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 Rapidly Exploring Random Tree Star Smart (RRT*-Smart) algorithm in an obstacle-filled environment. It is a modified version of the Rapidly Exploring Random Tree Star (RRT*) algorithm and it has been shown to be effective in path planning problems. RRT*-Smart employs a novel approach that combines two new techniques in addition to RRT*, path optimization and intelligent sampling.

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. It 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 from a randomly sampled nodes. The 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. The method has been extended and modified by a number of researchers to enhance the performance of RRT. One such modification is RRT-Connect method, which was put forth by Karaman and Frazzoli in 2000. This version constructs two trees from the beginning and goal configurations and links them to produce a path.

The produced pathways may not be optimal, nevertheless, because of the original RRT method. In 2010, Karaman and Frazzoli devised the RRT* method to overcome the optimality problem. In order to increase the number of optimum pathways, RRT* builds a more balanced tree and rewires the tree structure. The algorithm achieves convergence towards the best solution, assuring asymptotic optimality as well as probabilistic completeness. Yet, it has been demonstrated that it is time consuming and has a slow convergence rate especially when there are complicated obstacles.

As a result, RRT*-Smart was proposed to address limitations of RRT*. It was introduced by Jauwairia Nasir et al. in 2013 [2]. The algorithm uses heuristic information to direct sampling toward the objective region. Thus, it accelerates the rate of convergence in order to arrive at an optimal or near-optimal solution considerably faster, thereby lowering execution time. The algorithm's innovative method takes use of two new approaches in addi-

tion to RRT*, path optimization and intelligent sampling, which we will be implementing as our final project.

In terms of computing efficiency and route quality, RRT*-Smart has been demonstrated to perform better than previous RRT-based algorithms.

2   Goal

The goal of the project is the practical implementation of RRT*-Smart and a written technical report, and presentation. 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. [2], 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*" by Nasir, J et al [2].

The algorithm is as shown in Figure1.

T = (V,E) ← RRT*-Smart(z_{ini})

1 $T \leftarrow$ InitializeTree();
2 $T \leftarrow$ InsertNode($\emptyset$, z_{init}, T);
3 **for** i=0 to i=N **do**
4 **if** i=n+b, n+2b, n+3b…. **then**
5     $z_{rand} \leftarrow$ Sample(i, $z_{beacons}$);
6 **else**
7     $z_{rand} \leftarrow$ Sample(i);
8 $z_{nearest} \leftarrow$ Nearest(T, z_{rand});
9 (z_{new}, U_{new}, T_{new}) $\leftarrow$ Steer ($z_{nearest}$, z_{rand});
10 **if** Obstaclefree(z_{new}) **then**
11     $z_{near} \leftarrow$ Near(T, z_{new}, $|V|$);
12     $z_{min} \leftarrow$ Chooseparent (z_{near}, $z_{nearest}$, z_{new});
13     $T \leftarrow$ InsertNode(z_{min}, z_{new}, T);
14     $T \leftarrow$ Rewire (T, Z_{near}, Z_{min}, Z_{new});
15 **if** InitialPathFound **then**
16     $n \leftarrow i$;
17   (T, $directcost$) $\leftarrow$ PathOptimization(T, z_{init}, z_{goal});
18 **if** ($directcostnew < directcostold$) **then**
19     $Z_{beacons} \leftarrow$ PathOptimization(T, Z_{init}, Z_{goal});
20 **return** T

Figure 1: The algorithm for RRT*-Smart. [3]

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 S, et al. "Rapidly-Exploring Random Trees : A New Tool for Path Planning."

[2] 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

[3] Noreen I, et al. "A Comparison of RRT, RRT* and RRT* -Smart Path Planning Algorithms". IJCSNS