Supplementary Material
NSF: Neural Surface Fields for Human Modeling from Monocular Depth

Abstract

In this supplementary document, we provide further details about the network, training procedure, and method. We also report further results on per-outfit reconstruction, unseen subject adaptation, multi-resolution, and real data. Finally, we report timings for the method inference.

1. NSF: Implementation Details

1.1. Network Architecture

Implicit Fusion Shape For learning the implicit fusion shape f^{shape}, we follow the Auto-Decoder network architecture proposed in IGR [13], as described in sections A.1. We depict it in Figure 1.

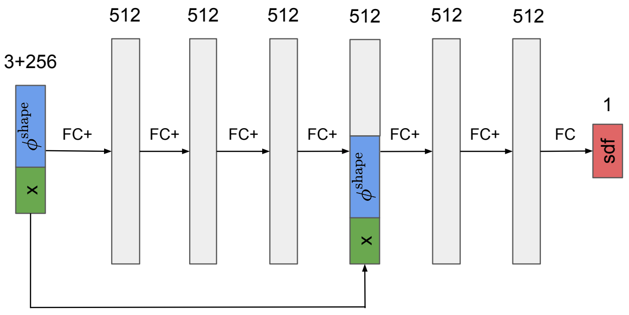


Figure 1: Auto-Decoder architecture proposed in IGR [13].

FC is a fully connected linear layer and FC+ is the FC followed by a softplus activation. We also use a skip connection from input to 4th layer. ϕ_{shape} is the subject-specific latent vector. This network predicts the 1-dimensional SDF value of given location $\mathbf{x}$ w.r.t. fusion shape of the specific subject.

Global Pose Feature Given the pose parameter θ in $SO(3)$, we use an MLP to extract the global pose features. Here, FC represents a fully connected layer, and FC+ has an additional ReLU() layer and dropout layer with $p = 0.3$ after each FC layer. This network extracts 24-dimensional global pose feature from 72-dimensional pose parameters θ. We visualize this in Figure 2.

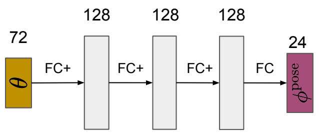


Figure 2: MLP to extract the global pose features.

Neural Surface Field Our proposed neural surface field predicts the continuous pose-dependent displacement field based on the subject-specific fusion shape with the Auto-Decoder network reported in Figure 3.

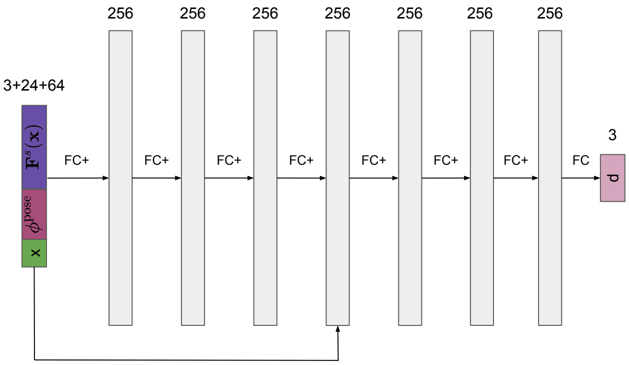


Figure 3: Auto-Decoder network for Neural Surface Field.

FC indicates fully connected layer. FC+ contains a FC layer, a batch normalization layer, and a softplus activation layer. $\mathbf{F}^s\left(\mathbf{x}^c\right)$ indicates the local feature at $\mathbf{x}$ on the feature surface, which has the dimension of 64.

1.2. Training Details

The whole model is trained on a single Nvidia 2080-Ti GPU with PyTorch deep learning framework [26]. We use two separate ADAM optimizers [15] for network parameters and the feature space. For both optimizers we use initial learning rate of $1e-4$ scheduled to decrease by a factor of 2 every 500 epochs. At the fine-tuning stage, we freeze network parameters and only optimize the feature space.

1.3. Evaluation Metrics

Chamfer Distance (CD) We report the bi-directional square-root Chamfer distance (CD), which measures L2 distances between the reconstructed surface to the ground-truth surface (lower is better). Given the reconstructed shape and the ground-truth shape, we sample with $200,000$ points $\mathbf{X}$ and $\mathbf{Y}$ from them. We calculate the root mean square bi-directional Chamfer distance with:

$$\left(\frac{1}{2M}\sum_{i=1}^{M}\min_{j}\|\mathbf{x}_i-\mathbf{y}_j\|_2^2+\frac{1}{2N}\sum_{j=1}^{N}\min_{i}\|\mathbf{x}_i-\mathbf{y}_j\|_2^2\right)^{\frac{1}{2}}, \tag{1}$$

where M, N are the number of points from the sampled point cloud $\mathbf{X}$ and $\mathbf{Y}$, respectively. The unit of reported result in the paper is centimeter (cm).

Normal Consistency (NC) We also report the normal consistency (NC), which measures the accuracy and the completeness of reconstructed shape normals (higher is better). Following the notion in Eq. 1, we compute the bi-directional normal consistency as

$$\frac{1}{2M}\sum_{i=1}^{M}\frac{\boldsymbol{n}\left(\mathbf{x}_{\boldsymbol{i}}\right)}{\left\|\boldsymbol{n}\left(\mathbf{x}_{\boldsymbol{i}}\right)\right\|}\frac{\boldsymbol{n}(\hat{\mathbf{y}})}{\|\boldsymbol{n}(\hat{\mathbf{y}})\|}+\frac{1}{2N}\sum_{j=1}^{N}\frac{\boldsymbol{n}(\hat{\mathbf{x}})}{\|\boldsymbol{n}(\hat{\mathbf{x}})\|}\frac{\boldsymbol{n}\left(\mathbf{y}_{\boldsymbol{j}}\right)}{\left\|\boldsymbol{n}\left(\mathbf{y}_{\boldsymbol{j}}\right)\right\|}, \tag{2}$$

where

$$\begin{aligned}\hat{\mathbf{y}}&=\arg\min_{\mathbf{y}_j\in\mathbf{Y}} d(\mathbf{x}_i,\mathbf{y}_j),\\ \hat{\mathbf{x}}&=\arg\min_{\mathbf{x}_i\in\mathbf{X}} d(\mathbf{x}_i,\mathbf{y}_j),\end{aligned} \tag{3}$$

$n(\cdot)$ denotes the normal of the given point and $d(\mathbf{x},\mathbf{y})$ the L_2 distance between point $\mathbf{x}$ and $\mathbf{y}$.

Intersection Over Union (IoU) Volumetric intersection over union (IoU) measures how well the the reconstructed mesh matches the ground-truth mesh (higher is better). Following IF-Nets [7], we perform the implicit waterproofing of ground-truth scan. We sample $200,000$ points $\mathbf{X}$ in the bounding box of the ground-truth scan and query the occupancy status of each sampled points in reconstructed shape and the ground-truth shape. We compute the IoU with:

$$\frac{\text{occ}_{pred}(\mathbf{X})\cap\text{occ}_{gt}(\mathbf{X})}{\text{occ}_{pred}(\mathbf{X})\cup\text{occ}_{gt}(\mathbf{X})}, \tag{4}$$

where occ_{pred} is the occupancy query function w.r.t. predicted shape and occ_{gt} is the occupancy query function w.r.t. ground-truth shape.

1.4. Baselines

Our baselines are PINA [10] and DSFN [4], which both model clothed humans from monocular depth input. Although the authors promised the code in their work, it is not yet available at the time of writing. Hence, we perform a quantitative comparison using results from respective original papers and a qualitative comparison using meshes provided by them. Moreover, we apply two naive baselines, namely SMPL [17] and our fusion shape. The fusion shape can be understood as SMPL+D. It shows how our proposed NSF learns the pose-dependent deformation.

1.5. SMPL Fitting

For the shapes without given SMPL registration (e.g. BuFF [33] for synthetic data as well as DSFN [4] for real data), we use a concurrent work based on LVD [8] to fit the SMPL [17] model to the input point cloud. Moreover, we use off-the-shelf methods [11, 31, 32] to estimate the SMPL [17] parameters from RGB images. Since the estimation from RGB images is always inaccurate in terms of root rotation and leg poses, we use the RVH registration library [1, 2, 3] to manually refine the SMPL fit with the input point cloud. Finally, we choose the visually plausible SMPL fit given the input point cloud.

2. NSF: Method Details

2.1. Extract Fusion Shape

Here we provide more detail in the learning of the canonical fusion shape. For the outfit which can be described by SMPL topology, we additionally fit SMPL+D using our learned implicit fusion shape surface with:

$$\min_{D} f^{\text{shape}}(\mathbf{X}(\text{SMPL+D})|\phi^{\text{shape}}), \tag{5}$$

where $\mathbf{X}$(SMPL+D) denotes the sampled points from the SMPL+D shape, f^{shape} together with ϕ^{shape} indicates the implicit fusion shape of the subject with outfit. We push the sampled points lie on the zero-level set to obtain the displacement of the SMPL surface. To ensure the smoothness of the SMPL+D surface, we additionally minimize the Laplacian of the optimized mesh [25].

Our learned fusion shape defines a coherent surface for our proposed Neural Surface Field (NSF). We can fuse the depth observation and texture from multiple partial shapes onto this unifying surface. This enables us to learn the deformation and the texture in the NSF.

2.2. Texture Transfer on Fusion Shape

To transfer the texture color from the posed partial shapes to the canonical fusion one, first We randomly select 10 partial shapes which could cover the most area of the fusion shape. Then, we canonicalize (Eq.1 in main paper) and project (Eq.2 in main paper) them to find the canonical correspondence on the fusion shape, as shown in Figure 5. Finally, we assign the color of each vertex on the fusion shape from the nearest canonical points. We found that

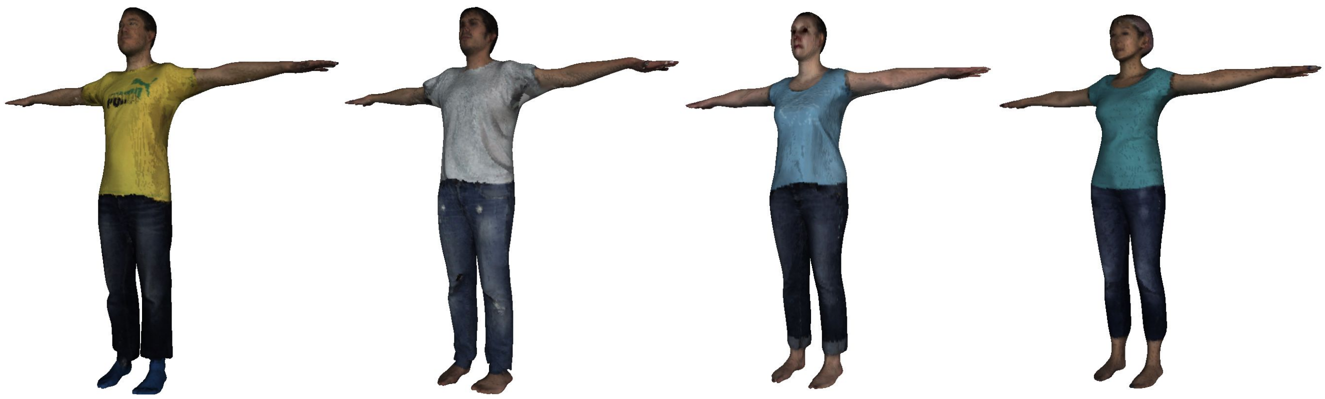


Figure 4: Texture transfer from colorful partial point cloud onto fusion shapes of multiple subjects in BuFF [27, 33] dataset. Fusion shapes below are extracted using Marching Cube [18] with a resolution of 512, producing a mesh of around 90k vertices for each shape. We can appreciate the preservation of fine details (e.g., the logo on the yellow t-shirt).

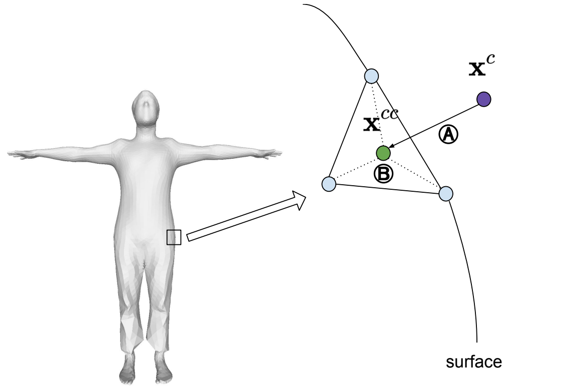


Figure 5: For a given pose-dependent canonical point $\mathbf{x}^c$ in $\mathbb{R}^3$, we project it onto the fusion shape surface using the gradient (A) and query the feature $\mathbf{F}^s(\mathbf{x}^{cc})$ with barycentric interpolation of basis features of nearest neighbors (B).

our texture transferring method produces a sharper texture compared to the vertex-color optimization using all available partial shapes. We show the textured fusion shape of subjects in BuFF [27, 33] dataset in Figure 4.

2.3. Surface-based Feature Query

We use the vertices of our learned fusion shape to form the basis of the surface-based feature space. As introduced in Eq. 6 in the main paper, we can lift the neural surface feature from $\mathbf{x}^{cc}$, $\mathbf{F}^s(\mathbf{x}^c) \leftarrow \mathbf{F}^s(\mathbf{x}^{cc})$ with the surface projection. To obtain the $\mathbf{F}^s(\mathbf{x}^{cc})$, we use Kaolin [12] to estimate the barycentric coordinate w.r.t. three vertices of the corresponding triangle and interpolate features of three vertex features using the barycentric weights.

3. Further Results

3.1. Per-Outfit Reconstruction Result

To prove the effectiveness of having a single general decoder for all the outfits, we also train outfit-specific ones and compare ourselves. We report the per-outfit reconstruction quantitative results of our method in Table 1. Our generalizable decoder achieves better reconstruction quality on every single outfit, and so also on avarage. We remark that our decoder uses $\frac{1}{9}$ of the parameters compared to the per-outfit training strategy.

3.2. Adapt to Unseen Subject

Here we study the method performance using a varying number of frames during the adaptation of surface-based features of unseen subjects. It shows that our method can reconstruct the given partial shape well using the only 10 frames. Besides, the more frames used to fine-tune, the more the model generalizes to the unseen poses.

Operation	BuFF Data [33]					
	Seen Frames			Unseen Poses		
	CD ↓	NC ↑	IoU ↑	CD ↓	NC ↑	IoU ↑
10 Frames	**0.75**	**0.920**	**0.894**	1.16	0.876	0.835
20 Frames	0.75	0.919	0.894	1.05	0.899	0.857
50 Frames	0.77	0.918	0.890	**0.93**	**0.915**	**0.876**

Table 2: Our model can be quickly adapted to unseen subjects by only fine-tuning the surface-based feature space with limited number of partial shapes. We use different splits from the one used in the main, thus the quantitative number here varies from Table 3 in main paper.

3.3. Multi-resolution Flexibility

Our proposed approach has high flexibility to generate desired mesh resolution and topology. We represent our fu-

Decoder Type	# Decoder	Subject	Garment	CAPE Data [21, 27, 33]					
				CD (cm) ↓	NC ↑	IoU ↑	CD (cm) ↓	NC ↑	IoU ↑
				Mean Value			Median Value		
Outfit-specific	1	00122	shortlong	0.58	0.941	0.919	0.52	0.946	0.928
Generalizable	/	00122	shortlong	**0.52**	**0.952**	**0.935**	**0.46**	**0.957**	**0.946**
Outfit-specific	1	00122	shortshort	0.65	0.922	0.906	0.66	0.922	0.904
Generalizable	/	00122	shortshort	**0.62**	**0.933**	**0.915**	**0.63**	**0.933**	**0.911**
Outfit-specific	1	00215	jerseyshort	0.49	0.949	0.934	0.49	0.948	0.933
Generalizable	/	00215	jerseyshort	**0.46**	**0.956**	**0.944**	**0.47**	**0.955**	**0.943**
Outfit-specific	1	00215	poloshort	0.70	0.935	0.904	0.65	0.938	0.911
Generalizable	/	00215	poloshort	**0.62**	**0.945**	**0.920**	**0.56**	**0.948**	**0.929**
Outfit-specific	1	00215	shortlong	0.63	0.927	0.910	0.60	0.929	0.914
Generalizable	/	00215	shortlong	**0.58**	**0.941**	**0.925**	**0.57**	**0.943**	**0.924**
Outfit-specific	1	03375	blazerlong	0.83	0.897	0.882	0.79	0.896	0.883
Generalizable	/	03375	blazerlong	**0.77**	**0.926**	**0.894**	**0.73**	**0.928**	**0.897**
Outfit-specific	1	03375	longlong	0.86	0.899	0.872	0.76	0.906	0.886
Generalizable	/	03375	longlong	**0.79**	**0.931**	**0.890**	**0.71**	**0.937**	**0.904**
Outfit-specific	1	03375	shortlong	0.81	0.919	0.878	0.76	0.919	0.883
Generalizable	/	03375	shortlong	**0.72**	**0.939**	**0.898**	**0.66**	**0.938**	**0.905**
Outfit-specific	1	03375	shortshort	0.86	0.907	0.871	0.83	0.906	0.873
Generalizable	/	03375	shortshort	**0.79**	**0.933**	**0.887**	**0.77**	**0.933**	**0.886**
Ours, Outfit-specific	9	3 subjects	9 outfits	0.71	0.922	0.897	0.67	0.912	0.902
Ours generalizable	1	3 subjects	9 outfits	**0.65**	**0.940**	**0.912**	**0.62**	**0.941**	**0.916**

Table 1: We evaluate our method on the task of reconstructing 3D shapes from monocular depth point clouds on CAPE [21, 27, 33] data. We compare the outfit-specific decoders against our general one for all outfits. We appreciate the better performance of our choice in all the settings and metrics.

sion shape as an implicit neural signed distance field (SDF), which let us extract an arbitrary resolution and mesh for the fusion shape. For instance, we can obtain fusion shape in SMPL [17] topology using Eq. 5. Also, we can instead apply Marching cube [18] and decide the resolution level. Our texture transfer approach reported in Sec. 2.2 is also flexible to different fusion shapes; we report in Figure 6 textured fusion shape discretized at different resolutions.

Our proposed Neural Surface Field formulates a continuous pose-dependent displacement field. Thus, our NSF maintains flexibility and can generate a posed mesh with arbitrary resolution and topology in reconstruction and animation tasks. In Figure 8, we show the different reconstructed mesh of a partial shape using our NSF.

3.4. Real Data Experiments

We show more qualitative reconstruction result of real data in DSFN [4] dataset in figure 9.

3.5. Inference Time Per Frame

We use the edge connectivity of our extracted fusion shape on the deformed vertices with NSF $\mathbf{V}^p$ (see sec. 3.5 in the main paper) to generate the posed mesh. Our formulation is much more efficient compared to other works since we get rid of per-frame Marching cube [18] as in [5, 6, 9, 23, 24, 29, 30] or per-frame Poisson reconstruction [14] as in [16, 19, 20, 22]. X-Avatar [28], a concurrent work based on SNARF [6], reports 7 seconds per frame as inference time which include remeshing via Marching cube on Nvidia RTX6000. In our proposed method, we report an average time of 0.039 seconds per frame, including deformation prediction, Linear Blend Skinning, and remeshing on RTX3080-Ti. which is $\sim$ 180 times faster than X-Avatar [28]. Furthermore, we compare the inference time with point-based works POP [22], which requires Poisson reconstruction to obtain mesh from point cloud for each frame. In the same setting, POP requires about 1.6 seconds to reconstruct the surface at each frame. Our approach is $\sim$ 40 times faster.

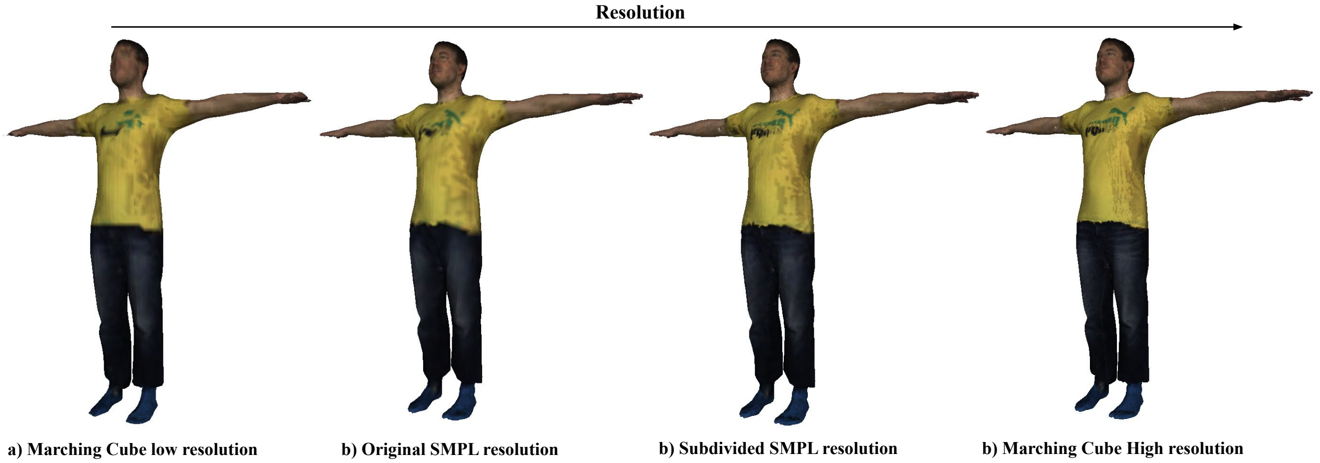

Resolution
a) Marching Cube low resolution
b) Original SMPL resolution
b) Subdivided SMPL resolution
b) Marching Cube High resolution

Figure 6: We show the textured fusion shape with different resolution and topology: (a) Marching cube with resolution of 128, $\sim$ 5k vertices; (b) SMPL+D topology, $\sim$ 7k vertices; (c) subdivided SMPL+D topology, $\sim$ 27k vertices; (d) Marching cube with resolution of 512, $\sim$ 86k vertices.

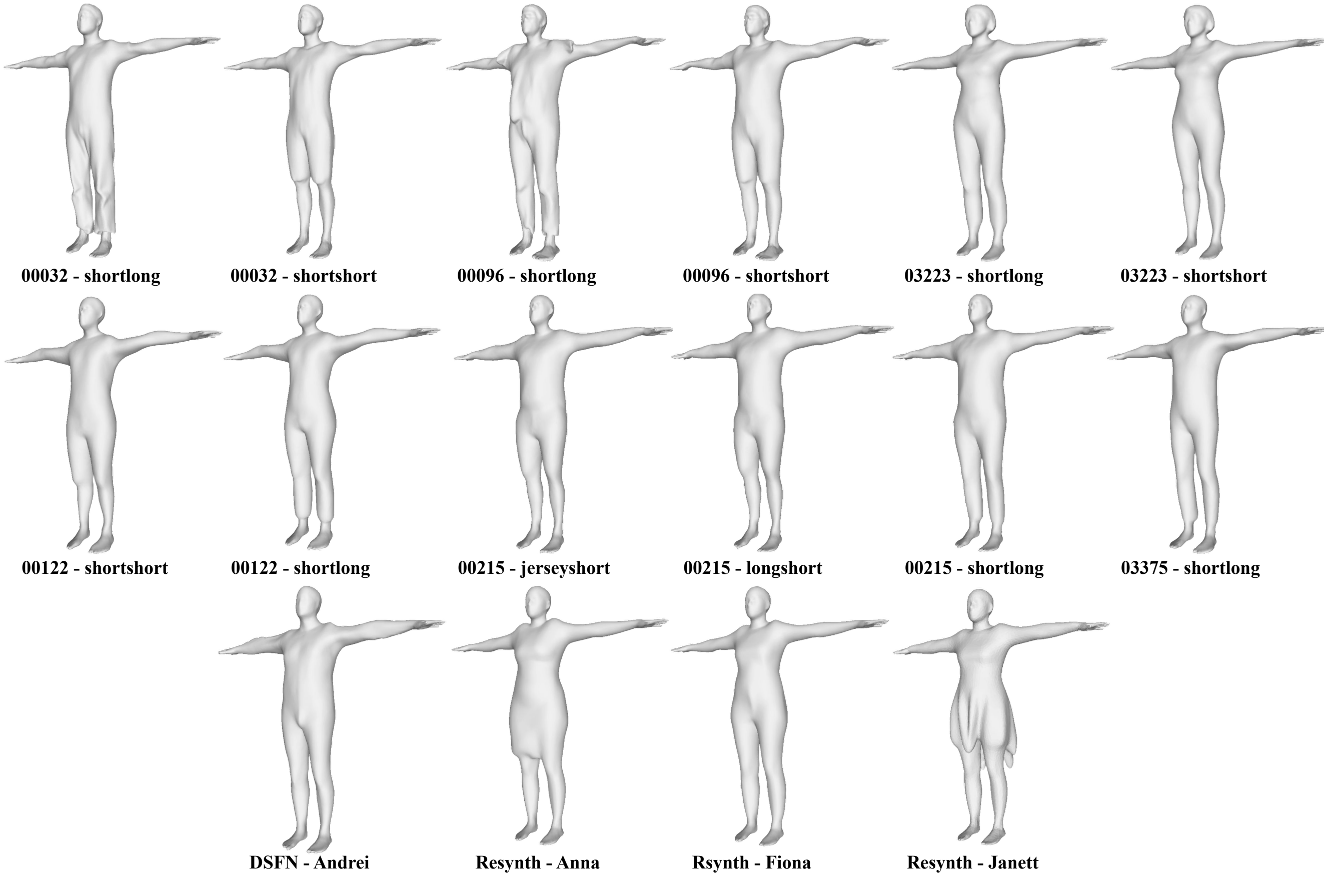
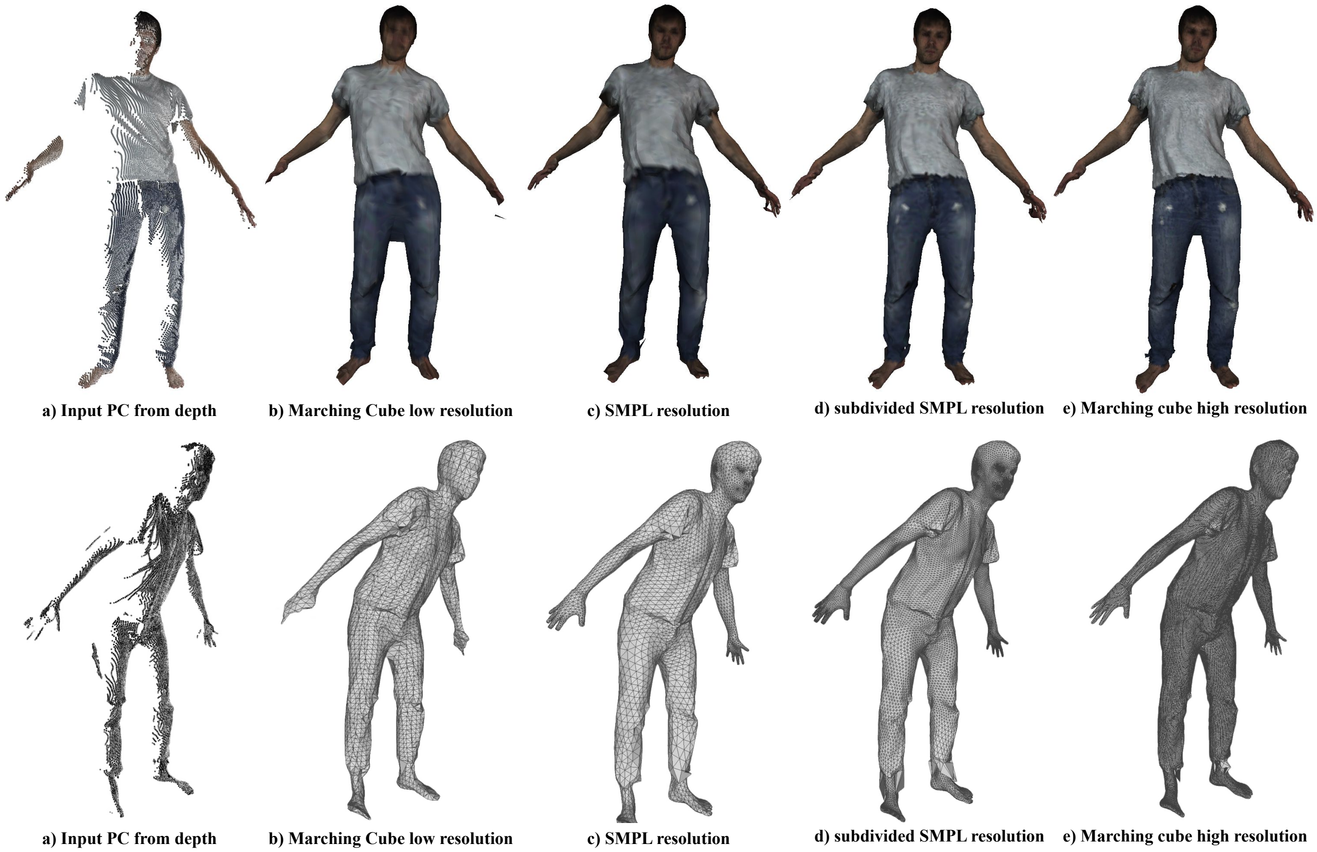


Figure 7: Learned fusion shapes of subjects from BuFF [33, 27] (1^{st} row), CAPE [21, 27] (2^{nd} row), DSFN [4] (left on 3^{rd} row), Resynth [22, 20] (3 right on 3^{rd} row).

References

[1] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human recon-



a) Input PC from depth b) Marching Cube low resolution c) SMPL resolution d) subdivided SMPL resolution e) Marching cube high resolution

a) Input PC from depth b) Marching Cube low resolution c) SMPL resolution d) subdivided SMPL resolution e) Marching cube high resolution

Figure 8: Given the partial shape (a), we show the reconstruction with different resolution and topology with our NSF: (b) Marching cube with resolution of 128, $\sim$ 5k vertices; (c) SMPL+D topology, $\sim$ 7k vertices; (d) subdivided SMPL+D topology, $\sim$ 27k vertices; (e) Marching cube with resolution of 512, $\sim$ 88k vertices.

struction. In *European Conference on Computer Vision (ECCV)*. Springer, aug 2020. 2

[2] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. In *Advances in Neural Information Processing Systems (NeurIPS)*, December 2020. 2

[3] Bharat Lal Bhatnagar, Xianghui Xie, and Ilya Petrov. Rvh mesh registration repository. `https://github.com/bharat-b7/RVH_Mesh_Registration`, 2022. 2

[4] Andrei Burov, Matthias Nießner, and Justus Thies. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction, 2021. 2, 4, 5, 7

[5] Xu Chen, Tianjian Jiang, Jie Song, Jinlong Yang, Michael J Black, Andreas Geiger, and Otmar Hilliges. gdna: Towards generative detailed neural avatars. In *CVPR*, 2022. 4

[6] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *International Conference on Computer Vision (ICCV)*, 2021. 4

[7] 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. 2

[8] Enric Corona, Gerard Pons-Moll, Guillem Alenyà, and Francesc Moreno-Noguer. Learned vertex descent: A new direction for 3d human model fitting. In *ECCV*, 2022. 2

[9] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Neural articulated shape approximation. In *The European Conference on Computer Vision (ECCV)*. Springer, August 2020. 4

[10] Zijian Dong, Chen Guo, Jie Song, Xu Chen, Andreas Geiger, and Otmar Hilliges. Pina: Learning a personalized implicit neural avatar from a single rgb-d video sequence. *arXiv*, 2022. 2

[11] Yao Feng, Vasileios Choutas, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. Collaborative regression of expressive bodies using moderation. In *International Conference on 3D Vision (3DV)*, 2021. 2

[12] Clement Fuji Tsang, Maria Shugrina, Jean Francois Lafleche, Towaki Takikawa, Jiehan Wang, Charles Loop, Wenzheng Chen, Krishna Murthy Jatavallabhula, Edward Smith, Artem Rozantsev, Or Perel, Tianchang Shen, Jun Gao, Sanja Fidler, Gavriel State, Jason Gorski, Tommy Xi-

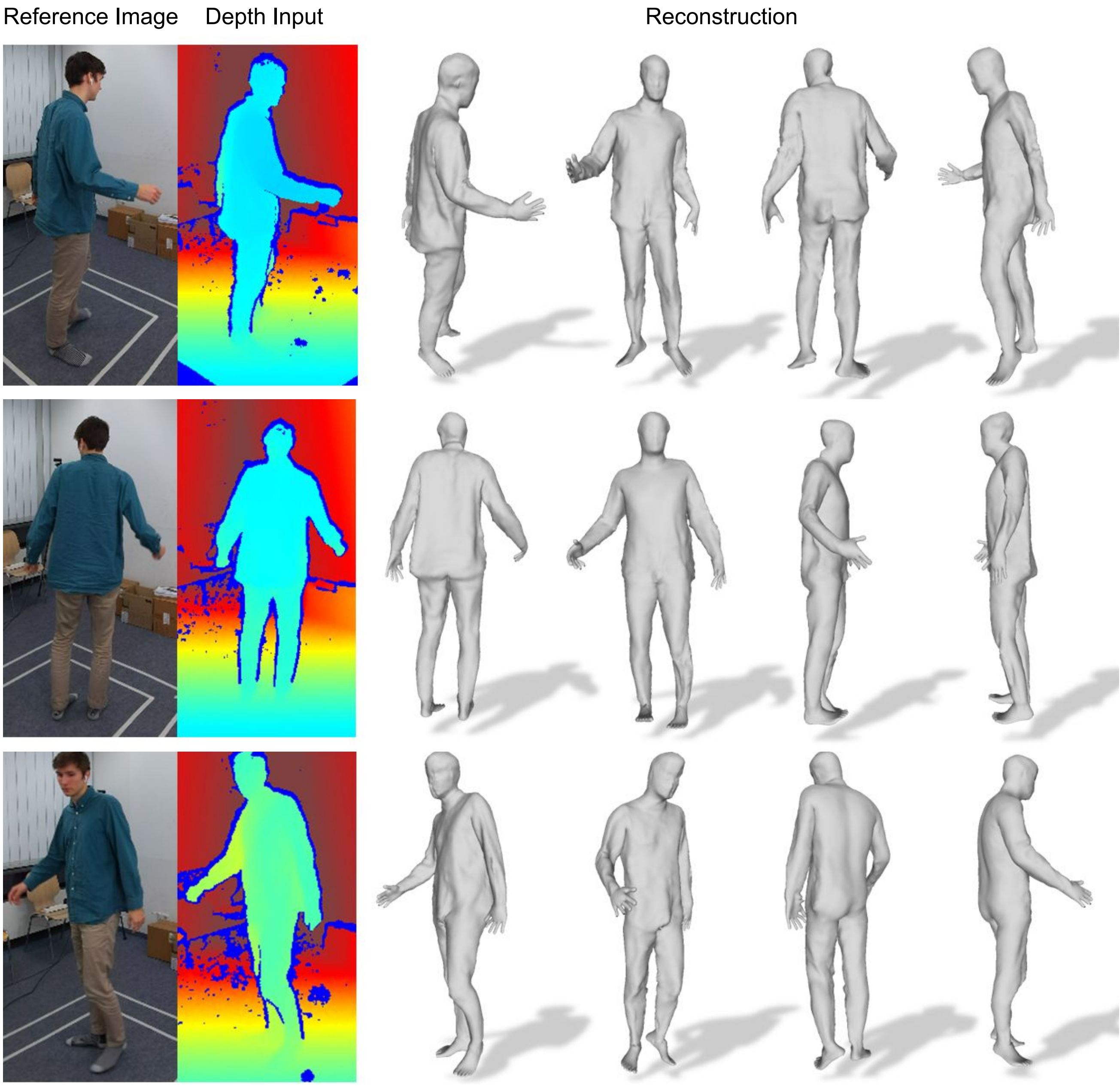


Figure 9: Our reconstruction result on Real Data from DSFN [4] dataset. Note that our method only takes noisy depth images as input. Our approach shows robustness while still capturing finer details of the input (e.g., wrinkles)

ang, Jianing Li, Michael Li, and Rev Lebaredian. Kaolin: A pytorch library for accelerating 3d deep learning research. https://github.com/NVIDIAGameWorks/kaolin, 2022. 3

[13] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *ICML*, 2020. 1

[14] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 4

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 1

[16] Siyou Lin, Hongwen Zhang, Zerong Zheng, Ruizhi Shao, and Yebin Liu. Learning implicit templates for point-based clothed human modeling. In *ECCV*, 2022. 4

[17] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 2, 4

[18] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH*, pages 163–169. ACM, 1987. 3, 4

[19] Qianli Ma, Shunsuke Saito, Jinlong Yang, Siyu Tang, and Michael J Black. Scale: Modeling clothed humans with a surface codec of articulated local elements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16082–16093, 2021. 4

[20] Qianli Ma, Jinlong Yang, Michael J. Black, and Siyu Tang. Neural point-based shape modeling of humans in challenging clothing. In *2022 International Conference on 3D Vision (3DV)*, September 2022. 4, 5

[21] 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)*, June 2020. 4, 5

[22] Qianli Ma, Jinlong Yang, Siyu Tang, and Michael J. Black. The power of points for modeling humans in clothing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021. 4, 5

[23] Marko Mihajlovic, Shunsuke Saito, Aayush Bansal, Michael Zollhoefer, and Siyu Tang. COAP: Compositional articulated occupancy of people. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 4

[24] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. LEAP: Learning articulated occupancy of people. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 4

[25] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian mesh optimization. In *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, GRAPHITE '06, page 381–389, New York, NY, USA, 2006. Association for Computing Machinery. 2

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 1

[27] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael Black. Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, 36(4), 2017. Two first authors contributed equally. 3, 4, 5

[28] Kaiyue Shen, Chen Guo, Manuel Kaufmann, Juan Zarate, Julien Valentin, Jie Song, and Otmar Hilliges. X-avatar: Expressive human avatars. *Computer Vision and Pattern Recognition (CVPR)*, 2023. 4

[29] Garvita Tiwari, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Neural-gif: Neural generalized implicit functions for animating people in clothing. In *International Conference on Computer Vision (ICCV)*, October 2021. 4

[30] Shaofei Wang, Marko Mihajlovic, Qianli Ma, Andreas Geiger, and Siyu Tang. Metaavatar: Learning animatable clothed human models from few depth images. In *Advances in Neural Information Processing Systems*, 2021. 4

[31] Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J. Black. ECON: Explicit Clothed humans Obtained from Normals. *arXiv:2212.07422*, 2022. 2

[32] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J. Black. ICON: Implicit Clothed humans Obtained from Normals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13296–13306, June 2022. 2

[33] 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. 2, 3, 4, 5