SurfaceSplat: Connecting Surface Reconstruction and Gaussian Splatting

Supplementary Material

A. Implementation Details

Mesh sampling for 3DGS. In the main text Sec. 3.2, we sample surface points from coarse mesh as 3DGS [8] initialization. Specifically, we render depth maps of training viewpoints using the coarse mesh and sample $5k$ points from each depth map. Then, we unproject the depth points into 3D color points. These points are then fused to generate a total of $50k$ points, which are subsequently combined with the sparse results from COLMAP [7].

3DGS training details. For 3DGS [8] training, convergence is achieved effectively with $7k$ iterations under sparse inputs. Specifically, densification begins at 500 iterations with intervals of 100 iterations. An opacity reset is performed at $3k$ iterations, while other parameters remain consistent with the original implementation. To ensure a fair comparison, other GS-based methods tested in the paper also follow this training strategy.

Camera position. Camera pose refers to the camera's position $\boldsymbol{c}$ and orientation matrix $\mathbf{R}$ in the world coordinate system. The proposed two methods in the main text Sec. 3.3 (Camera position perturbation and interpolation) focus exclusively on resampling camera positions while ensuring that the camera orientation consistently points toward the center of the object, located at $(0, 0, 0)$ in the world coordinate system.

B. More Experimental Results

B.1. Results on different sparsity levels.

To better understand the strengths and weaknesses of the SDF-based and 3DGS-based methods, we evaluate them under varying levels of sparse input. Specifically, we select 9 scenes from the MobileBrick dataset [3], and the reported results are averaged across all scenes. Table 1 presents results across different sparsity levels. 3DGS-based methods (e.g., GOF[14]) significantly outperform SDF-based methods (e.g., Voxurf[9]) in novel view rendering, while Voxurf consistently achieves better surface reconstruction than GOF. We hypothesize that this stems from SDF's dense representations, which effectively capture global geometry, and 3DGS's sparse representations, which excel at preserving local details. To leverage the strengths of both approaches, we propose a hybrid method, leading to our proposed SurfaceSplat framework.

Table 1. Rendering and mesh reconstruction results on SDF-based and GS-based methods with different input image numbers.

	Rendering (PSNR)		Mesh (F1 Score)	
Input	Voxurf[9]	GOF[14]	Voxurf[9]	GOF[14]
5	11.78	12.61	31.70	30.15
10	14.06	16.00	63.60	53.64
15	14.90	18.30	66.82	60.20
20	15.83	19.81	70.39	65.86
30	16.93	21.43	71.97	68.25

B.2. Per-scene 10-view mesh results on Mobilebrick

Table 2 presents the surface reconstruction results (F1 scores) for each MobileBrick scene, using 10 input images per scene for surface reconstruction. The best scores are highlighted in bold.

B.3. Per-scene 3-view reconstruction mesh on DTU

Previous methods [12, 13] use 3 manually selected images with the best overlap for surface reconstruction. However, we argue that this does not reflect real-world reconstruction scenarios. Instead, we propose evenly sampling 5 images for sparse-view reconstruction. Nonetheless, we also report results under the 3-view setting for fair comparison with previous methods. Table 3 presents the results, demonstrating that our method outperforms existing alternatives. Furthermore, our framework is compatible with a variety of SDF-based methods and 3D Gaussian representations. In addition to integrating Voxurf into our pipeline, we also experiment with incorporating SparseCraft, and observe similarly strong reconstruction performance, demonstrating the generality and versatility of our approach.

B.4. SDF-3DGS Mutual Enhancement.

Our method enables mesh reconstruction and 3DGS to enhance each other's performance. Table 4 presents the ablation study results on MobileBrick, demonstrating the effectiveness of this mutual enhancement. The results show that without support from the other module, performance drops significantly for both components. Additionally, we analyze the impact of cyclic optimization in our method. Running two cycles provides a slight performance improvement. However, for a trade-off between efficiency and performance, we use a single loop iteration as the default setting.

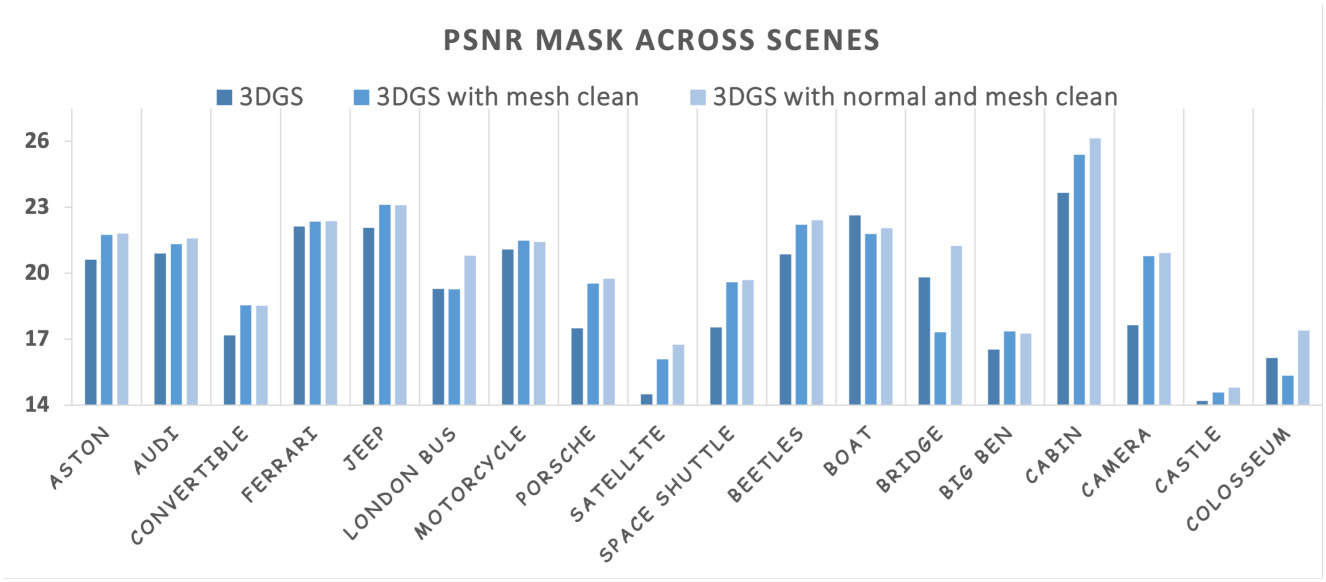

Figure 1. Ablation study on mesh-based sampling for enhancing 3DGS rendering. We report foreground PSNR here.

Table 2. Quantitative F1 score (↑) across all 18 MobileBrick test scenes.

F1 Score	Aston	Conv.	Ferrari	Jeep	Bus	Moto.	Porsche	Beetles	Big_ben	Boat	Audi	Bridge	Cabin	Camera	Castle	Colosseum	Satellite	Shuttle	Mean	Time
Voxurf [9]	55.8	53.1	69.4	**88.0**	58.7	88.2	59.7	64.8	55.0	60.6	**83.1**	67.9	78.4	91.0	11.5	22.3	61.7	60.9	62.9	55 mins
MonoSDF [13]	51.2	42.6	56.9	31.4	13.8	54.5	40.1	36.9	2.3	4.8	60.2	67.5	76.3	25.6	4.9	4.7	38.9	49.5	36.8	6 hrs
2DGS [1]	42.8	36.5	62.1	71.7	34.9	51.6	39.3	47.2	28.4	67.8	73.9	81.2	62.5	43.7	17.3	18.9	4.6	40.8	45.8	10 mins
GOF [14]	55.7	48.3	67.8	70.2	46.4	73.5	50.9	62.7	42.3	71.9	77.2	78.6	73.8	53.4	13.6	26.5	33.8	50.2	55.4	50 mins
Ours	**60.8**	**58.9**	**70.1**	86.5	**67.3**	**89.8**	**62.4**	**76.4**	**62.3**	**81.5**	80.9	**94.3**	**80.6**	**91.4**	**17.8**	**32.1**	**73.5**	**64.7**	**69.0**	1 hr

B.5. Efficacy of mesh-based sampling for 3DGS

Fig. 3 (e)&(f) in main paper provide a visual comparison between our mesh-based point sampling approach and COLMAP-generated sparse points. The comparison shows that our method achieves noticeably better visual quality in object regions, which leads to enhanced 3DGS rendering quality. The results across each scene on MobileBrick are summarized in Fig. 1. This demonstrates the effectiveness of our mesh cleaning and normal loss in enhancing 3DGS [8] rendering quality.

B.6. Efficacy of 3DGS for mesh reconstruction

Sec. 3.3 in the main text mentioned that 3DGS [8] can provide higher-quality novel view images, as extended views, are combined with the original inputs to refine the mesh. Specifically, we propose two novel view pose strategies, and we visualize the resulting novel view images in Fig. 2.

B.7. Visual reconstruction on BlendedMVS

We performed mesh reconstruction using 3-view input on the BlendedMVS dataset [10]. Fig. 3 presents the results, comparing our method with two representative approaches: Voxurf (SDF-based) and 2DGS (3DGS-based). While none of the methods perform well in this setting, our approach achieves slightly better results than the alternatives. We hypothesize that 3 input views are insufficient for real-world surface reconstruction, highlighting the challenges of extreme sparsity.

C. More Qualitative Results

C.1. DTU rendering results

We visualize and compare the novel view synthesis results of our method (based on SparseCraft) against the original SparseCraft on the DTU dataset under sparse input settings of 3, 6, and 9 views.

C.2. Mesh reconstruction on Mobilebrick and DTU

Fig. 5 Presents additional mesh reconstruction results on MobileBrick (10 images) and DTU (5 images). Training images are uniformly sampled to minimize overlap, making the task more challenging and reflect the real-world reconstruction problem.

Table 3. Quantitative results of 3-view reconstruction on DTU. Chamfer Distance (mm)↓ is reported. Note that SparseNeuS requires pertaining on large-scale dataset and ground-truth masks at inference time. The best results are **bolded**, while the second-best are underlined.

Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean	Time
Voxurf [9]	3.75	6.02	4.56	3.62	4.53	2.80	3.79	4.23	4.26	2.09	4.40	4.44	1.36	4.60	2.51	3.79	50 mins
MonoSDF [13]	6.76	3.50	**1.79**	0.73	1.95	**1.45**	1.25	1.63	**1.40**	0.98	4.03	1.75	0.94	2.54	3.55	2.28	6 hrs
SparseNeuS [5]	4.10	4.21	3.64	1.78	2.89	2.49	1.76	2.50	2.88	2.16	2.04	3.27	1.29	2.36	1.75	2.61	Pretrain + 2 hrs ft
VolRecon [6]	3.56	4.48	4.24	3.15	2.85	3.91	2.51	2.65	2.56	2.67	2.84	2.77	1.60	3.09	2.19	3.00	2days pre
ReTR [4]	3.78	3.91	3.95	3.15	2.91	3.50	2.79	2.76	2.50	2.35	3.56	4.02	1.70	2.72	2.16	3.05	3days pre
SparseCraft [11]	2.13	2.83	2.68	**0.70**	1.49	2.15	1.29	**1.37**	1.57	1.13	**1.22**	2.53	**0.61**	**0.83**	0.99	1.57	1.5 hours
Ours(Voxurf)	2.65	4.47	1.87	1.22	2.28	1.98	1.33	1.96	2.66	1.94	1.86	**1.67**	0.78	1.22	1.63	1.96	1hour
Ours(SparseCraft)	**1.86**	**2.56**	2.85	0.75	**1.40**	1.99	**1.13**	1.42	1.51	**0.90**	1.28	2.26	0.68	0.89	**0.94**	**1.49**	2 hours

Table 4. Ablations studies on effectiveness of our proposed modules on MobileBrick test scenes.

	Meshing		Rendering	
	F1↑	CD↓	PSNR↑	PSNR-F↑
SDF-based method w/o 3DGS	**62.42**	**13.3**	14.34	18.34
3DGS-based method w/o SDF	54.96	11.0	**16.52**	**18.36**
Ours (One cycle)	68.97	**9.7**	17.48	20.45
Ours (Two cycles)	**69.14**	9.9	**17.58**	**20.55**

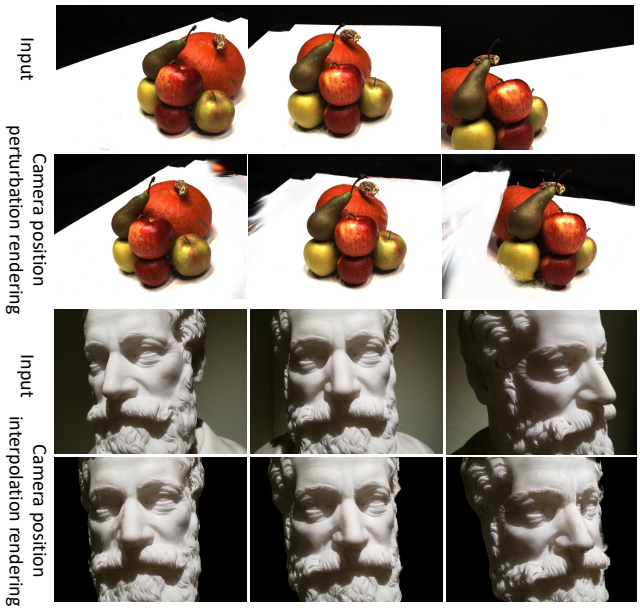

Figure 2. Visualization of newly rendered images with different pose expansion strategies. The top row presents results on DTU [2] (scan63), while the bottom row shows results on BlendedMVS [10] (Man).

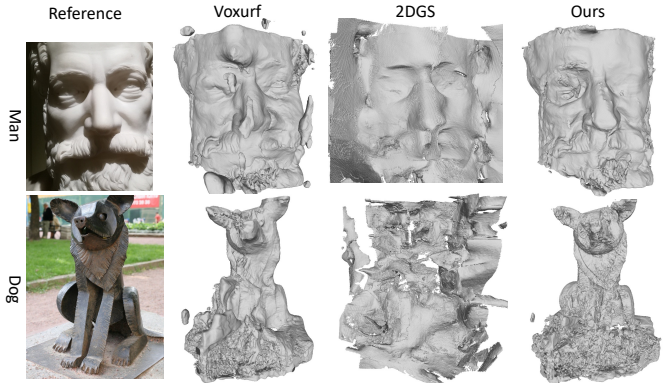

Figure 3. Qualitative comparison of 3-view mesh reconstruction on BlendedMVS dataset.

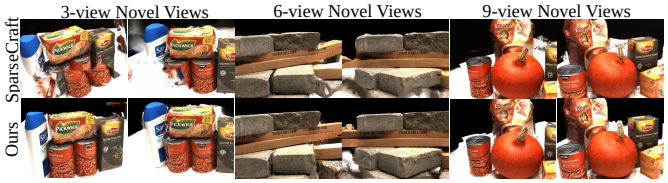

Figure 4. DTU novel view synthesis comparison.

C.3. MobileBrick rendering results

Fig. 6 presents additional novel view renderings on MobileBrick, demonstrating that our method achieves superior rendering quality. This improvement stems from the stable initialization point cloud provided by the coarse mesh.

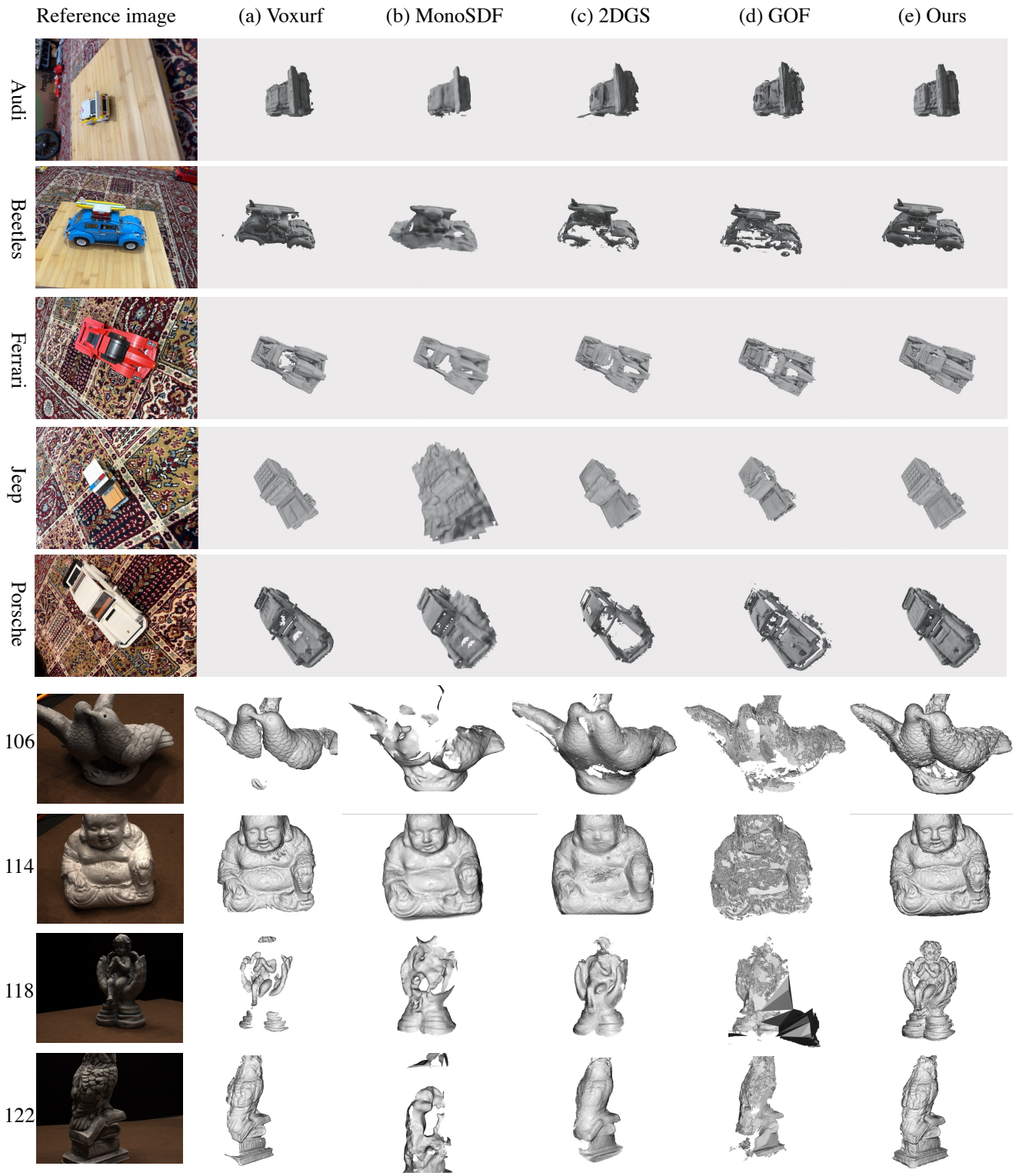

Figure 5. More qualitative mesh reconstruction results on MobileBrick and DTU.

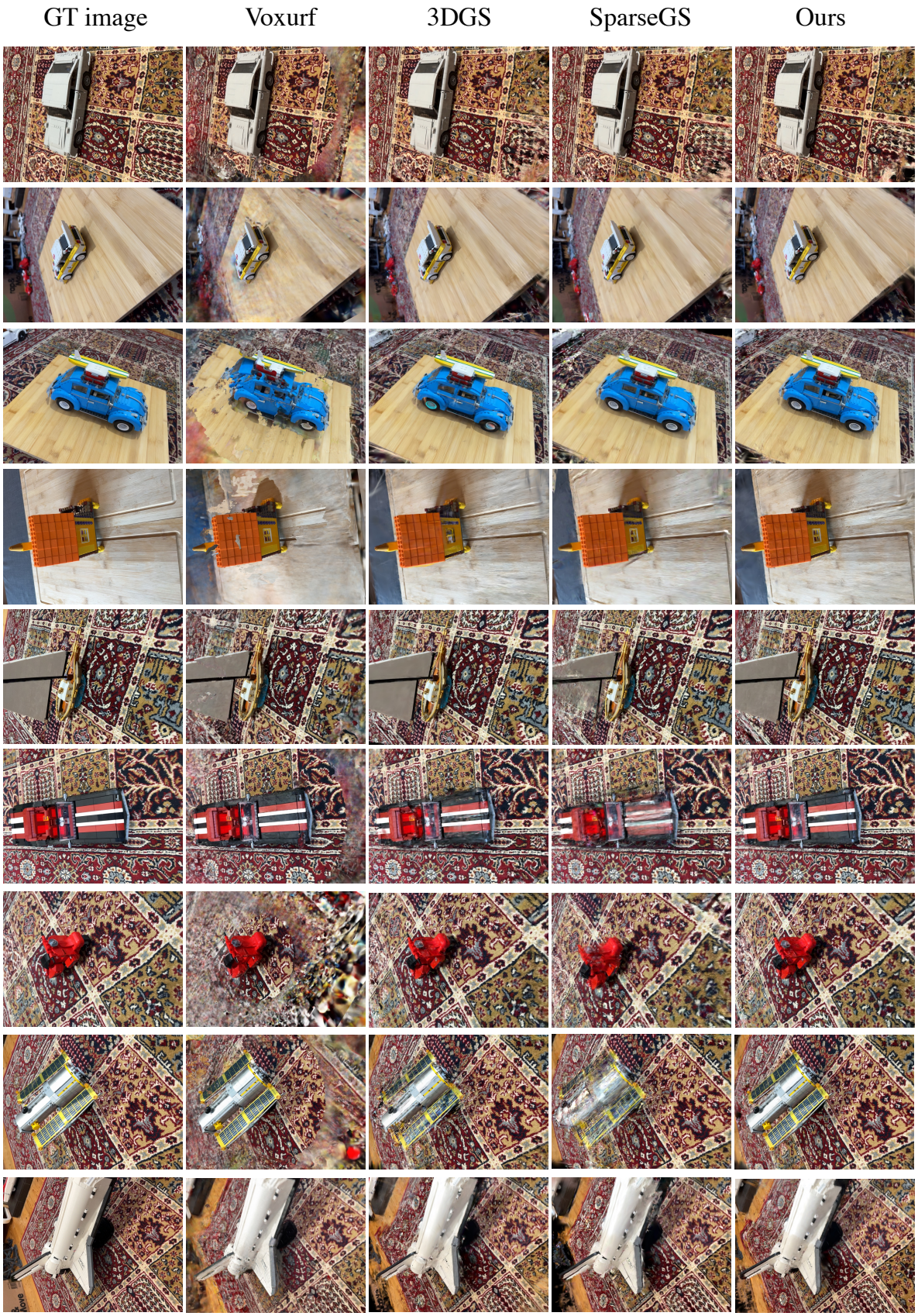

Figure 6. More qualitative novel view rendering results on MobileBrick.

References

[1] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 2

[2] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014. 3

[3] Kejie Li, Jia-Wang Bian, Robert Castle, Philip HS Torr, and Victor Adrian Prisacariu. Mobilebrick: Building lego for 3d reconstruction on mobile devices. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4892–4901, 2023. 1

[4] Yixun Liang, Hao He, and Yingcong Chen. Retr: Modeling rendering via transformer for generalizable neural surface reconstruction. *Advances in Neural Information Processing Systems*, 36:62332–62351, 2023. 3

[5] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *Eur. Conf. Comput. Vis.*, pages 210–227. Springer, 2022. 3

[6] Yufan Ren, Fangjinhua Wang, Tong Zhang, Marc Pollefeys, and Sabine Süsstrunk. Volrecon: Volume rendering of signed ray distance functions for generalizable multi-view reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16685–16695, 2023. 3

[7] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Eur. Conf. Comput. Vis.*, pages 501–518. Springer, 2016. 1

[8] Hao Sun, Junping Qin, Lei Wang, Kai Yan, Zheng Liu, Xinglong Jia, and Xiaole Shi. 3dgs-hd: Elimination of unrealistic artifacts in 3d gaussian splatting. In *2024 6th International Conference on Data-driven Optimization of Complex Systems (DOCS)*, pages 696–702. IEEE, 2024. 1, 2

[9] Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. Voxurf: Voxel-based efficient and accurate neural surface reconstruction. In *Int. Conf. Learn. Represent.*, 2023. 1, 2, 3

[10] 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 1790–1799, 2020. 2, 3

[11] Mae Younes, Amine Ouasfi, and Adnane Boukhayma. Sparsecraft: Few-shot neural reconstruction through stereopsis guided geometric linearization. In *European Conference on Computer Vision*, pages 37–56. Springer, 2024. 3

[12] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4578–4587, 2021. 1

[13] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Adv. Neural Inform. Process. Syst.*, 2022. 1, 2, 3

[14] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Trans. Graph.*, 2024. 1, 2