Motion Planning

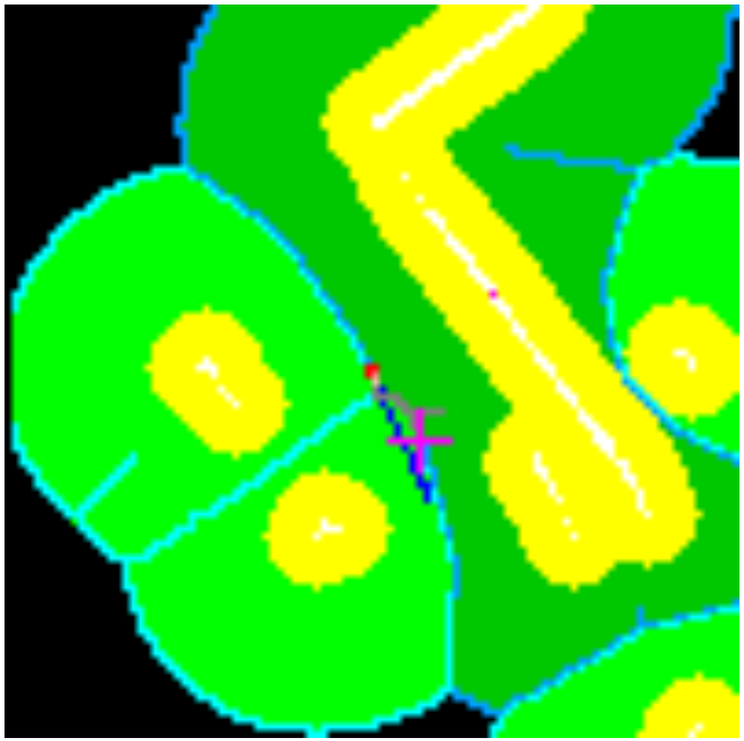
COMP3431 Robot Software Architectures

Motion Planning

- Task Planner can tell the robot discrete steps but can’t say how to execute them
- Discrete actions must be turned into operations in a continuous world
- Planning actions set goals and constraints, something else must implement motor actions

Configuration Space

- Treat robot as a point, expand obstacles
- More complicated if robot is not a regular shape (piano movers problem)



Robotic Motion Planning: RRT's

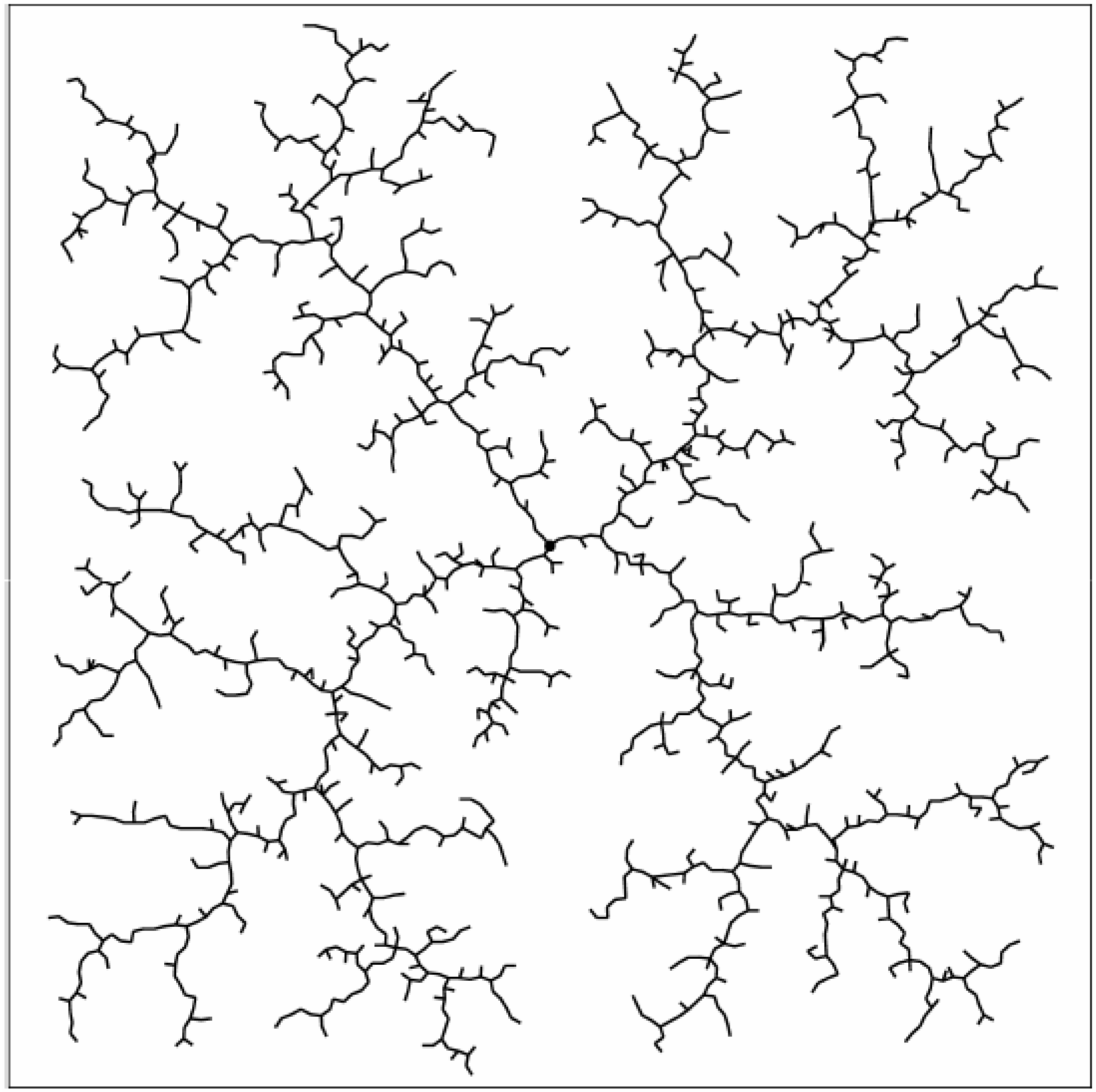
Robotics Institute 16-735

http://www.cs.cmu.edu/~motion

Howie Choset

http://www.cs.cmu.edu/~choset

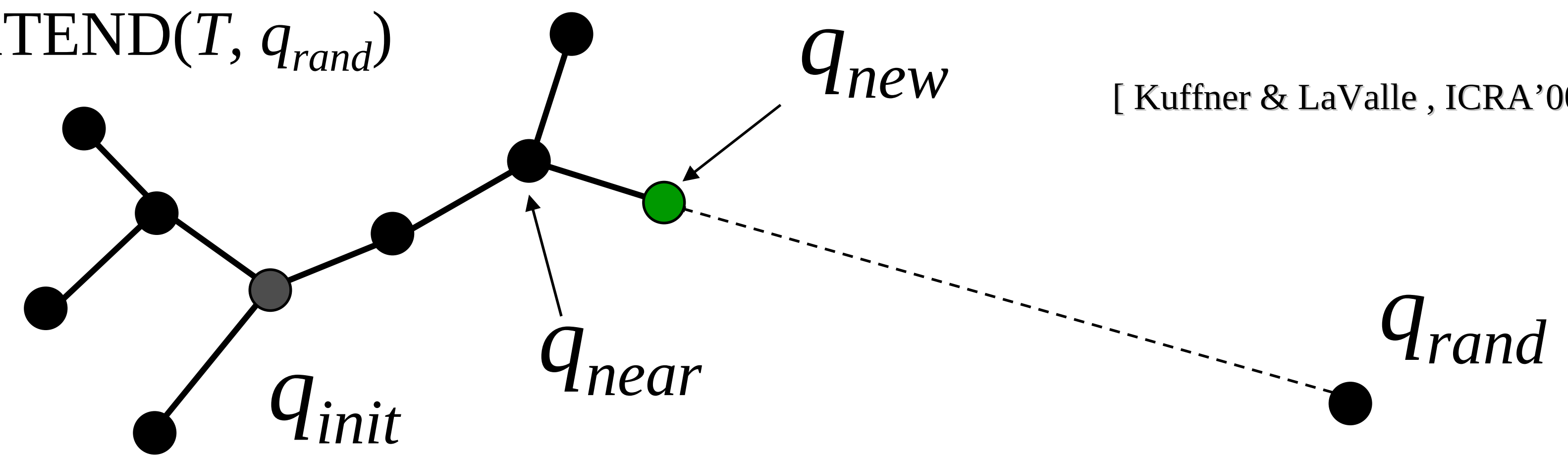
Rapidly-Exploring Random Tree



Path Planning with RRTs (Rapidly-Exploring Random Trees)

```
BUILD_RRT (q_init)  {
   T.init(q_init);
   for k =  1 to K do
      q_rand = RANDOM_CONFIG();
      EXTEND(T, q_rand)
}
```

EXTEND(T, q_{rand})

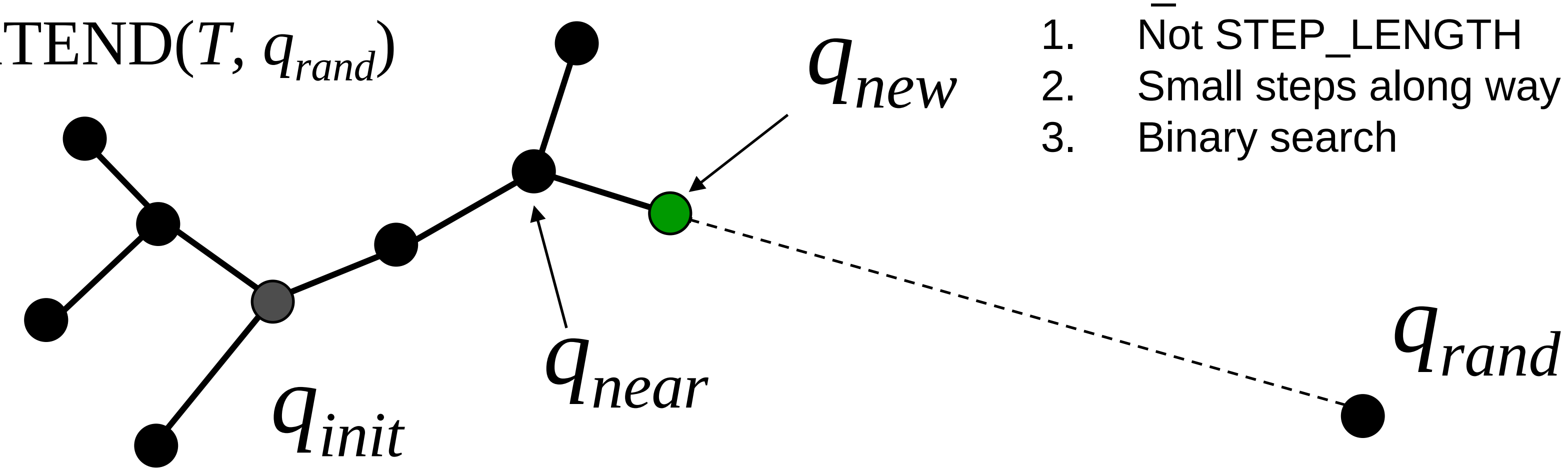


[Kuffner & LaValle , ICRA'00]

Path Planning with RRTs
(Some Details)

```
BUILD_RRT (q_init) {
   T.init(q_init);
   for k = 1 to K do
      q_rand = RANDOM_CONFIG();
      EXTEND(T, q_rand)
}
```

EXTEND(T, q_{rand})



STEP_LENGTH: How far to sample

1. Sample just at end point
2. Sample all along
3. Small Step

Extend returns

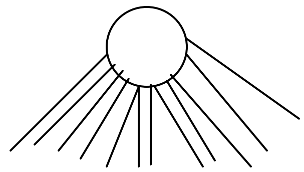
1. Trapped, cant make it
2. Extended, steps toward node
3. Reached, connects to node

STEP_SIZE

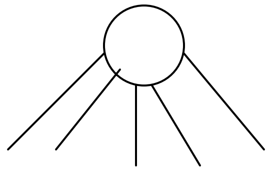
1. Not STEP_LENGTH
2. Small steps along way
3. Binary search

RRT vs. Exhaustive Search

- Discrete

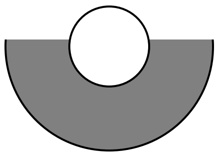


A* may try all edges

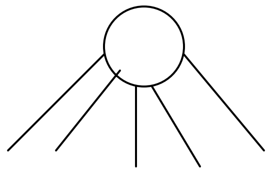


Probabilistically subsample all edges

- Continuous



Continuum of choices



Probabilistically subsample all edges

Naïve Random Tree

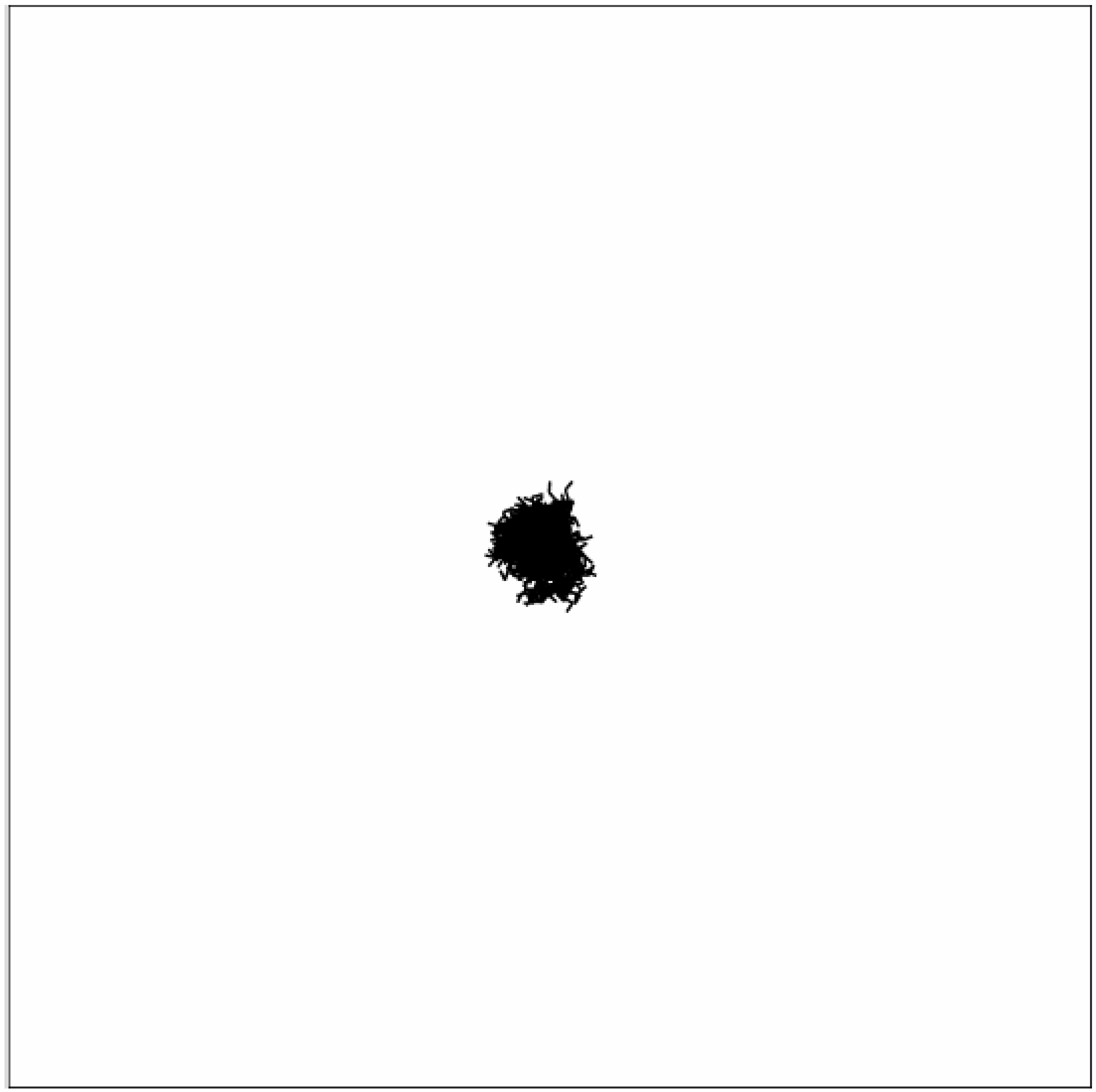
Start with middle

Sample near this node

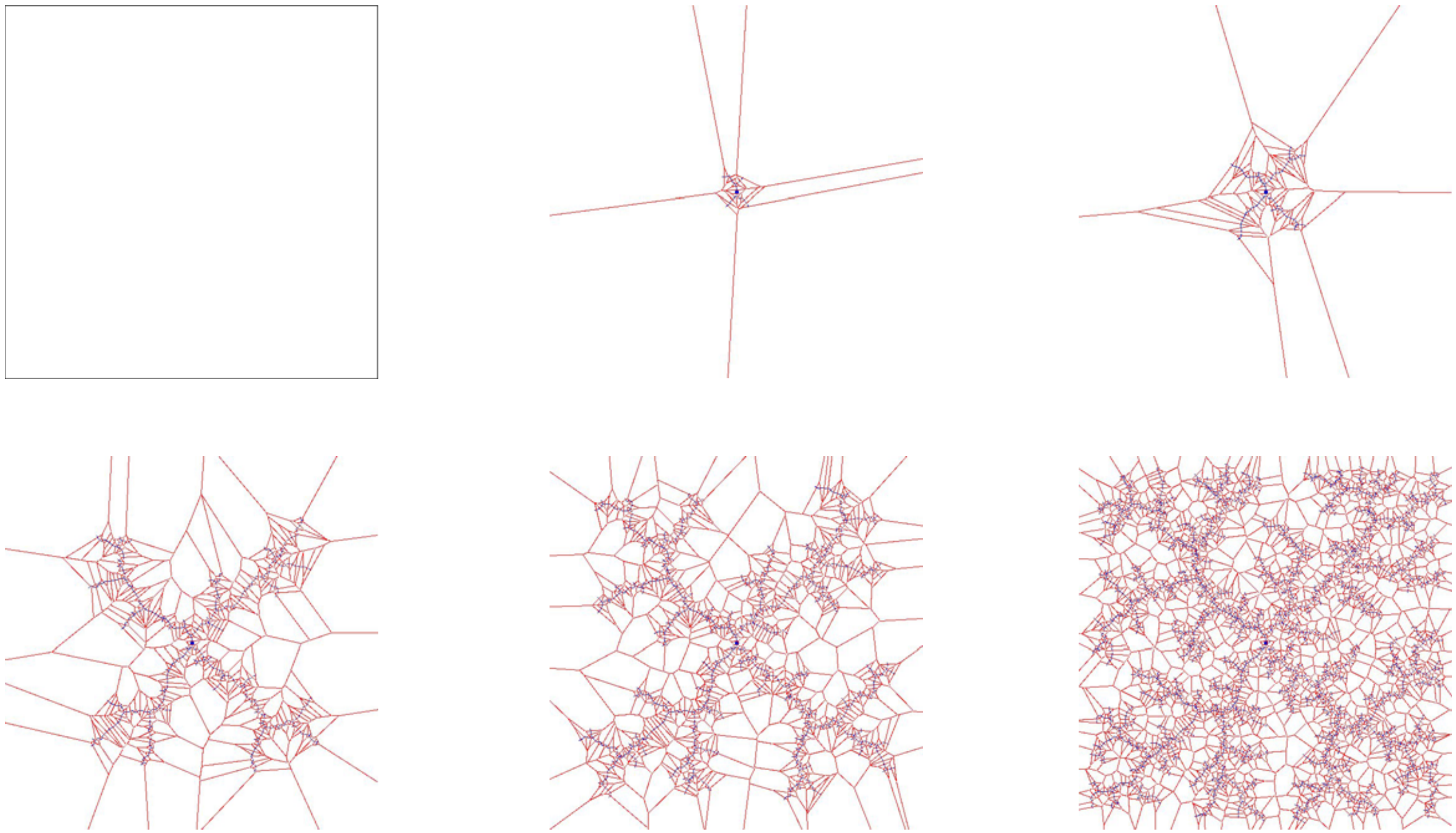
Then pick a node at random in tree

Sample near it

End up Staying in middle



RRTs and
Bias toward large Voronoi regions

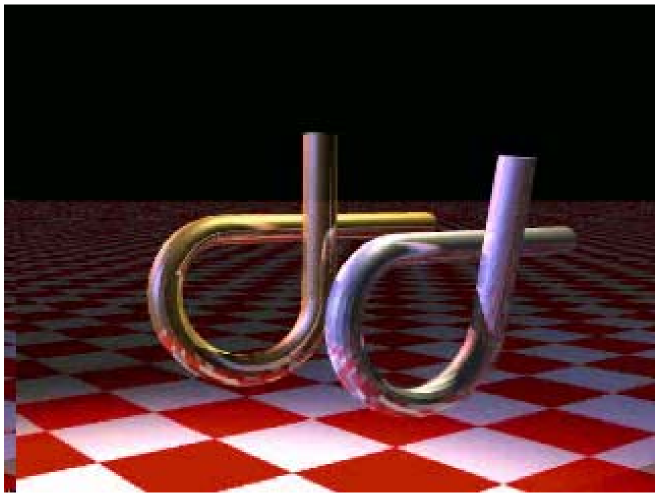


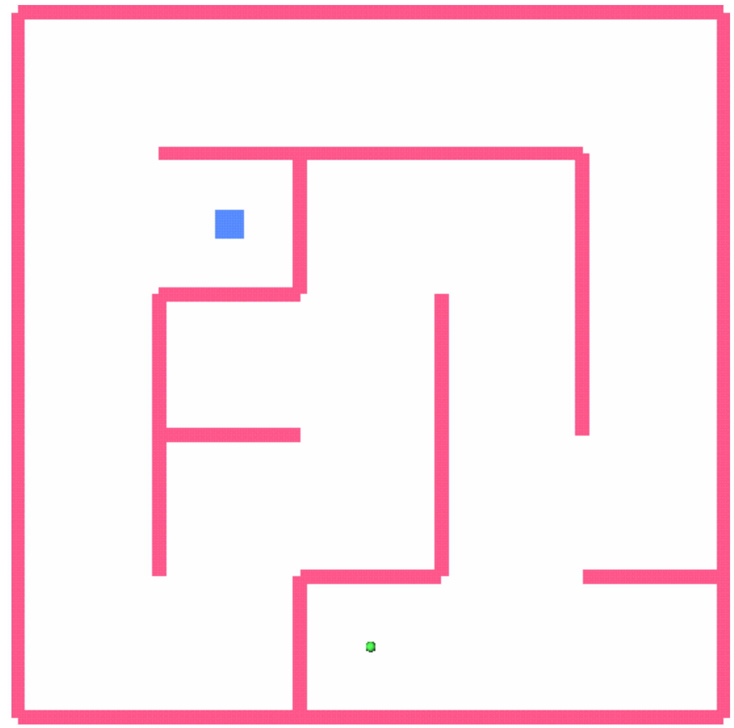
http://msl.cs.uiuc.edu/rrt/gallery.html

Biases

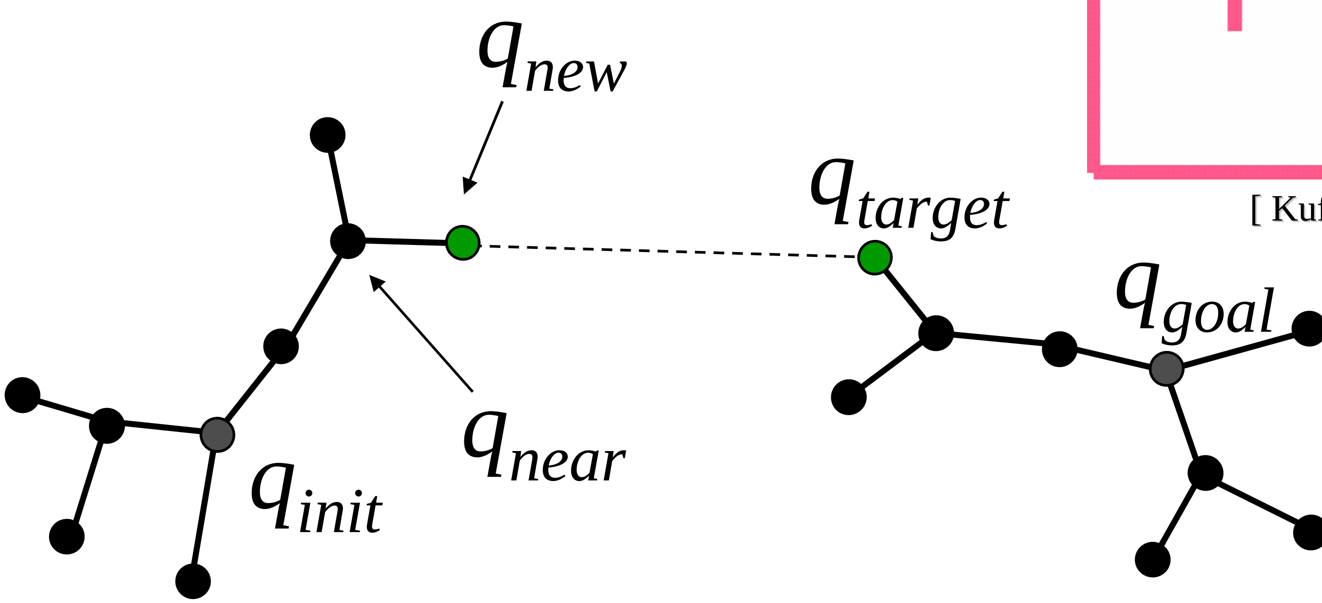
- Bias toward larger spaces
- Bias toward goal
  - When generating a random sample, with some probability pick the goal instead of a random node when expanding
  - This introduces another parameter
  - James' experience is that 5-10% is the right choice
  - If you do this 100%, then this is a RPP

Grow two RRTs towards each other

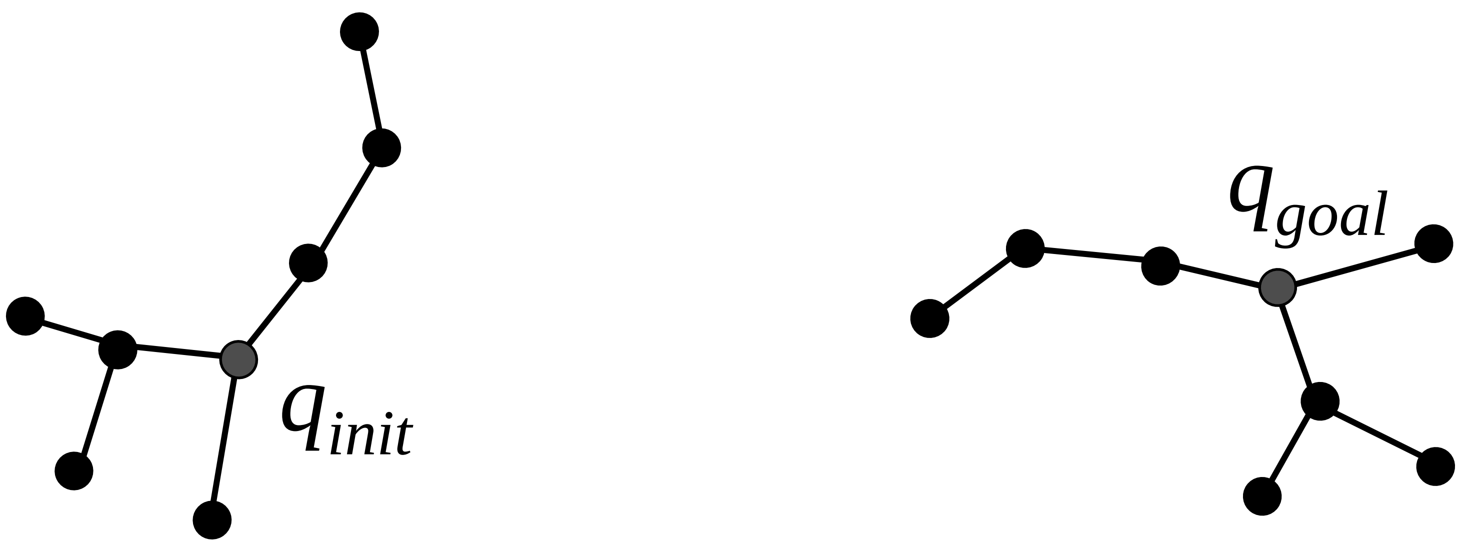




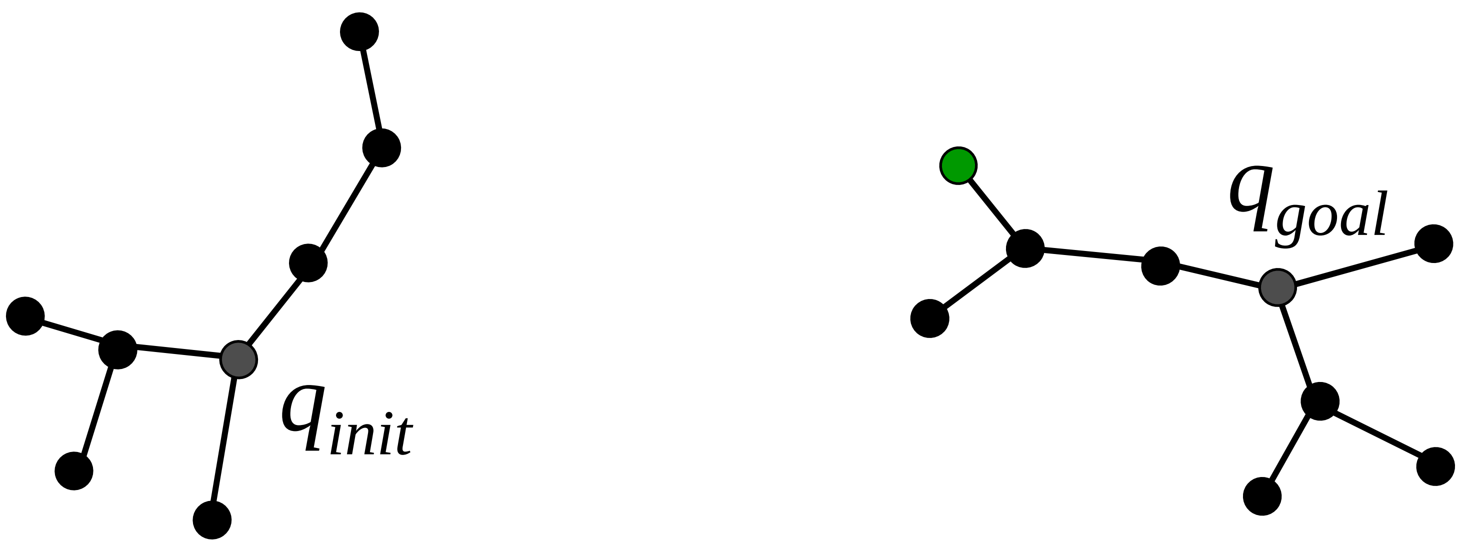
[Kuffner, LaValle ICRA '00]



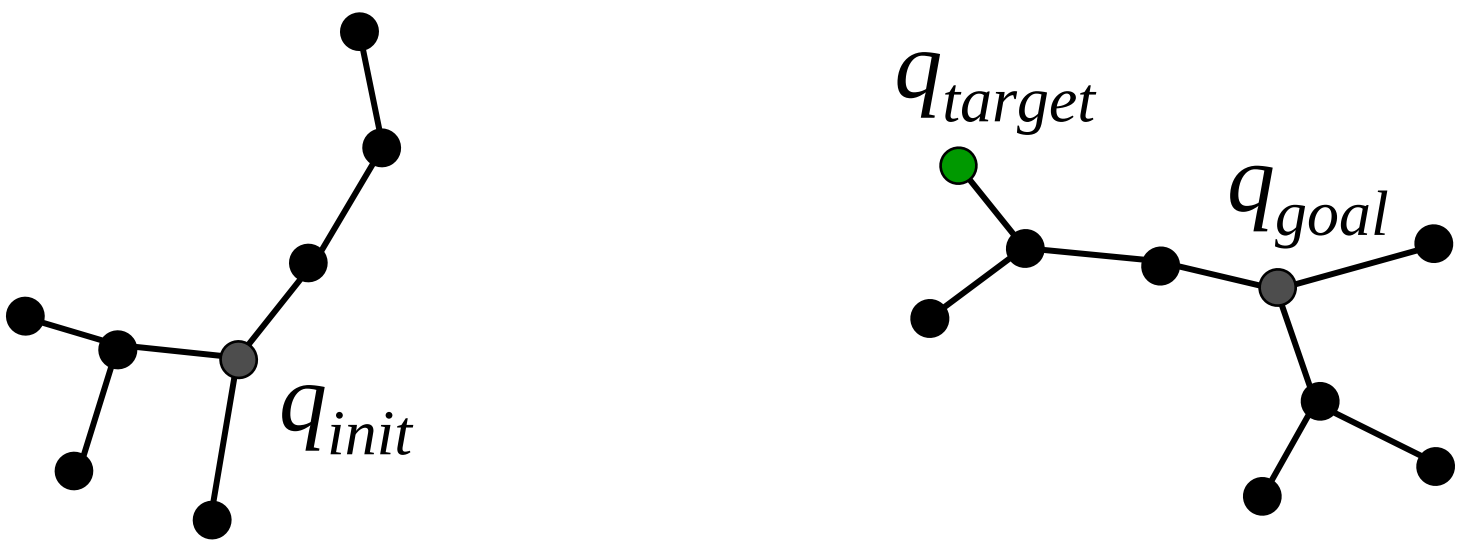
A single RRT-Connect iteration...



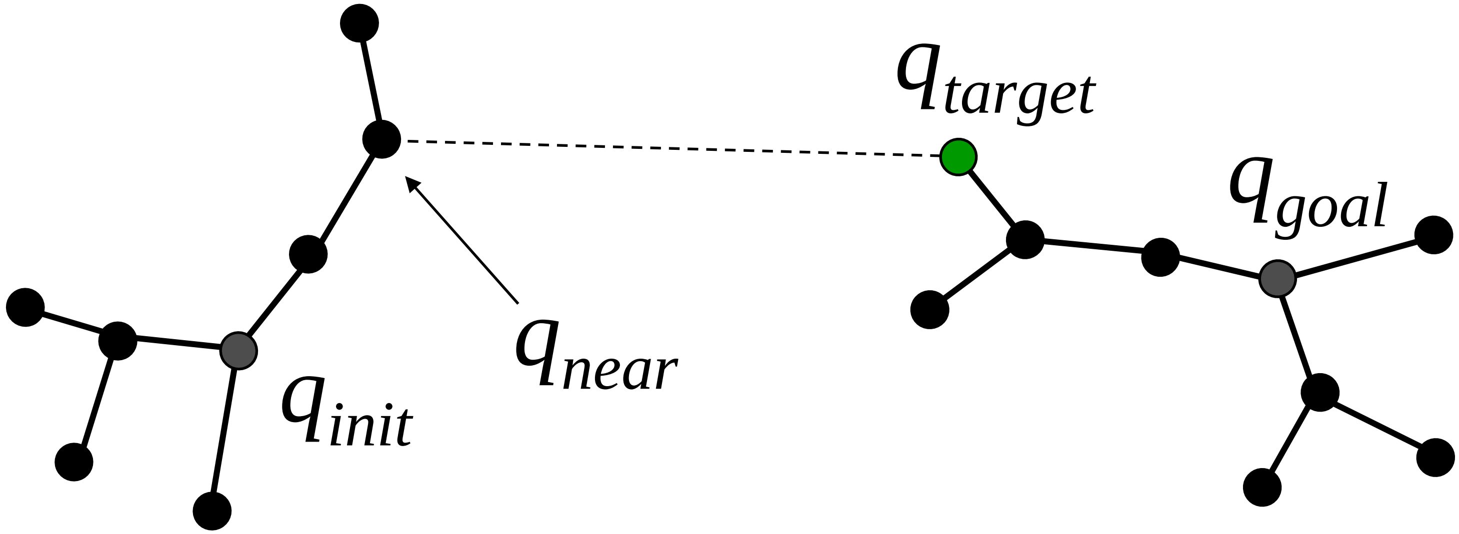
1) One tree grown using random target



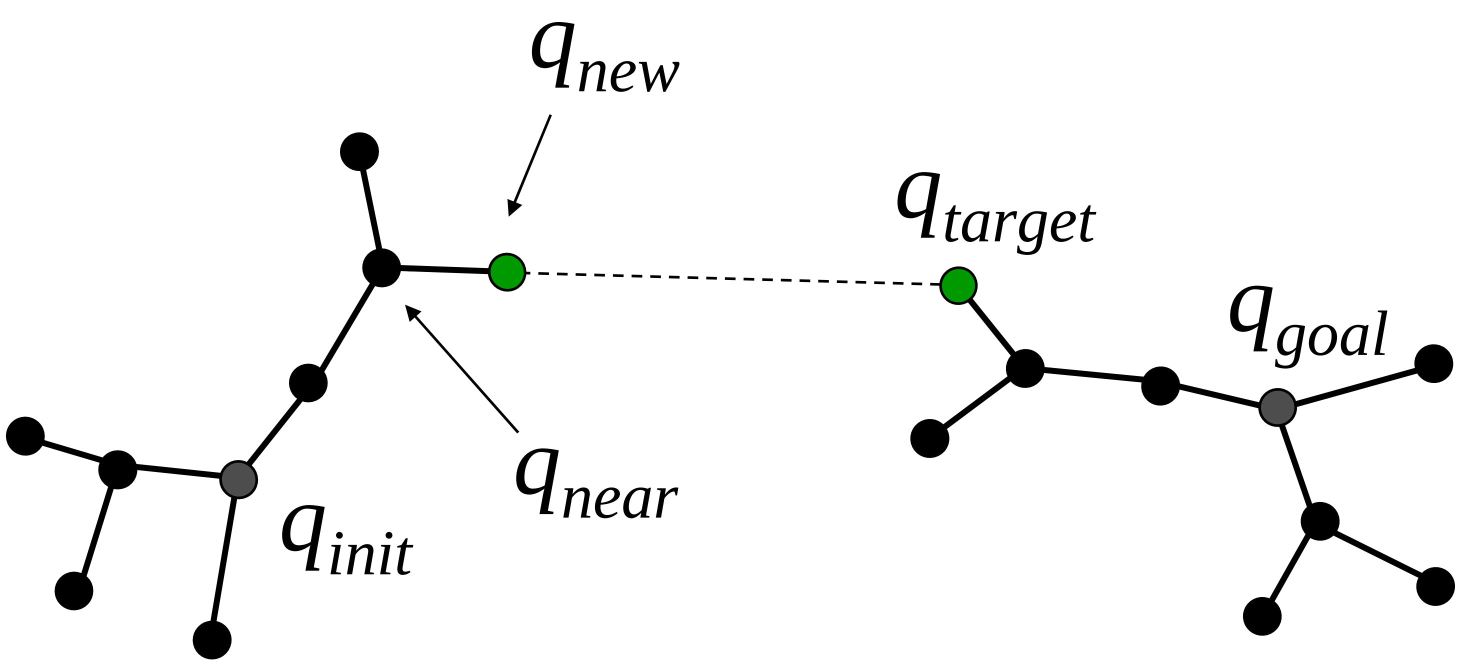
2) New node becomes target for other tree



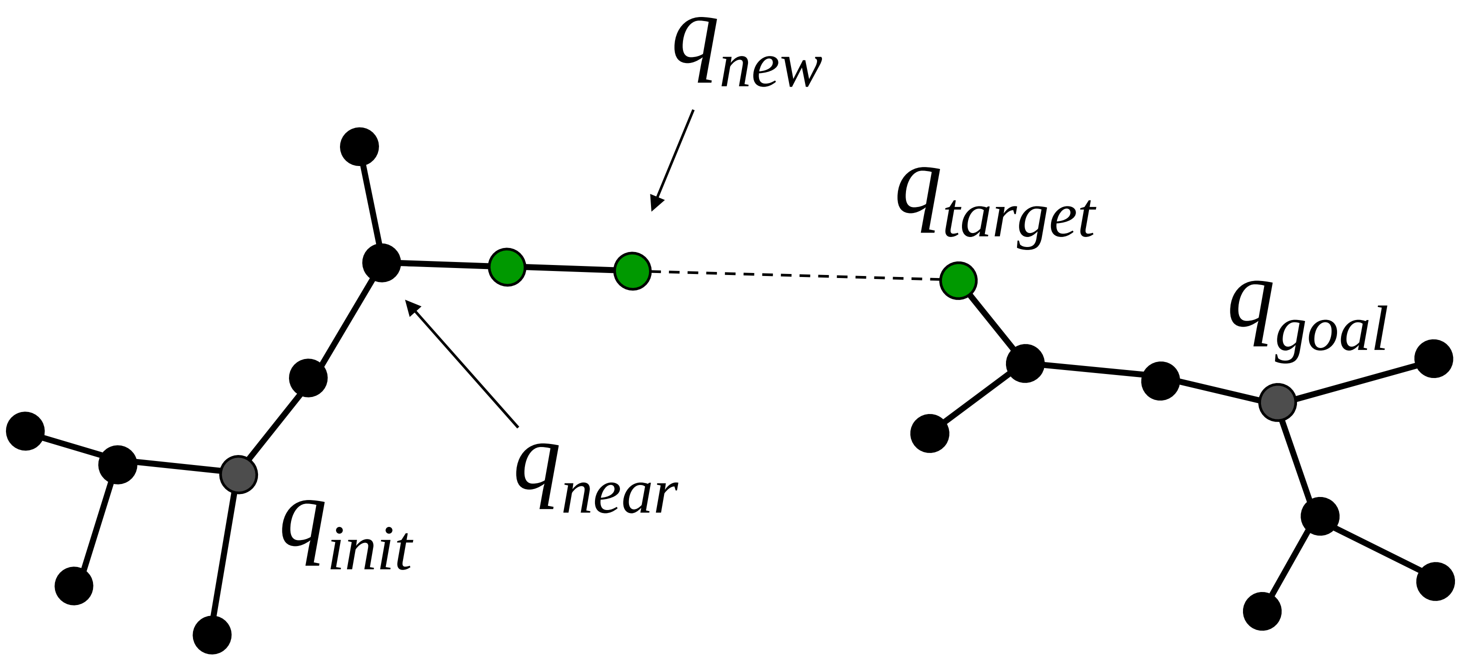
3) Calculate node "nearest" to target



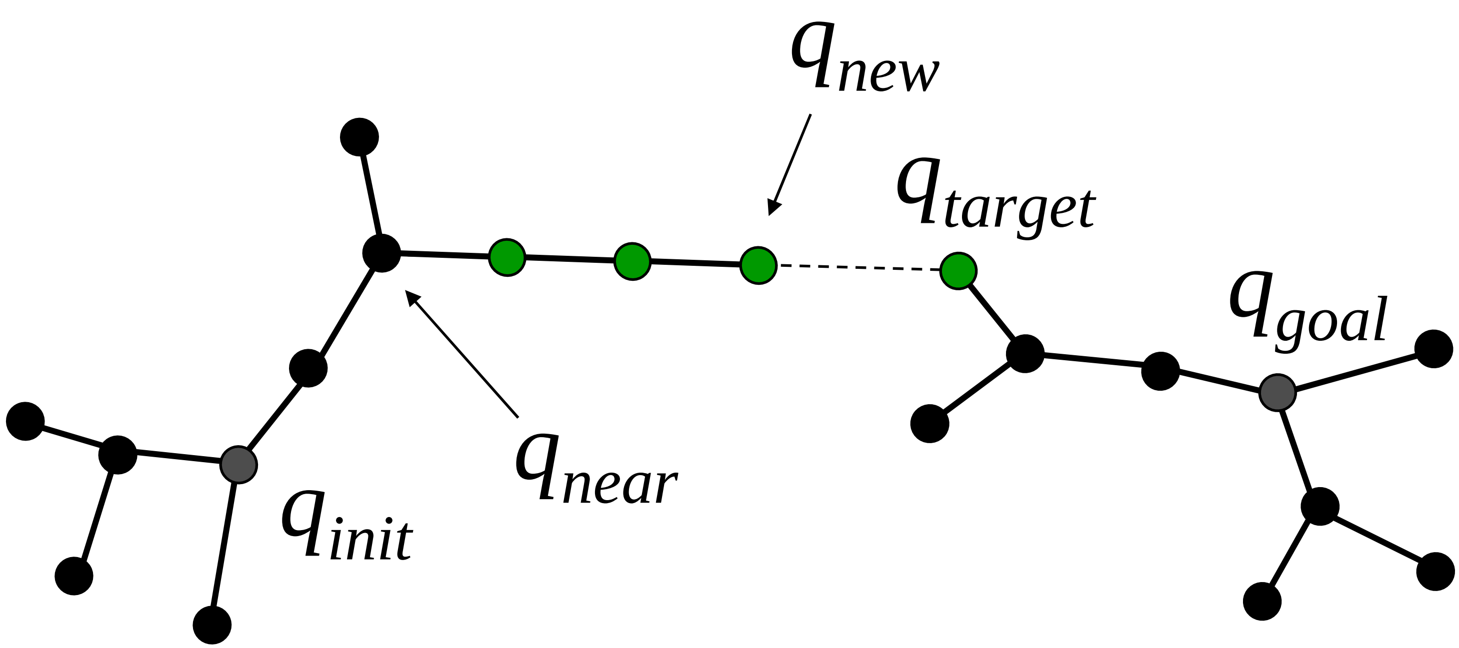
4) Try to add new collision-free branch



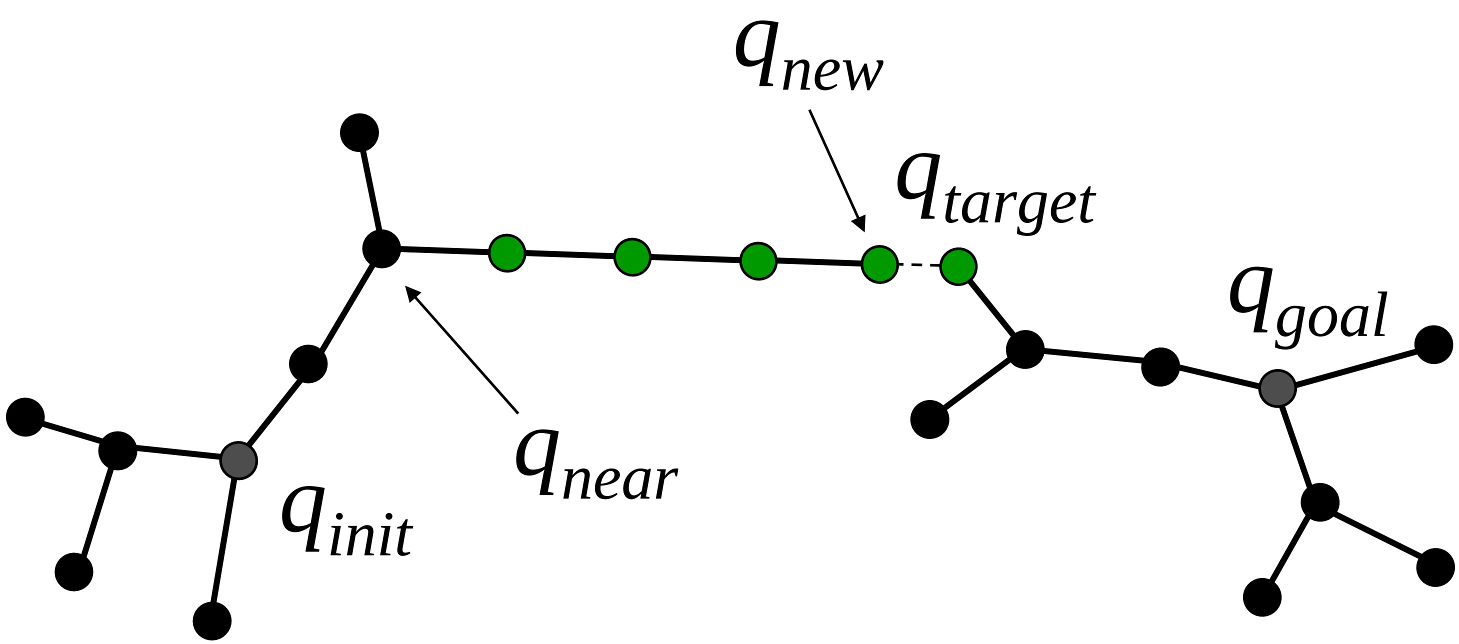
5) If successful, keep extending branch



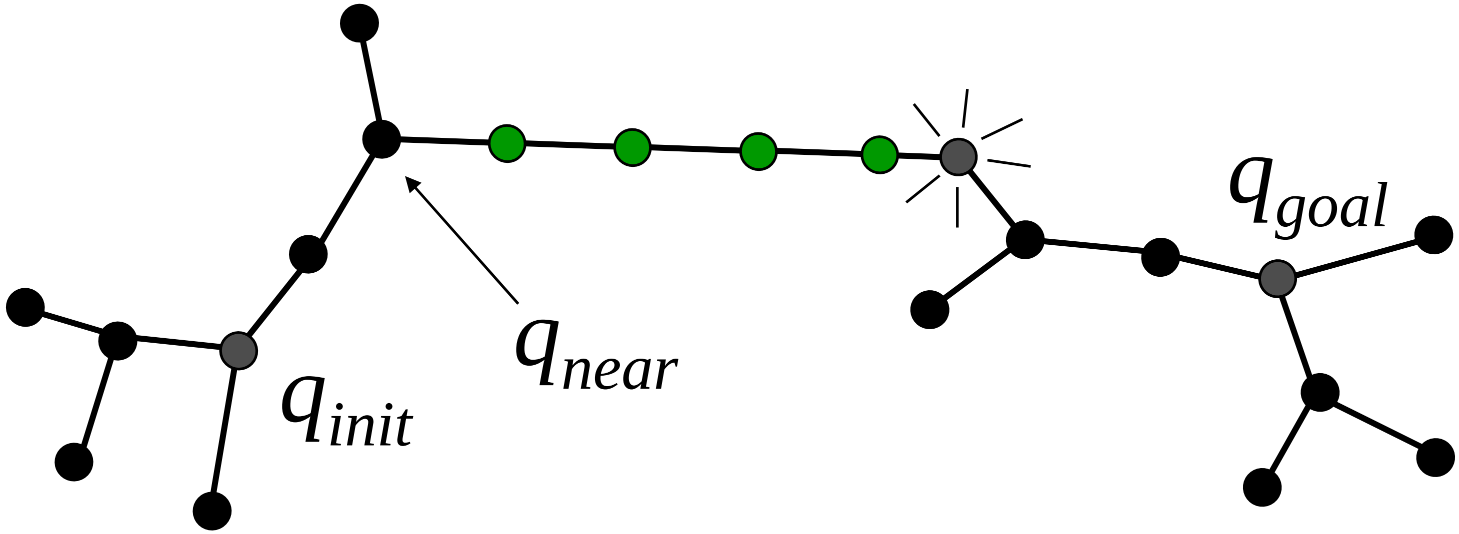
5) If successful, keep extending branch



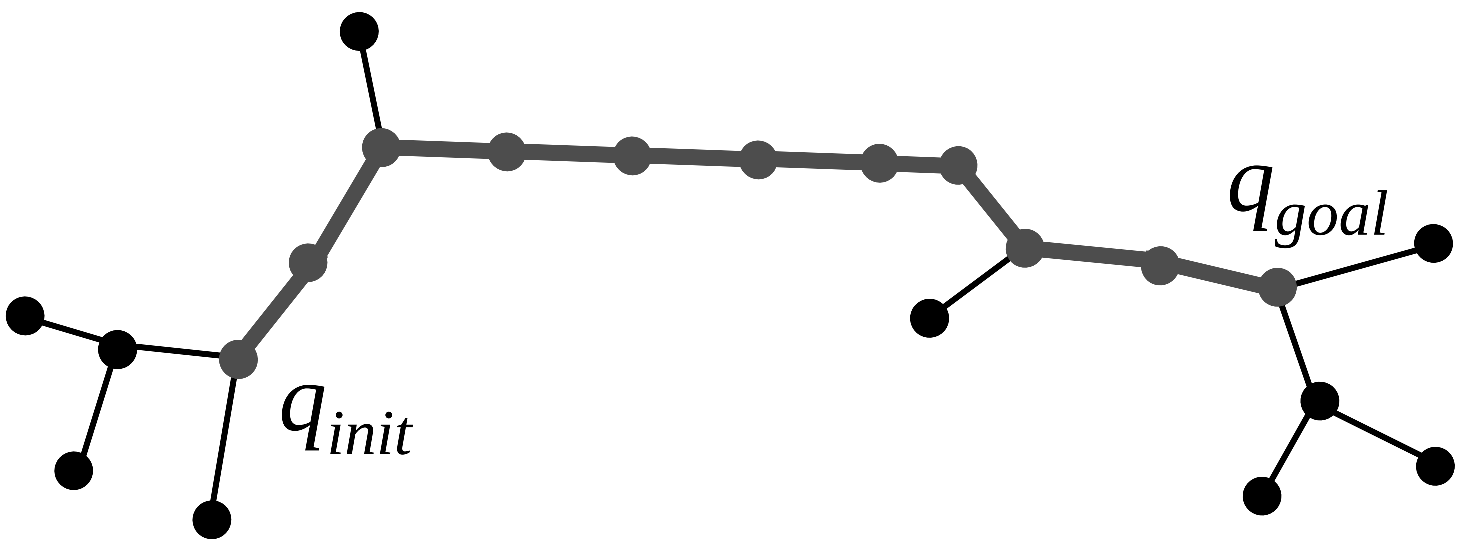
5) If successful, keep extending branch



6) Path found if branch reaches target



7) Return path connecting start and goal

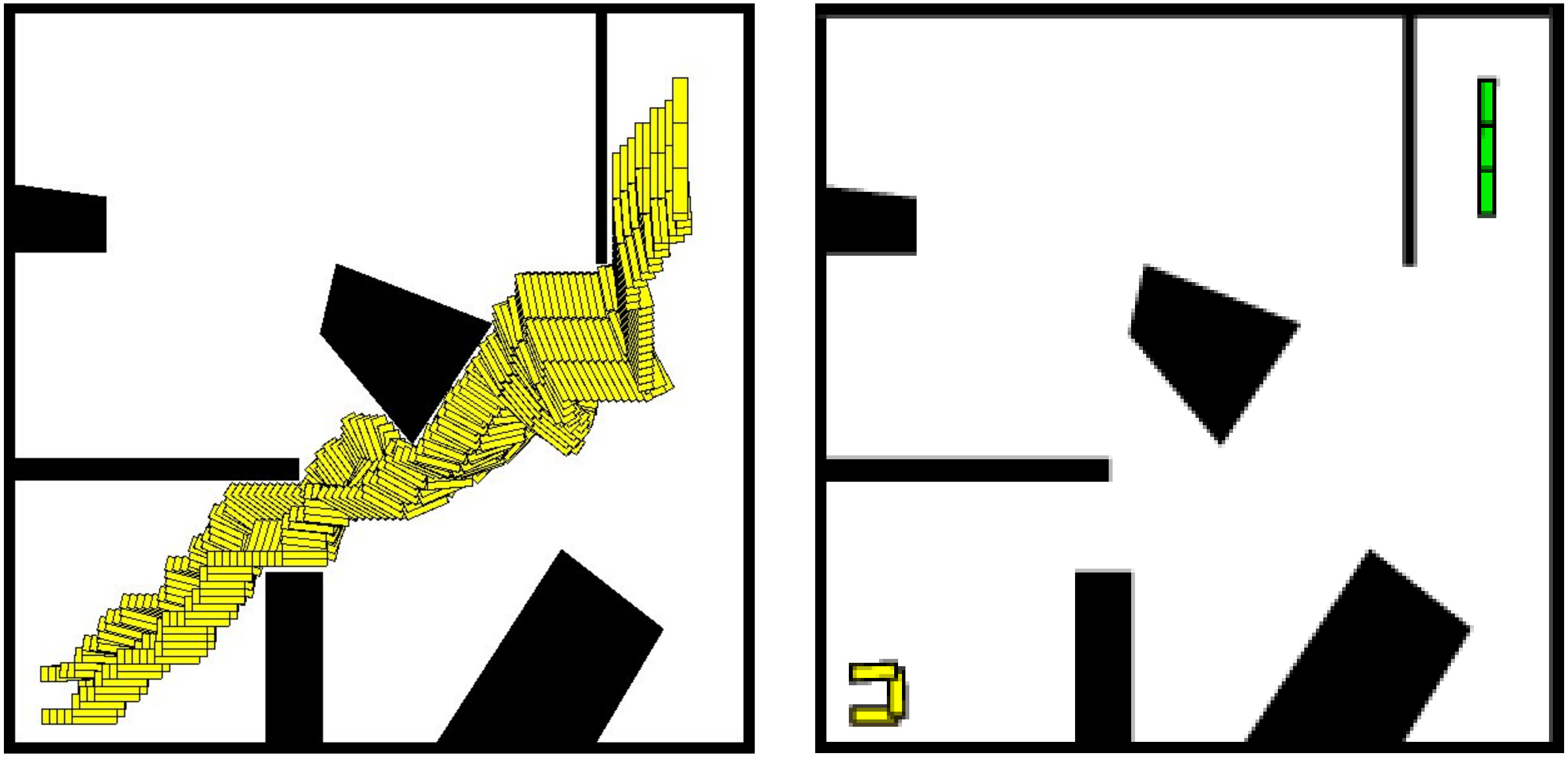


Basic RRT-Connect

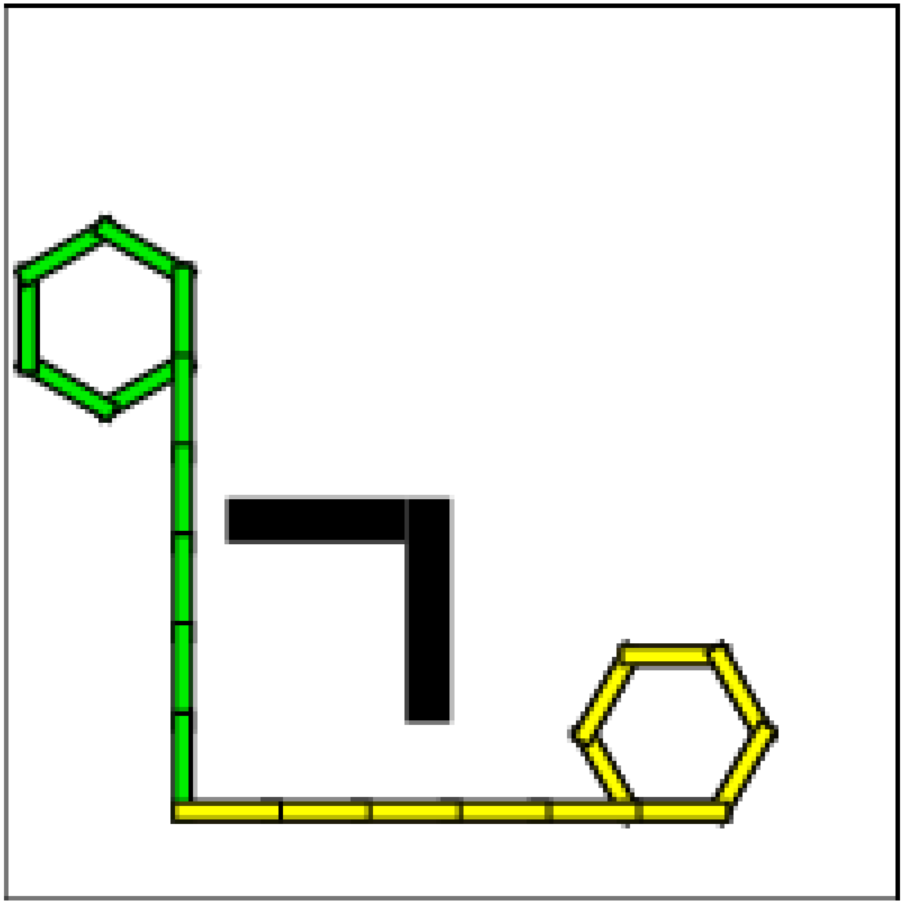
RRT_CONNECT ($q_{init,}$ q_{goal}) {
  T_a.*init*(q_{init}); T_b.*init*(q_{goal});
  for *k* = 1 to K do
    q_{rand} = RANDOM_CONFIG();
    if not (EXTEND(T_a, q_{rand}) = Trapped) then
      if (EXTEND(T_b, q_{new}) = Reached) then
        Return PATH(T_a, T_b);
    SWAP(T_a, T_b);
  Return Failure;
}

Instead of switching, use T_a as smaller tree. This helped James a lot

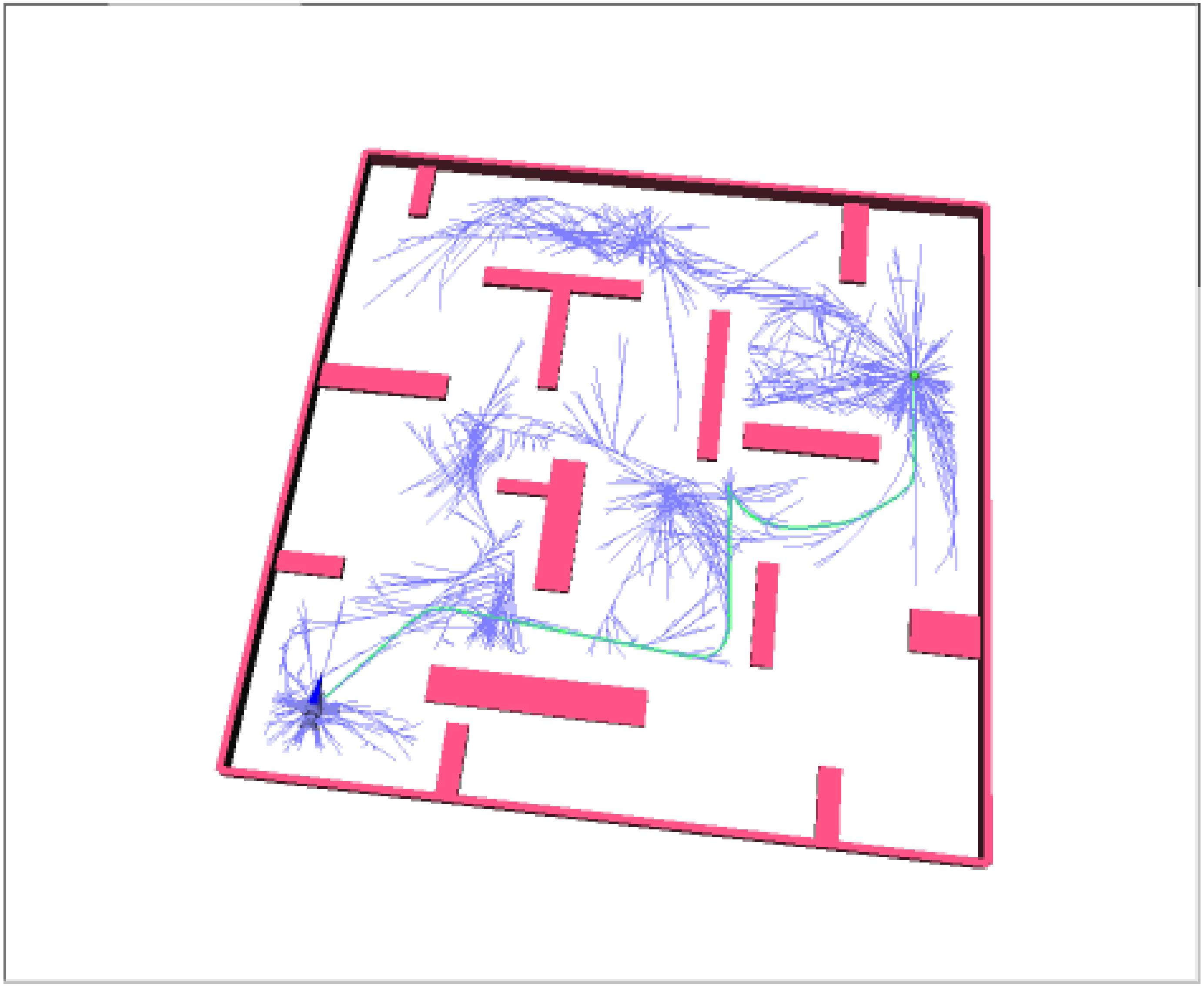
Articulated Robot



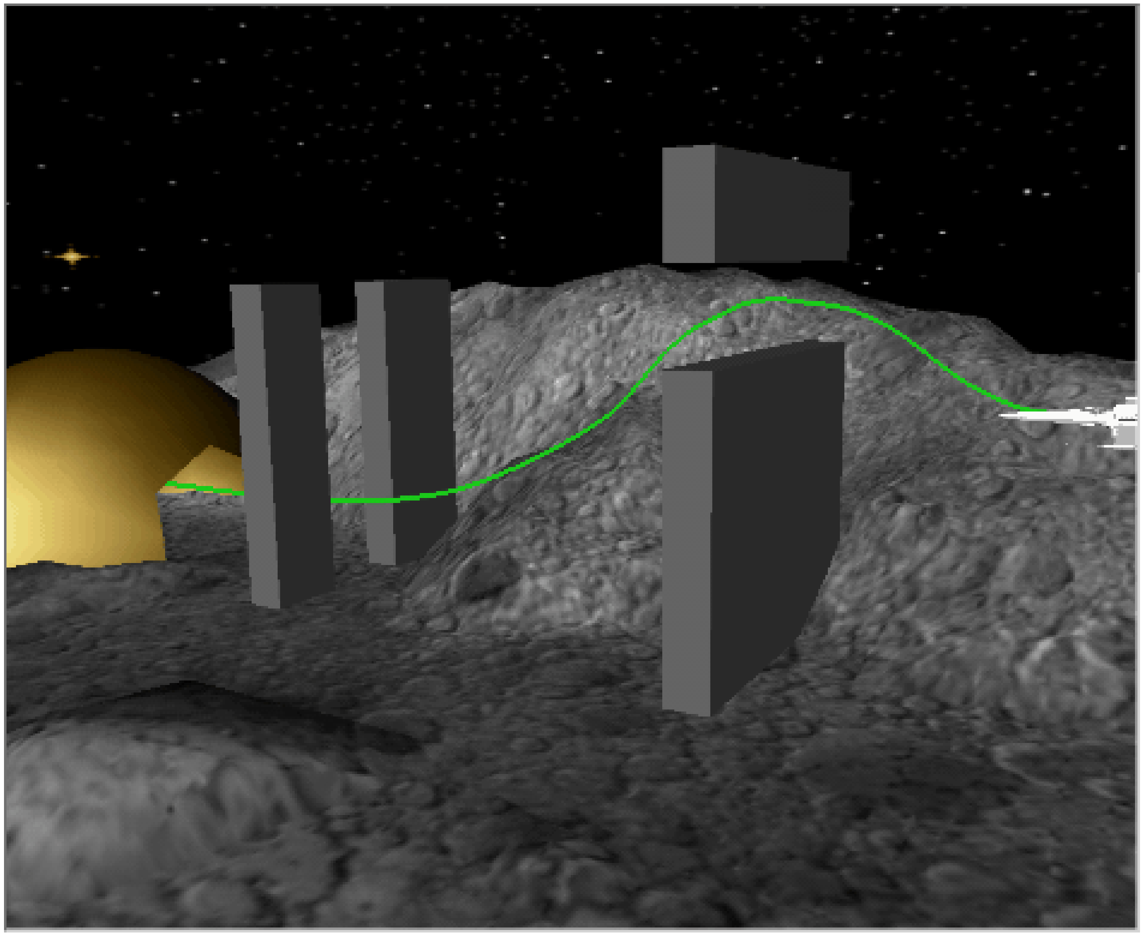
Highly Articulated Robot



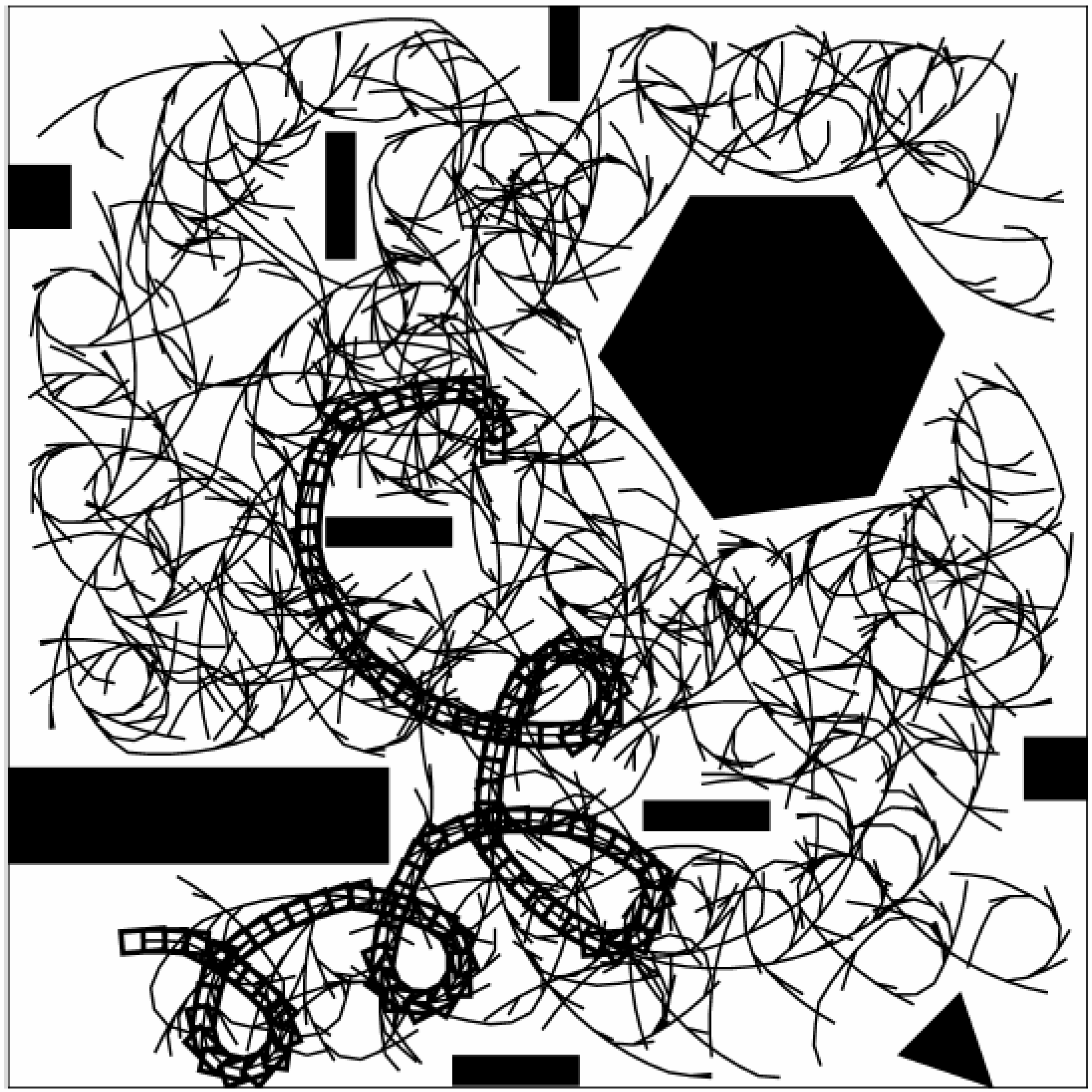
Hovercraft with 2 Thusters



Out of This World Demo



Left-turn only forward car



Open Problems

Open Problems

- Rate of convergence
- Optimal sampling strategy?

Open Issues

- Metric Sensitivity
- Nearest-neighbor Efficiency

Applications of RRTs

Robotics Applications
- mobile robotics
- manipulation
- humanoids

Other Applications
- biology (drug design)
- manufacturing and virtual prototyping (assembly analysis)
- verification and validation
- computer animation and real-time graphics
- aerospace

RRT extensions
- discrete planning (STRIPS and Rubik's cube)
- real-time RRTs
- anytime RRTs
- dynamic domain RRTs
- deterministic RRTs
- parallel RRTs
- hybrid RRTs

RRT Summary

Advantages

- Single parameter
- Balance between greedy search and exploration
- Converges to sampling distribution in the limit
- Simple and easy to implement

Disadvantages

- Metric sensitivity
- Nearest-neighbor efficiency
- Unknown rate of convergence
- “long tail” in computation time distribution

Links to Further Reading

- Steve LaValle’s online book:
  “Planning Algorithms” *(chapters 5 & 14)*
  http://planning.cs.uiuc.edu/

- The RRT page:
  http://msl.cs.uiuc.edu/rrt/

- Motion Planning Benchmarks
  Parasol Group, Texas A&M
  http://parasol.tamu.edu/groups/amatogroup/benchmarks/mp/