Map My World Project

Xiaodong Wang

Abstract

Mapping a location (e.g. an kitchen) using sensor data while localizing oneself in the continuously updated environment is referred to as Simultaneous Localization And Mapping (SLAM). In this project the Real-Time Appearance Based Mapping (RTABMap) is done using the rtabmap_ros package. A mobile robot is used to map two simulated worlds created in Gazebo.

1. Introduction

The goal of this project is to create maps for two simulated worlds using Real-Time Appearance Based Mapping (RTABMap). As the name suggests, SLAM is Simultaneous Localization and Mapping, it is used for estimating the pose of a robot and mapping of the environment at the same time. Localization is inferring location in given map. Mapping is inferring map given location.
Applications of SLAM include: vacuum cleaner, lawn mower, surveillance with unmanned air vehicles, reef monitoring, exploration of mines, terrain mapping for localization and many more...

2. Background

2.1 SLAM

Simultaneous localization and mapping (SLAM) is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it." To rephrase it, what the SLAM application does is basically translate the data gathered from the outside world into a virtual environment. This is used in apps like Tesla's Autopilot, where the sensors are able to translate the outside world's data into the car's head computer, who then compiles them in a virtual projection of the surroundings, that is used to avoid crashes and better inform the driver about the journey.

In SLAM, you will map the environment, giving the noisy measurements and localize robot relative to its own map giving the controls. This makes it a much more difficult problem than localization or mapping since both the map and the poses are now unknown to you. In real-world environments, you will primarily be faced with SLAM problems and you will aim to estimate the map and localize the robot.
An example of a robot solving its SLAM problem is robotic vacuum cleaner that uses the measurements provided by its laser finder sensors and data from the encoders to estimate the map and localize itself relative to it.

2.2 Form and Nature

SLAM has two forms: 1. Online SLAM: Robot estimates its current pose and the map using current measurements and controls. 2. Full SLAM: Robot estimates its entire trajectory and the map using all the measurements and controls.

SLAM has two natures: 1. Continuous: Robot continuously senses its pose and the location of the objects. 2. Discrete: Robot has to identify if a relation exists between any newly detected and previously detected objects.

2.3 SLAM Challenges

Computing the full posterior composed of the robot pose, the map and the correspondence under SLAM poses a big challenge in robotics mainly due to the continuous and discrete portion.
Continuous: The continuous parameter space composed of the robot poses and the location of the objects is highly dimensional. While mapping the environment and localizing itself, the robot will encounter many objects and have to keep track of each one of them. Thus, the number of variables

will increase with time, and this makes the problem highly dimensional and challenging to compute the posterior.
Discrete: The discrete parameter space is composed out of the correspondence values, and is also highly dimensional due to the large number of correspondence variables. Not only that, the correspondence values increase exponentially over time since the robot will keep sensing the environment and relating the newly detected objects to the previously detected ones. Even if you assume known correspondence values, the posterior over maps is still highly dimensional.

3. Scene and robot configuration

3.1 Robot Configuration

The robot used for this project was an updated version of the robot used in the localization project. The main modification was replacing the front camera with an RGB-D camera (kinect camera).

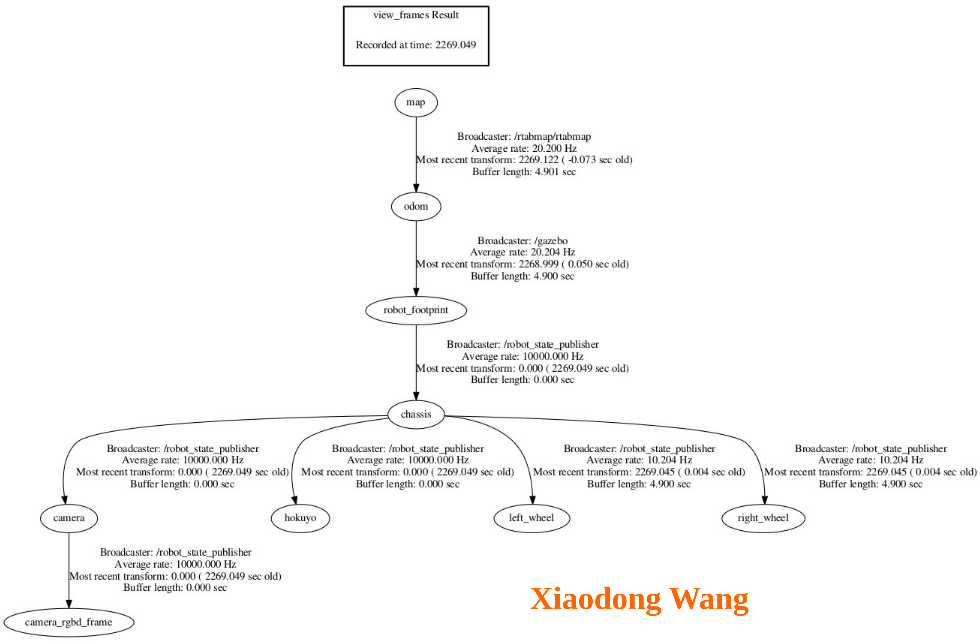

Fig. 1. Tf Tree of the Robot

3.2 World Configuration

3.2.1 Kitchen Dining - Udacity World

Mapping was done on two different worlds, first the 'kitchen and dining' world provided by Udacity, as shown in figure 2.

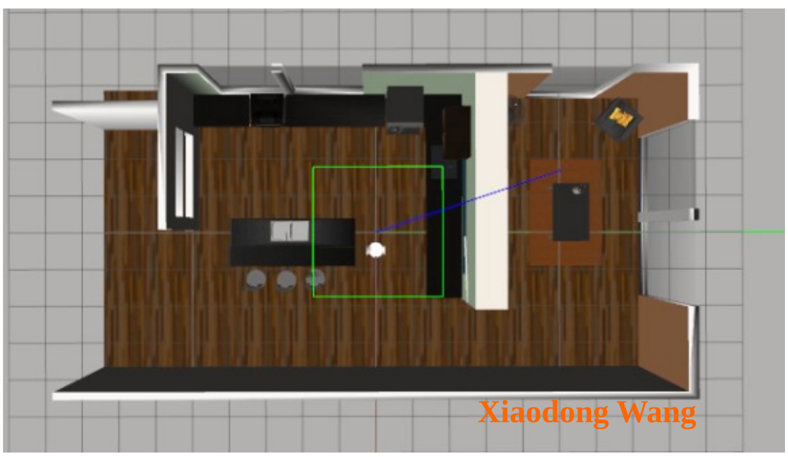

Fig. 2. Kitchen Dining – Udacity World

3.2.2 Custom World

A second custom world was made in Gazebo as shown in figure 3.

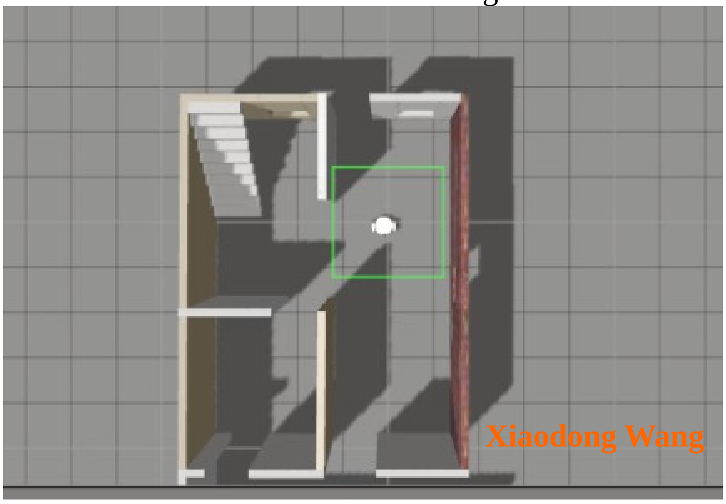

Fig. 3. Custom World

4. Results

4.1 Mapping Kitchen and Dining

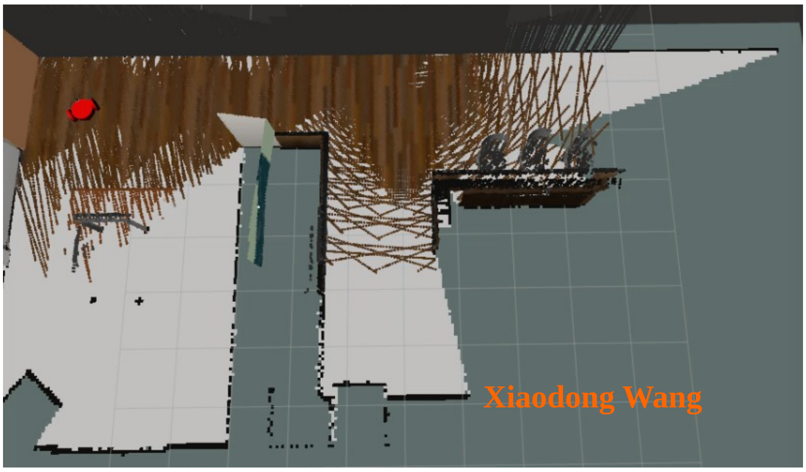

Fig. 3. Mapping the Kitchen Dining World

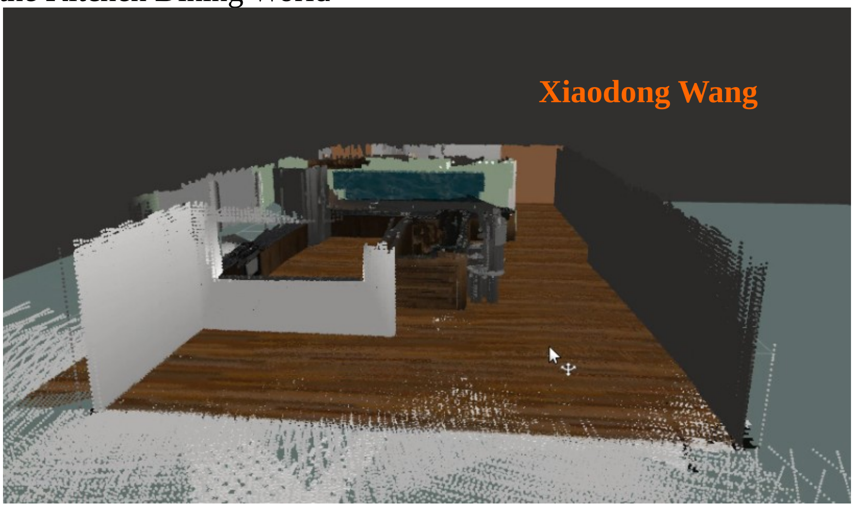

Fig. 4. 3D Map Generated for the Kitchen Dining World

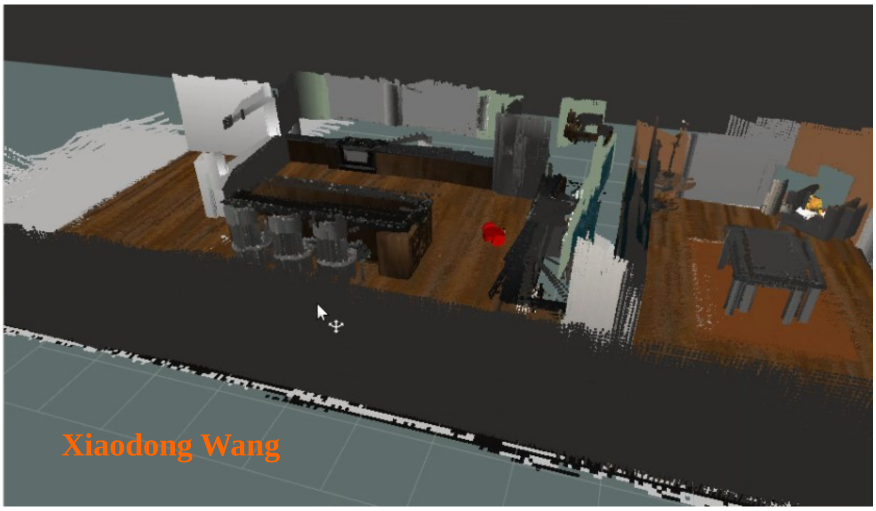

Fig. 5. 3D Map Generated for the Kitchen Dining World

4.2 Mapping the Custom World

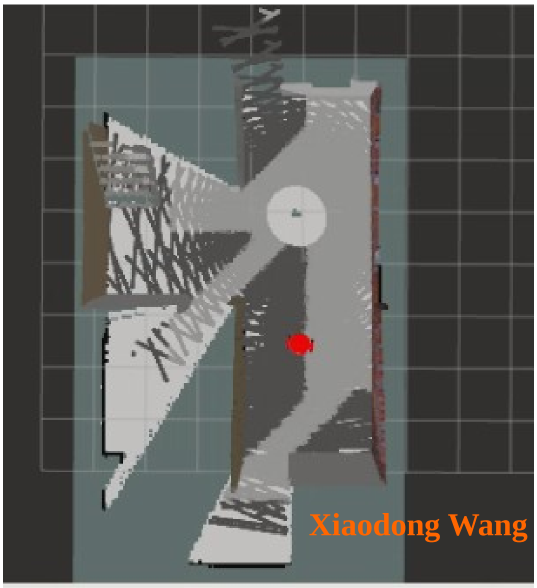

Fig. 6. Mapping the Custom World

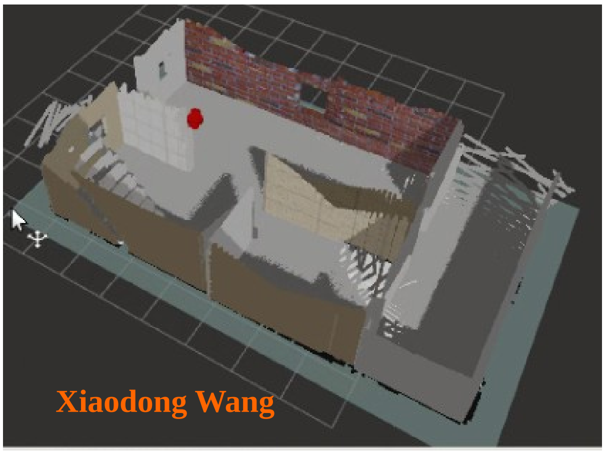

Fig. 7. 3D Map Generated for the Custom World

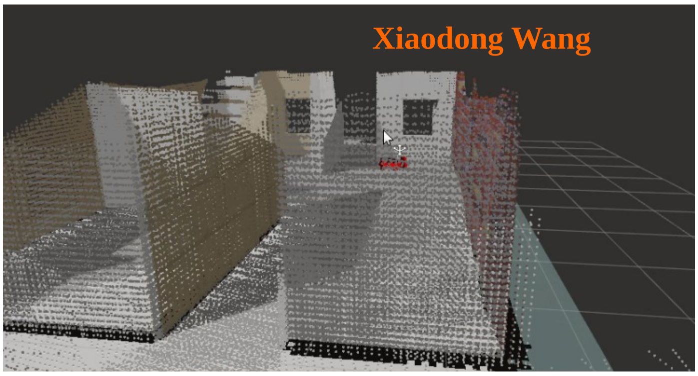

Fig. 8. 3D Map Generated for the Custom World

5. Discussion

When mapping the two worlds, a number of issues need to be solved. First, the size of the environment is often much more bigger than the perception range of the robot. Second, many similar scenes in the environment may increase the difficulty of mapping. Fortunately, RTABMap is able to solve these issues.

We can increase the number of robots or extend the perception range. These two methods can also be combined. New and more advanced sensors may be mounted to the robot. We can also tune the rtabmap's parameters to gain a better result.

6. Future Work

RTAB-Map is indeed a very powerful way of mapping. I am planning to deploy this method to a real robot, and to map indoor and outdoor environments. Then analyze the results and find ways to refine the algorithm. SLAM is a very challenging field, an outstanding SLAM engineer should not always use the package made by others.