Geometry-aware Two-scale PIFu Representation for Human Reconstruction

— *Supplementary Material*

Zheng Dong[1] **Ke Xu**[2] **Ziheng Duan**[1] **Hujun Bao**[1] **Weiwei Xu**[*1] **Rynson W.H. Lau**[2]

[1]State Key Lab of CAD&CG, Zhejiang University [2] City University of Hong Kong

A Implementation Details

Training $\&$ inference details. To train our network, we adopt PyTorch [11] with ADAM optimizer [7] and learning rate $1e^{-4}$ on 2 NVIDIA RTX3090 GPUs. The β_1, β_2 in ADAM are set to 0.5, 0.99, respectively. For geometry-aware PIFu-Body ($\mathcal{F}_b$), we first train $\mathcal{F}_b$ on three-view RGBD images (with resolutions of 512x512), end-to-end for 20 epochs with a batch size of 4, where the learning rate is reduced by a factor of 2 every 5 epochs and the regularization loss L_{reg} is not used in this phase. Then, we fix the backbone (the largest box in Fig. 2) in $\mathcal{F}_b$ and continue to train the PIFu-body (in Fig. 2) part with the additional loss L_{reg} for 5 epochs. The training strategy of fixing the backbone is mainly to prevent L_{reg} from affecting the depth denoising, thereby making $\mathbf{D}_{rf}$ smooth. For the RGBD images involved in training, we randomly select three views with intervals of approximately 135 degree, 90 degree, 135 degree, according to the captured setting (Sec. C). Besides, we sample 8000 3D points for training $\mathcal{F}_b$ according to the sampling strategy of PIFuHD [13].

For our high-resolution PIFu-Face ($\mathcal{F}_f$), in order to model vivid expressions, we first train the single-view $\mathcal{F}_f$ independently for approximately 100 epochs with a learning rate of $1e^{-3}$ on the processed *FaceScape* [19] dataset. Refer to Fig. 6 (a) in the main paper for the facial mesh processing procedure. In this phase, we sample 5000 points around the ground-truth facial model and render the front-view facial RGBD images to pre-train $\mathcal{F}_f$. The loss function used here is the same as PIFu [12].

Finally, we fine-tune the whole network ($\mathcal{F}_b$ $\&$ $\mathcal{F}_f$) with a learning rate of $1e^{-5}$ for 5 epochs on *THuman2.0* [20] training set, to optimize $\mathcal{F}_f$ to fit the whole body, where the backbone in $\mathcal{F}_b$ is fixed. To jointly train $\mathcal{F}_b$ and $\mathcal{F}_f$, we sample 5000 body and facial points equally, as shown in Fig. 7(a). Here, the facial points used for training are selected as stated in Sec. 3.2 in our main paper, but we replace $\mathbf{D}_{rf}^f$ with $\mathbf{D}_{gt}^f$ (*i.e.*, ground-truth facial depth map) to ensure the sampling correctness.

For the hyper-parameters during training, we set μ_0 and μ_1 in L_σ as 1.0, 1.0 respectively, ϵ in L_{reg} as 4mm, $\rho_D, \lambda_s, \rho_N$ in L_D as 10.0, $\{1, 3/4, 2/4, 1/4\}$, 1.0 respectively. For λ_{reg}, λ_D in L, we set them as 100.0, 10.0 respectively. Besides, we set the dilate rate x as 2, δ_p as 0.01m and α as 0.15m.

When evaluating the models on *THuman2.0* [20] test set, we first generate the three-view RGBD images with 5 different degrees of depth noise (0.5cm, 1.0cm, 1.5cm, 2.0cm, 2.5cm of Gaussian standard deviation) for each mesh, and then measure the Point-to-Surface, Chamfer and Normal errors between the reconstructed and the ground-truth surfaces.

When testing the models on our captured data, we adopt three-view RGBD images with intervals of 135 degree, 90 degree, 135 degree, as the capture setting. The user is required to face the middle camera (*i.e.*, front view f), and the real depth data is captured using Microsoft Azure Kinect-V4 sensors. The captured RGB resolution is 2560x1440, and depth resolution is 1440x1440. We crop and down-sample the captured RGBD images to a resolution of 512x512 as the inputs for our network. Here we use *background-matting-v2* [8] to obtain body binary mask $\mathbf{M}$. For high-resolution facial RGB image, we first use *RetinaFace* [3] to detect the front-view face region, and a face skin segmentation network [4] to obtain the facial binary mask $\mathbf{M}_f$, then crop the facial RGB image

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

from the original full-body image as $\mathbf{I}_f$. For the running time, it takes about 0.831s for our network to perform depth denoising and 12.287s to predict the occupancy field of resolution 256^3. After obtaining the occupancy fields O_b and O_f, we perform face-to-body fusion ($\mathcal{W}$) to get the fused field and then apply the Marching Cube algorithm [9] to recover the final human mesh.

Network details. In our geometry-aware PIFu-body $\mathcal{F}_b$, $\mathcal{M}_b$ is the implicit function that predicts the occupancy values, implemented as a multi-layer perceptron (MLP) with skip connections and the hidden neurons as (1024, 512, 256), (128, 128, 128, 128). The first group of neurons reduces the feature dimensions, and the second group is used to query occupancy values. The function $\mathcal{A}$ indicates the multi-view feature aggregation module, implemented by a multi-head Transformer [15] encoder with 8 heads and 6 layers. Besides, for the RGB and depth encoders, we use two independent HRNets [16] (*HRNetV2-W18-Small-v2*) as feature extraction backbone. In our PIFu-Face $\mathcal{F}_f$, since the depth input to $\mathcal{F}_f$ is the denoised facial depth, we adopt a single encoder to encode RGBD features. Specifically, the function $\mathcal{H}_f$ is the *HourGlass* [10] network used in PIFu [5] and the function $\mathcal{M}_f$ is the MLP with skip connections and the hidden neurons as (512, 256, 128, 32).

Details of competing methods. When comparing with PIFuHD [13], StereoPIFu [6], we directly use their pre-trained models since there are no open-source training codes. For IPNet [2], we fine-tune their pre-trained model on our training set, and when testing on our captured data, we use the fused point clouds (the former, latter and current frames) as inputs. For DoubleFusion [21] and Function4D [20], we re-implement the two methods following the details of the papers, but not in real time. For Function4D [20], we track the former and latter frames for the current frame and fuse the multi-frame point clouds to produce three new depth maps. The regenerated depth map is smoother and contains more details. We then feed these depth maps into the multi-view PIFu model with the truncated PSDF features. For multi-view RGBD-PIFu [12], we retrain the original PIFu model on three-view RGBD images with the same settings as our model.

Obtaining the points set $\mathcal{S}_j$. To obtain the set $\mathcal{S}_j$ where 3D points are projected on the depth-jumping regions, we first calculate the depth-jumping maps, denoted as $\mathbf{M}_d^i (i = 1, ..., \mathcal{N})$, from the ground-truth depth maps $\mathbf{D}_{gt}^i (i = 1, ..., \mathcal{N})$. In $\mathbf{M}_d^i$, the value of the 3x3 regions with a depth range greater than th (default as 6cm) is set to 1, and 0 otherwise. Then, for each queried 3D body point $\mathbf{X}_b$, if the 2D projection $\mathbf{x}_b^i$ is located in the depth-jumping regions, we put $\mathbf{X}_b$ in the $\mathcal{S}_j$. We can set a flag $s_j(\cdot)$ to mark these points as:

$$s_j(\mathbf{X}_b) = \begin{cases} 1 & \sum_i^{\mathcal{N}} \mathcal{B}(\mathbf{M}_d^i, \mathbf{x}_b^i) > k \\ 0 & else \end{cases}, \tag{1}$$

where $\mathcal{B}(\cdot, \mathbf{x}_b)$ is the nearest sampling function. We set the threshold k to 0, which means that as long as a point $\mathbf{X}_b$ is projected to the depth-jumping regions under one view, it will be marked in $\mathcal{S}_j$. This allows more points in $\mathcal{S}_b$ to be optimized by L_{reg}. As shown in Fig. A, the points marked in $\mathcal{S}_j$ are more clustered in the depth-jumping regions such as the body edges.

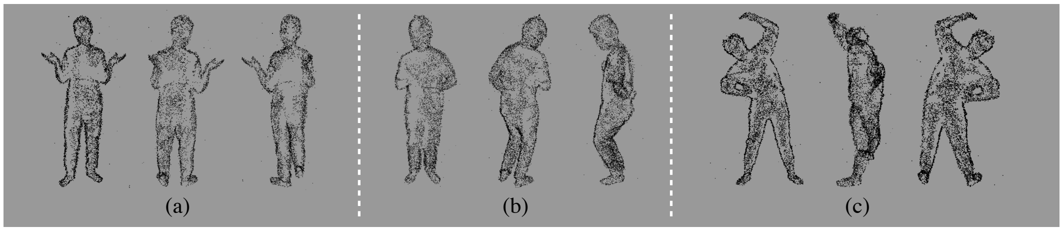

(a) (b) (c)

Figure A: Visualization of the sampled points set $\mathcal{S}_j$. We show the results from three perspective views. The points are mainly clustered in the depth-jumping regions (*e.g.*, body edges).

B More Results.

Qualitative comparisons on the test dataset. Fig. B shows the reconstructed results of four existing approaches and our method on our test dataset. It can be seen that our reconstructed results are closer to the ground-truth models (especially the details). As stated in Sec. 4.1 in our main paper, Multi-view RGBD-PIFu tends to lose some high-frequency details (*e.g.*, face surface), and suffer from floating geometry (Fig. B(a)). The topological error in the back view is obvious in PIFuHD and

StereoPIFu. Besides, the details in the front view tend to be incorrect (Fig. B(b,c)). For IPNet, even if we fine-tune the model, the topological errors and geometry missing are still evident (Fig. B(d)).

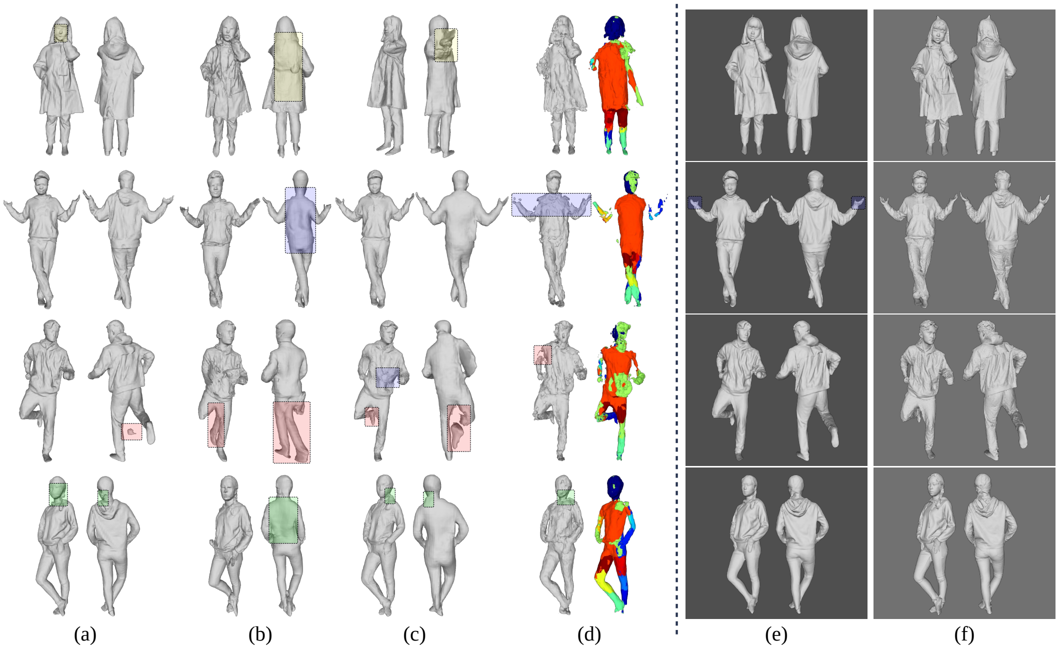


Figure B: Qualitative comparisons on our test dataset, between our proposed method and four state-of-the-art approaches. Multi-view RGBD-PIFu [12] (a). PIFuHD [13] (b). StereoPIFu [6] (c). IPNet [2] (the outer and inner reconstructed results) (d). Ours (e). Ground-truth models (f). We show the front and back results. Zoom in to see the details.

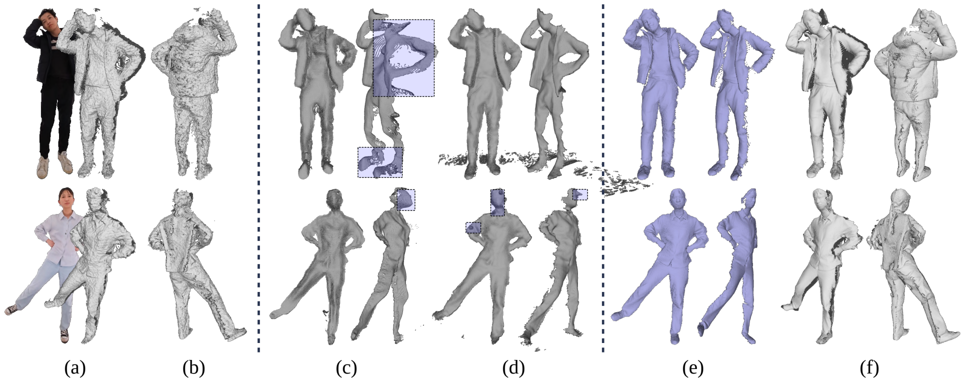


Figure C: Visualization of depth denoising on our captured data. Raw RGB images and depth maps (fused point clouds) (a,b). Results of DDRNet [18] and Sterzentsenko *et al.* [14] (c,d). Our refined depth maps and the fused point clouds (from three refined depths) (e,f). We show the refined depth maps in two different views. Zoom in to see details.

Comparisons with the depth denoising methods. Fig. C and Fig. 3 in our main paper show the depth denoising results of DDRNet [18], Sterzentsenko *et al.* [14], and ours on our captured data. We can see that both the previous two methods over-smooth the original depths (Fig. C(c,d)). Besides, DDRNet [18] suffer from topological errors from other perspectives (Fig. C(c)). In contrast, our results recover the high-frequency details and the topology information is correct (Fig. C(e,f)).

Comparisons of the view number. We conduct an experiment to evaluate the view number (Fig. D). We first pre-train the three-view model according to the details in Sec. A, then add input images and continue to train the pre-trained model. We can see that as the number of views increases, the geometric details gradually become better, especially in the invisible or small regions (boxes). But it can be seen that when the number of views is greater than 3, the overall topology information and details hardly change, probably because the three-view depth maps have covered most of the body (Fig. C(b,f)), which is why we choose to adopt the three-view capture setting (Sec. C).

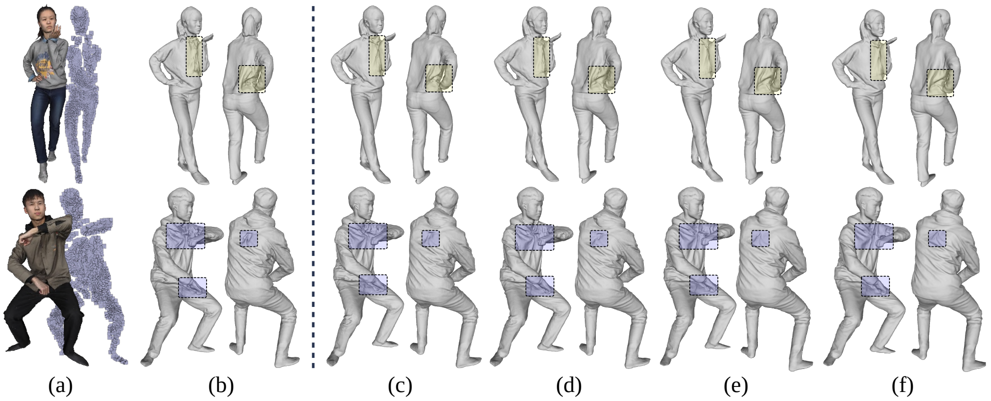

(a) (b) (c) (d) (e) (f)

Figure D: Comparisons on the number of views. RGBD images of the front view (a). 3 views (our setting) (b). 6 views (c). 8 views (d). 10 views (e). 12 views (f). Zoom in to see details (especially the details inside the boxes).

Comparisons with face reconstruction methods. We compared our PIFu-Face to three state-of-the-art face reconstruction methods. Visual comparisons are shown in Fig. E, which shows that our method produces competitive face reconstruction results. Compared to these methods that explicitly reconstruct the face model, our PIFu-Face implicitly predicts the face occupancy fields, which facilitates our face-body occupancy fields fusion scheme for full-body reconstruction.

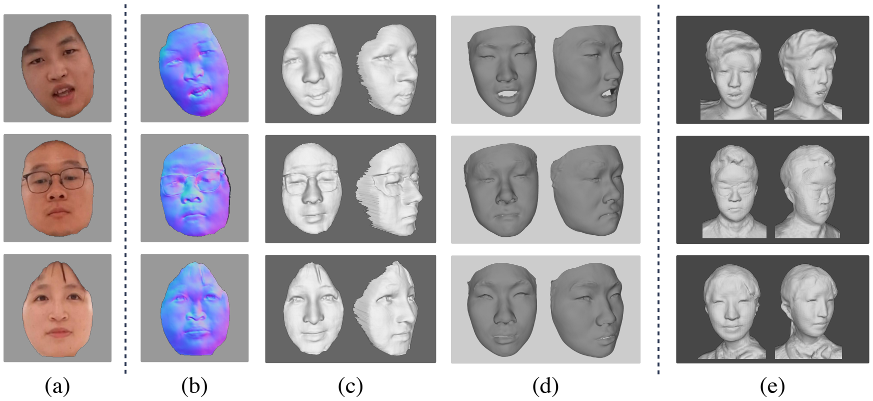

(a) (b) (c) (d) (e)

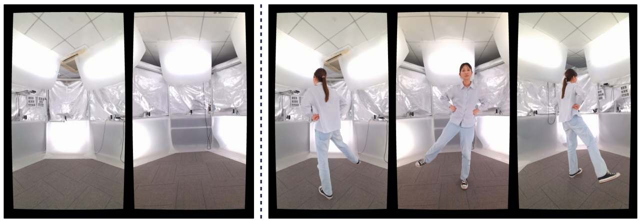
Figure E: Qualitative comparisons between three state-of-the-art face reconstruction methods and our PIFu-Face model. Input facial RGB image (a). Predicted facial normal maps of Abrevava *et al.* [1] (b). 3D reconstructed facial model of DF2Net [22] (c), FaceVerse [17] (d). Our reconstructed facial results (e). Zoom in to see the details.

More textured results. In Fig. G, we provide more reconstructed human geometric models along with the corresponding textured results. The textured results are overall high-quality and well aligned with the geometric models.

More geometric results. We show more of our reconstructed geometric results in Fig. H. It can be seen that our method can produce vivid facial/hair details and accurate bodies under different poses.

C Obtaining Real RGBD Data

During capturing, we placed color and depth (Kinect-V4) cameras in a circle at 45-degree intervals to ease the calibration, as illustrated in Fig. F(a). For each camera, we used the infrared calibration board to calibrate the intrinsic parameters and initial extrinsic Kinect parameters. Then we used the iterative closest point (ICP) method to match the points clouds captured by multiple depth sensors to further optimize the extrinsic parameters. Afterward, we recorded the captured RGBD images and performed the de-distortion operation on the captured RGB images. During testing, we obtain the three-view RGBD images with intervals of (135 degree, 90 degree, 135 degree) as shown in Fig. F(b), and the user is required to face the cameras in the middle.



(a) (b)

Figure F: Our capturing system. Two background images (a). Three selected RGB images (135 degree, 90 degree, 135 degree) (b).

Personal information of captured data. We captured RGBD images of different people. Although the captured data contains personal information, it is licensed only for academic research purposes.

D Border Impacts

While we do not foresee our method causing any direct negative societal impact, it may be leveraged to create malicious applications using 3D human reconstruction. The human information captured using Kinect sensors may have the risk of a leak that raises privacy concerns. We urge the readers to limit the usage of this work to legal use cases.

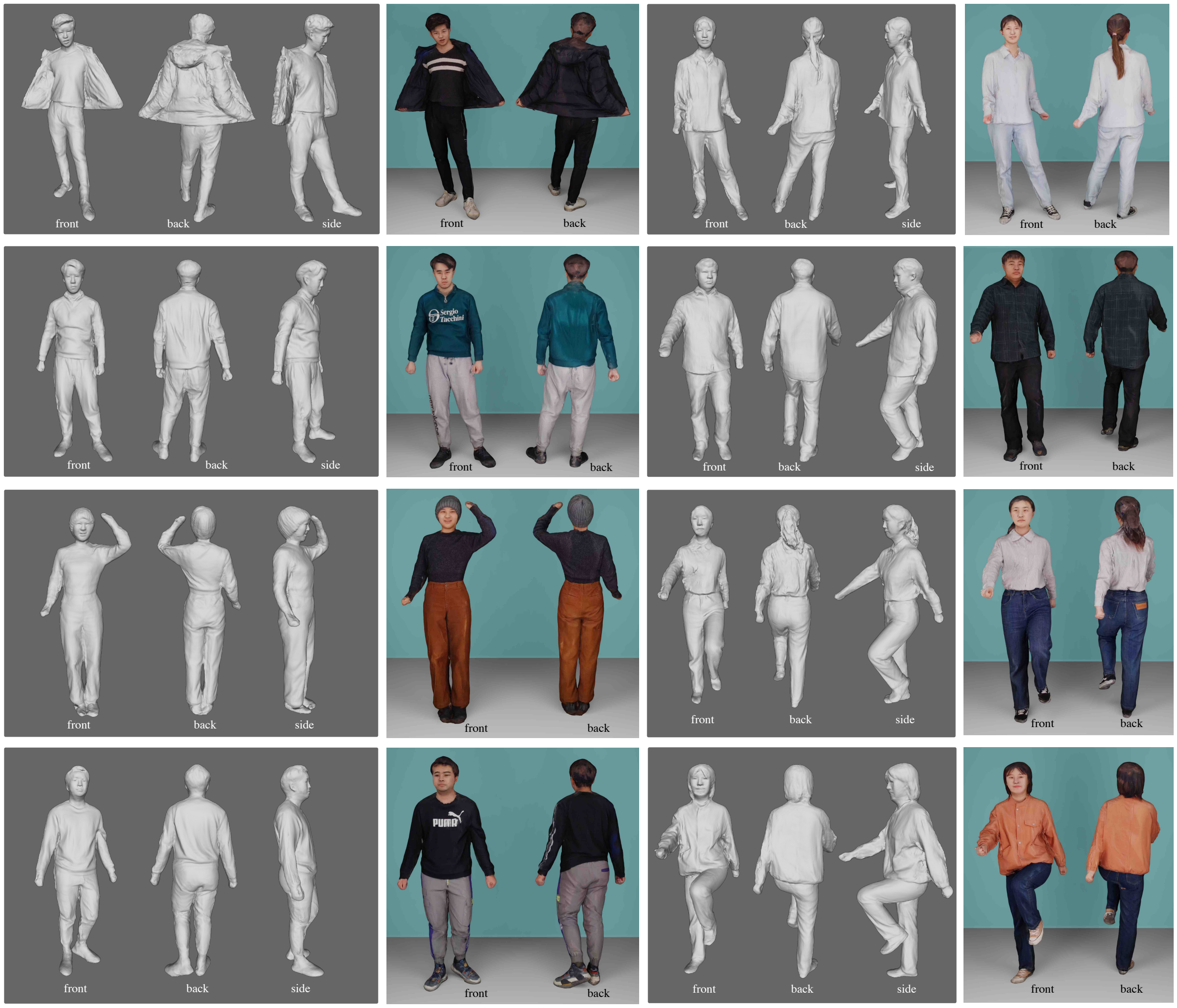


Figure G: More of our textured results.

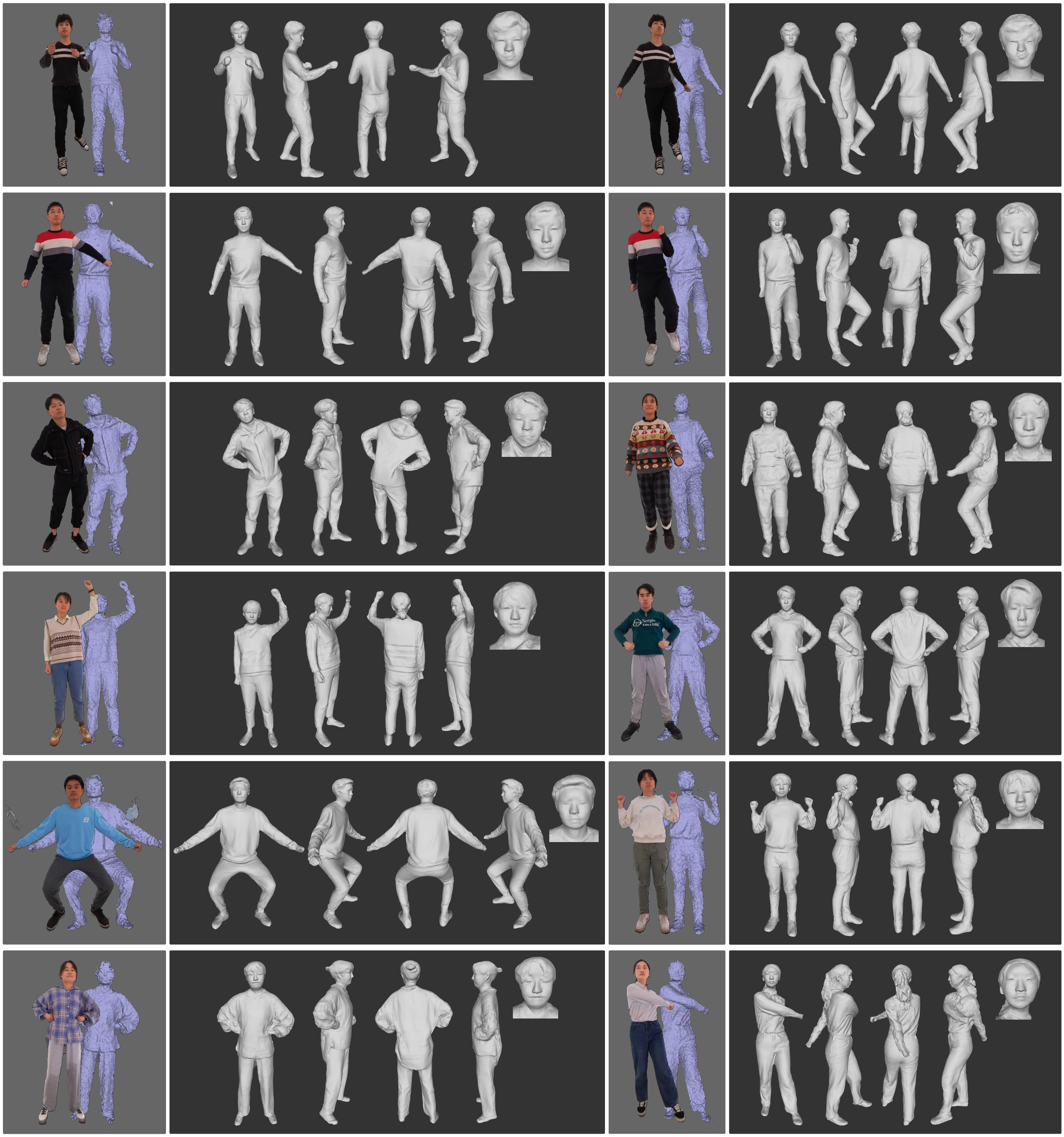


Figure H: More of our reconstructed results

References

[1] Victoria Fernández Abrevaya, Adnane Boukhayma, Philip Torr, and Edmond Boyer. Cross-modal deep face normals with deactivable skip connections. In *CVPR*, 2020. 4

[2] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. In *ECCV*, 2020. 2, 3

[3] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *CVPR*, 2020. 1

[4] Nasir Hayat. Face skin segmentation. [EB/OL]. `https://github.com/nasir6/face-segmentation/` Accessed Jun 5, 2021. 1

[5] Tong He, John Collomosse, Hailin Jin, and Stefano Soatto. Geo-pifu: Geometry and pixel aligned implicit functions for single-view human reconstruction. In *NeurIPS*, 2020. 2

[6] Yang Hong, Juyong Zhang, Boyi Jiang, Yudong Guo, Ligang Liu, and Hujun Bao. Stereopifu: Depth aware clothed human digitization via stereo vision. In *CVPR*, 2021. 2, 3

[7] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 1

[8] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *CVPR*, 2021. 1

[9] William Lorensen and Harvey Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH*, 1987. 2

[10] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 2

[11] 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. In *NeurIPS*, 2017. 1

[12] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, 2019. 1, 2, 3

[13] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, 2020. 1, 2, 3

[14] Vladimiros Sterzentsenko, Leonidas Saroglou, Anargyros Chatzitofis, Spyridon Thermos, Nikolaos Zioulis, Alexandros Doumanoglou, Dimitrios Zarpalas, and Petros Daras. Self-supervised deep depth denoising. In *ICCV*, 2019. 3

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[16] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE TPAMI*, 2021. 2

[17] Lizhen Wang, Zhiyuan Chen, Tao Yu, Chenguang Ma, Liang Li, and Yebin Liu. Faceverse: a fine-grained and detail-controllable 3d face morphable model from a hybrid dataset. In *CVPR*, 2022. 4

[18] Shi Yan, Chenglei Wu, Lizhen Wang, Feng Xu, Liang An, Kaiwen Guo, and Yebin Liu. Ddrnet: Depth map denoising and refinement for consumer depth cameras using cascaded cnns. In *ECCV*, 2018. 3

[19] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *CVPR*, 2020. 1

[20] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *CVPR*, 2021. 1, 2

[21] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *CVPR*, 2018. 2

[22] Xiaoxing Zeng, Xiaojiang Peng, and Yu Qiao. Df2net: A dense-fine-finer network for detailed 3d face reconstruction. In *ICCV*, 2019. 4