Motion/Path Planning II

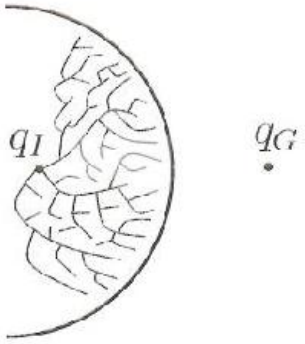
Sampling techniques

Approaches

- Cell decomposition
- Roadmaps
- Sampling Techniques (RRT, DRT, PRM,..)
- On-line algorithms D*, ARA*,..

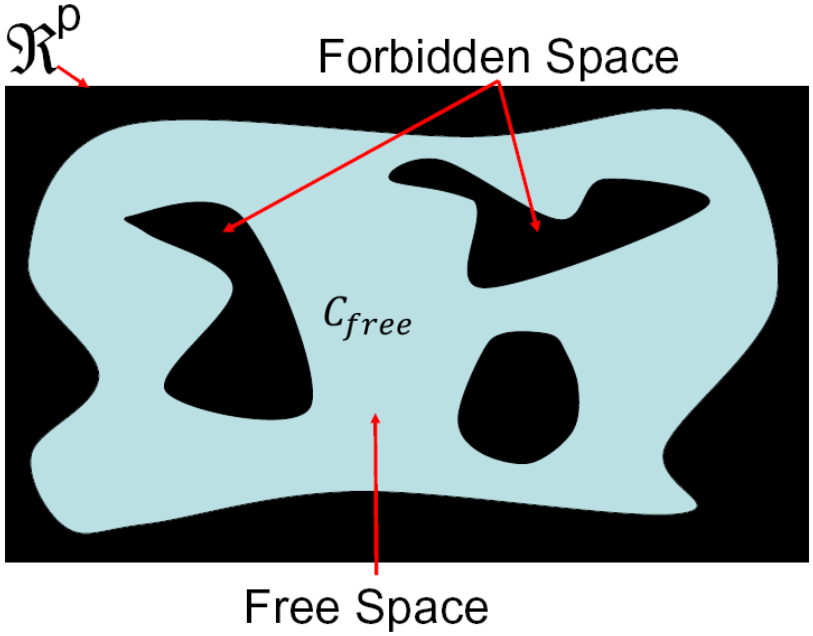
Completely describing and "optimally" exploring the C-space is too hard in high dimension + it is not necessary →
Limit ourselves to finding a "good" sampling of the C-space

- This is why sampling of the entire space (rather than searching from start state) is necessary:



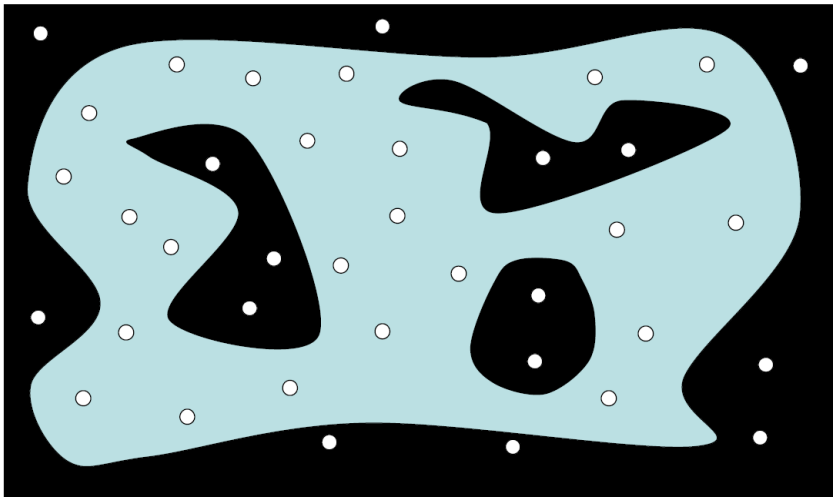
Example from Steve Lavalle

Sampling Techniques



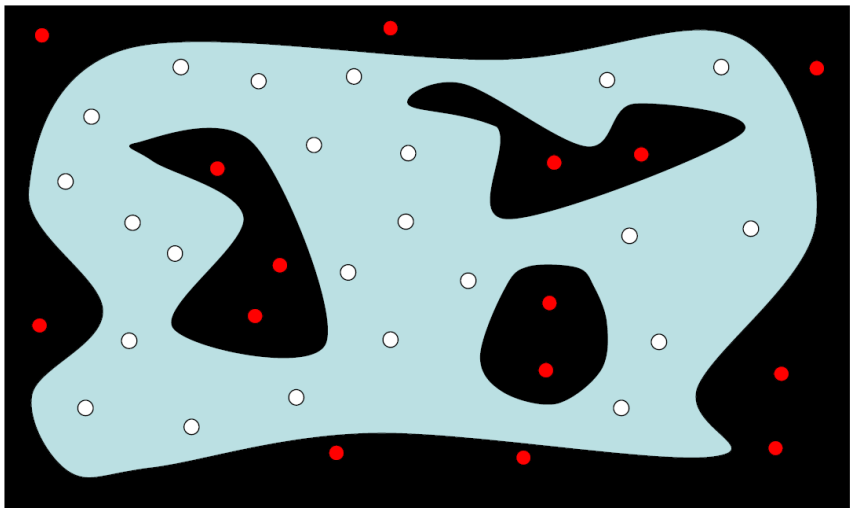
Sampling Techniques

Sample random locations



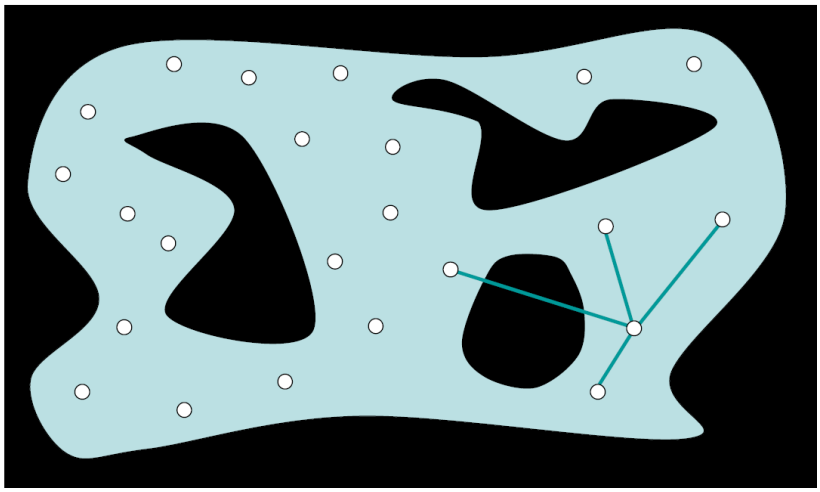
Sampling Techniques

Remove the samples in the forbidden regions



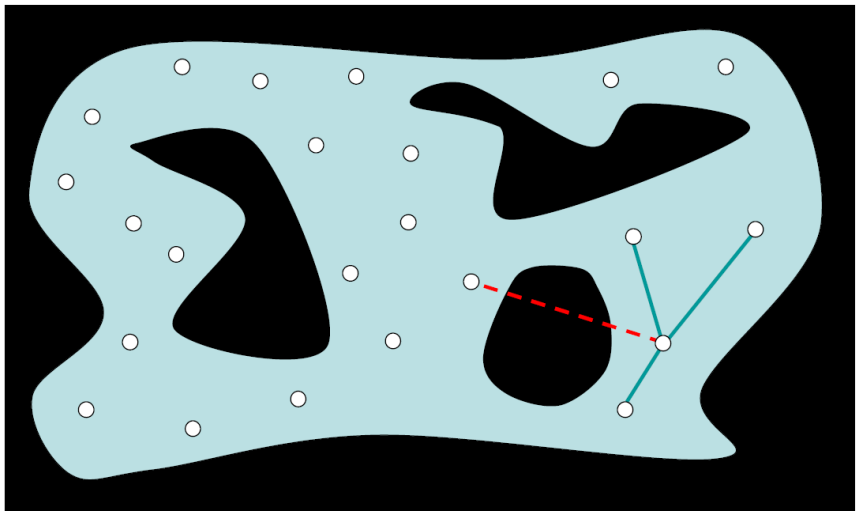
Sampling Techniques

Link each sample to its *K* nearest neighbors



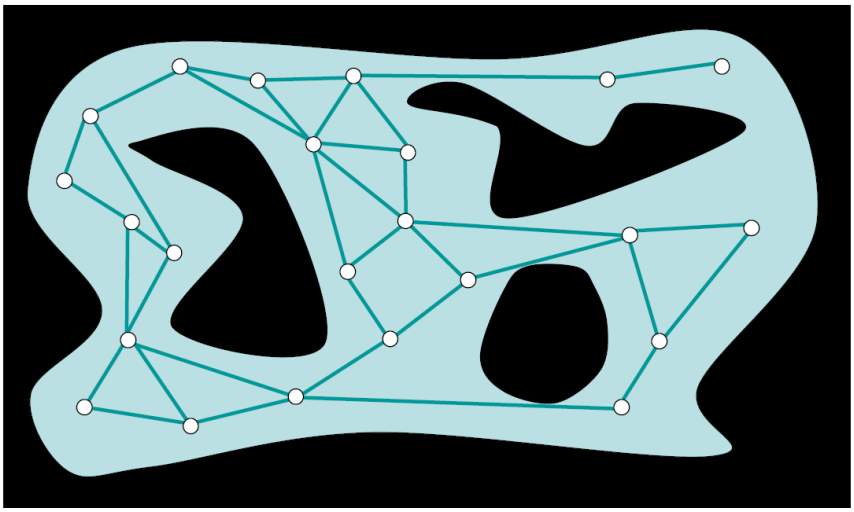
Sampling Techniques

Remove the links that cross forbidden regions



Sampling Techniques

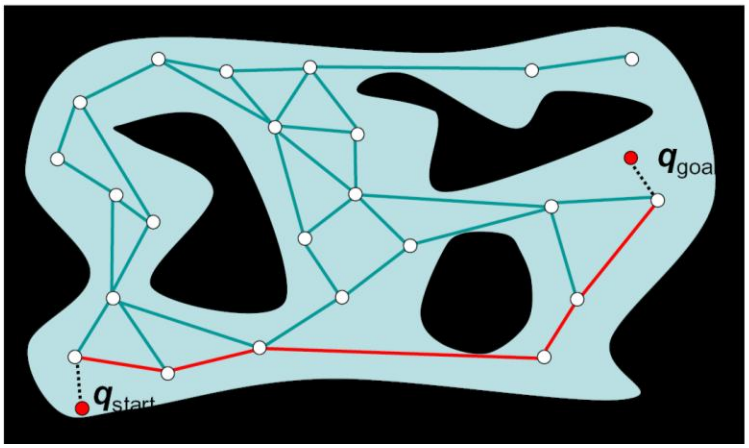
Remove the links that cross forbidden regions



The resulting graph is a *probabilistic roadmap (PRM)*

Sampling Techniques

Link the start and goal to the PRM and search using A*



Sampling Techniques

Continuous space

↓

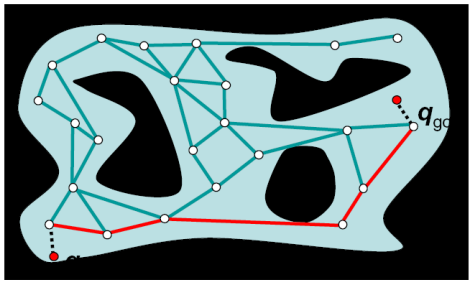
Sampling

↓

Local planner

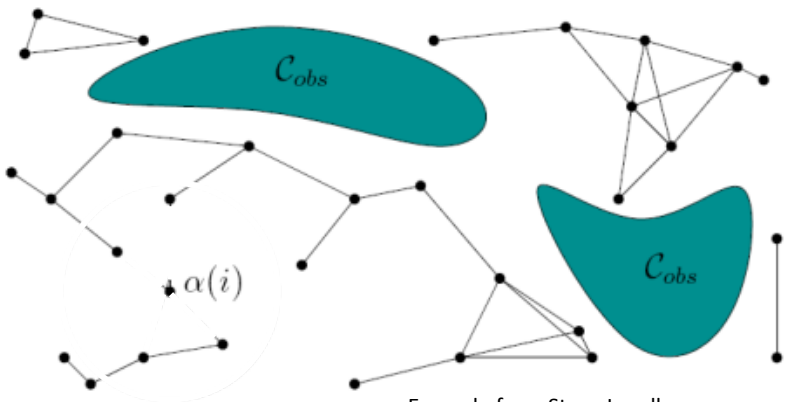
↓

A* through roadmap



- How to connect the samples?
- How to select the samples: "Good" sampling strategies are important
- What are good strategy to maximize "completeness" and to minimize time?

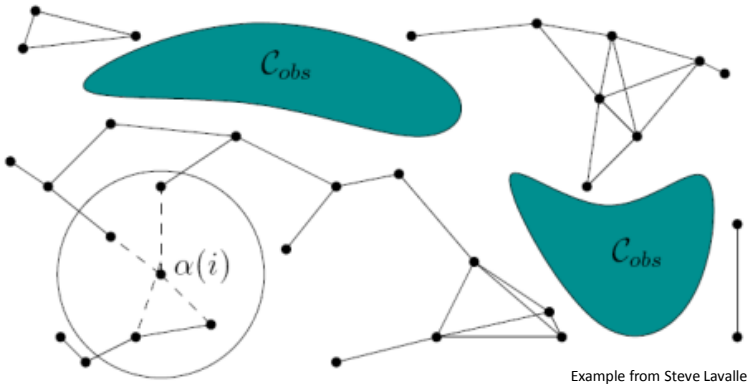
- Suppose that we have built a partial roadmap from i-1 samples.
- Let $\alpha(i)$ be the i-th sample
- **What strategy would you use to $\alpha(i)$ *connect* to the existing roadmap?**
- **What strategy would you use to select $\alpha(i)$?**



Example from Steve Lavalle

Connecting samples

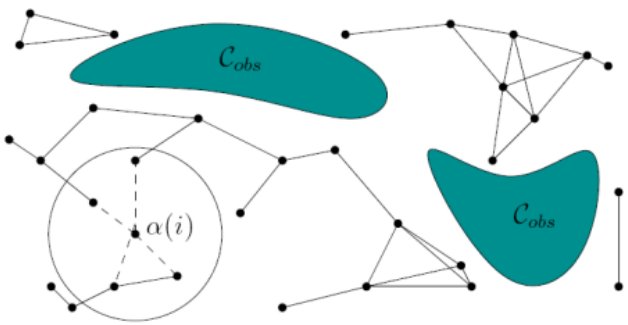
- Select *K* vertices closest to *α(i)*
- Select *K* (often just 1) closest points from each of the components in *G*
- Select all vertices within radius *r* from *α(i)*



Example from Steve Lavalle

Selecting samples I

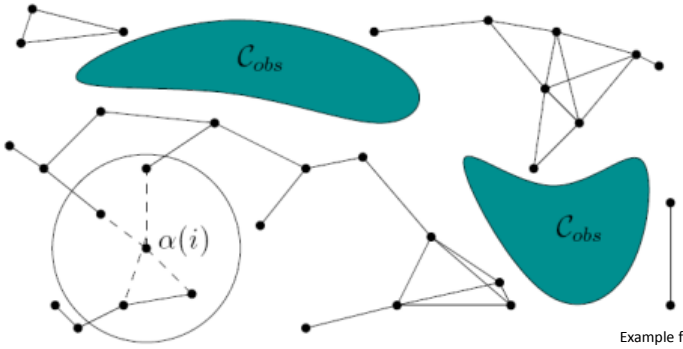
- Sample uniformly from C_{free}
- Select at random an existing vertex with a probability distribution inversely proportional to how well-connected a vertex is, and then generate a random motion from it to get a sample *α(i)*
- Bias sampling toward obstacle boundaries



Example from Steve Lavalle

Selecting samples II

- Sample q_1 and q_2 from Gaussian around q_1 and if either is in C_{obs}, then the other one is set as $\alpha(i)$
- Sample q_1, q_2, $q_3$and set q_2 as $\alpha(i)$ if q_2 is in C_{free}, and q_1 and q_3 are in C_{obs}
- Bias sampling away from obstacles



Example from Steve Lavalle

– Can we say something a little more formal about desired properties of $\alpha(i)$?

– Can we say something a little more formal about "approximate completeness"?

– Can we avoid the (expensive) pre-processing step?

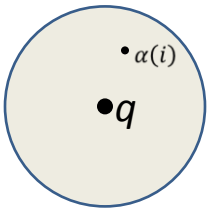
Density

- A set *Y* is dense in *X* iff for any *x* in *X* and any $\varepsilon > 0$, there exist a *y* in *Y* inside the ball $B(x,\varepsilon)$

(e.g., $\mathbb{Q}$ dense in $\mathbb{R}$)

- A sequence $\alpha(i)$ is dense in *C* if the corresponding set *A* is dense in *C*
- A random sequence is *probably* dense:
  - For any *q* in *C* and any $\varepsilon > 0$, there exist a *i* such that $\alpha(i)$ is inside the ball $B(q,\varepsilon)$ with probability 1

Note: The distance could be or L_2 or L_∞ … the denseness property remains



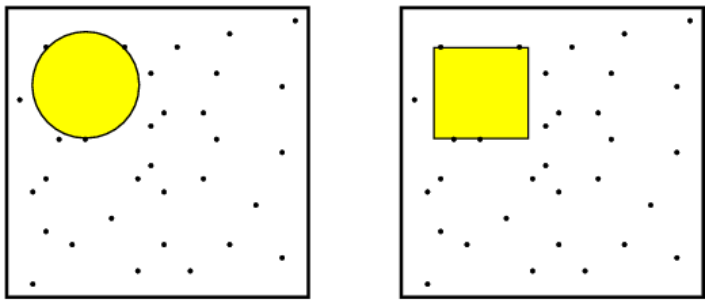
Deterministic sequences

- All we need is asymptotic denseness
- We don't absolutely need randomness

i	Naive Sequence	Binary	Reverse Binary	Van der Corput	Points in $[0,1]/\sim$
1	0	.0000	.0000	0	
2	1/16	.0001	.1000	1/2	
3	1/8	.0010	.0100	1/4	
4	3/16	.0011	.1100	3/4	
5	1/4	.0100	.0010	1/8	
6	5/16	.0101	.1010	5/8	
7	3/8	.0110	.0110	3/8	
8	7/16	.0111	.1110	7/8	
9	1/2	.1000	.0001	1/16	
10	9/16	.1001	.1001	9/16	
11	5/8	.1010	.0101	5/16	
12	11/16	.1011	.1101	13/16	
13	3/4	.1100	.0011	3/16	
14	13/16	.1101	.1011	11/16	
15	7/8	.1110	.0111	7/16	
16	15/16	.1111	.1111	15/16	

Dispersion

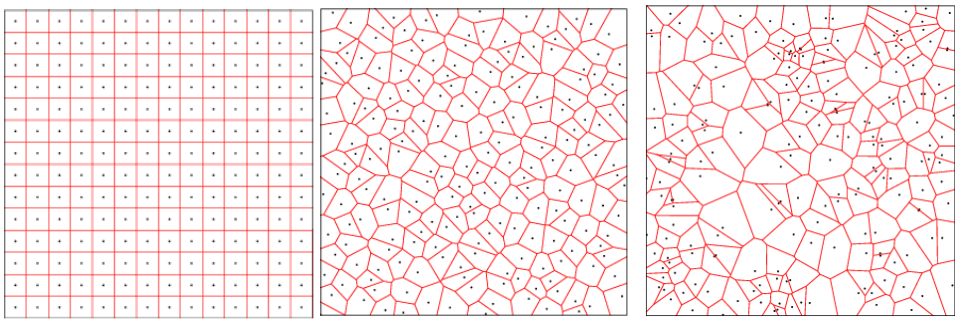
- We may want the samples to become uniformly close to each other
- Dispersion: $\delta(A) = \max_{q \in C}(\min_{\alpha \in A} d(q, \alpha))$



L_2 L_∞

Dispersion

For $L_\infty : \delta(A) \geq \frac{1}{2N^{\frac{1}{p}}}$ (dispersion for grids)



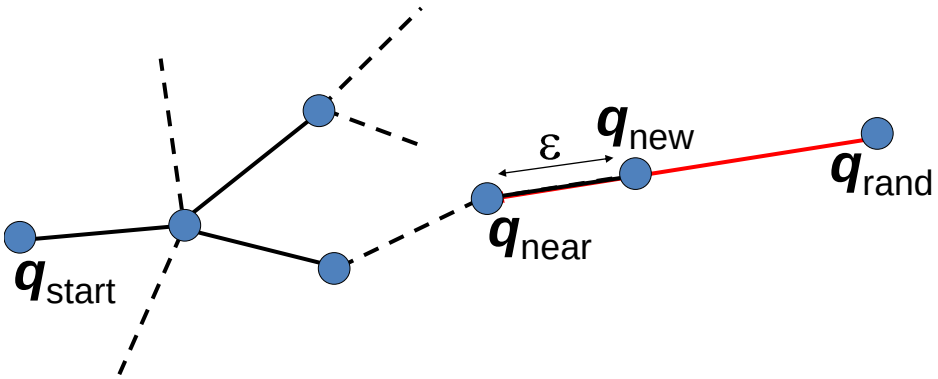
Grid Halton Random

- Can we say something a little more formal about desired properties of $\alpha(i)$?
- *Can we say something a little more formal about "approximate completeness"?*
- Can we avoid the (expensive) pre-processing step?

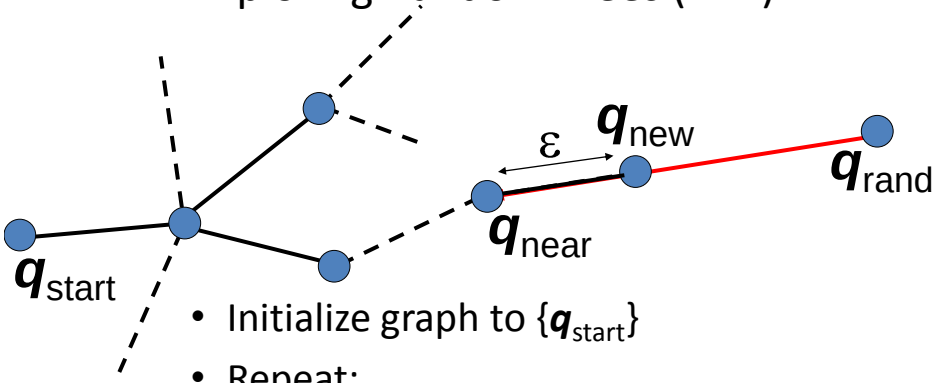
"Completeness"

- Deterministic sampling:
  - If α is dense sampling
  - Resolution complete <==> Guaranteed to find a path in finite time if one exists
- Random sampling:
  - If α is probabilistically dense
  - Probabilistically complete <==> Guaranteed to find a path in finite time if one exists with probability 1

– Can we say something a little more formal about desired properties of $\alpha(i)$?
– Can we say something a little more formal about "approximate completeness"?
– *Can we avoid the (expensive) pre-processing step?*

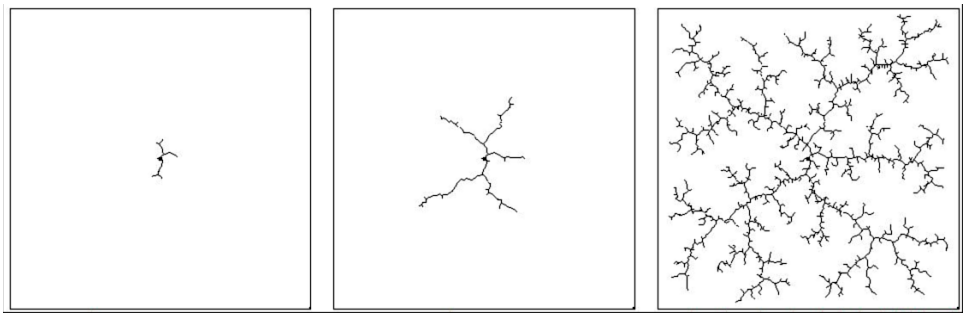


No pre-processing: Rapidly Exploring Random Trees (RRT)

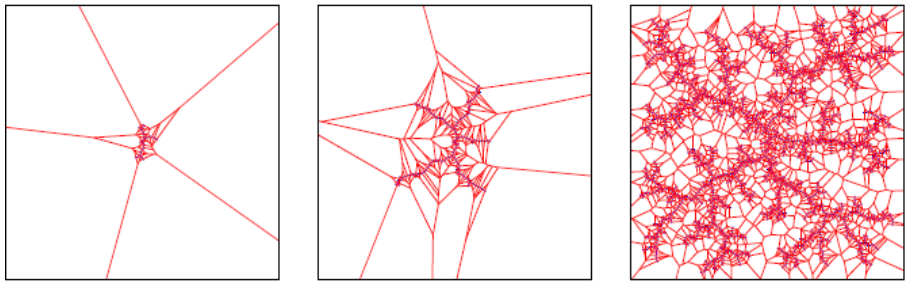


- Initialize graph to $\{q_{start}\}$
- Repeat:
  - Select random new sample q_{rand}
  - Find closest node q_{near} to q_{rand}
  - Create edge (q_{near}, q_{new}) if no collisions

Properties

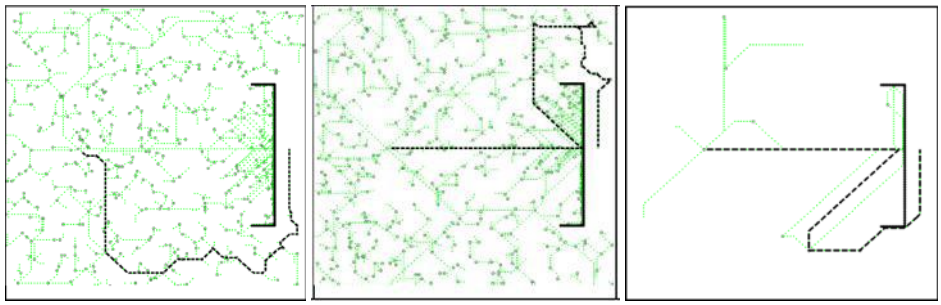


- Tends to explore the space rapidly in all directions
- Does not require extensive pre-processing
- Single query/multiple query problems
- Needs only collision detection test → No need to represent/pre-compute the entire C-space



uniform coverage of space:
the growth is always biased by the largest Voronoi region

How to connect the goal?



$P_g = 0$ $P_g = 0.1$ $P_g = 0.5$

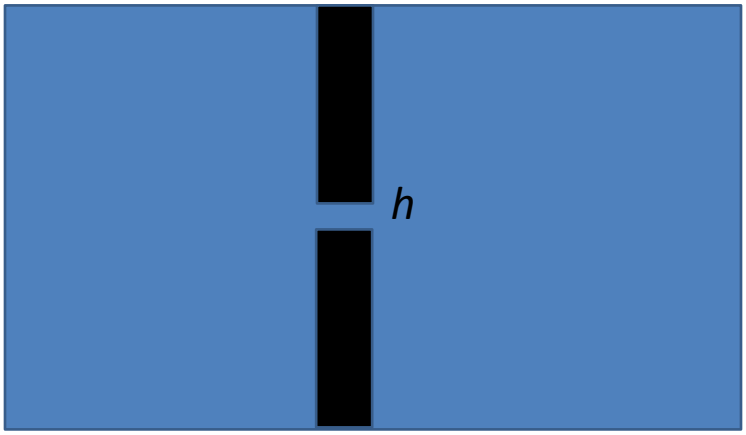
- With a probability $(1-P_g)$, q_{rand} is chosen as a random sample in C_{free}, with probability P_g, q_{rand} is set to q_G
- Variations (ERRT): If known set of "preferred waypoint" locations W,

$(1-P_g-P_w) \rightarrow q_{rand}$

$P_g \rightarrow q_G$

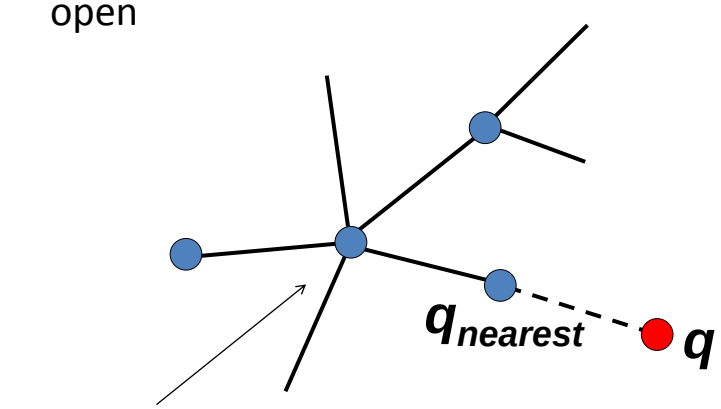
$P_w \rightarrow$ random q from W

- In general: Probability of finding path after *k* samples depends on the smallest gap in C_{obs}
- *P*(finding a path) after *k* samples assuming
  - Uniform sampling
  - Use nearest sample every iteration



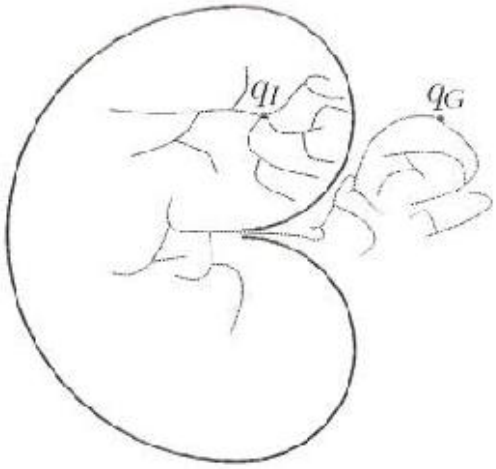
Convergence?

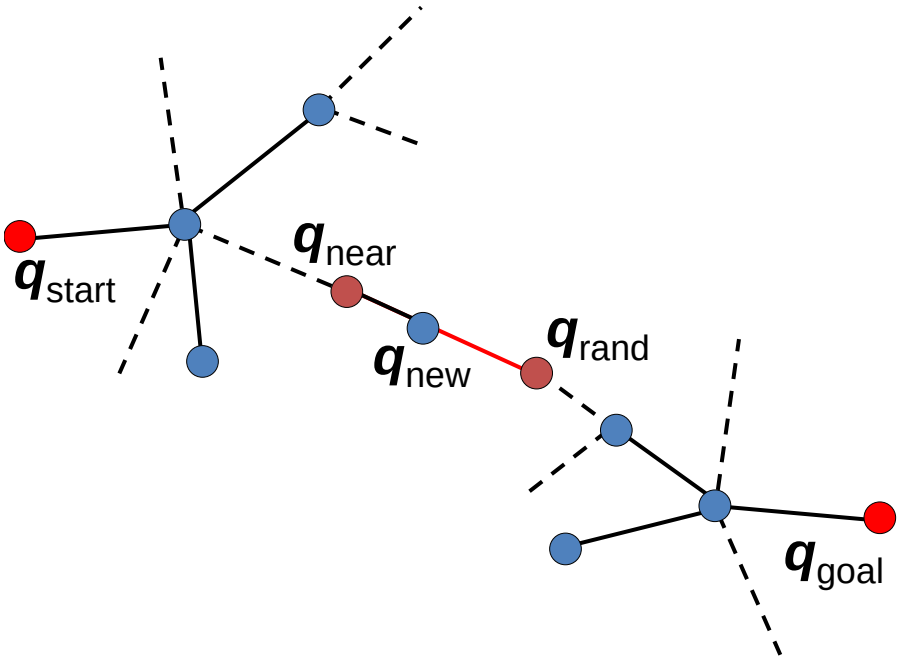
- For any $q \in C_{free}$, $\lim_{N\to\infty} P[d(q,q_{nearest}) < \varepsilon] = 1$
- Assumptions: C_{free} is connected, bounded and open

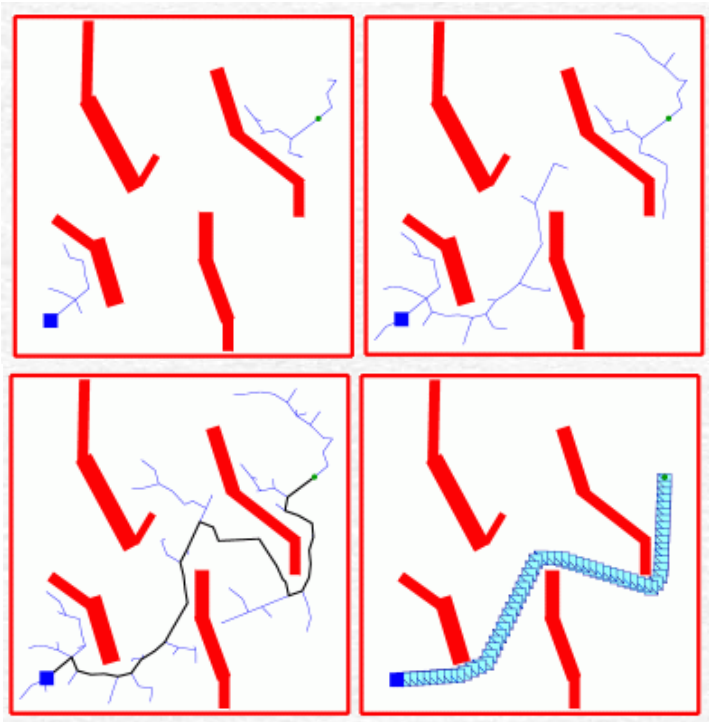


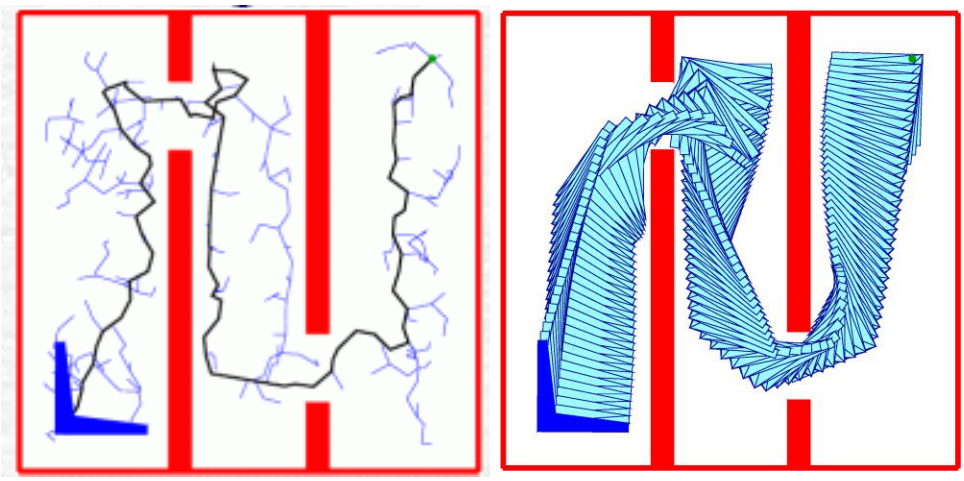
N samples drawn

- Should we use 2 trees?

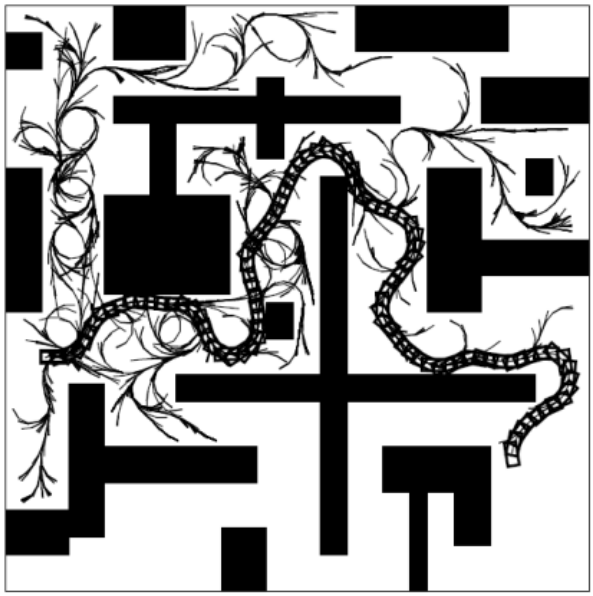




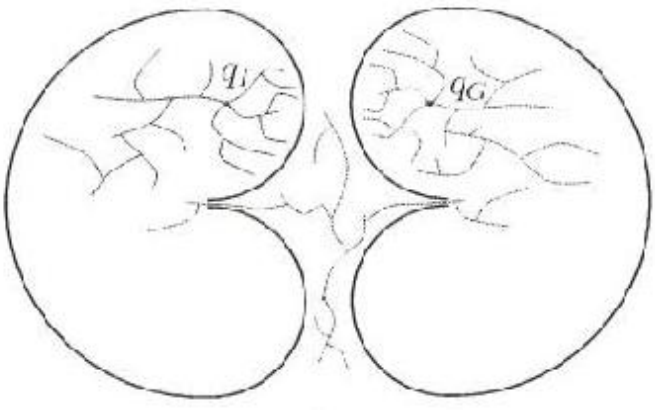




From Kuffner et al.

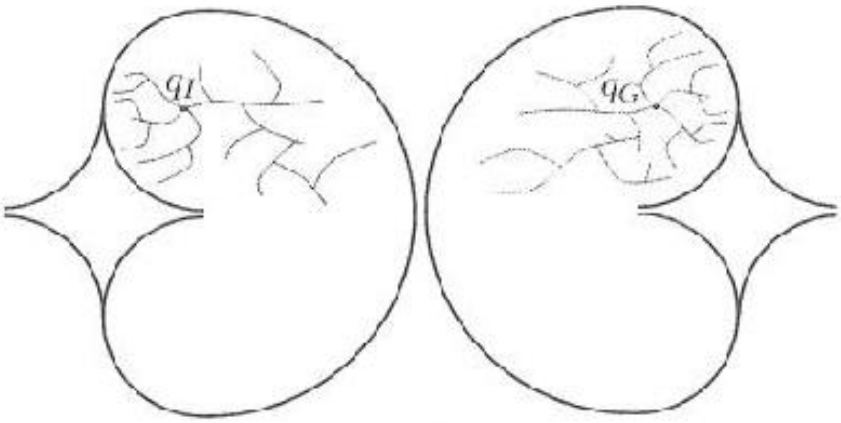


- Should we use more trees?



Example from Steve Lavalle

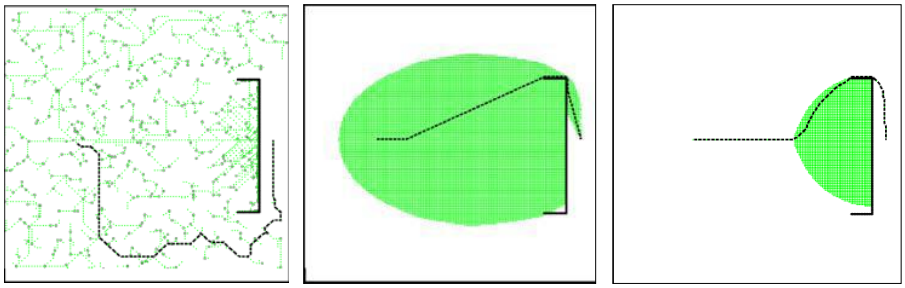
- Good luck!



Example from Steve Lavalle

Comparison

- Even for easier problems that can be solved by A* directly, needs much fewer node expansions
- Smaller memory
- Works well in (very) high dimensions
- (Very) sub-optimal solutions. Needs post-processing.
- More difficult to incorporate complex cost functions



RRT A* wA* ε=3

Note: wA* = A* with heuristic $\varepsilon h(.)$ where $h(.)$ is an admissible heuristic. Sub-optimal, guaranteed to find path at most εC^*

Example from Max Likhachev