Abstract

This work presents an attempt to map surrounding environment using ROS and Gazebo along with RTAB-Map, to create a 2D occupancy grid and a 3D octomap of two environments — one supplied and the other student created.

A previous robot creation was extended to modify different sensors output path to supply the necessary sensor data for RTAB-Map. This leverages the laser scanner, IMU/Wheel Encoder but replaces the camera with a RGB-D camera (i.e. kinect).

Further the ROS project is created with all links connected with appropriate naming and mapping.

The robot is launched and teleoped around the room to generate a map of the environment.

After successfully mapping the supplied environment, a student defined environment is created and mapped using the same technique.

Introduction

In this project a robot model uses a Simultaneous Localisation and Mapping (SLAM) technique called RTAB-Map (Real-Time Appearance-Based Mapping). It is a RGB-D Graph Based SLAM approach that uses incremental appearance based loop closure detection.

The RTAB-Map ROS package is coupled with visual representation in real time via rtabmapviz. The resultant map is stored in local database that be later explored via rtabmap-databaseViewer.

Background

When a robot encounters a new environment where there is no supplied map, it needs to be able to create this map and localise its pose using it. This combined localisation and mapping process is referred to as SLAM (Simultaneous Localisation and Mapping).

The main mapping algorithms are Occupancy Grid Mapping, Grid-based FastSLAM, Graph-SLAM and RTAB-Map.

The Occupancy Grid Mapping is a 2D algorithm where each grid cell is identified as Unknown/Undiscovered Zone, Free Zone or Occupied. This represents a slice of the 3D world.

The Grid-Based FastSLAM approach combines SLAM (Synchronised Location and Mapping) using a MCL (Monte Carlo Localisation) Algorithm and an Occupancy Grid Mapping. The main advantage of is the MCL particle filter approach but it always assumes there are known landmark positions. Thus it is unable to model an arbitrary environment.

Graph-SLAM uses a graph based approach to represent poses, features from the environment, motion constraints (between two poses) and measurement constraints (ties together a feature and a pose). It solves the full SLAM problem, it covers the entire path and map and not the most recent pose.

This project uses RTAB-Map, which is a Graph-SLAM approach that uses loop closure with Visual Bag-of-Words for optimisation.

The loop closure detection occurs against working memory to constrain the number of images interrogated. Working memory can be transferred and retrieved from long term memory to reduce complexity. The algorithm used for loop closure detection is SURF (Speeded Up Robust Features).

The possible outputs of RTAB-Map are 2D occupancy grid map, 3D octomap or a 3D point cloud.

Robots are of varying dimensions inclusive of height. Whilst mapping a 2d environment may show where fixed walls etc are it does not take into account height. A robot, that is propelled on the floor, may be able to navigate under some obstacles but not others e.g. a chair vs a large table. Hence it is required to understand the environment from a 3D perspective.

However, building a 3D map is costlier than a 2D map. This is not only in terms of computation power & data costs but also in the cost of the sensors required. Simple sensors such as a stereo camera may be used as a cheaper option but the algorithms required can be more complex.

Robot Model Configuration

The robot model used was based on the udacity_bot created in the previous project as the student robot model which had a square base with two actuators for the left and right wheels. The camera was replaced with a kinect leveraging the openni_camera ros package with the gazebo controller Openni Kinect.

No changes were made to the hokuyo laser range finder.

The udacity_bot configuration files can be found under the urdf directory which is renamed as slam_bot.

Visualization of the frames and rqt_graph as follows

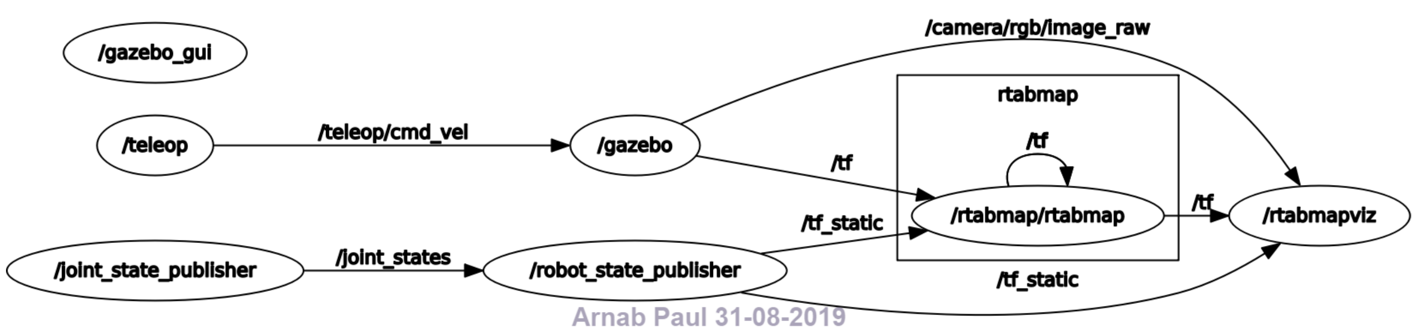


Figure1.rqt_graph

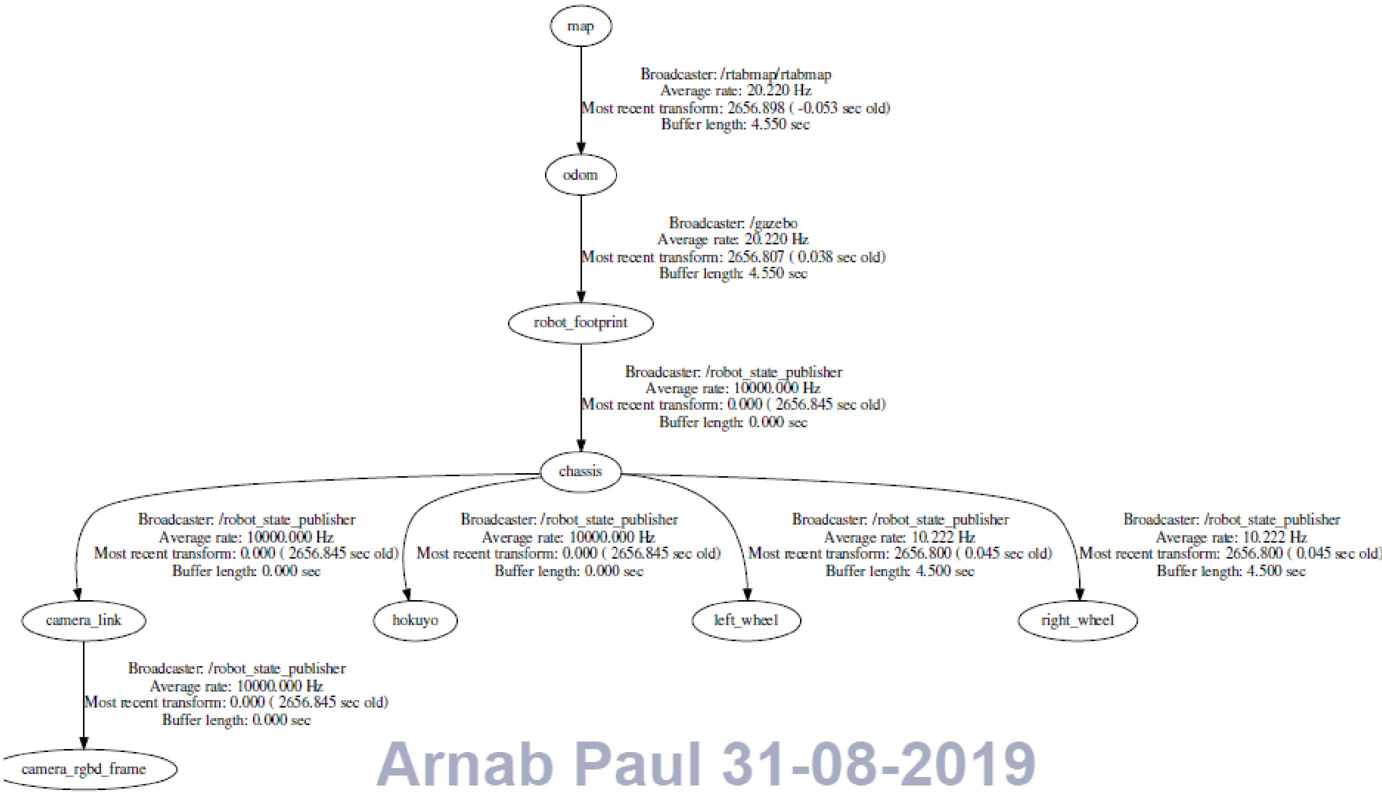


Figure2. view_frames output

World Creation

Two worlds were created in gazebo — one supplied as kitchen_dining.world and the other student customised my_street.world

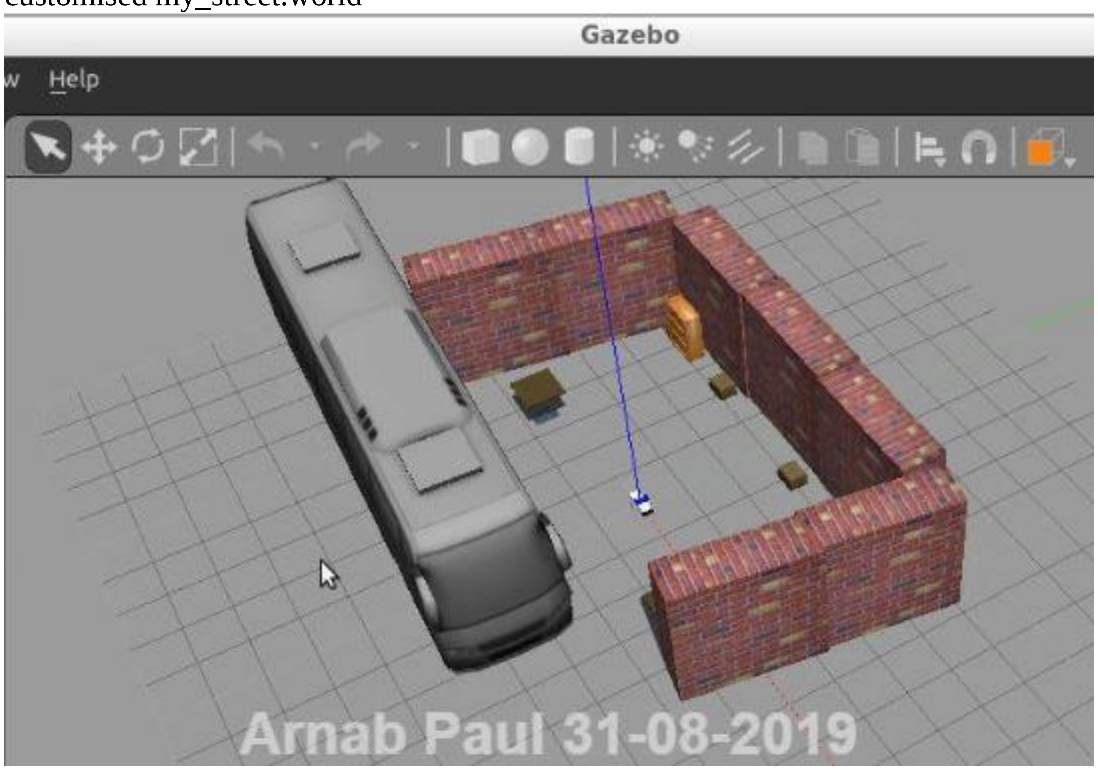


Fig 4. my_street.world

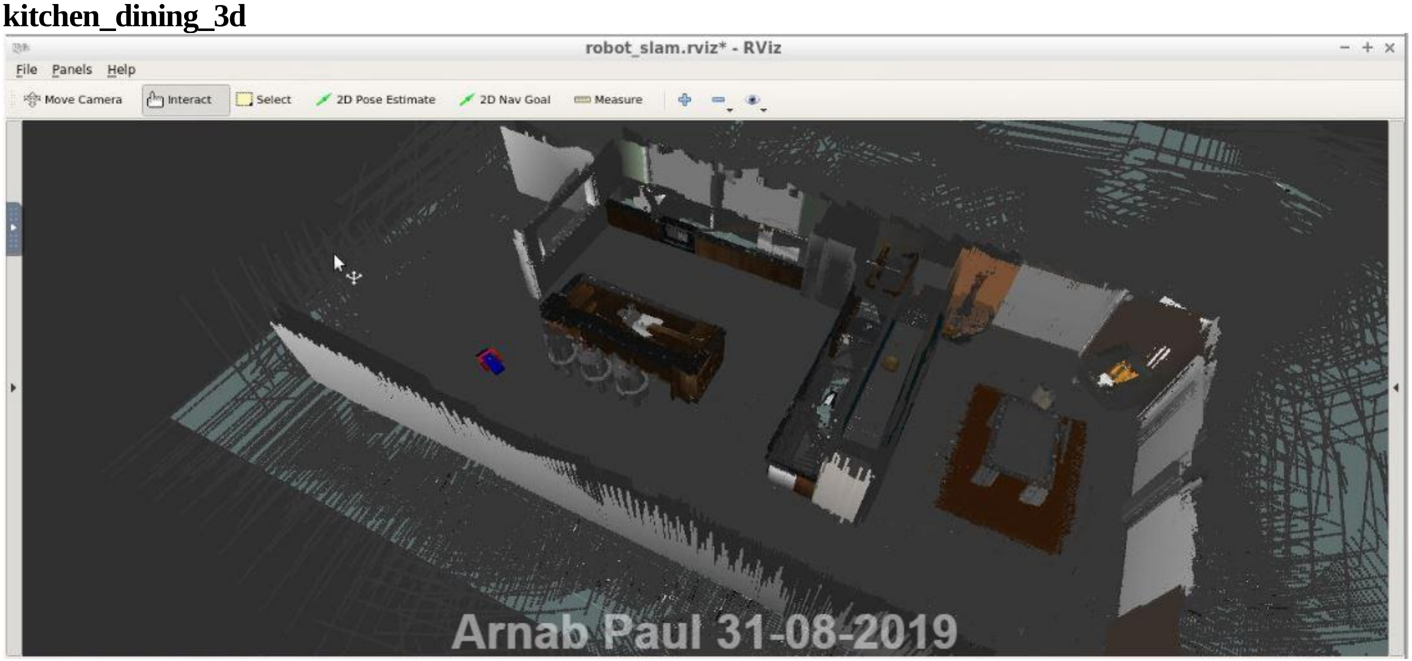
The following tree diagram depicts the package structure and files that made it up.

```
├── CMakeLists.txt
├── package.xml
├── launch
│   ├── config
│   │   └── robot_slam.rviz
│   ├── mapping.launch
│   ├── robot_description.launch
│   ├── rviz.launch
│   ├── teleop.launch
│   ├── teleop.py
│   └── slam_world.launch
├── materials
│   └── textures
│       └── kinect.png
├── meshes
│   ├── hokuyo2.dae
│   └── kinect.dae
├── rtab_run
├── urdf
│   ├── slam_bot.gazebo
│   └── slam_bot.xacro
└── worlds
    ├── kitchen_dining.world
    └── my_street.world
```

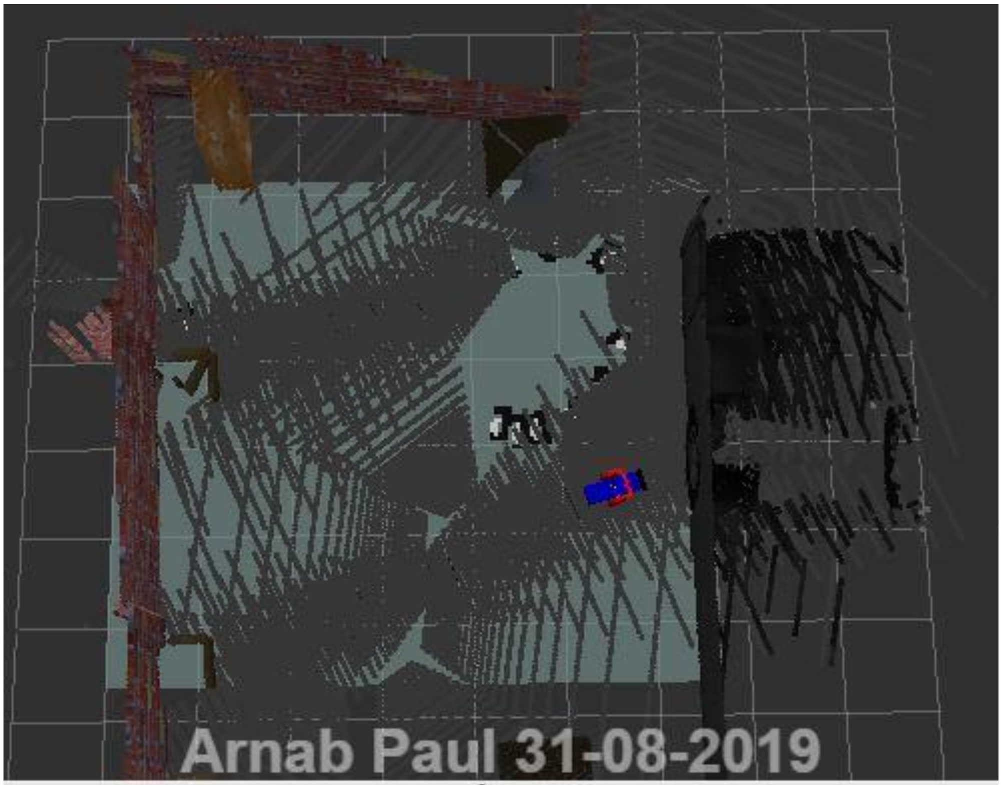
This tree structure was chosen based on the previous student project that conformed to known gazebo/ros standards.

Results

kitchen_dining_3d



my_street_3d



Discusion

The robot was teleoped (navigated via the keyboard) around the room. While mapping the student created world, at some points the robot did not move forward. This appeared to be when it started to perform loop closure. Kp/MaxFeatures was selected to 500 and Reg/Strategy was chosen 1(option for ICP), Vis/MinInliers was reduced from 15 to 10.

However the 3D map quickly started to resemble the physical kitchen dining gazebo model. To improve loop detection rates some, on the spot circles were performed. More SURF features were identified there as there was more variation in the surfaces.

The my_street gazebo model wall surfaces were bricked, repeatable pattern with lack of other discerning features sometimes caused the loop closure detection to map to an incorrect previous image. This then distorted the map. Additional features were added to achieve a successful map.

The kitchen_dining model performed significantly better than the student created my_street model. This was due to the richer and more complex features of the kitchen_dining model.

Future Work

We live in an ever-changing environment and 3D/2D mapping is a crucial part for robot to be able to safely navigate through the environment. 3D mapping such as the one presented here also has many application including navigating rooms for home robot, rescue mission, navigating unexplored terrain for both aerial and ground robots. As the price for 3D mapping sensors (RGBD camera, Laser scanner and IMU) continues to drop, it will open up opportunities to create richer and more realistic 3D maps at a cheaper price. This work could be further advanced to implement in a robotic prototype and use the simulation information found in this project to troubleshoot real world problems.