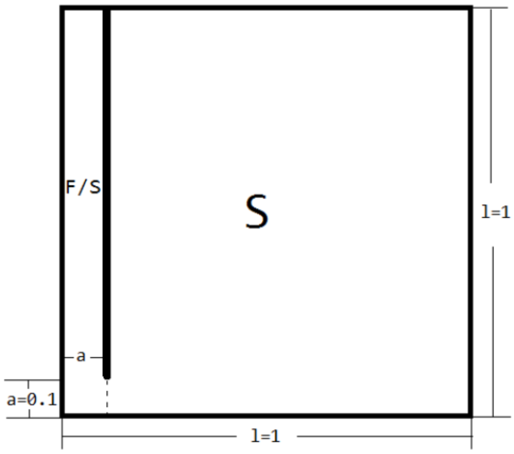
COMP550 Homework 3

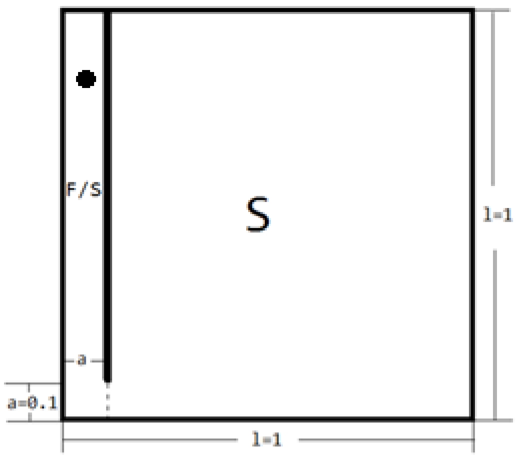
COMP550
Haoran Liang (hl74)
Nov, 13 2019

1. **An assumption when using a PRM based approach for motion planning is that the configuration space is not *pathologically* difficult.**



(a) Estimate the $\varepsilon - goodness$ of the space, i.e., what is the largest ε such that this space is $\varepsilon - good$

We know that ε stands for the least fraction of the free space that can be seen by every free configuration. Thus, if we have a free configuration at the top of the F/S, then it can only see the F/S space. Show as the image:

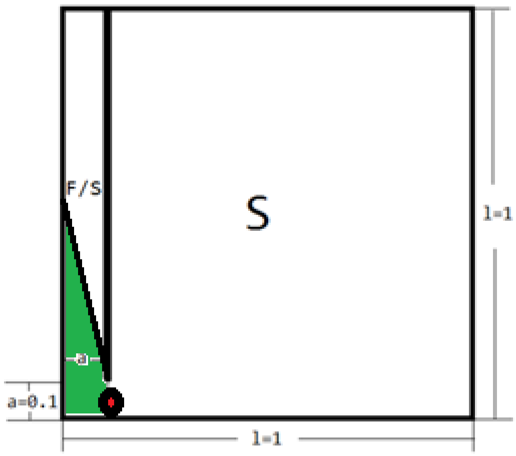


Based on the given value, we can compute that $\varepsilon = 0.1 \times 1 = 0.1$

(b) Alice the world-renowned roboticist claims that the 0.4-lookout of S is a 0.6 fraction of S. Do you think this is true? Why or why not?

False.

If $\beta = 0.4$, we can draw a set of free configurations in S which is able to see at least 0.4 fraction of F/S. The set looks like:



The green part is about 0.4 of the entire Free Space. The red point indicating the points in S that can see the green part. Based on the equation we know that $\alpha = \frac{Volumn(\beta - lookout)}{Volumn(S)}$. So that it is impossible to have $\alpha = 0.6$.

(c) For a space with low $\varepsilon, \beta, \alpha$-expansiveness, what kind of sampling techniques would you expect to work well? Explain.

If we have a low expansiveness, there would be more cost to construct a roadmap without good connectivity and coverage. Thus, this might lead obstacles having a narrow passage. Therefore, an **Expansive-Spaces Trees** would be a good solution to solve such problem since EST has two trees at initial point and goal point. Two trees will grow up toward each other, and they will finally merge in to one tree.

(d) Imagine that you extrude this configuration space to 3 dimensions. How will $(\varepsilon, \beta, \alpha)$ values be affected by this change?

The $(\varepsilon, \beta, \alpha)$ values are not affected by the change. We know that the 3D configuration space composed of stacked 2D slices. Therefore we can integrate the y-axis of the 3D configuration to several 2D configuration. For each 2D slice, the $(\varepsilon, \beta, \alpha)$ values are the same. Therefore, the entire 3D configuration will remain the same.

2. What is the configuration space of the robot? What is the control space of the robot? Is this a non-holonomic or a holonomic robot?

(a) What is the configuration space of the robot?

Since the robot can translate and rotate in 2D space, the configuration space for the robot is $R^2 \times SO(2) = SE(2)$.

(b) What is the control space of the robot?

The control space of the robot is v_r, the right wheel turning rate and v_l, the turning rate of left wheel. So the control space is $R^1 \times R^1$.

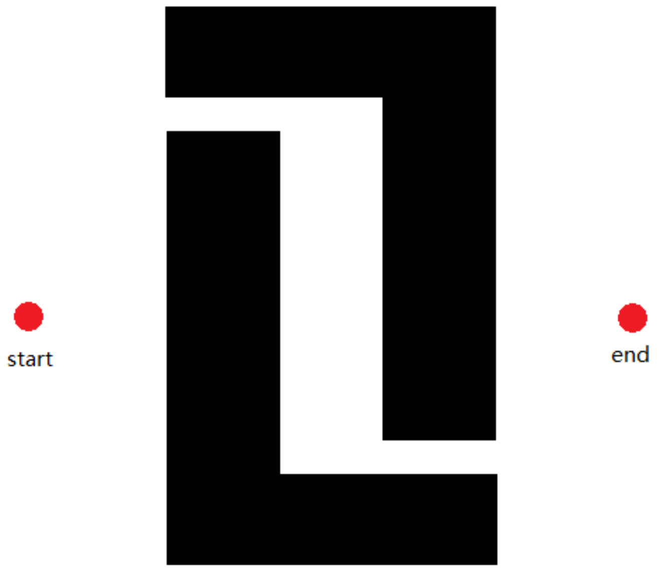
(c) Is this a non-holonomic or a holonomic robot?

The robot is **non-holonomic robot** since we have 3 parameters of the configuration space and only 2 parameters of control space. So we cannot describe the position and turning direction by only knowing the turning rates of two wheels.

3. Compare visibility graph and PRM. For each method, provide at least one scenario in which it would work well while the other would not.

(a) Visibility graph beats PRM:

If we have a space that contains narrow space between two obstacles, like the image shows.



For visibility graph, we know that it is complete. So that we could always find the shortest path between two points even there is a narrow path between two obstacles. However, since PRM is not complete, it may not return the shortest path between two points, or it may take longer time to sample configurations in that narrow space. Thus, under this situation, we can say that visibility graph can work well and PRM would not.

(b) PRM beats visibility graph:

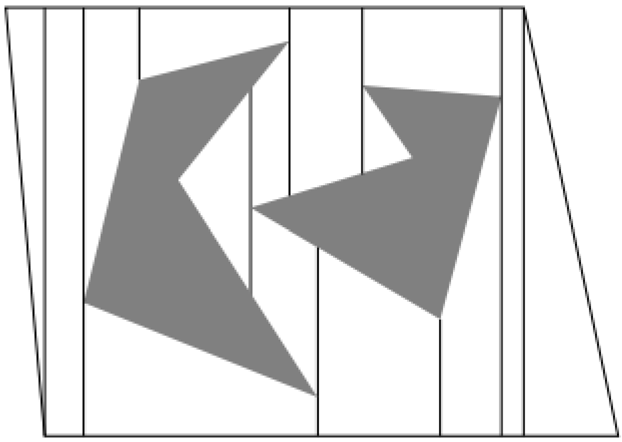
We know that visibility graph will try to connect each vertices of all obstacles. Therefore, if we have a lot of obstacles in a space, there will be a lot of edges connected between vertices. However, most of these edges are useless and connecting them will consume a lot of time. Therefore, if we have lots of polygon obstacles, choosing PRM to find shortest path would be a better choice comparing using visibility graph.

4. Explain why the Brushfire Algorithm does not scale to higher dimensions.

The basic idea of Brushfire Algorithm is to construct the shortest path between the start and the goal location. The algorithm will iterate through the graph and find all adjacent nodes of the current node, and mark the distance value to any unvisited adjacent node. So that we can consider the Brushfire Algorithm as applying Breath First Search to a graph, and this will visit all nodes in a graph. Consider we have 100 x 100 grids in 2D space, then we have 10000 nodes need to visit. If we increase the dimension to 3, then we have 10^6 nodes need to be visited. So if we increase the dimension, the number of nodes need to be visited grows exponentially. Then it will cost lots of time to use Brushfire Algorithm in high dimension. Therefore, Brushfire Algorithm does not scale to higher dimensions.

5. Bob wants to design a motion planning for a mobile manipulator which has a base can move freely in the plane and has a 6 degree-of-freedom arm. He has trapezoidal decomposition planner and RRT planner available. Which planner would you recommend to Bob for each of the components.

We know that the mechanism of trapezoidal decomposition planner is to divide the space into several subspaces based on the obstacle vertices. So that we can form a graph looks like this:



Thus, we can easily find two points in adjacent cells and connect them using a straight line. We can also find which cell contains the start point and which cell contains the goal point. After that we can do a path planning through the graph between start and goal nodes. So after we form a graph containing the trapezoidal cells, we can reuse it for different start and goal points. Therefore we can apply trapezoidal decomposition planner to the base robot since there is no need to change the graph after we setting up the trapezoidal cells.

For the 6 degree-of-freedom arm, the configuration space of the arm is translating and rotating in 3D space. It is a little bit time consuming and complex to extend the trapezoidal decomposition planner in a 3D space. Also, if we change the goal location of the arm, we have to rebuild a graph to re-initialize the trapezoidal cells. So in 3D space translating and rotating, using RRT planner is a better choice.