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

RI 16-735, Howie Choset with slides from James Kuffner
Path Planning with RRTs  
(Rapidly-Exploring Random Trees)

**BUILD_RRT**(\(q_{\text{init}}\))  
\[
\begin{align*}
T.init(q_{\text{init}}); \\
&\text{for } k = 1 \text{ to } K \text{ do} \\
&q_{\text{rand}} = \text{RANDOM\_CONFIG}(); \\
&\text{EXTEND}(T, q_{\text{rand}})
\end{align*}
\]

[ Kuffner & LaValle, ICRA’00]

**EXTEND**(\(T, q_{\text{rand}}\))

[ Kuffner & LaValle, ICRA’00]
Path Planning with RRTs
(Some Details)

BUILD_RRT \( (q_{init}) \)  { 
  \( T.init(q_{init}); \)
  for \( k = 1 \) to \( K \) do 
    \( q_{rand} = \text{RANDOM_CONFIG}(); \)
    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, can’t 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

RI 16-735, Howie Choset with slides from James Kuffner
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

RI 16-735, Howie Choset with slides from James Kuffner
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

$q_{new}$

$q_{init}$

$q_{near}$

$q_{target}$

$q_{goal}$

[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

$q_{\text{init}}$ $q_{\text{near}}$ $q_{\text{new}}$ $q_{\text{target}}$ $q_{\text{goal}}$
5) If successful, keep extending branch

$q_{\text{init}}$, $q_{\text{near}}$, $q_{\text{new}}$, $q_{\text{target}}$, $q_{\text{goal}}$
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 \((q_{\text{init}}, q_{\text{goal}})\)  
\[
\begin{aligned}
& T_a.\text{init}(q_{\text{init}}); \quad T_b.\text{init}(q_{\text{goal}}); \\
& \text{for } k = 1 \text{ to } K \text{ do} \\
& \quad q_{\text{rand}} = \text{RANDOM_CONFIG}(); \\
& \quad \text{if not (EXTEND} (T_a, q_{\text{rand}}) = \text{Trapped}) \text{ then} \\
& \quad \quad \text{if (EXTEND} (T_b, q_{\text{new}}) = \text{Reached}) \text{ then} \\
& \quad \quad \quad \text{Return } \text{PATH}(T_a, T_b); \\
& \quad \quad \text{SWAP}(T_a, T_b); \\
& \quad \text{Return Failure;}
\end{aligned}
\]

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

• 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)*

- The RRT page:

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