FBIVan

Final Project

Bryn Sorli
Adam Rosa
Ryan McGarvey
Ghost of Elvis

Links:

GitHub: https://github.com/bryn-sorli/Intro_to_Robotics/tree/master/Project
Demo Video 1 (Blocks environment):
https://drive.google.com/open?id=1k7Ln-VWms3Cw57N3ZG9U8mrghaEMjOl2
Demo Video 2 (Landscape Mountains environment):
https://drive.google.com/open?id=1kIIZSLyVcPsidSVRXzSFLigDfJjKv9UY

Abstract:

For our project we wanted to have a multi-agent system. To do this we chose to use Microsoft's AirSim. Initially, we wanted to combine a ground vehicle with a drone to map an environment for the vehicle to travel through. However, due to limitations with AirSim, we switched to a project on drone swarming. For this we used four drones equipped with LIDARs to survey an environment and map out their surroundings. We then plotted this data and visualized the map.

Introduction and Background:

As mentioned earlier we used AirSim. Airsim is a high level open source API developed by Microsoft. The simulator runs in Unreal Engine 4 and lets the user simulate drones or cars equipped with many types of sensors and instruments. Control can be either direct or through the API. Because AirSim implements most of the functionality for moving the drones around, we spent more time planning the path for the drones and trying to find a way to keep track of them and their data. To do this AirSim lets you set up clients to each drone. These clients let you asynchronously control each drone in its own thread. As a result we had one script that was responsible for the drones flight plan and then another script that was responsible for each drones LIDAR.

Methods:

First you need the AirSim repository from GitHub, the repository is publicly available. Note that AirSim requires lots of memory and a rather good CPU and GPU, so make sure your system can meet the qualifications. Next you need the Unreal Engine. The documentation on AirSim is rather helpful at getting started. We started out by using the Block's environment that was shipped with AirSim. Most of the environment setup was handled by Ryan. Everyone got an environment setup but seeing as Ryan had the most powerful PC a lot of the testing was done on his system. Next AirSim works by essentially using the information coming from an Unreal environment. It does this by using a JSON folder that tells Unreal what to do in terms of simulated robots. These can then send and receive info from Python or C++ scripts. There are two scripts that need to be used. The first is the script for pathfinding and navigation. Ryan and

Bryn were responsible for the initial drone script that allowed a drone to take off and survey the area. Adam was responsible for improving upon this to include support for multiple drones and assign each of them an appropriate survey path. The second script is for receiving and plotting LIDAR data. Bryn was responsible for the initial data collection script that could receive input data from the drones. Adam was responsible for the plotting of the LIDAR data, by rotating the points given by the LIDAR sensor to the drone coordinates and then by translating those points to the world coordinates.

Results/Discussion:

The results were fantastic; here is a list of things that worked great. Running multiple drones in AirSim worked as long as we used a powerful enough PC. Planning the paths for the four drones to survey different areas of the landscape worked great. Reading and using the LIDAR from the drones was a clean process. Manipulating the data to a world coordinate system to plot our representation of the landscape worked great. Although all these things worked in the end, they did not come without difficulty. Since none of us had experience with Unreal, the setup of AirSim was very difficult. Ryan led us to a solution to this. Then we experienced a lot of difficulty flying multiple drones at the same time. This caused us to dig deep into the AirSim API, which was a great learning experience. We all dug into the API, but Adam led us to the best solution. Another problem we encountered was getting good LIDAR data. At first we got one sensor reading at the end of our flight, which is not very helpful. So, Ryan suggested and we agreed that the best approach would be to split up our drone controller code and our LIDAR code. This enabled us to receive LIDAR data as the drones were surveying the area. Bryn got us to a solution to receiving the data for a drone. Then, we had some difficulty combining the data from multiple drones into one world coordinate system, but Adam led us to the solution by drawing from the Sparki simulator labs that really prepared us on how to rotate and translate the data from each drone coordinate system to the world coordinate system.

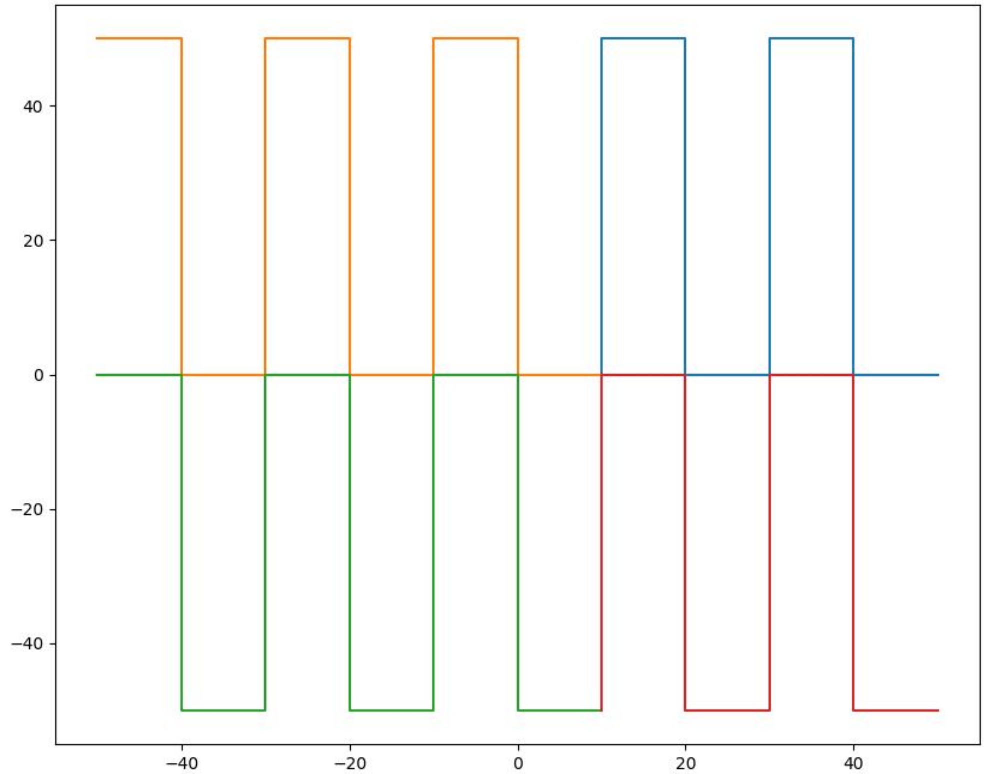

Figure 1. Path of four drones, each color represents a different drone

LIDAR Figures:

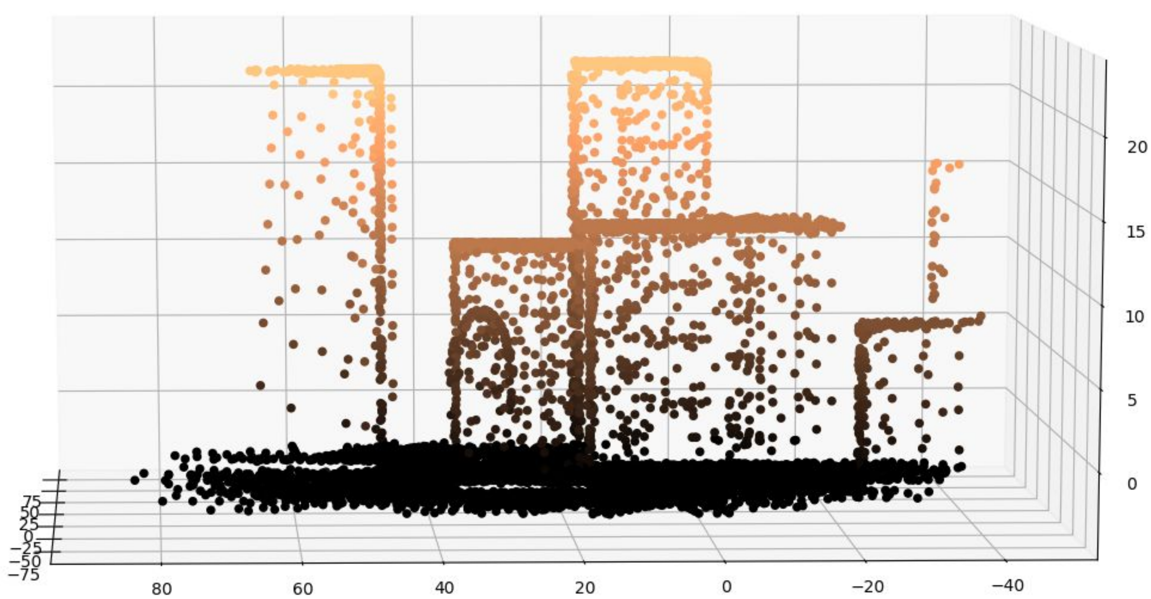

Figure 2. Side view of partial survey of Block environment

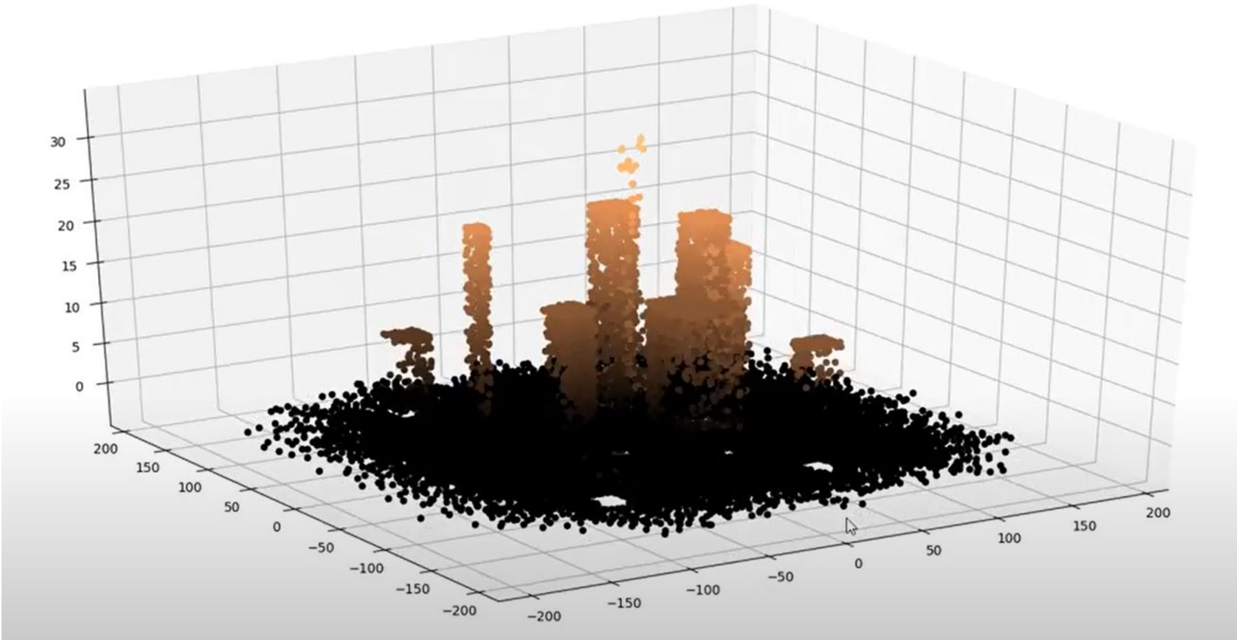

Figure 3. Full survey of Block environment

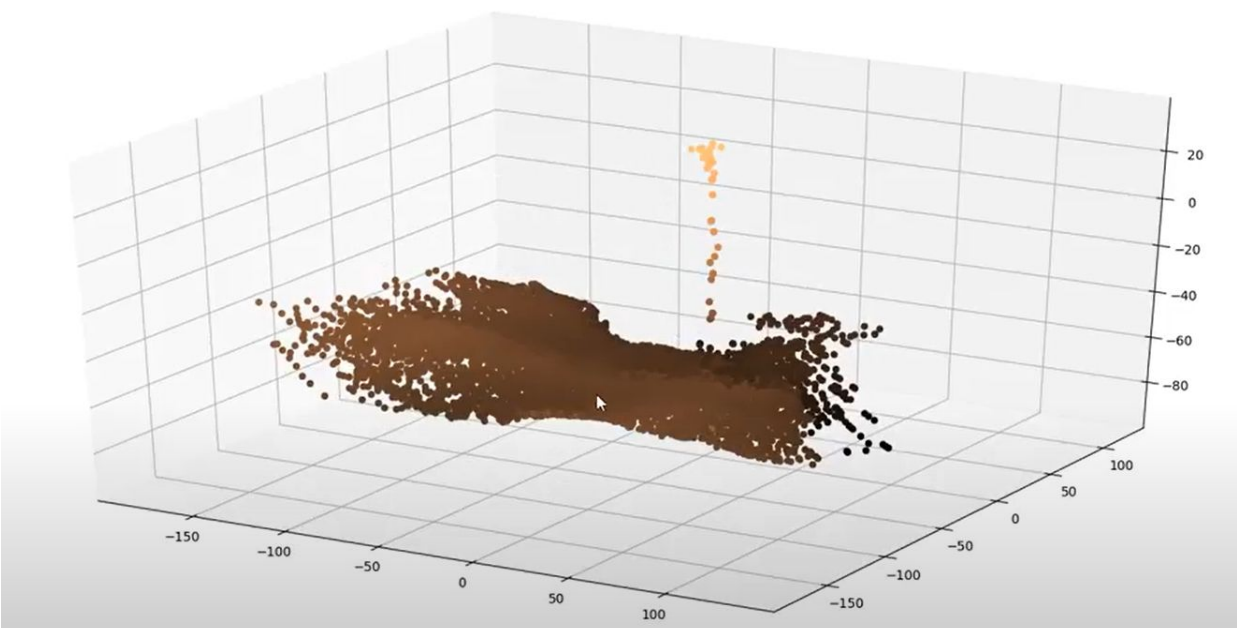

Figure 4. Partial survey of Landscape Mountains environment