Report: Map My World

Xuan He

Abstract—In this project, a Simultaneous Localization and Mapping (SLAM) algorithm was applied to simulated environments to generate maps. In the simulation, a robot buggy model was built with a RGBD camera and a laser range finder. The SLAM algorithm used is Real Time Appearance Based Mapping (RTAB-MAP), a type of GraphSLAM algorithm that uses traditional computer vision techniques (bag of words) to solve the correspondence problem. There are two simulated environments: the kitchen dining environment is provided and the other one is built. In both environments, the package *RTAB-MAP* package was utilized to generate the map. The algorithm works well in the first environment, where the map features are distinguishable from each other. However the mapping performance were worse when the robot explores an environment with visually similar regions and failed to correlate different frame correctly.

Index Terms—Robot, SLAM, RTABMAP

1 INTRODUCTION

IN order to efficiently solve many tasks performed by mobile robots including transportation, search and rescue or automated vacuum cleaning, a map of the environment is necessary for mobile robots. The availability of an accurate map allows for the design of the systems that can operate in complex environments only based on their on-board sensors. Learning maps under pose uncertainty is often referred to as the simultaneous localization and mapping (SLAM) problem [1]. One popular approach to the SLAM problem is via graph-based formulation, so called GraphSLAM. Solving a graph-based SLAM problem involves the construction of a graph whose nodes represent robot poses or landmarks and in which an edge between two nodes encodes a sensor measurement that constrains the connected poses. Once such a graph is constructed, the crucial problem is to find a configuration of the nodes that is maximally consistent with the measurements. This involves solving a large error minimization problem.

This project is an extension of previous localization project. It explores how the SLAM methods works in a simulated environment. The robot uses a RGB-D camera and estimates its trajectory and map features poses as it moves through the environment. If the SLAM algorithm is successful, the robot would be able to produce a map with recognizable features in the surroundings and its trajectory.

2 BACKGROUND

In localization, the robot is provided with map, measurements and motions to estimate its own poses where SLAM is much more challenging problem, since the only information the robot has is the measurements and control input. As a result, the solution space for SLAM problem is highly dimensional, rendering many analytic solutions unsuitable for the task. In addition, SLAM algorithms also need to identify object correspondences between images in order to resolve map features correctly. However, the correspondence values increase exponentially over time since the robot captures more images as it explores the environment.

The SLAM problem has two variations - Full SLAM and Online SLAM. The Online SLAM problem tries to estimate the current pose and the map features from the previous measurements and motion, while the Full SLAM attempts to estimate the entire trajectory and the map features at the same time. Many algorithms solve the SLAM problem on different scopes, including FastSLAM and GraphSLAM.

FastSLAM is a landmark-based algorithm that extends the Monte Carlo Localization particle filter algorithm to estimate both the landmarks and robot poses. The Grid-base FastSLAM algorithm is a version of FastSLAM that combines the MCL and Occupancy Grid mapping, a mapping algorithm that outputs grid cells labelled as occupied, free and unknown. However, either verions of FastSLAM have problems with arbitrary environments since they assume known landmark locations.

On the other hand, the GraphSLAM algorithm represents the SLAM problem as a graph where the poses of the trajectory and the measured locations are the nodes, the links are the estimated motion and measurement distances represented as constraints. The algorithm solves for the best configuration of the graph to satisfy the constraints as much as possible, thus solving the Full SLAM problem.

This project uses the Real Time Appearance Based Mapping (RTAB-Map), an upgraded version of GraphSLAM that uses a bag of words approach to identify correspondences between frames using visual similarity. The algorithm detects loops in the trajectory with features such as SURF or SIFT. At every frame, the camera image would be compared against previous found map features in a constant-size working memory. This allows the algorithm to map trajectories with repeated locations by reducing the constraint complexity, and the algorithm can run in constant time with a proper memory management scheme.

3 ROBOT MODEL CONFIGURATIONS

The robot was built from previous project. The robot was shown in the following figure. The old camera was replaced with a kinect RGB-D camera. The robot has four wheels and round chassis for easy turning. The laser range finder and camera was stacked to have a better sensing range.The visualization of the transform frames are also provided.

Fig. 1. Student Robot.

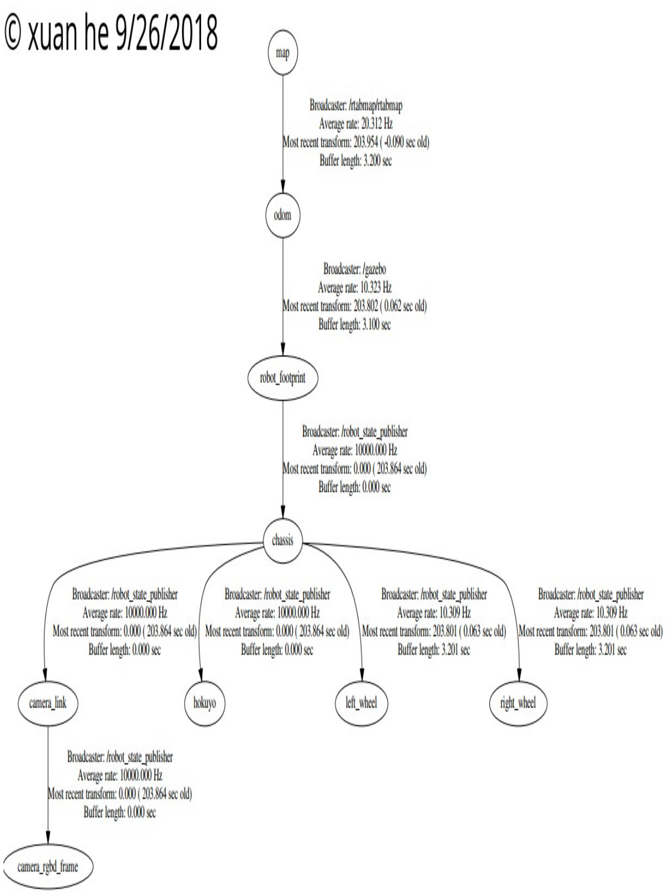

Fig. 2. tf frames.

4 Scene Creation

A new scene is built from scratch in Gazebo. The models in the scene are all from online model database. A square root was constructed with four stone walls, and placed many robot parts close to the walls in the room. Then stone walls are replaced with different types of walls to avoid featureless scenes. No repeated objects except for the walls parts were present in the scene. In addition,

Fig. 3. Udacity Kitchen Dining Scene

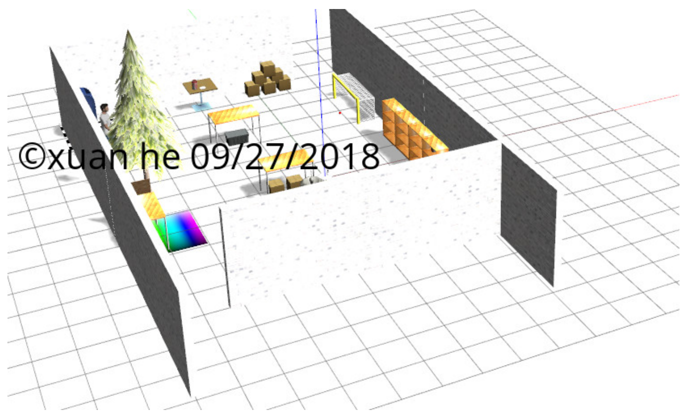

Fig. 4. Student Created Scene

5 Results

5.1 Kitchen Dining Scene

The generated mapping of the kitchen dining area is provided below including a 2D map and a 3D point cloud map. For kitchen dining environment, the results are very accurate by visual inspection. In 2D and 3D map, the boundaries are correctly generated. Compare the 2D map with 3D map, most obstacles such as walls and items were correctly detected and shown on the map. Although the robots were not able to detect higher items, those do not stop robots from moving. The location of the laser range finder and cameras are placed to make sure the robots can always get through the unobstructed space. The left side has a weird shape in the 2D map which is due to the missing boundary in the scene.

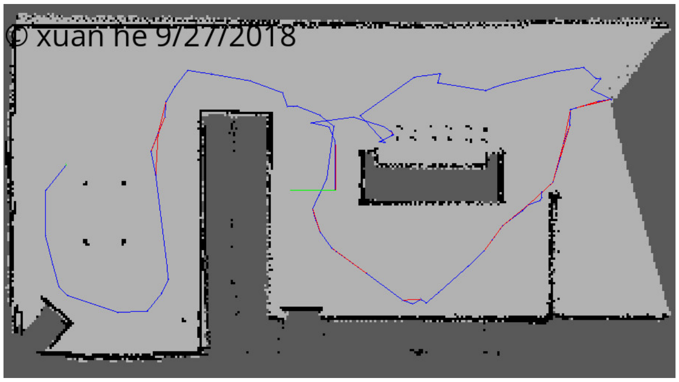

Fig. 5. Generated map: kitchen dining 2D map.

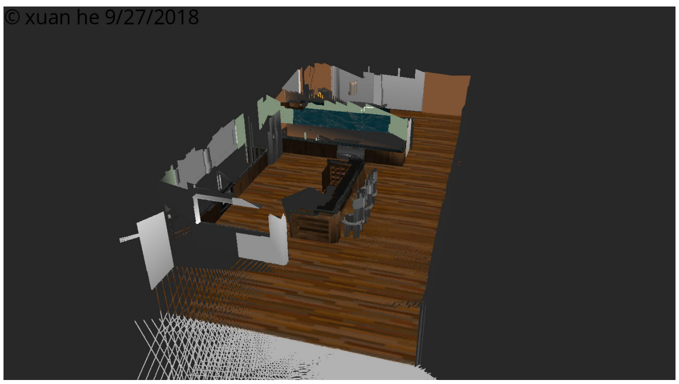

Fig. 6. Generated map: kitchen dining 3D map.

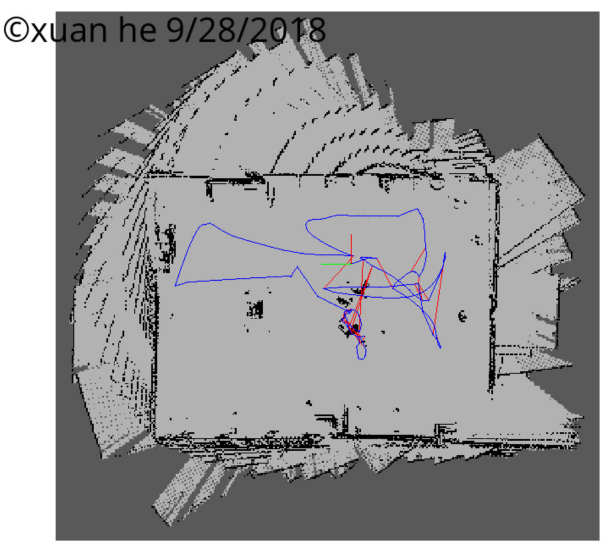

Fig. 7. Generated map: scene 2D map.

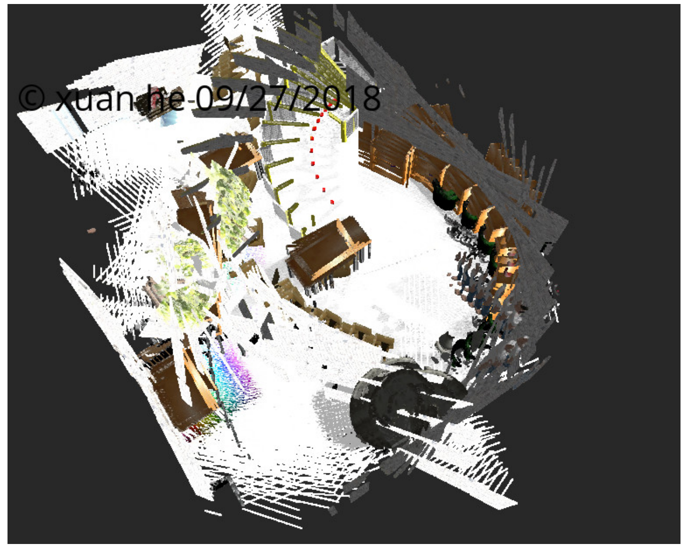

Fig. 8. Generated map: kitchen dining 3D map.

5.2 Custom World

For custom scene, the robot was able to map part of boundaries in the scene as shown in Fig 6 and Fig 7. But the accuracy was not good. This could be due to several reasons. First, the custom scene does not have richer features like kitchen dining scene. Since rtabmap is using bag of words, less features would result in bad accuracy. Also, the positions of the sensors might not be ideal which led to a poor perception.

6 Discussion

The robot mapping can perform much better in the kitchen dining area than the custom scene. The kitchen dining scene has richer and more complex features than the custom scene. RTAB-map uses the bag of words approach to identify reoccurring objects. However, the method needs tuning and sometimes produce false positive of loop closures.
RTAB-map does not perform well when there are dynamic objects in the environment. Since RTAB-map assumes the objects to be static in the environment, when it detects an object appearing at different locations. A more advanced detection algorithm would be required to distinguish a moving object.
In the simulation, the robot was navigated via keyboard input around the room. Sometimes, the robot would get stuck in places. This appeared to be when it started to perform loop closure.

7 Conclusion / Future work

SLAM has varieties of application in robotics such as automatic vacuum robots and self-driving cars. This technology is crucial for applications where the map of the environment is very difficult for humans to obtain, where robots can enter and perceive the environment directly. For future work, the algorithm could be deployed onto real robots easily with simple tuning parameters. However, the sensor position of the robot model built in this project might not be ideal. Experiments need to be carried out to find a better location so that the camera have a better field of view.

REFERENCES

[1] S. Thrun and M. Montemerlo, "The graph slam algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.