3D Path Planning: Pruning with Constraint Satisfaction (7A)
Final Report

Kushagra Khare IMT2015022
Rachit Jain IMT2015034

November 21, 2018

1 Introduction

Our project was to built a path planning algorithm which creates paths for robots, self-driving cars, drones, etc considering various non-holonomic constraints.
So our main aim was to built a pruning method which which takes into account various constraints. So we first read the paper referenced [1]. In the paper author talks about a process as to how the problem can be solved. So we followed the process and also simultaneously finding better ways to solve a particular subtask.

2 RRT

RRT(Rapidly exploring Random Trees)[2] is an algorithm to search a space by randomly building a space-filling tree.The tree is constucted by randomly selecting samples and biased to grow in large unsearched areas. This algorithm searches the whole space to find the path to the destination as shown in figure 1.

3 RRT-A*

A* algorithm is a graph traversal algorithm,i.e it gives a path between 2 nodes. It uses heuristics to guide it's search which makes it better than Dijkstra's algorithm.
RRT-A*[3] is an algorithm where we take the cost function of A* to determine selection of nodes in the RRT algorithm. This algorithm has resulted in a much faster search algorithm (approximately 10x times faster than RRT). So we have implemented RRT-A* as we will be using this as a search algorithm between start and end points.The algorithm working can be seen in figure 2 and 3.

Figure 1: RRT

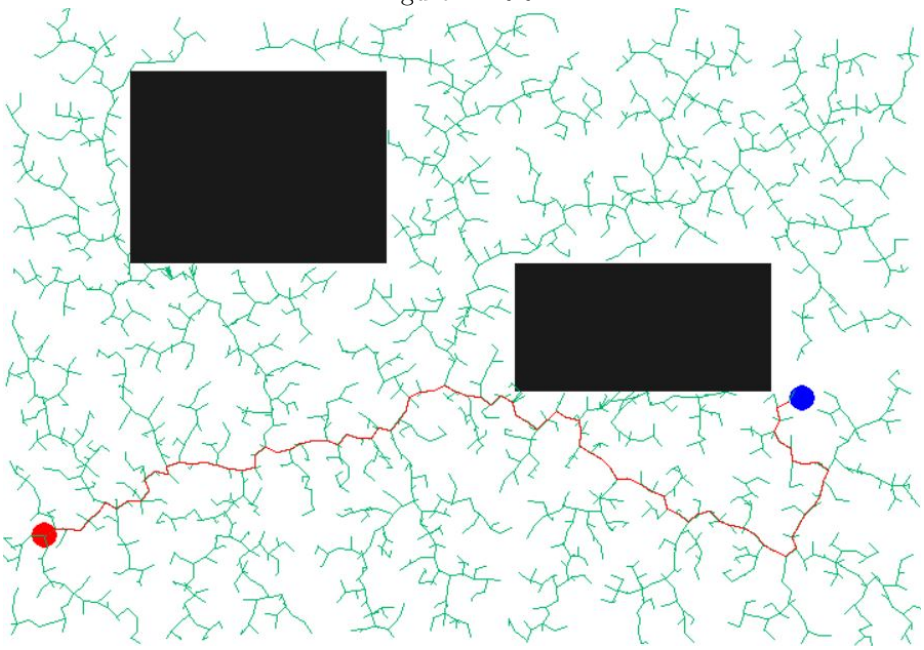
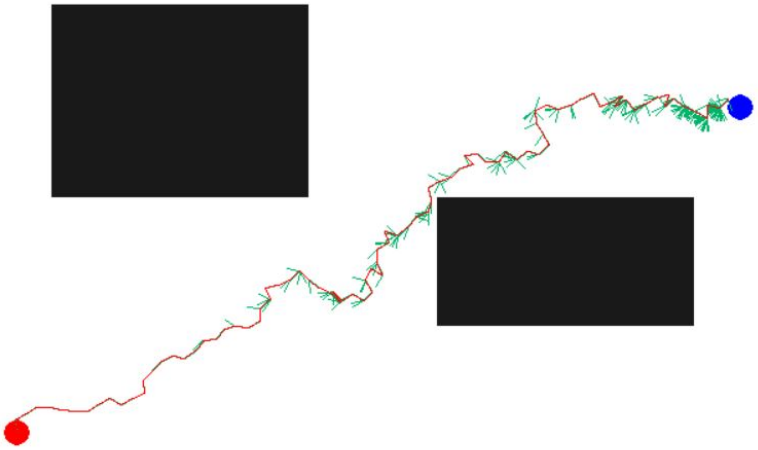


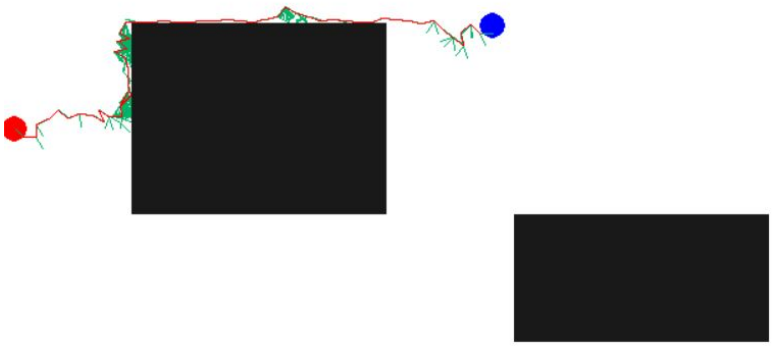
Figure 2: RRT-A*



4 RRT* Reeds-Shepp

So now we can combine Reeds-Shepp with RRT* to get the desired results,i.e, a path satisfying all non-holonomic constraints. As discussed earlier the equations

Figure 3: RRT-A*



for a car for non-holonomic constraints were:-
dx = v*cos(ϕ) $*$ cos(θ)
$dy = v * \cos(\phi) * \sin(\theta)$
$d\theta = (v/L) * \sin(\phi)$
$distance(p, q) = \sqrt{(p.x - q.x)^2 + (p.y - q.y)^2 + A * min[(p.\theta - q.\theta)^2), (p.\theta - q.\theta + 2\pi)^2, (p.\theta - q.\theta - 2\pi)^2]}$.

Here
L is the length of the car.
A = L*L.

After finding the nearest node (vn) of the randomized configuration (q), we have to find the optimal input(v, ϕ) to get the new node (qn) closest to q. This is a optimization problem, defined as following:-

distance(qn,q) $= \sqrt{(qn.x - q.x)^2 + (qn.y - q.y)^2 + A * min^2[(qn.\theta - q.\theta), (qn.\theta - q.\theta + 2\pi), (qn.\theta - q.\theta - 2\pi)}$
$Where$,
$qn.x = vn.x + v * dt * \cos(\phi) * \sin(vn.\theta)$
$qn.y = vn.y + v * dt * \cos(\phi) * \sin(vn.\theta)$
$qn.\theta = vn.\theta + v * dt * \sin(\phi)$

The above distance metric used is that of Reeds-Shepp algorithm. The combination of RRT* and Reeds-Shepp gives us a smooth, constraint and optimal path in real-time. Here now every time we select a random point we will also select a θ(direction) which will also decide the path to be taken.
But while implementation we faced some issues:-

- Due to very big resolutions screen the algorithm takes a lot of time to converge to a solution and also sometimes not converges to optimal path.
- Also there are a lot of paths to enumerate from to select an optimal path, which adds to the compplexity.

References

[1] Yang et al. (2013). Spline-Based RRT Path Planner for Non-Holonomic Robots.
https://link.springer.com/content/pdf/10.1007/s10846-013-9963-y.pdf

[2] Steven M. LaValle (1998). Rapidly-Exploring Randm Trees: A new tool for Path Planning
http://msl.cs.illinois.edu/ lavalle/papers/Lav98c.pdf

[3] Jiadong Li et al. (2014). RRT-A* Motion Planning Algorithm for Non- holonomic Mobile Robot
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=69353

[4] Bill Triggs. Motion Planning for Nonholonomic Vehicles: An Introduction
https://hal.inria.fr/inria-00548415/document

[5] J. A. Reeds AND L. A. Shepp. Optimal paths for a car that goes both forwards and backwards.
https://pdfs.semanticscholar.org/932e/c495b1d0018fd59dee12a0bf74434fac7af4.pdf

[6] Dubins Path.
https://gieseanw.wordpress.com/2012/10/21/a-comprehensive-step-by-step-tutorial-to-comp

[7] Jane Li. Non-holonomic Planning
http://users.wpi.edu/ zli11/teaching/rbe550$_2$017/$slides/9 - Non - holonomic\%20Planning.pdf$