Albert-Ludwigs Universität Freiburg
Technische Fakultät
Institut für Informatik

Full Map Posterior SLAM in ROS

Master Project in Autonomous Intelligent Systems

22.04.2021

José Arce y de la Borbolla

Professor: Prof. Dr. Wolfram Burgard

Advisors: Dr. Daniel Büscher, Dr. Lukas Luft

Outline

Background and Motivation

Grid-Based, Rao-Blackwellized Particle Filters

- Traditional Grid-based Particle Filter SLAM works by:
  - Factorizing the SLAM Posterior (Rao-Blackwellization):

$$p\left(\boldsymbol{x}_{0:t}, \boldsymbol{m} \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}\right) = \underbrace{p\left(\boldsymbol{x}_{0:t} \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}\right)}_{\text{Path Posterior}} \cdot \underbrace{p\left(\boldsymbol{m} \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}\right)}_{\text{Map Posterior}}$$

  - Modeling the Path Posterior Distribution non-parametrically using weighted samples (particles):

$$\mathcal{X} = \left\{ \left\langle \boldsymbol{x}^{[j]}, w^{[j]} \right\rangle \middle| \ j = 1, \dots, J \right\}$$

$$p\left(\boldsymbol{x} \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}\right) \simeq \sum_{j=1}^{J} w^{[j]} \delta_{x^{[j]}}\left(\boldsymbol{x}\right)$$

  - Each particle represents a possible trajectory and maintains its own map.

- Traditional Grid-based Particle Filter SLAM works by:
  - Using a Grid Map with a Reflection Model.

$$p\left(m_i\right) = \begin{cases} q & \text{if } m_i = \text{occupied} \\ 1 - q & \text{if } m_i = \text{free} \end{cases}$$

  - (FastSLAM V2.0) Improving the proposal pose distribution using ScanMatching.

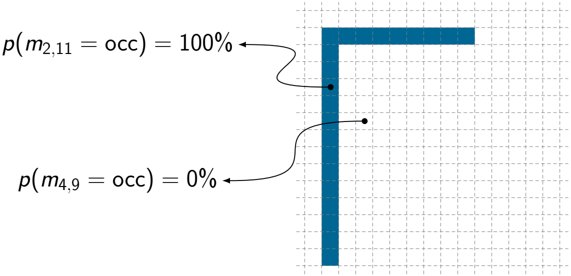
$$\boldsymbol{x}_t^{\star} = \underset{\boldsymbol{x}_t}{\arg\max} \left\{ p\left(\boldsymbol{z}_t \mid \boldsymbol{x}_t, \boldsymbol{m}_{t-1}\right) \cdot p\left(\boldsymbol{x}_t \mid \boldsymbol{u}_t, \boldsymbol{x}_{t-1}^{\star}\right) \right\}$$

  - Only use the most likely pose and map.

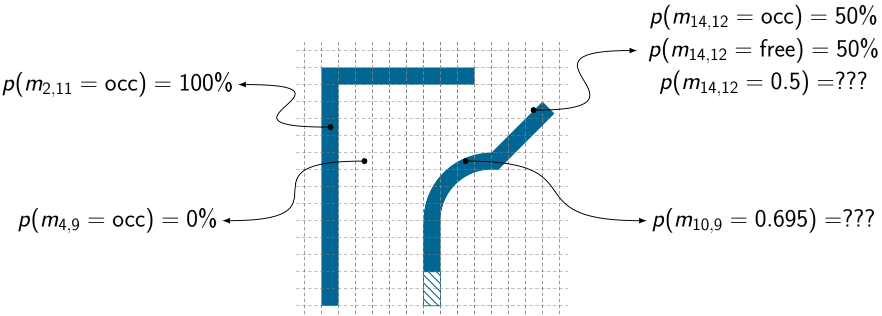
Background and Motivation

Approach

- The Reflection Map properly encodes obstacles aligned with the grid.



- The Reflection Map properly encodes obstacles aligned with the grid.



- But it does not account for partial occupancies, misalignments or transparency.

Background and Motivation
Approach

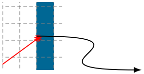
Idea

Extend the Reflection Map model by also storing the distance traversed by the beams within each cell (r), and use it to model the "permeability" of each cell as described by L. Luft, A. Shaefer and T. Schubert in *Closed-form full map posteriors for robot localization with lidar sensors*.

Background and Motivation

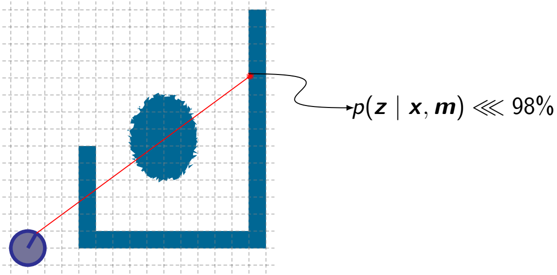
Approach

- The current Particle Weighting method only considers the endpoint of the beams.



$$p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{m}) = 98\%$$

- The current Particle Weighting method only considers the endpoint of the beams.



- This can lead to overconfidence in the particle weights.

Background and Motivation
Approach

Idea

Use a measurement likelihood distribution that takes into account all of the cells traversed by the scan's beams, and that also helps us compute the Full Map Posterior in closed form.

- The map is computed so that it maximizes the likelihood of the measured laser data.

★★★★★ 5 out of 5 stars

- The map is computed so that it maximizes the likelihood of the measured laser data.

★★★★★ 5 out of 5 stars
1 Rating

Background and Motivation
Approach

- The map is computed so that it maximizes the likelihood of the measured laser data.

5 out of 5 stars
1 Rating

4.6 out of 5 stars
17’893 Global Ratings

- But discards or ignores the uncertainty of the distributions.

- The map is computed so that it maximizes the likelihood of the measured laser data.

★★★★★ 5 out of 5 stars
1 Rating

★★★★☆ 4.6 out of 5 stars
17’893 Global Ratings

- But discards or ignores the uncertainty of the distributions.

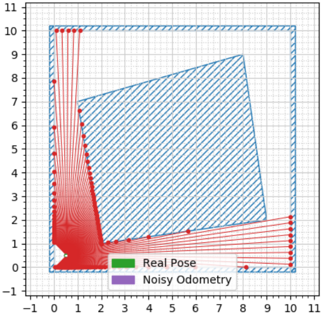
Idea

Preserve the confidence of the estimated map by modeling the Map Posterior as a parametric distribution, and not just the Most Likely Map.

Outline

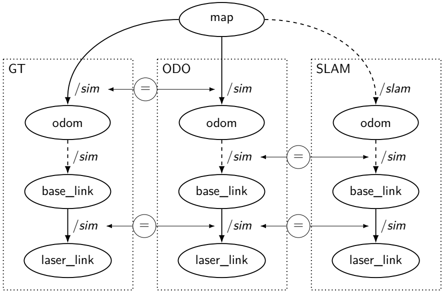
Full Map Posterior SLAM in ROS
José Arce y de la Borbolla
Background and Motivation
Solution
2D Simulator
OpenSLAM GMapping
Experiments
Results
Discussion

A simulator was created for generating test scenarios in a 2D environment.



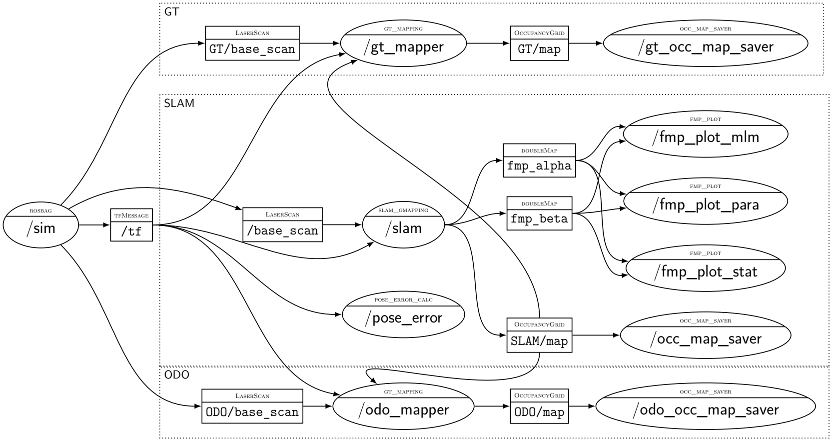
- Input:
  - JSON File(s) containing:
    - Simulation Parameters
    - Map Configuration
    - Robot Configuration
    - Robot Commands
- Outputs:
  - ROSBag File

Solution
2D Simulator



- The generated `ROSBags` publish `LaserScan` messages, `TF` odometry transformations and can include other custom messages.
- The scans and transformations are published under three coordinate system branches: Ground Truth, Pure Odometry and SLAM.

Solution
OpenSLAM GMapping: Overview



Solution

OpenSLAM GMapping: Overview

- Modify the `OpenSLAM GMapping` framework to support the features mentioned in the Motivation section.
  - Create custom ROS messages, especially a map message type that supports `double`-typed cells for publishing the Full Posterior.
  - Add `double` field `R` to the cell class to accumulate the beam's traveled distance within it.
  - Implement methods for computing the `R` value of the cells, improve the line discretization algorithm, improve the numerical stability, and more.

Solution

OpenSLAM GMapping: Full Map Posterior

- The new, Decay model was implemented alongside the Reflection Model. A parameter can be used to select between the two.
- Each model considers a prior map distribution and factorizing sensor model as shown in the following table.

	Reflection Model	Exponential Decay-Rate Model
Prior Dist. $p(m_i) =$	$\mathrm{Beta}(\alpha_0, \beta_0)$ (Uninf. with $\alpha_0 = \beta_0 = 1$)	$\mathrm{Gamma}(\alpha_0, \beta_0)$ (Uninf. with $\alpha_0 = 1,\ \beta_0 = 0$)
Forward Sensor Model $f(r_i, m_i, \delta_i) =$	Binomial Dist. $\mu_i^{\delta_i}(1-\mu_i)^{1-\delta_i}$	Poisson-Exponential Dist. $\lambda_i^{\delta_i}\mathrm{e}^{-\lambda_i r_i}$

	Reflection Model	Exponential Decay-Rate Model
Measurement Likelihoods $\ell\left(r_i, m_i\right) =$	$\frac{\left(H_i+\alpha_0\right)^{\delta_i}\left(M_i+\beta_0\right)^{1-\delta_i}}{H_i+\alpha_0+M_i+\beta_0}$	$\left(\frac{R_i+\beta_0}{R_i+\beta_0+r_i}\right)^{H_i+\alpha_0}\left(\frac{H_i+\alpha_0}{R_i+\beta_0+r_i}\right)^{\delta_i}$
Map Posterior $\mathrm{BEL}\left(m_i\right) =$	$\mathrm{Beta}\left(H_i+\alpha_0, M_i+\beta_0\right)$	$\mathrm{Gamma}\left(H_i+\alpha_0, R_i+\beta_0\right)$

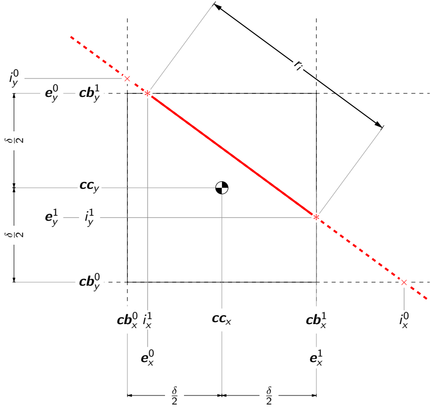
- The measurement likelihoods are computed as log-likelihoods for numerical stability and accumulated for each particle as their weights.
- Before re-sampling, the particle weights are normalized using $\mathrm{softmax}(\cdot)$ to revert them to likelihoods.

- The most likely map can then be computed as follows.

	Reflection Model	Exponential Decay-Rate Model
Most Likely Map $m_i^\star =$	$\dfrac{H_i + \alpha_0 - 1}{H_i + \alpha_0 + M_i + \beta_0 - 2}$	$\dfrac{H_i + \alpha_0 - 1}{R_i + \beta_0}$

Solution

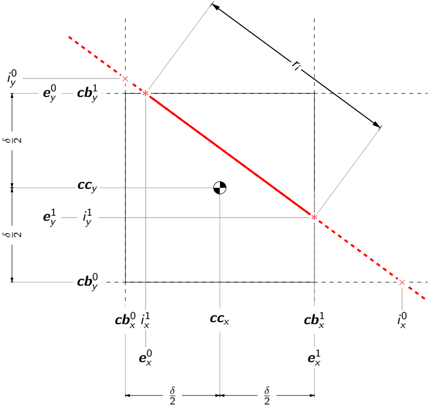
OpenSLAM GMapping: Cell R Values



- For each cell traversed by a beam:
  1. Compute beam's delta $\boldsymbol{\Delta b} = \boldsymbol{b}_e - \boldsymbol{b}_s$.
  2. Compute the cell boundaries $\boldsymbol{cb}$.
  3. Compute beam's slope $m = \frac{\boldsymbol{\Delta b}_y}{\boldsymbol{\Delta b}_x}$ and bias $b = \boldsymbol{be}_y - m \cdot \boldsymbol{be}_x$.
  4. Find the intersections $\boldsymbol{i}$ of the cell boundaries and the beam.

Solution

OpenSLAM GMapping: Cell R Values



5. Find the intersection between all intervals.

$$[\boldsymbol{e}^0, \boldsymbol{e}^1] = [\boldsymbol{b}_s, \boldsymbol{b}_e] \cap [\boldsymbol{cb}^0, \boldsymbol{cb}^1] \cap [\boldsymbol{i}^0, \boldsymbol{i}^1]$$

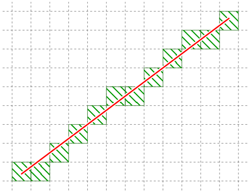
6. Return $r_i = \left\| \boldsymbol{e}^1 - \boldsymbol{e}^0 \right\|_2$.

- Accumulate in the cell over all scans: $R_i = R_i + r_i$.
- Edge Cases:
  - Beam's start or end fall inside cell.
  - Beam is horizontal or vertical.

Solution

OpenSLAM GMapping: Line Supercover

- GMapping uses Bresenham to discretize the beams to the map's grid.
- However, Bresenham can leave out some cells, which can be detrimental for the computation of the full-beam particle weights.

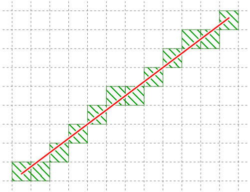


Bresenham's Algorithm

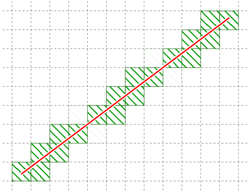
Solution

OpenSLAM GMapping: Line Supercover

- GMapping uses Bresenham to discretize the beams to the map's grid.
- However, Bresenham can leave out some cells, which can be detrimental for the computation of the full-beam particle weights.
- Thus, it was replaced by a Line Supercover algorithm that accounts for all crossed cells.



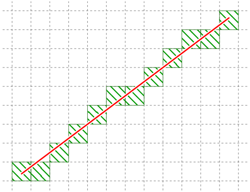
Bresenham's Algorithm



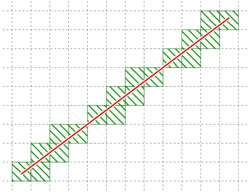
Line Supercover Algorithm

Solution
OpenSLAM GMapping: Line Supercover

- It consists of a few changes to the original Bresenham's:
  - Use both the current and previous iteration error.
  - Use double precision x and y differences for better accuracy.
  - Also check the upper-left and lower-right cells whenever a change in y takes place.



Bresenham's Algorithm



Line Supercover Algorithm

Solution

OpenSLAM GMapping: Miscellaneous

- An option for computing the average particle pose was added to estimate the robot pose more robustly and less susceptible to noise.

$$\begin{pmatrix} \overline{x} \\ \overline{y} \\ \overline{\theta} \end{pmatrix} = \begin{pmatrix} \sum_i w_i \cdot x_i \\ \sum_i w_i \cdot y_i \\ \arctan\left(\sum_i w_i \sin\theta_i,\ \sum_i w_i \cos\theta_i\right) \end{pmatrix}$$

- Uniform noise can be added to the particle weights to prevent overconfidence.

$$\ell_i = (1 - w_{oc}) \cdot \ell_i\left(r_i, m_i\right) + \frac{w_{oc}}{r_{\max}^{\delta_j}},\ w_{oc} \in [0, 1]$$

- Before thresholding, the Decay map must be converted to reflectivities via:

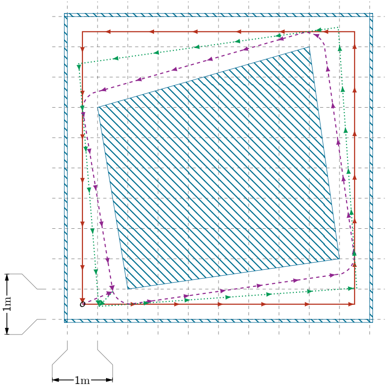
$$occ_i = 1 - \exp\left(-d\frac{H_i}{R_i}\right)$$

Outline

Experiments

Simulations

- A robot was simulated driving around a map as shown in the figure doing 1, 2 or 3 loops with varying number of measurement steps.
- After these initial moves, another 100-step loop around the map was executed, this time in the CW direction, with map updates turned off (Localization-Only), to evaluate the accuracy of the localization and the previously generated map.
- All four combinations of Map Models (REF/DEC) and Particle Weights (CMH/ML) were tested.



- The uncertainty parameters for the motion and measurement models used were:

$$\boldsymbol{\alpha} = \begin{pmatrix} \alpha_{\rho\rho} & \alpha_{\rho\theta} & \alpha_{\theta\rho} & \alpha_{\theta\theta} \end{pmatrix} = \begin{pmatrix} 0.01 & 0.05 & 0.01 & 0.05 \end{pmatrix}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{rr} & \sigma_{r\theta} \\ \sigma_{\theta r} & \sigma_{\theta\theta} \end{pmatrix} = \begin{pmatrix} 0.005 & 0.0 \\ 0.0 & 0.0 \end{pmatrix}$$

- Each combination of measurement steps, map model and particle weight was executed 1000 times.

Experiments
SLAM Parameters

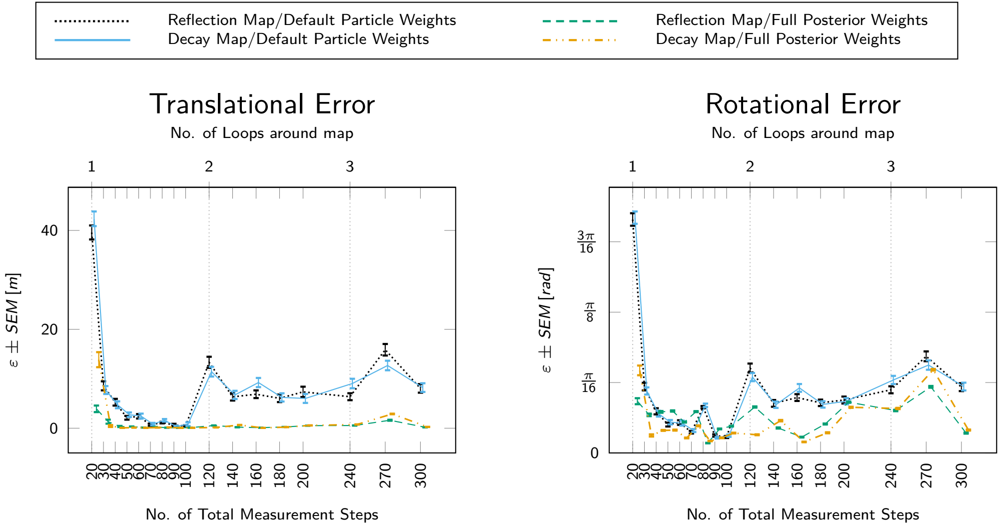
- Pose improvement was turned off.
- Uninformative Prior:
  - REF: $\alpha_0 = 1,\ \beta_0 = 1$
  - DEC: $\beta_0 = 1,\ \beta_0 = 0$
- Map Resolution $\delta = 5\text{cm}$
- Occupancy threshold $\tau = 25\%$
- Overconfidence Uniform Noise weight $w_{oc} = 0$

All other parameters were left with their default values.
The considered error metric was angular difference and the euclidean distance between the last Localization-only ground truth and corrected poses.

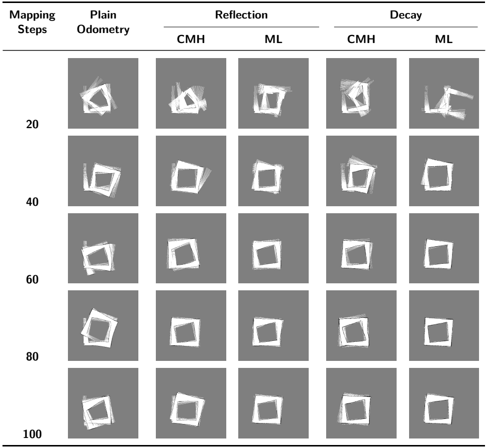
Outline

Results

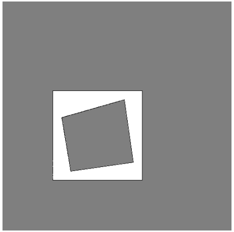
Errors in Simulated Data



Results
Qualitative Results

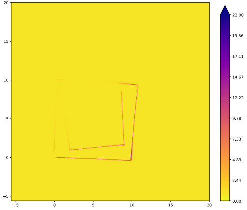


REF	Reflection Map (Default)
DEC	Exponential Decay Rate Map
CMH	Closest Mean Hit Likelihood Particle Weights (Default)
ML	Full Map Posterior Measurement Likelihood Particle Weights

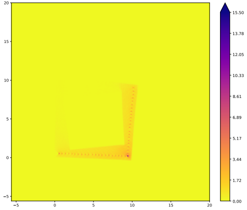

GT Occupancy Map

Results
Map Posterior Properties

Parameters

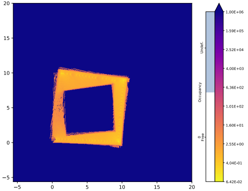


α

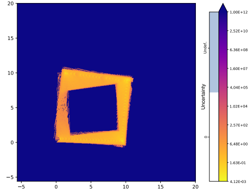


β

Moments

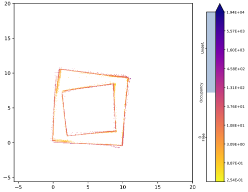


Mean



Variance

Mode



Most Likely Map

Outline

Discussion

Limitations

- Although the proposed method has clear advantages with respect to its accuracy and faster convergence due to the use of the additional information, it also comes with some limitations:
  - It depends highly on the map discretization
    - As the number of traversed cells increases (due to higher map resolutions, longer sensor range, . . .), the measurement likelihoods shrink or grow exponentially.
    - This can lead to potential numerical stability problems or particle deprivation at the very early stages of the SLAM.
  - It is computationally more expensive.
    - ML checks all traversed cells instead of just the endpoints.
    - The line discretization algorithm can return more cells for a given line, making the map updates and particle weighting more costly.
    - A trade-off between computational resources and accuracy must be made depending on the specific requirements.

Discussion

Future Work

- Relax the assumption of cell independence to improve the measurement likelihoods.
- Try the described methods online in a physical robot.
- Benchmark the exact computational requirements in term of memory and processing power.
- Research other mapping models, particle weighting methods and approximations to this method that reduce the computational demand.

References

- L. Luft, A. Schaefer, T. Schubert, and W. Burgard. Closed-form full map posteriors for robot localization with lidar sensors. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2017.
- A. Schaefer, L. Luft, and W. Burgard. An analytical lidar sensor model based on ray path information. *IEEE Robotics and Automation Letters (RA-L), 2(3):1405–1412*, July 2017.
- G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao- blackwellized particle filters. *IEEE Transactions on Robotics, 23(1):34–46*, 2007.
- R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner. On measuring the accuracy of SLAM algorithms. *Journal of Autonomous Robots, 27(4):387– 407*, 2009.

References

- J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal, 4(1):25–30*, 1965.
- E. Andres, P. Nehlig, and J. Françon. Supercover of straight lines, planes and triangles. In *E. Ahronovitz and C. Fiorio, editors, Discrete Geometry for Computer Imagery, pages 243–254, Berlin, Heidelberg*, 1997.
- J. Rao and A. Sengupta. *Topics in circular statistics*, volume 5. 01 2001.
- A. Elfes. Occupancy Grids: *A Probabilistic Framework for Robot Perception and Navigation*. Carnegie Mellon University, 1989.

- All of the code created for this project can be found here: https://github.com/joseab10/FMP_gmapping

Thank you very much for your attention!

Questions & Comments?