CSCI2240 Final Project Proposal

Arman Maesumi, Jianxin Gu, Nick Huang

April 2022

1 Choice of Paper

We choose to reproduce the result presented in Representing Scenes as Neural Radiance Fields for View Synthesis (NeRF). However, training a naive implementation of NeRF can take 1 to 2 days as pointed out by the paper. To resolve this issue, we think it would be helpful if we adapt the method presented in Instant Neural Graphics Primitives with a Multi-resolution Hash Encoding (INGP), which can reduce the training time significantly.

We are planning to implement our own version of Instant Neural Graphic Primitives and Neural Radiance Fields in PyTorch, and extend it to work with our project idea.

2 Idea for Extension

Conditional NeRF for rendered images.

The motivation for this project stems from the workflow of 3D artists that design and render complex scenes, which may take a considerable amount of time to render. Let's take a typical scenario as an example: an artist has already placed the objects in a scene, but is still tuning the environment configuration, such as the intensity of the scene lights. Every time an adjustment is made to any scene parameter, the entire scene needs to be re-rendered from scratch. Is there a way to quickly synthesize high fidelity previews of the scene under the new configuration?

To this end, we would like to utilize NeRF as the scene representation and add conditional parameters such as the intensity of a light source to the model. More specifically, we will modify the input of a traditional NeRF implicit function to be of the form $(x, y, z, \theta, \phi) \times \mathcal{P} \to (R, G, B, \sigma)$, where $\mathcal{P}$ is one, or many, scene parameters. Therefore, we can generate a quick preview of the scene under different configurations without the need to re-render the scene completely from scratch.

3 Data acquisition

In order to test the correctness of our own implementation of INGP and NeRF, we can reuse the dataset provided by the official code base of the NeRF paper.

For our project extension, we may have to generate new data from our own synthetic scenes. We are planning to utilize commercial renderers such as Blender or Mitsuba 2 because of their efficiency.

4 Timeline

Weekly subgoals:

- Milestone 1 items:
  - Week 1 (4/5 - 4/10) - Set up local environments, dig into our papers (INGP, NeRF), start writing the scaffolding of our codebase.
  - Week 2 (4/11 - 4/18) - Finish our NeRF implementation, test our implementation on original NeRF datasets, start working on INGP implementation.
- Milestone 2 items:

  - Week 3 (4/19 - 4/26) - Try to finish our INGP implementation (unforeseen issues may arise), start rendering custom synthetic data using commercial path tracers.
  - Week 4 (4/27 - 5/4) - Start implementing our novel ideas (condition NeRFs on scene parameters, such as light intensity).

- Remaining time: thoroughly test our novel idea on more data, write up our project report, create our project presentation.

5 Division of Labor

Currently our group has three members, and our project has three clear components that need to be developed (INGP, NeRF, synthetic data creation). Each team member can focus (mostly) on one task, while also helping with the other tasks. However, for the third task (generating synthetic data), this will mostly require using commercial software, which isn't really academically relevant to the course. This will be accounted for accordingly in the division of labor. We haven't yet assigned specific tasks to team members.