Project log - Robotica

Augello Andrea Castiglione Francesco Paolo
La Martina Marco

December 22, 2020

Contents

1 Setup

OS	Ubuntu 18.04 Ubuntu 20.04
ROS version	melodic noetic
Webots	R2020b revision 1
Target hardware	Raspberry Pi 4B Raspberry Pi 3B+

2 Name

Our team has chosen the name **Change**, which resembles **Chang'e 4** [2], the spacecraft mission part of the second phase of the Chinese Lunar Exploration Program, which achieved humanity's first soft landing on the far side of the moon.

3 Libraries and environment

We have used the **webots_ros** [3] package in order to gain deeper understanding of how to interface ROS nodes with the standard ROS controller for Webots. We have also studied the ROS documentation [4] in order to install and configure the ROS enviroment and also to understand fundamental ROS concepts related to nodes and topics. Moreover, we set-up the ROS interface in Webots following the cyberbotics documentation [4].

4 Task

Our robot will be deployed in a room (such as the one showed in our demo) and its aim is to identify humans and estimate their relative positions. If the distance between said humans is less that a specified value, the robot will go towards them and invite them to respect social distancing (with both visual and audio output).

5 Tiago Iron

The robot selected for the given task is the **TIAGo Iron**.
PAL Robotics TIAGo Iron[1] is a two-wheeled human-like robot with a torso and a head but no articulated arm. The model is a modular mobile platform that allows human-robot interaction. **We use a IMU with 6 degrees of freedom**. IMU:

1. gyro;
2. accelerometer;

We got rid of the compass in the IMU.

6 Movement primitives

[8]

7 Positioning

Implementing Positioning Algorithms Using Accelerometers.

8 Projection Matrix

[7]

9 Dependencies

- OpenCV 4.x
- Imutils

10 Object recognition

We evalued performance between YOLO V3, TinyYOLO, HoG , HoG + SVG , HoG + SVG + NMS. Yolo wins because it is 443% more efficient. Width and not height. Yolo yields much more tight bounding boxes.

11 TIAGo Wheels

We asked the developers: 200mm. We discovered that the webots model is not the same size as the TIAGo datasheet.

12 Clustering

We decided to lower the dimensionality of our data. We used cilindric coordinates and the feature vector is 2 dimensional. We used the Density-Based Scan with a threshold. The entities not belonging to the cluster are discarded.

13 SLAM

We decided to use gMapping, one of the most used SLAM algorithms[9].

- Gaussian (EKF) approximation of odometry model from Probabilistic Robotics
- Discrete time steps (=when updates happen) correspond to whenever the robot has traveled

14 ROS

15 Bugs found in the Webots ROS Controller

Logical values did not allow callbacks.

References

[1] *https://cyberbotics.com/doc/guide/tiago-iron.*
Webots TIAGo Iron documentation.

[2] *https://www.theguardian.com/science/2019/jan/03/china-probe-change-4-land-far-side-moon-basin-crater.*
The Guardian, 3 January 2019.

[3] *https://github.com/cyberbotics/webots_ ros.*
Github page for the `webots_ros` package from *cyberbotics*.

[4] *https://wiki.ros.org/ROS/Tutorials.*
ROS documentation from ROS.org.

[5] *https://www.cyberbotics.com/doc/guide/tutorial-8-using-ros.*
Cyberbotics documentation.

[6] *https://pal-robotics.com/wp-content/uploads/2019/07/Datasheet_ TIAGo_ Complete.pdf.*
Tiago IRON datasheet.

[7] *https://www.songho.ca/opengl/gl_ projectionmatrix.html.*
OpenGL Projection Matrix.

[8] *https://www.nxp.com/docs/en/application-note/AN3397.pdf.*
Implementing Positioning Algorithms Using Accelerometers.

[9] *https://people.eecs.berkeley.edu/ pabbeel/cs287-fa11/slides/gmapping.pdf.*
Gmapping from UC Berkeley EECS, Pieter Abbeel.