RRT and RRT* Algorithm Implementation

Om Vinayak Gaikwad [1]

1. RRT Algorithm Explaination

Rapidly Exploring Random Trees (RRT): Is a motion-planning algorithm that generates trees from the start to the goal point and finds an optimal path. The basic idea of the working of this algorithm is to first generate a set of nodes in the region of the start node, then generate a random point in the environment and find the nearest node to the generated random point among the set of nodes. Once we have the qnear the task is to extend one step toward the generated random point. This step is repeated and trees are generated based on the random point generated in the map when the random point reaches near the goal or on the goal point then the algorithm

2. RRT* Algorithm Implementation

The RRT* is an improved version of the RRT algorithm. It generates random trees in the environment based on the generated random point and attempts to connect the nearest node. This algorithm also performs optimization by using the rewiring function. This allows the path to be optimized even after we reach the goal. The rewire function takes the the new node (random generated node) and negihbors, first, it checks for a possible collision between the neighboring node and the new node if no collision is found the the cost and parent of the of the new node is updated.

3. Differences between RRT and RRT*

Even though both RRT and RRT* work on the same principles they have some differences. The major differences between them are: 1. RRT* uses an optimization method such as the rewiring method which ensures that the optimal path is found by comparing the cost of the new node with the neighbor cost. 2. While the optimization gives a better path in RRT* it also makes it computationally expensive. Therefore, RRT is computationally less expensive than RRT* 3. In RRT* we consider a certain neighbor size and choose the nodes within a certain distance are chosen where in RRT we get the nearest node based on comparing the distance with the node and the new point with a certain distance and if this distance is less then we update it.

4. Result comparison of RRT with PRM

In Probabilistic Roadmap the algorithm works by first building a roadmap in Cfree space by sampling in the entire Cspace whereas the RRT algorithm explores the space by randomly generating trees in the entire space until the goal node is found. The results produced in the previous assignment are based on four sampling methods: uniform, gaussian, random, and bridge however in RRT we generate a random point by taking in consideration the goal bias and comparing it with random uniform point. If it is less than the goal bias it is considered as the new random point. The RRT is algorithm is faster as compared to the PRM algorithm as it uses the same roadmap for constructing the tree whereas PRM uses different roadmaps after the query.

5. Results and Explaination

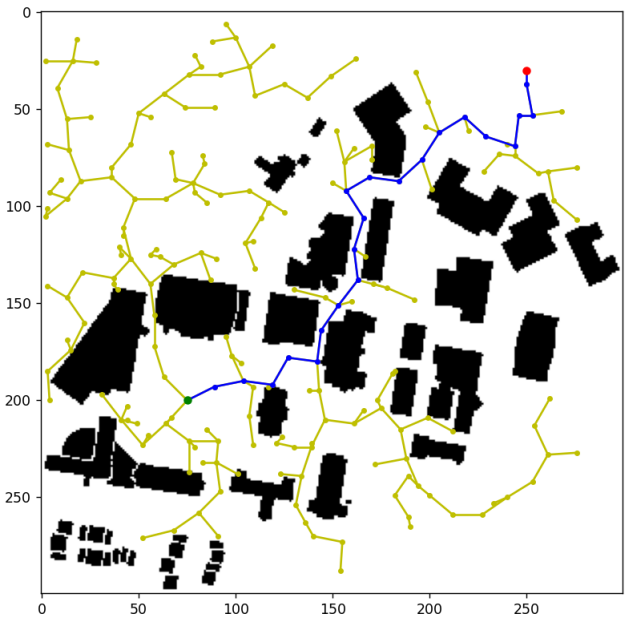

Figure 1. RRT Algorithm Output

Figure 1 shows the output of the RRT algorithm we can see that the one point from the start goal is randomly generated this is called the ”new point” then we find the qnear nodes to the point further we extend the node towards the new point and check collision which decides whether to add or

drop the node. If the node is added we check if it reaches the neighbor region of the goal.

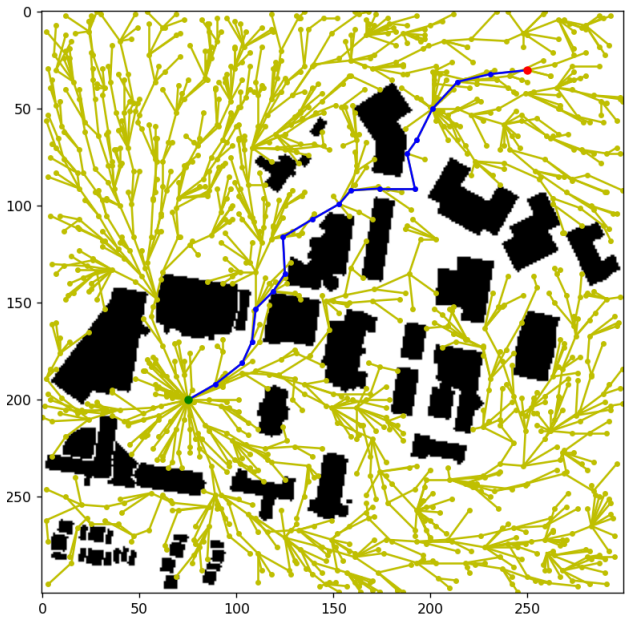

Figure 2. RRT* Algorithm Output

Figure 2 shows the output of the RRT* algorithm we can see that the one point from the start goal is randomly generated this is called the ”new point” the qnear nodes to the point then a certain neighbor size is defined and find the and compare the distance of the new node with the node and if its within the define neighbor size it is appended in that node in our neighbors list. Further the node is extended towards the new point and check collision which decides whether to add or drop the node. If the node is added the node and its neighbors are rewired to increase the optimality of the path.

References

[1] https://theclassytim.medium.com/robotic-path-planning-rrt-and-rrt-212319121378

[2] https://www.youtube.com/watch?v=Ob3BIJkQJEw

[3] Kevin M. Lynch and Frank C. Park: MODERN ROBOTICS MECHANICS, PLANNING, AND CONTROL.

[4] Introduction to Intelligent System (430.457, Fall 2016)

[5] Lecture Slides