Improving Exploration Graph Generation for Robotic Exploration

Jacob Dent, Sayantan Datta, and Srinivas Akella

University of North Carolina at Charlotte, NC, USA

Introduction

- A important application for robots is to send them into environments that are dangerous for humans
- For a robot to traverse and explore an unknown environment, it will need to be able to generate its own motion plans
- One way of doing this is by having a robot construct an exploration graph where each node represents a potential position and each edge represents a collision-free path between two different positions
- Our work is meant to improve the efficiency of graph construction in Datta and Akella [1]

Goals

- Identify alternative ways of generating exploration graphs
- Identify a method that produces graphs more efficiently than the current method

Prior Solution

- Current solution is to skeletonize maps using Zhang and Suen's method [2]. This involves removing pixels from the border until no more can be removed
- Recomputes already explored sections of the graph with each update
- A lidar scan is simulated at each node to detect if a node is on the frontier

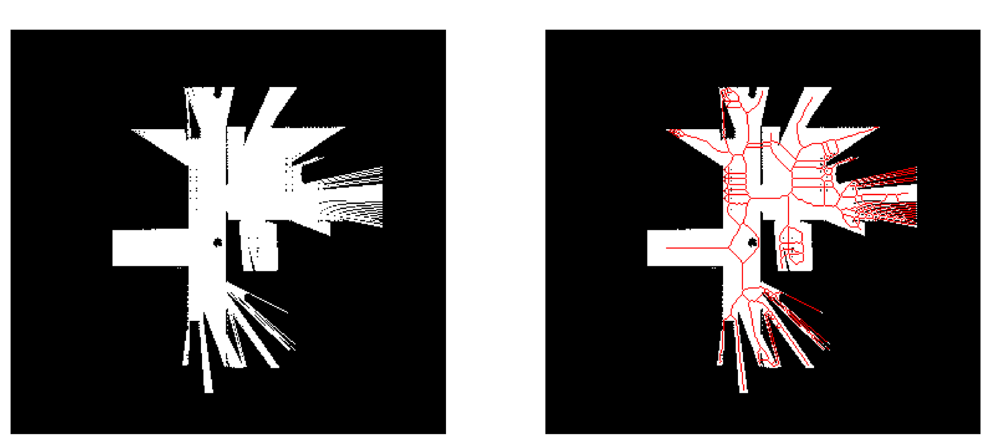

Figure 1. Image on the left is a binary image of a map generated using a simulated lidar. Image on the right is the same image but with its skeleton overlaid on top.

Probabilistic Roadmap (PRM)

- The Probabilistic Roadmap (PRM) algorithm is primarily aimed at multi-query applications [3]
- PRM takes a number of randomly generated "samples" in the free space and adds them as nodes to the graph.
- The algorithm then attempts to build collision-free connections between the nodes of the graph.

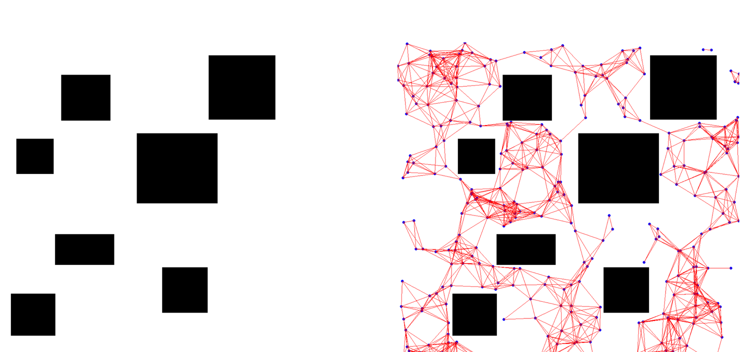

Figure 2. (Left) An unexplored map. (Right) A graph built by the PRM algorithm.

Incremental Computation of PRM Graph

- PRM is designed to work in a static workspace [3]
- Our workspace grows as the robot explores
- We are not generating completely new graphs with each map update. One lifetime graph is updated with new nodes and edges with each new map.
- An image difference is computed and space that is collision free in both the current and previous map is marked as forbidden space for node sampling.

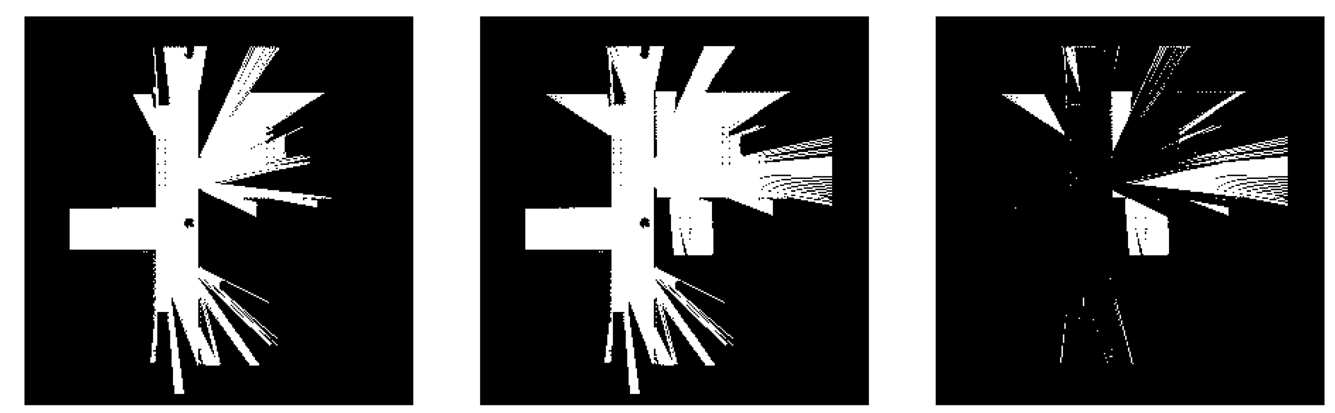

Figure 3. (Left) A map generated by lidar. (Middle) The next map generated after some exploration was performed (Right) The difference in collision-free space between the two.

- New nodes are allowed only in the newly discovered collision-free space
- Edges can be built between the new and old collision-free space allowing edges to connect these new nodes to the previous graph
- This prevents node clustering in the earlier sections of the map

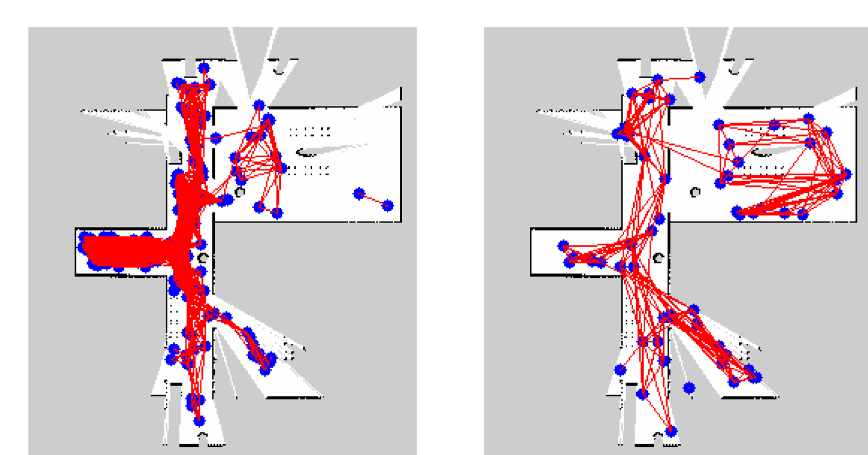

Figure 4. (Left) PRM without using the updated collision-free space. (Right) PRM computed using the updated collision-free space.

Comparison of The Methods

- A simulated robot was driven through a building layout. Eight maps were saved from the robot's lidar at different points in the environment.

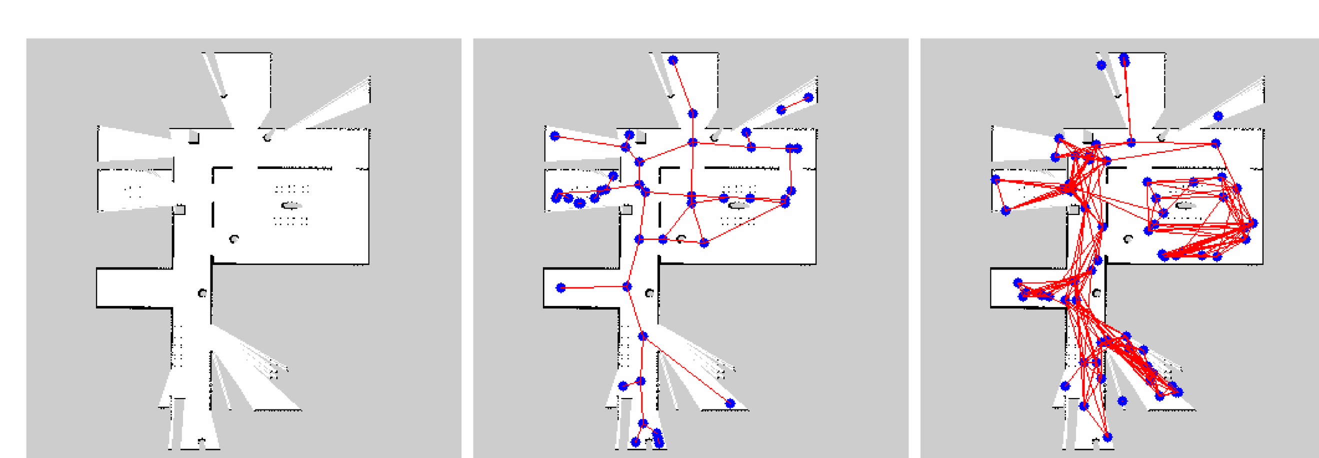

Figure 5. (Left) An example map. (Middle) The graph generated using the Skeleton method. (Right) PRM graph.

Results

- Each method was run 100 times over each map
- Average computation times for each method on eight different maps were computed

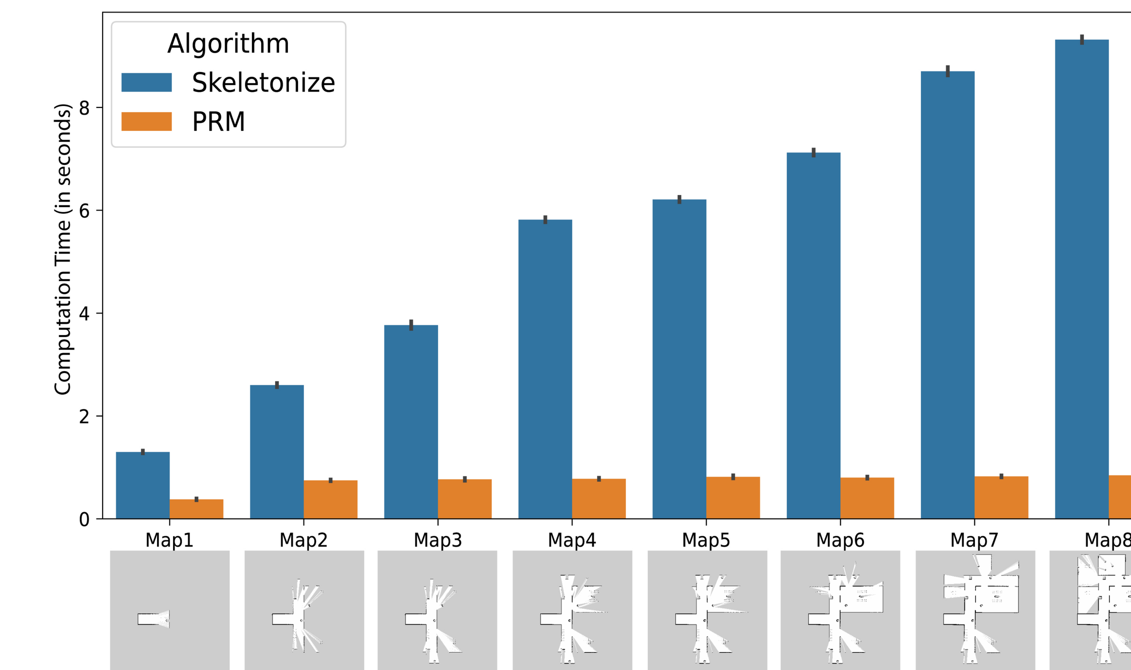

Figure 6. The maps increase in complexity (the number of independent obstacles in the map) and size. The PRM running time stays consistent across different map complexities.

Analysis

- PRM method is faster on average with a consistent running time
- Skeletonize method's computation time increased with complexity of the maps
- Skeletonize method gave simpler graphs

Future Work

- The PRM graph could be pruned of redundant nodes allowing for an even simpler graph
- The algorithm could be modified to dynamically add nodes whose number depends on the size of the newly discovered area
- This approach should be extended to work with multiple robots performing exploration simultaneously

References

[1] S. Datta and S. Akella, "Prioritized indoor exploration with a dynamic deadline," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 3108–3114.

[2] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, p. 236–239, Mar. 1984. [Online]. Available: https://doi.org/10.1145/357994.358023

[3] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

jdent4@charlotte.edu, sdatta3@charlotte.edu, sakella@charlotte.edu