OpenCV for Robotics

Unit 1: Introduction

Introduction

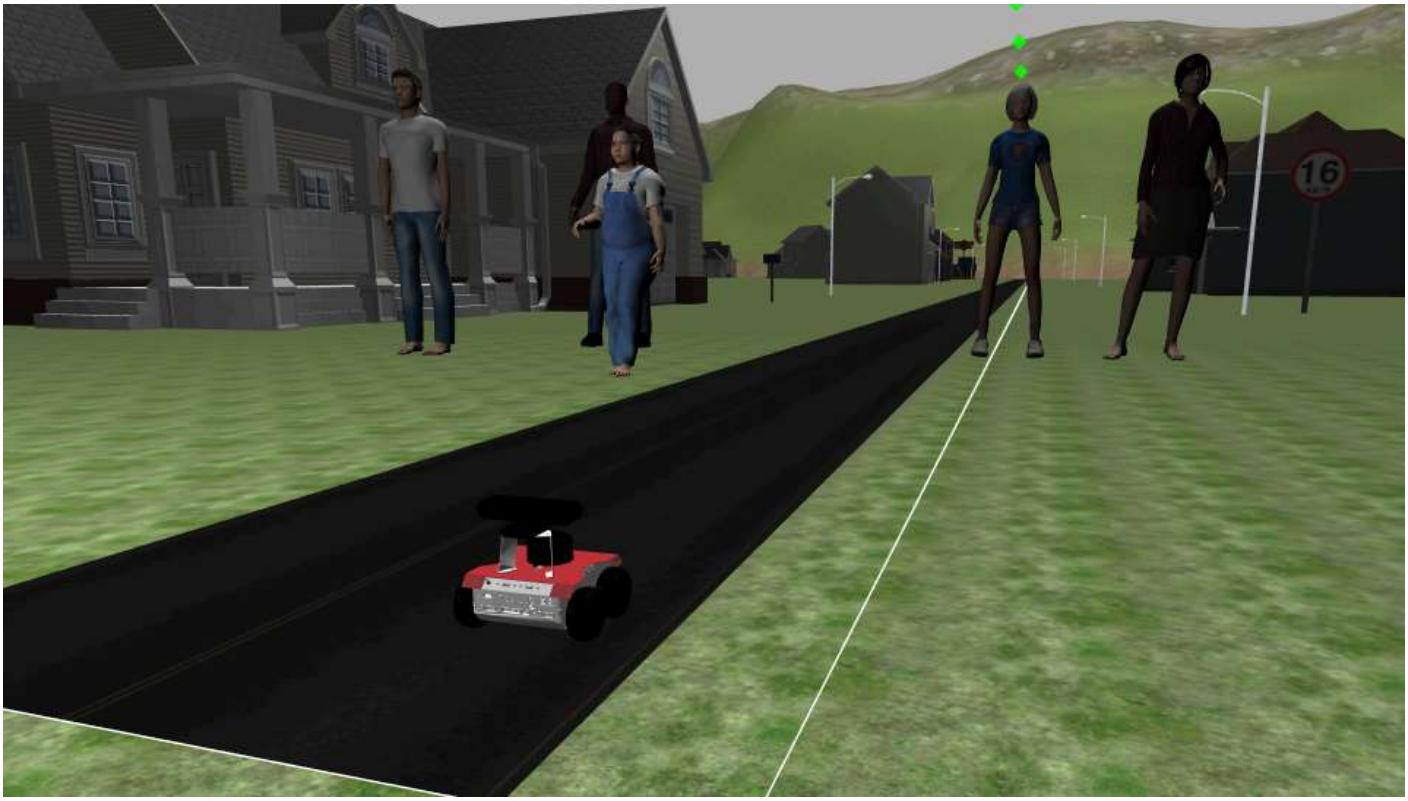
Anyone who has heard of computer vision has heard about Open CV, the great library that offers you a large number of computer vision algorithms that allow a study of one image or several images.

In the world of robotics, knowing how to extract information from a medium, such as an image, can be essential, which is why combining both can give an interesting result and is worth putting into practice.

Relate to the environment, recognize patterns, understand concepts of pixels, colors, borders, detection of objects, detection of people, faces, etc. They make this combination one of extraordinary potential when it comes to obtaining useful information for a robot about the system that surrounds it. A clear example is the use of cameras in robotics, from drones to mobile robots. The use of cameras has been a constant for quite some time, and it is a tool that cannot be underestimated.

Working Example: Let's detect people with the ROSBot

In the simulation on the right, you can see a robot called ***ROSBot***, which is not alone. You can see that there are some people pretty close to it. Hopefully, this robot is not too shy or uncomfortable, but let's see if it can detect them.



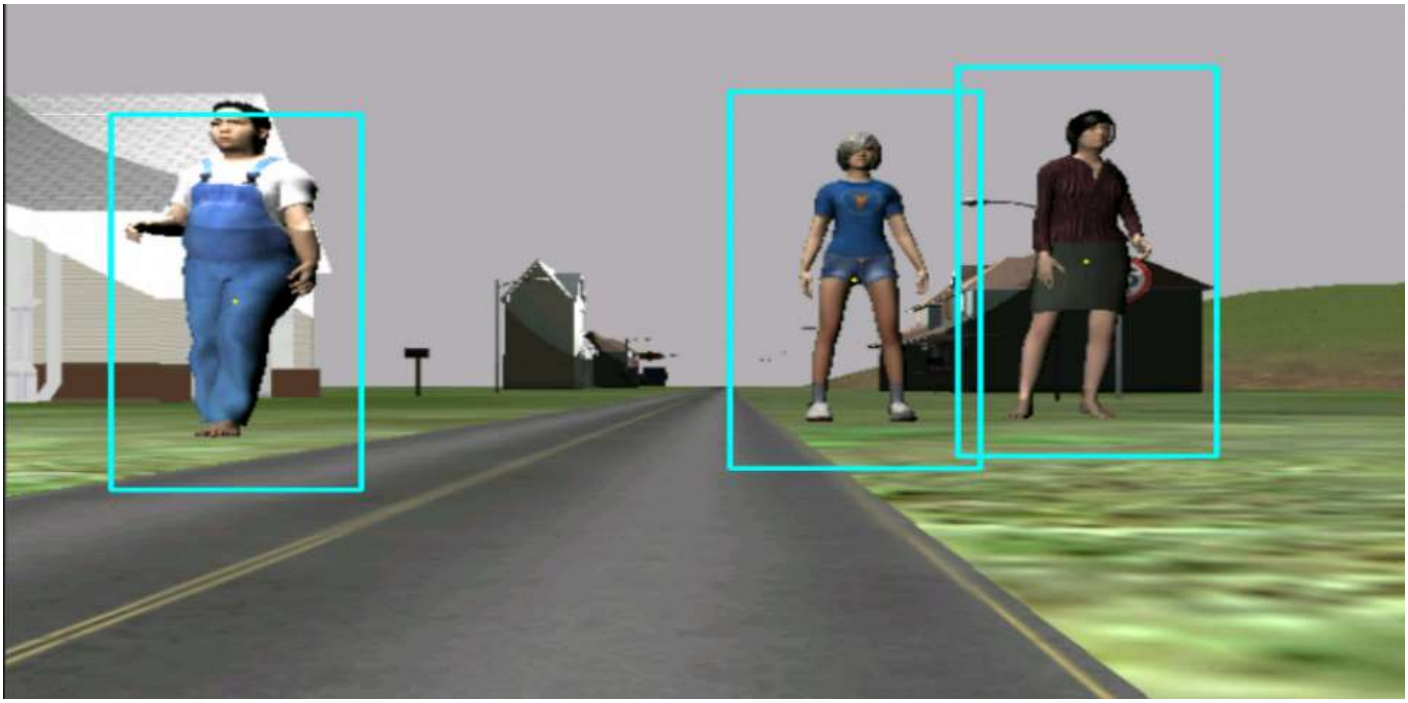
Step 1 Open Graphical Tools

Click on the Graphical Interface Icon:

Now, in a Web Shell, `execute` the following command:

In [ ]:
```
roslaunch project people.launch
```

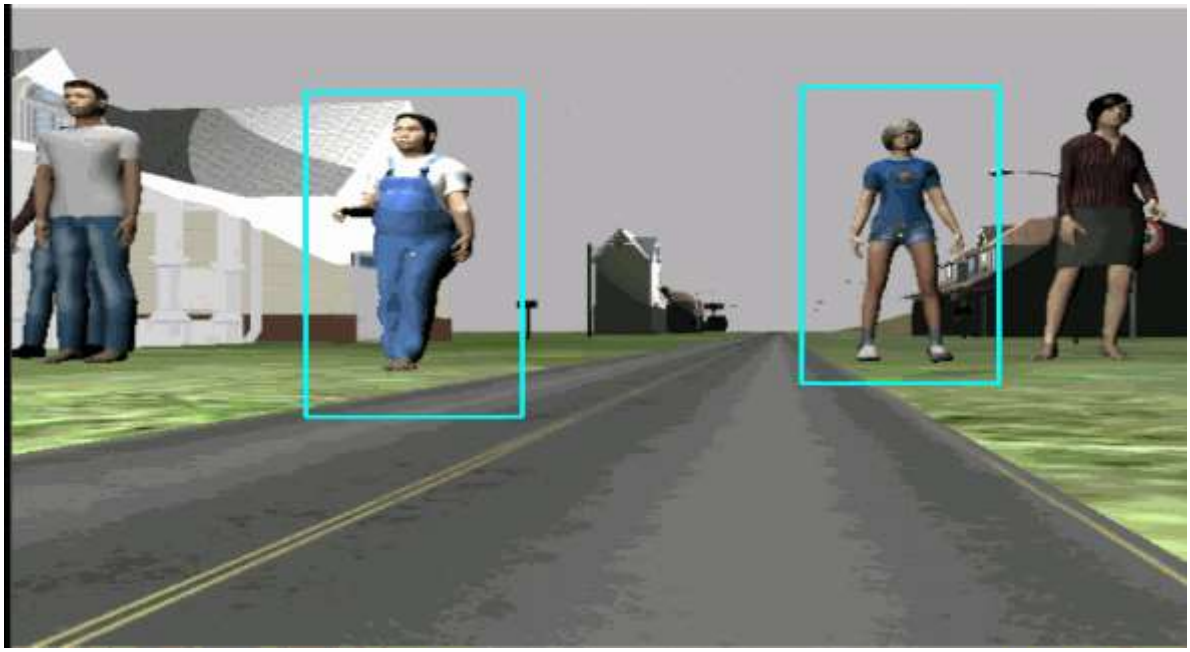
In the Graphical Tools window, you will see how the robot is detecting the people.



Step 2 Move the robot.

If you want to move ROSBot around in order to see how it will detect the rest of the people, just run the following command in another Web Shell.

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

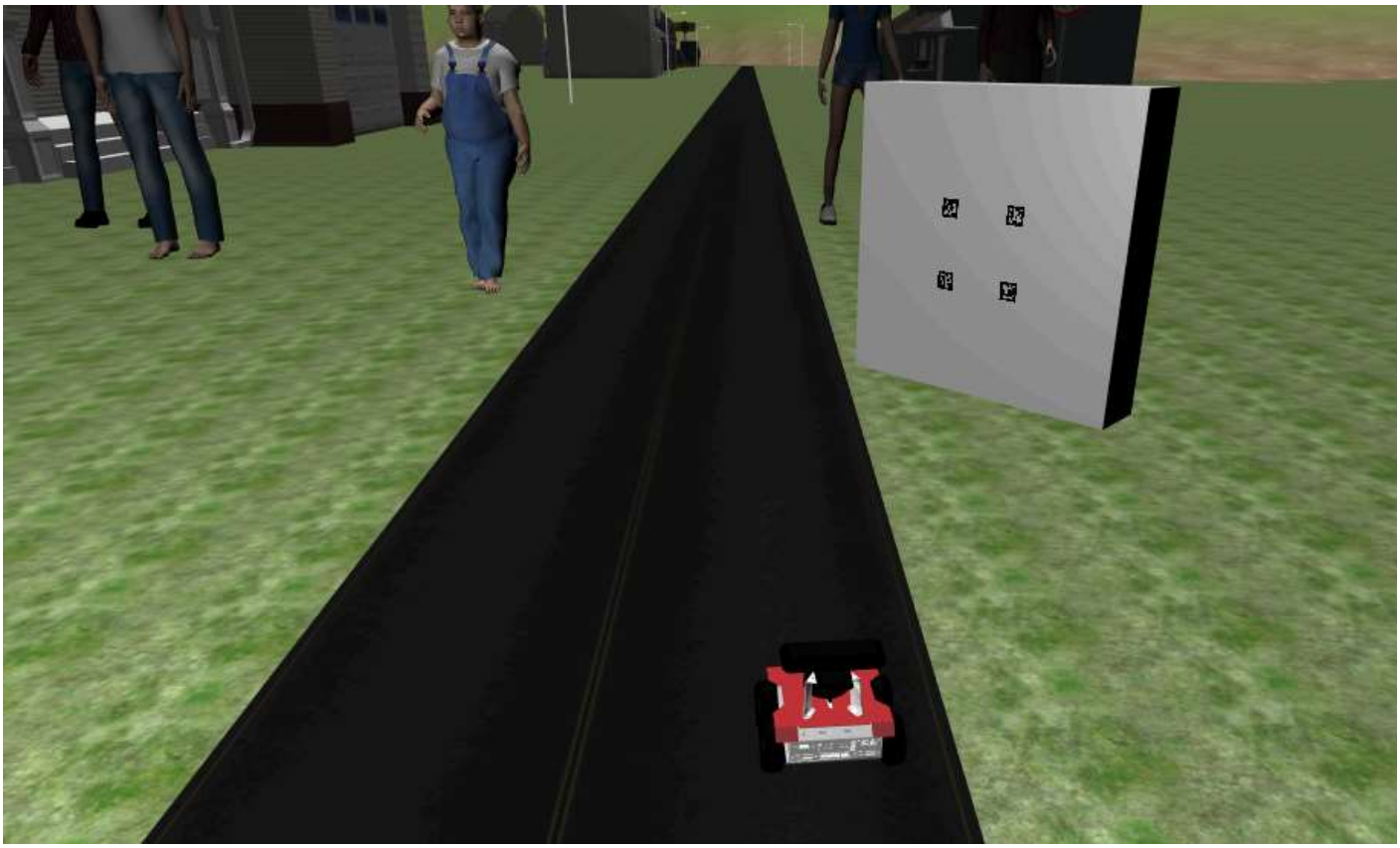


Although the Open CV algorithms are optimized for real images and not for simulations, we can see that they can also be applied. However, it will always work much better with real high resolution images.

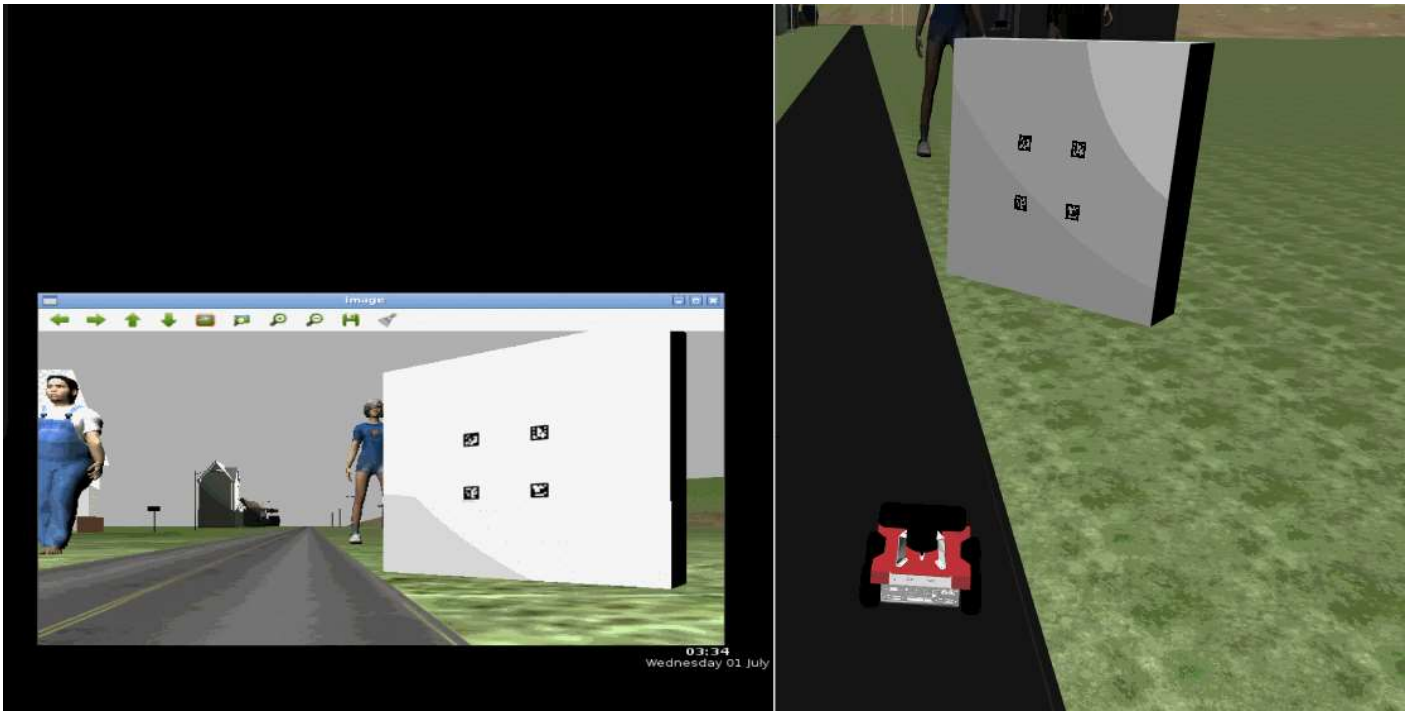
But, what will you learn in this course?

ARTags

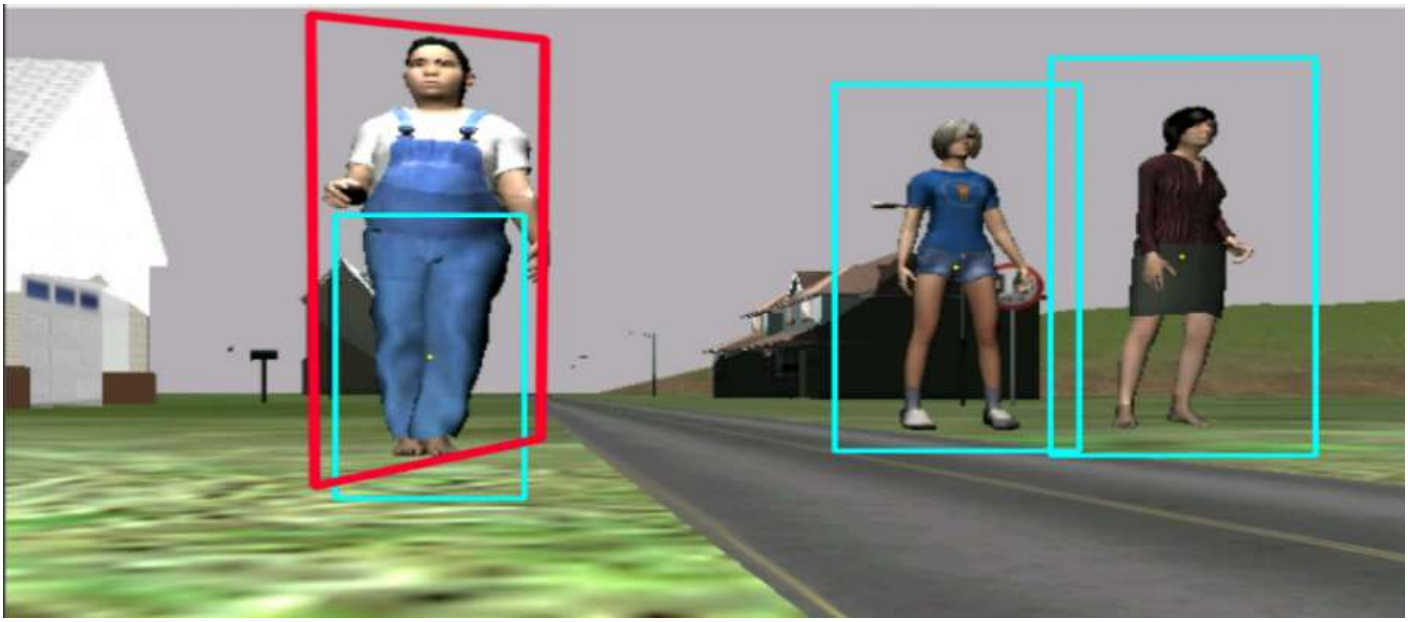
Imagine that in the simulation window, a wall like the one below appears:



What!? What is that? Well, those are ARtags. ARtags are a system that supports augmented reality. They can be used to facilitate the appearance of virtual objects within a real (or simulated) world. You can see an example in the below images and GIFs:

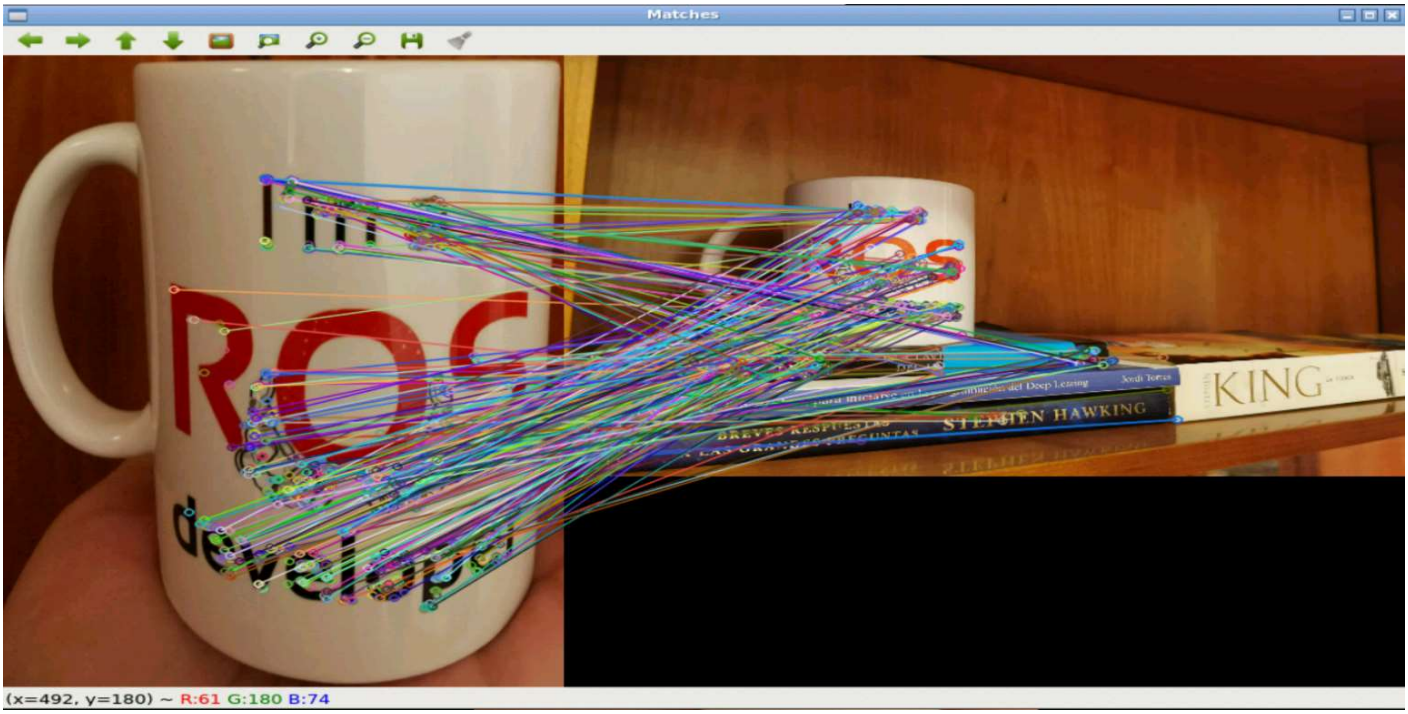


People Tracking and Object Detection

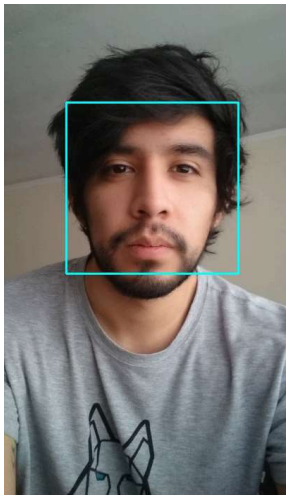


Well, in the previous demo, you have already tested a people tracking algorithm. But in this case, we are working with a simulation... So, if you use a real robot with a real camera, how will it be?

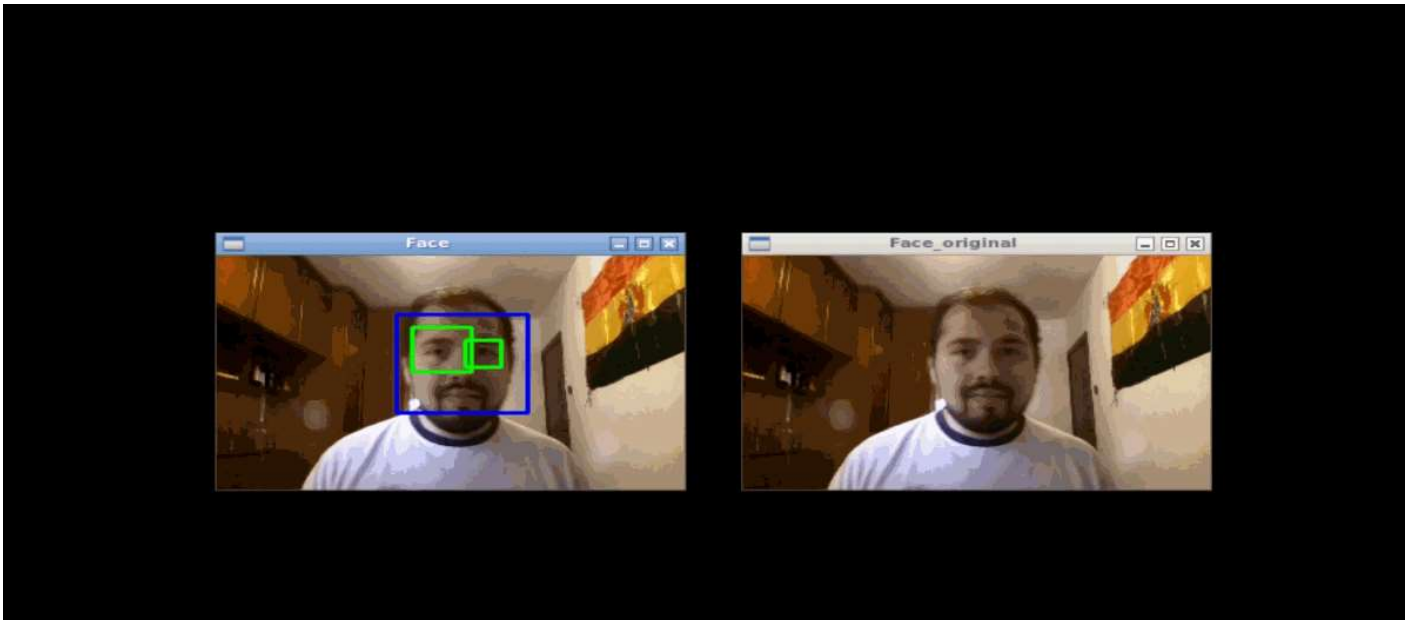
In this course, you will work at trying to detect real objects as well. For instance, you will try to detect a mug, or any ther object that you want.



Face Detection

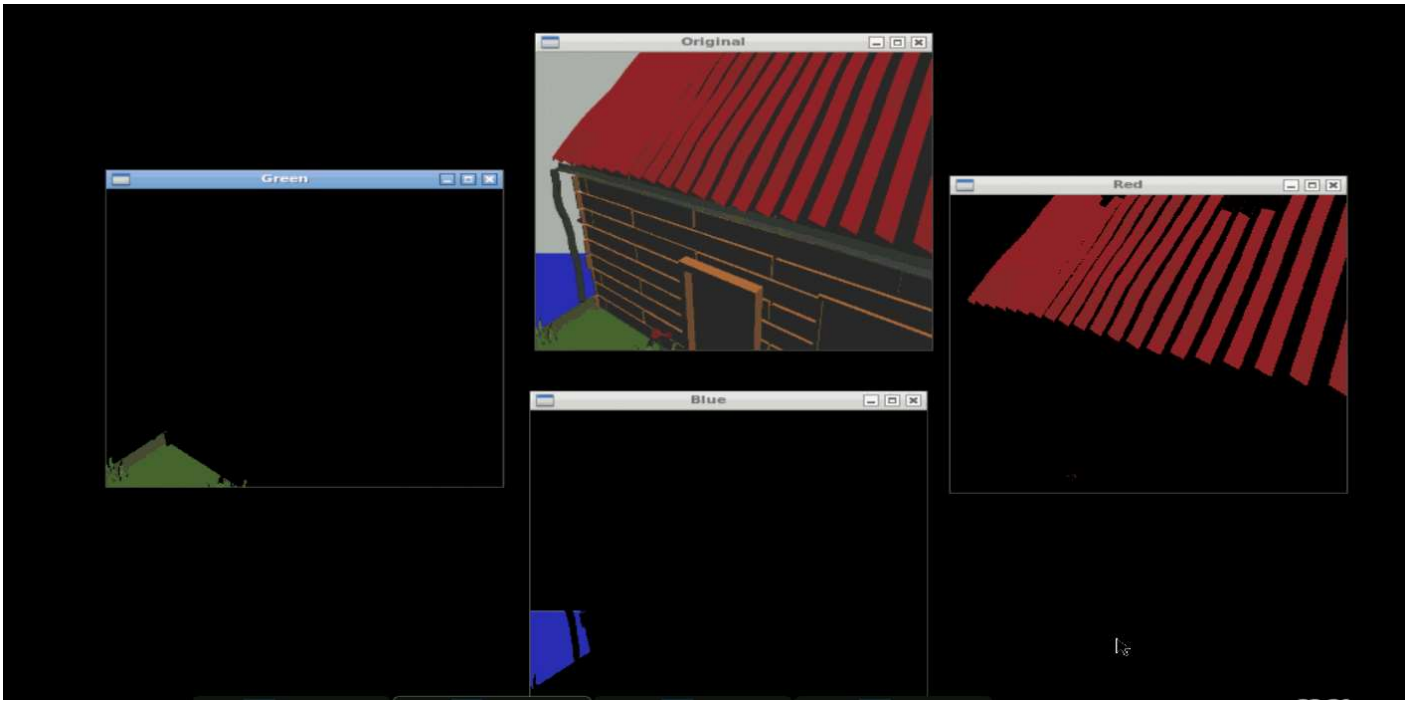


Also, you will see how to detect and track people's faces...

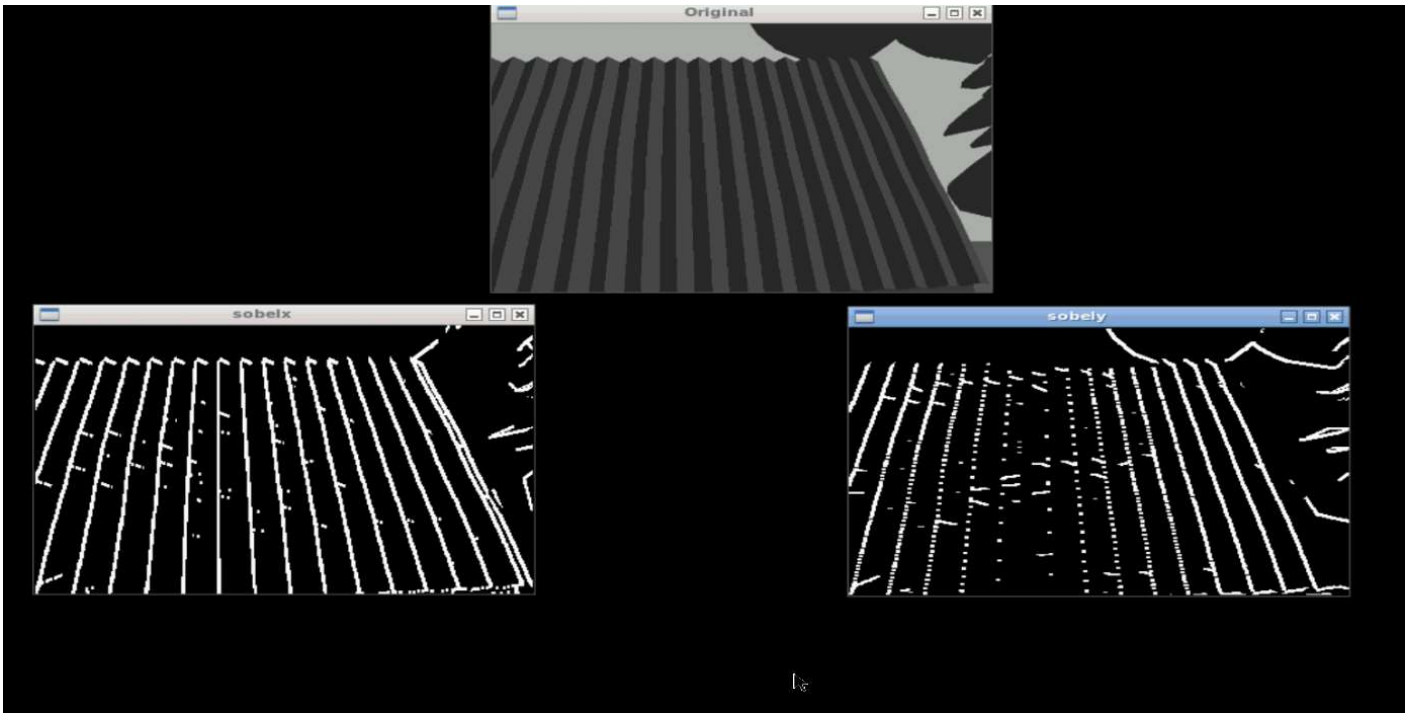


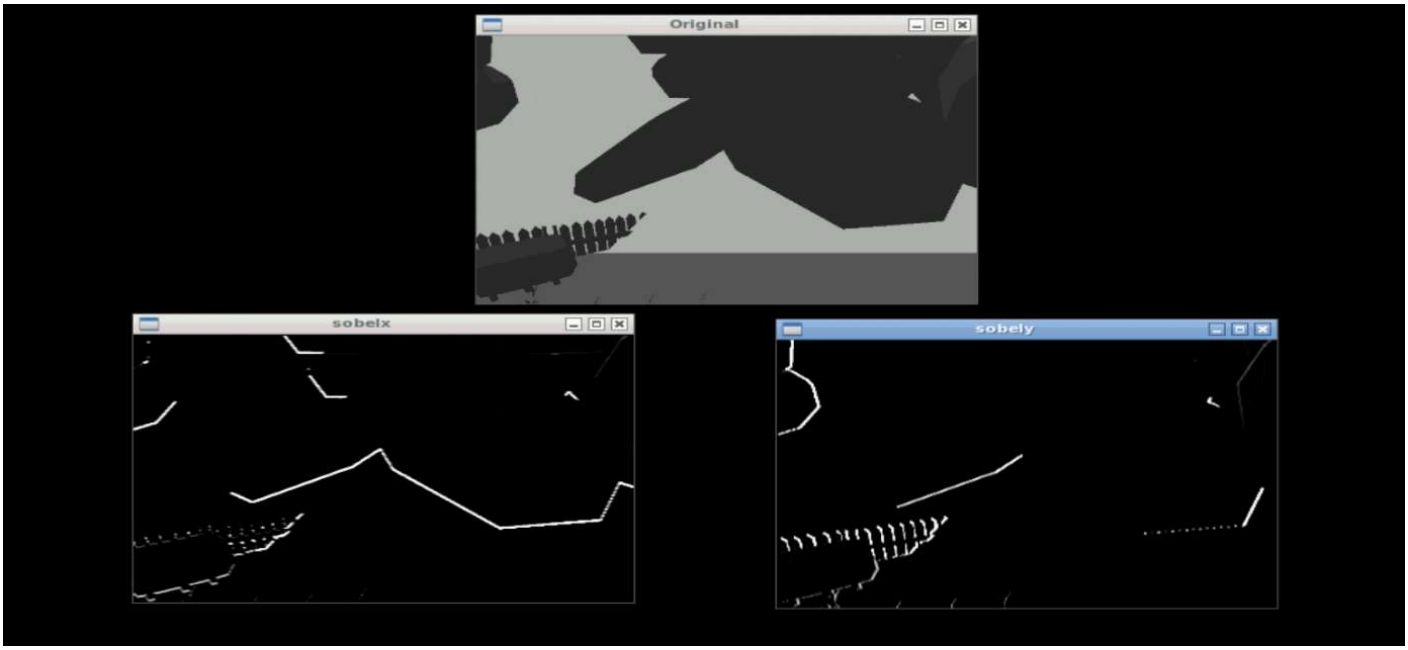
But you will learn about many other basic concepts as well...

Color Filtering

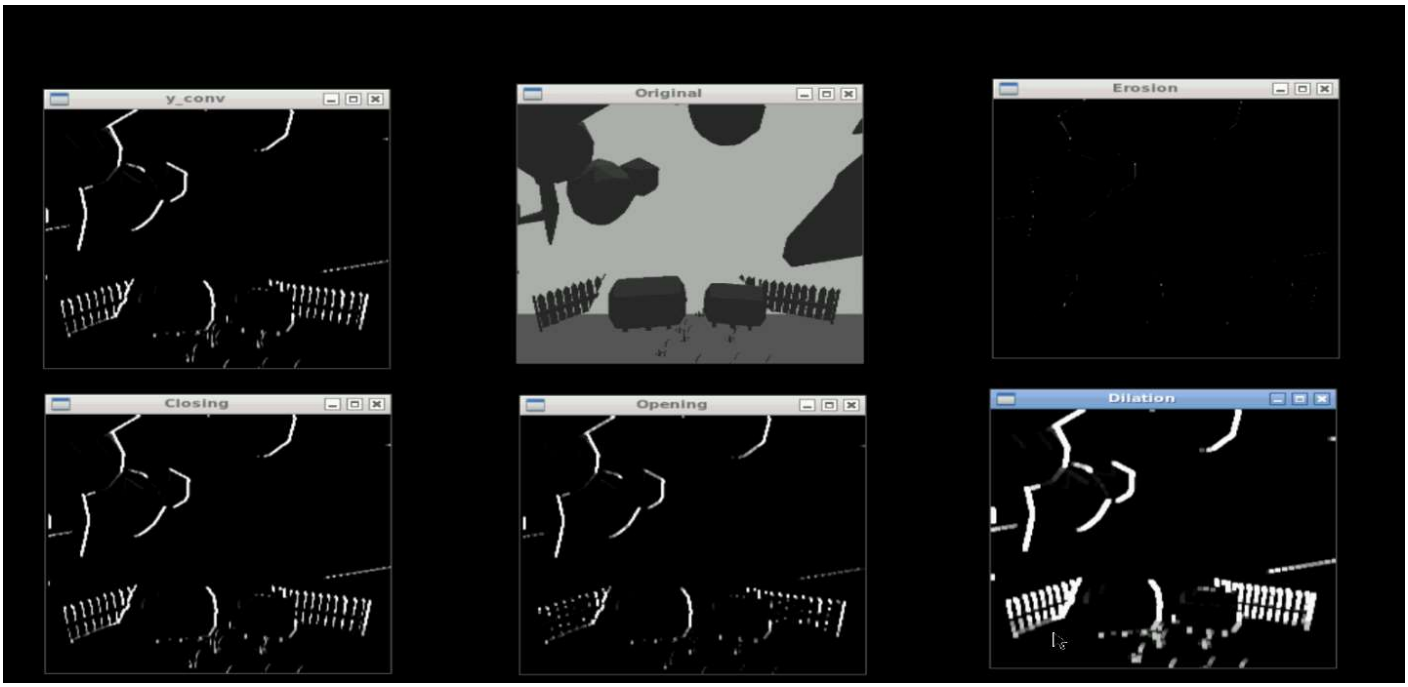


Edge Detection





Morphological Transformations



Summarizing... you will learn:

Unit 2: Computer Vision Basics

- cv_bridge
- Color spaces and color filtering
- Edge detection and a brief introduction to convolutions
- Morphological transformations

Unit 3: People related opencv functions:

- Face Detection (Haar cascades)
- People Detection and tracking (HOG)

Unit 4: Feature Matching

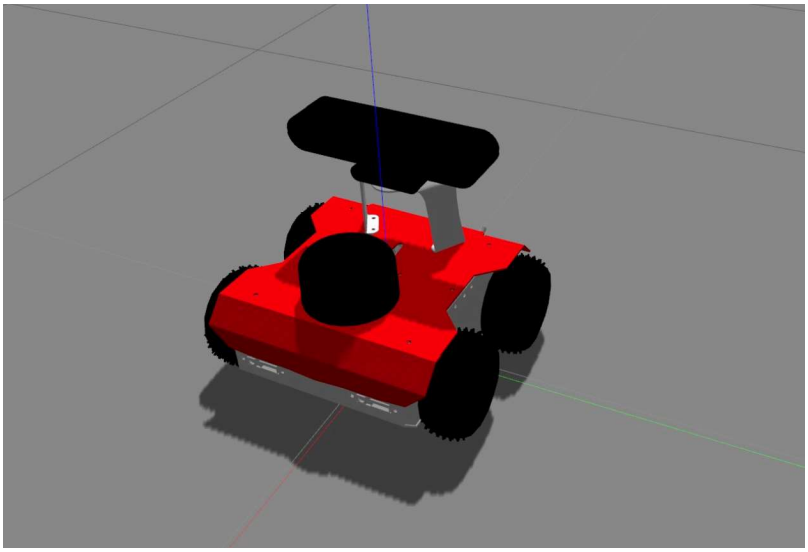
- Features from Accelerated Segment Test (FAST)
- Binary Robust Independent Elementary Features (BRIEF)
- Oriented FAST and Rotated BRIEF (ORB).

Unit 5: ARTags (Augmented Reality)

And how will you learn this?

All Robot Ignite Courses are based on hands-on learning because we have seen that it is the best and fastest way to learn about robotics. In this particular case, you will constantly compare using just python and opencv and integrating this code to ROS. That's why to learn all the above mentioned topics, you will work with different robots and environments. You will work with:

- ROSbot



- Hector Quadrotor

Requirements

It's essential that before starting this course, you know the following:

- ROS Basics: You need to know all the basics of ROS to be able to follow this course. If you don't, please do our **ROS Basics in 5 days Course** (https://www.robotigniteacademy.com/en/course/ros-in-5-days/details/)

- Basic knowledge of Python. If you don't have it, please do our **Python for Robotics Course** (https://www.robotigniteacademy.com/en/course/python-basics/details/)

- Some basic math knowledge, like arrays, etc.

Special Thanks

- This course was developed with the collaboration of the engineer Andres Guzman (https://www.linkedin.com/in/andres-guzman-a63757139/), specialist in the area of artificial vision.
- This course wouldn't have been possible without the knowledge and work of the ROS Community, OSRF, and Gazebo Team

Let's start this awesome course!