Smaller and Faster 3DGS via Post-Training Dictionary Learning

J. Gong[1], J. Unger[1] and E. Miandji[1]

[1]Linköping University, Sweden

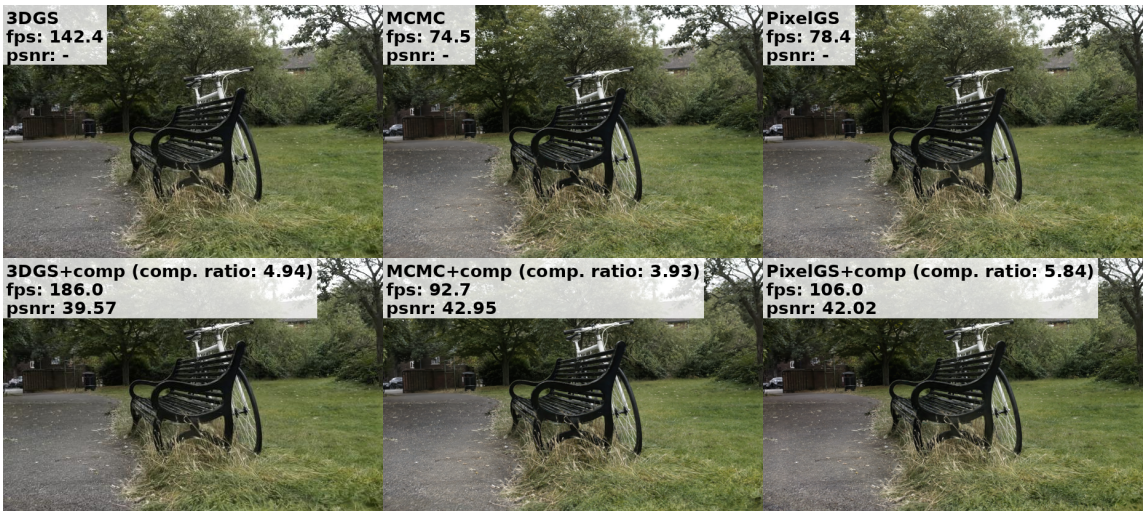

Figure 1: *Visual comparison of baseline methods (3DGS, MCMC, and PixelGS, top row) and their compressed counterparts (bottom row). Each panel in bottom row reports the corresponding compression ratio, rendering speed (FPS), and PSNR, highlighting that our dictionary-learning-based compression substantially reduces model size and improves rendering efficiency while preserving visual fidelity.*

Abstract

*3D Gaussian Splatting (3DGS) suffers from large memory footprints. Existing compression techniques often lead to architectures with several additional trainable parameters and noticeable drops in rendering. We introduce the first dictionary-learning-based compression framework for 3DGS. Our compression framework is straightforward to implement, yet provides significant compression capabilities, preserves image quality, and improves real-time rendering performance. Across 13 benchmark scenes, our approach achieves an average compression ratio of **3.95**×, **3.10**×, and **4.55**× when applied to 3DGS, 3DGS-MCMC, and PixelGS, respectively. This yields consistent rendering speedups of **23.3%**, **24.3%**, and **25.3%**, while maintaining image quality.*

CCS Concepts

• ***Computing methodologies*** → *Rendering; Image-based rendering;*

1. Introduction

3D Gaussian Splatting (3DGS) [KKLD] represents radiance fields using anisotropic Gaussians. The excessive memory costs often hinders its adoption for complex scenes. As such, 3DGS compression has become an active research field [GGS, NMKP]. For instance, Navaneet et al. [NMKP] quantize Gaussian parameters with K-means codebooks and reduce the number of Gaussians of a 3D model via opacity penalization and pruning. However, existing pipelines give rise to a drop in image quality and often need quantization-aware-training or per-scene optimization. In this paper, we propose a fundamentally different approach via a dictionary-learning-based compression pipeline that applies compression as a post-training process, hence enabling compression of existing SOTA 3DGS models with virtually no noticeable degradation in image quality.

Since each 3D Gaussian is described by a set of attributes, namely position, covariance, Spherical Harmonics (SH) coefficients for color, and opacity. SH coefficients account for over 80% of the storage cost, we, thus, train a dictionary based on SH coefficients, followed by Orthogonal Matching Pursuit (OMP) [PRK] to obtain sparse codes. We show that the radiance field reconstruction, e.g. during rendering, can be directly evaluated using the sparse co-

DOI: 10.2312/egp.20261016

Table 1: Mean per-scene PSNR (dB), SSIM, LPIPS, and FPS of compressive renderings relative to the original for tol = 0.10. Higher PSNR/SSIM and lower LPIPS indicate closer agreement with the original. "FPS(-Comp)" denotes the rendering speed of the original baselines.

Method	Metric	Bicycle	Bonsai	Counter	Garden	Kitchen	Room	Stump	Truck	Flowers	Playroom	Train	Treehill	Drjohnson	**Mean**
3DGS+comp	PSNR	38.54	41.03	40.69	37.97	39.59	41.20	38.51	39.34	37.80	40.03	39.64	39.18	40.84	**39.57**
	SSIM	0.9830	0.9863	0.9818	0.9810	0.9851	0.9829	0.9819	0.9859	0.9870	0.9821	0.9873	0.9829	0.9871	**0.9842**
	LPIPS	0.0301	0.0264	0.0263	0.0296	0.0217	0.0301	0.0392	0.0210	0.0257	0.0260	0.0210	0.0311	0.0311	**0.0276**
	Comp Ratio	4.94	4.65	3.76	3.16	3.36	4.91	4.15	3.92	3.40	3.56	3.39	3.76	4.33	**3.95**
	FPS	186.0	268.9	186.7	223.7	158.8	186.9	197.1	225.9	305.7	203.2	193.2	203.7	173.4	**208.7**
	FPS(-Comp)	142.4	240.0	170.5	188.4	143.9	167.7	147.7	193.4	250.4	177.7	178.6	163.6	136.2	**169.3**
MCMC+comp	PSNR	43.20	43.09	43.56	42.85	42.64	43.64	44.21	42.51	43.41	42.17	42.29	43.22	41.57	**42.95**
	SSIM	0.9974	0.9951	0.9947	0.9969	0.9964	0.9944	0.9966	0.9969	0.9974	0.9946	0.9970	0.9967	0.9952	**0.9961**
	LPIPS	0.0094	0.0067	0.0065	0.0080	0.0060	0.0056	0.0126	0.0051	0.0076	0.0045	0.0062	0.0096	0.0109	**0.0076**
	Comp Ratio	3.93	3.22	2.67	2.85	2.64	3.72	2.80	3.29	2.25	2.54	3.43	3.38	3.56	**3.10**
	FPS	92.7	182.3	125.5	101.5	129.7	142.5	92.1	148.8	74.1	182.3	216.5	85.3	253.2	**140.5**
	FPS(-Comp)	74.5	167.1	118.5	82.9	118.0	132.2	78.4	127.7	67.6	157.1	196.5	72.4	177.6	**113.0**
PixelGS+comp	PSNR	42.41	42.18	42.73	41.97	41.85	42.11	42.78	41.65	42.24	41.20	41.50	42.00	41.61	**42.02**
	SSIM	0.9954	0.9932	0.9914	0.9950	0.9948	0.9913	0.9943	0.9952	0.9958	0.9915	0.9958	0.9945	0.9940	**0.9940**
	LPIPS	0.0123	0.0115	0.0101	0.0104	0.0086	0.0152	0.0164	0.0078	0.0106	0.0114	0.0084	0.0140	0.0164	**0.0118**
	Comp Ratio	5.84	5.13	4.40	3.56	3.69	5.53	4.36	4.56	3.59	4.45	4.29	4.18	5.59	**4.55**
	FPS	106.0	216.0	123.9	138.0	129.4	158.5	116.8	117.7	114.4	152.5	119.9	97.3	134.5	**132.7**
	FPS(-Comp)	78.4	183.5	106.9	105.6	107.5	133.0	88.1	93.1	89.6	122.9	99.0	73.8	95.3	**105.9**

efficients to obtain directional radiance values. Given the inherent sparsity introduced by our dictionary, our method achieves a higher rendering speed, while reducing the memory footprint and preserving the image quality of a given 3DGS model.

2. Method

We briefly recall the dictionary learning framework, which underpins our compression method. Given a data matrix $X = [x_1, x_2, \ldots, x_N] \in R^{d\times N}$, the goal is to find an overcomplete dictionary $D = [a_1, a_2, \ldots, a_N] \in R^{d\times m}$, where $m > d$, and sparse codes $\alpha_i \in R^m$, $i \in \{1, \ldots, N\}$, such that

$$\min_{D,\{\alpha_i\}} \sum_{i=1}^{N} \|x_i - D\alpha_i\|_2^2, \quad \text{s.t. } \|\alpha_i\|_0 \le k,\ \|a_j\|_2 = 1,\ \forall j. \qquad (1)$$

The dictionary learning problem is typically solved approximately via an alternating minimization strategy, whereby the sparse codes $\{\alpha_i\}$ and the dictionary D are updated in turn while keeping the other fixed. Having the dictionary D, and given a new test set, the task is then to obtain the most sparse coefficients while minimizing the error. In our framework, sparse coefficients are achieved via OMP, which efficiently selects the most relevant atoms from the dictionary until either/both a target sparsity k or/and error threshold (tolerance) ϵ is reached. Since $k \ll m$, it has led to several applications in compression, see e.g. [MHU19].

3. Experiments

To quantify the rendering fidelity and efficiency of our method, Table 1 reports PSNR, SSIM, Lpips, and FPS means over each scene for the compressed renderings of each pipeline *3DGS*, *MCMC*, and *PixelGS* at tolerance 0.1, against the corresponding original SOTA baseline renderings. From Table 1, it is evident that our method preserves rendering quality since the minimum PSNR value is 37.80 dB and the maximum PSNR 44.21 dB; from **Metric** "Comp Ratio" and "FPS", we can also easily find that across all frameworks, our method consistently reduces memory usage and improves rendering efficiency. More concretely, when our compressive framework is applied on baselines, our method improves the rendering speed, with average FPS gains of 23.3%, 24.3%, and 25.3%. Meanwhile, the SH coefficients achieve significant compression, with mean ratios of 3.95×, 3.10×, and 4.55×, respectively.

4. Conclusion

Overall, this study firstly highlights dictionary learning as a powerful tool for balancing compression, speed, and visual fidelity in 3DGS, and lays the groundwork for future explorations at the intersection of dictionary learning and real-time rendering.

References

[GGS] GIRISH S., GUPTA K., SHRIVASTAVA A.: Eagles: Efficient accelerated 3d gaussians with lightweight encodings. In *European Conference on Computer Vision*. 1

[KKLD] KERBL B., KOPANAS G., LEIMKÜHLER T., DRETTAKIS G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* 1

[MHU19] MIANDJI E., HAJISHARIF S., UNGER J.: A unified framework for compression and compressed sensing of light fields and light field videos. URL: https://doi.org/10.1145/3269980, doi:10.1145/3269980. 2

[NMKP] NAVANEET K., MEIBODI K. P., KOOHPAYEGANI S. A., PIRSIAVASH H.: Compact3d: Smaller and faster gaussian splatting with vector quantization. 1

[PRK] PATI Y. C., REZAIIFAR R., KRISHNAPRASAD P. S.: Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*. 1