SLAM based on a Single Plane Laser Sensor

Maggie Du[1,2], Sean Andersson[2], Yancheng Zhu[2]

Monta Vista High School, 21840 McClellan Rd, Cupertino, CA 95014[1]; Boston University Department of Mechanical Engineering, 110 Cummington Mall, Boston, MA, 02215[2]

BOSTON UNIVERSITY

Introduction

Simultaneous Localization and Mapping (SLAM) is a powerful tool that allows us to **construct a map** of a given space while simultaneously **providing our estimated location** within the map. This technique is already being used in a variety of applications, including autonomous cars, robot vacuums, etc. The goal of this project is to **create a SLAM capable bot** with only a **single-plane laser sensor** that provides an array of distances from itself to the closest obstacle in each direction.

Methods

The hardware consists of a **raspberry pi 3** model B computer (fig. 1) connected to a time of flight **Scanse sweep laser range finder** with 360° horizontal FOV and 1 cm resolution (fig. 2).

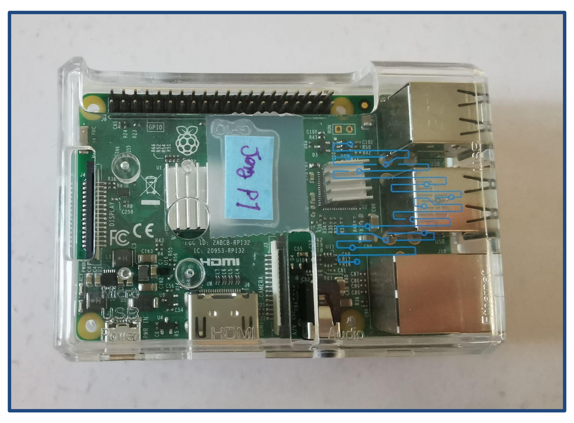

Figure 1: raspberry pi board

Figure 2: laser sensor

Software setup consisted of Ubuntu 20.04 as the running environment with **ROS (Robot Operating System)** running on top. ROS nodes are the actual code pieces that run the necessary tasks, and ROS topics connect these nodes together.

We used an off the shelf SLAM algorithm called **Hector Mapping** for this project. Hector SLAM uses only **laser scan data** (no odometry) to build a map and localize. However, it does **require transform data** that essentially manages the relevant coordinate frames (fig. 3).

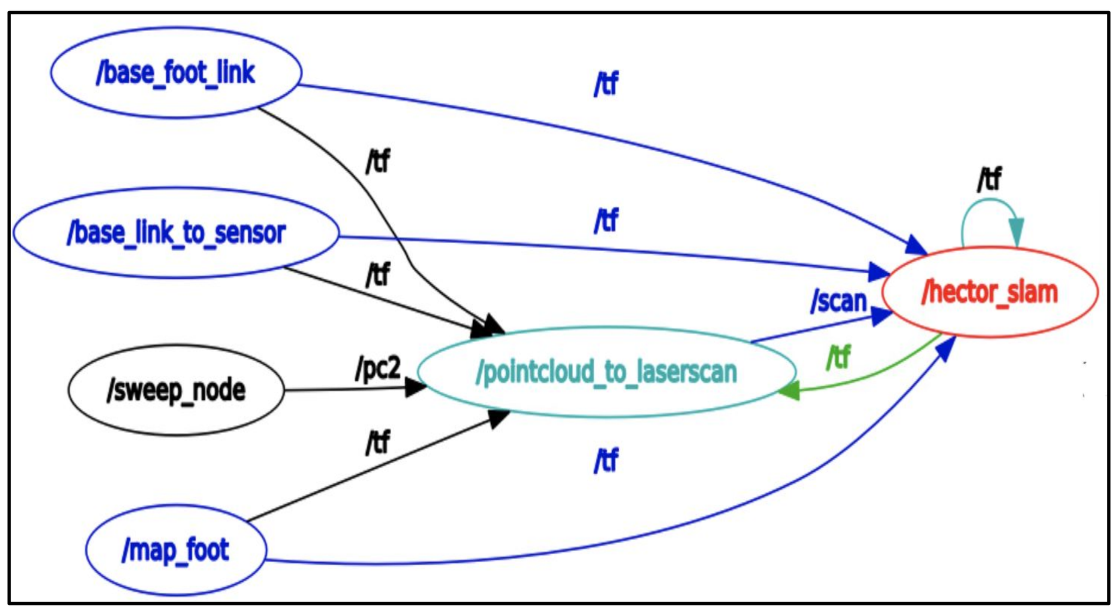

Figure 3: ROS node + topic connection graph

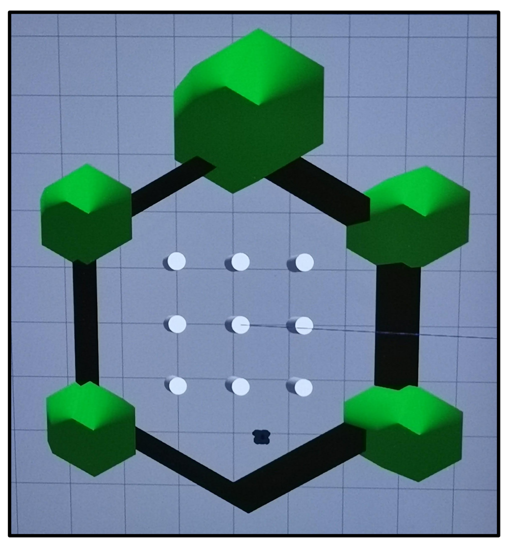

Figure 4: Turtlebot world

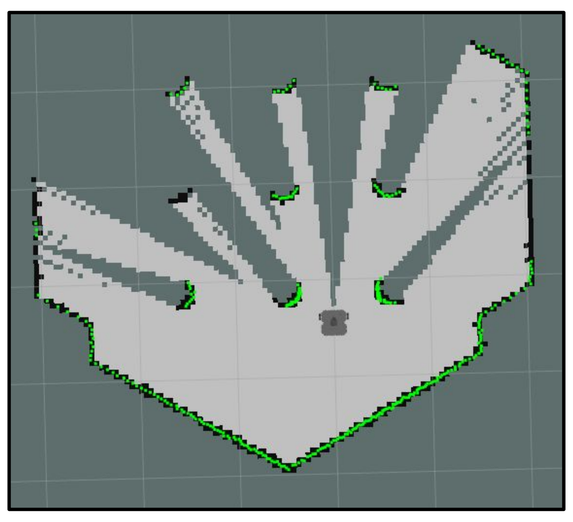

Figure 5: RViz map

Our initial research was on the **ROS turtlebot gazebo package** (fig. 4) to test Hector mapping on a virtual bot that published tf and scan data to rviz and hector mapping. Using the turtlebot teleop package, we could move the virtual bot around the environment and intake scan data. The **map was generated in real time on RViz**, a ROS graphing visualization tool (fig. 5).

For the actual sensor, we used a **physical room in our lab** to test the hector mapping algorithm (fig. 6). This required an **extra converting step** using the pointcloud_to_laserscan package to transform the range information for the mapping node to understand. After that, we published the tf and scan data from static transform publisher and the laser scan, respectively.

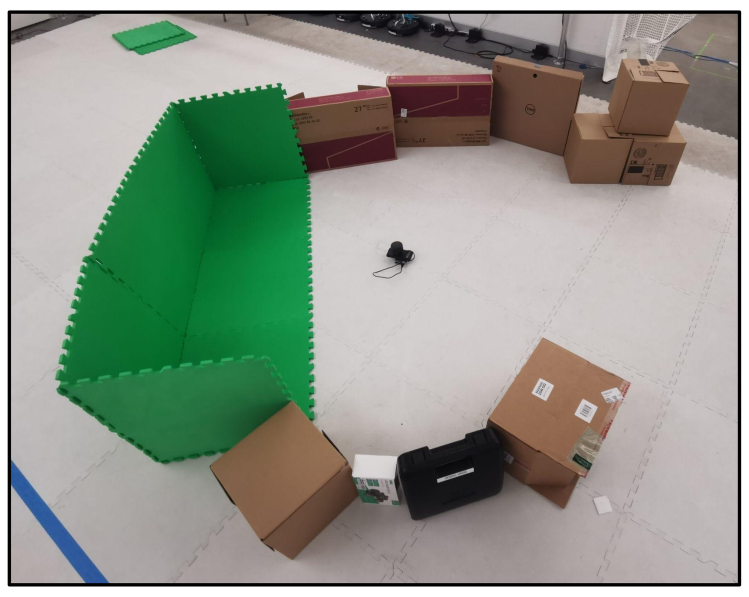

Figure 6: physical testing environment using boxes and tiles

Results

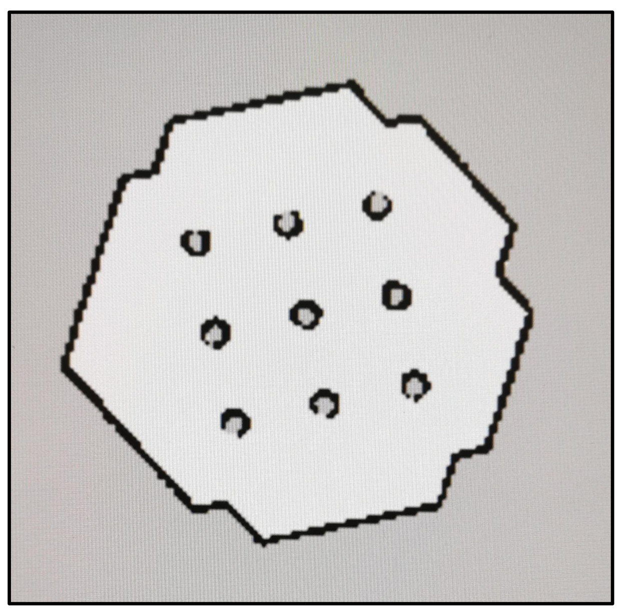

Figure 7: pgm map file generated from gazebo world by turtlebot slam launch file

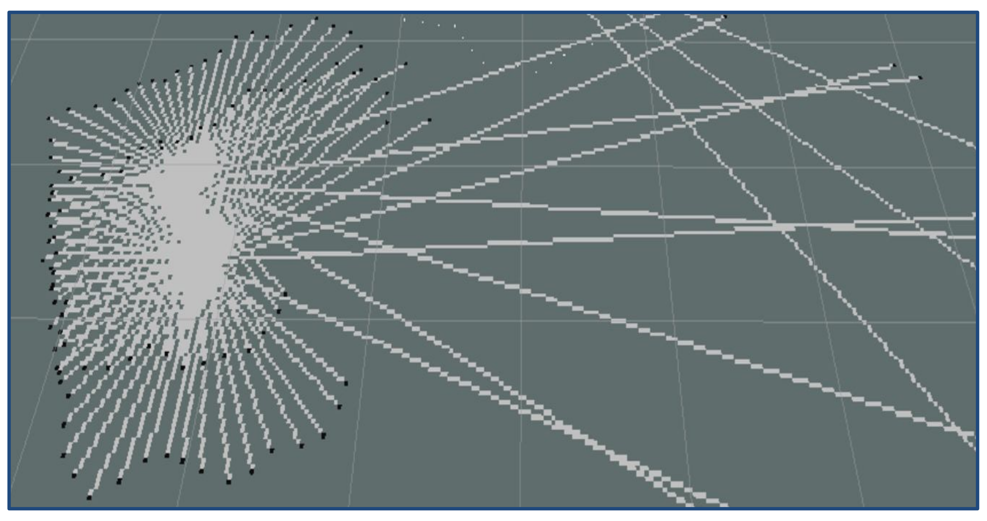

Figure 8: real time RViz map using laserscan data from range finder

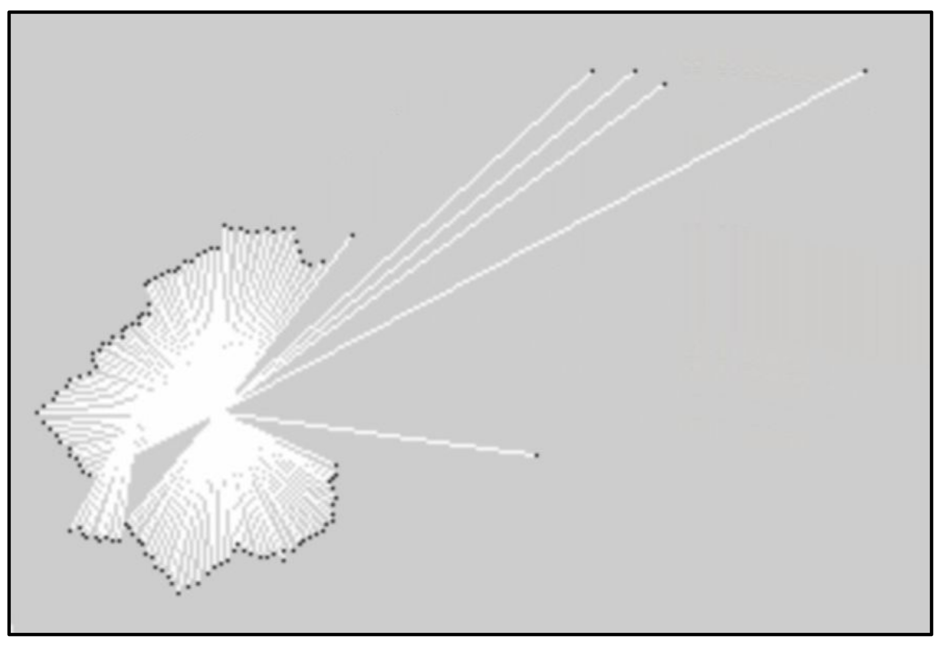

Figure 9: pgm map file generated from hector mapping in the physical world

We were able to successfully generate maps of both the virtual and physical environments using the hector mapping package in conjunction with transform information plus the scan data coming from either the turtlebot or the sweep sensor.

Discussion

This project raises the question of **how much data is really necessary** to create a decent map. This laser scanning sensor can only generate 2D maps because of its single plane limitations, but is also much more versatile than a single beam laser sensor. The objective is to achieve the same goal of generating a SLAM capable machine but **using as little resources as possible**, so further research could target this motive.

SLAM is still a growing field, and there are **numerous future possibilities** for low cost, low resource robots that could be used in conjunction with other fields. For instance, navigating newly discovered caves or burrows for geology or biology research using several SLAM capable robots would be safer than sending humans down to investigate first.

References

- Ponnu, G; George, J. Real-Time ROSberryPi SLAM Robot. Masters Dissertation, Cornell University, Ithaca, NY, 2016.
- ROS Wiki. http://wiki.ros.org/Documentation (accessed August 5, 2022)
- https://github.com/ros/geometry
- https://github.com/scanse/sweep-ros
- https://github.com/samialperen/oko_slam/blob/master/doc/hector_slam_tutorial.md
- https://github.com/turtlebot/turtlebot_simulator

Acknowledgements

I would like to thank my professor Sean Andersson and my mentor Yancheng Zhu for their constant guidance throughout this project, as well as the rest of the researchers at Andersson lab for their continued help and support during times of need.