Obby: The object recognition and obstacle avoidance robot simulation in Webots

Arvindeva Wibisono Saket Narendra

May 30, 2021

Abstract

Recognition of objects has always been of interest to many people. In this work we show how we achieved recognition of objects in Webots while maneuvering the robot manually as well as autonomously. In addition to recognising the objects, we also implement an object tracking mode and an object counting mode.

1 Introduction

In our work, we chose to make use of Webots[1] which is an open source platform as our software to simulate the robot. We chose to make use of webots as it is an easy to use software that allows us to program our robot in any programming language required, including Python, in which we are already familiar with. We will talk about the functionalities of the robot and it's design as well.

2 Novelty

Such a robot can be used in many fields. If an object has been lost by somebody, our robot can help find it in tricky areas. It can also be used in fun robotics competition where one has to traverse through multiple obstacles to reach the final destination. Another use can be to count the number of objects that are present in high profile work areas where the object count can be very useful. There are many more use cases for such robots.

3 Design

Our robot consists of a rectangular cuboidal body equipped with a Webots camera and a distance sensor at the periphery of the body. The robot has 4 wheels for stability. We made the robot from scratch using available shapes form Webots. The base of the robot, which is the body, is a rectangle shape with default Webots physics and bounding object. Connected to the body is the four wheels, which are identical cylinders. To connect the wheels to the body, we used default HingeJoints from Webots and we gave them motors, in which we will control later from the controller. The environment of the robot is a child's bedroom which has a lot of toys scattered all around. We also designed and implemented the environment from scratch.These objects are of different shapes, sizes and color such as spheres, cylinders, cones and cubes.

We also design our robot's controller ourself, in which we decided to use Python since

it's the language that we are most familiar with. In the robot controller, we name it 4_wheels_robot_python.py, is where all the magic happens: Initializing the robot, initializing the camera and the distance sensor, capturing keyboard input, controlling the motors, and a few modes for the robot to either drive itself or manually by the user.

4 Implementation

This robot is implemented entirely in Webots. Webots give us the capability to make our own robot, give it a camera, and give it a distance sensor. With these features, we don't need any external software such as OpenCV to process the camera image to achieve our goal. With the robot having a distance sensor and a camera, it could easily detect the objects placed in the room. It is able to detect the different objects telling what shape and colour it is. With the distance sensor it is also able to measure the distance between itself and the object. With this information, the robot could avoid the object by moving around it.

By using the built-in camera from Webots, our robot can use one of its most useful features, which is the Recognition feature. With the Recognition class that is already built-in, our camera was able to detect what an object is, as long as we give the right details on the object node in Webots. We use this feature heavily to implement our robot's other modes such as the object tracking and object counting modes. For example, in the object tracking mode, the robot can know the position of an object relative to the camera just by using the Recognition class.

We also implemented a manual drive which allows us to drive the robot by just pressing the keys W,A,S,D. For this, we had to use the Keyboard class from Webots, and then activate the keyboard in the robot controller to make sure that the simulation is able to detect keyboard inputs of the user. By doing so, the user has the choice of either moving the robot themself or letting the robot drive itself. We choose to use W,A,S,D control because it's the standard for doing four-directional movement on a keyboard (along with arrow keys,but we find that WASD is more comfortable).

We also implemented an object counting mode, where the robot is also able to tell the number of objects present in the room by continuously rotating, keeping track of the names or IDs of the object to avoid duplicates. To do this, again we use the Recognition class from Webots. The implementation is basically as follows: Initialize an empty list of objects to keep track of what is detected so far. For every timestep, list all of the objects that are in vision. For every object, check if that object's ID is already in the object list. If it's not, that add that ID to the list. Repeat until a keyboard input is pressed to stop the mode.

The last mode that we implemented is the object tracking mode. In this mode, we need to specify an input string of the name of the object that will be tracked. Once the mode starts, the robot will rotate in its place until it finds the specified object. Once it finds the object, the robot will move forward towards the object, and will make sure that the object is still in the middle of its camera vision. In every timestep, it will check the position of the object in the camera's image and it will turn left or right according to where it is, making sure that the object is in the center.

5 Results

We successfully implemented everything that we wanted to achieve, and here is the demo of our robot in action: https://youtu.be/XsFrYIHvWuI

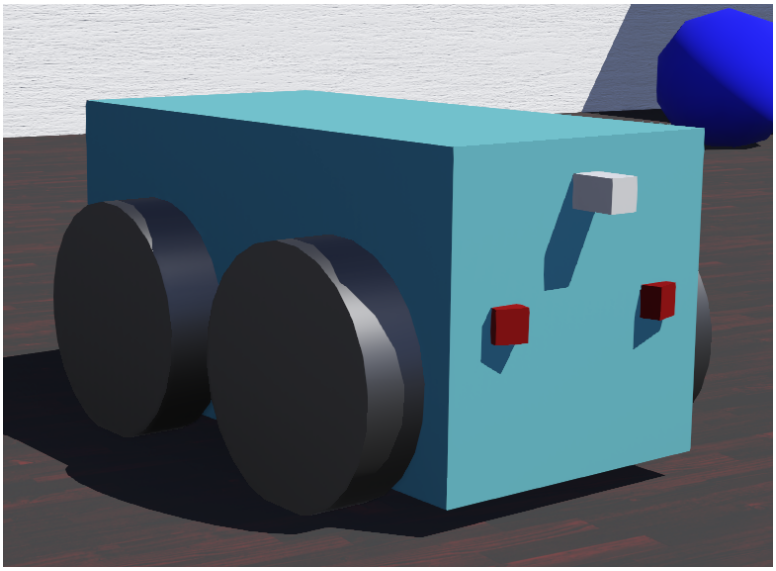

Figure 1: Latest version of Obby, an object detection and obstacle avoidance robot. The white block on its forehead is the camera, the red eyes are the distance sensors.

Figure 2: The children's bedroom environment with different objects detectable by our robot.

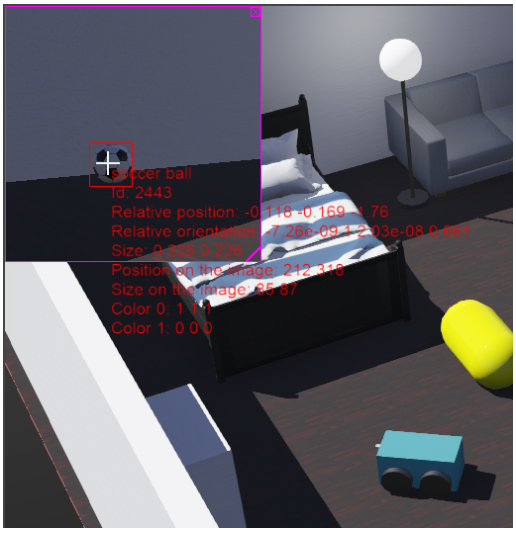

Figure 3: Webots object recognition in action, with this we were able to know the id, name/model, and position of an object easily.

6 Conclusion

In this work, we talked about how we worked with Webots software to make a robot that detects, avoids and counts objects placed in a room. The robot was successfully able to tell which object was placed in front of it and move around the obstacles that it encountered with the help of the distance sensor. We also provided a manual drive for user interaction with the environment. From this project, we learned a lot about robotics as a whole, especially regarding simulation in Webots. We did not have any experience with the software prior to this project, so it was a lot of fun going through the long learning process to learn about the different features the software has to offer. We were really excited to see our robot working as intended with its many modes.

References

[1] https://cyberbotics.com