Computational Geometry
csci3250

Laura Toma

Bowdoin College

Approximate Path Planning

Geometric (combinatorial) path planning

- Approach
  - **Compute free C-space geometrically**
    - compute C-obstacles as Minkowski sums and their union
  - **Compute a partition of C-free**
    - trapezoidal partition
  - **Compute a complete roadmap**

- **Geometric Roadmaps**
  - Via trapezoidal partition or triangulation.
  - Shortest Euclidian paths: Visibility graphs
  - Maximum clearance paths: Voronoi diagram of obstacles

- Comments
  - Complete
  - Works beautifully in 2D and for some simple cases in 3D
  - Worst-case bound for combinatorial complexity of C-objects in 3D is high
  - Unfeasible for high #DOF
    - A complete planner in 3D runs in $O(2^{n^{\wedge}\#DOF})$

Approximate path planning

- Roadmaps
- Sampling
- Cell decomposition

Approximate path planning

- **Roadmaps**
  - Compute a roadmap that captures connectivity of C-free

- **Sampling**
  - When dimension of C-space is high it is hard to construct the C-obstacles
  - However it is much easier to “sample” a certain placement and determine if it is in C-free or not.
    - this ends up being a collision test: would my robot , if placed at this position and in this configuration, intersect any obstacle?
  - By sampling, we can determine what points are in C-free or not
  - If the sampling is dense, it implicitly constructs a discretized version of C-free

Sampling

- You are not given the representation of C-free: Imagine being blindfolded in a maze
- Sampling: you walk around hitting your head on the walls
- Left long enough, after hitting many walls, you have a pretty good representation of the maze
- However the space is huge
  - e.g. DOF= 6: 1000 x 1000 x 1000 x 360 x 360 x 360
- So you need to be smart about it and chose the points you sample

Sampling and Roadmaps

- Instead of computing C-free explicitly, compute a roadmap that captures its connectivity
- Construct roadmap by sampling free space
- We can sample C-space
  - e.g. with a grid; or, randomly; or, …
  - one shot, or incremental
  - ..
- If C-free is not sampled adequately, we may not detect paths that go through narrow passages
- Tradeoff: to capture all possible paths in C-free we want dense sampling, but this is infeasible because it’s huge

Approximate path planning

- Goal: Approximate in some way C-free

- Approaches
  - Approximate the robot and/or obstacles by simple objects: (hierarchies of) bounding boxes or spheres
  - Cell decomposition method: Sample with uniform grid/quadtree/octree. This creates a huge roadmap which can be searched using AI techniques to direct the search towards a path
  - Potential method
  - Probabilistic roadmaps
  - RRT
  - …. any other approaches?
- Approaches are usually hybrid: combine ideas above

Use spheres and BB's

- Approximate the robot and/or obstacles by simple objects: (hierarchies of) bounding boxes or spheres
- Issues
  - cluttered scenes
  - complicated shapes

Grid roadmaps/cell decomposition

- Sample C-space with uniform grid/quadtree/octree.
- This essentially "pixelizes" the space
- May compute an occupancy grid
- Graph is implicit
  - given by grid topology: move +/-1 in each direction
- Can do it on one-shot (pre-compute it), or incrementally as needed

- This creates a huge roadmap which can be searched using AI techniques to direct the search towards a path

Potential field methods

- Idea [Latombe et al, 1992]
  - Define a potential field
  - Robot moves in the direction of steepest descent on potential function
- Ideally potential function has global minimum at the goal, has no local minima, and is very large around obstacles
- Algorithm outline:
  - place a regular grid over C-space
  - search over the grid with potential function as heuristic

- Pro: Nice framework that can be adapted to any specific scene
- Con: can get stuck in local minima
  - Potential functions that are minima-free are known, but expensive to compute
- RPP (Randomized path planner) is based on potential functions
  - escapes local minima by executing random walks
  - succesfully used to
    - performs riveting ops on plane fuselages
    - plan disasembly ops for maintenance of aircraft engines

Probabilistic Roadmaps

- Latombe, Overmaps et al , 1996
- Approach
  - **Construction phase:**
    - Construct roadmap representing C-free by repeatedly sampling and connecting
  - **Query phase:**
    - Use roadmap to find path between any two points

Probabilistic Roadmaps

- Constructing the roadmap
  - Start with uniform sampling of points in C-free and try to connect them
    - two points are connected by an edge if a simple quick planner can find a path between them
  - This will create a set of trees
  - Augment roadmap by selecting additional sample points in areas that are estimated to be "difficult"

- Comments
  - Roadmap adjusts to the density of free space and is more connected around the obstacles
  - Size of roadmap can be adjusted as needed
  - More time spent in the "learning" phase ==> better roadmap
  - Roadmap stored as set of tree for space efficiency
    - trees encode connectivity, cycles don't change it. Additional edges are useful for shortest paths, but not for completeness

Probabilistic Roadmaps

(1) $N \leftarrow \emptyset$

(2) $E \leftarrow \emptyset$

(3) **loop**

(4) $\quad c \leftarrow$ a randomly chosen free configuration

(5) $\quad N_c \leftarrow$ a set of candidate neighbors of c chosen from N

(6) $\quad N \leftarrow N \cup \{c\}$

(7) $\quad$ **for all** $n \in N_c$, in order of increasing $D(c, n)$ **do**

(8) $\quad\quad$ **if** $\neg \mathit{same\_connected\_component}(c, n) \wedge \Delta(c, n)$ **then**

(9) $\quad\quad\quad E \leftarrow E \cup \{(c, n)\}$

(10) $\quad\quad\quad$ update R's connected components

- Components
  - sampling C-free
  - the local planner
  - selecting the neighbors
  - the expansion step and the heuristical measure of difficulty of a node

Probabilistic Roadmaps

Comments

- One of the leading motion planning technique
- Efficient, easy to implement, applicable to many types of scenes
- Embraced by many groups, many variants of PRM's, used in many type of scenes.
- Not clear which technique better in which circumstances