Roadmaps

Vertex Visibility Graph

Full visibility graph

Reduced visibility graph, i.e., not including segments that extend into obstacles on either side.

(but keeping endpoints’ roads)

what else might we do?
An alternative roadmap
Voronoi diagrams

These line segments make up the \textbf{Voronoi diagram} for the four points shown here.

Solves the “Post Office Problem”
Voronoi diagrams

These line segments make up the Voronoi diagram for the four points shown here.

Solves the “Post Office Problem”

or, perhaps, more important problems...
Voronoi diagrams

“true” Voronoi diagram
(isolates a set of points)

generalized Voronoi diagram
What is it?
Voronoi diagrams

Let $B = \text{the boundary of } C_{\text{free}}$.

Let $q$ be a point in $C_{\text{free}}$. ( )
Let $B$ = the boundary of $C_{\text{free}}$. Let $q$ be a point in $C_{\text{free}}$.

Define $\text{clearance}(q) = \min \{ | q - p | \}$, for all $p \in B$.
Let $B = \text{the boundary of } C_{\text{free}}$. Let $q$ be a point in $C_{\text{free}}$.

Define $\text{clearance}(q) = \min \{ | q - p | \}$, for all $p \in B$

Define $\text{near}(q) = \{ p \in B \text{ such that } | q - p | = \text{clearance}(q) \}$
Voronoi diagrams

Let $B = \text{the boundary of } C_{\text{free}}$.

Let $q$ be a point in $C_{\text{free}}$.

Define $\text{clearance}(q) = \min \{ |q - p| \}$, for all $p \in B$

Define $\text{near}(q) = \{ p \in B \text{ such that } |q - p| = \text{clearance}(q) \}$

$q$ is in the Voronoi diagram of $C_{\text{free}}$ if $|\text{near}(q)| > 1$

Evaluation

+ maximizes distance from obstacles
+ reduces to graph search
+ can be used in higher-dimensions
- nonoptimal
- real diagrams tend to be noisy

Evaluation

+ maximizes distance from obstacles
+ reduces to graph search
+ can be used in higher-dimensions
- nonoptimal
- real diagrams tend to be noisy
Generalized Voronoi Graph (GVG)
Generalized Voronoi Graph (GVG)
Generalized Voronoi Graph (GVG)

• Access GVG
Generalized Voronoi Graph (GVG)

- Access GVG
- Follow Edge
Generalized Voronoi Graph (GVG)

- Access GVG
- Home to the MeetPoint
- Follow Edge

Free Space with Topological Map (GVG)
Generalized Voronoi Graph (GVG)

• Access GVG  
• Home to the MeetPoint
• Follow Edge  
• Select Edge
GVG construction using sonar

- Nomadic Scout
- Sonar (GVG navigation)
- Camera with omni-directional mirror (feature detection)
- Onboard 1.2 GHz processor
GVG construction using sonar
GVG construction using sonar
Slammer in Action
What is the Voronoi Graph?
Voronoi applications

A retraction of a 3D object == "medial surface"

Skeletonizations resulting from constant-speed curve evolution

in 2D, it’s called a medial axis
skeleton $\rightarrow$ shape

curve evolution
where wavefronts collide
centers of maximal disks

again reduces a 2d (or higher) problem to a question about graphs...
skeleton $\rightarrow$ shape

curve evolution  
where wavefronts collide  
centers of maximal disks

again reduces a 2d (or higher) problem to a question about graphs...
Problems

The skeleton is sensitive to small changes in the object’s boundary.

- graph isomorphism (and lots of other graph questions) : NP-complete
Roadmap problems

If an obstacle decides to roll away...  (or wasn’t there to begin with)

recomputing in less than $O(N^2)$ time?
Experiments at CSA

- Autonomous Over-the-Horizon Navigation
CSA Experiments
CSA Experiments
Practical Considerations
Path Planning

Potential Field methods

- compute a repulsive force away from obstacles
Local techniques

Potential Field methods

- compute a repulsive force away from obstacles
- compute an attractive force toward the goal
Local techniques

Potential Field methods

- compute a repulsive force away from obstacles
- compute an attractive force toward the goal

→ let the sum of the forces control the robot
Local techniques

Potential Field methods

- compute a repulsive force away from obstacles
- compute an attractive force toward the goal

→ let the sum of the forces control the robot

To a large extent, this is computable from sensor readings
Sensor Based Calculations
Major Problem?
Local Minima!
Simulated Annealing

- Every so often add some random force
Known Map

Brushfire Transform
The Wavefront Planner: Setup

![Wavefront Planner Setup Diagram]
The Wavefront in Action (Part 1)

- Starting with the goal, set all adjacent cells with “0” to the current cell + 1
  - 4-Point Connectivity or 8-Point Connectivity?
  - Your Choice. We’ll use 8-Point Connectivity in our example
The Wavefront in Action (Part 2)

• Now repeat with the modified cells
  – This will be repeated until no 0’s are adjacent to cells with values $\geq 2$
• 0’s will only remain when regions are unreachable
The Wavefront in Action (Part 3)

• Repeat
The Wavefront in Action (Part 3)

- Repeat
The Wavefront in Action (Part 3)

- Until Done
  - 0’s would only remain in the unreachable areas
The Wavefront in Action

- To find the shortest path, according to your metric, simply always move toward a cell with a lower number
  - The numbers generated by the Wavefront planner are roughly proportional to their distance from the goal

Two possible shortest paths shown
Introduction to Mapping

• What the world looks like?
• Knowledge representation
  – Robotics, AI, Vision
• Who is the end-user?
  – Human or Machine
• Ease of Path Planning
• Uncertainty!
Simultaneous Localization And Mapping

**SLAM** is the process of building a map of an environment while, at the same time, using that map to maintain the location of the robot.

- **Problems for SLAM in large scale environments:**
  - Controlling growth of uncertainty and complexity
  - Achieving autonomous exploration
Consider this Environment:
Three Basic Map Types

Topological:
Collection of nodes and their interconnections

Grid-Based:
Collection of discretized obstacle/free-space pixels

Feature-Based:
Collection of landmark locations and correlated uncertainty
# Three Basic Map Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Grid-Based</th>
<th>Feature-Based</th>
<th>Topological</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construction</td>
<td>Occupancy grids</td>
<td>Kalman Filter</td>
<td>Navigation control laws</td>
</tr>
<tr>
<td>Complexity</td>
<td>Grid size and resolution</td>
<td>Landmark covariance ($N^3$)</td>
<td>Minimal complexity</td>
</tr>
<tr>
<td>Obstacles</td>
<td>Discretized obstacles</td>
<td>Only structured obstacles</td>
<td>GVG defined by the safest path</td>
</tr>
<tr>
<td>Localization</td>
<td>Discrete localization</td>
<td>Arbitrary localization</td>
<td>Localize to nodes</td>
</tr>
<tr>
<td>Exploration</td>
<td>Frontier-based exploration</td>
<td>No inherent exploration</td>
<td>Graph exploration</td>
</tr>
</tbody>
</table>
# Other Maps

<table>
<thead>
<tr>
<th></th>
<th>Appearance Based</th>
<th>Geometry Based</th>
<th>Mesh Based</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Construction</strong></td>
<td>Images</td>
<td>Lines, planes, etc</td>
<td>Mesh</td>
</tr>
<tr>
<td><strong>Path Planning</strong></td>
<td>N/A</td>
<td>Geometry based</td>
<td>Graph based</td>
</tr>
<tr>
<td><strong>Localization</strong></td>
<td>Arbitrary localization</td>
<td>Arbitrary localization</td>
<td>Arbitrary localization</td>
</tr>
</tbody>
</table>