Today's lesson

- shortest paths
- high clearance paths
- other quality measures
- combined quality criteria and corridor maps
- path quality in sampling-based planners

Shortest paths among obstacles in the plane

First attempt: Dijkstra on the connectivity graph of the trapezoidal map

Lesson

- Does the graph on which we are searching for the best paths contain the best paths?

Properties of the shortest path

A polygonal line whose vertices are the start and goal configurations and vertices of the obstacles
Computing a shortest path

Algorithm \textsc{ShortestPaths}(p_{out}, p_{goal})
\begin{enumerate}
\item \textsc{VisibilityGraphs}(G)
\item Assign each arc \((v, w)\) in \(G\), a weight, which is the Euclidean length of the segment \(vw\).
\item Use Dijkstra's algorithm to compute a shortest path between \(p_{out}\) and \(p_{goal}\) in \(G\).
\end{enumerate}

Computing the visibility graph

Algorithm \textsc{VisibilityGraphs}(G)
\begin{enumerate}
\item Initialize a graph \(G' = (V, E)\) where \(V\) is the set of all vertices of the polygons in \(S\) and \(E = \emptyset\).
\item for all vertices \(v \in V\)
\item do \(W = \text{VisibleVertices}(v, S)\)
\item For every vertex \(w \in W\), add the arc \((v, w)\) to \(E\).
\item return \(G'\)
\end{enumerate}

Shortest paths in the plane, complexity

- the visibility-graph algorithm takes \(O(n^2 \log n)\) time where \(n\) is the number of obstacle vertices
- there are output sensitive algorithms (in the size of the visibility graph)
- near-optimal \(O(n \log n)\) algorithm by Hershberger and Suri
- the case of a simple polygon (whose complement is the obstacle) is much simpler

Shortest paths among polyhedra in 3-space

- the setting: point robot moving among polyhedra with a total of \(n\) vertices
- the problem is NP-hard [Canny-Reif]
  - algebraic complexity
  - combinatorial complexity

High clearance paths

- Voronoi diagrams/the medial axis
- the Voronoi diagram of line segments, and the retraction method for a disc [O'Dunlaing-Yap]

Other quality measures

- other \(L_p\) metrics, e.g., Manhattan (\(L_1\))
- link number
- number of reverse movements
- low energy
- weighted regions
- many more

- multiple criterion optimal paths

Video [Schirra/Rohnert]
Combined quality criteria and corridor maps

- a path is called Pareto optimal if no other path has a better value for one criterion without having a worse value for another criterion
- multiple criterion optimization is often hard

Clearance-length combination

- combined measure
- relaxed combination: the visibility Voronoi complex
- corridor maps

Optimizing a combined measure

[Wein-van den Berg-H]

\[ L^*(C) = \int \left( \frac{w_{\text{max}}}{w(t)} \right)^{d-1} dt \]

- examples:
  - the optimal path in the presence of a point obstacle is a logarithmic spiral
  - the optimal path in the presence of a segment obstacle is a circular arc
- approximation algorithms for the general polygonal case

The visibility Voronoi diagram (VVD)

[Wein-van den Berg-H]

- finding the shortest path with a given clearance c, while still allowing to make significant shortcuts with lesser clearance on the Voronoi diagram

The visibility Voronoi complex

- implicitly encodes the VVD for any clearance c
- interpolates between the visibility diagram (c=0) and the Voronoi diagram (c=\infty)
- \(O(n^2 \log n)\) construction time

Corridor maps

[Geraerts-Overmars]

- motivated by motion planning in games
- similar to VVD/VVC, augmenting the VD with clearance information
- instead of providing a single solution path, provides a corridor among static obstacles, where later one can easily maneuver among dynamic obstacles
Path quality in sampling-based planners

- the typical process: building a roadmap graph and running Dijkstra or similar
- recall our earlier test: does the graph on which we are searching for the best paths contain the best paths?
- path quality can be very low
- example: path length in BiRRT

Bi-RRT reminder: Growing two-trees
[Kuffner and LaValle ’00]

- maintain two trees rooted at source & goal
- construction step – sample configurations and expand either tree as in RRT
- merging step – connect configurations from both trees

Example (I) – in OOPSMP

![Example (I) diagram]

- 48.4% of paths are over three times worse than optimal (even after smoothing)
- much larger than the theoretical bound

Example (II) – close-by start and goal configurations

![Example (II) diagram]

- 5.9% of paths are over 140 times worse than optimal (even after smoothing)
- importance of visibility blocking – narrow passages not the only king (theoretical motivation for Visibility PRM, Laumond et al. ’00)

How low can path quality get?

Sampling-Diagram Automata: Analysis of path quality in tree planners
[Nechushtan-Raveh-Halperin, WAFR 2010]

![How low can path quality get? diagram]

- Short-cutting heuristics (‘path smoothing’)
- Retraction towards medial axis
  [e.g., Wilmarth et al. ’99, Gerarts and Overmars ’07]
- Useful Cycles in PRM [Nieuwenhuisen and Overmars ’04]
- Biasing tree growth by a cost-function
  [e.g., Urmson and Simmons ’03, Elfvin and Bleuler ’06, Jaillet et al. ’08, Raveh et al. ’09]

- RRT* – a modification of RRT [Karaman and Frazzoli ’10] (for more variants, see paper)
  - the modified RRT* algorithm converges to an optimal path as running time reaches infinity
  - “Standard” RRT misses the (precise) optimal path with probability one
    Still, might be ε-good, or within same homotopy class as optimal path

Improving path quality in sampling-based motion planning, sample work

- Short-cutting heuristics (‘path smoothing’)
- Retraction towards medial axis
  [e.g., Wilmarth et al. ’99, Gerarts and Overmars ’07]
- Useful Cycles in PRM [Nieuwenhuisen and Overmars ’04]
- Biasing tree growth by a cost-function
  [e.g., Urmson and Simmons ’03, Elfvin and Bleuler ’06, Jaillet et al. ’08, Raveh et al. ’09]

- RRT* – a modification of RRT [Karaman and Frazzoli ’10] (for more variants, see paper)
  - the modified RRT* algorithm converges to an optimal path as running time reaches infinity
  - “Standard” RRT misses the (precise) optimal path with probability one
    Still, might be ε-good, or within same homotopy class as optimal path
Improving quality by path hybridization

Example: move the rod from the bottom to the top of a 2D grid (rotation + translation)

3 randomly generated motion paths

H-Graphs: Hybridizing multiple motion paths (looking for shortcuts)

Hybridizing the paths

General quality criteria

Rod-in-Grid scene: 3 dof

Implemented in the OOPSMP package (Plaku, Moll and Kavraki), collision detection – PQP (Lin and Manocha)
Double-Wrench: 12 dof
Switching the two wrenches (rotation + translation x 2)

Running-time bottleneck for hybridization:
Trying to connect nodes from different paths

H-Graphs become particularly useful for high-dimensional problems (at least in this example)

Simple Heuristic – “Neighborhood H-Graphs”: compare only to nodes in local neighborhood – but can we do better?

Edit-distance string matching
⇒ Linear alignment of motion paths

Comparing “This dog” and “That Dodge” with insertion / deletions / replacement:
T H I – S D O – G –
T H A T – D O D G E

Alignment length is linear
Now testing only O(n) edges along the alignment

Comparison of running times

- hybridizing five motion paths in a 2-D maze:
  - from 3.52 seconds to 0.83 seconds on average (75% decrease), with comparable path quality
Why do Hgraphs work?

- wrong decision can be taken at every step
- can be solved by path-hybridization

References

- Shortest Path and Networks, J.S.B. Mitchell, Chapter 27 of the Handbook on DCG, Goodman-O'Rourke (eds)
- Visibility Graphs, Chapter 15 of the Computational Geometry book by de Berg et al
- more details, more experiments:
  - http://acg.cs.tau.ac.il/projects
    IEEE Trans. on Robotics, 2011

THE END