Map My World

Gao Peng

Abstract-Based on the positioning robot in the previous project, this paper introduces Occupancy Grid Mapping technology to implement mapping and reconstruction of unknown environments (including 2D and 3D mapping). At the same time, this paper introduces and implements Simultaneous Localization and Mapping (SLAM)-related technologies (Grid Based FastSLAM, GraphSLAM and RTAB Mapping).For completely unknown environments, the SLAM algorithm is used to implement 2D and 3D mapping of the environment, and localization of the mobile robot is implemented based on the mapping result. In order to realize this process, the mobile robot needs to add an RGB-D camera to acquire the environmental depth information and then realize The process of mapping. This article describes a variety of ways to implement this process, by comparing these methods to achieve the best SLAM algorithm for unknown environments.

1.Introduction

In the previous project, the localization process of mobile robots was established on the premise of the known environment map. However, in most real environments, the environmental information is unknown or constantly changing. Therefore, it is impossible to implement the localization process completely to solve real-world problems. For example, hoping to allow mobile robots to autonomously navigate in houses requires map information of houses, which is not provided in most houses. At the same time, even if map information can be provided, most furniture will often be moved, that is, map information will constantly change.

In this project, the robot is to map a given simulated environment and another one that is created in both 2D occupancy grid and 3D octomap using a SLAM algorithm known as RTAB-Map. The goal is to produce a great map of the environment with the least amount of passes possible and getting at least 3 loop closures.

2.Background

There are many difficulties in implementing the environment mapping, as follows:

Map size--Larger map sizes not only consume more computing resources, but also make it more difficult to map maps of mobile robots with a limited range of sensing.
Noise--In the process of map mapping, the noise will be introduced with the sensor, and it will accumulate and increase with the use of the sensor, which will have a greater impact on the final mapping result.
Perceptual ambiguity--when a robot frequently visits different places that very similar, the robot has a difficult time establishing correspondence between different locations traversed at different points in time.

Cycles--If a robot were to only move up and down a hallway, it can correct it odometry errors incrementally coming back. However, cycles make a robot return from different paths. When a cycle closes, the accumulated odometric error can be large.

In this article, we will introduce different Mapping and SLAM algorithms to solve these difficulties and problems. Including Occupancy Grid Mapping, Grid-based FastSLAM, and GraphSLAM.

2.1 Occupancy Grid Mapping

The Occupancy Grid Mapping Algorithm is different from the commonly used SLAM algorithm in that it realizes the map mapping process by using noisy and uncertain measurement data, and the premise is that the position and attitude of the mobile robot is known. The idea of this algorithm is to represent the map as a field of random.Each variable is binary and relates to the occupancy of the location it covers. Binary Bayes Filter Algorithm and Inverse Sensor Model are the core of the algorithm.

Because this algorithm needs the known mobile robot position, it is often used in postprocessing. In SLAM algorithm, the position information of the mobile robot is not required to be provided, so the occupancy grid mapping algorithm is often used for subsequent processing of SLAM algorithm results.

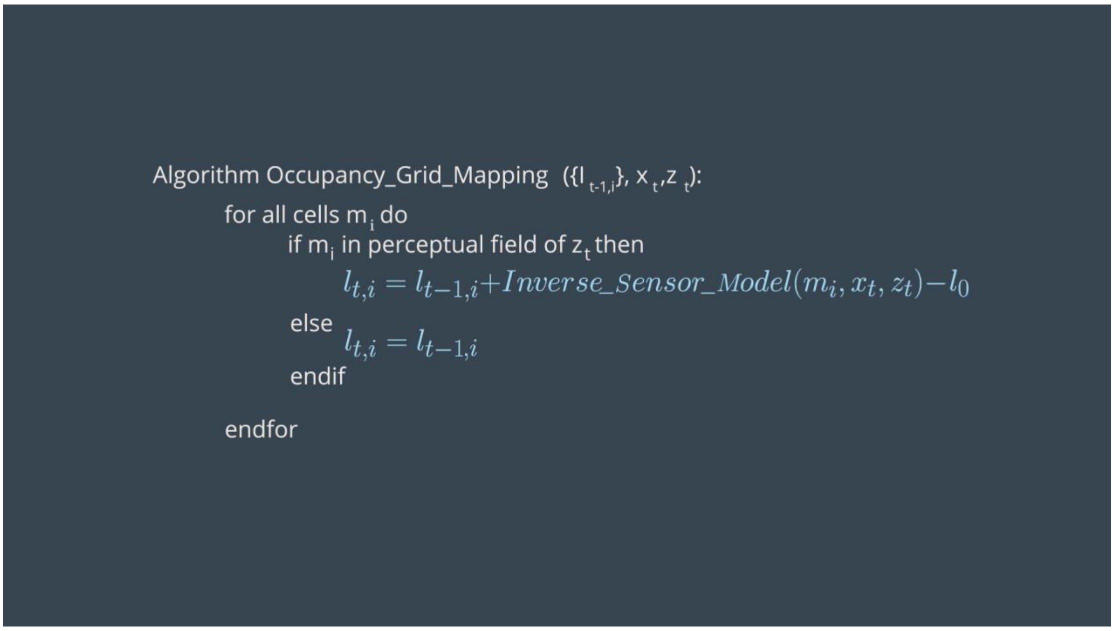

Figure 1. Occupancy Grid Mapping Algorithm

2.2 Grid-based FastSLAM

Simultaneous Localization and Mapping(SLAM) algorithm is a process that realizes the positioning of mobile robots and reconstruction of map mapping on the premise that the position and map information of the mobile robot are unknown. The challenge of this technology is that neither the position of the robot nor the map information is provided, and the accuracy of the map mapping and reconstruction depends on the accuracy of the localization of the robot. The SLAM algorithm is divided into two categories, online SLAM and full SLAM.Online SLAM estimates the current pose of the robot and map using the present measurements and controls. Meanwhile, full SLAM or offline SLAM, estimates the robot's complete trajectory and map using all measurements and control.

The idea of FastSLAM is to generate a random particle filter to approximate the posterior of each point on the mobile robot path, and it uses the low-dimensional Extended Kalman Filter (EKF) to obtain the independent features of the map. At the same time, these maps are estimated to be Gaussian. Since the FastSLAM can not only estimate the entire path of the mobile robot but also can estimate the instantaneous position information of the mobile robot, the algorithm can solve both full SLAM and online SLAM.

There are different types of FastSLAM, two of which are iterative upgrades of the FastSLAM algorithm, FastSLAM 1.0 and FastSLAM 2.0, and the other is an extension of the FastSLAM algorithm, Grid-based FastSLAM.The drawbacks of FastSLAM 1.0 and FastSLAM 2.0 are that the known landmark information is always required to be assumed, so this also weakens the algorithm's ability to generalize the random environment. However, Grid-based FastSLAM solves this problem. In addition, the algorithm not only preserves the original There are algorithms that use the advantages of particle filters and introduce an occupation grid mapping algorithm. The grid based mapping algorithm can use grid maps to map maps and locate mobile robots without assuming landmark information.

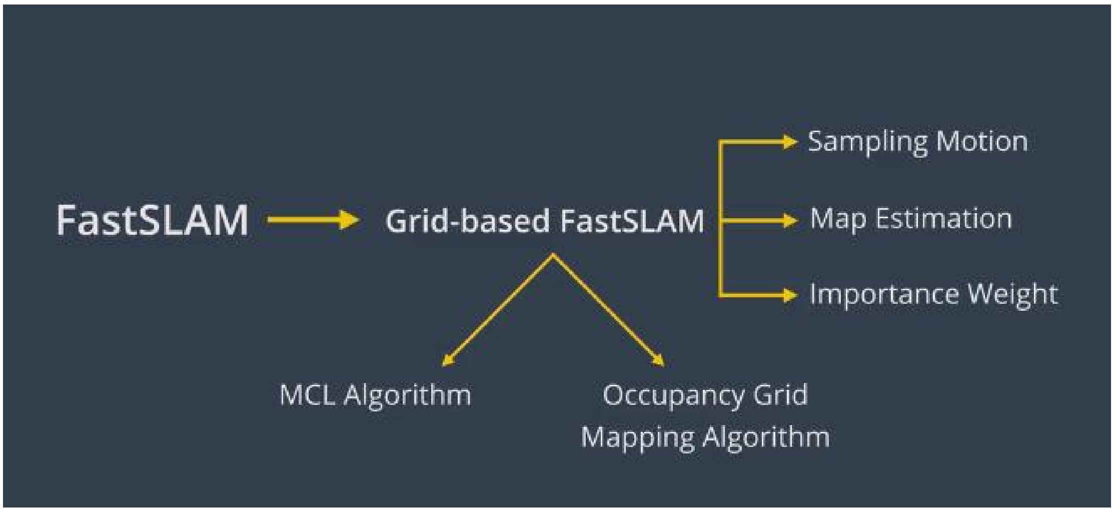

Figure 2. Grid-based FastSLAM

2.3 GraphSLAM

GraphSLAM is used to handle the full SLAM problem, which means that the algorithm can be used to map the full path of a mobile robot. Compared to FastSLAM, this algorithm has the advantage of reducing computational resource consumption and further improving the accuracy of mapping and positioning.Because FastSLAM uses particles to Estimate its pose, there's a possibility that a particle doesn't exist in the most likely position. And this disadvantage is especially obvious in large environments. However, the GraphSLAM algorithm can find a better solution to optimize this issue in a large environment.

GraphSLAM implements its Front-End process by extracting motion constraints and measurement constraints from the dataset. It looks like this:

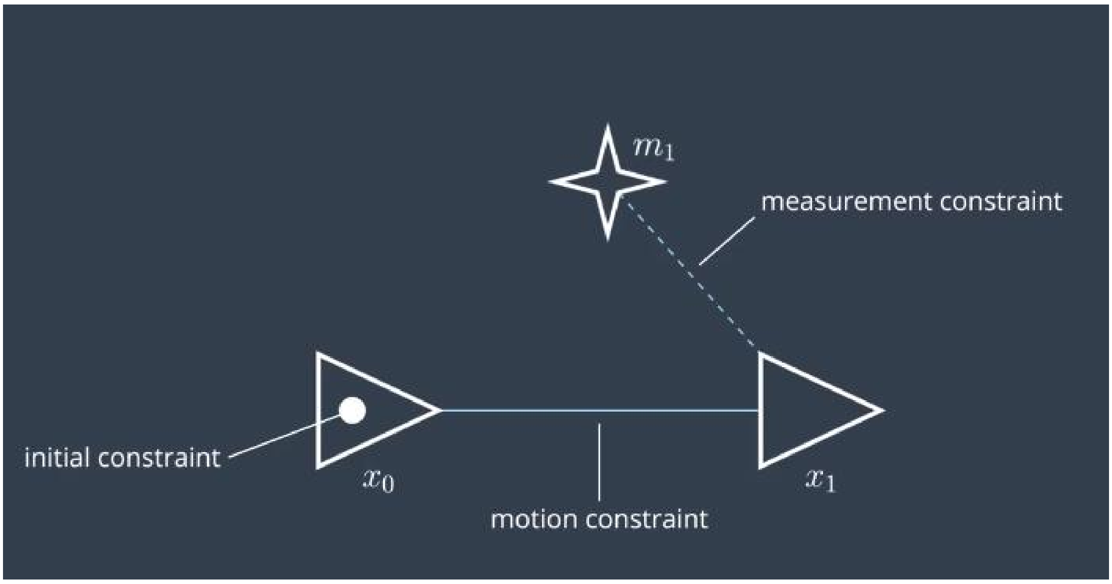

Figure 3. GraphSLAM

Then, the algorithm uses the Maximum Likelihood Estimation (MLE) algorithm to obtain the corresponding estimation information from the linear constraints. However, most of the constraints in the real environment are not linear, so Taylor's formula is used to perform the linearization process, and The gradient-descent method is used to perform the optimization process. At the same time, since the use of the Taylor formula introduces some errors, iteratively reduces the error.

RTAB-Map, real-time appearance-based mapping, is a graph-based SLAM method that uses data gathered from vision sensors to localize the robot and map the environment in real time. The algorithm establishes bag-of-words through the SURF feature detection algorithm. At the same time, it establishes loop closure through the memory management methods of STM, WM, and LSTM. This not only speeds up the use of memory, but also improves the accuracy of the 3D mapping process.

3.Scene and robot configuration

3.1 Scene

In this project, a total of two sets of scenes are used as objects that need to implement SLAM. One of them is a standard scene provided by udacity, and the other is a virtual scene created by myself through gazebo editing. As follows:

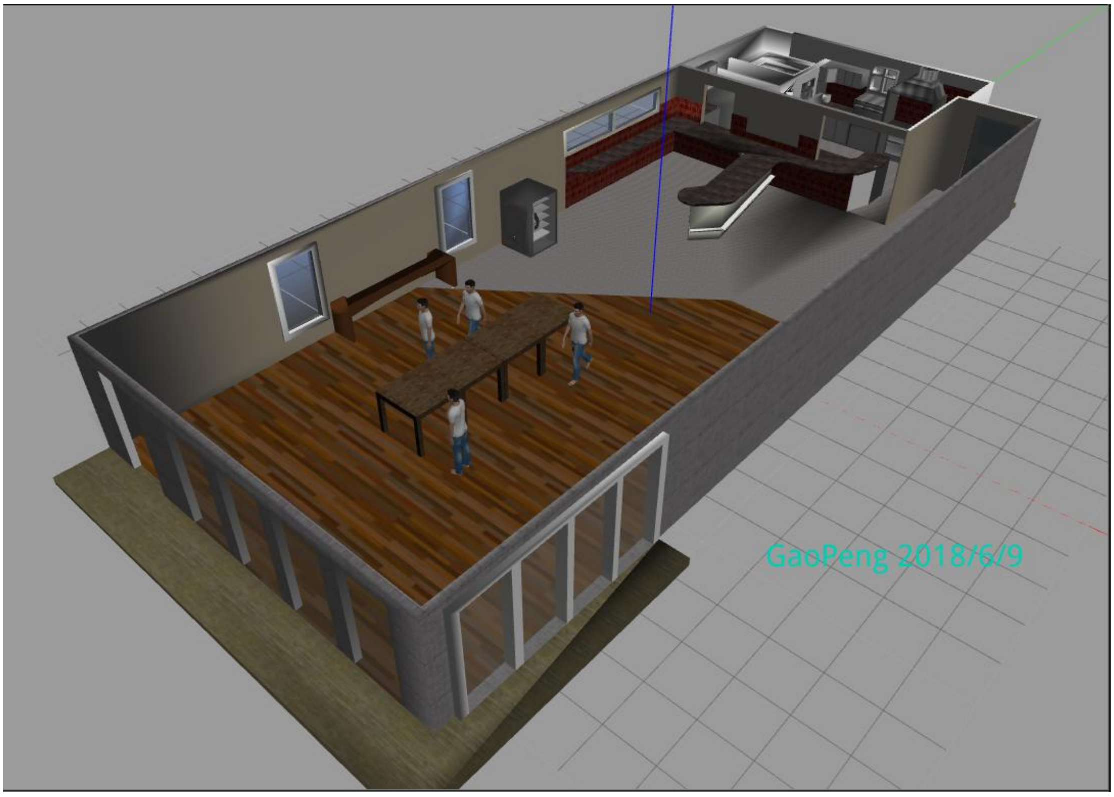

Figure 4. My_world

The scene I set up contains a coffee shop scene where people standing and walking and extra tables are added.

3.2 robot

The robot in this project uses the robot I created in the previous project. However, on the basis of the previous one, I added an RGB-D camera to the robot structure. Compared to the RGB camera, the RGB-D camera can capture the environment. Depth information. This is also the necessary information to refresh RTAB-Map.Then the camera object was lengthened and moved down to view more of the ground rather than just what’s in front of it. Meanwhile, I output the corresponding structure information through the tf command. As follows:

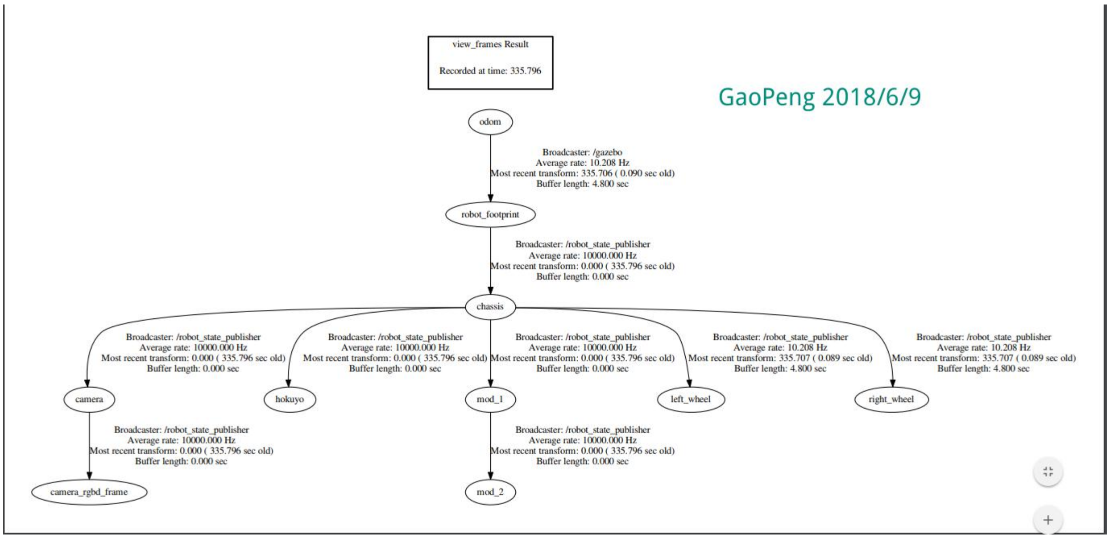

Figure 5. Frames

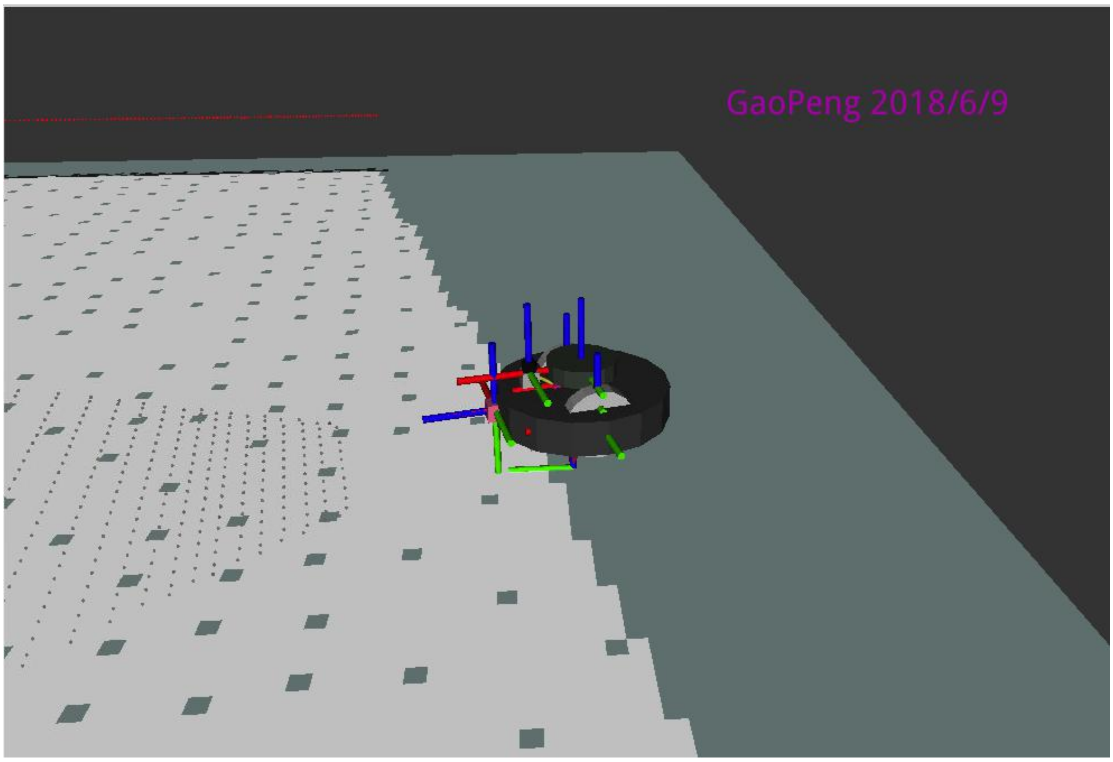

Figure 6. Construct

This is due to the depth cloud being generated along the z axis and therefore this needs to be rotated in order to visually correct. Adding this new RGB-D frame connected to the RGB-D camera allows the rotation of the point clouds without having to modify the camera's orientation. This frame is rotated by -90° on both the x axis and z axis to correctly visualize the point clouds in the project.

4.Results

4.1 udacity world

In order to map the provided virtual environment, we wrap 3 circles in two rooms in the environment. The resulting 2D map looks like this:

Figure 7. Udacity_world 2D Map

The robot performs the SLAM process in the provided virtual environment at a preset speed and angular velocity. The resulting 3D map is as follows:

LOOP1

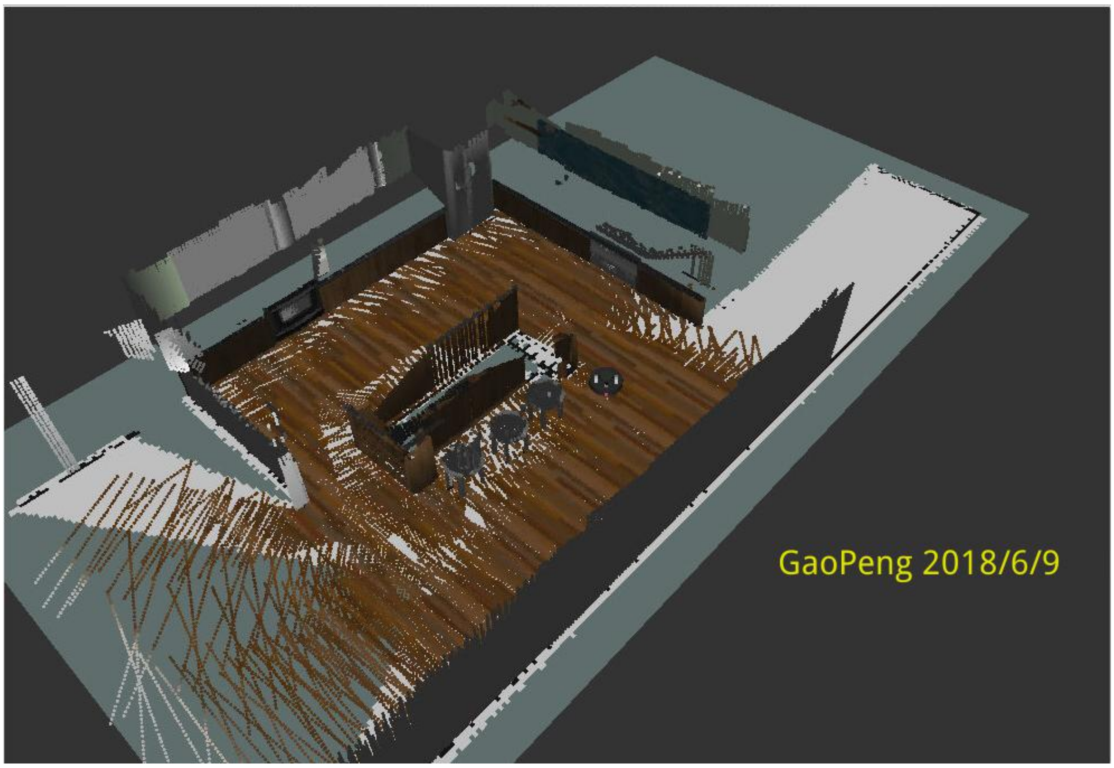

Figure 8. Udacity_world 3D Map LOOP1

LOOP2

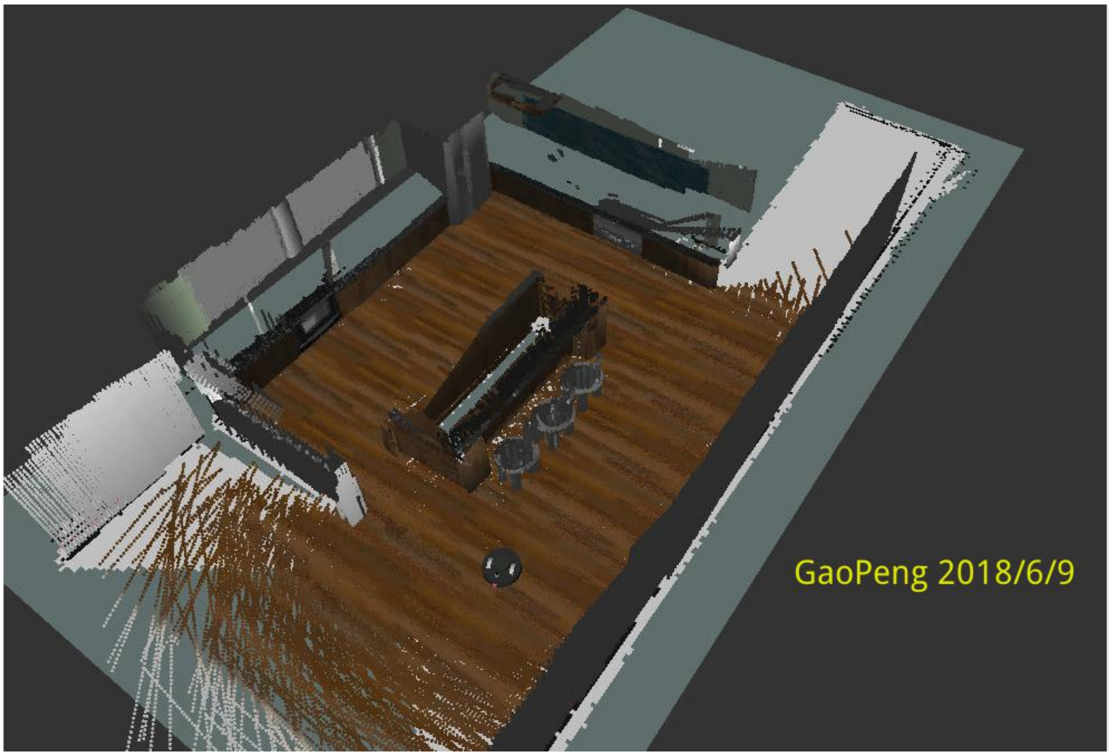

Figure 9. Udacity_world 3D Map LOOP2

LOOP_FINAL

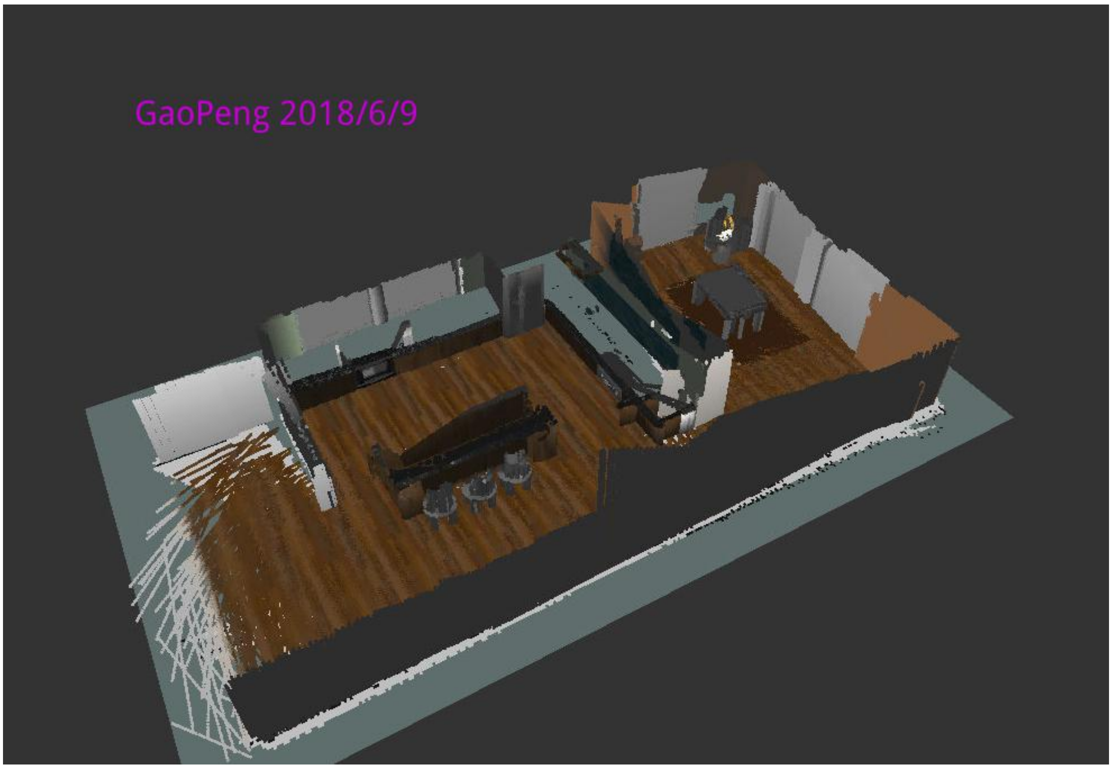

Figure 10. Udacity_world 3D Map LOOP_FINAL

The path taken by the robot during map mapping is as follows:

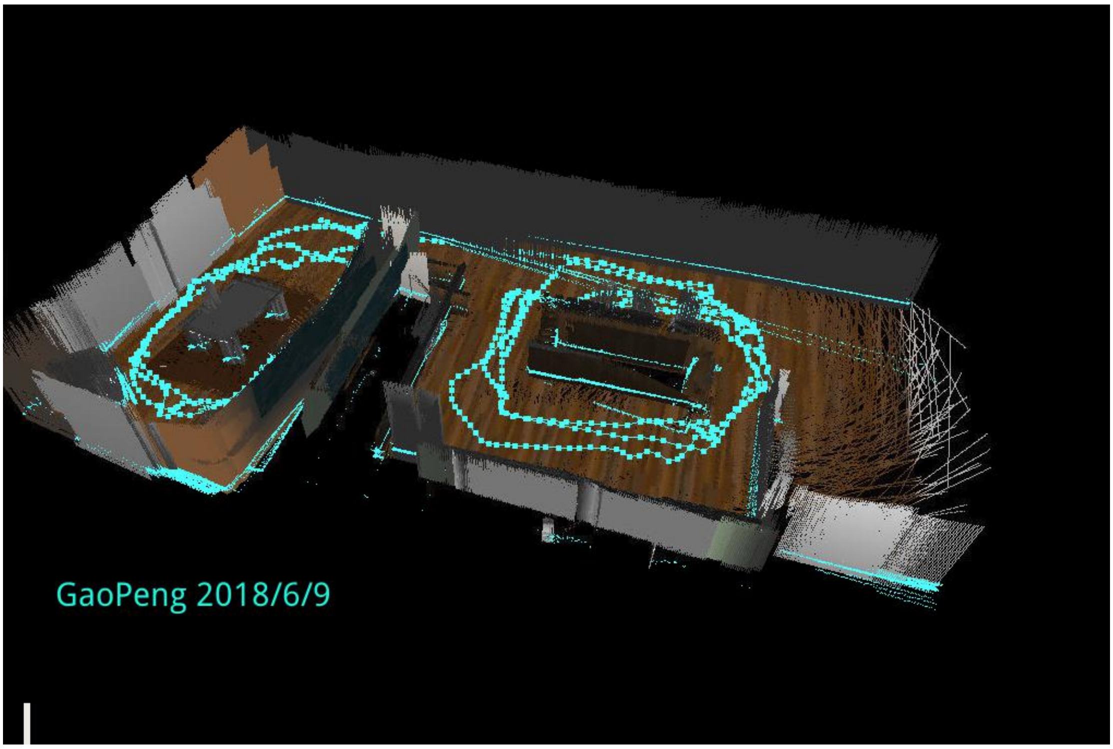

Figure 11. Udacity_world 3D Map Path

4.2 My world

Similar to the previous mapping of udacity maps, only the larger areas of the map I made were mapped. The 2D map results generated by SLAM are as follows:

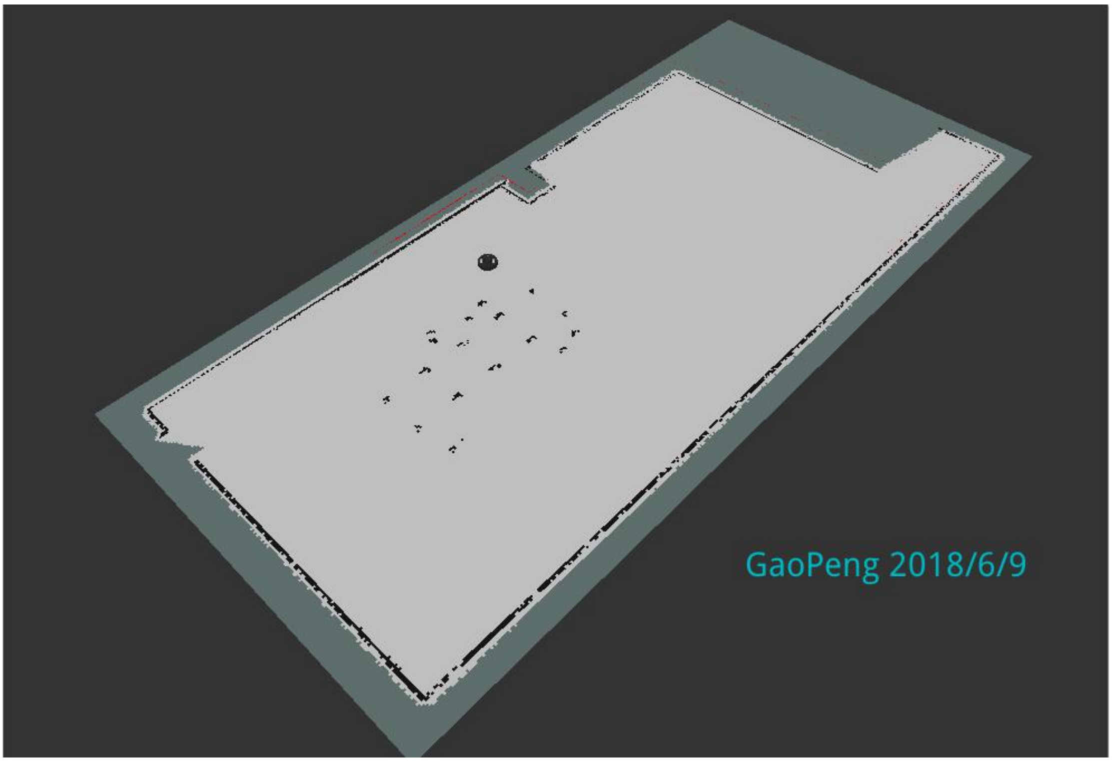

Figure 12. My_world 2D Map

The robot performs the SLAM process in the provided virtual environment at a preset speed and angular velocity. The resulting 3D map is as follows:

LOOP1

Figure 13. My_world 3D Map LOOP1

LOOP_FINAL

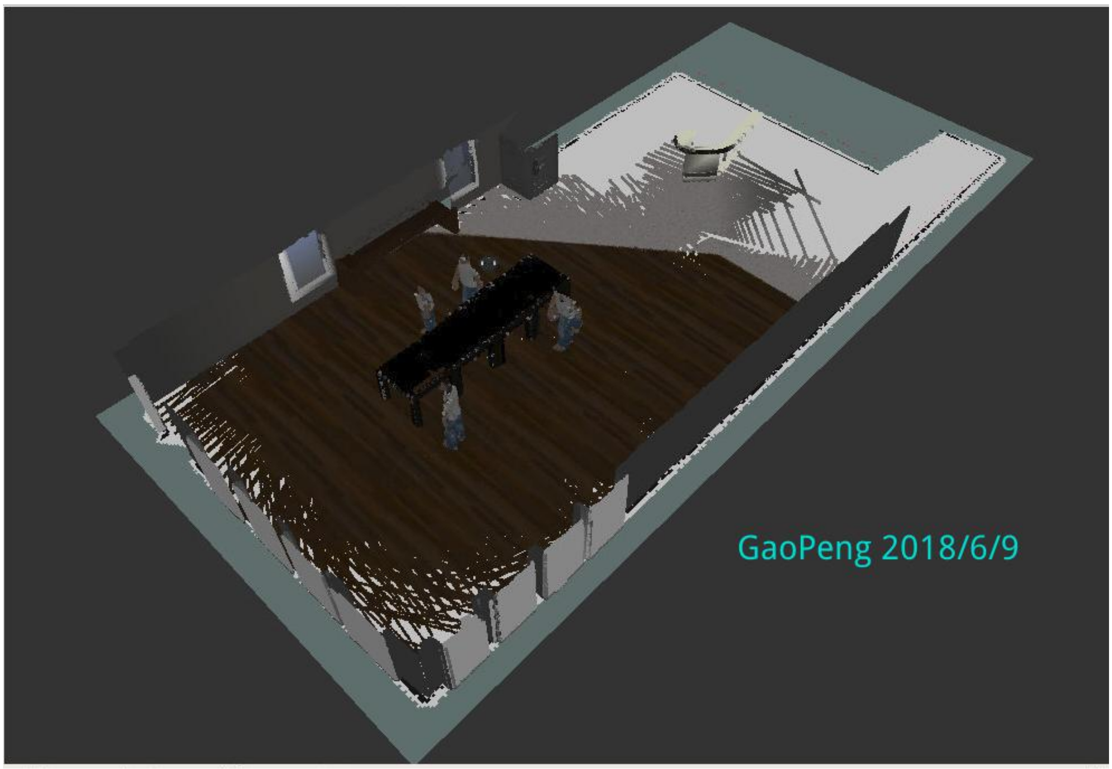

Figure 14. My_world 3D Map LOOP_FINAL

The path taken by the robot during map mapping is as follows:

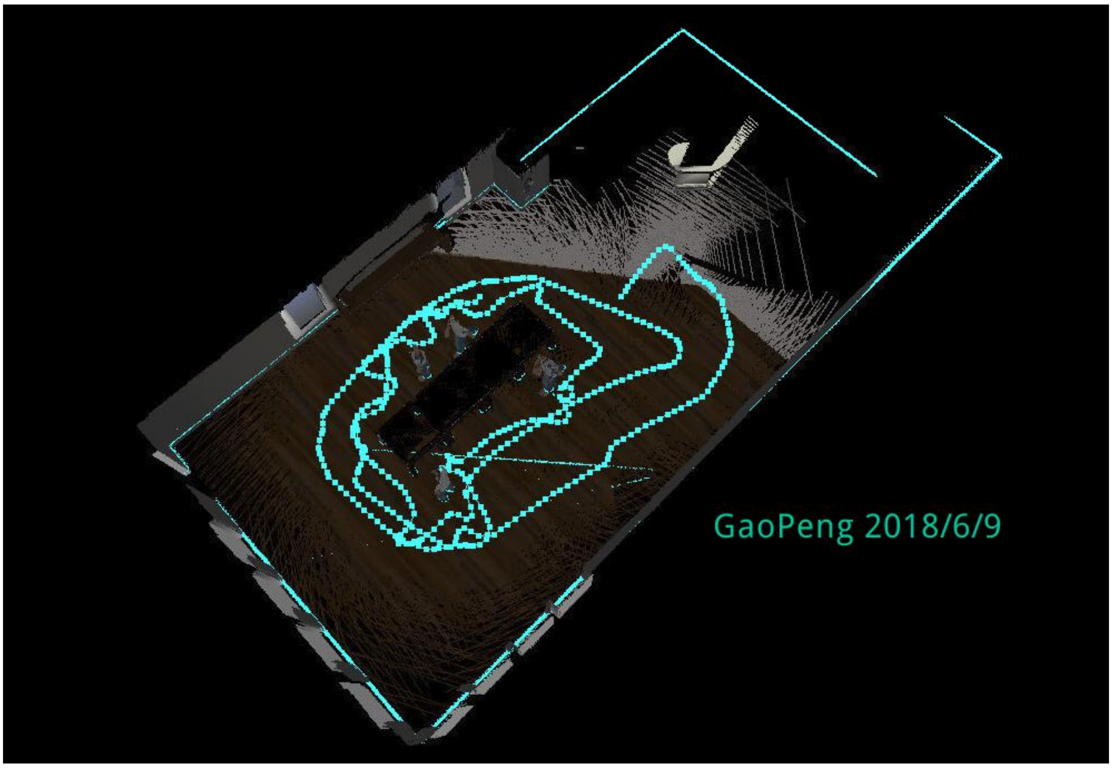

Figure 15. My_world 3D Map Path

5.Discussion

The mapping for two virtual environments is successful both in 2D and 3D. The mapping to 2D maps is relatively simple. The mapping of 2D maps can be implemented after circling 1 to 2 rounds. Subsequent 3D map mapping requires relatively more laps to implement.

The environment provided by Udacity requires that the mobile robot needs to wrap around more turns to get a more complete 3D mapping map. At the same time, the mobile robot also needs to use slower line speed and angular velocity to move. On the other hand, the virtual environment I provide is relatively It's simpler. The number of turns around the mobile robot is one less than the previous one, but a more complete 3D mapping map is also obtained. Although in the experimental process, the mobile robot moves in the same speed as the previous one, but you can speed up some speed to speed up the process of 3D map mapping.From the rtabdatabaseViewer, the kitchen and dining world had a total global loop closures of 43 (2 rooms and 3 circles) while the café world had 12(1 room and 1 circle) indicating that the café world had more features. On the surface, my provided environment is easier to implement the 3D map mapping process. However, specific analysis and experiments can be found mainly because the environment I provide only implements a 3D map mapping of a room and udacity provides a virtual environment. There are two rooms. Therefore, if you try another room 3D map mapping under the room I provide, there will be distortion and inability to locate the problem. The result is as follows:

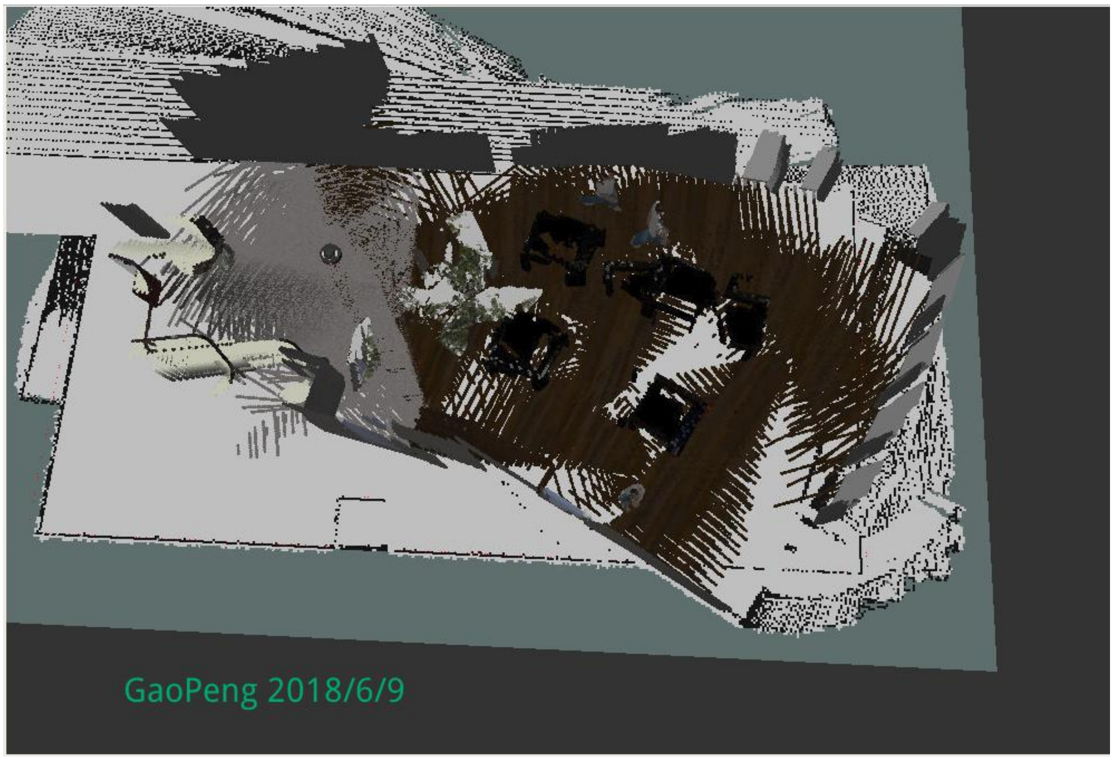

Figure 16. My_world 3D Map Mistakes

From the above figure, it can be seen that even mapping the other area of the virtual environment I provide will cause serious distortion and positioning errors. For the above situation, I found that the area I provided has different floors. Therefore, you can Modifying the number of global loop closures by modifying the number of global loop closures by modifying the different types of floor or other furniture in the environment allows the mobile robot to perform 3D map mapping. At the same time, it can accelerate 3D mapping by enhancing sensor detection and configuration.

6.Future Work

Simultaneous Localization and Mapping (SLAM) algorithm has a very important application value in real life. This technology can be applied to robotic arm control and driverless driving. At the same time, it will make important contributions to the realization of unmanned quadrotors.

In the space, the technology can help people map the environmental information of other planets in space, and also avoid people being in a dangerous environment. This technology can also fight criminal organizations in military security. We can even imagine that robots in the future will be Widely used in people's lives, the unmanned four-rotor is used in autonomous navigation to help people carry out the logistics transportation. The autonomous navigation of the unmanned vehicle will send the guests to the destination.

Therefore, SLAM technology will be a necessary condition for robot automation. Applying this technology to life will further improve people's quality of life.