Supplementary Material: Learning to Dress 3D People in Generative Clothing

Qianli Ma[1], Jinlong Yang[1], Anurag Ranjan[1,2], Sergi Pujades[4], Gerard Pons-Moll[5], Siyu Tang[*3], and Michael J. Black[1]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany [2]University of Tübingen, Germany
[3]ETH Zürich, Switzerland [4]Université Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, France
[5]Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
{qma, jyang, aranjan, black}@tue.mpg.de
sergi.pujades-rocamora@inria.fr gpons@mpi-inf.mpg.de siyu.tang@inf.ethz.ch

S1. Implementation Details

S1.1. CAPE network architecture

Here we provide the details of the CAPE architecture, as discribed in the main paper Sec. 4.1. We use the following notations:

- x: data, $\hat{x}$: output (reconstruction) from the decoder, z: latent code, p: the prediction map from the discriminator.
- LReLU: leaky rectified linear units with a slope of 0.1 for negative values.
- CONV$_n$: Chebyshev graph convolution layer with n filters.
- CONVBlock$_n$: convolution block comprising CONV$_n$ and LReLU.
- CResBlock$_n$: conditional residual block that uses CONV$_n$ as filters.
- DS$_n$: linear graph downsampling layer with a spatial downsample rate n.
- US$_n$: linear graph upsampling layer with a spatial upsample rate n.
- FC$_m$: fully connected layer with output dimension m.

Condition module: for pose θ, we remove the parameters that are not related to clothing, e.g. head, hands, fingers, feet and toes, resulting in 14 valid joints from the body. The pose parameters from each joint are represented by the flattened rotational matrix (see Sec. 4.1, "Conditional model"). This results in the overall pose parameter $\mathbb{R}^{9\times 14}$. We feed this into a small fully-connected network:

$$\begin{aligned}\theta \in \mathbb{R}^{9\times 14} &\rightarrow \text{FC}_{63} \rightarrow \text{LReLU}\\ &\rightarrow \text{FC}_{24} \rightarrow z_\theta \in \mathbb{R}^{24}\end{aligned}$$

The clothing type c refers to the type of "outfit", i.e. a combination of upper body clothing and lower body clothing. There are four types of outfits in our training data: *longlong*: long sleeve shirt / T-shirt / jersey with long pants; *shortlong*: short sleeve shirt / T-shirt / jersey with long pants; and their opposites, *shortshort* and *longshort*. As the types of clothing are discrete by nature, we represent them using a one-hot vector, $c \in \mathbb{R}^4$, and feed it into a linear layer:

$$c \in \mathbb{R}^4 \rightarrow \text{FC}_8 \rightarrow z_c \in \mathbb{R}^8$$

Encoder:

$$\begin{aligned}x \in \mathbb{R}^{3\times 6890} &\rightarrow \text{CONVBlock}_{64} \rightarrow \text{DS}_2\\ &\rightarrow \text{CONVBlock}_{64} \rightarrow \text{CONVBlock}_{128} \rightarrow \text{DS}_2\\ &\rightarrow \text{CONVBlock}_{128} \rightarrow \text{CONVBlock}_{256} \rightarrow \text{DS}_2\\ &\rightarrow \text{CONVBlock}_{256} \rightarrow \text{CONVBlock}_{512}\\ &\rightarrow \text{CONVBlock}_{512} \rightarrow \text{CONV}_{64}^{1\times 1}\\ &\rightarrow \text{FC}_{18} \rightarrow z_\mu \in \mathbb{R}^{18}\\ &\hookrightarrow \text{FC}_{18} \rightarrow z_\sigma \in \mathbb{R}^{18}\end{aligned}$$

Decoder:

$$\begin{aligned} z \in \mathbb{R}^{18} &\xrightarrow[z_\theta, z_c]{\text{concat}} z' \in \mathbb{R}^{18+24+8} \\ &\rightarrow \text{FC}_{64 \times 862} \rightarrow \text{CONV}_{512}^{1\times 1} \\ &\rightarrow \text{CResBlock}_{512} \rightarrow \text{CResBlock}_{512} \rightarrow \text{US}_2 \\ &\rightarrow \text{CResBlock}_{256} \rightarrow \text{CResBlock}_{256} \rightarrow \text{US}_2 \\ &\rightarrow \text{CResBlock}_{128} \rightarrow \text{CResBlock}_{128} \rightarrow \text{US}_2 \\ &\rightarrow \text{CResBlock}_{64} \rightarrow \text{CResBlock}_{64} \\ &\rightarrow \text{CONV}_3 \rightarrow \hat{x} \in \mathbb{R}^{3\times 6890} \end{aligned}$$

Discriminator:

$$\begin{aligned} \hat{x} \in \mathbb{R}^{3\times 6890} &\xrightarrow[\text{tile}\{z_\theta, z_c\}]{\text{concat}} \hat{x}' \in \mathbb{R}^{(3+24+8)\times 6890} \\ &\rightarrow \text{CONVBlock}_{64} \rightarrow \text{DS}_2 \\ &\rightarrow \text{CONVBlock}_{64} \rightarrow \text{DS}_2 \\ &\rightarrow \text{CONVBlock}_{128} \rightarrow \text{DS}_2 \\ &\rightarrow \text{CONVBlock}_{128} \rightarrow \text{DS}_2 \\ &\rightarrow \text{CONV}_1 \rightarrow p \in \mathbb{R}^{1\times 431} \end{aligned}$$

Conditional residual block: We adopt the graph residual block from Kolotouros et al. [5] that includes Group Normalization [13], non-linearity, graph convolutional layer and graph linear layer (i.e. Chebyshev convolution with polynomial order of 0). After the input to the residual block, we append the condition vector to every input node along the feature channel. Our CResBlock is given by

$$\begin{aligned} x_{\text{in}} \in \mathbb{R}^{i\times P} &\xrightarrow[\text{tile}\{z_\theta, z_c\}]{\text{concat}} x' \in \mathbb{R}^{(i+24+8)\times P} \\ &\rightarrow \text{ResBlock}_j \rightarrow x_{\text{out}} \in \mathbb{R}^{j\times P} \end{aligned}$$

where x_{in} is the input to the CResBlock. x_{in} has P nodes and i features on each node. ResBlock is the graph residual block from [5] that outputs j features on each node.

S1.2. Training details

The model is trained for 60 epochs, with a batch size of 16, using stochastic gradient descent with a momentum of 0.9. The learning rate starts from an initial value of 2×10^{-3}, increases with a warm-up step of 2×10^{-3} / epoch for 4 epochs, and then decays with a rate of 0.99 after every epoch.

The convolutions use the Chebyshev polynomial of order 2 for the generator, and of order 3 for the discriminator. An L2-weight decay with strength 2×10^{-3} is used as regularization.

We train and test our model for males and females separately. We split the male dataset into a training set of 26,574 examples and 5,852 test examples. The female dataset is split into a training set of 41,472 examples and a test set of 12,656 examples. Training takes approximately 15 minutes per epoch on the male dataset and 20 minutes per epoch on the female dataset.

S2. Image Fitting

Here we detail the objective function, experimental setup and extended results of the image fitting experiments, as described in the main manuscript Sec. 6.3.

S2.1. Objective function

Similar to [6], we introduce a silhouette term to encourage the shape of the clothed body to match the image evidence. The silhouette is the set of all pixels that belong to a body's projection onto the image. Let $\hat{S}(\beta, \theta, c, z)$ be the rendered silhouette of a clothed body mesh $M(\beta, \theta, c, z)$ (see main paper Eq. (4)), and S be the ground truth silhouette. The silhouette objective is defined by the bi-directional distance between S and $\hat{S}(\cdot)$:

$$\begin{aligned} E_S(\beta, \theta, z; c, S, K) = &\sum_{x \in \hat{S}} l(x, S) \\ &+ \sum_{x \in S} l(x, \hat{S}(\beta, \theta, c, z)) \end{aligned} \tag{1}$$

where $l(x, S)$ is the L1 distance from a point x to the closest point in the silhouette S. The distance is zero if the point is inside S. K is the camera parameter that is used to render the mesh to the silhouette on the image plane. The clothing type is derived from upstream pipeline and is therefore not optimized.

For our rendered scan data, the ground truth silhouette and clothing type are acquired for free during rendering. For in-the-wild images, this information can be acquired using human-parsing networks, e.g. [2].

After the standard SMPLify optimization pipeline, we apply the clothing layer to the body, and apply an additional optimization step on body shape β, pose θ and clothing structure z, with respect to the overall objective:

$$\begin{aligned} E_{\text{total}} = &E_J(\beta, \theta; K, J_{\text{est}}) + \lambda_S E_S(\beta, \theta, z; c, S, K) \\ &+ \lambda_\theta E_\theta(\theta) + \lambda_\beta E_\beta(\beta) \\ &+ \lambda_a E_a(\theta) + \lambda_z E_z(z) \end{aligned} \tag{2}$$

The overall objective is a weighted sum of the silhouette loss with other standard SMPLify energy terms. E_J is a weighted 2D distance between the projected SMPL joints and the detected 2D points, J_{est}. $E_\theta(\theta)$ is the mixture of Gaussians pose prior term, $E_\beta(\beta)$ the shape prior term, $E_a(\theta)$ the penalty term that discourages unnatrual joint bents, and $E_z(z)$ the L2-regularizer on z to prevent extreme clothing deformations. For more details about these terms please refer to Bogo et al. [1].

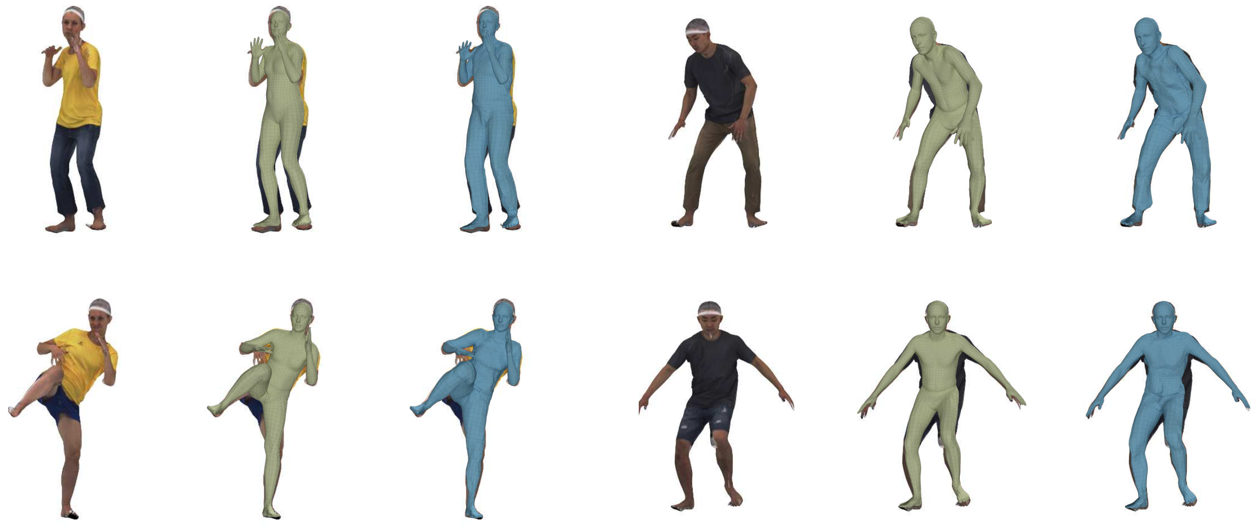

Figure 1: Qualitative results on the rendered meshes from CAPE dataset. Minimally-clothed fitting results from SMPLify [1] are shown in green; results from our method are shown in blue.

S2.2. Data

We render 120 textured meshes (aligned to the SMPL topology) randomly selected from the test set of the CAPE dataset that include variations in gender, pose and clothing type, at a resolution of 512×512. The ground truth meshes are used for evaluation. Examples of the rendering are shown in Fig. 1.

S2.3. Setup

We re-implement the SMPLify work by Bogo et al. [1] in Tensorflow, using the gender neutral SMPL body model. Compared to the original SMPLify, there are two major changes. First, we do not include the interpenetration error term, as it slows down the fitting but brings little performance gain [4]. Second, we use OpenPose for the ground truth 2D keypoint detection instead of DeepCut [8].

S2.4. Evaluation

We measure the mean square error (MSE) between ground truth vertices $\mathcal{V}_{\text{GT}}$ and reconstructed vertices from SMPLify $\mathcal{V}_{\text{SMPLify}}$, and from our pipeline (Eq. (2)) $\mathcal{V}_{\text{CAPE}}$, respectively. As discussed in Sec. 6.3, to eliminate the influence of the ambiguity caused by focal length, camera translation and body scale, we estimate the body scale s and camera translation t for both $\mathcal{V}_{\text{SMPLify}}$ and $\mathcal{V}_{\text{CAPE}}$. Specifically, we optimize the following energy function for $\mathcal{V} = \mathcal{V}_{\text{SMPLify}}$ and $\mathcal{V} = \mathcal{V}_{\text{CAPE}}$ respectively:

$$E = \underset{s,t}{\operatorname{argmin}} \frac{1}{N} \sum_{i \in \mathbf{C}} ||s(\mathcal{V}_i + t) - \mathcal{V}_{\text{GT},i}||^2 \tag{3}$$

where i is vertex index, $\mathbf{C}$ the set of clothing vertex indices, and N the number of elements in $\mathbf{C}$. Then, the MSE is computed with estimated scale $\hat{s}$ and translation $\hat{t}$ using:

$$\text{MSE} = \frac{1}{N} \sum_{i \in \mathbf{C}} ||\hat{s}(\mathcal{V}_i + \hat{t}) - \mathcal{V}_{\text{GT},i}||^2 \tag{4}$$

S2.5. Extended image fitting results

Qualitative results. Fig. 1 shows the reconstruction result of SMPLify [1] and our method on rendered meshes from the CAPE dataset. Quantitative results can be found in the main manuscript, Table 3. We also show qualitative results of CAPE fitted to images from the DeepFashion [7] dataset in Fig. 2. In general CAPE has better silhouette overlapping and in some cases improved pose estimation, but has also shown a few limitations that point to future work, as disucussed next.

Limitations and failure cases. Since our image fitting pipeline is based on SMPLify, it fails when SMPLify fails to predict the correct pose. Besides, while in this work the reconstructed clothing geometry only relies on the silhouette loss, it can further benefit from other losses such as the photometric loss. Recent regression-based methods have achieved improved performance on this task [3, 4, 5], and integrating CAPE with them is an interesting future line of work.

The CAPE model itself fails when the garment in the image is beyond its model space. As discussed in the main paper, CAPE inherits the limitations of the offset representation in terms of clothing types. Skirts, for example, have a different topology from human bodies, and can hence not be modeled by CAPE. Consequently, if CAPE is employed to fit images of people in skirts, it can only approximate with e.g. the outfit type *shortshort*, which fails to explain the observation in the image. The last row of Fig. 2 shows a few

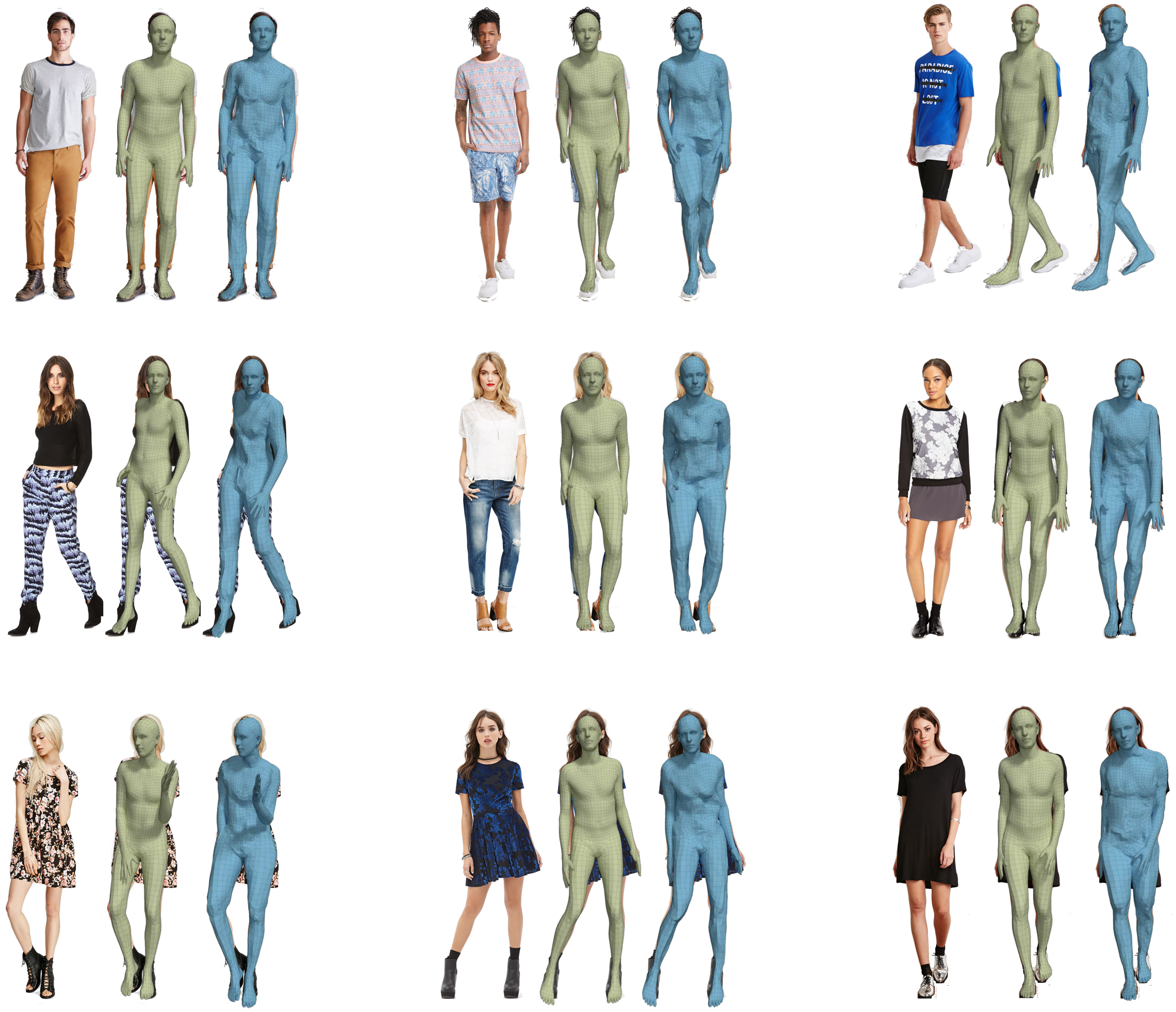

Figure 2: Qualitative results on fashion images from the DeepFashion dataset [7]. SMPLify [1] results are shown in green, our results are in blue.

of such failure cases on skirt images from the DeepFashion dataset [7]. Despite better silhouette matching than the minimally-clothed fitting, the reconstructed clothed bodies have the wrong garment type, which do not match the evidence in the image. Future work can explore multi-layer clothing models that can handle these garment types.

S2.6. Post-image fitting: re-dress and animate

After reconstructing the clothed body from the image, CAPE is capable of dressing the body with new styles by sampling the z variable, changing the clothing type by sampling c, and animating the mesh by sampling the pose parameter θ. We provide such a demo at `03:38` in the supplemental video[1]. This shows the potential in a wide range of applications.

S3. Extended Experimental Results

S3.1. CAPE with SMPL texture

As our model has the same topology as SMPL, it is compatible with all existing SMPL texture maps, which are mostly of clothed bodies. Fig. 3 shows an example texture applied to the standard minimally-clothed SMPL model (as done in the SURREAL dataset [11]) and to our clothed body model, respectively. Although the texture creates an illusion of clothing on the SMPL body, the overall shape remains skinny, oversmoothed, and hence unrealistic. In contrast, our model, with its improved clothing geometry, matches more naturally the clothing texture if the correct clothing type is given. This visual contrast becomes even stronger when the texture map has no shading information (albedo map), and when the object is viewed in a 3D setting. See `03:02` in the supplemental video for the comparison in 3D with the albedo map.

[1] available at https://cape.is.tue.mpg.de.

As a future line of research, one can model the alignment between the clothing texture boundaries and the underlying geometry by learning a texture model that is coupled to shape.

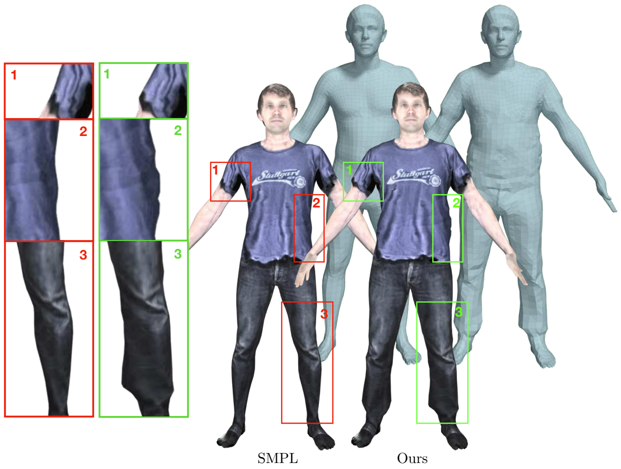

Figure 3: Front row: A clothing texture applied to the SMPL body and one of our generated clothed bodies. Back row: respective underlying geometry. Best viewed zoomed-in on screen.

S3.2. Pose-dependent clothing deformation

In the clip from `03:18` in the supplemental video, we animate a test motion sequence of a clothed body. We fix the clothing structure variable z and clothing type c, and generate new clothing offsets by only changing body pose θ (see main paper Sec. 6.2, "Pose-dependent clothing deformation"). Then the clothed body is brought to animation with the corresponding pose.

We compare it with traditional rig-and-skinning methods with fixed clothing offsets. An example of such a method is to dress a body with an instance of offset clothing layer using ClothCap [9], and re-pose using SMPL blend skinning.

The result is shown in both the original motion and in the zero-pose space (i.e. body is unposed to a "T-pose"). In the zero-pose space, we exclude the pose blend shapes (body shape deformation that is caused by pose variation), to highlight the deformation of the clothes. As the rig-and-skinning method uses a single fixed offset clothing layer, it looks static in the zero-pose space. In contrast, the clothing deformation generated by CAPE is pose-dependent, temporal coherent, and more visually plausible.

S4. CAPE Dataset Details

Elaborating on the main manuscript Sec. 5, our dataset consists of:

- 40K registered 3D meshes of clothed human scans for each gender.
- 8 male and 3 female subjects.
- 4 different types of outfits, covering 8 common garment types: short T-shirts, long T-shirts, long jerseys, long-sleeve shirts, blazers, shorts, long pants, jeans.
- Large variations in pose.
- Precise, captured minimally clothed body shape.

Table 1 shows a comparision with public 3D clothed human datasets. Our dataset is distinguished by accurate alignment, consistent mesh topology, ground truth body shape scans, and a large variation of poses. These features makes it not only suitable for studies on human body and clothing, but also for the evaluation of various Graph-CNNs. See Fig. 4 and `01:56` in the supplemental video for examples of the dataset. The dataset is available for research purposes at https://cape.is.tue.mpg.de.

Table 1: Comparison with other datasets of clothed humans.

Dataset	**Captured**	**Body Shape Available**	**Registered**	**Large Pose Variation**	**Motion Sequences**	**High Quality Geometry**
Inria dataset [14]	Yes	Yes	No	No	Yes	No
BUFF [15]	Yes	Yes	No	No	Yes	Yes
Adobe dataset [12]	Yes	No	Yes*	No	Yes	No
RenderPeople	Yes	No	No	Yes	No	Yes
3D People [10]	No	Yes	Yes*	Yes	Yes	Yes
Ours	Yes	Yes	Yes	Yes	Yes	Yes

∗ Registered per-subject, i.e. mesh topology is consistent only within the instances from the same subject.

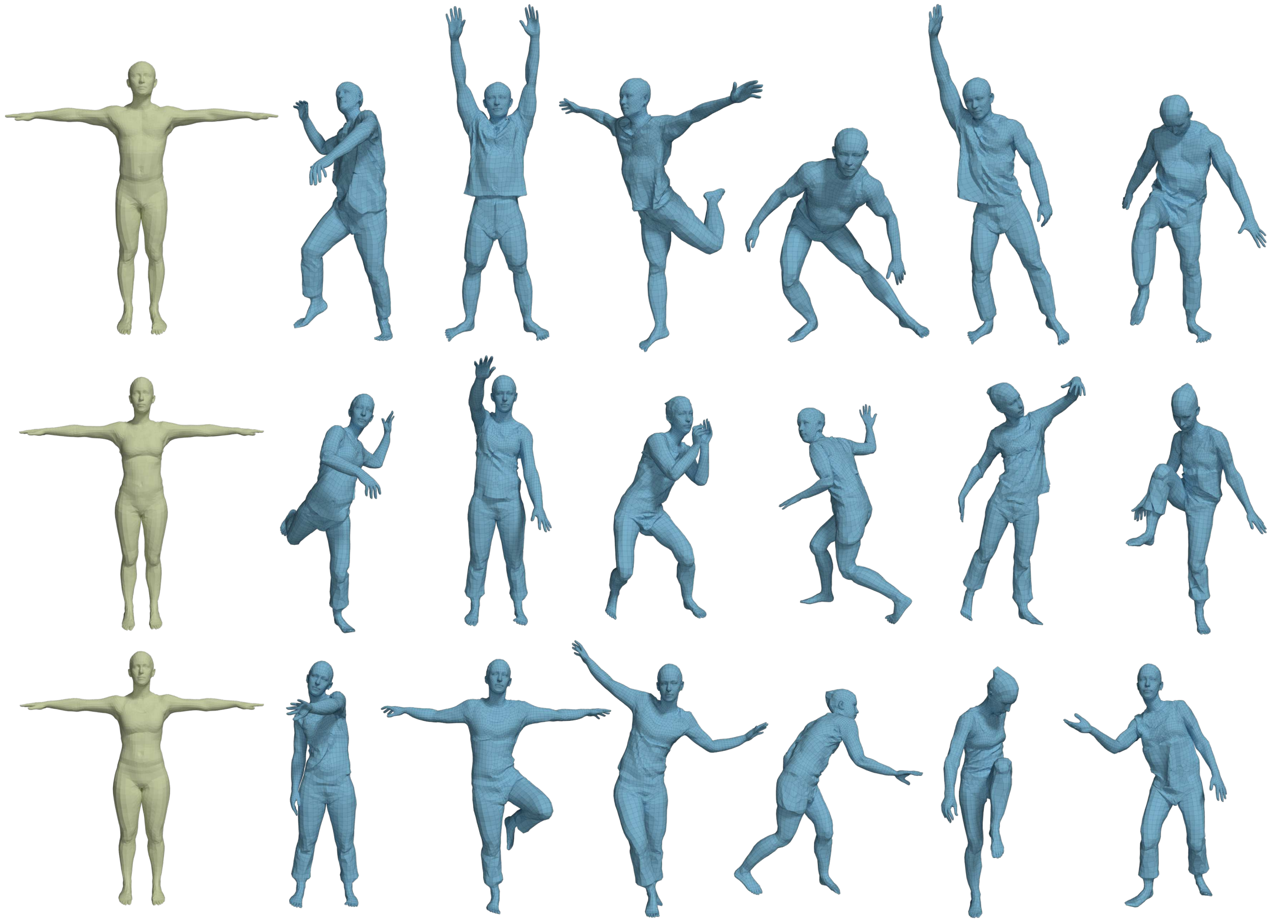

Figure 4: Examples from the CAPE dataset: we provide accurate minimal-dressed body shape (green), clothed body scans with large pose and clothing wrinkle variations, all registered to the SMPL mesh topology (blue).

References

[1] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision (ECCV)*. Springer, 2016. 2, 3, 4

[2] Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, Ming Yang, and Liang Lin. Instance-level human parsing via part grouping network. In *The European Conference on Computer Vision (ECCV)*. Springer, 2018. 2

[3] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[4] Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 3

[5] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[6] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3D and 2D human representations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[7] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 4

[8] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[9] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. Clothcap: Seamless 4D clothing capture and retargeting. *ACM Transactions on Graphics (TOG)*, 36(4):73, 2017. 5

[10] Albert Pumarola, Jordi Sanchez, Gary Choi, Alberto Sanfeliu, and Francesc Moreno-Noguer. 3DPeople: Modeling the Geometry of Dressed Humans. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 6

[11] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[12] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. In *ACM Transactions on Graphics (TOG)*, volume 27, page 97. ACM, 2008. 6

[13] Yuxin Wu and Kaiming He. Group normalization. In *The European Conference on Computer Vision (ECCV)*. Springer, 2018. 2

[14] Jinlong Yang, Jean-Sébastien Franco, Franck Hétroy-Wheeler, and Stefanie Wuhrer. Estimation of human body shape in motion with wide clothing. In *European Conference on Computer Vision*. Springer, 2016. 6

[15] Chao Zhang, Sergi Pujades, Michael J. Black, and Gerard Pons-Moll. Detailed, accurate, human shape estimation from clothed 3D scan sequences. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6