5.6. Roadmap Methods for Multiple Queries.

Suppose you will need solutions for many $(q_i, q_g)$ pairs.

Then it's worth spending extra compute time to construct a good approximation of $C_{free}$.

**Caveat:** $C_{free}$ can have many components; in worst case the # of components is exponential in dimension of $C$.

Build graph $G$ of $C_{free}$ such that:

1) LPM can easily connect any point to $G$.
2) $G$ is small so searching is fast.

General approach known as PRMs (Prob. Roadmap Methods)

(Lavalle calls them sampling-based roadmap methods)

2 Phases:

- Preprocessing phase - build $G$
- Query phase - connect $q_i$ & $q_g$, extract path

Preprocessing

0. Select # of samples, $N$ (from a sequence dense in $C$)
0. Select \# of samples, N (from a sequence dense in C)
1. \texttt{G.init(); i=0}
2. \texttt{while i < N}
3. \texttt{if \alpha(i) \in C_{free} then}
4. \texttt{G.add_vertex(\alpha(i)); i=i+1; (now call \alpha(i), \gamma)}
5. \texttt{for each \gamma \in \text{nbhd}(\alpha(i), \gamma)}
6. \texttt{if [(\text{not G.same-component}(\alpha(i), \gamma))}
   \text{and ( connect(\alpha(i), \gamma))]} \text{ then}
   \texttt{G.add_edge(\alpha(i), \gamma)}

If N too large, preprocessing takes too long
If N too small, queries take too long (or fail).
If \alpha poorly chosen, too many points are required
   to get a good roadmap.

Example.

After 14 pts, \texttt{G} has seven components.
After adding 7 more points \# of components of \texttt{G} is still seven.
Time to stop?

How do you know if N was large enough?
“““““\texttt{G}'s structure mirrors that of C_{free}?"""""""
"G's structure mirrors that of \( C_{\text{free}} \)?

In the example above, since \( C_{\text{free}} \) is given, we know that \( G \) should have 1 component and 2 holes.

In general, we don't know this.

Despite the short comings, PRM have solved many complex problems that no other methods have been able to solve.

Problem w/ simple algorithm given above:

don't connect points unless the connection reduces the # of components of \( G \).

\( G \) would never close the loop.
Path for \( q_1, q_6 \) pair shown will be very long.

If you want to close the loop, replace
\[
\begin{align*}
\text{not } G.\text{same\_comp}(x(i), q) \text{ with } \\
G.\text{vertex\_degree}(q) < k
\end{align*}
\]

Query Phase
Treat \( q_1, q_6 \) as next two elements in sequence \( \alpha \).
If PRM fails to connect \( q_{\alpha} \) on \( q_6 \) then \( N \) was too
If LPM fails to connect \( q_i \) or \( q_6 \), then \( N \) was too small.

If \( q_i \neq q_6 \) are connected, but no path can be found, then no soln exists, or \( N \) was too small. (Perhaps a path exists thru a narrow corridor without many samples.)

Visibility Roadmap

Idea is to cover \( C_{free} \) with a small \# of points

Generalization of visibility region is reachability region. This can be very hard to compute for complex dynamic systems.

Let \( q \) be a "guard" if it cannot see any other guard.

Let \( q \) be a "connector" if it can see at least 2 guards in at least two different components of \( G \).
Visibility Roadmap Algorithm

$q_i, q_g \rightarrow G$. If can't connect $q_i$ & $q_g$, then they are guards.

0. while $i < N$
1. place sample $\alpha(i)$
2. if $\alpha(i)$ is a guard, insert $\alpha(i)$ in $G$.
3. if $\alpha(i)$ is a connector, insert in $G$ & connect disconnected components.
4. otherwise discard $\alpha(i)$.

How do you choose $N$?
Can $G$ be constructed incrementally, stopping when done?
How do you know when to stop? When # of comps. of $G$ stabilizes? When 1000 $\alpha$'s in a row have been discarded? 1,000,000?

If $\alpha$ is dense on $C$ free, then the alg. is resolution complete.