Final Project 2019S2: Where's Wally the Brick?

Due Dates: Project Demonstration - 14 October 2019
Project Report - 18 October 2019

This assessment is worth 30% of the overall mark. It is a group project of 3 students.
The project covers most of the topics discussed during the lectures. The aim is to go through various stages to locate multiple objects in a previously mapped environment and navigate to Wally the Brick.

The stages are as follows:

Stage: MAINSTREAM - LIDAR SLAM	**Marks (Total 100 + bonus)**
Stage 1. 2D Lidar SLAM with teleoperated Turtlebot. The output of this stage is a 2D occupancy map. Use ROS Package GMapping.	15 points
Stage 2. Localization using a pre-built map with 2D Lidar. Use ROS Package AMCL.	10 points
Stage 3. Map exploration (autonomously drive around). Using ROS package. Use ROS Move base and Frontier Exploration.	15 points
Stage 3+. Map exploration (autonomously drive around). Using own algorithm.	30 points
Stage 4. Find Wally the Brick using the camera information and navigate towards it. Use OpenCV.	25 points
Written report.	20 points
First group to navigate to Wally the Brick.	Bonus 5 points
Most efficient exploration algorithm.	Bonus 5 points
Update occupancy map with the location of Wally the Brick.	Bonus 10 points

Stage: VARIATION 1 - ORBSLAM with RGBD Camera	**Marks (Total 100 + bonus)**
Stage 1. ORBSLAM with teleoperated Turtlebot.	25 points
Stage 2. Exploration (autonomously drive around) while running ORBSLAM. Using ROS package.	15 points
Stage 2+. Exploration (autonomously drive around) while running ORBSLAM. Using own algorithm.	20 points
Stage 3. Discover and navigate towards Wally the Brick while running ORBSLAM and exploration.	20 points
Written report.	20 points
First group to navigate to Wally the Brick without missing other bricks.	Bonus 5 points
Most efficient exploration algorithm.	Bonus 5 points
Update ORBSLAM map with Wally location.	Bonus 10 points

Stage: VARIATION 2 - Matlab SIMULATION	**Marks (Total 100 + bonus)**
Stage 1. Create simulated environment with 30 point landmarks and a robot.	5 points
Stage 2. Perform SLAM EKF with range and bearing observations in the simulated environment. The output of this stage is a 2D feature point map of 30 landmarks.	35 points
Stage 3. Add 15 more points to the simulated environment and 1 line (Wally the Brick).	5 points
Stage 4. Explore the existing map using own algorithm. Find Wally the Brick and navigate towards it.	15 points
Stage 5. Perform mapping only for the new points and visualise uncertainties.	20 points
Written report.	20 points
Most efficient exploration algorithm.	Bonus 5 points
Creative simulated environment.	Bonus 5 points

Note that VARIATION 1 Project is limited to 2 projects due to hardware availability and own laptops running Ubuntu and ROS are required (No Virtual Box). VARIATION 1 requires proficient C++ programers.

Hardware:

- Own Laptop / Raspberry Pi (own SSD card)/ Computer Labs
- Turtlebot 3 Waffle
- 360 Laser Distance Sensor LDS-01
- Intel® RealSense™ Camera R200 (MAINSTREAM)
- Orbbec 3D Camera (VARIATION 1) - Limited to 2 projects

Software:

ROS Kinetic installation:
http://wiki.ros.org/kinetic/Installation/Ubuntu
Turtlebot3 documentation:
http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/
Turtlebot3 simulator:
http://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/#turtlebot3-simulation-using-gazebo
VARIATION 1 - ORBSLAM
https://github.com/raulmur/ORB_SLAM2

Schedule:

- 2 September 2019 - Project assignment description is distributed on UTSOnline.
- 2-6 September 2019 - Team formation.
- 5 September 2019 - If you are planning to work in any of the VARIATION PROJECTS, submit your preference with a paragraph explaining the choice.
- 6 September 2019 - Confirmation of the VARIATION PROJECT.
- 16 and 23 September 2019 - 7:30pm Help sessions after the lecture at Computer Lab CB11.B1.400 and CB11.B1.402.
- 30 September 2019 - 6pm and 7:30pm Help sessions with robotic hardware at Mechatronics Lab CB11.10.403
- 14 October 2019 - 6PM Project demonstration: explain and demonstrate stages of the project.

Note that MAINSTREAM project is encouraged. Students who choose a VARIATION project have to make a case by submitting their preference elaborating the reason for the choice to Teresa.VidalCalleja@uts.edu.au.

A **written report** is required for each group detailing the project stages, the methodology adopted, the algorithms implemented, and the final results achieved. Images of the generated maps both in simulation and real robot and the navigation strategy. Note there is a **Bonus of 5 points if you submit a Video** with the report.

The mark of each student will be based on the performance of the group as well as their individual contribution. The report will contain a statement on the individual contributions.

Hints

General

- The model of the TurtleBot3 is "waffle".
- To simulate a TurtleBot, follow the "TurtleBot3 Simulation using Gazebo" tutorial and use either the "world" or "stage 4" worlds.
- To map an environment, follow the "SLAM" tutorial.
- For navigation, follow the "Navigation" tutorial.
- If you are using the computers in the FEIT computer labs, remember that you need to start a shell in the Singularity container.
- Whenever you start a shell, remember that you need to source a setup script (either the global one at "/opt/ros/melodic/setup.bash" or the one in your workspace).
- You also need to set the TurtleBot3 model before you run the launch files ("export TURTLEBOT3_MODEL=waffle")

Exploring the map

- To explore the map efficiently, you will want to read the map (nav_msgs/OccupancyGrid) published by map_server, and control move_base by publishing a goal (geometry_msgs/PoseStamped) or by using the Action API

Working with images

- To work with an image in a ROS node, you will want to use cv_bridge and OpenCV
- There are tutorials for OpenCV

Using Your Own Computer

- You can use Ubuntu 16.04 or 18.04, there shouldn't be much difference.
- If you are using Ubuntu 16.04, install ROS Kinetic.

- If you are using Ubuntu 18.04, install ROS Melodic.
- You will need at least these additional packages:
  - map-server
  - turtlebot3
  - turtlebot3-simulations
  - gmapping
  - dwa-local-planner
- To avoid having to source the setup script and set the TurtleBot3 model every time you open a terminal you can add the commands to the ".bashrc" file in your home directory.
- There is a bug in the "turtlebot3_navigation" package that will prevent navigation from working. You need to remove slashes from the "global_costmap_params.yaml" and "local_costmap_params.yaml" files. You can run these commands to do it:
  - sudo sed -i 's/\///g' /opt/ros/melodic/share/turtlebot3_navigation/param/global_costmap_params.yaml
  - sudo sed -i 's/\///g' /opt/ros/melodic/share/turtlebot3_navigation/param/local_costmap_params.yaml