Robot Motion Planning

Sampling Based Algorithms

A. Narayanan[1]

[1]Department of Informatics

August 16, 2022

Outline

Overview of Sampling Based Approaches

Probablilistic Roadmaps

- Expansiveness

Difficulty with Classical Approaches

- Curse of dimensionality
- Running time increases exponentially with dimensions of configuration space
- Several variants of the path planning problem have proven to be PSPACE-hard

Drawbacks of Combinatorial Approaches

Multiple Query Roadmaps

A multiple query approach tries to capture the connectivity of the free space as good as possible, such that multiple, different queries for paths can be answered very fast. In other words: create a roadmap that is suitable for as many use cases as p

PRM - Probablistic Roadmaps

Basic steps to construct a PRM:

1. sample vertices and keep vertices that do not lie on an obstacle
2. find neighbour vertices - k-nearest neighbour or neighbours within a specified radius
3. connect neighbouring vertices with edges (lines) (and check for collisions on connecting line using e.g. discretized line search)
4. add vertices and edges until roadmap is dense enough

PRM Visualized

Drawbacks

PRMs don't perform well when there are narrow passages.

How to Improve?

- Increase number of milestones
- Random walk
- Path Correction
- Sample at obstacle boundaries

OBPRM - Obstacle Based PRM

Obstacle-based PRMs are constructed by sampling only close to obstacles. During sampling, the first goal is to find a point that lies inside an obstacle. Then, another point is sampled at an arbitrary distance to the first point. Using step-wise approximation, a point sufficiently close to the obstacle border is searched.

Gaussian Sampler

- The Gaussian sampler addresses the narrow passage problem by sampling from a Gaussian distribution that is biased near the obstacles.

Steps:

1. first generate a configuration q_1 randomly from a uniform distribution.
2. Then a distance step is chosen according to a normal distribution to generate a configuration q_2 at random at distance step from q_1.
3. Both configurations are discarded if both are in collision or if both are collision-free. A sample is added to the roadmap if it is collision-free and the other sample is in collision.

Sampling Inside Narrow Passages - Bridge Planner

- The bridge planner uses a bridge test to sample configurations inside narrow passages.

1. sample 2 configurations q' and q" from a uniform distribution in $\mathcal{Q}$
2. add to roadmap if collision free
3. if both are in collision then add q_m halfway between them to the roadmap if it is collision free

Resample (random-bounce walk)

if there is no path found, using random-bounce walk to generate new milestones, improve the C-space connectivity.
A random-bounce walk from q refers to picking a random direction of motion and moving in this direction until a collision occurs, then a new random direction is chosen. The above steps are repeated until the max. distance is reached, then the current configuration q' and the edge (q, q) are inserted into the roadmap.

Single Query Roadmaps

Single query planners try to solve a single query as fast as possible, without trying to cover the whole free space

Single Query PRM

PRM itself can also be used as single-query planner. In that case, q_{init} and q_{goal} should be inserted to the roadmap at the beginning. The planner should check periodically if the given query can be solved, that is if q_{init} and q_{goal} belong to the same component of the roadmap. At that point, the construction of the roadmap should be aborted

Weighted Randomized Tree Expansion

1. expand trees from start and goal
2. pick a node with probability $= 1/w(x)$, with $w(x)$ being the amount of neighbors within radius (measurement for exploration around x)
3. sample k points $(y_1, \ldots, y_k)$ around x
4. add y_i to the tree if
   - $1/w(y_i) > 1/w(x)$
   - y_i is collision free
   - y_i can see x
5. if a pair of nodes from start tree and goal tree are close and can see each other, then connect them and terminate

Rapidly Exploring Random Trees

1. pick q_{start} as the first node
2. pick a random target location (every n^{th} iteration, choose q_{goal})
3. find closest vertex in roadmap
4. extend this vertex towards target location
5. repeat steps until q_{goal} is reached

For faster execution, the tree can be grown from both q_{start} and q_{goal} (RRT-Connect)

How to sample

Path Optimization

Outline

ϵ, α, β - Expansiveness

- A PRM has good `coverage` if the milestones are distributed in such a way that (almost) any point in the free C-space can be connected to one milestone via a straight line.
- A PRM has good `connectivity` if every milestone is reachable from any other milestone
- The coverage and connectivity are characterized by the **Expansiveness** of the space given by ϵ, α, β

Definition

A free space F is $(\epsilon, \alpha, \beta)$-expansive, if it satisfies these conditions:

1. $\mu(V(p)) \geq \epsilon, \forall p \in F$
   $\Longrightarrow$ each point p must see at least an ϵ fraction of the free space $F \rightarrow F$ is ϵ-good
2. $\beta - \mathit{lookout}(S) = \{q \in S \mid \mu(V(q) \setminus S) \geq \beta \cdot \mu(F \setminus S)\} \Longrightarrow$ for any $S \subseteq F$, the $\beta - \mathit{lookout}(S)$ is defined as the subset of points p that can see at least a β-fraction of the complementary space of $S (\equiv F \setminus S)$
3. α is then defined as the relative volume of these points p to S: $\frac{\mu(\beta - \mathit{lookout}(S))}{\mu(S)}$

with $\epsilon, \alpha, \beta \in (0, 1]$, $V(\cdot)$ as the set of visible points of a point and $\mu(\cdot)$ denoting the volume of the set of points