Out-of-core Multi-resolution Gaussian Splatting Optimization Pipeline for Large Scale Scene Reconstruction

Shiu Ka Heng

4th Year Project Report
Artificial Intelligence
School of Informatics
University of Edinburgh

2023

Abstract

Gaussian splatting enables high-quality view synthesis but struggles with large-scale reconstructions due to memory constraints. We propose an optimization pipeline that partitions the problem into manageable segments, stored out-of-core. Our approach first runs structure-from-motion on the entire scene. We then cluster cameras by distance to decompose the problem into discrete levels of detail. Optimization occurs coarsely, then finer grains are recursively split until convergence. Partitioned segments are merged into a coherent octree structure with inherent multi-resolution properties.

This pioneering framework pushes reconstructions to unprecedented scales previously infeasible. We comprehensively analyze limitations, recursively split/merge optimizations, develop interoperable infrastructure, and generate visualizations. Extensions like NeRF integration and LOD viewers are promising future work. In summary, our out-of-core gaussian splatting unlocks reconstructions at scales never before possible, overcoming barriers via principled partitioning. We enable scenes orders of magnitude larger, unlocking new frontiers in graphics and vision.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Shiu Ka Heng*)

Acknowledgements

TBD

Table of Contents

Chapter 1

Introduction

Gaussian splatting (Kerbl et al. [2023]) is currently the state-of-the-art in the field of view synthesis, offering unmatched reconstruction quality and training time. However, we find its application to scene reconstruction is constrained by memory limitations, especially for larger scenes.

To address these challenges, we introduce an out-of-core pipeline that integrates Gaussian splatting with established acceleration structure libraries (Rusu and Cousins [2011]) to enable massive reconstructions previously impossible to fully fit into memory. Our approach dynamically divides the optimization process, facilitating the reconstruction of expansive, multi-scale scenes, and simultaneously generates multiple Level of Detail (LOD) that could be used for subsequent LOD rendering.

Chapter 2

Background

Volumetric rendering and view synthesis have been popular in graphics for decades (Kang [1998]). Recently, learned volumetric representations like NeRF (Mildenhall et al. [2020]) have enabled high quality view synthesis, but with performance and quality tradeoffs (Gao et al. [2022], Barron et al. [2021]). More recently, gaussian splatting (Kerbl et al. [2023]) has shown that combining classical graphics primitives with learned representations can provide both performance and realism.

However, scaling the gaussian splatting training algorithm to reconstruct larger scenes poses challenges, primarily because of the memory overhead required to accommodate the spatial information parameters. Current techniques necessitate the optimization of all parameters concurrently (Kerbl et al. [2023]), leading to a surge in memory requirements. These constraints hinder the broader application and accessibility of the technology.

While there is no literature yet on effectively scaling gaussian splatting for vast scenes, there's a wealth of research on upscaling other forms of 3D reconstructions, which might offer insights into the scaling problem.

2.1 Introduction to Gaussian Splatting

Gaussian splatting combines classical computer graphics and contemporary machine learning techniques to optimize a set of points into 3D view-dependent gaussians using a SFM as input. Rooted in point-based rendering, this method aspires to streamline the synthesis process, paving the way for quicker renderings without compromising on the quality.

2.1.1 Gaussian Splatting Training Process

2.1.1.1 Initialization from Sparse SfM Points

Gaussian splatting takes off from a sparse Structure from Motion (SfM) point cloud. This set of essential points, capturing the essence of a scene's architecture, becomes the basis for the formation of 3D Gaussians. Currently, the official implementation uses

COLMAP (Fisher et al. [2021]) for SfM, an open source SfM library that is widely used in the research community.

2.1.1.2 3D Gaussian Optimization

Using the initial SfM, we extract the camera positions and the 3D points, and then optimize the 3D points into 3D gaussians using a differentiable renderer. The renderer takes in the camera positions, the 3D points, and the 3D gaussians, and renders the image. The loss function is then computed by comparing the rendered image with the ground truth image using a mix of L1 and D-SSIM loss (a form of perceptual loss proposed by Baker et al. [2022]), tunable by a hyperparameter. The loss is then backpropagated through the renderer to optimize the 3D gaussians.

2.1.1.3 Adaptive Control of Gaussians

In addition to simply optimizing the 3D gaussians, Kerbl et al. [2023] also proposes a technique to adaptively control the density of the gaussians, by splitting, merging, or removing gaussians depending on the posittional gradient of each individual splat. This allows for the dynamic addition and removal of gaussians, which is essential since the starting SfM points are sparse.

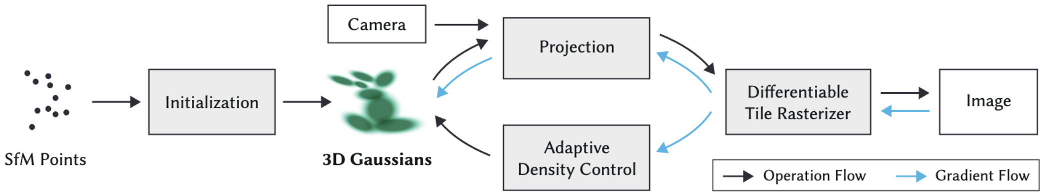

Figure 2.1: A diagram illustrating the Gaussian Splatting process from Kerbl et al. [2023]

2.2 Scaling NeRFs: Related work

2.2.1 Splitting NeRFs

Neural radiance fields (Mildenhall et al. [2020]) have been the state-of-the-art in view synthesis prior to gaussian splatting, and numerous works exist that try to scale the technique for larger scenes.

To address the same issue we are encountering regarding the memory limit, multiple smaller spatially segmented models have been used to represent scenes. In Block-NeRF (Tancik et al. [2022]), which specifically aims to solve the issue of applying the NeRF technique in city scenes, the reconstruction space heuristically using city block boundaries. However its application is domain specific to the reconstruction of multiple city blocks. Turki et al. [2021] later proposed using a geometric clustering algorithm to dynamically partition a scene into multiple smaller NeRF models by analyzing visibility statistics.

Inspired by the above techniques, we see that gaussian splatting could similarly benefit in being segmented into smaller models to make the reconstruction of larger scenes feasible. Furthermore, model segmentation in gaussian splatting should be even more straightforward because the parameters responsible of modeling the appearance of scenes can be spatially interpreted, and thus allows for directly segmenting gaussian splatting models after training.

2.2.2 NeRFs for multi-scale reconstruction

Other than just the practicality of scaling NeRFs to accommodate for computational power and memory limits, large scale scene reconstruction in learned representations also needs to consider the varying appearance of objects from different distances. Images of an object taken at a short distance may exhibit a high amount of detail, whereas images taken from further distances may have the object appear blurry, due to the finite amount of pixels in images. In the realm of NeRF training, Xiangli et al. [2021] has proposed a coarse-to-fine training schedule, as well as a dynamically growing NeRF model in the form of residual blocks to allow for the progressive learning of higher detail at closer distances.

While gaussian splatting utilizes a different architecture to represent the scene, optimizing one model over images of the same object at drastically different scales may see similar issues, and thus this problem will also be something we put into consideration in the design of our technique.

2.3 Scaling Structure from Motion: Related work

Another similar problem would be the scaling of Structure from Motion (SfM) to larger datasets, which similarly may run into computational limits when scaled naively.

As a more mature field of research, numerous techniques have been explored in this space. Steedly et al. [2003] proposed a novel technique that utilizes the Hessian of reprojection error and its eigenvectors to effectively factor the optimization into smaller problems, showing a theoretical interpretation on camera group clustering for the problem of SfM, and a technique that offers low residuals and can be efficiently computed. This clustering technique could potentially be applied in our optimization pipeline to quickly and effectively segment out camera camera groups.

Another paper, Lu et al. [2019], explored dividing the problem into multiple groups based on the image adjacency matrix, demonstrating that the problem could be divided into smaller parallelizable problems.

Other works (Farenzena et al. [2009], Sun and Tao [2018], Bhowmick et al. [2017]) have hierachically splitting nested groups and then merging them back up in a bottom-up fashion, which have a provably lower computational complexity (Farenzena et al. [2009]).

Chapter 3

Proposed methods

Given the variety of approaches to scaling NeRFs and SfM, in the exploratory stage, we should explore multiple approaches to scaling gaussian splatting training. The overarching theme is to divide the problem into smaller, more manageable problems, and then merging them back up to form a complete reconstruction.

However, it remains to be explored:

- How do we partition the problem into smaller problems?
- How do we merge the smaller problems back up?
- Should we partition recursively, or in a single pass?
- How do we ensure the boundaries of the smaller problems are consistent?

Preliminarily, I am interested in recursive partitioning, because it may allow for the simultaneous generation of LOD models, which will be useful for subsequent LOD rendering. Moreover, we may dynamically change the depth of the recursive partitioning depending the computer's memory limit, such that even arbritarily large scenes could eventually be split into smaller problems that could fit into memory.

Drawing inspiration from Xiangli et al. [2021], we would also like to incorporate distance awareness in the process to model the varying appearance of objects at different distances more effectively, which will be useful for reconstructing multi-scale scenes. This requirement also fits well with the recursive partitioning approach, since we may utilize the distance of the camera to the object as part of the partitioning criteria to weigh coarser views more heavily for top level partitions, and closer views more heavily for lower level partitions. We are also drawing inspiration from Steedly et al. [2003] to segment scenes with scene geometry in mind, which will reduce the amount of overlap between partitions and reduce artifacts.

Factoring in these considerations, here is a preliminary outline of building a pipeline we will explore. An overview is shown in Figure 3.1, but below is a more detailed description of certain steps of the pipeline.

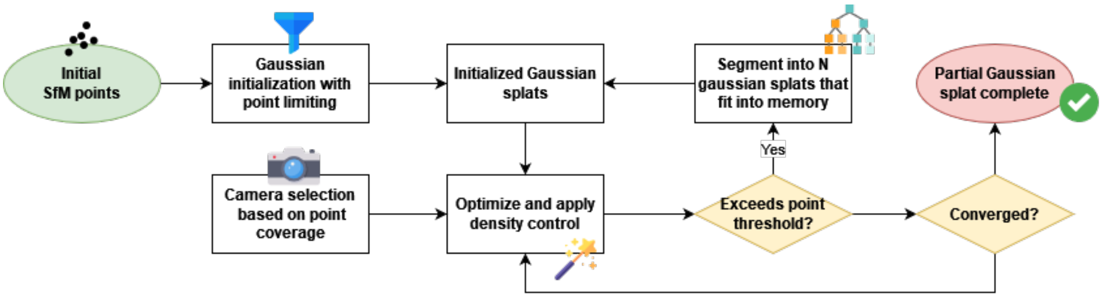
Initial SfM points
Gaussian initialization with point limiting
Initialized Gaussian splats
Segment into N gaussian splats that fit into memory
Partial Gaussian splat complete
Camera selection based on point coverage
Optimize and apply density control
Exceeds point threshold?
Yes
Converged?

Figure 3.1: A flow diagram illustrating the proposed Gaussian Splatting LOD training process

3.0.1 SfM, Gaussian Splatting Initialization, and Camera Selection

We assume SfM for the scene has already been performed, and thus we have a sparse point cloud of the scene, the camera parameters, and correspondances between the cameras and the points. We then initialize the gaussian splatting model using the sparse point cloud. However, unlike the original initialization process, we need to downsample the points to a size that would fit into memory. Moreover, we need to reduce the number of cameras to an amount that may still fit into memory, yet maximize the overall coverage of the downsampled points. As a side-effect, this reduces the amount of overlap and likely select cameras of the same scale.

3.0.2 Gaussian Splatting Optimization and Paritioning

Once we have the initial gaussian splatting model, we may then optimize the model using the same optimization process as the original gaussian splatting paper. However, since the optimization algorithm automatically grows the number of gaussians (and is likely to grow given we just downsampled the points), we will need to set a threshold for the maximum number of gaussians allowed in the model. Once the number of gaussians exceeds the threshold, we will need to partition the model into smaller models.

3.0.3 Recursive Partitioning

In partitioning the model into smaller models, we would desire the following properties:

- The smaller models are of similar size, and thus can more effectively fit into memory
- The smaller models' camera coverage of points should be orthogonal to each other, such that the smaller models can be merged back up without artifacts

We suggest representing cameras as a graph, where each camera is a node, and the edges between the nodes represent the amount of overlap between the cameras. We may then use the spectral graph partitioning algorithm with balanced partitioning constraints to partition the cameras into smaller groups (see Patil and Kulkarni [2021]).

3.1 Efficient algorithms for dealing with large spatial datasets

Implementationally, using such techniques for gaussian splatting reconstructions pose unique challenges, because in order to train gaussian splatting models, it needs to fit on the GPU; on the other hand, the process of splitting gaussian splats may be more suitable for the CPU due to the complexities of running data structures required to effectively filter and segment large gaussian splats.

Moreover, the merged results of multiple gaussian splatting reconstruction may not fit in GPU memory, and will need to be stored either on the CPU or on disk. Fortunately, gaussian splats could be easily represented as a regular point cloud with extra attributes, and for the task of manipulating and storing massive gaussian splats that could not fit on memory, we can simply leverage the mature field of point cloud research for existing tools (Rusu and Cousins [2011], Zampogiannis et al. [2018]). However we do need to write an efficient enough interchange that readily moves gaussian splats between these data structures and the GPU, when switching between splat optimization, and manipulating and storing splats.

Chapter 4

Research Roadmap

The following roadmap outlines the systematic approach that will be undertaken to carry out research for the dissertation:

- Comprehensive Evaluation: Begin by assessing Gaussian splatting's performance on large multi-scale scenes. Through this evaluation, we aim to identify and document instances where Gaussian splatting falls short or exhibits failure cases.
- Optimization Analysis: Undertake a rigorous investigation to determine the bottlenecks in the current optimization process. This will guide subsequent modifications and enhancements.
- Out-of-Core Database Development: Construct an out-of-core database infrastructure to facilitate seamless transformation between differentiable and non-differentiable formats. Our preliminary focus is on the interoperability between PCL and PyTorch.
- Recursive Partitioning: Implement a method for recursively splitting the scene during the reconstruction process. This is pivotal to managing and optimizing larger scenes more efficiently.
- Merging Technique Development: Post recursive splitting, devise a mechanism to merge the partitioned segments, ensuring consistency and accuracy in the final reconstructed scene.
- Visualization Creation: Generate rudimentary visualizations of the reconstructions. These visualizations will be instrumental in visually assessing the quality, accuracy, and efficiency of the proposed methods.

4.1 Evaluation Metrics for Dissertation Success

The dissertation's accomplishments will be measured through a combination of quantitative and qualitative assessments. These metrics will ensure a holistic evaluation of the innovations and optimizations proposed in the context of Gaussian splatting for large-scale view synthesis.

4.1.1 Quantitative Metrics

1. **Memory Usage Reduction**:
   - *Method*: Quantitatively compare the memory utilization of the standard Gaussian splatting methodology with the proposed optimized technique using the same input dataset.
   - *Success Criteria*: A noticeable diminution in memory footprint while retaining the synthesized view's fidelity.
2. **Processing Speed**:
   - *Method*: Quantify the time consumption for view synthesis between conventional and optimized techniques.
   - *Success Criteria*: Either a speed enhancement or a time performance parity, ensuring no compromise due to memory optimizations.
3. **Quality of Synthesized Views**:
   - *Method*: Implement renowned image quality metrics such as PSNR, SSIM, and MSE to benchmark the quality of generated views.
   - *Success Criteria*: Scores reflecting that the proposed technique's output either matches or surpasses the traditional method.

4.1.2 Qualitative Metrics

1. **Expert Reviews**:
   - *Method*: Solicit feedback on synthesized views from computer graphics connoisseurs.
   - *Success Criteria*: Affirmative responses emphasizing the visual quality and appeal of the optimized technique.

4.1.3 Practicality and Applicability

1. **Integration with Existing Libraries**:
   - *Method*: Document assimilating the optimized technique into prevailing libraries.
   - *Success Criteria*: An uncomplicated integration procedure, underscoring the proposed method's adaptability.
2. **Scalability Test**:
   - *Method*: Evaluate the technique's robustness and performance across diverse dataset magnitudes.
   - *Success Criteria*: Consistent augmentation in performance and memory conservation as data complexity escalates.

Chapter 5

Additional Research Directions

Beyond our primary research objectives, should our timeline permit, we are eager to explore several supplementary avenues that could significantly enhance the depth and breadth of our investigation.

5.1 Integration with NeRF Techniques

One area of promise lies in the potential to combine NeRF-related techniques with Gaussian splatting, especially when modeling distant objects. Gaussian models have frequently shown shortcomings when tasked with reconstructing expansive terrains such as the sky. This is primarily because the inherent featurelessness of such areas offers scant Structure from Motion (SfM) points to fit Gaussians accurately.

This challenge is further compounded by the fact that Gaussians struggle with representing objects positioned at infinite distances or those lacking a well-defined structure. Against this backdrop, we hypothesize that a dual-model approach might be beneficial. Here, Gaussian splatting could focus on proximate objects, while another distinct representation hones in on more distant subjects. This hybrid approach could potentially refine the function of traditional graphics skyboxes by endowing them with enhanced view-dependent properties.

5.2 Web-Based Gaussian Splat Visualization

Another intriguing proposition is the development of a web-based viewer for visualizing Gaussian splats with Levels of Detail (LODs). While this might seem straightforward on the surface, the process is replete with challenges.

For instance, we would require a data storage format that enables pinpointed retrieval of specific levels or segments. An octree format, potentially building upon the pioneering work of Potree (Thesis et al. [2016]), emerges as a likely candidate.

Then there is the matter of crafting an order-independent transparency that remains consistent across web platforms (antimatter15 [2023]).

Finally, the viewer has to intelligently selecting the right LOD, as well as ensuring seamless transitions between different LODs, and it’s evident that the task at hand is far from simple.

Chapter 6

Conclusion

In closing, while Gaussian splatting enables unparalleled quality in view synthesis, scaling it to large scenes poses severe memory challenges. Our solution entails an out-of-core optimization pipeline, leveraging techniques like recursive partitioning and merging. This will push reconstructions to unprecedented scales.

We will comprehensively evaluate current limitations, analyze optimization bottlenecks, develop out-of-core infrastructure, implement recursive splitting and merging algorithms, and generate visualizations. Beyond our core objectives, integrating Gaussian splatting with NeRF and building an interactive LOD viewer are exciting avenues for exploration.

In summary, this undertaking promises to unlock Gaussian splatting's true potential for expansive scene reconstruction via pioneering out-of-core innovations. We look forward to systematically overcoming existing barriers to facilitate reconstructions previously impossible. The project will be both academically impactful and graphically groundbreaking as we drive Gaussian splatting to new frontiers.

Bibliography

antimatter15. Webgl 3d gaussian splat viewer, 2023. URL https://github.com/antimatter15/splat. GitHub repository.

Allison H. Baker, Alexander Pinard, and Dorit M. Hammerling. Dssim: a structural similarity index for floating-point data. *ArXiv*, abs/2202.02616, 2022. URL https://api.semanticscholar.org/CorpusID:246634654.

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. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5835–5844, 2021. URL https://api.semanticscholar.org/CorpusID:232352655.

Brojeshwar Bhowmick, Suvam Patra, Avishek Chatterjee, Venu Madhav Govindu, and Subhashis Banerjee. Divide and conquer: A hierarchical approach to large-scale structure-from-motion. *Comput. Vis. Image Underst.*, 157:190–205, 2017. URL https://api.semanticscholar.org/CorpusID:207061308.

Michela Farenzena, Andrea Fusiello, and Riccardo Gherardi. Structure-and-motion pipeline on a hierarchical cluster tree. *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 1489–1496, 2009. URL https://api.semanticscholar.org/CorpusID:14569374.

Alex Fisher, Ricardo Cannizzaro, Madeleine Cochrane, Chatura V. D. Nagahawatte, and Jennifer L. Palmer. Colmap: A memory-efficient occupancy grid mapping framework. *Robotics Auton. Syst.*, 142:103755, 2021. URL https://api.semanticscholar.org/CorpusID:234802653.

Kyle Gao, Yin Gao, Hongjie He, Denning Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *ArXiv*, abs/2210.00379, 2022. URL https://api.semanticscholar.org/CorpusID:252683586.

Sing Bing Kang. Survey of image-based rendering techniques. In *Electronic imaging*, 1998. URL https://api.semanticscholar.org/CorpusID:1079639.

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. URL https://api.semanticscholar.org/CorpusID:259267917.

Luping Lu, Yong Zhang, and Kai Liu. Block partitioning and merging for processing large-scale structure from motion problems in distributed manner. *IEEE Access*, 7:114400–114413, 2019. URL `https://api.semanticscholar.org/CorpusID:196183843`.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf. *Communications of the ACM*, 65:99 – 106, 2020. URL `https://api.semanticscholar.org/CorpusID:213175590`.

Shobha V. Patil and Dinesh B. Kulkarni. K-way spectral graph partitioning for load balancing in parallel computing. *International Journal of Information Technology*, 13:1893 – 1900, 2021. URL `https://api.semanticscholar.org/CorpusID:238654435`.

Radu Bogdan Rusu and Steve B. Cousins. 3d is here: Point cloud library (pcl). *2011 IEEE International Conference on Robotics and Automation*, pages 1–4, 2011. URL `https://api.semanticscholar.org/CorpusID:206849822`.

Drew Steedly, Irfan Essa, and Frank Dellaert. Spectral partitioning for structure from motion. *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 996–1003 vol.2, 2003. URL `https://api.semanticscholar.org/CorpusID:2543530`.

Kun Sun and Wenbing Tao. A constrained radial agglomerative clustering algorithm for efficient structure from motion. *IEEE Signal Processing Letters*, 25:1089–1093, 2018. URL `https://api.semanticscholar.org/CorpusID:49348499`.

Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8238–8248, 2022. URL `https://api.semanticscholar.org/CorpusID:246706356`.

Diploma Thesis, Markus Schuetz, and Claus Scheiblauer. Potree : Rendering large point clouds in web. 2016. URL `https://api.semanticscholar.org/CorpusID:31971528`.

Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly- throughs. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12912–12921, 2021. URL `https://api.semanticscholar.org/CorpusID:245334780`.

Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *European Conference on Computer Vision*, 2021. URL `https://api.semanticscholar.org/CorpusID:251040899`.

Konstantinos Zampogiannis, Cornelia Ferm”uller, and Yiannis Aloimonos. cilantro: A lean, versatile, and efficient library for point cloud data processing. *Proceedings of the 26th ACM international conference on Multimedia*, 2018. URL `https://api.semanticscholar.org/CorpusID:49556051`.