Standard Search Algorithms Implementation Report

Name: Aniket Patil Email: apatil2@wpi.edu

Important: Results are also saved as .png images in the "results" folder

Questions and Answers:

1. For PRM, what are the advantages and disadvantages of the four sampling methods in comparison to each other?

Answer:

Sampling Method	Pros	Cons
Uniform Sampling	• Uniform distribution means adequate coverage of the graph • Connections are also of uniform weights, simpler and efficient implementation	• Resolution is a key feature, that could result in incomplete solution
Random Sampling	• Random nodes sampled implies no bias	• Cannot introduce preference for free space or ignoring obstacles • May need higher number of samples to get a good number as valid samples
Gaussian Sampling	• Tackles the obstacle boundary conditions, without deviating away from the region through which the shortest path is found	• May not be efficient for concave or obstacles which occlude the path, forcing it into local minima
Bridge Sampling	• Tackles the obstacle boundary conditions successfully without having to sample outside the required area	• Focused on the gaps between obstacles, which means it may form two disconnected trees near obstacle clusters • High number or tried samples needed (n_pts)

2. For RRT, what is the main difference between RRT and RRT*? What change does it make in terms of the efficiency of the algorithms and optimality of the search result?

Answer: The main difference between the RRT and RRT* algorithms is the connection between the new sampled node and its neighbors. This connection occurs in two steps; First, the new sampled node is connected to a neighbor with the lowest cost. Second, the neighbors are rewired to check connections which may result in lesser cost values. This rewiring optimizes the output path found, even though the goal node may be reached in the form of a sub-optimal solution.

This results in a more optimal search result from the RRT* implementation, though it may be slightly more computationally expensive to rewire all neighboring nodes.

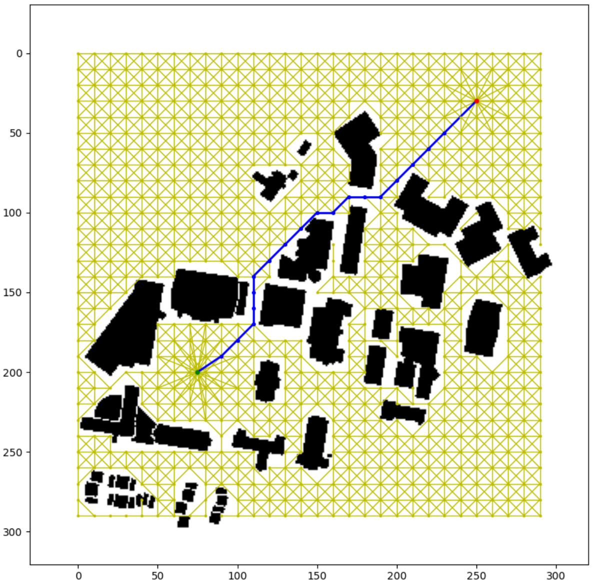
3. Comparing between PRM and RRT, what are the advantages and disadvantages?

Answer:

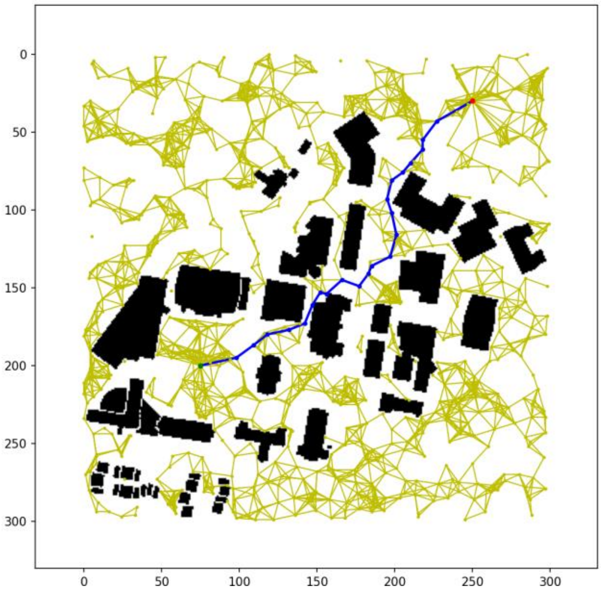
1. RRT is Tree based, whereas PRM is graph based algorithm. What this means is that PRM will sample the entire C-space and find nodes in the C-free (free space). RRT will always find nodes that can be connected to the tree.
2. Each node of RRT tree is always connected to the start node, which is not always true for PRM algorithm.
3. PRM sampling is done using different criteria, which are not based on the goal location. RRT explores the C-space for which we can give a goal bias that drives the exploration towards the goal node.
4. RRT is probabilistically complete, which means that if we go to T = infinity, we can find a solution if there exists one, or return that no path exists. PRM is not always complete, it may return no path found, due to various factors such as the K-nearest neighbor search radius or the resolution of uniform sampling.

Algorithm Results and Explanation:

PRM: This implementation uses 4 sampling methods: Uniform, Random, Gaussian and Bridge Sampling. The outputs received from these 4 sampling methods vary highly based on the sampling methods. Not only the method used, but also the parameters given to the sampling methods. I tried to concentrate these parameters to the top of the code in the form of static global constants.



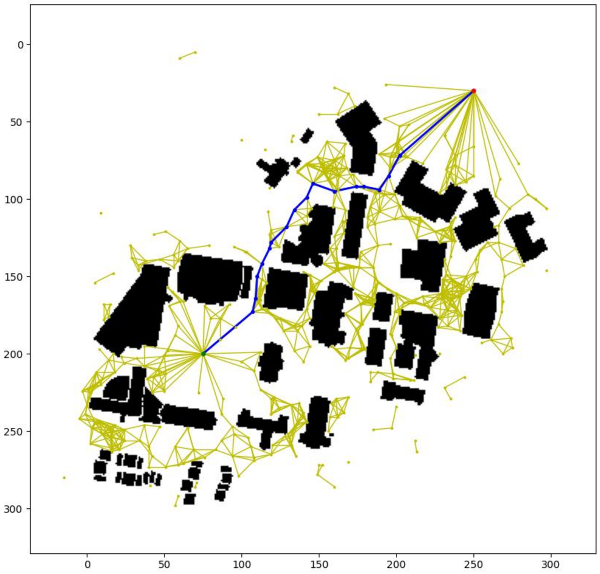
Uniform Sampling



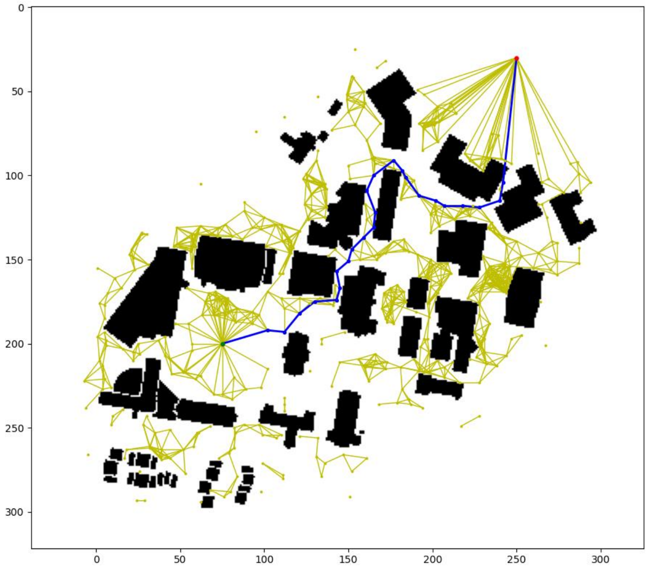
Random Sampling

Uniform Sampling: In Uniform sampling, the samples are evenly spaced throughout the map, this produces a grid like structure as seen in the output image. Since the nodes are evenly spaced and distributed, the path found is also along a straight line for most part of the solution. This is because the shortest path is found along the diagonals of the grid formed. The issue with this is that the resolution may make this method prone to small obstacles or narrow paths.

Random Sampling: In random sampling, the nodes are sampled in a random way and then connected using a nearest neighbors search and a check of whether the path in between two nodes is free or occluded. As expected, the graph for the random sampling is different for each run.



Gaussian Sampling



Bridge Sampling

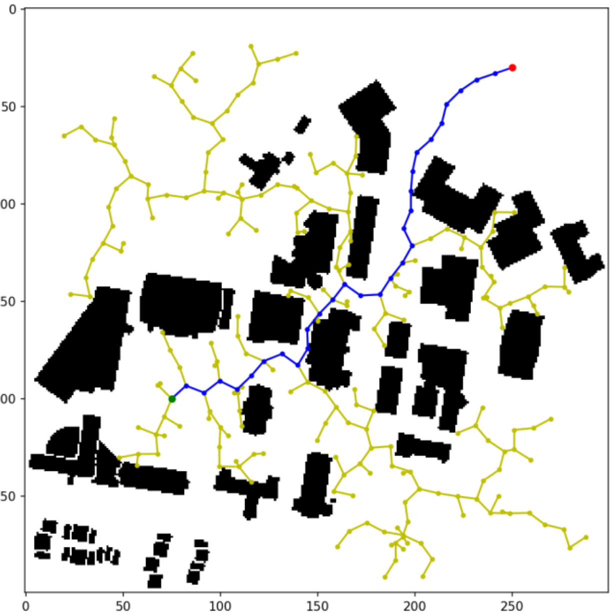
Gaussian Sampling: In Gaussian sampling, a random node (q1) is selected and then another node (q2) is picked at random from the gaussian distribution of the initial node (q1). Now checking the obstacles at location of q1 and q2, we decide which one to append to the tree. This means that if both q1 and q2 are free or if both are obstacles, we discard the points and start over. However, if one of them is free, we add that to the lift of valid vertices for the tree. As this method suggests, our intuition tells us that the gaussian sampling nodes will be focused more on the boundaries of the obstacles. This is what is seen in the output image for the gaussian sampling, that is, the nodes are not present out in the free space and are concentrated more near the obstacle space.

Bridge Sampling: In Bridge sampling, the nodes are placed in between two obstacle spaces. A random node is sampled, and another node is sampled using a probability distribution (Gaussian in this case). After this, we drop a sample at the mid-point of these and append it to the list if it is present in the free space. Here, we see that the nodes are placed in the regions between two obstacles. Due to this, the goal and start points (being in free space) need to be connected separately with neighboring nodes.

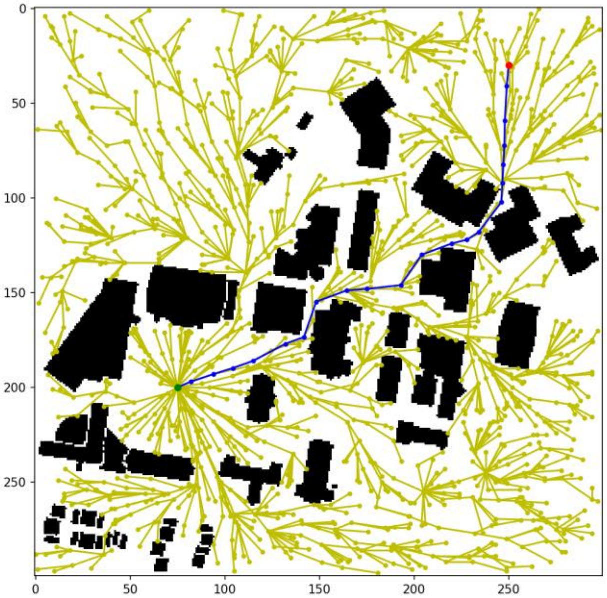
RRT vs RRT*: Both these methods produce different results; this can be seen in the images below.

The reason for these varied trees is that RRT* first adds the node to the tree by connecting it to the lowest cost neighbor. After that, the algorithm rewires the neighboring nodes to connect it to the current node, if this connection will reduce their cost. This rewiring step helps in reducing the nodes of the tree to lower cost neighbors. This ensures optimality of the planned path.

RRT can have two goal search methods, one where the tree stops exploring after the goal node is found and another where it keeps updating the cost of the goal node every time a shorter path to the goal is found. If we use the first method in RRT, that becomes a point of difference between the two algorithms as RRT* uses the second implementation method.



RRT Implementation



RRT* Implementation

Reference Papers and Resources:

1. A good RRT and RRT* Algorithm resource: https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-410-principles-of-autonomy-and-decision-making-fall-2010/lecture-notes/MIT16_410F10_lec15.pdf
2. Animations for understanding the RRT Implementation: https://www.youtube.com/watch?v=gP6MRe_IHFo
3. https://rrwiyatn.github.io/blog/robotik/2020/05/29/prm.html