Robot Environment Interaction Using TurtleBot

Andreia Rodrigues - Eduardo Leite - Francisco Queirós - Vincent De Clercq

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Topics

- Introduction
- Architecture
- Wall Following
- Line Following
- Object Following
- Object Avoidance
- Conclusions

Topics

- Introduction
- Architecture
- Wall Following
- Line Following
- Object Following
- Object Avoidance
- Conclusions

Introduction

- Technology has a big impact on our quality of lives
- Nowadays one of the ways it achieves that is through robots
- In recent years robots evolved from simple mechanical task performers into more intelligent machines capable of performing complex tasks
- One of the reasons why robots are capable of now doing such tasks, is their increasing level of autonomy
- One of the forms in which this autonomy comes is through autonomous navigation

Introduction

And so our goal was to explore some of the most common navigation problems:

- The ability to follow a wall
- The ability to follow a line
- The ability to follow an object
- The ability to avoid an object

Topics

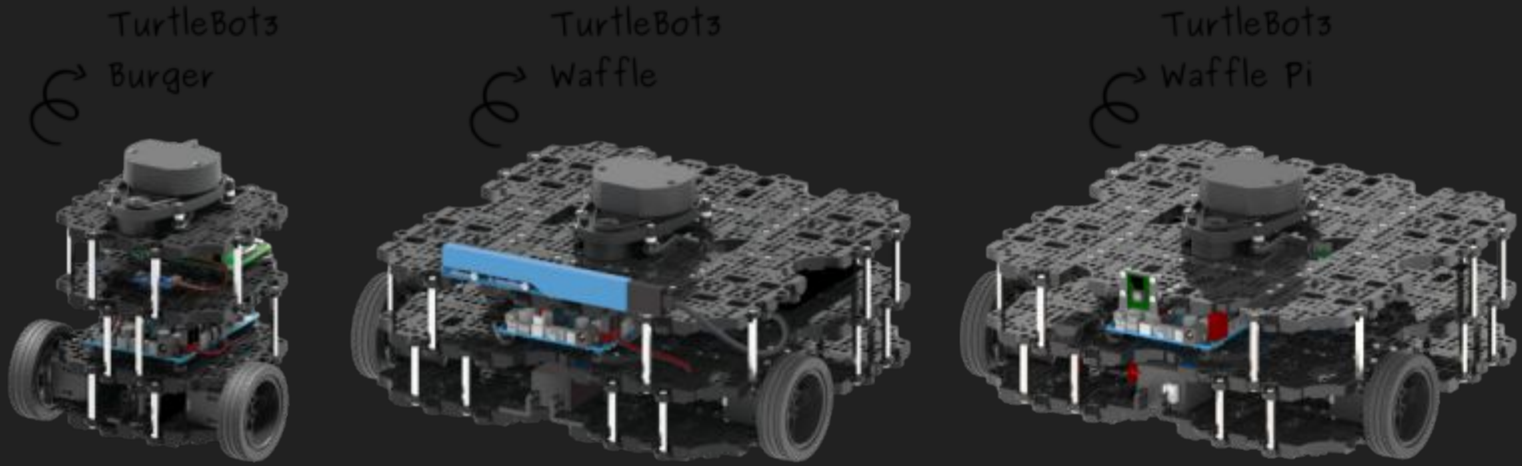
- Introduction
- Architecture
- Wall Following
- Line Following
- Object Following
- Object Avoidance
- Conclusions

Architecture

In order to explore these techniques we had to develop our own application and to do so we used the following set of technologies:

- Robot Operating System (ROS): A framework for building robots
- Turtlebot3 Waffle: A ROS-based mobile robot
- OpenCV: Computer vision and machine learning software
- C++: Programming language compatible with ROS and OpenCV

Architecture



Architecture

Our application starts by initializing ROS and subscribing to all the necessary topics and then enters a very simple loop with two phases:

- Sensor/Callback Phase:
  - Initialized with Spin Once (Single Threaded Spinning)
  - Call every waiting sensor callback and modifies the internal state
- Update Phase:
  - Calls the appropriate update function for the current mode
  - Runs a safety method that stops the robot if it’s in danger

Topics

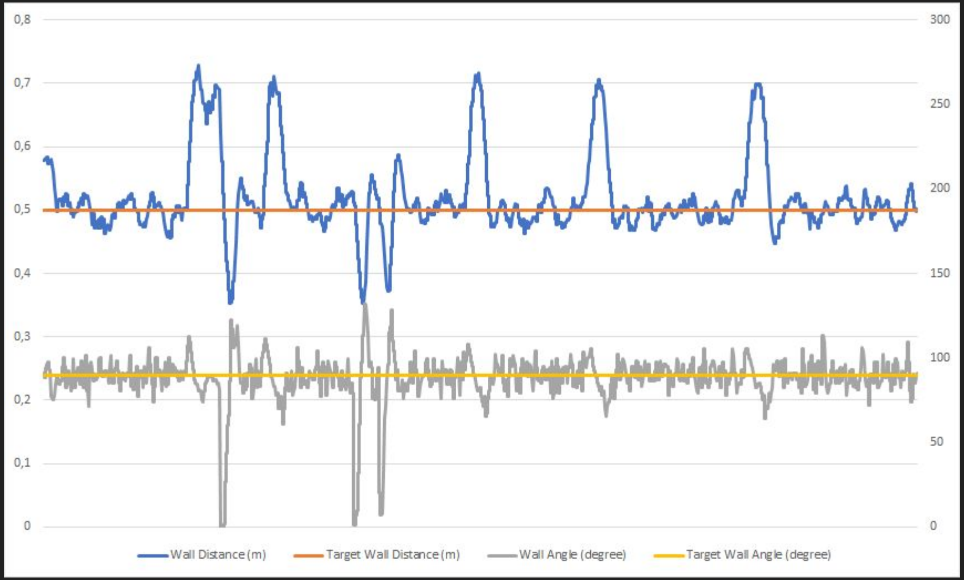
- Introduction
- Architecture
- Wall Following
- Line Following
- Object Following
- Object Avoidance
- Conclusions

Wall Following

- Constant distance
- Mapping, moving around stationary object
- LIDAR
- Choose closest side
- Imaginary parallel line
- $\omega = vk(\sin \alpha - (\sin \beta * ((d-T)/T))$

Wall Following

- Tested in real and simulated environments
- Satisfactory
- Exact distance fluctuates due to corners
- More aggressive in turns, more cautious when approaching wall



Wall Following

Topics

- Introduction
- Architecture
- Wall Following
- <u>Line Following</u>
- Object Following
- Object Avoidance
- Conclusions

Line Following

A line following robot is as the name suggest, a robot that can autonomously follow colored lines. This ability has many real life application, such as:

- Allowing a robot in a factory to move between workstations without human interaction
- Define a path in which a robot can travel to autonomously carry something from a sources to a destination
- And much more

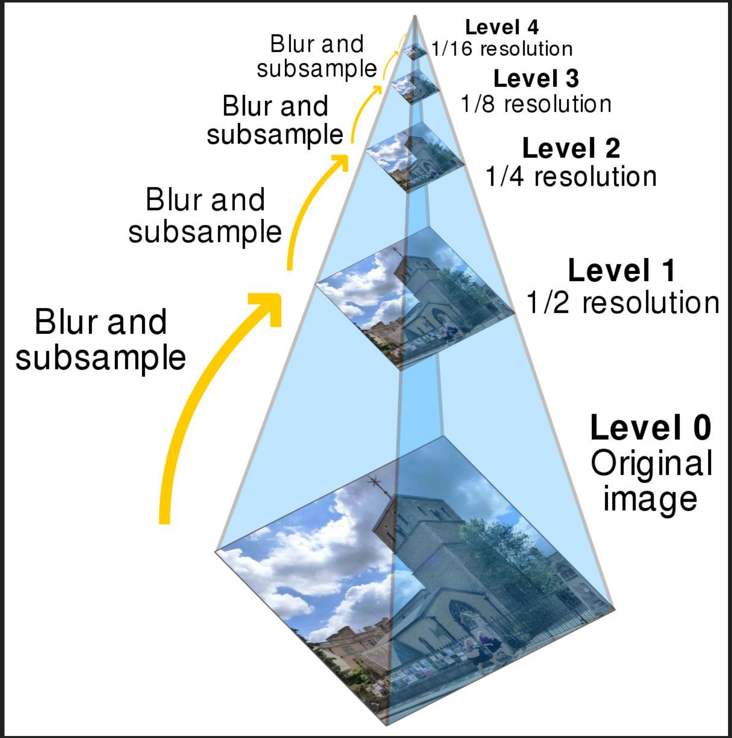
Line Following

In order to make our own robot capable of following a line we have developed a routine that used the image captured by the color camera. We also made use of the code for wall following so that the robot could go around obstacles. And so the steps in our routine are:

- Application of a gaussian filter in order to reduce noise
- Downsampling of the image using a gaussian pyramid representation technique, reducing processing time

Line Following

Gaussian Blur



Pyramid Down

Line Following

- Conversion to HSV, for better color space to detect objects of a certain color
- Conversion to a true/false matrix using the target HSV interval for the line in question
- Masking the top, left and right sides of the image, so that the robot would only consider what's directly in from of it
- Calculating the image moments
- Using the moments to get the area and the centroid of the object
- Posting the new velocity values accordingly

Line Following

Moments summarize a shape given image $I(x,y)$:

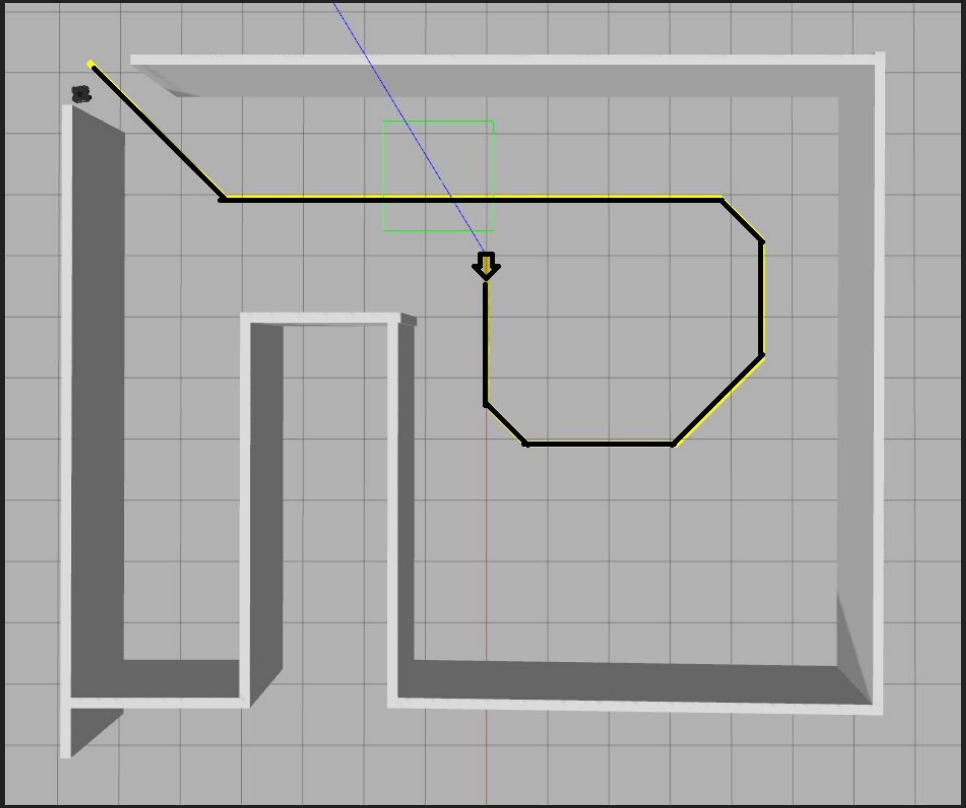
$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y)$$

Central moments are translation invariant:

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q I(x,y)$$

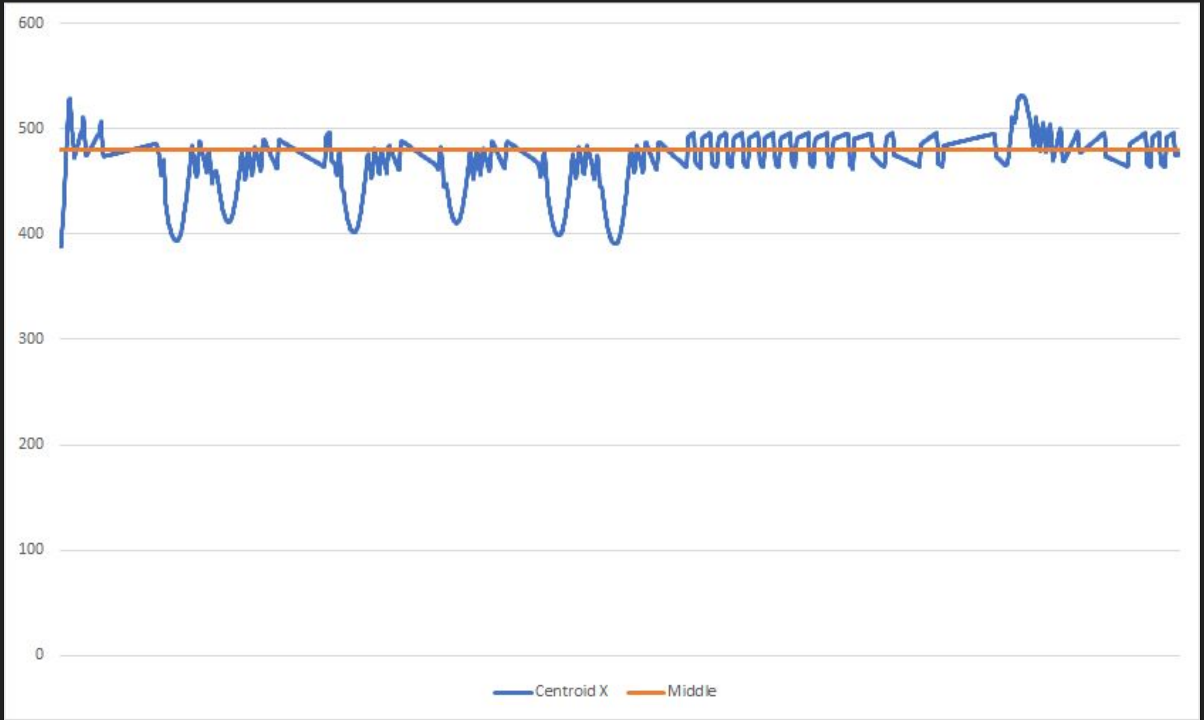
$$\bar{x} = \frac{M_{10}}{M_{00}} \quad \bar{y} = \frac{M_{01}}{M_{00}}$$

Line Following

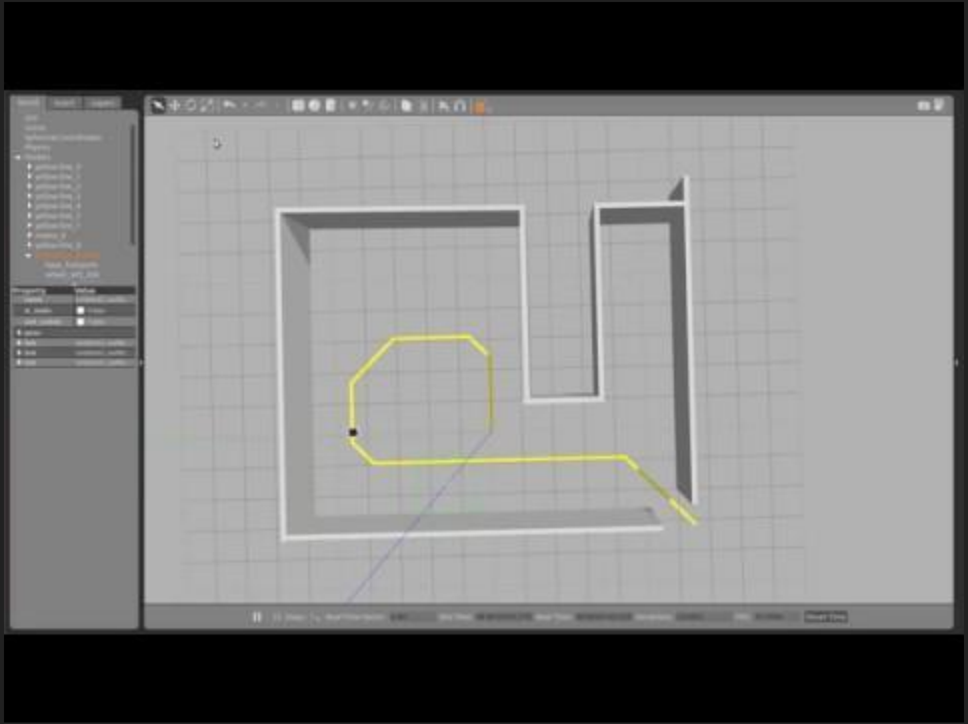


Line Following

- The robot was able to follow a line from start to finish
- It would sometimes deviated slightly from the center



Line Following



Topics

- Introduction
- Architecture
- Wall Following
- Line Following
- Object Following
- Object Avoidance
- Conclusions

Object Following

- Detect an object according to a target color.
- After object is detected, move forward and turn towards the object.
- Use the color camera for this
- Process is similar to the Line Following mode (blur, color detection, mask part of the image, detect centroid of the pixels and decide accordingly).

Object Following

- Tested in real environment
- Robot followed an object by turning towards the object

Object Following

Topics

- Introduction
- Architecture
- Wall Following
- Line Following
- Object Following
- <u>Object Avoidance</u>
- Conclusions

Object Avoidance

- Detect and move away
- Obstacle avoidance, safety
- LIDAR
- Solution:
  - 1) Find object within predefined distance
  - 2) Turn away: measure angle, rotate accordingly
  - 3) Object in front? Avoid new object

    Object in back? Keep avoiding

Object Avoidance

- Tested in real environment
- Turn away: successful
- Start moving away: successful
- If object follows robot: keep turning and moving away: successful

Object Avoidance

Topics

- Introduction
- Architecture
- Wall Following
- Line Following
- Object Following
- Object Avoidance
- Conclusions

Conclusions

- Seemingly simple, yet challenging
- Sensor noise, computation time
- Satisfactory outcome
- Improvements:
    - Parameter optimization (distances) through more testing
    - Edge cases, particularly in real life
    - More functionalities

Robot Environment Interaction Using TurtleBot

Andreia Rodrigues - Eduardo Leite -
Francisco Queirós - Vincent De Clercq

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO