3D Reconstruction Framework for Development of Robots and IoT for Precision Agriculture

Brian Park, Nahyeon Kim, Shivam Kumar Panda, Andrew Choi, Ph.D, Khalid M. Jawed, Ph.D, Jungseock Joo, Ph.D, Sriram Narasimhan, Ph.D
Structures-Computer Interaction Laboratory, Mechanical and Aerospace Engineering Department, UC Los Angeles, CA 90095

Introduction

In agricultural robotics, there is no testing framework for robots and sensors used in precision agriculture. It is expensive to test our physical robot in the outside world, especially in its earlier stages of development.

Therefore, we require a virtual testing environment that includes a 3D rendering of crops, weeds, and soil to simulate our robot's behavior. We will develop an active learning model that is trained to discover the view angles of the object that require the most additional information.

Objectives

1. Develop the AgriSim Python library that controls the Unity Sampler module and Instant-NGP API.
2. Leverage this Python module to run baseline RL experiments to evaluate the performance of our produced NeRFs.

Methodology

To achieve the above objectives, we will utilize the peaceful-pie module in order to successfully create a connection between our Python library and Unity environment.

Peaceful-pie is a library that our Python module imports to invoke our Unity environment functions, including the ability to sample images, change the camera's position and orientation, and write positional and rotational data to text files.

Generating NeRFs with Python Sampler

Within AgriSim, we have implemented various functionality to load a 3D object from the ShapeNet dataset and place it within our Unity environment.

AgriSim then samples default viewpoints, which is in the shape of a circular ring around the center of the 3D object. The user can also input additional camera poses by specifying the camera's position and orientation.

Once AgriSim captures these images, it uses the positional and rotational information to compile a transforms.json file, which contains the intrinsic and extrinsic parameters of the Unity camera. Instant-NGP takes these images and transforms.json file to generate a NeRF.

Conducting Experiments

This quarter, we were able to conduct baseline experiments on our AgriSim module. Using the Instant-NGP API, we can take screenshots of our resulting NeRF and compare them to the images in our testing dataset. We calculate the PSNR and SSIM of each image.

Results: Python Sampler

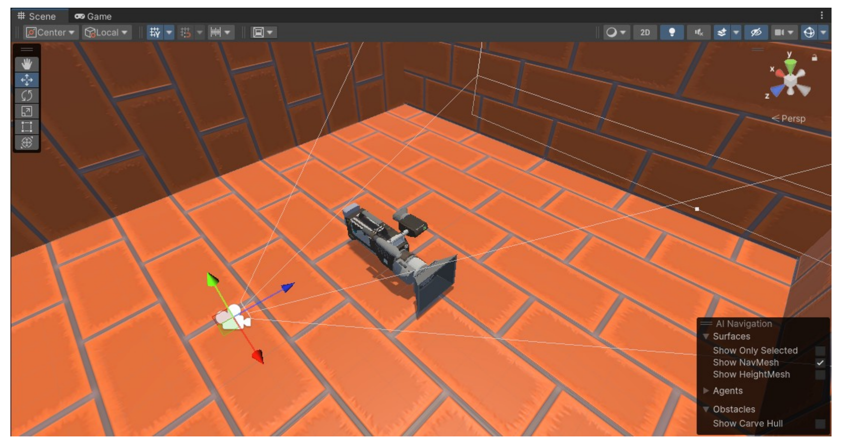

Figure 1: Unity Environment

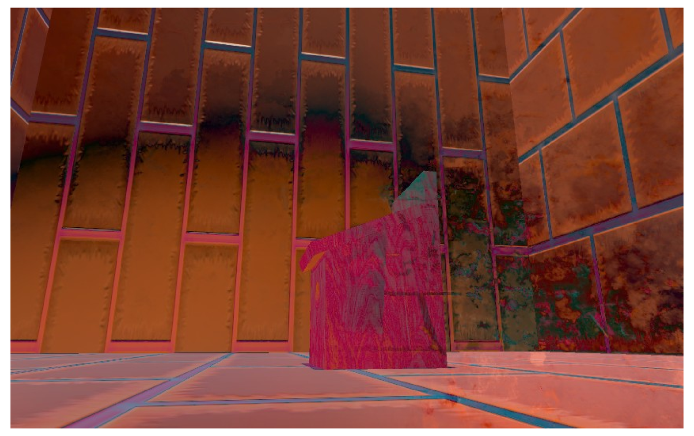

Figure 2: Combined difference of images

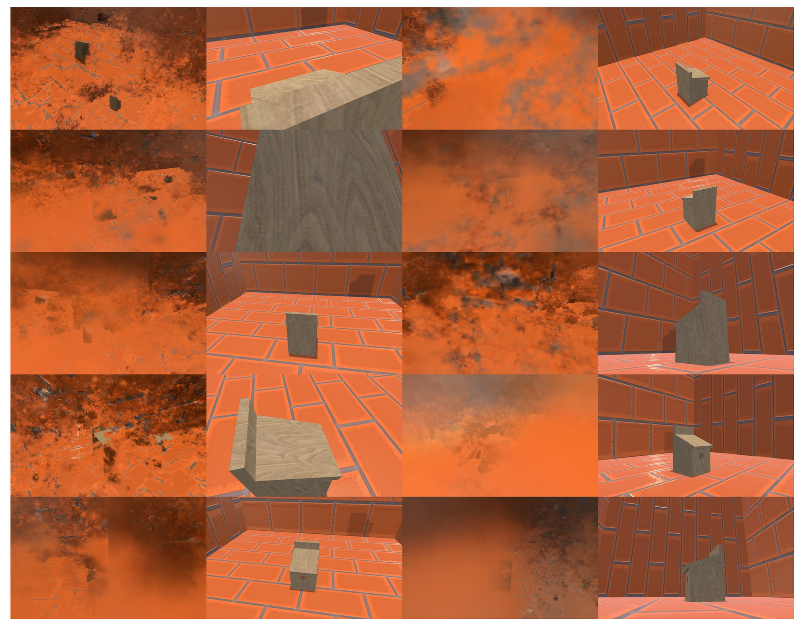

Figure 3: Collage of Predicted and ground truth images

Results: Baseline/RL Experiments

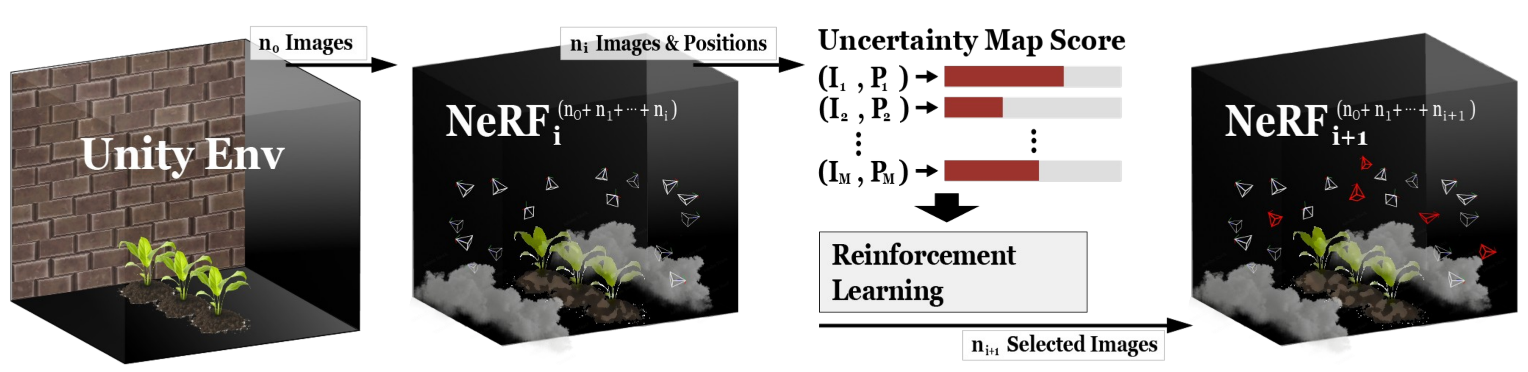

Figure 4: AgriSim-RL pipeline

Total Images	100	Iteration	35000
	Baseline_default image 0%	Baseline_default image 50%	Baseline_default image 100%
PSNR	13.976	15.161	14.15
SSIM	0.693	0.76	0.683
Total Images	50	Iteration	35000
	Baseline_default image 0%	Baseline_default image 50%	Baseline_default image 100%
PSNR	13.9822	15.802	14.144
SSIM	0.689	0.805	0.684

Figure 5: Baseline experiment results

Conclusion

As shown in Figures 1-3, AgriSim is able to successfully import and load an object within our Unity environmennt. Given an object ID, the user is able to select any 3D object within the ShapeNet dataset to use as the subject.

Our testing pipeline is also shown to be successful, as we are able to leverage Instant-NGP's API to analyze our resulting NeRF from specific viewpoints and evaluate them using the PSNR and SSIM metrics.

Figure 5 represents the results of our baseline experiments. Baseline experiments represent the quality of our NeRF without any reinforcement leaarning. Based on the table, it is not clear that the quality of our NeRF improves with larger quantity of sampled images. Rather, once the number of images reaches a certain threshold, the quality of the NeRF remains stagnant.

Additionally, it is shown that a combination of default and randomly sampled images gives us the highest quality NeRFs. In cases where our image dataset contains solely default images or solely randomly sampled images, Instant-NGP performs worse.

Future Works

1. In future quarters, we plan to conduct more thorough baseline and RL experiments, as we adjust various hyperparameters, including the number of images and the 3D sampled object.
2. Implement uncertainty map algorithms to allow the RL agent to predict the optimal image positions.

Literature Cited

Pan, Xuran., Lai, Zihang., Song, Shiji., Huang, Gao., "ActiveNERF: Learning Where To See With Uncertainty Estimation.", EECV 2022.
Mueller, Thomas., Evans, Alex., Scied Christopher., Keller, Alexander., "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding"., SISGRAPH 2022.

Acknowledgements

A special thanks to Professor Khalid Jawed, Andrew Choi, Shivam Kumar Panda, Nahyeon Kim, and the Structures-Computer Interaction Laboratory at the University of California, Los Angeles. This project was sponsored by the NSF REU program (Award No. 1925360).

NSF UCLA STRUCTURES-COMPUTER INTERACTION LABORATORY