arXiv:2008.08324v1 [cs.CV] 19 Aug 2020

FrankMocap: Fast Monocular 3D Hand and Body Motion Capture by Regression and Integration

Yu Rong[1,3] Takaaki Shiratori[2] Hanbyul Joo[3]

[1]The Chinese University of Hong Kong [2]Facebook Reality Labs [3]Facebook AI Research



Figure 1: We present a whole 3D body motion capture system, named *FrankMocap*, to simultaneously estimate 3D body and hand motion from monocular videos in the wild. Our system allows us to perform a live whole body motion capture demo using a single RGB webcam, as shown on the left. On the right, several example results on in-the-wild videos are demonstrated, where we show the input images (left), the 3D hand motion capture outputs (middle), and the whole body motion capture outputs (right).

Abstract

Although the essential nuance of human motion is often conveyed as a combination of body movements and hand gestures, the existing monocular motion capture approaches mostly focus on either body motion capture only ignoring hand parts or hand motion capture only without considering body motion. In this paper, we present FrankMocap[1], a motion capture system that can estimate both 3D hand and body motion from in-the-wild monocular inputs with faster speed (9.5 fps) and better accuracy than previous work. To construct FrankMocap, we build the state-of-the-art monocular 3D "hand" motion capture method by taking the hand part of the whole body parametric model (SMPL-X). Our 3D hand motion capture output can be efficiently integrated to monocular body motion capture output, producing whole body motion results in a unified parrametric model structure. We demonstrate the state-of-the-art performance of our hand motion capture system in public benchmarks, and show the high quality of our whole body motion capture result in various challenging real-world scenes, including a live demo scenario. [2]

1. Introduction

Billions of daily human activities are being recorded as videos and uploaded to public internet websites, capturing extremely diverse human behaviors in various real-world scenarios. A technology that can digitize the human motions from these videos has enormous potentials in various applications including human-computer interaction, social artificial intelligent, and robotics. A motion capture system with a commodity camera and reduced computation would enable people to make full use of such applications.

While the hands and body are equally important for human motion understanding toward these applications, the hands are physically small body parts, making it difficult to capture the motion of the hands and body jointly even with a professional motion capture system. The same is true for recent work on 3D body pose (*i.e.*, torso and limbs) estimation from a single RGB image [5, 20, 55, 25, 26]. Although the accuracy improvement of body pose estimation is significant, subtle finger gestures are ignored, losing the original nuance of the human motion. Similarly, there have been noticeable achievements in 3D hand pose estimation from a depth input [34, 48, 45, 47, 53, 58] or a single RGB image [62, 9, 15, 8, 11, 61]. However, these approaches are often demonstrated with hand-specific camera views, rather than the more challenging in-the-wild scenarios where a camera is capturing whole bodies of people and the hands

[1]"FrankMocap" is an homage to the Frankenstein's monster in *The Modern Prometheus.*

[2]Code and models are available at https://penincillin.github.io/frank_mocap.

tend to be in low resolutions with frequent motion blurs. A few recent approaches aim to capture 3D motions of the whole body (*i.e*., the body and hands) by leveraging the 3D parametric models that can express both hands and body [19, 55, 37]. However, these approaches rely on optimization techniques to fit the parametric models to image measurements, which are relatively slow and not suitable for real-time applications.

In this paper, we present a fast and accurate motion capture method to estimate both 3D body and hand poses from monocular RGB images or videos, as shown in Figure 1. Our method consists of two regression modules that predict 3D poses of the body and hands individually from a single RGB image input, followed by an integration module that produces the whole body pose from the outputs from the body and hand modules [3]. A main idea of our approach is to make the outputs from body module and hand module as compatible as possible, enabling us to efficiently integrate the outputs for whole body motion capture. To make this integration process tractable, we employ the SMPL-X model [37] that has a unified skeleton and mesh representation for both the body and hands. Based on that, the body module and hand module contribute the different part of the same output structures. Inspired by the achievements of the recent work on 3D body motion estimation [20, 25], the hand and body modules are designed based on deep neural network techniques, and directly regress 3D poses from an input single RGB image. Given the 3D body and hand pose estimations from the hand and body modules, the integration can be performed by direct *copy-and-paste* of the predicted body and hand poses to the SMPL-X model, achieving a near real-time performance for whole body 3D motion capture ($\sim$9.5 fps). A further improvement for better estimate of the whole body pose can be achieved via the optimization framework we also present. Follow the similar spirit of Joo *et al*. [19], we call our method as FrankMocap to represent this regression and integration manner.

We demonstrate the fast and accurate performance of FrankMocap on various real-world monocular videos, including a real-time demo. Notably, our 3D hand pose estimation outperforms previous approaches in public benchmarks. We also present thorough ablation studies to demonstrate the advantage of our method, and compare our method with previous hand motion capture methods and whole body motion capture methods.

2. Related Work

3D Parametric Human Body Models. 3D Parametric human body models are widely used for markerless motion capture, to model the deformation of 3D human (including body, face and hands) via low dimension parameters instead of the original vertices [3, 40, 37, 42, 19]. The SCAPE is a pioneering work that accounts for shape variations and pose deformations [3]. After that, Loper *et al*. introduce the SMPL [28] that learns local pose-dependent blendshape on top of linear blend skinning for holistic mesh deformation as well as shape variations. Later, Romero *et al*. [42] extend the SMPL to a hand and introduce the hand deformation model called MANO. They also develop a unified body and hand model called SMPL+H. Joo *et al*. build a unified model of body, hands and face, called Adam, and use it to achieve whole body motion capture of face, body and hands from a multiview setup [19]. Pavlakos *et al*. [37] also similarly develop a unified model of body, hands and face, called SMPL-X, in which all template body parts are designed by artists for consistent quality over the body parts, and learned deformation statistics.

Single Image 3D Body Pose Estimation. Many monocular 3D body pose estimation approaches consider to predict 3D body keypoint locations from single images [41, 50, 52, 29, 38]. As the major limitations, the output of these methods cannot be directly used for graphics applications, since 3D joint angles are missing and the lengths of parts are not preserved. More recent monocular 3D body pose estimation approaches adopt parametric 3D human model such as SMPL [28] or Adam model [19, 55] for 3D body representation. The use of 3D parametric models allows them to reconstruct a 3D body pose by fitting the 3D body model to 2D observations, such as 2D keypoints, via optimization framework [5]. More recent work [20, 25, 52, 43, 26, 56] leverages the deep learning framework to directly regress parameters of a body model.

There are also approaches that use a hybrid framework by using a deep learning framework to produce an intermediate representation such as 2D and depth heat maps and fitting a skeletal model on these outputs to reconstruct joint angles [31, 55, 30]. Various types of inputs are considered in these deep network approaches, including single RGB images [20], 2D keypoint heatmaps [39], body part segmentation [35] or densepose maps [44, 56]. Due to the lack of training data with 3D annotations, these models are trained with mixed datasets including indoor datasets such as Human3.6M [14] and in-the-wild datasets such as COCO [27] or 3DPW [54]. Most papers in this area such as HMR [20] and SPIN [25] use single images as input. There are some other works takes sequences as input. The representative ones are Zhang *et al*. [60] and VIBE [24]. In this paper, we mainly focus on processing single images and applying simple temporal smoothness to handle sequence inputs.

Single Image 3D Hand Pose Estimation. Previous works on 3D hand joint estimation takes depth images as in-

[3] We use *whole body* to represent all the body parts including the fingers, while we use *body* to represent the torso and limbs excluding the fingers.

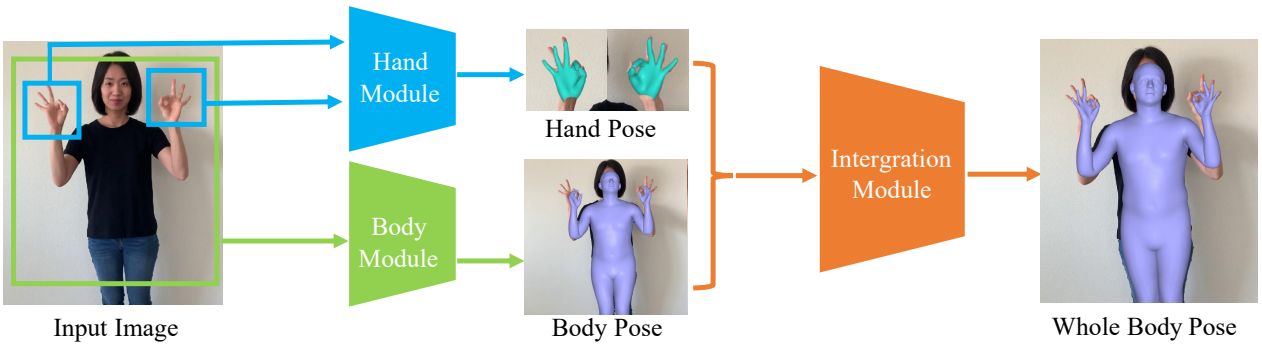


Figure 2: Overview of our pipeline for whole body motion capture. Given a single RGB image input, we apply our hand module and body module to estimate 3D hands and 3D body. Our integration module combines these outputs into a unified whole body output.

put [33, 34, 48, 45, 47, 49, 53, 51, 58]. Although achieving good performance, these methods cannot be easily applied to in-the-wild RGB images and videos. Recent work begins to use single RGB images as input [62, 32, 9, 57]. These approaches focus more on 3D hand joints location estimation instead of joint angles. Inspired by recent success in 3D body motion capture [20, 25], there are several methods on single image 3D hand pose estimation. Boukhayma *et al*. [8] uses images and 2D pose predicted from OpenPose [10] as input and regress the parameters of the MANO model [42]. Zhang *et al*. [61] share a similar framework as Boukhayma. The key difference is that their model contains a 2D heatmap prediction module, instead of using predictions from OpenPose. Baek *et al*.'s method [4] also has similar architecture. The difference lies in that they additionally adopt 2D masks as an intermediate representation. Different from these approaches, the work of Ge *et al*. [11] use a self-created 3D hand model instead of MANO. The proposed framework takes single images as input and predicts 2D heatmaps as intermediate representation. After that, graph convolutional network [23] is used to regress the vertices of the hand model.

Joint 3D Pose Estimation of Body and Hands. There are a few methods [55, 37] that pursue to estimate 3D poses of body and hands together. Due to the lack of annotated data for whole body capture, all these previous works resort to optimization methods. SMPLify-X [37] uses the SMPL-X model [37] to represent a whole body pose. The model parameters are optimized by fitting to 2D keypoints with additional constraints including body pose priors and collision penalizer. Monocular Total Capture (MTC) [55] is based on the Adam model [19]. It 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 with relatively slow computation time (from 10 seconds to a few minutes). Besides, when 2D heatmap detection failed, their accuracy degrades significantly.

3. Method

Our method aims to estimate 3D body (the torso and limb parts) and 3D hands (both left and right) from monocular inputs (either monocular images or videos). Our method produces, as output, the parameters of the SMPL-X model [37] to represent both 3D body and hand poses in a unified form. An important aspect of our method is to use separate expert modules for each body and hand pose estimation while both modules produce the compatible outputs as part of SMPL-X model. An overview of the framework is shown in Figure 2.

Notably, our hand pose estimator leverages the hand part of the SMPL-X model, by treating it as a stand-alone parametric hand model. While still showing the state-of-the-art monocular hand pose estimation performance, the output of our hand module can be directly merged to the body estimation output [25], to pose the whole body SMPL-X model. We also present an optimization framework to improve the hand and body pose estimation output with additional constraints for better accuracy. In the remaining part of this section, we assume the inputs to our model are single images. Details of processing sequence inputs are included in section 4.1.

3.1. Overview of SMPL-X Model

Given a single image input cropped around a single person, our method produces whole body motion capture output as a form of shape and pose parameters of the SMPL-X model. As an extension of the SMPL model [28], the SMPL-X model can represent the shape variations and pose-dependent deformation of human bodies via a combination of low-dimensional shape and pose parameters. As a key difference from the SMPL model that only focuses on body parts, SMPL-X can also express finger motions and facial expressions, by including additional sets of parameters for them.

We formulate the SMPL-X model, denoted by W, as:

$$\boldsymbol{V}_w = W(\boldsymbol{\phi}_w, \boldsymbol{\theta}_w, \boldsymbol{\beta}_w), \quad (1)$$

where W is parameterized by global orientation of the whole body $\boldsymbol{\phi}_w \in \mathbb{R}^3$, whole body pose parameters $\boldsymbol{\theta}_w \in \mathbb{R}^{(21+15+15)\times 3}$ accounting for pose-dependent deformation, and shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$ accounting for cross-identity shape variations of the body and hands. We divide $\boldsymbol{\theta}_w$ for each of the body and hands, namely body pose parameters $\boldsymbol{\theta}_w^b \in \mathbb{R}^{21\times 3}$, left hand pose parameters $\boldsymbol{\theta}_w^{lh} \in \mathbb{R}^{15\times 3}$ (*i.e.*, 3 joints per finger), right hand pose parameters $\boldsymbol{\theta}_w^{rh} \in \mathbb{R}^{15\times 3}$, and thus $\boldsymbol{\theta}_w = \{\boldsymbol{\theta}_w^b, \boldsymbol{\theta}_w^{lh}, \boldsymbol{\theta}_w^{rh}\}$. [4] All pose parameters are defined in the angle-axis 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, $\boldsymbol{V}_w \in \mathbb{R}^{10475\times 3}$. The 3D joint locations of the whole body can be obtained by applying a joint regression function R from the posed vertices:

$$\boldsymbol{J}_w^{3D} = R_w(\boldsymbol{V}_w), \quad (2)$$

where $\boldsymbol{J}_w^{3D} \in \mathbb{R}^{(22+15+15)\times 3}$.

Our hand models are defined by taking the hand parts of SMPL-X:

$$\boldsymbol{V}_h = H(\boldsymbol{\phi}_h, \boldsymbol{\theta}_h, \boldsymbol{\beta}_h), \quad (3)$$

where $\boldsymbol{\theta}_h \in \mathbb{R}^{3\times 15}$ is hand pose parameters and $\boldsymbol{\beta_h}$ is the shape parameters for hand model. Since our hand model is taken from SMPL-X, $\boldsymbol{\beta_h}$ shares the same parameterization space as β_w. For hand model H, we only focus on its influence on the hand part. For brevity, we use $\boldsymbol{\theta}_h$ to denote the hand pose parameters instead of $\boldsymbol{\theta}_{rh}$ or $\boldsymbol{\theta}_{lh}$ to describe our hand pose estimation method. $\boldsymbol{\phi}_h \in \mathbb{R}^3$ represents the global orientation of the hand meshes, which is necessary to use hand model as a stand-alone model, independent from the ancestry joints of the original SMPL-X model W. Our hand model H produces the hand mesh structure with 778 vertices, $\boldsymbol{V}_h \in \mathbb{R}^{778\times 3}$. Here, we define the hand mesh vertices from the whole body mesh $\boldsymbol{V}_w$ by cropping the vertices around the wrist area, where we choose all vertices from which any wrist joint and finger joints are closest. Since the hand mesh $\boldsymbol{V}_h$ is a subset of the whole body mesh $\boldsymbol{V}_w$, the exact vertex correspondences are known. We also consider the 3D joint regression function for hand $\boldsymbol{J}_h^{3D}$:

$$\boldsymbol{J}_h^{3D} = R_h(\boldsymbol{V}_h), \quad (4)$$

where $\boldsymbol{J}_h^{3D} \in \mathbb{R}^{21\times 3}$ contains a wrist, 15 finger joints (3 joints per finger), and 5 finger tips. To define the joint regression matrix of R_h, we take the wrist and finger joint parts from the whole body joint regression matrix R_w, and define additional rows for the 5 finger tips that are not defined in $\boldsymbol{J}_w^{3D}$. See Figure 4 for visualization of H and skeleton hierarchy.

[4] Note that we ignore other parameters of the original SMPL-X model, including facial expression parameters.

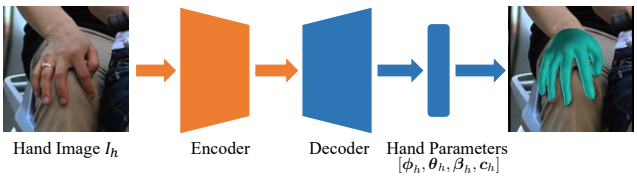


Figure 3: **Overall framework of our hand module.** Our hand module takes a cropped hand image $\mathbf{I}_H$ as input, and produces the parameters of hand model, $[\boldsymbol{\phi}_h, \boldsymbol{\theta}_h, \boldsymbol{\beta}_h, \boldsymbol{c}_h]$. Our hand module is built by a deep encoder-decoder network. The predicted hand parameter is used to produce the mesh shape and pose of the hand part of SMPL-X.

The major advantage of our representation is that the components of 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.

3.2. 3D Hand Estimation Module

We present a monocular 3D hand pose estimation module, denoted by M_H, estimating the parameters of the hand model H. In particular, our hand module is inspired by the recently proposed monocular body pose estimation approaches [20, 21, 25], thus follows the similar model architecture, parameterizations, and training stages. Leveraging the achievement in body pose estimation area, we found that our hand pose estimation method can be robustly applicable for various in-the-wild situations, showing the state-of-the-art performance in public hand pose estimation benchmarks.

Hand Module Architecture. Our hand module M_H is built upon an end-to-end deep neural network architecture to regress the hand pose parameters defined in Eq. (3). Our hand module M_H is defined as:

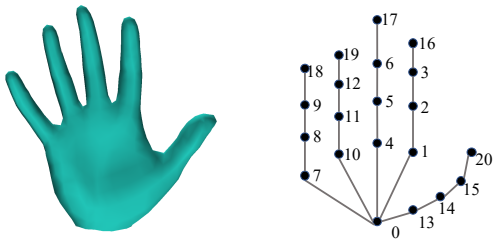
$$[\boldsymbol{\phi}_h, \boldsymbol{\theta}_h, \boldsymbol{\beta}_h, \boldsymbol{c}_h] = M_H(\mathbf{I}_H), \quad (5)$$

where $\mathbf{I}_H$ is an input RGB image cropped around a hand region. $\boldsymbol{c}_h = (\boldsymbol{t}_h, s_h)$ is weak-perspective camera parameters which allows to project a posed 3D hand model to an input image. Here, $\boldsymbol{t}_h \in \mathbb{R}^2$ is for for 2D translation on the image plane, and $s_h \in \mathbb{R}$ is a scale factor. Thus, the i-th 3D hand joint,$\boldsymbol{J}_{h,i}^{3D}$ can be projected as:

$$\boldsymbol{J}_{h,i}^{2D} = \boldsymbol{s}_h \Pi(\boldsymbol{J}_{h,i}^{3D}) + \boldsymbol{t}_h, \quad (6)$$

where Π is an orthographic projection.

Following the body pose estimation approaches [20, 25], the architecture of our hand module M_H is composed of an



(a) SMPL-X's Hand (b) Hand Skeleton Hirerarchy

Figure 4: Our hand model taken from SMPL-X. We take the hand part of SMPL-X as a stand-alone hand model for hand pose estimation. The example mesh is shown in (a) and the skeleton hierarchy is shown in (b).

encoder and a decoder structure, where the encoder outputs the encoded features from input images, and the decoder regresses the hand pose parameters from the features. See Figure 3 for the overview of our hand module. We use the ResNet-50 [13] for the encoder network. The decoder network is composed of a group of fully connected layers. Our hand module is trained with the data for the right hand. The images and annotations for the left hand are used after vertical flipping. During the testing time, the left hand images are flipped and processed as if they were a right hand, and their outputs are flipped back to the original left hand space.

Note that the shape parameter $\boldsymbol{\beta}_h$ is originally defined for whole body model $\boldsymbol{\beta}_w$, but we only consider the deformation for the hand vertices defined in 3, ignoring the body part. We describe how this can be handled in our integration module.

Training Method.

We consider three different types of annotations: (1) 3D pose annotations (in angle-axis representation), (2) 3D keypoint (joint) annotations, and (3) 2D keypoint annotations. The losses for each of the annotations, namely $L_{\boldsymbol{\theta}}$, L_{3D} and L_{2D}, are defined as follows:

$$\begin{aligned} L_{\theta} &= \|\boldsymbol{\theta}_h - \hat{\boldsymbol{\theta}}_h\|_2^2, \\ L_{3D} &= \|\boldsymbol{J}_h^{3D} - \hat{\boldsymbol{J}}_h^{3D}\|_2^2, \\ L_{2D} &= \|\boldsymbol{J}_h^{2D} - \hat{\boldsymbol{J}}_h^{2D})\|, \end{aligned} \tag{7}$$

where $\hat{\boldsymbol{\theta}}_h$, $\hat{\boldsymbol{J}}_h^{3D}$ and $\hat{\boldsymbol{J}}_h^{2D}$ are the ground-truth annotations of angle-axis pose parameters, 3D keypoints, and 2D keypoints. In particular, the 2D keypoint loss L_{2D} is necessary to estimate the camera projection parameters. We do not use the shape parameters provided by the 3D hand datasets such as FreiHAND [63], since these are defined for the MANO model [42] and not compatible with our hand model from SMPL-X. Instead, an additional shape parameter regularization loss L_{reg} is applied:

$$L_{reg} = \|\boldsymbol{\beta}_h\|_2^2. \tag{8}$$

The overall loss L used to train our hand module is defined as follows:

$$L = \lambda_1 L_{\boldsymbol{\theta}} + \lambda_2 L_{3D} + \lambda_3 L_{2D} + \lambda_4 L_{reg}. \tag{9}$$

In experiments, the balanced weights are set as $\lambda_1 = 10$, $\lambda_2 = 100$, $\lambda_3 = 10$ and $\lambda_4 = 0.1$.



Figure 5: Motion Blur Augmentation. We show example images of motion blur augmentation. From left to right: original images, augmented images after applying different motion blur kernels.

Datasets Preprocessing. 3D hand pose datasets are often built by multi-view setups in controlled environments to obtain ground-truth annotations. A model trained with these datasets often suffer from overfitting, showing limited performance when applied to outdoor in-the-wild data. Notably, recent 3D body pose estimation approaches have shown that leveraging diverse datasets can greatly improve its generalization ability [20, 21, 44, 25]. Following this, we include as many publicly available datasets as possible towards in-the-wild 3D hand pose estimation. More details of the datasets are discussed in section 4.2. The major challenge in using diverse datasets is that their annotation types vary. For example, there exist available ground-truth joint angle parameters in FreiHand [63] and HO-3D [12] datasets, while others do not contain it. Furthermore, the details of hand annotations including skeleton hierarchy and scales are also different across datasets. To handle this, we perform several pre-processing steps to make them

consistent and compatible with our hand model, including 1. Rescaling all 3D keypoint annotations to be compatible with our hand model, by using the middle finger's knuckle length[5] as a reference. 2. Re-ordering the 3D keypoints joints to be the same as our hand model's skeleton hierarchy shown in Figure 4.

Training Data Augmentation. Performing data augmentations during training is a common practice to enable model with better generalization ability. Following previous approaches [61], we 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 additional apply motion blur augmentation to the images. We first use the methods in previous papers [6, 7] to generate blur kernels and then use 2D filtering to add blurriness to images. The experiments show that our motion blur augmentation is beneficial to generalize our hand module for in-the-wild scenes. Examples of motion blur augmentation are shown in Figure 5.

3.3. 3D Body Estimation Module

We leverage the state-of-the-art monocular 3D pose estimation methods [20, 25] with a few modifications for our body estimation module. The recent monocular body pose estimation methods [20, 25] are based on the SMPL [28] model to capture torso and limb motion. Although our body module produces similar outputs, these previous methods cannot be directly applicable for our objective, since SMPL model's shape parameters are not compatible with SMPL-X. Thus, we fine-tune the publicly available state-of-the-art pose estimator [25] by replacing SMPL with SMPL-X in the training pipelines. For training, we use the publicly available indoor 3D pose datasets such as human3.6M [14]. We also include the pseudo-ground truth annotations introduced in [17] that provides SMPL fitting paired with the in-the-wild 2D keypoint datasets (*e.g.*, COCO [27] and MPII [2]). Since all these annotations are in SMPL format, we ignore the shape parameters of ground truth SMPL annotations, and only use the pose parameters and 2D keypoint annotations that are compatible to SMPL-X model. We follow the same neural network architecture with the similar training steps as in the work of [26], without using the SMPLify part.

Our body module M_B produces the torso and limb parameters defined in Eq. 1 from an single image:

$$[\boldsymbol{\phi}_b, \boldsymbol{\theta}_b, \boldsymbol{\beta}_b, \boldsymbol{c}_b] = M_B(\boldsymbol{I}_b), \tag{10}$$

where $\boldsymbol{I}_b$ is an input image cropped around a target single person's whole body. Similar to Eq. 1, $\boldsymbol{\phi}_b \in \mathbb{R}^3$ is the global body orientation, $\boldsymbol{\theta}_b \in \mathbb{R}^{21\times 3}$ is the body pose parameters (without any hand joints), and $\boldsymbol{\beta}_b \in \mathbb{R}^{10}$ is the shape parameter. Again, $\boldsymbol{\beta}_b$ shares the same parameterization space as $\boldsymbol{\beta}_w$, which is defined in Eq. 1. Similar to Eq. 5, we use weak perspective camera parameters $\boldsymbol{c}_b = (\boldsymbol{t}_b, \boldsymbol{s}_b)$.

Note that the existing body pose estimators including our fine-tuned version do not accurately estimate the the wrist and arm orientation due to inaccurate or insufficient annotations (*e.g.*, only one keypoint is annotated for a wrist), as shown in Figure 6. Our integration module solves this issue.

3.4. Whole Body Integration Module

Our integration module combines the outputs from the 3D body and hand modules into a unified representation as a form of SMPL-X model. For the integration, we present two strategies: (1) a fast method by simple copy-and-paste composition, and (2) an optimization framework to include additional 2D keypoint cues for more accurate output.

Fast Body and Hand Composition by Copy-and-Paste. Since the outputs from our hand and body modules are compatible with the SMPL-X model, they can be easily combined as the single form. A simple strategy is just transferring the corresponding joint angle parameters from the outputs of each of the hand and body modules. However, the wrist parts require additional processing, because we obtain two different outputs from the body and hand modules (represented by the global hand orientation $\boldsymbol{\phi}_h$). Let us denote the pose parameters for the wrist joint as $\boldsymbol{\theta}^{\text{wrist}}$, then $\boldsymbol{\theta}_b = \tilde{\boldsymbol{\theta}}_b \cup \{\boldsymbol{\theta}_b^{\text{rwrist}}, \boldsymbol{\theta}_b^{\text{lwrist}}\}$, where $\tilde{\boldsymbol{\theta}}_b$ includes all body pose parameters except wrists. We use the similar notations for the whole body pose parameters, $\boldsymbol{\theta}_w^{\text{rwrist}}$, $\boldsymbol{\theta}_w^{\text{lwrist}}$, and $\tilde{\boldsymbol{\theta}}_w^b$. Then, whole body integration by copy-and-paste can be performed as:

$$\begin{aligned} \boldsymbol{\phi}_w &= \boldsymbol{\phi}_b, \\ \boldsymbol{\beta}_w &= \boldsymbol{\beta}_b, \\ \boldsymbol{c}_w &= \boldsymbol{c}_b, \end{aligned} \tag{11}$$

$$\begin{aligned} \left(\tilde{\boldsymbol{\theta}}_w^b, \boldsymbol{\theta}_w^{lh}, \boldsymbol{\theta}_w^{rh}\right) &= \left(\tilde{\boldsymbol{\theta}}_b, \boldsymbol{\theta}_{lh}, \boldsymbol{\theta}_{rh}\right), \\ \left(\boldsymbol{\theta}_w^{\text{lwrist}}, \boldsymbol{\theta}_w^{\text{rwrist}}\right) &= \left(\Gamma_l\left(\boldsymbol{\theta}_b, \boldsymbol{\phi}_{lh}\right), \Gamma_r\left(\boldsymbol{\theta}_b, \boldsymbol{\phi}_{rh}\right)\right), \end{aligned} \tag{12}$$

where Γ_l and Γ_r are the functions to convert the global wrist orientation $\boldsymbol{\phi}_h$ 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 $\boldsymbol{\phi}_h$ with the global orientation of the current wrist pose from $\boldsymbol{\theta}_b$ that can be computed by following the forward kinematics of the body skeleton hierarchy. This strategy requires almost no extra computation, making

[5] the skeleton between 4-th and 5-th joints efined in Figure 4.

our separate modules to 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.

Hand and Body Composition via Optimization. As an alternative integration method, 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-and-paste strategy, and also can take advantage from the 2D keypoint estimation output [10] for better 2D localization quality. In particular, our optimization framework finds the whole body model parameters that minimize the following objective cost function:

$$\mathcal{F}([\boldsymbol{\phi}_w, \boldsymbol{\theta}_w, \boldsymbol{\beta}_w, \boldsymbol{c}_w]) = \mathcal{F}^{2d} + \mathcal{F}^{pri}, \qquad (13)$$

where $\mathcal{F}^{2d}$ is the 2d reprojection cost term between the 2D keypoint estimation [10] and the projection of 3D joints (body and both hands), and the prior term $\mathcal{F}^{pri}$ is needed to keep the 3D pose and shape parameters in plausible space, as in SMPLify method[5]. We first initialize all parameters by our copy-and-paste strategy except that we do not apply Γ to transfer the global hand orientation to whole body model. Instead, the wrist orientations of the hands can be obtained by minimizing the Eq. (13) with other parameters. While a Gaussian mixture model learnt from motion capture dataset [1] is often used for the body pose prior term as in SMPLify method [5], we use the the exemplar fine-tuning approach introduced in [17] for the similar goal, by applying neural network fine-tuning of M_B for each frame independently, which does not require additional regularization term but still keep the 3D pose in plausible space. Note that our optimization framework requires only a few iteration (20 iterations in all our experiments), since outputs from the body and hand modules output is already close to the target status. See Figure 6 for the example of our optimization.

4. Experiments

In this section, we first describe the implementation details. Then we summarize the datasets used for our hand module training. After that, we quantitatively and qualitatively compare our methods with the state-of-the-art approaches. We also perform ablation studies to examine the key designs of our methods.

4.1. Implementation Details

Bounding Boxes. For the online version, we use OpenPose [10] to obtain body bounding boxes. After processing the body, the hand bounding boxes are obtained by projecting the hand part of the estimated 3D body to image space. For the offline processing of internet videos, we use OpenPose detections to localize both bodies and hands.

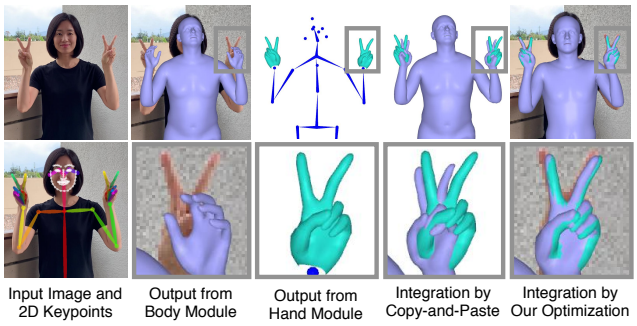


Figure 6: Optimizing the whole body model (SMPL-X) with 3D hand prediction and 2D keypoint estimation. (a) An input image and the estimated 2D keypoints by OpenPose [10]; (b) 3D body pose estimation from our body module; (c) The output of 3D hand module aligned to the wrist joints of SMPL-X; (d) Integration output by copy-and-paste strategy; (e) Integration output by our optimization framework.

Table 1: Processing time of various methods.

Method → Time (fps) ↓	SMPLify-X	MTC	Online (CP)	Offline (CP)	Offline (OP)
Preprocess (fps)	7.5	7.5	35	7.5	7.5
Model (fps)	0.01	0.1	13	13	1.1
Overal (fps)	0.01	0.1	9.5	4.7	0.95

Video Processing. For copy-and-paste strategy (used in online demo and offline internet videos), the videos are processed frame-by-frame without any post-processing. For optimization-based strategy, after obtaining per-frame outputs, we apply a naive temporal smoothing for each separate dimension of parameters (shape, pose, and camera). We use a 5-frame-size smoothing kernel with the weight $[0.1, 0.2, 0.5, 0.2, 0.1]$. It is noted that our copy-and-paste method can generate temporally-stable results even without smoothness. We believe it is due to the fact that recent CNN pose regressors tend to produce such output as demonstrated in recent papers (*e.g.* SPIN [26]), thanks to multiple augmentation tricks in training. The optimization-based method (SMPLify-X and MTC) suffers from temporal instability due to the complicated optimization procedures with multiple-stages (*e.g.* torso first and others later) and elaborated balancing issues between data term and prior term. The processing time of each method are compared in Table 1. The processing time of our copy and paste method is about 9.5 fps, where the code is implemented in python and runs in a single GeForce RTX 2080 GPU. In our supplementary video, we also show a live demo using a single webcam, which cannot be performed by alternative approaches.

Hand Module. Input images of the hand module are center-cropped surrounding the hands, where the bounding boxes for cropping are given by 2D hand keypoints. Ground-

truth 2D keypoints are used for training time, the predicted keypoints from OpenPose [10] are used for testing time. The cropped images are further padded and resized to size of 224×224. During training, we apply data augmentations to each of training images via random scaling, translation, rotation, color jittering, and synthetic motion blur. The hand module architecture is based on ResNet-50 [13] with two additional fully connected layers to map the output features of ResNet to vectors with 61 dimension, which is composed of camera parameters C (3 dimensions), hand global rotation ϕ_h (3 dimensions), hand pose parameters θ_h (45 dimensions) and shape parameters β_h (10 dimensions). The hand module is implemented with PyTorch [36]. The Adam optimizer [22] with learning rate $1e{-}4$ is used to train the model. The hand module is trained until converge, which takes about 100 epochs.

Body Module. We follow the similar training steps to the state-of-the-art method [25] using the Human3.6M [14] and COCO datasets [27] with the pseudo SMPL annotations [17]. Our training starts from the pre-trained model of SPIN [25] with substituting the SMPL parameters to the SMPL-X parameters. The model is then finetuned using the Adam optimizer [22] with learning rate $5e{-}5$ for about 20 epochs. Besides, we use neutral SMPL-X model for both hand and body module.

4.2. Datasets

FreiHAND. FreiHAND [63] is a dataset with ground truth 3D hand joints and MANO parameters for real human hand images. The 3D annotations are obtained by a multi-camera system and a semi-automated approach. The obtained data is further augmented with synthetic backgrounds. In our experiments, we randomly select 80% of samples from original training set as training data and use the remaining 20% of samples for validation.

HO-3D. HO-3D dataset [12] is a dataset aiming to study the interaction between hands and objects. The dataset has 3D joints and MANO pose parameters for hands, and also has 3D bounding boxes for objects the hands interact with. In this paper, we only use 3D annotations of hands. The training set is composed of different sequences, each of which records one type of hand-object interaction. Following the similar practice in processing FreiHAND, we randomly choose 80% of sequences from the original training set as training data and use the remaining 20% of sequences for validation.

MTC. Monocular Total Capture [55] is a dataset captured by Panoptic Studio [16, 18] in a multi-view setup with 30 HD cameras. It has 3D hand joints annotations for both body and hands. The sequences are mainly the range of motion data of multiple subjects. To polish the dataset, we filter out erroneous samples where hands are not visible or too small.

Table 2: Comparison of our hand module with the state-of-the-art hand methods on three public benchmarks, namely STB, RHD and MPII+NZSL. For STB and RHD, we use 3D AUC as the evaluation metric. The threshold ranges from 20mm to 50mm. For MPII+NZSL, we use 2D AUC as the evaluation metric. The threshold ranges from 0px to 30px.

Dataset → Method ↓	STB	RHD	MPII+NZSL
[62]	0.948	0.675	0.171
[8]	0.994	-	0.501
[4]	-	0.926	-
[11]	**0.995**	0.92	0.15
[55]	0.994	-	0.340
[61]	**0.995**	0.901	-
Ours-less-datasets	0.992	0.918	0.556
Ours-no-data-augment	0.991	0.893	0.608
Ours-no-shape-params	0.987	0.910	0.647
Ours	0.992	**0.934**	**0.655**

Table 3: Ablation study on dataset. We show the results of our hand module trained with different datasets. These models are evaluated on MPII+NZSL [46] using 2D AUC as metric. For data augmentation, we use all the available datasets.

FreiHAND	HO-3D	MTC	STB	RHD	MPII+NZSL
✓					0.482
	✓				0.367
✓	✓				0.526
		✓	✓	✓	0.556
✓	✓	✓			0.595
✓	✓	✓	✓		0.598
✓	✓	✓		✓	0.645
✓	✓	✓	✓	✓	0.655

STB. Stereo Hand Pose Tracking Benchmark [59] is composed of 15,000 training samples and 3,000 testing samples. The provided annotations include 3D joints and depth images. In our experiments, we use 3D joints only. We use training set of STB to train our model and compare with other state-of-the-art methods on the validation set. To unify definition of joints, following the practice of [9, 11], we move the root joint from palm center to wrist.

RHD. Rendered Hand Dataset [62] is a synthetic dataset that has 2D and 3D hand joint annotations. It is composed of 41,258 training samples and 2,728 testing samples. We train our model on the training set and compare with other state-of-the-art methods on the testing set.

MPII+NZSL. MPII+NZSL dataset [46] is composed of in-the-wild images with manually annotated 2D hand joints. It includes challenging images with occlusion, blur, and low resolution. To show our models' generalization ability, our models are no trained on the MPII+NZSL, we only use it for validation.

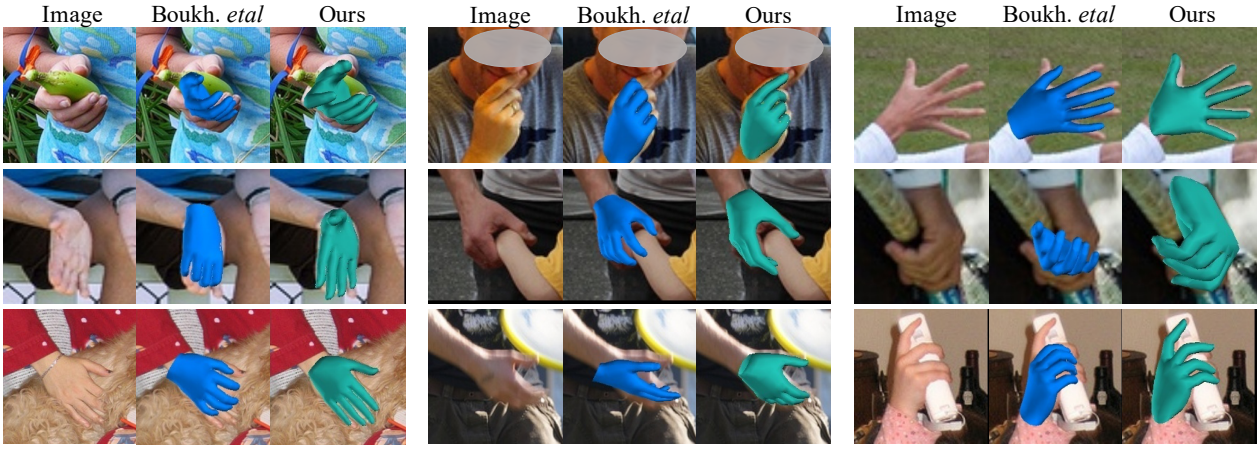


Figure 7: Qualitative comparison with State-of-the-are methods. The images are selected from COCO dataset [27]. We qualitatively compare our models performance with Boukhayma *et al*. [8].

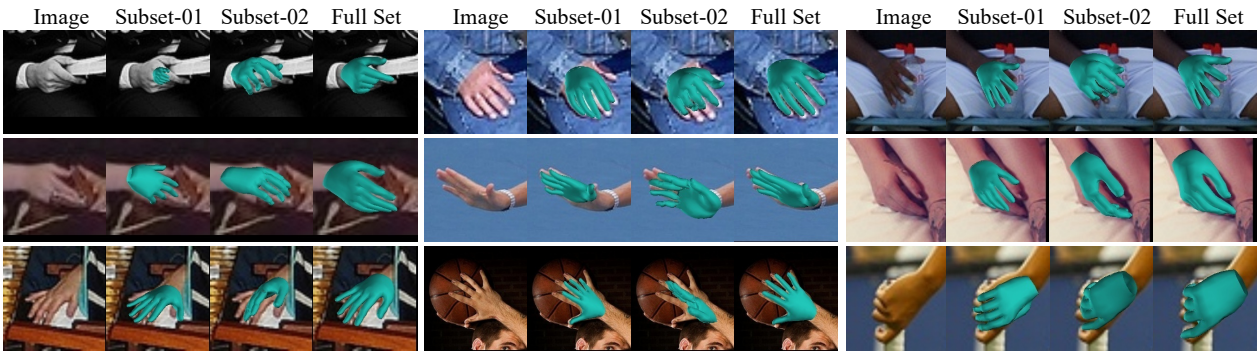


Figure 8: Ablation study on training dataset. We show qualitative ablation study on using different datasets in training our hand model. "Subset-01" means using the combination of datasets FreiHADN [63] and HO-3D [12]. "Subset-02" means using the combination of datasets: STB [59], RHD [62] and MTC [55]. "Full set" means using all the above datasets. The images are selected from COCO dataset [27].

Table 4: Ablation study on data augmentation. We show the results of our hand module trained with different data augmentation strategies. These models are evaluated on MPII+NZSL [46] using 2D AUC as metric.

Position	Rescale	Color Jittering	Rotation	Motion Blur	MPII + NZSL
✗	✗	✗	✗	✗	0.608
✓	✓				0.610
✓	✓	✓			0.618
✓	✓	✓	✓		0.622
✓	✓	✓	✓	✓	0.655

4.3. 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, STB [59], RHD [62] and MPII+NZSL [46]. 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 [59] and RHD [62], we use 3D AUC and the threshold ranges from 20mm to 50mm. For MPII+NZSL [46], we use 2D AUC and the threshold ranges from 0px to 30px.

The results are listed in Table 2. For fair comparison, all the methods takes single RGB image as input. "Ours" refers to our best model trained with all the datasets and all data augmentation srategies. It outperforms previous methods on RHD and MPII+NZSL and shows a comparable performance in STB. Notably, our method shows significantly better 2D localization accuracy on challenging in-the-wild dataset MPII+NZSL, demonstrating its generalization ability to in-the-wild scenarios.

We also compare our own best model with variants of our method. "Ours-no-shape-params" differs from "Ours" in that the shape parameters β are not used and fixed to zero. "Ours-no-data-augment" refers to the model trained without using any data augmentation. "Ours-less-datasets" refers to the model trained without using latest datasets, namely

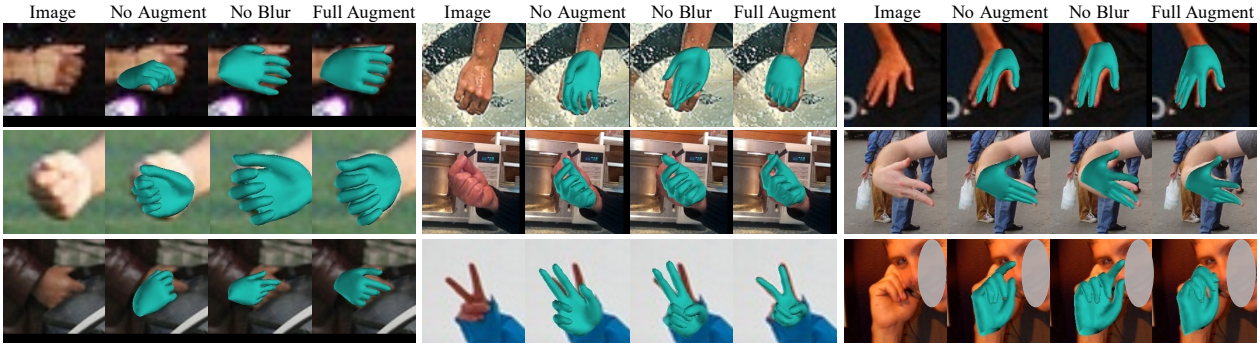


Figure 9: Ablation study on data augmentation. We show qualitative ablation study on training our hand model using different data augmentation. "No Blur" refers to model trained with all data augmentation strategies except motion blur augmentation. "Full Augment" refers to model trained with all data augmentation strategies. The images are selected from COCO dataset [27].

FreiHAND and HO-3D. This model is trained with MTC, STB and RHD. These datasets are also used by previous methods.

Comparison between the variants of our model verifies that including various datasets and applying data augmentation are important in achieving better results. Inferring shape variations by estimating shape parameters is also helpful to improve the accuracy. Note that the results of "Ours-less-datasets" show that our model still achieves comparable performance on STB and RHD with the previous methods by using only limited datasets, and shows better 2D localization accuracy on in-the-wild MPII+NZSL dataset. These results demonstrate that our method takes advantage from both network design (including the training strategy) and larger training datasets.

We also qualitatively compare our method with previous work, as shown in Figure 7 and our supplementary video. The results indicate that our hand model can generate more precise 3D hand poses under challenging in-the-wild scenarios with occlusion, blur and low resolution.

Ablation Study. We further examine two key designs used in training our hand module, the mixture of different datasets and data augmentation. The results for ablation study on the datasets are listed in Table 3 and Figure 8. As expected, the results in Table 3 shows that using more datasets will lead to better performance. We also show the examples of qualitative comparison in Figure 8. Similar to the conclusion from the quantitative study, the qualitative results show that incorporating more datasets can increase the models' generalization ability and generate more precise results for in-the-wild images. In the figure, "Subset-01" means using the combination of datasets FreiHADN [63] and HO-3D [12]. "Subset-02" means using the combination of datasets: STB [59], RHD [62] and MTC [55]. "Full set" means using all the datasets.

The results for the ablation study on data augmentation are listed in Table 4 and Figure 9. The results in Table 4 demonstrate that applying data augmentation leads to better results. We also show qualitative results in Figure 9, where, by adopting data augmentation, our models can generalize better to challenging scenarios including blur, challenging poses and occlusion. In Figure 9, "No Augment" refers to model trained without any data augmentation, and "No Blur" refers to model trained with all data augmentation strategies except motion blur augmentation. "Full Augment" refers to model trained with all data augmentation strategies.

4.4. Integration Module Evaluation

We qualitatively compare our method with previous whole body motion capture approaches, MTC [55] and SMPLif-X [37]. We also compare between two version of our model. We refer to the copy-and-paste integration and the optimization-based integration as "Ours-CP" and "Ours-OP", respectively The results are shown in Figure 10 and our supplementary videos, indicating that our method outperforms previous approaches in terms of both speed and accuracy. Notably, as shown in Table 1, our copy-and-paste method runs in two orders of magnitude faster speed than the alternative approaches, yet showing better 3D pose estimation quality.

5. Discussion

We present FrankMacop, a fast motion capture system to estimate both 3D hand and 3D body motion from monocular inputs in the wild. We design the body and hand expert modules to produce compatible outputs for whole body motion capture. We present two integration strategies, copy-and-paste for faster speed and an optimization framework for better quality. The performance of our method has been demonstrated in-the-wild monocular videos. In particular, we also demonstrate our whole body motion capture system in a live demo at near real-time speed (9.5 fps), which is orders of magnitude faster than alternative methods. Our 3D hand pose estimation module outperforms previous state-

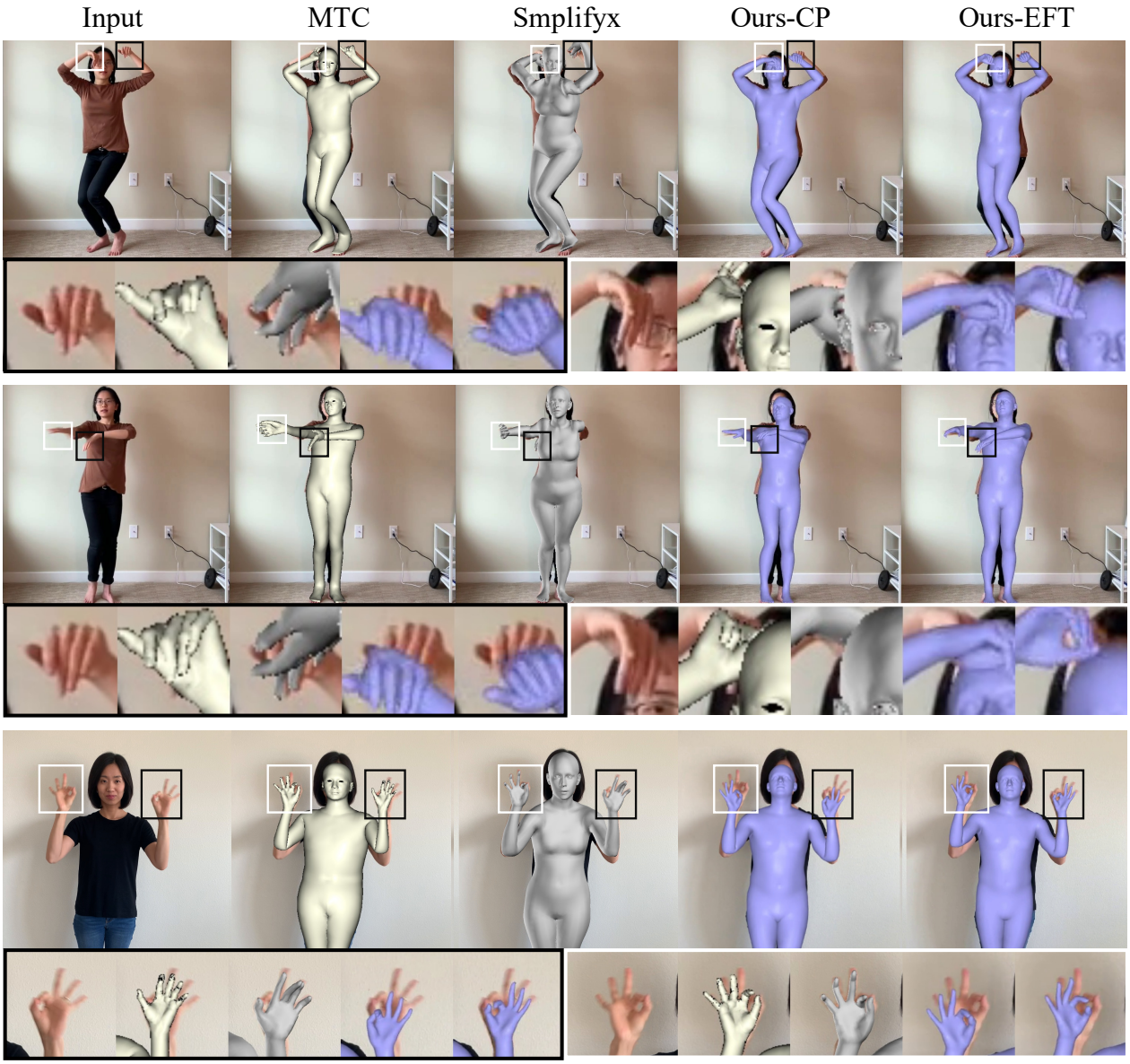


Figure 10: Qualitative comparison over the previous whole body pose estimation methods. We compare our method with MTC [55] and SMPLify-X [37]. Our method from two different strategies, by the copy-and-paste method (Ours-CP) and by an optimization framework (Ours-OP), is not only faster, but also produces more convincing whole body motion capture outputs.

of-the art on hand only methods in public benchmarks, and ours also can be used as a stand-alone monocular 3D hand pose estimator.

Our method still suffered from a few limitations: 1. Hand pose estimation become erroneous if two hands are too close each other. 2. Bounding boxes are required to infer 3D body and hands. It would be an interesting future direction to mitigate these problems and extend the method to handle cases of multiple people interacting with each other, such as two people greeting with hand shaking.

Acknowledgements. We thank Yuting Ye for her helpful discussions and feedbacks. We also want to thank Xintao Wang for helping us in implementing motion blur augmentation.

References

[1] Cmu motion capture database. http://mocap.cs.cmu.edu/resources.php. 7

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark

and state of the art analysis. In *CVPR*, 2014. 6

[3] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *TOG*, 2005. 2

[4] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1067–1076, 2019. 3, 8

[5] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, 2016. 1, 2, 7

[6] Giacomo Boracchi and Alessandro Foi. Uniform motion blur in poissonian noise: Blur/noise tradeoff. *IEEE Transactions on Image Processing*, 20(2):592–598, 2010. 6

[7] Giacomo Boracchi and Alessandro Foi. Modeling the performance of image restoration from motion blur. *IEEE Transactions on Image Processing*, 21(8):3502–3517, 2012. 6

[8] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019. 1, 3, 8, 9

[9] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV*, 2018. 1, 3, 8

[10] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 3, 7, 8

[11] 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 *Proc CVPR*, pages 10833–10842, 2019. 1, 3, 8

[12] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and objects poses. *arXiv preprint arXiv:1907.01481*, 2019. 5, 8, 9, 10

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5, 8

[14] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 2014. 2, 6, 8

[15] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5d heatmap regression. In *ECCV*, 2018. 1

[16] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *CVPR*, 2015. 8

[17] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human pose fitting towards in-the-wild 3d human pose estimation. *arXiv preprint arXiv:2004.03686*, 2020. 6, 7, 8

[18] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, et al. Panoptic studio: A massively multiview system for social interaction capture. *TPAMI*, 2017. 8

[19] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *CVPR*, 2018. 2, 3

[20] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 1, 2, 3, 4, 5, 6

[21] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *CVPR*, 2019. 4, 5

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Proc. ICLR*, 2016. 8

[23] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc ICLR*, 2017. 3

[24] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5253–5263, 2020. 2

[25] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019. 1, 2, 3, 4, 5, 6, 8

[26] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proc. CVPR*, 2019. 1, 2, 6, 7

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 6, 8, 9, 10

[28] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *TOG*, 2015. 2, 3, 6

[29] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017. 2

[30] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. Xnect: Real-time multi-person 3d human pose estimation with a single rgb camera. *arXiv preprint arXiv:1907.00837*, 2019. 2

[31] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *TOG*, 2017. 2

[32] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *CVPR*, 2018. 3

[33] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BmVC*, volume 1, page 3, 2011. 3

[34] Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Tracking the articulated motion of two strongly interacting hands. In *CVPR*, 2012. 1, 3

[35] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehle, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *3DV*, 2018. 2

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. 8

[37] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10975–10985, 2019. 2, 3, 10, 11

[38] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *CVPR*, 2017. 2

[39] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3D human pose and shape from a single color image. In *Proc. CVPR*, 2018. 2

[40] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J Black. Dyna: A model of dynamic human shape in motion. *TOG*, 2015. 2

[41] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Reconstructing 3d human pose from 2d image landmarks. In *CVPR*, 2012. 2

[42] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *TOG*, 2017. 2, 3, 5

[43] Yu Rong, Ziwei Liu, Cheng Li, Kaidi Cao, and Chen Change Loy. Delving deep into hybrid annotations for 3d human recovery in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5340–5348, 2019. 2

[44] Yu Rong, Ziwei Liu, Cheng Li, Kaidi Cao, and Chen Change Loy. Delving deep into hybrid annotations for 3d human recovery in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5340–5348, 2019. 2, 5

[45] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *CHI*, 2015. 1, 3

[46] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 8, 9

[47] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and robust hand tracking using detection-guided optimization. In *CVPR*, 2015. 1, 3

[48] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using RGB and depth data. In *ICCV*, 2013. 1, 3

[49] Andrea Tagliasacchi, Matthias Schröder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. Robust articulated-icp for real-time hand tracking. In *Computer Graphics Forum*, 2015. 3

[50] V. Tan, I. Budvytis, and R. Cipolla. Indirect deep structured learning for 3D human body shape and pose prediction. In *Proc. BMVC*, 2017. 2

[51] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. *ACM Transactions on Graphics (ToG)*, 35(6):1–11, 2016. 3

[52] Hsiao-Yu Fish Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *NIPS*, 2017. 2

[53] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *IJCV*, 2016. 1, 3

[54] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, 2018. 2

[55] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *Proc. CVPR*, pages 10965–10974, 2019. 1, 2, 3, 8, 9, 10, 11

[56] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *ICCV*, 2019. 2

[57] Linlin Yang and Angela Yao. Disentangling latent hands for image synthesis and pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9877–9886, 2019. 3

[58] Qi Ye, Shanxin Yuan, and Tae-Kyun Kim. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In *ECCV*, 2016. 1, 3

[59] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 3d hand pose tracking and estimation using stereo matching. *Proc ICIP*, 2017. 8, 9, 10

[60] Jason Y Zhang, Panna Felsen, Angjoo Kanazawa, and Jitendra Malik. Predicting 3d human dynamics from video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7114–7123, 2019. 2

[61] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2354–2364, 2019. 1, 3, 6, 8

[62] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proc ICCV*, pages 4903–4911, 2017. 1, 3, 8, 9, 10

[63] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proc. ICCV*, pages 813–822, 2019. 5, 8, 9, 10