МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Национальный исследовательский Томский государственный университет
Томский государственный университет систем управления и радиоэлектроники
Болгарская Академия наук
Академия инженерных наук им. А.М. Прохорова
Международная научно-техническая организация «Лазерная ассоциация»

ИННОВАТИКА-2020

СБОРНИК МАТЕРИАЛОВ

**XVI Международной школы-конференции студентов,
аспирантов и молодых ученых
23–25 апреля 2020 г.
г. Томск, Россия**

Под редакцией А.Н. Солдатова, С.Л. Минькова

Scientific & Technical Translations
STT
ИЗДАТЕЛЬСТВО
Томск – 2020

GOOGLE CARTOGRAPHER FOR NAVIGATION AND MAPPING (SLAM)

M. Mondal, S.I. Poslavskiy
National Research Tomsk State University
mainakme2140@gmail.com

This article examines the viability of using Google Cartographer as a platform for mapping a nearby area using a LIDAR (Light Detection and Ranging) and using the map data generated by it to estimate an accurate position and navigate using a flight controller accordingly.

Keywords: Google Cartographer, ROS, LIDAR, SLAM, Mapping, Navigation.

Google Cartographer – Cartographer is a system that provides real-time simultaneous localization and mapping (SLAM) in 2D and 3D across multiple platforms and sensor configurations.

Robot Operating System – ROS (Robot Operating System) provides libraries and tools to help software developers create robot applications. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more. ROS is licensed under an open source, BSD license.

Google Cartographer with ROS is used as the software stack with some changes to the scripts according to the hardware selection.

The hardware:

1) Velodyne VLP-16 – It's a LIDAR or a Light Detection and Ranging device. It is used to measure distance to a target by illuminating the target with pulsed laser light and measuring the reflected pulses with a sensor. Differences in laser return times and wavelengths can then be used to make digital 3-D representations of the target.

2) Pixhawk PX4 (Cube) or Ardupilot– PX4 (Cube) autopilot is an open-source autopilot system oriented toward inexpensive autonomous aircraft. It is the flight controller which is the heart of the navigation side of this project.

3) NVIDIA Jetson TX1 – The Jetson TX1 is a full-featured development platform for visual computing. It comes with a Linux environment, which includes support for many common APIs, and is supported by NVIDIA's complete development tool chain. Here, the Jetson is used as the main controller board which has our software stack (ROS and Google Cartographer) and communicates between the Pixhawk and the LIDAR.

4) Veyron 2x25a – This is the motor controller which reads PWM signals from the pixhawk and translates it to the actuator motors.

5) Motors – 2 motors are used in tank configuration, with tank tread (wheels) for efficient and multidirectional movement of the robot.

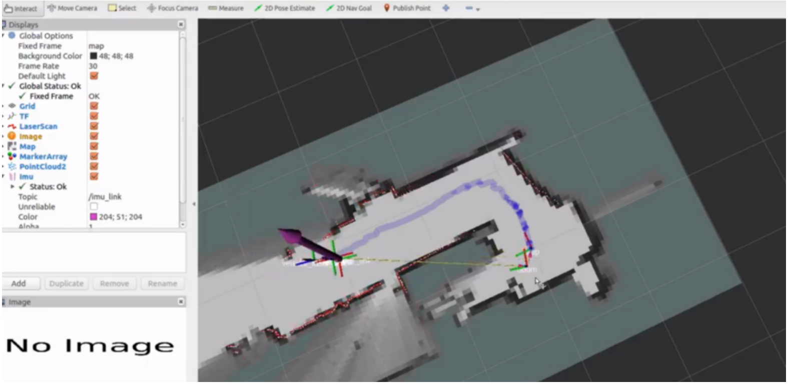
The initial step is to download and compile all the ROS Libraries in the Jetson (or any chosen board) from ROS official website. Once compiled the Google Cartographer needs to be downloaded and compiled satisfying all the dependencies.

Assuming the we already have our hardware setup ready and connected (Jetson, Pixhawk, LIDAR and Motor controllers and the motors) we can move on to testing each component but launching the rostopics for the major components. (Jetson, Pixhawk and LIDAR).

The next step is to calibrate the flight controller (pixhawk) and verify the map data (topic: /velodyne_points) from the LIDAR. Once everything is verified and calibrated, the flight controller should be set to guided mode following which it should be armed.

The guided mode helps us send commands in form of goals to the flight controller, and the controller will automatically navigate simple goals. The initial setup is complete and the robot is now ready for testing.

For testing, another computer is used to remotely access the Jetson's graphical interface (optional). Assuming that the flight controller is already armed and the points from the LIDAR are published (initial setup). Rviz (comes with ROS), a graphical interface for visualizing map data, should be launched and the all the map data from the topics should be added (/map, /point_cloud2, /pose etc.) to form a visual map and a position estimate on the 3D space in the program (Rviz) as seen in Figure 1.



Global Options
Fixed Frame map
Background Color 48; 48; 48
Frame Rate 30
Default Light ☑
✓ Global Status: Ok
✓ Fixed Frame OK
Grid ☑
TF ☑
LaserScan ☑
Image ☑
Map ☑
MarkerArray ☑
PointCloud2 ☑
Imu ☑
✓ Status: Ok
Topic /imu_link
Unreliable □
Color 204; 51; 204
Alpha 1

Figure 1. Mapping in Rviz

The final step is to give it the robot a nav goal using the ‘2D Nav Goal’ button on Rviz and checking the accuracy of the robot to navigate simple paths.

In our tests, we found that the aforementioned algorithm works well in open spaces.

Today, in the age of driverless or self-drive cars, research in the area of SLAM and navigation algorithms are the key to having a safer environment for all living beings.

References

1. ROS.org. Powering the world’s robots [Electronic resource]. – URL: http://www.ros.org/ (date: 08.04.2019).
2. ROS Willow Garage [Electronic resource]. – URL: http://www.willow garage.com/pages/software/rosplatform (date: 06.04.2018).
3. Continuous-Time SLAM–Improving Google’s Cartographer 3D Mapping [Electronic Resource]. – URL: https://www.mdpi.com/books/pdfdownload/article/789 (date: 06.04.2018).
4. Cartographer ROS. Documentation [Electronic Resource]. – URL: https://google-cartographer-ros.readthedocs.io/en/latest/ (date: 06.04.2018).
5. Hess W., Kohler D., Rapp H., Andor D. Real-Time Loop Closure in 2D LIDAR SLAM // Robotics and Automation (ICRA), 2016, IEEE International Conference on. IEEE. – 2016. – P. 1271–1278.