Monocular Total Capture: Posing Face, Body, and Hands in the Wild

Donglai Xiang

CMU-RI-TR-19-19

May, 2019

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Thesis Committee:
Yaser Sheikh (Chair)
Martial Hebert
Aayush Bansal

Submitted in partial fulfillment of the
requirements for the degree of Master of Science in Robotics

Abstract

We present the first method to capture the 3D total motion of a target person from a monocular view input. Given an image or a monocular video, our method reconstructs the motion from body, face, and fingers represented by a 3D deformable mesh model. We use an efficient representation called 3D Part Orientation Fields (POFs), to encode the 3D orientations of all body parts in the common 2D image space. POFs are predicted by a Fully Convolutional Network, along with the joint confidence maps. To train our network, we collect a new 3D human motion dataset capturing diverse total body motion of 40 subjects in a multiview system. We leverage a 3D deformable human model to reconstruct total body pose from the CNN outputs with the aid of the pose and shape prior in the model. We also present a texture-based tracking method to obtain temporally coherent motion capture output. We perform thorough quantitative evaluations including comparison with the existing body-specific and hand-specific methods, and performance analysis on camera viewpoint and human pose changes. Finally, we demonstrate the results of our total body motion capture on various challenging in-the-wild videos.

Contents

Chapter 1

Introduction

Human motion capture is essential for many applications including visual effects, robotics, sports analytics, medical applications, and human social behavior understanding. However, capturing 3D human motion is often costly, requiring a special motion capture system with multiple cameras. For example, the most widely used system [2] needs multiple calibrated cameras with reflective markers carefully attached to the subjects' body. The actively-studied markerless approaches are also based on multi-view systems [19, 21, 25, 26, 29] or depth cameras [7, 50]. For this reason, the amount of available 3D motion data is extremely limited. Capturing 3D human motion from single images or videos can provide a huge breakthrough for many applications by increasing the accessibility of 3D human motion data, especially by converting all human-activity videos on the Internet into a large-scale 3D human motion corpus.

Reconstructing 3D human pose or motion from a monocular image or video, however, is extremely challenging due to the fundamental depth ambiguity. Interestingly, humans are able to almost effortlessly reason about the 3D human body motion from a single view, presumably by leveraging strong prior knowledge about feasible 3D human motions. Inspired by this, several learning-based approaches have been proposed over the last few years to predict 3D human body motion (pose) from a monocular video (image) [4, 9, 27, 33, 35, 36, 44, 58, 60, 69, 73] using available 2D and 3D human pose datasets [1, 5, 22, 25, 28]. Recently, similar approaches have been introduced to predict 3D hand poses from a monocular view [12, 37, 74]. However, fundamental difficulty still remains due to the lack of available in-the-wild 3D body or hand datasets that provide paired images and 3D pose data; thus most of the previous methods only demonstrate results in controlled lab environments. Importantly, there exists no method that can reconstruct motion from all body parts including body, hands, and face altogether from a single view, although this is important for fully understanding human behavior.

In this thesis, we aim to reconstruct the **3D total motions** [26] of a human using a monocular imagery captured in the wild. This ambitious goal requires solving challenging 3D pose estimation problems for different body parts altogether, which are often considered as separate research domains. Notably, we apply our method to in-the-wild situations (e.g., videos from YouTube), which has rarely been demonstrated in previous work. We use a 3D representation named Part Orientation Fields (POFs) to efficiently encode the 3D orientation of a body part in the 2D space. A POF is defined for each body part that connects adjacent joints in torso, limbs, and fingers, and represents relative 3D orientation of the rigid part re-

Figure 1.1: We present the first method to simultaneously capture the 3D total body motion of a target person from a monocular view input. For each example, (left) input image and (right) 3D total body motion capture results overlaid on the input.

gardless of the origin of 3D Cartesian coordinates. POFs are efficiently predicted by a Fully Convolutional Network (FCN), along with 2D joint confidence maps [15,63,68]. To train our networks, we collect a new 3D human motion dataset containing diverse body, hands, and face motions from 40 subjects. Separate CNNs are adopted for body, hand and face, and their outputs are consolidated together in a unified optimization framework. We leverage a 3D deformable model that is built for total capture [25] in order to exploit the shape and motion prior embedded in the model. In our optimization framework, we fit the model to the CNN measurements at each frame to simultaneously estimate the 3D motion of body, face, fingers, and feet. Our mesh output also enables us to additionally refine our motion capture results for better temporal coherency by optimizing the photometric consistency in the texture space.

This thesis presents the first approach to monocular total motion capture in various challenging in-the-wild scenarios (e.g., Fig. 1.1). We demonstrate that our single framework achieves comparable results to existing state-of-the-art 3D body-only or hand-only pose estimation methods on public benchmarks. Notably, our method is applied to various in-the-wild videos, which has rarely been demonstrated in either 3D body or hand estimation area. We also conduct thorough experiments on our newly collected dataset to quantitatively evaluate the performance of our method with respect to viewpoint and body pose changes. The major contributions of this thesis are summarized as follows:

- We present the first method to produce **3D total motion capture** results from a monocular image or video in various challenging in-the-wild scenarios.
- We introduce an optimization framework to fit a deformable human model on 3D POFs and 2D keypoint measurements for total body pose estimation, showing comparable results to the state-of-the-art methods on both 3D body and 3D hand estimation benchmarks.
- We present a method to enforce photometric consistency across time to reduce motion jitters.
- We capture a new 3D human motion dataset with 40 subjects as training and evaluation data for monocular total motion capture.

Chapter 2

Related Work

In this chapter, we review various previous work related to this thesis.

2.1 Single Image 2D Human Pose Estimation

Over the last few years, great progress has been made in detecting 2D human body keypoints from a single image [11, 15, 38, 63, 64, 68] by leveraging large-scale manually annotated datasets [5, 28] with deep Convolutional Neural Network (CNN) framework. In particular, the major breakthrough is boosted by using the fully convolutional architectures to produce confidence scores for each joint with a heatmap representation [15, 38, 63, 68], which is known to be more efficient than directly regressing the joint locations with fully connected layers [64]. A recent work [15] learns the connectivity between pairs of adjacent joints, called the Part Affinity Fields (PAFs) in the form of 2D heatmaps, to assemble 2D keypoints for different individuals in the multi-person 2D pose estimation problem.

2.2 Single Image 3D Human Pose Estimation

Early work [4, 44] models the 3D human pose space as an over-complete dictionary learned from a 3D human motion database [1]. More recent approaches rely on deep neural networks, which are roughly divided into two-stage methods and direct estimation methods. The two-stage methods take 2D keypoint estimation as input and focus on lifting 2D human poses to 3D without considering input image [9,17,20,33,36,39]. These methods ignore rich information in images that encodes 3D information, such as shading and appearance, and also suffer from sensitivity to 2D localization error. Direct estimation methods predict 3D human pose directly from images, in the form of direct coordinate regression [46, 55, 56], voxel [32, 42, 66] or depth map [73]. Similar to ours, a recent work uses 3D orientation fields [31] as an intermediate representation for the 3D body pose. However, these models are usually trained on MoCap datasets, with limited ability to generalize to in-the-wild scenarios.

Due to the above limitations, some methods have been proposed to integrate prior knowledge about human pose for better in-the-wild performance. Some work [41,48,67] proposes to use ordinal depth as additional supervision for CNN training. Additional loss functions are introduced in [18, 73] to enforce constraints on predicted bone length and joint angles.

Some work [27, 70] uses Generative Adversarial Networks (GAN) to exploit human pose prior in a data-driven manner.

2.3 Monocular Hand Pose Estimation

Hand pose estimation is often considered as an independent research domain from body pose estimation. Most of previous work is based on depth image as input [40, 49, 52, 54, 65, 71]. RGB-based methods have been introduced recently, for 2D keypoint estimation [51] and 3D pose estimation [12, 23, 74].

2.4 3D Deformable Human Models

3D deformable models are commonly used for markerless body [6, 30, 43] and face motion capture [8, 13] to restrict the reconstruction output to the shape and motion spaces defined by the models. Although the outputs are limited by the expressive power of models (e.g., some body models cannot express clothing and some face models cannot express wrinkles), they greatly simplify the 3D motion capture problem. We can fit the models based on available measurements by optimizing cost functions with respect to the model parameters. Recently, a generative 3D model that can express body and hands is introduced by Romero et al. [47]; the Adam model is introduced by Joo et al. [26] to enable the total body motion capture (face, body and hands), which we adopt for monocular total capture.

2.5 Photometric Consistency for Human Tracking

Photometric consistency of texture has been used in various previous work to improve the robustness of body tracking [45] and face tracking [61, 62]. Some work [10, 16] also uses optical flow to align rendered 3D human models. In this work, we improve temporal coherency of our output by a photo-consistency term which significantly reduces jitters. This is the first time that such technique is applied to monocular body motion tracking to the best of our knowledge.

Chapter 3

Proposed Method

3.1 Method Overview

Our method takes as input a sequence of images capturing the motion of a single person from a monocular RGB camera, and outputs the 3D total body motion (including the motion from body, face, hands, and feet) of the target person in the form of a deformable 3D human model [26, 30] for each frame. Given an N-frame video sequence, our method produces the parameters of the 3D human body model, including body motion parameters $\{\boldsymbol{\theta}_i\}_{i=1}^N$, facial expression parameters $\{\boldsymbol{\sigma}_i\}_{i=1}^N$, and global translation parameters $\{\boldsymbol{t}_i\}_{i=1}^N$. The body motion parameters $\boldsymbol{\theta}$ includes hands and foot motions, together with the global rotation of the body. Our method also estimates shape coefficients $\boldsymbol{\phi}$ shared among all frames in the sequence, while $\boldsymbol{\theta}$, $\boldsymbol{\sigma}$, and $\boldsymbol{t}$ are estimated for each frame respectively. Here, the output parameters are defined by the 3D deformable human model Adam [26]. However, our method can be also applied to capture only a subset of total motions (e.g., body motion only with the SMPL model [30] or hand motion only by separate hand model of Frankenstein in [26]). We denote a set of all parameters $(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{t})$ by $\boldsymbol{\Psi}$, and denote the result for the i-th frame by $\boldsymbol{\Psi}_i$.

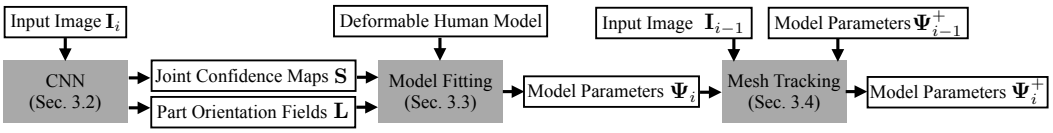

Figure 3.1: An overview of our method. Our method is composed of CNN part, mesh fitting part, and mesh tracking part.

Our method is divided into 3 stages, as shown in Fig. 3.1. In the first stage, each image is fed into a Convolutional Neural Network (CNN) obtain the joint confidence maps and the 3D orientation information of body parts, which we call the 3D Part Orientation Fields (POFs). In the second stage, we estimate total body pose by fitting a deformable human mesh model [26] on the image measurements produced by the CNN. We utilize the prior information embedded in the human body model for better robustness against the noise in CNN outputs. This stage produces the 3D pose for each frame independently, represented by parameters of the deformable model $\{\boldsymbol{\Psi}_i\}_{i=1}^N$. In the third stage, we additionally enforce

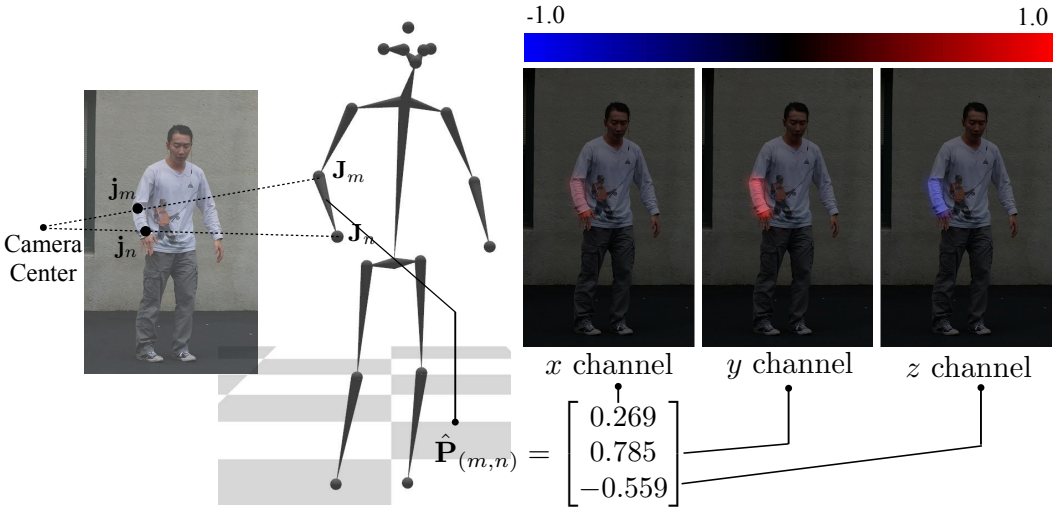

Figure 3.2: An illustration of a Part Orientation Field. The orientation $\hat{\mathbf{P}}_{(m,n)}$ for body part $\mathbf{P}_{(m,n)}$ is a unit vector from $\mathbf{J}_m$ to $\mathbf{J}_n$. All pixels belong to this part in the POF are assigned the value of this vector in x, y, z channels.

temporal consistency across frames to reduce motion jitters. We define a cost function to ensure photometric consistency in the texture domain of mesh model, based on the fitting outputs of the second stage. This stage produces refined model parameters $\{\mathbf{\Psi}_i^+\}_{i=1}^N$. This stage is crucial for obtaining realistic body motion capture output.

3.2 Predicting 3D Part Orientation Fields

The 3D Part Orientation Field (POF) encodes the 3D orientation of a body part of an articulated structure (e.g., limbs, torso, and fingers) in 2D image space. The same representation is used in a very recent literature [31], and we describe the details and notations used in our framework. We pre-define a human skeleton hierarchy $\mathbb{S}$ in the form of a set of '(parent, child)' pairs[1]. A rigid body part connecting a 3D parent joint $\mathbf{J}_m \in \mathbb{R}^3$ and a child joint $\mathbf{J}_n \in \mathbb{R}^3$ is denoted by $\mathbf{P}_{(m,n)}$, with $\mathbf{J}_m, \mathbf{J}_n$ defined in the camera coordinate, if $(m,n) \in \mathbb{S}$. Its 3D orientation $\hat{\mathbf{P}}_{(m,n)}$ is represented by a unit vector from $\mathbf{J}_m$ to $\mathbf{J}_n$ in $\mathbb{R}^3$:

$$\hat{\mathbf{P}}_{(m,n)} = \frac{\mathbf{J}_n - \mathbf{J}_m}{||\mathbf{J}_n - \mathbf{J}_m||} \,. \tag{3.1}$$

For a specific body part $\mathbf{P}_{(m,n)}$, its Part Orientation Field $\mathbf{L}_{(m,n)} \in \mathbb{R}^{3\times h\times w}$ encodes its 3D orientation $\hat{\mathbf{P}}_{(m,n)}$ as a 3-channel heatmap (in x, y, z directions respectively) in the image space, where h and w are the size of image. The value of the POF $\mathbf{L}_{(m,n)}$ at a pixel $\mathbf{x}$ is defined as,

$$\mathbf{L}_{(m,n)}(\mathbf{x}) = \begin{cases} \hat{\mathbf{P}}_{(m,n)} & \text{if } \mathbf{x} \in \mathbf{P}_{(m,n)}, \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{3.2}$$

[1]See Appendix B for our body and hand skeleton definition.

Note that the POF values are non-zero only for the pixels belonging to the current target part $\mathbf{P}_{(m,n)}$ and we follow [15] to define the pixels belonging to the part as a rectangle. An example POF is shown in Fig. 3.2.

3.2.1 Implementation Details

We train a CNN to predict joint confidence maps $\mathbf{S}$ and Part Orientation Fields $\mathbf{L}$. The input image is cropped around the target person to 368×368. The bounding box is given by OpenPose[2] [14, 15, 51] for testing. We follow [15] for CNN architecture with minimal change. 3 channels are used to estimate POF instead of 2 channels in [15] for every body part in $\mathbb{S}$. L_2 loss is applied to network prediction on $\mathbf{S}$ and $\mathbf{L}$. We also train our network on images with 2D pose annotations (e.g. COCO). In this situation we only supervise the network with loss on $\mathbf{S}$. Two networks are trained for body and hands separately.

3.3 Model-Based 3D Pose Estimation

Ideally the joint confidence maps $\mathbf{S}$ and POFs $\mathbf{L}$ produced by CNN provide sufficient information to reconstruct a 3D skeletal structure up to scale [31]. In practice, $\mathbf{S}$ and $\mathbf{L}$ can be noisy, so we exploit a 3D deformable mesh model to more robustly estimate 3D human pose with the shape and pose priors embedded in the model. In this section, we first describe our mesh fitting process for body, and then extend it to hand pose and facial expression for total body motion capture. We use superscripts B, LH, RH, T and F to denote functions and parameters for body, left hand, right hand, toes, and face respectively. We use Adam [26] which encompasses the expressive power for body, hands and facial expression in a single model. Other human models (e.g., SMPL [30]) can be also used if the goal is to reconstruct only part of the total body motion.

3.3.1 Deformable Mesh Model Fitting with POFs

Given 2D joint confidence maps $\mathbf{S}^B$ predicted by our CNN for body, we obtain 2D keypoint locations $\{\mathbf{j}_m^B\}_{m=1}^J$ by taking channel-wise argmax on $\mathbf{S}^B$. Given $\{\mathbf{j}_m^B\}_{m=1}^J$ and the other CNN output POFs $\mathbf{L}^B$, we compute the 3D orientation of each bone $\hat{\mathbf{P}}_{(m,n)}^B$ by averaging the values of $\mathbf{L}^B$ along the segment from $\mathbf{j}_m^B$ to $\mathbf{j}_n^B$ as in [15]. We obtain a set of mesh parameters $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, and $\boldsymbol{t}$ that agree with these image measurements by minimizing the following objective:

$$\mathcal{F}^B(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{t}) = \mathcal{F}_{\text{2D}}^B(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{t}) + \mathcal{F}_{\text{POF}}^B(\boldsymbol{\theta}, \boldsymbol{\phi}) + \mathcal{F}_p^B(\boldsymbol{\theta}), \tag{3.3}$$

where $\mathcal{F}_{\text{2D}}^B$, $\mathcal{F}_{\text{POF}}^B$, and $\mathcal{F}_p^B$ are different constraints as defined below. The 2D keypoint constraint $\mathcal{F}_{\text{2D}}^B$ penalizes the discrepancy between network-predicted 2D keypoints and the projections of the joints in the human body model:

$$\mathcal{F}_{\text{2D}}^B(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{t}) = \sum_m \|\mathbf{j}_m^B - \boldsymbol{\Pi}(\tilde{\mathbf{J}}_m^B(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{t}))\|^2, \tag{3.4}$$

where $\tilde{\mathbf{J}}_m^B(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{t})$ is m-th joint of the human model and $\boldsymbol{\Pi}(\cdot)$ is projection function from 3D space to image, where we assume a weak perspective camera model. The POF constraint

[2] https://github.com/CMU-Perceptual-Computing-Lab/openpose

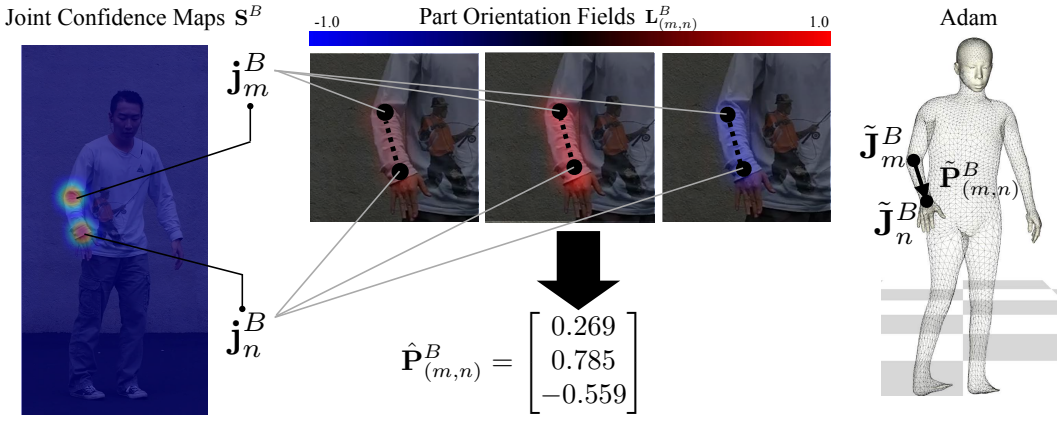

Figure 3.3: Human model fitting on estimated POFs and joint confidence maps. We extract 2D joint locations from joint confidence maps (left) and then body part orientation from POFs (middle). Then we optimize a cost function (Eq. 3.3) that minimizes the distance between $\mathbf{\Pi}(\tilde{\mathbf{J}}^B_m)$ and $\mathbf{j}^B_m$ and angle between $\tilde{\mathbf{P}}^B_{(m,n)}$ and $\hat{\mathbf{P}}^B_{(m,n)}$.

$\mathcal{F}^B_{\text{POF}}$ penalizes the difference between POF prediction and the orientation of body part in mesh model:

$$\mathcal{F}^B_{\text{POF}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = w^B_{\text{POF}} \sum_{(m,n)\in\mathbb{S}} 1 - \hat{\mathbf{P}}^B_{(m,n)} \cdot \tilde{\mathbf{P}}^B_{(m,n)}(\boldsymbol{\theta}, \boldsymbol{\phi}), \tag{3.5}$$

where $\tilde{\mathbf{P}}^B_{(m,n)}$ is the unit directional vector for the bone $\mathbf{P}^B_{(m,n)}$ in the human mesh model, w^B_{POF} is a balancing weight for this term, and $\cdot$ is the inner product between vectors. The prior term $\mathcal{F}^B_p$ is used to restrict our output to a feasible human pose distribution (especially for rotation around bones), defined as:

$$\mathcal{F}^B_p(\boldsymbol{\theta}) = w^B_p \|\mathbf{A}^B_\theta(\boldsymbol{\theta} - \boldsymbol{\mu}^B_\theta)\|^2, \tag{3.6}$$

where $\mathbf{A}^B_\theta$ and $\boldsymbol{\mu}^B_\theta$ are pose prior learned from CMU Mocap dataset [1], and w^B_p is a balancing weight. We use Levenberg-Marquardt algorithm [3] to optimize Eq. 3.3. The mesh fitting process is illustrated in Fig. 3.3.

3.3.2 Total Body Capture with Hands, Feet and Face

Given the output of the hand network $\mathbf{S}^{LH}, \mathbf{L}^{LH}$ and $\mathbf{S}^{RH}, \mathbf{L}^{RH}$, we can additionally fit the Adam model to estimate the hand pose using similar optimization objectives:

$$\mathcal{F}^{LH}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{t}) = \mathcal{F}^{LH}_{\text{2D}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{t}) + \mathcal{F}^{LH}_{\text{POF}}(\boldsymbol{\theta}, \boldsymbol{\phi}) + \mathcal{F}^{LH}_p(\boldsymbol{\theta}). \tag{3.7}$$

$\mathcal{F}^{LH}$ is the objective function for left hand and each term is defined similarly to Eq. 3.4, 3.5, 3.6. Similar to previous work on hand tracking [57,59], we use a hand pose prior constraint $\mathcal{F}^{LH}_p$, learned from the MANO dataset [47]. The objective function for the right hand $\mathcal{F}^{RH}$ is similarly defined.

Once we fit the body and hand parts of the deformable model to the CNN outputs, the projection of the model on the image is already well aligned to the target person. Then

we can reconstruct other body parts by simply adding more 2D joint constraints using additional 2D keypoint measurements. In particular, we include 2D face and foot keypoints from the OpenPose detector. The additional cost function for toes is defined as:

$$\mathcal{F}^T(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{t}) = \sum_m \|\mathbf{j}_m^T - \boldsymbol{\Pi}(\tilde{\mathbf{J}}_m^T(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{t}))\|^2, \tag{3.8}$$

where $\{\mathbf{j}_m^T\}$ are 2D tiptoe keypoints on both feet from OpenPose, and $\{\tilde{\mathbf{J}}_m^T\}$ are the 3D joint location of the mesh model in use. Similarly for face we define:

$$\mathcal{F}^F(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{t}, \boldsymbol{\sigma}) = \sum_m \|\mathbf{j}_m^F - \boldsymbol{\Pi}(\tilde{\mathbf{J}}_m^F(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{t}, \boldsymbol{\sigma}))\|^2. \tag{3.9}$$

Note that the facial keypoints $\tilde{\mathbf{J}}_m^F$ are determined by all the mesh parameters $\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{t}, \boldsymbol{\sigma}$ together. In addition, we also apply regularization for shape parameters and facial expression parameters:

$$R^{\boldsymbol{\phi}}(\boldsymbol{\phi}) = \|\boldsymbol{\phi}\|^2, R^{\boldsymbol{\sigma}}(\boldsymbol{\sigma}) = \|\boldsymbol{\sigma}\|^2. \tag{3.10}$$

Putting them together, the total optimization objective is

$$\begin{aligned} \mathcal{F}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{t}, \boldsymbol{\sigma}) = \mathcal{F}^B + \mathcal{F}^{LH} + \mathcal{F}^{RH} + \\ \mathcal{F}^T + \mathcal{F}^F + R^{\boldsymbol{\phi}} + R^{\boldsymbol{\sigma}}, \end{aligned} \tag{3.11}$$

where the balancing weights for all the terms are omitted for simplicity. We optimize this total objective function in multiple stages to avoid local minima. We first fit the torso, then add limbs, and finally optimize the full objective function including all constraints. This stage produces 3D total body motion capture results for each frame independently in the form of Adam model parameters $\{\boldsymbol{\Psi}_i\}_{i=1}^N$. For more detail on deformable model fitting, please refer to Appendix C.

3.4 Enforcing Photo-Consistency in Textures

In the previous stages, we perform per-frame processing, which is vulnerable to motion jitters. Inspired by previous work on body and face tracking [45, 62], we propose to reduce the jitters using the pixel-level image cues given the initial model fitting results. The core idea is to enforce photometric consistency in the model textures, extracted by projecting the fitted mesh models on the input images. Ideally, the textures should be consistent across frames, but in practice there exist discrepancies due to motion jitters. In order to efficiently implement this constraint in our optimization framework, we compute optical flows from projected texture to the target input image. The destination of each flow indicates the expected location of vertex projection. To describe our method, we define a function $\mathcal{T}$ which extracts a texture given an image and a mesh structure:

$$\boldsymbol{\mathcal{T}}_i = \mathcal{T}\left(\mathbf{I}_i, M(\boldsymbol{\Psi}_i)\right), \tag{3.12}$$

where $\mathbf{I}_i$ is the input image of the i-th frame $M(\boldsymbol{\Psi}_i)$ is the human model determined by parameters $\boldsymbol{\Psi}_i$. The function $\mathcal{T}$ extracts a texture map $\boldsymbol{\mathcal{T}}_i$ by projecting the mesh structure on the image for the visible parts. We ideally expect the texture for (i+1)-th frame $\boldsymbol{\mathcal{T}}_{i+1}$ to

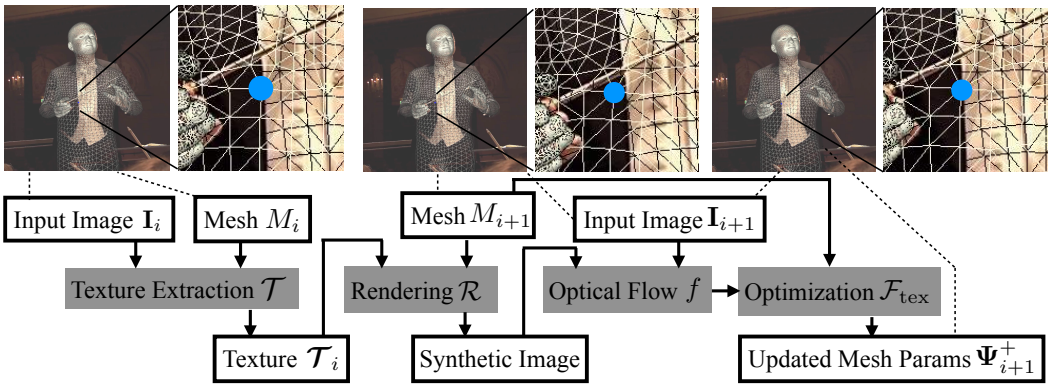

Figure 3.4: Illustration of our temporal refinement algorithm. The top row shows meshes projected on input images at previous frame, current target frame, and after refinement. In zoom-in views, a particular vertex is shown in blue, which is more consistent after applying our tracking method.

be the same as $\mathcal{T}_i$. Instead of directly using this constraint for optimization, we use optical flow to compute the discrepancy between these textures for easier optimization. More specifically, we pre-compute the optical flow between the image $\mathbf{I}_{i+1}$ and the rendering of the mesh model at (i+1)-th frame with the i-th frame's texture map $\mathcal{T}_i$, which we call 'synthetic image':

$$\mathbf{f}_{i+1} = f(\mathcal{R}(M_{i+1}, \mathcal{T}_i), \mathbf{I}_{i+1}), \tag{3.13}$$

where $M_{i+1} = M(\mathbf{\Psi}_{i+1})$ is the mesh for the (i+1)-th frame, and $\mathcal{R}$ is a rendering function that renders a mesh with a texture to an image. The function f computes optical flows from the synthetic image to the input image $\mathbf{I}_{i+1}$. The output flow $\mathbf{f}_{i+1} : \mathbf{x} \longrightarrow \mathbf{x}'$ maps a 2D location $\mathbf{x}$ to a new location $\mathbf{x}'$ following the optical flow result. Intuitively, the computed flow mapping $\mathbf{f}_{i+1}$ drives the projection of 3D mesh vertices toward the directions for better photometric consistency in textures across frames. Based on this flow mapping, we define the texture consistency term:

$$\mathcal{F}_{\text{tex}}(\mathbf{\Psi}^+_{i+1}) = \sum_n \|\mathbf{v}^+_n(i+1) - \mathbf{v}'_n(i+1)\|^2, \tag{3.14}$$

where $\mathbf{v}^+_n(i+1)$ is the projection of the n-th mesh vertex as a function of model parameters $\mathbf{\Psi}^+_{i+1}$ under optimization. $\mathbf{v}'_n(i+1) = \mathbf{f}_{i+1}(\mathbf{v}_n(i+1))$ is the destination of each optical flow, where $\mathbf{v}_n(i+1)$ is the projection of n-th mesh vertex of mesh M_{i+1}. Note that $\mathbf{v}'_n(i+1)$ is pre-computed and constant during the optimization. This constraint is defined in image space, and thus it mainly reduces the jitters in x, y directions. Since there is no image clue to reduce the jitters along z direction, we just enforce a smoothness constraint for z-components of 3D joint locations:

$$\mathcal{F}_{\Delta z}(\boldsymbol{\theta}^+_{i+1}, \boldsymbol{\phi}^+_{i+1}, \boldsymbol{t}^+_{i+1}) = \sum_m (\mathbf{J}^{+z}_m(i+1) - \mathbf{J}^z_m(i))^2, \tag{3.15}$$

where $\mathbf{J}^{+z}_m(i+1)$ is z-coordinate of the m-th joint of the mesh model as a function of parameters under optimization, and $\mathbf{J}^z_m(i)$ is the corresponding value in previous frame as a fixed constant. Finally, we define a new objective function:

$$\mathcal{F}^+(\mathbf{\Psi}^+_{i+1}) = \mathcal{F}_{\text{tex}} + \mathcal{F}_{\Delta z} + \mathcal{F}_{\text{POF}} + \mathcal{F}^F, \tag{3.16}$$

where the balancing weights are omitted. We minimize this function to obtain the parameter of the (i+1)-th frame $\mathbf{\Psi}_{i+1}^{+}$, initialized from output of last stage $\mathbf{\Psi}_{i+1}$. Compared to the original full objective Eq. 3.11, this new objective function is simpler since it starts from a good initialization. Most of the 2D joint constraints are replaced by $\mathcal{F}_{\text{tex}}$, while we found that the POF term and face keypoint term are still needed to avoid error accumulation. Note that this optimization is performed recursively—we use the updated parameters of the i-th frame $\mathbf{\Psi}_{i}^{+}$ to extract the texture $\boldsymbol{\mathcal{T}}_i$ in Eq. 3.12, and update the model parameters at the (i+1)-th frame from $\mathbf{\Psi}_{i+1}$ to $\mathbf{\Psi}_{i+1}^{+}$ with this optimization. Also note that the shape parameters $\{\boldsymbol{\phi}_i^{+}\}$ should be the same across the sequence, so we take $\boldsymbol{\phi}_{i+1}^{+} = \boldsymbol{\phi}_i^{+}$ and fix it during optimization. We also fix the facial expression parameters in this stage.

Chapter 4

Results

In this chapter, we present thorough quantitative and qualitative evaluation of our method.

4.1 Dataset

Body Pose Dataset: *Human3.6M* [22] is an indoor marker-based human MoCap dataset, and currently the most commonly used benchmark for 3D body pose estimation. We quantitatively evaluate the body part of our algorithm on it. We follow the standard training-testing protocol as in [42].
Hand Pose Dataset: *Stereo Hand Pose Tracking Benchmark (STB)* [72] is a 3D hand pose dataset consisting of 30K images for training and 6K images for testing. *Dexter+Object (D+O)* [53] is a hand pose dataset captured by an RGB-D camera, providing about 3K testing images in 6 sequences. Only the locations of finger tips are annotated.
Newly Captured Total Motion Dataset: We use the Panoptic Studio [24, 25] to capture a new dataset for 3D body and hand pose in a markerless way [26]. 40 subjects are captured when makeing a wide range of motion in body and hand under the guidance of a video for 2.5 minutes. After filtering we obtain about 834K body images and 111K hand images with corresponding 3D pose data. We split this dataset into training and testing set such that no subject appears in both. For more details on the dataset, please refer to Appendix A.

4.2 Quantitative Comparison with Previous Work

4.2.1 3D Body Pose Estimation

Comparison on Human3.6M

We compare the performance of our single-frame body pose estimation method with previous state-of-the-arts. Our network is initialized from the 2D body pose estimation network of OpenPose. We train the network using COCO dataset [28], our new 3D body pose dataset, and Human3.6M for 165k iterations with a batch size of 4. During testing time, we fit Adam model [26] onto the network output. Since Human3.6M has a different joint definition from Adam model, we build a linear regressor to map Adam mesh vertices to 17 joints in Human3.6M definition using the training set, as in [27]. For evaluation, we follow [42] to rescale our output to match the size of an average skeleton computed from the training

Method	MPJPE
Pavlakos [42]	71.9
Zhou [73]	64.9
Luo [31]	63.7
Martinez [33]	62.9
Fang [20]	60.4
Yang [70]	58.6
Pavlakos [41]	56.2
Dabral [18]	55.5
Sun [56]	49.6
*Kanazawa [27]	88.0
*Mehta [35]	80.5
*Mehta [34]	69.9
*Ours	**58.3**
*Ours+	64.5

Table 4.1: Quantitative comparison with previous work on Human3.6M dataset. The '*' signs indicate methods that show results on in-the-wild videos. The evaluation metric is Mean Per Joint Position Error (MPJPE) in millimeter. The numbers are taken from original papers. 'Ours' and 'Ours+' refer to our results without and with prior respectively.

set. The Mean Per Joint Position Error (MPJPE) after aligning the root joint is reported as in [42].

The experimental results are shown in Table 4.1. Our method achieves competitive performance; in particular, we show the lowest pose estimation error among all methods that demonstrate their results on in-the-wild videos (marked with '*' in the table). We believe it important to show results on in-the-wild videos to ensure the generalization beyond this particular dataset. As an example, our result with pose prior shows higher error compared to our result without prior, although we find that pose prior helps to maintain good mesh surface and joint angles in the wild.

Ablation Studies

We investigate the importance of each dataset through ablation studies on Human3.6M. We compare the result by training networks with: (1) Human3.6M; (2) Human3.6M and our captured dataset; and (3) Human3.6M, our captured dataset, and COCO. Note that setting (3) is the one we use for the previous comparison. We follow the same evaluation protocol and metric as in Table 4.1, with result shown in Table 4.2. First, it is worth noting that with only Human3.6M as training data, we already achieve the best performance among results marked with '*' in Table 4.1. Second, comparing (2) with (1), our new dataset provides an improvement despite the difference in background, human appearance and pose distribution between our dataset and Human3.6M. This verifies the value of our new dataset. Third, we see a drop in error when we add COCO to the training data, which suggests that our framework can take advantage of this dataset with only 2D human pose annotation for 3D pose estimation.

Training data	MPJPE
(1) Human3.6M	65.6
(2) Human3.6M + Ours	60.9
(3) Human3.6M + Ours + COCO	58.3

Table 4.2: Ablation studies on Human3.6M. The evaluation metric is Mean Per Joint Position Error in millimeter.

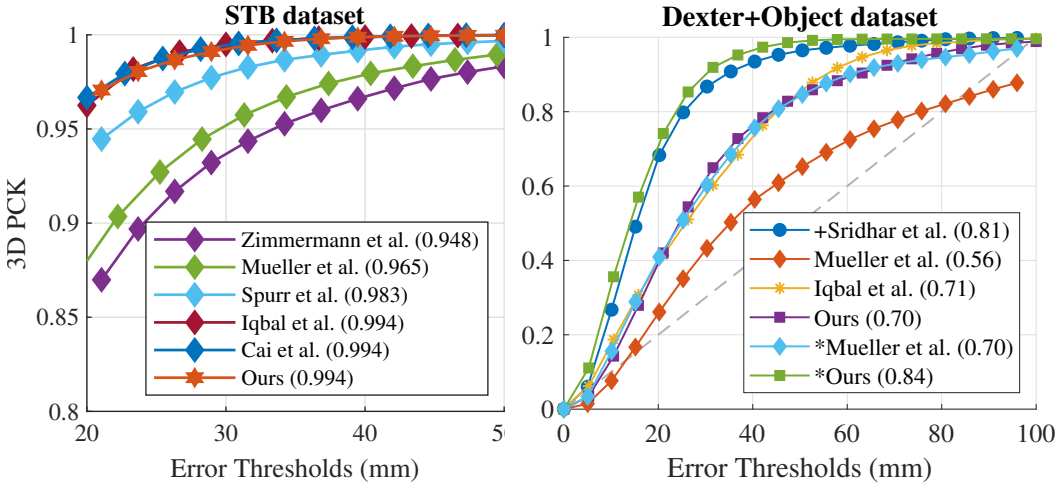

Figure 4.1: Comparison with previous work on 3D hand pose estimation. We plot PCK curve and show AUC in bracket for each method in legend. Left: results on the STB dataset [72] in 20mm-50mm; right: results on Dexter+Object dataset [53] in 0-100mm. Results with depth alignment are marked with '*'; the RGB-D based method is marked with '+'.

4.2.2 3D Hand Pose Estimation

We evaluate our method on the Stereo Hand Pose Tracking Benchmark (STB) and Dexter+Object (D+O), and compare our result with previous methods. For this experiment we use the separate hand model of Frankenstein in [26].

STB

Since the STB dataset has a palm joint rather than the wrist joint used in our method, we convert the palm joint to wrist joint as in [74] to train our CNN. We also learn a linear regressor using the training set of STB dataset. During testing, we regress back the palm joint from our model fitting output for comparison. For evaluation, we follow the previous work [74] and compute the error after aligning the position of root joint and global scale with the ground truth, and report the Area Under Curve (AUC) of the Percentage of Correct Keypoints (PCK) curve in the 20mm-50mm range. The results are shown in the left of Fig. 4.1. Our performance is on par with the state-of-the-art methods that are designed particularly for hand pose estimation. We also point out that the performance on this dataset has almost saturated, because the percentage is already above 90% even at the lowest threshold.

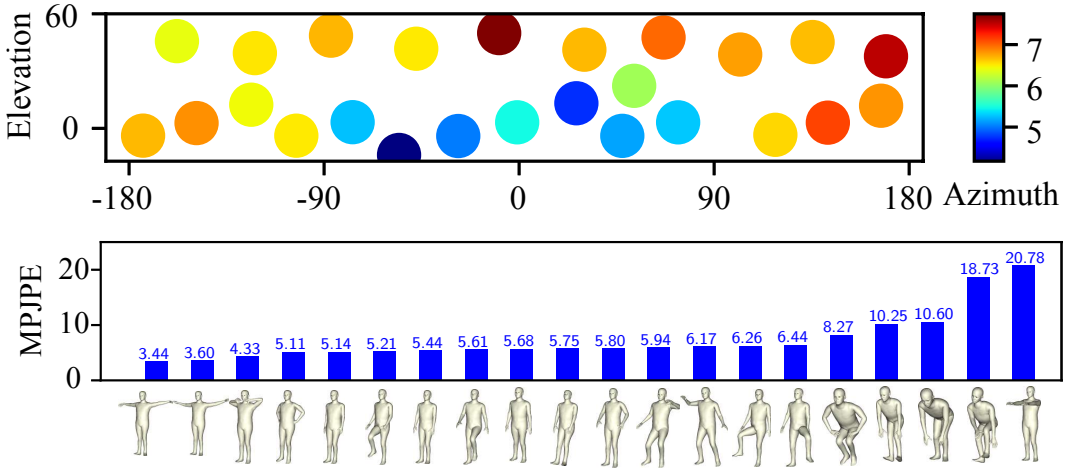

Figure 4.2: Evaluation result in Panoptic Studio. Top: accuracy vs. view point; bottom: accuracy vs. pose. The metric is MPJPE in cm. The average MPJPE for all testing samples is 6.30 cm.

D+O

Following [37] and [23], we report our results using a PCK curve and the corresponding AUC in the right of Fig. 4.1. Since previous methods are evaluated by estimating the absolute 3D depth of 3D hand joints, we follow them by finding an approximate hand scale using a single frame in the dataset, and fix the scale during the evaluation. In this case, our performance (AUC=0.70) is comparable with the previous state-of-the-art [23] (AUC=0.71). However, since there is fundamental depth-scale ambiguity for single-view pose estimation, we argue that aligning the root with the ground truth depth is a more reasonable evaluation setting. In this setting, our method (AUC=0.84) outperforms the previous state-of-the-art method [37] (AUC=0.70) in the same setting, and even achieves better performance than an RGB-D based method [53] (AUC=0.81).

4.3 Quantitative Study for View and Pose Changes

Our new 3D pose data contain multi-view images with the diverse body postures. This allows us to quantitatively study the performance of our method in view changes and body pose changes. We compare our single view 3D body reconstruction result with the ground truth. Due to the scale-depth ambiguity of monocular pose estimation, we align the depth of root joint to the ground truth by scaling our result along the ray directions from the camera center, and compute the Mean Per Joint Position Error (MPJPE) in centimeter. The average MPJPE for all testing samples is 6.30 cm. We compute the average errors per each camera viewpoint, as shown in the top of Fig. 4.2. Each camera viewpoint is represented by azimuth and elevation with respect to the subjects' initial body location. We reach two interesting findings: first, the performance worsens in the camera views with higher elevation due to the severe self-occlusion and foreshortening; second, the error is larger in back views compared to the frontal views because limbs are occluded by torso in many poses. At the

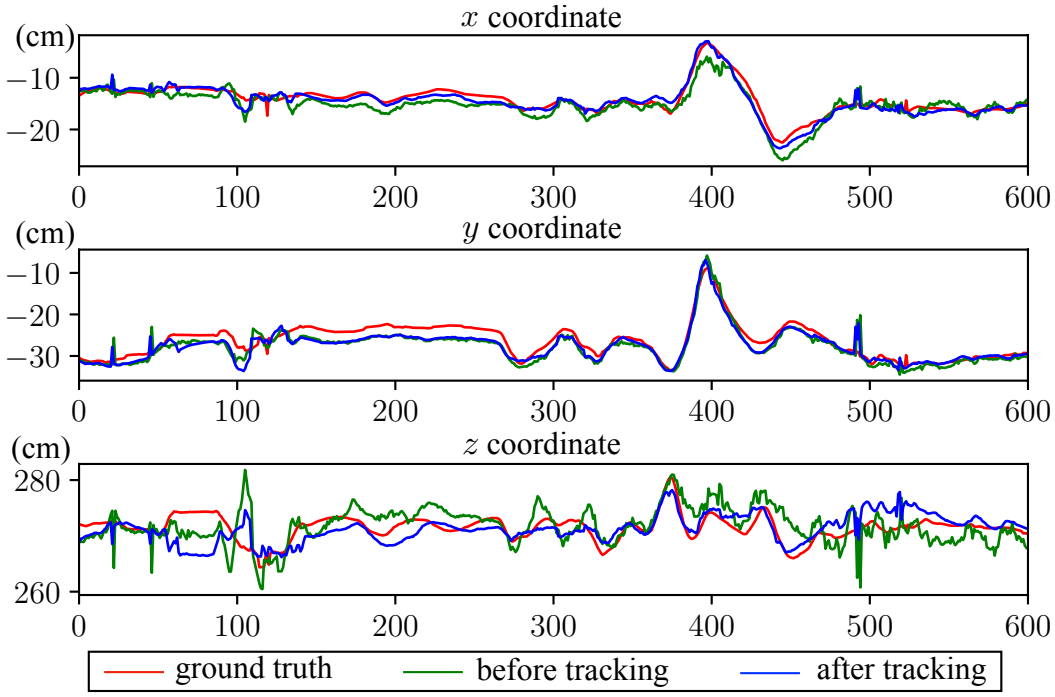

Figure 4.3: The comparison of joint location across time before and after tracking with ground truth. The horizontal axes show frame numbers (30fps) and the vertical axes show joint locations in camera coordinate. The target joint here is the left shoulder of the subject.

bottom of Fig. 4.2, we show the performance for varying body poses. We run k-means algorithm on the ground truth data to find body pose groups (the center poses are shown in the figure), and compute the error for each cluster. Body poses with more severe self-occlusion or foreshortening tend to have higher errors.

4.4 The Effect of Mesh Tracking

To demonstrate the effect of our temporal refinement method, we compare the result of our method before and after this refinement stage using Panoptic Studio data. We plot the reconstructed left shoulder joint in Fig. 4.3. We find that the result after tracking (in blue) tends to be more temporally stable than that before tracking (in green), and is often closer to the ground truth (in red).

4.5 Qualitative Evaluation

We demonstrate our total motion capture results in various videos captured by us or obtained from YouTube in the supplementary videos[1]. For videos where only the upper body

[1] `http://domedb.perception.cs.cmu.edu/mtc`

of the target person is visible, we assume that the orientation of torso and legs is pointing vertically downward in Eq. 3.5.

Chapter 5

Discussion

In this thesis, we present a method to reconstruct 3D total motion of a single person from an image or a monocular video. We thoroughly evaluate the robustness of our method on various benchmarks and demonstrate monocular 3D total motion capture results on in-the-wild videos.

There are some limitations with our method. First, we observe failure cases when a significant part of the target person is invisible (out of image boundary or occluded by other objects) due to erroneous network prediction. Second, our hand pose detector fails in the case of insufficient resolution, severe motion blur or occlusion by objects being manipulated. Third, we use a simple approach to estimating foot and facial expression that utilizes only 2D keypoint information. More advanced techniques and more image measurements can be incorporated into our method. Finally, our CNN requires bounding boxes for body and hands as input, and cannot handle multiple bodies or hands simultaneously. Solving these problems points to interesting future directions.

Appendix A

New 3D Human Pose Dataset

In this section, we provide more details of the new 3D human pose dataset that we collect.

A.1 Methodology

We build this dataset in 3 steps:

- We randomly recruit 40 volunteers on campus and capture their motion in a multi-view system [24, 25]. During the capture, all subjects follow the motion in the same video of around 2.5 minutes recorded in advance.
- We use multi-view 3D reconstruction algorithms [24, 25, 51] to reconstruct 3D body, hand and face keypoints.
- We run filters on the reconstruction results. We compute the average lengths of all bones for every subject, and discard a frame if the difference between the length of any bone in the frame and the average length is above a certain threshold. We further manually verify the correctness of hand annotations by projecting the skeletons onto 3 camera views and checking the alignment between the projection and images.

A.2 Statistics and Examples

To train our networks, we use our captured 3D body data and hand data, include a total of **834K** image-annotation pairs for bodies and **111K** pairs for hands. Example data are shown in Fig. A.1 and our supplementary video.

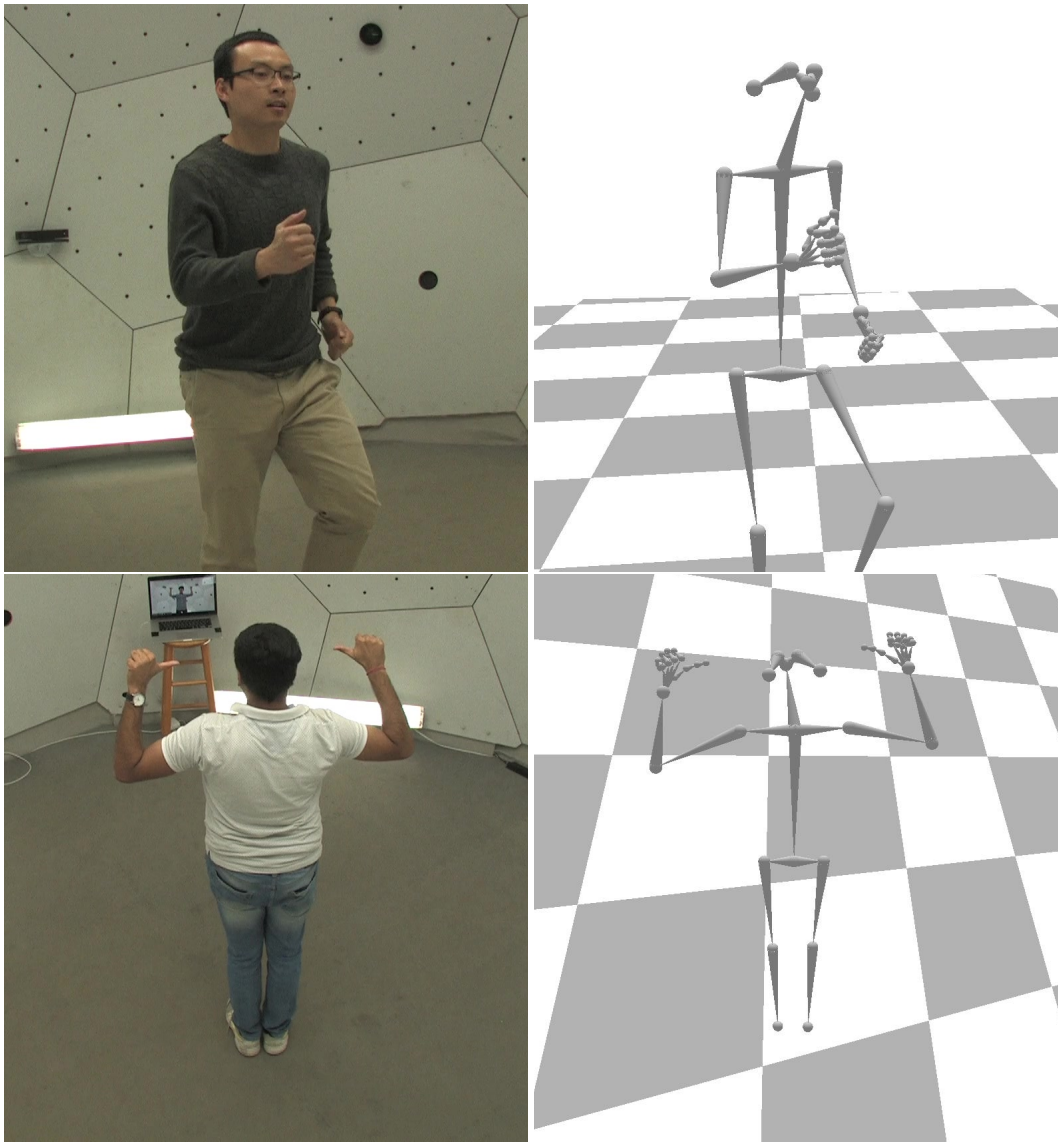

Figure A.1: Example images and 3D annotations from our new 3D human pose dataset.

Appendix B

Network Skeleton Definition

In this section we specify the skeleton hierarchy $\mathbb{S}$ we use for our Part Orientation Fields and joint confidence maps. As shown in Fig. B.1, we predict 18 keypoints for the body and POFs for 17 body parts, so $\mathbf{S}^B \in \mathbb{R}^{18\times368\times368}$, $\mathbf{L}^B \in \mathbb{R}^{51\times368\times368}$. Analogously, we predict 21 joints for each hand and POFs for 20 hand parts, so $\mathbf{S}^{LH}$ and $\mathbf{S}^{RH}$ have the dimension $21 \times 368 \times 368$, while $\mathbf{L}^{LH}$ and $\mathbf{L}^{RH}$ have the dimension $60 \times 368 \times 368$. Note that we train a CNN only for left hands, and we horizontally flip images of right hands before they are fed into the network during testing. Some example outputs of our CNN are shown in Fig. B.2, B.3, B.4, B.5.

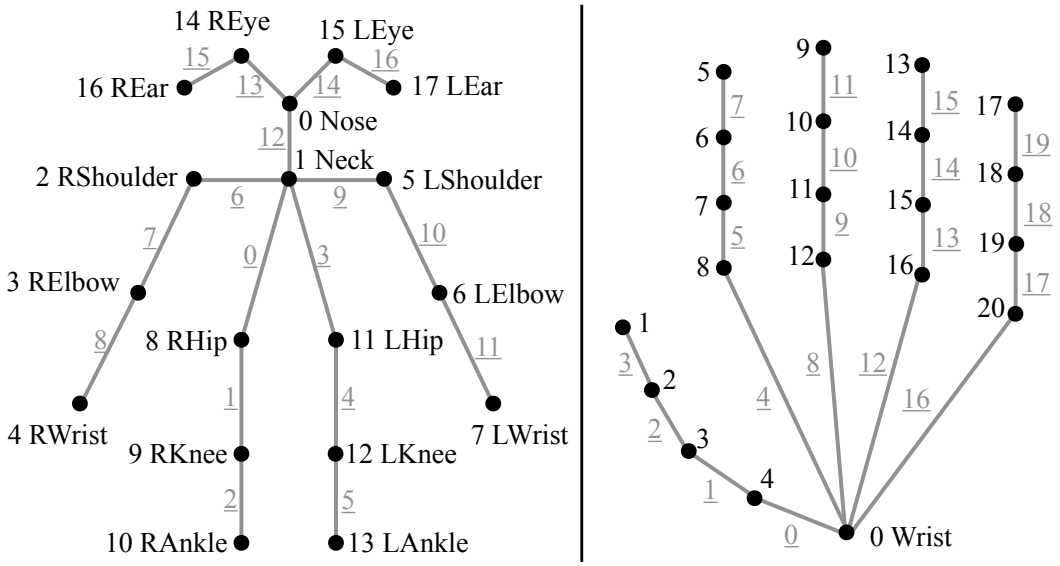

Figure B.1: Illustration on the skeleton hierarchy $\mathbb{S}$ in our POFs and joint confidence maps. The joints are shown in black, and body parts for POFs are shown in gray with indices underlined. On the left we show the skeleton used in our body network; on the right we show the skeleton used in our hand network.

Figure B.2: Joint confidence maps predicted by our CNN for a body image.

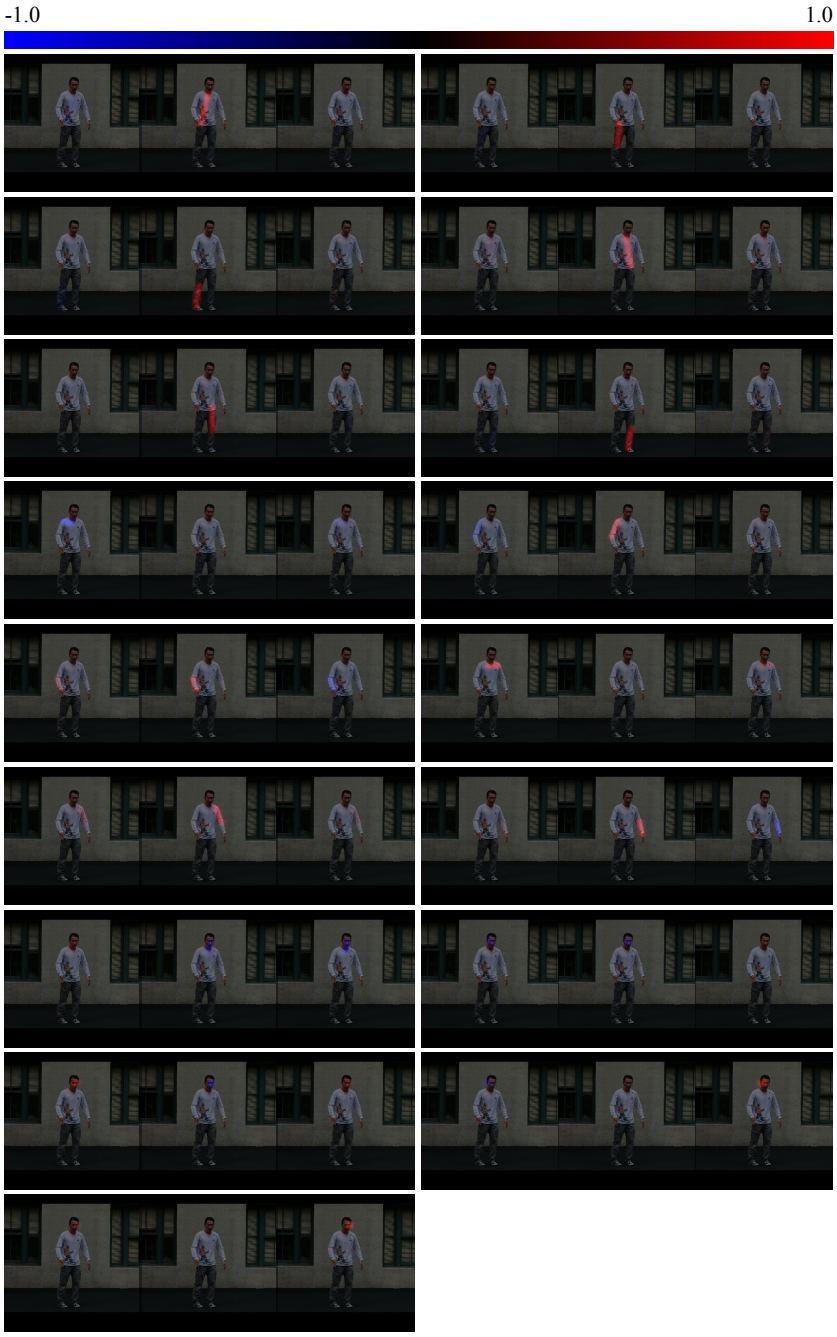

Figure B.3: Part Orientation Fields predicted by our CNN for a body image. For each body part we visualize x, y, z channels separately.

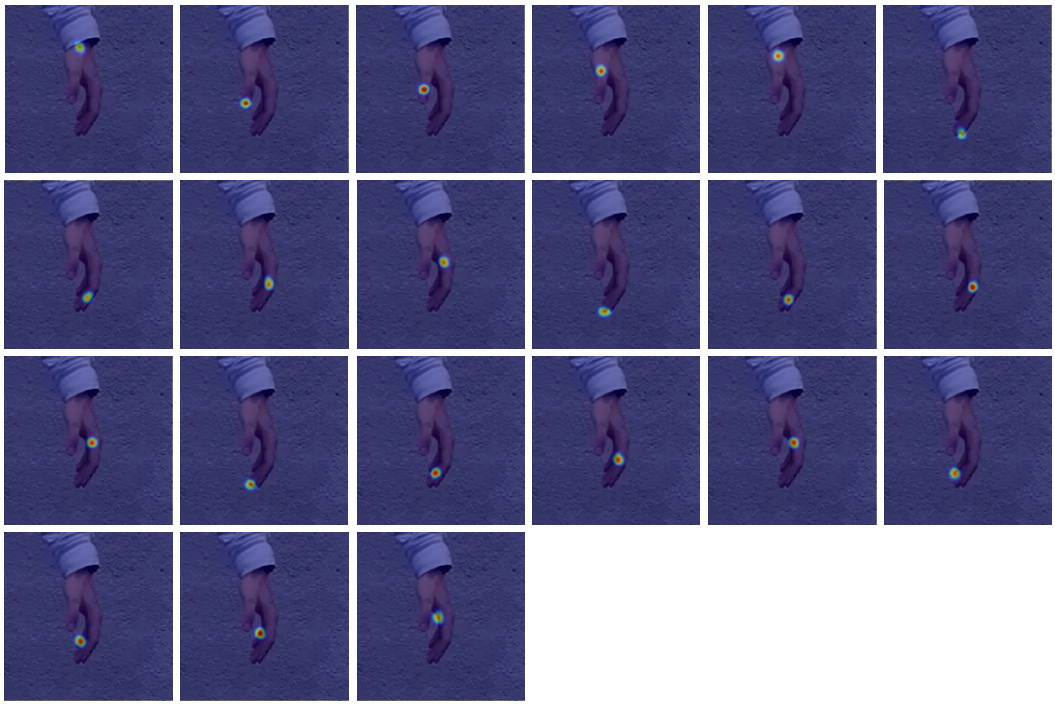

Figure B.4: Joint confidence maps predicted by our CNN for a hand image.

Figure B.5: Part Orientation Fields predicted by our CNN for a hand image. For each hand part we visualize x, y, z channels separately.

Appendix C

Deformable Human Model

C.1 Model Parameters

As explained in the main paper, we use Adam model introduced in [26] for total body motion capture. The model parameters $\boldsymbol{\Psi}$ include the shape parameters $\boldsymbol{\phi} \in \mathbb{R}^{K_\phi}$, where $K_\phi = 30$ is the dimension of shape deformation space, the pose parameters $\boldsymbol{\theta} \in \mathbb{R}^{J\times 3}$ where the $J = 62$ is the number of joints in the model[1], the global translation parameters $\boldsymbol{t} \in \mathbb{R}^3$, and the facial expression parameter $\boldsymbol{\sigma} \in \mathbb{R}^{K_\sigma}$ where $K_\sigma = 200$ is the number of facial expression bases.

C.2 3D Keypoints Definition

In this section we specify the correspondences between the keypoints predicted by our networks and Adam keypoints.

Regressors for the body are directly provided by [26], which define keypoints as linear combination of mesh vertices. During mesh fitting (Section 5 of the main paper), given current mesh $M(\boldsymbol{\Psi})$ determined by mesh parameters $\boldsymbol{\Psi} = (\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{t}, \boldsymbol{\sigma})$, we use these regressors to compute joints $\{\tilde{\mathbf{J}}^B_m\}$ from the mesh vertices, and further $\{\tilde{\mathbf{P}}^B_{(m,n)}\}$ by Eq. 3.1 in the main paper. $\{\tilde{\mathbf{J}}^B_m\}$ and $\{\tilde{\mathbf{P}}^B_{(m,n)}\}$ follow the skeleton structure in Fig. B.1. $\{\tilde{\mathbf{J}}^B_m\}$ and $\{\tilde{\mathbf{P}}^B_{(m,n)}\}$ are used in Eq. 3.4 and 3.5 in the main paper respectively to fit the body pose.

Joo *et al.* [26] also provides regressors for both hands, so we follow the same setup as body to define keypoints and hand parts $\{\tilde{\mathbf{J}}^{LH}_m\}, \{\tilde{\mathbf{J}}^{RH}_m\}, \{\tilde{\mathbf{P}}^{LH}_{(m,n)}\}, \{\tilde{\mathbf{P}}^{RH}_{(m,n)}\}$, which are used in Eq. 3.7 in the main paper to fit hand pose. Note that the wrists appear in both skeletons of Fig. B.1, so actually $\tilde{\mathbf{J}}^{LH}_0 = \tilde{\mathbf{J}}^B_7, \tilde{\mathbf{J}}^{RH}_0 = \tilde{\mathbf{J}}^B_4$. We only use 2D keypoint constraints from the body network, i.e., $\mathbf{j}^B_4, \mathbf{j}^B_7$ in Eq. 3.4, ignoring the keypoint measurements from hand network $\mathbf{j}^{LH}_0$ and $\mathbf{j}^{RH}_0$ in Eq. 3.7, since the body network is usually more stable in output.

For Eq. 3.8 in the main paper, we use 2D foot keypoint locations from OpenPose as $\{\mathbf{j}^T_m\}$, including big toes, small toes and heels of both feet. On the Adam side, we directly use mesh vertices as keypoints $\{\tilde{\mathbf{J}}^T_m\}$ for big toes and small toes on both feet. We use the middle point between a pair of vertices at the back of each feet as the heel keypoint, as shown in Fig. C.1 (left).

[1]The model has 22 body joints and 20 joints for each hand.

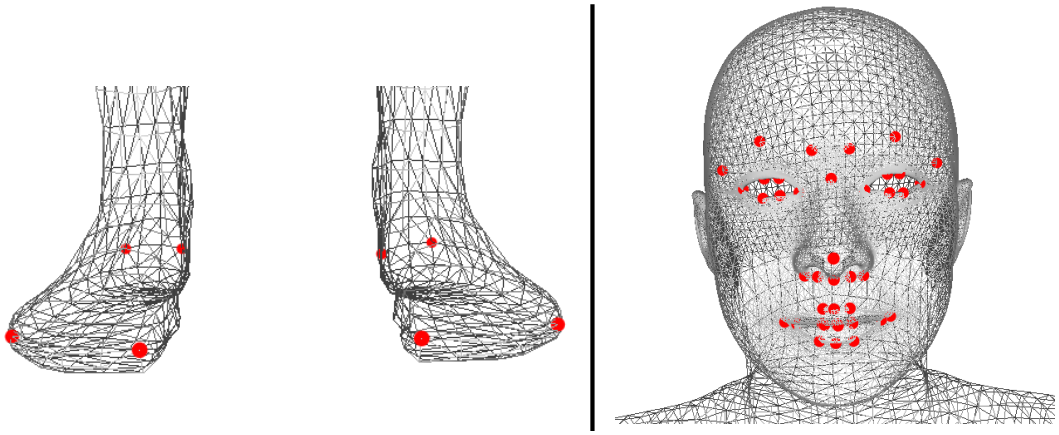

Figure C.1: We plot Adam vertices used as keypoints for mesh fitting in red dots. Left: vertices used to fit both feet (the middle points between the 2 vertices at the back are keypoints); right: vertices used to fit facial expression.

In order to get facial expression, we also directly fit Adam vertices using the 2D face keypoints predicted by OpenPose (Eq. 3.9 in the main paper). Note that although OpenPose provides 70 face keypoints, we only use 41 keypoints on eyes, nose, mouth and eyebrows, ignoring those on the face contour. The Adam vertices used for fitting are illustrated in Fig. C.1 (right).

Appendix D

Implmentation Details

In this chapter, we provide details about the parameters we use in our implementation.

In Eq. 3.5 and 3.6, we use

$$w^B_{\text{POF}} = 22500, w^B_p = 200.$$

We have similarly defined weights for left and right hands omitted in Eq. 3.7, for which we use

$$w^{LH}_{\text{POF}} = w^{RH}_{\text{POF}} = 2500, w^{LH}_p = w^{RH}_p = 10.$$

Weights for Eq. 3.10 (omitted in the main paper) are

$$w^{\boldsymbol{\phi}} = 0.01, w^{\boldsymbol{\sigma}} = 100.$$

In Eq. 3.14, a balancing weight is omitted for which we use

$$w_{\Delta z} = 0.25.$$

In Eq. 3.16, $\mathcal{F}_{\text{POF}}$ consists of POF terms for body, left hands and right hands, i.e., $\mathcal{F}_{\text{POF}} = \mathcal{F}^B_{\text{POF}} + \mathcal{F}^{LH}_{\text{POF}} + \mathcal{F}^{RH}_{\text{POF}}$. We use weights $25, 1, 1$ to balance these 3 terms.

Bibliography

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

[2] Vicon motion systems. `www.vicon.com`.

[3] S. Agarwal, K. Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[4] I. Akhter and M. J. Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *CVPR*, 2015.

[5] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.

[6] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *TOG*, 2005.

[7] A. Baak, M. M, G. Bharaj, H.-p. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *ICCV*, 2011.

[8] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999.

[9] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, 2016.

[10] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black. Dynamic faust: Registering human bodies in motion. In *CVPR*, 2017.

[11] A. Bulat and G. Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, 2016.

[12] Y. Cai, L. Ge, J. Cai, and J. Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV*, 2018.

[13] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *TVCG*, 2014.

[14] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. OpenPose: realtime multi-person 3D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*, 2018.

[15] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[16] D. Casas, M. Volino, J. Collomosse, and A. Hilton. 4d video textures for interactive character appearance. In *Computer Graphics Forum*, 2014.

[17] C.-H. Chen and D. Ramanan. 3d human pose estimation = 2d pose estimation + matching. In *CVPR*, 2017.

[18] R. Dabral, A. Mundhada, U. Kusupati, S. Afaque, A. Sharma, and A. Jain. Learning 3d human pose from structure and motion. In *ECCV*, 2018.

[19] A. Elhayek, E. Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt. Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras. In *CVPR*, 2015.

[20] H. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *AAAI*, 2018.

[21] J. Gall, C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *CVPR*, 2009.

[22] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 2014.

[23] U. Iqbal, P. Molchanov, T. Breuel Juergen Gall, and J. Kautz. Hand pose estimation via latent 2.5d heatmap regression. In *ECCV*, 2018.

[24] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *CVPR*, 2015.

[25] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. Nabbe, I. Matthews, et al. Panoptic studio: A massively multiview system for social interaction capture. *TPAMI*, 2017.

[26] H. Joo, T. Simon, and Y. Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *CVPR*, 2018.

[27] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[29] Y. Liu, J. Gall, C. Stoll, Q. Dai, H.-P. Seidel, and C. Theobalt. Markerless motion capture of multiple characters using multiview image segmentation. *TPAMI*, 2013.

[30] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *TOG*, 2015.

[31] C. Luo, X. Chu, and A. Yuille. Orinet: A fully convolutional network for 3d human pose estimation. In *BMVC*, 2018.

[32] D. C. Luvizon, D. Picard, and H. Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *CVPR*, 2018.

[33] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017.

[34] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. In *3DV*, 2018.

[35] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *TOG*, 2017.

[36] F. Moreno-noguer. 3d human pose estimation from a single image via distance matrix regression. In *CVPR*, 2017.

[37] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *CVPR*, 2018.

[38] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[39] B. X. Nie, P. Wei, and S.-C. Zhu. Monocular 3d human pose estimation by predicting depth on joints. In *ICCV*, 2017.

[40] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Tracking the articulated motion of two strongly interacting hands. In *CVPR*, 2012.

[41] G. Pavlakos, X. Zhou, and K. Daniilidis. Ordinal depth supervision for 3D human pose estimation. In *CVPR*, 2018.

[42] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *CVPR*, 2017.

[43] G. Pons-Moll, J. Romero, N. Mahmood, and M. J. Black. Dyna: A model of dynamic human shape in motion. *TOG*, 2015.

[44] V. Ramakrishna, T. Kanade, and Y. Sheikh. Reconstructing 3d human pose from 2d image landmarks. In *CVPR*, 2012.

[45] N. Robertini, D. Casas, H. Rhodin, H.-P. Seidel, and C. Theobalt. Model-based outdoor performance capture. In *3DV*, 2016.

[46] G. Rogez and C. Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *NIPS*, 2016.

[47] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *TOG*, 2017.

[48] M. R. Ronchi, O. Mac Aodha, R. Eng, and P. Perona. It's all relative: Monocular 3d human pose estimation from weakly supervised data. In *BMVC*, 2018.

[49] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *CHI*, 2015.

[50] J. Shotton, A. Fitzgibbon, M. Cook, and T. Sharp. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

[51] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[52] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt. Fast and robust hand tracking using detection-guided optimization. In *CVPR*, 2015.

[53] S. Sridhar, F. Mueller, M. Zollhöfer, D. Casas, A. Oulasvirta, and C. Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *ECCV*, 2016.

[54] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive markerless articulated hand motion tracking using RGB and depth data. In *ICCV*, 2013.

[55] X. Sun, J. Shang, S. Liang, and Y. Wei. Compositional human pose regression. In *ICCV*, 2017.

[56] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei. Integral human pose regression. In *ECCV*, 2018.

[57] A. Tagliasacchi, M. Schröder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly. Robust articulated-icp for real-time hand tracking. In *Computer Graphics Forum*, 2015.

[58] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *CVIU*, 2000.

[59] J. Taylor, V. Tankovich, D. Tang, C. Keskin, D. Kim, P. Davidson, A. Kowdle, and S. Izadi. Articulated distance fields for ultra-fast tracking of hands interacting. *TOG*, 2017.

[60] B. Tekin, A. Rozantsev, V. Lepetit, and P. Fua. Direct prediction of 3d body poses from motion compensated sequences. In *CVPR*, 2016.

[61] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Headon: Real-time reenactment of human portrait videos. *TOG*, 2018.

[62] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, 2016.

[63] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014.

[64] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014.

[65] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall. Capturing hands in action using discriminative salient points and physics simulation. *IJCV*, 2016.

[66] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *ECCV*, 2018.

[67] M. Wang, X. Chen, W. Liu, C. Qian, L. Lin, and L. Ma. Drpose3d: Depth ranking in 3d human pose estimation. In *IJCAI*, 2018.

[68] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[69] W. Xu, A. Chatterjee, M. Zollhoefer, H. Rhodin, D. Mehta, H.-P. Seidel, and C. Theobalt. Monoperfcap: Human performance capture from monocular video. *TOG*, 2018.

[70] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang. 3d human pose estimation in the wild by adversarial learning. In *CVPR*, 2018.

[71] Q. Ye, S. Yuan, and T.-K. Kim. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In *ECCV*, 2016.

[72] J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, and Q. Yang. 3d hand pose tracking and estimation using stereo matching. *arXiv preprint arXiv:1610.07214*, 2016.

[73] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *ICCV*, 2017.

[74] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single rgb images. In *ICCV*, 2017.