Techno-Science Research Journal 8 (2020) 134-139

I.T.C

Content list available at ITC

Techno-Science Research Journal

Journal Homepage: www.ric.itc.edu.kh

Techno-Science Research Journal

Simultaneous Localization and Mapping Using Intel RealSense Camera

Tongly Mork[*], Sarot Srang, Daro Van

Department of Industrial and Mechanical Engineering, Institute of Technology of Cambodia, Russian Federation Blvd., P.O. Box 86, Phnom Penh, Cambodia

Received: 29 June 2020; Accepted: 24 October 2020; Available online: xxxx

Abstract: *This paper provides Simultaneous Localization and Mapping (SLAM) for generating a 3D map of an environment. It takes the approach of graph-based SLAM with loop closure detection in the use of RGB-D (Red, Green, Blue and Depth) camera to generate the 3D map of an unknown environment. In real-time applications, localization and mapping require both accuracy and robustness. Thus, in this paper, a lighter weight Intel RealSense d435i RGB-D (with build-in IMU) camera is chosen as the sensor. The data from Intel RealSense d435i camera is computed by Jetson Nano, a single board computer which runs Robotic Operating System (ROS) and extracted to RTAB-Map node (Real-Time Appearance-Based Mapping) in ROS environment to perform the SLAM. The RTAB-Map uses the RGB data along with depth and IMU information to build the 3D map of an environment. RGB image data and depth data with IMU data are computed to get key features such as dense point cloud, and depth of RGB pixels. These features in different scenes are matched to calculate the motion of the camera. This can lead to find the odometry of the camera and construct the 3D map of the surrounding environment. Particularly, the result in this paper was conducted indoor and outdoor environment and compared to observe the difference of the quality of both environments by using Rviz simulator based on ROS to visualize the 3D map in real-time.*

Keywords: SLAM; RGB-Depth d435i camera; ROS; Rtab-map

1. INTRODUCTION

Since the 21st century, with the rapid economic and social development and the ever-changing science and technology, some problems in military, transport, traffic safety, industrial production and cruise protection need to be solved and optimized urgently (Duzhen, 2018). Within the rapid development of artificial intelligence, machine vision, automatic control, navigation and other disciplines, the enthusiasm for researching and developing robot is noticeably increasing (Das, 2018). Moreover, researches robots for exploring an unknown environment are establishing every year. For exploration, the localization and mapping in a variety of complex environments are prerequisite for the robots to accomplish tasks.

Simultaneous localization and mapping (SLAM) of the robot relies on various types of sensors, for example, LiDAR (Light Detection and Ranging, depth sensor, tracking sensor, RGB image sensor or other related sensors that are needed in SLAM algorithm (Apriaskar, 2017). Robots use SLAM algorithm by accurately digesting the information of itself and the environment to accomplish tasks more effectively and safely. The SLAM method can improve the autonomous capabilities and environmental adaptability of a robot to achieve autonomous localization and navigation in an unknown environment (Carrio, 2019).

SLAM can process data from several different types of sensors, and the powers and limits of various sensor types have been a major driver of new algorithms. Recently, many kinds of depth camera are chosen for SLAM algorithm to generate a 3D point cloud of an environment surrounding the robot, for example, Stereolabs ZED, Kinect, XtionPRO Live, Realsense camera, and so on (Magnabosco & Breckon, 2013).

Intel RealSense camera d435i is chosen in this study. This camera is very special for the SLAM algorithm because it has built-in IMU (Inertial Measurement Unit) sensor inside (Duzhen, 2018). IMU senor data are generated at the same time as image data and depth data for use for 3D mapping and navigation of the robots. Furthermore, SLAM algorithm does not only need IMU data, depth data, and RGB data (image data) that get from RealSense camera for building 3D map, but also needs a feature matcher of all these data together to perform 3D map. This feature matcher data is called RTAB-

[*] Corresponding author: Tongly Mork
E-mail: tongly_mork@oac.itc.edu.kh; Tel: +855-964292365

Map which runs in ROS (Robotic Operating System) environment.

The emergence of modern RGB-D sensors had a significant impact in many application fields, including robotics, augmented reality (AR) and 3D scanning for 3D scene reconstruction. Therefore, the purpose of this paper is to investigate the following research directions: using SLAM algorithm for an efficient exploration, performing simultaneous localization and mapping for real-time 3D reconstruction map. Also, this paper mainly designed and made a SLAM for 3D mapping based on RGB-D camera. The principle and framework of the SLAM algorithm are introduced. The Intel RealSense D435i camera is used as a RGB-D camera sensor to implement the SLAM algorithm with Nvidia Jetson Nano as an embedded system.

2. METHODOLOGY

2.1. Configuration of 3D mapping

In this paper, the configuration of the 3D mapping system is proposed based on ROS (Robotic Operating System). The main components of the 3D mapping process are divided into two parts, mapping part: SLAM algorithm using RTAB-Map package in ROS run on Ubuntu 18.04 LTS with Jetson Nano, a single board computer, and sensor part: RGB-D camera from Intel RealSense camera d435i.

2.2. Intel RealSense camera

The data from the RealSense camera is very important for using for the next section. RealSense cameras project infrared light onto a scene and detect the reflection to measure depth as shown in Fig. 1.

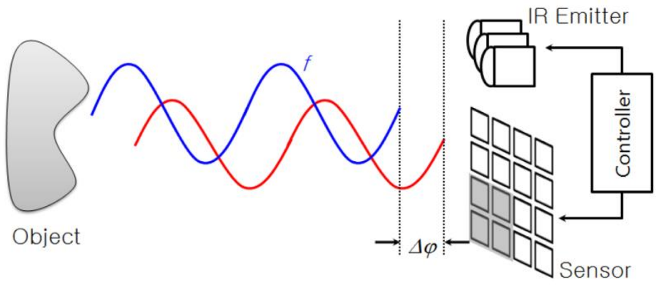


Fig. 1. The principle of depth measurement of RealSense camera

The formula of the depth information from RealSense camera to the object is retrieved in (Miles Hansard, 2012).

$$Depth = \frac{c}{2}\frac{\Delta\varphi}{2\pi f} \qquad \text{(Eq. 1.)}$$

where: c is the speed of light (m/s)
$\Delta\varphi$: phase different (rad)
f: IR light emitter signal frequency (rad/s)

There are two principle sensors to measure the depth, and place in the specific space with a small distance apart, namely, IR emitter and receiver sensor. The camera takes the two images from these two sensors and compares them. Since the distance between the sensors is known, these comparisons give depth information given in Eq. 1. The depth of the object from the camera is represented depth colour of the image as shown in Fig. 2. Moreover, from RealSense camera d435i, we can get raw data of depth data, RGB data and IMU data. These data will be used in the SLAM algorithm.

The data from the RealSense camera for using in section 2.6 need to be transfere from the camera to the computation processor. For device communication, the Librealsense is used. Librealsense is a cross-platform ROS package library for using in Robotic Operating System (ROS) environment. This can give the raw data form camera for computation in the specific computational system. This effort was initiated to better support researchers, creative coders, and computer vision developers in domains such as robotic navigation, object recognition, virtual reality, and the internet of things, Unman Aerial Vehicle (UAV) and so on.

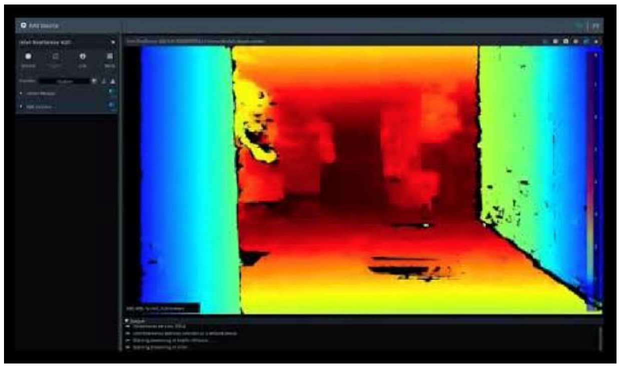


Fig. 2. Depth colour data from RealSense camera

2.3. Robotic Operating System

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms. When running rosnodes, those nodes perform computations or processing data, and they may require data from other nodes.

Furthermore, ROS is an open-source software framework which is developed in the use of robot software application. It provides functions like an operating system in a computer, such as hardware layer abstraction, low-level device control, implementation of commonly-used functionality, messagepassing between processes and packages management from one node to other nodes in the specific topic as shown in Fig. 3. ROS also provides tools and libraries that can be reused in robotics research and development, particularly SLAM in this study. One of the advantages of using ROS is that it can run code across multiple platforms. There are important terms in the architecture system of ROS, rosnode, rostopic and ros master.

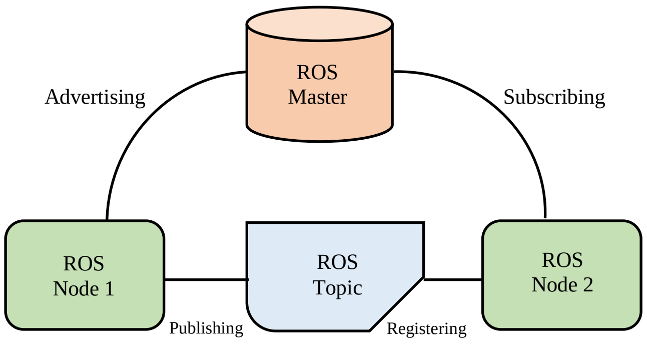


Fig. 3. Architecture of ROS

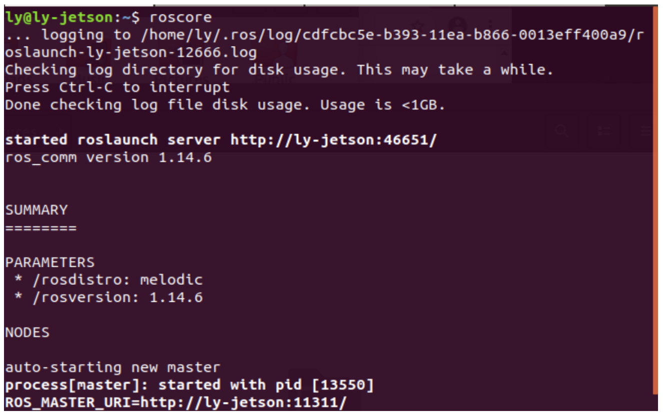
The ROS Master provides names and registration services to the rest of the nodes in the ROS system. It tracks publishers and subscribers to topics as well as services. The role of the Master is to enable individual ROS nodes to locate one another. Once these nodes have located each other, they can communicate with each other peer-to-peer.

2.4. Intel RealSense camera and ROS communication

There are two prerequisites needed in RealSense-ROS communication, namely ROS distribution for Ubuntu operating system and librealsense_ros which is the RealSense camera library for running in ROS environment. The two prerequisites are important parts for achieving the RealSense camera to communicate in ROS.

Thereafter, the ROS master needs to be launched. ROS master or roscore is a collection of nodes and programs that are pre-requisites of a ROS-based system. The roscore is running for communication between ros node and other nodes in specific of rostopics. It is launched using the roscore command as shown in Fig. 4. Next, ROS nodes of RealSense camera are being launched. Therefore, ROS environment for publishing from RealSense camera node is required and shown in Fig. 5. RealSense camera publishes nodes such as IMU node, RGB node and depth node. For monitoring the result, rviz ROS tool must be launched as shown in Fig. 6.

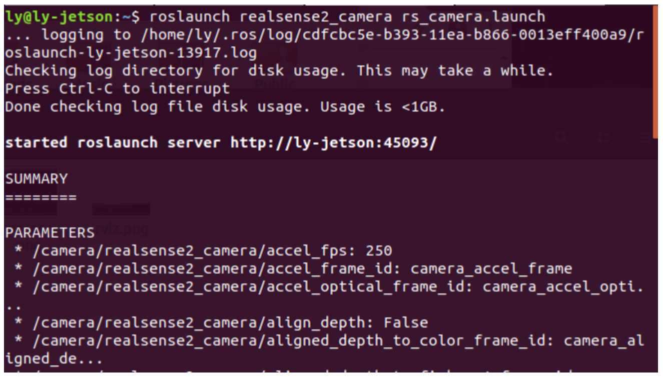
Terminal_1: Run ros master
```
$roscore
```



```
ly@ly-jetson:~$ roscore
... logging to /home/ly/.ros/log/cdfcbc5e-b393-11ea-b866-0013eff400a9/r
oslaunch-ly-jetson-12666.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ly-jetson:46651/
ros_comm version 1.14.6


SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.6

NODES

auto-starting new master
process[master]: started with pid [13550]
ROS_MASTER_URI=http://ly-jetson:11311/
```

Fig. 4. ROS master terminal

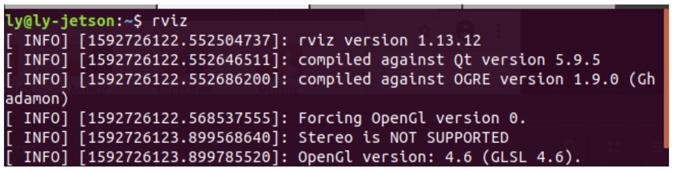
Terminal_2: Launch realsense camera node
```
$cd catkin_ws
$source devel/setup.bash
$cd src/realsense-ros/realsense2_camera
$roslaunch realsense2_camera
rs_camera.launch
```



```
ly@ly-jetson:~$ roslaunch realsense2_camera rs_camera.launch
... logging to /home/ly/.ros/log/cdfcbc5e-b393-11ea-b866-0013eff400a9/r
oslaunch-ly-jetson-13917.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ly-jetson:45093/

SUMMARY
========

PARAMETERS
 * /camera/realsense2_camera/accel_fps: 250
 * /camera/realsense2_camera/accel_frame_id: camera_accel_frame
 * /camera/realsense2_camera/accel_optical_frame_id: camera_accel_opti.
..
 * /camera/realsense2_camera/align_depth: False
 * /camera/realsense2_camera/aligned_depth_to_color_frame_id: camera_al
igned_de...
```

Fig. 5. launching RealSense camera node terminal

Terminal_3: rviz for monitoring
```
$cd catkin_ws
$source devel/setup.bash
$rviz
```



```
ly@ly-jetson:~$ rviz
[ INFO] [1592726122.552504737]: rviz version 1.13.12
[ INFO] [1592726122.552646511]: compiled against Qt version 5.9.5
[ INFO] [1592726122.552686200]: compiled against OGRE version 1.9.0 (Gh
adamon)
[ INFO] [1592726122.568537555]: Forcing OpenGl version 0.
[ INFO] [1592726123.899568640]: Stereo is NOT SUPPORTED
[ INFO] [1592726123.899785520]: OpenGl version: 4.6 (GLSL 4.6).
```

Fig. 6. rviz terminal

After realsense_camera node is launched in ROS, there are many topics published by RealSense camera. Actually, there is a 3D visualization tool for ROS-rviz that allows visualizing the 3D image as shown in Fig. 7. This does not have the ability to make point cloud reconstruction or 3D mapping, but it can configure RealSense camera d435i to communicate to the ROS environment. The next section deals with 3D mapping algorithm by using the data from the communication of this section.

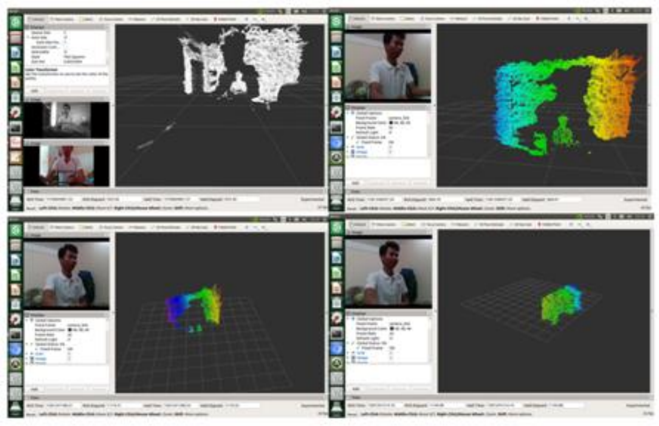


Fig. 7. Depth output of RealSense camera D435i visualize in Rviz

2.5. RTAB-Map

RTAB-Map (Real-Time Appearance-Based Mapping) is a ROS package that uses the data received from the depth camera to perform Graph-Based SLAM, generating a dense, colour point cloud, and odometry of the camera. Moreover, Jetson Nano (single-board computer) which is running ROS for computing the data from the camera to perform SLAM base on RTAB-Map node as shown in Fig. 8.

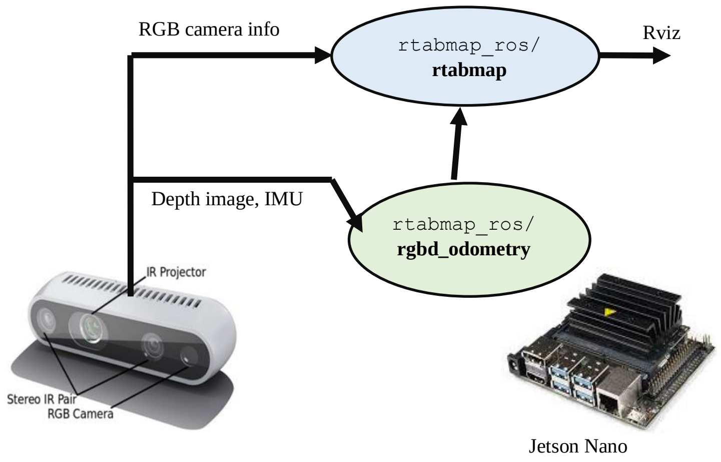


Fig. 8. rtabmap ROS node process

When running the rtabmap node on ROS, rtabmap node subscribes to the RealSense camera topic via ROS master that included RGB image data, IMU data and depth data or odometry data published at a different rate for using in SLAM algorithm in section 2.6. Furthermore, Rtabmap can make sure that images from the RealSense camera are correctly synchronized together (François, 2019). If the camera data are published on the network, the data format must be also configured to synchronize RGB images, IMU data, depth data before sending all these data to other node for SLAM algorithm.

2.6. SLAM

The SLAM algorithms can be mainly divided into three parts: raw data from RealSense camera (RealSense data info), 3D feature matching raw data (Rtab-map), and 3D map performance as shown in Fig. 9.

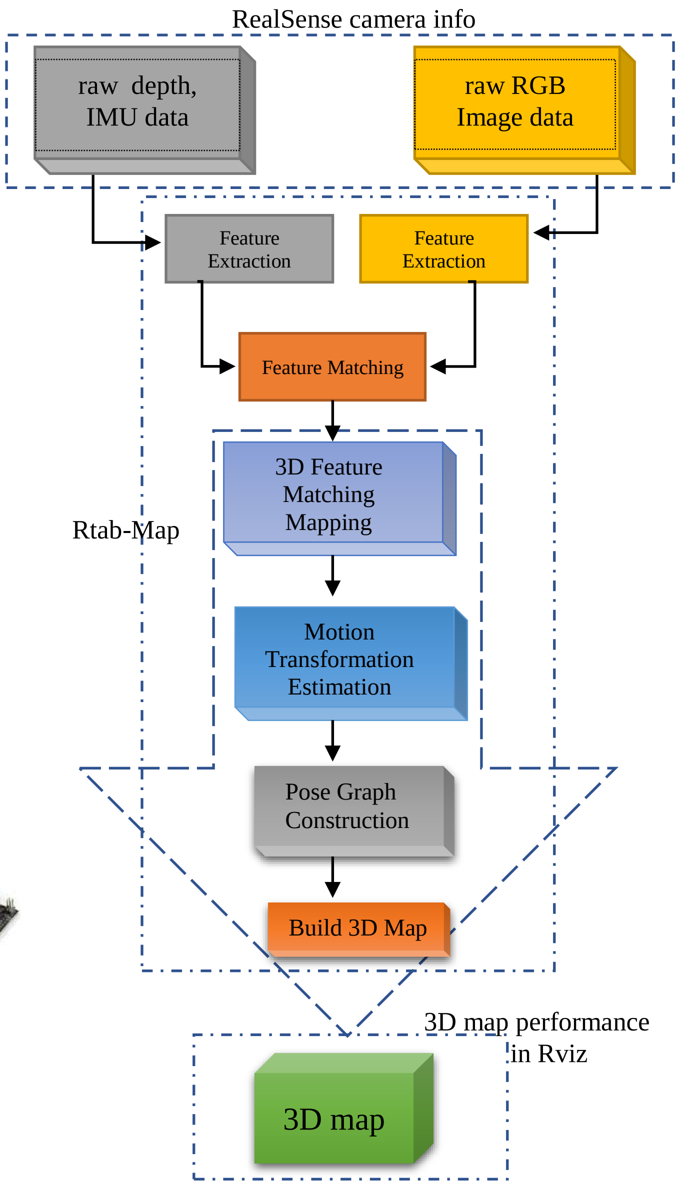


Fig. 9. Block diagram of SLAM using RealSense camera

The raw data from RealSense camera is performed in feature detection and descriptor extraction into RGB images, depth, and IMU feature, or points descriptors of the adjacent frames are matched together, forming a 3D feature. When the motion of the camera is detected, the 3D feature matching is performed according to the matching results. Then, the motion transformation is estimated and optimized for pose graph construction. The SLAM algorithm constructs a pose map according to the results of the feature matching data, then performs closed-loop detection and optimization of the pose map. Finally, the 3D mapping is generated from reconstructing the environment that the RealSense camera is pointing at.

3. RESULTS AND DISCUSSION

The result of the generated 3D map from reconstructing the real-time mapping in this paper was obtained by conducting experiment inside and outside of a building with mapping time around 15 minutes. Fig. 10. and Fig. 11. show the indoor 3D mapping. The 3D feature matching of RGB data and depth data point cloud are better results than outdoor mapping shown in Fig. 12. The 3D mapping indoor environment is mapped with path planing continuously for 3D reconstruction. However, the 3D reconstruction in the large area of the outdoor environment is not feasible due to lighting conditions and low depth range.

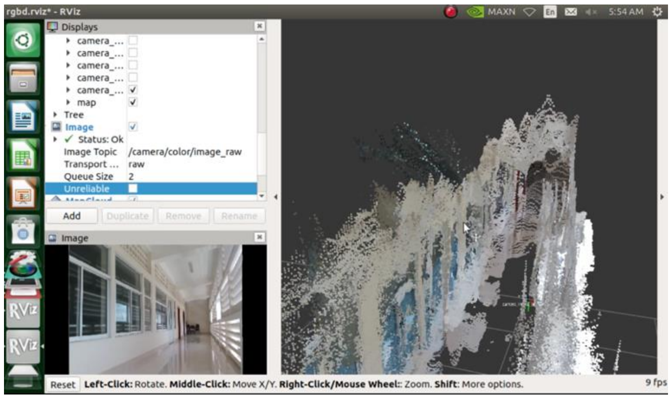

rgbd.rviz* - RViz

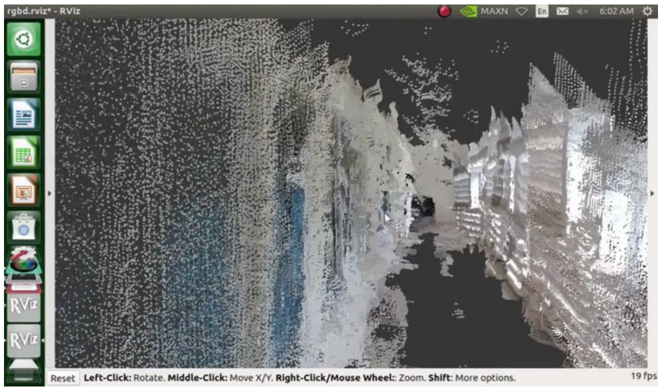
Image Topic /camera/color/image_raw
Transport ... raw
Queue Size 2

Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.

9 fps

Fig. 10. Result of 3D mapping (Indoor)

Fig. 11. is the different perspectives of the indoor 3D map that build using RealSense d435i. As we can see from the 3D map, the wall, window, hollow space and other things can be easily seen in the 3D map that was reconstructed by thousands of point cloud matching with each RGB pixel. However, the clear mirror cannot be found as it has absorbed much-infrared ray, therefore the camera cannot get its depth information. Also, the reflexed floor cannot be mapped due to the reflexing surface will cause the infrared ray not to return to the IR pair of the camera.


rgbd.rviz* - RViz

Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.

19 fps

Fig. 11. Indoor 3D mapping in the detailed point cloud

The result of 3D mapping that is shown in Fig. 12. was mapped in outdoor environment. The appearance of the outdoor 3D mapping generated by using Intel RealSense d435i camera can be seen clearly in the detail, and the environment can be recongnized. The point cloud and RGB texturing simultaneously match to generate how the things look like, for example, the plants, the grass, concrete, and so on. However, the 3D reconstructs of point cloud did not match well due to light condition and limit of mapping rang of the Intel RealSense d435i camera.

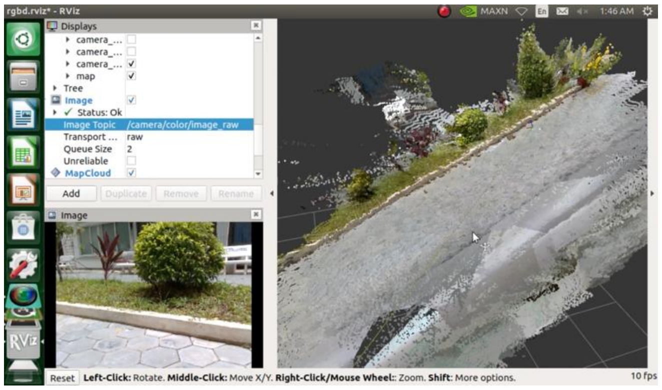

rgbd.rviz* - RViz

Image Topic /camera/color/image_raw
Transport ... raw
Queue Size 2

Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.

10 fps

Fig. 12. Result of 3D mapping (Outdoor)

Moreover, to map these environments both indoor and outdoor, the RealSense camera needs to move smoothly steadily to get better 3D mapping as shown in Fig. 10. and Fig. 12. However, the built-in IMU of RealSense d435i camera can only keep track for a very short time. Moving or turning too quickly would break the sequence of successful point cloud matches, and cause the process losing track. The system could recover immediately if the camera stops moving, but if not, the longer the time passed, the farther away it will drift from the correct position, and this will affect 3D mapping with large error.

4. CONCLUSIONS

In this paper, localization and mapping are based on RTAB-Map for generating a 3D map in an environment. SLAM algorithm uses RealSense d435i camera as 3D map sensor which has RGB data, IMU data, and depth information for publishing in ROS. The rtabmap is used for matching all data from RealSense camera to build the 3D map and gets updated from time to time upon receiving new data. The SLAM algorithm extracts the features from the data of depth camera and computes the odometry information through the features matching of the adjacent image from rtabmap_ros node, and build an environment map based on the location and posture of the RealSense camera. It determine the location and posture from an unknown environment to reconstruct the 3D map. The results of the 3D mapping of both indoor and outdoor are generated in Rviz of ROS environment with the quality depend on the limited of depth camera range and light condition of each environment.

As mentioned in section 3, the built-in IMU of RealSense d435i camera can only keep track for a very short time. Moving or turning too quickly would break the sequence of successful point cloud matches and result in losing track. Also, the 3D mapping that is generated from matching point cloud and RGB texturing take a lot of computational of the single-board computer Jetson Nano's processor. So, for future work, the computational of 3D mapping will be reduced by lowering the the quality of depth point cloud and lowering the resolution of the RGB texturing or using grayscale instead. Expectedly, the computational task would be more efficient for 3D mapping.

REFERENCES

Apriaskar, B. R. (2017). Simulation of Simultaneous Localization and Mapping Using Hexacopter and RGBD Camera. *Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)*.

Duzhen, F. (2018). VSLAM and Navigation System of Unmanned Ground Vehicle Based on RGB-D camera. *Faculty of Science and Engineering*.

ShaneLoretz. (2020). *Ubuntu install of ROS.* Retrieved on 23 May 2020 from http://wiki.ros.org/Installation/Ubuntu

Carrio, A. (2019). Onboard Detection and Localization of Drone Using Depth Maps. *IEEE*.

François, M. (2019). *Loop closure detection approach using RTAB-Map.* Retrieved on 17 April 2019 from https://introlab.3it.usherbrooke.ca/mediawiki-introlab/index.php/RTAB-Map

Das, S. (2018). Simultaneous Localization and Mapping (SLAM) using RTAB-MAP. *computer science bibliography, DBLP: /corr/abs-1809-02989*.

Doronhi. (2020). *Github.com.* Retrieved on 23 June 2020 from ROS Wrapper for Intel® RealSense™ Devices: https://github.com/IntelRealSense/realsense-ros

Magnabosco, M., & Breckon, T. (2013). Cross-Spectral Visual Simultaneous Localization And Mapping (SLAM) with Sensor Handover. *Robotics and Autonomous Systems. 63 (2): 195–208. doi:10.1016/j.robot.2012.09.023*.

Miles Hansard, S. L. (2012). Time of Flight Cameras: Principles, Methods, and Applications. *SpringerBriefs in Computer Science, ISBN 978-1-4471-4658- 2. 10.1007/978-1-4471-4658-2 . hal-00725654*, pp.95,.

Intel. (2020). *Intel® RealSense™ Technology*. Retrieved on 25 May 2020 from https://www.intelrealsense.com/stereo-depth/