Lecture 17: *Motion Planning I*

Scribe: *Duong Le, Keyu Han, Baiyu Huang, Seunghee Yoon*

1   Complete Motion Planning

1.1   Cell Decompose

1.2   Visibility Graph

2   Sampling-based Motion Planning

Sampling-based motion planning is not complete: not guarantee to find a solution when one exists. However, as the number of samples goes to infinity, there is a strong guarantee that it can find a solution.

There are 2 types of planning algorithms:

Multiple Query Algorithm
- Roadmap is built beforehand. Then the roadmap can be used multiple times for different queries.
- Query is fast since roadmap is already generated.
- Keeping roadmap all the time is expensive.
- Can only deal with static environment - has to rebuild the roadmap when environment changed.

Single Query Algorithm
- No roadmap is built beforehand. Find a path from start to goal then finish
- Query is slower.
- No saved roadmap - better memory
- Can deal with dynamic environment

2.1   Probabilistic Road Map (PRM)

Probabilistic Roadmap (PRM) is a multi-query algorithm.
There are 2 steps:
1. Preprocessing state: build roadmap
2. Query state: search roadmap given start and goal

2.1.1   Roadmap Construction

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 $G = (V, E)$

1: $V \leftarrow \emptyset$
2: $E \leftarrow \emptyset$
3: **while** $|V| < n$ **do**
4:     **repeat**
5:        $q \leftarrow$ a random configuration in Q
6:     **until** q is collision-free
7:     $V \leftarrow V \cup \{q\}$
8: **end while**
9: **for all** $q \in V$ **do**
10:    $N_q \leftarrow$ the k closest neighbors of q chosen from V according to *dist*
11:    **for all** $q' \in N_q$ **do**
12:       **if** $(q, q') \notin E$ **and** $\Delta(q, q') \neq \texttt{NIL}$ **then**
13:          $E \leftarrow E \cup \{(q, q')\}$
14:       **end if**
15:    **end for**
16: **end for**

Figure 1: Roadmap Construction Algorithm

Let $\Delta(q, q')$ be a function that returns either a collision-free path from q to q' or NIL if it cannot find such a path. There are many algorithms can be used for Δ. For example, if we want to check whether there is a straight-line path from q to q, we can use binary search to check for collision.

Figure 1 shows an algorithm to construct a roadmap:

- Initialize graph $G = (V, E)$ to be empty.
- From line (4) to (8) describes sampling process: keep sampling to get n numbers of collision-free samples.
- From line (9) to end describes process to build roadmap from n samples: for every sample node $q \in V$, create a set N_q of k closest neighbors. Then from every $q' \in N_q$, whenever function $\Delta(q, q')$ succeeds to compute a collision-free path from q to q', the edge (q, q') is added to the roadmap. Figure 2 shows an example of roadmap using Δ as straight-line planner.
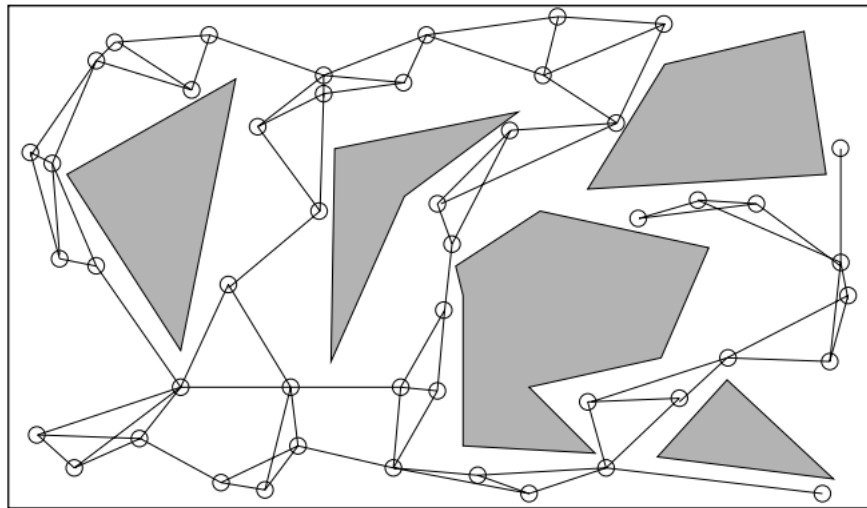

Figure 2: An example of a roadmap

2.1.2   Query

When you have a start point and end point, you connect them to the graph and then using graph searching algorithm to find path from start to end.

2.1.3   Failure

Some reasons to failure:
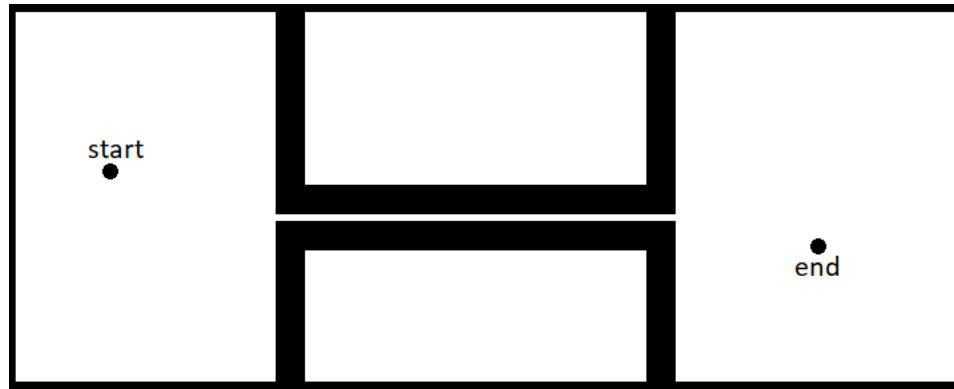
- Connectivity: disconnected graph

Figure 3: An example when obstacles are very closed to each other

- Obstacles are very closed to the others.

Solution: generate more samples.

But in the case when obstacles are closed to the others, the probability to sample between the obstacles is very unlikely. For example, in figure 3, the chance to sample points in the narrow road between 2 obstacles is very low. If you keep sampling more points, the graph will become so dense and will take a lot of time to find a path between start and end point.

Solution: sample more points toward edges of the obstacles. We can use binary search to find the closest point to the obstacle that's collision-free, and then sample more points near that point. The process of sampling becomes more expensive, but result graph is less dense and faster to compute path.

2.2   Visibility PRM