Homework #4

Deep Learning for Computer Vision

NTU, Fall 2024

113/11/19

113/12/3 (Tue.) 11:59 PM (GMT+8) due

TAs: Bin-Shih Wu, Yu-Hsiang Huang

Update

- 11/27: if your GPU has higher CUDA version, you can install the corresponding Pytorch version (can use version other than 1.12.1)

Outline

- Problems & Grading
- Dataset
- Submission & Rules
- Supplementary

Problems – Overview

- **Problem :** 3D Novel View Synthesis

Please refer to “Dataset” section for more details about datasets.

Problem: 3D Novel View Synthesis

In this problem, you will have to train your own **3D gaussians** as explicit scene representation for the given scene.
To be more specific, given a set of training images (with camera pose) of this scene, learn the parameters of all 3D gaussians to fit this scene. After that, given a test set camera pose, by splatting the gaussians, you should be able to synthesize these unseen novel views of this scene.

- Grading:
  - Report (60%)
  - Performance (40%)

Reference:

1. [A Survey on 3D Gaussian Splatting]
2. [3D Gaussian Splatting for Real-Time Radiance Field Rendering]

Problem: 3D Novel View Synthesis

1. (15%) Please explain:

   a. Try to explain 3D Gaussian Splatting in your own words

   b. Compare 3D Gaussian Splatting with NeRF (pros & cons)

   c. Which part of 3D Gaussian Splatting is the most important you think? Why?

Please read the references below or other materials to realize their ideas.

Reference:

1. 3D Gaussian Splatting for Real-Time Radiance Field Rendering
2. A Survey on 3D Gaussian Splatting
3. NeRF: Representing Scene as Neural Radiance Fields for View Synthesis
4. 3D 重建技術 (三): 3D Gaussian Splatting, 讓 3D 重建從學術走向實際應用的革新技術！用彩色橢圓球球表示場景！

Problem: 3D Novel View Synthesis

2. (15%) Describe the implementation details of ***your 3D Gaussian Spaltting*** for the given dataset. You need to explain your ideas completely.

- **You have to train 3D gaussians to represent a scene on the given dataset without loading pretrained weights.**
- **We provide the following Github links for your reference.**
    - **3D Gaussian Splatting** - **LINK** (recommended)

Problem: 3D Novel View Synthesis

3. (15%) Given novel view camera pose, your 3D gaussians should be able to render novel view images. Please evaluate your generated images and ground truth images with the following three metrics (mentioned in the 3DGS paper). Try to use at least **three** different hyperparameter settings and discuss/analyze the results.
   - **Please report the PSNR/SSIM/LPIPS on the public testing set.**
   - **Also report the number of 3D gaussians.**
   - **You also need to explain the meaning of these metrics.**
   - **Different settings such as learning rate and densification interval, etc.**

Setting	**PSNR**	**SSIM**	**LPIPS (vgg)**	**Number of 3D gaussians**
Setting 1 (You need to write your setting)	*TODO*	*TODO*	*TODO*	*TODO*
Setting 2 (You need to write your setting) …	*TODO*	*TODO*	*TODO*	*TODO*

Problem: 3D Novel View Synthesis

4. (15%) Instead of initializing from SFM points [dataset/sparse/points3D.ply], please try to train your 3D gaussians with random initializing points.
    - Describe how you initialize 3D gaussians
    - Compare the performance with that in previous question.

Problem: 3D Novel View Synthesis

Performance (40%)

- PSNR and SSIM scores should **both** be above the baseline scores to get points
  - Public baseline (on the public testing set):
    - Baseline (20%):
      - PSNR: **35**
      - SSIM: **0.97**
  - Private baseline (on the private testing set):
    - Baseline (20%):
      - PSNR: **TBD**
      - SSIM: **TBD**

Outline

- Problems & Grading
- Dataset
- Submission & Rules
- Supplementary

Problem Dataset: indoor scene

- The dataset consists of **109** RGB images of size 861 X 478 of a scene.

- In the dataset, you will get

```
dataset/
    train/
        images/                 # training images (59 images)
            {id}.png
        sparse/0/               # training set (59 camera poses)
            cameras.txt    )
            images.txt     )    # including camera poses & SFM points.
            points3D.ply   )
    public_test/
        images/                 # public testing images (50 images)
        sparse/0/               # public testing set (50 camera poses)
    private_test/
        sparse/0/               # private testing set (170 camera poses)
```

- You **CANNOT** use testing data for training purposes.

Problem Dataset: indoor scene

- cameras.txt:
  - camera intrinsic parameters: camera position, focal length, principle point
- images.txt:
  - camera extrinsic parameters: quaternion, translation
- points3D.ply:
  - SFM points (keypoints) used for initialization
  - **only in train folder**

Reference: [Output Format — COLMAP 3.11.0.dev0 documentation]()

Download Problem Dataset

- Download the dataset

  https://drive.google.com/file/d/1B2rY6cfCD-UbxBGM24sgxk5NWDgiXg8_/view?usp=drive_link

Outline

- Problems & Grading
- Dataset
- Submission & Rules
- Supplementary

Submission

- Click the following link and sign in to your GitHub account to get your submission repository
        - https://classroom.github.com/a/qybCIqy6
    - You should connect your Github account to the classroom with your **student ID**
    - If you cannot find your student ID in the list, please contact us (ntudlcv@gmail.com)
- By default, we will only grade your last submission before the deadline (**NOT** your last submission). Please e-mail the TAs if you'd like to submit another version of your repository, and let us know which commit to grade.
- We will clone the **main** branch of your repository.

Submission

- Your GitHub repository should include the following files
  - hw4_<studentID>.pdf (report)
  - hw4.sh (for Problem)
  - Python files (e.g., training code & inference code & visualization code)
  - Model files (can be loaded by your python file)
- **Don’t push the dataset to your repo.**
- If any of the file format is wrong, you will get zero point.

Bash Script - Problem

- Please provide a script to the target split of folder which contain camera pose, and save the generated images into the specified directory.
- TA will run your code as shown below target
  - bash hw4.sh $1 $2
    - $1: path to the **folder** of split (e.g., */dataset/private_test)

      It also contains the folder of **sparse/0/**. The camera poses are in **sparse/0/cameras.txt** and **sparse/0/images.txt**. You should predict novel views base on the private test split.
    - $2: path of the **folder** to put output **images** (e.g., xxxxxxxxx.png, please follows sparse/0/images.txt to name your output images.)

      The filename should be {id}.png (e.g. xxxxxxxxx.png). The image size should be the same as training set.
- Please follow the naming rules strickly
- Note that you should **NOT** hard code any path in your file or script.
- Your testing code have to be finished in **20 mins.**
- We provide the grade.py for grading, see the supplementary page

Bash Script (cont'd)

- You must not use commands such as **rm**, **sudo**, **CUDA_VISIBLE_DEVICES**, **cp**, **mv**, **mkdir**, **cd**, **pip** or other commands to change the Linux environment.
- In your submitted script, please use the command **python3** to execute your testing python files.
  - For example: python3 test.py $1 $2
- We will execute you code on **Linux** system, so try to make sure you code can be executed on Linux system before submitting your homework.

Rules – Submission

- If your model checkpoints are larger than GitHub’s maximum capacity (50 MB), you could download and preprocess (e.g. unzip, tar zxf, etc.) them in hw4_download.sh.
    - TAs will run `bash hw4_download.sh` prior to any inference if the download script exists, i.e. it is **NOT** necessary to create a blank `hw4_download.sh` file.
- Do **NOT** delete your model checkpoints before the TAs release your score and before you have ensured that your score is correct.

Rules – Submission

- Please use **wget** to download the model checkpoints from cloud drive (e.g. Dropbox) or your working station.
  - You should use **-O argument** to specify the filename of the downloaded checkpoint.
- Please refer to this Dropbox Guide for a detailed tutorial.
- Google Drive is a widely used cloud drive, so it is allowed to use **gdown** to download your checkpoints from your drive.
  - It is also recommended to use -O argument to specify the filename.
  - Remember to set the permission visible to public, otherwise TAs are unable to grade your submission, resulting in zero point.
  - If you have set the permission correspondingly but failed to download with **gdown** because of Google's policy, TAs will manually download them, no worries!!

Rules – Environment

- Ubuntu 22.04.4 LTS
- NVIDIA RTX A4500(20 GB)
- GNU bash, version 5.1.16(1)-release
- Python 3.7

Rules – Environment

- Ensure your code can be executed successfully on **Linux** system before your submission.
- Use only **Python3** and **Bash** script conforming to our environment, do not use other languages (e.g., CUDA) and other shell (e.g., zsh, fish) during inference.
    - Use the command **"python3"** to execute your testing python files.
- You must **NOT** use commands such as **sudo, CUDA_VISIBLE_DEVICES** or other commands to interfere with the environment; **any malicious attempt against the environment will lead to zero point in this assignment**.
- You shall **NOT** hardcode any path in your python files or scripts, while the dataset given would be the absolute path to the directory.

Packages

- torch==1.12.1
- torchaudio==0.12.1
- torchvision==0.13.1
- plyfile==0.8.1
- opencv-python==4.10.0.84
- scipy==1.7.3
- imageio==2.31.2
- lpips==0.1.4
- numpy==1.21.6
- diff-gaussian-rasterization (check 3DGS)
- simple-knn (check 3DGS)
- fused-ssim (check 3DGS)
- Others if you need (e.g., tqdm, gdown, glob, yaml, etc.)
- Any dependencies of above packages, and other standard python packages
- If you use the official 3DGS repo and your GPU has higher CUDA version, you can install corresponding Pytorch version

- **E-mail or ask TA first if you want to import other packages.**

Packages and Reminders

- If you use other github repository other than the reference link, LINK, please give us your environment's requirements.txt and describe how we can reproduce your environment in your report.
- Python==3.7
- Do not use **imshow()** or **show()** in your code or your code will crash.
- Use **os.path.join** to deal with path as often as possible.
- If you train on GPU ids other than 0, remember to deal with the "**map location**" issue when you load model. (More details about this issue, please refer to https://github.com/pytorch/pytorch/issues/15541)

Deadline and Academic Honesty

- Deadline: **113/12/3 (Tue.) 11:59 PM (GMT+8)**
- Late policy : Up to 3 free late days in a semester. After that, late homework will be deducted 30% each day.
- Taking any unfair advantages over other class members (or letting anyone do so) is strictly prohibited. Violating university policy would result in F for this course.
- Students are encouraged to discuss the homework assignments, but you must complete the assignment by yourself. TA will compare the similarity of everyone’s homework. Any form of cheating or plagiarism will not be tolerated, which will also result in F for students with such misconduct.

Penalty

- If we cannot execute your code, TAs will give you a chance to make minor modifications to your code. After you modify your code,
    - If we can execute your code, you will still receive a 30% penalty in your model performance score.
    - If we still cannot execute your code, no point will be given.

Reminder

- Please start working on this homework as early as possible.
- The training may take hours on a GPU or days on CPUs.
- Please read and follow the HW rules carefully.
- If not sure, please ask your TAs!

How to Find Help

- Google!
- Use TA hours (please check course website for time/location).
- Post your question under HW4 discussion section on NTU COOL.
- Contact TAs by e-mail: ntudlcv@gmail.com.

DOs and DONTs for the TAs (& Instructor)

- Do NOT send private messages to TAs via Facebook.
- TAs are happy to help, but they are not your tutors 24/7.
- TAs will NOT debug for you, including addressing coding, environmental, library dependency problems.
- TAs do NOT answer questions not related to the course.
- If you cannot make the TA hours, please email the TAs to schedule an appointment instead of stopping by the lab directly.

Outline

- Problems & Grading
- Dataset
- Submission & Rules
- Supplementary

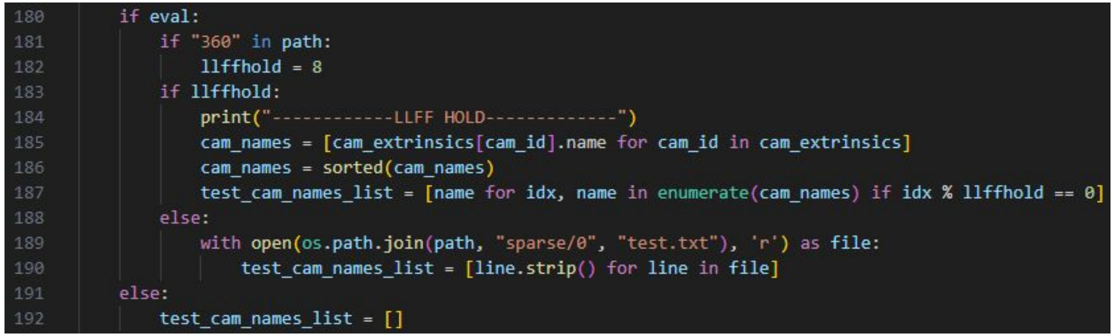
Supplementary - Installing 3DGS

- If you encounter `*OSError: CUDA_HOME environment variable is not set. Please set it to your CUDA install root.*`
  - try `*conda install -c conda-forge cudatoolkit-dev*`
- Make sure you have GCC.
- If you have other question when installing 3DGS, google first!
  - If you cannot solve it, e-mail or ask TA with error message.

Supplementary - 3DGS Official Repo

- If you use the 3DGS official repository as reference, here is a notice for implementation.

- gaussian-splatting/scene/dataset_readers.py (for dataset construction)
  - this part controls “validation set”.



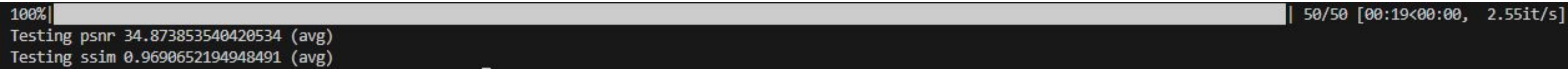
```
    if eval:
        if "360" in path:
            llffhold = 8
        if llffhold:
            print("------------LLFF HOLD------------")
            cam_names = [cam_extrinsics[cam_id].name for cam_id in cam_extrinsics]
            cam_names = sorted(cam_names)
            test_cam_names_list = [name for idx, name in enumerate(cam_names) if idx % llffhold == 0]
        else:
            with open(os.path.join(path, "sparse/0", "test.txt"), 'r') as file:
                test_cam_names_list = [line.strip() for line in file]
    else:
        test_cam_names_list = []
```

Supplementary - grade.py

- This file is on github and it is the one we will evaluate your score after running your hw4.sh.
  - python grade.py $1 $2
    - $1 is the path to the folder of generated image
    - $2 is the path to the folder of gt image
  - you can also use this file to evaluate on eval dataset

ground truth

generated



```
100%|████████████████████████████████████████████████████████████| 50/50 [00:19<00:00,  2.55it/s]
Testing psnr 34.873853540420534 (avg)
Testing ssim 0.9690652194948491 (avg)
```