Robotic Motion Planning: RRT’s

Robotics Institute 16-735
http://www.cs.cmu.edu/~motion

Howie Choset
http://www.cs.cmu.edu/~choset
Overview

• Probabilistic RoadMap Planning (PRM) by Kavraki
  – samples to find free configurations
  – connects the configurations (creates a graph)
  – is designed to be a multi-query planner

• Expansive-Spaces Tree planner (EST) and Rapidly-exploring Random Tree planner (RRT)
  – are appropriate for single query problems

• Probabilistic Roadmap of Tree (PRT) combines both ideas
Next HW Assignment

• Implement a PRM planner for a multi-link (at least four) robot arm. The arm can be a simple planar arm (which will simplify the graphics), or a 3D arm. The arm can be composed of line segments (which will make collision checking easier) rather than finite volume links. All you need to do is write code to detect the intersection between line segments and polygons. If you want, you can use collision checking software that is available on the web.

• How was the previous?

• This is the last one
Rapidly-Exploring Random Trees (RRTs)  
[Kuffner, Lavalle]

The Basic RRT  
  single tree  
  bidirectional  
  multiple trees (forests)

RRTs with Differential Constraints  
  nonholonomic  
  kinodynamic systems  
  closed chains

Some Observations and Analysis  
  number of branches  
  uniform convergence  
  resolution completeness  
  leaf nodes vs. interior nodes

Performance & Implementation Issues  
  Metrics and Metric sensitivity  
  Nearest neighbors  
  Collision Checking  
  Choosing appropriate step sizes
High-Dimensional Planning as of 1999

**Single-Query:**

Barraquand, Latombe ’89; Mazer, Talbi, Ahuactzin, Bessiere ’92; Hsu, Latombe, Motwani ’97; Vallejo, Jones, Amato ’99;

**Multiple-Query:**

Kavraki, Svestka, Latombe, Overmars ’95; Amato, Wu ’96; Simeon, Laumound, Nissoux ’99; Boor, Overmars, van der Stappen ’99;

**EXAMPLE: Potential-Field**

Greedy, can take a long time but good when you can dive into the solution

**EXAMPLE: PRM**

Spreads out like uniformity but need lots of sample to cover space

RI 16-735, Howie Choset with slides from James Kuffner
Rapidly-Exploring Random Tree

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

BUILD_RRT \( (q_{init}) \)  
\[
\text{T.init}(q_{init}); \\
\text{for } k = 1 \text{ to } K \text{ do} \\
\quad q_{rand} = \text{RANDOM_CONFIG}(); \\
\quad \text{EXTEND}(T, q_{rand})
\]

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

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

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

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

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

- Continuous
  
  Continuum of choices

  Probabilistically subsample all edges

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
RRT vs. RPP

Greedy gets you stuck here

RRT’s will pull away and better approximate cost-to-go

goal
Grow two RRTs towards each other

$q_{new}$

$q_{near}$

$q_{target}$

$q_{goal}$

[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

$q_{init}$

$q_{near}$

$q_{new}$

$q_{target}$

$q_{goal}$

RI 16-735, Howie Choset with slides from James Kuffner
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
Mixing position and velocity, actually mixing position, rotation and velocity is hard
So, what do they do?

• Use nearest neighbor anyway

• As long as heuristic is not bad, it helps
  (you have already given up completeness and optimality, so what the heck?)

• Nearest neighbor calculations begin to dominate the collision avoidance (James says 50,000 nodes)

• Remember K-D trees
Articulated Robot
Highly Articulated Robot
Hovercraft with 2 Thusters
Out of This World Demo
Left-turn only forward car
Analysis

The limiting distribution of vertices:

- **THEOREM**: $X_k$ converges to $X$ in probability
  - $X_k$: The RRT vertex distribution at iteration $k$
  - $X$: The distribution used for generating samples

- **KEY IDEA**: As the RRT reaches all of $Q_{free}$, the probability that $q_{rand}$ immediately becomes a new vertex approaches one.

Rate of convergence:

- The probability that a path is found increases exponentially with the number of iterations.

"This is the bain or the worst part of the algorithm," J. Kuffner
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
Diffusion Limited Aggregation

- Often used to model natural physical processes (e.g. snow accumulation, rust, etc.)
Exploring Infinite Space
Polar Sampling

RI 16-735, Howie Choset with slides from James Kuffner
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/
PRT (Prob. Roadmap of Trees)

• Basic idea:
  – Generate a set of trees in the configuration space
  – Merge the trees by finding nodes that can be connected

• Algorithm
  – pick several random nodes
  – Generate trees $T_1, T_2 \ldots T_n$ (EST or RRT)
  – Merge trees
    • generate a representative super-node
    • Using PRS ideas to pick a neighborhood of trees
    • $\Delta$ is now the tree-merge algorithm
  – For planning
    • generate trees from initial and goal nodes towards closest supernodes
    • try to merge with “roadmap” of connected trees

• Note that PRS and tree-based algorithms are special cases