Assignment #4:
Running SLAM with RGB-D sensor data

Release: Jan 24, 2017 ***Due: Feb 2 (depends on signup)***
Checkoff Location: John Crerar Library - Kathleen A. Zar Room

We will setup the space at John Crerar Library - Kathleen A. Zar Room for driving the car. In this time, we only allow one connection to the car at the same time. You will need to sign for a time slot to lab to connect to the Magoo system debugging your design, and you can only stay in the room during your debug time slot, and MUST leave the room before the end of your time slot.

This lab has two parts. The first part you can do on your own with your laptop. The second part requires a Magoo car. You must sign up for exactly one 1-hour debug/check-off time slot.
http://bit.ly/2k9TBS9

Please make sure you finish the Part 1 before you come to debug lab hours.

Goals for this lab:
- SLAM (Simultaneous Localization and Mapping) package installation
- Use SLAM to build a good 3D model

Part 1:
- Rtabmap_ros installation
- Running SLAM with recorded sensor data

Part 2:
- Construct 3D model with real-time sensor data

Part 1: Rtabmap_ros Installation and Running SLAM with Recorded Sensor Data

Rtabmap_ros installation:

Please follow the instruction to install rtabmap_ros package in your ubuntu system. We are using ROS indigo distribution, and able to install the binary via apt-get.
https://github.com/introlab/rtabmap_ros#installation

When launching rtabmap_ros's nodes, if you have the error while loading shared libraries, add the next line at the end of your ~/.bashrc to fix it:

$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/ros/kinetic/lib/x86_64-linux-gnu

Running SLAM with recorded sensor data

Please download lab4 materials: http://bit.ly/2jL1SMI
Extract the tarball
$ tar -zxvf lab4.tgz

You will get 4 files, keyboard.py, lab4.launch, lab4_tutorial.launch, and lab4_tutorial_slam.bag.

- lab4_tutorial.launch: rtabmap launch file used for part 1 of the lab, running SLAM with log data.
- lab4_tutorial_slam.bag: ros sensor log data used for part 1 of the lab.
- keyboard.py: keyboard controller to control the car for part 2.
- lab4.launch: rtabmap launch file used for part 2, running SLAM with real-time data.

To running ROS system at your local ubuntu, first you need to change .bashrc to set your ubuntu as the ROS Master node. Edit ~/.bashrc to set ROS_MASTER_URI and ROS_IP to localhost as below.

- export ROS_MASTER_URI=http://localhost:11311
- export ROS_IP=localhost

And then source the rc file.
$ source ~/.bashrc

Use the following command to check if your environment settings correctly.
ROS_MASTER_URI and ROS_IP should be localhost.

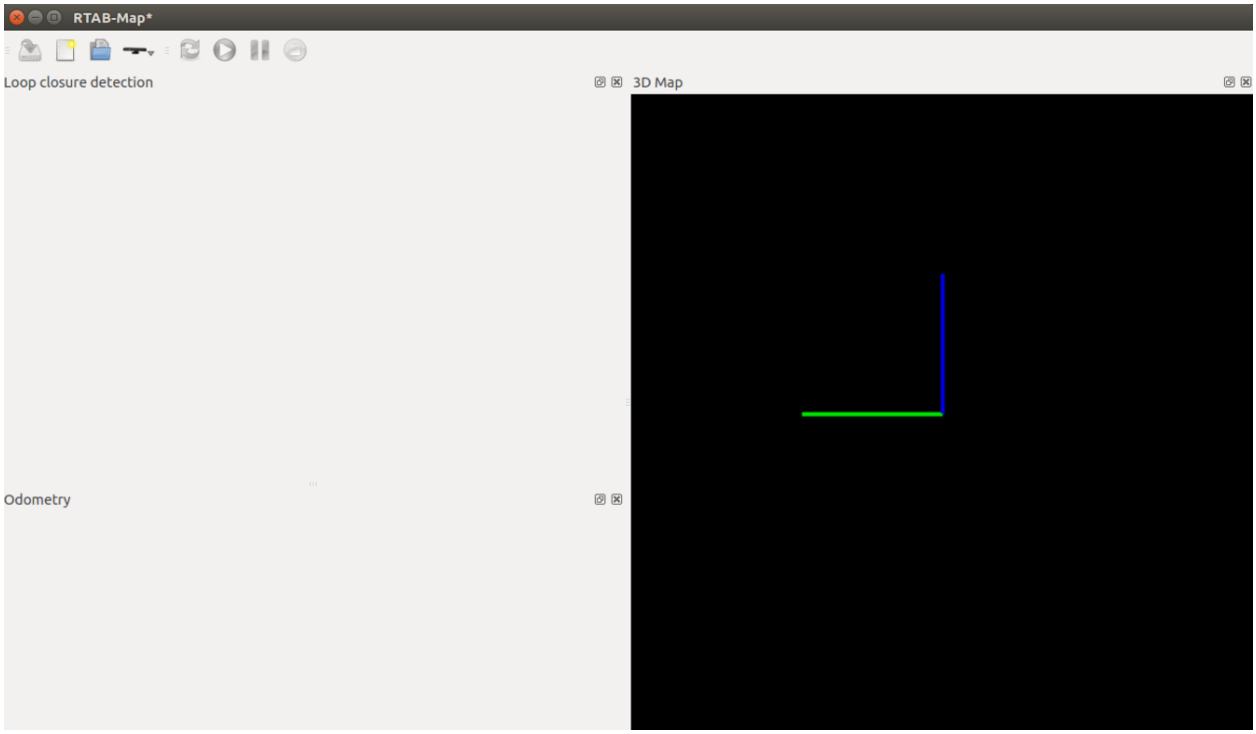
$ env | grep ROS

Now, we need you copy lab4_tutorial.launch file to rtabmap launch folder. Assume you put lab4_tutorial.launch under your ~/Downloads folder.

$ roscd rtabmap_ros
$ cd launch
$ sudo cp ~/Downloads/lab4_tutorial.launch lab4_tutorial.launch

Execute the following command to launch rtabmap
$ roslaunch rtabmap_ros lab4_tutorial.launch

You should be able to see the RTAB-Map window shows up like this. It is empty because we don't have any sensor data coming yet.



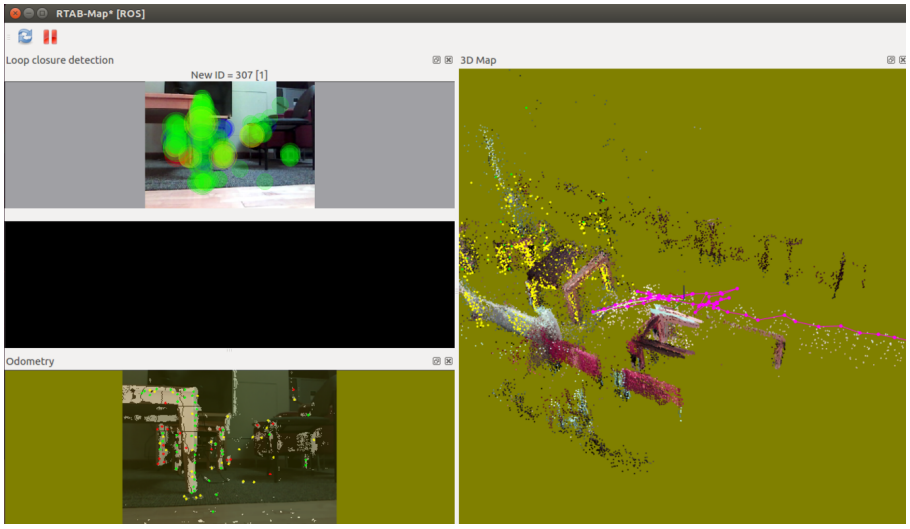
Open a new terminal, and check if the ROS_MASTER_URI and ROS_IP are set correctly. If it is not correct, you need to modify the ~/.bashrc and source the file in this terminal.

$ env | grep ROS

Now, you will play the ROS log data file to publish topics on your ROS system. Assume you put the downloaded lab4_tutorial_slam.bag in ~/Downloads.

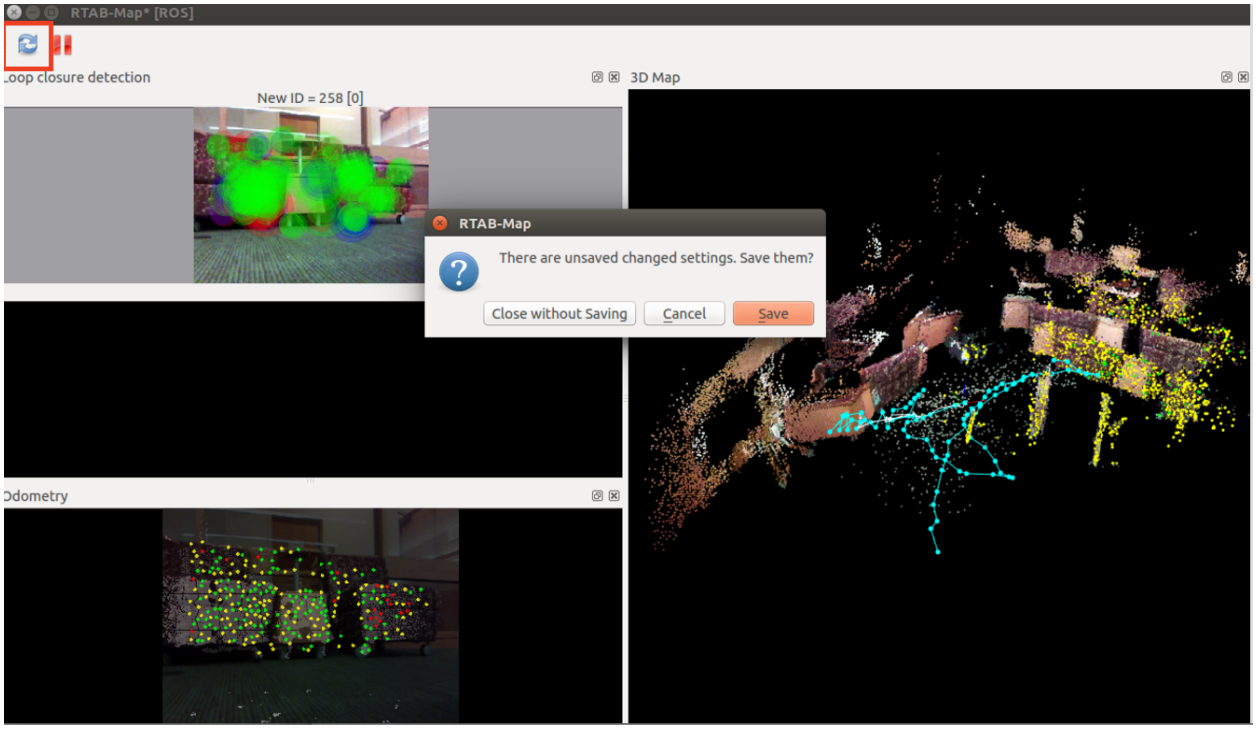
$ cd ~/Downloads
$ rosbag play lab4_tutorial_slam.bag --clock

You should see the images on RTAB-Map window constructing a 3D model. Normally, rtabmap background should be black, but sometimes it change to yellow, which means the camera is about to lose position tracking.



When the background turns RED, it means the SLAM doesn't work because its losing camera tracking. To recover the SLAM function, we will move the camera back to the previous successful position. From the ros log data, you can see we did this a few times.

You can press the button to save your map database or close the window and choose Save the map database. The saved database file will be at ~/.ros/rtabmap.db. You must rename it, otherwise, whenever you run rtabmap, the new map database will overwrite this map database.



After running all the logged ROS data and saved the map database, you can close rtabmap and the terminal. To open map database, you simply need to open a new terminal and run rtabmap.

$ rtabmap

And then keep the mouse focus on the rtabmap window, select the top left system bar -> file -> open database. Choose the map database you saved to reconstruct your 3D model.

Part 2: Construct 3D Model with Real-time Sensor Data

Please make sure you finish the Part 1 before you come to debug lab hours.

In this part, you will drive the car to capture the sensor data and build a 3D model. If you need to manually move the car, please hold the handle of the car, DO NOT just hold the car chassis!!

While you drive the car, you should try to avoid rtabmap background RED screen. When the background turns RED, it means that odometry cannot be computed, i.e. the odometry is lost and the mapping cannot continue. These events would cause RED screens:

1. Featureless environments: white walls, uniform texture, dark areas, etc. Without enough features, odometry cannot be computed. Odometry quality would be over 100 for environments with enough visual features.
2. "Empty space" environments: For sensors like the Realsense R200, good features are within around 0.5 to 5 meters of the sensor. By default, features farther than 5 meters are not used. For example, if you point the sensor over a clear area where there are no objects under 5 meters, there are good chances that the odometry will lose tracking.
3. Too fast camera motion: If the camera is moved too fast, there will be less features to match between successive frames.

When you come to the lab hours, you need to setup your wifi connection, just like in the previous lab, and modify the ~/.bashrc file, setting ROS_MASTER_URI to the car system.

Network settings (Running Ubuntu on laptop)
1. Connect the wireless network to SSID:f1tenthcar or f1tenthcar2.
Edit connection:

- IP address: 192.168.1.X
- Netmask: 255.255.255.0
- Gateway: 192.168.1.X
- (X is not 1 or 20)

2. In the .bashrc, set your ROS_IP as 192.168.1.X, and ROS_MASTER_URI as 192.168.1.1

$ export ROS_MASTER_URI=http://192.168.1.1:11311
$ export ROS_IP=192.168.1.X

3. Source ~/.bashrc
$ source ~/.bashrc

Network settings (Running Ubuntu on a VM)
1. Setup VM network interface (VirtualBox -> devices -> Network Settings) as Bridge Adapter.

2. On your host machine (Mac or Windows), connect the wireless network to SSID:f1tenthcar or f1tenthcar2. Open network preference -> advanced -> TCP/IP:
- IP address: 192.168.1.X
- Netmask: 255.255.255.0
- Gateway (Router): 192.168.1.X
- (X is not 1 or 20).

3. In the VM (Ubuntu), set Ubuntu connection (top right of the system bar, edit connection)
- IP address: 192.168.1.Y
- Netmask: 255.255.255.0
- Gateway: 192.168.1.X
- (X, Y are not 1 or 20).

4. In the .bashrc, set your ROS_IP as 192.168.1.Y, and ROS_MASTER_URI as 192.168.1.1

$ export ROS_MASTER_URI=http://192.168.1.1:11311
$ export ROS_IP=192.168.1.Y

5. Source ~/.bashrc
$ source ~/.bashrc

Implementation:
Check if your environment settings correctly. ROS_MASTER_URI should be http://192.168.1.1:11311, and ROS_IP should be your ubuntu IP.

$ env | grep ROS

Put the keyboard.py to replace the one you use for lab3
$ cp ~/Downloads/keyboard.py ~/lab3_race/src/race/src/keyboard.py
$ chmod 755 ~/lab3_race/src/race/src/keyboard.py

Prepare to drive the car
$ cd ~/lab3_race
$ source devel/setup.bash
$ rosrun race keyboard.py

Try to drive the car forward speed at 8 units, or backward speed at -11 units, which is the slowest moving speed.

Prepare to launch rtabmap:
Copy lab4.launch file to rtabmap launch folder. Assume you put lab4_tutorial.launch under your ~/Downloads folder.

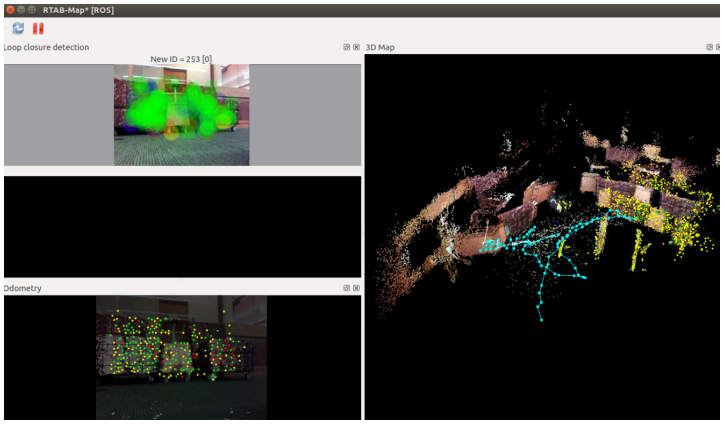
$ roscd rtabmap_ros
$ cd launch
$ sudo cp ~/Downloads/lab4.launch lab4.launch

Set ros parameter use_sim_time to false
$ rosparam set use_sim_time false

Check if the setting is correct. It should be false.
$ rosparam get use_sim_time

Now, place the car facing a environment with a lot of features, and then launch rtabmap
$ roslaunch rtabmap_ros lab4.launch

Drive the car carefully and build the 3D map.



Remember save map db when you close the window.

Some tips when driving the car to run SLAM:

1. Don't drive too fast. Drive the car at 8 units forward speed, and -11 units backward.
2. Don't steering the wheels with a large angle. Make the turning within 50 units.
3. If rtabmap background turns to red screen, don't steer the wheel. Replace the camera where it failed to compute the odometry. The camera pose may be tracked again (the RED background disappears) and the mapping should continue.
4. If rtabmap always becomes RED background at the same place in the environment, try to add texture/objects to the scene to add more visual features to track.

Requirement:

- Show the 3D model you build with the logged data. (Part 1)

- Show the 3D model you build in the Zar Room. (Part 2)
    - Drive the car in the designate area in Zar Room.
    - Find an efficient driving path or cycle in the area to build the 3D model.
    - Make the 3D model clear (high quality). Complete the 3D scene construction with at least 3 driving loops, and with rtabmap's loop closure it should improve model with each iteration.