2S-UDF: A Novel Two-stage UDF Learning Method for Robust Non-watertight Model Reconstruction from Multi-view Images Supplementary Material

Junkai Deng[1,2] **Fei Hou**[1,2*] **Xuhui Chen**[1,2] **Wencheng Wang**[1,2] **Ying He**[3]
[1]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]School of Computer Science and Engineering, Nanyang Technological University
{dengjk, houfei, chenxh, whn}@ios.ac.cn yhe@ntu.edu.sg

A. Proof of Theorem 1

Proof: We consider the scenario where the ray only intersects with the plane once. The weight w_2 is the derivative of the logistic sigmoid function scaled by a constant factor $|\cos(\theta)|$, which is a bell-shaped function centered at $f(t^*) = 0$ implying that the point t^* is on the surface. Put it in another way, the color weight w_2 attains maximum value when the point is on the surface, therefore w_2 is unbiased. With the ray truncation mechanism, if there are multiple ray-plane intersections along a single ray, only the first intersection is in effect. Therefore, it is also occlusion-aware.

B. Bias and Translucency Analysis of Stage 1

The density function σ_1 in our Stage 1 coarse training is neither unbiased nor fully opaque, but we select $c = 5$ for a good balance. In fact, we can estimate the bias and translucency. For points in front of the surface, the incident angle θ between the ray and the surface normal is obtuse, so we restrict θ to the range of $[91°, 180°]$. Assuming $s = 1000$, by setting $c = 5$, in theory, the offset width between 0.00161 and 0.00566 is obtained relative to the true zero level set, indicating that the maximum relative bias is below 0.5%. This error level is acceptable for most application scenarios. Moreover, assuming $s = 1000$, the surface transparency in the extreme case mentioned above is less than 0.001. When a ray has a larger incident angle, its transparency becomes even smaller, resulting in an almost opaque density σ_1. As a result, the weight function w_1 is approximately occlusion-aware. Thus, setting the constant $c = 5$ offers a good balance between occlusion-awareness and unbiasedness in the first stage training.

*Corresponding author

C. Discussion of MeshUDF

Unlike signed distance fields (SDFs), from which extracting a mesh is extensively studied, extracting a mesh from unsigned distance fields is still an actively developing research field with several challenges, which can lead to sub-optimal reconstruction results. MeshUDF [3] is a UDF-mesh extraction method that has enjoyed considerable popularity, yet it still contains some limitations. Figure 9 showcases two common limitations of MeshUDF: the extracted mesh exhibits a visible "staircase effect" and hole artifacts resulting in a negative visual impact. "Staircase effect" and holes are pervasive across the results of NeuralUDF [8], NeUDF [7] and our method. To eliminate these artifacts, we can use DoubleCoverUDF [4] for mesh extraction from UDF in the future, but we use MeshUDF in this work for fair comparisons.

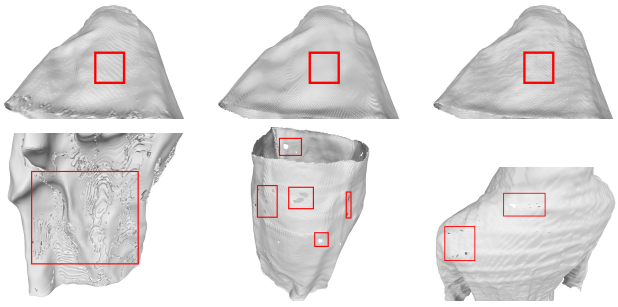

Figure 9. The "staircase effect" and hole artifacts found in extracted meshes using MeshUDF [3]. The first row shows raw meshes that have visible "staircases" widely found in NeuralUDF [8], NeUDF [7] and our method, all using MeshUDF. The second row shows the hole artifacts found in extracted meshes. These artifacts may negatively impact on visual effects.

D. Implementation Details

The UDF network is an MLP, consisting of 8 hidden layers, each with 256 elements. We use skip connections after every 4 hidden layers. The output of the UDF network is a single value representing the predicted UDF and a 256-dimensional feature vector used in the color network.

For the color network, we use another MLP with 4 hidden layers, each having 256 elements. We use the coarse-to-fine strategy proposed by Park *et al*. [10] for position encoding, setting the maximum number of frequency bands to 16 for the UDF network and 6 for the color network. For background rendering, we use NeRF++ [14] for background prediction. During training, we use the Adam optimizer [6] with a global learning rate of 5e-4. We sample 512 rays per batch and train our model for 250,000 iterations for the first stage and another 50,000 iterations for the second stage, making up a total of 300,000 iterations. We leverage MeshUDF [3] to extract meshes from trained UDFs.

For the weights of each loss function term, we empirically set $\lambda_1 = 0.1$, $\lambda_2 = 0.01$, and $\lambda_3 = 0.001$, although λ_2 is occasionally set to 0.02, and λ_3 is optional. The weight λ_m for mask loss $\mathcal{L}_{loss}$ is set to 0.1 aligning with other works [8, 11], if mask supervision is adopted.

E. More Ablation Studies

We conduct additional ablation studies in this section.

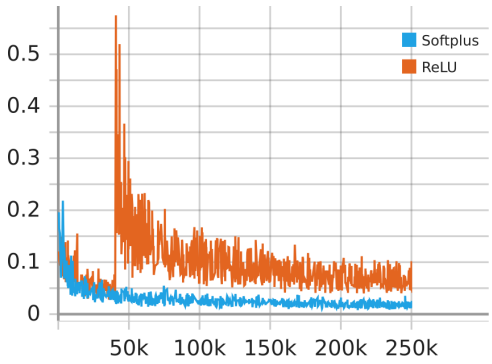

Figure 10. Ablation study on the usage of ReLU (orange) [2] versus softplus (blue) [1, 7] in the MLP output layer. The former is non-differentiable at 0 and its gradient vanishes for negative input, whereas the latter is differentiable everywhere. Using ReLU after the output layer of the MLP, the network makes progress at the early stage of training, but collapses after 40K iterations, leading to a training loss reduction through the rendering of only backgrounds. In contrast, softplus leads to correct learning of both geometry and color, and consistently decreases the training loss over iterations.

Non-negativity. Ensuring that the computed distances in the proposed method are non-negative is important, and can be achieved by applying either ReLU [2] or softplus [1, 7] to the MLP output. However, ReLU is not differentiable at 0 and has vanishing gradients for negative inputs, which can make the network difficult to train. An ablation study confirms that training with ReLU only results in early progress, but fails to learn a valid UDF later on. See Figure 10 for details.

S-value loss. Although $\mathcal{L}_s$ is optional, it is still important that the learned s is large enough so that the model has better convergence, and the result is sharper. As shown in Figure 11, there are cases where omitting $\mathcal{L}_s$ results in a worse reconstruction result, as the Chamfer distances are higher. However, the impact is negligible both in quantitative metrics and qualitative comparisons, hinting at the optional nature of $\mathcal{L}_s$.

Reference Image | CD=0.891 w/ $\mathcal{L}_s$ | CD=0.904 w/o $\mathcal{L}_s$

Figure 11. Qualitative and quantitative ablation study on the s-value loss $\mathcal{L}_s$. The visual impact and the quantitative impact are both very small.

F. Complete Results

We present the remaining results on DeepFashion3D [15] dataset in Figure 12 for UDFs and Figure 13 for reconstructed models. The UDFs of NeuralUDF exhibit apparent oscillation. The UDFs of NeUDF are nearly closed possibly resulting in watertight models. In contrast, our learned UDFs are closest to the ground truth.

NeuralUDF [8] performs poorly on some cases in Figure 13, possibly due to its complicated visibility indicator function. SDF-based methods such as VolSDF [13] and NeuS [11] produce closed or double-cover models, leading to large reconstruction loss. Note that the UDF-based method NeUDF [7] also fails to learn open models in case SS-D0. The reason is that the learned UDF of NeUDF is usually nearly closed, so it is liable to generate watertight models.

We also present the results on DTU [5] dataset and BlendedMVS [12] in Figure 14. For DTU dataset where quantitative comparisons are feasible, our Stage 2 optimization generally improves the reconstruction results (measured by Chamfer distances) of NeUDF [7] by around 10%. The reason is presented in the main text. For BlendedMVS dataset, we encourage readers to focus on the "bear" data. The brochure held by the bear (marked in red box) is an open part of the model. NeuralUDF [8] and NeAT [9],

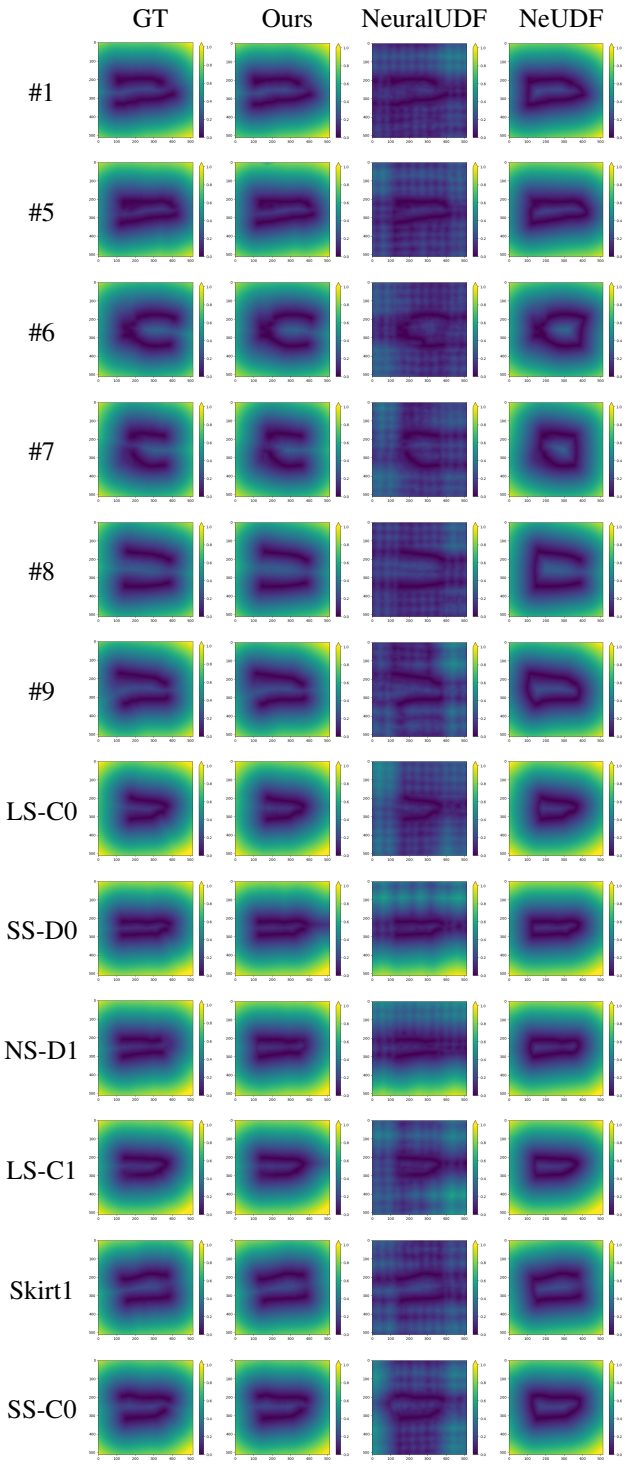

Figure 12. Visualization of the learned UDFs on cross sections for the remaining garments from DeepFashion3D.

both of which use SDF implicitly or explicitly, as explained in the main text, fail to reconstruct the open brochure. NeUDF [7] correctly reconstructs the brochure as a single-layer open surface but with large holes. Our method can generate a visually better open surface for such parts in real-life captured data.

References

[1] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating Second-Order Functional Knowledge for Better Option Pricing. In *Adv. Neural Inform. Process. Syst.* MIT Press, 2000. 2

[2] Kunihiko Fukushima. Cognitron: a self-organizing multilayered neural network. *Biological Cybernetics*, 20(3-4):121–136, 1975. 2

[3] Benoît Guillard, Federico Stella, and Pascal Fua. MeshUDF: Fast and Differentiable Meshing of Unsigned Distance Field Networks. In *Eur. Conf. Comput. Vis.*, pages 576–592, Cham, 2022. Springer Nature Switzerland. 1, 2

[4] Fei Hou, Xuhui Chen, Wencheng Wang, Hong Qin, and Ying He. Robust Zero Level-Set Extraction from Unsigned Distance Fields Based on Double Covering. *ACM Trans. Graph.*, 42(6), 2023. 1

[5] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large Scale Multi-view Stereopsis Evaluation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 406–413, 2014. 2, 6

[6] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Int. Conf. Learn. Represent.*, San Diego, CA, USA, 2015. 2

[7] Yu-Tao Liu, Li Wang, Jie Yang, Weikai Chen, Xiaoxu Meng, Bo Yang, and Lin Gao. NeUDF: Leaning Neural Unsigned Distance Fields with Volume Rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 237–247, 2023. 1, 2, 3, 5, 6

[8] Xiaoxiao Long, Cheng Lin, Lingjie Liu, Yuan Liu, Peng Wang, Christian Theobalt, Taku Komura, and Wenping Wang. NeuralUDF: Learning Unsigned Distance Fields for Multi-View Reconstruction of Surfaces with Arbitrary Topologies. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 20834–20843, 2023. 1, 2, 5, 6

[9] Xiaoxu Meng, Weikai Chen, and Bo Yang. NeAT: Learning Neural Implicit Surfaces with Arbitrary Topologies from Multi-View Images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 248–258, 2023. 2, 5, 6

[10] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable Neural Radiance Fields. In *Int. Conf. Comput. Vis.*, pages 5845–5854, 2021. 2

[11] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. In *Adv. Neural Inform. Process. Syst.*, pages 27171–27183. Curran Associates, Inc., 2021. 2, 5

[12] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. BlendedMVS: A Large-Scale Dataset for Generalized Multi-View Stereo Networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1787–1796, 2020. 2, 6

[13] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume Rendering of Neural Implicit Surfaces. In *Adv. Neural Inform. Process. Syst.*, pages 4805–4815. Curran Associates, Inc., 2021. 2, 5

[14] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NeRF++: Analyzing and Improving Neural Radiance Fields, 2020. 2

[15] Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. Deep Fashion3D: A Dataset and Benchmark for 3D Garment Reconstruction from Single Images. In *Eur. Conf. Comput. Vis.*, pages 512–530, Cham, 2020. Springer International Publishing. 2, 5

Figure 13. The remaining qualitative comparisons with VolSDF [13], NeuS [11], NeAT [9] (with mask supervision), NeuralUDF [8] and NeUDF [7] on the DeepFashion3D [15] dataset.

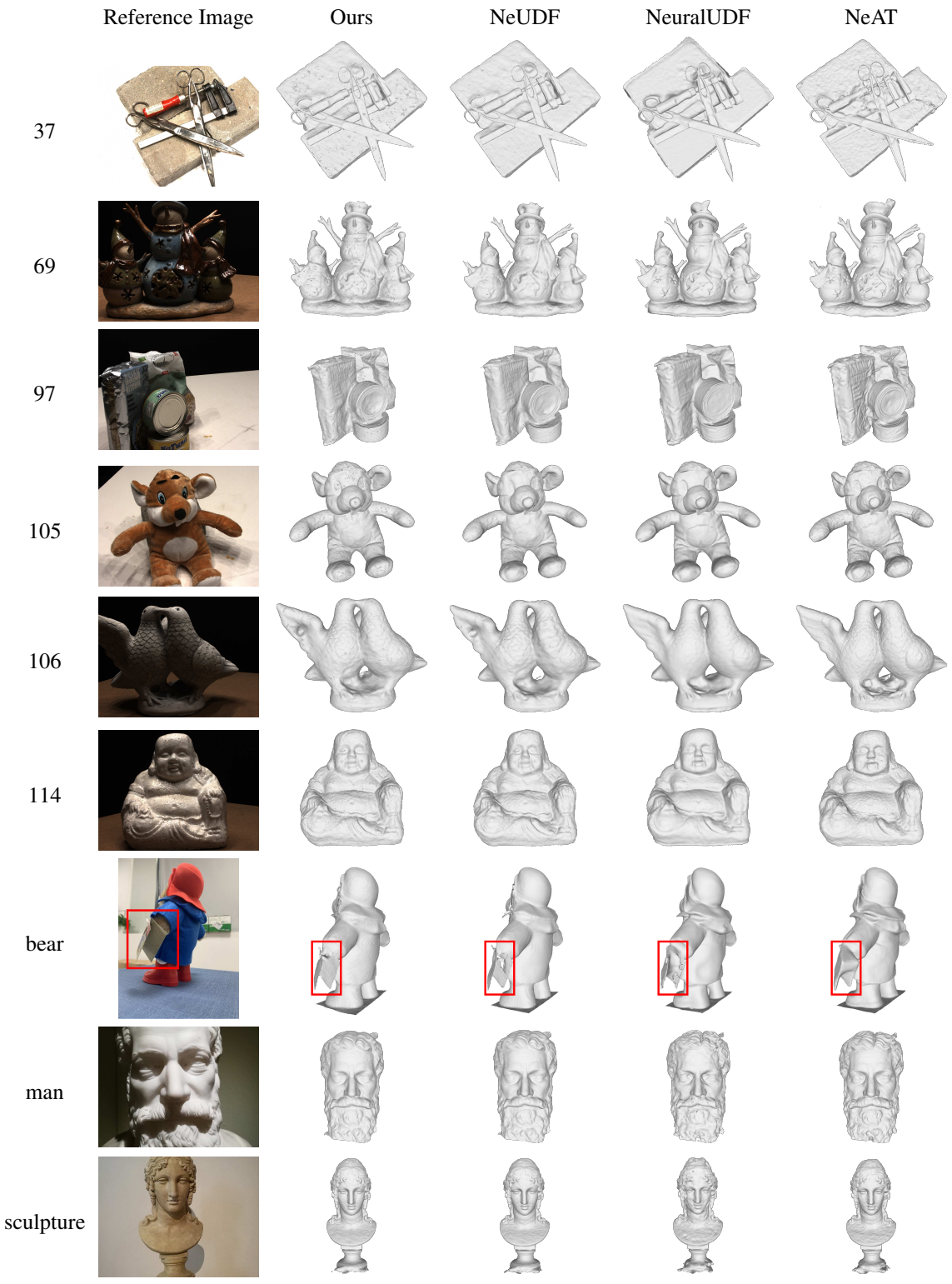

Figure 14. Qualitative comparisons with NeAT [9], NeuralUDF [8] and NeUDF [7] on the DTU [5] dataset and BlendedMVS [12] dataset.