Mobile Robot Control 2020: Tutorial lecture #2
Object Detection

APRIL 2020

Marzieh Dolatabadi

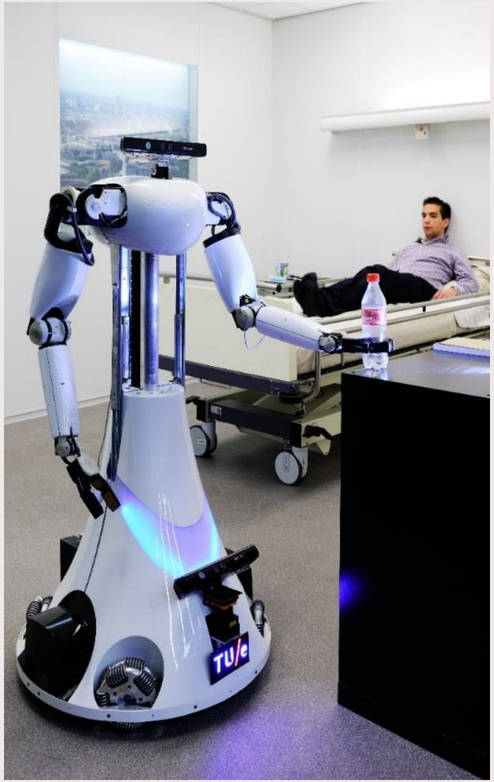
Department of Mechanical Engineering

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Motivation

Goal : How we can detect objects and features?

- What are objects and features?
- What is input data?



Motivation

Features :

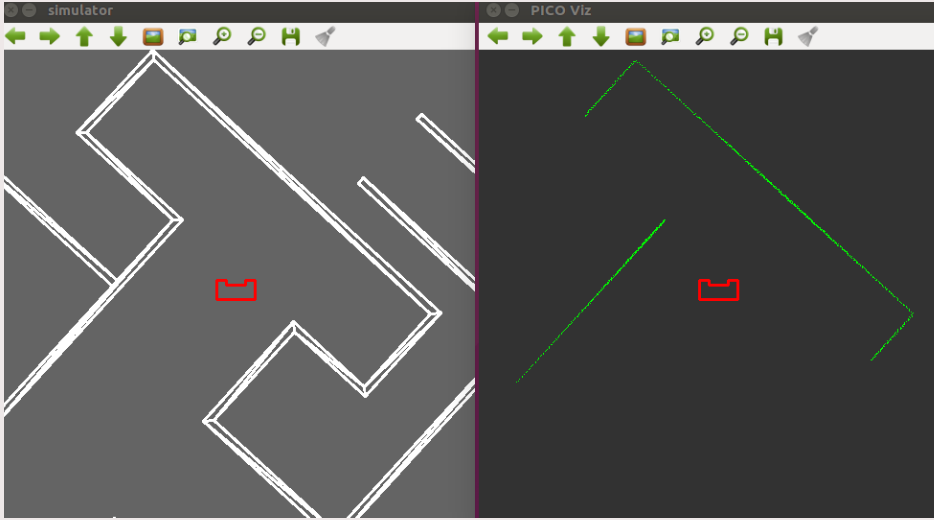
- Line
- Edge
- Corner
- Room
- Door

Objects :

- Moving and static clutter object

Input :

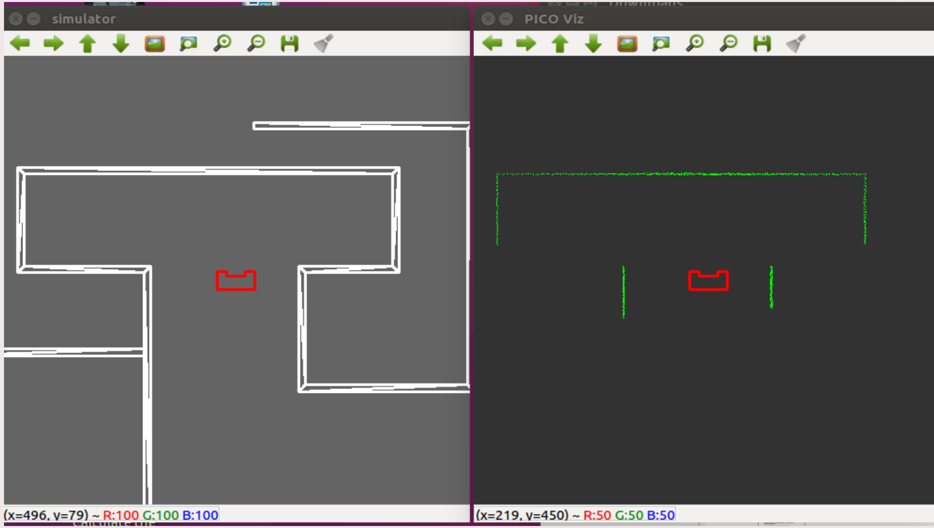
Laser Data : Measures distance to a target by illuminating the target with light and measuring the reflected light with a sensor.



Input data:

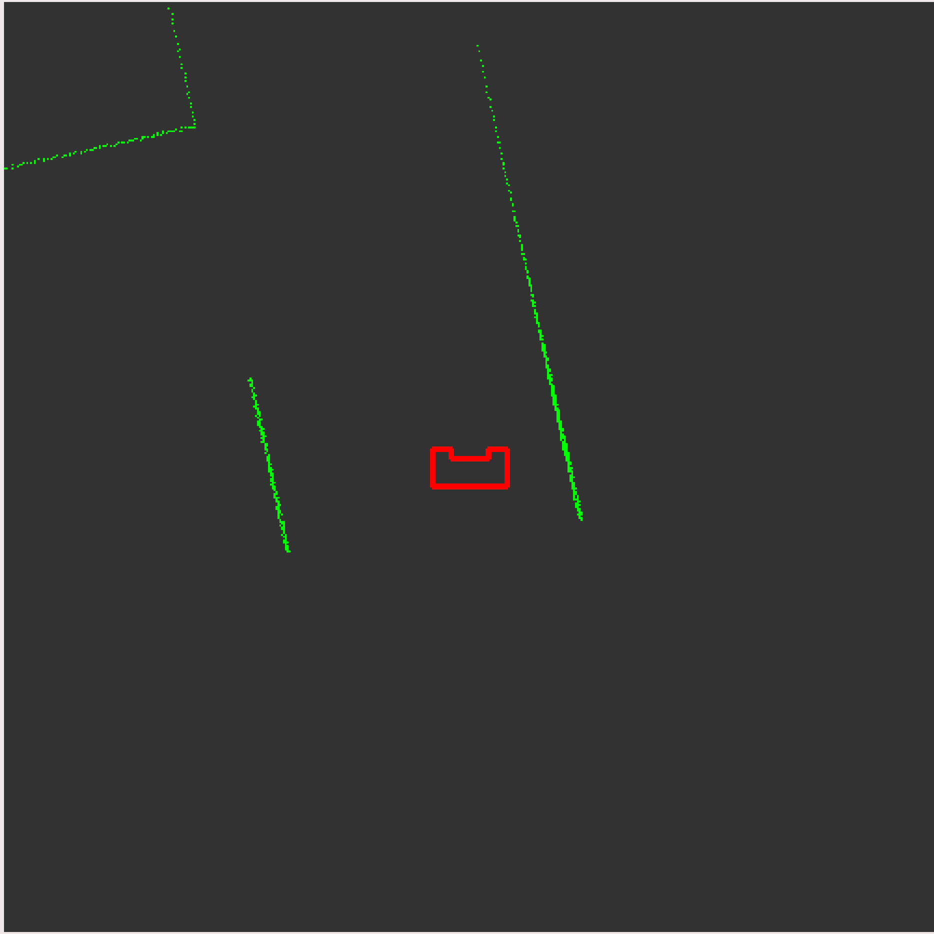
Question :

- What does laser data look like?
- How can we use laser data?


simulator
(x=496, y=79) ~ R:100 G:100 B:100
PICO Viz
(x=219, y=450) ~ R:50 G:50 B:50

OUTLINE

- LaserData struct
- Line extraction
- Features
- Segmentation
- Vision library

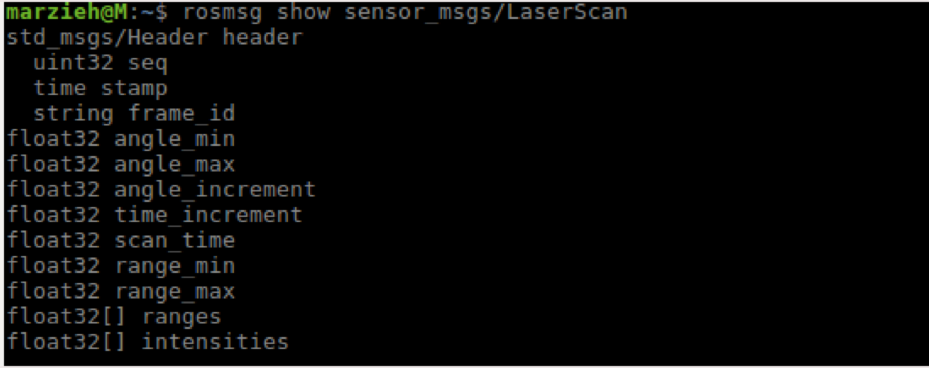


LaserData struct

1 – Open a Terminal (Ctrl - Alt + T)

2-

rosmsg show sensor_msgs/LaserScan

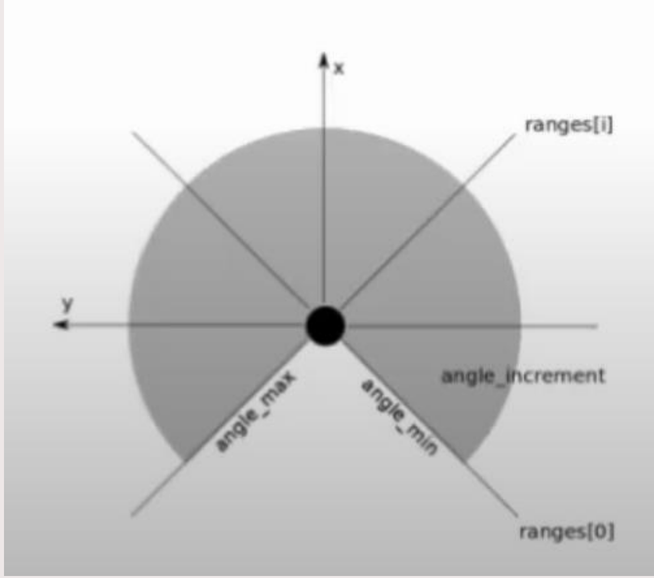


```
marzieh@M:~$ rosmsg show sensor_msgs/LaserScan
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
float32 angle_min
float32 angle_max
float32 angle_increment
float32 time_increment
float32 scan_time
float32 range_min
float32 range_max
float32[] ranges
float32[] intensities
```

LaserData struct

The *LaserData* struct is defined as follows:

- **Range** = the smallest and largest measurable distances
- **Angle** = the angle of the first and last beam in the measurement.
- ***Angle increment*** = the angle difference between two beams.
- ***Timestamp*** = specifies at which point in time the data was measured.

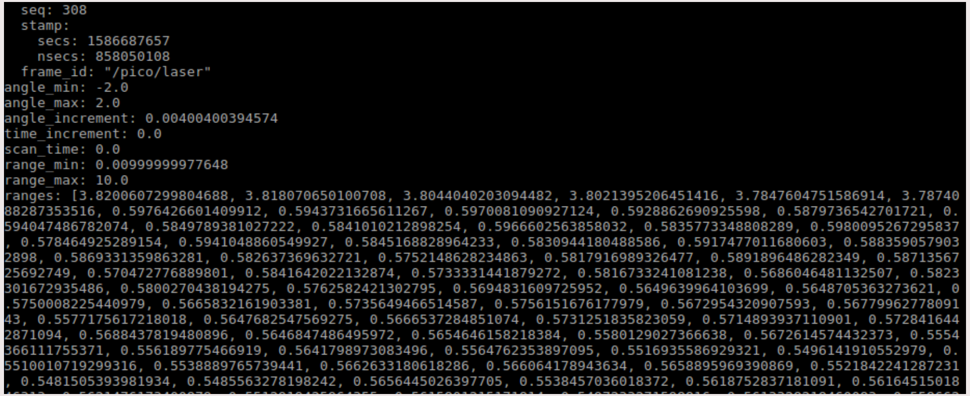


LaserData struct

How we can get a LaserData struct ?

Open a Terminal (Ctrl - Alt + T)

Type : rostopic echo /pico/laser



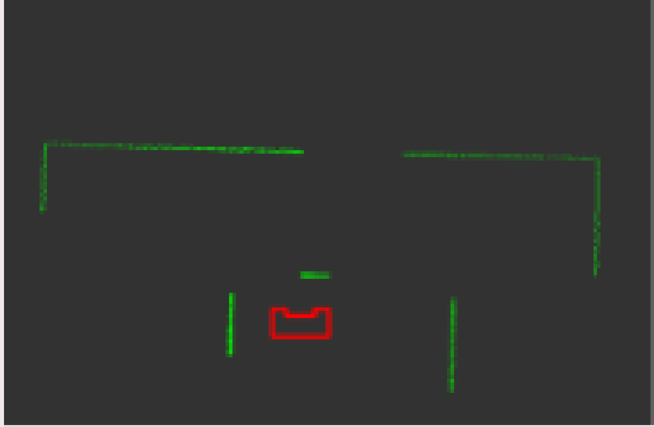
```
  seq: 308
  stamp:
    secs: 1586687657
    nsecs: 858050108
  frame_id: "/pico/laser"
angle_min: -2.0
angle_max: 2.0
angle_increment: 0.00400400394574
time_increment: 0.0
scan_time: 0.0
range_min: 0.00999999977648
range_max: 10.0
ranges: [3.8200607299804688, 3.818070650100708, 3.8044040203094482, 3.8021395206451416, 3.7847604751586914, 3.78740
88287353516, 0.5976426601409912, 0.5943731665611267, 0.5970081090927124, 0.5928862690925598, 0.5879736542701721, 0.
594047486782074, 0.5849789381027222, 0.5841010212898254, 0.5966602563858032, 0.5835773348808289, 0.5980095267295837
, 0.578464925289154, 0.5941048860549927, 0.5845168828964233, 0.5830944180488586, 0.5917477011680603, 0.588359057903
2898, 0.5869331359863281, 0.582637369632721, 0.5752148628234863, 0.5817916989326477, 0.5891896486282349, 0.58713567
25692749, 0.570472776889801, 0.5841642022132874, 0.5733331441879272, 0.5816733241081238, 0.5686046481132507, 0.5823
301672935486, 0.5800270438194275, 0.5762582421302795, 0.5694831609725952, 0.5649639964103699, 0.5648705363273621, 0
.5750008225440979, 0.5665832161903381, 0.5735649466514587, 0.5756151676177979, 0.5672954320907593, 0.56779962778091
43, 0.5577175617218018, 0.5647682547569275, 0.5666537284851074, 0.5731251835823059, 0.5714893937110901, 0.572841644
2871094, 0.5688437819480896, 0.5646847486495972, 0.5654646158218384, 0.5580129027366638, 0.5672614574432373, 0.5554
366111755371, 0.5561897775466919, 0.5641798973083496, 0.5564762353897095, 0.5516935586929321, 0.5496141910552979, 0.
5510010719299316, 0.5538889765739441, 0.5662633180618286, 0.566064178943634, 0.5658895969390869, 0.5521842241287231
, 0.5481505393981934, 0.5485563278198242, 0.5656445026397705, 0.5538457036018372, 0.5618752837181091, 0.5616451518
```

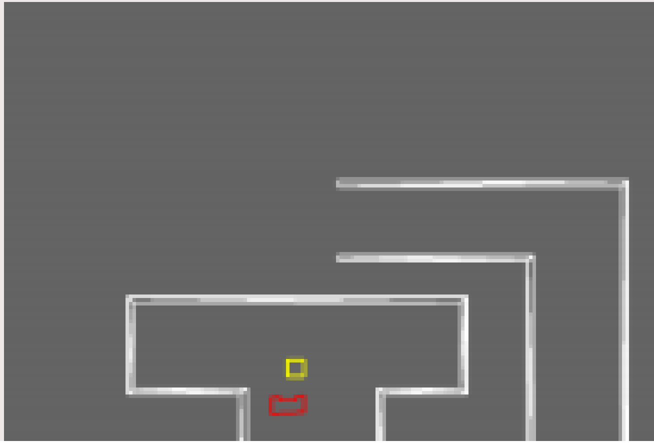
How do we know which number correspond to which laser beam?

cout<<"value at 1degree ="<<msg->ranges[1]<<endl;

LaserData struct

- How to check the distance to an obstacle using a laser?
- What does inf mean in the laser range?



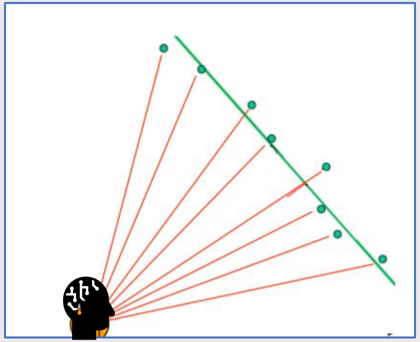


Line extraction

Imagine we have two scans. In the second scan scene has changed a little. How do we match them?

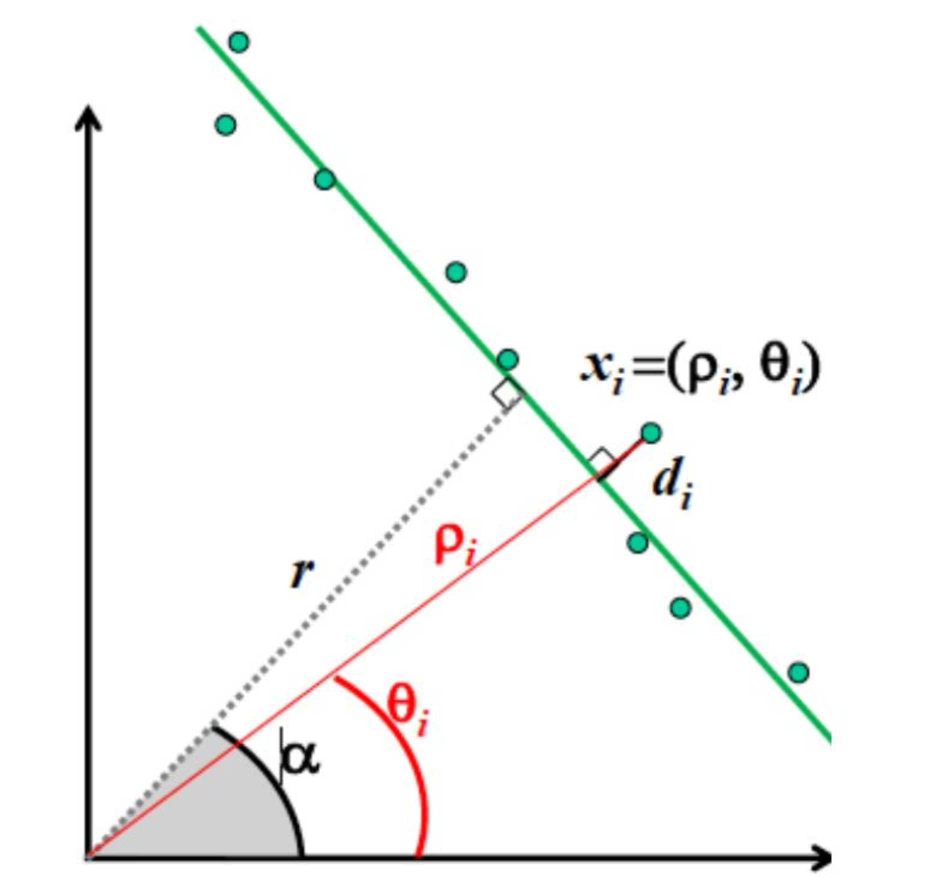
There are many different techniques to find features:

I will explain one here but It is good to have a look to find easier and more robust techniques.



Line extraction : Problem

Given a measurement vector of N range ,X_i = ($\rho_{i\,,}$ θ_i) for i=1..N, what are the parameters r,α that define a line feature for these measurements.



Line extraction : Solution

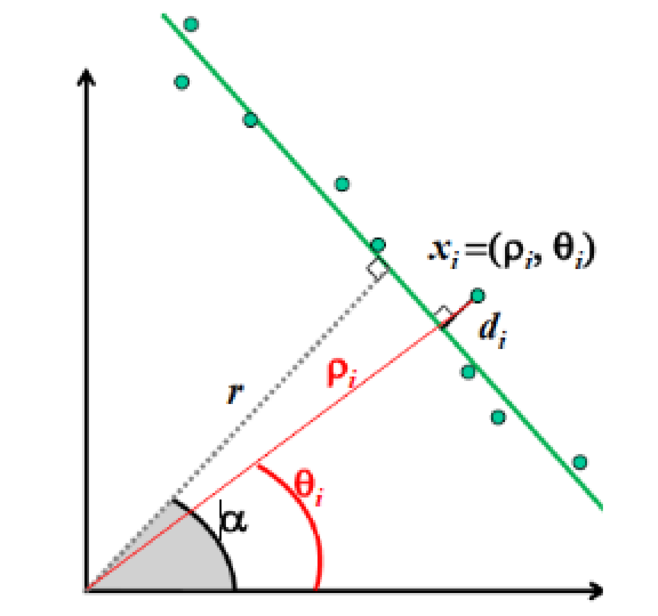
We need a simple projection.

All measurements should satisfy the linear equation:

$\rho_i * \cos(\theta_i - \alpha) = r$

But measurements are noisy, and points will be some distance di from the line.

$\rho_i * \cos(\theta_i - \alpha) - r = d_i$

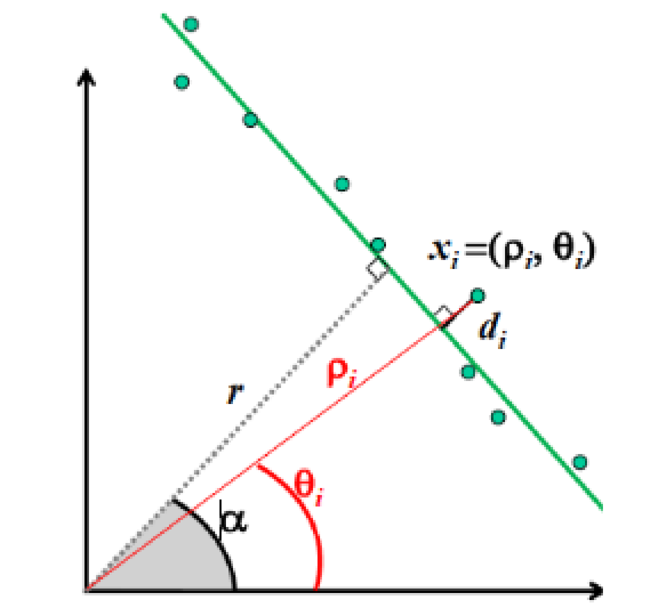


Line extraction : Solution

Minimize Sum of Squared Errors:

$$\sum (\rho_i * \cos (\theta_{i\,\_} \alpha)- r)^2$$

If the error is smaller than a predefined value, a line segment is found.

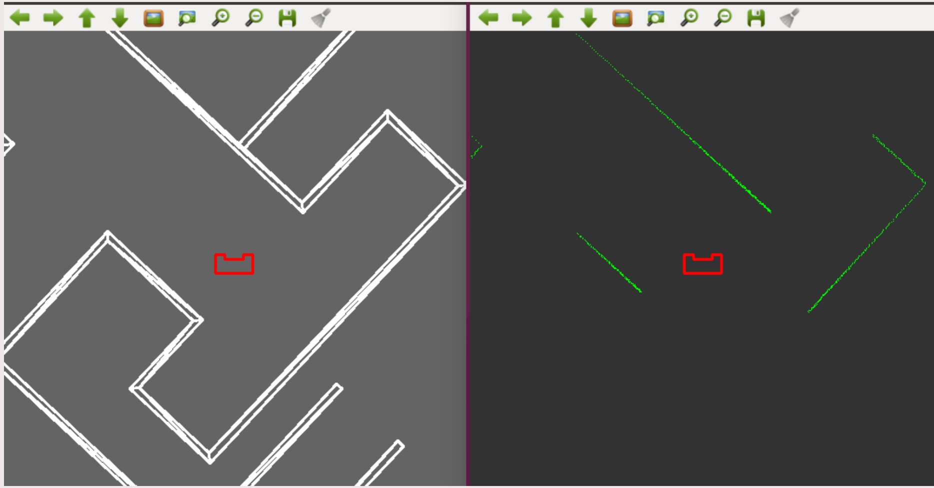


Feature extraction

Fit a line to points in laser data.

Find the most distance point to the line.

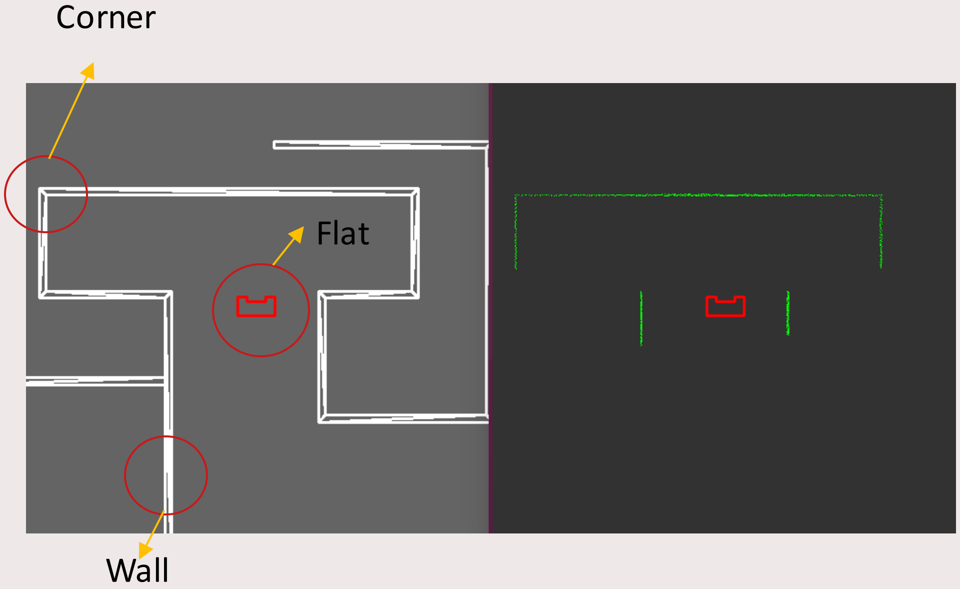
If two lines are close enough, merge two lines.



Segmentation

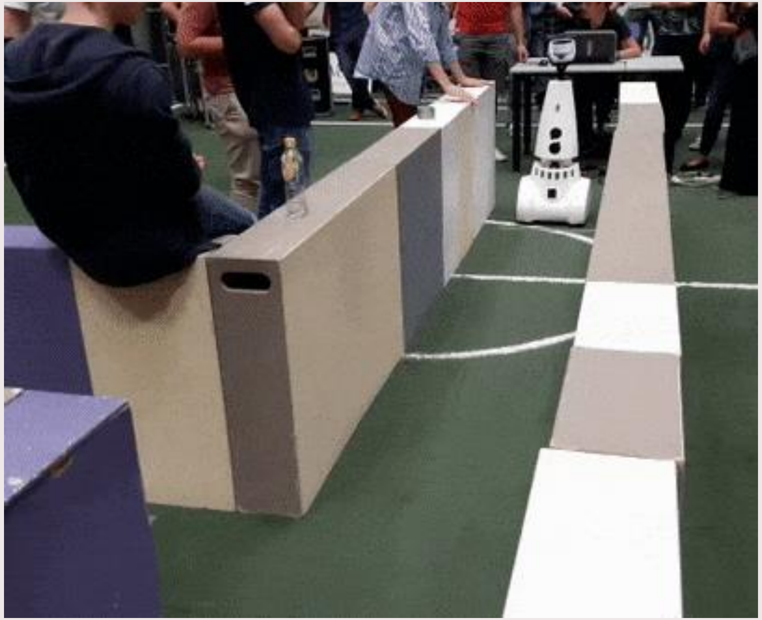
How do we know that our detection is a wall or an object?

Flat : no change in all directions.
Wall: no change along the wall direction.
Corner : significant change in all directions.



Segmentation

- Movement detection algorithm is employed to distinguish the difference between human movement and static objects.
  - The segments in two scans are stored into two matrixes and compared together.
    - If there is a distinct distance between these two segments , it is classified as a human.



Vision library

- OpenCV is an image processing library .

- An efficient implementation of the line extraction algorithm exist in this library.

- Supports Windows, Linux, Mac OS, iOS and Android.

Question :

m.dolatabadi.farahani@tue.nl