Supplementary Material of "I2L-MeshNet: Image-to-Lixel Prediction Network for Accurate 3D Human Pose and Mesh Estimation from a Single RGB Image"

Gyeongsik Moon and Kyoung Mu Lee

ECE & ASRI, Seoul National University, Korea
{mks0601,kyoungmu}@snu.ac.kr

In this supplementary material, we present more experimental results that could not be included in the main manuscript due to the lack of space.

1 Various settings of I2L-MeshNet

1.1 When to marginalize 2D to 1D?

We report how the MPJPE, PA MPJPE, and GPU memory usage change when the marginalization takes place on the ResNet output (*i.e.*, $\mathbf{F}_{\mathrm{P}}$ or $\mathbf{F}_{\mathrm{M}}$), which is the input of the first upsampling module, instead of the output of the last upsampling module (*i.e.*, $f_{\mathrm{up}}^{\mathrm{P}}(\mathbf{F}_{\mathrm{P}})$ or $f_{\mathrm{up}}^{\mathrm{M}}(\mathbf{F}_{\mathrm{M}})$) in Table 1. For the convenience, we removed PoseNet from our I2L-MeshNet and changed MeshNet to take the input image. The table shows that the early marginalization increases the errors while requiring less amount of GPU memory. This is because the marginalized two 1D feature maps can be generated from multiple 2D feature map, which results in spatial ambiguity. To reduce the effect of this spatial ambiguity, we designed our I2L-MeshNet to extract a sufficient amount of 2D information and then apply the marginalization at the last part of the network instead of applying it in the early stage.

When the marginalization is applied on the ResNet output $\mathbf{F}_{\mathrm{M}}$, all 2D layers (*i.e.*, deconvolutional layers and batch normalization layers) in the upsampling modules are converted to the 1D layers. All models are trained on Human3.6M dataset. The z-axis heatmap prediction part is not changed.

settings	MPJPE	PA MPJPE	GPU mem.
avg on $\mathbf{F}_{\mathrm{M}}$	93.5	64.1	**4.4 GB**
avg on $f_{\mathrm{up}}^{\mathbf{M}}(\mathbf{F}_{\mathbf{M}})$ (ours)	**86.2**	**59.8**	4.6 GB

Table 1. The MPJPE, PA MPJPE, and GPU memory usage comparison between various marginalization settings on Human3.6M dataset.

1.2 How to marginalize 2D to 1D?

We report how the MPJPE and PA MPJPE change when different marginalization methods are used in Table 2. For the convenience, we removed PoseNet from our I2L-MeshNet and changed MeshNet to take the input image. The table shows that our average pooling achieves the lowest errors. Compared with the max pooling that provides the gradients to one pixel position per one x or y position, our average pooling provides the gradients to all pixel positions, which is much richer ones. We implemented the weighted sum by constructing a convolutional layer whose kernel size is $(8h, 1)$ and $(1, 8w)$ for x- and y-axis lixel-based 1D heatmap prediction, respectively, without padding. The weighted sum provides lower error than that of the max pooling, however still worse than our average pooling. We believe the large size of a kernel of the convolutional layer (*i.e.*, $(8h, 1)$ and $(1, 8w)$) is hard to be optimized, which results in higher error than ours. For all settings, models are trained on Human3.6M dataset, and the z-axis heatmap prediction part is not changed.

settings	MPJPE	PA MPJPE
max pooling	93.5	64.1
weighted sum	89.4	61.4
avg pooling (ours)	**86.2**	**59.8**

Table 2. The MPJPE and PA MPJPE comparison between various marginalization settings on Human3.6M dataset.

2 Comparison with previous 2.5D heatmap regression

We compare the MPJPE and GPU memory usage between a model that predicts our lixel-based 1D heatmap and a model that predicts the 2.5D heatmap [2] in Table 3. The 2.5D heatmap [2] consists of xy heatmap and z heatmap, where xy one is the pixel-based 2D heatmap and z one has the same spatial size with that of xy heatmap and contains root joint-relative depth on the activated xy position for all mesh vertices. They predict the depth values on z heatmap, not the likelihood, thus cannot model uncertainty of the z-axis prediction. As the table shows, our lixel-based one achieves significantly lower error under the same resolution while requiring a much smaller amount of GPU memory. We think that this is because the 2.5D heatmap of Iqbal et al. [2] cannot model uncertainty of the prediction in z-axis, while ours can. For all settings, models are trained on Human3.6M dataset, and we removed PoseNet and changed MeshNet to take an input image and predict the heatmap.

3 Effect of each loss function

We show the effectiveness of the MeshNet pose loss $L_{\text{pose}}^{\text{MeshNet}}$ in Table 4. Although we supervise mesh vertices by the mesh vertex loss $L_{\text{pose}}^{\text{MeshNet}}$, additional $L_{\text{pose}}^{\text{MeshNet}}$

settings	resolution	uncertainty in z-axis	MPJPE	GPU mem.
2.5D heatmap [2]	8×8, 8×8	✗	107.4	3.6GB
2.5D heatmap [2]	32×32, 32×32	✗	100.4	8.4GB
lixel-based 1D heatmap	8, 8, 8	✓	100.2	3.4GB
lixel-based 1D heatmap	32, 32, 32	✓	94.8	4.0GB
lixel-based 1D heatmap	64, 64, 64	✓	86.2	4.6GB

Table 3. The MPJPE and GPU memory usage comparison between various marginalization settings on Human3.6M dataset.

is helpful for human joint-aligned mesh prediction. Both models are trained on Human3.6M dataset.

For visually pleasant mesh estimation, we use normal vector loss L_{normal} and edge length loss L_{edge}. We show the effectiveness of the two loss functions in Figure 1. As the figure shows, the two loss functions improves visual quality of output meshes. We checked that L_{normal} and L_{edge} marginally affect the MPJPE and PA MPJPE. For all settings, all models are trained on Human3.6M dataset and MSCOCO dataset.

settings	MPJPE	PA MPJPE
wo. $L_{\text{pose}}^{\text{MeshNet}}$	84.5	58.5
w. $L_{\text{pose}}^{\text{MeshNet}}$	81.8	58.0

Table 4. The MPJPE and PA MPJPE comparison between models trained with and without $L_{\text{pose}}^{\text{MeshNet}}$ on Human3.6M dataset.

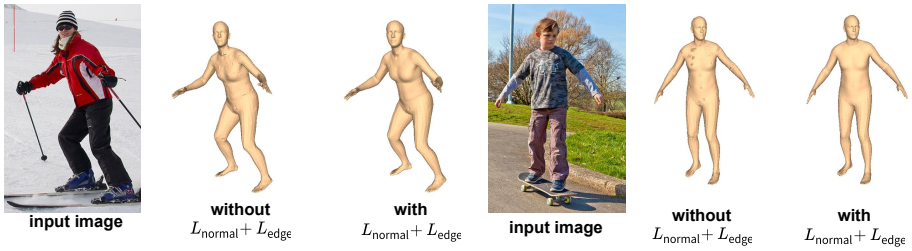

Fig. 1. Estimated meshes from models trained with different combinations of loss functions.

4 Accuracy of PoseNet

We provide the MPJPE and PA MPJPE of PoseNet from I2L-MeshNet in Table 5. The PoseNet is trained with MeshNet by minimizing the loss function L. As our PoseNet predicts 3D joint coordinates of the SMPL body joint set or MANO hand joint set, we calculate the errors using groundtruth SMPL or MANO 3D joint coordinates. We could not calculate the MPJPE on FreiHAND dataset because the official evaluation server does not support it.

datasets	MPJPE	PA MPJPE
Human3.6M	62.2	47.2
3DPW	112.2	72.3
SURREAL	40.0	29.5
FreiHAND	n/a	8.0

Table 5. The MPJPE and PA MPJPE of PoseNet on each dataset.

5 Pseudo-groundtruth SMPL parameters of Human3.6M dataset

All the previous works [4–6, 10] used SMPL parameters obtained by applying Mosh [7] on the marker data of Human3.6M dataset as the groundtruth parameters. However, currently, the distribution of the SMPL parameters from Mosh is disallowed because of the license problem. In addition, the source code of Mosh is not publicly released. Alternatively, we obtain groundtruth SMPL parameters by applying SMPLify-X [9] on the groundtruth 3D joint coordinates of Human3.6M dataset. Although the obtained SMPL parameters are not perfectly aligned to the groundtruth 3D joint coordinates, we checked that the error of the SMPLify-X is much less than those of current state-of-the-art 3D human pose estimation methods, as shown in Table 6. Therefore, we think using SMPL parameters from SMPLify-X as groundtruth is reasonable. Note that for a fair comparison, all the experimental results of previous works are reported by training and testing them on our SMPL parameters from SMPLify-X. When fitting, we used neutral gender SMPL body model. However, we found that it produces gender-specific body shapes, although we did not specify gender for each subject. As most of the subjects of the training set in Human3.6M dataset are female, we found that our I2L-MeshNet trained on Human3.6M dataset tends to produce female body shape meshes. We tried to fix the identity code of the SMPL body model obtained from the T-pose; however it produces higher errors. Thus, we did not fix the identity code for each subject.

methods	MPJPE
Moon et al. [8]	53.3
Sun et al. [11]	49.6
Iskakov et al. [3]*	20.8
SMPLify-X from GT 3D pose	**13.1**

Table 6. The MPJPE comparison between SMPLify-X fitting results and state-of-the-art 3D human pose estimation methods. "*" takes multi-view RGB images as inputs.

methods	MPVPE	MPJPE
SMPLify [1]	75.3	-
BodyNet [12]	65.8	40.8
I2L-MeshNet (Ours)	**44.7**	**37.7**

Table 7. The MPVPE and MPJPE comparison between state-of-the-art methods and the proposed I2L-MeshNet on SURREAL.

6 Evaluation on SURREAL

We additionally provide evaluation results on SURREAL [13] that contains 67K clips synthesized by animating SMPL body model. We followed the same training and test set split of BodyNet [12]. For evaluation, mean per-vertex position error (MPVPE), which is averaged per-vertex Euclidean distance error (mm) between predicted and groundtruth 3D mesh coordinates, and MPJPE are used after root joint alignment. We compare MPVPE and MPJPE of our I2L-MeshNet with previous state-of-the-art 3D human body pose and mesh estimation methods [1, 4, 12] on the SURREAL test set. To this end, we reduced the clips in the training set to 1 fps to make the training image set. Table 7 shows that the proposed I2L-MeshNet significantly outperforms all previous state-of-the-art methods. Especially, it achieves much lower test error compared with BodyNet [12], model-free approach.

7 Qualitative results

We provide qualitative results comparison between ours and previous state-of-the-art model-free method (*i.e.*, GraphCMR [6]) in Figure 2. As the figure shows, our I2L-MeshNet provides much more visually pleasant mesh results than GraphCMR. We think this is because the graph convolutional network (GraphCNN) often tends to smooth the meshes by averaging the vertex feature with that of neighboring vertices.

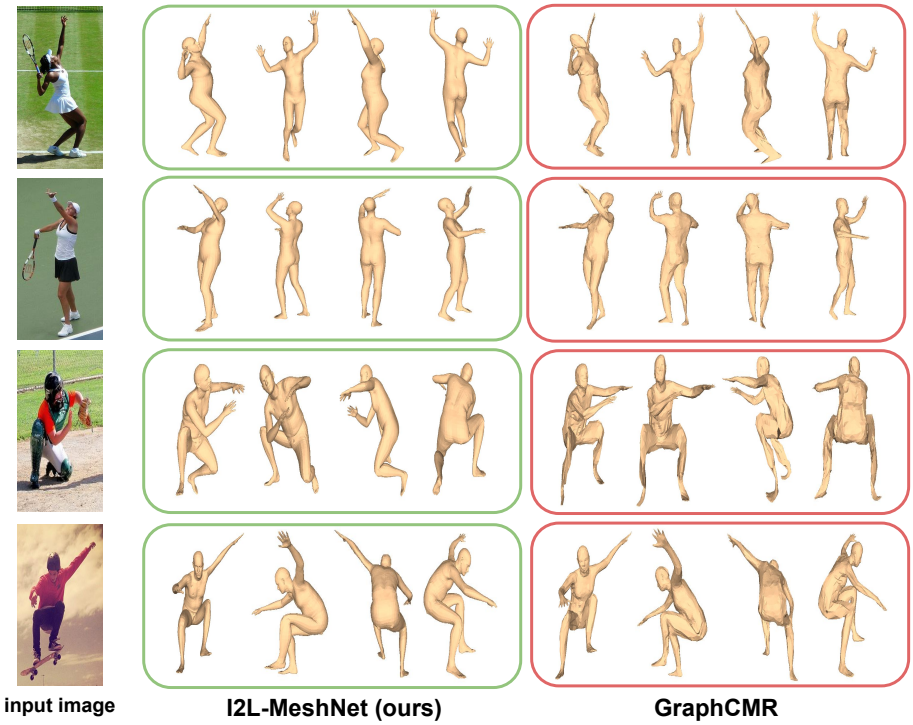

Fig. 2. Estimated meshes comparisons between our I2L-MeshNet and GraphCMR [6].

References

1. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep it smpl: Automatic estimation of 3D human pose and shape from a single image. In: ECCV (2016)
2. Iqbal, U., Molchanov, P., Breuel Juergen Gall, T., Kautz, J.: Hand pose estimation via latent 2.5 d heatmap regression. In: ECCV (2018)
3. Iskakov, K., Burkov, E., Lempitsky, V., Malkov, Y.: Learnable triangulation of human pose. In: ICCV (2019)
4. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: CVPR (2018)
5. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In: ICCV (2019)
6. Kolotouros, N., Pavlakos, G., Daniilidis, K.: Convolutional mesh regression for single-image human shape reconstruction. In: CVPR (2019)
7. Loper, M., Mahmood, N., Black, M.J.: Mosh: Motion and shape capture from sparse markers. ACM TOG (2014)
8. Moon, G., Chang, J.Y., Lee, K.M.: Camera distance-aware top-down approach for 3D multi-person pose estimation from a single RGB image. In: ICCV (2019)
9. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3D hands, face, and body from a single image. In: CVPR (2019)
10. Pavlakos, G., Zhu, L., Zhou, X., Daniilidis, K.: Learning to estimate 3D human pose and shape from a single color image. In: CVPR (2018)
11. Sun, X., Xiao, B., Wei, F., Liang, S., Wei, Y.: Integral human pose regression. In: ECCV (2018)
12. Varol, G., Ceylan, D., Russell, B., Yang, J., Yumer, E., Laptev, I., Schmid, C.: Bodynet: Volumetric inference of 3D human body shapes. In: ECCV (2018)
13. Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning from synthetic humans. In: CVPR (2017)