# Simultaneous localization and mapping

Gabriel Garcia

**Abstract**—SLAM is a key feature for mobile robots and an important field of research in robotics. Several SLAM solutions are available based in different sensory approaches. This paper describes tests performed using a wheeled robot with a RGB-D camera and a LIDAR, aiming to 3D map a simulated environment employing the RTAB-map. Results proves the efficiency of RTAB-map in two different environments.

**Index Terms**—Robot, slam, localization, mapping, gazebo, RTAB-map.

## 1 Introduction

SIMULTANEOUS localization and mapping (SLAM) is an important field of research for robotics. It can be defined as the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. A key feature in SLAM is to recognize previously visited sites. This process is also known as loop closure detection, referring to the fact that coming back to a previously visited location makes it possible to associate this location with another one recently visited [1].

This work performs tests using an available SLAM solution in the ROS framework, the RTAB-Map. Tests were conducted in a simulated environment, using a robot embedded with an RGB-D camera and a LIDAR.

## 2 Background / Formulation

The availability of a map of the robots workspace is an important requirement for the autonomous execution of several tasks including localization, planning, and navigation. Especially for mobile robots that work in complex, dynamic environments, e.g., fulfilling transportation tasks on factory floors or in a hospital, it is important that they can quickly generate (and maintain) a 3-D map of their workspace using only onboard sensors. Manipulation robots, for example, require a detailed model of their workspace for collision-free motion planning and aerial vehicles need detailed maps for localization and navigation.

Wheeled robots usually rely on 2-D laser range scanners, which commonly provide very accurate geometric measurements of the environment at high frequencies. To compute the relative motion between observations, most state-of-the-art SLAM (and also localization only) systems use variants of the iterative closest point (ICP) algorithm, [2]. Disadvantages of ICP include the dependence on a good initial guess to avoid getting stuck in a local minimum and the lack of a measure of the overall quality of the match, [3]

Visual SLAM approaches [4], compute the robots motion and the map using cameras as sensors. Stereo cameras are commonly used to gain sparse distance information from the disparity in textured areas of the respective images. In contrast with laser-based SLAM, Visual SLAM systems typically extract sparse keypoints from the camera images. Visual feature points have the advantage of being more distinctive than typical geometric structures, which simplifies data association. Descriptors can easily be combined with different keypoint detectors.

Recently introduced RGB-D cameras such as the Microsoft Kinect or the Asus Xtion Pro Live offer a valuable alternative to laser scanners, as they provide dense, high-frequency depth information at a low price, size, and weight. The depth sensor projects structured light in the infrared spectrum, which is perceived by an infrared camera with a small baseline. As structured light sensors are sensitive to illumination, they are generally not applicable in direct sunlight. Time-of-flight cameras are less sensitive to sunlight but have lower resolutions, are more noisy, more difficult to calibrate, and much more expensive. The first scientifically published RGB-D SLAM system was proposed by [5], who used visual features in combination with Generalized-ICP to create and optimize a pose graph.

Through all available SLAM solutions, the RTAB-Map was chosen based on its speed and memory management. The objective of RTAB-map is to provide a solution independent of time and size, to achieve real-time loop closure detection for long-term operation in large environments. The idea resides in only using a certain number of locations for loop closure detection so that real-time constraint can be satisfied, while still having access to the locations of the entire map when necessary. When the number of locations in the map makes processing cycle time for finding matches greater than a real-time threshold, our approach transfers locations less likely to cause loop closure detection from the robots Working Memory (WM) to a Long-Term Memory, so that they do not take part in the detection of loop closures. However, if a loop closure is detected, neighbor locations can be retrieved and brought back into WM to be considered in loop closure detections, [1].

## 3 Model Configuration

The SpeleoRobot, shown in Figure 1, is capable of inspecting underground locations adjacent to existing mining operations and other unstructured environments. The robot is currently in test phase and can detect the threat of falling rocks or the presence of animals, for example. The robot is equipped with a camera and lighting system that affords a real-time view of the underground environment, enabling the operator to conduct simultaneous inspections or surveys for future access, thereby reducing the risks inherent to speleological activities.

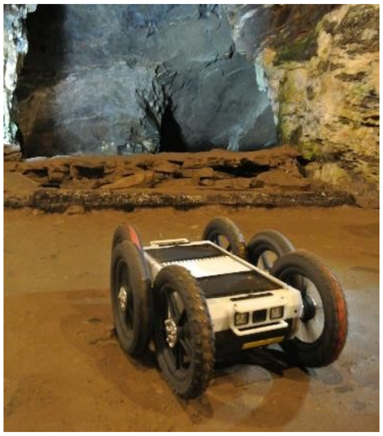

Fig. 1. Robot in field test.

For this project, a simulated version of the SpeleoRobot was used. It was added with a Hokuyo Lidar located in such a way that the wheels will not interfere in measurements. And its RGB camera was replaced by an RGB-D camera. The robot's body has a box format (54 cm x 25 cm x 12 cm), and it has six wheels of 14 cm radius. Figure 2 presents the model used to represent the SpeleoRobot, the Hokuyo is the gray and black object, and the RGB-D camera is represented by the green box. The robot uses a six wheels Skid Steer controller.

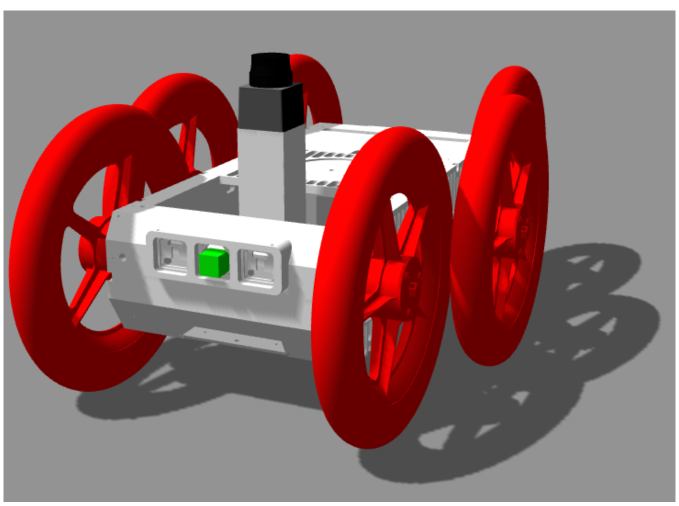

Fig. 2. SpeleoRobot model.

Inside the ROS framework, the system uses three packages:

- Espeleo-simulator: this package provides all the necessary files to correctly spawn and control the robot in Gazebo.
- Slam-project: inside this package, one can find the Gazebo worlds, a node to teleoperate the robot and launch files.
- Rtabmap-ros: this is the package responsible for the SLAM part.

## 4 World Creation

The primary objective of the robot used is to explore unstructured environments. Therefore, the world created is a simulation of a disaster in an urban environment. The world, presented in 3, has a destructed building, an airplane, other robots, an ATV, a bridge, a wall, a gazebo, and some other objects. To create the world, I started by locating a piece of asphalt to be the ground, and then I put all the following models. The models used are available in the online Gazebo dataset.

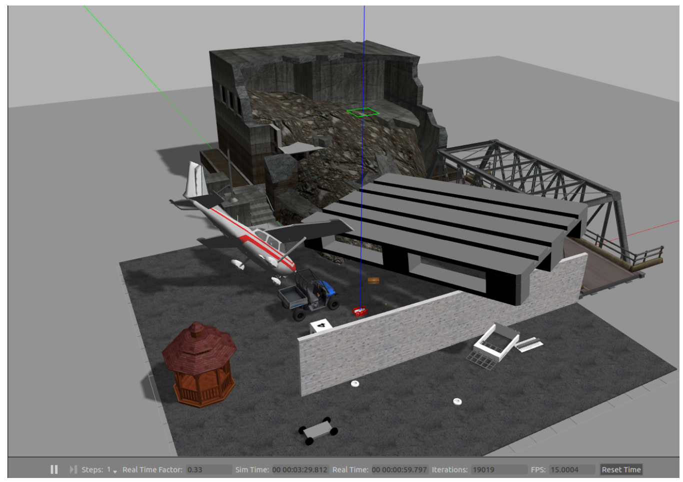

Fig. 3. Created world.

## 5 Results

We have conducted the mapping tests in a house environment, with a living room and a kitchen. I teleoperated the robot using a keyboard through an hour, trying to visit all the places that were not correctly mapped. Figure 4 shows the process of mapping, in the image top one can see the Gazebo simulator, and at the bottom, it is presented the RViz interface with the resultant map. The final 3D map can be seen in Figure 5, and the respective occupancy grid is shown in Figure 6. In Figure 7 one can see that 15 loops closures were achieved. Figure 8 shows the process of mapping in the created world.

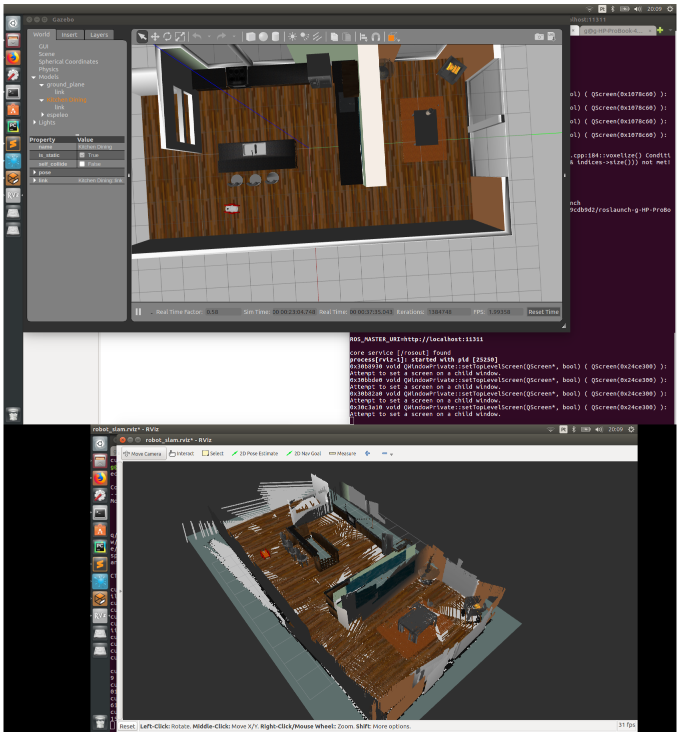

Fig. 4. Robot mapping house.

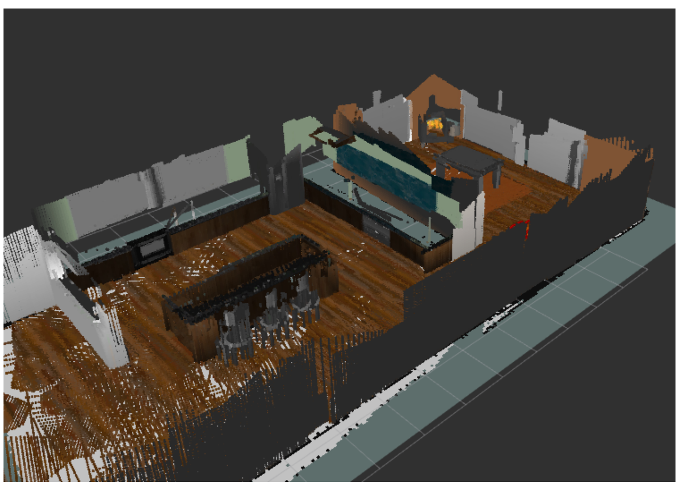

Fig. 5. Final 3D map of the simulated house.

## 6 Discussion

The SLAM process relies basically on feature extracted from its sensory source. Therefore, if the environment has different features as objects, different colors, windows and doors, the SLAM would perform well. Both worlds where the SLAM was tested present those characteristics, what allowed the construction of a reliable map and occupancy grid. A total of 15 loop closures were found, another point that proves the efficiency of RTAB-Map, whereas the robot covered a path reaching the same points different times.

## 7 Future work

The robot used was developed to map underground environments, which usually do not have easily recognizable

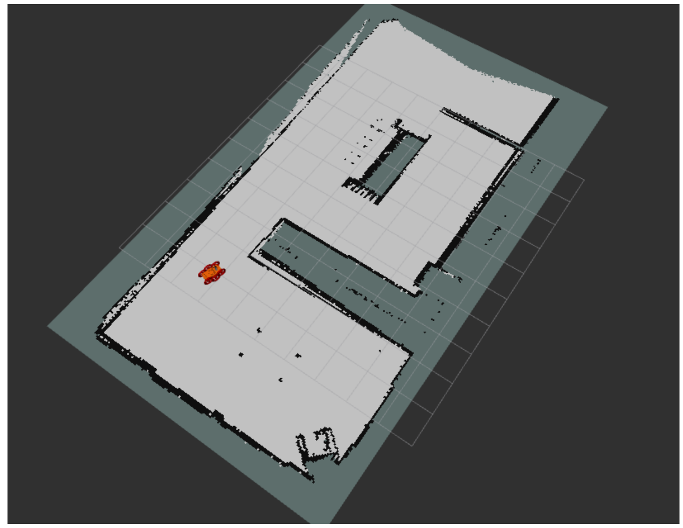

Fig. 6. Final occupancy grid of the simulated house.

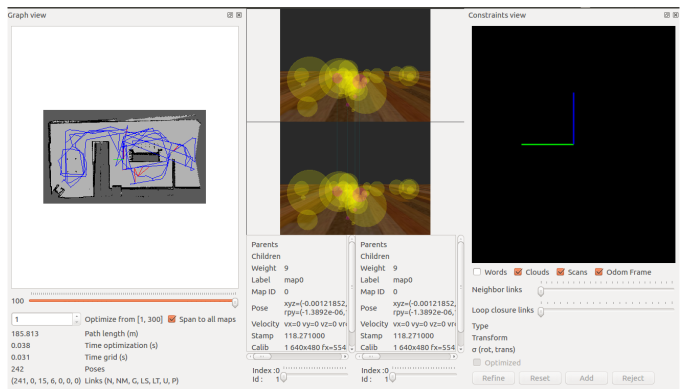

Fig. 7. Screenshot of rtabmap-databaseViewer.

features. Therefore, future work should be focused on perform tests in abandoned mines and natural caves. Through those tests, some points could be improved as the location of the sensors and the illumination system.

## References


[1] M. Labb and F. Michaud, "Memory management for real-time appearance-based loop closure detection," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1271–1276, Sept 2011.

[2] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pp. 145–152, IEEE, 2001.

[3] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-d mapping with an rgb-d camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, 2014.

[4] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *null*, p. 1403, IEEE, 2003.

[5] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.

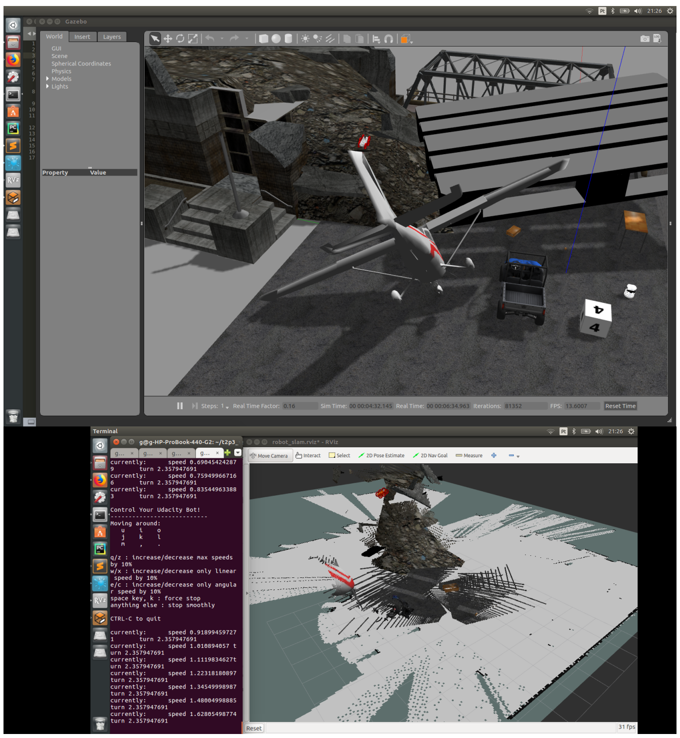

Fig. 8. Robot mapping created world.