Multi-Person 3D Pose and Shape Estimation via Inverse Kinematics and Refinement – Supplementary Material

Junuk Cha[1 [0000−0003−2321−2797]], Muhammad Saqlain[1,2† [0000−0001−5877−6432]], GeonU Kim[1‡], Mingyu Shin[1,3‡], and Seungryul Baek[1 [0000−0002−0856−6880]]

[1]UNIST, South Korea [2]eSmart Systems, Norway [3]Yeongnam Univ., South Korea

In this supplemental document, we try to involve more detailed explanation about our pipeline. For the purpose, we draw the schematic diagram for the entire framework in Fig. 1 and detail the overall sequence of the framework including the rearrange and concatenation method for $\mathbf{F}_{\text{img}}$ and $\boldsymbol{\Theta}_{\text{init}}$ in Sec. 1. We further offer more details on our training process in Sec. 2, summarize how to calculate absolute 3D skeletons $\mathbf{P}$ in Sec. 3 and discuss the additional quantitative results in Sec. 4. Finally, we present additional qualitative examples in Sec. 5.

1 Overall sequence of framework

In this section, we detail the overall sequence of framework.

Feature extraction. We use ResNet architecture [2] to extract image feature $\mathbf{F}_{\text{img}} \in \mathbb{R}^{8\times8\times2048}$ from the cropped image $\mathbf{X} \in \mathbb{R}^{256\times256\times3}$. The cropped image $\mathbf{X}$ goes through CONV, BN, ReLU, MAX Pool, Block 1, Block 2, Block 3, and Block 4 layers to produce the $8 \times 8 \times 2,048$-dimensional array. This feature array is used to further 1) estimate initial skeletons from the initial skeleton estimation network f^{P}, 2) estimate the twist angle Φ and shape $\boldsymbol{\beta}_{\text{init}}$ from twist angle and shape estimator f^{TS} and 3) refine the initial mesh parameters via the relation-aware refiner f^{Ref}.

Initial skeleton estimation network $f^{\mathbf{P}}$. As denoted in Table 1, the obtained feature $\mathbf{F}_{\text{img}}$ is fed to the 1×1 conv1 layer to generate 3D heatmaps whose dimensions are $8\times8\times8\times K$. After that, 3D soft-argmax operation is applied on the 3D heatmaps to obtain the root-relative 3D skeletons $\mathbf{P}_{\text{rel}}$. Similarly, $\mathbf{F}_{\text{img}}$ is fed to 1×1 conv2 layer to generate 2D heatmaps whose dimensions are $8\times8\times K$. Then, 2D soft-argmax operation is applied on the 2D heatmaps to generate 2D skeletons $\mathbf{P}_{\text{img}}$. Finally, the absolute 3D skeletons $\mathbf{P}$ are differentiably calculated by combining $\mathbf{P}_{\text{img}}$ and $\mathbf{P}_{\text{rel}}$ with camera intrinsic matrix as in [18].

Global average pooling. To be used for twist angle and shape estimator f^{TS} and relation-aware refiner f^{Ref}, the global average pooling (GAP) is applied on $\mathbf{F}_{\text{img}}$ to reduce its dimension to $1 \times 1 \times 2,048$.

Twist and shape estimation network $f^{\mathbf{TS}}$. The twist and shape estimation network f^{TS} is composed of multiple fully-connected layers as denoted in Table 2. It first changes the $1\times1\times2,048$-dimensional feature array into the $1\times1\times1,024$-dimensional array via two fully-connected layers (i.e. FC1 and FC2). Then, it

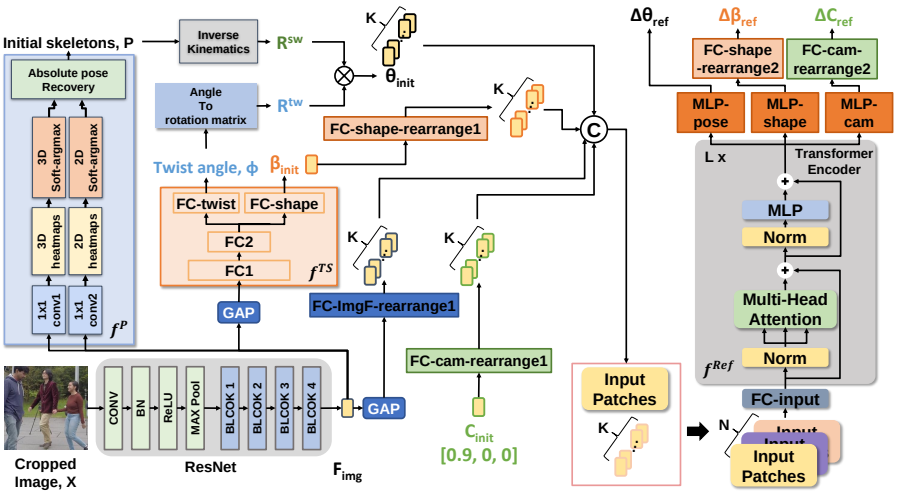

Fig. 1: The detailed diagram of our entire framework. From the cropped image $\mathbf{X}$, feature array $\mathbf{F}_{\text{img}}$ is extracted and it is used for 1) initial skeleton estimation via f^{P}, for 2) estimating initial mesh parameter $\boldsymbol{\Theta}_{\text{init}} = [\boldsymbol{\theta}_{\text{init}}; \boldsymbol{\beta}_{\text{init}}; \mathbf{C}_{\text{init}}]$ via f^{TS} and for 3) refining the initial mesh parameter $\boldsymbol{\Theta}_{\text{init}}$ by $\boldsymbol{\Theta}_{\text{ref}} = \boldsymbol{\Theta}_{\text{init}} + \Delta\boldsymbol{\Theta}_{\text{ref}}$ where $\Delta\boldsymbol{\Theta}_{\text{ref}} = [\Delta\boldsymbol{\theta}_{\text{ref}}; \Delta\boldsymbol{\beta}_{\text{ref}}; \Delta\mathbf{C}_{\text{ref}}]$ via f^{Ref}. Overall sequences are described in Sec. 1 and detailed architecture for f^{P}, f^{TS} and f^{Ref} are shown in Tables 1, 2 and 3, respectively.

maps it to the twist angle vector $\boldsymbol{\Phi} \in \mathbb{R}^{K\times 2}$ and shape vector $\boldsymbol{\beta}_{\text{init}} \in \mathbb{R}^{1\times 10}$ via respective fully-connected layers (i.e. FC-twist, FC-shape).

Relation-aware refinement network f^{Ref}. While both initial skeleton estimation and twist and shape estimation have been performed individually for each person in each image, the relation-aware refinement network f^{Ref} needs to deal with multi-persons contained in the image altogether. For the purpose, we rearrange and concatenate the intermediate output vectors as follows:

Concatenating responses for N persons. The intermediate feature obtained by applying GAP on the $\mathbf{F}_{\text{img}}$ is the $1\times1\times2,048$-dimensional array. We concatenate it for N sampled persons and reshape it towards the $N \times 1 \times 2,048$-dimensional array. In parallel, the pose parameter $\boldsymbol{\theta}_{\text{init}}$, shape parameter $\boldsymbol{\beta}_{\text{init}}$ and camera parameter $\mathbf{C}_{\text{init}}$ are concatenated for N persons and reshaped into $N \times K \times 6$-dimensional array, $N \times 1 \times 10$ and $N \times 1 \times 3$-dimensional array, respectively.

Re-arranging vectors for K joints. Afterwards, we need to align the dimension of the shape $\boldsymbol{\beta}_{\text{init}}$, camera parameters $\mathbf{C}_{\text{init}}$ and feature $\mathbf{F}_{\text{img}}$ to that of the pose parameter $\boldsymbol{\theta}_{\text{init}}$ obtained for each joint. For the purpose, we apply fully-connected layers (i.e. FC-shape-rearrange1, FC-cam-rearrange1 and FC-imgF-rearrange1 layers) to make their second dimensions as K. As a result, we obtain the $N\times K\times10$-dimensional array, $N\times K\times6$-dimensional array and $N\times K\times2,048$-dimensional array. Finally, we obtain the $N \times K \times 2,067$-dimensional array by

concatenating resultant shape, camera, pose and image feature arrays. This is the input to the relation-aware refinement network f^{Ref} whose architecture is denoted in Table 3.

Transformer outputs. From the Transformer, three arrays (i.e. $N \times K \times 6$-dimensional array, $N \times K \times 10$-dimensional array and $N \times K \times 3$-dimensional array) are obtained. The first array is the residual pose vector $\Delta\boldsymbol{\theta}_{\text{ref}} \in \mathbb{R}^{N \times K \times 6}$. The second and third arrays are transformed via FC-shape-rearrange2 and FC-cam-rearrange2 layers towards $N \times 1 \times 10$ and $N \times 1 \times 3$-dimensional arrays, respectively to obtain the shape $\Delta\boldsymbol{\beta}_{\text{ref}} \in \mathbb{R}^{N \times 1 \times 10}$, and camera $\Delta\mathbf{C}_{\text{ref}} \in \mathbb{R}^{N \times 1 \times 3}$ residual vectors.

Pose and shape discriminators D_θ and D_β. We used the same architecture of [8] as pose discriminator D_θ and shape discriminator D_β.

Table 1: Architecture of initial 3D skeleton estimation network f^{P}. Input is $\mathbf{F}_{\text{img}}$ from ResNet.

Layer	Operation	Kernel	Dimensionality
	Input: $\mathbf{F}_{\text{img}}$	-	$8 \times 8 \times 2048$
1×1 conv1	Conv.	1×1	$8 \times 8 \times 8 \times K$
3D Soft-argmax	3D Soft-argmax	-	$K \times 3$
1×1 conv2	Conv.	1×1	$8 \times 8 \times K$
2D Soft-argmax	2D Soft-argmax	-	$K \times 2$
Absolute pose recovery	-	-	$K \times 3$

Table 2: Architecture of twist angle and shape estimation network f^{TS}. Input is global average pooled $\mathbf{F}_{\text{img}}$ from ResNet.

Layer	Operation	Kernel	Dimensionality
	Input: global average pooled $\mathbf{F}_{\text{img}}$	-	2048
FC1	Linear + dropout(0.5)	-	1024
FC2	Linear + dropout(0.5)	-	1024
FC-twist	Linear	-	$K \times 2$
FC-shape	Linear	-	10

Table 3: Architecture of relation-aware refiner f^{Ref}. Input is $\boldsymbol{\Theta}_{\text{init}}$ concatenated with global average pooled $\mathbf{F}_{\text{img}}$ from ResNet.

Layer	Operation	Kernel	Dimensionality
	Input: Input patches	-	$N \times K \times 2067$
FC-input	Linear	-	$N \times K \times 1024$
Norm	LayerNorm	-	$N \times K \times 1024$
Multi-Head Attention	Attention	-	$N \times K \times 1024$
Norm	LayerNorm	-	$N \times K \times 1024$
MLP	Linear + GELU	-	$N \times K \times 2048$
	Linear	-	$N \times K \times 1024$
Norm	LayerNorm	-	$N \times K \times 1024$
Multi-Head Attention	Attention	-	$N \times K \times 1024$
Norm	LayerNorm	-	$N \times K \times 1024$
MLP	Linear + GELU	-	$N \times K \times 2048$
	Linear	-	$N \times K \times 1024$
MLP-pose	Linear	-	$N \times K \times 6$
MLP-shape	Linear	-	$N \times K \times 512$
	Linear	-	$N \times K \times 10$
MLP-cam	Linear	-	$N \times K \times 512$
	Linear	-	$N \times K \times 3$

Algorithm 1: The summary of our entire training process

Input: Image I
Output: P, $\boldsymbol{\beta}_{\mathbf{init}}$**,** ϕ**,** $\Theta_{\mathbf{init}}$**,** $\Theta_{\mathbf{ref}}$
for $t = 1, \ldots, T$ **do**
  Crop the image to produce $\mathbf{X}$ using M bounding boxes.
  Obtain initial 3D skeletons ($\mathbf{P}$) of each person from f^{P}.
  Obtain SMPL shape parameters ($\boldsymbol{\beta}_{\mathrm{init}}$) and twist angles ($\boldsymbol{\Phi}$) from f^{TS}.
  Obtain initial SMPL pose parameters ($\boldsymbol{\Theta}_{\mathrm{init}}$) by inverse kinematics.
  Sample $\boldsymbol{\Theta}_{\mathrm{init}}$ of N persons.
  Feed them to f^{Ref} concatenated with image features ($\mathbf{F}_{\mathrm{img}}$) and refine them to get $\boldsymbol{\Theta}_{\mathrm{ref}}$.
  Feed $\boldsymbol{\Theta}_{\mathrm{ref}}$ to discriminators D_θ and D_β.
  Calculate gradient ∇L (Eq. 1) and update f^{P}, f^{TS}, and f^{Ref}
end

2 More details on training process

We obtain the initial skeleton estimation network f^{P}, twist angle and shape estimation network f^{TS} and relation-aware refinement network f^{Ref} via optimizing the following loss functions:

$$L(f^{\mathrm{P}}, f^{\mathrm{TS}}, f^{\mathrm{Ref}}) = L_{\mathrm{P}}(f^{\mathrm{P}}) + L_{\mathrm{TS}}(f^{\mathrm{TS}}) + L_{\mathrm{Ref}}(f^{\mathrm{Ref}}). \tag{1}$$

where individual terms are defined in the main paper. The overall training procedure is summarized in Algorithm 1.

Datasets. For training, we involved multiple datasets to train our framework. We used the full or partial losses for each datasets according to their ground-truth types: Human3.6M [4] dataset is used for calculating the full losses (i.e. $L_{\mathrm{P}}(f^{\mathrm{P}})$, $L_{\mathrm{TS}}(f^{\mathrm{TS}})$, and $L_{\mathrm{Ref}}(f^{\mathrm{Ref}})$), as it provides the SMPL pose and shape parameters, 3D skeleton, 2D skeleton ground truths. MPI-INF-3DHP [15] is used for calculating the partial losses $L_{\mathrm{P}}(f^{\mathrm{P}})$ and $L_{\mathrm{pose}}(f^{\mathrm{Ref}})$, as it provides only 2D and 3D skeleton ground-truths. LSP [6], MSCOCO [13] and MPII [1] datasets are used to calculate only the 2D losses in $L_{\mathrm{P}}(f^{\mathrm{P}})$ and $L_{\mathrm{pose}}(f^{\mathrm{Ref}})$, as they provide only 2D skeleton ground-truth. Additionally, MuCo-3DHP [16], CMU-Panoptic [7], SAIL-VOS [3], SURREAL [21], AIST++ [12] are used to calculate the $L_{\mathrm{P}}(f^{\mathrm{P}})$.

3 Details on calculating P

We followed [18] for calculating the absolute 3D skeletons $\mathbf{P}$. Our aim is obtaining absolute 3D skeletons $\mathbf{P}$ from root-relative 3D skeletons $\mathbf{P}_{\mathrm{rel}}$ and 2D skeletons $\mathbf{P}_{\mathrm{img}}$. The root-relative 3D skeletons $\mathbf{P}_{\mathrm{rel}}$, 2D skeletons $\mathbf{P}_{\mathrm{img}}$, and absolute 3D skeletons can be expressed as $\{(\mathbf{X}_k, \mathbf{Y}_k, \mathbf{Z}_k)\}_{k=1}^{K}$, $\{(x_k, y_k)\}_{k=1}^{K}$, and $\{(\mathbf{X}_k + \mathbf{X}_o, \mathbf{Y}_k + \mathbf{Y}_o, \mathbf{Z}_k + \mathbf{Z}_o)\}_{k=1}^{K}$, respectively, where K is the number of joints

Table 4: Ablation study on loss on 3DPW.

Method	MPJPE(↓)	dOrder(↑)
Ours-L_{pose}-L_{adv}	68.8	47.9
Ours-L_{mesh}	66.3	94.7
Ours-L_{pose}	67.0	51.2
Ours-L_{adv}	67.0	95.3
Ours	66.0	96.5

Table 5: The effectiveness of refiner f^{Ref} on 3DPW.

Method	MPJPE(↓)
HybrIK [11]	80.0
HybrIK [11] w/ Ref	78.4
Ours w/o Ref	67.3
Ours	66.0

Table 6: Inference frame rate.

Method	FPS
ROMP [11]	22.7
PARE [11]	21.4
SPEC [11]	36.4
METRO [11]	16.6
HybrIK [11]	24.8
Ours	21.5

and $(\mathbf{X}_o, \mathbf{Y}_o, \mathbf{Z}_o)$ is the offset. We should recover the offset $(\mathbf{X}_o, \mathbf{Y}_o, \mathbf{Z}_o)$. A normalized image coordinates can be calculated as $(\tilde{x}_k, \tilde{y}_k)^T = K^{-1}(x_k, y_k)^T$. It can be expressed as follows:

$$\begin{bmatrix} \tilde{x}_k \\ \tilde{y}_k \end{bmatrix} = \begin{bmatrix} (\mathbf{X}_k + \mathbf{X}_o)/(\mathbf{Z}_k + \mathbf{Z}_o) \\ (\mathbf{Y}_k + \mathbf{Y}_o)/(\mathbf{Z}_k + \mathbf{Z}_o) \end{bmatrix} \tag{2}$$

where $\tilde{x}_k$, $\tilde{y}_k$, $\mathbf{X}_k$, $\mathbf{Y}_k$, and $\mathbf{Z}_k$ are estimated values. This equation can be arranged to

$$\begin{bmatrix} \mathbf{X}_o - \tilde{x}_k \mathbf{Z}_o \\ \mathbf{Y}_o - \tilde{y}_k \mathbf{Z}_o \end{bmatrix} = \begin{bmatrix} \tilde{x}_j \mathbf{Z}_k - \mathbf{X}_k \\ \tilde{y}_j \mathbf{Z}_k - \mathbf{Y}_k \end{bmatrix} \tag{3}$$

There is K joints, thus we can obtain $2K$ linear equations with the variables $(\mathbf{X}_o, \mathbf{Y}_o, \mathbf{Z}_o)$. We used Cholesky decomposition to solve them. After calculating the offset $(\mathbf{X}_o, \mathbf{Y}_o, \mathbf{Z}_o)$, the absolute 3D skeletons $\mathbf{P}$ whose 2D skeleton lies outside the image are calculated as $(\mathbf{X}_k + \mathbf{X}_o, \mathbf{Y}_k + \mathbf{Y}_o, \mathbf{Z}_k + \mathbf{Z}_o)^T$ and the others are calculated as $(\tilde{x}_k, \tilde{y}_k, 1)^T \cdot (\mathbf{Z}_k + \mathbf{Z}_o)$.

4 More quantitative results

4.1 Ablation study on losses

In this section, We studied the effectiveness of losses L_{pose}, L_{mesh}, and L_{adv} for training f^{Ref}. The results are shown in Table 4. When using only L_{mesh}, the f^{ref} is trained on Human3.6M which has SMPL parameters ground-truth. Its generalization performance is worse than using a variety of dataset. When we do not involve L_{mesh}, it is trained on various datasets (MPI-INF-3DHP, LSP, COCO, etc.). Its performance is better than other methods except 'Ours'. When we do not involve L_{pose}, it is trained on Human3.6M by L_{mesh} and various datasets by L_{adv}. It has better performance compared to 'Ours-L_{pose}-L_{adv}'. L_{adv} punishes it for wrong estimated pose and shape parameters. When we do not involve L_{adv}, it is trained on a variety of datasets and has better performance compared to 'Ours-L_{pose}-L_{adv}'. Finally, the method using three losses (ie. L_{pose}, L_{mesh} and L_{adv}) has the best performance. Further, we extend our framework to use the original camera parameter and measure the depth-order accuracy similar to [5, 20, 22] and its result is shown in 'dOrder' column of the Table. We could see that results are gradually increasing as more loss functions are involved.

4.2 Effectiveness of refiner $f^{\mathbf{Ref}}$

We conducted the experiment to find out whether the refiner module f^{Ref} is really improve the final mesh quality. Table 5 shows that mesh obtained from HybrIK [11] can be improved by our refiner module f^{Ref} as well as our coarse mesh. However, its gain is limited because HybrIK uses less competitive 3D poses.

4.3 Inference time

We compare the inference frame rate of our method with other state-of-the-arts. In Table 6, it is confirmed that the inference speed is comparable to other methods.

Fig. 2: Qualitative comparisons with HybrIK [11]. Red circles highlight wrongly estimated parts.

5 More qualitative results

Fig. 2 shows our qualitative results compared to HybrIK [11]. In Figs. 3, 4 and 5, we present more qualitative results compared to three state-of-the-art methods [10, 19, 9] on three multi-person pose estimation benchmark datasets (i.e. 3DPW [14], AGORA [16] and MuPoTS [17]), respectively. Fig. 6 illustrates the intermediate outputs (i.e. initial skeletons, initial meshes obtained by inverse kinematics and refined meshes) obtained from our pipeline. Figs. 7 and 8 show the top-view and second-view results and failure cases, respectively.

Fig. 3: Qualitative comparisons on 3DPW [14]. Red circles highlight wrongly estimated parts.

Fig. 4: Qualitative comparisons on AGORA [17]. Red circles highlight wrongly estimated parts.

Fig. 5: Qualitative comparisons on MuPoTS [16]. Red circles highlight wrongly estimated parts.

(a) (b) (c) (d)

Fig. 6: Example outputs from our pipeline: (a) input RGB image, (b) initial skeleton estimation results obtained from the input image, (c) initial meshes obtained from the inverse kinematics process, (d) refined meshes obtained from the refinement Transformer.

(a) (b) (c) (d)

Fig. 7: Example outputs from our pipeline: (a) input RGB image, (b) refined meshes overlaid on input RGB image, (c) top view, (d) side view.

(a) (b) (c) (d)

Fig. 8: Failure cases: (a) input RGB image, (b) initial skeleton estimation results obtained from the input image, (c) initial meshes obtained from the inverse kinematics process, (d) refined meshes obtained from the refinement Transformer.

References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2d human pose estimation: New benchmark and state of the art analysis. In: CVPR (2014)
2. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: ECCV (2016)
3. Hu, Y.T., Chen, H.S., Hui, K., Huang, J.B., Schwing, A.G.: Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines. In: CVPR (2019)
4. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. TPAMI (2013)
5. Jiang, W., Kolotouros, N., Pavlakos, G., Zhou, X., Daniilidis, K.: Coherent reconstruction of multiple humans from a single image. In: CVPR (2020)
6. Johnson, S., Everingham, M.: Clustered pose and nonlinear appearance models for human pose estimation. In: BMVC (2010)
7. Joo, H., Liu, H., Tan, L., Gui, L., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., Sheikh, Y.: Panoptic studio: A massively multiview system for social motion capture. In: ICCV (2015)
8. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: CVPR (2018)
9. Kocabas, M., Huang, C.H.P., Hilliges, O., Black, M.J.: Pare: Part attention regressor for 3d human body estimation. In: ICCV (2021)
10. Kocabas, M., Huang, C.H.P., Tesch, J., Müller, L., Hilliges, O., Black, M.J.: Spec: Seeing people in the wild with an estimated camera. In: ICCV (2021)
11. Li, J., Xu, C., Chen, Z., Bian, S., Yang, L., Lu, C.: Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In: CVPR (2021)
12. Li, R., Yang, S., Ross, D.A., Kanazawa, A.: Learn to dance with aist++: Music conditioned 3d dance generation. arXiv:2101.08779 (2021)
13. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
14. von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: ECCV (2018)
15. Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., Theobalt, C.: Monocular 3d human pose estimation in the wild using improved cnn supervision. In: 3DV (2017)
16. Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Sridhar, S., Pons-Moll, G., Theobalt, C.: Single-shot multi-person 3d pose estimation from monocular rgb. In: 3DV (2018)
17. Patel, P., Huang, C.H.P., Tesch, J., Hoffmann, D.T., Tripathi, S., Black, M.J.: Agora: Avatars in geography optimized for regression analysis. In: CVPR (2021)
18. Sárándi, I., Linder, T., Arras, K.O., Leibe, B.: Metrabs: Metric-scale truncation-robust heatmaps for absolute 3d human pose estimation. IEEE Transactions on Biometrics, Behavior, and Identity Science (2020)
19. Sun, Y., Bao, Q., Liu, W., Fu, Y., Black, M.J., Mei, T.: Monocular, one-stage, regression of multiple 3d people. In: ICCV (2021)
20. Ugrinovic, N., Ruiz, A., Agudo, A., Sanfeliu, A., Moreno-Noguer, F.: Body size and depth disambiguation in multi-person reconstruction from single images. In: 3DV (2021)

21. Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning from synthetic humans. In: CVPR (2017)
22. Zhang, J., Yu, D., Liew, J.H., Nie, X., Feng, J.: Body meshes as points. In: CVPR (2021)