Algorithmic Robotics and Motion Planning

Spring 2011

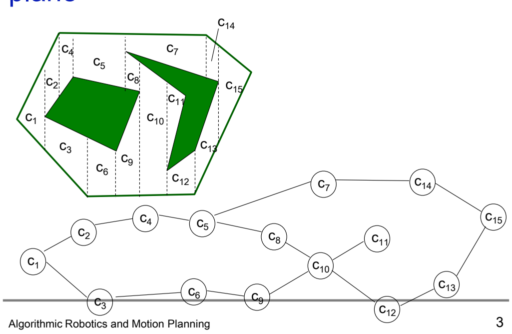
Path quality

Dan Halperin
School of Computer Science
Tel Aviv University

Today's lesson

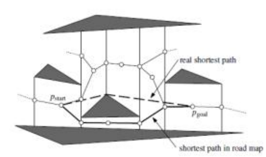
- shortest paths
- high clearance paths
- other quality measures
- combined quality criteria and corridor maps
- path quality in sampling-based planners

Shortest paths among obstacles in the plane



Shortest paths among obstacles in the plane [from de Berg et al, Ch. 15]

- first attempt: Dijkstra on the connectivity graph of the trapezoidal map

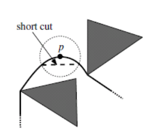


lesson

- 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 SHORTESTPATH($S, p_{\text{start}}, p_{\text{goal}}$)
Input. A set S of disjoint polygonal obstacles, and two points p_{start} and p_{goal} in the free space.
Output. The shortest collision-free path connecting p_{start} and p_{goal}.
1. $\mathcal{G}_{\text{vis}} \leftarrow$ VISIBILITYGRAPH($S \cup \{p_{\text{start}}, p_{\text{goal}}\}$)
2. Assign each arc (v,w) in $\mathcal{G}_{\text{vis}}$ a weight, which is the Euclidean length of the segment $\overline{vw}$.
3. Use Dijkstra's algorithm to compute a shortest path between p_{start} and p_{goal} in $\mathcal{G}_{\text{vis}}$.

Computing the visibility grpah

Algorithm VISIBILITYGRAPH(S)
Input. A set S of disjoint polygonal obstacles.
Output. The visibility graph $\mathcal{G}_{\text{vis}}(S)$.
1. Initialize a graph $\mathcal{G} = (V,E)$ where V is the set of all vertices of the polygons in S and $E = \emptyset$.
2. **for** all vertices $v \in V$
3. **do** $W \leftarrow$ VISIBLEVERTICES(v, S)
4. For every vertex $w \in W$, add the arc (v,w) to E.
5. **return** $\mathcal{G}$

Shortest paths in the plane, complexity

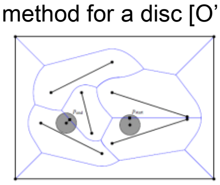
- the visibility-graph algorithm takes $O(n^2 \log n)$ time where n is the number of obstacle vertices
- there are output sensitive algorithms (in the size of the visibility graph)
- near-optimal $O(n \log n)$ algorithm by Hershberger and Suri
- the case of a simple polygon (whose complement is the obstacle)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



- video [Schirra/Rohnert]

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

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
- corridor trees in the complement of molecules

Optimizing a combined measure

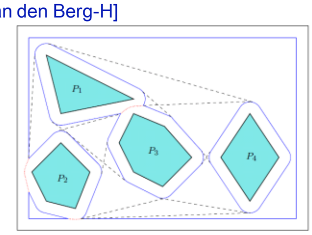
[Wein-van den Berg-H]

$$L^*(C) = \int_\gamma \left(\frac{w_{\max}}{w(t)}\right)^{d-1} dt$$

- 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

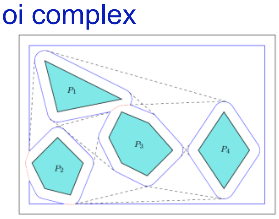
The visibility Voronoi diagram (VVD)

[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

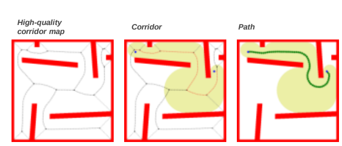
The visibility Voronoi complex



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

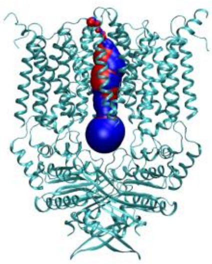
Corridor maps

[Geraerts-Overmars]

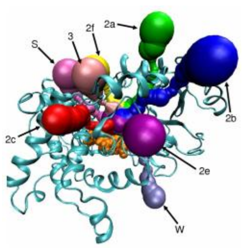


- motivated by motion planning in games
- similar to VVD/VVC, augmenting the VD with clearance information
- instead of providing a single solution path, provides a corridor among static obstacles, where later one can easily maneuver among dynamic obstacles

Corridor trees in the complement of molecules [Yaffe-H]

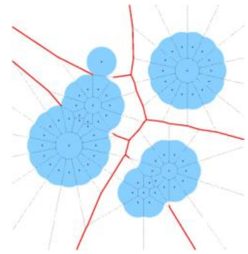


ABC Transporter



Cytochrome P450

Constructing the pathway axis in the complement of molecules, a 2D analog



Corridor trees

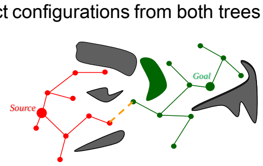
- switch to a one-dimensional network (graph) problem by discarding Voronoi facets
- select a source point
- give weight to edges: $w(e) = \frac{len(e)}{\pi r^2}$
- compute a Corridor Tree using weight minimization
- pathways in the tree are called corridors

Path quality in sampling-based planners

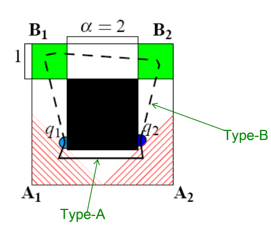
- the typical process: building a roadmap graph and running Dijkstra or similar
- recall our earlier test: does the graph on which we are searching for the best paths contain the best paths?
- path quality can be very low
- example: path length in BiRRT

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

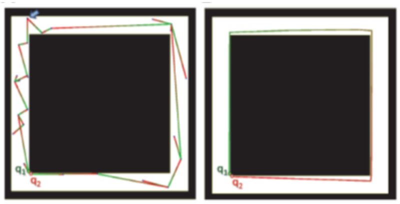


Example (I) – in OOPSMP



- 49.4% of paths are over three times worse than optimal (even after smoothing)
- much larger than the theoretical bound

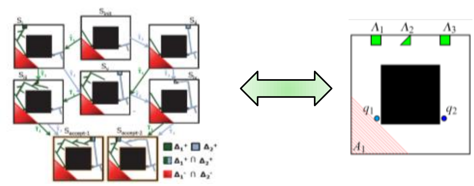
Example (II) – close-by start and goal configurations



- 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]



Path quality in sampling-based planners, our primary motivation: Simulating molecular pathways

robots=proteins, quality=low energy

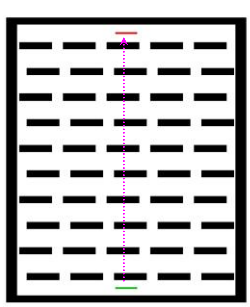
- motion of transmembrane helices
  [Enosh, Fleishman, Ben-Tal, Halperin
  Bioinofrmatics 07] (54 dofs)
- modelling the motion of KcsA potassium channel
  [Enosh, Raveh, Furman-Schueler, Halperin, Ben-Tal
  Biophysical Journal 08] (104 dofs)
- PathRover: prior knowledge + Rosetta
  [Raveh, Enosh, Furman-Schueler, Halperin
  PLoS Computational Biology 09] (200 dofs)
- related work: Amato et al, Kavraki et al, Latombe et al

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

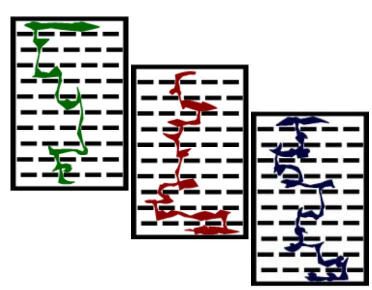
- Short-cutting heuristics ("path smoothing")
- Retraction towards medial axis
  [e.g., Wilmarth et al. '99, Geraerts and Overmars '07]
- Useful Cycles in PRM [Nieuwenhuisen and Overmars '04]
- Biasing tree growth by a cost-function
  [e.g., Urmson and Simmons '03, Ettlin and Bleuler '06, Jaillet et 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 <u>misses</u> 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

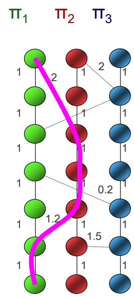
example: move the rod from the bottom to the top of a 2D grid *(rotation + translation)*



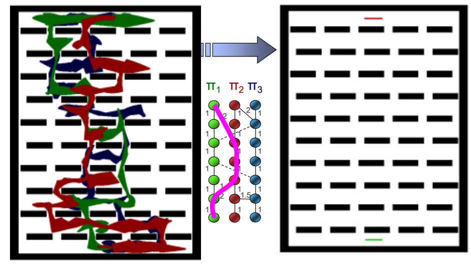
3 randomly generated motion paths



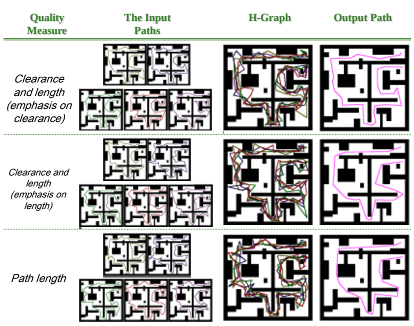
H-Graphs: Hybridizing multiple motion paths
(= looking for shortcuts)



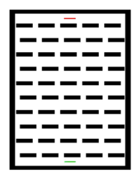
Hybridizing the paths

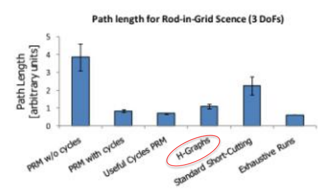


General quality criteria



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)

long runs of PRM same time as total time of HGraphs

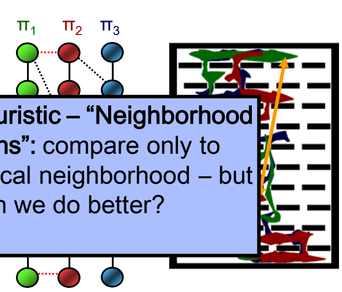
H-Graphs become particularly useful for high-dimensional problems (at least in this example)

Scene adapted from Nieuwenhuisen et al., ICRA 04'

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

in a naïve implementation: $O(n^2)$ potential edges need to be tested

we ass... of paths



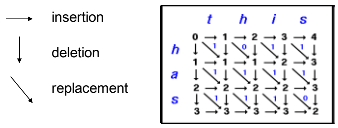
Simple Heuristic – "Neighborhood H-Graphs": compare only to nodes in local neighborhood – but can we do better?

Edit-distance string matching
➔ Linear alignment of motion paths

Comparing “This dog” and “That Dodge” with insertion / deletions / replacement:

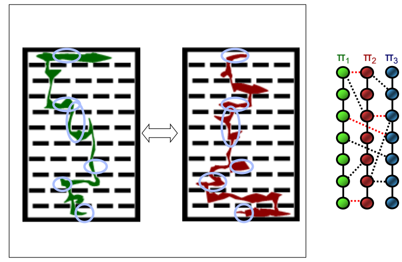
T H I – S   D O – G –
T H A T –   D O D G E

dynamic-programming algorithm:



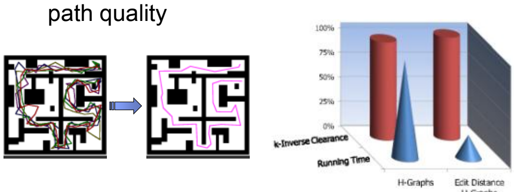
Alignment length is linear

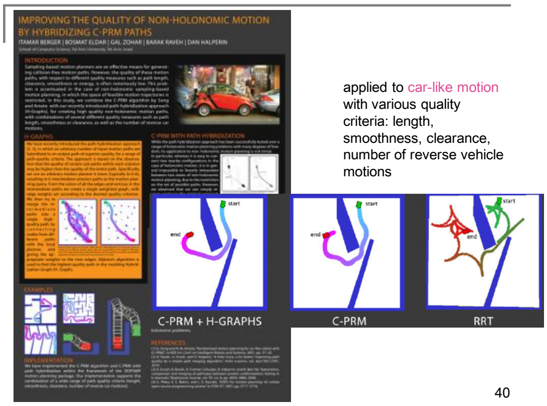
Now testing only O(n) edges along the alignment



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

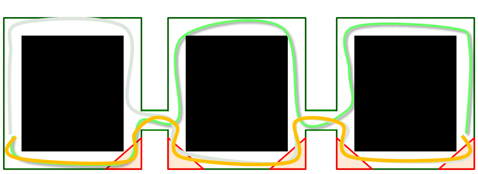




applied to car-like motion with various quality criteria: length, smoothness, clearance, number of reverse vehicle motions

40

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 ot 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
  - *A Little More, A Lot Better: Improving Path Quality by a Path-Merging Algorithm* [Raveh-Enosh-Halperin] IEEE Trans. on Robotics, 2011

42

THE END