




Plenoxels: Radiance Fields without Neural Networks

The University of Georgia

Subject: Advanced Topic in Computer Science

Presented By: Ratish Jha, Sakshi Seth, Likitha Karnati



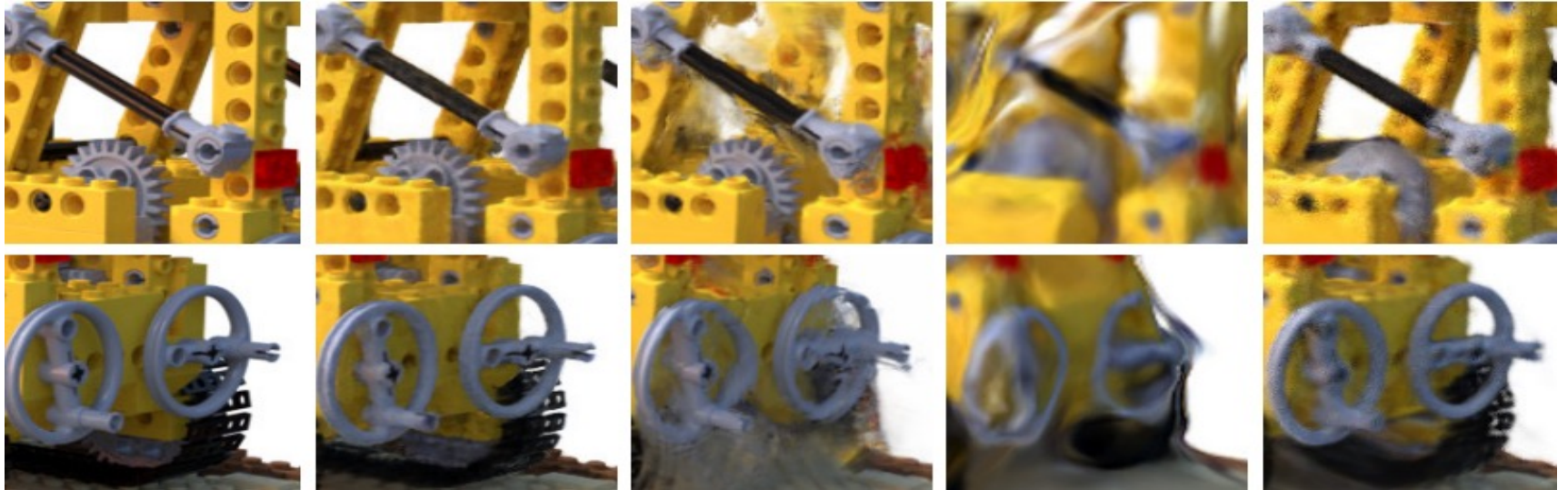
Agenda

1. Introduction
2. Motivation of the proposed work
3. Key Contribution
4. Related work
5. Method
6. Results
7. Limitation and future directions



Imagine taking a few photos with your mobile phone and quickly converting them into a 3D scene that you could navigate.

2D images of a Lego bulldozer from different angles



Navigable in 3D space



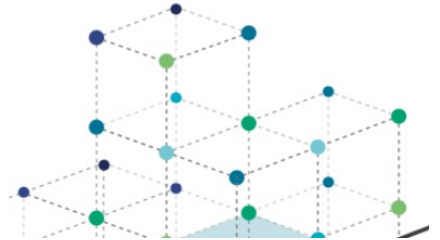
Introduction

What does Plenoxels (Plenoptic voxels) means?

- It is a sparse voxel grid where each occupied voxel corner stores a scalar opacity value and a vector of spherical harmonic (SH) coefficients for each color channel.

What are Voxels?

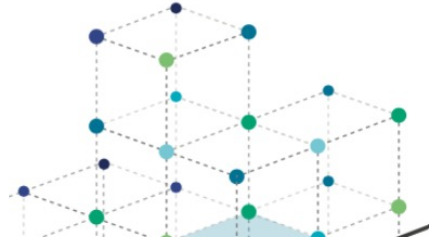
- Voxels are small, three-dimensional (3D) units or "volume elements" that are used to represent the properties of objects or scenes in 3D space.
- These voxels can be used to render the scene from any angle quickly and efficiently, without the need for complicated neural networks.





How Plenoxels are made?

- It works by first capturing the light and reflection in a scene from different angles using virtual cameras.
- Then, this information is used to create 3D voxels called Plenoxels that store the light and reflection information at each point in the scene.





What is a Radiance Fields?

- Essentially, a radiance field describes how light interacts with the surface of an object and how it is reflected or absorbed by the surface.

Why Radiance Field is necessary in Plenoxels?

- Plenoxels rely on radiance fields to capture the lighting and reflectance properties of the scene, which are then used to render realistic images of the object from any viewpoint.

Motivation



- Large amount of time taken for training and rendering by NeRF.
- Subsequent papers attempted to reduce the computational cost of NeRF, but training still take multiple hours on a single GPU, making it impractical for many applications.
- These limitation led to an alternative method such as Plenoxels.

Key contribution



- Authors have built a custom CUDA implementation of their radiance field optimization method.
- Custom CUDA implementation can optimize a bounded scenes in just 11 minutes and 27 minutes for unbounded scenes.

Related Work



- Classical Volume Reconstruction.
- Neural Volume Reconstruction.
- Accelerating NeRF.

Classical Volume Reconstruction

- Classical Volume Reconstruction refers to a method used to represent 3D volumes.
- Classical methods for volume rendering include voxel grids and multiplane images.
- Sparse array only stores the non-zero elements, and it can greatly reduce the memory requirements of the data structure and make it more computationally efficient.

Neural Volume Reconstruction



- The proposed method in this paper is most similar with Neural volume.
- Both uses a voxel grid with interpolation.
- While the proposed method shows that the voxel grid can be optimized by pruning and coarse to fine optimization, without any neural networks or warping functions.

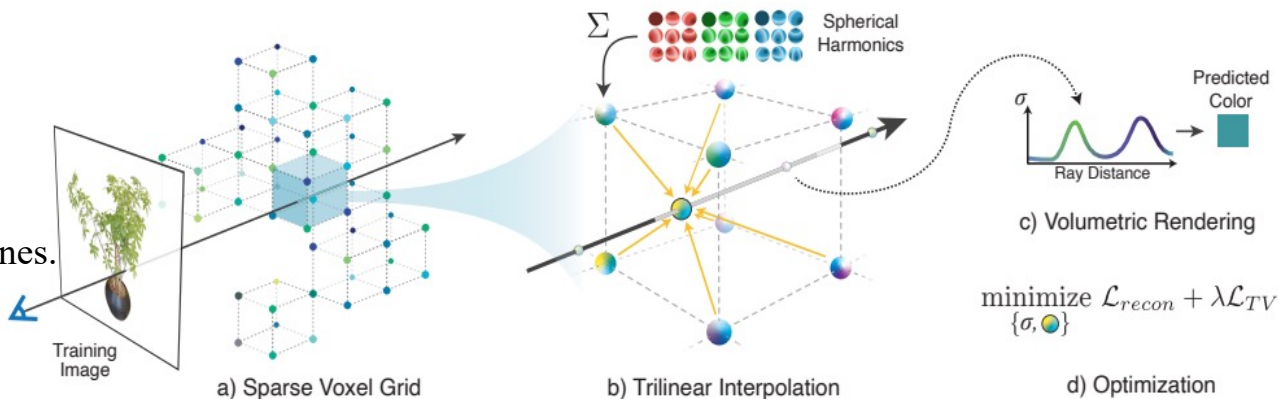
Accelerating NeRF



- One approach is to subdivide the 3D volume into regions that can be processed more efficiently.
- Some methods have focused on a range of computational and pre- or post-processing techniques to remove the bottlenecks.
- Another approach involves pretraining a NeRF model and then extracting it into a different data structure that supports fast inference.

Method

- Volume Rendering.
- Voxel Grid with Spherical Harmonic.
- Interpolation.
- Coarse to Fine.
- Optimization.
- Unbounded Scenes.
- Regularization.
- Implementation.



Volume Rendering

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i$$

where

$$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$$

T_i = Light transmitted through ray \mathbf{r} to each sample i .

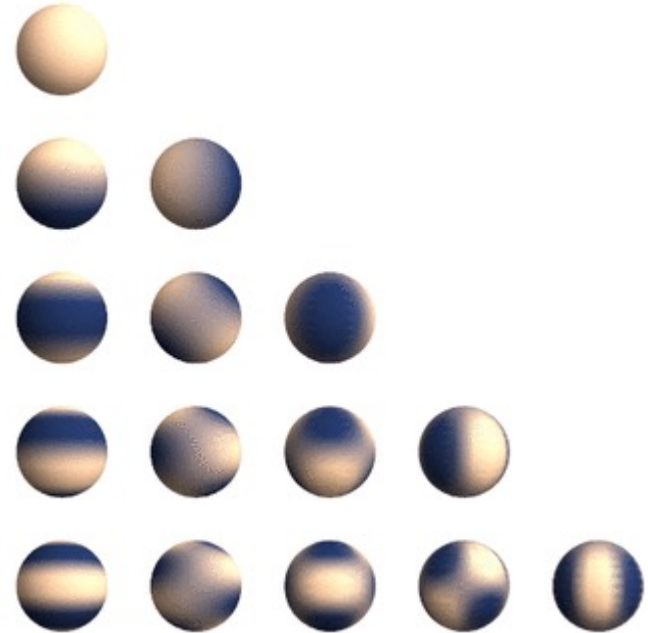
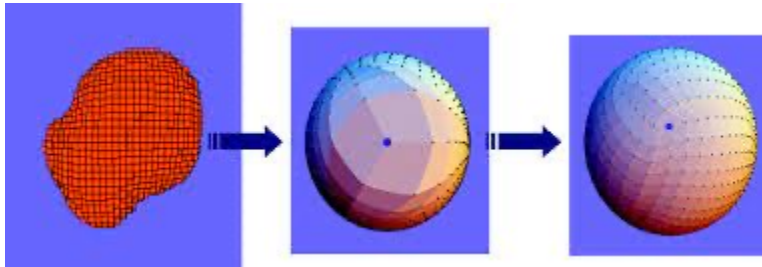
σ_i = Opacity of sample i

\mathbf{c}_i = color of sample i

δ_i = distance to next sample

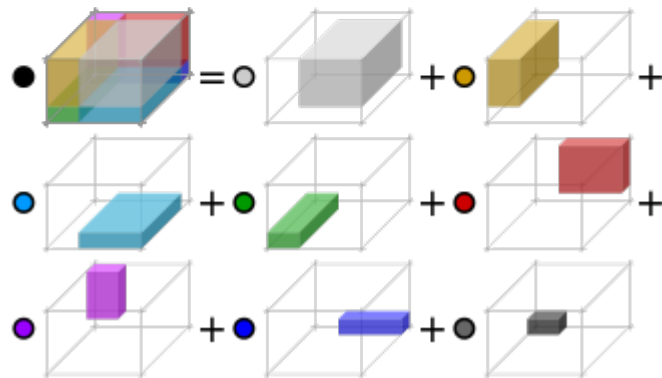
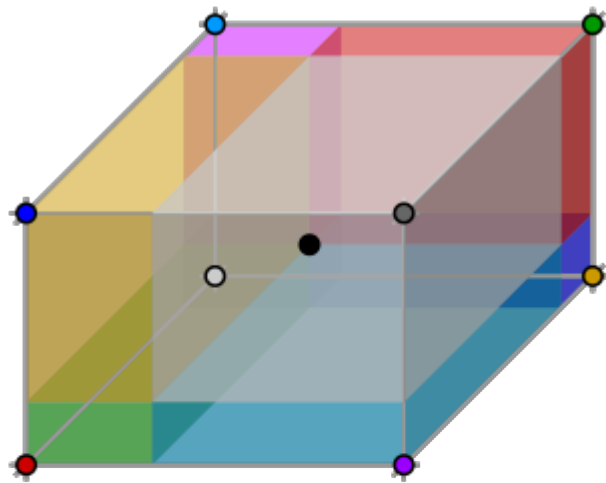
Voxel with Spherical Harmonics

- Spherical harmonic functions are a set of orthogonal functions defined on the surface of a sphere.



TriLinear Interpolation

- The opacity and color at each sample point along each ray are computed by trilinear interpolation of opacity and harmonic coefficients stored at the nearest 8 voxels.
- Trilinear Interpolation significantly outperforms Nearest Neighbor.



	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
Trilinear, 256^3	30.57	0.950	0.065
Trilinear, 128^3	28.46	0.926	0.100
Nearest Neighbor, 256^3	27.17	0.914	0.119
Nearest Neighbor, 128^3	23.73	0.866	0.176

Coarse to Fine & Optimization

Coarse iters.: 1
Eps. time: 00:00

Coarse iters.: 1
Eps. time: 00:00

Coarse iters.: 1
Eps. time: 00:00

MSE over rendered pixel with Total Variation Regularization:

$$\mathcal{L}_{recon} = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \|C(\mathbf{r}) - \hat{C}(\mathbf{r})\|_2^2$$

$$\mathcal{L} = \mathcal{L}_{recon} + \lambda_{TV} \mathcal{L}_{TV}$$

$$\mathcal{L}_{TV} = \frac{1}{|\mathcal{V}|} \sum_{\substack{\mathbf{v} \in \mathcal{V} \\ d \in [D]}} \sqrt{\Delta_x^2(\mathbf{v}, d) + \Delta_y^2(\mathbf{v}, d) + \Delta_z^2(\mathbf{v}, d)}$$

Unbounded Scenes



- 360 model is made by essentially combining the foreground and background model using Equirectangular projection.
- Model is kept lightweight by only storing RGB color and Sparse layer which have values below Opacity threshold limit.

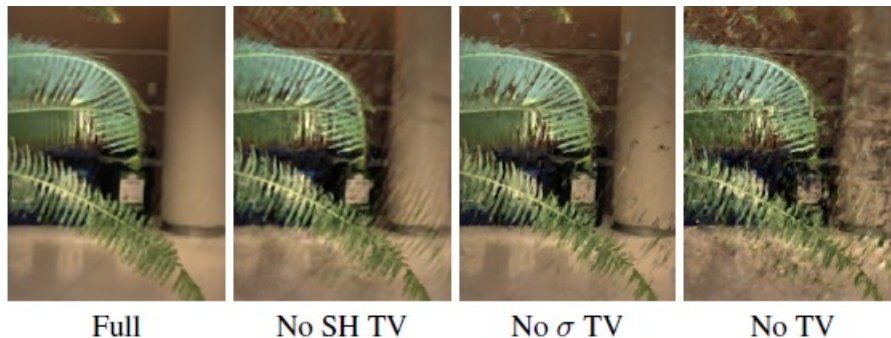
Regularization

- Cauchy Regularizer: For real Unbounded Scenes

$$\mathcal{L}_s = \lambda_s \sum_{i,k} \log(1 + 2\sigma(\mathbf{r}_i(t_k))^2)$$

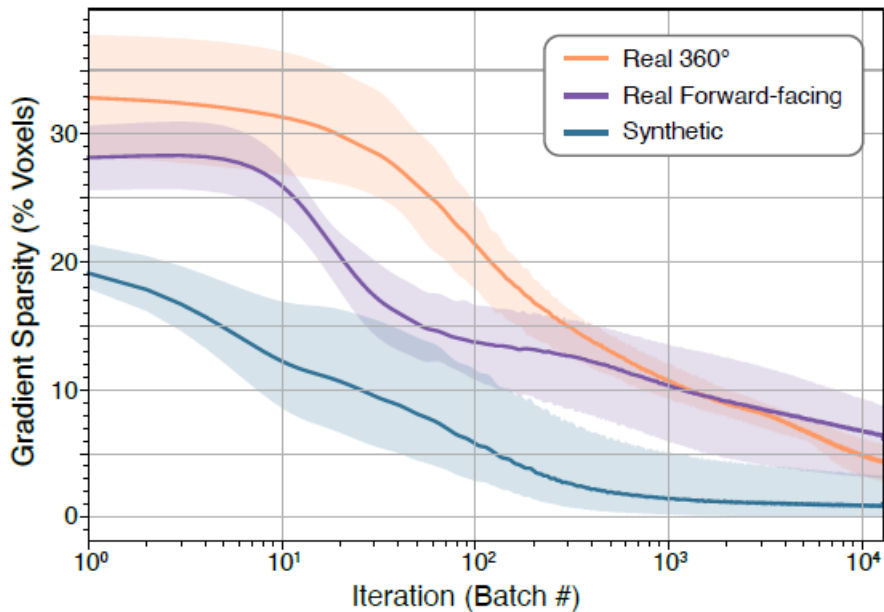
- Beta Distribution Regularizer: For real 360 scenes

$$\mathcal{L}_\beta = \lambda_\beta \sum_{\mathbf{r}} (\log(T_{FG}(\mathbf{r})) + \log(1 - T_{FG}(\mathbf{r})))$$



Implementation

- Sparsity Gradient drops very quickly, fewer than 10% of the voxels have nonzero gradients.
- Lesser computation needed resulting in faster convergence.



Results

Experiments are performed on three types of Visual Content they are:

- **Synthetic, bounded scenes** which are typically designed with a fixed perspective, which can be controlled and adjusted by the creator of the scene.
- **Real, forward-facing scenes** which are captured from a single, fixed position, which limits the perspective to a specific viewpoint.
- **Real 360-degree scenes** which capture a full 360-degree view of the surrounding environment, allowing the viewer to explore the scene from a variety of different perspectives.

Synthetic, bounded scenes.

- The authors compared their method to other method for representing 3D scenes, such as JaxNERF.
- Plenoxels were found to be able to represent the scenes with greater accuracy and detail than the other methods.

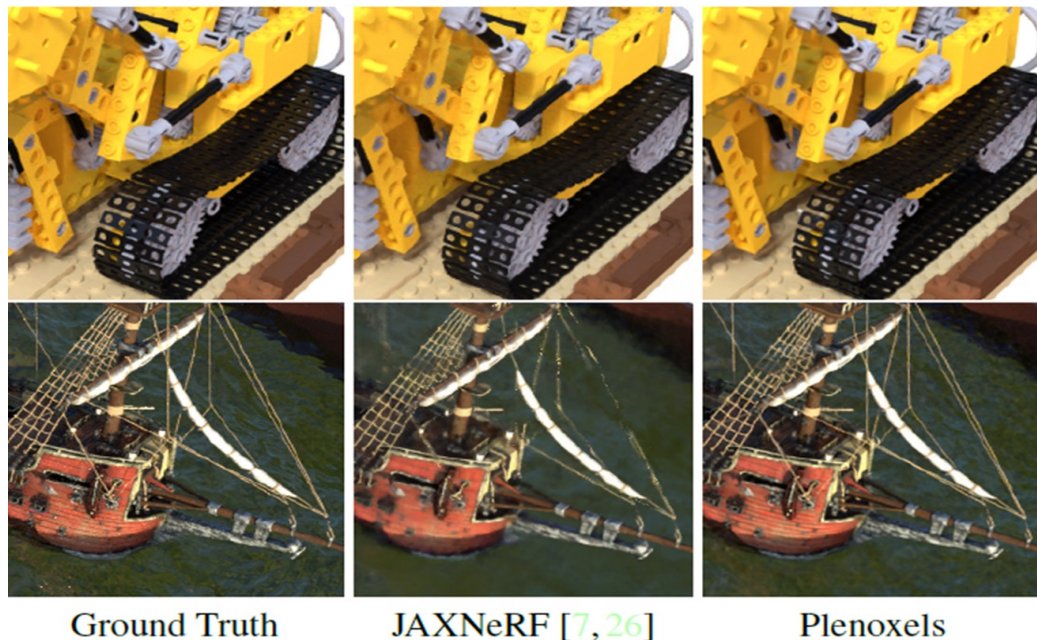


Figure. Synthetic, bounded scenes. Example results on the lego and ship synthetic scenes from NeRF [26]. Please see the supplementary material for more images.

Real, forward-facing scenes

- Real, unbounded, forward-facing scenes:
- The scenes were captured using a camera.



Real, forward-facing scenes (..)



Ground Truth

JAXNeRF [7, 26]

Plenoxels

Figure. Real, forward-facing scenes. Example results on the fern and orchid forward-facing scenes from NeRF.

Results on 360° Scenes

- Real, unbounded, 360-degree scenes
- The scenes were captured using a 360-degree camera.



Figure Real, 360° scenes. Example results on the playground and truck 360° scenes from Tanks and Temples [15].

Results on 360° Scenes

The creator will be able to capture it from different perspectives.



Results on 360° Scenes (Cont...)



Ablation Studies

	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
Ours: 100 images (low TV)	31.71	0.958	0.050
NeRF: 100 images [26]	31.01	0.947	0.081
Ours: 25 images (low TV)	26.88	0.911	0.099
Ours: 25 images (high TV)	28.25	0.932	0.078
NeRF: 25 images [26]	27.78	0.925	0.108

Table 3. **Ablation over the number of views.** By increasing our TV regularization, we exceed NeRF fidelity even when the number of training views is only a quarter of the full dataset. Results are averaged over the 8 synthetic scenes from NeRF.

- **Plenoxels** takes 11 minutes to optimize a bounded scene on a single Titan RTX GPU, whereas **NeRF** takes roughly a day.
- **Plenoxels** take roughly 27 minutes to optimize an unbounded scene, whereas **NeRF++** takes about four days.

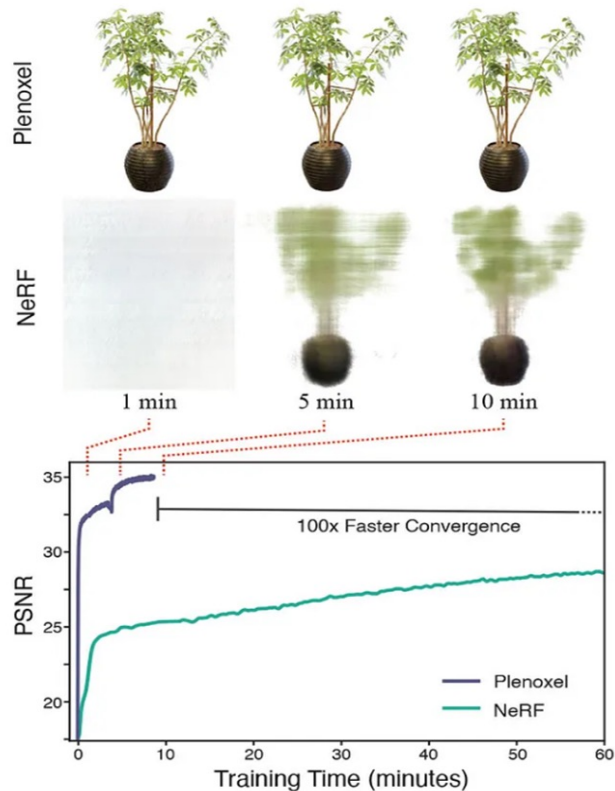


Figure: Comparison between the results of NeRF and Plenoxel

Limitations

- The method used to create 3D models can still have some errors, which may create artifacts (imperfections in 3D image formed).
- The settings like parameters used for the plenoxil method may not be the best for every single scene and may need to be adjusted depending on the specific details of each one.
- The method may not be as good as other methods in certain situations, such as when trying to create 3D models with very detailed textures.

Future Work:

- Finding ways to reduce or remove the artifacts that may appear in the final 3D models.
- Investigating ways to adjust the settings for the method on a scene-by-scene basis, to improve the accuracy of the resulting models.
- Adding new features to the method, such as support for multi-scale rendering and tone-mapping, to create even more realistic 3D models.



Thank You!