KNN Gaussian Splatting:
Enhancing Continuity in 3D Scene Rendering

Wuyue Lu, Kehui Li, and Shimei Jin
Simon Fraser University, CMPT 742

1. Introduction

With the ability to produce incredibly detailed renderings, Neural Radiance Fields (NeRF) [1] have represented a major breakthrough in 3D scene reconstruction. Nevertheless, their computational requirements, especially with regard to rendering and training speed, have driven research toward substitute methods. Gaussian Splatting [2] was a workable approach that improved efficiency by using rasterization and 3D gaussians. However, Gaussian Splatting has discontinuity problems despite its speed advantages; these are particularly apparent in close-up or low-density gaussian scenarios.

Another creative concept, PointNeRF [3], uses a neural point cloud and the k-nearest neighbors (kNN) algorithm to solve the continuity issue. Smoother transitions between points are ensured by using this technique, which considers each point as a basis and creates a continuous space. Our effort, which draws inspiration from PointNeRF, attempts to tackle the discontinuity issues with Gaussian Splatting by including the kNN method.

2. Related work

In 3D scene representations, both traditional and neural methods have been extensively explored. Traditional techniques include volumes, point clouds, meshes, depth maps, and implicit functions, each serving specific roles in vision and graphics applications. Recent advancements in neural scene representations, such as Neural Radiance Fields (NeRF) [1] and Gaussian Splatting [2], have significantly progressed novel view synthesis and realistic rendering, enabling impressive 3D scene reconstruction from 2D images.

The global nature of NeRFs, reconstructed as global MLPs, encoding the entire scene space, proves inefficient and expensive for complex and large-scale scenes. Although optimization techniques have been explored for NeRF, the optimization landscape for Gaussian Splatting remains relatively unexplored. Particularly, 3D Gaussian Splatting has emerged as a significant development, achieving superior visual quality and competitive training times while facilitating real-time novel-view synthesis. However, it still faces challenges in effectively representing empty space.

An innovative alternative, Point-NeRF [3], introduces a localized neural representation by combining volumetric radiance fields with point clouds. This approach excels in modeling intricate local scene details, outperforming global NeRF in rendering quality. Our idea is inspired by Point-NeRF, striving for a balance between quality and speed, while fixing the incontinuity problem which spurred research.

3. Methodology

Gaussian Splatting involves projecting a 3D gaussian cloud onto a 2D pixel grid using rasterization. Our pipeline maintains this structure but introduces modifications for enhanced continuity.

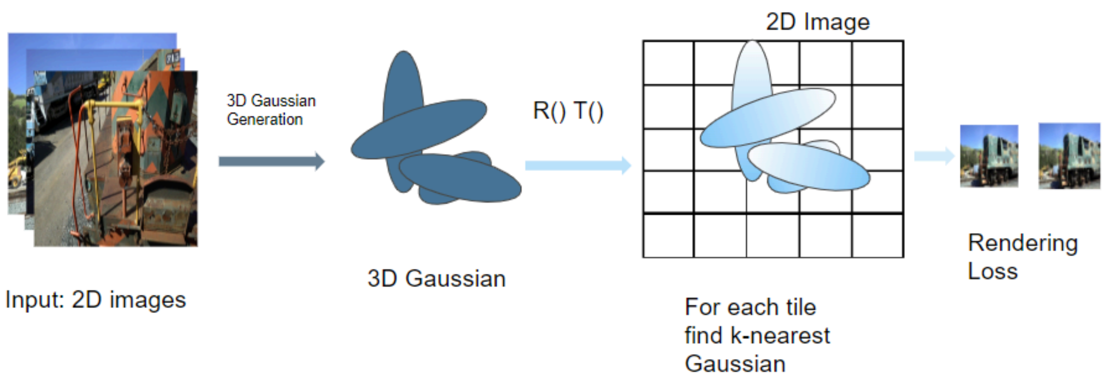


Figure 1. Pipeline of the model. After finding K nearest gaussians of a tile, MLP R and T learns how each gaussian affects each pixel.

After projecting 3D gaussians onto a 2D image grid, in each 16x16 pixel region, we gather and order the gaussians by their depth, and finally do alpha blending to blend them together. When generating parameter opacity and SH coefficients which are used in the blending process, we need to consider the gaussian surrounding it.

To overcome artifacts arising from low-density gaussians, we employ a Multi-Layer Perceptron (MLP) to predict parameters such as opacity and SH coefficients. Inspired by PointNeRF, we incorporate a k-nearest neighbors (kNN) technique into Gaussian Splatting. Each point (gaussian) is treated as a basis, creating a continuous space that addresses discontinuity challenges. This MLP-based approach allows for a more precise understanding of the influence of neighboring gaussians on each point.

Our iterative process involves projecting, blending, calculating loss against ground truth images, and refining the model. This process ensures the model's ability to predict accurate parameters for optimal blending.

4. Intermediate Results

In the intermediate phase of our project, we focused on implementing the kNN algorithm in Python to apply it on each tile and identify trained Gaussians. This step was crucial before integrating the learning mechanism for color and opacity MLPS to render the image. Our choice to use Python was dictated by our team's current familiarity with CUDA programming.

The primary outcome at this stage revealed a persistent issue: the discontinuity problem in Gaussian splatting was not resolved. Despite our approach to use neighboring Gaussians, discrete Gaussians were still visible in the rendered images. This result was consistent across different numbers of gaussians.

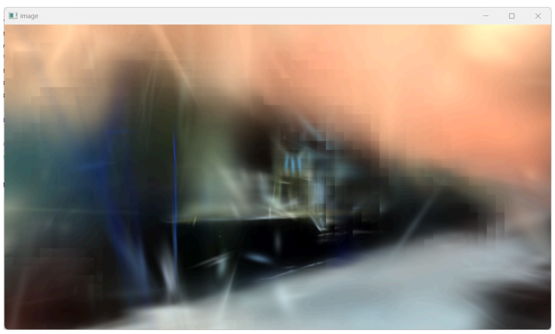
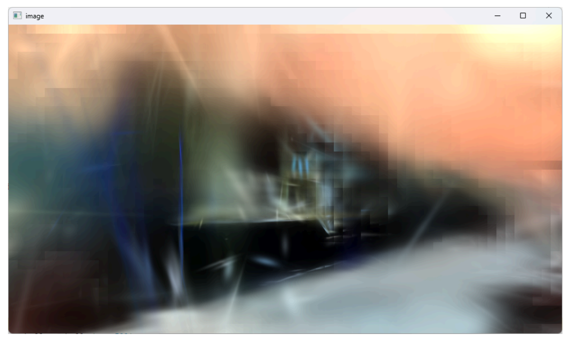
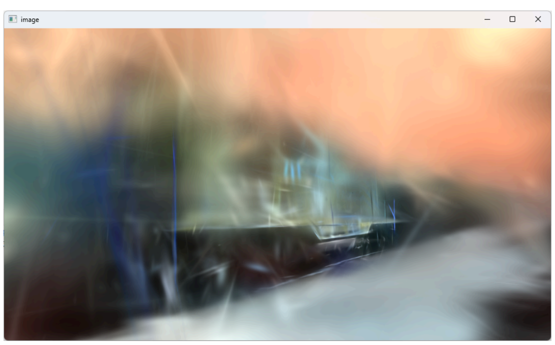
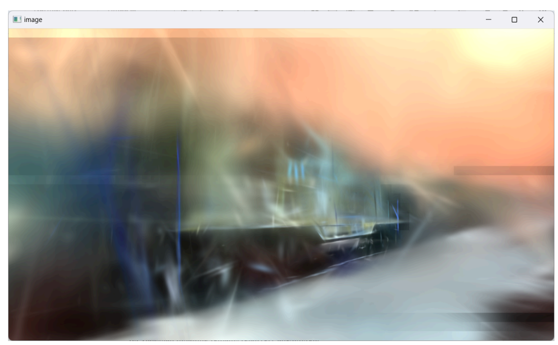
500 Gaussians for each tile		
2129 Gaussians for each tile		
	Ground Truth	KNN

Table 1: Rendered result for ground truth and with KNN for 500 Gaussians and 2129 Gaussians per tile.

Upon evaluating our intermediate results, we identified a critical aspect of the rendering method that contributed to the observed issue. Our current rendering approach still visualizes how individual Gaussians would appear, independent of their neighbors. This method inherently leads to a representation where Gaussians remain distinct entities, regardless of the number of neighboring Gaussians incorporated.

This insight suggests that the key challenge lies not just in identifying neighboring Gaussians but in fundamentally altering the rendering methodology to account for the collective influence of these neighbors by utilizing MLPs to learn the color and opacity. In essence, the current rendering approach does not fully utilize the relational data provided by the kNN algorithm, leading to a superficial aggregation of Gaussians rather than a cohesive, continuous image.

Moving forward, our focus will shift towards developing a more sophisticated rendering technique. This will involve the implementation of MLPs to learn and apply the combined influence of neighboring Gaussians on color and opacity, thereby aiming to achieve a more seamless and continuous visual output. Additionally, enhancing our team's proficiency in CUDA programming is identified as a crucial step to optimize the computational efficiency of our model, especially considering the heavy processing requirements of neural network-based rendering.

5. Conclusion

In this study, we integrated the K-Nearest Neighbors (KNN) algorithm directly into Gaussian splatting, bypassing the use of a MLP. Our results indicate that while KNN contributes to the process, it alone is insufficient to address the discontinuity issues inherent in Gaussian splatting. The primary objective of employing KNN was to standardize the number of input neurons in the MLP layer. This approach, however, highlighted the critical role of the MLP in managing these discontinuities. Notably, the success observed in Point-NeRF models suggests that the effectiveness predominantly stems from the MLP layer rather than the KNN component. In light of these findings, our future research in the upcoming semester will be dedicated to a more thorough examination of the MLP component. This focus aims to unravel its mechanisms and further optimize our approach, potentially leading to more refined and effective solutions in the field.

Reference

[1] Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." *Communications of the ACM* 65.1 (2021): 99-106.
[2] Kerbl, Bernhard, et al. "3d gaussian splatting for real-time radiance field rendering." *ACM Transactions on Graphics (ToG)* 42.4 (2023): 1-14.
[3] Xu, Qiangeng, et al. "Point-nerf: Point-based neural radiance fields." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.