Coverage
Coverage

- A task performed quite often in everyday life:
  - Cleaning
  - Painting
  - Plowing/Sowing
  - Tile setting
  - etc.
Motivation

Humanitarian Demining
Motivation
Lawn Mowing
Motivation
Vacuum Cleaning
Robotic Coverage

- More than 5 million Roombas sold!
- Automated Car Painting
Roomba Costumes

From: http://www.myroombud.com/
Coverage

• First Distinction
  – Deterministic
  – Random
  • Demining
  • Vacuum Cleaning

• Second Distinction
  – Complete
  – No Guarantee

• Third Distinction
  – Known Environment
  – Unknown Environment
Non-Deterministic Coverage

- Complete Random Walk
- Ant Robotics
  - Leave trail
  - Bias the behavior towards or away from the trails

Andrew Russell, Monash University, Australia

Ant Robotics: I. Wagner, IBM & Technion

S. Koenig Ant Robotics, terrain coverage
Deterministic Coverage

- Complete Algorithm
- Guarantees Complete Coverage
Cell-Decomposition Methods

Two families of methods:

- **Exact cell decomposition**
  The free space $F$ is represented by a collection of non-overlapping cells whose union is exactly $F$.

**Examples:** trapezoidal and cylindrical decompositions
BOUSTROPHEDON CELLULAR DECOMPOSITION

The way of the Ox!
Single Robot Coverage

- Deterministic algorithm
- Guarantee of completeness
- Sensor based
- Unknown Environment

Single Robot Coverage

Direction of Coverage

Cellular Decomposition

Reeb graph
Vertices: Critical Points
Edges: Cells
Critical Points

- There are four types of critical points:
  - Forward Concave critical point
  - Reverse Concave critical point
  - Reverse Convex critical point
  - Forward Convex critical point
Optimal Coverage

• Find an order for traversing the Reeb graph such that the robot would not go through a cell more times than necessary

Solution

• Use the Chinese Postman Problem
Chinese Postman Problem

• The Chinese postman problem (CPP), is to find a shortest closed path that visits every edge of a (connected) undirected graph. When the graph has an Eulerian circuit (a closed walk that covers every edge once), that circuit is an optimal solution.

Offline Analysis Algorithm

<table>
<thead>
<tr>
<th>Offline Analysis</th>
<th>Online Trajectory Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direction Alignment (Optional)</td>
<td>Per-Cell Planner</td>
</tr>
<tr>
<td>Cellular Decomp.</td>
<td>Non-Holonomic Controller</td>
</tr>
<tr>
<td>Chinese Postman Problem</td>
<td></td>
</tr>
</tbody>
</table>
Offline Analysis Algorithm

- **Input:** binary map separating obstacle from free space
- **Boustrophedon Cellular Decomposition (BCD)**

![Diagram showing the process of Offline Analysis and Online Trajectory Control]

- **Direction Alignment (Optional)**
- **Cellular Decomp.**
- **Chinese Postman Problem**
- **Per-Cell Planner**
- **Non-Holonomic Controller**

- Intersections = vertices
- Cells = edges
Offline Analysis Algorithm (cont.)

- **Chinese Postman Problem**
  - Eulerian circuit, i.e. *single* traversal through all cells (edges)
• Seed Spreader: piecewise linear sweep lines

• Footprint width
Coverage Direction Alignment

- Static alignment methods

- Alignment with average wind heading (pre-flight)

CSCE 774: Robotic Systems
Non-Holonomic Robot Controller

- Turning strategies

- Direction Alignment (Optional) → Cellular Decomp. → Chinese Postman Problem → Per-Cell Planner → Non-Holonomic Controller

- Greedy Waypoint Controller

- Curlicue Controller
Chinese Postman Problem

- The solution of the CPP guarantees that no edge is doubled more than once
- That means some cells have to be traversed twice
- Cells that have to be traversed/covered are divided in half
Double Coverage of a Single Cell

• By dividing the cell diagonally we control the beginning and end of the coverage
Double Coverage of a Single Cell

• By dividing the cell diagonally we control the beginning and end of the coverage
Optimal Coverage Algorithm

- Given a known environment:
  - Calculate the Boustrophedon decomposition
  - Construct the Reeb graph
  - Use the Reeb graph as input to the Chinese Postman Problem (CPP)
  - Use the solution of the CPP to find a minimum cost cycle traversing every edge of the Reeb graph
  - For every doubled edge divide the corresponding cell in half
  - Traverse the Reeb graph by covering each cell in order
Traversal order of the Reeb graph
Example
Example: Boustrophedon Decomposition
Example: Reeb Graph
Example: CPP solution
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example 2
Example 2 Boustrophedon Decomp.
Example 2
Example 2
Example 2
Example 2
Example 2
Example 2
Example 2
Example 2
Example 2
Example 2
Example 2
Example 2
Example 2
Example 2
Example 2
Example 2
Example 2
Example 2
UAV-Optimal Coverage
UAV-Optimal Coverage

• UAVs non-holonomic constraints require special trajectory planning

• 120 Km of flight during coverage
Image Mosaic
Multi-UAV
Video at ICRA 2011

Complete Optimal Terrain Coverage using an Unmanned Aerial Vehicle

Anqi Xu
Chatavut Viriyasuthee
Ioannis Rekleitis

McGill
Coverage of Known Worlds

Decompose the free space into simple cells and represent the connectivity of the free space by the adjacency graph of these cells.
Trapezoidal decomposition
Spatial decompositions

• Dividing free space into pieces and using those...

![Diagram of spatial decomposition](image-url)
Spatial decompositions

- Dividing free space into pieces and using those...

Exact cell decomposition

sweepline algorithm

Running time?
Spatial decompositions

- Dividing free space into pieces and using those...

Exact cell decomposition

sweepline algorithm

Running time?

O( N log(N) )

Path?
Spatial decompositions

- Dividing free space into pieces and using those...

Exact cell decomposition
sweepline algorithm

Running time?
O( N log(N) )

Path?
via centroids
Spatial decompositions

- Dividing free space into pieces and using those...

Exact cell decomposition
sweepline algorithm

Running time?
$O(N \log(N))$

Path?
via centroids + edge midpoints
Spatial decompositions

- Dividing free space into pieces and using those...

Exact cell decomposition
  - sweepline algorithm

Running time?
  - $O( N \log(N) )$

Path?
  - via centroids + edge midpoints + graph search

Why?
  - why else?
Optimality

- Obtaining the minimum number of convex cells is NP-complete.

Trapezoidal decomposition is exact and complete, but not optimal -- even among convex subdivisions.
Cell-Decomposition Methods

Two families of methods:

- **Exact cell decomposition**
- **Approximate cell decomposition**

F is represented by a collection of non-overlapping cells whose union is contained in F

**Examples:** quadtree, octree, $2^n$-tree
further decomposing...

- Approximate cell decomposition

Quadtree: recursively subdivides each *mixed* obstacle/free (sub)region into four quarters...
further decomposing...

• Approximate cell decomposition

Quadtree: recursively subdivides each *mixed* obstacle/free (sub)region into four quarters...
further decomposing...

- Approximate cell decomposition

Quadtree: recursively subdivides each *mixed* obstacle/free (sub)region into four quarters...
• Approximate cell decomposition

Again, use a graph-search algorithm to find a path from the start to goal.

Quadtree

is this a complete path-planning algorithm? i.e., does it find a path when one exists?
Octree Decomposition

![Octree Diagram]

- **EMPTY cell**
- **MIXED cell**
- **FULL cell**