Siddhant Gupta
EE367: Computational Imaging
02/21/2025

Final Project Proposal

Recent advancements in sensing and display technologies have spurred rapid growth in extended reality (XR) devices and applications. Modern XR systems typically feature dual high-resolution displays—one per eye—combined with specialized optics that replicate human 3D perception. These devices integrate multiple cameras and advanced sensors (such as time-of-flight and LiDAR) to capture detailed environmental data, enabling accurate depth representation and immersive experiences.

While augmented reality (AR) primarily overlays digital content onto the user's natural surroundings, another important XR application is the processing and display of pre-captured or synthetic 3D scenes. A key challenge in these systems is managing the enormous computational workload required to process high-fidelity data in a manner that is both power-efficient and low in latency. To address this challenge, many research efforts have focused on accelerating critical compute elements in XR workflows, including tasks such as visual simultaneous localization and mapping (SLAM) and novel view synthesis (NVS).

Novel view synthesis is the process of generating new perspectives of a scene from a limited set of input images. For several years, Neural Radiance Fields (NeRF) have been the standard approach for NVS, with extensive optimization efforts—including hardware acceleration by industry leaders—targeting its performance bottlenecks. More recently, a novel algorithm known as Gaussian splatting has emerged. This technique represents 3D scenes using collections of Gaussian primitives, enabling efficient projection from 3D space to a 2D view. Preliminary results indicate that Gaussian splatting can outperform NeRF in terms of both inference speed and image quality.

In this project, we evaluate the potential for accelerating the Gaussian splatting algorithm using a low-power, lightweight hardware accelerator. By doing so, we aim to bridge the gap between computational efficiency and the high-quality visual performance demanded by next-generation XR applications.

A few related works to this project are listed as follows:

1. Reducing the Memory Footprint of 3D Gaussian Splatting (Proc. ACM Comput. Graph. Interact. Tech., Vol. 7, No. 1, Article 1. Publication date: May 2024.)
    a. Discusses how gaussian data sets can be up to 20 gb for a scene which leads to challenges for edge computing and increased time and power usage. Authors discuss quantization and pruning methods that reduce the memory footprint by over 25x factor
2. Guikun Chen and Wenguan Wang. 2024. A Survey on 3D Gaussian Splatting. *CoRR* abs/2401.03890 (2024), 1–19. arXiv:https://arXiv.org/abs/2401.03890

    a. This work discusses the foundations on gaussian splatting algorithms and includes comparisons with other methods such as Nerf for rendering.
3. L. Wu, H. Zhu, S. He, J. Zheng, C. Chen and X. Zeng, "GauSPU: 3D Gaussian Splatting Processor for Real-Time SLAM Systems," 2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO), Austin, TX, USA, 2024, pp. 1562-1573, doi: 10.1109/MICRO61859.2024.00114. keywords: {Backpropagation;Simultaneous localization and mapping;Three-dimensional displays;Instruction sets;Graphics processing units;Rendering (computer graphics);Throughput;Real-time systems;Energy efficiency;Image reconstruction;3D Gaussian Splatting;Simultaneous Localization and Mapping;Accelerator;Graphic Processing Unit},
    a. This work is one of the only works that has been published that analyzes gaussian splatting for hardware acceleration. They propose a co-processing paradigm where they accelerate/offload the volume rendering from the GPU to a custom device they designed. The main benefit of this is to better handle the sparsity present.

Project Overview and Goals

While there have been some works that specifically aim to accelerate the process of NVS using gaussian splatting, they usually focus on the algorithms perspective and/or utilize CUDA for the implementation and processing rather than any custom hardware designed for this purpose. In this project, I am to explore which possible architecture the gaussian splatting based NVS (if any) can map to for a lightweight accelerator. I will discuss with my PI, Professor Priyanka Raina, for specific constraints, but tentatively my goal is to figure out a way of mapping this algorithm using at most a 64 by 64 array of processing elements. Related works suggest this algorithm is severely memory bound so the end goal of this project is to explore the feasibility of different architectures that try to achieve inference within a 10 ms latency that can help mitigate the memory bottleneck.

The steps of my final project will be in three main steps as follows:

1. Implementation of a python or c++ model of gaussian splatting inference (can be inspired by or use source code from the original authors)
2. Implementation of (discussed in related works) proposed algorithms that reduce the memory needed of the algorithm down to under a couple of gigabytes (the papers propose a 27x reduction in dataset size)
3. Refactoring the code to simulate some reasonable degree of parallelism (parallel processing elements that all divide up the frame for example) and implementing some mechanism to estimate the latency and how much memory is used by each parallel