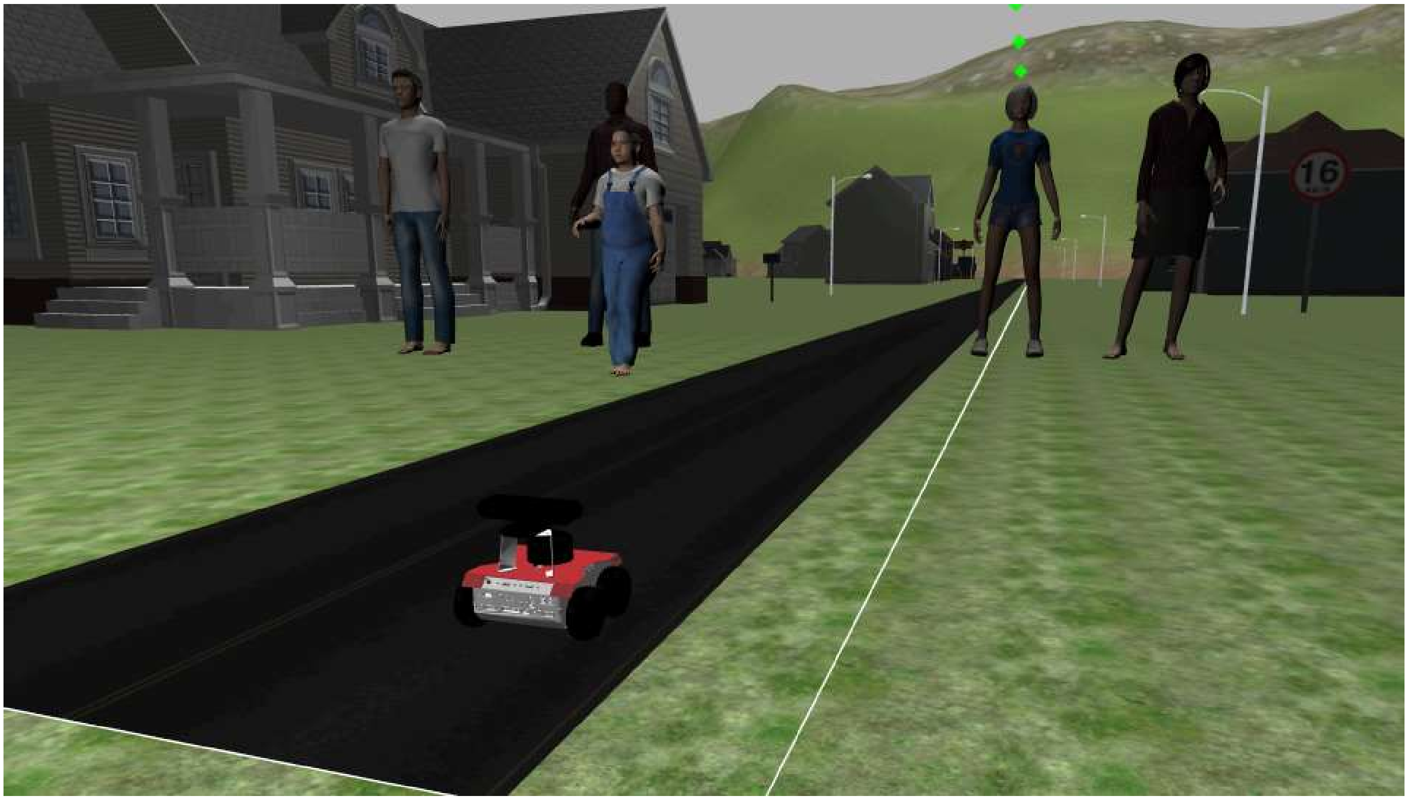
OpenCV for Robotics

Project: The reward

Introduction

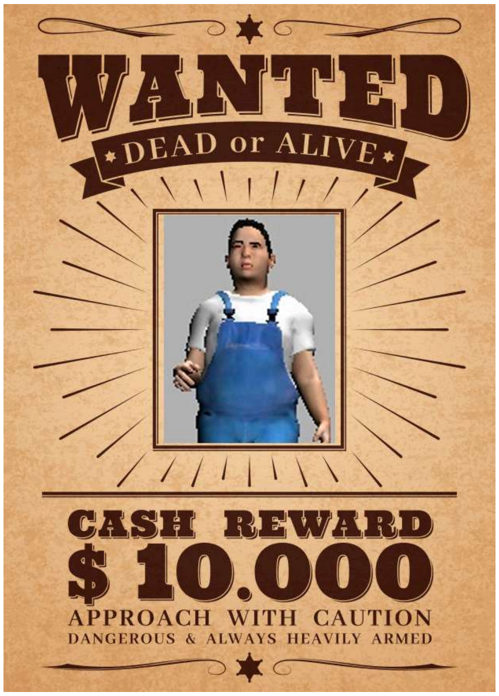


The time has come to put all the knowledge you've learned during the course into practice... and get a great reward! Bear in mind that you are now in the world of artificial vision...

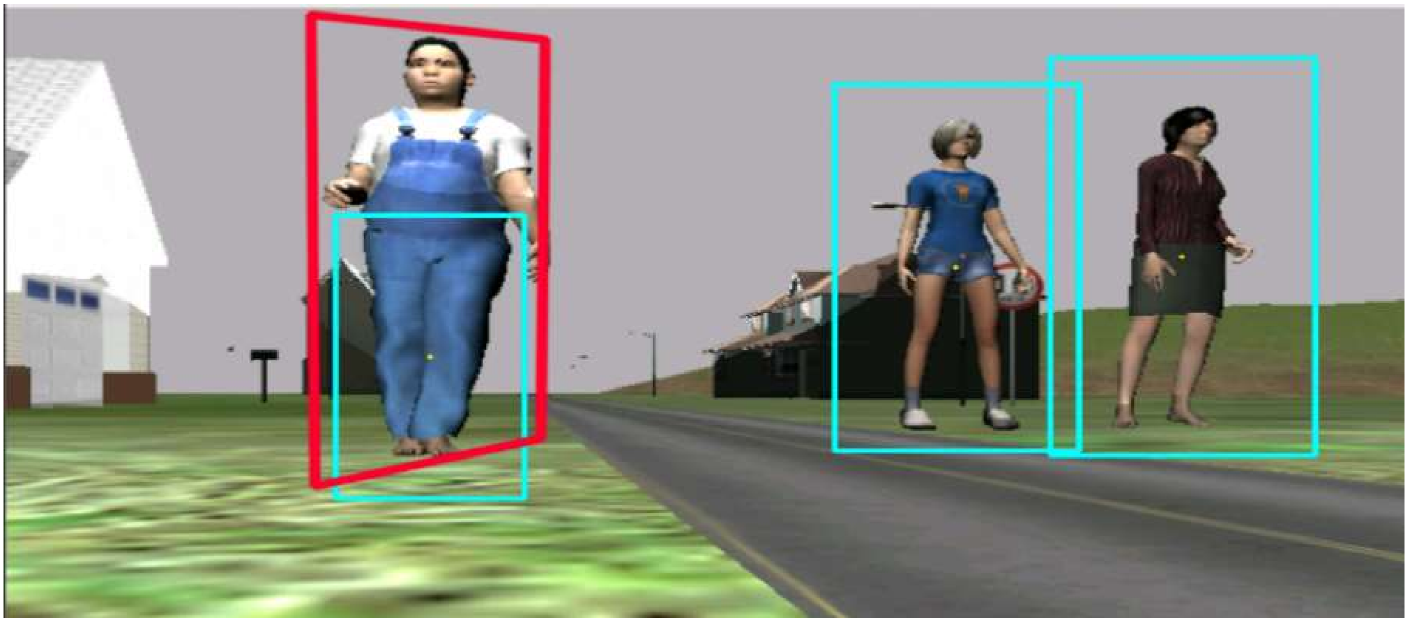
This project will be quite interesting. You already had a glimpse when you started the course, during the demo. However, the time has come for you to develop it with your own hands!

Scene

There is a dangerous person in this city, and many possible suspects are close to your robot. You must detect all the people and highlight the dangerous person.

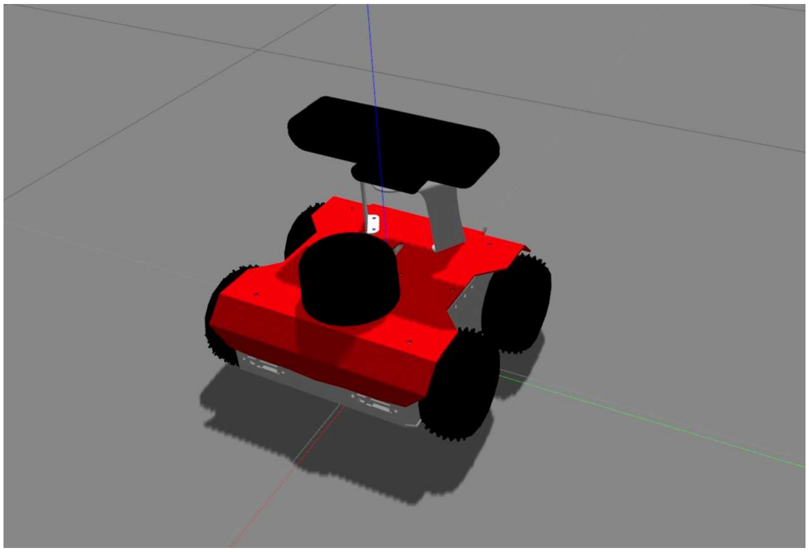


Below you can see an image that demonstrates the detection and highlights the dangerous person:



The ROSbot

For this project, you will use the ROSbot:



Tasks to accomplish.

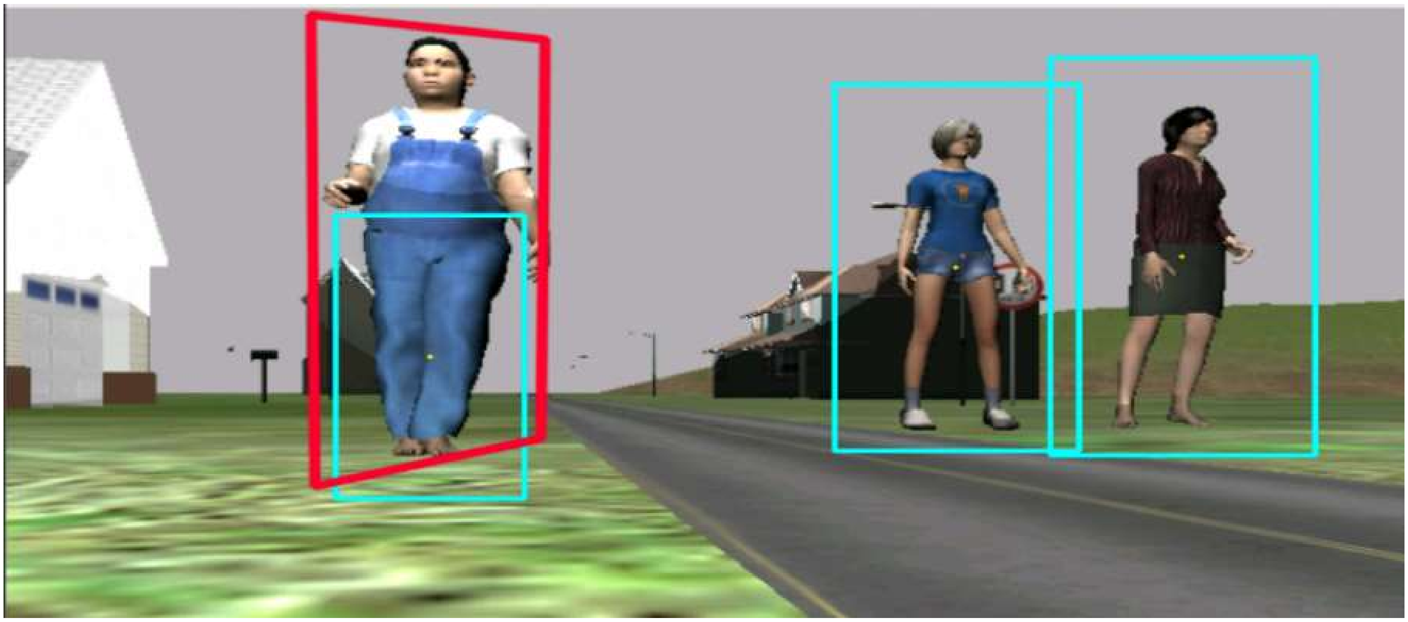
- Create a new package and call it **project**, with rospy as dependency.
- Create a folder named **launch** inside the package. Inside this folder, create a launch file to spawn the wall with the ARTags, like in **Unit 5**.
- Create the code to show the image of the wanted person (shown above) in the ARTags. You can find the image in the following path: **'/home/user/catkin_ws/src/opencv_for_robotics_images/Project/Course_images/wanted.png'** and show it.
- Afterwards, you can delete the ARTags wall using the following command:

In [ ]:
```
rosservice gazebo/delete_model
```

- Create a file to detect all the people who are close to the robot.
- Move the robot and try to find the wanted person. Remember, you can move the robot using the keyboard with the following command:

In [ ]:
```
roslaunch rosbot_navigation rosbot_teleop.launch
```

- Once you have found the wanted person, create a file to take a picture of him and save it in a new folder in your package, named **images**.
- Use this picture in order to detect the wanted person in a red square. The rest of the people will be marked with a blue square.
- The final result should be something like in the image below:



- Notes -

Remember that in simulations, some algorithms won't work as well as with real images. However, if the result is close to the one shown, it will be fine.

- End of Notes -