Supplementary Material for "Three Recipes for Better 3D Pseudo-GTs of 3D Human Mesh Estimation in the Wild"

Gyeongsik Moon[1] Hongsuk Choi[2] Sanghyuk Chun[3] Jiyoung Lee[3] Sangdoo Yun[3]

[1] Meta Reality Labs [2] Samsung AI Center - New York [3] NAVER AI Lab

mks0601@gmail.com redstonepo@gmail.com {sanghyuk.c, lee.j, sangdoo.yun}@navercorp.com

In this supplementary material, we provide more experiments, discussions, and other details that could not be included in the main text due to the lack of pages. The contents are summarized below:

A. Effectiveness of a combination of VPoser [6] and L2 regularizer
B. Qualitative comparisons
C. Implementation details
D. Limitations

A. Effectiveness of a combination of VPoser and L2 regularizer

Table A shows the effectiveness of 1) usage of VPoser [6] and 2) weight of L2 regularizer during the training of the annotation network f. The combination of VPoser and L2 regularizer is the third recipe, introduced in Section 2.2 of the main manuscript. Regardless of the usage of VPoser, setting the non-zero weight of the L2 regularizer produces lower 3D errors of g. This indicates that despite its simplicity, the L2 regularizer helps to prevent anatomically implausible 3D meshes and produce beneficial 3D pseudo-GTs. In addition, using VPoser achieves lower 3D errors of g compared to not using it. This is also because VPoser can effectively limit the 3D mesh to anatomically plausible space. Training sets of all annotation networks f in the table are H36M+MI+MSCOCO+3DPW. The ResNet backbone [1] of all annotation networks in the table are initialized with ResNet, pre-trianed on ImageNet [7] classification dataset.

Fig. A shows the effectiveness of 1) using VPoser and 2) applying L2 regularizer. Without VPoser and L2 regularizer, the 3D pseudo-GT has an anatomically implausible 3D mesh although its 2D pose is fit to the image. Using VPoser makes the 3D pseudo-GT anatomically plausible; however, it still produces the wrong 3D mesh. The right leg is too much bent to the left side. Finally, using both VPoser and L2 regularizer makes the 3D pseudo-GT anatomically plausible and correct. In particular, additionally using the L2 regularizer enforces the 3D mesh in the latent space of VPoser.

B. Qualitative comparisons

Fig. B shows qualitative comparisons between 3D pseudo-GTs from our annotation network f and NeuralAnnot [4]. The comparisons show that our 3D pseudo-GTs are more accurate than those of NeuralAnnot. The results from the first row to the fourth row show that ours are more robust to the depth ambiguity. For example, in the fourth row example, both have almost the same 2D position of the right knee. However, our 3D position of the right knee is more accurate as it does not penetrate inside of the left leg.

Fig. C shows (a) rendered 3D pseudo-GTs on various images of MSCOCO and (b) 3D pseudo-GTs on truncated images of MSCOCO. The rendered results show that our 3D pseudo-GTs are well-aligned with the image. In addition, ours produce robust 3D pseudo-GTs on severely truncated images by utilizing strong contextual information of image features.

C. Implementation details of annotation network f

As described in Section 2.1 of the main manuscript, our annotation network f is based on Pose2Pose network [3]. Hence, most of the details follow theirs. PyTorch [5] is used for implementation. For the training, we use Adam optimizer [2] with a mini-batch size of 192. Data augmentations, including scaling, rotation, random horizontal flip, and color jittering, are performed during the training. The initial learning rate is set to 10^{-4} and reduced by a factor of 10 at the 11th and 13th epoch. We train our annotation network f for 15 epochs. A single NVIDIA A100 GPU is used for the experiments, where it takes 6 hours to train our annotation network f. We modified the Pose2Pose network

Annotation network f			Estimation network g	
ID	Use VPoser	L2 reg. weight	Training sets	3D errors
f1-1	✗	0.0	H36M+MI+[MSCOCO]$_{f1\text{-}1}$	65.98
f1-2	✗	10^{-1}	H36M+MI+[MSCOCO]$_{f1\text{-}2}$	79.07
f1-3	✗	10^{-2}	H36M+MI+[MSCOCO]$_{f1\text{-}3}$	64.25
f1-4	✗	10^{-3}	H36M+MI+[MSCOCO]$_{f1\text{-}4}$	64.04
f1-5	✗	10^{-4}	H36M+MI+[MSCOCO]$_{f1\text{-}5}$	64.67
f1-6	✗	10^{-5}	H36M+MI+[MSCOCO]$_{f1\text{-}6}$	64.39
f2-1	✓	0.0	H36M+MI+[MSCOCO]$_{f2\text{-}1}$	56.57
f2-2	✓	10^{-1}	H36M+MI+[MSCOCO]$_{f2\text{-}2}$	59.22
f2-3	✓	10^{-2}	H36M+MI+[MSCOCO]$_{f2\text{-}3}$	**51.61**
f2-4	✓	10^{-3}	H36M+MI+[MSCOCO]$_{f2\text{-}4}$	53.27
f2-5	✓	10^{-4}	H36M+MI+[MSCOCO]$_{f2\text{-}5}$	55.06
f2-6	✓	10^{-5}	H36M+MI+[MSCOCO]$_{f2\text{-}6}$	56.03

Table A. Comparison of 3D errors of estimation networks g, trained with different 3D pseudo-GTs of MSCOCO. The subscript at the square brackets denotes a method to obtain the 3D pseudo-GTs. The 3D errors of g (PA MPJPE) are calculated on 3DPW.

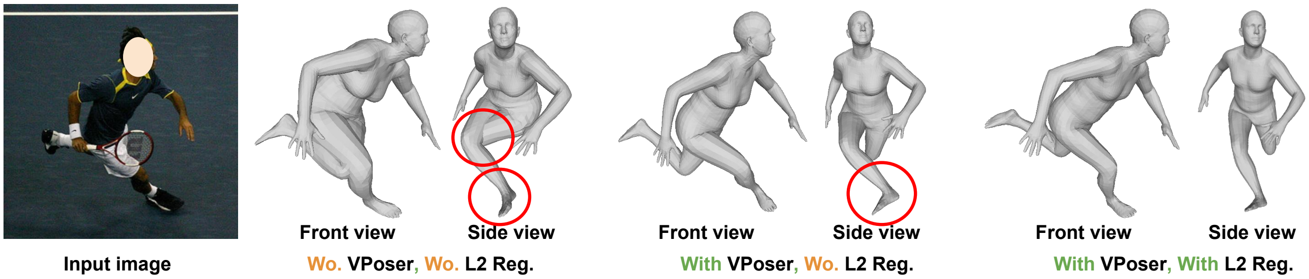


Figure A. Visual comparison between 3D pseudo-GTs, obtained from annotation networks whose IDs are f1-1, f2-1, and f2-3 of Table A. Wrong parts are highlighted.

to predict the latent code of VPoser instead of SMPL pose parameters. The predicted VPoser latent code is passed to the decoder of VPoser, which outputs the SMPL pose parameter. The L2 regularizer is applied to the predicted latent code of VPoser and SMPL shape parameter, where its weight is determined to 10^{-2} following Table A. A neutral gender SMPL model is used for the experiments. All other details are available in the codes of Pose2Pose [3][1].

D. Limitations

Although our annotation network f produces much more beneficial 3D pseudo-GTs than previous attempts, our 3D pseudo-GTs still contain some errors in nature. This could be addressed by collecting more ITW 3D datasets, such as 3DPW, as we made our 3D pseudo-GTs more beneficial by utilizing 3DPW to train annotation network f. Collecting ITW 3D datasets is challenging; however, we believe it is worthwhile considering its usefulness. In particular, Table 8 of the main manuscript shows that 3DPW is much more helpful than existing large-scale ITW 2D datasets, such as InstaVariety, despite the small scale of 3DPW. We observed from an additional study that using 50% and 10% of 3DPW when training the annotation network f decreases 3D error of the estimation network g only 4% and 7%, respectively. As such analysis shows that even a small amount of ITW 3D datasets are helpful, it relieves a concern on collection costs of ITW 3D datasets.

[1] https://github.com/mks0601/Hand4Whole_RELEASE/tree/Pose2Pose

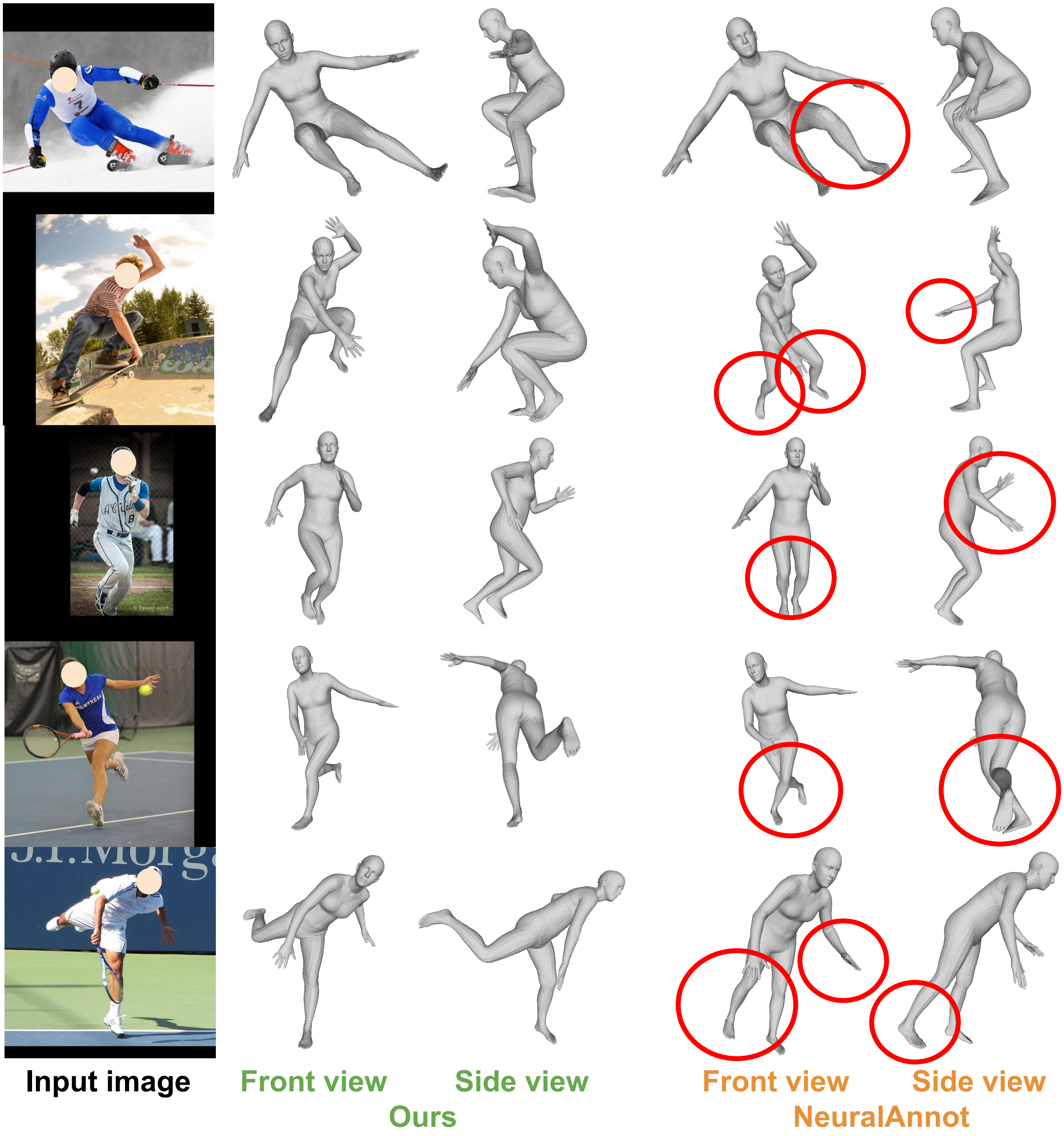
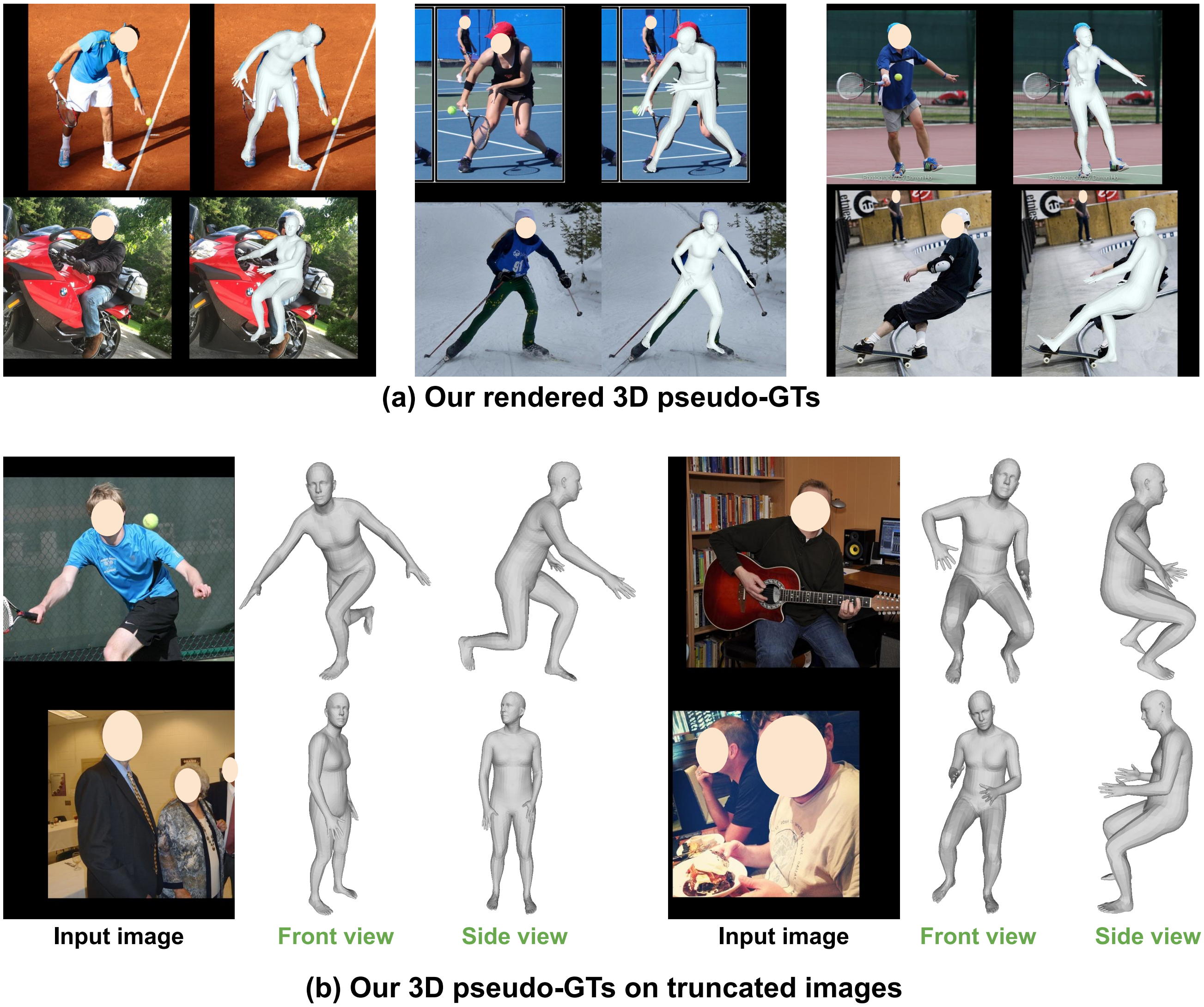


Figure B. Visual comparison between 3D pseudo-GTs of MSCOCO from ours and NeuralAnnot [4]. Wrong parts are highlighted.


(a) Our rendered 3D pseudo-GTs

(b) Our 3D pseudo-GTs on truncated images

Figure C. (a) Our rendered 3D pseudo-GTs on images of MSCOCO. (b) Our 3D pseudo-GTs on truncated images of MSCOCO.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 1

[3] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Accurate 3D hand pose estimation for whole-body 3D human mesh estimation. In *CVPRW*, 2022. 1, 2

[4] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. NeuralAnnot: Neural annotator for 3D human mesh training sets. In *CVPRW*, 2022. 1, 3

[5] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 1

[6] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3D hands, face, and body from a single image. In *CVPR*, 2019. 1

[7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *IJCV*, 2015. 1