ILR09 - Progress Review 10

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

October 29, 2020

1 Challenges

As the end of this project draws near, our window to kick the can down the road on the most challenging aspects of the project closes. In the past two weeks, we have been required by our schedule to demonstrate results on some major portions of the rover's functionality which had mostly existed in theory until now. Taking these aspects of the project from concept to reality has been exhausting and difficult, and has demanded all of our cleverness and focus.

One major task since the previous Progress Review has been to port functionality from the simulation to our rover surrogate. Our goal for this progress review was to demonstrate a short waypoint-based navigation sequence using dead reckoning. Our dead reckoning would use combined odometry from the wheel encoders and the IMU. This method was certain to create more error than navigation in the simulator, but over the distances we had in mind we thought we could make it work.

As it turned out, even developing functional dead reckoning with combined odometry in the first place was a challenge. Our first attempt only used the wheel odometry, and it demonstrated promising results. The rover traveled fairly straight, but calculated the angle of turns poorly, and could not recover from conditions where the wheels slipped or spun without touching the ground. We figured that once we incorporated IMU data, we would be in great shape. However, we first had to figure out the code that would combine the two odometry sources in a logical way, which took longer than expected. Then, when we attempted to test the rover, it mostly spun in circles. We determined that this was because the IMU information we were using was absolute, and therefore would produce different results depending on the rover's orientation at the beginning of the test (which were highly unlikely to be the results we wanted). It wasn't until Professor Dolan pointed out that the relative position information provided by the IMU would be more accurate and useful that we were able to effectively use the IMU to improve our waypoint navigation. At this point, however, we have managed to produce fairly accurate navigation results with the combined odometry.

Our other goals for Progress Review 10 both involved the triangular meshes that form the basis of our brinkmanship subsystem. Previously, we had developed a method to distinguish between safe and unsafe terrain by examining the unit normal vectors of each triangle in the generated meshes. However, we knew that this would not cover cases where the rover encountered a sheer brink, in which case there would be areas of the point cloud where there was no terrain to represent as a mesh. Furthermore, our method for stopping the rover near edges only actually relied on a narrow band of triangles from the mesh (those closest to the rover) to make judgments. We needed to come up with additional heuristics that would (a) account for sheer brinks, and (b) make more use of the mesh information. The latter of these two goals is still active, but thanks to Awadhut we've arrived at a version of the former.

Because the mesh is stored as an unordered set, it's difficult to discern which points should be the subject of our analysis when searching for brinks, but PyVista provides a convenient function for extracting the edges that make up the boundaries of a triangular mesh. We can then assess only the points which are connected by these edges. We can then look at the points in this set that are the furthest from the rover's position, and treat those as the far edge of the mesh. The distance of this far edge from the rover is then used to indicate the presence of a sheer brink. If the far edge of the mesh is closer than it would be when the rover is facing flat, level terrain, then we can assume that the rover is perceiving empty space beyond the edge of a brink instead of terrain.

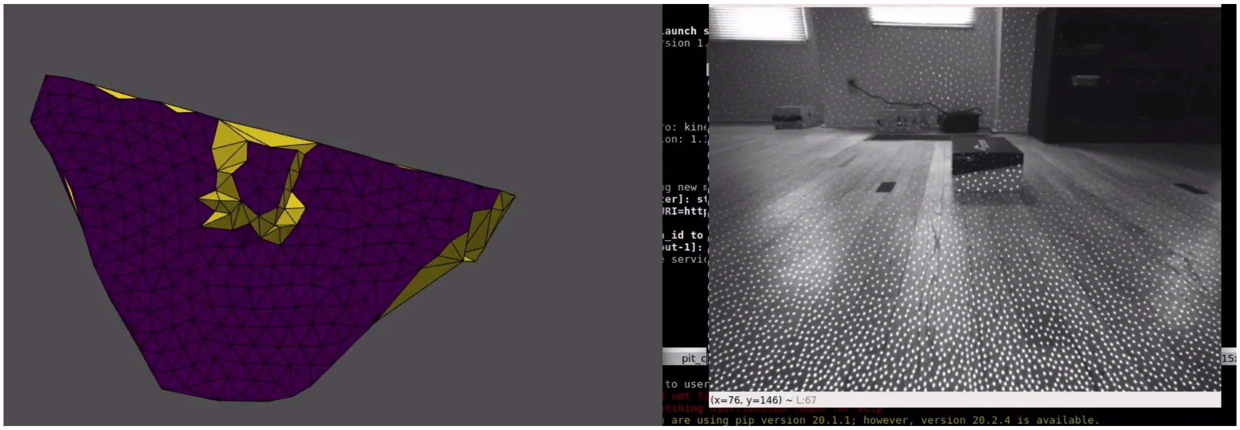

Figure 1: A mesh generated from stereo imagery captured by the rover surrogate's RealSense camera. The shoebox visible in the right-hand image can be seen in the mesh.

Our last goal was to generate triangular meshes from point clouds that originated from stereo imagery of real-life terrain. The biggest challenge to this task was very pedestrian in nature. For example, the USB cable that connected the RealSense camera to the rover computer was damaged when it was dragged along the ground during a field test, delaying any RealSense-related testing until a replacement cable could be found. We initially tried replacing the cable with a USB 2.0 cable, and spent a significant amount of time ruling out other possible causes of the errors that resulted. Once we got our hands on the proper USB 3.0 cable, however, the code we had written to generate meshes in the simulator worked just as well with the stereo-reconstructed point clouds. The results of this test can be seen in Figure 1, above.

2 Teamwork

I chose not to add an "Individual Progress" section in this report, because all the work I have performed since the previous Progress Review has been in a team with one or more of my partners. At this stage in our project, with much of our focus on integration of the various subsystems, there are fewer opportunities than before for individual work that does not involve multiple team members.

The rover has been stored at my apartment, and my yard has suitable terrain for testing, so Alex has been coming over almost every day to work on the waypoint navigation feature. Alex is the expert on the planning subsystem, so much of the code being tested originated from him. I've been assisting with partner debugging, troubleshooting, and providing an extra pair of hands when moving the rover around during and between tests. I've described most of our process and the issues we encountered in Section 1. Needless to say, we have been working together almost constantly.

Awadhut and I have also worked together quite a bit. The two mesh-related goals for this Progress Review involved the intersection of our areas of expertise, so it made sense for us to work on them together. Awadhut did most of the work on the sheer brink detection, as I was occupied with assisting Alex, but we had several conversations about possible heuristics and how to implement them. For the real-life mesh test, I was the one to conduct the testing because I had custody of the rover surrogate. Awadhut helped me adjust the code that we had previously written so that the point cloud-producing code and the mesh-generating code worked well together, and we worked together to troubleshoot the issues we encountered when trying to use a USB 2.0 cable to connect to the RealSense.

3 Plans

While I admit to having made this claim multiple times in the past month, it is more true than ever that our project has transitioned almost completely into the integration stage. The various functions are all present, and we've begun the process of linking them together into a single coherent system. In the simulator, we have made good progress in this regard. We have the ability to execute an entire mission plan, in which the rover travels to multiple locations around the edges of a modeled lunar pit, returning to a point distant from the pit at appropriate intervals. At each waypoint, the rover executes its brinkmanship routine, which creates meshes of the environment to guide the rover as it moves as close to the pit edge as possible. The final steps for this portion of the project are to integrate the sheer brink detection and use the mesh information to update the map that the rover uses for navigation and planning.

When it comes to the surrogate rover, we have somewhat further to go. However, our progress prior to this Progress Review puts us in a good position to complete our objectives before the next review. We have planned a field test at the Gascola site for this Friday (10/30). We will perform another test of our waypoint navigation routine, and also test brinkmanship on whatever slopes or brinks we can find at the site. Based on the results of this test, we will address any issues we encounter and put together a single test that involves navigation to multiple waypoints and repeated execution of the brinkmanship subroutine, to be executed ahead of Progress Review 11. This integrated test will also serve as a dry run for the real-life portion of our Fall Validation Demonstration.