COMP 790-058:

Fall 2007

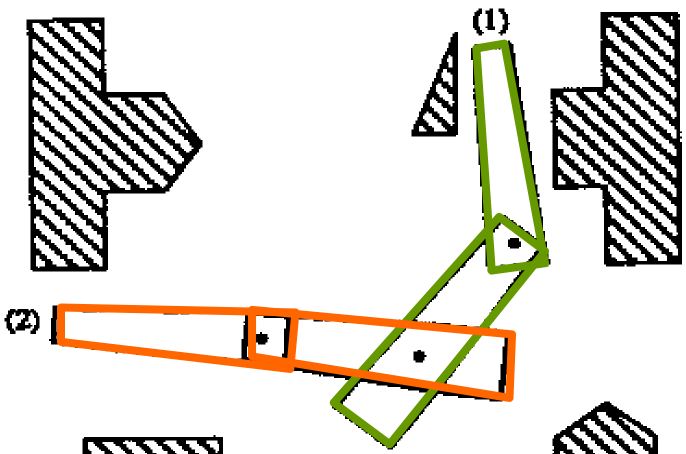
(Based on slides from J. Latombe @ Stanford & David Hsu @ Singapore)

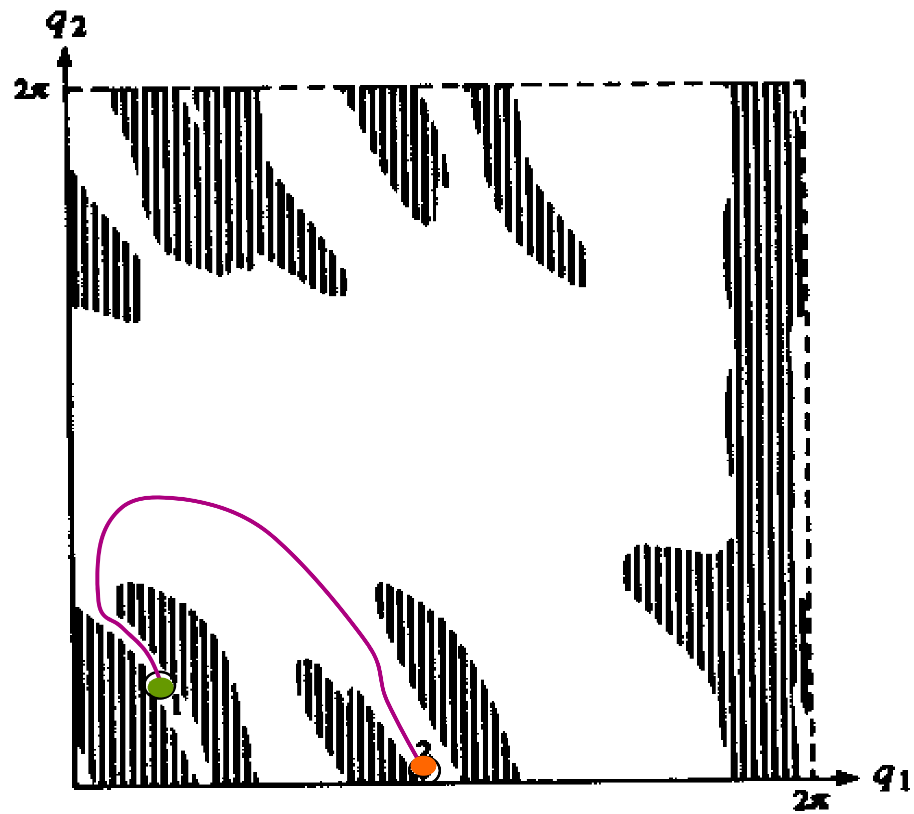
Path Planning for a Point Robot

Main Concepts

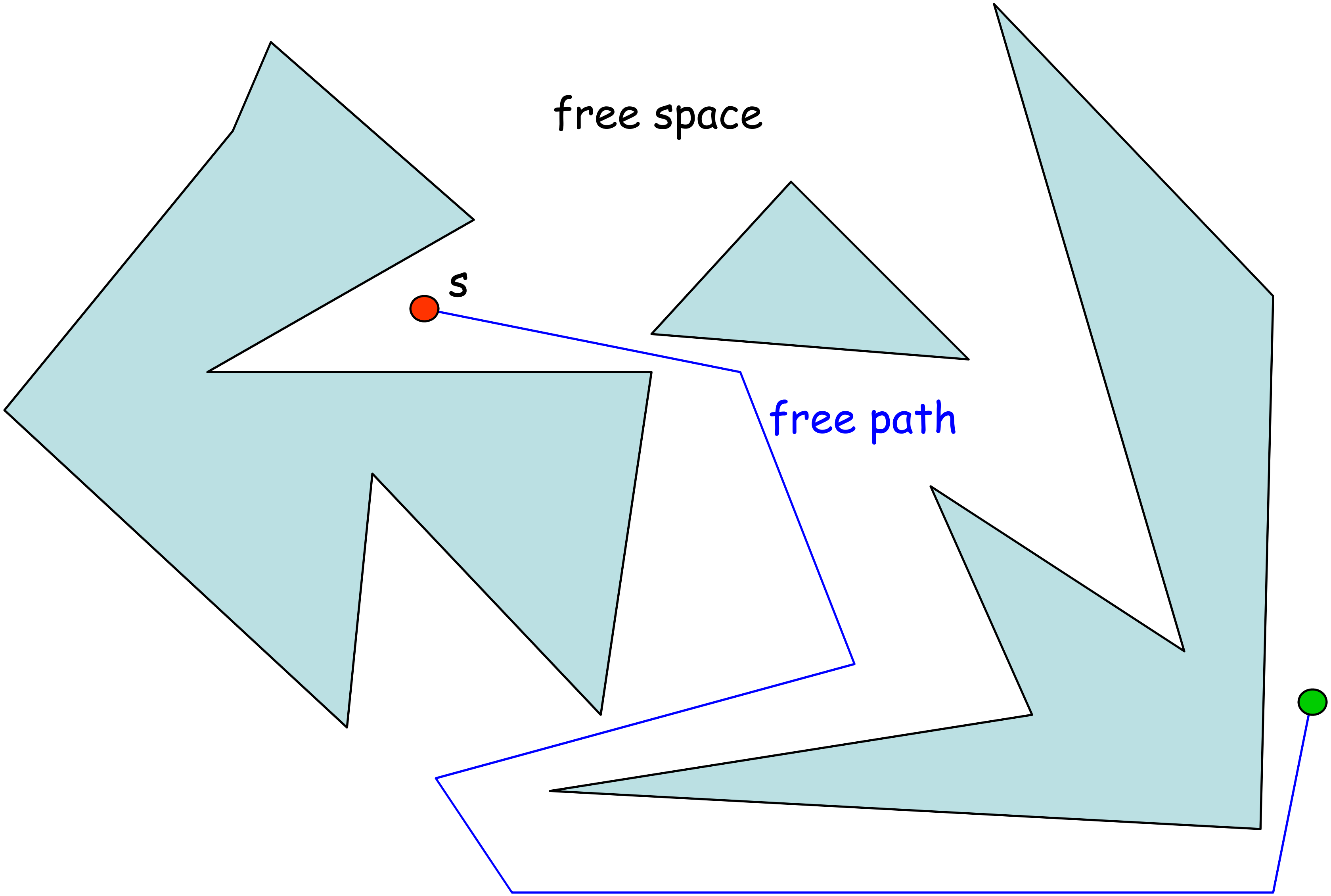
- Reduction to point robot
- Search problem
- Graph search
- Configuration spaces

Configuration Space: Tool to Map a Robot to a Point

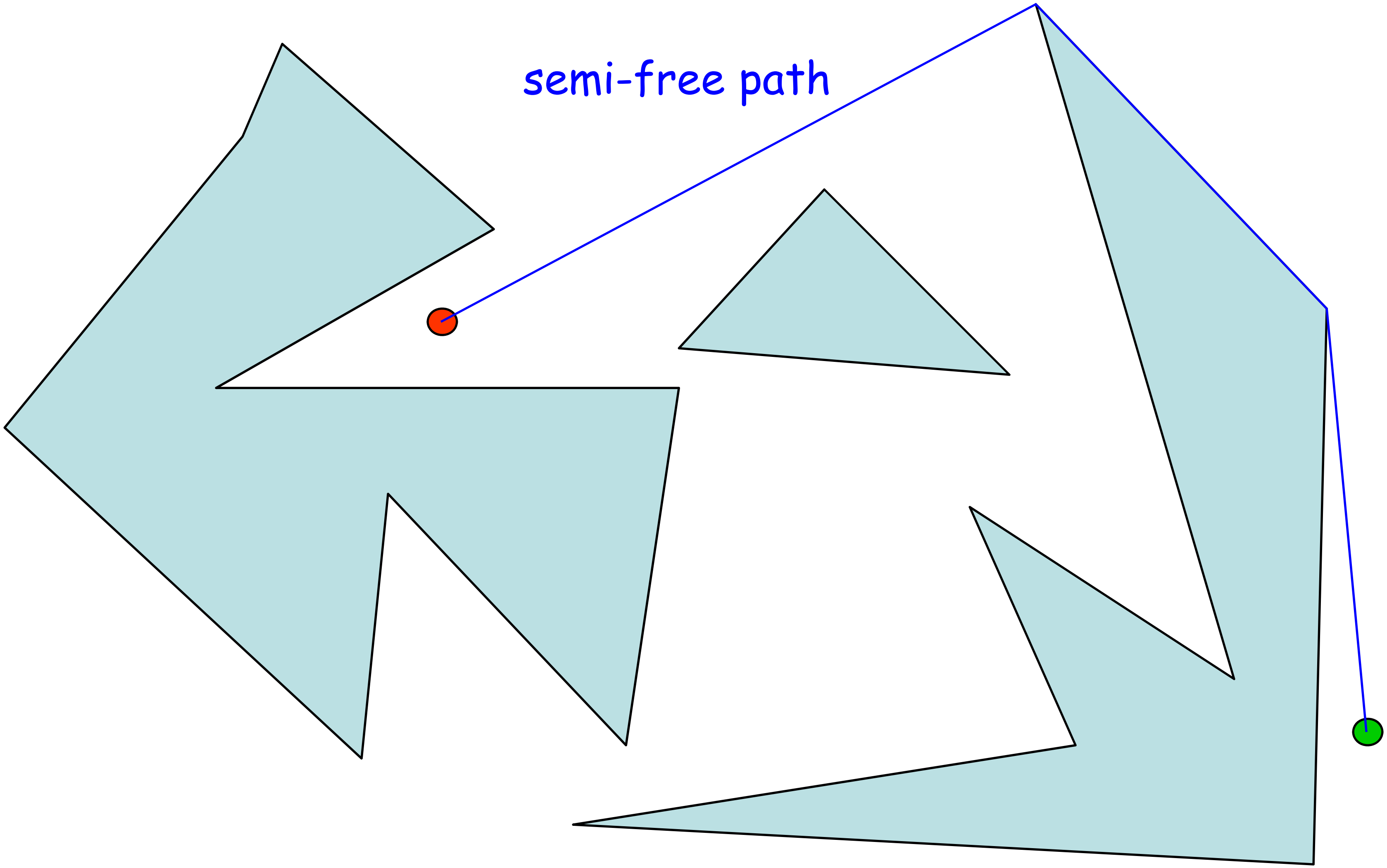




Problem



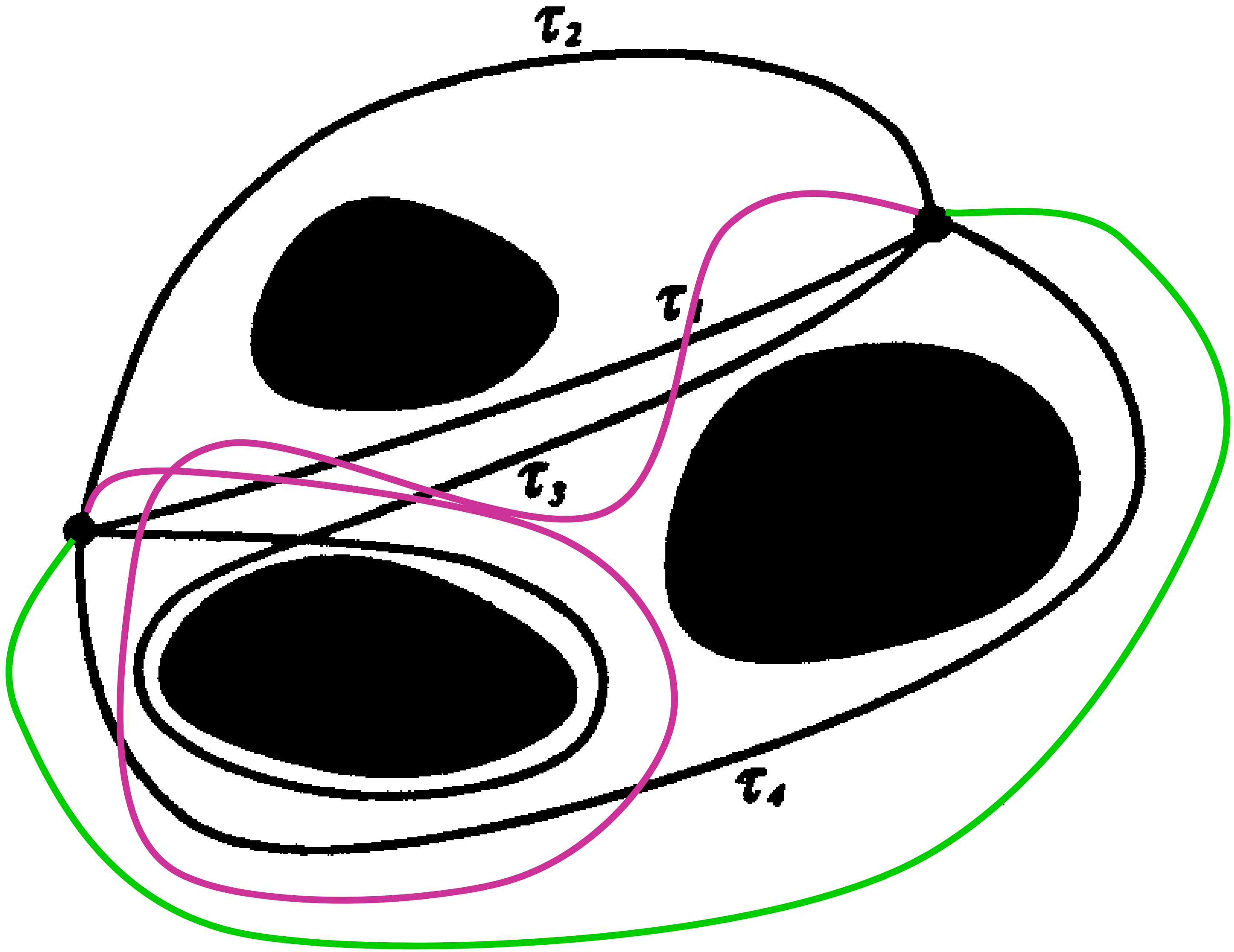
Problem



Types of Path Constraints

- Local constraints:
  lie in free space
- Differential constraints:
  have bounded curvature
- Global constraints:
  have minimal length

Homotopy of Free Paths



Motion-Planning Framework

Continuous representation

↓

Discretization

↓

Graph searching

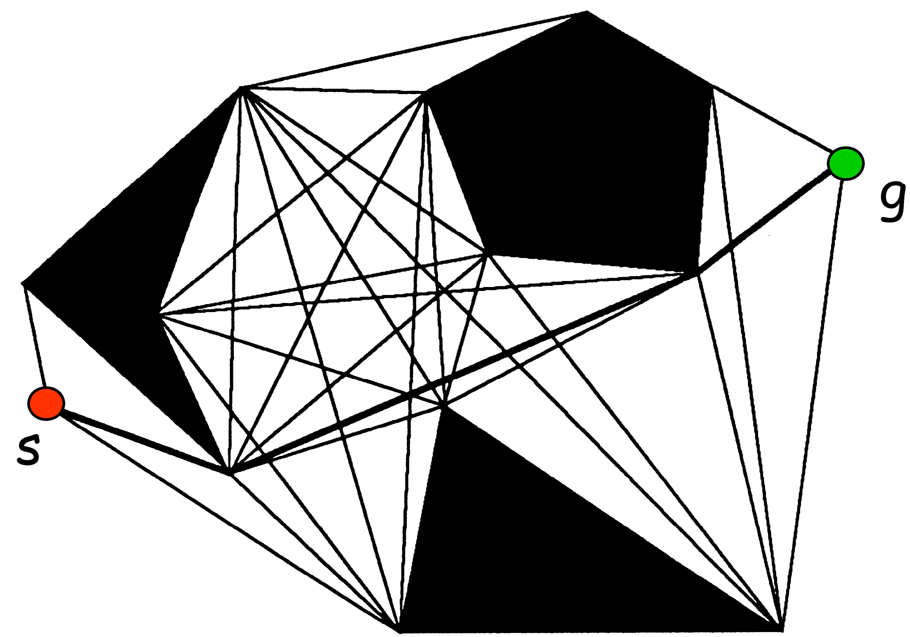
(blind, best-first, A*)

Path-Planning Approaches

1. **Roadmap**
   Represent the connectivity of the free space by a network of 1-D curves
2. **Cell decomposition**
   Decompose the free space into simple cells and represent the connectivity of the free space by the adjacency graph of these cells
3. **Potential field**
   Define a function over the free space that has a global minimum at the goal configuration and follow its steepest descent

Roadmap Methods

- **Visibility graph**
  Introduced in the Shakey project at SRI in the late 60s. Can produce shortest paths in 2-D configuration spaces



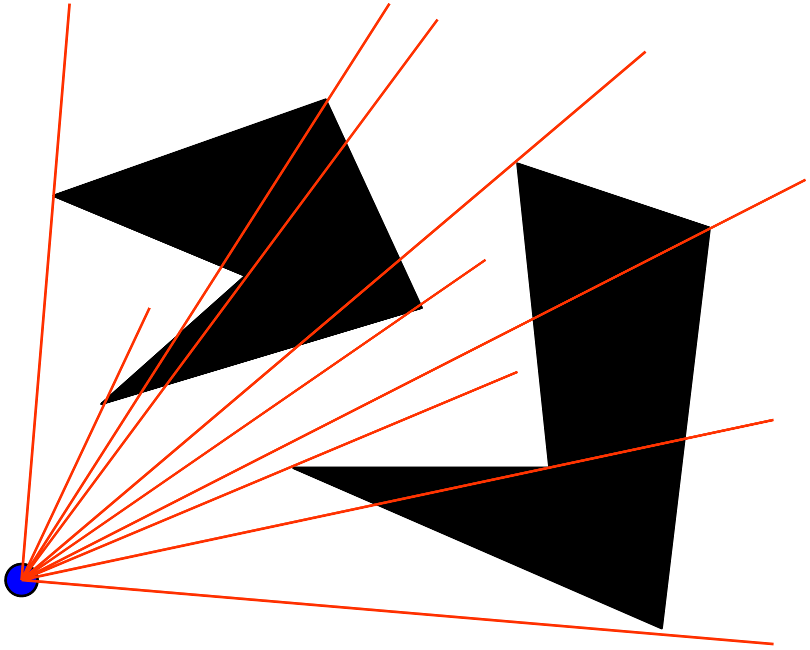
Simple Algorithm

```
1.  Install all obstacles vertices in VG, plus the start
    and goal positions
2.  For every pair of nodes u, v in VG
3.      If segment(u,v) is an obstacle edge then
4.          insert (u,v) into VG
5.      else
6.          for every obstacle edge e
7.              if segment(u,v) intersects e
8.                  then goto 2
9.          insert (u,v) into VG
10. Search VG using A*
```

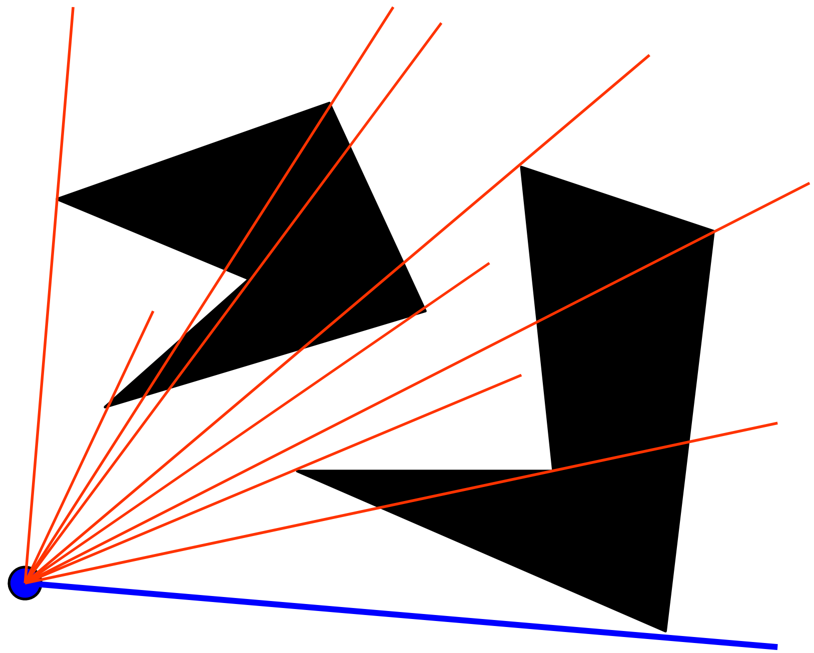
Complexity

- Simple algorithm: $O(n^3)$ time
- Rotational sweep: $O(n^2 \log n)$
- Optimal algorithm: $O(n^2)$
- Space: $O(n^2)$

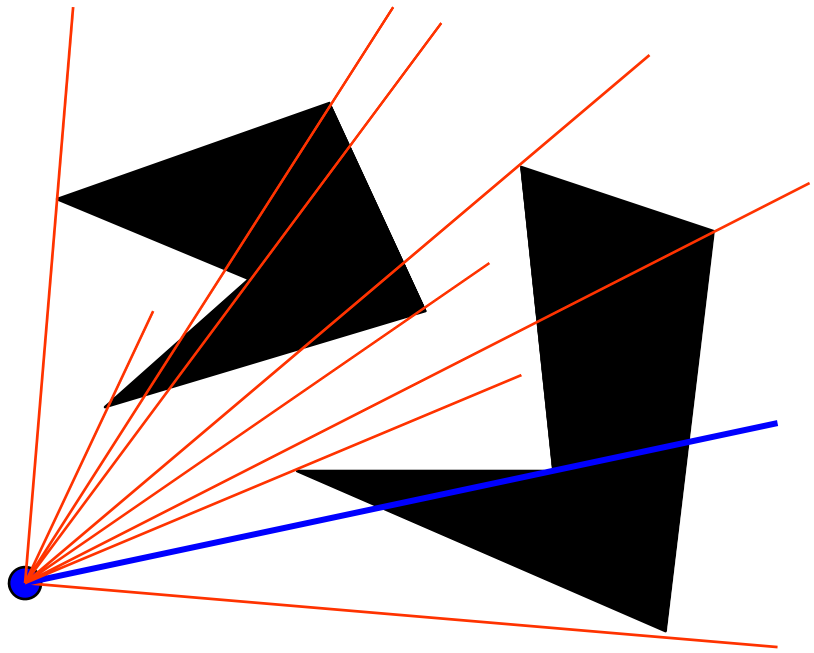
Rotational Sweep



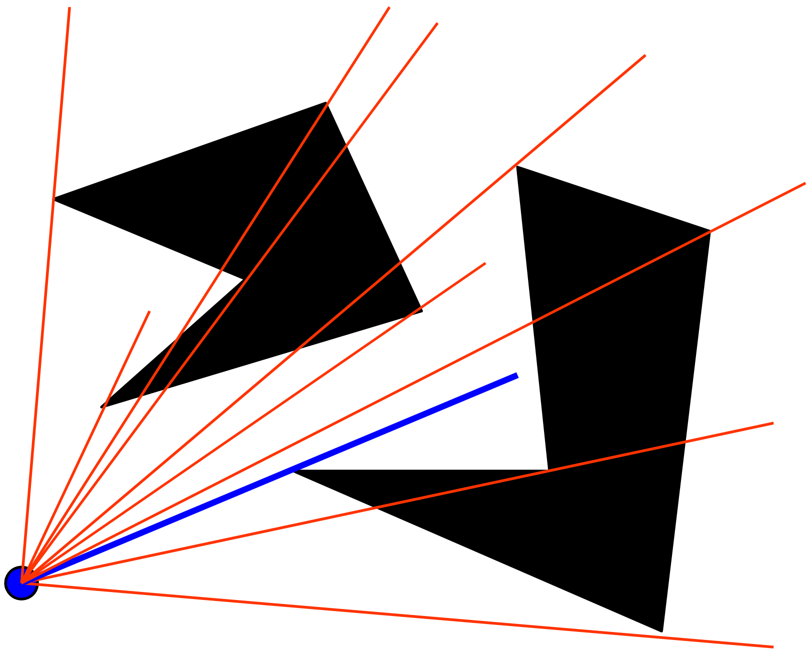
Rotational Sweep



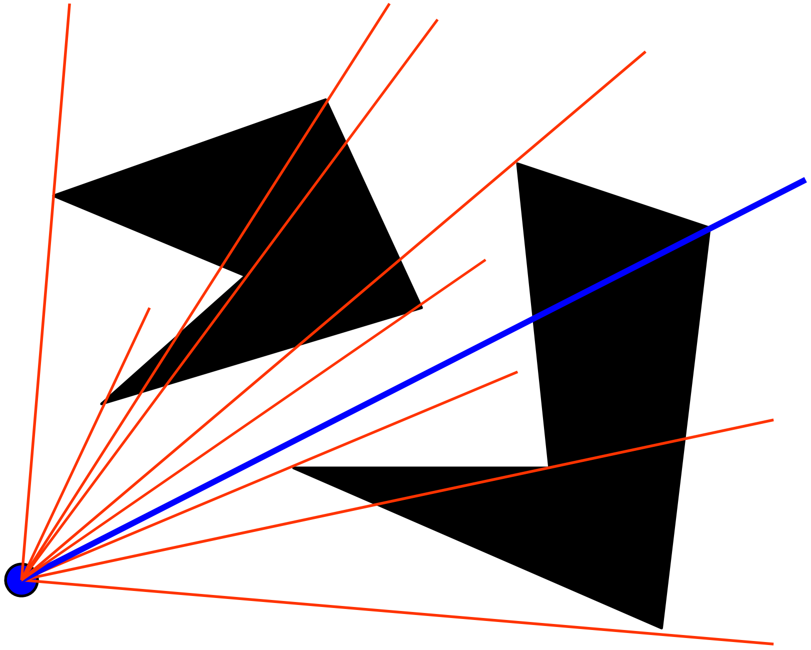
Rotational Sweep



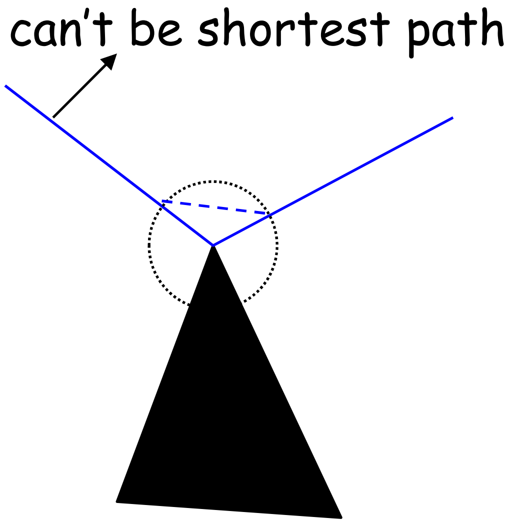
Rotational Sweep

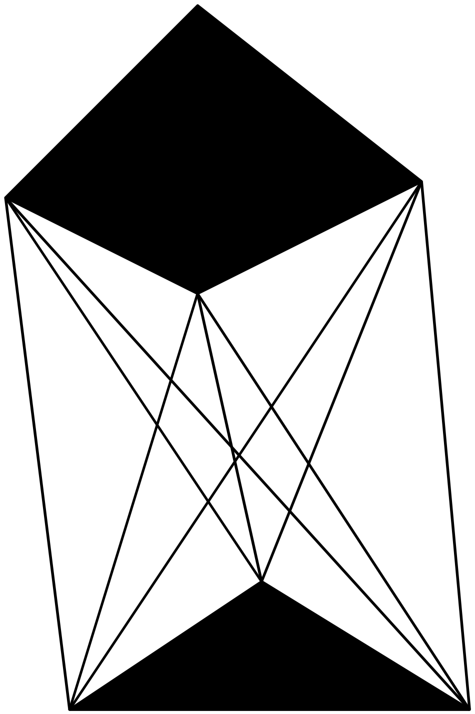


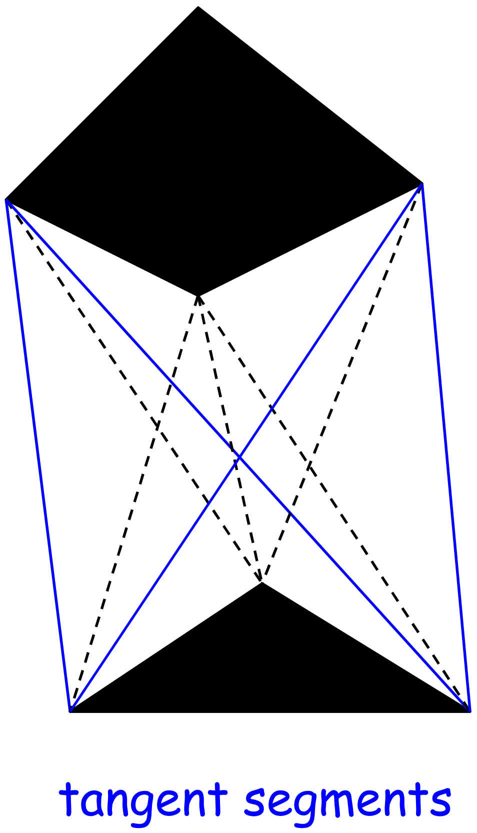
Rotational Sweep



Reduced Visibility Graph

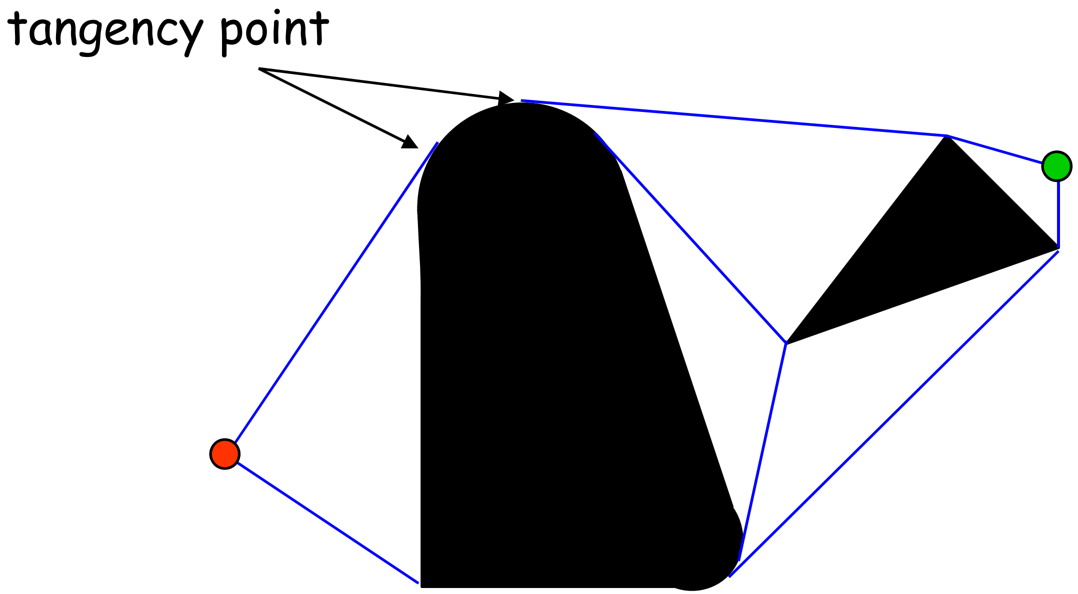




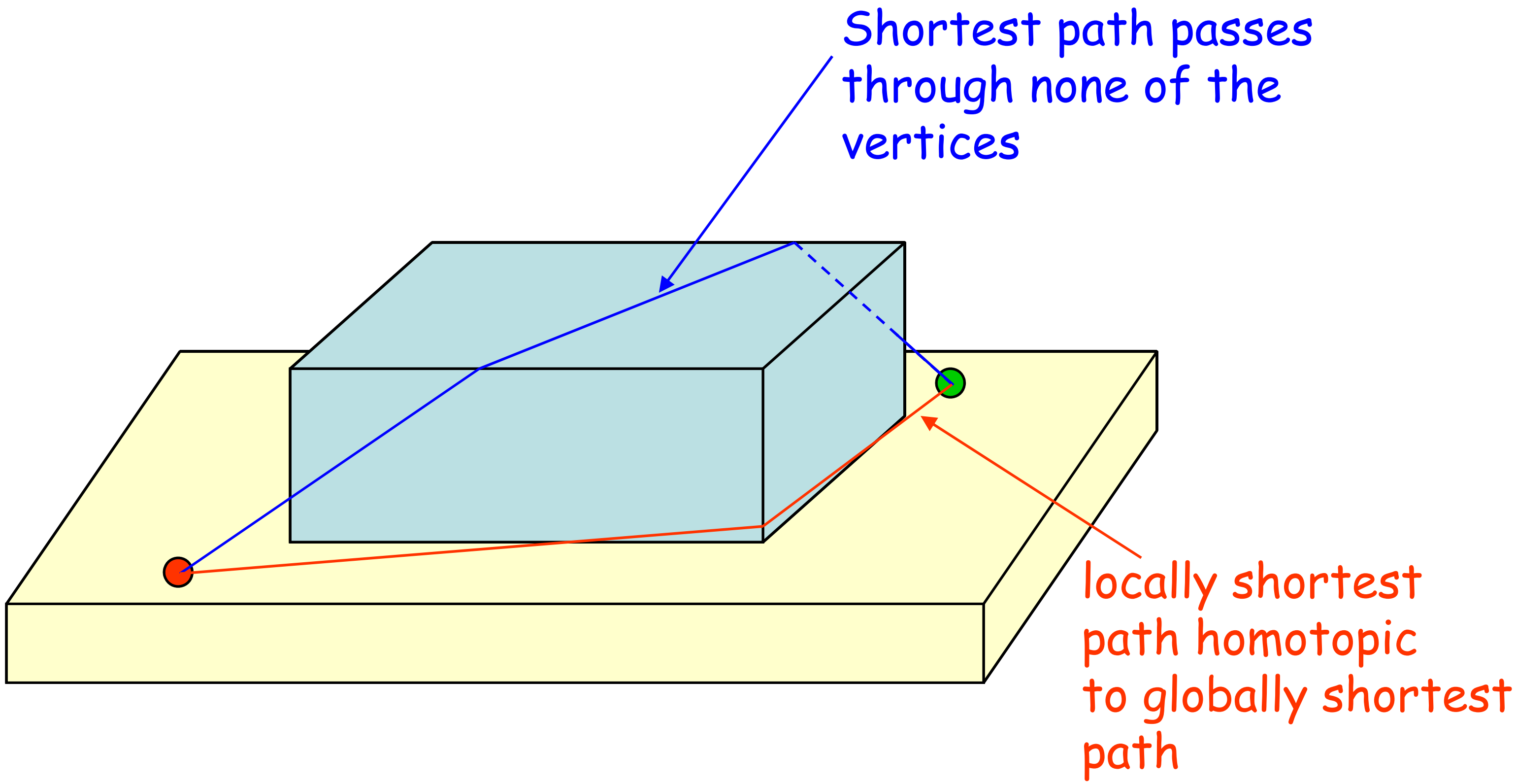


→ Eliminate concave obstacle vertices

Generalized (Reduced) Visibility Graph



Three-Dimensional Space

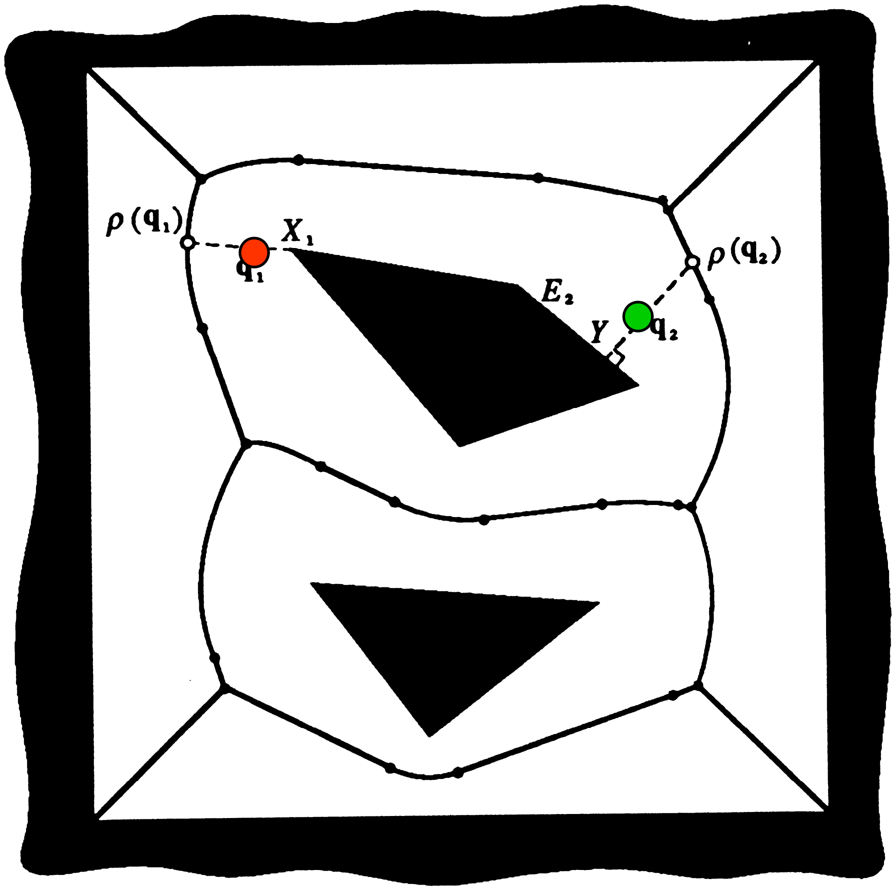


Computing the shortest collision-free path in a polyhedral space is NP-hard

Roadmap Methods

- **Voronoi diagram**
  Introduced by Computational Geometry researchers. Generate paths that maximizes clearance.

  O(n log n) time
  O(n) space



Roadmap Methods

- **Visibility graph**
- **Voronoi diagram**
- **Silhouette**

  First complete general method that applies to spaces of any dimension and is singly exponential in # of dimensions [Canny, 87]
- **Probabilistic roadmaps**

Path-Planning Approaches

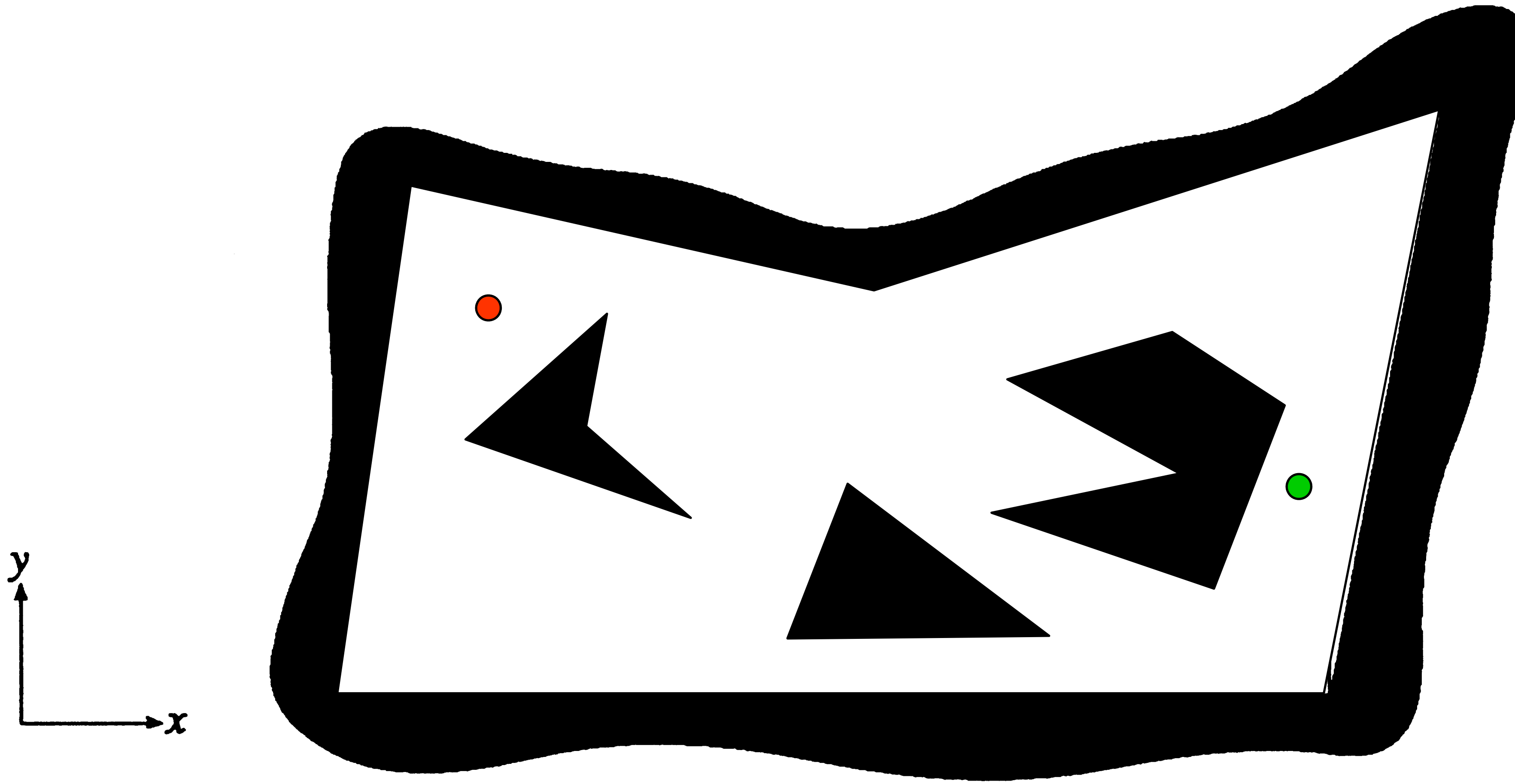
1. **Roadmap**
   Represent the connectivity of the free space by a network of 1-D curves
2. **Cell decomposition**
   Decompose the free space into simple cells and represent the connectivity of the free space by the adjacency graph of these cells
3. **Potential field**
   Define a function over the free space that has a global minimum at the goal configuration and follow its steepest descent

Cell-Decomposition Methods

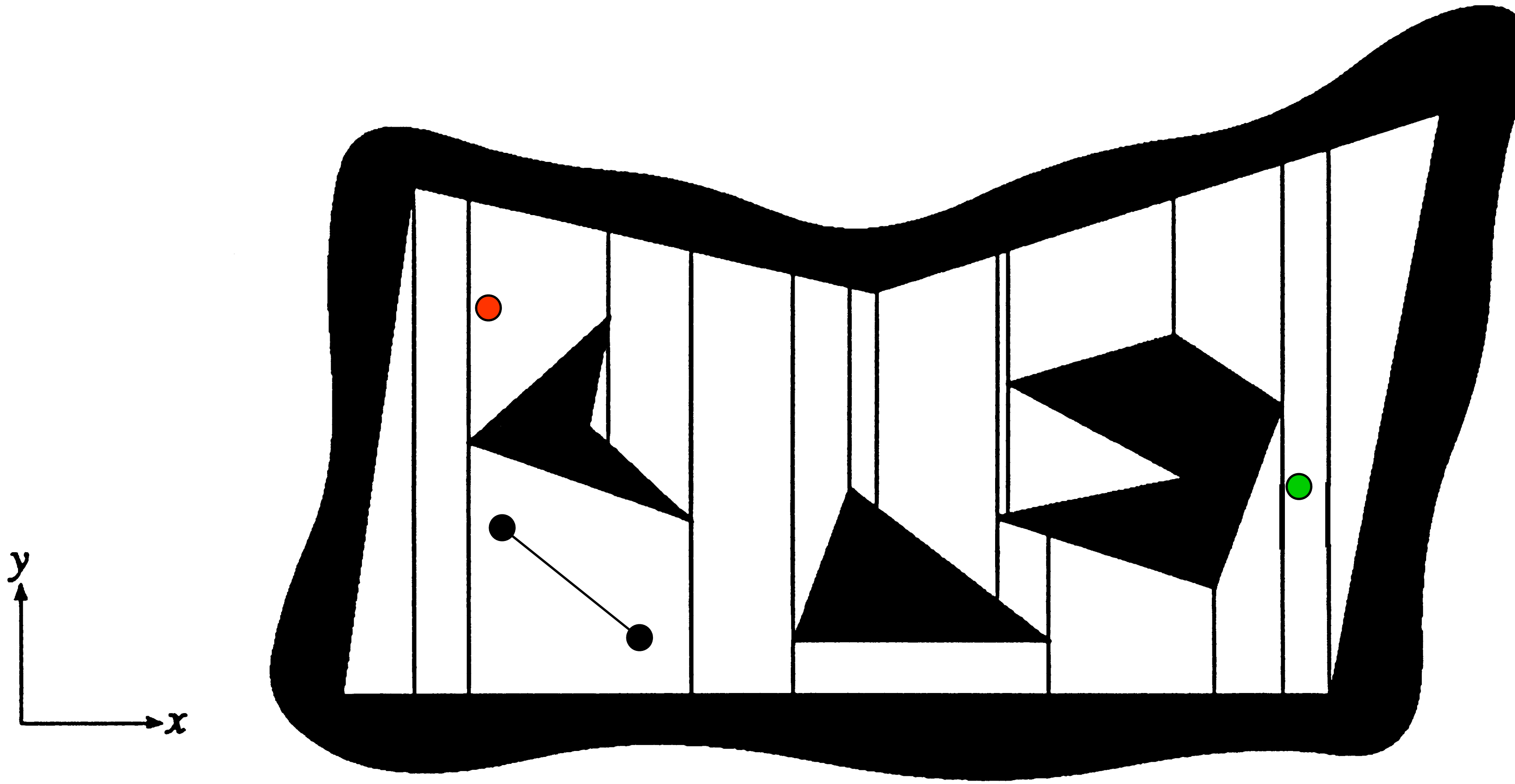
Two classes of methods:

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

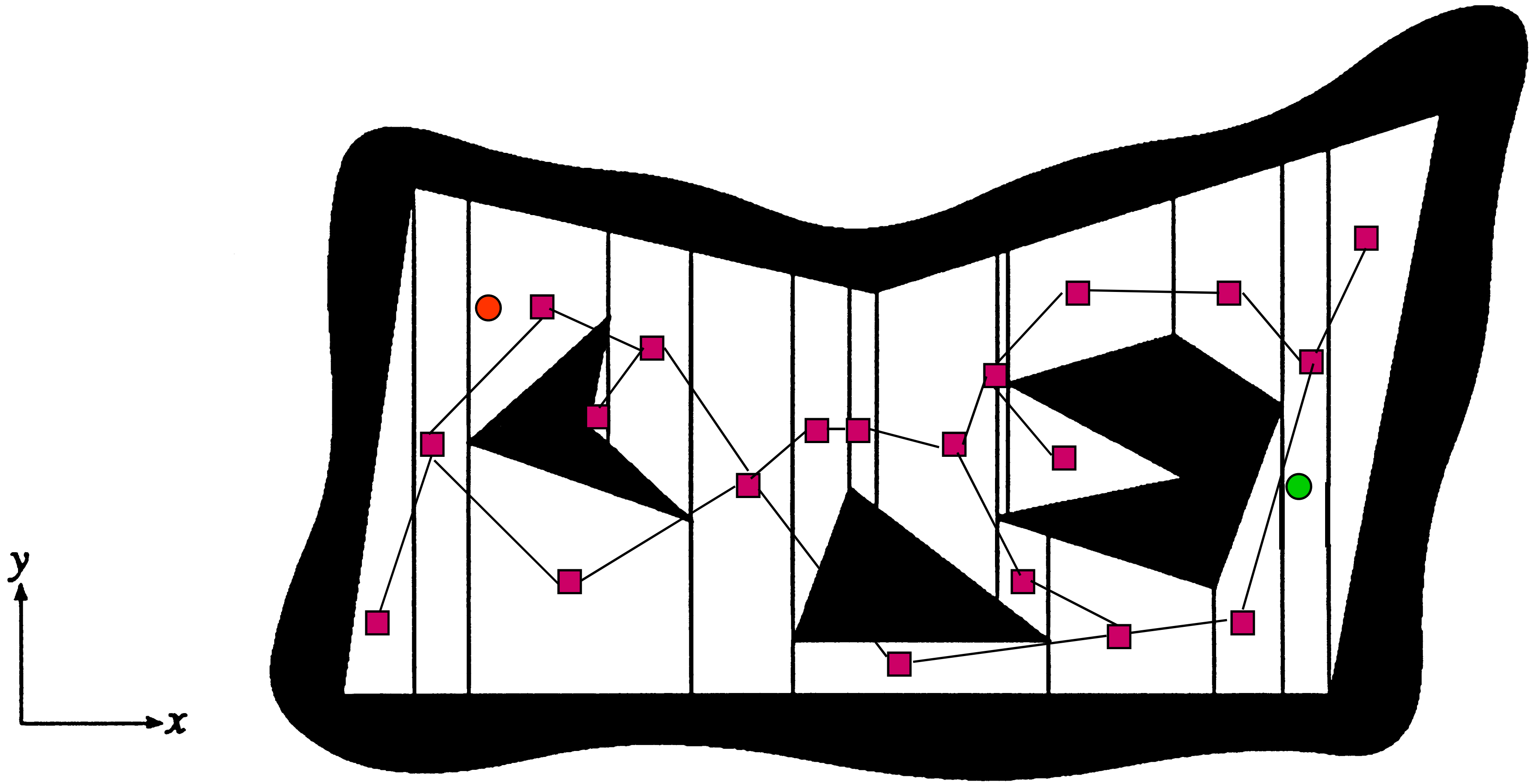
Trapezoidal decomposition



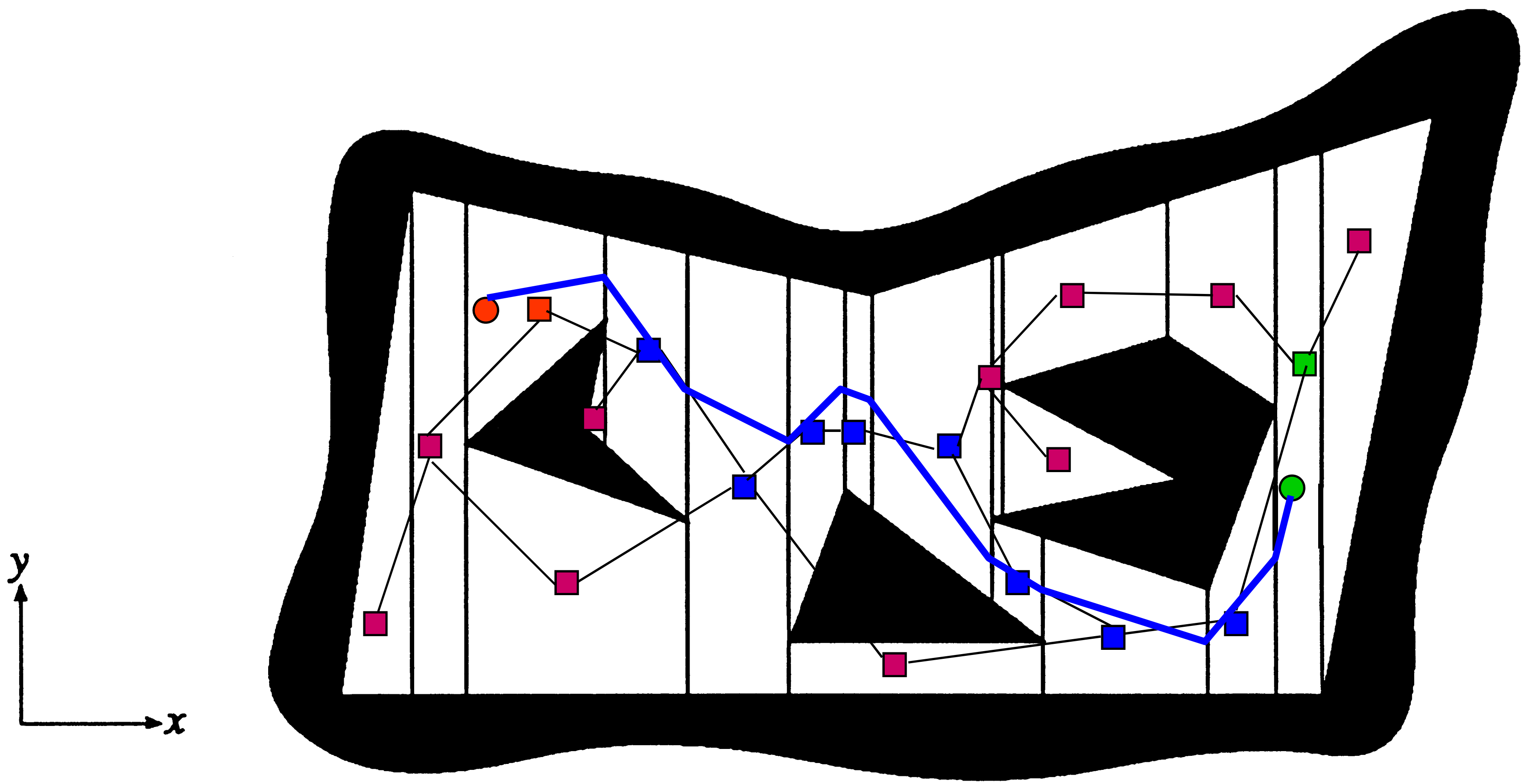
Trapezoidal decomposition



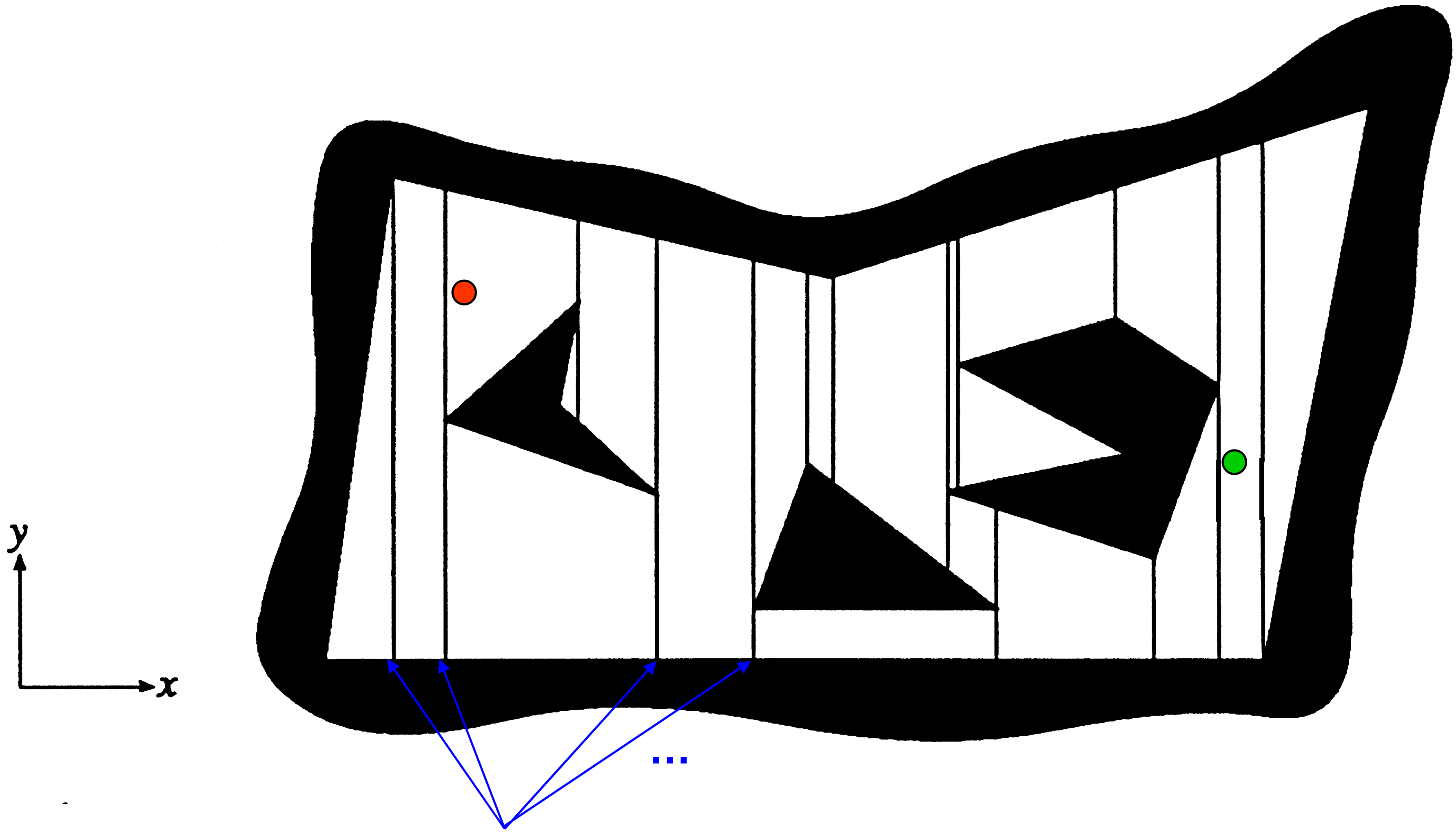
Trapezoidal decomposition



Trapezoidal decomposition

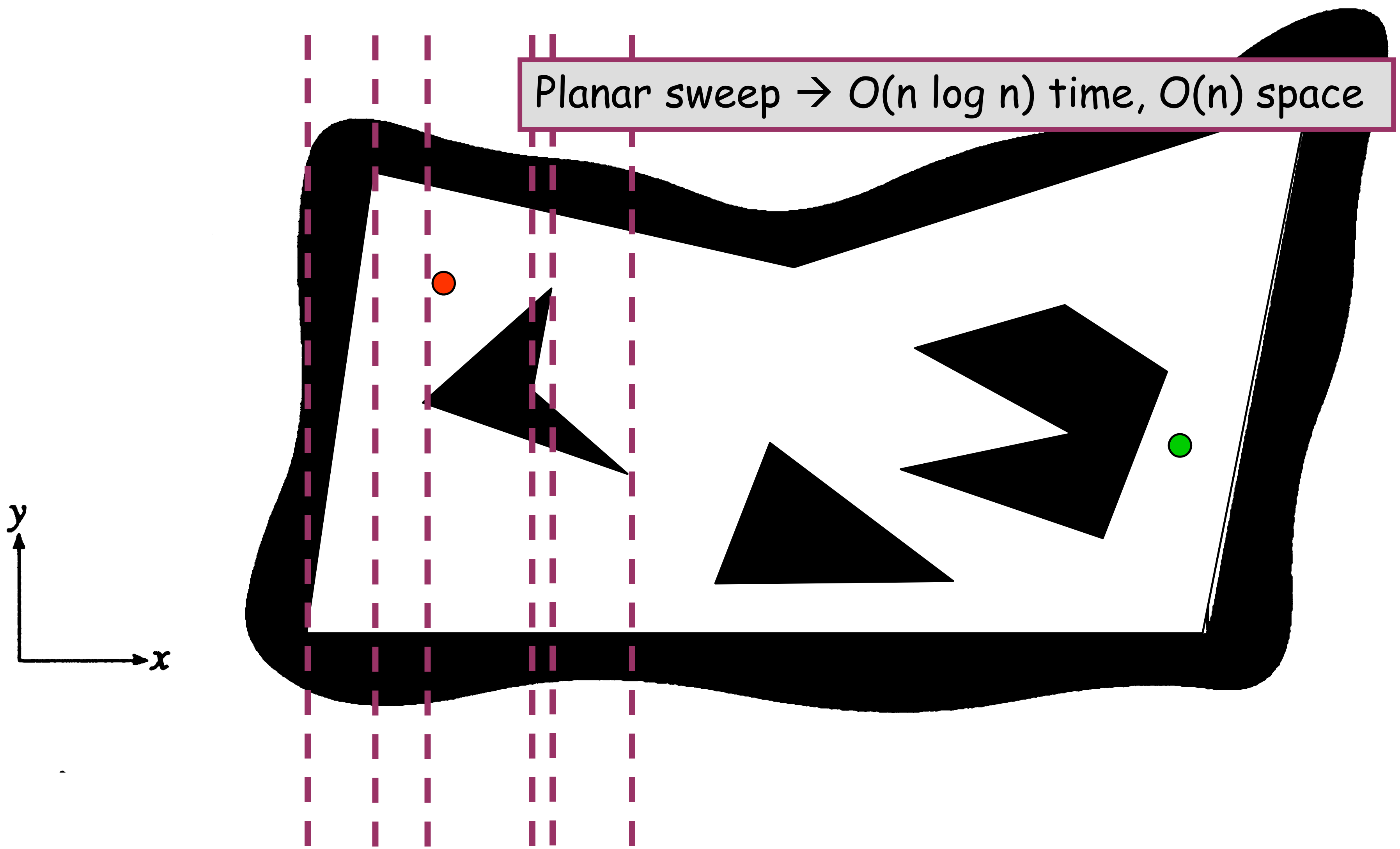


Trapezoidal decomposition



critical events → criticality-based decomposition

Trapezoidal decomposition

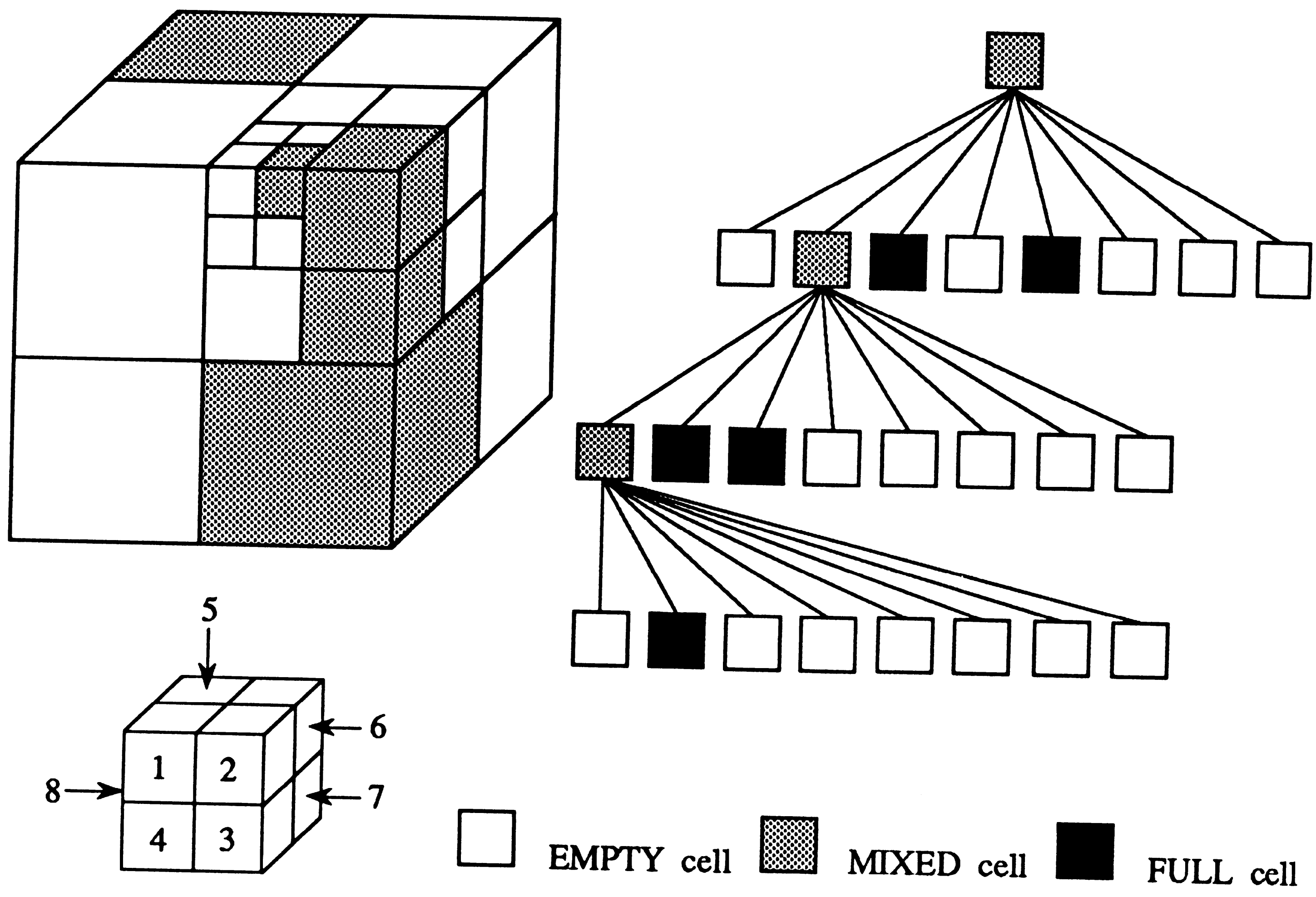


Cell-Decomposition Methods

Two classes 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

Octree Decomposition



Sketch of Algorithm

1. Compute cell decomposition down to some resolution
2. Identify start and goal cells
3. Search for sequence of empty/mixed cells between start and goal cells
4. If no sequence, then exit with no path
5. If sequence of empty cells, then exit with solution
6. If resolution threshold achieved, then exit with failure
7. Decompose further the mixed cells
8. Return to 2

Path-Planning Approaches

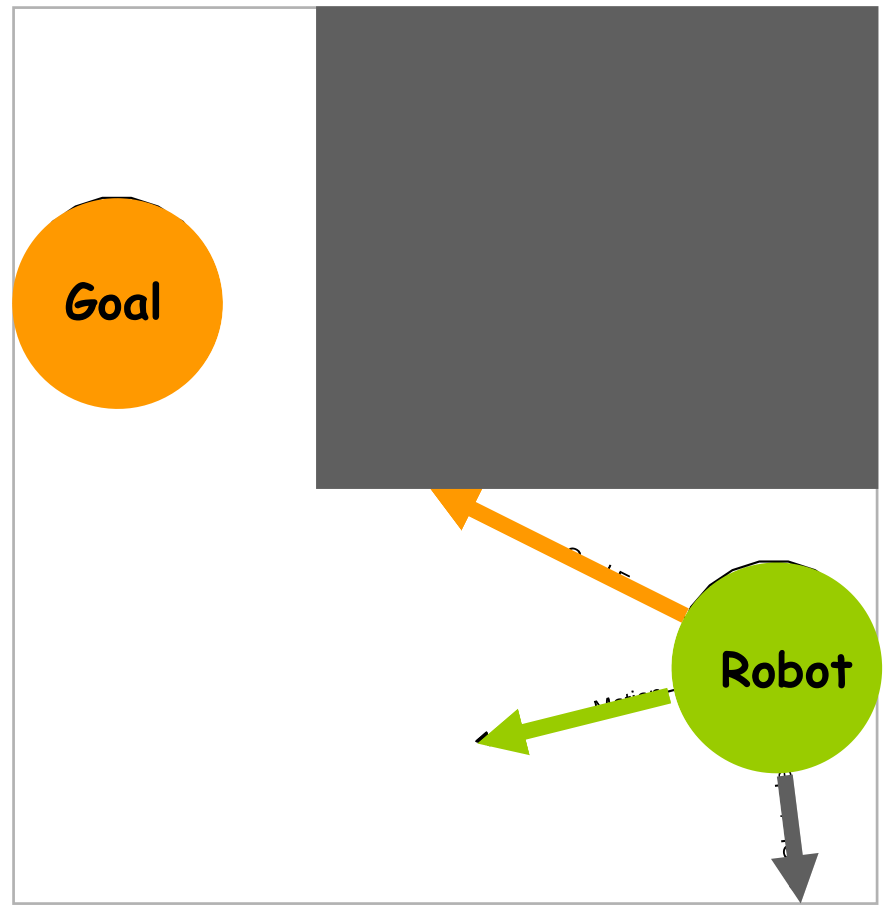
1. **Roadmap**
   Represent the connectivity of the free space by a network of 1-D curves
2. **Cell decomposition**
   Decompose the free space into simple cells and represent the connectivity of the free space by the adjacency graph of these cells
3. **Potential field**
   Define a function over the free space that has a global minimum at the goal configuration and follow its steepest descent

Potential Field Methods

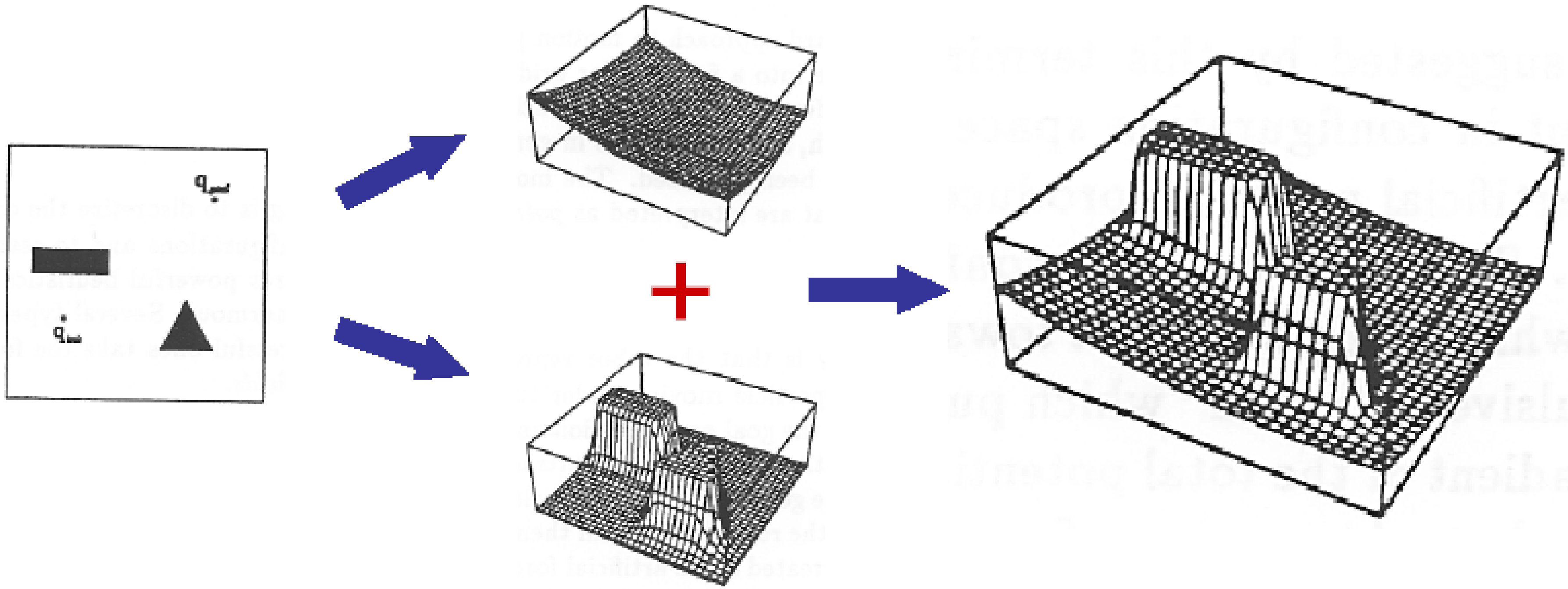
- Approach initially proposed for real-time collision avoidance [Khatib, 86]. Hundreds of papers published on it.

$$F_{Goal} = -k_p\left(x - x_{Goal}\right)$$

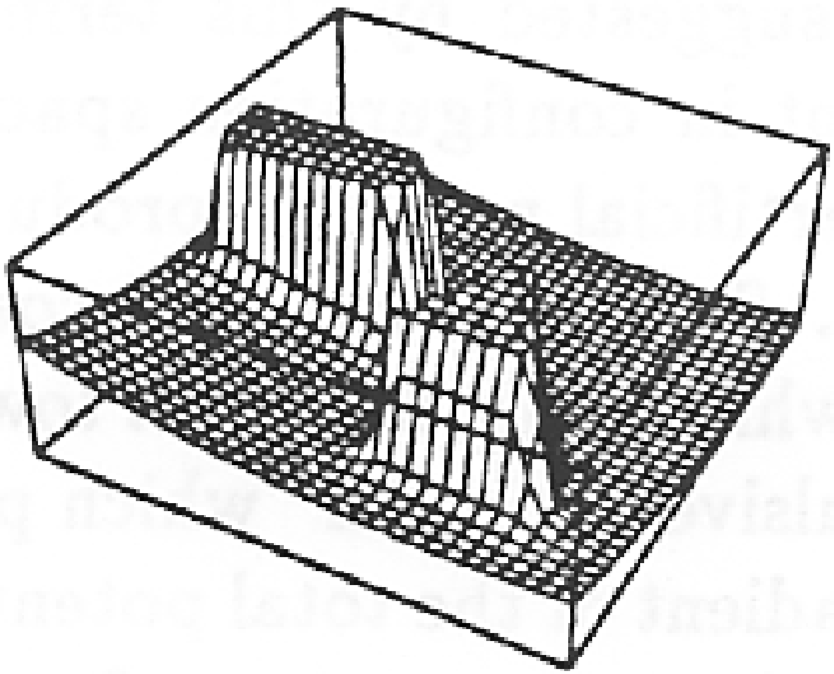
$$F_{Obstacle} = \begin{cases} \eta\left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)\frac{1}{\rho^2}\frac{\partial \rho}{\partial x} & \text{if } \rho \le \rho_0, \\ 0 & \text{if } \rho > \rho_0 \end{cases}$$



Attractive and Repulsive fields



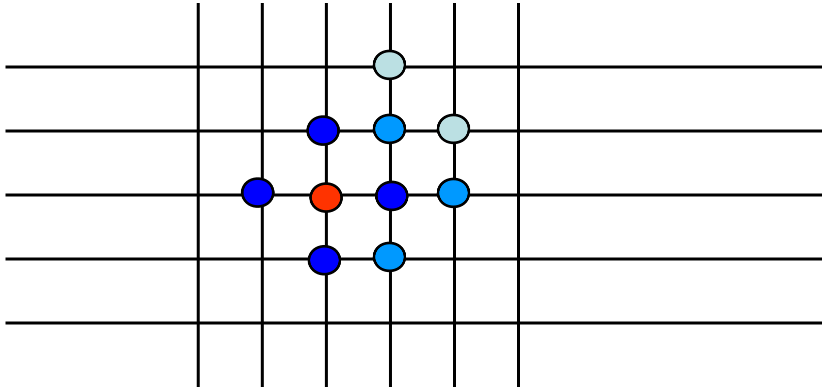
Local-Minimum Issue



- Perform best-first search (possibility of combining with approximate cell decomposition)
- Alternate descents and random walks
- Use local-minimum-free potential (navigation function)

Sketch of Algorithm (with best-first search)

1. Place regular grid G over space
2. Search G using best-first search algorithm with potential as heuristic function



Simple Navigation Function

2	1	2	3
1	0	1	2
2			3
3	4	5	4

Simple Navigation Function

2	1	2	3
1	0	1	2
2			3
3	4	5	4

Simple Navigation Function

2	1	2	3
1	0	1	2
2			3
3	4	5	4

Completeness of Planner

- A motion planner is complete if it finds a collision-free path whenever one exists and return failure otherwise.
- Visibility graph, Voronoi diagram, exact cell decomposition, navigation function provide complete planners
- Weaker notions of completeness, e.g.:
  - resolution completeness
    (PF with best-first search)
  - probabilistic completeness
    (PF with random walks)

- A probabilistically complete planner returns a path with high probability if a path exists. It may not terminate if no path exists.

- A resolution complete planner discretizes the space and returns a path whenever one exists in this representation.

Preprocessing / Query Processing

- Preprocessing:
  Compute visibility graph, Voronoi diagram, cell decomposition, navigation function

- Query processing:
  - Connect start/goal configurations to visibility graph, Voronoi diagram
  - Identify start/goal cell
  - Search graph

Issues for Future Classes

- Space dimensionality
- Geometric complexity of the free space
- Constraints other than avoiding collision
- The goal is not just a position to reach
- Etc …