Supplementary Materials

Xiangrui Liu*
City University of Hong Kong
Hong Kong, China
xiangrliu3-c@my.cityu.edu.hk

Xinju Wu*
City University of Hong Kong
Hong Kong, China
xinjuwu2-c@my.cityu.edu.hk

Pingping Zhang
City University of Hong Kong
Hong Kong, China
ppingyes@gmail.com

Shiqi Wang†
City University of Hong Kong
Hong Kong, China
shiqwang@cityu.edu.hk

Zhu Li
University of Missouri-Kansas City
Kansas City, Missouri, USA
zhu.li@ieee.org

Sam Kwong
Lingnan University
Hong Kong, China
samkwong@ln.edu.hk

1 Per-scene Results on Evaluation Datasets

We provide per-scene evaluation results between our method and comparison methods [2, 4, 6–8] on the three used datasets. We focus on scene-by-scene comparisons to evaluate the effectiveness of the proposed CompGS in various scenarios.

Specifically, Table 1 illustrates the comparison results on two scenes, namely *Train* and *Truck*. It can be observed that our method attains 29.93× compression ratio on the *Train* scene, and achieves more than 37.59% size reduction as compared to the most advanced compression method [8]. Similarly, the proposed CompGS demonstrates superior compression performance on the *Truck* scene, attaining a compression ratio of 57.62× while maintaining comparable rendering quality.

Additionally, Table 2 presents the results on the Deep Blending dataset [3]. For the *DrJohnson* scene, our approach successfully reduces the data size from 782.10 MB to 8.21 MB without compromising visual quality. Meanwhile, our method achieves significant compression advancements on the *Playroom* scene, exhibiting a compression ratio ranging from 76.91× to 108.67× and decreasing the data size to under 8 MB from the original 549.88 MB.

Moreover, Table 3 depicts the evaluation results on the Mip-NeRF 360 dataset [1]. Our framework achieves a remarkable average compression ratio of up to 89.35×. Specifically, the proposed CompGS exhibits substantial bitrate reductions for scenes such as *Bicycle* and *Garden*, lowering the size from 1431.12 MB and 1383.79 MB to 21.74 MB and 28.66 MB, respectively, while preserving rendering quality. It is noteworthy that for the *Stump* scene, our proposed CompGS demonstrates exceptional performance with a compression ratio of 175.20×. This might be attributed to the inherent local similarities within this particular scene. For the scenes that have smaller sizes such as the *Room*, *Counter*, and *Bonsai* scenes with sizes ranging from 285.04 MB to 366.62 MB, our method still achieves a compression ratio of 41.43×, 29.66×, and 25.25×, respectively.

The per-scene evaluation demonstrates the versatility and efficacy of the proposed CompGS across various scenes on the prevalent datasets [1, 3, 5]. Our method consistently achieves superior compression ratios compared to existing techniques [2, 4, 6–8], highlighting its potential for real-world applications.

Table 1: Per-scene Results on the Tanks&Templates dataset [5].

Methods	Train			
	PSNR (dB)	SSIM	LPIPS	Size (MB)
Kerbl et al. [4]	22.02	0.81	0.21	257.44
Navaneet et al. [7]	21.63	0.80	0.22	27.54
Niedermayr et al. [8]	21.92	0.81	0.22	13.78
Lee et al. [6]	21.69	0.80	0.24	37.38
Girish et al. [2]	21.68	0.80	0.23	24.67
Proposed	22.12	0.80	0.23	8.60
	21.82	0.80	0.24	6.72
	21.49	0.78	0.26	5.51
Methods	Truck			
	PSNR (dB)	SSIM	LPIPS	Size (MB)
Kerbl et al. [4]	25.41	0.88	0.15	611.31
Navaneet et al. [7]	25.04	0.88	0.16	66.48
Niedermayr et al. [8]	25.24	0.88	0.15	21.51
Lee et al. [6]	25.10	0.87	0.16	41.55
Girish et al. [2]	25.10	0.87	0.17	42.46
Proposed	25.28	0.87	0.18	10.61
	24.97	0.86	0.20	7.82
	24.72	0.85	0.21	6.27

2 Other Experiments

The proposed CompGS leverages G-PCC to compress locations of anchor primitives. We have experimentally found that directly storing uncompressed locations incurs a size increase of 0.40 MB, highlighting the rationality of the proposed scheme. Additionally, we devise a variant named "PCC-based" to demonstrate the superiority of the proposed method against existing point cloud compression methods. Specifically, we employ the advanced GRASP-Net [9] for positions and G-PCC for other attributes of Gaussian primitives. As illustrated in Table **??**, our method outperforms this variant, owing to the inter-primitive prediction and rate-constrained optimization fully tailored for Gaussian primitives.

*Both authors contributed equally to this research.
†Corresponding author.

Table 2: Per-scene Results on the Deep Blending dataset [3].

Methods	DrJohnson			
	PSNR (dB)	SSIM	LPIPS	Size (MB)
Kerbl et al. [4]	29.14	0.90	0.24	782.10
Navaneet et al. [7]	29.34	0.90	0.25	85.10
Niedermayr et al. [8]	29.03	0.90	0.25	27.86
Lee et al. [6]	29.26	0.90	0.25	48.11
Girish et al. [2]	29.52	0.91	0.24	80.09
Proposed	29.33	0.90	0.27	10.38
	29.21	0.90	0.27	8.21
	28.99	0.90	0.28	7.00
Methods	Playroom			
	PSNR (dB)	SSIM	LPIPS	Size (MB)
Kerbl et al. [4]	29.94	0.91	0.24	549.88
Navaneet et al. [7]	30.43	0.91	0.24	59.82
Niedermayr et al. [8]	29.86	0.91	0.25	19.88
Lee et al. [6]	30.38	0.91	0.25	38.17
Girish et al. [2]	30.27	0.91	0.25	43.28
Proposed	30.04	0.90	0.29	7.15
	29.59	0.89	0.30	5.43
	29.61	0.89	0.31	5.06

Table 3: Per-scene Results on the Mip-NeRF 360 dataset [1].

Methods	Bicycle				Flowers				Garden			
	PSNR (dB)	SSIM	LPIPS	Size (MB)	PSNR (dB)	SSIM	LPIPS	Size (MB)	PSNR (dB)	SSIM	LPIPS	Size (MB)
Kerbl et al. [4]	25.18	0.77	0.21	1431.12	21.55	0.61	0.34	858.08	27.39	0.87	0.11	1383.79
Navaneet et al. [7]	24.99	0.76	0.23	158.14	21.29	0.59	0.35	92.93	27.05	0.86	0.12	153.50
Niedermayr et al. [8]	24.99	0.75	0.23	46.77	21.27	0.59	0.35	31.19	26.98	0.85	0.14	46.63
Lee et al. [6]	24.82	0.74	0.25	64.32	21.07	0.57	0.38	51.23	26.89	0.84	0.14	63.41
Girish et al. [2]	24.80	0.74	0.25	102.32	21.11	0.58	0.37	63.83	26.81	0.84	0.15	76.21
Proposed	24.70	0.74	0.26	21.74	21.31	0.58	0.35	25.44	27.45	0.85	0.13	28.66
	24.42	0.72	0.28	15.05	21.18	0.57	0.37	17.27	27.07	0.84	0.15	17.73
	24.21	0.71	0.30	12.30	20.97	0.55	0.39	12.85	26.74	0.83	0.17	15.41
	Stump				Tree Hill				Room			
	PSNR (dB)	SSIM	LPIPS	Size (MB)	PSNR (dB)	SSIM	LPIPS	Size (MB)	PSNR (dB)	SSIM	LPIPS	Size (MB)
Kerbl et al. [4]	26.56	0.77	0.22	1149.30	22.49	0.63	0.33	893.52	31.44	0.92	0.22	366.62
Navaneet et al. [7]	26.53	0.77	0.23	126.12	22.50	0.63	0.34	97.24	31.05	0.91	0.23	39.01
Niedermayr et al. [8]	26.31	0.76	0.25	39.88	22.45	0.62	0.35	33.24	31.15	0.91	0.23	14.67
Lee et al. [6]	26.28	0.76	0.26	56.63	22.59	0.63	0.34	61.29	30.76	0.91	0.23	34.26
Girish et al. [2]	26.44	0.76	0.24	104.52	22.56	0.63	0.35	78.21	31.52	0.92	0.23	36.13
Proposed	26.24	0.75	0.26	12.02	23.11	0.64	0.33	20.02	30.85	0.91	0.23	8.85
	26.05	0.74	0.28	8.23	23.05	0.64	0.35	12.63	30.14	0.90	0.25	6.76
	25.78	0.72	0.30	6.56	22.94	0.62	0.37	10.23	29.88	0.89	0.26	5.58
	Counter				Kitchen				Bonsai			
	PSNR (dB)	SSIM	LPIPS	Size (MB)	PSNR (dB)	SSIM	LPIPS	Size (MB)	PSNR (dB)	SSIM	LPIPS	Size (MB)
Kerbl et al. [4]	28.99	0.91	0.20	285.04	31.34	0.93	0.13	426.82	32.16	0.94	0.20	297.50
Navaneet et al. [7]	28.24	0.90	0.22	30.48	30.53	0.92	0.14	45.82	31.21	0.93	0.22	31.65
Niedermayr et al. [8]	28.69	0.90	0.21	13.58	30.75	0.92	0.14	18.46	31.48	0.93	0.21	13.08
Lee et al. [6]	28.60	0.90	0.22	34.49	30.50	0.92	0.14	45.36	31.92	0.93	0.22	35.40
Girish et al. [2]	28.32	0.90	0.22	31.73	30.40	0.92	0.14	56.64	31.38	0.93	0.22	36.22
Proposed	29.09	0.90	0.22	9.61	30.82	0.92	0.15	10.39	31.80	0.93	0.22	11.78
	28.40	0.89	0.24	6.96	30.00	0.90	0.16	6.88	30.76	0.92	0.24	7.63
	27.73	0.87	0.26	5.41	29.29	0.89	0.18	5.55	29.79	0.91	0.26	5.56

Table 4: Performance comparison on the Tanks&Temples dataset

	PSNR (dB)	SSIM	LPIPS	Size (MB)
PCC-based	17.05	0.52	0.44	43.90
Proposed	22.12	0.80	0.23	8.60

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-aliased Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5470–5479.

[2] Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. 2023. EAGLES: Efficient Accelerated 3D Gaussians with Lightweight Encodings. *arXiv preprint arXiv:2312.04564* (2023), 1–10.

[3] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep Blending for Free-viewpoint Image-based Rendering. *ACM Transactions on Graphics* 37, 6 (2018), 1–15.

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (2023), 1–14.

[5] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 2017. Tanks and Temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics* 36, 4 (2017), 1–13.

[6] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. 2023. Compact 3D Gaussian Representation for Radiance Field. *arXiv preprint arXiv:2311.13681* (2023), 1–10.

[7] KL Navaneet, Kossar Pourahmadi Meibodi, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. 2023. Compact3D: Compressing Gaussian Splat Radiance Field Models with Vector Quantization. *arXiv preprint arXiv:2311.18159* (2023), 1–12.

[8] Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. 2023. Compressed 3D Gaussian Splatting for Accelerated Novel View Synthesis. *arXiv preprint arXiv:2401.02436* (2023), 1–10.

[9] Jiahao Pang, Muhammad Asad Lodhi, and Dong Tian. 2022. GRASP-Net: Geometric Residual Analysis and Synthesis for Point Cloud Compression. In *Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing and Analysis*. 11–19.