# ROB 550 BotLab Report

Saptadeep Debnath - saptadeb@umich.edu

## I. MAPPING AND SLAM

### *1.1 - Mapping - Occupancy Grid*

Write a paragraph or two description including the following:

- Provide the values of the incremental log odds you're using for a free & occupied cells.
- Describe the method used determine which cells to update.
- Comment on mapping behavior.

The mapping function is implemented in two stages - first we score the cells which are located at the endpoint of the rays. If the cells corresponding to the endpoint of the rays are in the given map, log odds for the cell are increased by '3', defining the boundaries. Next, the cells between the robot cell and the endpoint cell are scored. Bresenham's line algorithm is then used to rasterize the given ray. The log odds of these free cells are then decreased by '1'.

The maps generated when the robot is stationary as is in the case of convex_grid_10mx10m_5cm.log are not quite accurate at far away points. As the robot is stationary the laser rays originating from that position diverge farther away from each other at long distances, because of which some of the cells in between those rays don't get scored. This problem is mitigated when the robot is in motion; wherein because of the motion of the robot the laser rays tend to cover most of the grid cells and hence we obtain a better map in the end.

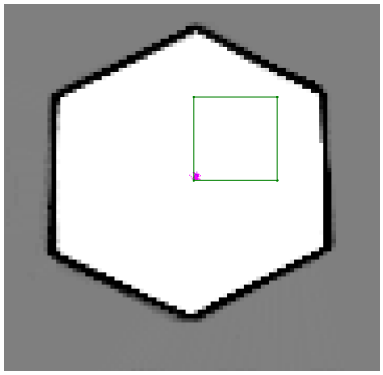

Fig. 1: Final png showing the particle distribution you obtained at the end of drive square with action only

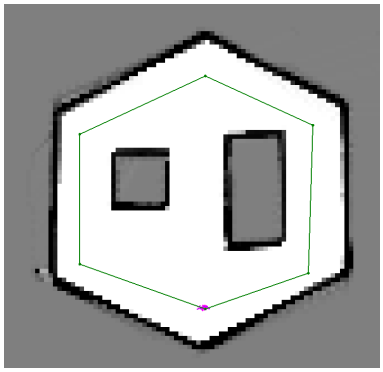

Fig. 2: Final png showing the particle distribution you obtained at the end of obstacle slam with action only

### *1.2.1 - MCL - Action Model*

Write a paragraph or two description including the following:

- Specify what type of action model you're using.
- Provide all noise constants that you're using.
- Comment on the your action model:
  - Does the distribution grow as expected for the particles?
  - Do you think the action model constants will differ drastically from one mbot to another?

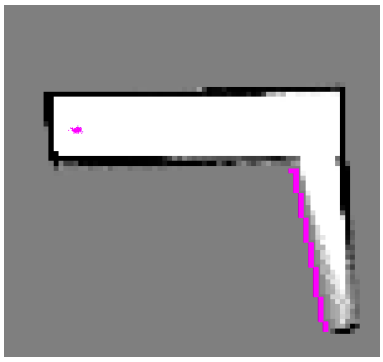

Fig. 3: Final png showing the particle distribution you obtained at the end of straight line calm with action only

*1.2.2 - MCL - Sensor Model & Particle Filter*

Write a paragraph or two description including the following:

- Specify what type of sensor model you're using, and how you are weighting and resampling.
- Provide all noise constants that you're using.
- Is the estimated pose closer to the true pose with your localized pose estimate compared to that from odometry?
- Do the particles remain in a tight region as the robot moves?
- Do the particles spread more aggressively with a certain motion type? (rotation, for example).

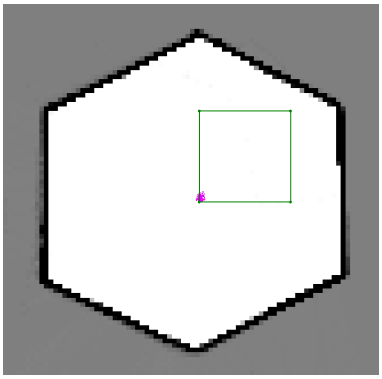

Fig. 4: Final png showing the particle distribution you obtained at the end of drive square with localization only

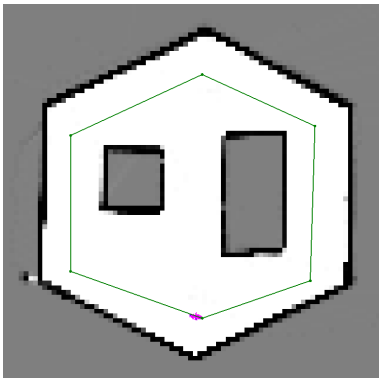

Fig. 5: Final png showing the particle distribution you obtained at the end of obstacle slam with localization only

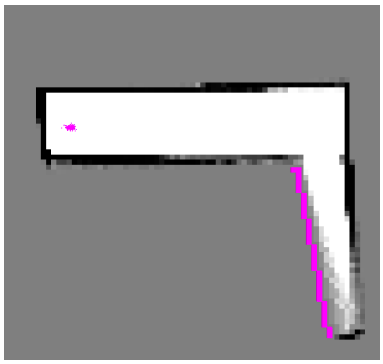

Fig. 6: Final png showing the particle distribution you obtained at the end of straight line with localization only

*1.3 - Simultaneous Localization and Mapping (SLAM)*

Write a paragraph or two description including the following:

- Did you obtain similar/close performance for obstacle_slam_10mx10m_5cm.log compared to the previous two runs? Why or why not?
- Did the change of map resolution affect your computation time? Did it improve/harm your slam results?
- Did you have to change your mapping log odds to improve on performance? If so, report them.

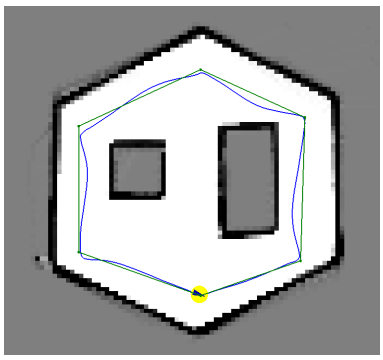

Fig. 7: Final png showing the particle distribution you obtained at the end of Obstacle Slam with Full SLAM


Fig. 8: Final png showing the particle distribution you obtained at the end of Maze LowRes with Full SLAM


Fig. 9: Final png showing the particle distribution you obtained at the end of Maze HiRes with Full SLAM


Fig. 10: Final png showing the particle distribution you obtained at the end of Maze Hard with Full SLAM

## II. PATH PLANNING AND EXPLORATION

### *2.1 - Obstacle Distances*

Comment on whether your code passed or failed the three tests in obstacle_distance_grid_test. Provide remarks on the code's performance and whether there is anything major slowing it down.

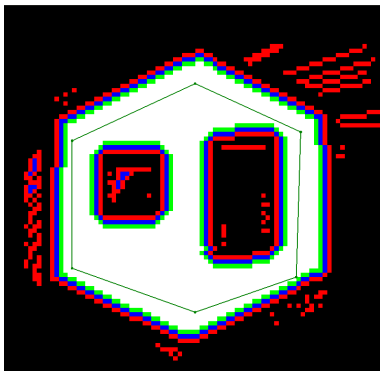

Fig. 11: Final png showing the obstacle distance grip obtained at the end of Obstacle Slam

### *2.2 - A* Path Planning*

Report statistics on your path planning execution times for each of the example problems in the data/astar folder. If your algorithm is optimal and fast, great. If not please discuss possible reasons and strategies for improvement.


Fig. 12: Final png showing the path you obtained at the end of Maze HiRes with A* path planning

### *2.3 - Map Exploration*

Comment on your exploration performance:

- What are the factors that are preventing your exploration from being ideal?
- Provide your logic behind tackling frontiers (how do you go about chosing your next frontier to go to)
- Comments on the performance of Astar and path planning with automated exploration commands.