CS5182 Course Project
(2024/2025 Semester A)

1 Objectives

The objectives of this project are for students to have some hands-on experience with recent research development in computer graphics-related areas, especially deep learning-based methods, such as deep modeling of 3D geometric data, neural rendering, and 2D/3D content generation.

2 Requirements

Students can choose **any** topics, which are **learning-based** and **related to computer graphics**, to explore, including the implementation of the algorithm, and performance analysis. To have a fast training and evaluation process, it is highly encouraged to proceed with the project on a machine equipped with GPUs. Both Pytorch and Tensorfolow platforms are allowed. Some topics are listed here for your reference.

List of 3D Geometric Data Processing:

- Point cloud shape classification/part segmentation/scene segmentation/...
- Point cloud upsampling/downsampling/registration/compression/denoising/completion/...
- Object detection in 3D point clouds
- 3D surface reconstruction
- ...

List of Neural Rendering topics:

- Faster Inference
- Faster Training
- Unconstrained Images
- Deformable
- Generalization
- Lighting
- 3D Reconstruction
- ...

3D Content Generation from Images/Texts/...

Other tasks are also encouraged. Some references are also provided at the end of this document. Students can refer to the open-source codes of those methods in GitHub.

This course project can be completed either individually or as a group. The maximum number of members allowed in each group is **3**. However, it is important to note that group-based projects are expected to have a higher workload and require more creativity. Additionally, in the report, the responsibilities of each group member should be clearly indicated to ensure accountability.

The program has two levels of requirements: basic and advanced, in order to accommodate students with different backgrounds and interests. The basic requirements are designed to provide all students with an opportunity to practice deep learning methods for computer graphics-related tasks, which will account for 70 percent of the final score. On the other hand, the advanced requirements are intended for students who wish to explore and propose a novel algorithm, contributing to 20 percent of the final score. The remaining 10 percent will be allocated to the presentation component.

2.1 Basic Requirements (70 percent)

1. **Introduction of the project**
   Students should provide an introduction to the selected task, which should include a concise statement of the input and output requirements. Additionally, it is important to mention the dataset being used, the source paper that presents your algorithm, the objective of your work, the motivation behind it, and the challenging issues that need to be addressed

2. **Deploy deep learning environment**
   Students should install deep learning platforms, such as TensorFlow or PyTorch, on their own computers or remote servers. One can refer to Anaconda for environmental management.

3. **Run Demo code**
   You will need to refer to research papers to select the algorithm for your specific task. Many of these papers provide demo code on GitHub, which may include training code or an evaluation code with a pre-trained model. It is important for students to successfully run this demo code and provide a detailed description of the purpose and function of each step.

4. **Train model**
   Students are expected to train their own model using the provided dataset. It is necessary to download the corresponding dataset for this purpose. Please note that training the model may take several hours when using a GPU. However, training on a CPU is not recommended as it will significantly increase the training time.

5. **Compare your experimental results with those provided by the corresponding paper**
   Students are responsible for evaluating the performance of both the trained model and, if provided, the pre-trained model. It is important to analyze the quantitative error and accuracy of the models. Additionally, students are expected to showcase the visualized results. For viewing 3D point clouds, MeshLab can be referred to as a suitable tool.

6. **Analyze the performance on other dataset**
   Students should pick another dataset to verify the performance and generality of this method.

7. **Analyze the drawbacks of the method**
   Students should carefully examine the failure cases of the algorithm and analyze the reason. Some ideas should be proposed to solve the problem.

8. **Implement and compare with other state-of-the-art methods**
   For the same task, students should pick another deep learning-based algorithm, implement it, and compare the results.

2.2 Advanced Requirements (20 percent)

Students are expected to expand upon this algorithm by applying it to new tasks or attempting to enhance it through your proposed method. This may involve modifying the network structure, the loss function, or the dataset. It is important for students to clearly explain the reasons behind these modifications and provide the final results following these changes.

2.3 Presentation (10 percent)

The project should be presented during the lecture/tutorial time. Each project has 7 minutes (5 m presentation + 2 m Q&A). Presentation Date: **02 Dec. 2024**

3 Submission Details

Each group needs to submit the following items via Canvas, and the submission link in Canvas will be open later. The submission deadline is **15 Dec. 2024**.

Program:

1. A source subdirectory containing all the source files, do not include the platform source code and training dataset.
2. Txt files which contain the results for training and evaluation periods.
3. Data that are evaluated in your report.

Report:

1. A cover that indicates your name(s) and student ID(s).
2. Describe the results listed in requirements in last section.

4 Marking

This course project contributes 25% of the final course mark.

5 References

- A comprehensive collection of papers on deep learning-based 3D point cloud processing: `https://github.com/NUAAXQ/awesome-point-cloud-analysis-2023`
- Neural Rendering Courses and Survey:
  `https://github.com/weihaox/awesome-neural-rendering/blob/master/docs/INTRODUCTION-A`
  `md`
- 3D Content generation. `https://github.com/topics/3d-generation`

- Libraries for Geometry Processing
  https://github.com/zishun/awesome-geometry-processing
- Advanced computer vision papers: Arxiv https://arxiv.org/list/cs.CV/recent
- Open source code: Github http://github.com/
- Environment management: Anaconda https://www.anaconda.com/
- TensorFlow API https://www.tensorflow.org/api_docs/python/tf/
- PyTorch API https://pytorch.org/docs/stable/index.html
- 3D point cloud viewer: MeshLab http://www.meshlab.net/