

Path Planning in Unknown Environment with Obstacles Using RRT-SMART

Sagar Ojha
as03050@umd.edu

Obaid Ur Rahman
obdurhmn@umd.edu

April 11, 2023

1 Introduction

For our final project, we propose to implement a path planning algorithm in an obstacle-filled environment. The algorithm we will use is a modified version of the Rapidly Exploring Random Tree Star (RRT*) algorithm, which has been shown to be effective in path planning problems; it is called RRT* - Smart, and it employs a novel approach that combines two new techniques in RRT*, path optimization and intelligent sampling.

1.1 Definitions

RRT (Rapidly Exploring Random Tree) is a widely used algorithm for path planning in robotics. The algorithm builds a tree of feasible paths in the configuration space of a robot. The tree is randomly sampled, and new nodes are added to the tree by selecting a nearby node and extending the tree towards a randomly sampled configuration. This process is repeated until a feasible path from the initial configuration to the goal configuration is found.

RRT Star (RRT*), promises to achieve convergence towards the best solution, assuring asymptotic optimality as well as probabilistic completeness. Yet, it has been demonstrated that doing so takes an endless amount of time and has a slow convergence rate.

As a result, RRT*-Smart was proposed to address limitations of RRT*. The purpose of the proposed method is to accelerate the rate of convergence in order to arrive at an optimal or near-optimal solution considerably faster, thereby lowering execution time. The suggested algorithm's innovative method takes use of two new RRT* approaches, path optimization and intelligent sampling, which we will be implementing as our final project.

1.2 Background

1.3 Literature Review

Path planning is a crucial task in robotics and automation, and several methods have been suggested to address this issue. One of the most popular methods for resolving path planning issues is the Rapidly Exploring Random Tree (RRT) algorithm, which LaValle and Kuffner [1] initially presented in 1998. RRT is a probabilistic technique that builds a tree of viable robot configuration options. The RRT method has been extended and modified

by a number of researchers to solve this issue and enhance performance. An example of an RRT version that constructs two trees from the beginning and goal configurations and links them to produce a path is the RRT-Connect method, which was put forth by Karaman and Frazzoli in 2000.

The produced pathways may not be ideal, nevertheless, because of the original RRT method. In 2010, Karaman and Frazzoli [2] devised the RRT* method to overcome this problem. In order to increase the number of optimum pathways, RRT* builds a more balanced tree and rewires the tree structure. The RRT* method is effective, but it still has delayed convergence, especially in situations with barriers that are complicated.

The RRT* - Smart method, which was introduced by Jauwairia Nasir et al. in 2013 [3], is another development that we will be concentrating on. This algorithm uses heuristic information to direct sampling toward the objective region and speed up convergence.

In terms of computing efficiency and route quality, RRT* - Smart has been demonstrated to perform better than previous RRT*-based algorithms. Throughout this project, we will make reference to a number of pertinent works, especially the work of Nasir J et al. [3].

2 Goal

The goal of the project is a written technical report and presentation on the practical implementation of RRT*-Smart. Below is the list of complete goals.

1. Implementation of RRT*-Smart, a cutting-edge sampling based path-planning technique presented by Nasir J et al. [3], in simulation.
2. A 6-8 page technical report in LaTeX written in two column IEEE conference format describing introduction/motivation, a brief background and related work, methodology including high-level pseudo code and system details, tests/demos/experiments that were run, discussion, conclusions, and bibliography.
3. In class presentation with 10-12 slides.

The simulation will be performed using Gazebo and OpenCV. The primary goal will be the implementation of RRT*-Smart that is capable of

precisely and efficiently guiding a robot in a given area with obstacles. The algorithm's performance will be assessed using a variety of measures, including path length, computing time, and obstacle avoidance.

3 Method

The path planning method is sampling-based and it will be forward search using RRT*-Smart. The title of the paper that we will be implementing is "RRT*-SMART: A Rapid Convergence Implementation of RRT*".

Below are the software packages and libraries in Python that we plan to use.

- Gazebo
- ROS
- Rviz
- Numpy
- OpenCV
- Heapq

4 Timetable

1. Literature Review (5 days)
 - Reading and analyzing papers (3 days)
 - Understanding key concepts and algorithms (2 days)
2. Software setup and simulation environment (5 days)
 - Installing software and packages (1 day)
 - Setting up the simulation environment (4 days)
3. Implementation of RRT* - SMART (20 days)
 - Implementing the algorithm (12 days)

- Testing in simulation (8 days)
- 4. Evaluation and optimization (5 days)
 - Evaluating algorithm performance (2 days)
 - Optimizing the algorithm (3 days)
- 5. Documentation and report writing (6 days)
 - Writing the documentation (1 day)
 - Writing the report (3 days)
 - Preparing the presentation (2 days)

References

- [1] LaVelle and Kuffner bibliography
- [2] Karaman and Frazzoli bibliography
- [3] Nasir J, Islam F, Malik U, et al. RRT*-SMART: A Rapid Convergence Implementation of RRT*. International Journal of Advanced Robotic Systems. 2013;10(7). doi:10.5772/56718