

GaussianUDF: Inferring Unsigned Distance Functions through 3D Gaussian Splatting

Shujuan Li¹, Yu-Shen Liu¹, Zhizhong Han²

¹School of Software, Tsinghua University, Beijing, China

²Department of Computer Science, Wayne State University, Detroit, USA

lisj22@mails.tsinghua.edu.cn, liuyushen@tsinghua.edu.cn, h312h@wayne.edu

Method	Train		Test	
	2DGS	Ours	2DGS	Ours
SSIM \uparrow	0.938	0.937	0.903	0.904
PSNR \uparrow	34.48	33.78	28.37	28.40
LPIPS \downarrow	0.169	0.171	0.195	0.195

Table 1. Rendering quality on DTU [4] dataset.

1. Source Codes

We provide our demonstration code as a part of our supplementary materials. We will release the source code, data and pretrained models upon acceptance.

2. Implementation Details

Details for Self-supervision. For sampling root points, we first sample random points from a standard normal distribution and then transform them onto the 2D Gaussian’s plane using the corresponding rotation matrix, variance, and mean. The offset direction of these root points is determined by the normal vector of the Gaussian that generated them. For Gaussians whose maximum scale exceeds a certain threshold, we sample more root points. Specifically, we set this threshold to three times the mean of the maximum scales of all Gaussians.

Data Preparation. To validate the effectiveness of our method on real-world data, we additionally captured four scenes with open surfaces. We took 50–90 images per scene with a smartphone, each image with a resolution of 4032×3024 . Following the processing procedure of NeuS [7], we first used COLMAP [6] to estimate a sparse point cloud, then manually adjusted the regions of interest, and finally estimated the corresponding camera parameters for training.

3. Visualization for Optimization Process

We show the training process in Figure 1 and Figure 2. During training, we first optimize the rendering loss to learn the Gaussian representations, then use the far supervision to guide the implicit field in learning a coarse shape from the Gaussian centers. Based on this, we incorporate the projection loss to optimize the Gaussian point cloud, leading to thinner surfaces and much less noises around geom-

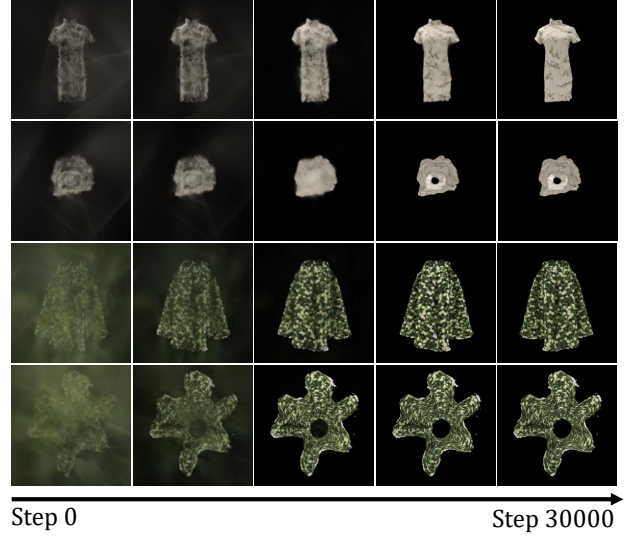


Figure 1. Training process of rendering image.

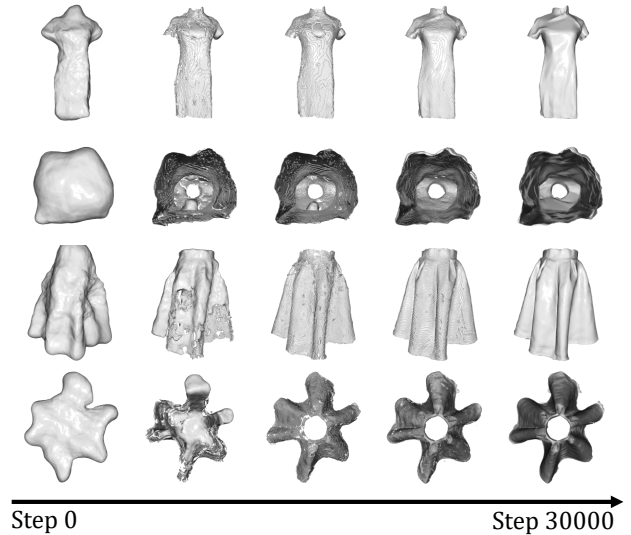


Figure 2. Training process of reconstructing meshes.

etry structures. Meanwhile, we add the self-supervision to achieve more accurate and complete reconstruction. Please refer to the whole process in our video for more details.

Method	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
NeuS[7]	1.00	1.37	0.93	0.43	1.10	0.65	0.57	1.48	1.09	0.83	0.52	1.20	<u>0.35</u>	0.49	0.54	0.84
3DGS[5]	2.14	1.53	2.08	1.68	3.49	2.21	1.43	2.07	2.22	1.75	1.79	2.55	1.53	1.52	1.50	1.96
SuGaR[2]	1.47	1.33	1.13	0.61	2.25	1.71	1.15	1.63	1.62	1.07	0.79	2.45	0.98	0.88	0.79	1.33
2DGS[3]	0.48	0.91	<u>0.39</u>	0.69	1.01	0.83	0.81	1.36	1.27	0.76	0.70	1.40	0.40	0.76	0.52	0.80
GSPull[9]	0.51	0.56	<u>0.46</u>	<u>0.39</u>	0.82	0.67	0.85	1.37	1.25	0.73	<u>0.54</u>	1.39	<u>0.35</u>	0.88	0.42	0.75
GOF[8]	<u>0.50</u>	0.82	0.37	0.37	1.12	0.74	0.73	1.18	1.29	0.68	<u>0.77</u>	0.90	0.42	0.66	<u>0.49</u>	0.74
NeuralUDF	0.69	1.18	0.67	0.44	0.90	<u>0.66</u>	0.67	1.32	0.94	0.95	0.57	<u>0.86</u>	0.37	0.56	0.55	0.75
2S-UDF*[1]	–	0.89	–	0.55	–	<u>0.68</u>	0.88	–	1.15	<u>0.70</u>	0.74	–	0.41	0.61	0.51	0.71
VRPrior[10]	0.55	0.95	0.45	0.43	0.94	0.75	<u>0.61</u>	1.40	<u>1.01</u>	0.74	0.60	0.94	0.31	<u>0.50</u>	0.50	0.71
Ours	0.62	<u>0.67</u>	0.43	0.42	<u>0.83</u>	0.86	<u>0.72</u>	<u>1.20</u>	1.03	0.68	0.61	0.63	0.43	<u>0.56</u>	0.52	0.68

Table 2. Numerical comparisons in terms of CD on DTU [4] dataset. Bold and underlined numbers indicate the first and second best performance, respectively. *We use the metrics reported in the paper of 2S-UDF.

Setting	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
Only Far	0.76	0.95	0.68	0.73	0.92	1.14	1.09	1.4	1.35	0.81	0.99	1.41	0.96	0.84	0.77	0.99
Far & Near	0.6	0.69	0.4	0.48	0.84	0.84	0.84	1.37	1.13	0.7	0.76	1.16	0.49	0.74	0.62	0.78
Far & Proj	0.7	0.73	0.6	0.54	0.85	1.3	0.99	1.25	1.2	0.7	0.96	1.29	0.57	0.81	0.64	0.88
w/o Warp	0.65	0.69	0.41	0.48	0.82	0.85	0.76	1.21	1.08	0.67	0.65	1.14	0.51	0.65	0.53	0.74
w/o Near	0.72	0.79	0.53	0.4	0.8	1.02	0.91	1.29	1.02	0.72	0.68	0.79	0.65	0.67	0.6	0.77
w/o Proj	0.81	0.85	0.49	0.5	0.87	0.87	0.79	1.25	1.1	0.89	0.58	0.67	0.49	0.6	0.64	0.76
Full Model	0.62	0.67	0.43	0.42	0.83	0.86	0.72	1.2	1.03	0.68	0.61	0.63	0.43	0.56	0.62	0.68

Table 3. The detailed CD on DTU [4] dataset for ablation studies.

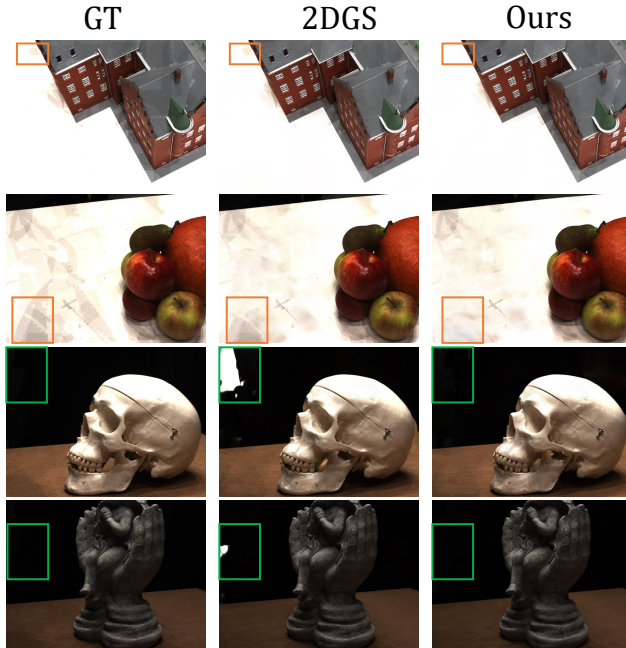


Figure 3. Rendering quality analysis on DTU [4] dataset. Our method renders images with fewer floating artifacts.

4. Novel View Synthesis

Although our method focuses on multi-view surface reconstruction tasks, Gaussian splatting naturally supports novel view synthesis. Since we adopt a splatting process similar to 2DGS [3], we compare our rendering results with 2DGS to analyze how our approach impacts rendering quality. On the DTU dataset, we use 12.5% of the images for validation, while the remaining images are kept as the training set, and we report the rendering metrics including PSNR, SSIM and LPIPS.

The numerical comparisons are reported in Table 4. We notice a drop for PSNR on the training set but the results on test set show that our method achieves comparable rendering quality with 2DGS in novel view synthesis. The impact of our method on the rendering process primarily comes from the projection loss L_{proj} that reduces the noises in the Gaussian point clouds. To further analyze the reason, we visualize the rendered images in Figure 4. The scenes in the DTU dataset feature complex illumination conditions, and 2DGS tends to overfit these illumination variations, resulting in the generation of additional noisy Gaussians during the densification process. Our method smooths out these noises, which compromises the representation of the illumination. However, this smoothing removes the floating artifacts and leads to an improvement in rendering quality.



Figure 4. All reconstruction results on DF3F [11] dataset. Our method accurately reconstructs the open surfaces.

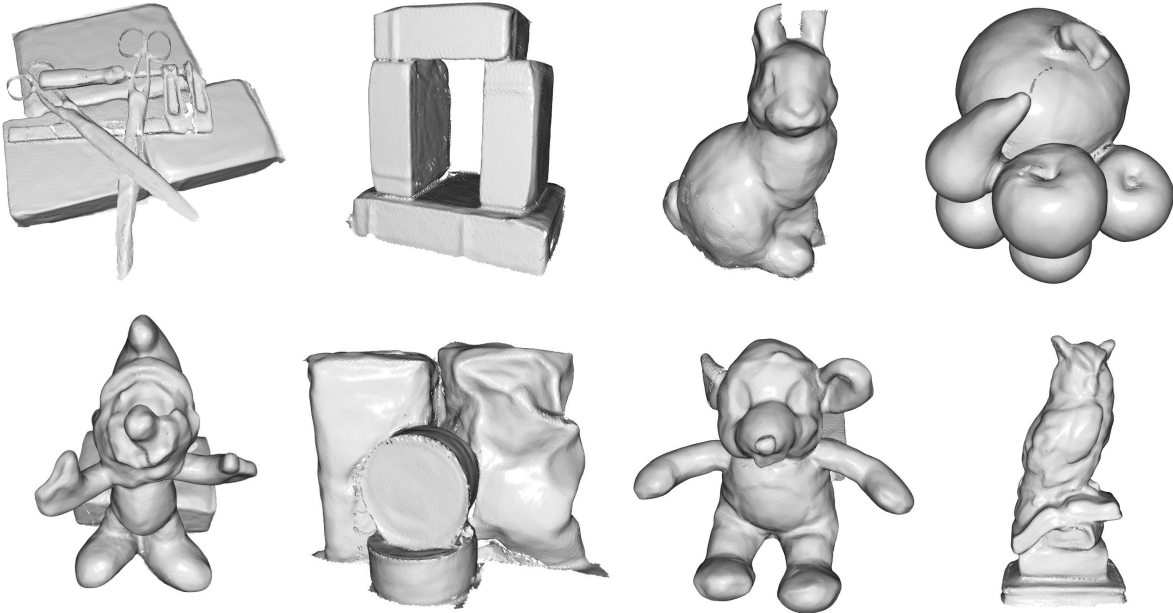


Figure 5. More reconstruction results on DTU [4] dataset.

Overall, our method achieves results that are quantitatively comparable to 2DGS.

5. More Results

We show more visual comparisons in our video. The comparisons show that our method recovers more accurate and thinner surfaces for open structures either in synthetic or real scenes. We visualize more reconstructed meshes on

Method	Train		Test	
	2DGS	Ours	2DGS	Ours
SSIM \uparrow	0.938	0.937	0.903	0.904
PSNR \uparrow	34.48	33.78	28.37	28.40
LPIPS \downarrow	0.169	0.171	0.195	0.195

Table 4. Rendering quality on DTU dataset.

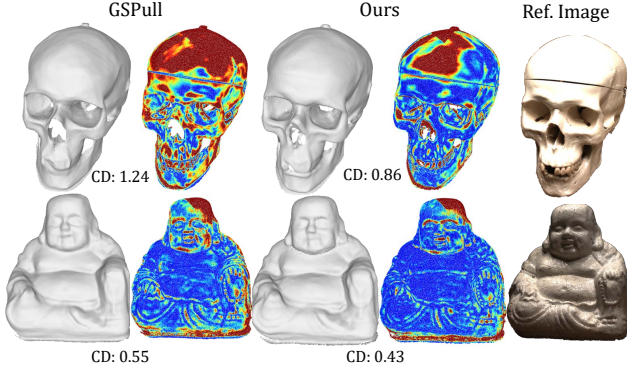


Figure 6. Visual comparisons with GSPull [9] show that our method reconstructs surfaces with less errors.

DF3D [11] in Figure 4 and on DTU [4] in Figure 5. And we provide visual comparisons with GSPull [9] on the DTU dataset shown in Fig. 6. The results show our method produces more accurate and complete surfaces.

The detailed results of the comparison experiments and ablation studies on the DTU dataset are reported in Table 2 and Table 3, respectively. Our method achieves a lower average Chamfer distance, indicating that the reconstructed surfaces have smaller errors and that our approach is more robust compared to baseline methods.

6. Video

We provide a video containing training process visualization, field visualization, visual comparisons on all datasets, and application of point cloud deformation as a part of our supplementary materials.

References

- [1] Junkai Deng, Fei Hou, Xuhui Chen, Wencheng Wang, and Ying He. 2S-UDF: A novel two-stage UDF learning method for robust non-watertight model reconstruction from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5084–5093, 2024. 2
- [2] Antoine Guédon and Vincent Lepetit. SuGaR: Surface-aligned Gaussian splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2024. 2
- [3] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2
- [4] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 406–413, 2014. 1, 2, 3, 4
- [5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2
- [6] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1
- [7] 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 *Advances in Neural Information Processing Systems*, pages 27171–27183, 2021. 1, 2
- [8] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Trans. Graph.*, 2024. 2
- [9] Wenyuan Zhang, Yu-Shen Liu, and Zhizhong Han. Neural signed distance function inference through splatting 3D Gaussians pulled on zero-level set. In *Advances in Neural Information Processing Systems*, 2024. 2, 4
- [10] Wenyuan Zhang, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. Learning unsigned distance functions from multi-view images with volume rendering priors. In *Proceedings of the European Conference on Computer Vision*, 2024. 2
- [11] 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 *Proceedings of the European Conference on Computer Vision*, pages 512–530. Springer, 2020. 3, 4