ILR07 - Progress Review 8

Justin Morris
Teammates: Awadhut Thube, Alex Withers
Team G: The Pit Crew

October 2, 2020

1 Individual Progress

My focus over the past two weeks has been on the second of the two goals that Team G set for ourselves for Progress Review 8. This task was to implement a pit image capture control sequence, to be executed whenever the rover determined that it had reached a vantage point on its mission. In the past semester, we had created a panorama by hard-coding nine pan/tilt angle combinations for the camera mount to move to, and capturing an image at each position. These nine images were then post-processed to create a single panoramic image. Our goal for this semester was to create a parameterized version of this behavior that could be incorporated into the rover's state machine.

A major reason for capturing the pit images is to facilitate the creation of a 3D model of the pit interior on the rover's onboard computer. This 3D model would represent a more data-efficient product to be transmitted back to Earth than a series of individual images. We learned that the process for generating the model required panoramic images that were aligned along the horizontal axis. Therefore, I designed the image capture control sequence to stitch together the images captured at each of the camera tilt angles, rather than aggregating all the images into a single large panorama (Figure 1). These panoramas are also now generated within the code which moves the camera mount and captures the images, eliminating the need for a separate program for post-processing.

Figure 1: Panoramic images generated from images captured at 0 and +/- 30 degrees from the horizontal.

In addition to these changes, we knew that our code would have to account for the fact that the rover might approach the pit at an oblique angle and on uneven terrain. Therefore, the image capture code was designed take inputs for robot yaw, pitch, and roll. These values will be used to adjust the reference point around which the pan/tilt angles are calculated, so that the images captured by the rover camera are always centered on the pit interior. There is also a value stored in the code for the camera FOV, which can be used to determine how many levels of panorama will be required to image the entire pit from top to bottom. Lastly, I added code to gradually move the camera mount from position to position, because in our previous implementation the motion had been very jerky. I felt that slowing it down could potentially improve the fidelity of the captured images, since the motors would come to a complete stop before the image was captured.

2 Challenges

I foresee several challenges ahead of our team. We anticipate that the bulk of our work in October will be in integrating our subsystems into a single coherent system, a process that always involves unexpected difficulties. It will be essential for us to maintain awareness of the interaction points between various parts of our Pit Navigator project, as well as the ways in which it interacts with the greater body of the PitRanger mission. When we encounter errors, these interaction points should be where we begin our troubleshooting.

Our commitment to maintaining a hardware-dependent portion of our Fall Validation Demonstration will also prevent challenges. Finding a suitable location and time to test code on Blue will be difficult, and we will need to make the most of our limited resources in this regard. Code that works as expected in the simulated environment may not translate perfectly into the real world, and will need to be tweaked in order to function properly. Lastly, there will be logistical challenges in presenting a demonstration in Fall that is more than just a pre-recorded video of Blue in action, although as a last resort that may be the only format we can manage.

The generation and interpretation of a triangular mesh from a point cloud may also bring challenges. We have determined that there are existing algorithms for generating such meshes, but nobody on our team has first-hand experience with them. I am confident that we will be able to figure out how to create these triangular meshes and derive useful heuristic information from them, but for the time being they are an unknown quantity within our project plan. It will be necessary for us to develop familiarity with triangular meshes as soon as possible, in order to integrate their useful functionality into our project.

3 Teamwork

A smaller percentage of my work leading up to this progress review was teamwork, because I was interacting with Blue while Alex and Awadhut primarily developed in the simulation. However, Awadhut did provide me with the code he had written last semester for generating the panoramas, which I was able to integrate easily into my program. As we integrate our subsystems further, and work to transfer code designed in the simulation onto Blue, the amount of teamwork required will only increase.

Our team has returned to doing daily stand-up meetings every morning. Alex observed that holding these meetings last semester was beneficial in terms of keeping us focused and informed about our progress on the project. These meetings have helped us to stay on the same page, and contribute suggestions or advice when one member of the team is facing some hurdle in their work. As project manager, I have led these meetings, and I make sure that items on the project Trello board are discussed and updated when appropriate.

4 Plans

One of our goals for Progress Review 9 is to generate a triangle mesh and extract information about individual triangles from that mesh. We have learned that the MoonRanger team has created an implementation of a triangle mesh that can be constructed from a point cloud. I will contact the members of the team who have worked on this task, and learn how to use their code for our purposes. Awadhut and I will collaborate to turn a point cloud into one of these triangle maps, and augment the code from MoonRanger in any places where the necessary functionality is not present. We can then use this triangle map to implement more robust heuristics for slope and brink detection, and begin testing these heuristics in the simulator.

I will also continue to have primary responsibility for our rover test platform, Blue. I will work with my teammates to use Blue to test various subsystems in the physical world. When we are confident that we have tested our code sufficiently in the simulator, we will port those features to Blue and verify that they behave as expected in a real-world environment. Eventually, we will perform a integration test of most, if not all, of the functions of our system using Blue at a field site like Gascola. This integration test will represent a portion of our Fall Validation Demonstration deliverables.