# DLCV HW4 Report

NTUEE
B10901187
Hung Kai Chung

Dec. 2023

## Problem : 3D Novel View Synthesis

### 1. Please explain: (a) the NeRF idea in your own words (b) which part of NeRF do you think is the most important (c) compare NeRF's pros/cons w.r.t. other novel view synthesis work

(a) **NeRF**(Neural Radiance Fields) is a model to create novel view synthesis. The concept of the NeRF is using a model to learn the implicit representation of the given object from differenct view. The prerequisite knowledge of NeRF is ray tracing. In Fig 1, the camera actually can see is the light after the point p emit and pass through the volume cloud. Hence, we can imagine that the ray shoot from the camera may go through many volume cloud, so the NeRF have to get each point of the density to get the final color the camera can see. The formula of the final color is in Fig 2. Due to the computing limitation, we used the **Hierarchical Sampling** to replace integral computation. Fig 3 is overall process of NeRF model training, given images from different camera direction, we can using Hierarchical sampling like in Fig 4, feed these points to the model and get the rgb color & volume density, after that using volume rendering to do alpha accumulation to get final color. NeRF use MSELoss to do training.

(b) I think the volume rendering & view-dependent color is the most important part in NeRF, volume rendering is essential for generating realistic images by simulating how light interacts with the 3D scene. The volume rendering formula is in Fig 5. Notice that the NeRF only considers the light at that time in the img. As NeRF is not modeling the reflected part, it is not possible to relight scenes using NeRF.

(c) DVGO uses explicit voxel grid to represent 3D. It reduces the training time to 15 min compared to NeRF which needs at least 10 hours. Instant-ngp get idea from NeRF Positional Encoding, if we can efficiently get the encode feature, we can use less MLP layer to get color. Instant-ngp used Multiresolution Hash Encoding, using different resolution to encode, thus getting larger feature and doing the NeRF process(adding MLP and do volume rendering to calculate MSELoss). Also the Hash table can also backpropagate by the loss.

**Pros:**

- Training NeRF didn't have to get the 3D model, just using different view of 2D image.

**Cons:**

- Training NeRF needs many times
- The model only applicable to a single scene.

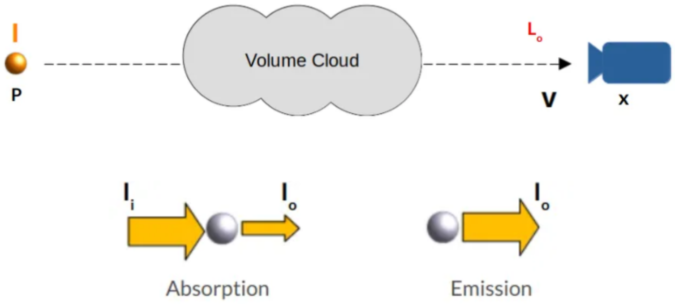


Figure 1: Ray Tracing

$$\bar{c} = \int_{t_n}^{t_f} T(t)\sigma(t)c(t)dt$$

Figure 2: Color Formula

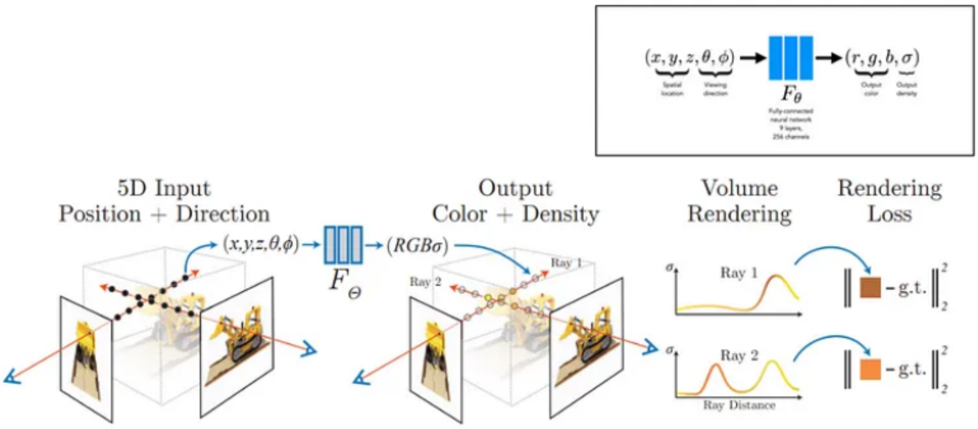


Figure 3: NeRF Trainging process

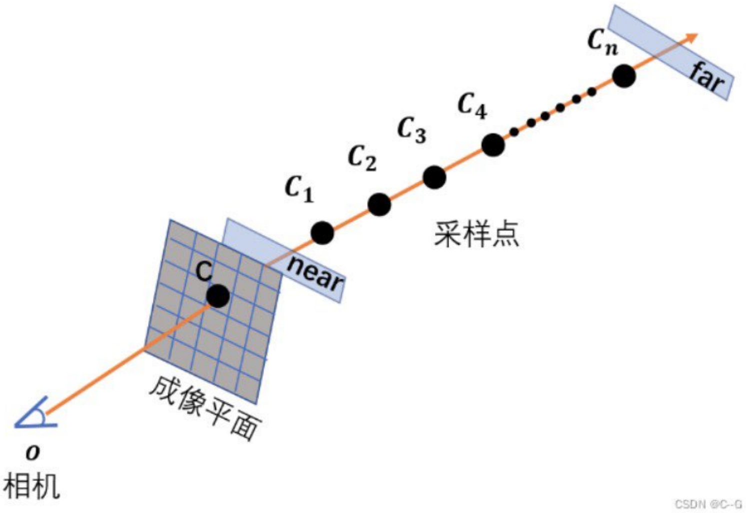
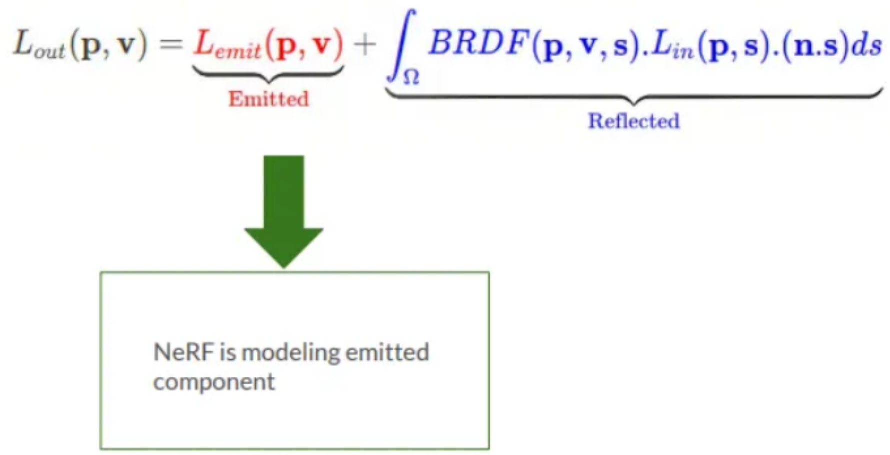


Figure 4: Sampling



$$L_{out}(\mathbf{p},\mathbf{v}) = \underbrace{L_{emit}(\mathbf{p},\mathbf{v})}_{\text{Emitted}} + \underbrace{\int_{\Omega} BRDF(\mathbf{p},\mathbf{v},\mathbf{s}).L_{in}(\mathbf{p},\mathbf{s}).(\mathbf{n}.\mathbf{s})ds}_{\text{Reflected}}$$

Figure 5: Rendering Equation

## 2. Describe the implementation details of your NeRF model for the given dataset. You need to explain your ideas completely.

The meta.json provides the pose and view direction of the camera and the rotation matrix and so on. We can get the [coordinate, direction, far, near] of each rays in 2D images. Second, we have to sample **coarse** points from each rays by using uniform sampling after that using inverse transform sampling to get **fine** points. After we get coarse & fine points, we employ positional encoding to the coordinate & view direction, which let the model can learn high frequency scene. Finally, we can use the NeRF model in Fig 6, notice that NeRF model has two distinct output branches, due to the volume density don't influence by the view direction. We can use the output rgb & sigma to do volume rendering in order to get final color and then using MSELoss to train the model.

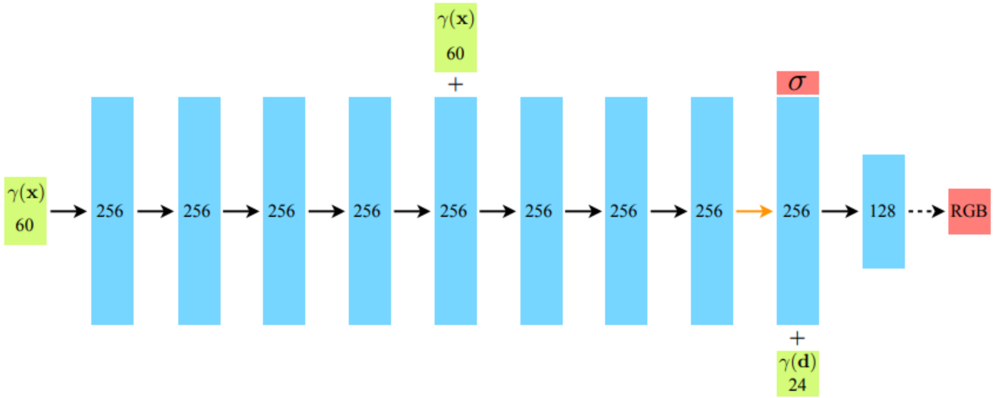


Figure 6: NeRF model Architecture

### 3. Given novel view camera pose from metadata.json, your model should render novel view images. Please evaluate your generated images and ground truth images with the following three metrics (mentioned in the NeRF paper). Try to use at least three different hyperparameter settings and discuss/analyze the results.

The meaning of three metrics:

- PSNR(Peak Signal-to-Noise Ratio): calculate the max & MSE using log, it will increase when MSE decrease.
- SSIM(Structural Similarity): calculate the luminance, contrast & structure
- LPIPS(vgg): Computing Perceptual Distance by using neural network

Settings:

- A: Coarse Sampling + Ranger optimizer
- B: Coarse & Fine Sampling + Ranger optimizer
- C: Coarse & Fine Sampling + Adam optimizer

| setting | PSNR | SSIM | LPIPS(vgg) |
|---|---|---|---|
| A | 34.70 | 0.969 | 0.1756 |
| B | 37.47 | 0.981 | 0.1575 |
| C | 30.43 | 0.9354 | 0.2532 |

Discussion:
The three different setting above show apparent results in the metrics. I found that the Hierarchical Sampling have a huge effect on the model compared with setting A & B. Compared with setting B & C, I found that using Ranger optimizer can do more well on the task.

## 4. With your trained NeRF, please implement depth rendering in your own way and visualize your results.

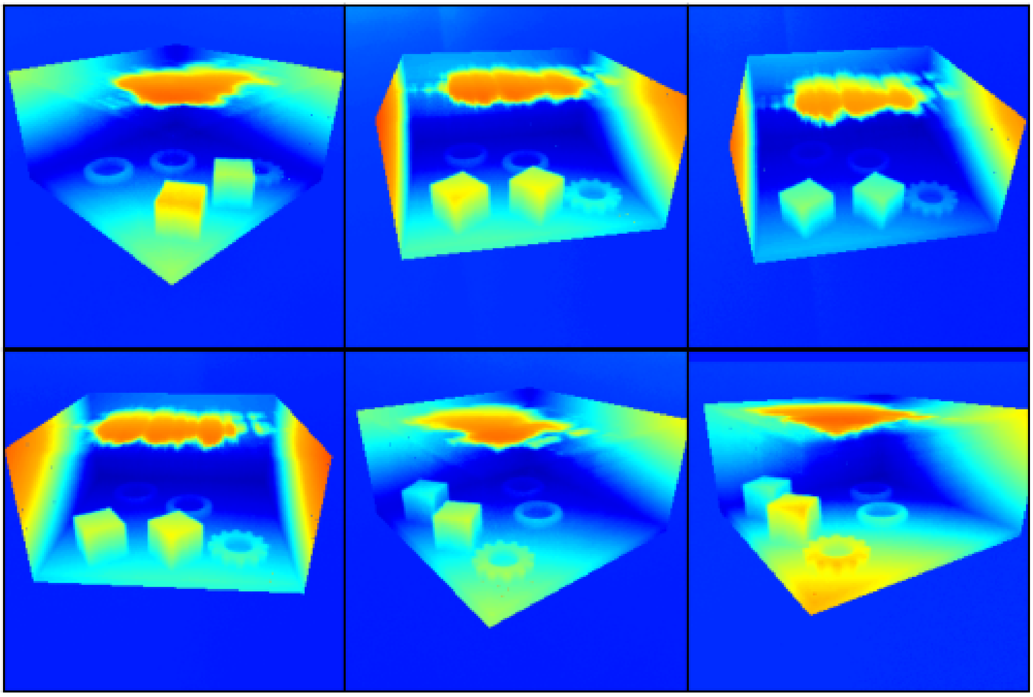


Figure 7: Depth rendering img