

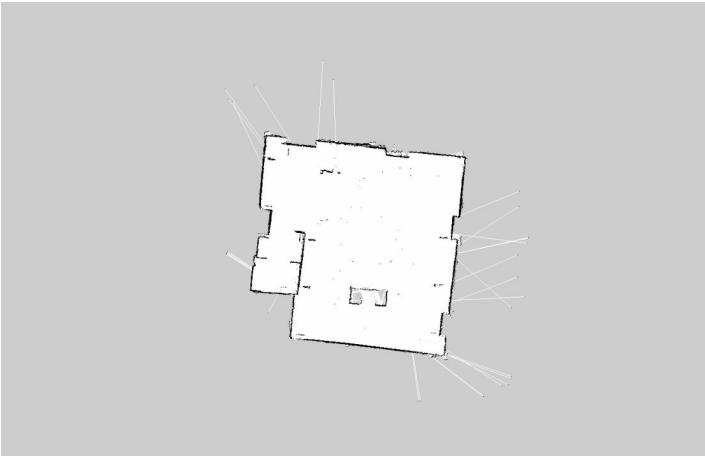
Presentation Mapping Project



Group 13

Haohua Dong 93079
André Ferreira 96158
Duarte Cerdeira 96195
Inês Pinto 96230

Introduction



 ROS
Robot Operating System

- **P3-DX Pioneer robot**
- **URG-04LXUG01 laser**
- **ROS Interface**
- **Occupancy Grid Mapping Algorithm**

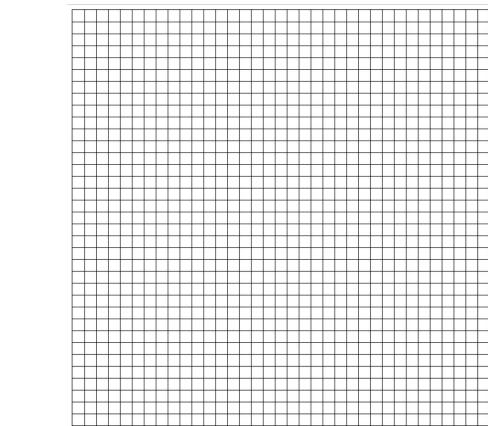
Occupancy Grid Mapping

$$m^* = \underset{m}{\operatorname{argmax}} \ p(m|z_{1:t}, x_{1:t}),$$

$$l_{t,i} = l_{t-1,i} + \text{InverseSensorModel}(m_i, x_i, z_i) + l_0,$$

Assumptions:

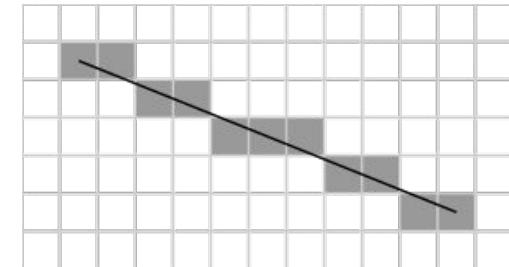
- ❖ The world map is static
- ❖ No correlation between individual grid cells
- ❖ The laser's position is known



Each cell is a **binary random variable** that models the occupancy
Cell is occupied: $p(m_i) = 1$
Cell is not occupied: $p(m_i) = 0$
No knowledge: $p(m_i) = 0.5$

Bresenham's Line Algorithm

- The Hokuyo LRF is very precise, with a resolution of 1 mm.
- Approximate the beams as a line - > Apply Bresenham's Line to update the map !
- Discretize the coordinates of the laser and the measured range to (x, y).



Slope $m \leq 1$

$$y_{k+1} = y_k + m$$

$$x_{k+1} = x_k + 1$$

Slope $m \geq 1$

$$y_{k+1} = y_k - m$$

$$x_{k+1} = x_k - 1$$

Slope $m > 1$

$$x_{k+1} = x_k + \frac{1}{m}$$

$$y_{k+1} = y_k + 1$$

Slope $m < 1$

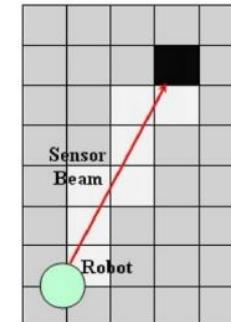
$$x_{k+1} = x_k - \frac{1}{m}$$

$$y_{k+1} = y_k - 1$$

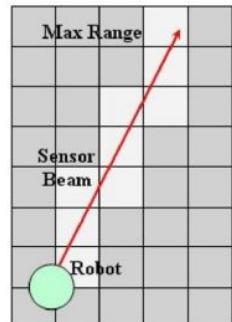
A=(X0, Y0) = position of the laser

B=(X1, Y1) = position of the detected range

C=(X2, Y2) = position of the detected range with assumed obstacle thickness



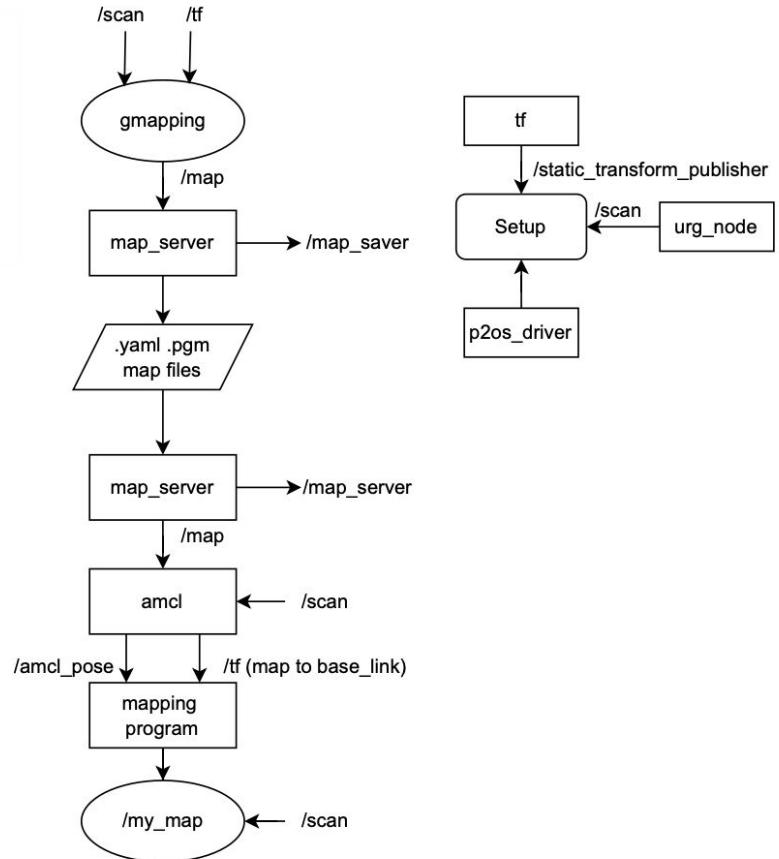
(a)



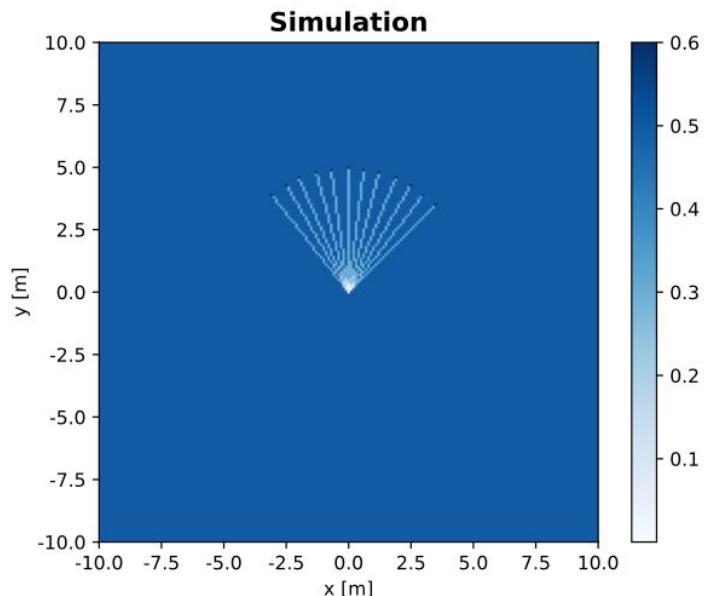
(b)



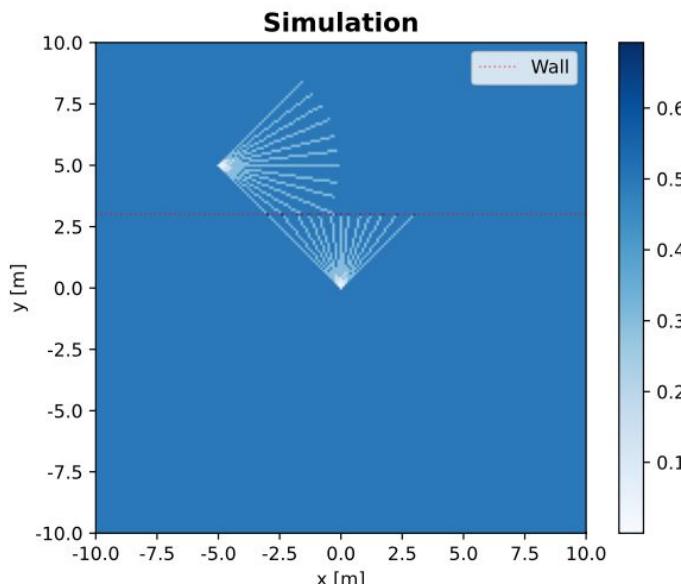
Implementation



Simulation



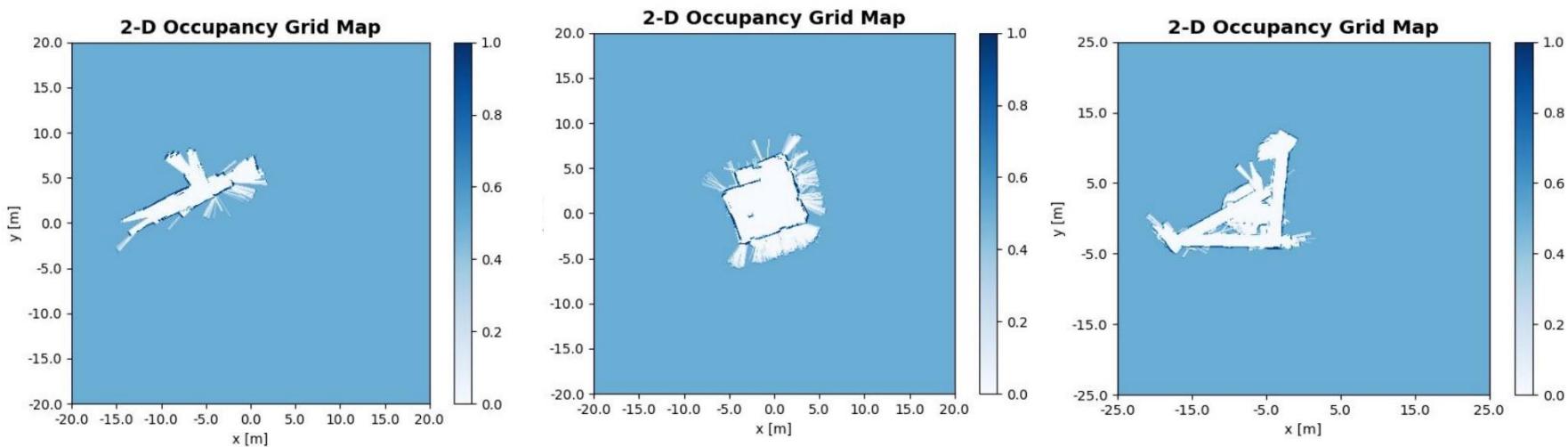
Simulation of the laser in Python



Simulation of a Wall in Python

Experimental results

Maps obtained using /pose

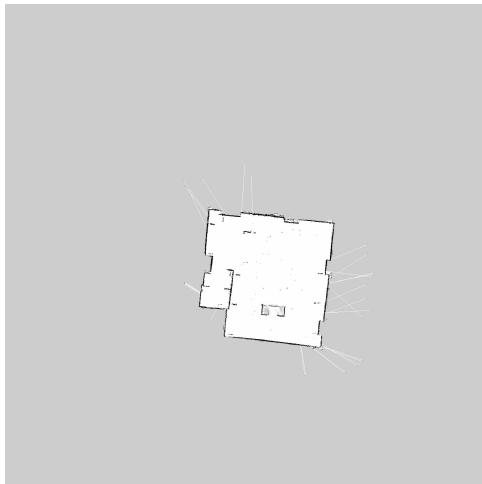


Evaluation metric for Occupancy-grid maps

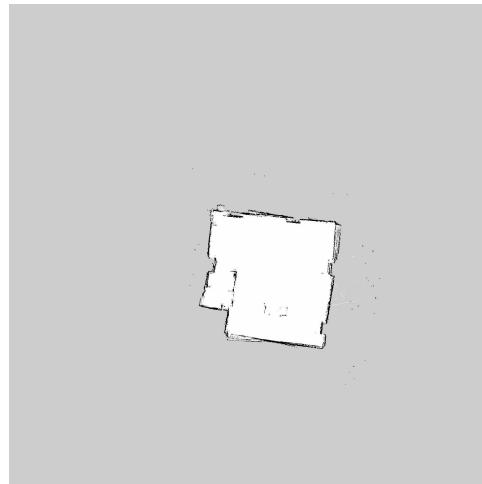
- Difference map:
0 - the map values are coincident
0.5 or 1 - the map values are different
- $D(i, j)$ is the value of each pixel, N_{free} is the number of free cells and $N_{occupied}$ is the number of occupied cells of the reference map.

$$Error[\%] = \frac{\sum_{i,j} D(i, j)}{N_{free} + N_{occupied}} \times 100$$

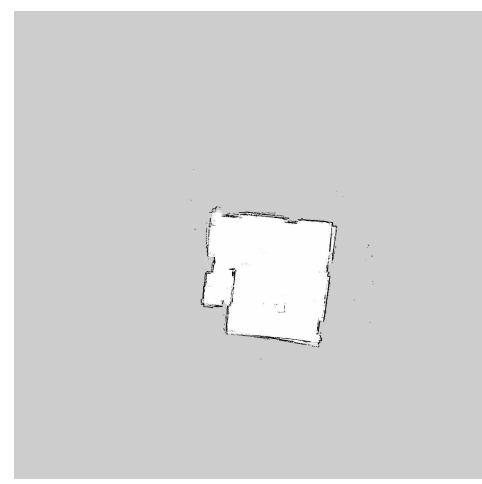
Room LSDC4 maps



Reference map
(obtained from the gmapping)

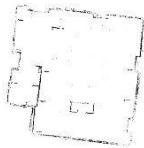


Map A



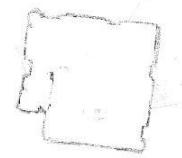
Map B

Differences and Errors



Difference between reference map and map A

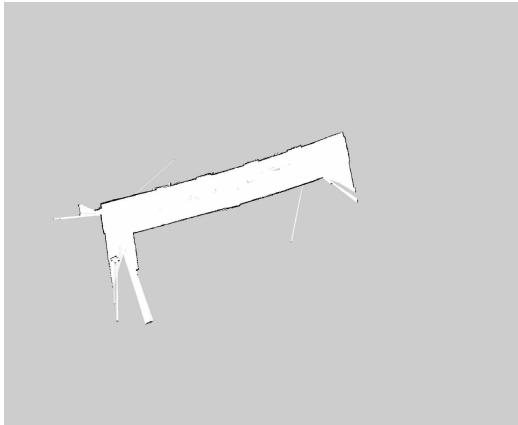
Error = 5.75 %



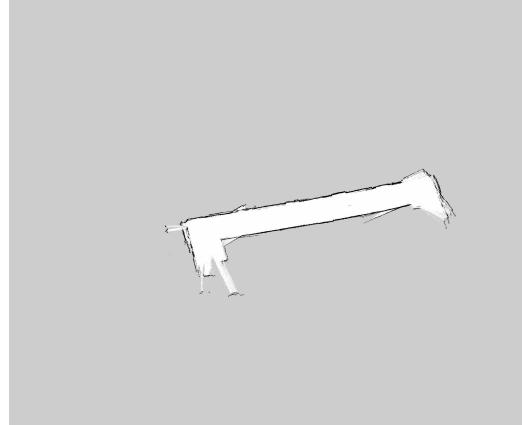
Difference between map A and map B

Error = 2.62%

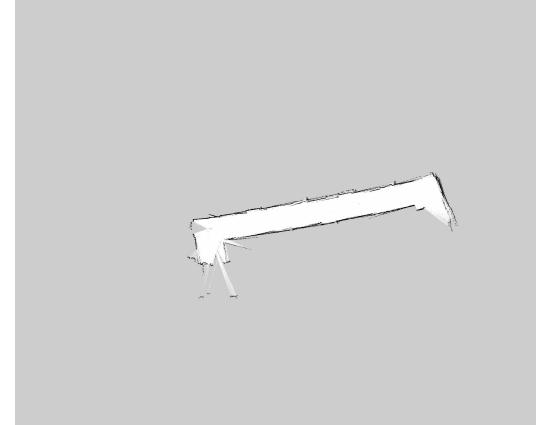
Corridor of the 5th floor



Reference map
(obtained from the gmapping)

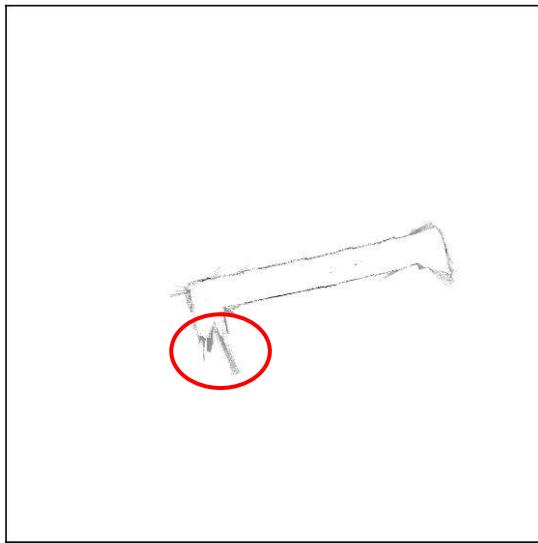


Map A



Map B

Differences and Errors

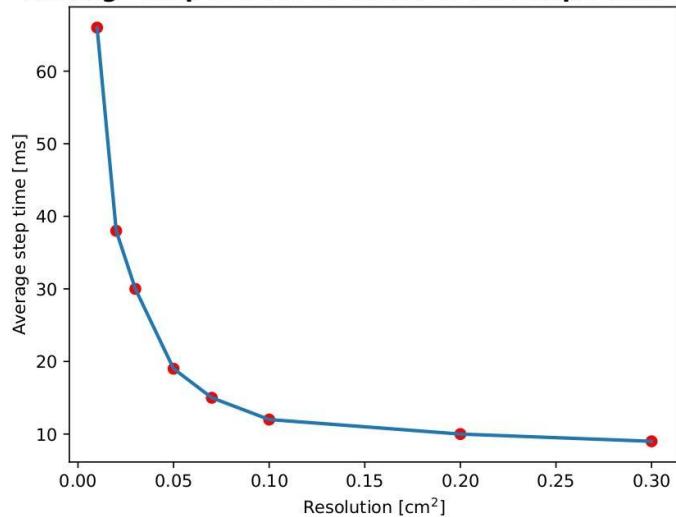


Difference between map A and map B

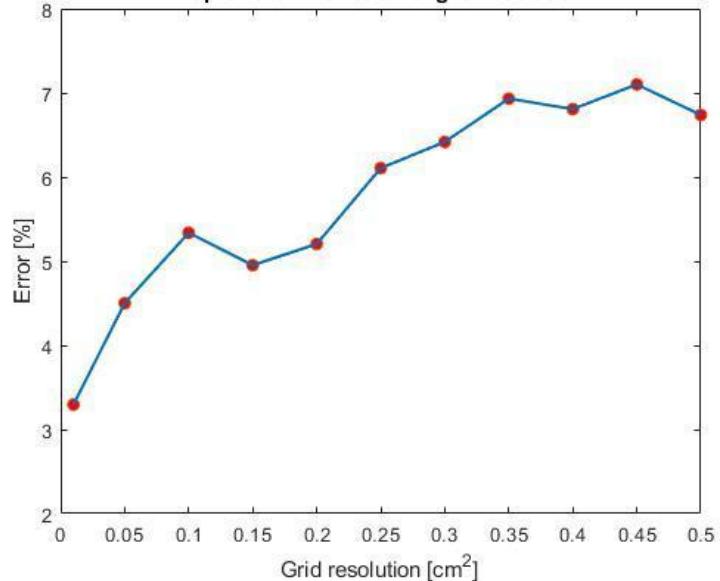
Error = 8.2%

Results' analysis

Average step time in function of the map resolution

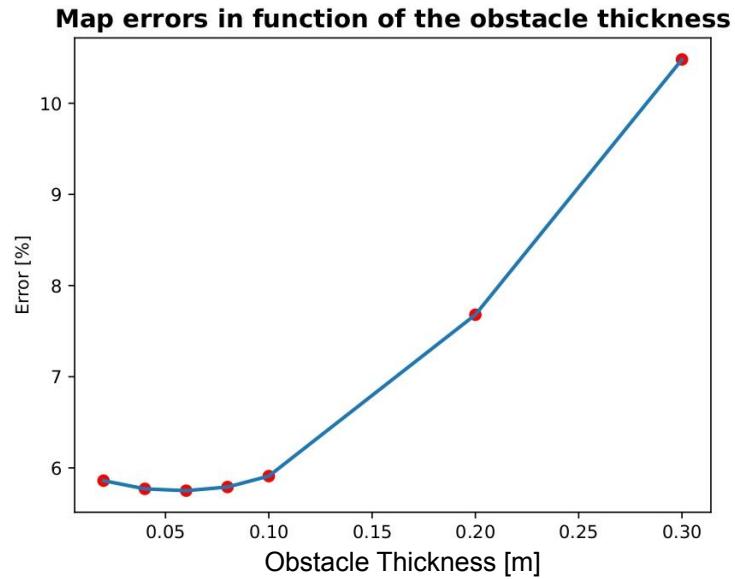


Map errors in function of grid resolution



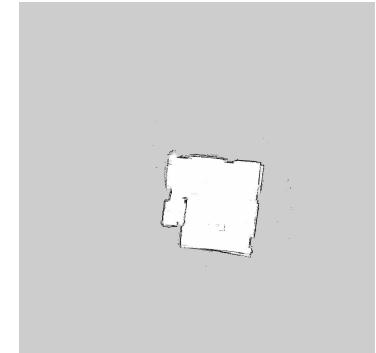
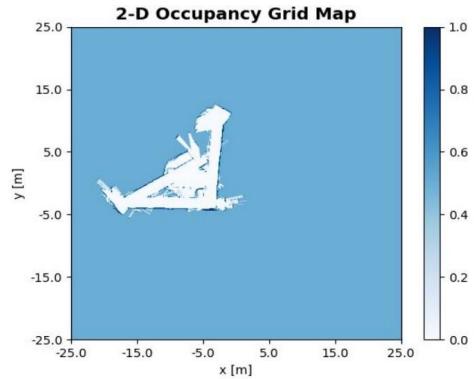
- Map is an NxN matrix;
- Complexity is $O(N^2)$ with N depending on the resolution chosen.

Results' analysis



Conclusions

- Several complications made the process longer than required
- Obtained accurate maps of the surrounding environment
- Highly dependent on outside factors
- Small errors in relation to ground truth
- There are other experiments that we thought were interesting to make



THE END