Convolutional Mesh Regression for Single-Image Human Shape Reconstruction **Supplementary Material**

Nikos Kolotouros, Georgios Pavlakos, Kostas Daniilidis
University of Pennsylvania

This supplementary material provides additional details that were not included in the main text due to space limitations. First, in Section 1 we provide additional informations about the training process. Then, we investigate the potential of regressing details like hair and clothing with our approach (Section 2) and extend our empirical evaluation (Section 3). Finally, in Section 4 we describe in detail the architecture of the models used in our experiments.

1. Training Details

During training we randomly rotate and flip the input images, rescale the bounding boxes and also introduce color jittering in the RGB input case. All input images are rescaled to 224×224 before feeding them in the encoder. For mixed training with Human3.6M [6] and UP-3D [10], since UP-3D is significantly smaller than Human3.6M we do not uniformly sample from all images. Instead, first we randomly pick one of the two datasets with probability 0.5, and then we select an image from this dataset uniformly at random. This ensures that an equal number of in-the-wild and indoor examples are included in our batch.

2. Clothing and hair

As suggested in the main manuscript, our approach should be able to capture details like hair and clothing which are not modeled by typical human body models. To demonstrate this potential, we use the data of Alldieck *et al*. [1] for training and apply our model on hold-out sequences. Interestingly, our regressed mesh (Fig. 1) indeed captures some rough details (e.g., hair bun and shorts). We clarify that these results are purely to demonstrate feasibility and the data is limited for a proper evaluation, but we believe this is a promising direction for future work.

3. Further experimental exploration

In the main manuscript we report results using images from Human3.6M and UP-3D, that provide 3D shape ground truth for training. Although we can expect to have access to such ground truth for indoor datasets (i.e., Human3.6M), in-the-wild examples do not typically come with 3D annotations. Here we demonstrate that our approach is applicable even when these annotations are not available.

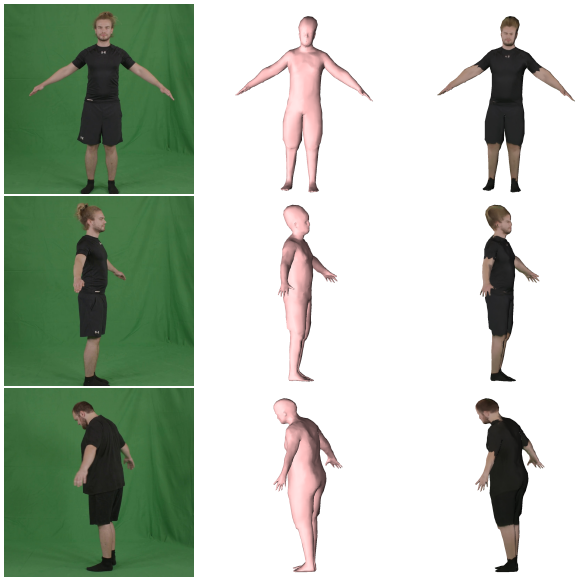

Figure 1: Qualitative results on the data of Alldieck *et al*. [1]. From left to right: input image, regressed mesh, ground truth texture applied on the regressed mesh.

Method	MPJPE	Reconst. Error
HMR [8]	88.0	58.1
Ours (H3.6M + LSP + MPII)	78.6	56.6
Ours (H3.6M + UP-3D)	**74.7**	**51.9**

Table 1: Evaluation of our approach on Human3.6M (Protocol 1) for weaker 2D annotations. The numbers are mean joint errors in mm. Training with 2D ground truth only for in-the-wild examples leads to less accurate results compared to our model trained on UP-3D data. However, we are still able to outperform [8] which is trained on significantly more data than our approach.

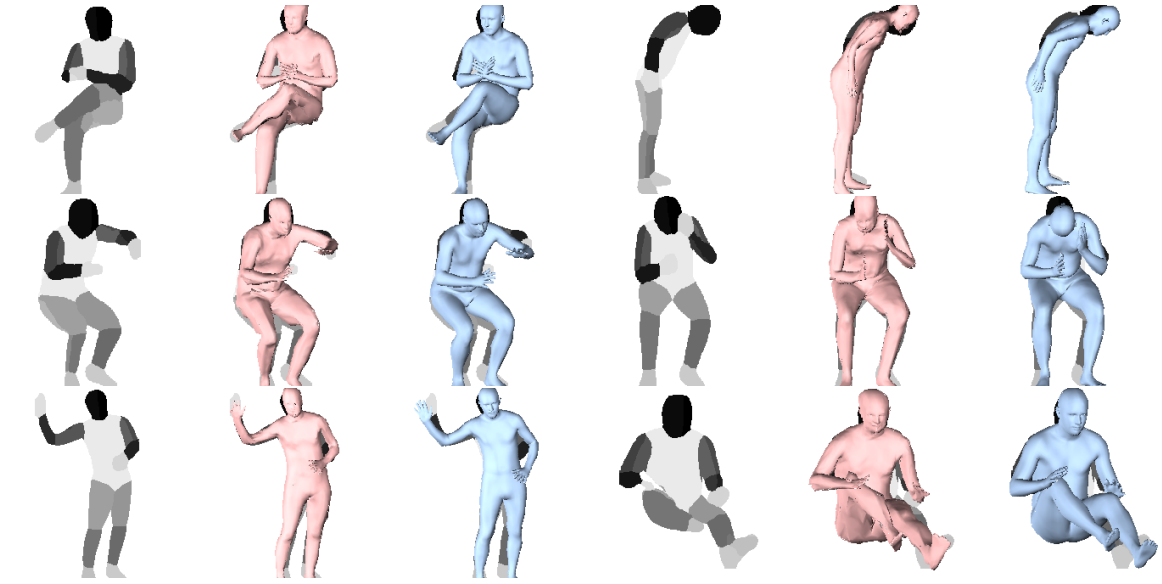

Figure 2: Qualitative results on Human 3.6M with parts segmentation input. With light pink color we indicate the regressed non parametric shape and with light blue the SMPL model regressed from the previous shape.

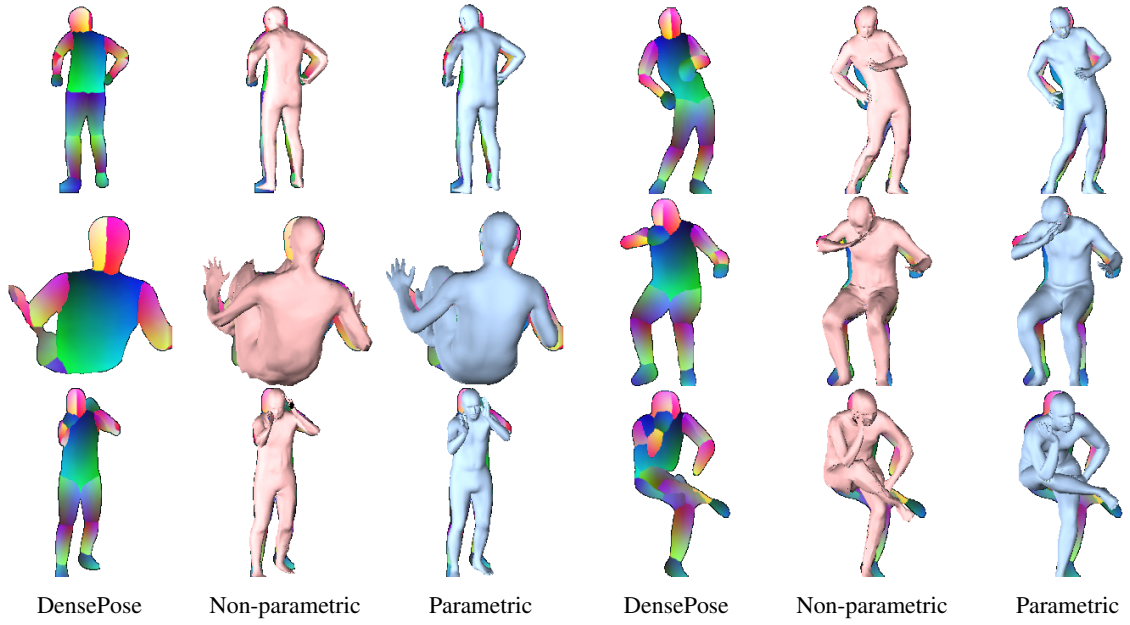

Figure 3: Qualitative results on Human3.6M with DensePose input. With light pink color we indicate the regressed non parametric shape and with light blue the SMPL model regressed from the previous shape.

To this end, we ignore the UP-3D data, and instead train with images from MPII [2] and LSP [7] that provide only 2D keypoint annotations. Effectively, for these examples, we use only the 2D reprojection loss and not the 3D shape loss. The results for this setting are reported in Table 1. We have also included the results of [8] that trains in a similar setting, i.e., using only 2D annotations for in-the-wild examples. Although the results of this training setting are worse compared to our best model trained on Human3.6M and UP-3D, we are still able to outperform [8] although they use significantly more data (i.e., COCO [11] and MPI-INF-3DHP [12]).

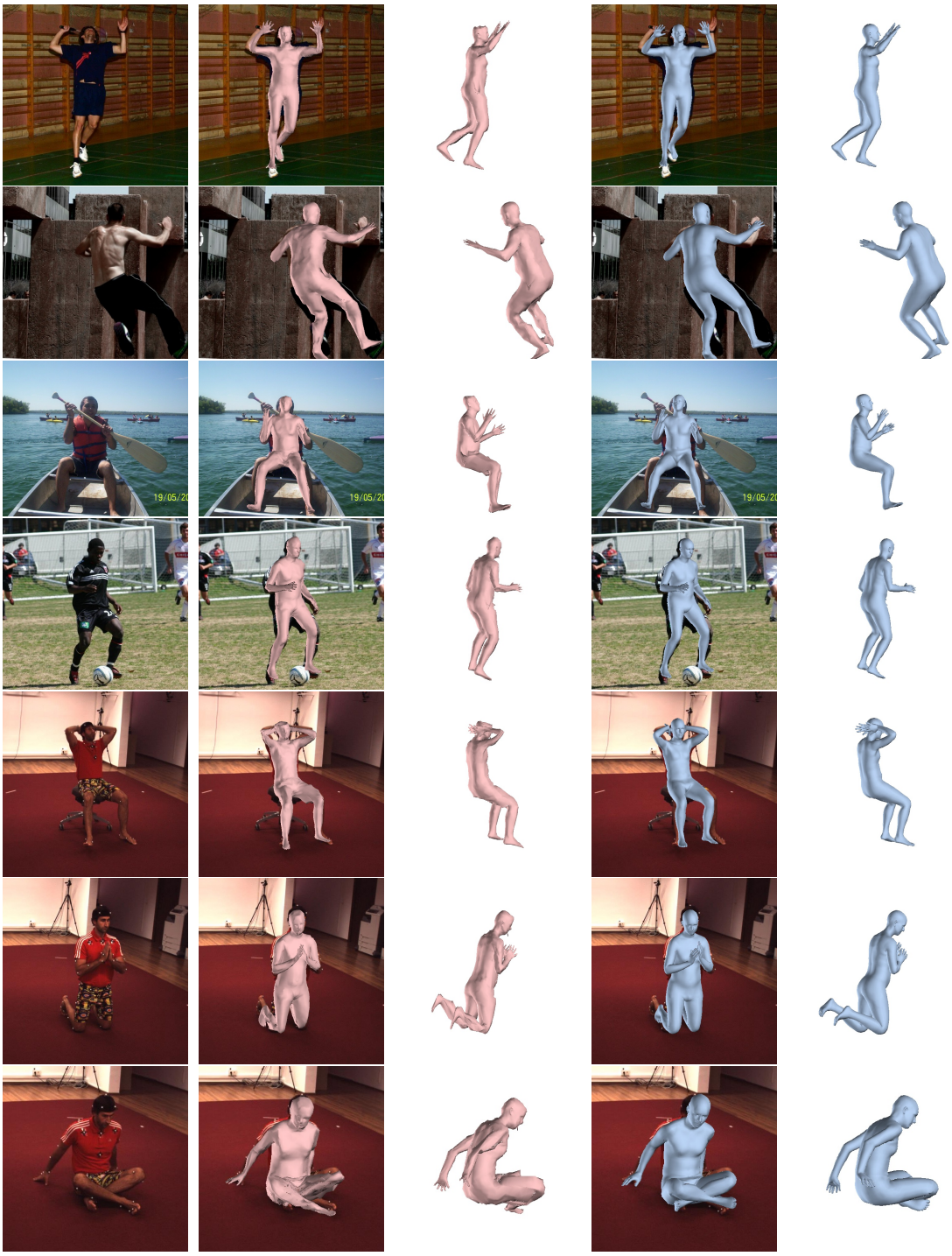

Figure 4: Visualization of our results from novel viewpoints. Rows 1-5: LSP [7]. Rows 6-7: Human3.6M [6]. From left to right: Input image, Non-parametric shape, Non-parametric shape (side view), Parametric shape, Parametric shape (side view).

Moreover, in Figure 2 and Figure 3 we include qualitative results when using part segmentations and DensePose [3] images respectively as the input representation. We can see that even with non-perfect detections, i.e., parts of the

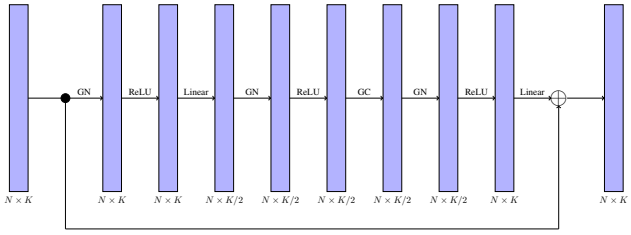

Figure 5: **Graph Residual Block**. Our basic building block for the Graph CNN is a redesign of the Bottleneck Residual Block [4]. GN stands for Group Normalization [13], and the Linear layers are simply per-vertex fully connected layers.

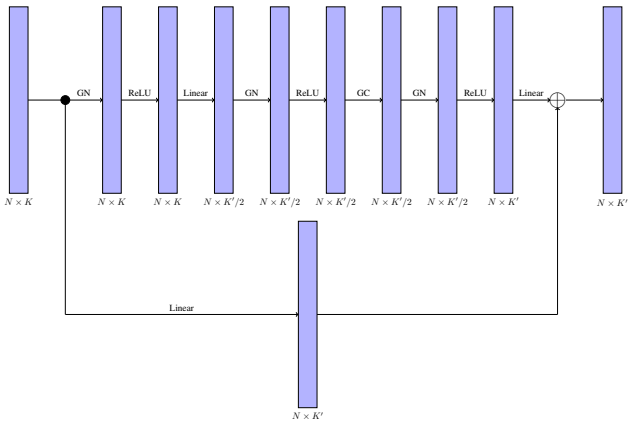

Figure 6: **Graph Residual Block v2**. This is the modified version of the Graph Residual Block when the number of input features is different from the number of output features.

body missing in some difficult poses, the network is still able to correctly regress the 3D shape.

Finally, in Figure 4 we present results of our approach visualized also from a novel viewpoint. This type of visualization allows us to inspect the accuracy of our results beyond the visible side which, and focus on the non-visible parts which is where we typically observe most errors.

4. Model Architecture

4.1. Graph CNN

As discussed in the main manuscript, the basic building block that we use in the Graph CNN is the Graph Residual Block depicted in Figure 5. It resembles the Bottleneck Residual Block [4], but we replace 1×1 convolutions with per-vertex Linear (fully connected) layers, 3×3 convolutions with the Graph convolutions proposed in [9] and Batch Normalization [5] with Group Normalization [13]. Whenever the number of input channels is different from the number of output channels, we feed the input to an additional

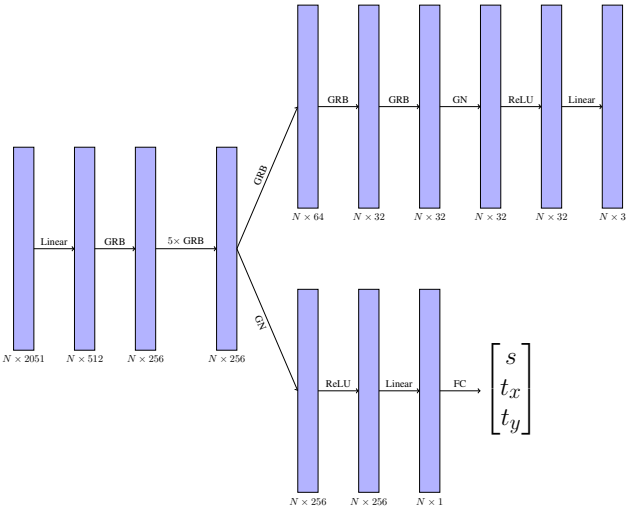

Figure 7: **Graph CNN**. Our Graph CNN makes use of the Graph Residual Block (GRB) of Figure 5 and Figure 6. The network eventually splits in 2 branches that predict the 3D shape and camera parameters s, t_x, t_y respectively.

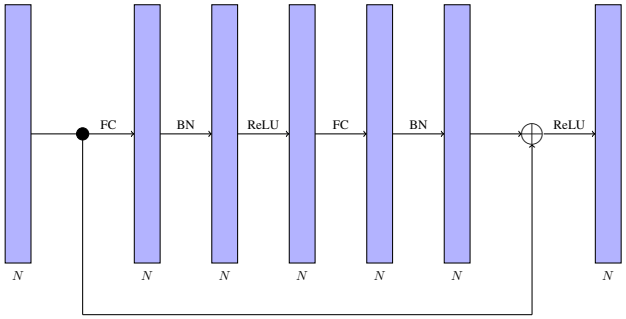

Figure 8: **Fully Connected Residual Block**. This figure depicts the residual block that is used in the MLP that regresses the SMPL parameters from the 3D shape. BN stands for Batch Normalization [5].

Linear layer that maps it to the correct feature map size before adding it to the output, as seen in Figure 6. Using this Graph Residual Block, the full network architecture used in all our experiments is depicted in Figure 7.

4.2. SMPL regressor

To estimate the SMPL parameters from the regressed shape we use a simple MLP with skip connections. The input to the MLP is the regressed 3D shape together with the template SMPL shape, both subsampled by a factor of 4. Subsampling here is essential to avoid the explosion in the number of parameters for the fully connected layers. Also, we found that including the template 3D shape in the input speeds up the learning significantly. The network architec-

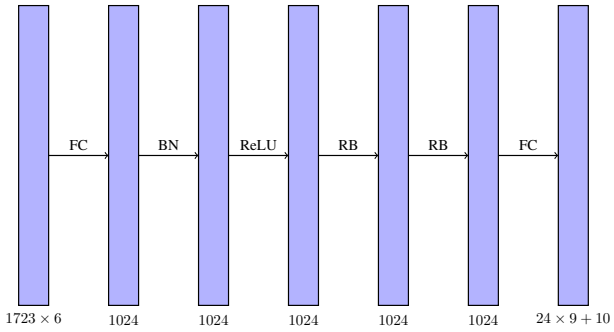

Figure 9: **SMPL Regressor**. This figure presents the architecture of the MLP that regresses the SMPL parameters from the 3D shape. RB stands for the Fully Connected Residual Block of Figure 8.

ture is shown in Figure 9. The input size is $1723 \times 3 \times 2$ (3D vertex coordinates for both the output and the template mesh), whereas the output size is $24 \times 3 \times 3 + 10$ (rotation matrices for each of the 24 joints and 10-dimensional SMPL shape parameters).

References

[1] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3D people models. In *CVPR*, 2018. 1

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 2

[3] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018. 3

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[6] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *PAMI*, 36(7):1325–1339, 2014. 1, 3

[7] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010. 2, 3

[8] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 1, 2

[9] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 4

[10] 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 *CVPR*, 2017. 1

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 2

[12] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3D human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):44, 2017. 2

[13] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 4