RRT - Rapidly-Exploring Random Trees

Tim Jagla, André Keuns, Anne Reich

Otto-von-Guericke-Universität Magdeburg

February 1, 2015

Collision Free Path Planning

Motivation

- path planning: find a path from location *A* to *B*
- Example for path planning:
  - mobile robot inside a build
  - shall go to location XY
- Example extension for collision free path planning:
  - avoiding walls and not falling stairs

Simple Example

- Simple General Forward Search
  - State: Unvisited, Dead, Alive
  - Priority queue, Q, with the set of alive states
  - Start loop over Q
  - In each while iteration check next state
    - It is the goal, is terminate
    - Otherwise, it tries applying every possible action

Algorithms

- Other known collsion free path planning algorithms
  - Breadth first
  - Deep first
  - Dijikstra‘s algorithm
  - A*
  - Best First
  - Backward search
  - ...

Principles

- Basic Ingredients of Planning
  - State
  - Input
  - Initial and goal states
  - A criterion: Feasibility and/or Optimality
  - a plan

Rapidly-Exploring Random Trees

Principles

- Grows a tree tooted at the starting configuration by using random samples from the search space
- As each sample is drawn, a connection is attempted between it an the nearest state in the tree
- If the connection is feasible, this results in the addition of the new state to the tree
- The probability of expanding an existing state is proportional to the size of ist Voronoi region
  - As the largest Voronoi regions belong to the states on the frontier of the search = the tree preferentially expands towards large unsearched areas

Notations

$$
\begin{aligned}
X &= \text{metric space} \\
x_{init} &= \text{initale state} \\
X_{goal} &= \text{goal region, } X_{goal} \subset X \\
C &= \text{configuration space of a rigid body or} \\
& \text{systems of bodies in a world} \\
T &= \text{RRT (Tree of vertices)} \\
T(C) &= \text{tanget bundle of the configuration space} \\
X_{obs} &= \text{obstacle region, } X_{obs} \subset X \\
X_{free} &= \text{region without obstacles, } X_{free} \subset X
\end{aligned}
$$

Each edge of the RRT will correspond to a path that lies entirely in X_{free}

General Procedure

- start with x_init and T, with K vertices
- in each iteration a random state, x_rand, is selected from X
- find the closest vertex to x_rand with the terms of a distance metric
- select an input that minimizes the distance from closest vertex to x_rand and check that the state is in X_{free}
- is it free, add this new state as a vertex to T

Nice Properties

- The expansion is heavily biased toward unexplored portions of the state space
- The distribution of vertices approaches the sampling distribution, leading to consistent behavior
- Is probabilistically complete under very general conditions
- sThe algorithm is relatively simple, which facilitates performance analysis

Nice Properties

- It always remains connected, even though the number of edges is minimal
- Can be considered as a plat planning module, which can be adapted and incorporated into a wide variety of planning systems
- Entire plath planning algorithms can be constructed without requiring the ability to steer the system between two prescibed states, which greatly broadens the applicability of RRTs

Challanges of our work

überlegung 2d -> 3d -> 6d / configuration space

Pseudo Code

Live Demo

graph aufbau vollständigen graphen

Sources

1 Rapidly-Exploring Random Trees: A New Tool for Path Planning - Steven M. LaValle
http://coitweb.uncc.edu/~xiao/itcs6151-8151/RRT.pdf (03.02.2015)

2 Planning Algorithms - Steven M. LaValle
http://planning.cs.uiuc.edu/ (03.02.2015)

3 http://en.wikipedia.org/wiki/Rapidly_exploring_random_tree (03.02.2015)

Thank you for your attention