SLAM Project: Map my world

Gassó Loncan, J.C.

Abstract—A custom robot model equipped with a Kinect RGB-D camera and hokuyo laser maps two different environments; one provided by Udacity and another designed by the author. This report focuses on the implemetation of SLAM principles using Real-Time Appearance-Based Mapping (RTAB-Map). This ROS package is based on camera feedback from an RGB-D sensors.

Index Terms—Robot, Udacity, SLAM.

1 Introduction

One of the most important and difficult challenge in robotics is to estimate where a robot is located. The previous Udacity's project was referred to Localisation, where a robot is provided to map its environment. The robot had access to its movement and sensor data and uses them to estimate its pose. However, in occasions there isn't a known map because the area is unexplored or because the surrounding change often; therefore the map is not up to date. In such case, the robot is need to create a new map, and this leads to mapping.

Mapping assumes that the robot knows its pose and as usual has access to its movement and sensor data. The robot must produce a map of the environment using the known trajectory and measurement data. However, even such case can be quite uncommon in the real world. Most of the time the robot would have neither a map nor know its pose, and this is where SLAM comes in.

The project's aims to use a custom model robot equipped with an RGB-D camera and laser range finder to generate a map of two different worlds. The first environment is provided by Udacity and the second one is created using the Gazebo Building tool. ROS packages which implement RTAB-Mapping was used to create 2D and 3D maps of each environment.

2 Background

SLAM is difficult task to solve as map is needed for localisation and the Robot's poses is need for mapping. The problem is more challenge than localisation and mapping isolated since neither the map nor the robot poses are usually provided.

There are several algorithm to face this problem. The two more important are Fast-SLAM and Graph-SLAM. Graph-SLAM is a SLAM algorithm that solves the full SLAM problem. This means that the algorithm recovers the entire path and map, instead of just the most recent pose and map.

This difference allows it to consider dependencies between current and previous poses. One advantage is the reduced need for significant on board processing capability. Another advantage that can be immediately appreciated is Graph-SLAM's improved accuracy over Fast-SLAM.

Fast-SLAM uses particles to estimate the robot's most likely pose. However, at any point in time, it's possible that there isn't a particle in the most likely location. In fact, the chances are slim to none especially, in large environments. Since Graph-SLAM solves the full SLAM problem, this means that it can work with allow of the data at once to find the optimal solution. Fast-SLAM uses titbits of information with a finite number of particles, so there is room for error.

In this project, RTAB-Map together with sensor information gathered from a Kinect RGB-D camera and hokuyo laser are used to obtain a 2-D occupancy grid map and 3-D point cloud of two simulated environments.

RTAB-Map (Real-Time Appearance-Based Mapping) is an Open Source RGB-D Graph-Based SLAM approach based on an incremental appearance-based loop closure detector. The loop closure detector uses a bag-of-words approach to determine how likely a new image comes from a previous location or a new location. When a loop closure hypothesis is accepted, a new constraint is added to the map's graph, then a graph optimiser minimises the errors in the map. A memory management approach is used to limit the number of locations used for loop closure detection and graph optimisation, so that real-time constraints on large-scale environments are always respected. [1]

3 Model configuration

The robot model is similar to the UNER bot used in the previous project. The robot model chassis is made by a rectangular box of 0.4x0.2x0.1 meter with a cylinder on the front. The cylinder radius is 0.1 meters and 0.1 of length. Two wheels with a radius of 0.1 meters are located at both sides of the chassis, in the front side while a passive back caster is at the back.

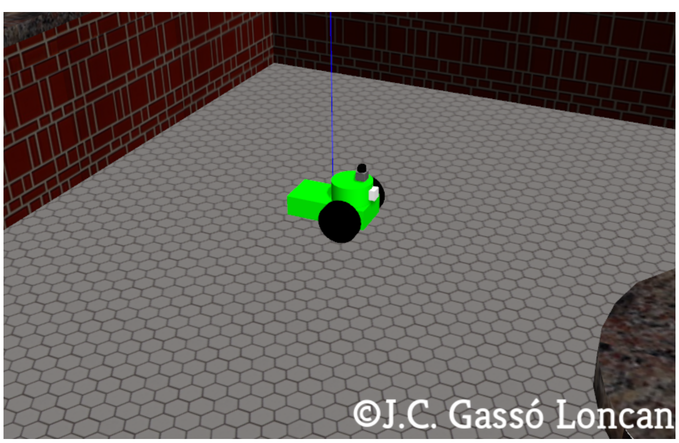

Fig. 1. Robot model Uner bot.

The main difference with the last UNER bot model lies in the location of the sensors. The robot is equipped with a laser range finder, as the previous model, and a RGB-D camera.

The laser range finder was moved to the top of the robot's neck. The camera is located at the front of the chassis. The robot can be found in figure 2.

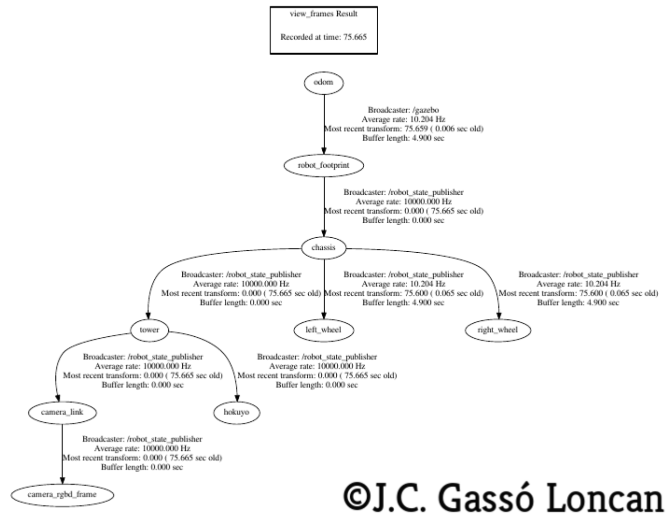

Fig. 2. Frames Uner bot.

4 World creation

Two worlds was used in this project. One provided by Udacity called Kitchen dining and the another was designed by report's author.

Fig. 3. Map provided by Udacity.

The designed custom world can be seen in figure below. It was used one template available in Gazebo called Cafe. Chairs, tables and walls was added in order to increase the complexity of the scene.

Fig. 4. Custom map: "My cafe".

5 RESULTS

In order to simulate the project, it was necessary to launch world, mapping and Rviz files, and run the teleop script. This last was used to navigate the map via keyboard.

5.1 Kitchen dining world

The 5 shows the results of SLAM. It was recorded by Rviz and saved in a database using RTAB-Map package. The robot was able to correctly map the environment, rebuilding almost everything with a great level of detail. Fourteen loop closure were achieved during the simulation.

Fig. 5. 3D view Kitchen dining after SLAM.

From the the 2D map showed in 6, one can see where the kitchen island, chairs and table lay as well as the sofa in the corner of room.

5.2 Custom My cafe world

The results of SLAM using the custom map was successfully mapped and navigated. As in the kitchen dining case, 3D map was created with great detail.

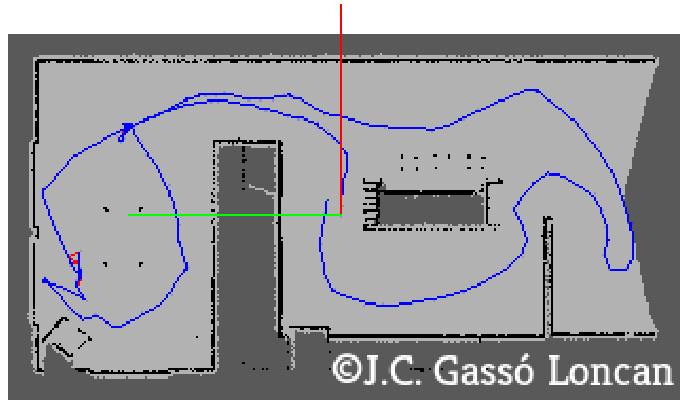

Fig. 6. 2d map of Kitchen dining.

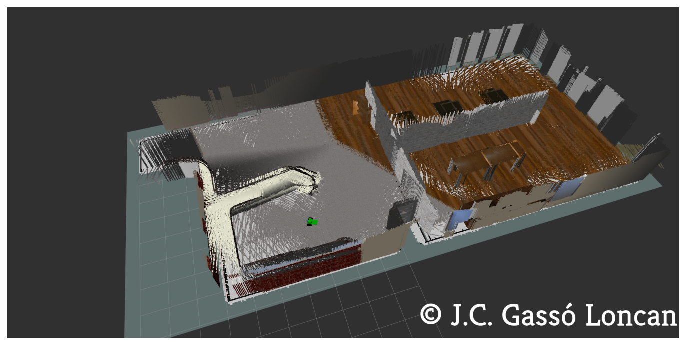

Fig. 7. 3D view Kitchen dining after SLAM.

The 2D map showed in 8 reflect on the successful result. It was necessary just one loop around the world to obtain a decent map.

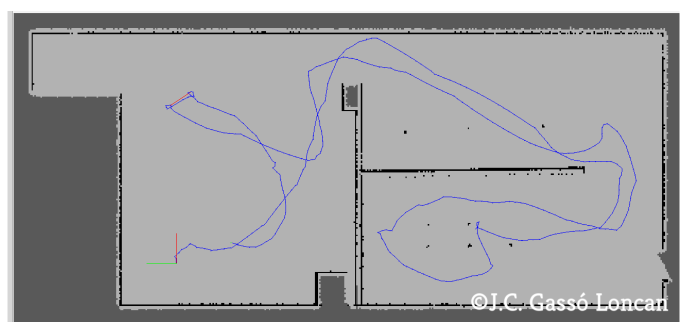

Fig. 8. 2d map of Kitchen dining.

6 DISCUSSION

The performance of the robot's mapping for each environment have a few similarities. Both were capable of clearly mapping 2D map boundaries, except for the unbounded section of the Kitchen Dining area. The missing piece unmapped on the right side is likely due to that side there are windows which could affect the detection. Both world have been navigated via keyboard at very low velocity. Increasing the velocity produce mismatches in the 3d reconstruction. The results of the 3D maps are a slightly better in Kitchen dining than the customised world. More features in the provided map may be the reason of this difference.

7 Conclusion / Future work

SLAM is an important approach in robotics. This face real life problems where location and mapping are needed simultaneously. On the other hand, designing environment demonstrates being a important matter.

Future works could analyse the parameters of this SLAM package in depth. RTAB-Map has been demonstrated being a very powerful tool.

References

[1] . ROS wiki: Open Source Robotics Foundation, “Roswiki: Rtab-map.” ”http://wiki.ros.org/rtabmap_ros”, 2009 (accessed Jun 10, 2018.