3D Path Planning: Pruning with Constraint Satisfaction (7a)

Kushagra Khare IMT2015022
Rachit Jain IMT2015034

August 22, 2018

1 Goal

In this project, we aim to provide an algorithm for 3D Path Planning. We will implement a RRT-A* based 3D Path Planning algorithm. The algorithm would include path pruning with constraint satisfaction and account for non-holonomic constraints.
We will go ahead with Manhattan based RRT-A* in the initial stages but will also try to find an optimized distance metric function using Voronoi bias property for the algorithm.

2 Introduction

3D path planning is required in various applications such as robotics, self-driving cars, protein folding, games etc. It ensures to find a trajectory from the initial point to the destination, subject to rules of motion and any other constraints, such as collision avoidance, balance and joint limits.
Algorithms like Dijkstra, A* can be used but they are quite expensive to compute for large clustered space Random sampling based planning algorithm like RRT can solve motion planning problem while also taking the differential constraint into consideration. But the paths so produced are jagged, with several unnecessary branches. They need to be pruned and smoothed. An approach could be to fit a spline over the points which would produce a smooth path.

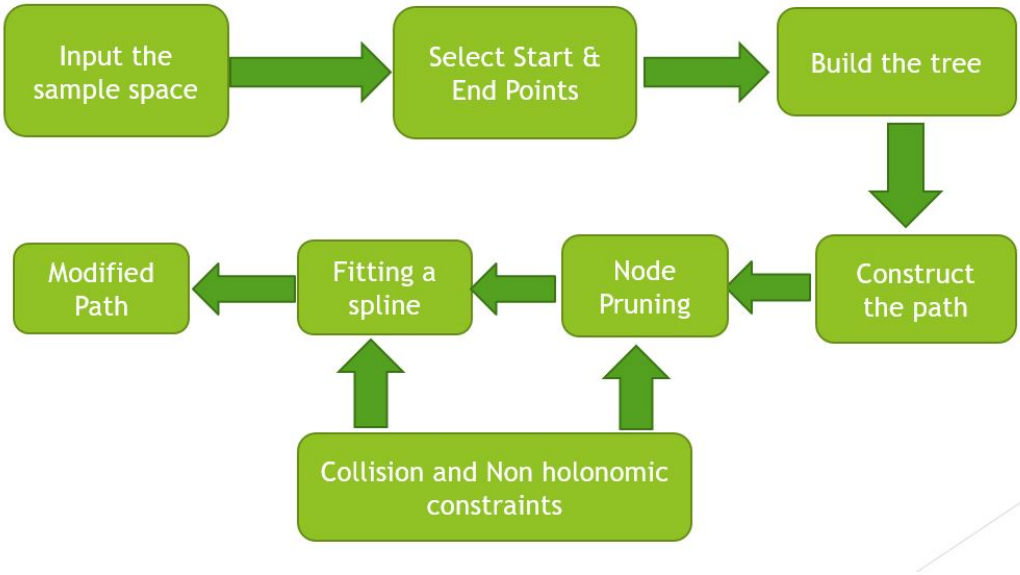
3 Gap Analysis

There has been some work done in this field. One of the attempts of Dynamic 3-D path following was done during 2004 was done by Conte et.al where they managed to create a way to fly a Yamaha RMAX Helicopter, but the limitation was that the focus was mainly on the hardware side and the software work done was very limited to the

hardware of the Yamaha RMAX helicopter. Then some work done in this field is by Yang where he proposed a path planner for non-holonomic robots, but there are some drawbacks to it:-
1) The algorithm used is RRT to which there was a paper in 2014 by Jiadong Li et.al where they improved the speed of the algorithm by 10 times by taking into account the cost function of A* algorithm. This algorithm is called RRT-A*.
2) There was not much focus on non-holonomic constraints. They talked about it on a superficial level but never clearly stated the various constraints that can be taken into picture.
3) The RRT-A* is implemented for 2D. So to extend it to 3D first we have to extend the A* algorithm to 3D, where the number of neighbours taken into consideration will increase from 8 to 26,then we have to use the cost function of 3D A* in RRT-A* to use it in 3D scenarious.

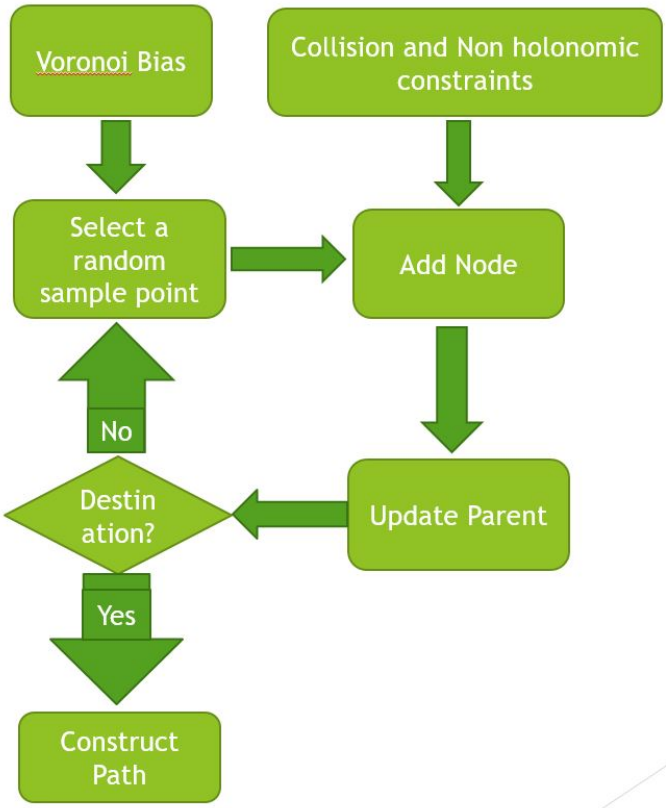
Our work will to create a 3D path planner with pruning with constraint satisfaction(including non-holonomic constraints). We will be adding a Voronoi Bias property to the RRT-A* algorithm to make path finding even faster.

4 Architecture



So to get a path first we take start and end points in the sample space as inputs. Then these inputs are used to build a tree(which has been discussed below). After the tree is constructed the best path is chosen which is then pruned and splines are fit, both of which are done by taking into consideration various constraints. Finally a modified optimal path is given as output.

4.1 Building the tree



To build a RRT, we first generate a random sample point. The voronoi bias property is used while sampling so that we can sample more towards the unexplored areas in the map. We then add node at step length in the direction of the sampled point. We then try to find out if the path to this node is collision free or not and whether it satisfies the holonomic constraints. We then update the parent of the node to the closest node in the tree using the distance metric function. We continue to sample the points until we reach the destination. Upon reaching the destination, we generate the path from source to destination.

5 Technical Requirements

Cloud: Microsoft Azure/Amazon AWS
Programming Language and Libraries: Python, NumPy, Scikit, PyGame, POGL(for 3D), TensorFlow
Dataset: Not decided

6 Milestones

S. No.	Objective	Deadline
1.	Project Report & Interface Setup	22-Aug
2.	RRT-A* algorithm	29-Aug
3.	Spline Based RRT-A*	12-Sep
4.	Node Pruning	19-Sep
	Mid Term Exams	
6.	Two phase sampling	14-Oct
7.	Voronoi Bias Property	14-Oct
	Mid Term Break	
8.	Non-Holonomic Constraints	5-Nov
9.	Analysis of Model	12-Nov

References

[1] Jiadong Li et al. (2014). RRT-A* Motion Planning Algorithm for Non-holonomic Mobile Robot
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=6935304

[2] 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

[3] Jun Qu. Non-holonomic Mobile Robot Motion Planning,
http://msl.cs.uiuc.edu/ lavalle/cs576$_1$999/*projects*/*junqu*/

[4] Gianpaolo Conte et al. (2004) Dynamic 3D path following for an autonomous helicopter
https://www.sciencedirect.com/science/article/pii/S1474667017320219