ROBOT AUTONOMY SCRIBE

Scribe 1: Siddharth Raina andrew id: sraina1
Scribe 2: Sumit Saxena andrew id: sumits1
Scribe 3: Akshay Bhagat andrew id: abhagat1
Date: Feb 15 2017

SAMPLING BASED PLANNING

In sampling based planning, instead of building a configuration space we follow a sample based approach, where we sample a point and check if its in collision. A collision detector function probes points corrosponding to a geometric model and returns a corrosponding value based on whether the point is in collision or not.

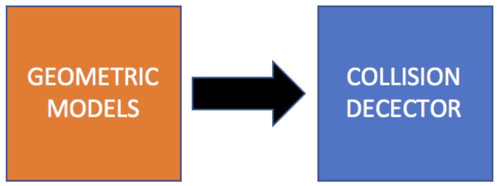

Figure 1: SAMPLE BASED PLANNING

Probe returns 1 if $q \in C_{obstacle}$
Probe returns 0 if $q \in C_{free}$
Both self collisions and environment-object collisions are checked by the probe.

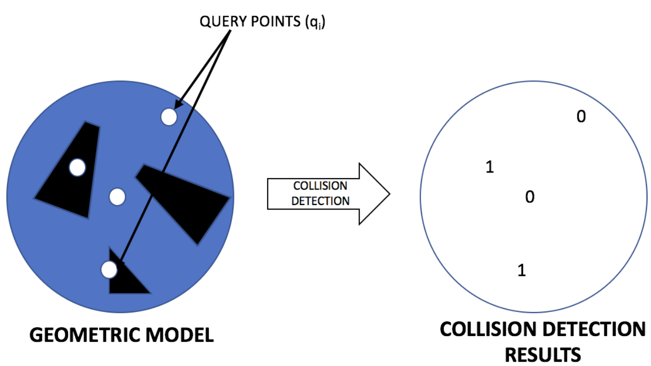

Figure 2: COLLISION DETECTION AND RESULTS ON A GEOMETRIC MODEL

However, using a probe is computationally expensive and a fixed resoultion probe can overlook certain points (for ex. obstacles of sizes less than the resolution of the probe)

THE PROBABILISTIC ROADMAP

If the collision checker is queried multiple times, It is a good idea to create a probabilistic roadmap. In general, the multi-query methods work in two stages:

1. **Preprocessing:** Even before a single query is made, a probabilistic roadmap is created. The probabilistic roadmap is an undirected graph G, whose nodes belong to C_{free} space
2. **Query:** In the query phase, the robot is given start and end configurations and it searches the roadmap to connect the two configurations.

However, one disadvantage of creating a probabilistic roadmap is that it is not robust enough to tolerate changes in the environment.

CHECKING WHETHER AN EDGE HAS A COLLISION

Given two nodes of a roadmap there are two ways of checking whether the edge between the nodes is collision free.

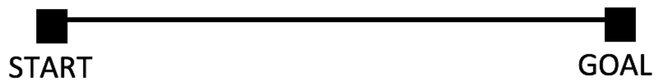

Figure 3: START AND GOAL CONFIGURATIONS

LINEAR SWEEP

In a linear sweep, the planner moves linearly in steps along the edge between two configurations till it reaches the goal or encounters a collision. This method has several advantages:

1. Even if it does not reach the goal position, it can find out a collision free patch which can be cached in its memory and used later.
2. It takes smaller number of steps as compared to the bisection test.

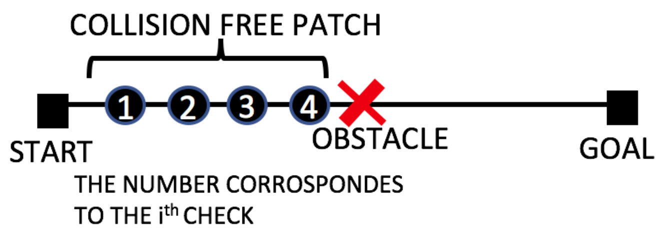

Figure 4: LINEAR SWEEP

BISECTION TEST

In a bisection , a planner performs a binary search for feasible segments between the two configurations. However it has the follwing disadvantages:

1. It is based on the assumption that obstacles are adjacent points and so are the free segments.
2. It gives no intermediate solution.
3. It is computationally more expensive than a linear sweep of the same resolution.

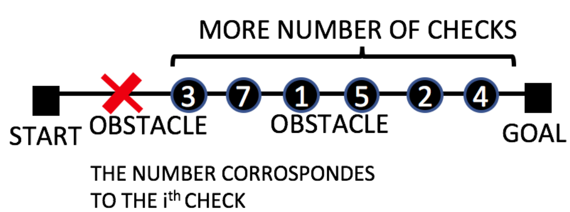

Figure 5: BISECTION TEST

CREATING THE ROADMAP

Following are the steps to create a roadmap:

1. **SAMPLING:** Sample random milestones in C_{free}.
2. **SIMPLE PERMEATION:** Try to connect all points with the simple planner β_s.
3. **REPAIR:** Try to connect all the milestones to the start configuration and the end configuration. If connected , hop on to the roadmap. The preservation of the network will ensure the connectivity between the start configuration and the goal configuration.

ASYMPTOTIC OPTIMALITY

As the number of milestones increase, the probability of a path between the start configuration and the end configuration reaches close to 1.

CONNECTIONS IN A ROADMAP

The connections in a roadmap can be made in the following ways:

1. **K NEAREST NEIGHBORS:** Connect to K closest neighboring configurations. A disadvantage of this method is that the planner might attempt to connect really far points which might not be collision free.
2. **CHECKING A RADIUS:** Connect to every point within a radius R and connect to it. The radius R is a critical parameter here, If R is too small, the connections will be faster but it would lead to the creation of a dense map. On the other hand, for very large values of R, the planner would be able to connect to distant points but the chances of getting a collision free edge are low.

CREATING A SINGLE QUERY PROBABILISTIC ROADMAP

1. Sample M milestones.
2. Add the start configuration and end configuration to the set of milestones.
3. Perform a simple permeation on this set of milestones.
4. If the start configuration and the end configuration are in the same connected graph, Plan the path in the graph.

REFERENCES

1. *https://personalrobotics.ri.cmu.edu/files/courses/16662/notes/rrt/16662_Lecture12.pdf*