CS172 Computer Vision I:
Homework 3 Nerf

First Author
Student ID 2021533088
juya@shanghaitech.edu.cn

Abstract

In HW3, I finished the following tasks: 1. Implementation of NeRF accelerate method(TensoRF and NGP) 2. Record and process three custom dataset and train on them.

1. Introduction

Since it is acceptable to directly use the open source codes, I use torch-ngp as my main codebase. For dataset processing, I also use COLMAP-3.8-windows-cuda, LLFF as tools to convert a video into LLFF dataset.

2. Part1 Implementation of NeRF accelerate method

2.1. basic settings

As mentioned above, I use torch-ngp as my main codebase. The environment is as follows: CUDA 11.3, python 3.6.6, torch 1.10.0, torchvision 0.11.0, rest of the packages are installed according to the instructions in torch-ngp. I ran my codes on a 4070 laptop GPU.

For TensoRF[1] and NGP[2] implementation, I choose the TanksAndTemple/Caterpillar dataset. The train/test split is specified in "transforms_test.json", "transforms_train.json" and "transforms_val.json"
There are 368 rgb images in the dataset, 322 of them are use as train data, 46 of them are used as test data, the train-test ratio is 7:1.

2.2. TensoRF

I train it for about 7 hours, the training result is PSNR = 24.974518
Here's some qualitative result comparison between rendered test view and GT test views:


GT view 1


rendered view 1

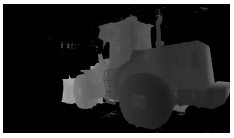

GT view 2

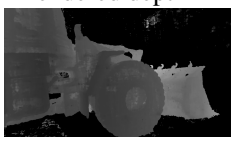

rendered view 2


GT view 3


rendered view 3


rendered depth 1


rendered depth 2


rendered depth 3

TensoRF

2.3. NGP

I train it for about 5 hours, The training result is PSNR = 24.850865, LPIPS (alex) = 0.151360
Here's some qualitative result comparison between rendered test view and GT test views:


GT view 1

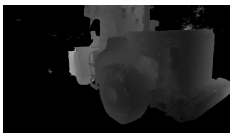

rendered view 1

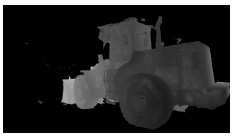

GT view 2


rendered view 2


GT view 3


rendered view 3


rendered depth 1


rendered depth 2


rendered depth 3

NGP

3. My custom dataset

I constructed three dataset, the first one is a camera, the second one is a mouthwash, the third one is my desk. I constructed the datasets by filming with my phone first, then process them into LLFF dataset form. Here's a preview of my datasets: (Figure 1)

4. Camera

There are 349 rgb images in the scene, 305 of them are use as train data, 44 of them are used as test data, the train-test ratio is 6.93:1. I train this scene for 116484 steps and reached a PSNR = 28.641409, LPIPS (alex) = 0.038063 Here's some qualitative result comparison between rendered test view and GT test views:

5. mouthwash

There are 323 rgb images in the scene, 282 of them are use as train data, 41 of them are used as test data, the train-test ratio is 6.88:1. I train this scene for 596698 steps and reached a PSNR = 27.950325, LPIPS (alex) = 0.073165


(a) camera


(b) desk

(c) mouthwash

Figure 1: Custom Datasets


GT view 1


rendered view 1

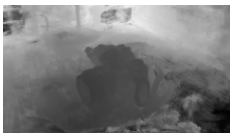

GT view 2


rendered view 2


GT view 3


rendered view 3


rendered depth 1


rendered depth 2


rendered depth 3

camera

Here's some qualitative result comparison between rendered test view and GT test views:


GT view 1


rendered view 1


GT view 2


rendered view 2


GT view 3


rendered view 3


rendered depth 1


rendered depth 2


rendered depth 3

mouthwash

6. desk

There are 371 rgb images in the scene, 324 of them are use as train data, 47 of them are used as test data, the train-test ratio is 6.89:1. I train this scene for 113659 steps and reached a PSNR = 28.539164, LPIPS (alex) = 0.069691
Here's some qualitative result comparison between rendered test view and GT test views:

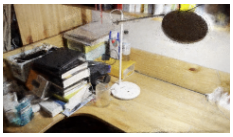

GT view 1

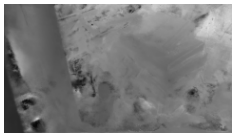

rendered view 1

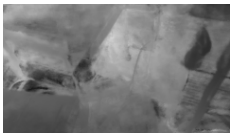

GT view 2

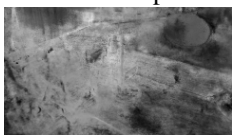

rendered view 2


GT view 3


rendered view 3


rendered depth 1


rendered depth 2


rendered depth 3

desk

References

[1] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields, 2022.

[2] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):1–15, July 2022.