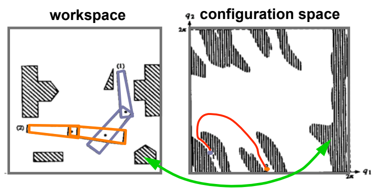
Last lectures

- Configuration space
- Collision detection and distance computation



Probabilistic Roadmaps

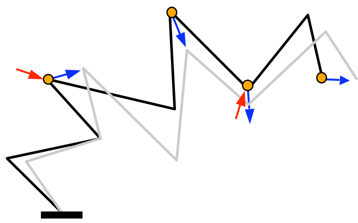
Difficulty with classic approaches

- Running time increases exponentially with the dimension of the configuration space.
  - For a d-dimension grid with 10 grid points on each dimension, how many grid cells are there?

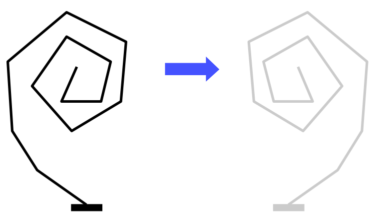
10^d

- Several variants of the path planning problem have been proven to be PSPACE-hard.

First attempt: potential field + random walk



But pathological cases …



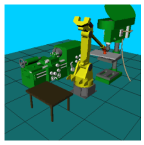
Multiple-Query PRM

Classic multiple-query PRM

- *Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces.* L. Kavraki *et al.*, 1996.

Assumptions

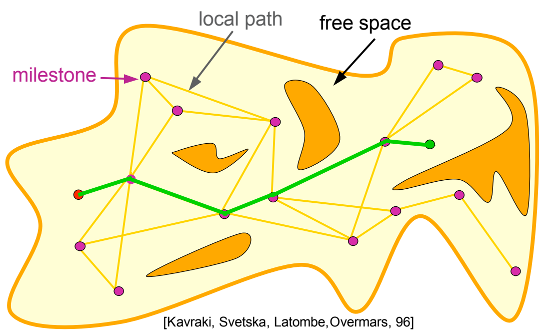
- Static obstacles
- Many queries to be processed in the same environment
- Examples
  - Navigation in static virtual environments
  - Robot manipulator arm in a workcell



Overview

- Precomputation: roadmap construction
  - Uniform sampling
  - Resampling (expansion)
- Query processing

Probabilistic Roadmap (PRM): multiple queries



[Kavraki, Svetska, Latombe,Overmars, 96]

Some terminology

- The graph G is called a **probabilistic roadmap**.
- The nodes in G are called **milestones**.

- Questions?

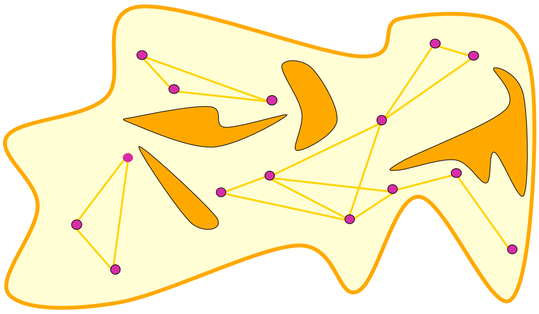
Uniform sampling

```
Input:  geometry of the moving object & obstacles
Output: roadmap G = (V, E)

V ← ∅ and E ← ∅.
repeat
  q ← a configuration sampled uniformly at random from C.
  if CLEAR(q) then
    Add q to V.
    N_q ← a set of nodes in V that are close to q.
    for each q'∈ N_q, in order of increasing d(q,q')
      if LINK(q',q) then
        Add an edge between q and q' to E.
```

Difficulty

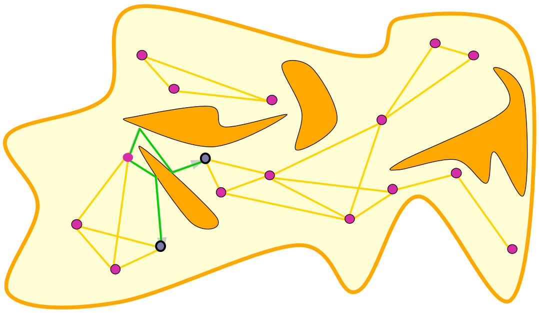
- Many small connected components



Resampling (expansion)

- Failure rate $r(q) = \dfrac{\text{no. failed LINK}}{\text{no. LINK}}$
- Weight $w(q) = \dfrac{r(q)}{\sum_p r(p)}$
- Resampling probability $\Pr(q) = w(q)$

Resampling (expansion)



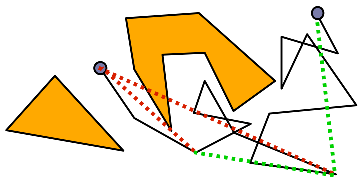
Query processing

- Connect q_{init} and q_{goal} to the roadmap
- Start at q_{init} and q_{goal}, perform a random walk, and try to connect with one of the milestones nearby
- Try multiple times

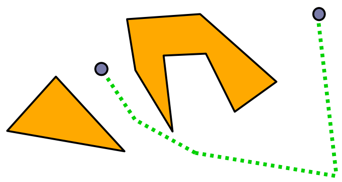
Error

- If a path is returned, the answer is always correct.
- If no path is found, the answer may or may not be correct. We hope it is correct with high probability.
- If either q_{init} or q_{goal} cannot be connected to a node of the roadmap, then the roadmap does not have sufficient information to answer the query. It is a failure.

Smoothing the path



Smoothing the path



Summary

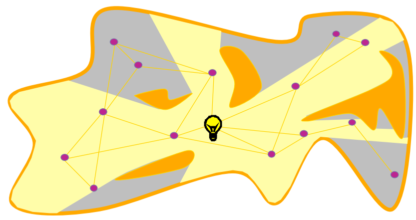
- What probability distribution should be used for sampling milestones?
- How should milestones be connected?
- A path generated by a randomized algorithm is usually jerky. How can a path be smoothed?

Expansive Spaces

Analysis of Probabilistic Roadmaps

Issues of probabilistic roadmaps

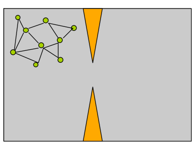
- Coverage
- Connectivity



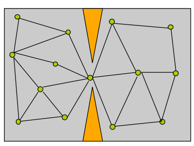
Is the coverage adequate?

- Milestones should be distributed so that almost **any** point of the configuration space can be connected by a straight line segment to one milestone.

Bad



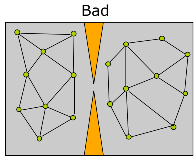
Good



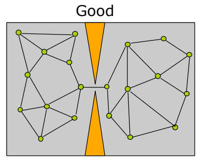
Connectivity

- There should be a one-to-one correspondence between the connected components of the roadmap and those of F.

Bad

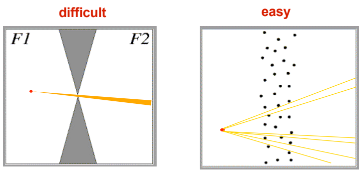


Good



Narrow passages

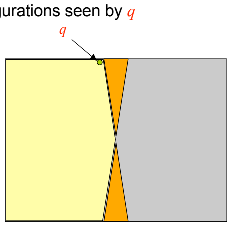
- Connectivity is difficult to capture when there are narrow passages.
- Narrow passages are difficult to define.



- How to characterize coverage & connectivity?
  Expansiveness

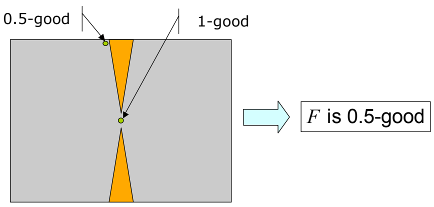
Definition: visibility set

- Visibility set of q
  - All configurations in F that can be connected to q by a straight-line path in F
  - All configurations seen by q



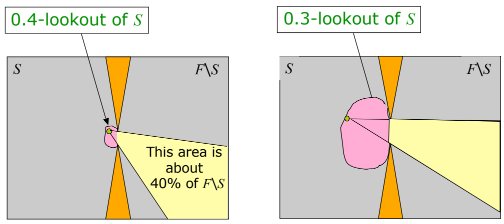
Definition: ϵ-good

- Every free configuration sees at least ε fraction of the free space, ε in (0,1].



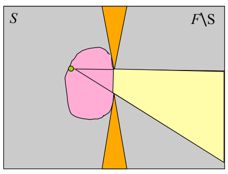
Definition: lookout of a subset S

- Subset of points in S that can see at least β fraction of $F\backslash S$, β is in (0,1].



Definition: (ϵ, α, β)-expansive

- The free space F is $(\varepsilon,\alpha,\beta)$-expansive if
  - Free space F is ε-good
  - For each subset S of F, its β-lookout is at least α fraction of S. ε,α,β are in (0,1]



F is ε-good → ε=0.5

β-lookout → β=0.4

$\frac{\text{Volume}(\beta\text{-lookout})}{\text{Volume}(S)}$ → α=0.2

F is $(\varepsilon, \alpha, \beta)$-expansive, where ε=0.5, α=0.2, β=0.4.

Why expansiveness?

- ε, α, and β measure the expansiveness of a free space.
- Bigger ε, α, and β → lower cost of constructing a roadmap with good connectivity and coverage.

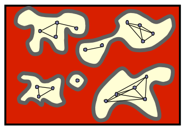
Theorem 1(Connectivity)

- Probability of achieving good connectivity increases exponentially with the number of milestones (in an expansive space).
- If ε, α, β decreases, then need to increase the number of milestones (to maintain good connectivity)

Path planning in expansive configuration spaces. Hsu, *et al.*, 1999.

Uniform sampling

- All-pairs path planning



- **Theorem 1** : A roadmap of $\frac{16\ln(1/\gamma)}{\varepsilon\alpha}+\frac{6}{\beta}$ uniformly-sampled milestones has the correct connectivity with probability at least $1-\gamma$.

Theorem 2 (Coverage)

- Probability of achieving good coverage, increases exponentially with the number of milestones (in an expansive space).

Completeness

- Complete algorithms are slow.
  - A **complete** algorithm finds a path if one exists and reports no otherwise.
  - Example: Canny's roadmap method
- Heuristic algorithms are unreliable.
  - Example: potential field
- **Probabilistic completeness**
  - Intuition: If there is a solution path, the algorithm will find it with high probability.

Probabilistic completeness

In an expansive space, the probability that a PRM planner fails to find a path when one exists goes to 0 exponentially in the number of milestones (~ running time).

[Kavraki, Latombe, Motwani, Raghavan,95]
[Hsu, Latombe, Motwani, 97]

Summary

- Main result
  - If a C-space is expansive, then a roadmap can be constructed efficiently with good connectivity and coverage.
- Limitation in practice
  - It does not tell you when to stop growing the roadmap.
  - A planner stops when either a path is found or max steps are reached.