Project log - Robotica

Augello Andrea Castiglione Francesco Paolo
La Martina Marco

December 24, 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 Environment

We have explored 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 Dependencies

This is a list of the libraries used in our project and a brief explanation of their relevance:

- opencv 4.x, a library aimed at computer vision[10];
- imutils, series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization[11];
- sklearn, a machine learning library, featuring various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN[12];
- numpy, a library that adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays[13];
- matplotlib, a comprehensive library for creating static, animated, and interactive visualizations[14];

5 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 avoid gatherings. It must estimate people's relative positions and, 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).

6 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 added a **speaker** and a **display** with a corresponding support solid to the base TIAGo model available in Webots. We also had to ask the Webots developers for the precise size of the **wheels** since the model does not exactly match the specifications given in the data TIAGo datasheet[6] and we discovered that they are 200mm.
We also decided to modify the IMU in order to best fit our goals and use an IMU with 6 degrees of freedom. The IMU consists of the following components:

1. gyro;
2. accelerometer;

We decided to **not use the compass** because in a real scenario it would have been subject to various degrees of interference (significantly more so than a gyro), especially in an environment with many metal objects.

7 Movement primitives

[8]

8 Positioning

Implementing Positioning Algorithms Using Accelerometers.

9 Projection Matrix

[7]

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 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.

12 ROS

13 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.

[10] *https://opencv.org/.*
OpenCV Website.

[11] *https://github.com/jrosebr1/imutils.*
Imutils GitHub page.

[12] *https://scikit-learn.org/stable/.*
Scikit-learn website.

[13] *https://numpy.org/.*
Numpy website.

[14] *https://matplotlib.org/.*
Matplotlib website.