ResGS: Residual Densification of 3D Gaussian for Efficient Detail Recovery

Supplementary Material

A. Additional Experiment Results

In this section, we provide additional experimental results of our work.

A.1. Evaluation on large-scale urban datasets

To evaluate the performance of our approach on large-scale urban datasets, we performed experiments using the Waymo [12] dataset and compared our results against GaussianPro [3]. The results are shown in the table below. It can be seen that our method achieves better results across all metrics.

Table 1. **Results on the Waymo dataset compared with GaussianPro.**

Dataset / Method | Metrics	Waymo PSNR ↑	SSIM ↑	LPIPS ↓	Mem.
GaussianPro	34.68	0.949	**0.191**	297MB
Ours	**35.29**	**0.951**	**0.191**	284MB

A.2. Variance of PSNR on Deep Blending Dataset

A notable variance in the PSNR metrics of our method is observed on the Deep Blending [5] dataset. We tested our method on the Deep Blending dataset 10 times and reported the average performance and population standard deviation. The results are shown in Table 2. The table demonstrates that the PSNR scores on the Deep Blending dataset exhibit a notable variance. The underlying reason, as discussed in our limitations, is that our method does not address issues related to occlusion and overfitting, making the Deep Blending dataset particularly challenging for our approach. This highlights opportunities for future optimization and improvement. Furthermore, we report the average performance and population standard deviation over 10 runs for the other two datasets, Mip-NeRF360 [1] and Tanks&Temples [7], as presented in Table 3 and Table 4, respectively. The PSNR metrics for these two datasets do not show a notable variance, showcasing the robustness of our method in most scenarios.

Table 2. **Average and standard deviation results of three metrics on the Deep Blending dataset.**

Dataset	Deep Blending					
Method | Metrics	PSNR ↑		SSIM ↑		LPIPS ↓	
	Avg	Stdev	Avg	Stdev	Avg	Stdev
Ours-Small (AbsGS)	29.71	0.211	0.903	0.002	0.235	0.002
Ours (3D-GS)	29.64	0.205	0.900	0.003	0.233	0.002
Ours (AbsGS)	29.68	0.173	0.900	0.002	0.228	0.002

Table 3. **Average and standard deviation results of three metrics on the Mip-NeRF360 dataset.**

Dataset	Mip-NeRF360					
Method | Metrics	PSNR ↑		SSIM ↑		LPIPS ↓	
	Avg	Stdev	Avg	Stdev	Avg	Stdev
Ours-Small (AbsGS)	27.93	0.010	0.830	0.001	0.191	0.001
Ours (3D-GS)	27.99	0.011	0.831	0.001	0.187	0.001
Ours (AbsGS)	27.99	0.011	0.833	0.001	0.174	0.001

Table 4. **Average and standard deviation results of three metrics on the Tank&Temples dataset.**

Dataset	Tank&Temples					
Method | Metrics	PSNR ↑		SSIM ↑		LPIPS ↓	
	Avg	Stdev	Avg	Stdev	Avg	Stdev
Ours-Small (AbsGS)	24.21	0.081	0.862	0.001	0.151	0.001
Ours (3D-GS)	24.26	0.058	0.864	0.001	0.141	0.001
Ours (AbsGS)	24.27	0.063	0.866	0.001	0.133	0.001

A.3. Further Analysis of Residual Split on 3D-GS

In our paper, we tested the compatibility of our residual split across multiple pipelines without modifying any hyper-parameters. Here, we present the results of applying residual split to 3D-GS [6] while adjusting the densification threshold τ to maintain the same storage size as 3D-GS, further demonstrating the capability of our method to enhance rendering quality. The results are shown in Table 5, showing that our method yields a further performance increase.

A.4. Results using random initialization

To assess the robustness of our method to random initialization, we conducted experiments using randomly initialized point clouds and compared our approach with other methods trained on the same random point clouds. The results are shown in Table 7. For the Mip-NeRF360 [1] and Tanks&Temples [7] datasets, our model shows a noticeable drop in quality compared to when initialized with SfM [11]. However, compared to other methods, it still achieves the best performance across all three metrics on Mip-NeRF360, and the highest SSIM and LPIPS on Tanks&Temples. As for Deep Blending [5], our method and others, excluding Scaffold-GS and Octree-GS, do not exhibit a significant performance drop. While we do not achieve the highest scores, our method remains competitive, ranking second across all metrics.

A.5. Periodic Opacity Reduction

3D-GS [6] often encounters redundancy issues, which we categorize into two main causes: the first is the use of an ex-

Table 5. **Results of residual split upon 3D-GS with varied gradient threshold.**

Dataset / Method | Metrics	Mip-NeRF360				Tanks&Temples				Deep Blending			
	PSNR ↑	SSIM ↑	LPIPS ↓	Mem	PSNR ↑	SSIM ↑	LPIPS ↓	Mem	PSNR ↑	SSIM ↑	LPIPS ↓	Mem
3D-GS	27.21	**0.815**	0.214	734MB	23.14	0.841	0.183	411MB	29.41	**0.903**	0.243	676MB
3D-GS + residual split	**27.48**	**0.815**	**0.213**	722MB	**23.85**	**0.846**	**0.174**	374MB	**29.83**	0.901	**0.238**	640MB

Table 6. **Results of our method without extended periodic opacity reduction.** Here, **No EPOR** refers to our method trained without applying the extended periodic opacity reduction. **No EPOR + CT** refers to the model without the extended periodic opacity reduction but with an adjusted gradient threshold to ensure that the final storage size is approximately the same as our full model with extended periodic opacity reduction.

Dataset / Method | Metrics	Mip-NeRF360				Tanks&Temples				Deep Blending			
	PSNR ↑	SSIM ↑	LPIPS ↓	Mem	PSNR ↑	SSIM ↑	LPIPS ↓	Mem	PSNR ↑	SSIM ↑	LPIPS ↓	Mem
Ours (No EPOR)	**28.02**	**0.833**	**0.173**	973MB	24.32	**0.867**	**0.130**	591MB	29.86	0.902	**0.227**	911MB
Ours (No EPOR + CT)	28.00	**0.833**	0.181	608MB	24.35	0.866	0.139	380MB	**29.96**	**0.904**	0.230	564MB
Ours (full)	28.00	**0.833**	0.174	698MB	**24.38**	**0.867**	0.132	351MB	29.91	0.902	**0.227**	586MB

Table 7. **Results using random initialized point clouds, compared with other methods.**

Dataset / Method | Metrics	Mip-NeRF360				Tanks&Temples				Deep Blending			
	PSNR ↑	SSIM ↑	LPIPS ↓	Mem.	PSNR ↑	SSIM ↑	LPIPS ↓	Mem.	PSNR ↑	SSIM ↑	LPIPS ↓	Mem.
Scaffold-GS	25.90	0.743	0.301	157MB	22.00	0.768	0.282	51MB	29.48	0.891	0.285	56MB
Octree-GS	25.70	0.739	0.292	190MB	22.15	0.794	0.226	81MB	28.29	0.873	0.300	92MB
3D-GS	25.90	0.769	0.266	632MB	21.19	0.777	0.241	339MB	29.42	0.897	0.253	605MB
AbsGS	25.50	0.773	0.226	741MB	20.89	0.786	0.211	304MB	29.52	0.897	0.241	478MB
Mini-Splatting-D	25.18	0.792	0.207	1.10GB	19.77	0.760	0.220	1.02GB	29.76	0.901	0.212	1.09GB
Pixel-GS	26.23	0.792	0.221	1.14GB	21.10	0.788	0.207	939MB	28.76	0.889	0.256	1.04GB
Ours-Random	26.38	0.799	0.200	750MB	22.02	0.815	0.178	282MB	29.57	0.899	0.234	464MB
Ours-SfM	28.00	0.833	0.174	698MB	24.38	0.867	0.132	351MB	29.91	0.902	0.227	586MB

cessive number of small-scale Gaussians to represent coarse areas, and the second is the overlap between Gaussians, resulting in redundant Gaussians that contribute minimally to the rendered image. Our residual split can solve the first issue but does not address the second. To tackle the second issue, as mentioned in our experiment settings, we implemented the periodic opacity reduction technique in [2, 13]. Specifically, the opacity of Gaussians is periodically reduced, causing the opacity of those that contribute little to the rendered image to diminish to a negligible value, allowing them to be effectively pruned. Additionally, we extended this operation to occur after the densification stage, discovering that it further reduces redundancy without compromising fidelity. Here, we present the experimental results of our method without extending the periodic opacity reduction operation, as shown in Table 6. From the table, we can observe that extending the periodic opacity reduction operation effectively reduces redundancy without compromising the fidelity of the rendered images. Furthermore, when the operation is not extended and the gradient threshold is adjusted to maintain storage approximately equal to our full model, the rendering quality shows only a slight decrease in LPIPS metrics. This indicates that while extending the operation can enhance the performance of our method, its overall impact on the performance improvement is relatively minor.

A.6. Image Resolution of Mip-NeRF360 Dataset

In our paper, we mentioned that the image resolution on the Mip-NeRF360 [1] dataset varies across different works. 3D-GS [6] uses the provided "images_4", the official downscaled 4 times images from the dataset, for outdoor scenes and "images_2" for indoor scenes. Most works [4, 13, 15] follow the 3D-GS setting, but Scaffold-GS [8], Octree-GS [10], Pixel-GS [16] uses different settings. Specifically, Scaffold-GS and Octree-GS downscale original images to 1.6K resolution, while Pixel-GS downscales original images by a factor of 4 for outdoor scenes and 2 for indoor scenes during the training process, instead of using the provided downscaled images. This discrepancy results in differences in experimental settings. To further demonstrate the performance of our method, we provide its results under the two different settings, as shown in Table 8 and 9. The tables show that our method prevails even under these alternate settings.

Table 9. **Results on the Mip-NeRF360 dataset with same setting as Pixel-GS.**

Dataset	Mip-NeRF360		
Method | Metrics	PSNR ↑	SSIM ↑	LPIPS ↓
Pixel-GS	27.88	0.834	0.176
Ours-Small (AbsGS)	28.21	0.841	0.176
Ours (3D-GS)	28.28	0.843	0.172
Ours (AbsGS)	28.30	0.845	0.160

Table 8. **Results on the Mip-NeRF360 dataset with same setting as Scaffold-GS and Octree-GS.**

Dataset	Mip-NeRF360		
Method | Metrics	PSNR ↑	SSIM ↑	LPIPS ↓
Scaffold-GS	27.71	0.813	0.221
Octree-GS	27.73	0.815	0.217
Ours-Small (AbsGS)	28.04	0.831	0.191
Ours (3D-GS)	28.10	0.832	0.187
Ours (AbsGS)	28.14	0.834	0.174

A.7. Per Scene Results

We present the per-scene results of the used metrics in Table 10-15. For works that did not report per-scene results, we obtained and reported them using their released code. The works selected for comparison are mainly the same as in the paper: 3D-GS [6], Scaffold-GS [8], Octree-GS [10], AbsGS [13], Pixel-GS [16], Mini-Splatting-D [4], Plenoxels [14], Instant-NGP [9] and Mip-NeRF360 [1]. Note that we did not show the results of FreGS [15] since they did not report their per-scene results or release their code. Works retrained on the Mip-NeRF360 [1] are marked with a *.

Table 10. **Per-scene PSNR metrics of Mip-NeRF360 dataset.**

Method | Scenes	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
Plenoxels	21.91	20.10	23.49	20.66	22.25	27.59	23.62	23.42	24.67
Instant-NGP	22.17	20.65	25.07	23.47	22.37	29.69	26.69	29.48	30.69
Mip-NeRF360	24.31	21.65	26.88	26.18	22.93	31.47	29.45	31.99	33.40
Scaffold-GS*	25.07	21.36	27.35	26.64	22.93	32.04	29.50	31.63	32.71
Octree-GS*	25.05	21.33	27.58	26.41	22.81	32.09	29.48	30.89	31.71
3D-GS	25.25	21.52	27.41	26.55	22.49	30.63	28.70	30.32	31.98
AbsGS	25.29	21.35	27.48	26.71	21.99	31.61	29.03	31.62	32.32
Pixel-GS*	25.21	21.49	27.42	26.84	22.09	31.45	29.05	31.65	32.53
Mini-Splatting-D	25.55	21.50	27.67	27.11	22.13	31.41	28.72	31.75	31.72
Ours-Small (AbsGS)	25.62	21.80	27.72	27.27	22.90	32.41	29.32	31.95	32.48
Ours (3D-GS)	25.55	22.07	27.71	27.12	22.82	32.48	29.41	32.09	32.74
Ours (AbsGS)	25.62	21.78	27.82	27.19	22.57	32.51	29.50	32.26	32.78

Table 11. **Per-scene SSIM metrics of Mip-NeRF360 dataset.**

Method | Scenes	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
Plenoxels	0.496	0.431	0.6063	0.523	0.509	0.8417	0.759	0.648	0.814
Instant-NGP	0.512	0.486	0.701	0.594	0.542	0.871	0.817	0.858	0.906
Mip-NeRF360	0.685	0.584	0.809	0.745	0.631	0.910	0.892	0.917	0.938
Scaffold-GS*	0.755	0.590	0.858	0.765	0.639	0.923	0.910	0.926	0.943
Octree-GS*	0.759	0.597	0.864	0.763	0.643	0.924	0.907	0.916	0.930
3D-GS	0.771	0.605	0.868	0.775	0.638	0.914	0.905	0.922	0.938
AbsGS	0.783	0.623	0.871	0.780	0.617	0.925	0.911	0.929	0.945
Pixel-GS*	0.775	0.633	0.867	0.784	0.630	0.920	0.911	0.929	0.944
Mini-Splatting-D	0.798	0.642	0.878	0.804	0.640	0.928	0.913	0.934	0.946
Ours-Small (AbsGS)	0.792	0.634	0.872	0.802	0.652	0.929	0.914	0.932	0.945
Ours (3D-GS)	0.790	0.639	0.874	0.800	0.650	0.929	0.916	0.934	0.947
Ours (AbsGS)	0.797	0.647	0.876	0.804	0.645	0.931	0.918	0.934	0.948

Table 12. **Per-scene LPIPS metrics of Mip-NeRF360 dataset.**

Method | Scenes	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
Plenoxels	0.506	0.521	0.3864	0.503	0.540	0.4186	0.441	0.447	0.398
Instant-NGP	0.446	0.441	0.257	0.421	0.450	0.261	0.306	0.195	0.205
Mip-NeRF360	0.301	0.344	0.170	0.261	0.339	0.211	0.204	0.127	0.176
Scaffold-GS*	0.233	0.352	0.120	0.236	0.331	0.216	0.203	0.130	0.207
Octree-GS*	0.225	0.340	0.108	0.233	0.310	0.203	0.202	0.141	0.219
3D-GS	0.205	0.336	0.103	0.210	0.317	0.220	0.204	0.129	0.205
AbsGS	0.171	0.270	0.100	0.195	0.278	0.200	0.189	0.121	0.190
Pixel-GS*	0.182	0.263	0.100	0.186	0.280	0.213	0.185	0.120	0.193
Mini-Splatting-D	0.158	0.255	0.090	0.169	0.262	0.190	0.172	0.114	0.175
Ours-Small (AbsGS)	0.175	0.290	0.099	0.188	0.279	0.196	0.184	0.117	0.189
Ours (3D-GS)	0.168	0.295	0.092	0.179	0.280	0.192	0.177	0.114	0.185
Ours (AbsGS)	0.156	0.251	0.089	0.170	0.250	0.187	0.172	0.112	0.179

Table 13. **Per-scene PSNR metrics of Tank&Temples and Deep Blending dataset.**

Method | Scenes	Truck	Train	Dr Johnson	Playroom
Plenoxels	23.22	18.93	23.14	22.98
Instant-NGP	23.38	20.46	28.26	21.67
Mip-NeRF360	24.91	19.52	29.14	29.66
Scaffold-GS	25.77	22.15	29.80	30.62
Octree-GS	26.27	22.77	29.87	30.95
3D-GS	25.19	21.10	28.77	30.04
AbsGS	25.74	21.72	29.20	30.14
Pixel-GS	25.49	22.13	28.02	29.79
Mini-Splatting-D	25.43	21.04	29.32	30.43
Ours-Small (AbsGS)	26.07	22.40	29.64	30.38
Ours (3D-GS)	26.08	22.65	29.54	30.32
Ours (AbsGS)	26.14	22.61	29.51	30.32

Table 14. **Per-scene SSIM metrics of Tank&Temples and Deep Blending dataset.**

Method | Scenes	Truck	Train	Dr Johnson	Playroom
Plenoxels	0.774	0.663	0.787	0.802
Instant-NGP	0.800	0.689	0.854	0.779
Mip-NeRF360	0.857	0.660	0.901	0.900
Scaffold-GS	0.883	0.822	0.907	0.904
Octree-GS	0.896	0.835	0.912	0.914
3D-GS	0.879	0.802	0.899	0.906
AbsGS	0.888	0.818	0.898	0.907
Pixel-GS	0.883	0.823	0.885	0.899
Mini-Splatting-D	0.890	0.817	0.905	0.908
Ours-Small (AbsGS)	0.889	0.834	0.905	0.907
Ours (3D-GS)	0.890	0.840	0.901	0.905
Ours (AbsGS)	0.893	0.841	0.900	0.905

Table 15. **Per-scene LPIPS metrics of Tank&Temples and Deep Blending dataset.**

Method | Scenes	Truck	Train	Dr Johnson	Playroom
Plenoxels	0.335	0.422	0.521	0.499
Instant-NGP	0.249	0.360	0.352	0.428
Mip-NeRF360	0.159	0.354	0.237	0.252
Scaffold-GS	0.147	0.206	0.250	0.258
Octree-GS	0.118	0.198	0.231	0.244
3D-GS	0.148	0.218	0.244	0.241
AbsGS	0.132	0.193	0.241	0.233
Pixel-GS	0.122	0.180	0.258	0.243
Mini-Splatting-D	0.100	0.181	0.218	0.204
Ours-Small (AbsGS)	0.124	0.177	0.233	0.235
Ours (3D-GS)	0.118	0.164	0.232	0.231
Ours (AbsGS)	0.106	0.159	0.231	0.223

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5835–5844, 2021. 1, 2, 3

[2] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Revising densification in gaussian splatting. *ArXiv*, abs/2404.06109, 2024. 2

[3] Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. *International Conference on Machine Learning (ICML)*, 2024. 1

[4] Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*, 2024. 2, 3

[5] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel J. Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)*, 37:1 – 15, 2018. 1

[6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42:1 – 14, 2023. 1, 2, 3

[7] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples. *ACM Transactions on Graphics (TOG)*, 36:1 – 13, 2017. 1

[8] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20654–20664, 2024. 2, 3

[9] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41:1 – 15, 2022. 3

[10] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2024. 2, 3

[11] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[12] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott M. Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2443–2451, 2020. 1

[13] Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details in 3d gaussian splatting. *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024. 2, 3

[14] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5491–5500, 2022. 3

[15] Jiahui Zhang, Fangneng Zhan, Muyu Xu, Shijian Lu, and Eric P. Xing. Fregs: 3d gaussian splatting with progressive frequency regularization. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21424–21433, 2024. 2, 3

[16] Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In *European Conference on Computer Vision*, 2024. 2, 3