Robotics Project 2021/2022

Arslan Ali
Lugano, Switzerland
Email: aliar@usi.ch

***Abstract*—An example of Thymio running in a simulated environment(CoppeliaSim) using YOLOv5 to detect plants.**

Keywords: YOLO, ROS2, Random Walk.

I. Introduction

This project is divided in **5** main steps, and in the following sections they will be explained.
The main idea is to reproduce an environment as much as similar as USI west and east campus in a simulated world, where the Thymio navigate through it without crashing and in the meanwhile recognize objects using a real time object detection algorithm. The following fundamental steps will be explained in detail as sections of this paper:

- World Preparation
- Thymio Preparation
- Path Planning
- Object Detection
- Results

II. World Creation

The world, representing the map of USI west campus and east is recreated using OpenStreetMap, where the portion of interested area has been selected and exported as *.osm* file. The latter is then converted from *.osm* to *.obj* using an open source tool OSM2World. The *.obj* file is then imported into

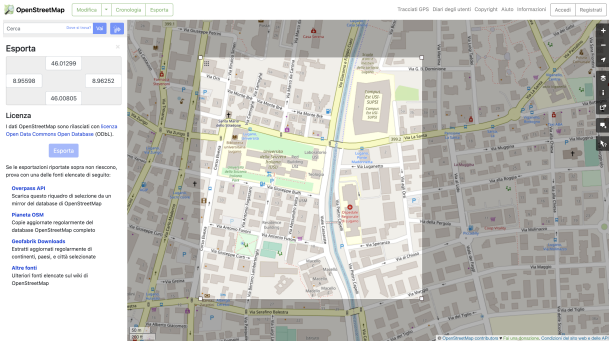

Fig. 1: Exporting map from OpenStreetMap

CoppeliaSim using *mesh* option, with Auto Scaling enabled and scaling it by a factor of *0.5* keeping also all the textures. Then, in the simulator a resizable floor of medium size has been chosen, and the map is successfully imported. In order to make the walls of the map collidable and detectable, for each of the building of the map, they're added manually, and also in order to keep the Thymio from falling off the map, the perimeter of the map has been surrounded by walls.

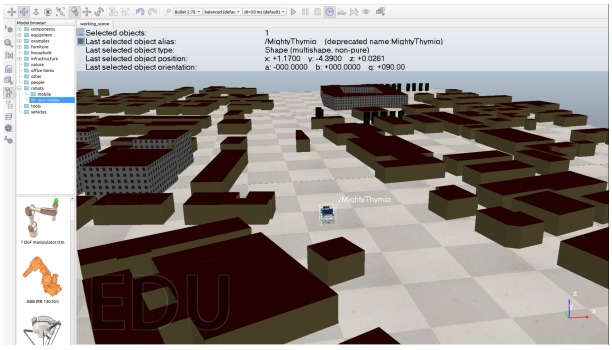

Fig. 2: Imported map into CoppeliaSim

III. Thymio Preparation

The robot that has been used into this project is Thymio, an open-source educational robot designed by researchers from the EPFL (1). Once the Thymio has been imported into the world, some little modifications to the default parameters has been applied. In particular the *CameraSensor* parameters has been modified:

- Pers.angle[deg]/ortho.size[m] has been modified from 40.0 to 115.0
- Resolution X/Y has been modified from 320/240 to 640/480

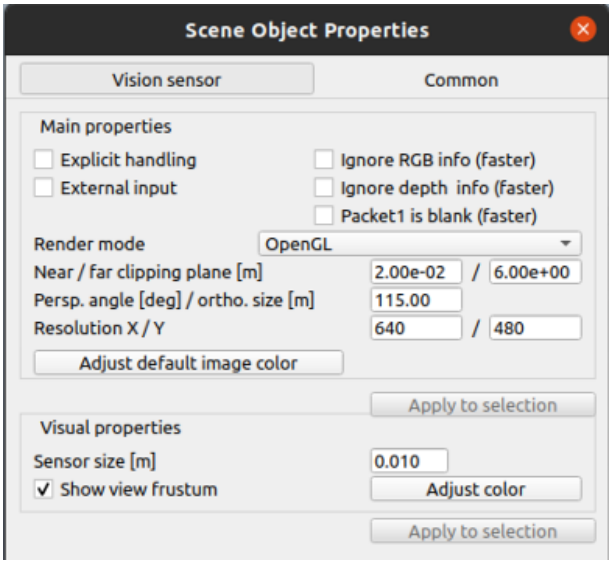

Fig. 3: Thymio configuration settings

IV. Path Planning

In order to randomly explore the map, the Thymio is using a very naive algorithm, known also as *Random Walk* (used by the first versions of Roomba (2)).
As the name suggests the main idea of this algorithm is to randomly move the robot. A random walk does not require any precise realization of the route plan. The robot moves in forward direction until an obstacle is sensed and then it stops if there is any barrier. Next, it turns by comparing sensor readings (in Thymio *proximity sensors* has been used) from the left or right direction in which to turn and finally by generation a random number it decides how much to turn.

Algorithm 1 Avoid hitting obstacles

```
while Thymio is moving forward do
    if sensors detect obstacle then
        Stop moving
        repeat
            Move Thymio backwards
        until sensors detect NO obstacle
        Rotate randomly in a range
    else
        Keep going forward
    end if
end while
```

V. Object Detection

In order to detect objects in the map a NN approach has been adopted. In particular YOLO algorithm has been used, it uses a neural networks to provide real time object detection. This algorithm is popular because of its speed and accuracy. It has been used in various applications to detect traffic signals, people, parking meters, and animals. As the name suggests, the algorithm requires only a single forward propagation through a NN to detect objects (3).
For this project YOLOv5 has been used, the architecture is pretrained on the COCO dataset. In order to use YOLO to detect and perform object recognition in camera images using ROS2, an external package has been used CvBridge, which converts between ROS2 Images messages and OpenCV images, where the latter are fed to YOLO to perform inference.

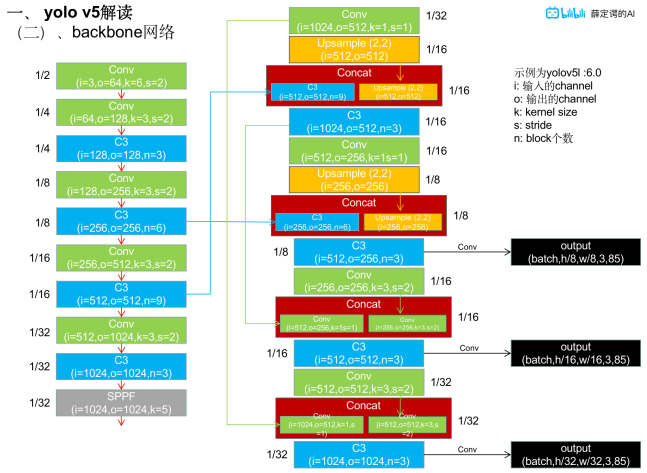

Fig. 4: Yolov5 architecture adopted

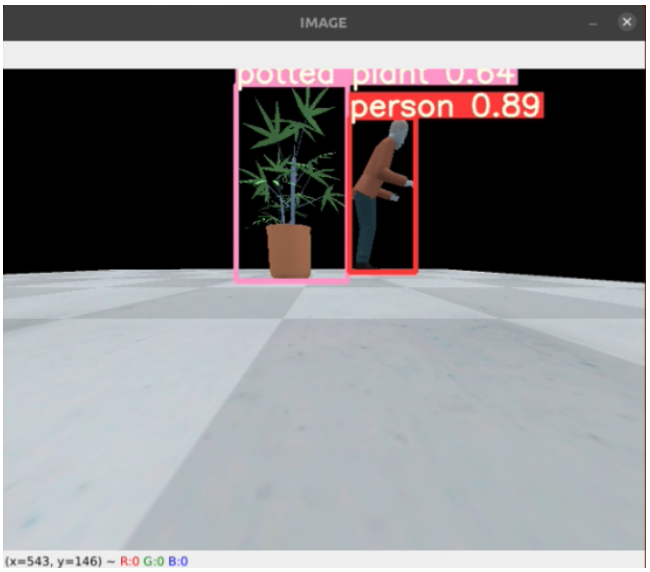

Fig. 5: Object detection in CoppeliaSim

VI. Results

As explained in previous sections, the Thmyio randomly explores the map while it's performing object detection. In particular, the task that the Thmyio is accomplishing in this project is, whenever a *potted plant* is detected in the map (plants have been randomly added to the map) it's stop moving and it begins to turn in on itself, meaning that the task is accomplished successfully. Otherwise it keeps moving randomly as explained in the **Algorithm 1**. The latter is modified in this way. Originally the aim of the project was, instead of

Algorithm 2 Avoid hitting obstacles and detect potted plants

```
while Thymio is moving forward do
    if Camera detects potted plant then
        Stop moving
        Start spinning (task completed)
        BREAK
    end if
    if sensors detect obstacle then
        Stop moving
        repeat
            Move Thymio backwards
        until sensors detect NO obstacle
        Rotate randomly in a range
    else
        Keep going forward
    end if
end while
```

detecting only potted plant, any other object in CoppeliaSim, and based on the object detected, the Thymio adopt a different behavior. Unfortunately, because the simulator is run in a virtualized environment, and running ROS with CoppeliaSim and YOLO is computationally heavy, after a few minutes the system stopped being responsive by crashing, that's why the task is simplified.

REFERENCES

[1] "Home," Dec 2021. [Online]. Available: https://www.thymio.org/

[2] J. Sörme and T. Edwards, "A comparison of path planning algorithms for robotic vacuum cleaners," 2018.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.