PyMAF-X: Towards Well-Aligned Full-Body Model Regression From Monocular Images

Hongwen Zhang, Yating Tian, Yuxiang Zhang, Mengcheng Li, Liang An, Zhenan Sun, Senior Member, IEEE, and Yebin Liu, Member, IEEE

Abstract—We present PyMAF-X, a regression-based approach to recovering a parametric full-body model from a single image. This task is very challenging since minor parametric deviation may lead to noticeable misalignment between the estimated mesh and the input image. Moreover, when integrating part-specific estimations into the full-body model, existing solutions tend to either degrade the alignment or produce unnatural wrist poses. To address these issues, we propose a Pyramidal Mesh Alignment Feedback (PyMAF) loop in our regression network for well-aligned human mesh recovery and extend it as PyMAF-X for the recovery of expressive full-body models. The core idea of PyMAF is to leverage a feature pyramid and rectify the predicted parameters explicitly based on the mesh-image alignment status. Specifically, given the currently predicted parameters, mesh-aligned evidence will be extracted from finer-resolution features accordingly and fed back for parameter rectification. To enhance the alignment perception, an auxiliary dense supervision is employed to provide mesh-image correspondence guidance while spatial alignment attention is introduced to enable the awareness of the global contexts for our network. When extending PyMAF for full-body mesh recovery, an adaptive integration strategy is proposed in PyMAF-X to produce natural wrist poses while maintaining the well-aligned performance of the part-specific estimations. The efficacy of our approach is validated on several benchmark datasets for body, hand, face, and full-body mesh recovery, where PyMAF and PyMAF-X effectively improve the mesh-image alignment and achieve new

The project page with code and video results can be found at https://www.liuyebin.com/pymaf-x.

Index Terms—Expressive human mesh recovery, full-body motion capture, mesh alignment feedback, monocular 3D reconstruction.

Recent years have witnessed the rise of the regression-based paradigm in recovering body [1], [2], [3], [4], [5], [6], hand [7], [8], [9], [10], face [11], [12], [13], [14], and even full-body [15], [16], [17], [18] models from monocular images. These methods [1], [2], [3], [19] learn to predict model parameters directly from images in a data-driven manner. Despite the high efficiency and promising results, regression-based methods typically suffer from coarse alignment between the predicted meshes and image observations.

When recovering the parametric body or full-body models [20], [21], minor rotation errors accumulated along the kinematic chain may lead to noticeable drifts in joint positions (see the top-left example in Fig. 1), since joint poses are represented as relative rotations w.r.t. their parent joints. In order to generate well-aligned results, optimization-based methods [21], [22], [23] include data terms in the objective function so that the alignment between the projection of meshes and 2D evidence can be optimized explicitly. Similar strategies are also exploited in regression-based methods [1], [2], [3], [19] 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 [1] in their architectures. As shown in Fig. 2(a), IEF reuses the same global feature in its feedback loop, making the regressor hardly perceive the mesh-image misalignment in the inference phase.

As suggested in previous works [24], [25], [26], [27], neural networks tend to retain high-level information and discard detailed local features when reducing the spatial size of feature maps. In order 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 representation such as part segmentation [28] or dense correspondences [29], [30] in their regression networks. Though pixel-level evidence is considered, it is still challenging for those methods to learn structural priors and get hold of spatial details simultaneously based merely on high-resolution contexts.

Motivated by the above observation, we design a Pyramidal Mesh Alignment Feedback (PyMAF) loop in our regression
An auxiliary pixel-wise supervision and spatial alignment from three aspects. First, PyMAF is improved of the elbow poses is the rotation around the A feature pyramid is incorporated with the mesh alignment A mesh alignment feedback loop is proposed for PyMAF is further extended as PyMAF-X for full-body contexts. Since the SMPL family includes the hand and face meshes. We leverage three part-specific PyMAF networks as part experts to predict body, hand, and face parameters, and propose PyMAF-X for expressive full-body mesh recovery. Benefiting from the well-aligned results of each PyMAF-based expert, PyMAF-X can produce plausible full-body mesh results in common scenarios even using the most naive integration strategy [16]. However, as shown in Fig. 1, the naive “Copy-Paste” integration may lead to unnatural wrist poses under challenging cases. To address this issue, we propose an adaptive integration strategy to adjust the twist rotation of the elbow poses so that the elbow and wrist poses could be more compatible. In this way, the updated twist rotation of the elbow joint serves as compensation for the wrist joint and helps to produce natural wrist poses in the full-body model. Moreover, since the twist component [34] of the elbow poses is the rotation around the elbow-to-wrist bone, it barely changes the position of the body and hand joints, which is the key to maintaining the well-aligned performances of body and hand experts. Different from existing full-body solutions [15], [16], [17], [18], our method do not rely on additional networks to infer the wrist poses, and hence bypass the learning issue raised by insufficient full-body mesh annotations.

The contributions of this work can be summarized as follows:

- A mesh alignment feedback loop is proposed for regression-based human mesh recovery, where mesh-aligned evidence is exploited to correct parametric errors explicitly so that the estimated meshes can be better aligned with the input images.
- A feature pyramid is incorporated with the mesh alignment feedback loop so that the regression network can leverage multi-scale contexts. This yields the Pyramidal Mesh Alignment Feedback (PyMAF) loop, a novel architecture for human mesh recovery.
- An auxiliary pixel-wise supervision and spatial alignment attention are introduced in PyMAF to enhance the mesh-aligned features such that they can be more informative, relevant, and aware of the whole image contexts.
- PyMAF is further extended as PyMAF-X for full-body mesh recovery, where an adaptive integration strategy with the elbow-twist compensation is proposed to avoid unnatural wrist poses while maintaining the alignment of the body and hand estimations.

An early version of this work has been published as a conference paper [6]. We have made significant extensions to our previous work [6] from three aspects. First, PyMAF is improved to be more accurate with the newly introduced spatial alignment attention, which effectively enhances the feature learning and further improves the mesh-image alignment. Second, PyMAF goes beyond body mesh recovery and is extended to reconstruct hand and full-body models from monocular images. The well-aligned performance of the body- and hand-specific PyMAF makes it more promising to produce well-aligned full-body meshes. Third, an adaptive integration strategy is proposed to assemble predictions from body and hand experts. Such a strategy effectively addresses the unnatural wrist issues while maintaining the part-specific alignment. Based on these updates, our final method PyMAF-X achieves new state-of-the-art network to exploit multi-scale and position-sensitive contexts for better mesh-image alignment. The central idea of our approach is to correct parametric deviation explicitly and progressively based on the alignment status. In PyMAF, mesh-aligned evidence will be extracted from the spatial features according to the 2D projection of the estimated mesh and then fed back to the regressors for parameter updates. As illustrated in Fig. 2, the mesh alignment feedback loop takes advantage of more informative features for parameter correction compared with the commonly used iterative error feedback loop [1], [31]. In order to leverage multi-scale contexts, mesh-aligned evidence is extracted from a feature pyramid so that the coarse-aligned meshes can be corrected with large step sizes based on the lower-resolution features. To enhance these mesh-aligned features, an auxiliary task is imposed on the highest-resolution feature to infer pixel-wise dense correspondence, guiding the image encoder to preserve the most related information in the spatial feature maps. Meanwhile, a spatial alignment attention mechanism is introduced to fuse the grid and mesh-aligned features so that the regressor could be aware of the whole image contexts.

Since the SMPL family includes the hand [32] and face [33] models, PyMAF can be easily modified to reconstruct the hand and face meshes.
results both qualitatively and quantitatively, contributing novel solutions towards the well-aligned and natural recovery of full-body models from monocular images.

II. RELATED WORK

A. Monocular Human Mesh Recovery

Monocular recovery of human meshes has been actively studied in recent years. Aiming at the same goal of producing well-aligned and natural results, two different paradigms for human mesh recovery have been investigated in the research community. In this subsection, we give a brief review of these two paradigms and refer readers to [35] for a more comprehensive survey.

Optimization-based Approaches. Pioneering work in this field mainly focuses on the optimization process of fitting parametric models (e.g., SCAPE [36] and SMPL [20]) to 2D observations such as keypoints and silhouettes [22], [37], [38]. 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 evidence. Based on this paradigm, different updates have been investigated to incorporate information such as 2D/3D body joints [22], [39], silhouettes [23], [40], part segmentation [41], dense correspondences [42] 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. [43] 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 end-to-end learning and is also able to leverage rich 2D datasets thanks to the progress (e.g., SPIN [3], EFT [44], and NeuralAnnot [45]) in the generation of more precise pseudo 3D ground-truth for 2D datasets [46], [47], [48].

Regression-based Approaches. Alternatively, taking advantage of the powerful nonlinear mapping capability of neural networks, recent regression-based approaches [1], [3], [15], [19], [28], [49], [50], [51] have made significant advances in predicting human models directly from monocular images. These deep regressors take 2D evidence as input and learn model priors implicitly in a data-driven manner under different types of supervision signals [1], [52], [53], [54], [55], [56], [57], [58] 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 [19], [59], 2D/3D joints [4], [18], [19], [42], [49], [50], [52], [60], [61], segmentation [28], [62] and dense correspondences [29], [30]. Such strategies can benefit from synthetic data [29], [63] and the progress in the estimation of proxy representations [27], [64], [65], [66], [67]. In these regressors, though supervision signals are imposed on the re-projected models to penalize the mismatched predictions during training, their architectures can hardly perceive the misalignment during the inference phase. In comparison, the proposed PyMAF is a close-loop for both training and inference, which enables a feedback loop in our regressor to leverage spatial evidence for better mesh-image alignment of the estimated human models.

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. Among them, volumetric representation [59], [68], mesh vertices [2], [51], [69], and position maps [70], [71], [72], [73] 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, the mesh surfaces of non-parametric outputs tend to be rough and more sensitive to occlusions without additional structure priors. In our solution, the deep regressor uses spatial features at multiple scales for both high-level and fine-grained perception. It produces parametric models directly with no further processing required.

Recently, there are also numerous efforts devoted to achieving or handling multi-person recovery [61], [74], [75], [76], [77], [78], [79], video inputs [80], [81], [82], [83], [84], [85], occlusions [5], [30], [86], [87], more accurate shape [63], [88], [89], ambiguities [90], [91], camera estimation [92], [93], imbalanced data [94], [95], pseudo ground-truth generation [3], [44], [45], and clothed human reconstruction [96], [97], [98], [99]. Our work is complementary to them and focuses on the design of regressor architectures for single-image well-aligned body and full-body mesh recovery.

B. Full-Body Mesh Recovery

Compared with the large number of solutions for the body-only [1], [2], [3], [4], [30], hand-only [7], [9], [10], [100], [101], [102], [103], [104], [105], [106], [107], and face-only [11], [12], [13], [14], [108], [109] mesh recovery, the full-body mesh recovery receives less attention due to its challenging nature and the lack of annotated datasets. Similar to the developments of body-only mesh recovery algorithms, the research in the field of full-body mesh recovery begins with the proposal of full-body models, including Frank [110], Adam [110], SMPL-X [21], and GHUM [111], etc., and their corresponding optimization-based methods [21], [39], [110], [111], [112], [113]. Recently, several regression-based methods [15], [16], [17], [18], [114] have been proposed to overcome the slow and unnatural issues of optimization-based methods.

Following the pioneering work ExPose [15], regression-based methods [15], [16], [17], [18], [115], [116], [117] typically consist of three part-specific modules, namely part experts, to predict parameters of body, hand, and face from the corresponding part images cropped from original inputs. They differ mainly in the architecture of the part experts and the strategy to integrate part estimations. As the part experts are basically chosen from the body- or hand-only mesh recovery solutions, the integration strategy to sew up independent estimations becomes an essential aspect of a regression-based full-body method. The most straightforward strategy to integrate the body and hand estimations would be the “Copy-Paste” [15], [16]. To obtain
more natural integration results, learning-based strategies are proposed in recent state-of-the-art methods [16], [17], [115]. For instance, FrankMocap [16] learns to correct the arm poses based on the distance between the wrist positions predicted by body and hand experts. Zhou et al. [115] incorporate body features in the learning of the hand expert so that the predicted hand poses could be more compatible with the arm. PIXIE [17] introduces a learnable moderator to merge body and hand features for the regression of wrist and finger poses. All the above solutions rely on additional networks to predict or correct the wrist poses with the condition of body information, which is typically inferior to the original hand poses predicted by the hand expert, resulting in degraded alignment on the hand parts. Recently, Hand4Whole [18] proposes to learn wrist poses based on the positions of selected hand joints but does not consider the compatibility of arm poses. In contrast to existing solutions, PyMAF-X resorts to the adjustment of the twist components of wrist and elbow poses, which produces natural wrist rotations while maintaining the well-aligned performances of each part expert during the integration. Besides, our motivation and method also differ from the previous work [50], [118] that decomposes the twist components in the inverse kinematics problem.

C. Iterative Fitting in Regression Tasks

Strategies for incorporating fitting processes along with regression tasks have also been investigated in the literature. For human mesh recovery, Kolotouros et al. [3] combine an iterative fitting procedure with the training procedure to generate more accurate ground truths for better supervision. Several attempts have been made to deform human meshes so that they can be aligned with the intermediate estimates such as depth maps [119], part segmentation [116], and dense correspondences [42]. These approaches adopt intermediate estimations as fitting objectives and hence rely on their quality. In contrast, our approach uses the currently estimated meshes to extract deep features for refinement, enabling the fully end-to-end learning of the deep regressor.

In a broader view, remarkable efforts have been made to involve iterative fitting strategies in other computer vision tasks, including facial landmark localization [120], [121], [122], human/hand pose estimation [31], [123], etc. For generic objects, Pixel2Mesh [124] progressively deforms an initial ellipsoid by leveraging perceptual features. Following the spirit of these works, we exploit new strategies to extract fine-grained evidence and contribute novel solutions in the context of human mesh recovery.

III. Method

In this section, we will elaborate technical details of our approach. We first present PyMAF, a powerful model for regression-based human mesh recovery, then extend it to PyMAF-X for full-body mesh recovery.

A. PyMAF for Body-Only Mesh Recovery

As illustrated in Fig. 3, PyMAF consists of a feature pyramid for mesh recovery in a coarse-to-fine fashion. Coarse-aligned predictions will be improved by utilizing the mesh-aligned evidence extracted from finer-resolution features accordingly and fed back to a regressor for parameter rectification.
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 \( \phi^t(x) \in \mathbb{R}^{C_x \times 1} \) 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.,

\[
\phi_p^t = \mathcal{F}(\phi_s^t, X^t) = \oplus \{ f(\phi_s^t(x)), \text{for } x \in X^t \},
\]

(1)

where \( \mathcal{F}() \) 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 \( \Theta_t \) (the subscript \( t \) is omitted for simplicity) at each level, a mesh with vertices of \( M = M(\theta, \beta) \in \mathbb{R}^{N \times 3} \) can be generated accordingly, where \( N = 6890 \) denotes the number of vertices in the SMPL model. These mesh vertices are 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 keypoints \( \hat{K} = \Pi(J) \in \mathbb{R}^{N_j \times 2} \), 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 noticeable misalignment between the 2D projection and image evidence. To penalize such misalignment during the training of the regression network, we follow common practices [1], [3] to add 2D supervisions on the 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

\[
\mathcal{L}_{\text{reg}} = \lambda_2 d ||K - \hat{K}|| + \lambda_3 d ||J - \hat{J}|| + \lambda_{\text{para}} ||\Theta - \hat{\Theta}||, \tag{2}
\]

where \( d || \cdot || \) 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 straightforward strategy to extract point-wise features would be using grid-pattern points \( X_{\text{grid}} \) 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 current estimates when \( t > 0 \). We will show that such a mesh conditioned sampling strategy helps the regressor produce well-aligned reconstruction results.

2) Mesh Alignment Feedback Loop: As mentioned in HMR [1], 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 evidence can be leveraged in our regressor to rectify current parameters and improve the mesh-image alignment of the estimated model.

Mesh-aligned Features. In the proposed mesh alignment feedback loop, we extract mesh-aligned features from \( \phi_s^t \) based on the currently estimated mesh \( M_t \) when \( t > 0 \) to obtain more fine-grained and position-sensitive evidence. 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 set as the mesh-aligned points \( X_{\text{mesh}}^t \) which are obtained by first downsampling the mesh \( M_t \) to \( \hat{M}_t \) and then projecting it on the 2D image plane, i.e., \( X^t = X_{\text{mesh}}^t = \Pi(\hat{M}_t) \). Based on \( X_{\text{mesh}}^t \), the mesh-aligned features \( \phi_m^t \) will be extracted from \( \phi_s^t \) using (1), i.e.,

\[
\phi_m^t = \phi_p^t = \mathcal{F}(\phi_s^t, \Pi(\hat{M}_t)). \tag{3}
\]

Spatial Alignment Attention. Though the mesh-aligned features \( \phi_m^t \) are position-sensitive, these features are confined to the re-projection regions of the current mesh result. To enable the perception of the relative positions in the whole image context, we further design spatial alignment attention to fuse the information from both grid and mesh-aligned features. Considering that both these two features are extracted from the same spatial feature map, we adopt a self-attention module to process them. Specifically, the point-wise features extracted based on the grid-pattern points \( X_{\text{grid}} \) and the mesh-aligned points \( X_{\text{mesh}}^t \) are first concatenated together as \( \phi_{gm}^t \)

\[
\phi_{gm}^t = \{ \phi_s^t(x), \text{for } x \in (X_{\text{grid}} \cup X_{\text{mesh}}^t) \} \in \mathbb{R}^{N_{gm} \times C_g}, \tag{4}
\]

where \( N_{gm} \) is the total number of the grid-pattern and mesh-aligned points. Then, spatial alignment attention is applied to learn attentive relations among \( \phi_{gm}^t \) so that the mesh-aligned features can be more effectively enhanced with the spatial information in the grid features. In our solution, a self-attention module [125] is employed to process the features \( \phi_{gm}^t \)

\[
Q, K, V = \phi_{gm}^t W^Q, \phi_{gm}^t W^K, \phi_{gm}^t W^V, \phi_{gm}^t = \text{Att}(Q, K) V, \tag{5}
\]

where \( W^Q, W^K, \) and \( W^V \) are the learnable matrices used to generate different subspace representations of the query, key, and value features \( Q, K, V \), respectively. \( \text{Att}(\cdot) \) denotes the scaled dot-product attention function [125] with softmax. In this way, the messages of the grid and mesh-aligned features can be fully fused together since the self-attention mechanism captures the relationships between all elements of the features \( \phi_{gm}^t \). After that, the enhanced mesh-aligned features \( \tilde{\phi}_{gm}^t \) are obtained by reducing the dimension of \( \phi_{gm}^t \) and concatenating them together. Finally, the enhanced mesh-aligned features \( \tilde{\phi}_{gm}^t \) are fed into the
supervision is written as
\[ L_{aux} = \lambda_{\theta fb} \text{CrossEntropy}(P, \hat{P}) + \lambda_{uv} \text{SmoothL1}(\hat{P} \circ U, \hat{P} \circ \hat{U}) + \lambda_{uv} \text{SmoothL1}(\hat{P} \circ V, \hat{P} \circ \hat{V}), \]

where \( \circ \) denotes the mask operation. Note that the auxiliary prediction is required in the training phase only.

Fig. 4 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 input variations after applying auxiliary supervision. Note that the dense correspondence is not limited to the IUV representation, the Projected Normalized Coordinate Code (PNCC) [126] can be also adopted as dense correspondences when IUV is not defined in the mesh model. More discussions about the choice of dense correspondences can be found in the Supplementary Material, which can be found on the Computer Society Digital Library at http://ieeecomputersociety.org/10.1109/TPAMI.2023.3271691.

B. PyMAF-X for Full-Body Mesh Recovery

The body-specific PyMAF can be easily modified to reconstruct hand and face meshes by simply changing the SMPL model in the above formulation to the MANO [32] and FLAME [33] models. Based on the regression power of PyMAF, we extend it to PyMAF-X for full-body mesh recovery.

Following previous works [15], [16], [17], [18], PyMAF-X consists of three experts, i.e., three part-specific PyMAFs, to predict the parameters of body, hand, and face, as illustrated in Fig. 5. To ensure high-resolution observations of part regions, part experts perform individual predictions on the body, hand, and face images cropped from the original inputs. At each iteration of the mesh alignment feedback loop, the predictions of the body-, hand-, and face-specific PyMAF are collected and integrated as the parameters \( \Theta_{fb} = \{\theta_{fb}, \beta_{fb}, \psi, \pi\} \) of the full-body model SMPL-X [21], where \( \theta_{fb}, \beta_{fb}, \psi \) and \( \pi \) denotes the pose, shape, and facial expression parameters, respectively. The pose parameters \( \theta_{fb} \) consist of the rotational poses of 55 joints in total, including 22 joints for the body, 30 finger joints for the hands, and 3 jaw joints for the face. The camera parameters \( \pi \) are taken from the predictions of the body-specific PyMAF and used to project body, hand, and face vertices on the image plane. Moreover, considering that the positions of hand and face are susceptible to inaccurate body pose estimations, we align the center of their re-projected points to the image center of hand and face to ensure their mesh-aligned features are meaningful.

Naïve Integration. After individual regression of each part, we need to figure out the rotation of wrist joints to integrate the body and hand meshes. The most straightforward strategy would be the naïve “Copy-Paste” integration [16]. Specifically, the poses of the wrist joints are calculated based on the body poses predicted by the body expert and the global orientation of hands predicted by the hand expert. Let \( \theta_{hand} \) be the global orientation of the left or right hand, which is also the global
rotation of the wrist joint. The wrist pose of the full-body model can be solved by first computing the global rotation \( \theta_{\text{elbow}} \) of the elbow joint and then the relative rotation \( \theta_{\text{wrist}} \) of the wrist joint, i.e.,

\[
\hat{\theta}_{\text{elbow}} = \prod_{j \in A(\text{elbow})} \theta_j,
\]

\[
\theta_{\text{wrist}} = \hat{\theta}_{\text{hand}}^{-1} \hat{\theta}_{\text{elbow}},
\]

where \( \theta_j \) denotes the relative rotation of the \( j \)-th body joint, \( A(\text{elbow}) \) the ordered set of joint ancestors of the elbow joint and itself in the kinematic tree, and \( \hat{\theta}_{\text{hand}}^{-1} \) the inverse global rotation of the hand. Benefiting from the well-aligned results of each part, PyMAF-X can produce plausible results in common scenarios using such a simple integration strategy.

Adaptive Integration with Elbow-Twist Compensation. As pointed out in previous work [15], the body expert hardly perceives the hand poses due to the small proportion of hand region in the body images. It may lead to incompatible configurations of the arm and hand poses predicted individually by the body and hand experts, resulting in unnatural wrist poses of the full-body model, as illustrated in Fig. 6. Previous work [15], [16], [17], [18] alleviates this issue by learning wrist poses from the body and hand features but typically degrades the accuracy of the wrist poses and alignment. In our work, we propose an adaptive integration strategy to correct the elbow poses directly based on the solved wrist poses such that the elbow and wrist poses could be more compatible. To maintain the mesh-image alignment, we only correct the twist rotation of the elbow joints as it is the rotation along the elbow-to-wrist bone and barely affects the position of the body and hand joints. To this end, we first compute the twist angle of the wrist poses w.r.t. the elbow-to-wrist bone, then update the elbow and wrist poses by adding and subtracting the compensated twist rotation, respectively.

Step 1: Computing the original twist angle. The twist component around the elbow-to-wrist vector can be decomposed from the wrist poses. Without loss of generality, let the quaternion representation of the left or right wrist pose solved in (8) be \( q_{\text{wrist}} = (w_{\text{wrist}}, \bar{v}_{\text{wrist}}) \). By using Huyhge’s method [127], [128], the quaternion \( q_{\text{tw}} \) of the twist rotation around the normalized elbow-to-wrist vector \( \bar{v}_{\text{tw}} \) can be calculated as

\[
u_{\text{proj}} = \frac{\bar{v}_{\text{wrist}} \cdot \bar{v}_{\text{tw}}}{\|\bar{v}_{\text{wrist}}\|},
\]

\[
q_{\text{proj}} = (w_{\text{wrist}}, u_{\text{proj}} \bar{v}_{\text{tw}}),
\]

\[
q_{\text{tw}} = \frac{q_{\text{proj}}}{\|q_{\text{proj}}\|},
\]

where \( u_{\text{proj}} \bar{v}_{\text{wrist}} \) in \( q_{\text{proj}} \) is the projection vector of the normalized \( \bar{v}_{\text{wrist}} \) onto \( \bar{v}_{\text{tw}} \). Let \( w_{\text{tw}} \) be the first element of the twist quaternion \( q_{\text{tw}} \), then the twist rotation angle can be computed as \( \alpha_{\text{tw}} = 2\cos^{-1}(w_{\text{tw}}) \in [-\pi, \pi] \).

Step 2: Updating elbow and wrist poses. The angle \( \alpha_{\text{tw}} \) reflects the intensity of the wrist rotation around the elbow-to-wrist bone, and an out-of-range twist angle typically leads to unnatural
wrist poses. To tackle this issue, an additional twist rotation is added to the elbow pose and serves as a compensation to the wrist pose. Specifically, the elbow/wrist poses are updated as \( \theta_{\text{elbow}}(\theta_{\text{wrist}}) \) by adding/subtracting a twist rotation \( \theta_{\text{cp}} \) around the elbow-to-wrist vector \( \vec{v}_{tw} \) with a compensation angle of \( \alpha_{\text{cp}} \), i.e., \( \theta_{\text{elbow}} = \theta_{\text{elbow}}(\theta_{\text{wrist}} + \theta_{\text{cp}}) \) and \( \theta_{\text{wrist}} = \theta_{\text{wrist}}(\theta_{\text{elbow}}(\theta_{\text{cp}})) \). In our solution, we empirically set a range \( [\alpha_{\text{tmin}}, \alpha_{\text{tmax}}] \) to constraint \( \alpha_{\text{tw}} \) and adopt the compensation angle \( \alpha_{\text{cp}} \) as

\[
\alpha_{\text{cp}} = \begin{cases} 
\alpha_{\text{tw}} - \alpha_{\text{tmax}}, \\
\min(\alpha_{\text{tw}} - \alpha_{\text{tmin}}, 0), 
\end{cases}
\]

As shown in our experiments, with the twist compensation from the elbow joint, the wrist pose becomes more natural while maintaining the mesh-image alignment of the body and hands. In practice, the adaptive integration is not applied for those invisible hands since the global orientation predicted by the hand expert is not reliable when the hand is invisible. In our implementation, the hand expert of PyMAF-X also predicts the confidence of the visibility status of hands. When the hand is invisible, the full-body model simply adopts the wrist poses predicted by the body expert and the mean poses of hands.

IV. EXPERIMENTS

A. Implementation Details

The part-specific PyMAF primarily adopts ResNet-50 [129] as the backbone of the image encoder. We also follow Ex-Pose [15] and PIXIE [17] to adopt HRNet-W48 [27] as the encoder backbone for the body model regression. For each part-specific PyMAF, the image encoder takes a \( 224 \times 224 \) image as input and produces spatial feature maps with resolutions of \{14 \times 14, 28 \times 28, 56 \times 56\}. When generating mesh-aligned features, the vertex number of body, hand, and face meshes is down-sampled to 431, 195, and 708, respectively. The mesh-aligned features extracted from feature maps of each point will be processed by MLPs so that their channel dimensions will be reduced to 5. Hence, the mesh-aligned feature vector for the body model has a length of 2155 = 431 \times 5, which is similar to the length of the global features used in HMR [1]. The maximum number \( T \) is set to 3, which is equal to the iteration number used in HMR. For the grid features used at \( t = 0 \), they are uniformly sampled from \( \phi^0 \) 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 downsampling. The regressors \( R_j \) have the same architecture as the one in HMR, except that they have slightly different input dimensions. The twist angle constraint \( [\alpha_{\text{tmin}}, \alpha_{\text{tmax}}] \) is empirically set to \([-72^\circ, 72^\circ]\) in our implementation. Following previous work [3], [5], we adopt the continuous 6D representation [130] for pose parameters in the regressor. Following PARE [5], the body encoder is initialized with the model pretrained on 2D pose datasets [46], [48]. During training, we use the Adam [131] optimizer and set the learning rate to \( 5e^{-5} \) without decay. The part-specific PyMAFs are first pre-trained individually and then assembled together for finetuning on full-body datasets. Similar to PARE [5], we also observed a slight performance gain when removing the auxiliary supervision at the final stage of training, but we do not apply such a strategy in our experiments for more consistent ablation studies of our newly introduced components. More details of the implementation can be found in our code and the Supplementary Material, available in the online supplemental material.

Camera Setting. We follow previous work [3] to use a weak perspective camera with a pre-defined focal length of 5,000 by default for training and evaluation. When running experiments on AGORA [132], we use a perspective camera with the focal lengths estimated by SPEC [93] as there are stronger perspective distortions in this dataset. Incorporating the camera setting of SPEC [93] with our method for more accurate mesh recovery is left for future work.

Runtime. The PyTorch implementation of the body-only PyMAF takes about 22 ms to process one sample on the machine with an NVIDIA RTX 3090 GPU. For full-body mesh recovery, PyMAF-X takes about 80 ms to process one sample, which is on par with existing regression-based approaches [15], [17], [18]. In our current implementation, the part-specific backbone networks run in sequence to process the body, hand, and face images. Running them in parallel would further reduce the runtime.

B. Datasets

Following the practices of previous work [1], [3], [5], the body expert is trained on a mixture of data from several datasets with 3D and 2D annotations, including Human3.6M [134], MPI-INF-3DHP [139], MPII [46], LSP [47], LSP-Extended [140], and COCO [48]. For the hand expert, we use images from FreiHAND [8], InterHand2.6M [9] and COCO-Wholebody [141] for training. For the face expert, we use the images from VGGFace2 [142] for training. Detailed descriptions of the datasets can be found in the Supplementary Material, available in the online supplemental material.

Pseudo Ground-truth. Following previous work [5], the SMPL/SMPL-X models fitted in EFT [44] and ExPose [15] are used as pseudo ground-truth annotations for the training of body and full-body model regressors. For the training of the face expert, we use DECA [14] and a face alignment algorithm FAN [143] to generate pseudo ground-truth FLAME models and facial landmarks on the training set of VGGFace2 [142].

Dense Correspondence. Note that we do not use the DensePose annotations in COCO for auxiliary supervision but render dense correspondence maps based on the pseudo ground-truth meshes using the method described in [30].

C. Evaluation Metrics

We report the results of our approach in various evaluation metrics for quantitative comparisons with existing state-of-the-art methods, where all metrics are computed in the same way as previous work [1], [3], [5], [15], [17], [18] in literature.

To quantitatively evaluate the performance of the 3D pose estimation, PVE, MPJPE, PA-PVE, and PA-MPJPE are adopted as the primary evaluation metrics. They are all reported in millimeters (mm) by default. Among these metrics, PVE denotes the mean Per-vertex Error, defined as the average point-to-point
TABLE I

| Backbone Architectures are Highlighted in the Brackets |

| Method                  | 3DPW PVE | MPJPE | PA-MPJPE | MPJPE | PA-MPJPE |
|-------------------------|---------|-------|----------|-------|----------|
| Kanazawa et al. [135]   | 119.3   | 72.6  |          |       |          |
| Doersch et al. [55]     | -       | 74.7  |          |       |          |
| Arman et al. [136]      | -       | 72.2  |          |       |          |
| VIBE [80]               | 113.4   | 56.5  | 65.9     | 41.5  |          |
| MEVA [138]              | -       | 54.7  |          |       |          |
| TCMR [81]               | 111.5   | 50.8  | 62.3     | 41.1  |          |

Euclidean distance between the predicted and ground truth mesh vertices, while MPJPE denotes the Mean Per Joint Position Error. PA-PVE and PA-MPJPE denote the PVE and MPJPE after rigid alignment of the prediction with the ground truth using Procrustes Analysis. Note that the metrics PA-PVE and PA-MPJPE are not aware of the global rotation and scale errors since they are calculated after rigid alignment.

D. Comparison With the State of the Art

1) Evaluation on Body-Only Mesh Recovery: 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 [133] and Human3.6M [134] datasets in Table I. Our PyMAF achieves competitive or superior results among previous approaches, including frame-based and temporal approaches. Note that the approaches reported in Table I are not strictly comparable since they may use different training data, pseudo ground-truths, learning rate schedules, training epochs, etc. For a fair comparison, we report our baseline results in Table I, which is trained under the same setting as PyMAF. The baseline approach has the same network architecture with HMR [1] and also adopts the 6D rotation representation [130] for pose parameters. Under the setting of using ResNet-50 backbone and without training on 3DPW, PyMAF reduces the MPJPE over the baseline by 4.7 mm and 5.5 mm on 3DPW and Human3.6M datasets, respectively.

From Table I, we can see that PyMAF has more notable improvements on the metrics MPJPE and PVE. We would argue that the metric PA-MPJPE cannot reveal the mesh-image alignment performance since it is calculated as the MPJPE after rigid alignment. As depicted in the Supplementary Material, available in the online supplemental material, a reconstruction result with a smaller PA-MPJPE value can have a larger MPJPE and worse alignment between the reprojected mesh and the input image.

2) Keypoint Localization APs on the COCO [48] Validation Set. Results of SMPLify [22] are evaluated based on the implementation in SPIN [3], results of HMR [1], GRAPHCMR [2], and SPIN [3] are evaluated based on their publicly released code and models.

The bold entries indicate the best result of the comparison methods.
TABLE III
Hand Reconstruction Errors on the FreiHAND [8] Dataset. † denotes the methods using extra training data more than FreiHAND

| Method          | PA-PVE ↓ | PA-MPJPE ↓ | F-Scores † @ 5 mm / 10 mm |
|-----------------|----------|------------|--------------------------|
| * Hand-only methods |          |            |                          |
| FreiHAND [8]    | 10.7     | -          | 0.529 / 0.935            |
| Pose2Mesh [49]  | 7.8      | 7.7        | 0.674 / 0.969            |
| I2L-MeshNet [4] | 7.6      | 7.4        | 0.681 / 0.973            |
| METRO [51]      | 6.7      | 6.8        | 0.717 / 0.981            |
| * Full-body methods |        |            |                          |
| ExPose [15]     | 11.8     | 12.2       | 0.464 / 0.918            |
| Zhou et al. [115]| -        | 15.7       | -/-                      |
| FrankMocap [16] | 11.6     | 9.2        | 0.553 / 0.951            |
| PIXIE [17] †    | 12.1     | 12.0       | 0.468 / 0.919            |
| Hand4Whole [18] †| 7.7      | 7.7        | 0.664 / 0.971            |
| ‘Baseline’      | 8.6      | 8.9        | 0.608 / 0.965            |
| PyMAF           | 8.1      | 8.4        | 0.638 / 0.969            |
| PyMAF †         | 7.5      | 7.7        | 0.671 / 0.974            |

The bold entities indicate the best result of the comparison methods.

TABLE IV
Face Reconstruction Errors on Stirling3D [144] and NoW [13] Datasets

| Method          | PA-P2S (mm) ↓ |
|-----------------|---------------|
| * Stirling3D LO/HQ |               |
| RingNet [13]     | 1.63/1.58     |
| ExPose [15]      | 1.27/1.91     |
| ‘Baseline’       | 1.55/1.57     |
| PyMAF            | 1.51/1.48     |
| PyMAF †          | 1.47/1.24     |
| * NoW            |               |
| DECA [14]        | 1.09          |
| ExPose [15]      | 1.26          |
| PIXIE [17]       | 1.18          |
| ‘Baseline’       | 1.17          |
| PyMAF            | 1.13          |
| PyMAF †          | 1.42          |

The bold entities indicate the best result of the comparison methods.

be found in the supplementary material, available in the online supplemental material.

2) Evaluation on Hand-Only Reconstruction: We compare the hand-only PyMAF with state-of-the-art approaches on the FreiHAND [8] dataset. As shown in Table III, PyMAF outperforms the baseline and previous full-body methods and is comparable with recent hand-only methods [4, 51]. It is also worth noting that full-body methods typically adopt the parametric representation of the hand mesh, which tends to be numerically inferior to the non-parametric representation used in recent hand-only methods [4, 51], as pointed out in previous works [4, 35].

3) Evaluation on Face-Only Reconstruction: Following previous work [15, 17], we compare the face-only PyMAF with state-of-the-art face reconstruction approaches on the test set of Stirling3D [144] and NoW [13] datasets. Table IV reports the performances of different methods in Point-to-Surface after Procrustes Alignment (PA-P2S). It shows that the PyMAF outperforms the face expert of previous full-body methods ExPose [15] and PIXIE [17], while achieving similar results compared with the strong face-only method DECA [14]. Qualitative comparisons of face reconstruction results are visualized in Fig. 7. For more consistent comparisons, we only show the intermediate parametric FLAME model predicted by DECA [14] without using detail displacements. We can see that PyMAF is able to capture expressive face shapes and has competitive results against DECA [14].

4) Evaluation on Full-Body Mesh Recovery: Following previous work [15, 16, 17, 18] on full-body mesh recovery, we evaluate the performance of PyMAF-X on two benchmark datasets, i.e., EHF [21] and AGORA [132].

Table V reports the results of different methods for full-body mesh recovery, including the optimization-based MTC [112] and SMPLify-X [21], and the regression-based ExPose [15], FrankMocap [16], Zhou et al. [115], PIXIE [17], and Hand4Whole [18]. We can see that PyMAF-X achieves the best results among existing solutions on most metrics, especially on the evaluation of the body and full-body reconstruction.

Table VI compares the results of PyMAF-X and other full-body methods on the test set of AGORA [132], where all the evaluation results are taken from the official evaluation platform. Recent state-of-the-art approaches to body-only mesh recovery are also included in Table VI for comprehensive comparisons. Note that the evaluation on AGORA is also affected by the detection results as the predictions are first matched with the ground truth and then used to calculate the reconstruction error. We use an off-the-shelf tool OpenPifPaf [145] to detect persons and the corresponding hands and face regions, of which the person detection result is slightly worse than the recent solutions Hand4Whole [18] and BEV [78]. For matched predictions, PyMAF-X outperforms other methods, especially in the metrics for hand and full-body reconstruction on this challenging dataset.

Qualitative comparisons of different full-body methods on real-world images are shown in Fig. 8, where we can see that PyMAF-X produces more accurate body, hand, and wrist poses than recent state-of-the-art approaches, including FrankMocap [16], PIXIE [17], and Hand4Whole [18]. The video results of PyMAF-X and other full-body methods can be found on our project page and supplementary materials, available in the online supplemental material.

E. Ablation Study

In this part, we will perform ablation studies under various settings to validate the key components proposed in
TABLE V

| Method            | PVE       | PA-PVE    | PA-MPJPE |
|-------------------|-----------|-----------|----------|
|                   | Full-body | Hands     | Body     | Hands     | Body     | Hands    |
| MTC [112]†        | -         | -         | 67.2     | -         | -        | 107.8    |
| SMPlyf-X [21]†    | -         | -         | 65.3     | 73.4      | 12.3     | 6.3      |
| ExPose [15]       | 77.1      | 51.6      | 35.0     | 54.5      | 52.6     | 12.8     |
| FrankMocap [16]   | 107.6     | 42.8      | -        | 57.5      | 52.7     | 12.5     | 62.3     |
| Zhou et al. [113] | 90.8      | 51.7      | 28.1     | 66.2      | 60.3     | 14.6     | 7.0      |
| PIXIE [17]        | 89.2      | 42.8      | 32.7     | 55.0      | 53.0     | 11.1     | 4.6      |
| Hand4Whole [18]   | 76.8      | 39.8      | 26.1     | 50.3      | -        | 10.8     | 60.4     |

The bold entities indicate the best result of the comparison methods.

TABLE VI

| Method            | Detection | MVE        | MPJPE     |
|-------------------|-----------|------------|-----------|
|                   | F1 Score  | Full-Body  |           |           |
|                   |           | Body       | Face      | L/R-Hand  |           |           |
|                   |           | Body       | Face      | L/R-Hand  |           |           |
| SPIN [3]†         | 0.77      | 148.9      | -         | -         | -         | -         |
| PARE [5]†         | 0.84      | 140.9      | -         | -         | -         | -         |
| SPEC [93]†        | 0.84      | 106.5      | -         | -         | -         | -         |
| ROMP [77]†        | 0.91      | 103.4      | -         | -         | -         | -         |
| BEV [78]†         | 0.93      | 100.7      | -         | -         | -         | -         |
| SMPlyf-X [21]†    | 0.71      | 236.5      | 187.0     | 48.9      | 48.3      | 51.4      | 231.8     |
| ExPose [15]       | 0.82      | 217.3      | 151.5     | 51.1      | 74.9      | 71.3      | 215.9     |
| FrankMocap [16]   | 0.80      | -          | 168.3     | -         | 54.7/55.7 | -         | 165.2     |
| PIXIE [17]        | 0.82      | 191.8      | 142.2     | 50.2      | 49.5/49.0 | -         | 188.9     |
| Hand4Whole [18]†  | 0.94      | 135.5      | 90.2      | 41.6      | 46.3/48.1 | -         | 132.6     |
| PyMAYF-X (Res50)† | 0.89      | 134.4      | 90.4      | 38.7      | 45.9/47.3 | -         | 132.8     |
| PyMAYF-X (HR48)†  | 0.89      | 125.7      | 84.0      | 35.0      | 44.6/45.6 | -         | 124.6     |

The bold entities indicate the best result of the comparison methods.

PyMAYF and PyMAYF-X. The efficacy of the mesh-aligned features, pyramidal design, auxiliary dense supervision, and spatial alignment attention proposed in PyMAYF will be validated on Human3.6M [134]. As the Human3.6 M dataset includes large-scale amounts of images and the corresponding ground-truth 3D labels, ablation approaches of PyMAYF are trained and evaluated on Human3.6 M. As for the proposed adaptive integration in PyMAYF-X, ablation approaches are evaluated on EHF [21], where different approaches are trained under the same setting.

Efficacy of Mesh-aligned Features. In PyMAYF, 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 the mesh-aligned features with the global features or the grid features uniformly sampled from spatial features as the input for parameter regressors. Table VII reports the performance of approaches using different types of features in the feedback loop. The results under the non-pyramidal setting are also included in Table VII, where the grid and mesh-aligned features are extracted from the feature maps with the highest resolution (i.e., 56 × 56), and the mesh-aligned features are extracted on the reprojected points of the mesh under the mean pose at t = 0. Note that all approaches in Table VII do not use 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 global features but is worse than the mesh-aligned counterpart. The mesh-aligned solution achieves even more performance gain when using pyramidal feature maps since multi-scale mesh-alignment evidence is leveraged in the feedback loop. Though the grid features contain primary spatial cues on uniformly distributed pixel positions, they cannot reflect the alignment status of the current estimation. This implies that mesh-aligned features are the most informative ones for the regressor to rectify the current mesh parameters.

TABLE VII

| Feedback Feature | Mesh-aligned? | MPJPE | PA-MPJPE |
|------------------|--------------|-------|----------|
| Global (Baseline)| No           | 84.1  | 55.6     |
| Grid             | 80.5         | 54.7  |          |
| Mesh-aligned     | 79.6         | 53.4  |          |
| Grid             | Yes          | 79.7  | 54.3     |
| Mesh-aligned     | 76.8         | 50.9  |          |

The bold entities indicate the best result of the comparison methods.

[1] Online. Available: https://agora-evaluation.is.tuebingen.mpg.de
Benefit from Auxiliary Supervision. The auxiliary pixel-wise supervision helps to enhance the reliability of the mesh-aligned evidence extracted from spatial features. Using alternative pixel-wise supervision such as part segmentation rather than dense correspondences is also possible in our framework. In our approach, these auxiliary predictions are solely needed for supervision 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 VIII compares 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 clean and rich information in foreground regions. Moreover, using feature maps for point-wise feature extraction consistently performs better than 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 VIII, 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.

Efficacy of Spatial Alignment Attention. In our approach, Spatial Alignment Attention (SAA) is designed to enable the awareness of the whole image context in the regressor. To validate its efficacy, we replace the spatial alignment attention with fully-connected layers to fuse the grid and mesh-aligned features. As reported in Table IX, simply fusing the grid features (the second row) only brings marginal improvements in comparison with the approach using the spatial alignment attention (the third row). The performances of PyMAF with or without spatial alignment attention across each refinement iteration are reported in Table X, where the PyMAF with spatial alignment attention improves the reconstruction results more quickly.

| 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. | MPJPE | PA-MPJPE |
|-------------|-------------|-------|----------|
| Feature     | 77.8        | 54.7  |          |
|            | 75.1        | 48.9  |          |

The bold entities indicate the best result of the comparison methods.

| Feedback Feat | PVE | MPJPE | PA-MPJPE |
|---------------|-----|-------|----------|
| Mesh-aligned  | 89.1| 76.8  | 50.9     |
| + Grid        | 88.9| 76.6  | 51.0     |
| + SAA         | 85.1| 73.6  | 50.1     |

The bold entities indicate the best result of the comparison methods.
In this paper, we first present Pyramidal Mesh Alignment Feedback (PyMAF) for regression-based human mesh recovery and further extend it as PyMAF-X for full-body mesh recovery. PyMAF is primarily motivated by the observation of the reprojection misalignment between the parametric mesh results and the input images. At the core of PyMAF, the parameter regressor leverages spatial information from a feature pyramid to correct the parameter deviation explicitly in a feedback loop based on the alignment status of the currently estimated meshes. To achieve this, given a coarse-aligned mesh estimation, the mesh-aligned features are first extracted from the spatial feature maps and then fed back into the regressor for parameter rectification. Moreover, an auxiliary dense supervision is employed to enhance the learning of mesh-aligned features while spatial alignment attention is introduced to enable the awareness of the global contexts in our deep regressor. When extending PyMAF for full-body model recovery, an adaptive integration with the elbow-twist compensation strategy is proposed in PyMAF-X to produce natural wrist poses while maintaining the alignment performances of part-specific PyMAF. The efficacy of PyMAF and PyMAF-X is validated on indoor and in-the-wild datasets, where our approaches effectively improve the mesh-image alignment over the baseline and previous regression-based solutions.

**Efficacy of Adaptive Integration.** In PyMAF-X, an elbow-twist compensation is used to adaptively correct the elbow poses in the integration of body and hand estimations. Such an adaptive integration strategy could produce physically-plausible wrist poses while preserving the mesh-image alignment. We investigate different integration strategies and compare our solution with two alternatives: i) a learned integration strategy similar to PIXIE [17], which predicts the wrist poses based on the fused features of body and hand features; ii) the naive copy-paste integration strategy [16], which calculates the wrist poses based on the estimated body and hand poses. Table XI reports the performances of the three different integration strategies on the EHF dataset. Here, we use the MPJPE of body and hand joints to measure the mesh-image alignment and the PA-PVE of wrist vertices to measure the physical plausibility of the wrist joint. As shown in the first row, the learned integration strategy can also produce natural wrist poses but degrade the alignment of hand parts in the full-body model. Compared with the learned and copy-paste strategies, the proposed adaptive integration produces both well-aligned and natural poses of the body, hand, and wrist parts. Fig. 9 provides a visual comparison of different integration strategies under challenging cases in real-world scenarios. We can see that our adaptive integration maintains the alignment and effectively improves the plausibility of the wrist poses by leveraging the twist compensation from elbow joints.

**Limitations and Future Work.** In our experiments, we found that PyMAF-X fails to reconstruct interacting hands due to the separated regression of two hands. Meanwhile, when handling images with strong perspective distortions or with only upper-torso observations, common issues such as bent legs and erroneous limb poses remain unsolved in this work. We will leave these issues for future work to incorporate the merits of recent datasets [146], [147] and solutions such as SPEC [93], PIXIE [17], Hand4Whole [18], and IntagHand [10] into our framework.

Moreover, similar to existing methods [15], [16], [17], [18], the full-body alignment performance of PyMAF-X heavily relies on the pose and shape estimation of the body expert. Due to the lack of full-body mesh annotations, the estimated body shapes are typically inaccurate in challenging cases, resulting in erroneous bone lengths of arms and coarse alignment of hands. Combining PyMAF-X with SPIN [3], EFT [44], or NeuralAnnot [45] for the generation of more precise pseudo 3D ground-truth full-body mesh annotations on in-the-wild data would be interesting future work. Besides, the elbow-twist

---

**TABLE X**

| Method         | Metric   | $M_0$ | $M_1$ | $M_2$ | $M_3$ |
|----------------|----------|-------|-------|-------|-------|
| PyMAF w/o SAA  | PVE      | 312.1 | 97.0  | 90.5  | 88.7  |
|                | MPJPE    | 274.0 | 80.3  | 76.6  | 75.1  |
|                | PA-MPJPE | 131.7 | 52.1  | 49.9  | 48.9  |
| PyMAF w. SAA   | PVE      | 312.1 | 91.2  | 83.6  | 81.8  |
|                | MPJPE    | 274.0 | 78.2  | 73.2  | 72.1  |
|                | PA-MPJPE | 131.7 | 51.6  | 49.8  | 48.7  |

The bold entities indicate the best result of the comparison methods.

**TABLE XI**

| Integration      | (Alignment) Body | Hands | (Plausibility) Wrist |
|------------------|------------------|-------|---------------------|
| Learned          | 64.9             | 42.3  | 5.8                 |
| Copy-Paste       | 64.9             | 31.2  | 6.7                 |
| Adaptive (Ours)  | 64.9             | 31.2  | 5.9                 |

The bold entities indicate the best result of the comparison methods.
rotations are adjusted empirically in PyMAF-X based on the twist components of wrist poses. Learning the compensation angle $\alpha_{op}$ via networks is also possible when large-scale full-body mesh annotations are available.

ACKNOWLEDGMENTS

We would like to thank Xinchi Zhou, Wanli Ouyang, and Limin Wang for their help, feedback, and discussions in the early work of this paper.

REFERENCES

[1] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, “End-to-end recovery of human shape and pose,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 7122–7131.
[2] N. Kolotouros, G. Pavlakos, and K. Daniilidis, “Convolutional mesh regression for single-image human shape reconstruction,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4501–4510.
[3] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, “Learning to reconstruct 3D human pose and shape via model-fitting in the loop,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 2252–2261.
[4] G. Moon and K. M. Lee, “I2L-MeshNet: Image-to-ixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image,” in Proc. Eur. Conf. Comput. Vis., Springer, 2020, pp. 752–768.
[5] M. Kocabas, C.-H. P. Huang, O. Hilliges, and M. J. Black, “PARE: Part attention regressor for 3D human body estimation,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 11127–11137.
[6] H. Zhang et al., “PyMAF: 3D human pose and shape regression with pyramidal mesh alignment feedback loop,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 11446–11456.
[7] X. Zhang, Q. Li, H. Mo, W. Zhang, and W. Zheng, “End-to-end hand mesh recovery from a monocular RGB image,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 2354–2364.
[8] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox, “FreiHAND: A dataset for markerless capture of hand pose and shape from single RGB images,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 813–822.
[9] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee, “InterHand2.6M: A dataset and baseline for 3D interacting hand pose estimation from a single RGB image,” in Proc. Eur. Conf. Comput. Vis., Springer, 2020, pp. 548–564.
[10] M. Li et al., “Interacting attention graph for single image two-hand reconstruction,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 2761–2770.
[11] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, “Joint 3D face reconstruction and dense alignment with position map regression network,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 557–574.
[12] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos, “Large pose 3D face reconstruction from a single image via direct volumetric CNN regression,” in Proc. IEEE Int. Conf. Comput. Vis. Workshop, 2021, pp. 1749–1759.
[13] S. Sanyal, T. Bolkart, H. Feng, and M. J. Black, “Learning to regress 3D hand shape and expression from an image without 3D supervision,” in Proc. Int. Conf. 3D Vis., 2019, pp. 10884–10894.

Gavril P. Pavlovsk, L. Zhu, X. Zhou, and K. Daniilidis, “Learning to estimate 3D human pose and shape from a single color image,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 459–468.
M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, “SMPL: A skinned multi-person linear model,” ACM Trans. Graph., vol. 34, no. 6, pp. 1–16, 2015.
G. Pavlovsk et al., “Expressive body capture: 3D hands, face, and body from a single image,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., 2019, pp. 10975–10985.
F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, “Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image,” in Proc. Eur. Conf. Comput. Vis., Springer, 2016, pp. 561–578.
C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler, “Unite the people: Closing the loop between 3D and 2D human representations,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 6050–6059.
O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Interv., Springer, 2015, pp. 234–241.
A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” in Proc. Eur. Conf. Comput. Vis., Springer, 2016, pp. 689–699.
T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2117–2125.
K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 5693–5703.
M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, “Neural body fitting: Unifying deep learning and model-based human pose and shape estimation,” in Proc. IEEE Int. Conf. 3D Vis., 2018, pp. 484–494.
Y. Xu, S.-C. Zhu, and T. Tung, “DenseReC: Joint 3D pose and shape estimation by dense render-and-compare,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 7760–7770.
H. Zhang, J. Cao, G. Liu, W. Ouyang, and Z. Sun, “Learning 3D human shape and pose from dense body parts,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 5, pp. 2610–2627, May 2022.
J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, “Human pose estimation with iterative error feedback,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 4733–4742.
J. Romero, D. Tzionas, and M. J. Black, “Embodied hands: Modeling and capturing hands and bodies together,” ACM Trans. Graph., vol. 36, no. 6, pp. 1–17, 2017.
T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, “Learning a model of facial shape and expression from 4D scans,” ACM Trans. Graph., vol. 36, no. 6, pp. 194:1–194:17, 2017.
F. S. Grassia, “Practical parameterization of rotations using the exponential map,” J. Graph., GPU, Game Tools, vol. 3, no. 3, pp. 29–48, 1998.
Y. Tian, H. Zhang, Y. Liu, and L. Wang, “Recovering 3D human mesh from monocular images: A survey,” 2022, arXiv:2203.01923.
D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, “SCAPE: Shape completion and animation of people,” ACM Trans. Graph., vol. 24, pp. 408–416, 2005.
L. Sigal, A. Balan, and M. J. Black, “Combined discriminative and generative articulated pose and non-rigid shape estimation,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2008, pp. 1337–1344.
P. Guan, A. Weiss, A. O. Balan, and M. J. Black, “Estimating human shape and pose from a single image,” in Proc. IEEE Int. Conf. Comput. Vis., 2009, pp. 1381–1388.
Y. Zhang, Z. Li, L. An, M. Li, T. Yu, and Y. Liu, “Lightweight multi-person total motion capture using sparse multi-view cameras,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 5560–5569.
Y. Huang et al., “Towards accurate marker-less human shape and pose estimation over time,” in Proc. IEEE Int. Conf. 3D Vis., 2017, pp. 421–424.
A. Zanfir, E. Mariniou, and C. Smichisescu, “Monocular 3D pose and shape estimation of multiple people in natural scenes - the importance of multiple scene constraints,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 2148–2157.
R. A. Güler and I. Kokkinos, “HoloPose: Holistic 3D human reconstruction in-the-wild,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 10884–10894.
ZHANG et al.: PyMAF-X: TOWARDS WELL-ALIGNED FULL-BODY MODEL REGRESSION FROM MONOCULAR IMAGES 12301

[65] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, “OpenPose: Performance analysis of state-of-the-art 2D human pose estimation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12919–12928.

[66] Z. Zeng, W. Ouyang, P. Luo, W. Liu, and X. Wang, “3D human mesh regression with dense correspondence,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7376–7385.

[67] S. Wu et al., “Graph-based 3D multi-person pose estimation using multi-view images,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 11128–11137.

[68] Z. Zheng, T. Yu, Y. Wei, Q. Dai, and Y. Liu, “DeepHuman: 3D human reconstruction from a single image,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 7738–7748.

[69] K. Lin, L. Wang, and Z. Liu, “Mesh graphormer,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 12919–12928.

[70] P. Yao, Z. Fang, F. Wu, Y. Feng, and J. Li, “DenseBody: Directly regressing dense 3D human pose and shape from a single color image,” 2019, arXiv:1903.11217.

[71] T. Zhang, B. Huang, and Y. Wang, “Object-occluded human shape and pose estimation from a single color image,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7376–7385.

[72] W. Zeng, W. Ouyang, P. Luo, W. Liu, and X. Wang, “3D human mesh regression with dense correspondence,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7054–7063.

[73] Z. Zhang, J. Yang, and C. Fowlkes, “The best of both worlds: Combining model-based and nonparametric approaches for 3D human body estimation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* Workshop, 2022, pp. 2318–2327.

[74] M. Fieraru, M. Zanfir, E. Oneata, A.-I. Popa, V. Olaru, and C. Sminchisescu, “Three-dimensional reconstruction of human interactions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7212–7221.

[75] W. Jiang, N. Kolotouros, G. Pavlakos, X. Zhou, and K. Daniilidis, “Coherent reconstruction of multiple humans from a single image,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5579–5588.

[76] J. Zhang, D. Yu, J. H. Liew, X. Nie, and J. Feng, “Body meshes as points,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 546–556.

[77] Z. Sun, Q. Bao, W. Liu, Y. Fu, M. J. Black, and T. Mei, “Monocular, one-stage, regression of multiple 3D people,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 11179–11188.

[78] Y. Sun, W. Liu, Q. Bao, Y. Fu, T. Mei, and M. J. Black, “Putting people in their place: Monocular regression of 3D people in depth,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13243–13252.

[79] M. Fieraru, M. Zanfir, T. Szente, E. Bazavan, V. Olaru, and C. Sminchisescu, “REMIPS: Physically consistent 3D reconstruction of multiple interacting people under weak supervision,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 19385–19397.

[80] M. Kocabas, N. Athanasious, and M. J. Black, “VIBE: Video inference for human body pose and shape estimation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5253–5263.

[81] H. Choi, G. Moon, J. Y. Chang, and K. M. Lee, “Beyond static features for temporally consistent 3D human pose and shape from a video,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 19964–19973.

[82] Z. Wan, Z. Li, M. Tian, J. Liu, S. Yi, and H. Li, “Encoder-decoder with multi-level attention for 3D human shape and pose estimation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 13033–13042.

[83] S. Zhang, Y. Zhang, F. Bogo, M. Pollefeys, and S. Tang, “Learning motion priors for 4D human body capture in 3D scenes,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 11343–11353.

[84] D. Rempe, T. Birdal, A. Hertzmann, J. Yang, S. Sridhar, and L. J. Guibas, “HuMoR: 3D human motion model for robust pose estimation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 11468–11479.

[85] G. Pavlakos, J. Malik, and A. Kanazawa, “Human mesh recovery from multiple shots,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1485–1495.

[86] R. Khurolkar, S. Tripathi, and K. Kitan, “Occluded human mesh recovery,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1715–1725.

[87] Y. Yuan, U. Iqbal, P. Molchanov, K. Kitan, and J. Kautz, “GLAMR: Global occlusion-aware human mesh recovery with dynamic cameras,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11038–11049.

[88] A. Sengupta, I. Budvytis, and R. Cipolla, “Hierarchical kinematic probability distributions for 3D human shape and pose estimation from images in the wild,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 11219–11229.

[89] V. Choutas, L. Müller, C.-H.-P. Huang, S. Tang, D. Tzionas, and M. J. Black, “Accurate 3D body shape regression using metric and semantic attributes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2718–2728.
[90] A. Sengupta, I. Budvytis, and R. Cipolla, “Probabilistic 3D human shape and pose estimation from multiple unconstrained images in the wild,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 16094–16104.

[91] N. Koutsouras, G. Pavlakos, D. Jayaraman, and K. Daniilidis, “Probabilistic modeling for human mesh recovery,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 11160–11170.

[92] I. Kissos, L. Fritz, M. Goldman, O. Meir, E. Oks, and M. Kliger, “Beyond weak perspective for monocular 3D human pose estimation,” in Proc. Eur. Conf. Comput. Vis., Springer, 2020, pp. 541–554.

[93] M. Kocaba, G. Zhou, F. Huang, J. Tesch, L. Muller, O. Hilliges, and M. J. Black, “SPEG: Seeing people in the wild with an estimated camera,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 11035–11045.

[94] S. Guan, J. Xu, Y. Wang, B. Ni, and X. Yang, “Bilevel online adaptation for out-of-domain human mesh reconstruction,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 10472–10481.

[95] Y. Rong, Z. Liu, and C. C. Loy, “Chasing the tail in monocular 3D human reconstruction with prototype memory,” IEEE Trans. Image Process., vol. 31, pp. 2907–2919, 2022.

[96] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, “PPIFu: Pixel-aligned implicit function for high-resolution clothed human digitization,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 2929–2938.

[97] Z. Zheng, T. Yu, Y. Liu, and Q. Dai, “PaMR: Parametric model-conditioned implicit representation for image-based human reconstruction,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 6, pp. 3170–3184, Jun. 2022.

[98] Y. Zheng et al., “DeepMultiCap: Performance capture of multiple characters using sparse multiview cameras,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 6239–6249.

[99] Y. Xiu, J. Yang, D. Tzionas, and M. J. Black, “ICON: Implicit cloaked humans obtained from normals,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 13296–13306.

[100] S. Baek, K. I. Kim, and T.-K. Kim, “Pushing the envelope for RGB-based dense 3D hand pose estimation via neural rendering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1067–1076.

[101] A. Boukhayma, R. D. Bem, and P. H. Torr, “3D hand shape and pose estimation from images in the wild,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 10843–10852.

[102] Y. Hasson et al., “Learning joint reconstruction of hands and manipulated objects,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 11807–11816.

[103] D. Kulon, H. Wang, R. A. Guler, M. M. Bronstein, and S. Zafeiriou, “Single image 3D hand reconstruction with mesh convolutions,” in Proc. Brit. Mach. Vis. Conf., 2019, pp. 1–4.

[104] L. Ge et al., “3D hand shape and pose estimation from a single RGB image,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1067–1076.

[105] D. Kulon, R. A. Guler, I. Kokkinos, M. M. Bronstein, and S. Zafeiriou, “Weakly-supervised mesh-convolutional hand reconstruction in the wild,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 4989–4999.

[106] B. Zhang et al., “Interacting two-hand 3D pose and shape reconstruction from single color image,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 11354–11363.

[107] Y. Rong, J. Wang, Z. Liu, and C. C. Loy, “Monocular 3D reconstruction of interacting hands via collision-aware factorized refinements,” in Proc. Int. Conf. 3D VI., 2021, pp. 432–441.

[108] A. Tewari et al., “Self-supervised multi-level face model learning for monocular reconstruction at over 250 Hz,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 2549–2559.

[109] L. Wang, Z. Chen, T. Yu, C. Ma, L. Li, and Y. Liu, “FaceVerse: A fine-grained and detail-controllable 3D face morphable model from a hybrid dataset,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 20332–20342.

[110] H. Joo, T. Simon, and Y. Sheikh, “Total capture: A 3D deformation model for tracking faces, hands, and bodies,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8320–8329.

[111] H. Xu, E. G. Bazavan, A. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, “GHUM & GHUML: Generative 3D human shape and articulated pose models,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6184–6193.

[112] D. Xiang, H. Joo, and Y. Sheikh, “Monocular total capture: Posing face, body, and hands in the wild,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 10965–10974.
[138] Z. Luo, S. A. Golestaneh, and K. M. Kitani, “3D human motion estimation via motion compression and refinement,” in Proc. Asian Conf. Comput. Vis., 2020, pp. 324–340.

[139] D. Mehta et al., “Monocular 3D human pose estimation in the wild using improved CNN supervision,” in Proc. Int. Conf. 3D Vis., 2017, pp. 506–516.

[140] S. Johnson and M. Everingham, “Learning effective human pose estimation from inaccurate annotation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2011, pp. 1465–1472.

[141] S. Jin et al., “Whole-body human pose estimation in the wild,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 196–214.

[142] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “VGGFace2: A dataset for recognising faces across pose and age,” in Proc. IEEE 13th Int. Conf. Autom. Face Gesture Recognit., 2018, pp. 67–74.

[143] A. Bulat and G. Tzimiropoulos, “How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks),” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 1021–1030.

[144] Z.-H. Feng et al., “Evaluation of dense 3D reconstruction from 2D face images in the wild,” in Proc. IEEE Int. Conf. Autom. Face Gesture Recognit., 2018, pp. 780–786.

[145] S. Kreiss, L. Bertonni, and A. Alahi, “OpenPosePaf: Composite fields for semantic keypoint detection and spatio-temporal association,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 8, pp. 13498–13511, Aug. 2021.

[146] L. Müller, A. A. Osman, S. Tang, C.-H. P. Huang, and M. J. Black, “On self-contact and human pose,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 9990–9999.

[147] C.-H. P. Huang et al., “Capturing and inferring dense full-body human-scene contact,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 13274–13286.

Hongwen Zhang received the BE degree from the South China University of Technology, Guangzhou, China, in 2015, and the PhD degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2021. He is currently a post-doctoral researcher with the Department of Automation, Tsinghua University. His research interests include computer vision, computer graphics, and their applications in 3D human modeling.

Yating Tian received the BE degree from Wuhan University, Wuhan, China, in 2020. She is currently working toward the postgraduate degree with the Department of Computer Science and Technology, Nanjing University. Her current research interest lies in human pose and shape estimation.

Yuxiang Zhang received the BE degree from Tsinghua University, China, in 2019. He is currently working toward the PhD degree with the Department of Automation, Tsinghua University, Beijing, China. His research focuses on computer vision and graphics, specifically on multi-person markless motion capture and 3D human reconstruction.

Liang An received the BE degree from the Tsinghua University, China, in 2017. He is currently working toward the PhD degree with the Department of Automation, Tsinghua University, Beijing, China. His research areas include 3D reconstruction of human and animals.

Mengcheng Li received the BS degree from the Department of Physics, Tsinghua University, Beijing, China, in 2018. He is currently working toward the PhD degree with the Department of Automation, Tsinghua University. His research areas include computer vision and 3D reconstruction.

Zhenan Sun (Senior Member, IEEE) received the BS degree in industrial automation from the Dalian University of Technology, Dalian, China, in 1999, the MS degree in system engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2002, and the PhD degree in pattern recognition and intelligent systems from the Chinese Academy of Sciences, Beijing, China, in 2006. Since 2006, he has been a faculty member with the Institute of Automation, Chinese Academy of Sciences, where he is currently a professor. His current research interests include biometrics, pattern recognition, and computer vision. He is a fellow of the IAPR.

Yebin Liu (Member, IEEE) received the BE degree from the Beijing University of Posts and Telecommunications, China, in 2002, and the PhD degree from the Department of Automation, Tsinghua University, Beijing, China, in 2009. He is currently a professor with Tsinghua University. He was a research fellow with the Computer Graphics Group of the Max Planck Institute for Informatik, Germany, in 2010. His research areas include computer vision, computer graphics, and computational photography.