Progress Report
CS4681 – Advanced Machine Learning

Kulasekara K.M.S.N. - 210302P

Project Code : 3DV009

Table of Contents

1. Introduction

Recent advancements in neural scene representation, particularly Neural Radiance Fields (NeRFs), have significantly transformed the field of novel view synthesis. These models enable the generation of highly photorealistic images of a 3D scene from sparse 2D views, opening up possibilities in areas such as virtual reality, augmented reality, robotics, and computer graphics. By learning a continuous volumetric representation of a scene parameterized by neural networks, NeRFs can produce images from arbitrary viewpoints that capture fine geometric details and complex lighting effects. Despite their impressive visual fidelity, traditional NeRF models are extremely computationally intensive and require substantial memory resources. Training a single scene often takes hours to days on high-end GPUs, and inference is relatively slow, which makes real-time applications practically infeasible.

To overcome these limitations, recent research has turned to sparse and explicit voxel-based representations. By discretizing the scene into voxels and storing radiance and opacity information directly, these methods drastically reduce the computational and memory burden while still maintaining high-quality reconstructions. Among these approaches, Plenoxels [1] has emerged as a state-of-the-art method. Plenoxels represents scenes using a sparse voxel grid where each voxel contains an opacity value and a set of spherical harmonic (SH) coefficients for each colour channel. Unlike NeRF, which requires a neural network to approximate radiance and optimize millions of parameters indirectly, Plenoxels directly optimizes the voxel values themselves. This direct optimization approach significantly accelerates training, enabling the model to converge in minutes rather than days, while achieving real-time rendering performance on both synthetic and real-world datasets.

While Plenoxels demonstrates strong performance in terms of speed and visual quality, there are still several avenues for improvement. Challenges remain in regularization stability, particularly when training on noisy or incomplete real-world datasets, where artifacts can appear. Furthermore, reconstruction accuracy can vary across different scene types, and the model's sensitivity to hyperparameters can impact convergence and robustness. The goal of this project is to investigate targeted enhancements to Plenoxels by improving loss functions, adopting more sophisticated training strategies, and performing hyperparameter optimization. Through these interventions, the project aims to push the performance boundaries of voxel-based neural rendering, making it more robust, accurate, and applicable to a wider variety of scenes and real-time applications.

2. Literature Review

The rapid development of neural scene representations has led to a wide variety of methods for novel view synthesis, each balancing trade-offs between quality, efficiency, and memory usage. Early neural approaches like NeRF demonstrated the potential of continuous volumetric functions for photorealistic rendering but suffered from slow training and rendering. Subsequent works have progressively introduced structural and algorithmic innovations to accelerate performance, improve generalization, and reduce resource requirements. This section reviews the evolution from NeRF to voxel-based representations, highlighting key contributions and the state-of-the-art Plenoxels framework, while identifying remaining gaps that motivate this project.

In their work Mildenhall et al. [2] Neural Radiance Fields (NeRF) which they introduced marked a breakthrough in neural scene representation by modeling a continuous volumetric scene function using a multilayer perceptron (MLP). Given a sparse set of calibrated input images, NeRF optimizes the network weights to synthesize novel photorealistic views. A notable strength of NeRF is its memory efficiency, despite its expressiveness, the entire neural network representation occupies less than 5 MB of memory. However, this efficiency comes at the cost of extremely high training and inference times, often requiring hours to days to converge, making NeRF unsuitable for real-time applications.

Subsequent works sought to address NeRF's limitations. In their work Baroron et al. [3] they introduced Mip-NeRF which is an anti-aliasing solution inspired by mipmaps in graphics pipelines, significantly improving rendering quality when dealing with varying levels of detail. Similarly, SNeRG [4] shifted towards a sparse voxel grid representation, caching features at discrete grid points to accelerate rendering. While this reduced query costs, SNeRG still relied on neural networks for radiance estimation, leaving training time relatively high. PlenOctrees [5] extended this idea by using octree-based hierarchical data structures in combination with voxel features, enabling real-time rendering but requiring a pre-trained NeRF backbone as input, which limited practicality.

Plenoxels [1] departed radically from the NeRF paradigm by eliminating the neural network entirely. Instead, scenes are represented as a sparse voxel grid where each voxel stores a density (opacity) value and spherical harmonic (SH) coefficients for colour representation. Rendering is performed by volume integration over these explicit voxel parameters, and optimization directly updates the voxel grid instead of network weights. This innovation reduced training time from days to minutes while achieving real-time rendering performance on both synthetic and real-world datasets. To stabilize optimization, Plenoxels introduced Total Variation (TV) regularization, sparsity losses, and a beta loss for opacity control.

The authors used the same differentiable volume rendering model as NeRF but replaced the continuous MLP representation with a voxel-based geometry model. Like PlenOctrees, Plenoxels adopt a sparse voxel grid, but for simplicity and to enable efficient trilinear interpolation, they avoided octrees. Instead, a dense 3D index array is maintained with pointers to a separate data array that stores values only for occupied voxels. Each voxel encodes appearance using spherical harmonics of degree 2, which requires 9 coefficients per colour channel, resulting in 27 coefficients per voxel. The choice of degree 2 harmonics follows findings from PlenOctrees, which showed that higher-order harmonics provide minimal additional benefits. To further improve quality, the method prunes unnecessary voxels, refines surviving voxels, and applies trilinear interpolation to significantly outperform simpler nearest-neighbour interpolation.

The study also highlighted the importance of interpolation in voxel-based rendering. In particular, trilinear interpolation was shown to provide much better results than nearest-neighbour methods. By smoothly blending voxel values, trilinear interpolation not only captures subtle variations within a single voxel but also ensures that the overall representation behaves like a continuous function rather than a blocky approximation. This continuity is especially valuable during optimization, as it stabilizes training and leads to more accurate reconstructions. Their experiments further demonstrated that increasing resolution in combination with trilinear interpolation produced significant gains in rendering quality.

For optimization, Plenoxels employed a coarse-to-fine training strategy, where the voxel grid resolution was progressively refined over time. To accelerate convergence, they used stochastic sampling, drawing a random subset of rays at each step to evaluate the reconstruction loss and a separate subset of voxels to evaluate the total variation (TV) regularization. This sampling strategy allowed faster iterations while maintaining reconstruction quality. Training was performed using an adaptive optimizer with different learning rate schedules for spherical harmonic coefficients, tone mapping parameters, and voxel opacities. Regularization was applied more strongly in the early stages, with TV loss active before grid upsampling, ensuring smoothness and stability during initialization. A carefully balanced batch size was used to trade-off between memory efficiency and training speed. Together, these design choices enabled Plenoxels to achieve fast training and real-time rendering while preserving high visual fidelity across both synthetic and real-world datasets.

3.Methodology

To address the limitations identified in current voxel-based neural rendering methods, this project adopts a focused strategy aimed at enhancing the Plenoxels framework. Rather than proposing a completely new representation, the goal is to build upon its efficiency and strengths through carefully designed modifications. The methodology emphasizes incremental yet impactful improvements that directly target known bottlenecks in training stability, reconstruction fidelity, and adaptability across diverse scenes. In particular, the project explores enhancements in three main areas: hyperparameter optimization, loss function design, and training strategies.

3.1 Hyperparameter Optimization

To improve training stability and efficiency, I propose exploring adaptive learning rate schedules such as cosine annealing and cyclical learning rates [6] of relying on a fixed exponential decay. These adaptive strategies allow for more flexible control of the optimization process, enabling faster convergence while avoiding premature stagnation. In addition, automated hyperparameter search methods like Bayesian optimization [7] will be employed to tune critical parameters, including TV regularization weights, the degree of spherical harmonics, and voxel pruning thresholds. Currently, TV regularization is applied with separate weights for different categories (e.g., forward-facing scenes), but I plan to simplify this by introducing a unified TV weight per scene. By combining adaptive learning dynamics with principled hyperparameter tuning and more generalizable TV weighting, I expect the model to achieve higher reconstruction quality with reduced manual intervention.

3.2 Loss Function Enhancement

Most NeRF-like models, including Plenoxels, are trained using Mean Squared Error (MSE) between the predicted pixel colours and ground-truth image pixels. MSE is simple and mathematically convenient, but it only evaluates differences at the pixel level. This means that two images that look perceptually similar to humans (e.g., one edge shifted by a pixel) can still have a high MSE. As a result, MSE-optimized models often produce reconstructions that are accurate in colour but visually blurry, especially around edges and fine textures. Instead of relying purely on MSE, I propose using perceptual losses such as LPIPS, which measures similarity in deep feature space to capture semantic details (edges, textures) aligned with human perception, and SSIM-weighted loss, which evaluates luminance, contrast, and structural consistency to better preserve edges and textures critical for realistic 3D reconstructions.

3.3 Training Strategy Improvements

In terms of training strategies, I propose a progressive regularization scheme where strong TV and sparsity penalties are applied during coarse optimization stages, followed by weaker or adaptive regularization in finer stages. This gradual adjustment prevents over-smoothing while ensuring stability throughout training, as opposed to abruptly disabling regularization. Additionally, I will investigate transfer learning from pretrained voxel grids across different scenes to accelerate convergence, leveraging prior knowledge to reduce training time. Finally, I plan to explore ensemble voxel models, where multiple voxel grids trained with varying hyperparameter settings are averaged or blended. Such ensemble approaches are expected to enhance robustness and generalization, mitigating the sensitivity of reconstruction quality to specific hyperparameter configurations.

4. Project Timeline

TASK NUMBER	TASK TITLE	START DATE	DUE DATE	DURATION (DAYS)
1	**Background Study and Project Setup**			
1.1	Research on Project Background	14/08/25	20/08/25	6
1.2	Literature Review	14/08/25	24/08/25	10
1.3	Setting Up the Project	15/08/25	18/08/25	3
1.4	Plenoxel Baseline Analysis	18/08/25	28/08/25	10
2	**Trying Out Methods and Testing**			
2.1	Hyperparameter Optimization	26/08/25	24/09/25	28
2.2	Loss Function Enhancement	28/08/25	26/09/25	28
2.3	Training Stratergy Improvement	01/09/25	29/09/25	28
3	**Documentation and Finalization**			
3.1	Creating the Progress Report	21/08/25	24/08/25	3
3.2	Creating the Short Paper	16/09/25	21/09/25	5
3.3	Creating the Final Paper	25/09/25	02/10/25	7
3.4	Submitting the Paper to a Confernce	03/10/25	05/10/25	2

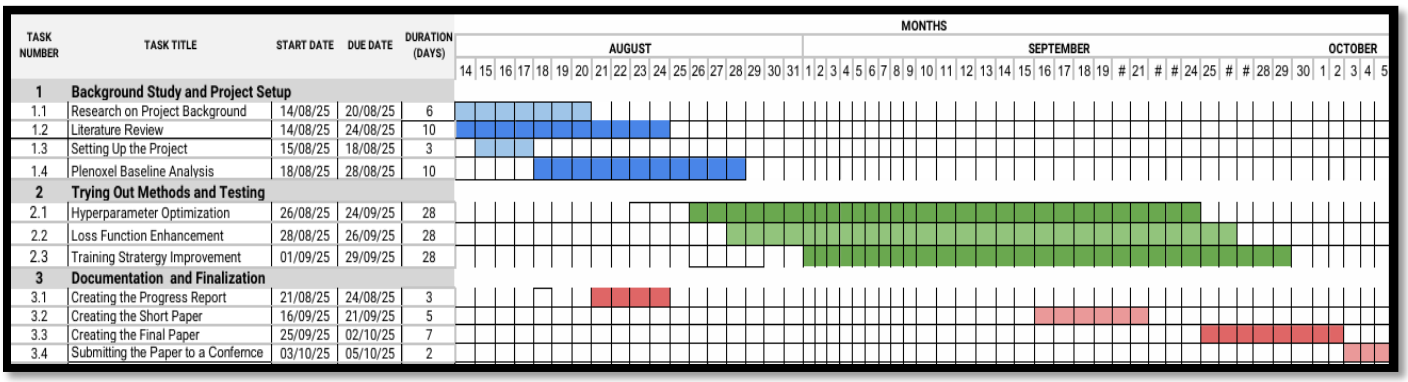

Figure 1: Project Timeline

5.Conclusion

This project builds upon the efficiency and scalability of Plenoxels while addressing its key limitations in stability, reconstruction accuracy, and generalization. By introducing adaptive hyperparameter optimization strategies, perceptual and structure-aware loss functions, and more flexible training schedules, I aim to enhance both the visual fidelity and robustness of voxel-based scene representations. The proposed improvements ranging from adaptive learning rate schedules to perceptual loss integration and progressive regularization are designed to push voxel-based methods closer to achieving high-quality reconstructions in real-world, diverse, and noisy datasets, all while maintaining real-time performance. Ultimately, the project's goal is to move beyond incremental acceleration and toward developing a more reliable, generalizable framework for neural rendering, making practical applications such as VR/AR, robotics, and graphics production more feasible.

6.References

[1]. S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, “Plenoxels: Radiance fields without neural networks,” 2022. https://openaccess.thecvf.com/content/CVPR2022/html/Fridovich-Keil_Plenoxels_Radiance_Fields_Without_Neural_Networks_CVPR_2022_paper.html

[2]. B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “NERF: Representing scenes as neural radiance fields for view synthesis,” *arXiv.org*, Mar. 19, 2020. https://arxiv.org/abs/2003.08934

[3]. J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “MIP-NERF: a multiscale representation for Anti-Aliasing Neural Radiance fields,” *arXiv.org*, Mar. 24, 2021. https://arxiv.org/abs/2103.13415

[4]. P. Hedman, P. P. Srinivasan, B. Mildenhall, J. T. Barron, and P. Debevec, “Baking neural radiance fields for Real-Time view synthesis,” *arXiv.org*, Mar. 26, 2021. https://arxiv.org/abs/2103.14645

[5]. A.Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, “PlenOctrees for real-time renderng of neural radiance fields,” *arXiv.org*, Mar. 25, 2021. https://arxiv.org/abs/2103.14024

[6]. L. N. Smith, “Cyclical learning rates for training neural networks,” *arXiv.org*, Jun. 03, 2015. https://arxiv.org/abs/1506.01186

[7]. J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian optimization of machine learning algorithms,” *arXiv.org*, Jun. 13, 2012. https://arxiv.org/abs/1206.2944