RBE 550 - Programming Assignment 3

Abizer Patanwala

February 23, 2023

1 Advantages and Disadvantages of 4 sampling methods

- **Runtime**: Runtime of bridge sampling method in environment where there are less number of obstacles is higher than other three since we have to find two points both lying on the obstacles.
- **Environment**: For environment with a large number of obstacles with narrow passages, both bridge based sampling and gaussian sampling produce better results than other 2. This is because in gaussian sampling points which lie near obstacle and are in passage will be found, and in bridge based sampling points in between obstacles and lying in the passage will be found. In this scenario among the gaussian based and bridge based sampling, bridge based sampling will give better results, since it finds points in between obstacles which have a higher chance of lying in the passage. For environment with less number of obstacles it is advantageous to use uniform sampling since it takes less time and the path can easily be found.
- **Connectivity**: Connectivity of gaussian based sampling and bridge based sampling is inferior to uniform and random based sampling, since the points in the former two methods are distributed around obstacles. This may present problems when trying to connect start and goal nodes which are far from obstacles. Connectivity of uniform sampling is better than the other three in environments with less no of obstacles

2 Algorithm explanation and result

PRM is an algorithm for creating roadmap. Its basic working is as follows: First it finds points in the configuration space using a sampling method.Then it creates a roadmap by conneting each point to its nearest neighbours such that the edge doesn't intersect obstacle. This creates a roadmap. To find a pth between start and goal node, start and goal nodes are added to the roadmap. Then any graph based search algorithm can be used to find a path between start and goal node. The PRM psuedocode is shown in Figure 1.

2.1 Sampling algorithms

Uniform Sampling: It samples points uniformly across the configuration space which results in a good connectivity.
Random Sampling: Samples points randomly in a configuration space
Gaussian Sampling: First samples a point q1. Then finds a point q2 using gaussian sampling. If q1 and q2 both lie on obstacles or both are obstacle free, it is discarded. Else the point which is obstacle free is added.
Bridge Sampling: Sample a point q1 in the obstacle space. Sample another point q2. If q2 also lies in obstacle space, check if midpoint is in free space. If so, add the midpoint, else discard it.

For the implementation, the number of points to search for are chosen as 1000. The number of nearest neighbours to search for, while creating the roadmap, is chosen as 10. The number of nearest neighbours to search for, while connecting the start and goal node to the roadmap, is chosen as 20. The implementation results are shown in Fig.2, Fig.3, Fig.4, and Fig.5. Fig 2 shows resultant path when using uniform sampling for creating roadmap. We can see that the points in the roadmap are evenly spaced in the roadmap, as is the nature of uniform sampling. The points cover the entire environment. Fig 2 shows resultant path when using random sampling for creating roadmap. The points in the roadmap are randomly distributed, because of random nature of sampling. Fig 3 shows resultant path when using gaussian sampling for creating roadmap. We can see that the points are sampled near the obstacles which justify the working of gaussian sampling. Fig 4 shows resultant path when using bridge based sampling for creating roadmap. When using bridge based sampling points in between the obstacles are sampled, as evident from Fig 4. The resultant path when using bridge based sampling and gaussian sampling is in between the obstacles, since maximum points lie near the obstacles. The path of random sampling goes above the obstacle which lie just before the goal as evident from Fig 2. This is due to random nature of sampling.

Algorithm 6 Roadmap Construction Algorithm

Input:

n : number of nodes to put in the roadmap

k : number of closest neighbors to examine for each configuration

Output:

A roadmap $G = (V, E)$

```
V ← ∅
E ← ∅
while |V| < n do
  repeat
    q ← a random configuration in Q
  until q is collision-free
  V ← V ∪ {q}
end while
for all q ∈ V do
  N_q ← the k closest neighbors of q chosen from V according to dist
  for all q' ∈ N_q do
    if (q, q') ∉ E and Δ(q, q') ≠ NIL then
      E ← E ∪ {(q, q')}
    end if
  end for
end for
```

Figure 1: Psuedocode of PRM [1].

2.2 Working of DFS

References

[1] Choset, Howie, et al. Principles of robot motion: theory, algorithms, and implementations. MIT press, 2005.

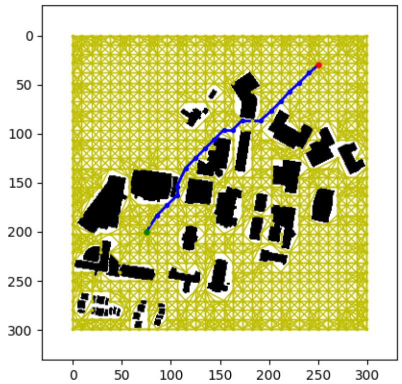

Figure 2: Resultant path when using uniform sampling for creating roadmap.

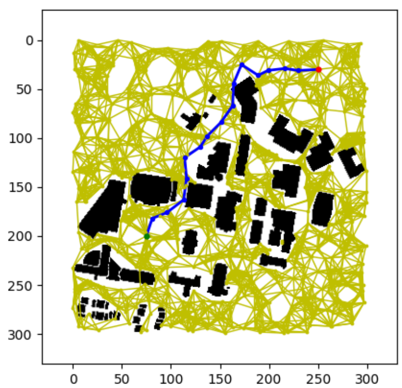

Figure 3: Resultant path when using random sampling for creating roadmap.

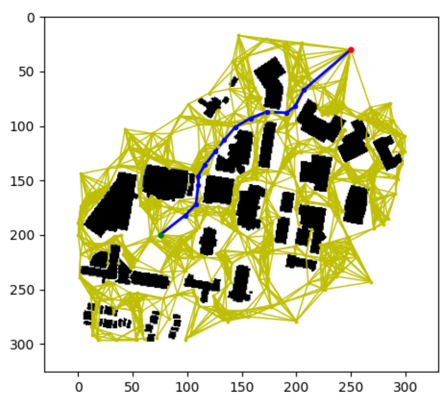

Figure 4: Resultant path when using gaussian sampling for creating roadmap.

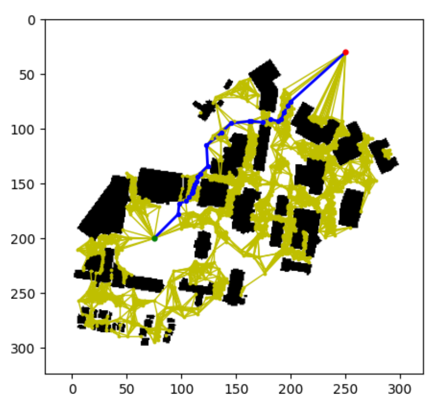

Figure 5: Resultant path when using bridge sampling for creating roadmap.