Robotics & IOT

MOSIG M1 • May 2022

Teacher : Philip Scales

Manon ROUVELET • Jad JAWLAKH • Jizong ZHAN • Yucheng XIAO

Outline

1) Introduction: Project Idea & Structure

Project Idea

- Advanced version of Localization lab
- Except initial positions don’t need to be given at initial stage
- Relies on 3 essential features: 1st filter, 2nd filter, and weight system

Project Structure

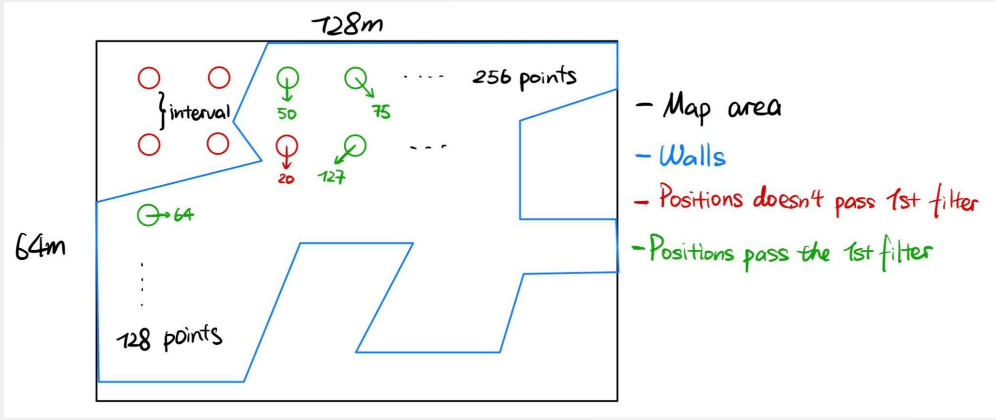
- Update()
- Estimate_Localization()
- First_Filter()
- Second_Filter()
- Construct_Weight()
- Update_Weight()
- Update_Position()
- Calculate_Score()
- Cell_Value()

2) Update Function

- If localization not initialized, initialize it
- Stores 16 positions as estimated position of robot
- Recomputes and re-estimates upon robot moving

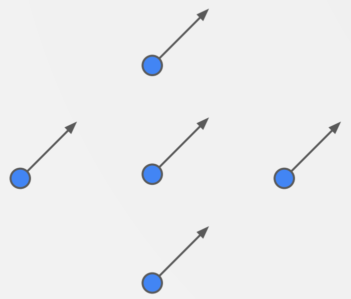
3) First Filter

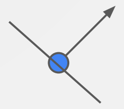
- Choose appropriate intervals
- For each position:
    - calculate score of 8 different orientations
    - save orientation with highest score
- Save position with highest score (> 25)


128m
256 points
interval
50
75
20
127
64
64m
128 points
- Map area
- Walls
- Positions doesn't pass 1st filter
- Positions pass the 1st filter

4) Second Filter

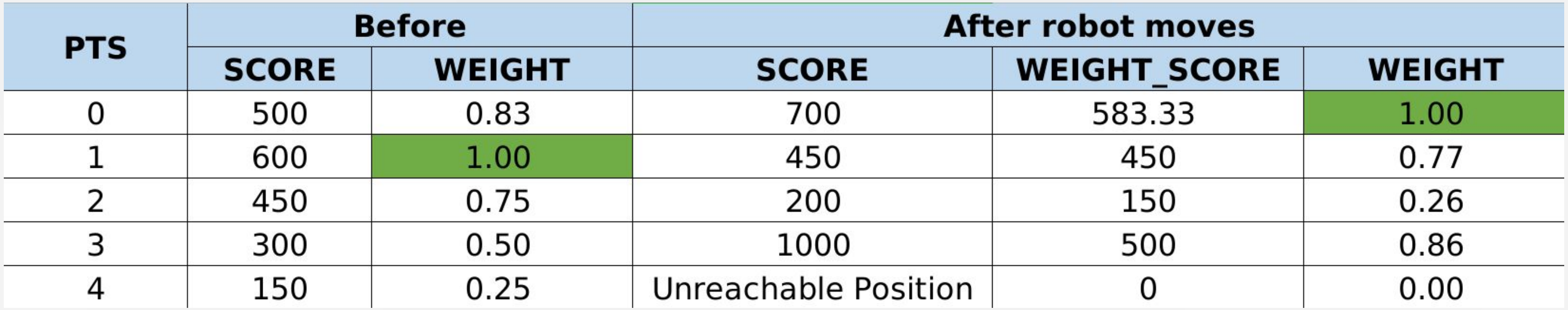
- Upscale: add 4 points around each point for more precision/accuracy (Upscaled points will have same orientation as original point),
- Each point and its 4 extra points around it, given min and max orientation so that each can scan orientation once,
- Update orientation to be that with the highest score ,
- A parameter called step to control the precision of this filter,
- Filters out 95% of the positions depending on the score,
- Loop (2nd filter) stops when there’s less than 16 points remaining.





5) Weight System: Re-verification when Robot Moves

- **construct_weight()**: weight = score/highest_score
- **update_position()**: add odometry data to current position
- **estimate_updated_position()**: like the relocalize() done in the labs but with 16 points instead of 1 point
- **update_weight()**: update the weight table, weighted_score = new_score * weight



PTS	Before		After robot moves		
	SCORE	WEIGHT	SCORE	WEIGHT_SCORE	WEIGHT
0	500	0.83	700	583.33	1.00
1	600	1.00	450	450	0.77
2	450	0.75	200	150	0.26
3	300	0.50	1000	500	0.86
4	150	0.25	Unreachable Position	0	0.00

6) Conclusion: Shortcomings & Future Improvements

Shortcomings

- We have an unresolved bug due to time constraints, when program executed with ROSBAG file played, if ROSBAG file not paused, it'll falsely think that robot rotated with ridiculously high angle
- First filter problem with tossing away points within interval of score less than 25 (the threshold), and to solve that problem we could possibly increase the number of points by from 256x256 to 512x512 for example → tradeoff: execution time and accuracy

Future Improvements

- Fix the bug
- Many cool possibilities now that we know the position of the robot on the map, for example:
  - A smart and optimal vacuum cleaner that would scan a floor and only clean dirty areas thanks to some hi-tech sensors and of course some algorithm
  - Have a better follow-me behavior where the robot is environment-aware

Demo