Monocular Total Capture: Posing Face, Body, and Hands in the Wild (Supplementary Material)

Donglai Xiang[1] Hanbyul Joo[1,2] Yaser Sheikh[1]
[1]Carnegie Mellon University [2]Facebook AI Research
{donglaix,hanbyulj,yaser}@cs.cmu.edu

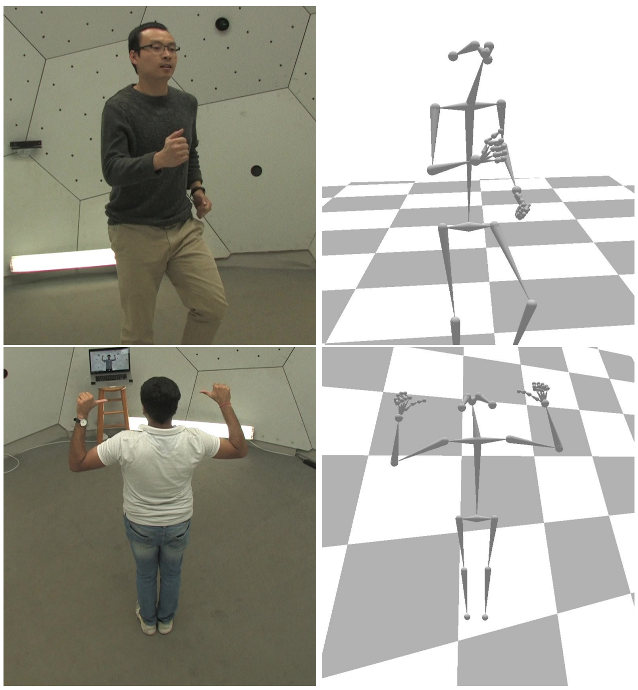

Figure 1. Example images and 3D annotations from our new 3D human pose dataset.

1. New 3D Human Pose Dataset

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

1.1. Methodology

We build this dataset in 3 steps:

- We randomly recruit 40 volunteers on campus and capture their motion in a multi-view system [1, 2]. 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 [1, 2, 4] 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.

1.2. Statistics and Examples

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

2. 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. 2, 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. 4, 5, 6, 7.

3. Deformable Human Model

3.1. Model Parameters

As explained in the main paper, we use Adam model introduced in [3] for total body motion capture. The model parameters $\mathbf{\Psi}$ 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 $\boldsymbol{\theta} \in \mathbb{R}^{J\times3}$ where the $J = 62$ is the number of joints in the model[1], the global transla-

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

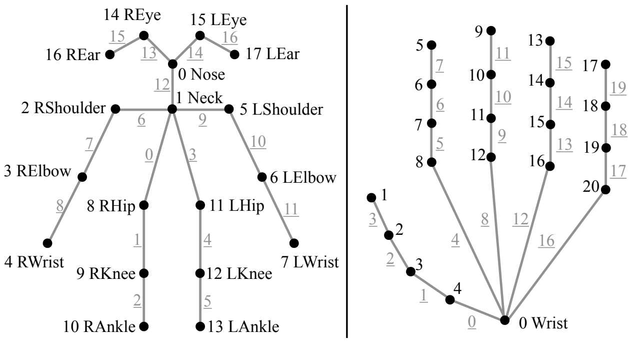

Figure 2. Illustration on the skeleton hierarchy $\mathbb{S}$ in our POFs and joint confidence maps. The joints are shown in black, and body parts for POFs are shown in gray with indices underlined. On the left we show the skeleton used in our body network; on the right we show the skeleton used in our hand network.

tion 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.

3.2. 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 [3], 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 $\{\tilde{\mathbf{J}}_m^B\}$ from the mesh vertices, and further $\{\tilde{\mathbf{P}}_{(m,n)}^B\}$ by Equation 1 in the main paper. $\{\tilde{\mathbf{J}}_m^B\}$ and $\{\tilde{\mathbf{P}}_{(m,n)}^B\}$ follow the skeleton structure in Fig. 2. $\{\tilde{\mathbf{J}}_m^B\}$ and $\{\tilde{\mathbf{P}}_{(m,n)}^B\}$ are used in Equation 4 and 5 in the main paper respectively to fit the body pose.

Joo *et al*. [3] also provides regressors for both hands, so we follow the same setup as body to define keypoints and hand parts $\{\tilde{\mathbf{J}}_m^{LH}\}, \{\tilde{\mathbf{J}}_m^{RH}\}, \{\tilde{\mathbf{P}}_m^{LH}\}, \{\tilde{\mathbf{P}}_m^{RH}\}$, which are used in Equation 7 in the main paper to fit hand pose. Note that the wrists appear in both skeletons of Fig. 2, so actually $\tilde{\mathbf{J}}_0^{LH} = \tilde{\mathbf{J}}_7^B, \tilde{\mathbf{J}}_0^{RH} = \tilde{\mathbf{J}}_4^B$. We only use 2D keypoint constraints from the body network, i.e., $\mathbf{j}_4^B, \mathbf{j}_7^B$ in Equation 4, ignoring the keypoint measurements from hand network $\mathbf{j}_0^{LH}$ and $\mathbf{j}_0^{RH}$ in Equation 7, since the body network usually produces more stable output.

For Equation 8 in the main paper, we use 2D foot keypoint locations from OpenPose as $\{\mathbf{j}_m^T\}$, including big toes, small toes and heels of both feet. On the Adam side, we directly use mesh vertices as keypoints $\{\tilde{\mathbf{J}}_m^T\}$ 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. 3 (left).

In order to get facial expression, we also directly fit

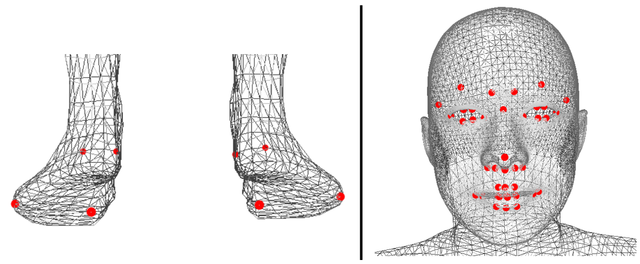

Figure 3. We plot Adam vertices used as keypoints for mesh fitting in red dots. Left: vertices used to fit both feet (the middle points between the 2 vertices at the back are keypoints); right: vertices used to fit facial expression.

Adam vertices using the 2D face keypoints predicted by OpenPose (Equation 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. 3 (right).

4. Implmentation Details

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

In Equation 4 and 5 of the main paper, we use

$$w_{\text{POF}}^B = 22500, w_p^B = 200.$$

We have similarly defined weights for left and right hands omitted in Equation 7, for which we use

$$w_{\text{POF}}^{LH} = w_{\text{POF}}^{RH} = 2500, w_p^{LH} = w_p^{RH} = 10.$$

Weights for Equation 10 (omitted in the main paper) are

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

In Equation 15, a balancing weight is omitted for which we use

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

In Equation 16, $\mathcal{F}_{\text{POF}}$ consists of POF terms for body, left hands and right hands, i.e., $\mathcal{F}_{\text{POF}} = \mathcal{F}_{\text{POF}}^B + \mathcal{F}_{\text{POF}}^{LH} + \mathcal{F}_{\text{POF}}^{RH}$. We use weights $25, 1, 1$ to balance these 3 terms.

References

[1] 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.

[2] 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.

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

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

Figure 4. Joint confidence maps predicted by our CNN for a body image.

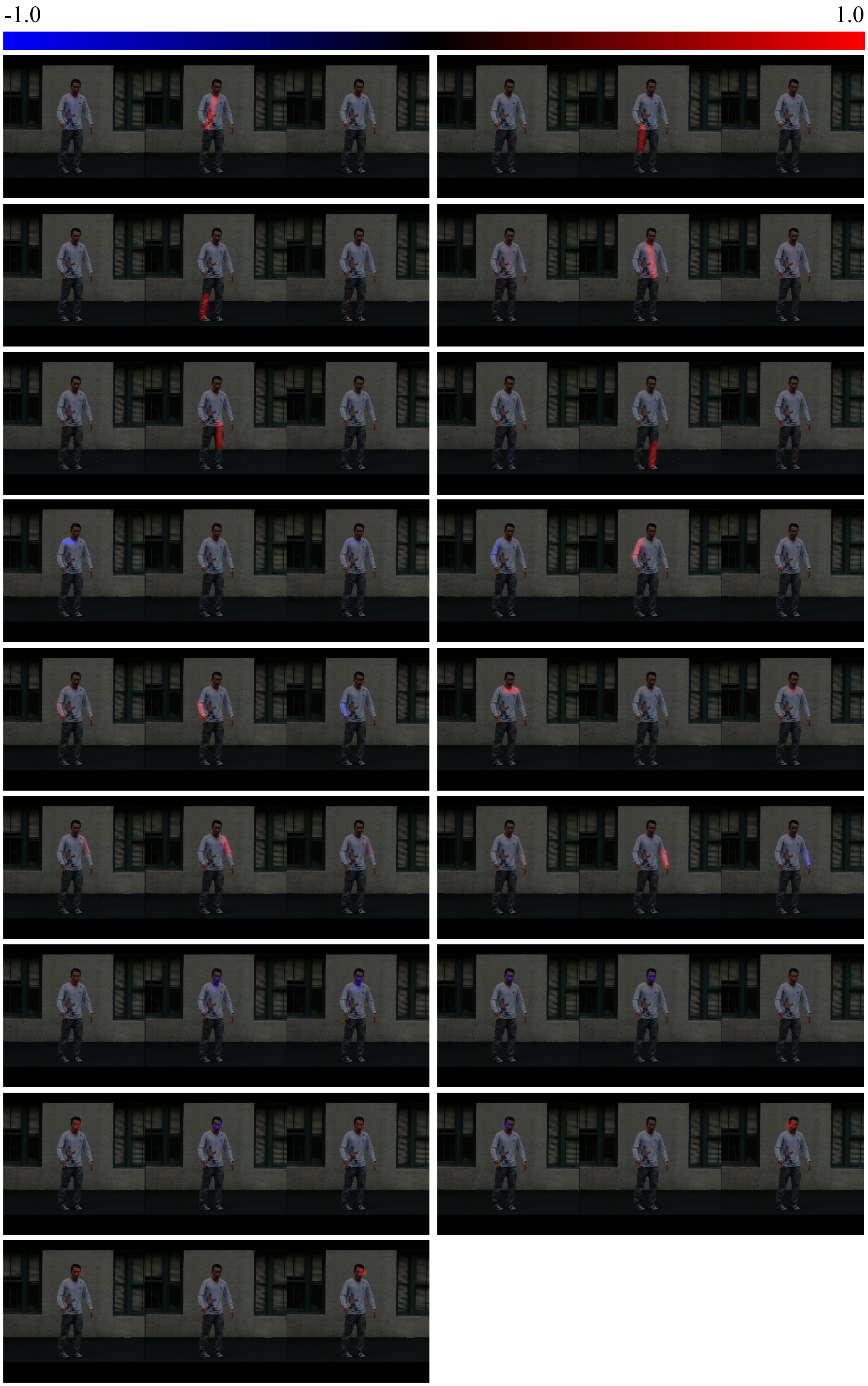

Figure 5. Part Orientation Fields predicted by our CNN for a body image. For each body part we visualize x, y, z channels separately.

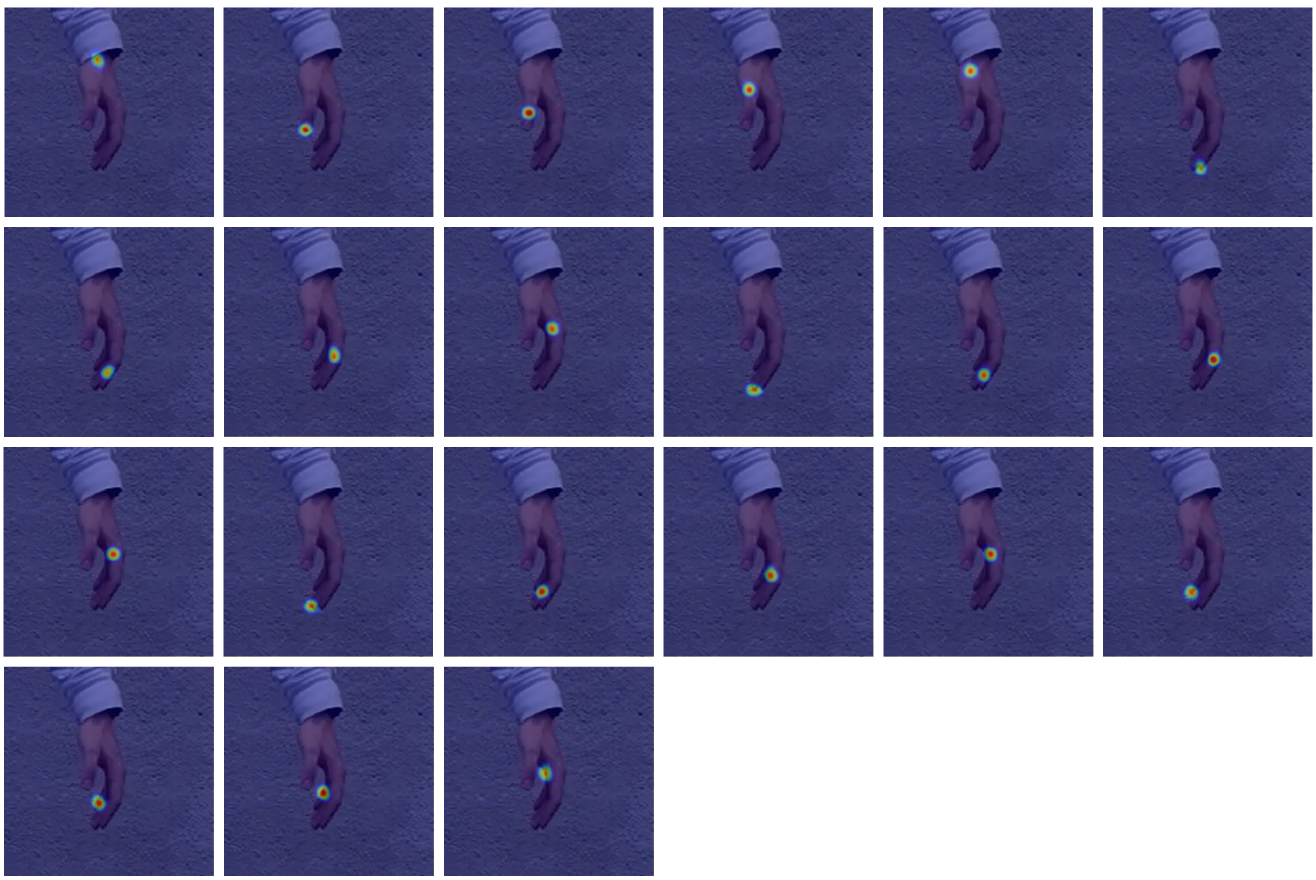

Figure 6. Joint confidence maps predicted by our CNN for a hand image.

Figure 7. Part Orientation Fields predicted by our CNN for a hand image. For each hand part we visualize x, y, z channels separately.