Project: Map My World Robot

Jd Fiscus

1 Abstract

In this project for Map My World there were two main objectives to complete the goal both of these objectives were to use ROS, and Gazebo utilizing RTAB-Map[1] to produce a 2D map and a 3D map in separate worlds. The first world was to provide a kitchen and dining room to map. The second world was to be built by the student.

The built robot from the previous project was to be expanded on in this project. However it was enhanced with new sensors including a RGB-D camera which replaced the previous one. The robot is expected to be able to teleop around the world in order to collect data for generating the 2D & 3D maps.

2 Introduction

This project is utilizing the technology Simultaneous Localisation and Mapping (SLAM) partnered with a technique called RTAB-Map (Real-Time Appearance-Based Mapping). It bases this SLAM approach on a RGB-D Graph which uses loop closure detection.

A real time visual representation is created with rtabmapviz and the RTAB-Map ROS wrapper. All the information is stored in a SQLite database file which can be viewed in both 2D and 3D maps utilizing rtabmap-databaseViewer[2].

3 Background

The robot is placed in a world where there is no predefined map, so it needs to create a map in order to determine where it is in relation to the rest of the world. This SLAM is utilizing multiple algorithms including Graph-SLAM[3], RTAB-Map, Occupancy Grid Mapping[4], and Grid-based FastSLAM.

---

[1] "RTAB-Map | Real-Time Appearance-Based Mapping - GitHub Pages." http://introlab.github.io/rtabmap/. Accessed 2 Jul. 2018.

[2] "rtabmap_ros - ROS Wiki - ROS.org." 30 Apr. 2018, http://wiki.ros.org/rtabmap_ros. Accessed 4 Jul. 2018.

[3] "GraphSLAM - Wikipedia." https://en.wikipedia.org/wiki/GraphSLAM. Accessed 4 Jul. 2018.

[4] "Occupancy grid mapping - Wikipedia." https://en.wikipedia.org/wiki/Occupancy_grid_mapping. Accessed 4 Jul. 2018.

Graph-SLAM is a SLAM algorithm thats uses information matrices to create a graph-based approach in representing two observation data points from a landmark. This is a full SLAM solution because it covers the path and map.

RTAB-Map is a Graph-SLAM approach utilizing loop closure. This detection works by gathering images to be analyzed as working memory. Long-term memory can be used to reduce complexity. The data collected is then outputted as an SQLite database where it can be viewed as a 2D or 3D map.

Occupancy Grid Mapping is a 2D algorithm which is the basis for the 2D map being generated by RTAB-Map.

Grid-based FastSLAM is a combination of SLAM, MCL and Occupancy Grid Mapping. By leveraging MCL particle filter and assuming there are known landmarks.

There is a lot of complexity with scale to consider in robotics, especially when mapping. Consider the height or sensor radius available can affect the quality of mapping results. For instance if a robot is to short it may not be able to scan the height of the map, but will probably have an easier time navigating whereas a larger robot can scan the world easier but might have mobility problems. Also different sensors produce a different quantity and quality of data. The cost of storing and processing data can affect the robots decision making ability.

4 Scene and Robot Configuration

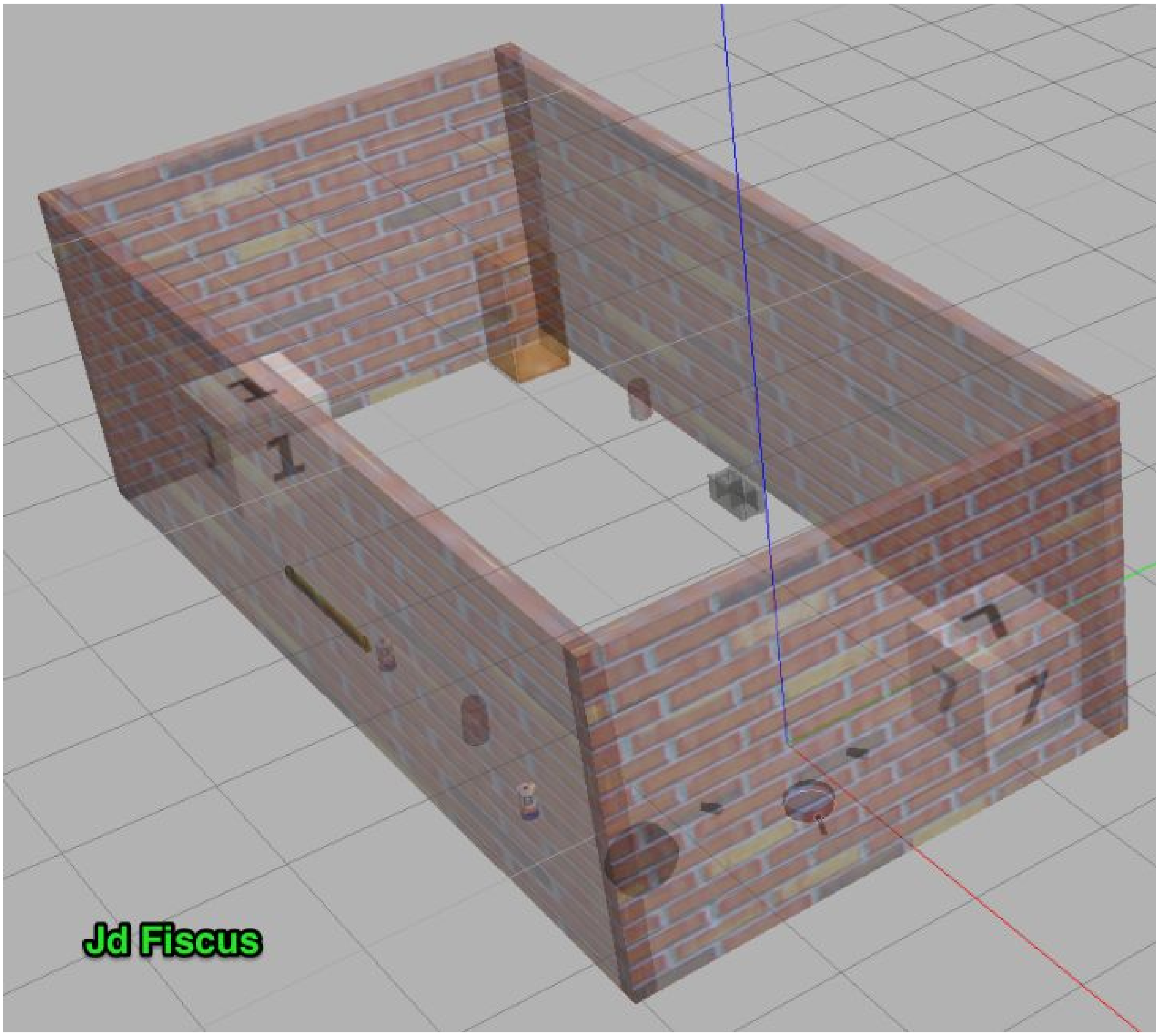
Scene

There are two scenes or worlds used in this project, first is the supplied world of a kitchen & dining room (kitchen_dining.world). The other is a custom built world, which in this case there were multiple different world created in an attempt to get results.

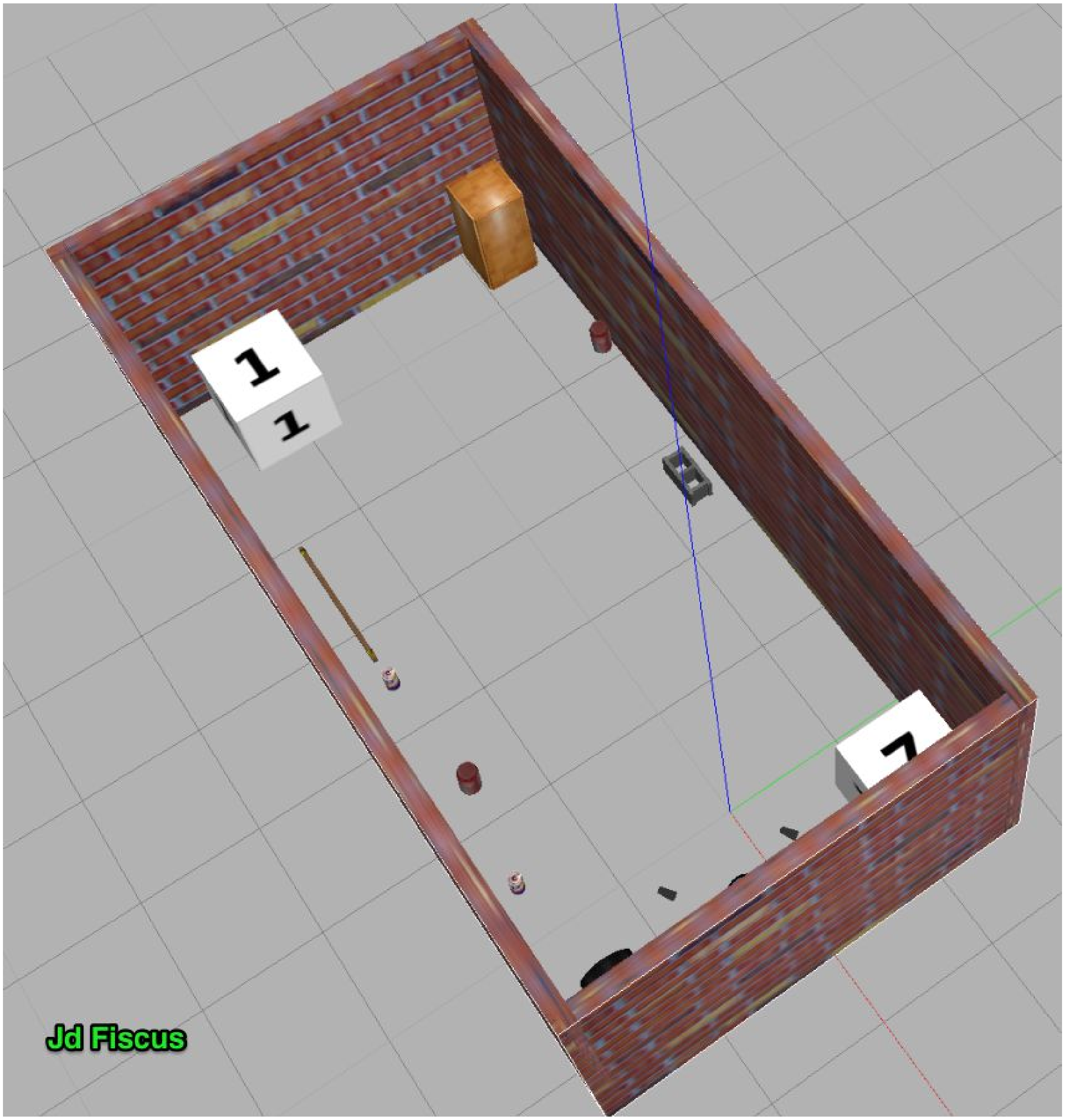
When creating the custom world larger objects where chosen to attempt providing enough detection points for a SLUR. This makes it easier for the robot to determine where it is placed without making false positives.

This project and structured based on the previous project. However it also includes the RTAB-Map database file.

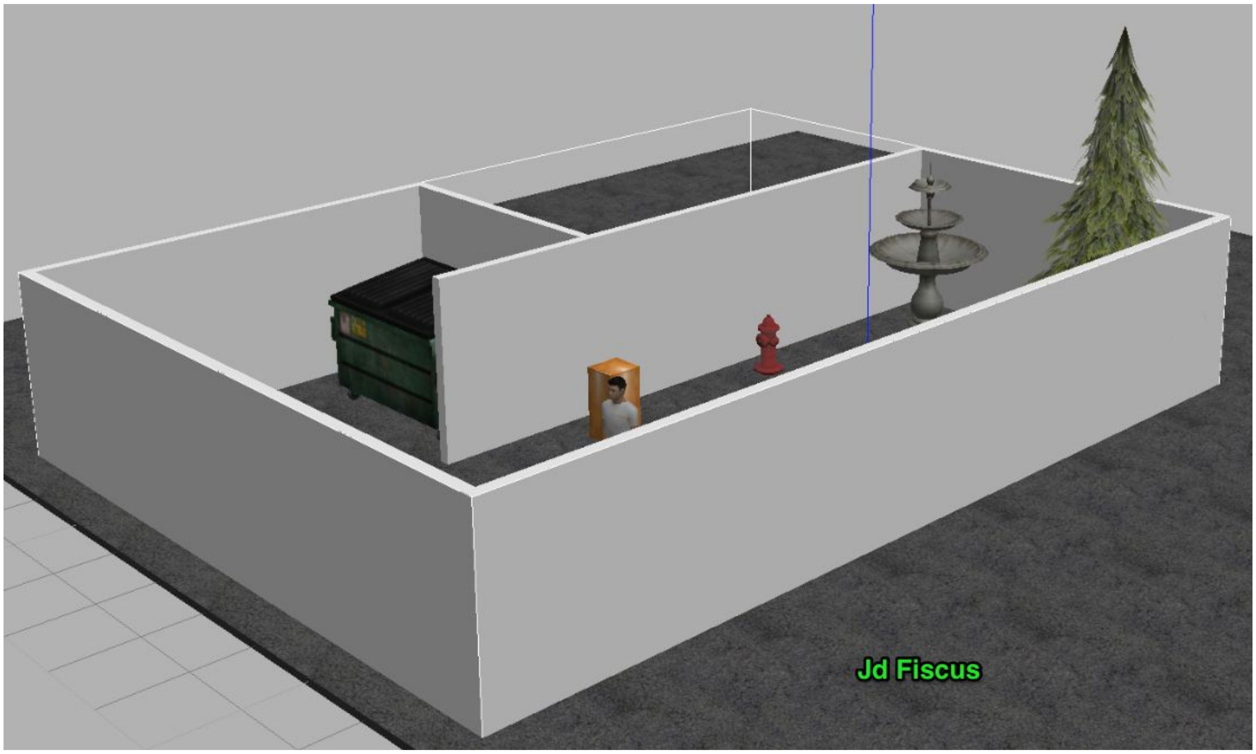
kitchen_dining

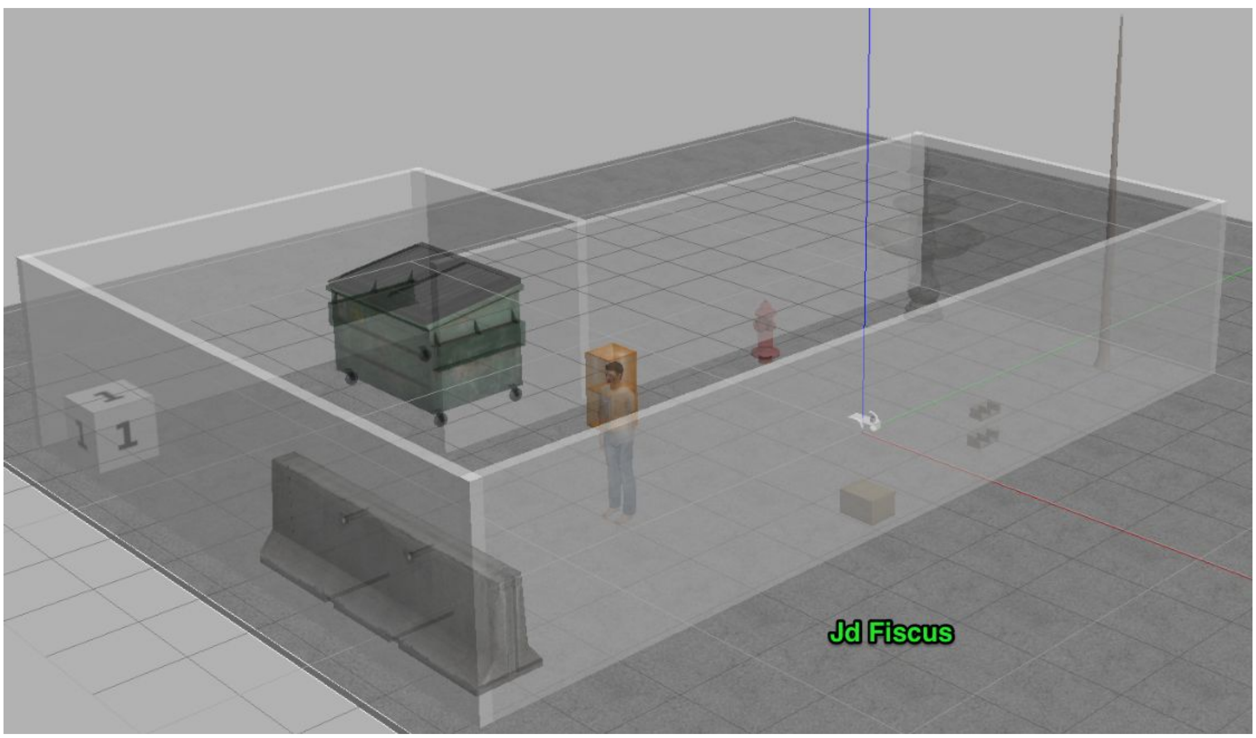


my_world



my_world

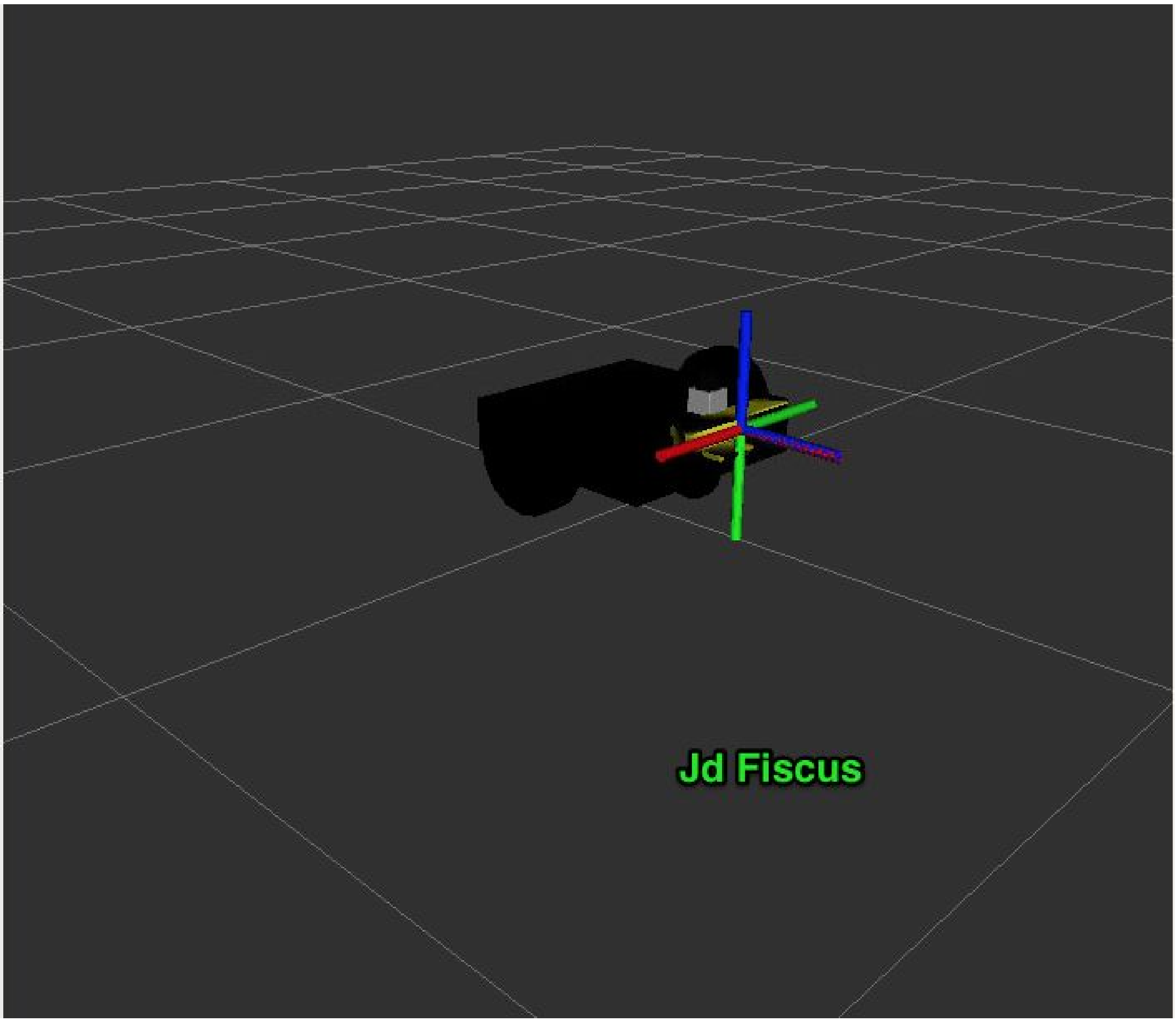




My_world_j

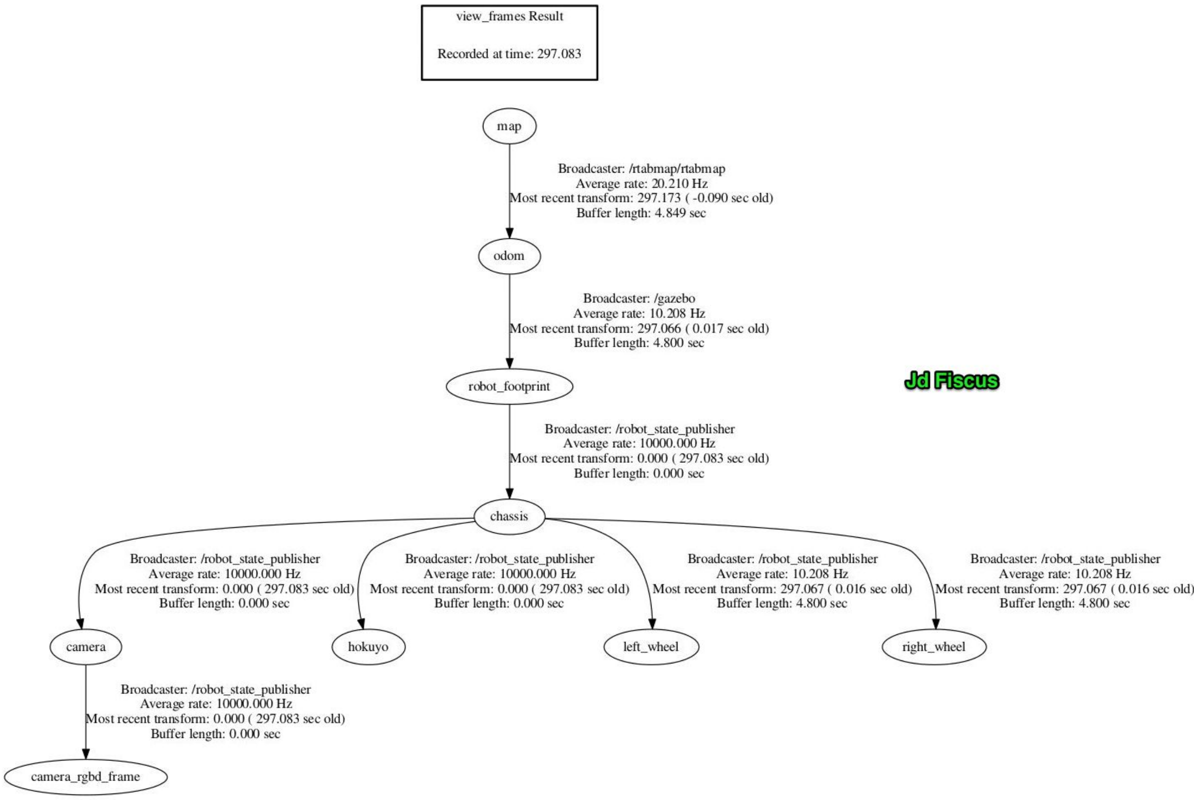
Robot Configuration

This robot model was based on the previous project called iamfiscus_bot, which is more square with actuators for the wheels. The hokuyo laser range finder stayed the same. However the camera previously used was replaced with a Kinect[5] leveraging openni_camera ros package[6] and Openni Kinect. Another change was adding an additional joint to rotate the the Kinect 180 degrees for data collection.



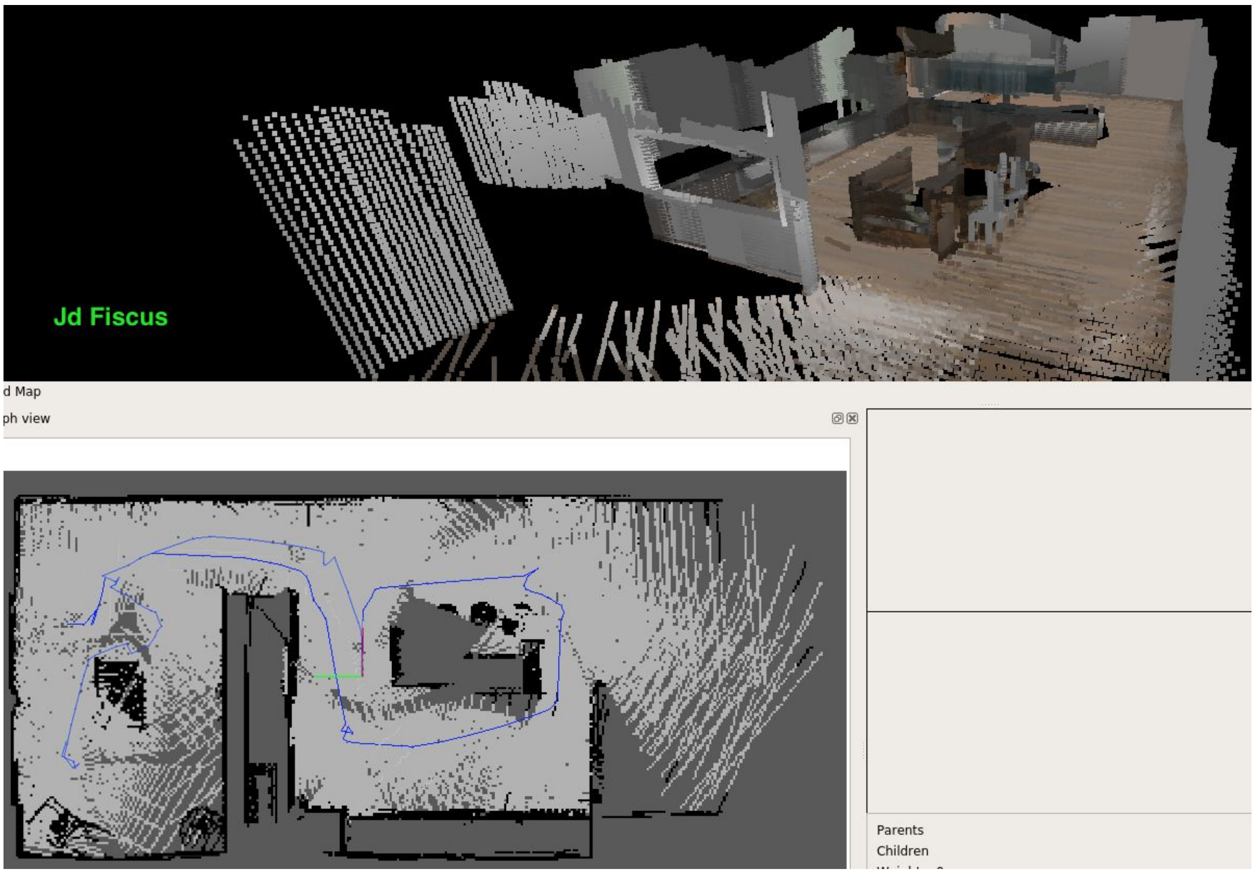
[5] "Study on the use of Microsoft Kinect for robotics ... - IEEE Xplore." https://ieeexplore.ieee.org/document/6236985/. Accessed 4 Jul. 2018.
[6] "openni_camera - ROS Wiki - ROS.org." http://wiki.ros.org/openni_camera. Accessed 4 Jul. 2018.

Below is the TF Frames graph of the robots configuration:



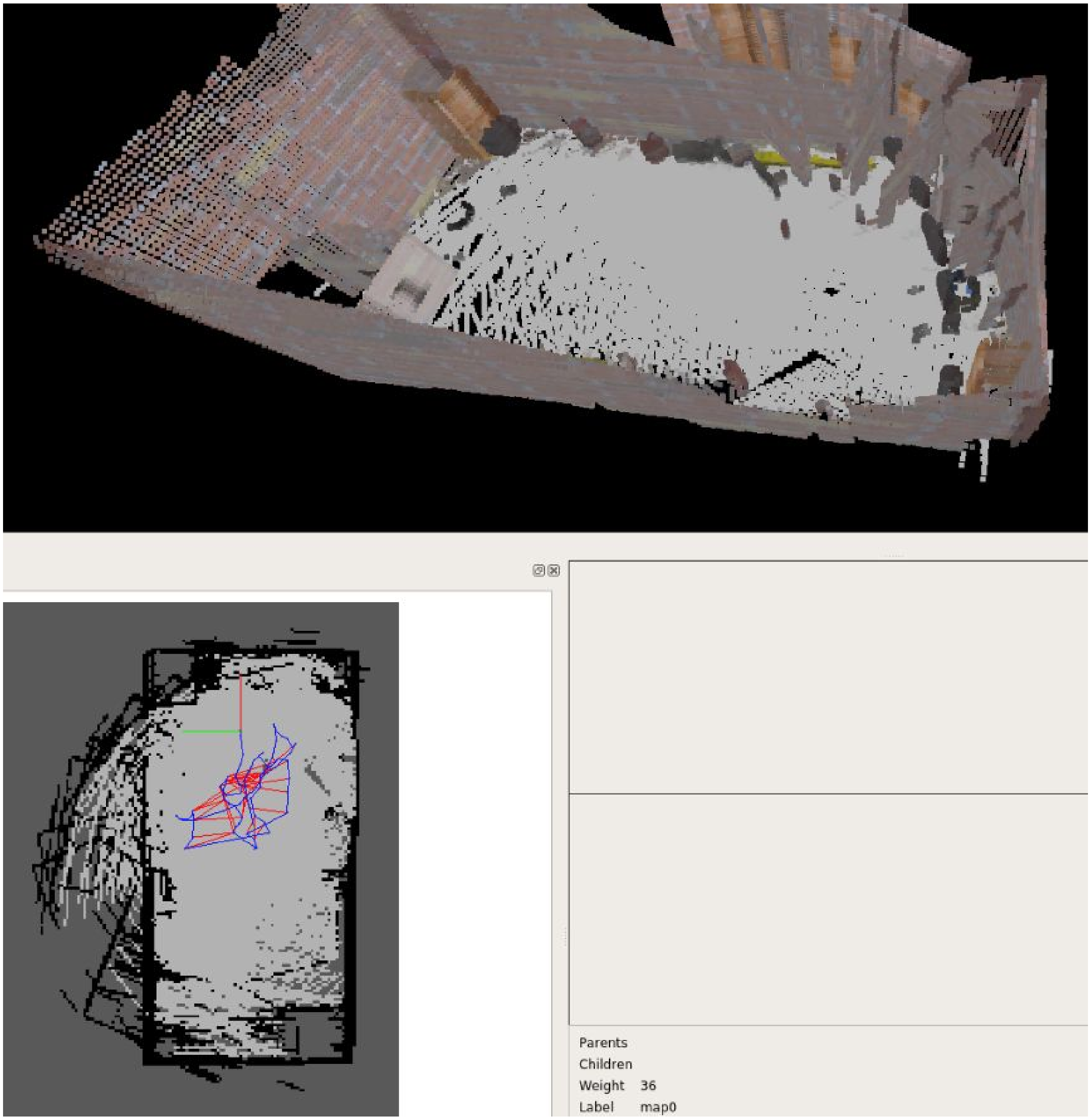
5 Results

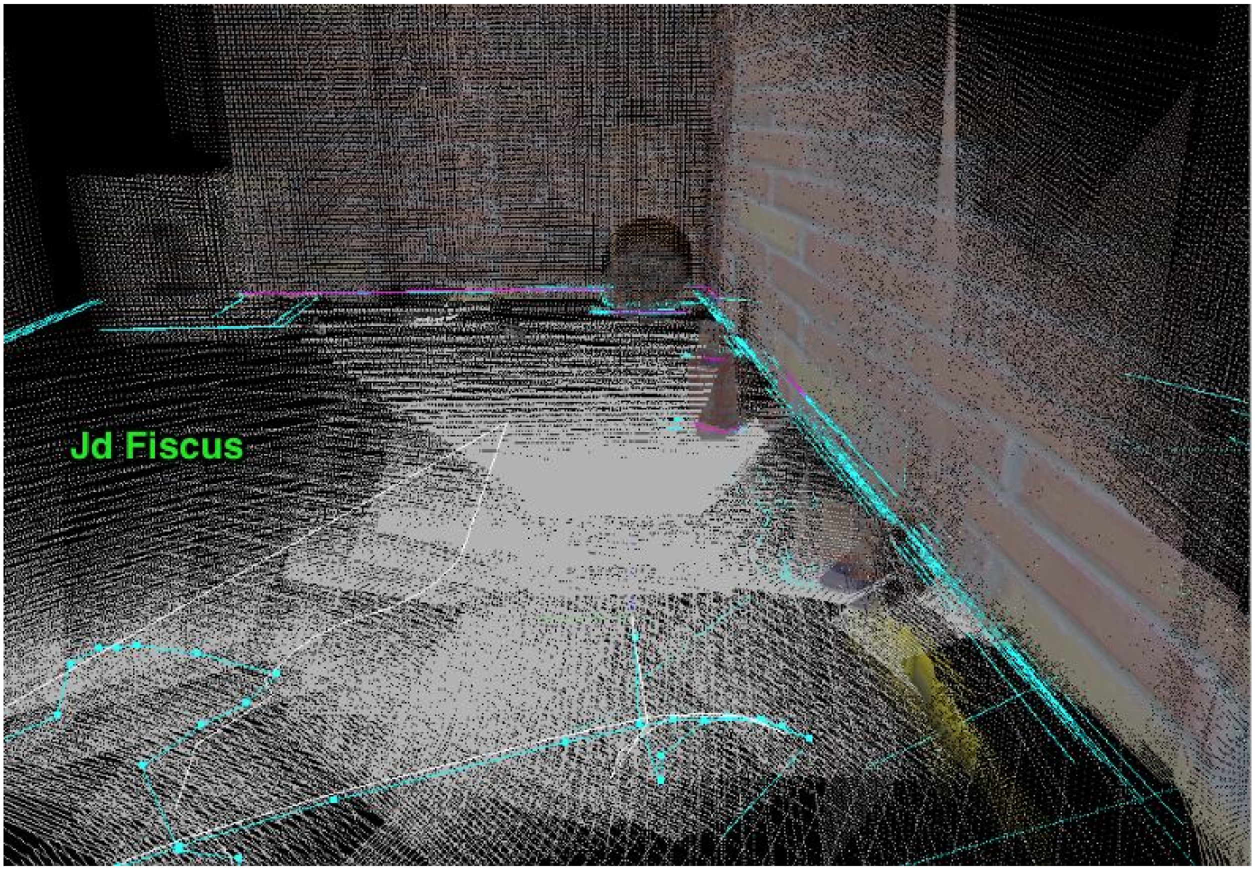
kitchen_dining



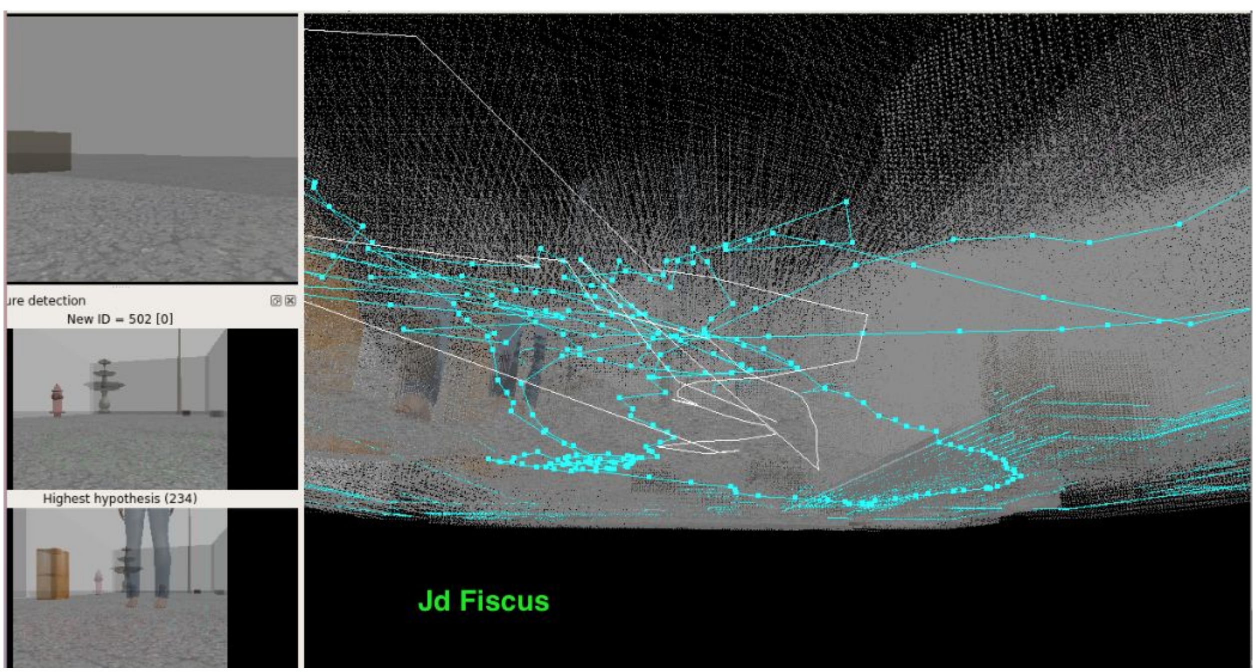
my_world

(3D & 2D maps)




Jd Fiscus

my_world_j


ure detection
New ID = 502 [0]
Highest hypothesis (234)
Jd Fiscus

6 Discussion

In this project the robot was controlled by teleop utilizing the keyboard and sometimes the robot would get stuck. It would take multiple passes in order to get enough data.

The kitchen and dining world seemed to produce the best result with the least amount of passes. This seems to be because there was plenty of unique objects and patterns that was collected by the camera. Also I think the fact that there were plenty of corners and short hallways.

However the custom created worlds my_world and my_world_j produced worse results and took more iterations. My world had to be customized multiple times in order to stop the robot from getting stuck because of false positives. Eventually it was able to make 3 passes of the room and produced an acceptable result. My world J probably would have worked better because of the shape of the room and content for SLUR. Unfortunately my computer would often crash under the resource strain which made data collection difficult

7 Future Work

LIDAR is one of the most interesting sensors for mapping and multiple LIDAR are often used in autonomous vehicles. Hopefully this will drive this technology down in cost and increase in quality. An example of this is Luminar's[7] technology which uses a longer wavelength and higher power.

Another thing to consider when talking about mapping is the data. The sheer quantity of data is massive so storage and data structure will impact the industry. It's estimated that a self driving car will produce 25 gigabytes[8] of data an hour. Sending that data effectively from the robot to the cloud will be an interesting problem to solve.

---

[7] "This new lidar sensor could equip every autonomous car in the world ...." 12 Apr. 2018, https://www.technologyreview.com/s/610858/this-new-lidar-sensor-could-equip-every-autonomous-car-in-the-world-by-the-end-of-2018/. Accessed 4 Jul. 2018.

[8] "Connected cars will send 25 gigabytes of data to the cloud every hour ...." https://qz.com/344466/connected-cars-will-send-25-gigabytes-of-data-to-the-cloud-every-hour/. Accessed 4 Jul. 2018.