ROS Perception in 5 Days

Chapter 0:   Introduction

Congratulations! You're on the path to becoming a ROS Developer.

The demand for robotics developers is growing, but there are not enough developers with the knowledge to program robots. This is your chance to stand out among the world's engineers. If you jump on the robotics development bandwagon now, you will have a clear advantage over other developers.

(Further reading: Why programming robots has a big future (https://www.theconstructsim.com/become-robotics-developer/).)

0.1 Why Should You Take This Course?

When we think of robotics, we all think of robots that recognize people, follow them, can distinguish a tomato from a cup, and can move around like any other form of life on this planet.

This has been the primary battle in robotics since the 70s. In reality, it isn't very easy for a robot to tell the difference between a person and an object or know whether it's John or Sakura saying,"Hi". It takes a lot of work to perform just one of those tasks.

Fortunately, ROS is here to help. Thanks to its package system, you can use the work of countless other roboticists before you to have a robot do all those things in a relatively short time.

Learning ROS is NOT an Easy Task

We know from our own experience how difficult it is to learn ROS. **It requires a lot of time and effort**. Also, learning from just the ROS official documentation, slides, posts, or videos is frustrating because there is no opportunity for practice. This is not very helpful for a beginner and can even get you lost.

That is why we have created this specific program to help you learn ROS with more efficiency and depth.

This platform's primary purpose is to help you learn ROS fast in an easy, intuitive, and practice-based way. **Our method is called 30/70**. You will dedicate 30% of your time to theory and 70% of your time to practicing with simulated robots.

Let's see how this method works with your very first exercise.

0.2 Hands-On Right Now!

- Demo -

Working Example: PR2 Makes Yolo Object Detections

In this simulation, you can see the famous **PR2** robot in a scene with **two humans** and an **elephant**. You will launch a **YOLO object detection system,** and you will be able to detect humans and elephants in a scene through the RGB camera.

Execute the launch file to start the recognition through this command in the WebShell:

► **Execute in WebShell #1**

```
In [ ]:  DISPLAY=:1 roslaunch darknet_ros_3d darknet_ros_3d.launch
```
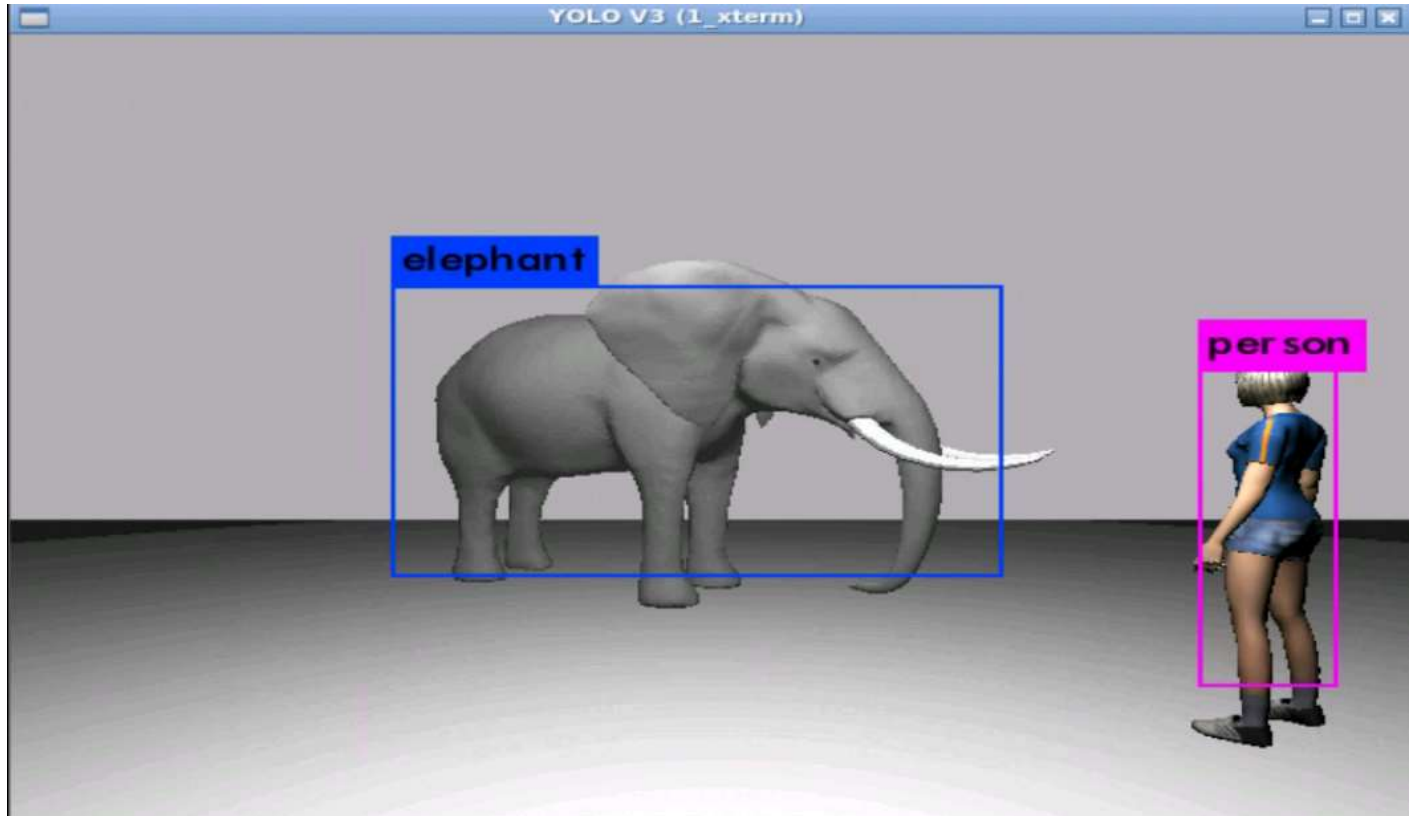
In approximately 30 seconds, you should have an output for the detections, similar to the image below:

Click on the Graphical Interface icon to see the detection as an image:

🖥️

You should see an image similar to the image below:

Move the PR2 Robot with your Keyboard

You can move the PR2 robot with your keyboard. **INCREASE SPEED WITH THE 'Q' KEY** until it has a value similar to the image below:

Use the command below to start keyboard control:

```
In [ ]:   roslaunch pr2_tc_teleop keyboard_teleop.launch
```

By executing this command, you can now move PR2 robot

IN THIS SIMULATION, the PR2 ONLY moves in translation, NO ROTATION.

TO MOVE SIDE TO SIDE YOU NEED TO KEEP the `SHIFT-KEY` pressed an dthen it will translate side to side and forwards and backwards.

ALSO: Please INCREASE THE SPEED wih the `Q-Key` untill at least 8.0 meters per second. Otherwise , the robo wont move at all.

The basic keyboard commands:

Key	Action
SHIFT+i	Move Forwards
SHIFT+,	Nothing
j	Strafe Left
l	Strafe Right
k	Stop
SHIFT+u	Move Forwards + Strafe Left
SHIFT+o	Move Forwards + Strafe Right
SHIFT+m	Move Backwards + Strafe Left
SHIFT+.	Move Backwards + Strafe Right
q	Increase Speed
z	Decrease Speed

Move the head of the **PR2** robot using the **rqt_gui** that you can see in the **Graphical Tools** window.

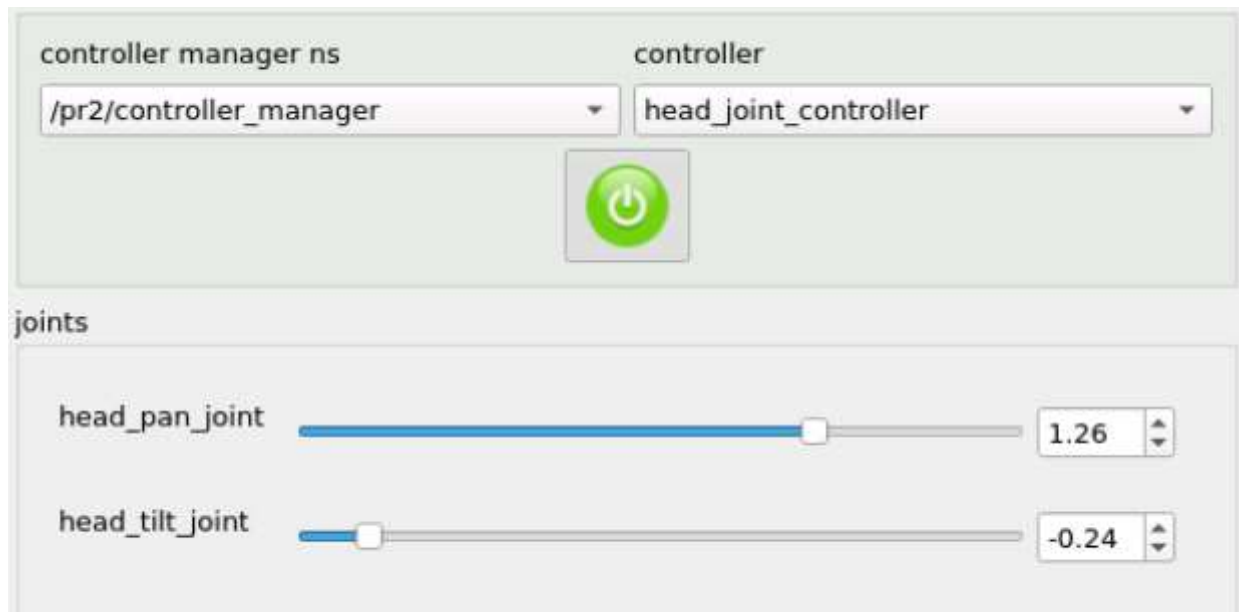You have to select form the dropdown menus this configuration:

Then click on the `RED ON Button` to activate the controllers with that configuration:
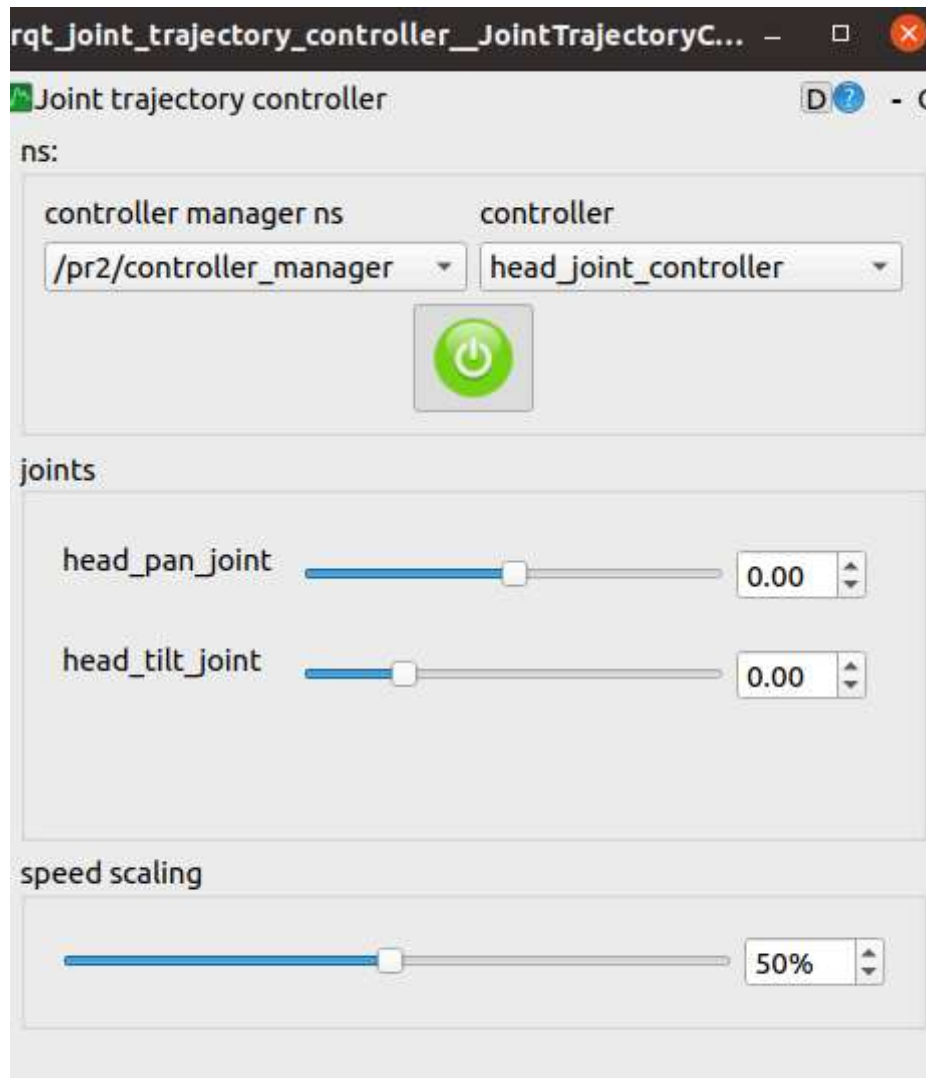
Move the slider named

- `head_pan_joint` to move the head of PR2 `LEFT` or `RIGHT`, depending if the values are `BIGGER THAN 0.0 Turn Head Left`, `SMALLER than 0.0 Turn Head Right`.
- `head_tilt_joint` to move the head of PR2 `UP` or `DOWN`, depending if the values are `BIGGER THAN 0.0 Turn Head DOWN`, `SMALLER than 0.0 Turn Head UP`.

For example with this configuration:

controller manager ns | controller
/pr2/controller_manager | head_joint_controller

joints

head_pan_joint ————————————————————[ ]———— 1.26

head_tilt_joint ———[ ]————————————————————— -0.24

You should see that the PR2 robot has turned its head left and up:

- End Demo -

In this course, you will learn to:

- Use OpenCV with ROS
- Make robots track objects by their color blobs
- Make robots navigate following floor lines with only an RGB camera
- Make robots detect human faces and track them
- Make robots recognize different faces
- Make robots track a person through a 3D environment
- Make robots recognize flat surfaces, like tables, where objects might be placed
- Make robots recognize objects and track them in a 3D space with point-cloud sensors

How will you learn all this?

Robot Ignite courses are based on hands-on learning because, from experience, it is the best and fastest way to learn robotics. For that reason, you will learn the topics mentioned above using different robots in various environments.
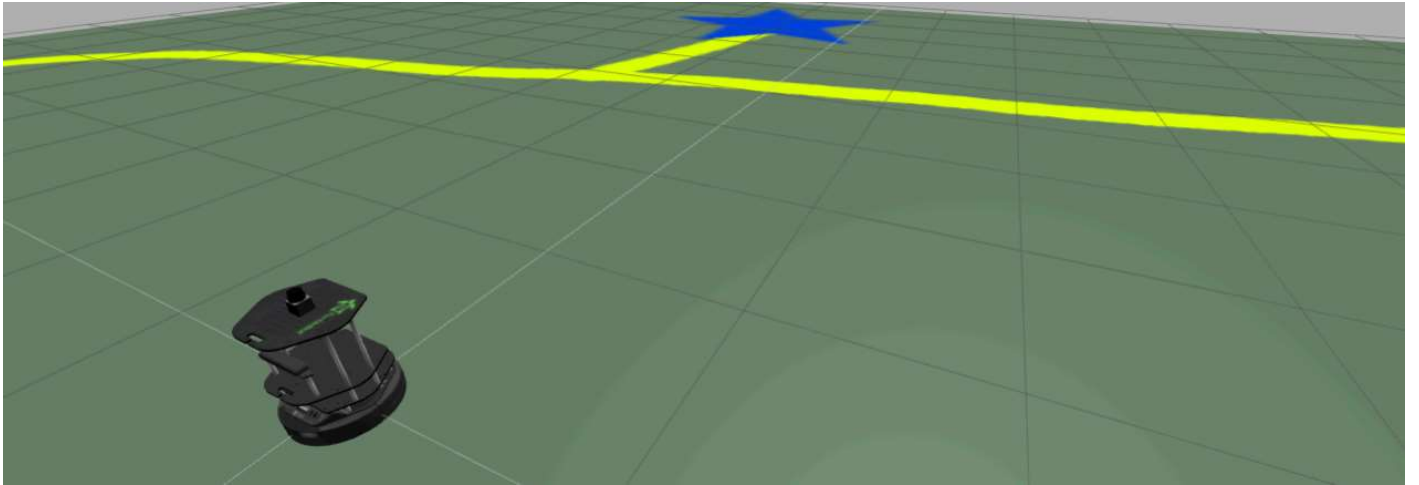
You will work with:

- The Mira robot for blob tracking
- TurtleBot for line following
- The Fetch robot for people and object tracking and detection in different environments
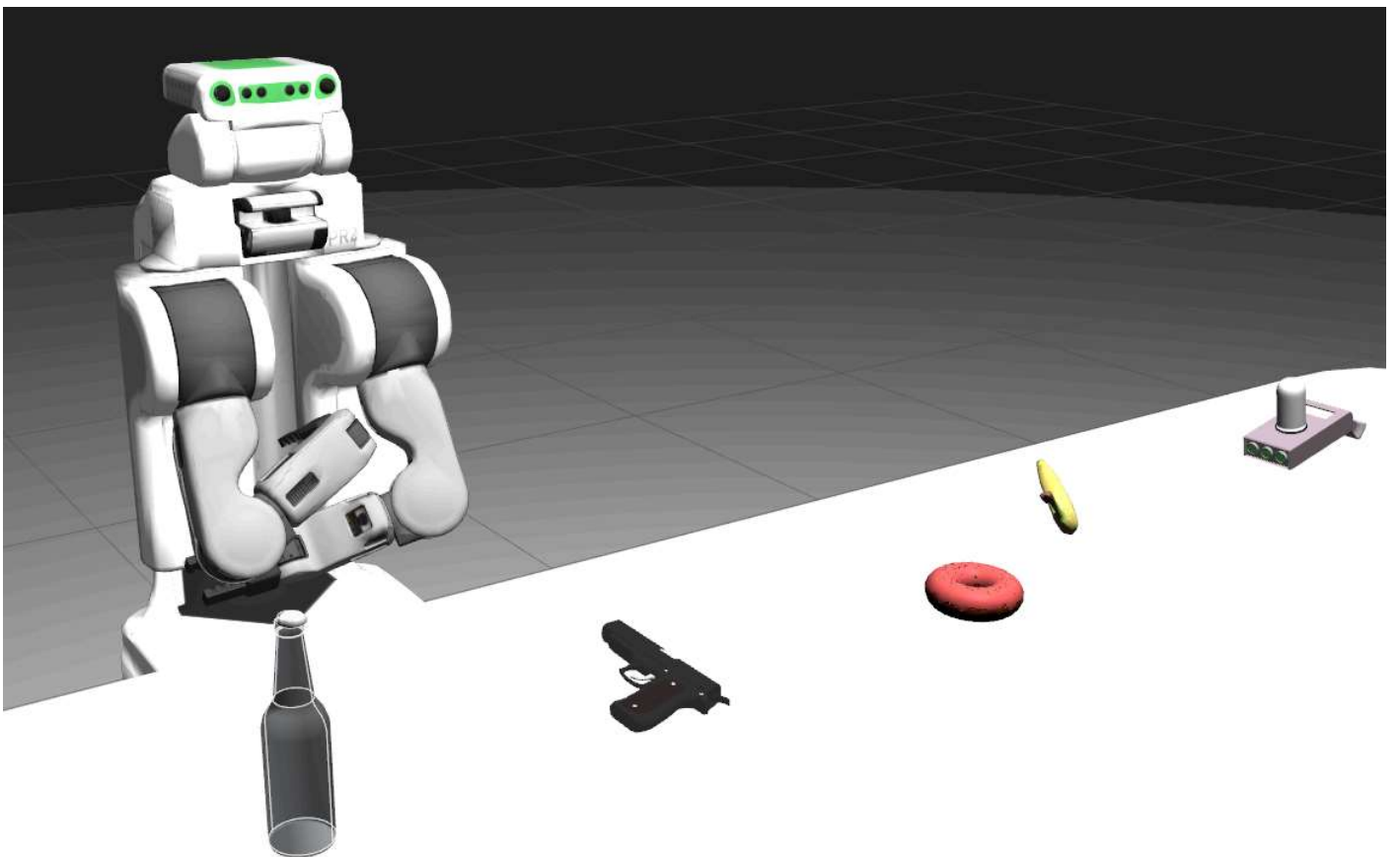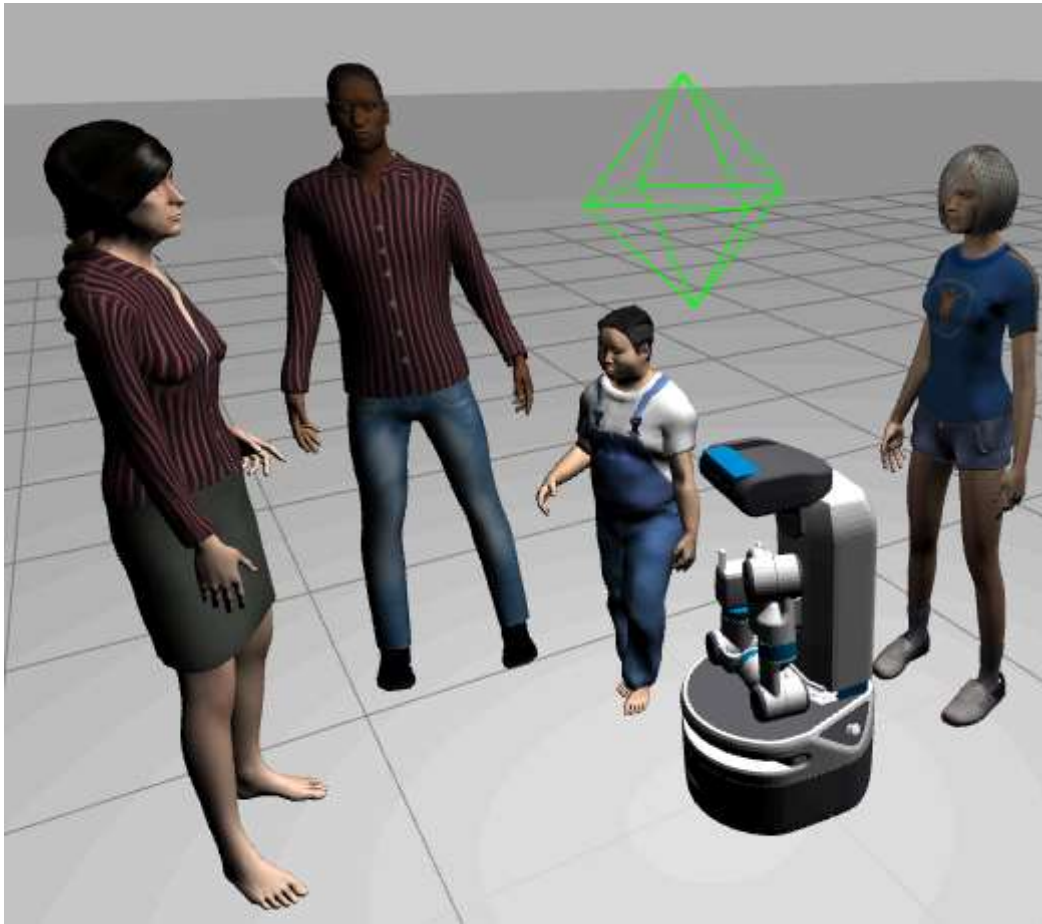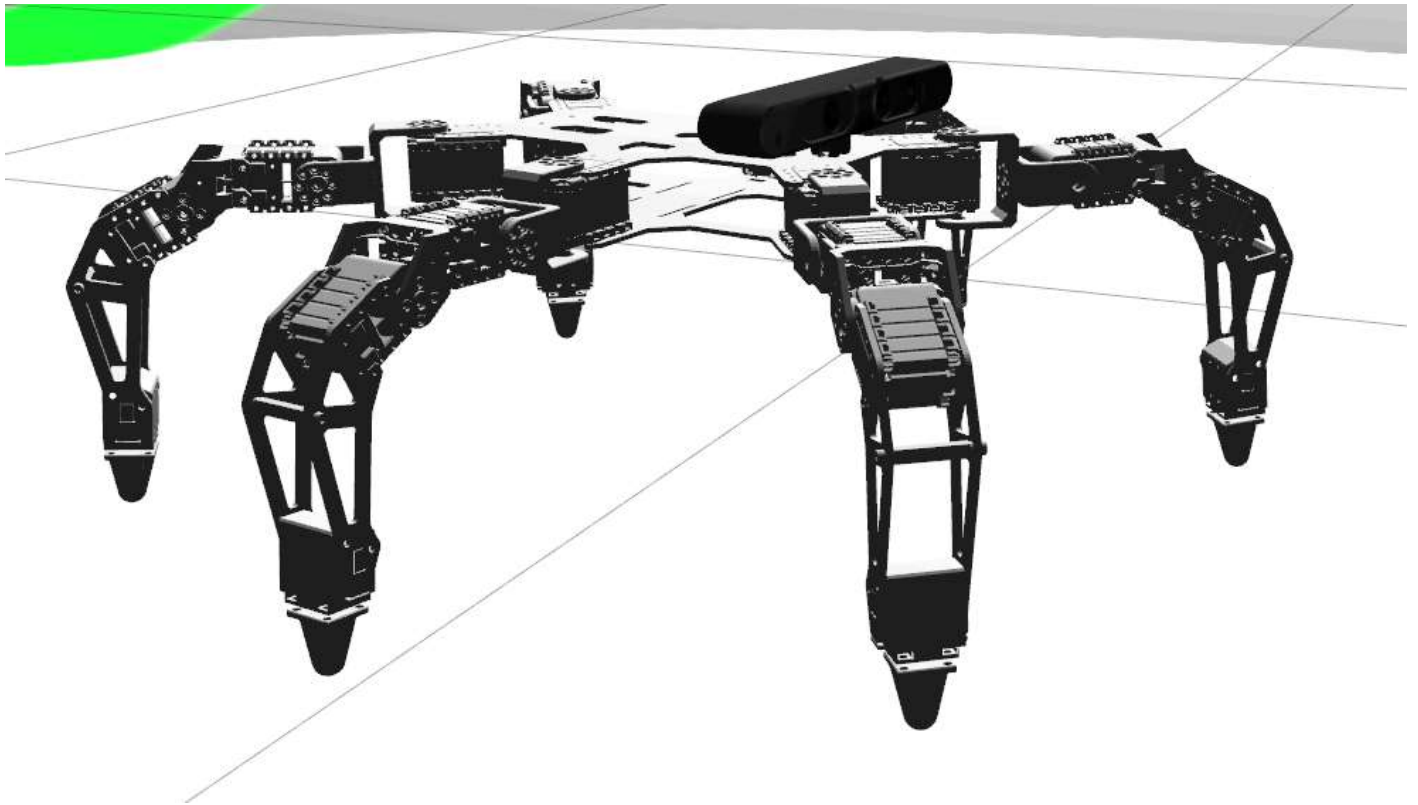- The Aibo robot for the final project

Mira
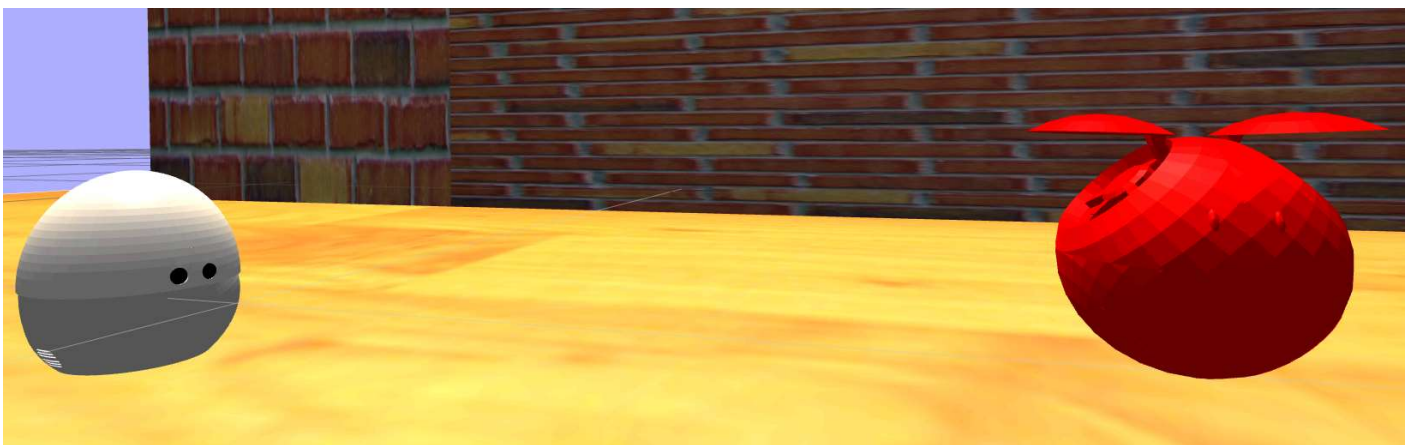
TurtleBot

PR2

Fetch

PhantomX Hexapod

0.3   What You Will Learn

In this ROS Perception In 5 Days course, you will focus 100% on learning what matters, avoiding unnecessary stuff. Also, you are going to practice with robots from the very beginning. No more only listening to long videos with lecturers or watching videos of a robot doing cool stuff without learning anything.

In this course, you will MAKE the robots do perception in matters of minutes.

Topics in this course:

- Blob tracking
- Line following with OpenCV
- Object detection using the expanded object detection package
- Object detection and localization in 3D space with deep learning models YOLO
- Human face tracking in 3D space
- Human face recognition
- People detection in a crowded environment and tracking in 3D space

Main points

- You will use different robots
- You will have access to our experts to ask them questions in the forum
- You will have our full encouragement and support
- You will practice -- a lot!

Requirements

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

- ROS basics: You need to know ROS basics to complete the course. If you don't know ROS basics, take our **ROS Basics in 5 Days course.**
- Basic knowledge of **Python**.

Special Thanks

We want to thank the following individuals and organizations:

- Blob tracking built upon amazing work from tizianofiorenzani (https://github.com/tizianofiorenzani/ros_tutorials/blob/master/opencv/src/find_ball.py) and jrosebr1 (https://github.com/jrosebr1/imutils/blob/master/bin/range-detector).
- Extended Object Recognition ROS package from tizianofiorenzani (https://github.com/Extended-Object-Detection-ROS/extended_object_detection), created by MoscowskyAnton (https://github.com/MoscowskyAnton).
- DarkNet YOLO package yolov4-for-darknet_ros (https://github.com/Tossy0423/yolov4-for-darknet_ros), created by Tossy0423 (https://github.com/Tossy0423).
- YOLO 3D gb_visual_detection_3d (https://github.com/IntelligentRoboticsLabs/gb_visual_detection_3d), created by Francisco Martin (https://gsyc.urjc.es/~fmartin/).
- Pedestrian Simulation (https://github.com/srl-freiburg/pedsim_ros) and People Tracking (https://github.com/spencer-project/spencer_people_tracking)
- Face Recognition module creators, https://github.com/ageitgey (https://github.com/ageitgey), https://github.com/ageitgey/face_recognition (https://github.com/ageitgey/face_recognition).
- Alonso Martinez (https://twitter.com/alonsorobots) for his amazing Mira robot. We were delighted to generate a Mira simulation for this course.
- The Phantom X simulation (https://github.com/HumaRobotics/phantomx_gazebo), used for the project.
- The MakerHuman Software (http://www.makehuman.org/), used for the creation of people in the simulations.
- This course would not have been possible without the ROS Community (http://www.ros.org/), OSRF (https://www.osrfoundation.org/), and Gazebo Team's (http://gazebosim.org/) knowledge and work.

It will be challenging, but with your perseverance and our guidance, you'll succeed and **become one of the few robotics developers in the world!**

The robots of the future are waiting for you :). Welcome to the new adventure!

So Get Started on Perception with ROS!