Dynamic Surface Function Networks for Clothed Human Bodies
– Supplemental Document –

Andrei Burov[1] Matthias Nießner[1] Justus Thies[1,2]
[1]Technical University of Munich [2]Max Planck Institute for Intelligent Systems, Tübingen

Abstract

In this supplemental document, we provide additional information about our Dynamic Surface Function Networks. Specifically, we detail the training procedure and weightings of the energy terms presented in the main paper (see Sec. A). In Fig. 4, we provide additional qualitative results and zoom-ins, to show the expressiveness of the proposed method. Additional comparisons and experiments are presented in Sec. B.

A. Implementation Details

A.1. Network Architecture

The dynamic surface function is represented as a multilayer perceptron (MLP). In our experiments, we use an 8-layer MLP with ReLU activation functions for the intermediate layer (each intermediate layer has a feature dimension of 256). The final output layer uses a tanh activation function, allowing us to specify a maximal amplitude of the offset surface (in our experiments 25cm). The network architecture is inspired by Mildenhall et al. [7], using the positional encoding for the sample point coordinate input. To represent pose-dependent deformations, we condition the dynamic surface function network also on pose parameters. Specifically, we compute the 'pose feature' $\mathcal{F} = [\mathbf{F}_1, \dots, \mathbf{F}_{23}] \in \mathbb{R}^{23\times 9}$, where $\mathbf{F}_k = (R_k - Id)$ is the feature component of a body part k (the root part is not included). This pose feature is describing the global pose of a human. Since most deformations are local (e.g., the pose of the leg does not influence the surface of an arm), we compute a local pose conditioning of a sample point based on the linear blend-skinning weights of SMPL. Specifically, we enable the pose conditioning of the corresponding joints defined by the SMPL skinning weights, as well as for the adjacent nodes (2-ring neighborhood, i.e., parent and grandparent node, as well as child and grandchild node):

$$\hat{\mathcal{F}} = (lbs_k \cdot \mathbf{N}_k) \cdot \mathcal{F},$$

where $lbs \in \mathbb{R}^{23}$ are the skinning weights of a sample point, $\mathbf{N} \in \mathbb{R}^{23\times 23}$ the 2-ring adjacency matrix.

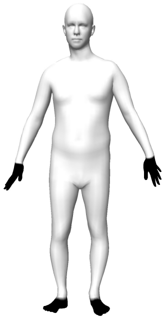

Figure 1: We exclude the hands and feet from optimization.

Note, during training we augment the pose conditioning $\mathcal{F}$ with noise to control overfitting. Specifically, we apply additive normal distributed noise with a standard deviation of 0.1.

A.2. Optimization

Optimizer settings The energy function is optimized in two stages. First, we fit the SMPL template to match the observations sequentially. We apply the L-BFGS [5] optimizer with line search, history size of 20 and 20 maximum iterations. We observe that using Adam [4] at this stage is not efficient, since it struggles to reconstruct rotations in the axis-angle form. The optimization is executed for 15 passes through the dataset with a fixed learning rate of 0.1 and then for another 15 passes with the learning rate linearly decreasing to 0.

During the second stage our objective is to reconstruct all parameters $\mathcal{P}$ jointly (MLP and SMPL parameters). At this stage, we use a standard Adam optimizer with $(0.9, 0.999)$ blending weights for the first and second momentum respectively. The optimization is carried out on random samples from the sequence with the first 100 global passes updated by a static learning rate of 0.00005 and remaining 300 passes by a linearly decaying learning rate. As soon as the learning rate is starting to decrease, we enable the dynamic conditioning, to capture the pose specific clothing deformations from reconstructed subjects.

Energy Term	Symbol	Value	Space
Sparse OpenPose	w_{OP}	1500	normalized image space
Dense Densepose	w_{DP}	25	3D space in meters
Dense Projective	w_{Proj}	100	3D space in meters
Silhouette	w_{Sil}	50	normalized image space
Surface Smoothness	w_{Reg}	1500	3D space in meters
Temporal Smoothness	w_T^{Surf}	100	3D space in meters
Temporal Smoothness	w_T^{Rot}	15	rotation matrices
Pair-wise Consistency	w_C	15	3D space in meters

Table 1: Energy term weights during optimization.

The optimization using ADAM takes approximately $60s$ per epoch (200 frames) while the initial fitting with L-BFGS takes around $700s$ per epoch.

Loss weights As described in the main paper, our optimization is based on a set of different energy terms. In Tab. 1, we specify the used weights during optimization. Note that we optimize in two stages as described above. For the initial fitting of the SMPL parameters, we increase the OpenPose weight w_{OP} to 10000 and disable the projective energy term during the first two optimization iterations (since the body is not yet roughly aligned with the body in the image, thus, leading to wrong projective correspondences). The temporal regularizers in this initial fitting procedure are turned on after the $5th$ pass. Note that all terms are normalized by their respective number of residuals (i.e., by the number of pixels). We prune projective correspondences based on distance ($0.5m$) and deviation in normals ($45°$).

A.3. Surface Sampling

For rendering, we need to sample the surface. We use the SMPL triangulation and subdivide it with a 1-to-4 subdivision scheme (each triangle is subdivided into 4). Based on these samples and the corresponding topology, we evaluate the dynamic surface function network to retrieve the actual surface position. These positions are then sent to the GPU rasterizer to render the surface, used for the analysis-by-synthesis process. Note that correspondences from DensePose [8] lead to additional samples on the surface.

A.4. Baseline Implementation

In the main paper, we discuss results based on the CAPE cloth model [6]. We leverage our fitting pipeline to optimize the energy with respect to the latent codes of the CAPE model. Specifically, we take the publicly available checkpoints for the *male* and *female* subjects (with clothing latent space of size 64, pose condition size of 32 and clothing type condition size of 32) and define the objective as latent codes' optimization for the CAPE decoder. In particular, we append the losses from the first stage of our optimization procedure to the Tensorflow [1] graph of the CAPE decoder, and initialize the reconstruction process with the parameters from the SMPL only optimization.

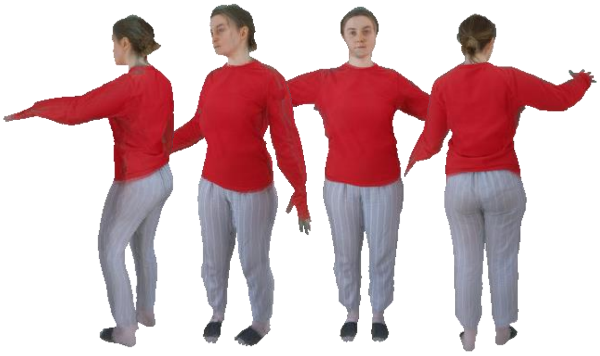

Figure 2: Our model can be extended to reconstruct the surface color. We use an MLP similar to the shape MLP to predict the color.

B. Additional Results

B.1. Comparisons

We provide an additional comparison to the incremental reconstruction method DoubleFusion [9] and to the topology-aware generative clothed human model SMPLicit [2]. As can be seen in Fig. 3, our approach is able to reconstruct more details than DoubleFusion, especially in the face region and also on the body. In contrast to DoubleFusion, our approach optimizes for a globally consistent representation that does not overfit to latest observations. The result produced by SMPLicit depicts a possible garment configuration for the specimen, however, as a method based on a generative model it does not match observed data as closely as an actual reconstruction method.

B.2. Reconstructing Surface Colors

In Fig. 2, we show the result of optimizing an additional MLP for the surface color. Specifically, we use the same architecture as the shape MLP and predict color values for each surface point. The MLP used in this experiment has 6 layers and a latent size of 256. We use 16 frequency bands for the positional encoding. We use a pretrained shape MLP, and train the color MLP with an ℓ_1 reconstruction and perceptual losses [3]. This experiment shows, that you can easily reconstruct the surface appearance of the person within our framework. Note that the incorporation of the color to refine the tracking and shape prediction is still open for follow-up works (i.e., joint optimization of the color and the shape MLP).

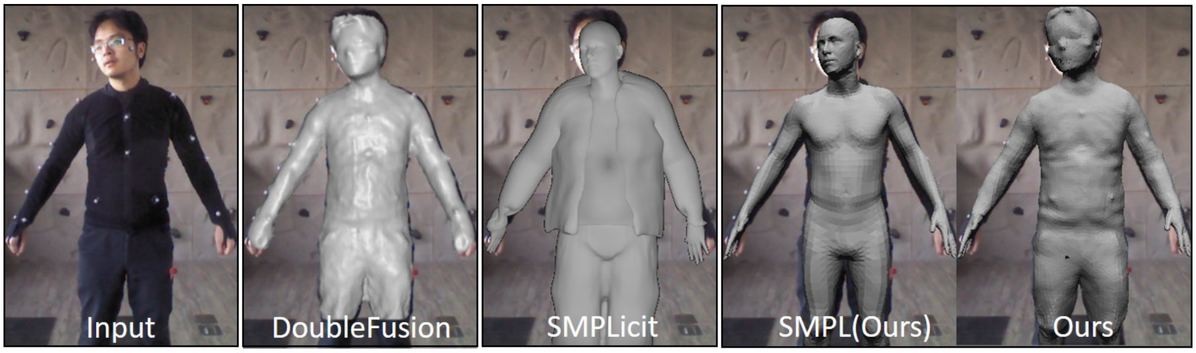

Figure 3: Additional qualitative comparison to DoubleFusion [9] and SMPLicit [2]. DoubleFusion is incrementally fusing the depth-observations to reconstruct the final body shape, while SMPLicit uses a generative approach to produce an output garment configuration that is close to the input. In contrast, our method globally optimizes for the actual shape.

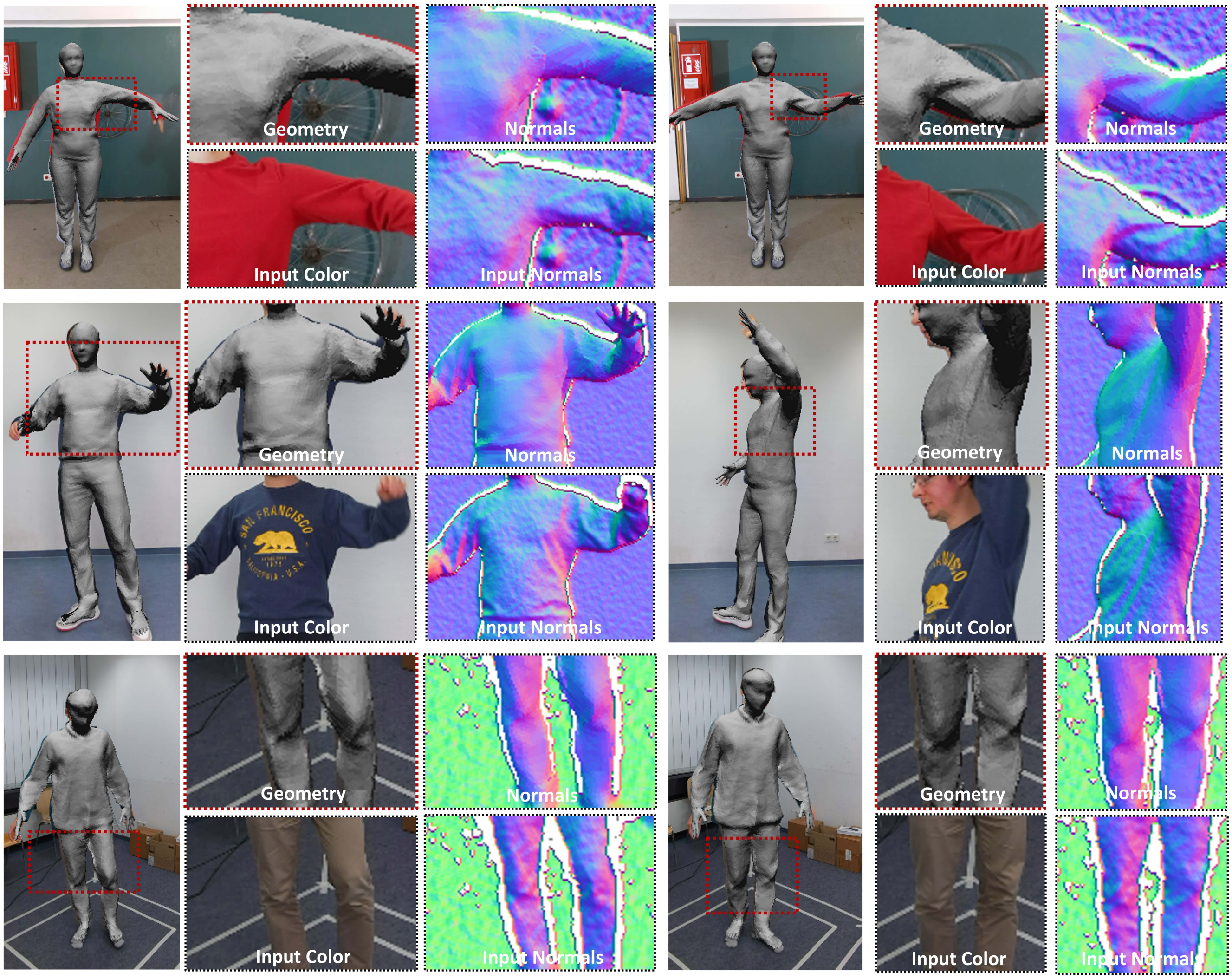

Figure 4: Dynamic Surface Function Networks are able to represent pose dependent wrinkles. Here, we show some sequences with corresponding close-ups to regions where pose dependent wrinkles occur (arms, upper-body and legs).

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016. 2

[2] Enric Corona, Albert Pumarola, Guillem Alenyà, Gerard Pons-Moll, and Francesc Moreno-Noguer. Smplicit: Topology-aware generative model for clothed people. In *CVPR*, 2021. 2, 3

[3] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 2

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 1

[5] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(1–3):503–528, Aug. 1989. 1

[6] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. Learning to Dress 3D People in Generative Clothing. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[7] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1

[8] Iasonas Kokkinos Riza Alp Güler, Natalia Neverova. Densepose: Dense human pose estimation in the wild. 2018. 2

[9] Tao Yu, Jianhui Zhao, Zhang Zerong, Kaiwen Guo, Dai Quionhai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performance with inner body shape from a depth sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, july 2019. 2, 3