Neural Networks and Deep Learning
Machine Learning Optimization Algorithms

Project Report On

# Novel View Synthesis and 3D Reconstruction with Neural Radiance Fields

**Authors**

Vansh Shah
C093

Siddh Sanghvi
C079

**Under the Mentorship of**

Archana Nanade

Abhay Kolhe

**Department of Computer Engineering**
**Mukesh Patel School of Technology Management & Engineering**
**NMIMS (Deemed-to-be University), Mumbai**

22nd March 2024

## 1. Introduction

Scene understanding is a computer vision task that involves implementation of algorithms that allow a computer to represent and process a 3D environment. The field of scene understanding and novel view synthesis has witnessed significant advancements with the involvement of deep learning techniques. This research project explores the capabilities of Neural Radiance Fields (NeRF) for reconstructing 3D scenes and explores complementary optimized methods for further analysis. NeRF, introduced in 2020, revolutionized the field by leveraging deep learning MLP based architectures to efficiently learn and represent a scene's radiance and geometry from a set of 2D images. This research project implements and evaluates NeRF for its effectiveness in reconstructing scenes. This project also explores and compares the potential of K-Planes, a recent approach that complements NeRF by offering a more optimized and sophisticated representation technique. To understand the core functionalities and practical implementations behind NeRF and other related methods, this project also explores the process and techniques of ray tracing and volumetric rendering. Ray tracing simulates the path of light through a scene, allowing for realistic image generation and also facilitating novel view synthesis. Volumetric rendering is a density based differentiable rendering technique that builds upon ray tracing by considering the opacity and color information within a 3D volume, leading to more accurate reconstructions. Furthermore, this research explores the application of ray marching, a technique for efficiently traversing a 3D scene along rays, to extract surface meshes from the reconstructed volume. This allows for the creation of a tangible 3D model from the NeRF representation.

By combining NeRF with K-planes, ray tracing, volumetric rendering, and ray marching, this research project aims to achieve a comprehensive understanding of scene reconstruction and explore the potential for generating high-fidelity 3D representations and extractable meshes from sparse input data.

**Keywords**
Neural Radiance Fields, MLPs, Novel View Synthesis, Ray Tracing, Volumetric Rendering, Differentiable Rendering, K-Planes, Ray Marching, Mesh Extraction, 3D Mesh, 3D Reconstruction.

## 2. Literature Survey

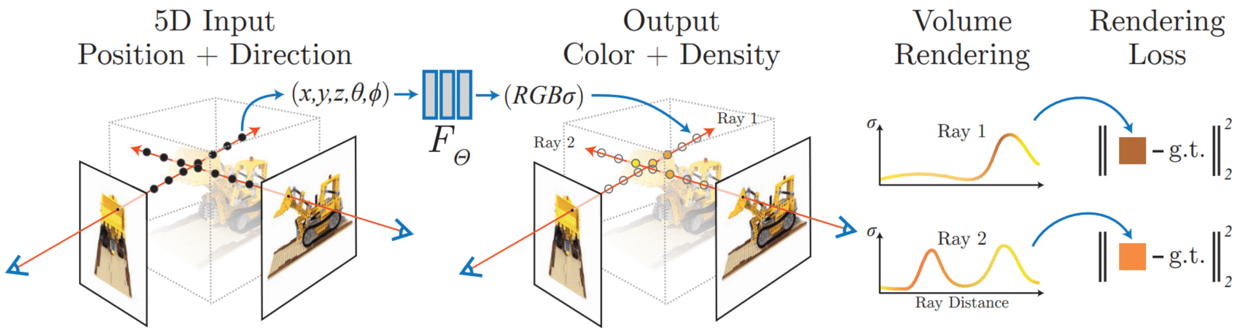
3D Scene interpretation and novel view synthesis have long been difficult problems in computer vision, with researchers exploring various techniques to reconstruct 3D environments from sparse input data. Traditional techniques, such as multi-view stereo (Seitz et al., 2006), attempted to extract depth information from multiple overlapping images. However, these approaches frequently struggled with occlusions, texture-less regions, and scenes with complex geometry.

Structure from motion (Schönberger & Frahm, 2016) approaches aimed at estimating both camera poses and 3D scene geometry from a collection of unsorted images. While effective for certain situations, these methods required significant manual involvement and often failed in extracting fine details or handling challenging overexposed images.
Another technique which was involved in scene representation was voxel-based representations (Szeliski, 1993). In this technique the 3D scene was divided into a regular grid of voxels, each comprising information about the occupancy and color. These representations were computationally expensive and suffered from quantization artifacts, making them unsuitable for high resolution scenes.
Prior to NeRF, multiple techniques were explored for understanding 3D scenes and generating novel views, but these methods often struggled with capturing fine details and required manual interference with complex scenes.

Neural Radiance Fields (NeRF) (Mildenhall et al., 2020) transformed the field by introducing deep learning and neural networks to represent a scene's radiance and geometry from 2D images. Researchers have improved and developed variants of NeRF, e.g., PixelNeRF (Yu et al., 2021) for unbounded scenes and NeRF++ (Zhang et al., 2020) for improved rendering quality and convergence.
NeRF's representing dense volumetric output can be inefficient leading to a considerable computation time to render the output. Researchers have explored improvements, e.g., Kernel Predicting Neural Rendering (Srinivasan et al., 2021) for complex light transport, and Scene Representation Networks (Sitzmann et al., 2019) for single-viewpoint rendering. However, integrating complementary techniques like K-Planes, ray tracing, volumetric rendering, and ray marching for efficient scene reconstruction and mesh extraction remains unexplored.

## 3. Proposed Work

We propose Neural Radiance Fields (NeRF) models for scene understanding. NeRF models represent a scene using deep learning. By overfitting to the scene and using positional encoding, a NeRF model can capture and represent complex lighting effects and finer details. Unlike the earlier methods that rely on multi-view stereo reconstruction, NeRFs have the capabilities to learn from sparse views and generalize well to novel view points, making them a powerful tool for creating realistic and detailed 3D scene understanding.

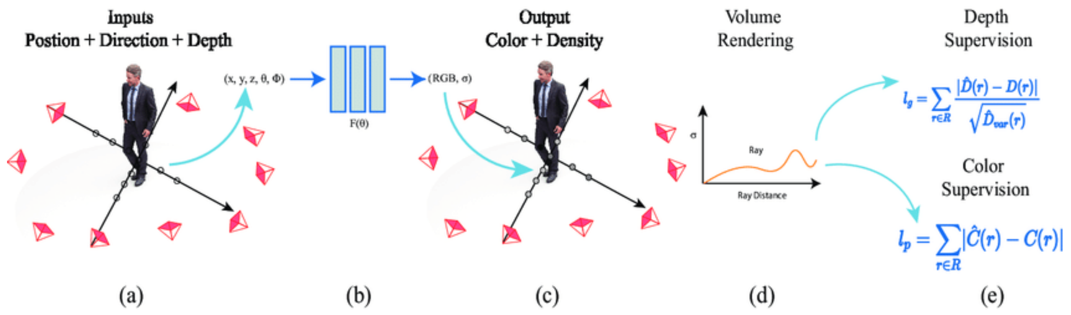


*1. NeRF Model Architecture Source*

We also explore the implementation of K-Planes NeRF, an optimized version of NeRF that uses explicit planes to represent 3D scenes. This superior method allows us to gain an approximate 1000x performance compared to the traditional NeRF approach. We now discuss the entire pipeline and implementation of these models.

### 3.1. Understanding NeRF Model

The NeRF model has a rather simple architecture. The core model consists of a fully connected deep and dense network having around 256 hidden layers. The earliest implementations of NeRF used Multilayer Perceptrons as their hidden units. The model takes in a 5D vector (x, y, z, θ, φ) as input. The input is (x, y, z) component are coordinates representing a point in 3D space. This point is usually the origin point for a ray in a w x h x 3 dimensional plane. The θ component represents a directional vector of a ray with respect to the centre of the 3D plane. x, y, z and θ can be used with another time component t to track the movement of the ray across a plane for a given timeframe t' essentially giving us a simplified version of ray tracing. The φ component is the angle of elevation of the ray relative to the x, y plane.


Inputs
Postion + Direction + Depth
(x, y, z, θ, Φ)
F(θ)
(RGB, σ)
Output
Color + Density
Volume
Rendering
σ
Ray
Ray Distance
Depth
Supervision
$l_g = \sum_{r\in R} \frac{|\hat{D}(r) - D(r)|}{\sqrt{\hat{D}_{var}(r)}}$
Color
Supervision
$l_p = \sum_{r\in R} |\hat{C}(r) - C(r)|$
(a)
(b)
(c)
(d)
(e)

*2. Inputs and Outputs of NeRF* *Source*

The NeRF model overfits to the scene data in order to memorize the entire 3D scene representation. This means there is only 1 NeRF model per scene and it does not follow the traditional deep learning concept of generalization. The output from the NeRF model is a 4D vector (R, G, B, alpha). The R G B component of the output represents the Red Green and Blue value for a point predicted by the NeRF model in the particular scene. The alpha component represents the accumulated transmittance which is used t represent the opacity of that point and occlusion produced by that point. The final image or novel view synthesis process of NeRF s performed through volumetric rendering wherein a novel 5D input that represents a non-existent view point is used as input. This point casts rays onto the representation and the model predicts the output and accumulations. This output is then rendered using volumetric rendering to reconstruct a novel image. Volumetric rendering can be mathematically represented as:

$$C(r) = \int_{t_n}^{t_f} T(t)\sigma\left(r(t)\right)c(r(t), d)dt \text{ where } T(t) = \exp\left(-\int_{t_n}^{t_f} \sigma\left(r(s)ds\right)\right)$$

Where $C(r)$ is the expected colour of camera ray $r(t) = o + td$ and $\sigma(x)$ is the differential probability of a ray terminating at an infinitesimal particle at location x. $T(t)$ is the accumulated transmittance of the traced ray. This rendering function can be differentially represented as:

$$\hat{C}(r) = \sum_{i=1}^{N} T_i \left(1 - \exp(-\sigma_i\delta_i)\right)ci \text{ where } T_i = \exp(-\sum_{j=1}^{i-1} -\sigma_i\delta_i)$$

Now we discuss the various data acquisition and preparation methods involved in training a NeRF model.

## 3.2. Data Acquisition: Structure from Motion Approach using Colmap

Colmap is an open-source software used to create sparse and dense 3D reconstructions from multiple images of a scene. It uses the photogrammetric process of extracting structure from motion using extraction and feature matching to provide sparse scene reconstructions.

### 3.2.1 Feature Extraction

Colmap utilizes a detector-descriptor apporach for feature extraction.

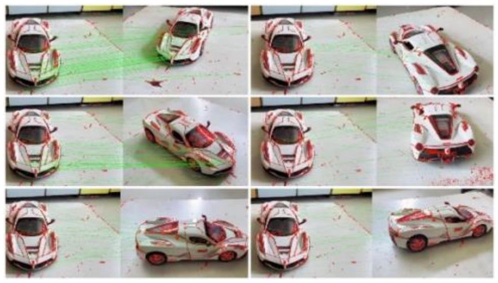
Detection: COLMAP leverages corner detectors like Harris corner detection or SIFT (Scale-Invariant Feature Transform). These algorithms identify points in the image where there's significant variation in intensity across neighboring pixels in multiple directions. These points often correspond to edges, corners, or blobs and are considered informative for reconstruction.

Descriptor Extraction: COLMAP employs a descriptor like SIFT or SURF (Speeded Up Robust Features) to create a unique mathematical representation of each feature's local neighborhood. This descriptor captures the key characteristics of the region around the feature point, including its intensity distribution and gradients. It's crucial for matching these features across different images despite variations in lighting, viewpoint, or even small occlusions.

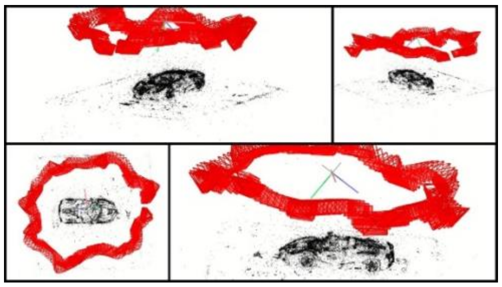
### 3.2.2. Feature Mapping

The extracted features are then exhaustively matched and overlapping images are found. This allows analysing the feature disparities across the various

*3. Feature Extraction*



*4. Exhaustive Feature Mapping*



*5. Sparse 3D Reconstruction*

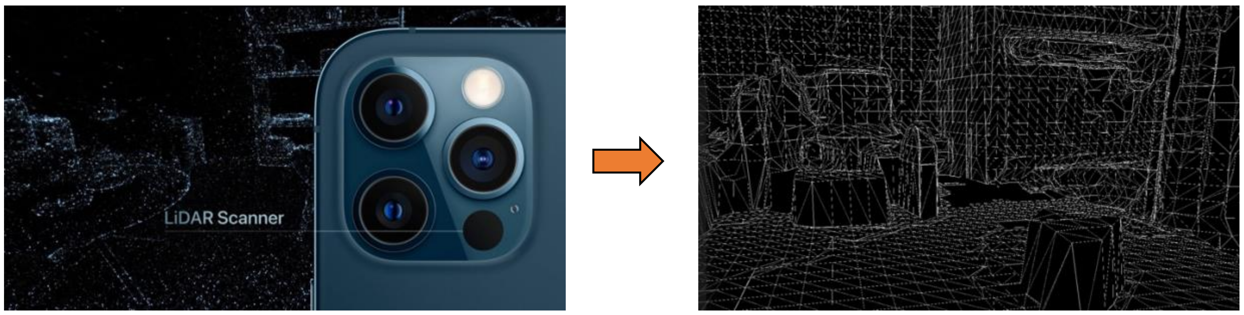
frames allowing the determination of structure of the scene based on feature disparity.

### 3.2.3. Sparse Reconstruction

After establishing feature correspondences across images, it estimates the 3D locations of these features (points) and the camera poses (position and orientation) for each image. This is achieved through iterative bundle adjustment, which minimizes the reprojection error - the difference between the locations of the features projected back from the estimated 3D points and camera poses, and their actual positions in the images. This process refines the initial estimates, resulting in a sparse 3D model consisting of these reconstructed 3D points and camera locations.

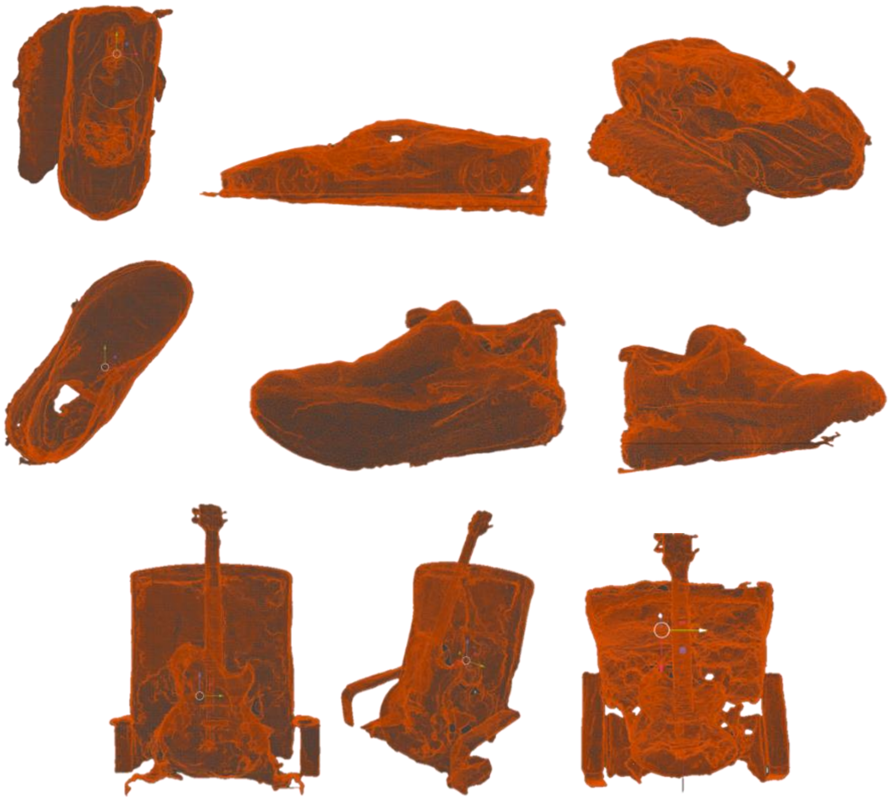
## 3.3. Data Acquisition: LiDAR Assisted

Light Detection and Ranging is a mapping technology that can capture the 3D structure of an object by calculating the time taken by a shot laser pulse to return to origin. LiDAR technology has been adopted by iPhones since the iPhone 12 line-up allowing convenient pose extraction and data creation of 3D objects. We used the application Polycam which utilises the LiDAR sensor to scan a 3D object and export it in the form of poses (x,y,z) and angles.

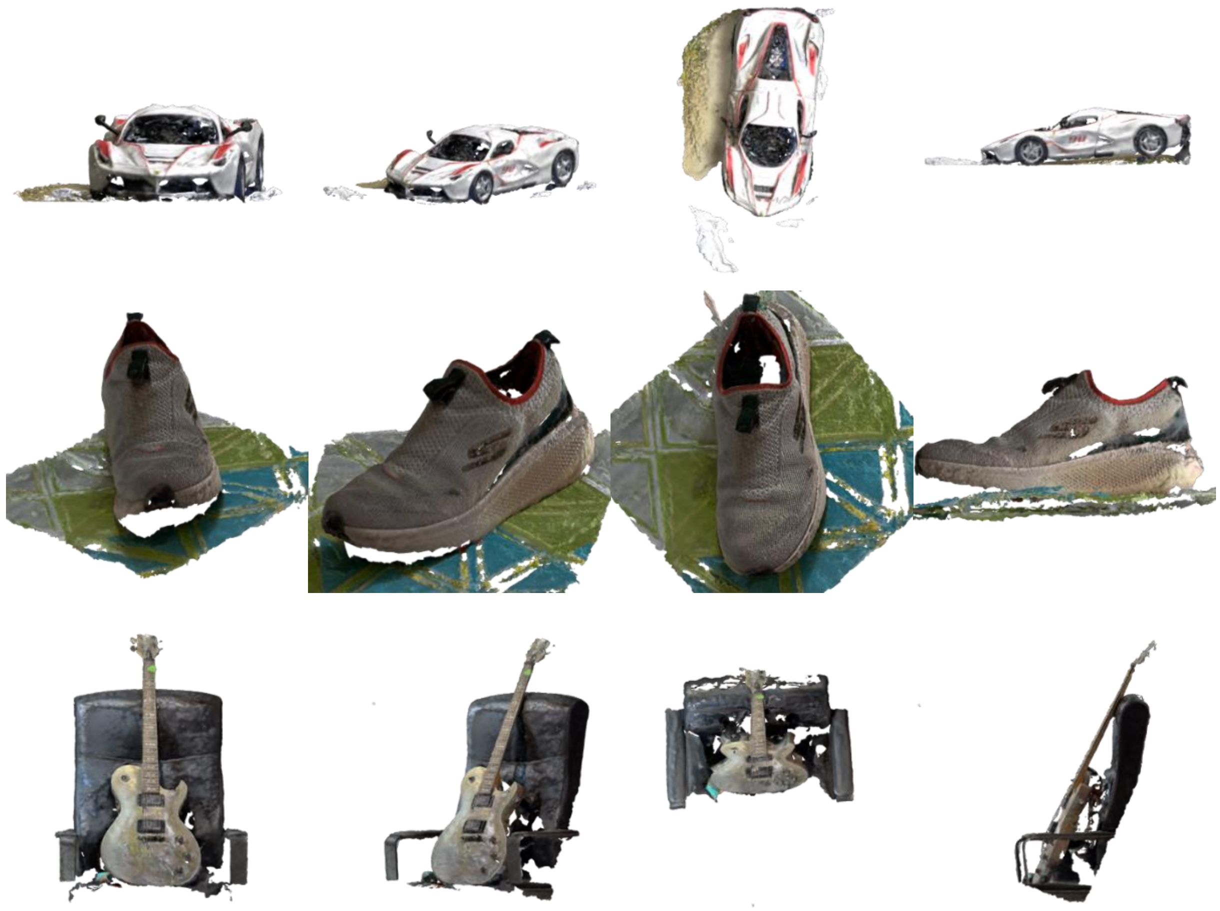


## 3.4. Optimization with K-Planes (WIP)

## 4. Results and Outputs



*4. Reconstructed Meshes*



*5. 3D Models with Materials*