Path Planning

Ritankar Mondal

May 26, 2024

1 Introduction

The Rapidly-Exploring Random Tree (RRT) algorithm is a popular path planning method used in robotics for navigating complex environments. It incrementally builds a space-filling tree that efficiently searches non-convex spaces. However, the basic RRT algorithm has limitations, such as generating non-smooth paths and not maintaining adequate spacing from obstacles. This report documents the implementation of the basic RRT algorithm and its enhancements: trajectory smoothing, ensuring spacing from obstacles, and a greedy approach that biases sampling towards the goal. We also evaluate the performance of these variants through extensive simulations.

2 Environment Setup

2.1 How to create Sections and Subsections

The environment is defined by a bounding rectangle and several fixed rectangular obstacles. The start and goal points are chosen within this environment:

- Bounding Rectangle: 100 units wide and 100 units tall.
- Obstacles: Four fixed rectangular obstacles, each approximately one-tenth the size of the bounding rectangle.
- Start Point: (10, 10)
- Goal Point: (90, 90)

The specific placements of the obstacles are as follows:

- ((20, 20), (30, 30))
- ((50, 50), (60, 60))
- ((70, 20), (80, 30))
- ((20, 70), (30, 80))

3 Basic RRT Algorithm (Part a)

The basic RRT algorithm iteratively samples random points in the environment, steers towards these points from the nearest node, and adds new nodes to the tree if the path is collision-free. The process continues until a node is found within a specified distance (goal tolerance) from the goal.

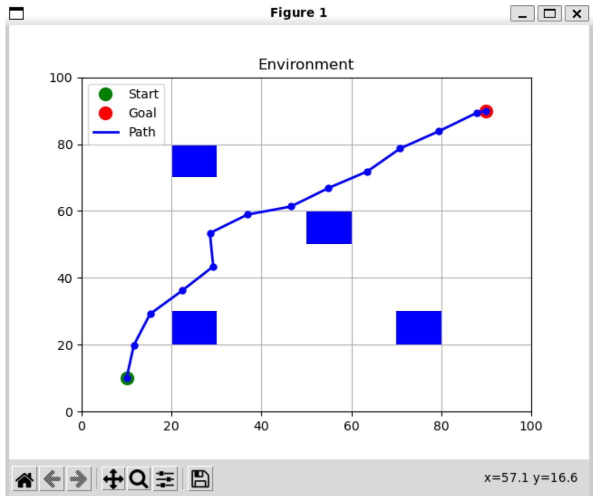

Figure 1: Basic RRT

4 Trajectory Smoothing (Part b)

The paths generated by the RRT algorithm are often non-smooth and contain unnecessary intermediate nodes, which are not feasible for real-world applications. Trajectory smoothing is applied to the path generated by the RRT algorithm to make it smoother and more feasible. This involves removing intermediate nodes that do not contribute to significant changes in direction while ensuring the path remains collision-free.

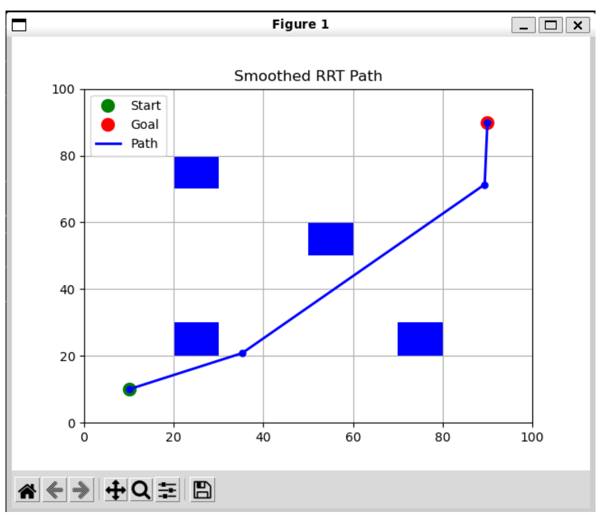

Figure 2: RRT after smoothing

5 RRT with Spacing from Obstacles (Part c)

To ensure the generated path maintains a safe distance from obstacles, the RRT algorithm is modified to bias random points away from obstacle edges by a specified spacing distance. This involves checking whether sampled points and the path segments are at a safe distance from obstacles before adding them to the tree. This ensures a safer path for navigation.

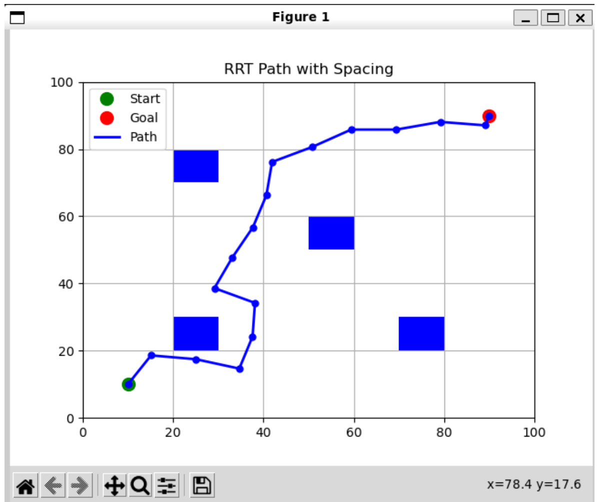

Figure 3: RRT with spacing

6 Greedy RRT Approach (Part d)

A greedy approach is introduced to bias the sampling of random points towards the goal. This approach increases the likelihood of finding a path more quickly by directing the search process closer to the goal. A goal bias parameter is used to control the probability of sampling the goal directly versus sampling a random point in the environment.

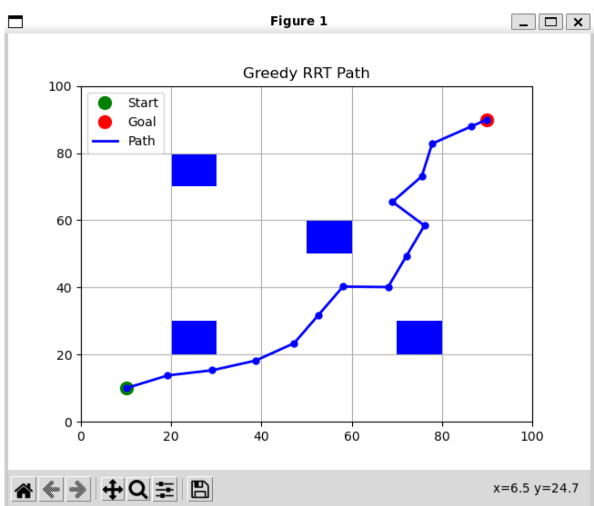

Figure 4: greedy RRT

7 Performance Evaluation (Part e)

The performance of each RRT variant is evaluated by running each algorithm 1000 times and recording the average number of nodes required to reach the goal. This evaluation helps to understand the efficiency and effectiveness of each enhancement.

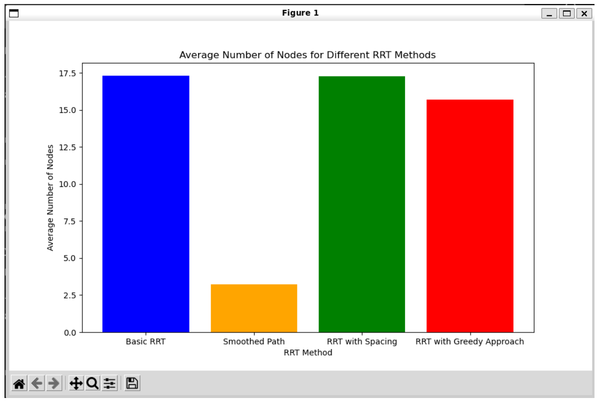

Figure 5: performace evaluation

8 Experimental Setup

- Number of Runs: 1000
- Max Distance: 10 units (maximum distance between nodes)
- Goal Tolerance: 5 units (distance within which the goal is considered reached)
- Goal Bias: 0.2 (probability of sampling the goal directly in the greedy approach)
- Spacing: 5 units (minimum distance from obstacles in the spacing variant)

9 Second Environment Setup and Evaluation

A second environment was created to further test the RRT algorithm, particularly focusing on the implementation of part (c) where random points are more likely to be spaced away from the edges of obstacles. The environment consists of two large obstacles with a narrow passage between them.

- Bounding Rectangle: 50 units wide and 100 units tall.
- Passage Width: 10 units.
- Wall Thickness: 1 unit.
- Obstacle Width: 24 units.
- Spacing: 2 units (minimum distance from obstacles).

The start and goal points are defined as follows:

- Start Point: (25, 1)
- Goal Point: (25, 99)

The RRT algorithm was used to plan a path from the start to the goal in this narrow passage environment, with the spacing constraint applied. The path found by the algorithm successfully navigated through the narrow passage while maintaining a safe distance from the obstacles.

The figure below shows the environment setup and the path generated by the RRT algorithm:

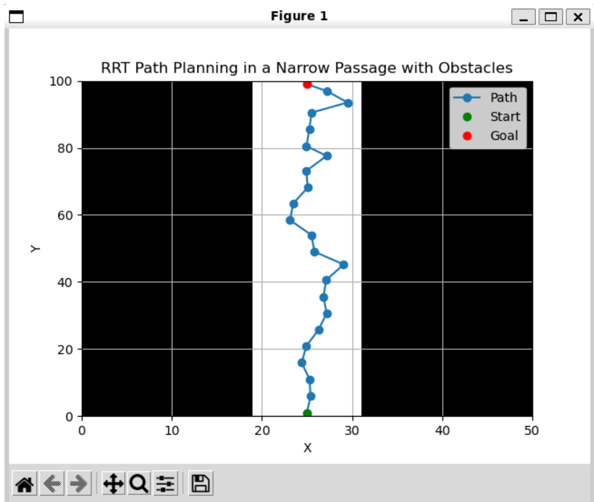

Figure 6: second evironment

10 Results

The results are summarized below, showing the average number of nodes required to reach the goal for each variant:

- Basic RRT: Average number of nodes = 17.389
- RRT with Smoothed Path: Average number of nodes = 3.227
- RRT with Spacing: Average number of nodes = 17.416
- Greedy RRT: Average number of nodes = 15.661

These values are bound to change with subsequent runnings of the algorithm.

11 Conclusion

The basic RRT algorithm provides a foundation for path planning in complex environments. However, the enhancements made to the algorithm address key limitations:

- Trajectory Smoothing improves the feasibility of the path for real-world applications.
- RRT with Spacing ensures a safer path by maintaining adequate distance from obstacles.
- Greedy RRT accelerates the search process by biasing the sampling towards the goal.

The performance evaluation shows that each enhancement contributes to a more efficient and practical path planning solution, suitable for use in various robotic applications. The choice of enhancement depends on the specific requirements of the application, such as the need for smooth paths or safe navigation near obstacles.

The additional evaluation in a narrow passage environment further demonstrates the effectiveness of the RRT with spacing approach in complex scenarios, highlighting its potential for practical robotic navigation tasks.