Dynamic Surface Function Networks for Clothed Human Bodies

Andrei Burov Matthias Nießner Justus Thies
Technical University of Munich

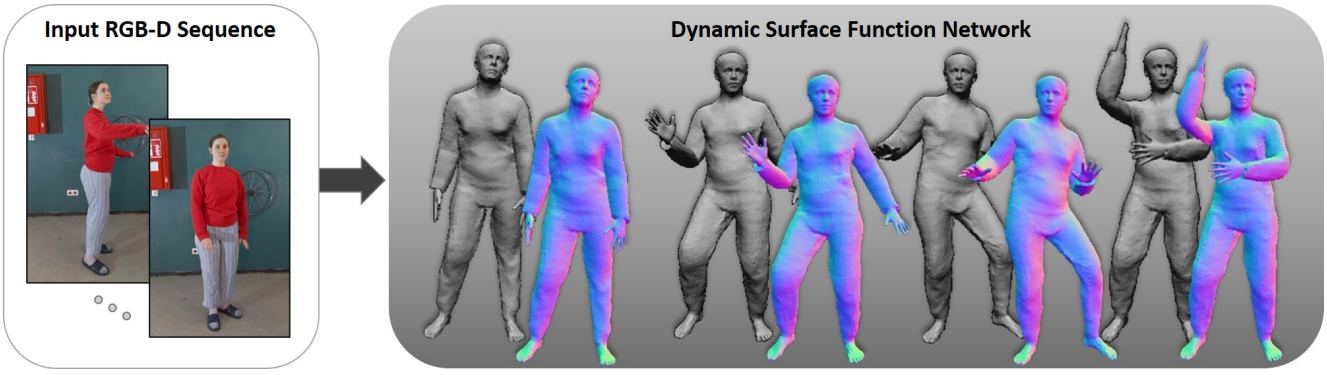

Figure 1: We introduce dynamic surface function networks which are trained on a short RGB-D input video sequence of a clothed human. Our continuous surface representation allows the modeling of clothes as well as the dynamic pose dependent deformations. We parametrize this representation by the kinematic model of SMPL [27], which facilitates full pose control at inference time (e.g., through joint rotations). Here, we show an animation sequence taken from the MOSH [28, 32] dataset.

Abstract

We present a novel method for temporal coherent reconstruction and tracking of clothed humans. Given a monocular RGB-D sequence, we learn a person-specific body model which is based on a dynamic surface function network. To this end, we explicitly model the surface of the person using a multi-layer perceptron (MLP) which is embedded into the canonical space of the SMPL body model. With classical forward rendering, the represented surface can be rasterized using the topology of a template mesh. For each surface point of the template mesh, the MLP is evaluated to predict the actual surface location. To handle pose-dependent deformations, the MLP is conditioned on the SMPL pose parameters. We show that this surface representation as well as the pose parameters can be learned in a self-supervised fashion using the principle of analysis-by-synthesis and differentiable rasterization. As a result, we are able to reconstruct a temporally coherent mesh sequence from the input data. The underlying surface representation can be used to synthesize new animations of the reconstructed person including pose-dependent deformations.

1. Introduction

Digital capture of human bodies is a rapidly growing research area in computer vision and computer graphics. There are many high impact applications, in particular, in the field of telepresencing and man-machine interaction that rely on the reconstruction of the surface as well as the motion of a person. For instance, for telepresencing in virtual reality (VR) or augmented reality (AR), the ultimate goal is to photo-realistically re-render the people who want to interact with each other. A key challenge is to capture and reproduce the natural motions, including the dynamically changing surface (especially clothing). In our work, we present a novel representation for the surface of a clothed human called *dynamic surface function networks* which captures pose-dependent surface deformations such as wrinkles of the clothing. In contrast to recent works on implicit representations [11, 7, 23], our representation explicitly models the surface of the human. Both representations have their advantages and disadvantages. We use an explicit surface representation to leverage fast forward rendering (exploiting the rasterization units of a GPU), and global surface correspondences between different frames. Specifically, our dynamic surface function network is attached to the surface of the SMPL [27] body model which gives us access to the

kinematic chain, as well as to a topology for rendering. In particular, our model represents a continuous offset surface that can be evaluated at arbitrary points of the SMPL surface (also within triangles). However, note that our representation is agnostic to the underlying parametric model and can also be used for other dynamically changing surfaces, e.g., human faces. The dynamic surface function network is trained in a person-specific fashion. To this end, we assume a short sequence (few seconds) of the person moving in front of a single consumer-level RGB-D camera. Note that we do not assume any specific motion sequences (e.g., standing in T-pose or alike). We jointly optimize the surface representation network as well as the pose parameters of the underlying SMPL body model. This global optimization strategy allows us to fuse all captured data into a consistent surface representation. This person-specific surface model can be animated explicitly using the joint control handles of the SMPL model, thus, allowing pose transfer (see Fig. 1).

To summarize, we present a method that allows reconstruction of a person-specific controllable body model based on a monocular input sequence captured by a commodity RGB-D sensor. Our key contributions are:

- an explicit surface representation network which is able to model the pose-dependent deformations of the surface, such as wrinkles of the clothing.
- a global analysis-by-synthesis formulation that allows for the joint optimization of the pose and the surface over the entire sequence, leading to a temporally consistent tracking of the person in the input sequence.

2. Related Work

Our work is based on an *explicit surface representation*. In the literature, explicit surface representations (especially, triangle meshes) are the most prominent representations for human faces and bodies [27, 37, 57]. However, implicit representations are also used to represent the surface of a body including the clothing [7, 11, 13, 23, 41, 42]. Implicit surface representations have the advantage of not needing special care for handling topological changes (as would be required for an explicit representation like a mesh). On the other hand, implicit representations do not provide explicit correspondences over a time series. For instance, methods such as Occupancy Flow [36] predict scene flow, but are limited by the sequence length. Methods like [16, 15, 48, 9] reconstruct implicit function of an object's geometry in a patch-based manner and empirically show consistency of patches across a sequence. PIFu [41] and PIFuHD [42] are able to reconstruct a pixel-aligned implicit function only based on RGB input images. They leverage a learned prior trained on a synthetic dataset of humans. Similarly, IF-Nets [11] learn a prior to reconstruct an implicit function of a human, based on pointcloud or depth-map inputs. IF-Nets have been extended in a follow-up work called IP-Nets [7] to fit an explicit surface to the reconstructed implicit surface (SMPL model + displacements).

Aside from these per-frame reconstruction methods, there exist methods that incrementally fuse observations into a discretized implicit function (volumetric SDF grid) [12]. The seminal work of DynamicFusion by Newcombe et al. [35] is able to reconstruct dynamically changing objects only based on a depth sequence. Follow-up works [24, 18, 14, 19] added additional color constraints or dense SDF alignment [44, 45]. BodyFusion [52] and DoubleFusion [53] added a deformation prior using a human skeleton. In contrast to these methods, our approach reconstructs a controllable mesh including pose-dependent deformations, allowing us to animate the reconstructed mesh. Note that these fusion methods work on a frame-to-frame tracking scheme.

Our method is an optimization-based approach that jointly optimizes the surface over the entire input sequence and does not require a learned prior as [7, 11, 41, 42]. The global optimization scheme of our method is closely related to MonoClothCap [51]. MonoClothCap gets an RGB sequence as input to track, and reconstructs a clothed human using an explicit surface representation. In order to handle clothing, they rely on a PCA model and shading-based refinement of the mesh. Another technique to model the surface of clothing explicitly has been shown in CAPE [30]. They learn a variational auto-encoder to represent offsets for clothing relative to the SMPL model based on high quality multi-view reconstructions using ClothCap [38]. This learned prior allows them to reconstruct clothed humans from single RGB inputs only, including pose-dependent deformations. Alldieck et al. [6] reconstruct a detailed human body shape based on a single RGB input leveraging a learned aligned image-to-image translation technique to regress texture maps for the geometry and color of the model. In contrast, our approach is not based on a learned prior and optimizes the surface based on the observations from an RGB-D camera. Our dynamic surface function network represents the pose-dependent geometry and is not restricted to a static surface [8, 49, 56, 22]. Methods that estimate the body shape under cloth [54, 50, 46, 34, 39] can be used as an initialization of our method. Our approach works with a consumer-grade RGB-D camera, and does not require a multi-camera setup to learn or reconstruct a person-specific template in advance [20, 21]. We use RGB-D data to explore the representation power of our surface network, but we see the potential of our representation to be used for video-based reconstruction similar to [5, 4, 3]. Note that our method is not designed for real-time use such as LiveCap [20], since our focus lies on the global reconstruction of a controllable mesh with a pose-dependent surface.

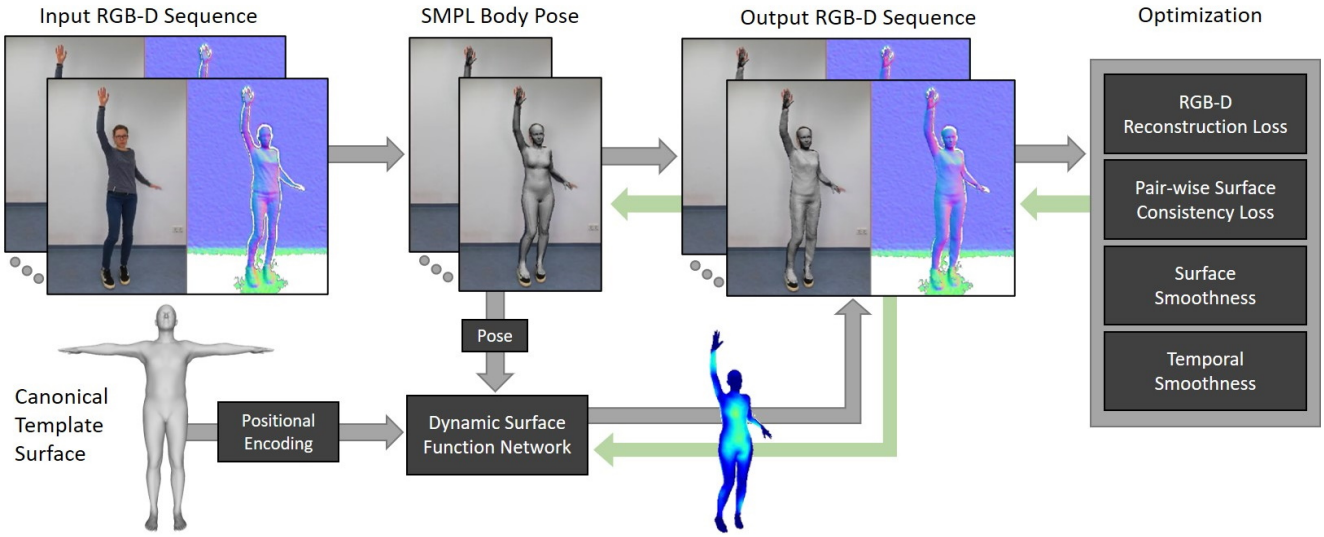

Figure 2: Given an RGB-D input sequence, we jointly optimize for the SMPL body parameters and a pose-conditioned offset surface function (Dynamic Surface Function Network) which is represented by an MLP. Specifically, the MLP gets a positionally encoded surface point of the canonical template surface and the pose parameters as input and predicts the offset surface location. The pose parameters and the weights of the network are optimized based on the optimization of a global analysis-by-synthesis energy formulation (green arrow).

3. Method

We formulate the reconstruction of the human body as an energy optimization problem, globally, across the entire input sequence. To model deforming clothing, we use an explicit offset surface function that is embedded on the SMPL surface. This offset surface is represented as a multi-layer perceptron (MLP). For an overview of our method refer to Fig. 2. In the following, we will detail the model definition and the energy formulation.

3.1. Model Definition

Our method is built upon the SMPL body model [27] and extends it to handle non-rigid offsets represented as a multi-layer perceptron (MLP).

Constrained SMPL model The SMPL model is parametrized by the shape parameters $\beta \in \mathbb{R}^{10}$ and the joint parameters $\delta \in \mathbb{R}^{72}$. We use a hard constraint to enforce the parameters to stay in a predefined range. That is, we use a differentiable mapping function $\delta(x)$ that maps our unconstrained joint parameters $x \in \mathbb{R}^{72}$, in an element-wise fashion, to a fixed range of the SMPL parameters:

$$\delta(x) = \delta_{min} + \frac{\tanh(x)+1}{2} \cdot (\delta_{max} - \delta_{min}).$$

Note that we use axis-angle representation. Thus, we constrain the angle-scaled axis to be in a min-max bounding box. We compute δ_{min} and δ_{max} from the MOSH dataset [28]. In our experiments, this hard constraint was effective in stabilizing the tracking without the need of any additional regularization, e.g., using VPoser [37] (especially, for scenarios with more occlusions) or pose-conditioned joint angle limits [2].

Dynamic Offset Surface To model the dynamic offset surface on top of the SMPL model, we employ an MLP which is conditioned on the reparameterized SMPL parameters x. Specifically, we embed the offsets $O_\Theta(v, x)$ in the canonical pose of the SMPL template surface T. The offset of the surface point v, given the pose parameters x is computed using a 8-layer ReLU-MLP with 256 feature channels per intermediate layer (Θ being the learnable parameters of the MLP). We apply positional encoding [33] to the input point v before feeding it into the MLP (using 10 frequencies). The pose conditioning is concatenated to the point input in matrix form. Note that we do not use the pose of the root joint into consideration. This representation of the pose conditioning is similar to the linear pose-dependent correctives of the SMPL model. Note that the 3D space allows us to have a continuous input to the MLP without the need of handling texture seams, distortions or alike. To rasterize the surface represented by the MLP, we sample the surface at surface points using a subdivided SMPL topology.

Model Given the constrained SMPL model and the dynamic offset model, we represent the vertices of a body as:

$$V(\beta, x, \Theta) = LBS(V_{SMPL}(\beta, \delta(x)) + O_\Theta(T, x), \delta(x))$$

$LBS(V', \delta)$ is the linear blend skinning function that applies the rotations defined by δ to the 'unposed' vertices $V' = V_{SMPL} + O$. $V_{SMPL}(\beta, \delta(x))$ computes the unposed SMPL model surface based on the shape PCA and the pose-dependent corrective space (see SMPL [27]). In our experiments, we exclude the hands and feet from the optimization (see supplemental material).

3.2. Fitting Energy Formulation

Given an RGB-D video sequence with $N+1$ frames, we minimize the following global energy:

$$E_{Seq}(\mathcal{P}) = \sum_{i=0}^{N} E_F^i(\mathcal{P}) + \sum_{i=1}^{N-1} E_T^i(\mathcal{P}) + \sum_{i=0}^{N}\sum_{j=0}^{N} E_C^{i,j}(\mathcal{P})$$

This energy formulation considers per-frame energies E_F^i, temporal energies E_T^i, and pair-wise surface consistency constraints $E_C^{i,j}$. $\mathcal{P}$ is the set of unknowns; namely the MLP weights Θ, the shape β and the per frame pose parameters x_i.

Per-frame Energy The per-frame energy is based on data-terms using the color frame $\mathcal{C}_i$ and depth frame. Given the intrinsics of the depth camera, we back-project the depth maps into the camera space using the pinhole camera model Π, resulting in 3D locations per pixel which we call $\mathcal{D}_i$. Based on the color frame $\mathcal{C}_i$, we estimate the 2D joint positions $\mathcal{J}_i^{OP}$ using OpenPose [10] and dense correspondences $\mathcal{M}_i^{DP}$ using DensePose [40]. In addition, we use Graphonomy [17] to estimate the silhouette $\mathcal{S}_i^G$ of the person in the input image. With these inputs, we define the per-frame energy as:

$$\begin{aligned} E_F^i(\mathcal{P}) = {} & w_{OP} \cdot E_{OpenPose}^i(\mathcal{P}) + w_{DP} \cdot E_{DensePose}^i(\mathcal{P}) \\ & + w_{Proj} \cdot E_{Projective}^i(\mathcal{P}) + w_{Sil} \cdot E_{Silhouette}^i(\mathcal{P}) \\ & + w_{Reg} \cdot E_{Reg}^i(\mathcal{P}) \end{aligned}$$

The detected 2D joint locations are used as a sparse energy term:

$$E_{OpenPose}^i(\mathcal{P}) = 1/K_J \cdot |\mathcal{J}_{i,j}^{OP} - \Pi(\mathcal{J}_{i,j})|$$

where $K_J = 25$ is the number of joints and $\mathcal{J}_i$ are the corresponding regressed joints of the SMPL model. The per-pixel DensePose energy term $E_{DensePose}^i$ is defined as:

$$E_{DensePose}^i(\mathcal{P}) = \sum_{(p,c)\in\mathcal{M}_i^{DP}} \frac{P2P(V_c, \mathcal{D}_i(p))}{K_{DP}}$$

K_{DP} is the number of valid correspondences $\mathcal{M}_i^{DP}$ estimated by DensePose (p is the pixel and c the correspondence on the SMPL surface). $P2P(V_c, xyz)$ measures the ℓ_1 point-to-point distance from the observation xyz to the surface point of the model $V_c = Sample(c, V(\beta, x_i, \Theta))$. The function $Sample(c, V(\beta, x, \Theta))$ computes the surface point based on the vertex indices (i_0, i_1, i_2) and barycentric coordinates (b_0, b_1, b_2) provided by the correspondence $c = (i_0, i_1, i_2, b_0, b_1, b_2)$. In addition to the DensePose term, we use a dense data term using projective correspondences $\mathcal{M}_i^{Proj}$. The projective energy term is defined as:

$$\begin{aligned} E_{Projective}^i(\mathcal{P}) = \sum_{(p,c)\in\mathcal{M}_i^{Proj}} & P2P(V_c, \mathcal{D}_i(p)) + N2N(V_c, \mathcal{D}_i(p)) \\ & + P2N(V_c, \mathcal{D}_i(p)) \end{aligned}$$

$N2N(V_c, \mathcal{D}_i(p))$ measures the cosine similarity of the normals from the observation point $\mathcal{D}_i(p)$ to the surface point V_c. $P2N(V_c, \mathcal{D}_i(p))$ measures point to plane distance between the observation point $\mathcal{D}_i(p)$ and the tangent plane at the source point V_c.

The per-frame silhouette energy term $E_{Silhouette}^i$ computes the difference between the mask predictions from [17] and the rendered subject's silhouette $\mathcal{S}_i(\mathcal{P})$. It is defined as:

$$E_{Silhouette}^i(\mathcal{P}) = |\mathcal{S}_i^G - \mathcal{S}_i(\mathcal{P})|$$

Note that our differentiable rasterizer explicitly applies edge sampling at the boundary of the silhouette of the surface to compute gradients.

To optimize for a smooth surface, we use the regularizer $E_{Reg}^i(\mathcal{P})$. Based on the topology of the template surface T, we apply a Laplacian regularizer:

$$E_{Regularizer}^i(\mathcal{P}) = \sum_{m\in T} \left| V_m - \frac{\sum_{n\in neigh_m} V_n}{|neigh_m|} \right|^2$$

Here, $neigh_m$ denotes the 1-ring neighborhood of the m-th vertex of the template topology.

Temporal Energy The temporal regularizer $E_T^i(\mathcal{P})$ is defined as:

$$E_T^i(\mathcal{P}) = w_T^{surf} \cdot E_{T_{surf}}^i(\mathcal{P}) + w_T^{rot} \cdot E_{T_{rot}}^i(\mathcal{P})$$

We use a Laplacian regularizer on the surface of the model in the temporal domain:

$$E_{T_{surf}}^i(\mathcal{P}) = |V_i - (V_{i-1} + V_{i+1})/2|^2$$

In addition, we apply a temporal regularizer on the joint rotation matrices $R_i = R(\delta(x_i))$:

$$E_{T_{rot}}^i(\mathcal{P}) = |R_i - R_{i-1}|^2 + |R_i - R_{i+1}|^2$$

Pair-wise Surface Consistency In each optimization step for each frame i, we randomly select another frame j from the sequence to measure surface consistencies. Specifically, we measure the difference of the offset surfaces:

$$E_C^{i,j}(\mathcal{P}) = w_C \cdot \omega(x_i, x_j) |\mathcal{V}_j \cdot (O_\Theta(T, x_i)) - O_\Theta(T, x_j))|$$

$\omega(x_i, x_j) = exp(-|R(\delta(x_i)) - R(\delta(x_j))|^2)$ measures the similarity of the poses in the two frames (excluding the root joint pose) and $\mathcal{V}_j$ denotes the surface visibility in frame j.

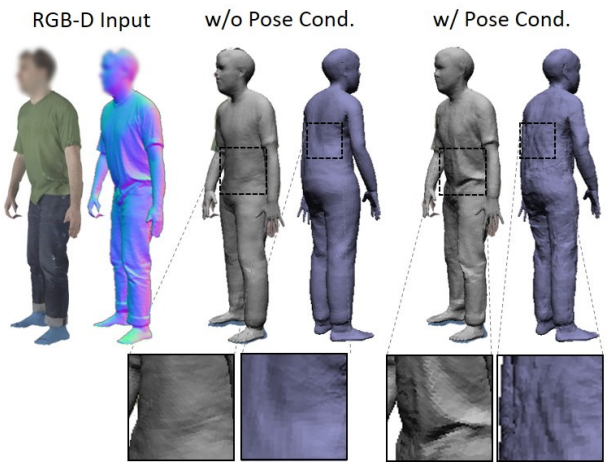

Figure 3: Without pose conditioning, only a static surface function is learned, and dynamically changing pose-dependent details such as wrinkles are not reconstructed. In blue, we show the view rotated by 180°.

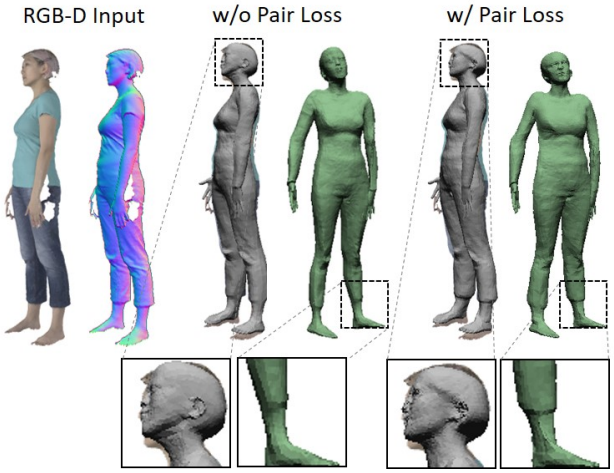

Figure 4: To reconstruct a consistent surface, we apply a pair-wise surface consistency loss, which measures the surface difference of pairs of frames weighted by their pose similarity. In green, we show the view rotated by 90°.

3.3. Optimization Scheme

The energy formulation in Eq. 3.2 is optimized in two stages. Specifically, we first minimize the energy only considering the SMPL body parameters for shape and pose to get a good initial estimate. We use L-BFGS [26] to optimize for these shape and pose parameters in sequential order (we initialize the parameters with the parameters of the previous frame). Using this initial fitting, we optimize for all parameters $\mathcal{P}$ in a joint optimization using ADAM [25] with random sampling of the input images. We refer to the supplemental material for the used hyperparameters.

4. Results

In this section, we evaluate our method on synthetic as well as on real-world data. For quantitative evaluations, we use synthetic sequences based on the BUFF [55] dataset. Real data inputs are captured with a Microsoft Kinect Azure at a depth resolution of 640x576 and color image resolution of $1080p$ at 15 fps (each recording is 200 frames long).We align color images with depth maps using OpenCV and apply distortion correction. These sequences are used to demonstrate the applicability of our method to real-world data, as well as to show qualitative comparisons. Specifically in Fig. 9, we show reconstructions of different persons in varying clothing. As can be seen, we faithfully reproduce the input data including the pose-dependent deformations of the surface. The temporal coherence of our reconstructions can be seen in the supplemental video. In Fig. 1, we show results for novel poses using our representation based on poses from the MOSH dataset [28].

4.1. Ablation Studies

The key components of our approach are the temporally consistent tracking of the SMPL body as well as the reconstruction of the surface using a pose-dependent MLP. In the following, we will analyze the effects of our optimization scheme as well as the dynamic surface MLP.

Static vs. Dynamic Surface MLP To handle pose dependent deformation of the geometry such as wrinkles of the clothing, we use a dynamic surface function network. This network is conditioned on the pose parameters of the underlying SMPL model. In Fig. 3, we show a comparison of using a dynamic surface function network and a static one (i.e., without providing the pose conditioning to the network). As can be seen, only the pose conditioned network is able to represent the dynamically deforming surface.

Pair-wise Surface Consistency The pair-wise consistency regularizes the surface network to predict similar surfaces if the pose is similar. In Fig. 4, we show the effect of this regularizer. As can be seen, the pair-wise surface consistency loss leads to more details as well as to a 3D consistent surface reconstruction. Without the pair-wise loss, the surface tends to be closer to the SMPL surface.

4.2. Comparisons

Implicit Surface Representations Our dynamic surface function networks estimate the surface points explicitly. A classical rasterizer can be used to render these surfaces. In contrast, implicit surface representations (like an occupancy function) need to be ray-casted or converted into an explicit surface representation [29]. Since implicit functions

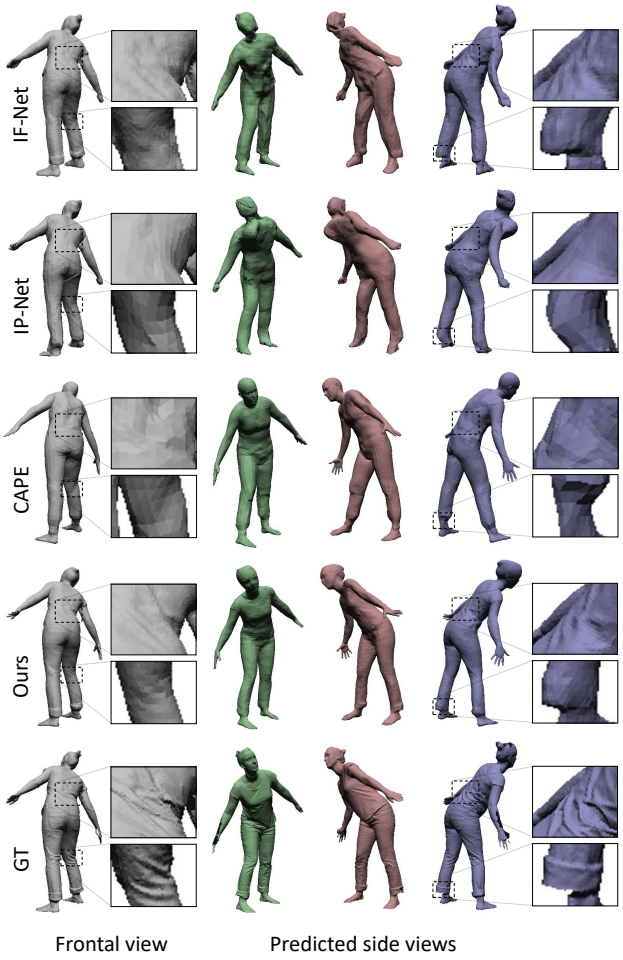

Figure 5: Our model is an explicit surface function. CAPE [31] also models an explicit surface, it predicts the offsets from SMPL with a graph based decoder. In contrast, IF-Nets [11] represent the surface implicitly using an occupancy function. IP-Net [7] is a hybrid approach that fits the SMPL model with additional vertex displacements to an estimated implicit function.

like IF-Nets [11] do not provide explicit correspondences over a time series, regularizing the surface to deform in a smooth and temporally consistent manner is non-trivial (regions can disappear and appear at a different location). In Fig. 10, we show a comparison to IF-Nets and IP-Nets on an RGB-D sequence. IP-Net [7] is a hybrid, leveraging the reconstruction abilities of IF-Nets and the controllability of the SMPL body model. Both IF-Nets and IP-Nets take a point cloud of a single frame as input and predict the implicit surface. We use the masks predicted by Graphonomy to remove the background points from the input. This input is fed into the pretrained networks for single views provided by the authors as shown in their publications. Note that the authors explicitly state that the networks generalize

Method	**IoU** ↑	**C-ℓ_2** ↓	**NC** ↑
IF-Nets [11]	0.818	1.8cm	0.903
IP-Nets [7]	0.783	2.1cm	0.861
CAPE [31]	0.648	2.5cm	0.844
Ours	**0.832**	**1.6cm**	**0.916**

Table 1: Quantitative comparisons based on a sequence of the BUFF dataset [55] (see Fig. 10). For all methods, we provide synthetically rendered monocular RGB-D data inputs. In the table, we report the numbers for IoU, chamfer distance (using ℓ_2-norm) and normal consistency w.r.t. the complete meshes from the dataset.

Method	**EPE** ↓
BodyFusion [52]	2.77cm
IP-Nets [7]	14.17cm*
CAPE [31]	5.51cm
SMPL	4.33cm
Ours	**2.63cm**

Table 2: Quantitative comparisons based on the BodyFusion dataset [52] which provides tracked VICON marker locations for each RGB-D frame. The mean end-point-error (EPE) is measured in 3D based on an ℓ_1 distance between the tracked VICON markers and the corresponding points on the reconstructed mesh. Note that both the CAPE baseline and our method use the SMPL tracking as initialization. $*$ not using temporal information.

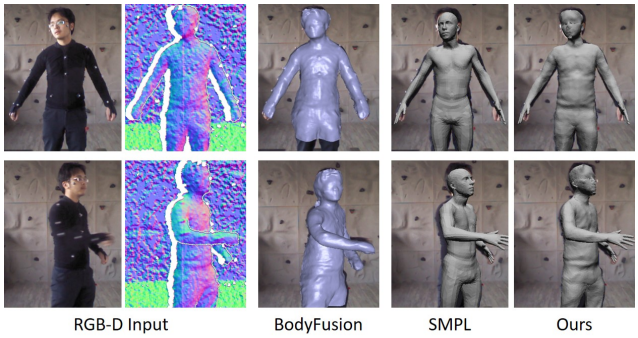

Figure 6: Qualitative comparison to BodyFusion [52] based on a sequence of their dataset. Note that this dataset has low quality inputs, but still our approach is able to integrate details over the sequence to reconstruct a high quality model.

to continuous articulations of temporal data from new data sources. In Tab. 1, we show the corresponding errors w.r.t. IoU, chamfer-distance and normal consistency. We observe that our method leads to better reconstructions, with a temporally consistent output mesh (see supplemental video).

BodyFusion [52] is a fusion method that integrates the

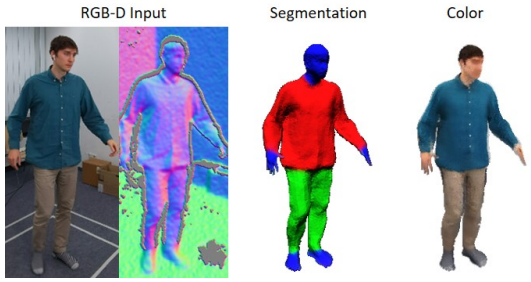

Figure 7: Our model can be extended to reconstruct the surface color as well as cloth segmentation (using Graphonomy [17] masks for training).

depth measurements into a volumetric SDF representation. It leverages the skeleton of a body as a kinematic prior, thus, outperforming general non-rigid fusion methods like DynamicFusion [35] or VolumeDeform [24] in the scenario of body reconstruction. In Tab. 2, we quantitatively compare our method to BodyFusion as well as to IP-Nets. Note that both BodyFusion as well as our method are using temporal information, while IP-Net is applied frame-by-frame, thus, leading to the highest tracking error. The tracking error is measured by a mean ℓ_1 end-point error on the dataset provided by the authors of BodyFusion [52]. Aside from the low-quality depth and color images, the dataset provides the coordinates of tracked VICON markers used for evaluation (see Fig. 6). Our method results in the lowest error and qualitatively results in sharper results.

Explicit Surface Representations In the results discussed above, we already mention IP-Nets [7] which model the surface explicitly with per-vertex displacements on top of the SMPL model, also known as SMPL-D. Our approach with an MLP *without* pose conditioning is closely related to SMPL-D (see Fig. 3). Instead of a discrete number of displacement vectors, we represent the surface using a continuous function that can be evaluated on any surface point of the SMPL model. In Tab. 1, we also include a comparison to the graph neural network-based GAN for clothing CAPE [31]. We used our optimization framework to estimate the latent codes, based on the RGB-D inputs (see supplemental material for more details). While the generative approach has several advantages for reconstructing humans from partial views or RGB images, it does not capture the detail that our reconstruction method recovers.

5. Discussion

Our Dynamic Surface Function Networks are able to reconstruct and track a variety of sequences of different people. The explicit representation of the surface allows us to use differentiable rasterization for rendering, and to generate a temporally consistent mesh as well as global correspondences, but it also has limitations. In particular, using a fixed topology our approach is not able to represent topologically changing surfaces.

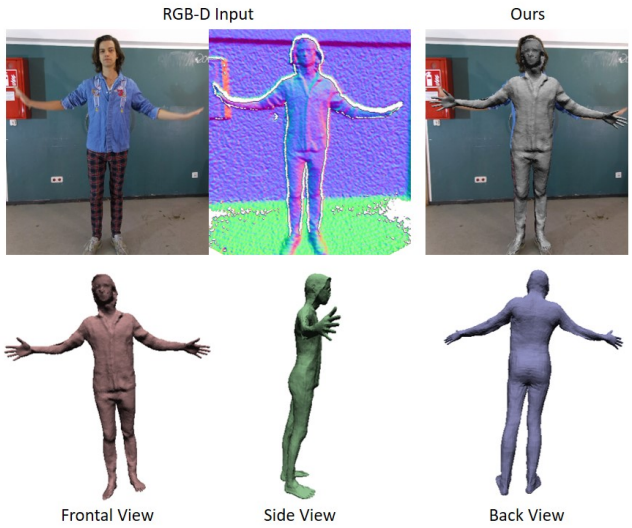

Figure 8: Our approach jointly optimizes the pose and the dynamic surface function network based on the given input data. It does not reconstruct regions that were not visible in the sequence. In the shown example the input data only contained frontal views of the person.

Our approach is an optimization approach and belongs to the class of methods that must be trained for each new input sequence [33, 47, 43]. In particular, for the reconstruction of a controllable representation this retraining of the dynamic surface function network is practical and crucial to get fine scale details such as wrinkles. If regions of the body are not visible in the input sequence (i.e., no data terms), the surface in these regions is only regularized to be smooth (see Fig. 8). While not the focus of this work, our representation can be extended to other modalities such as surface color or cloth segmentation, as can be seen in Fig. 7.

6. Conclusion

Our approach reconstructs and tracks a clothed human using a single commodity RGB-D sensor, obtaining a temporally-consistent surface reconstruction. The underlying dynamic surface function network is able to represent pose-dependent deformations of the surface and allows to re-animate the body. In our experiments, we demonstrate the effectiveness of this representation and show state-of-the-art reconstruction performance. The proposed representation offers a variety of advantages such as a consistent mesh structure, dynamically changing surface and the possibility to extend it to other outputs like color or segmentation.

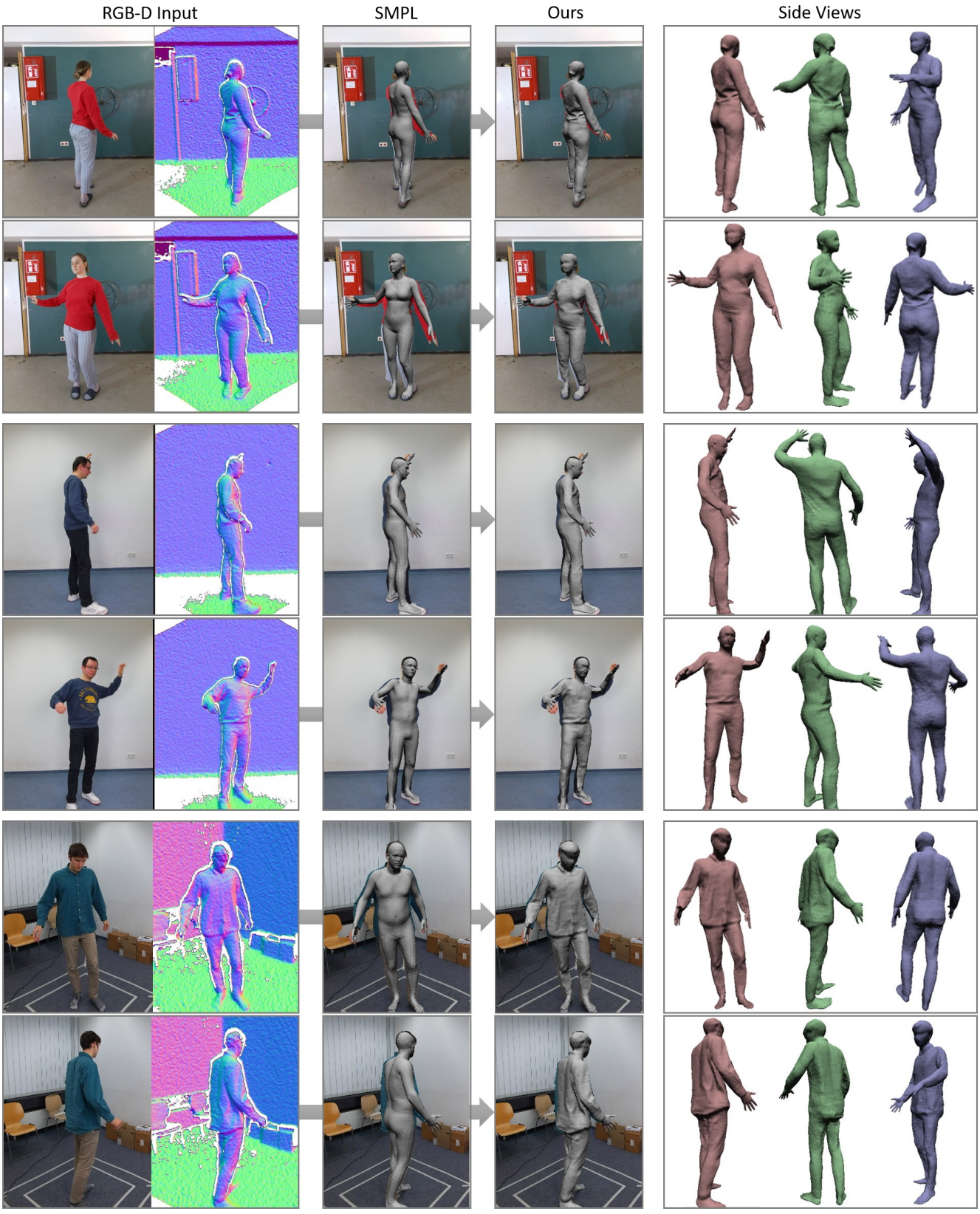

Figure 9: Temporally coherent reconstructions of clothed humans using our proposed dynamic surface function networks. The input data has been captured by a Microsoft Kinect Azure. Each sequence is 200 frames long.

Acknowledgements

The work is funded by Huawei, a TUM-IAS Rudolf Mößbauer Fellow-ship, the ERC Starting Grant Scan2CAD (804724), the German Research Foundation (DFG) Grant *Making Machine Learning on Static and Dynamic 3D Data Practical*, and the BMBF-funded Munich Center for Machine Learning.

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. 13

[2] Ijaz Akhter and Michael J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015)*, pages 1446–1455, June 2015. 3

[3] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single RGB camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2019. 2

[4] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Detailed human avatars from monocular video. In *International Conference on 3D Vision (3DV)*, sep 2018. 2

[5] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[6] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019. 2

[7] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. In *European Conference on Computer Vision (ECCV)*. Springer, August 2020. 1, 2, 6, 7

[8] Federica Bogo, Michael J. Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *International Conference on Computer Vision (ICCV)*, pages 2300–2308, Dec. 2015. 2

[9] Aljaz Bozic, Pablo Palafox, Michael Zollöfer, Justus Thies, Angela Dai, and Matthias Nießner. Neural deformation graphs for globally-consistent non-rigid reconstruction. 2021. 2

[10] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 4

[11] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 1, 2, 6

[12] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. 2

[13] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa: Neural articulated shape approximation. In *The European Conference on Computer Vision (ECCV)*, August 2020. 2

[14] Mingsong Dou, Jonathan Taylor, Henry Fuchs, Andrew Fitzgibbon, and Shahram Izadi. 3d scanning deformable objects with a single rgbd sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 493–501, 2015. 2

[15] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020. 2

[16] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T. Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[17] Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, Meng Wang, and Liang Lin. Graphonomy: Universal human parsing via graph transfer learning. In *CVPR*, 2019. 4, 7

[18] Kaiwen Guo, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu. Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera. *ACM Trans. Graph.*, 36(4), June 2017. 2

[19] Kaiwen Guo, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu. Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera. *ACM Transactions on Graphics (TOG)*, 36(3):32, 2017. 2

[20] Marc Habermann, Weipeng Xu, , Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Livecap: Real-time human performance capture from monocular video. *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, jul 2019. 2

[21] Marc Habermann, Weipeng Xu, , Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 2

[22] Thomas Helten, Andreas Baak, Gaurav Bharaj, Meinard Müller, Hans-Peter Seidel, and Christian Theobalt. Personalization and Evaluation of a Real-time Depth-based Full Body Tracker. In *Proceedings of the Joint 3DIM/3DPVT Conference*, pages 279–286, Seatle, Canada, 2013. 2

[23] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2

[24] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. Volumedeform:

Real-time volumetric non-rigid reconstruction. In *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, pages 362–379, Cham, 2016. Springer International Publishing. 2, 7

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5, 12

[26] 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. 5, 12

[27] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 1, 2, 3, 4

[28] Matthew M. Loper, Naureen Mahmood, and Michael J. Black. MoSh: Motion and shape capture from sparse markers. 33(6):220:1–220:13, Nov. 2014. 1, 3, 5

[29] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, page 163–169, New York, NY, USA, 1987. Association for Computing Machinery. 5

[30] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael Black. Learning to dress 3d people in generative clothing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 2

[31] 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. 6, 7, 13

[32] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, Oct. 2019. 1

[33] 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. 3, 7, 12

[34] A. Neophytou and A. Hilton. A layered model of human body and garment deformation. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 171–178, 2014. 2

[35] Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2, 7

[36] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *International Conference on Computer Vision*, Oct. 2019. 2

[37] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[38] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael Black. ClothCap: Seamless 4D clothing capture and retargeting. *ACM Transactions on Graphics (SIGGRAPH)*, 36(4), 2017. 2

[39] Gerard Pons-Moll, Jonathan Taylor, Jamie Shotton, Aaron Hertzmann, and Andrew Fitzgibbon. Metric regression forests for correspondence estimation. *International Journal of Computer Vision*, pages 1–13, 2015. 2

[40] Iasonas Kokkinos Riza Alp Güler, Natalia Neverova. Densepose: Dense human pose estimation in the wild. 2018. 4, 13

[41] Shunsuke Saito, , Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. *arXiv preprint arXiv:1905.05172*, 2019. 2

[42] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2020. 2

[43] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *arXiv*, 2020. 7

[44] Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1386–1395, 2017. 2

[45] Miroslava Slavcheva, Maximilian Baust, and Slobodan Ilic. Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2646–2655, 2018. 2

[46] Carsten Stoll, Juergen Gall, Edilson de Aguiar, Sebastian Thrun, and Christian Theobalt. Video-based reconstruction of animatable human characters. In *ACM SIGGRAPH Asia 2010 Papers*, SIGGRAPH ASIA '10, New York, NY, USA, 2010. Association for Computing Machinery. 2

[47] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics 2019 (TOG)*, 2019. 7

[48] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. Patchnets: Patch-based generalizable deep implicit 3d shape representations. In *European Conference on Computer Vision*, pages 293–309. Springer, 2020. 2

[49] A. Weiss, D. Hirshberg, and M.J. Black. Home 3D body scans from noisy image and range data. In *Int. Conf. on Computer Vision (ICCV)*, pages 1951–1958, Barcelona, Nov. 2011. IEEE. 2

[50] Stefanie Wuhrer, Leonid Pishchulin, Alan Brunton, Chang Shu, and Jochen Lang. Estimation of human body shape and posture under clothing. *CoRR*, abs/1312.4967, 2013. 2

[51] Donglai Xiang, Fabian Prada, Chenglei Wu, and Jessica Hodgins. Monoclothcap: Towards temporally coherent clothing capture from monocular rgb video, 2020. 2

[52] Tao Yu, Kaiwen Guo, Feng Xu, Yuan Dong, Zhaoqi Su, Jianhui Zhao, Jianguo Li, Qionghai Dai, and Yebin Liu. Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. In *The IEEE International Conference on Computer Vision (ICCV)*. IEEE, October 2017. 2, 6, 7

[53] 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

[54] Chao Zhang, Sergi Pujades, Michael Black, and Gerard Pons-Moll. Detailed, accurate, human shape estimation from clothed 3D scan sequences. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[55] 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)*, July 2017. 5, 6

[56] Qing Zhang, Bo Fu, Mao Ye, and Ruigang Yang. Quality dynamic human body modeling using a single low-cost depth camera. pages 676–683, 06 2014. 2

[57] Michael Zollhöfer, Justus Thies, Darek Bradley, Pablo Garrido, Thabo Beeler, Patrick Péerez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. 2018. 2

A. Implementation Details

A.1. Network Architecture

The dynamic surface function is represented as a multi-layer 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. [33], 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' $\mathcal{F} = [\mathbf{F}_1, \ldots, \mathbf{F}_{23}] \in \mathbb{R}^{23\times 9}$, 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.

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. At first, we warp the SMPL template to match the observations sequentially by reconstructing intrinsic SMPL parameters. We apply the L-BFGS [26] optimizer with the strong Wolfe search, history size of 20 and 20 maximum iterations. We observe that using the Adam [25] optimizer at this stage is not efficient, since it struggles to reconstruct rotations in the axis-angle form. The optimization is executed globally 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 of the model parameters $\mathcal{P}$ jointly. 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.

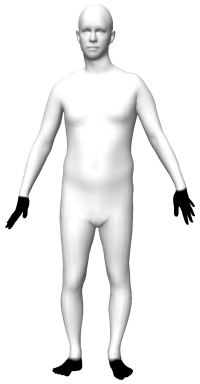

Figure 10: We exclude the hands and feet from optimization.

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 3: Energy term weights during optimization.

Loss weights As described in the main paper, our optimization is based on a set of different energy terms. In Tab. 3, 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 roughly aligned with the body in the image). 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 $(45°)$.

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

A.3. Surface Sampling

For rendering, we need to sample the surface. We use the original 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 [40], lead to additional samples on the surface.

A.4. Baseline Implementation

In the main paper, we discuss results based on the CAPE cloth model [31]. 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.