LONDON'S GLOBAL UNIVERSITY
Department of Computer Science

Using Neural Radiance Fields (NeRFs) on Dark Cave Scenes

Suraj Kothari

Supervisors: Simon Julier

Faculty of Engineering

Department of Computer Science

University College London

A Project Report Presented in Partial Fulfillment of the Degree

Your degree title

December 2022

Abstract)

Abstract

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Keywords— keyword 1 - keyword 2 - keyword 3

I dedicate this ...

The Book of Nature is written in the language of mathematics.

— Galileo Galilei

Acknowledgements

Declaration

I, Name, I declare that the thesis has been composed by myself and that the work has not be submitted for any other degree or professional qualification. I confirm that the work submitted is my own, except where work which has formed part of jointly-authored publications has been included and referenced. The report may be freely copied and distributed provided the source is explicitly acknowledged.

12/09/22

Signature *Date*

Table of Contents

1 | Introduction

1.1 Motivation

2 | Background

This chapter introduces the reader to the general background concepts used throughout the project. Section 2.1 is about the COLMAP software, where we describe the pipeline and mention the two types of camera parameters. Then, in Section 2.2, we explain what NeRFs are and how they can be used to generate novel views of a 3D scene.

2.1 COLMAP

COLMAP is a Structure-from-Motion [1] software package that provides a pipeline to reconstruct a 3D scene from a set of 2D images. The images overlap and are taken from different viewpoints around a particular object, such as a building, tree, or car.

2.1.1 COLMAP Pipeline

We present the reader with the full COLMAP pipeline for structure-from-motion reconstruction in Figure 2.1. We will expand upon the most important sections of the process.

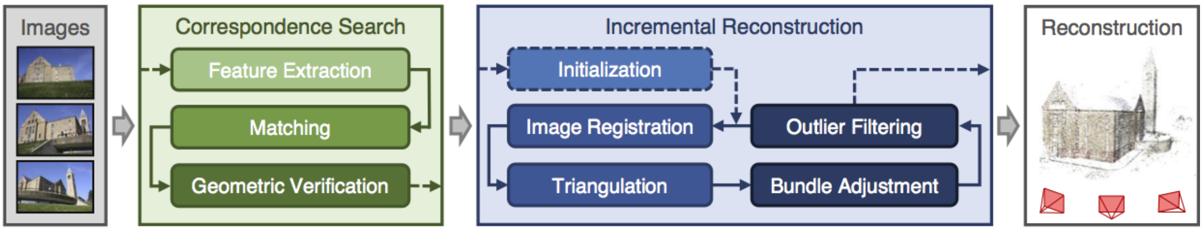

Figure 2.1: COLMAP Pipeline (Figure 2: [1])

Feature Extraction

The first stage of the pipeline is about detecting the particular features, such as points and corners, that appear in multiple images. This allows COLMAP to estimate the location and orientation of the camera relative to other images. One state-of-the-art method is SIFT [2] which is robust to noise, changes in illumination, and affine distortion. Additionally, this method can be used to match features across a wide range of scales and orientations.

Feature Matching

Using the features extracted in the previous stage, COLMAP searches for correspondences between features. They indicate which points in one image correspond to the same points in another image. The K-Nearest-Neighbour algorithm [3] is one technique that is used to find the k closest features in one image for each feature in another image.

Reconstruction initialisation

COLMAP initialises the pose estimation by choosing two images that are more likely to give a robust estimate of the pose. It will choose from a dense location in its image graph where there are many overlapping images.

Triangulation

Once COLMAP has estimated the poses for two images, it will solve for a new feature that is in these two images. This process is called triangulation because the two image poses are two vertices, and the new feature is the third vertex.

Bundle Adjustment

After estimating the poses for each image, COLMAP will optimise them further by minimising the reprojection error in a procedure known as bundle adjustment [4]. This is the error between the predicted 2D projection of a feature and its ground truth location in an image.

2.1.2 Extracted COLMAP Data

We will now outline the two types of camera parameters that COLMAP extracts after its reconstruction pipeline finishes.

2.1.2.1 Extrinsic Camera Parameters

Extrinsic camera parameters define the location and rotation of the camera for each viewpoint. These can be represented in a 4 × 4 matrix, also known as the pose matrix.

Given a 3D point p of the camera's initial location, the rotation and translation of this point is given by

$$\begin{aligned} p' &= \boldsymbol{R}_x(\theta, \boldsymbol{R}_z(\phi, p)) + \mathbf{t}p \\ &= \boldsymbol{R}_{xz}(\theta, \phi, p) + \mathbf{t}p \\ &= \begin{pmatrix} \boldsymbol{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \end{aligned}$$

, where θ is the angle of rotation on the x-axis and ϕ is the angle of rotation on the z-axis; rotation on the y-axis is not necessary.

In the second line, we simplify the equation to a single rotation matrix by multiplying the individual rotation matrices.

In the last line, we get the final pose matrix by combining the rotation and translation into a single pose matrix, which is done by concatenating the columns. A row, $[0, 0, 0, 1]$, is added to the bottom to ensure the matrix is square (4×4).

2.1.2.2 Intrinsic Camera Parameters

Intrinsics are parameters specific to the internal components of the camera used to capture the scene. COLMAP supports multiple types of cameras, including Pinhole, Radial, and Fisheye. The default is a Radial camera, and its intrinsics are focal lengths, optical centers, and the skew coefficient. These can be represented in the following 3×3 matrix

$$\begin{pmatrix} f_x & k & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

The focal length for a camera can differ for the x-axis and y-axis, but COLMAP assumes they are the same and returns one value. The optical centers will be assumed to be half the resolution ($c_x = WIDTH/2, c_y = HEIGHT/2$). The skew coefficient is the amount by which a pixel is shifted in the x-axis. It is 0 if the axes are perpendicular.

2.2 Neural Radiance Fields (NeRFs)

NeRFs (Neural Radiance Fields) are fully-connected deep neural networks that generate novel views of 3D scenes by training on a sample set of 2D images and then predicting the opacity and radiance at each point in space. Radiance is a measure of the intensity of light emitted by an object

The input to the neural network is a 5D vector, $[x, y, z, \theta, \phi]$, where $[x, y, z]$ is the location of a point in the scene and $[\theta, \phi]$ is the viewing direction.

The output is a 4D vector, $[R, G, B, \sigma]$ where $[R, G, B]$ is the emitted radiance and σ is the volume density, which is the opacity at the point and controls the amount of radiance emitted. The radiance is a function of both the location $[x, y, z]$ and viewing direction $[\theta, \phi]$. This means that as the viewing angle changes, the radiance also does, which enables the rendering of specular reflection.

2.2.1 NeRF Dataset

To train a NeRF, we need a dataset with a specific format. This includes a set of RGB images of the scene taken from multiple viewpoints, the extrinsic camera parameters, which are 4x4 pose matrices corresponding to each image, and the intrinsic camera parameters, which in this case is just the focal length. The extrinsic and intrinsic camera parameters are discussed in more detail in Section 2.1.2.

2.2.2 Training a NeRF

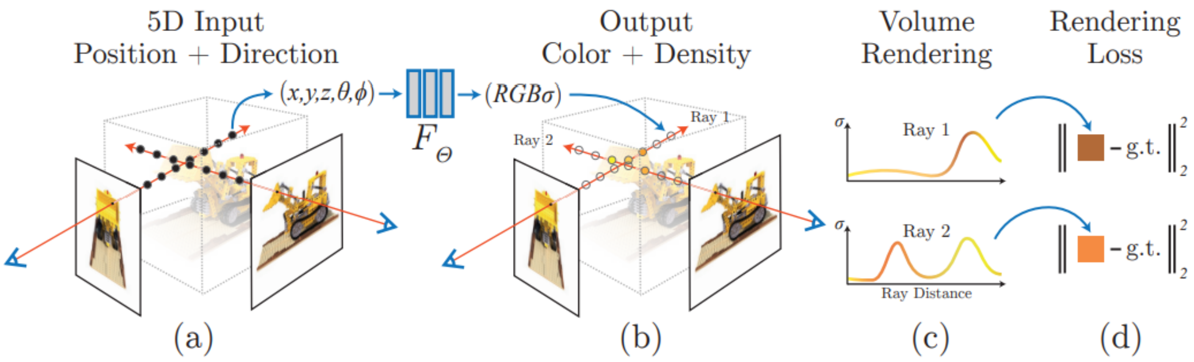

Figure 2.2: Steps to render a novel view (Figure 2: [5])

There are 4 steps to training a NeRF and then rendering a novel view from the model. These are labelled a, b, c, and d in Figure 2.2.

In step (a), we send a camera ray through the scene for each pixel in the view. We then sample 5D coordinates $[x, y, z, \theta, \phi]$ along the camera ray and input them into the model.

Next, in (b) the model outputs its prediction of the volume density σ and the view-dependent radiance $[R, G, B]$.

Then, in step (c), we use classical volume rendering techniques [6] to integrate the function of density and radiance between the near t_n and far t_f bounds of the ray. This results in a $[R, G, B]$ colour for each pixel.

Finally, in step (d), we compute the loss which is the squared error between the predicted pixel colour from step (c) and the ground truth from the input image. We minimise the loss using gradient descent, resulting in a trained NeRF that can generate novel views.

2.2.3 Optimisation of a NeRF

The 4 steps we described in 2.2.2 would result in a basic NeRF model. To model more complex scenes, the model needs to be able to handle high-frequency variations such as sharp changes in lighting and shadows, fine-grained details (e.g. tree leaves), or small bumps/cracks on a surface (e.g. pebbles, dirt).

We can optimise a NeRF using two techniques. Positional encoding which creates a high-frequency function, and Hierarchical sampling which allows us to sample from this high-frequency function.

2.2.3.1 Positional Encoding

When a NeRF is trained using the original 5D input coordinates, $[x, y, z, \theta, \phi]$, it is unable to learn high-frequency variations. This is due to the fact that deep neural networks tend to be biased towards learning low-frequency functions [7].

A clever trick is to map the input to a higher dimensional space and then pass this encoded input to the NeRF allowing it to learn high-frequency changes. The mapping is performed using a composition of sine and cosine functions. Given an input point p, its encoding is

$$\texttt{ENCODING}(p) = (sin(2^0\pi p), cos(2^0\pi p), sin(2^1\pi p), cos(2^1\pi p), \cdots, sin(2^{L-1}\pi p), cos(2^{L-1}\pi p))$$

, where L is a hyper-parameter for the number of pairs of sines and cosines used.

2.2.3.2 Hierarchical Sampling

In most 3D scenes, the volume is sparse, meaning that many points don't contribute much to the final rendered view. The simple approach to evaluating the integral for volume rendering is to use stratified sampling, which samples N points uniformly along the ray. However, this results in sampling unnecessary points in free space or regions hidden from the camera.

Hierarchical Sampling is a technique that proportionally over-samples parts of the ray that have a high likelihood of contributing to the final render. In Figure 2.3, the red ray has more samples in areas that correspond to useful objects/features in the scene and fewer in areas that are sparse.

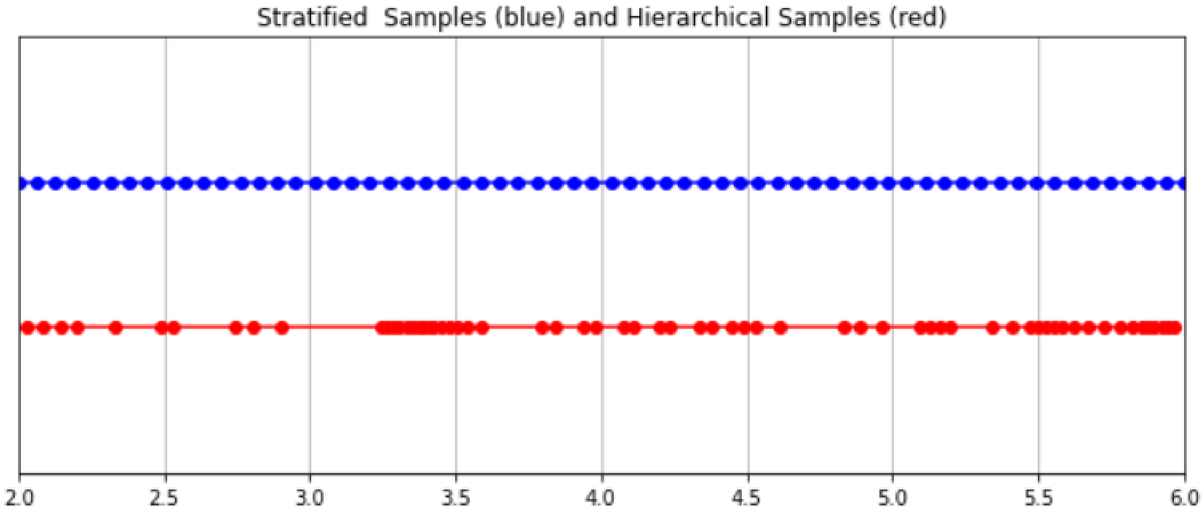

Figure 2.3: Stratified vs Hierarchical sampling

3 | Context

4 | Requirements and Analysis

5 | Design and Implementation

5.1 Post-processing of Extracted COLMAP Data

After COLMAP has completed its reconstruction, it stores the extrinsic and intrinsic parameters in binary files. We need to convert these to text files which can be done using the provided binary-to-text tool in COLMAP. After that, we can read the text files and extract the necessary data. required for training a NeRF.

5.1.1 Extracting Data from Text Files

The intrinsic parameters are stored in the `cameras.txt` file and the extrinsic parameters are stored in the `images.txt` file.

5.1.1.1 Cameras Text File Data

Here, we have shown the data from the `cameras.txt` file in a table. We only need to extract the focal length parameter from this file.

CAMERA_ID	MODEL	WIDTH	HEIGHT	FOCAL	C_X	C_Y	K
1	SIMPLE_RADIAL	3072	2304	2558.42	1536	1152	-0.020

5.1.1.2 Images Text File Data

Here, we have shown the data from the `images.txt` file in a table. We need to extract the quarternions s

Image_ID	QW	QX	QY	QZ	TX	TY	TZ	CAMERA_ID	FILENAME
1	0.862	0.0185	0.485	-0.146	-0.677	1.002	3.645	1	P1180141.JPG

5.1.2 Getting Image Dataset of Scene

To Do: - Insert code for getting images

5.1.3 Creating Pose Matrix from Extrinsics

To Do: - Insert code for creating pose matrix

5.1.4 Visualising Pose Data

To Do: - Insert code for visualising (include reference) - Insert pose data visualisation images - Compare tractor and building visuals.

6 | Testing

7 | Conclusion

References

[1] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 11 2004.

[3] C. Zhao, Z. Cao, C. Li, X. Li, and J. Yang, "Nm-net: Mining reliable neighbors for robust feature correspondences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[4] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment — a modern synthesis," in *Vision Algorithms: Theory and Practice* (B. Triggs, A. Zisserman, and R. Szeliski, eds.), (Berlin, Heidelberg), pp. 298–372, Springer Berlin Heidelberg, 2000.

[5] 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," *Communications of the ACM*, vol. 65, pp. 99–106, 1 2022.

[6] J. T. Kajiya and B. P. V. Herzen, "Ray tracing volume densities," *ACM SIGGRAPH Computer Graphics*, vol. 18, pp. 165–174, 7 1984.

[7] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310, PMLR, 09–15 Jun 2019.

A | System Manual

B | Supporting Documentation

C | Project Plan

D | Interim Report

E | Code Listing