Supplementary Material for M3D-VTON

Fuwei Zhao[1], Zhenyu Xie[1], Michael Kampffmeyer[2], Haoye Dong[1]
Songfang Han[3], Tianxiang Zheng[4], Tao Zhang[4], Xiaodan Liang[1*]
[1]Shenzhen Campus of SYSU, [2]UiT The Arctic University of Norway, [3]UC San Diego, [4]Momo

1. Architecture Details of M3D-VTON

1.1. MPM Architecture

In MPM, all the three sub-branches share the encoders $\mathcal{E}_C$ and $\mathcal{E}_A$, which have the same structure and only differ in the number of input channels (3 for clothes, 29 for agnostic person). They consist of four convolutional layers with stride 2 followed by two convolutional layers with stride 1. The regressor $\mathcal{R}$ in the warping branch has a sequence of two 2-strided convolutional layers, two 1-strided convolutional layers and one dense layer. The dense layer regresses the x and y coordinate offsets of the TPS control points and therefore has an output size of $2 \times 3 \times 3 = 18$. The decoder $\mathcal{D}_S$ and $\mathcal{D}_Z$ are used for the segmentation branch and the depth estimation branch, respectively. They both contain one 1-strided convolutional layer, succeed by four upsample-convolutional blocks to output the 512×320 resolution results. See Table 1 for a detailed overview of this module.

1.2. TFM Architecture

In TFM, we use the same 12-layer UNet generator $\mathcal{G}_T$ as in CP-VTON, which contains six 2-strided down-sampling convolutional layers and six up-sampling layers. Each convolutional layer is followed by an Instance Normalization layer and Leaky ReLU with a slope of 0.2. The 4 output channels are split into 3 channels for the rendered person and 1 channel for the fusion mask. See Figure 1 and Table 2 for the mask fusion process and the detailed architecture of TFM.

1.3. DRM Architecture

For DRM, we adopt an UNet-like generator $\mathcal{G}_Z$ with two residual blocks similar to that in NormalGAN [4]. The output channel is set to 2 (one for front depth, one for back depth). See Table 3 for the detailed architecture of DRM.

Encoder $\mathcal{E}_C$ / $\mathcal{E}_A$		
Layer	Type	Output Size
Input	Target Clothes / Agnostic Representation	(512,320,3/29)
Conv1	Conv 4x4 stride 2, ReLU, INorm	(256,160,64)
Conv2	Conv 4x4 stride 2, ReLU, INorm	(128,80,128)
Conv3	Conv 4x4 stride 2, ReLU, INorm	(64,40,256)
Conv4	Conv 4x4 stride 2, ReLU, INorm	(32,20,512)
Conv5	Conv 3x3 stride 1, ReLU, INorm	(32,20,512)
Conv6	Conv 3x3 stride 1, ReLU, INorm	(32,20,512)
Conv7	Conv 3x3 stride 1, ReLU, L2Norm	(32,20,512)
Feature Correlation		
Output Size	(32,20,640)	
Feature Concatenation		
Output Size	(32,20,1024)	
Regressor $\mathcal{R}$		
Layer	Type	Output Size
Input	Correlated Fature	(32,20,640)
Conv1	Conv 4x4 stride 2, INorm, ReLu	(16,10,512)
Conv2	Conv 4x4 stride 2, INorm, ReLu	(8,5,256)
Conv3	Conv 3x3 stride 1, INorm, ReLu	(8,5,128)
Conv4	Conv 3x3 stride 1, INorm, ReLu	(8,5,64)
Linear	Linear	50
Decoder $\mathcal{D}_S$ / $\mathcal{D}_Z$		
Layer	Type	Output Size
Input	Concatenated Featrue	(32,20,1024)
Conv1	Conv 3x3 stride 1, INorm, ReLu	(32,20,512)
Upsample1	Upsample x2, Conv 3x3 stride 1, INrom, ReLu	(64,40,256)
Upsample2	Upsample x2, Conv 3x3 stride 1, INrom, ReLu	(128,80,128)
Upsample3	Upsample x2, Conv 3x3 stride 1, INrom, ReLu	(256,160,64)
Upsample4	Upsample x2, Conv 3x3 stride 1, INrom, ReLu	(256,160,2/20)

Table 1. The architecture of MPM. (INorm refers to InstanceNorm).

TFM Generator $\mathcal{G}_T$		
Layer	**Type**	**Output Size**
Input	Input	(512,320,9)
Conv1	Conv 4x4, LReLU	(256,160,64)
Conv2	Conv 4x4, INorm, LReLU	(128,80,128)
Conv3	Conv 4x4, INorm, LReLU	(64,40,256)
Conv4	Conv 4x4, INorm, LReLU	(32,20,512)
Conv5	Conv 4x4, INorm, LReLU	(16,10,512)
Conv6	Conv 4x4, INorm, LReLU	(8,5,512)
Upsample1	Upsample x2, Conv 3x3, INorm, ReLU, Skip connection from Conv5	(16,10,1024)
Upsample2	Upsample x2, Conv 3x3, INorm, ReLU, Skip connection from Conv4	(32,20,1024)
Upsample3	Upsample x2, Conv 3x3, INorm, ReLU, Skip connection from Conv3	(64,40,512)
Upsample4	Upsample x2, Conv 3x3, INorm, ReLU, Skip connection from Conv2	(128,80,256)
Upsample5	Upsample x2, Conv 3x3, INorm, ReLU, Skip connection from Conv1	(256,160,128)
Upsample6	Upsample x2, Conv 3x3	(512,320,4)

Table 2. The architecture of TFM.

2. Dataset Examples

Figure 3.(a) illustrates how the proposed dataset is constructed. We firstly apply PIFuHD [3], a state-of-the-art single-image 3D human reconstruction method, on the MPV2D dataset [1] to obtain high-fidelity 3D human meshes, before the meshes are orthographically projected to front and back depth maps. A data point in our MPV3D dataset is consequently represented by a four-tuple (person,

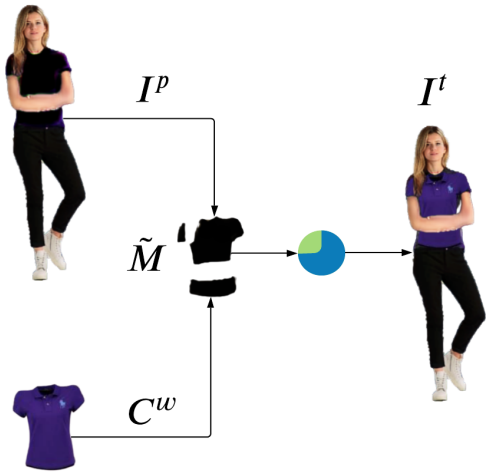

Figure 1. The mask fusion process in TFM. The fusion mask $\tilde{M}$ is used to fuse C^w and I^p to the refined try-on result I^t, which can be formulated as: $I^t = C^w \odot \tilde{M} + \tilde{I}^c \odot (1 - \tilde{M})$.

DRM Generator $\mathcal{G}_Z$		
Layer	**Type**	**Output Size**
Input	Input	(512,320,12)
Conv1	Conv 3x3 stride=1, ELU	(512,320,64)
	Conv 3x3 stride=2, ELU, InstanceNorm	(256,160,128)
Conv2	Conv 3x3 stride=1, ELU	(256,160,128)
	Conv 3x3 stride=1, ELU	(256,160,128)
	Conv 3x3 stride=2, ELU, InstanceNorm	(128,80,256)
Conv3	Conv 3x3 stride=1, ELU	(128,80,256)
	Conv 3x3 stride=1, ELU	(128,80,256)
	Conv 3x3 stride=2, ELU, InstanceNorm	(64,40,512)
Conv4	Conv 3x3 stride=1, ELU	(64,40,512)
	Conv 3x3 stride=1, ELU	(64,40,512)
	Conv 3x3 stride=2, ELU	(32,20,1024)
	Conv 3x3 stride=1, InstanceNorm	(32,20,1024)
Res1	Residual module block	(32,20,1024)
Res2	Residual module block	(32,20,1024)
Upsample1	Upsample x2, skipp connection from Conv3	(64,40,1536)
	Conv 3x3 stride=1, ELU	(64,40,512)
	Conv 3x3 stride=1, ELU	(64,40,512)
	Conv 3x3 stride=1, ELU, InstanceNorm	(64,40,512)
Upsample2	Upsample x2, skipp connection from Conv2	(128,80,768)
	Conv 3x3 stride=1, ELU	(128,80,256)
	Conv 3x3 stride=1, ELU	(128,80,256)
	Conv 3x3 stride=1, ELU, InstanceNorm	(128,80,256)
Upsample3	Upsample x2, skipp connection from Conv1	(256,160,384)
	Conv 3x3 stride=1, ELU	(256,160,128)
	Conv 3x3 stride=1, ELU	(256,160,128)
	Conv 3x3 stride=1, ELU, InstanceNorm	(256,160,128)
Upsample4	Upsample x2	(512,320,64)
	Conv 3x3 stride=1, ELU	(512,320,64)
	Conv 3x3 stride=1, ELU	(512,320,64)
	Conv 3x3 stride=1, Tanh	(512,320,2)

Table 3. The architecture of the DRM.

clothing, front depth, back depth). More examples from our dataset are shown in Figure 3.(b).

3. Limitation and Future Work

Monocular depth estimation is a highly ill-posed problem due to the well-known depth ambiguity. The proposed M3D-VTON tends to fail in predicting depth for ambiguous person parts in poses that are rarely presented in the training data. Figure. 2 illustrates monocular examples that lack sufficient information for our method to perceive accurate relative depth relations. Features extracted from cross legs (Figure 2.(a)) and occluded bent arms (Figure 2.(b)) in 2D space are not reliable enough for depth estimator to tell apart the front and back side, or to stitch them up. We suspect that the results for those would improve with better representation of diverse poses in the training data or more prior information like the 3D pose, which we aim to solve in future work.

(a) (b)

Figure 2. Failure cases for our methods on ambiguous occluded person parts. We display the front part and back part of the reconstructed point cloud in red and blue, respectively. Our method typically fails when poses are not represented well in the training dataset.

4. Additional Results

4.1. 2D Texture Fusion Results

We show more qualitative comparisons of the texture fusion results among our proposed M3D-VTON and other existing 2D try-on methods in Figure 4.

4.2. 3D Virtual Try-on Results

We show more qualitative comparison of the 3D virtual try-on results between M3D-VTON and other hybrid methods in Figure 5. For rotated views, please see the supplementary video.

References

[1] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bochao Wang, Hanjiang Lai, Jia Zhu, Zhiting Hu, and Jian Yin. Towards multi-pose guided virtual try-on network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9026–9035, 2019. 1

[2] 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 *The IEEE International Conference on Computer Vision (ICCV)*, pages 2304–2314, 2019. 5

[3] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for

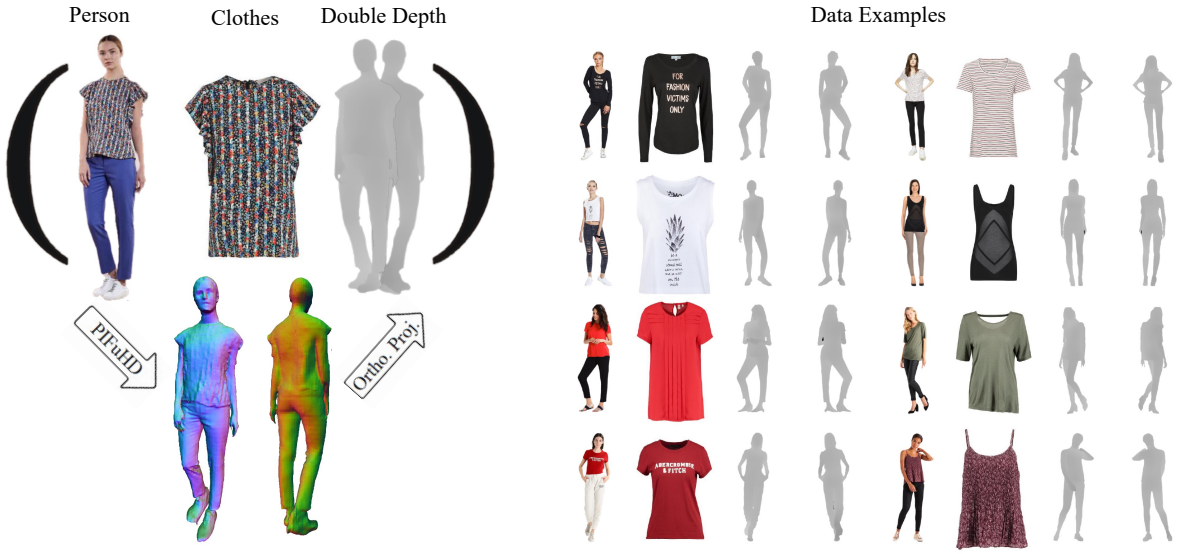

Figure 3. Procedure of how the MPV3D dataset is generated. (a) The three items enclosed in parentheses form one data data point (person, clothing, front depth, back depth) in our dataset. (b) More data examples.

high-resolution 3d human digitization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 84–93, 2020. 1

[4] Lizhen Wang, Xiaochen Zhao, Tao Yu, Songtao Wang, and Yebin Liu. Normalgan: Learning detailed 3d human from a single rgb-d image. In *Proceedings of the European Conference on Computer Vision*, 2020. 1, 5

Figure 4. Additional visual comparison with the other methods in the texture fusion module. The proposed M3D-VTON produces more realistic results.

Figure 5. Additional qualitative comparisons of 3D try-on results. The human mesh generated by the proposed M3D-VTON contains more texture details and accurate shape compared with PIFu [2] and NormalGAN [4].