# Abstract

The neural radiance field (Nerf)[1] has been proven to be a feasible solution to the novel view synthesis problem. But there are many limitations in practical application. One of them is that the pose matrix information of the image is required in the training process of Nerf. The report proposes a 6Dof poses estimation algorithm that could make Nerf capable of building implicit representation with one known pose matrix. We propose Transformation Net, a neural network architecture that uses the optical flow fields of two images to predict the transformation between two images. The report also illustrates the relationship between transformation and the optic flow field and shows that the features of the optic flow field are useful for transformation prediction.

**KEYWORDS**

**Domain adaptation, UOSA, visualisation technique**

Table of content

[Abstract i](#_Toc133282021)

[Chapter1 Introduction 1](#_Toc133282022)

[Chapter2 Related work 2](#_Toc133282023)

[1.1 Neural Radiance Field 2](#_Toc133282024)

[1.2 Nerf-mm 2](#_Toc133282025)

[1.3 Bundle adjusting Nerf 2](#_Toc133282026)

[1.4 FlowNet2 2](#_Toc133282027)

[Chapter3 Methodology 3](#_Toc133282028)

[3.1 Pose matrix 3](#_Toc133282029)

[3.2 Transformation between two coordinate systems 3](#_Toc133282030)

[3.3 Relation between optic flow field and transformation 3](#_Toc133282031)

[3.4 Transformation Net 4](#_Toc133282032)

[Chapter4 Dataset and data preprocessing 6](#_Toc133282033)

[4.1 Dataset 6](#_Toc133282034)

[4.2 Data Preprocessing 6](#_Toc133282035)

[4.3 Data Augmentation 7](#_Toc133282036)

[Chapter5 Results and visualization 8](#_Toc133282037)

[5.1 Experiment results 8](#_Toc133282038)

[5.1 Result visualization 8](#_Toc133282039)

[Chapter6 Key insights 9](#_Toc133282040)

[Chapter7 Future work 10](#_Toc133282041)

[References 11](#_Toc133282042)

Chapter1   
Introduction

How to effectively represent geometric objects has always been a hot topic in the field of graphics; To a large extent, the representations of geometric objects are divided into two types: explicit and implicit representation. The explicit representation shows how to construct the 3d object by giving the mapping functions of points, the point cloud and laser scan methods are two commonly used explicit cases. While the implicit methods describe the relation between points and target, such as the distance function from the surface and the level sets. The traditional investigations focus on the explicit representation of geometric objects, but the Neural radiance field (Nerf)[1] was learned from the Plenoptic Function, using the neural network to simulate the implicit radiance field from a series of 2D images. The success of their work started a wave of modeling using implicit representations

Nerf is a sensational work, but it has to know the information of the pose matrices of the images. The Local Light Field Fusion (LLFF) dataset[5], which was used in the training process of Nerf, estimated the pose matrix information with the help of Colmap, a software that could reconstruct the geometric information through feature mapping. Two subsequent researchers wanted to replace the utility of Colmap with neural networks, the nerf-mm proposed by Wang, Z et al.[2] integrate the learning of extrinsic information (Pose matrix and focal of the lens) into the training process of Nerf, while the Barf proposed by Lin, C. H. et al.[3] is motivated by mathematical insights, using bundle adjusting to correct the pose matrix.

These works have yielded great results, but the learning of the pose matrix is largely bound with the training of Nerf, which means they may lack generalization(overfitting). It may need to be retrained with Nerf in new scenes. This project wants to establish a two-step training process. We proposed a transformation net that would learn the rotation and translation between two images from the optic flow field. Then the model could be further fine-tuned by integrating it with Nerf.

The project also makes use of mature research in the optic flow field generate area. The work flownet2[4] investigates how to generate optic flow fields from two images. They design two kinds of neural network structures FlowNetS and FlowNetC for feature extracting and feature matching responsively. Then the neural networks are stacked together to generate a high-precision optic flow field.

Chapter2   
 Related work

## 1.1 Neural Radiance Field

Neural radiance field[1] is a concept of simulating the Plenoptic Function with a neural network, which was proposed by Mildenhall, B. et al. There are mainly two steps in the training process. In the first step, the neural network would receive spatial location (x, y, z) and viewing direction (θ, φ)) as input, then predict volume density and view-dependent emitted radiance (or the RGB color) at that spatial location. The second step is volume rendering. In theory, the cumulation of emitted radiance along a directed ray will finally be reflected as a pixel in the image. The algorithm would sample points along the ray directed from image pixel to focal, using the loss between RGB of pixel and cumulated radiance to supervise the training of the neural network. Though this is a work with a solid foundation, there are too many limitations in implementation. For instance, it can’t be applied to dynamic scenes, the process of sampling points is time-consuming, and the pose matrices of images must be known.

## 1.2 Nerf-mm

Some of the researchers want to make Nerf capable of training without known camera information. Nerf-mm[2] is a successful one among them. This work uses another two neural networks to estimate the Pose matrix and the focal length, then the neural radiance field would be generated on the estimated parameters. There is only one loss function in this process: the difference between ground truth pixel color and rendered color. Though using neural networks in estimation tasks is an experiential approach. It may lead to the cool start problem as the initial position is unknown, they use Colmap in the early training stage to solve the cool start problem. Another thing is that the learning of intrinsic information is highly bound with the training of Nerf, it may need to be retrained with Nerf in new scenes.

## 1.3 Bundle adjusting Nerf

Bundle adjusting Nerf(Barf)[3] is the concurrent work of Nerf-mm, they want to adjust the Nerf model from imperfect or unknown intrinsic information. The learning of the Pose matrix is also learned from the training of Nerf, but the backpropagation function is inferred from the 2d/3d image-aligned function. Besides the backpropagation function, they also improve the encoding method to make the backpropagation works better.

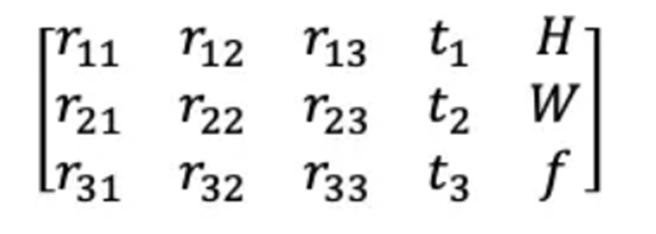
## 1.4 FlowNet2

The flownet2 is one of the mature researches that aim3 to generate the optic flow field from two input images, there are mainly two neural network structures used in FlowNet2, the FlowNet\_Simple that directly extracts features from the stacked images (H, W, 6), and the FlowNet\_Corr that would calculate the correlation between two images. In the research, they found that the flownet2 would have the best performance when adding two FlowNet\_Simple blocks after the FlowNet\_Corr block. The model could generate optic flow fields for both the slightly changed image pairs and the huge changed image pairs.

Chapter3   
 Methodology

## 3.1 Pose matrix

**Figure 3-1: The format of the pose matrix**



In the Local Light Field Fusion (LLFF) dataset, the pose matrix is a 3\*5 format matrix, in which the first three columns represent the x, y, and z axes of the camera direction in the real-world coordinate system, the fourth column is the position of the camera, while the last column represents the intrinsic of the camera (focal length, height and width of image)

## 3.2 Transformation between two coordinate systems

The project aims to estimate the transformation of the camera coordinate system. The transformation of the camera could be decoupled into one translation and one rotation. Assume p1 is the position of points in the camera coordinate system 1, likewise, p2 represents the position of points in the camera coordinate system 2.

Then the transformation could be written as

(1)

In which T and R represents the translation vector and rotation matrix respectively. Then It can be expanded in the following format.

(2)

We could also get the reverse transformation matrix

(3)

## 3.3 Relation between optic flow field and transformation

We could easily map the position in the camera coordinate 2 to that in the pixel image coordinate of camera 1 with the following formula.

(4)

U and V represents the position in pixel image, and means the intrinsic, which is assumed to be known in this project. and are the centre point of image, which is easy to get. The corresponding pixel position in the camera 2 could be simply written as

(5)

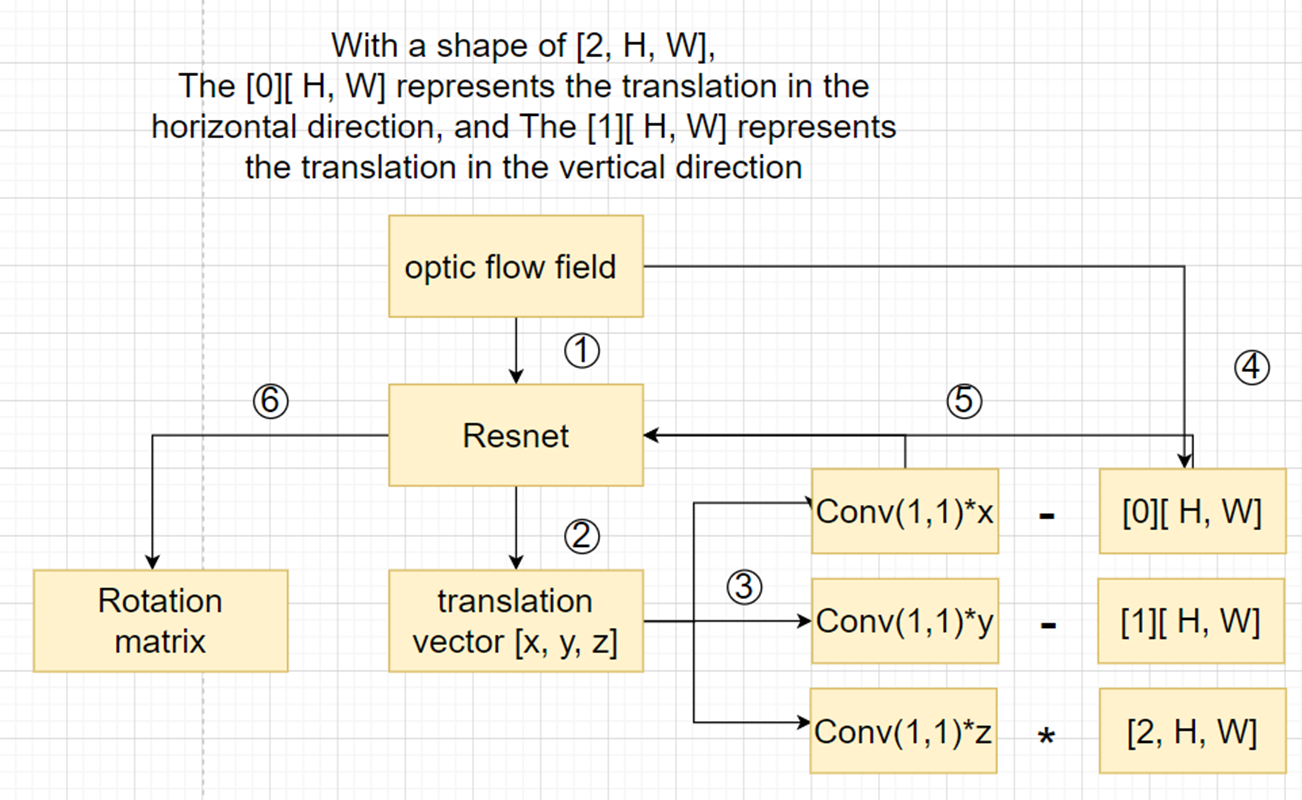
Then the optic flow could be represent as

(6)

In other words, the optic flow is highly correlated with the transformation, what we’d like to do is to decouple the rotation matrix R and translation t from the optic flow with the help of a neural network. The problem is that the depth information would be lost when we map the real scene into a picture, which means that we don’t know the precision of real-world coordinates (x, y, z), this could be solved by introducing the depth neural network or Nerf into our neural network structure, both the method would mitigate the loss brought by the missing depth information.

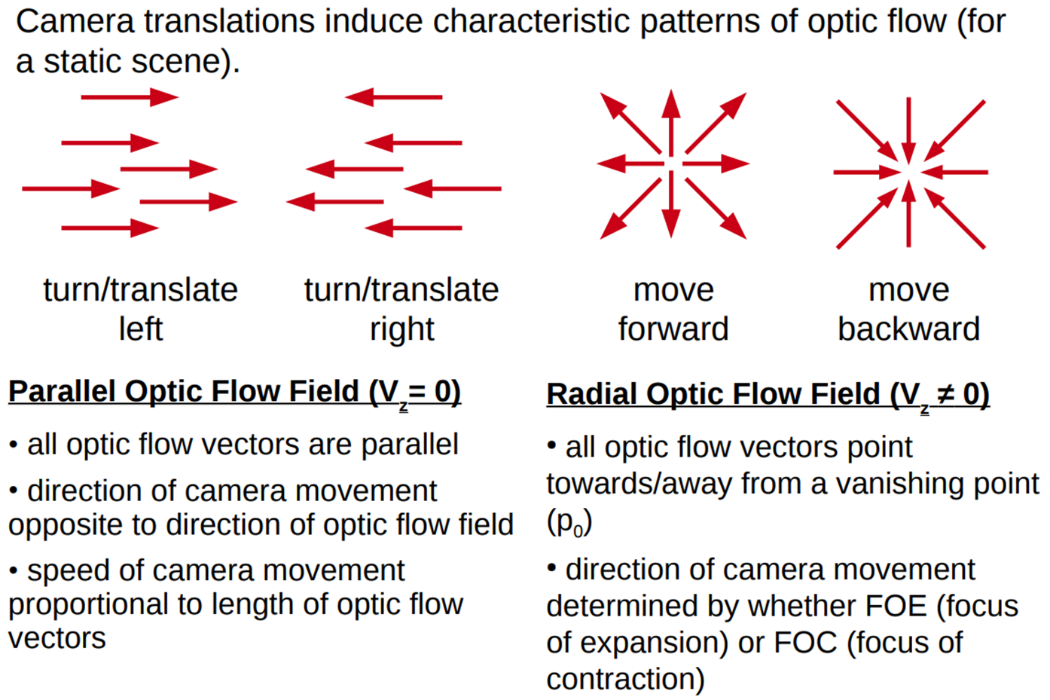
## 3.4 Transformation Net

Figure 3-2: Structure of Transformation Net



In the first stage, Two photos of the same static scene would be fed to FlowNet2. The output of FlowNet2 is the optic flow field tensor with the shape of (2, H, W); The values in the two-dimensional vector represent the translation of pixels in horizontal and vertical directions. The optic flow tensor would be regarded as the input of our Transformation Net, and it will output the estimated translation vector and rotation matrix. The Residual Net is chosen as the backbone of the Transformation Net, as it could integrate multi-scale features. The neural network used for translation estimation and that for rotation estimation have shared weights in the first three layers because the high-level features of translation and rotation should be the same. The network would first estimate the translation vector, after which, we would remove the effect brought by the translation from the optic flow tensor. This is easy as the translation of the camera will have a simple representation in the optic flow field.

**Figure 3-3: The patterns of translation**



As what was shown in Figure 3-2. We’d like to subtract the estimated horizontal translation x and vertical translation y from the optic flow tensor with a certain magnitude. Then the whole optic flow tensor would be divided by a certain magnitude to remove the optic flow brought by translation in the z-direction. The magnitude (weight) is also learned through the neural network, with the help of 1\*1 convolutional layers. Then the remained optic flow tensor would only be generated from rotation, so it would be used to estimate the rotation. The MSE loss between the true value and the estimated value was used to supervise the neural network.

Another point worth noting is that the activation layer used in the last layer is the tanh function as the value in the translation vector and rotation matrix ranges from (-1, 1). But the gradient value would rapidly grow if the initial weights of the neural network are too high, the strategy used to control the gradient is by dividing the parameter in the tanh function by a certain magnitude to make the tanh function won’t easily converge to 1 or -1.

Chapter4   
 Dataset and data preprocessing

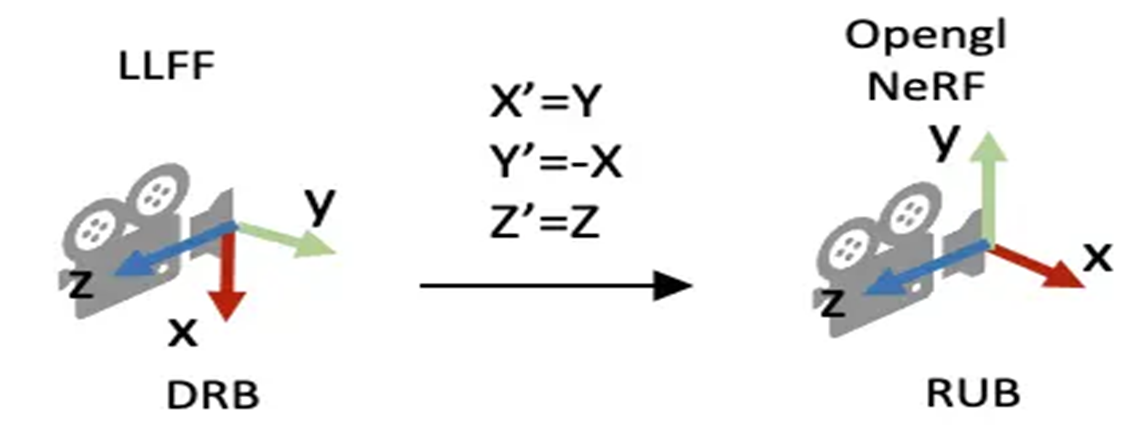
## 4.1 Dataset

The dataset used in this project is the same as that used in Nerf project. The Local Light Field Fusion(LLFF)[5] dataset already satisfies all the requirements of this project. There are totally eight folders in this dataset, and each one of them contains a series of images of the same static object taken from different angles and positions. The LLFF dataset also contains intrinsic and extrinsic information about the camera and images, which could be used to supervise the training of transformation.

## 4.2 Data Preprocessing

The pose matrix (extrinsic) in the LLFF dataset is using the Colmap camera coordinate system, to make it works in the OpenGL coordinate system, the x and y axes would be reversed first.

**Figure 4-1: The coordinate change**



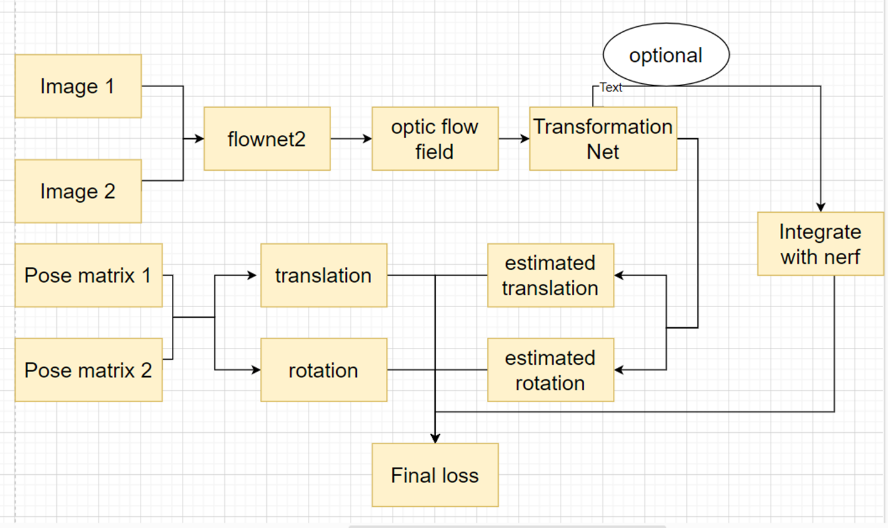
After which, all the pose matrices would be mapped inside the unit sphere after centralized and normalizing operations. We get the pose matrix in the format

(7)

For each pair of images, the rotation could easily be calculated with the following formula

(8)

**Figure 4-2: The data preprocessing pipeline**

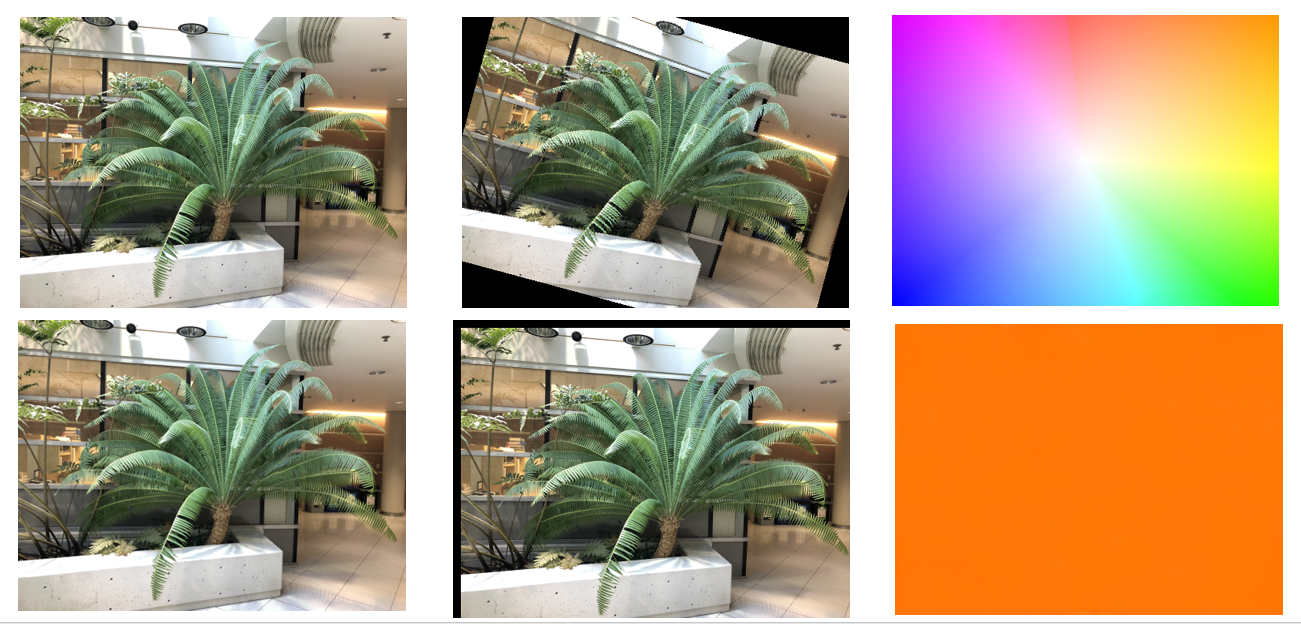


Then with a single subtraction , we get the translation vector. We used Flownet2 to generate optic flow fields for image pairs, the optic flow fields are bundled as training inputs, while [R, t] would be bundled as target sets. These data would be fed to the transformation net.

## 4.3 Data Augmentation

We believe that the neural network should learn the easy pattern first; so that it could have better performance in complicated transformation estimation tasks. To achieve this, we apply simple augmentation to the images in the dataset, the augmentations include the random rotation and translation along the x and y axes. From the figure below, we could observe the pattern of these augmentations from their optic flow field.

**Figure 4-2: The data augmentation and corresponding optic flow field**

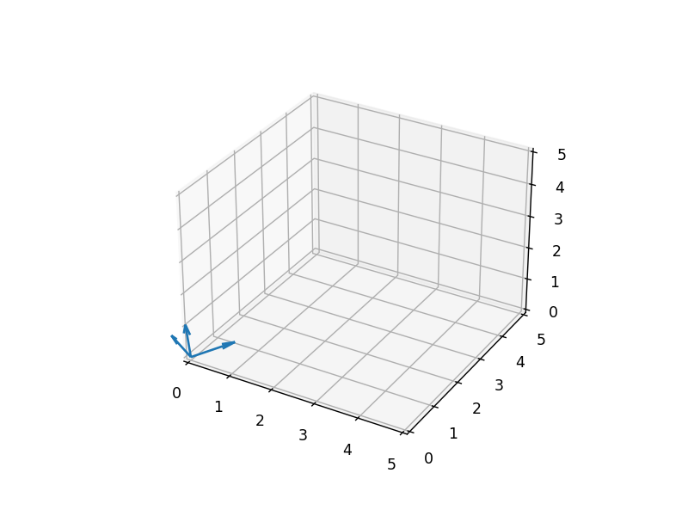
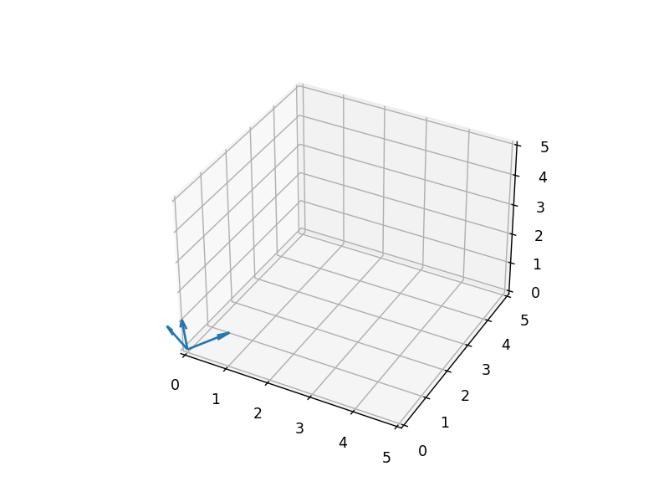


The rotation could be simulated with a 3d rotation matrix along the z-axis, with the following format.

(9)

The rotation reflects on the camera coordinate system, which is visualized in Figure 4-3

**Figure 4-3: The rotation in the coordinate system**

Chapter5   
 Results and visualization

## 5.1 Experiment results

With the combination of the original dataset and the augmented data, we finally get thousands of training image pairs. The neural network was trained on these training image pairs for about 500 echoes. The training loss ends up bouncing around 0.02, and the validate loss is around 0.07. We cannot tell whether it is fully converged. The result is not perfect enough, but the results show that it has better performance in translation prediction, the rotation estimation is not that well, the reason could be that the parameters needed to be estimated in rotation triple that in translation.

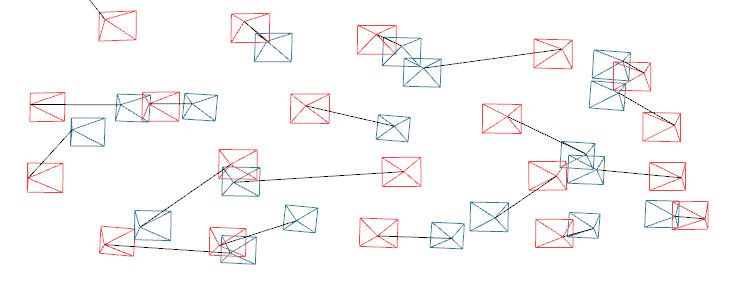
**Form 5-1:validation loss**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Fern | Flower | Horns | Leaves |
| Translation loss | 0.0086 | 0.0102 | 0.0076 | 0.0113 |
| Rotation loss | 0.0597 | 0.0644 | 0.0615 | 0.0568 |

## 5.1 Result visualization

We’d like to visualize the original camera poses and our estimations so that we could see the error between the estimated value and the true value. The visualization is based on opencv3d, some of the visualization code is modified from nerf-mm. We started from one known pose matrix and use it to estimate poses for all the other image frames,

**Figure 5-2:visualization of estimated poses**



The images show the true position from Colmap(red) and the estimation of the camera position(blue) based on one known pose matrix. As you can see, the estimation error of the pose matrix will accumulate as we continue to calculate the transformation matrix based on the previously estimated pose matrix. This means that our method has much room for improvement.

Chapter6   
 Key insights

The idea of the project is that integrating the pose estimation with Nerf directly could be time-consuming and also cause overfitting problems. What we are trying to do is to use a prior method to train a neural network that could be capable of estimating the camera transformation, then further integrate it with Nerf to fine-tune the model. During the period of the project, we tried to find the approaches that are helpful for transformation estimation. It appears that the solutions are buried in what we’ve learned in the past courses. We found that most of the research used feature-matching before the estimation, so it is easy to find that the optic flow field itself is also a feature-matching algorithm, With the mathematical derivation we also find the relationship between the optic flow field and camera transformation, so it was finally decided to use it in our project. Fortunately, mature research like flownet2 makes it possible for us to implement the idea. Thanks to past workers for their contributions.

Chapter7   
 Future work

As we’ve discussed in the result section, there are still a lot of problems in this project. We’d like to go through these obstacles in future work. The first thing that matters is that the data used in the LLFF dataset is also estimated with the help of Colmap, which means that the information on pose matrices could be inaccurate. Only when we get a dataset that has ground truth data, we could improve the performance of the transformation Net. Another thing is that estimating all the other pose matrices based on only one pose matrix could lead to the accumulation of errors, which will lead to terrible estimations in the last few frames. We should integrate information from more images.

So far, we haven’t integrated the Transformation Net with Nerf because of the time limitation, we consider completing this section in the future to see whether this approach could help us improve the performance of the transformation net.

References

1. Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, *65*(1), 99-106.Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S.: Transfer feature learning with joint distribution adaptation. In: ICCV (2013)
2. Wang, Z., Wu, S., Xie, W., Chen, M., & Prisacariu, V. A. (2021). NeRF--: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064.
3. Lin, C. H., Ma, W. C., Torralba, A., & Lucey, S. (2021). Barf: Bundle-adjusting neural radiance fields. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 5741-5751).
4. Mildenhall, B., Srinivasan, P. P., Ortiz-Cayon, R., Kalantari, N. K., Ramamoorthi, R., Ng, R., & Kar, A. (2019). Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, *38*(4), 1-14.
5. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., & Brox, T. (2017). Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2462-2470).