FrankMocap: A Monocular 3D Whole-Body Pose Estimation System via Regression and Integration

Yu Rong1,3* Takaaki Shiratori2 Hanbyul Joo3
1The Chinese University of Hong Kong 2Facebook Reality Labs 3Facebook AI Research

Abstract

3D human pose estimation can facilitate various applications, such as assistive technologies and AR/VR. Most existing monocular 3D pose estimation approaches only focus on a single body part, neglecting the fact that the essential nuance of human motion is conveyed through a concert of subtle movements of face, hands, and body. In this paper, we present FrankMocap, a fast and accurate whole-body 3D pose estimation system that can produce 3D face, hands, and body simultaneously from in-the-wild monocular images. The idea of FrankMocap is its modular design: We first run 3D pose regression methods for face, hands, and body independently, followed by composing the regression outputs via an integration module. The separate regression modules allow us to take full advantage of their state-of-the-art performances without compromising the original accuracy and reliability in practice. We develop three different integration modules that trade off between latency and accuracy. All of them are capable of providing simple yet effective solutions to unify the separate outputs into seamless whole-body pose estimation results. We quantitatively and qualitatively demonstrate that our modularized system outperforms both the optimization-based and end-to-end methods of estimating whole-body pose.

*Work done during Yu Rong’s internship at Facebook AI Research

1. Introduction

Estimating 3D human pose from single RGB images is one of the core technologies to build a computational model to understand human behavioral cues. It can facilitate numerous applications including assistive technology [23, 37, 49], sign language understanding [10], AR/VR [3, 62], and social signal understanding [47]. Importantly, the essential nuance of human behaviors is conveyed through a concert of subtlest movements of face, hands, and body. Thus, it is necessary to estimate whole-body motions to capture the authentic signals.

Estimating whole-body 3D poses including a face and hands, however, remains challenging. One of the major difficulties comes from the fact that the scale of faces and hands is much smaller than that of torsos and limbs. Furthermore, hands in motion are susceptible to artifacts caused by abrupt viewpoint changes, self-occlusions, and motion blur. These factors largely raise the difficulties of creating large-scale datasets with whole-body 3D poses, even in a controlled environment [29, 36, 50], not to mention capture such data in the wild. The lack of whole-body 3D data is a major obstacle preventing the construction of a unified system to estimate whole-body 3D poses simultaneously. Therefore, most existing methods focus on one of the major body parts individually: single image-based estimation of 3D body pose (i.e., torso and limbs) [5, 30, 33, 34, 54, 63, 64], 3D hand pose [8, 9, 20, 27, 69, 72] or 3D
In this paper, we present \textit{FrankMocap}, a modular system for estimating whole-body 3D poses in a unified output in SMPL-X form \cite{SMPL-X}, as visualized in Fig. 1. Our system is built upon the insight that training a single model to jointly estimate whole parts is intrinsically limited by the lack of accurate and diverse whole-body motion data. Instead, we design a modularized system to first run 3D pose regression methods for face, hands, and body independently. The three regression outputs are then composed via an integration module. The separate regression modules allow us to take full advantage of their state-of-the-art performances without compromising the original accuracy and reliability in practice. Our integration module provides a simple yet effective solution to unify them into seamless whole-body pose estimation outputs. Unlike previous whole-body pose estimation methods that require computationally heavy optimization \cite{Adam, SMPL-X, MTC}, our method also allows us to run at an interactive frame rate. We quantitatively and qualitatively demonstrate that our modular framework outperforms existing optimization-based methods \cite{SMPL-X, MTC} and end-to-end models trained to produce whole-body outputs jointly \cite{Adam}.

In conclusion, we make three major contributions. First, towards a practical system for whole-body pose estimation, we present the idea of \textit{regression-and-integration} to take full advantage of existing single-part pose datasets and circumvent the hurdle of lacking whole-body pose datasets. Especially, our hand module is comparable with SOTA hand-only methods on in-the-wild scenarios. Second, we present three effective strategies to integrate outputs from separate single-part pose estimation modules, namely the fastest “copy-paste” approach, a full-optimization approach, and an intermediate approach via a simple integration network. Third, we quantitatively and qualitatively demonstrate that our modular system outperforms the alternative approaches based on optimization-based methods or end-to-end methods training all parts jointly.

2. Related Work

3D Parametric Human Body Models. 3D parametric human models are widely used as a strong prior to estimate 3D pose and shape of humans. A main idea is to model the deformation of 3D human (including face, hands and body) via low dimension parameters \cite{3DMM, Adam, SMPL-X, SMPLify-X, MANO}. SCAPE \cite{3DMM} is a pioneering work that accounts for shape variations and pose deformations of human body. Loper \textit{et al.} introduce SMPL \cite{SMPL} that learns local pose-dependent blendshape on top of linear blend skinning for holistic mesh deformation as well as shape variations. Similarly, there exists a hand deformation model called MANO \cite{MANO} and several face models \cite{3DMM, Adam, FLAME}. Recent approaches also introduce whole-body models such as Adam \cite{Adam}, SMPL-X \cite{SMPL-X}, and GHUM & GHUML \cite{GHUM} that are capable of expressing face, body, and hands in a unified space.

Single Image 3D Body, Hand and Face Estimation. Recent monocular 3D body motion capture approaches adopt parametric model such as the SMPL \cite{SMPL} or Adam model \cite{Adam, MTC} for 3D body representation\footnote{Note that \textit{body} indicates torso and limbs excluding finger joints and \textit{whole-body} indicates all of face, hands and body.}. A pioneering work \cite{Adam} uses an optimization framework to fit the 3D body model to 2D observations. More recent methods \cite{SMPL-X, MTC, SCAPE, Adam, FLAME, MANO, GHUM} leverage a deep learning framework to regress parameters of the body models from RGB images. Non-parametric methods have been also introduced by directly regressing the model vertices \cite{Adam, FLAME} or corresponding UV maps \cite{GHUML}. There are also approaches that use a hybrid framework by using a deep learning framework to produce intermediate representations such as 2D heatmaps and then fitting the skeletal models on these outputs to reconstruct joint angles \cite{Adam, FLAME, MTC}. Due to the lack of in-the-wild training data with 3D annotations, most of these models are trained with mixed datasets, including indoor datasets such as Human3.6M \cite{H36M} and in-the-wild datasets such as COCO \cite{COCO}. While most methods are based on single images as input, recent approaches \cite{MTC, FLAME} predict 3D motions from a video.

Monocular 3D Hand pose estimation approaches share a similar pipeline as the body counterparts, based on the parametric 3D hand models, such as MANO \cite{MANO}. Deep neural networks are leveraged to either predict model parameters \cite{Adam, FLAME, SMPLify-X, MANO} or directly regress hand mesh vertices \cite{Adam, MANO}. Similarly, single image 3D face prediction methods \cite{Adam, FLAME, MANO} also share similar ideas by leveraging deep neural networks to regress the face landmarks or parameters of face models, such as 3DMM \cite{3DMM} or FLAME \cite{FLAME}. We refer the readers to recent surveys on 3D face reconstruction \cite{3DFaceSurvey} for more details.

Joint 3D Estimation of Body, Hands and Face. There are a few methods \cite{Adam, SMPL-X, MTC} aiming to capture whole-body 3D motions. SMPLify-X optimizes the parameter of the SMPL-X model \cite{SMPL-X} to fit it to 2D keypoints with additional constraints, including body pose priors and collision penalizer. Monocular Total Capture (MTC) \cite{MTC} is based on the Adam model \cite{Adam}, and adopts deep neural networks to get 2.5D predictions first. Then the parameters of Adam are obtained through optimization. Both of these methods rely on optimization procedures with relatively slow computational time (from 10 seconds to a few minutes). Zhou \textit{et al.} \cite{Zhou} use SMPL-H \cite{MANO} to represent body and hands and 3DMM face model \cite{3DMM} to represent 3D faces. They first predict 3D body and hand poses and then separately predict the parameters of 3DMM model, including shape, expression, albedo, and illumination. The predicted full-body motion is not represented in a unified format and this paper does not present generalization ability to in-the-wild scenar-
ios. ExPose [18], avoids using the optimization procedure by presenting a neural network to simultaneously predict the body, hand, and face parameters of SMPL-X. In particular, ExPose curates a pseudo-ground truth dataset by fitting SMPL-X model on in-the-wild images, followed by manual quality checking by annotators, and use it to train their model to jointly produces output.

3. Method

Given a single image as input, FrankMocap firstly estimates the 3D poses of face, hands (both left and right), and body (the torso and limb parts), in the form of SMPL-X model [50]. Each part’s output is produced by a separate 3D pose regressor trained on public datasets in each sub-field. Their outputs are combined through our integration module, to produce a seamless and unified whole-body pose estimation output. Fig. 2 illustrates an overview of the proposed framework.

3.1. Whole-Body Parameterization

SMPL-X Model. We formulate the SMPL-X model as:

$$V_w = W(\phi_w, \theta_w, \beta_w, \psi_f)$$  \hspace{1cm} (1)

where $W$ is parameterized by global orientation of the whole-body $\phi_w \in \mathbb{R}^3$, facial expressions parameters $\psi_f \in \mathbb{R}^{10}$, whole-body pose parameters $\theta_w \in \mathbb{R}^{(21+15+15+1) \times 3}$ accounting for pose-dependent deformation, and shape parameters $\beta_w \in \mathbb{R}^{10}$ accounting for cross-identity shape variations of the face, hands, and body. We divide $\theta_w$ for each of the body, hands, and faces, namely body pose parameters $\theta^b_w \in \mathbb{R}^{15 \times 3}$, left hand pose parameters $\theta^{lh}_w \in \mathbb{R}^{15 \times 3}$, right hand pose parameters $\theta^{rh}_w \in \mathbb{R}^{15 \times 3}$, and face pose parameters $\theta^f_w \in \mathbb{R}^{1 \times 3}$. In this way, $\theta_w = \{\theta^b_w, \theta^{lh}_w, \theta^{rh}_w, \theta^f_w\}$. All pose parameters are defined in the axis-angle representation, which stores the relative rotation to the parent joints defined in the kinematics map. As output, the SMPL-X model produces a mesh structure with 10,745 vertices, $V_w \in \mathbb{R}^{10475 \times 3}$. The 3D joint locations of the whole-body can be obtained by applying a joint regression function $R$ from the posed vertices as $J^{3D}_w = R_w(V_w)$, where $J^{3D}_w \in \mathbb{R}^{(22+20+20+3) \times 3}$.

Stand-Alone Hand Model. Our hand model is defined by taking the hand parts of SMPL-X:

$$V_h = H(\phi_h, \theta_h, \beta_h),$$  \hspace{1cm} (2)

where $\theta_h \in \mathbb{R}^{1 \times 15}$ is hand pose parameters and $\beta_h$ is the shape parameters for the hand model. Since our hand model is taken from SMPL-X, $\beta_h$ shares the same shape space as $\beta_w$. For brevity, we use $\theta_h$ instead of $\theta_{lh}$ or $\theta_{rh}$, to denote the hand pose parameters of either part in our description. $\phi_h \in \mathbb{R}^3$ represents the global orientation of the hand meshes. Our hand model $H$ produces the hand mesh structure with 778 vertices, $V_h \in \mathbb{R}^{778 \times 3}$, where the hand vertices are selected from the hand area of the original SMPL-X vertices. Given hand vertices $V_h$, 3D hand joints can be regressed as $J^{3D}_h = R_h(V_h)$, where $J^{3D}_h \in \mathbb{R}^{21 \times 3}$ contains a wrist, 15 finger joints (3 joints per finger), and 5 finger tips. See supplemental material for hand visualization and skeleton hierarchy. The major advantage of our hand representation is that the components of this 3D hand model, including pose parameters, vertices, and 3D joints, are directly compatible with the whole-body parameterization. This enables us to efficiently integrate outputs from the body module and the hand module.

\[\text{1751}\]
3.2. 3D Hand Estimation Module

We present a monocular 3D hand pose estimation module, denoted by $M_H$, to the parameters of the hand model $H$. In particular, our hand module is inspired by the recently proposed monocular body pose estimation approaches [30, 31, 33]. We follow similar model architecture, parameterization, and training strategies as the pioneering body methods. In particular, we introduce blur augmentation to make the hand module more robust to blurry hands that are frequently observed in full body motion capture scenarios.

Network Architecture. Our hand module $M_H$ is built upon an end-to-end deep neural network architecture that can predict hand model parameters defined in Eq. (2) from given input images. This process is defined as:

$$[\phi_h, \theta_h, \beta_h, c_h] = M_H(I_H),$$  

(3)

where $I_H$ is an input RGB image cropped around the hand region. $c_h = (t_h, s_h)$ is a set of weak-perspective camera parameters to project a posed 3D hand model to the input image. $t_h \in \mathbb{R}^2$ is for 2D translation on the image plane, and $s_h \in \mathbb{R}$ is a scale factor. Given camera parameters, the $i$-th 3D hand joint, $J_{h, i}^D$, can be projected as:

$$J_{h, i}^D = s_h \Pi(J_{h, i}^{3D}) + t_h,$$

(4)

where $\Pi$ is an orthographic projection. Following the practice of HMR [30, 33], our hand module $M_H$ adopts an encoder-decoder structure, where the encoder (ResNet-50 [25]) outputs features from an input image, and the decoder regresses the hand model parameters from the features. See Fig. 3 for the overview of our hand module. In particular, our hand module is trained with the assumption that the inputs are right hands. The images and annotations for the left hand are used after vertical flipping. During inference, the left-hand images are first flipped and processed as if they were a right hand. Then the outputs are flipped back to the original left-hand space. Note that the shape parameter $\beta_h$ is originally defined for whole-body model $\beta_w$, but we only consider the deformation for the hand vertices defined in Eq. (2), ignoring the body part.

Training Method. We consider three different types of annotations: (1) 3D pose annotations (in axis-angle representation), (2) 3D keypoint (joint) annotations, and (3) 2D keypoint annotations. The losses for each of the annotation, namely $L_\theta$, $L_{3D}$ and $L_{2D}$, are defined as follows: $L_\theta = \|\theta_h - \bar{\theta}_h\|_2^2$, $L_{3D} = \| J_{h}^{3D} - \bar{J}_{h}^{3D} \|_2^2$, and $L_{2D} = \| J_{h}^{2D} - \bar{J}_{h}^{2D} \|_1$, where $\theta_h$, $\bar{J}_{h}^{3D}$ and $\bar{J}_{h}^{2D}$ are the ground-truth axis-angle pose parameters, 3D keypoints, and 2D keypoints, respectively. In particular, the 2D keypoint loss are leveraged to estimate the camera parameters and facilitate the hand module generalize to in-the-wild images with only 2D keypoints annotation. We do not use the shape parameters provided by the 3D hand datasets such as FreiHAND [73], since these are defined for the MANO model [52] and not compatible with our hand model from SMPL-X. Instead, an additional shape parameter regularization loss $L_{reg} = \|\beta_h\|_2^2$ is applied. The overall loss $L$ used to train our hand module is defined as follows:

$$L = \lambda_1 L_\theta + \lambda_2 L_{3D} + \lambda_3 L_{2D} + \lambda_4 L_{reg}.$$

(5)

In experiments, the balanced weights are set as $\lambda_1 = 10$, $\lambda_2 = 100$, $\lambda_3 = 10$ and $\lambda_4 = 0.1$. Other hand module training details, including data preprocessing, dataset information, and implementation details, are included in the supplemental material.

Motion Blur Augmentation. Performing data augmentation during training is a common practice to enable a model with better generalizability. We firstly apply common data augmentation strategies, including random scale, random translation, color jittering, and random rotation. Importantly, we recognize that in-the-wild videos are often accompanied by severe motion blur. To achieve robustness to motion blur, we additionally apply motion blur augmentation to the images. We use the methods [6, 7] to generate blur kernels and then use 2D filtering to add blurriness to images. Examples of applying motion blur augmentation are shown in Fig. 4. Our experiments clearly show that our motion blur augmentation can effectively insist our hand module generalize to in-the-wild scenarios.

3.3. 3D Body Estimation Module

Our body module $M_B$ produces the torso and limb parameters defined in Eq. (1) from a single image:

$$[\phi_b, \theta_b, \beta_b, c_b] = M_B(I_b),$$

(6)
where \( I_h \) is an input image cropped around a target single person’s whole-body. Similar to Eq. (1), \( \phi_b \in \mathbb{R}^3 \) is the global body orientation, \( \theta_b \in \mathbb{R}^{21 \times 3} \) is the body pose parameters (without any finger joints), and \( \beta_b \in \mathbb{R}^{10} \) is the shape parameter. \( \beta_b \) shares the same parameterization space as \( \beta_w \), as defined in Eq. (1). Similar to Eq. (3), we use weak perspective camera parameters \( c_0 = (k_b, s_b) \).

We leverage the publicly available body pose estimation model (SPIN [33]) trained by the EFT dataset [28]. Since the original SPIN model is built with SMPL, we fine-tune the network by replacing SMPL with SMPL-X. For the fine-tuning process, we use the existing indoor 3D pose dataset Human3.6M [26] and outdoor pseudo-GT dataset by EFT [28] that provides SMPL annotations. Since the annotation (SMPL) and our target model (SMPL-X) are not exactly compatible, we only use the pose parameters and 2D keypoint from the annotations during fine-tuning process, ignoring the shape parameters.

### 3.4. 3D Face Estimation Module

We use the off-the-shelf public 3D face estimation model RingNet [55] to estimate facial expressions \( \psi_f \) and face poses \( \theta_f \) as follows:

\[
[\theta_f, \psi_f] = M_F(I_f).
\]

Since RingNet is based on FLAME [39], its predictions are compatible with the face part of SMPL-X. As an adjustment, we only keep the first 10 expression parameters from the original 50 dimensional expression parameters \( \psi_f \) predicted by RingNet, to make it compatible with the expression space defined in SMPL-X. It is worth noting that the FLAME and SMPL-X share the same PCA-based expression space, while SMPL-X only uses the 10 foremost principal components.

### 3.5. whole-body Integration Module

Our integration module combines the outputs from the face, hands, and body modules into a unified representation of the SMPL-X [50]. While the output of the face module can be easily applied to the face part of the SMPL-X model by copying the expression parameters and jaw poses, additional procedures are required to integrate the outputs of the hand and body modules because different wrist poses are estimated from these two modules. For this objective, we present three strategies: (1) by simple copy-paste composition for fastest processing, (2) by an optimization framework, (3) by an integration network to approximate the optimization via a simple neural network.

**By Copy-Paste.** Our hand and body modules’ outputs can be efficiently combined, since they are both defined in SMPL-X parameterization. As the simplest strategy, we transfer the corresponding joint angle parameters from the outputs of each module into the whole-body model. The wrist pose parameters require additional processing because we obtain two different outputs from the body module and the hand module (the wrist pose of hand module is represented by the global hand orientation \( \phi_h \)). Let us denote the pose parameters for the wrist joint as \( \theta_{wrist} \), then \( \theta_b = \theta_t \cup \{ \theta_{wrist}^b, \theta_{wrist}^w \} \), where \( \theta_t \) represents all body pose parameters except wrists. We use the similar notations for the whole-body pose parameters, \( \theta_w = \theta_t \cup \{ \theta_{wrist}^w, \theta_{wrist}^b \} \).

Then, whole-body integration by copy-paste can be performed as:

\[
(\phi_w, \beta_w, c_w, \theta_w^b, \theta_w^h, \theta_{wrist}^w, \theta_{wrist}^b) = (\phi_h, \beta_h, c_b, \theta_b, \theta_{lh}, \theta_{rh}),
\]

\[
(\theta_{wrist}^w, \theta_{wrist}^b) = (\Gamma_l(\theta_b, \phi_{lh}), \Gamma_r(\theta_b, \phi_{rh})),
\]

where \( \Gamma_{l,r} \) are the functions to convert the global wrist orientation \( \phi_h \) obtained from the hand module to the local wrist pose parameters w.r.t. its parent joint in the SMPL-X skeleton hierarchy. This can be implemented by comparing \( \phi_h \) with the global orientation of the current wrist pose from \( \theta_b \) that can be computed by following the forward kinematics of the body skeleton hierarchy. This strategy requires almost no extra computation, making our separate modules contribute a common whole-body model simultaneously. We found this simple integration produces convincing results, especially for the scenarios with computational bottlenecks as in our live demo.

**By Optimization.** As an alternative integration method for better accuracy, we build an optimization framework to fit the whole-body model parameters given the outputs from body and hand modules. This strategy is particularly helpful to reduce the artifact around the wrist parts over the copy-paste strategy. It can also take advantage from the 2D keypoint estimation output [13] for better 2D localization quality. Our optimization framework finds the whole-body model parameters that minimize the following objective cost function:

\[
F([\phi_w, \theta_w, \beta_w, c_w]) = F^{2d} + F^{mesh} + F^{pri},
\]

where \( F^{2d} \) is the 2D re-projection cost term between the 2D keypoint estimation [13] and the projection of 3D joints (hand, body and faces). \( F^{mesh} \) is the 3D distance loss between the hand mesh of whole-body model and the mesh vertices of the hand module’s output localized on the body’s wrist joints, as shown in Fig. 5. The prior term \( F^{pri} \) is needed to keep the 3D pose and shape parameters in plausible space. We use the EFT introduced in [28] for the similar goal, by applying neural network finetuning to replace explicit 3D pose prior terms used by previous works [50]. Therefore, the prior term only accounts for the shape regularization \( F^{pri} = \| \beta_w \|^2_2 \).
Figure 5: Optimizing the whole-body model (SMPL-X) with 3D hand prediction and 2D keypoint estimation. (a) Input images and estimated 2D keypoints by OpenPose [13]; (b) 3D body pose estimation from body module; (c) The output of 3D hand module aligned to the wrist joints of SMPL-X; (d) Integration output by copy-paste; (e) Integration output by optimization framework.

We use a multi-stage approach in which optimization is firstly performed without $F_{\text{mesh}}$. Then we optimize all terms in the second stage. During the second stage, we add an additional constraint $F_{3D}$ to maintain the 3D joints within the final locations of the first stage. Our experiments show that this term can produce reliable results in practice. See Fig. 5 for the example of our optimization.

By Wrist Integration Network. The result of the hand module shows precise 2D localization quality as shown in Fig. 6. But we lose its precision by applying copy-paste integration strategy since the wrist joint location is determined by the body module, as shown in Fig. 5 (second last column). Although the optimization method can achieve better precision and resolve this problem, it suffers from relatively slow computation speed due to the requirement of iterative gradient computations. As another alternative strategy to achieve both precision and fast-runtime performance, we introduce an integration network to adjust the arm pose of the whole-body model toward the given 2D location without gradient computation. Specifically, we develop a small neural network to adjust the arm parameters of whole-body model:

$$\tilde{\theta}_{\text{arm}}^w = \mathcal{H}(\theta_{\text{arm}}^w, \mathbf{d}),$$ (10)

where $\mathbf{d} \in \mathbb{R}^2$ is a 2D directional vector that the arm needs to follow in the image space. It is obtained from the wrist locations calculated from the body module and those from the hand module, followed by a normalization with the length of the arms. $\theta_{\text{arm}}^w$ is the pose parameters of the elbow and shoulder joints after the copy-paste integration. Conceptually, $\mathcal{H}$ predicts an approximation of optimization gradient to adjust the arm parameters, similar to the recent work of Song et al. [57]. We use a MLP with six layers to implement $\mathcal{H}$. To train $\mathcal{H}$, we generate a synthetic dataset by capturing our own a range of motion videos to cover diverse arm pose variations and by simulating arbitrary $\mathbf{d}$ directions. See supplemental material for more details of the wrist integration.

4. Experiments

We first demonstrate that our hand module achieves high performance competing to the previous state-of-the-art 3D hand methods, along with ablation studies to examine the designs of our method. Then, we demonstrate that our whole-body pose estimation method outperforms previous approaches on a public benchmark. See supplemental material for the details of implementation and datasets.

4.1. Hand Module Evaluation

Comparison with State-of-the-art Methods. We compare our hand module with the previous state-of-the-art hand approaches on three public hand benchmarks, namely

Table 1: Comparison of our hand module with the state-of-the-art 3D hand methods using 2D/3D AUC of PCK.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>STB</td>
<td>0.994</td>
<td>-</td>
<td>0.995</td>
<td>-</td>
<td>0.992</td>
<td>-</td>
<td>0.934</td>
</tr>
<tr>
<td>RHD</td>
<td>0.926</td>
<td>0.92</td>
<td>0.901</td>
<td>0.95</td>
<td>0.944</td>
<td>0.934</td>
<td>0.925</td>
</tr>
<tr>
<td>MPII+NZSL</td>
<td>0.501</td>
<td>-</td>
<td>0.15</td>
<td>0.701</td>
<td>-</td>
<td>-</td>
<td>0.655</td>
</tr>
</tbody>
</table>

Table 2: Ablation study on hand module data augmentation. Models are evaluated on MPII+NZSL [56] with 2D AUC.

<table>
<thead>
<tr>
<th>Translation</th>
<th>Rescale</th>
<th>Color</th>
<th>Rotation</th>
<th>Blur</th>
<th>2D AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>0.608</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.610</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>0.618</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>0.622</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>0.655</td>
</tr>
</tbody>
</table>

Figure 6: Qualitative comparison with state-of-the-art 3D hand methods. Images are selected from COCO [41].

Figure 7: Ablation study on data augmentations.
STB [68], RHD [72] and MPII+NZSL [56]. The results are listed in Tab. 1 and Fig. 6. For each validation dataset, we calculate the percentage of correct keypoints (PCK) under different thresholds and calculate the corresponding Area Under Curve (AUC) for PCK. For STB [68] and RHD [72], we use 3D AUC and the threshold ranges from 20 mm to 50 mm. For MPII+NZSL [56], we use 2D AUC and the threshold ranges from 0 px to 30 px. For fair comparison, all the methods take a single RGB image as input. As shown in the results, our hand module is comparable with the previous state-of-the-art hand-only methods based on MANO [52], though our hand model, taken from the whole-body model, has restricted shape variations 4.

Ablation Study. We examine the data augmentation strategies used in training hand module. The results are listed in Tab. 2 and Fig. 7. The results in Tab. 2 demonstrate that applying data augmentation leads to better results. The qualitative results in Fig. 7 further show that adopting data augmentation helps our model generalize better on challenging scenarios, including blur, pose variations, and occlusion. In Fig. 7, “No Augment” refers to the model trained without any data augmentation, and “No Blur” refers to the model trained with all data augmentation strategies except the motion blur augmentation. “Full” refers to the model trained with all data augmentation strategies. We also do an ablation study on mixture usage of datasets, demonstrating that using diverse datasets improves the performances. See supplemental material for details on this part.

4Note that the code of Kulton et al. [35] is not publicly available, and, thus, we are unable to make more comparisons in challenging in-the-wild scenes, our target applications. CMR [15] requires camera intrinsic as known, which prevents applying them to in-the-wild scenarios.

4.2. Body Module Evaluation

We compare our body module with the previous SOTA body-only methods on the test set of 3DPW [61]. Following the previous approaches, we use Mean Per Joint Position Errors (MPJPE) and PA-MPJPE (after Procrustes analysis [22]). The results are listed in Tab. 3. Our body module is comparable to the previous state-of-the-arts, including ExPose [18]. Our body model is a fine-tuned version of SPIN [33] and EFT [28]. The slightly degraded performance might be due to the reason that we only use fewer datasets, COCO and Human3.6M datasets. We only compare with HMR [30]-based methods. We believe that the experimental results shown in Tab. 3 can demonstrate the feasibility of extending previous SMPL-based methods to

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MPJPE</td>
<td>130</td>
<td>127.2</td>
<td>96.9</td>
<td>92.3</td>
<td>93.4</td>
<td>94.3</td>
</tr>
<tr>
<td>PA-MPJPE</td>
<td>81.3</td>
<td>70.2</td>
<td>59.2</td>
<td>54.2</td>
<td>60.7</td>
<td>60.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>V2V</td>
<td>PA-V2V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Body</td>
<td>146.2</td>
<td>81.9</td>
<td>69.9</td>
<td>69.3</td>
<td>63.5</td>
</tr>
<tr>
<td>FM-CP</td>
<td>68.0</td>
<td>57.4</td>
<td>58.2</td>
<td>57.1</td>
<td>54.7</td>
</tr>
<tr>
<td>FM-WI</td>
<td>231.3</td>
<td>116.4</td>
<td>83.7</td>
<td>83.9</td>
<td>93.1</td>
</tr>
<tr>
<td>FM-OPT</td>
<td>75.4</td>
<td>55.1</td>
<td>52.7</td>
<td>53.2</td>
<td>53.3</td>
</tr>
<tr>
<td>L Hand</td>
<td>V2V</td>
<td>PA-V2V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FM-CP</td>
<td>34.6</td>
<td>39.3</td>
<td>27.1</td>
<td>27.0</td>
<td>28.6</td>
</tr>
<tr>
<td>FM-WI</td>
<td>11.4</td>
<td>13.1</td>
<td>10.9</td>
<td>10.9</td>
<td>10.8</td>
</tr>
<tr>
<td>FM-OPT</td>
<td>13.1</td>
<td>12.9</td>
<td>11.2</td>
<td>11.2</td>
<td></td>
</tr>
<tr>
<td>R Hand</td>
<td>V2V</td>
<td>PA-V2V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FM-CP</td>
<td>34.3</td>
<td>39.3</td>
<td>25.4</td>
<td>25.5</td>
<td>25.6</td>
</tr>
<tr>
<td>FM-WI</td>
<td>13.3</td>
<td>12.9</td>
<td>11.2</td>
<td>11.2</td>
<td></td>
</tr>
<tr>
<td>FM-OPT</td>
<td>23.6</td>
<td>19.5</td>
<td>30.7</td>
<td>30.7</td>
<td>20.4</td>
</tr>
<tr>
<td>Face</td>
<td>V2V</td>
<td>PA-V2V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FM-CP</td>
<td>8.4</td>
<td>5.6</td>
<td>5.7</td>
<td>5.7</td>
<td>5.4</td>
</tr>
<tr>
<td>FM-WI</td>
<td>8.4</td>
<td>5.6</td>
<td>5.7</td>
<td>5.7</td>
<td></td>
</tr>
<tr>
<td>FM-OPT</td>
<td>8.4</td>
<td>5.6</td>
<td>5.7</td>
<td>5.7</td>
<td></td>
</tr>
</tbody>
</table>

Figure 8: Qualitative comparison of FrankMocap model with different integration modules.

SMPL-X based body module. Thanks to our modularized design, we can also easily leverage the most recent 3D body poses methods [38, 40] to build our body module.

4.3. Integration Module Evaluation

We quantitatively compare our methods with previous SMPL-X based whole-body methods (SMPLify-X [50] and ExPose [18]) on EHF dataset [50]. The evaluation metrics are vertex-to-vertex distance (V2V) and Procrustes Analysis vertex-to-vertex distance (PA-V2V). The results are listed in Tab. 4. We run the officially released codes of SMPLify-X and ExPose to obtain the evaluation results. FM-CP, FM-WI, and FM-OPT indicate FrankMocap using the copy-paste, wrist integration network, and optimization-based as the integration module, respectively. The results demonstrate that all our methods significantly outperform the previous methods in terms of whole-body estimation, body estimation, and hand estimation. Our face estimation is comparable with previous methods, where we only use an off-the-shelf 3D face model [55]. Comparing across different versions of FrankMocap reveal that the model with optimization achieves the best performance in general. Applying our wrist integration network can also lead to better performance than copy-paste version. We also provide qualitative comparisons between different integration modules in Fig. 8. It is revealed that using the wrist integration network can bring better hand location while adopting optimization can further improve hand localization while up-
We also qualitatively compare our method with previous whole-body motion capture approaches, MTC [64], SMPLify-X [50], and ExPose [18], with our output with wrist integration network. The results are shown in Fig. 9 and in supplementary videos, demonstrating that our method outperforms previous approaches, with fast and accurate performance in challenging in-the-wild scenes. Qualitative comparison with Zhou et al. [70] is included in the supplemental material.

5. Conclusion

We present FrankMocap, a monocular 3D whole-body motion capture system built upon the motivation to take full advantages of state-of-the-art part estimation modules, followed by effective integration algorithms. We develop three integration modules, namely copy-paste, wrist integration network, and an optimization-based method. The copy-paste and wrist integration modules are suitable for time-sensitive applications, while the optimization-based module is suitable for offline applications requiring better precision. Our system surpasses previous state-of-the-art 3D whole-body estimation methods on both public benchmarks and in-the-wild scenarios. Notably, our approach can directly leverage future algorithms by replacing each module, which is a main advantage of our modular design.

Acknowledgements. We acknowledge Yuting Ye from FRL, Donglai Xiang and Yu Ding from CMU, Yu Xiong, Anyi Rao, and Jingbo Wang from CUHK, and Jie Chen from Microsoft.
References


[20] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3D hand shape and pose estimation from a single RGB image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 6


[47] Jie Song, Xu Chen, and Omur Hilliges. Human body model fitting by learned gradient descent. In *European Conference...


