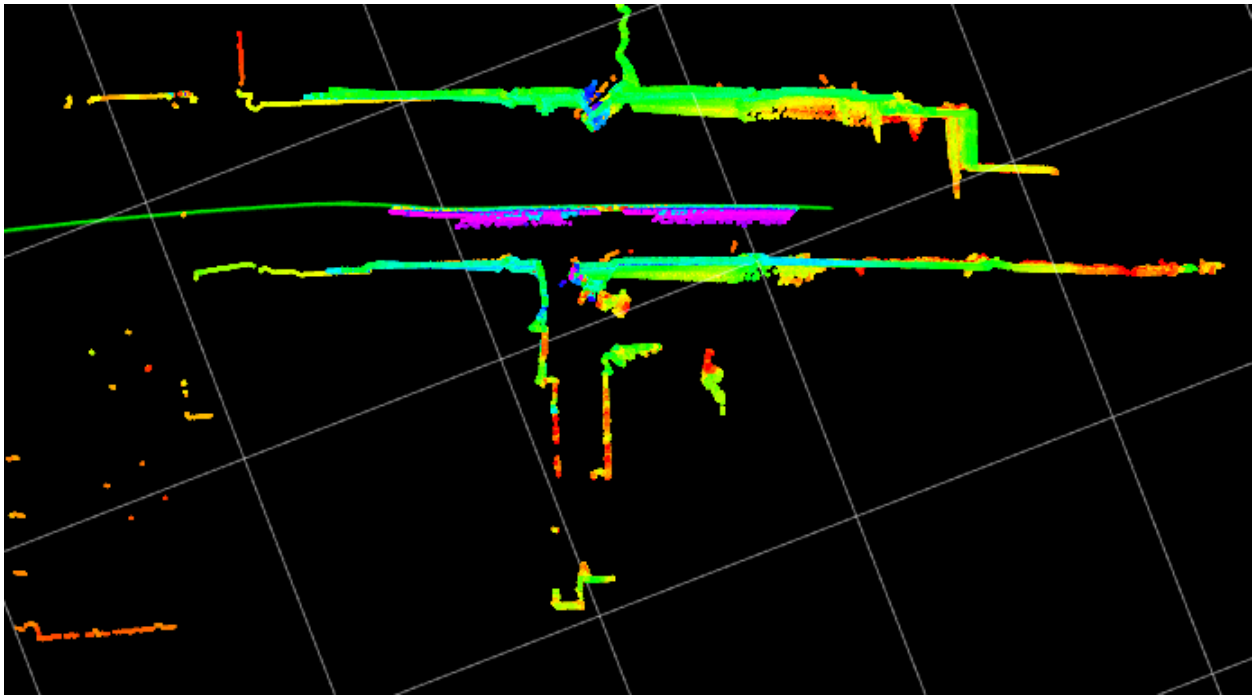
Vaibhav Raheja

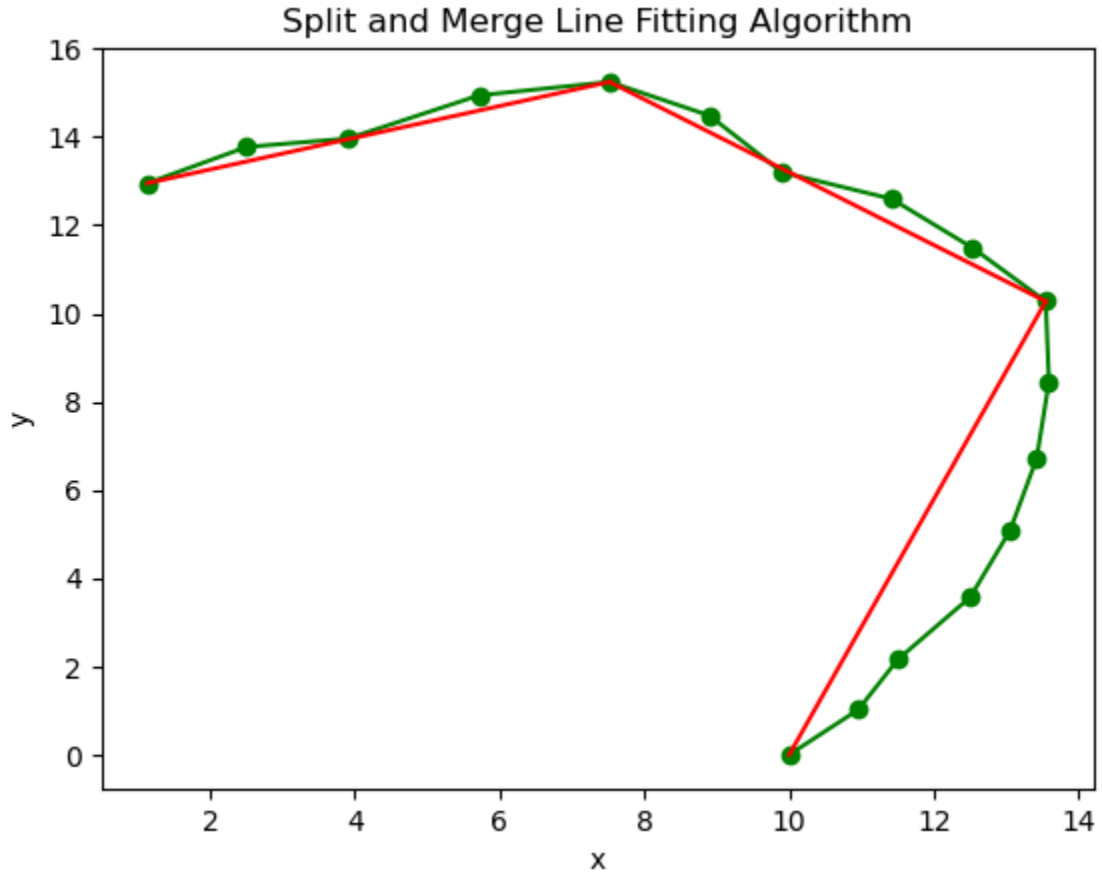
1. Play the rosbag file ‘lidar’ and visualize the robot trajectory and the LiDAR measurements on rviz2. Add screenshots to your report.



2. Write code in Python to find and fit lines and corners from LIDAR data, using one of the line fitting algorithms discussed in chapter 4.7.2 of Siegwart. Test your algorithm on the following data. Add the results to your report (you don’t need the ros template for this question).

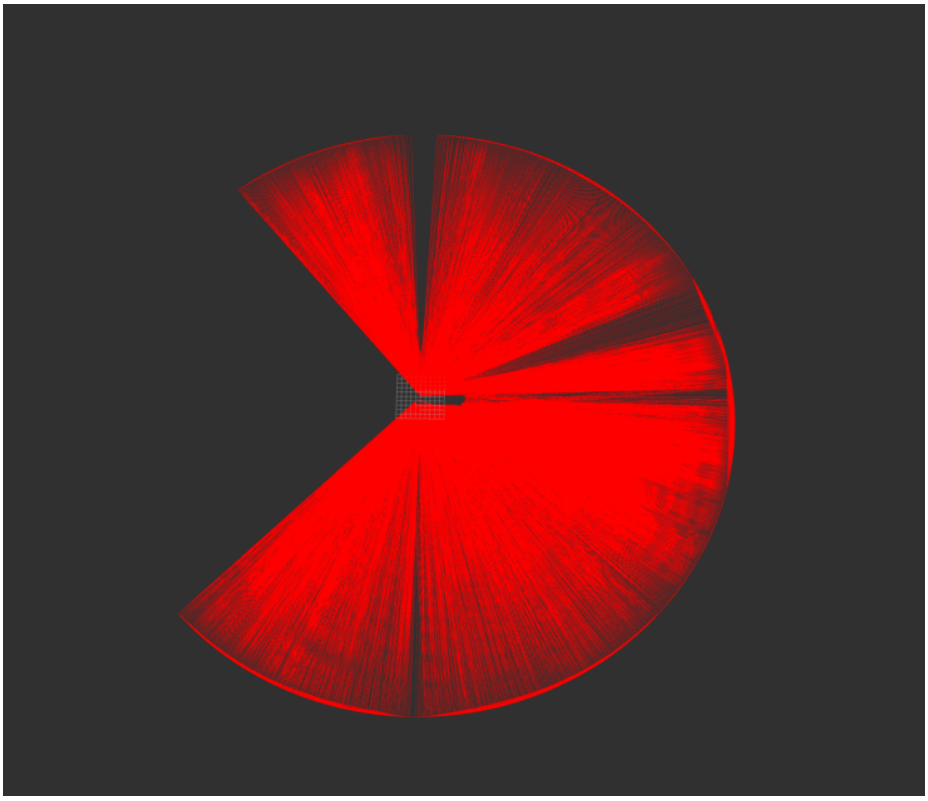
Vaibhav Raheja

Split and Merge:

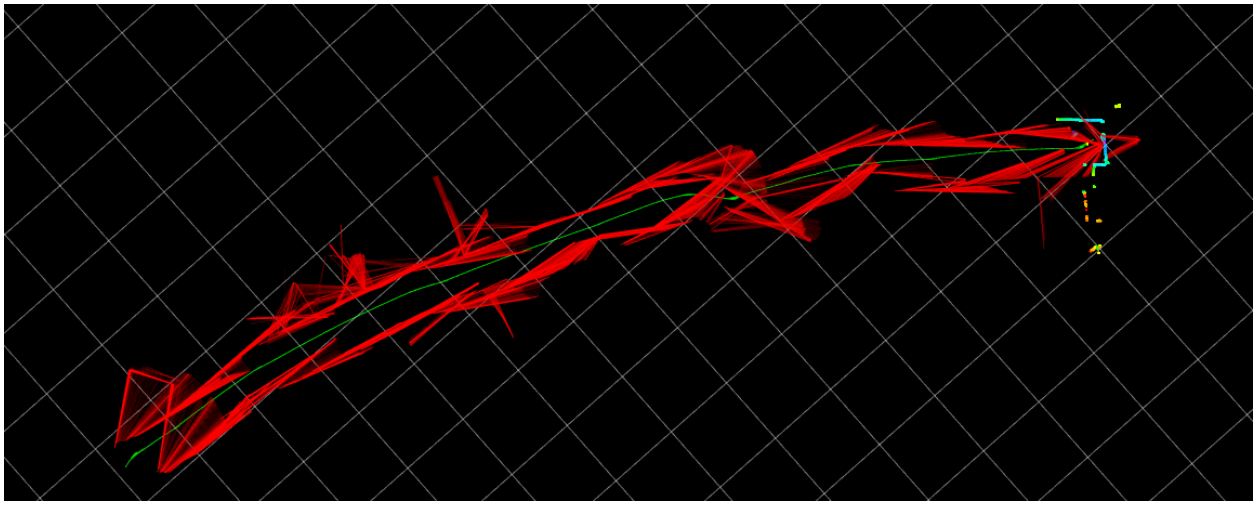


3. Use the function created in the previous question to find and fit lines and corners from the LiDAR data in the rosbag file. Visualize the lines on rviz2. Add screenshots to your report.

Vaibhav Raheja

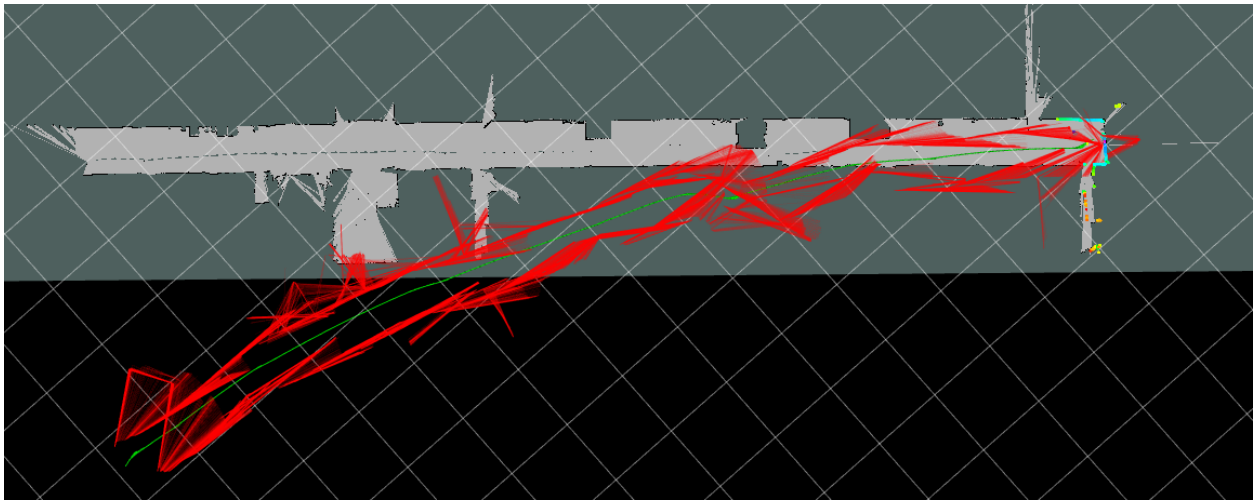


4. Mapping. Use the line fitting algorithm and the trajectory of the robot to create a complete geometric map of the corridor. Add screenshots to your report.



5. SLAM. Use Gmapping to create a map based on SLAM (see tutorial below). Compare this map with the one you obtained with your implementation. Explain the differences you find.

Vaibhav Raheja



The discrepancy is caused by the difference in timing between the odometry (odom) data and the simultaneous localization and mapping (SLAM) data, as well as the fact that the line fitting is not synchronized with the SLAM time steps and is moving with respect to the odometry. Therefore, there is a variance. Moreover, outliers are noticeable in the output, suggesting that there is room for enhancement. This could potentially be addressed by employing a different algorithm or by adjusting the parameters for range filtering to better exclude anomalies.