Comparison Of Sampling Based Path Planning Algorithms - RRT, RRT*, RRT* Quick

11.04.2021

—

Paras Savnani, Pooja Kabra

UMD-Robotics

661 Planning for autonomous robots

Introduction

Definitions

Self-explanatory

Background

Conventional graph-based planning methods require us to discretize the search space. However, as the dimension of the search space increases, the memory requirement grows exponentially and the solution becomes more difficult to reach. Sampling-based algorithms are shown to work well and are employed extensively in a high dimensional space. Besides they are also theoretically proven to be probabilistically complete and asymptotically optimal. One such algorithm that has recently become abundantly popular in the path planning fraternity is Rapidly-exploring Random Tree star (RRT*).
In this project, we aim to implement RRT*-Quick, an improved version of RRT*. It claims to have a faster convergence rate without being computationally inefficient by building on the fact that in RRT*, the nodes in a local region have common parents. Also, as the cost of the best solution gets lowered, more nodes are ripped off by branch-and-bound, one of the key extensions of RRT*, therefore the solution gets improved more.

Literature review

Incremental Sampling based algorithms for optimal motion planning: This paper presented the thorough analysis of the RRT and RRG algorithms for optimal motion planning. If the number of samples increases the RRT converges To a suboptimal solution The results obtained in this paper can be extended in a number of directions and applied to a number of sampling based algorithms. A new algorithm RRG (Rapidly Exploring Random Graph) is introduced in this paper and it is shown that the cost of the best path returned by RRG converges to the optimum almost surely.

RRT* Quick: A motion Planning Algorithm with Faster Convergence Rate: This paper proposes RRT* Quick as an improved version of RRT*. It uses ancestor nodes to efficiently enlarge the pool of parent candidate for faster convergence rate. They provide statistical analysis of the proposed algorithm and experiments with a robot in the real envt. as a further research topic.

Anytime Motion Planning using the RRT: This paper describes an anytime algorithm based on the RRT∗which (like the RRT) finds an initial feasible solution quickly, but (unlike the RRT) almost surely converges to an optimal solution. They present two key extensions to the RRT∗, committed trajectories and branch-and-bound tree adaptation, that together enable the algorithm to make more efficient use of computation time online, resulting in an anytime algorithm for real-time implementation. Furthermore, they evaluate the method using a series of Monte Carlo runs in a high-fidelity simulation environment,and compare the operation of the RRT and RRT∗methods. They also demonstrate experimental results for an outdoor wheeled robotic vehicle.

Goal

Option: We will implement **Option 1:** Implement a cutting-edge technique from the literature in simulation or on a real robot using **sampling based path planning method**. For this we will simulate RRT, RRT*, and RRT* Quick algorithms and we will compare them based on their computational efficiency and convergence rate.

Simulation:

We will simulate these algorithms on real world images obtained from Google maps in pygame/matplotlib.

Outcome:

- Develop sampling based planning algorithms and implement them on a custom C-space.
- Compare the optimality of each algorithm
- Compare the computation time of each algorithm
- Simulate and visualize each algorithm

Method

Planning Algorithms:

RRT : RRT is an algorithm that works by randomly generating and connecting to the closest available node. Each time a new node is created it should be checked if it lies outside the obstacle space. Also, nodes must be connected avoiding the obstacles. The algorithm ends when the goal node is found. It is a non-deterministic algorithm, but given a good amount of iterations it converges to find a path although not optimal, but is quick. RRT is probabilistically complete i.e. it almost covers the search space at infinite samples.

RRT* : RRT* is an optimized version of RRT, when the number of nodes approaches infinity, this algorithm will output the shortest path possible. We saw that RRT simply connects the random sample to the nearest node in the tree. On the other hand, RRT* looks within a volume sphere with a certain radius to find the node which takes the cheapest path to get to the random sample. It reroutes the existing nodes through the random sample if it lessens their cost.

RRT* Quick : RRT* Quick algorithm is a modified version of RRT* with the benefits of RRT* and a high convergence rate. It enlarges the pool of effective parent candidates without increasing the computation time. For a just as many number of iterations the RRT* Quick algorithm will give much shorter and almost straight paths as compared to RRT/ RRT*.

In RRT, as the radius of the sphere grows, the solution improves with it. However, the number of nodes contained grows too slow down the algorithm. RRT*-Quick cashes in on the attributes of the shape of the tree to satisfactorily force the converge rate without taking up much computation time. The central idea of RRT*-Quick is based on the observation that nodes in local area tend to share a common parent because of the “choose parent and the rewire operation” of RRT*. As most commonly used metrics to measure the cost satisfy the triangular inequality, the parents are good candidates for the parent with the lowest cost.

Depending on the degree, RRT*-Quick can also be modified to consider the ancestors. Due to triangular inequality, the ancestors always provide a path with a lower cost. Even if it doesn’t, the time overhead is insignificant compared to that caused from a larger radius.

The shapes of trees of RRT, RRT*, and the proposed RRT*-Quick are shown below.

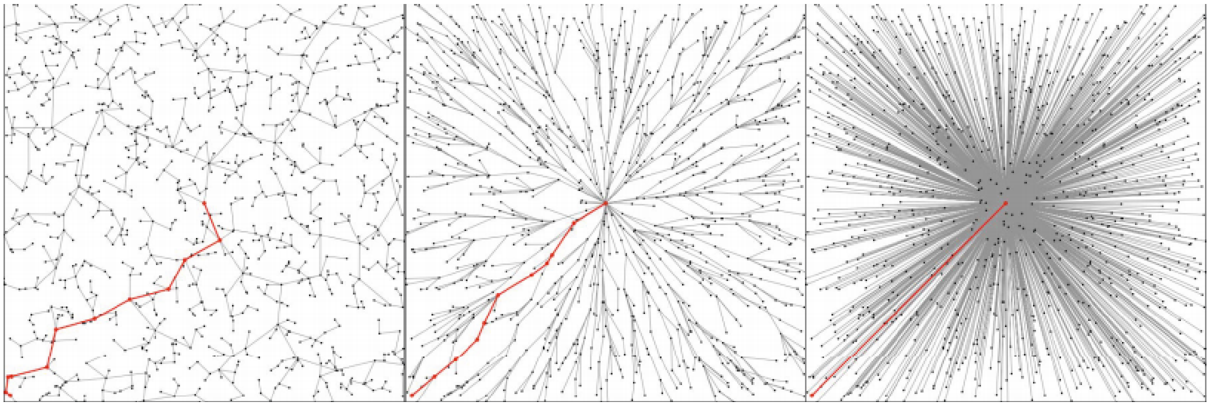

Fig. Extended Trees for RRT, RRT* and RRT* Quick after 1000 iterations

Research Papers:

- [RRT*-Quick: A Motion Planning Algorithm with Faster Convergence Rate]
- [Incremental Sampling Based Methods for Optimal motion Planning]
- [Anytime Motion Planning using the RRT*]

Software Requirements:

This project will be implemented in **python** and we will require following 3rd party libraries to write and visualize the path planning algorithm:

- Pygame
- Matplotlib
- Numpy
- Opencv

Hardware Requirements:

As we are not implementing this project on a real robot, we don't require a mobile robot platform.

Timeline

Milestone		Anticipated Deadline
Project Proposal		11th April
Defining and generating obstacle space		15th April
RRT	Implementation	16th April
	Visualization	18th April
RRT*	Implementation	20th April
	Visualization	22nd April
RRT* Quick	Implementation	28th April
	Visualization	30th April
Simulation in a ROS Gazebo environment		1st May
Project Report and Presentation		9th May

Future Work

If time permits we would like to implement following extensions with this project:

1. Simulate the planning algorithms on a turtlebot platform in a ROS Gazebo world.
2. Apply these algorithms on real world data like data from Google maps (This step will require preliminary image processing to generate the map from the images).
3. Implement BRRT* (Bidirectional RRT*) algorithm as another comparative technique.

References

[1] n-Bae Jeong, Seung-Jae Lee, and Jong-Hwan Kim- RRT*-Quick: A Motion Planning Algorithm with Faster Convergence Rate Robot Intelligence Technology and Applications 3 pp 67-76

[2] Sertac Karaman, Emilio Frazzoli- Incremental Sampling Based Methods for Optimal motion Planning

[3] Karaman, Sertac et al. Anytime Motion Planning using the RRT ."2011 IEEE International Conference on Robotics and Automation(ICRA) May 9-13, 2011, Shanghai International Conference Center,Shanghai, China.