Roadmaps

Robots in Context

Leon Bodenhagen lebo@mmmi.sdu.dk

Associate Professor

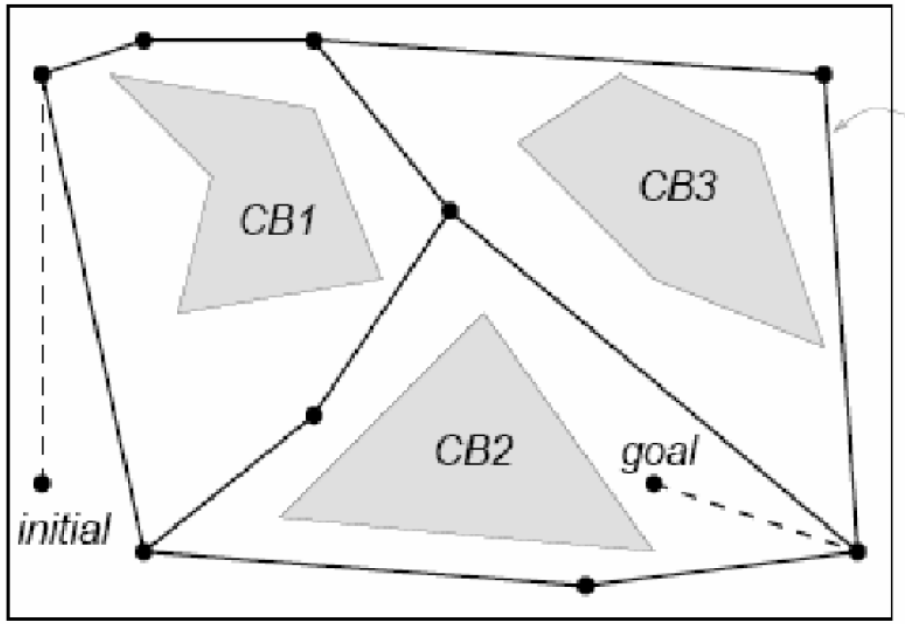
SDU Robotics, Maersk Mc-Kinney Moller Institute

Content

- Roadmaps
- Visibility Roadmaps
- Generalized Voronoi Map
- Canny's Roadmap Algorithm
- Graph search

Roadmaps

Basic idea: Capture the connectivity of the free space by a graph.



Roadmap Definition 1

- General
  - A roadmap is graph where nodes correspond to configurations and edges correspond to collision free paths between configurations.

Roadmap Definition 2

- A union of one-dimensional curves is a roadmap, RM, if for all q_{start} and q_{goal} in Q_{free} that can be connected the following holds
  - Accessibility: There exists a path from q_{start} to some q_{start}' in RM
  - Departability: There exists a path from q_{goal} to some q_{goal}' in RM
  - Connectivity: There exists a path in RM between q_{start}' and q_{end}'

Path Planning using a Roadmap

- Build the Roadmap
- Connect q_{start} and q_{goal} to q_{start}' and q_{end}' both belonging to the roadmap.
- Find a path between q_{start}' and q_{end}'

- What is the hard part here?
- Given a roadmap according to definition 2, what are the properties of the planner?

Building a Roadmap

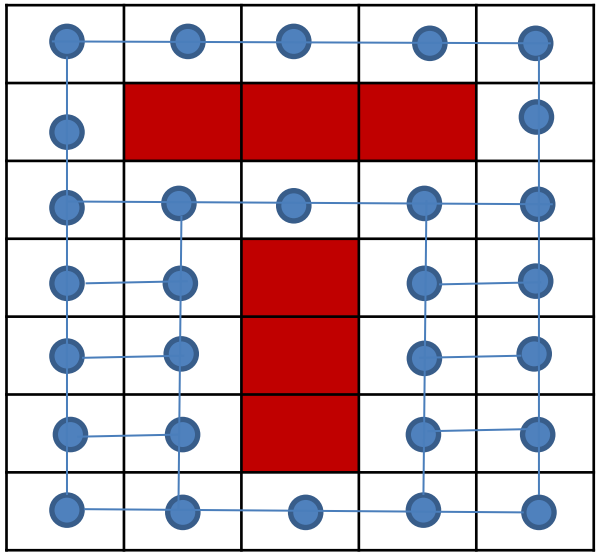
- Grid sampling
- Visibility Graph
- Voronoi Diagram
- Cannys roadmap method
- Probabilistic roadmaps (to be discussed later in the course)

Building a Roadmap

- **Grid sampling**
- Visibility Graph
- Voronoi Diagram
- Cannys roadmap method

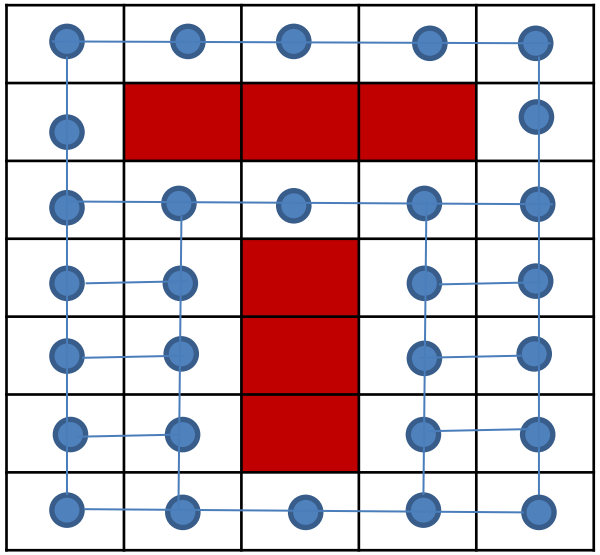
Grid Samling

- Divide the space into a grid.
- Create a node for each collision free cell.
- Connect all neighboring collision free cells (either as 4 or 8 connected)



Grid Samling

- Divide the space into a grid.
- Create a node for each collision free cell.
- Connect all neighboring collision free cells (either as 4 or 8 connected)



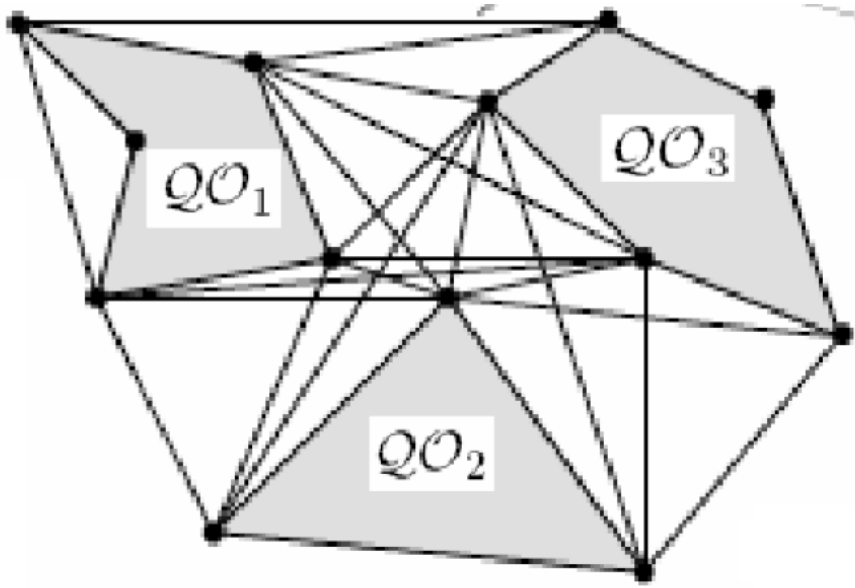
Pros and cons?

Building a Roadmap

- Grid sampling
- **Visibility Graph**
- Voronoi Diagram
- Cannys roadmap method

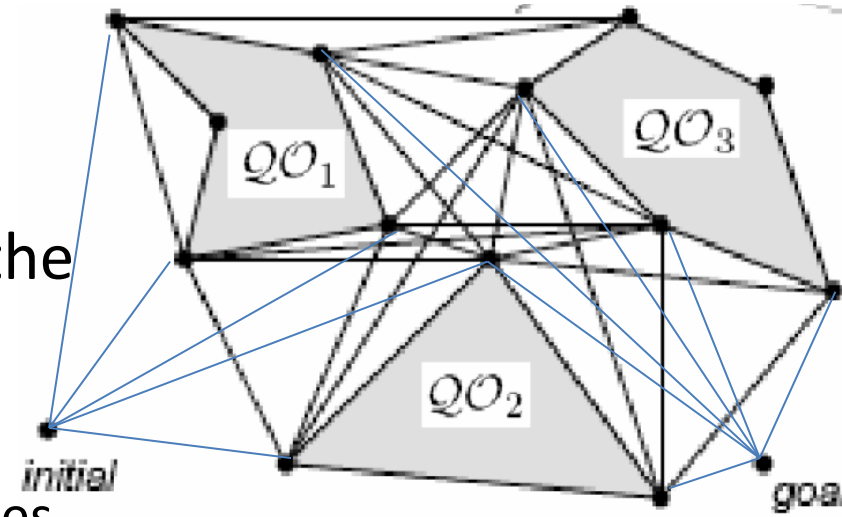
Visibility Graph

- Assumes all obstacles are polygons
- Nodes in the graph are verticies of the obstacles.
- Nodes are connected if
  - They are already connected by an edge on the obstacle.
  - The line segment joining them is collision free.



Visibility Graph

- We may include the initial and goal configurations in the same way.
- The algorithm for constructing the roadmap can be done in $O(N^3)$
  - N is the total number of verticies
  - Each have O(N) neighbor candidates.
  - Checking each candidate is O(N) because there are O(N) edges on the polygon to check collision against.



Visibility Graph

- **Discuss properties and pros and cons.**

Visibility Graph

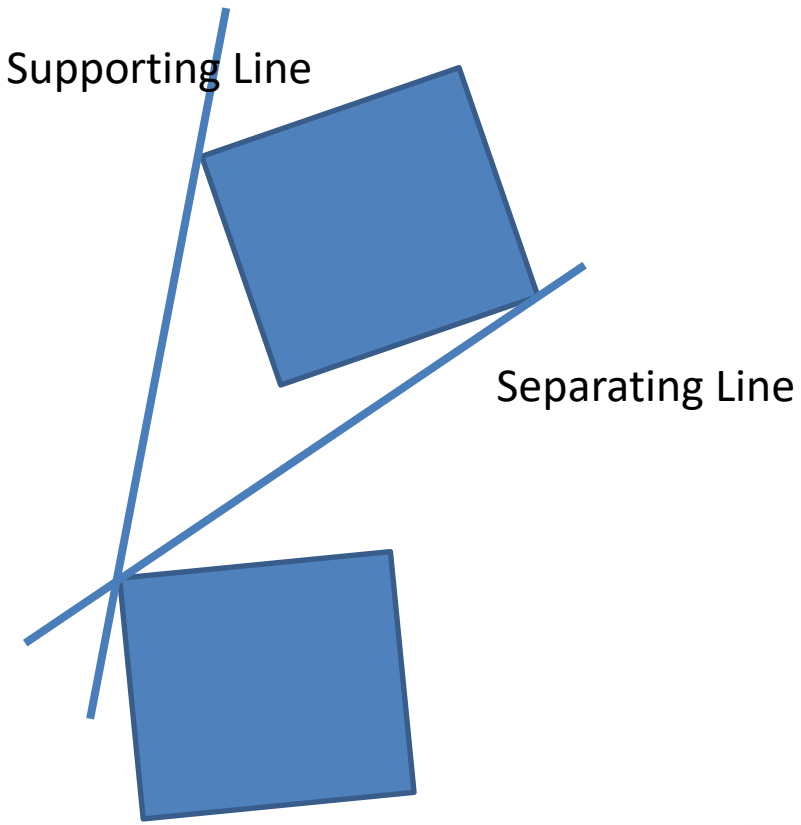
- Discuss properties and pros and cons.

- Pros
  - Gives the shortest possible path.
  - Small number of nodes (for simple obstacles)
  - Complete roadmap
- Cons
  - Goes close to obstacles
  - Requires polygons as obstacles.
  - May have many points
  - Many redundant edges

Reduced Visibility Graph

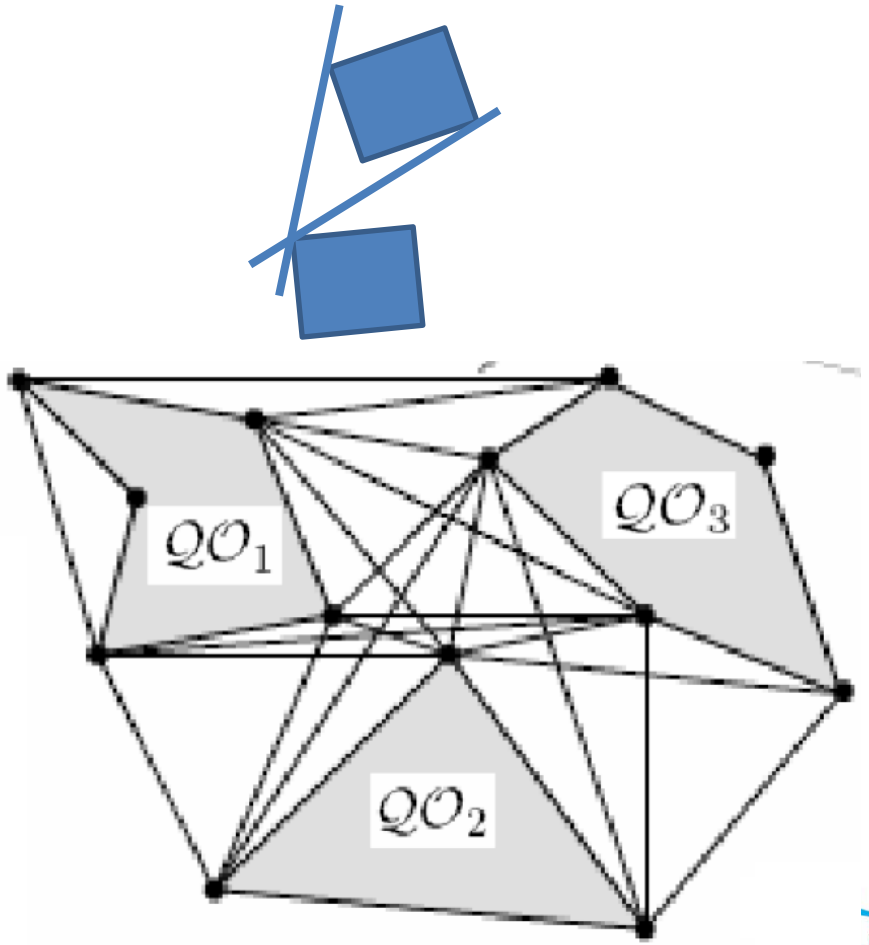
- Define supporting and seperating lines.
  - Supporting: Tangent to two obstacles both on the same side of the line.
  - Separating: Tangent to two obstacle on either side of the line.

- **Only** use those edges which corresponds to either a **separating** or **supporting** line



Reduced Visibility Graph - Exercise

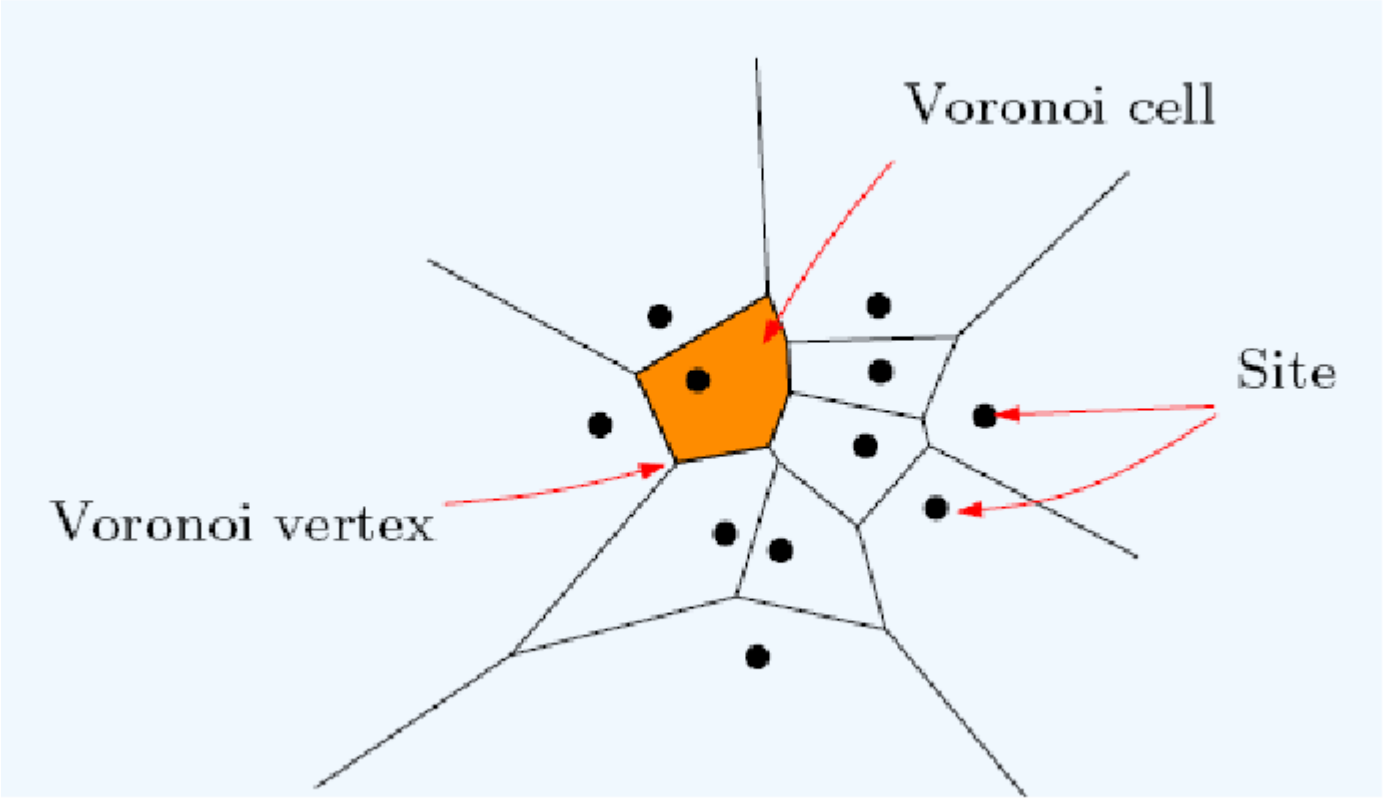
- Define supporting and seperating lines.
- Only use those edges which corresponds to either a separating or supporting line
- Use the principle to update the visibility roadmap



Building a Roadmap

- Grid sampling
- Visibility Graph
- **Voronoi Diagram**
- Cannys roadmap method

Voronoi Diagram



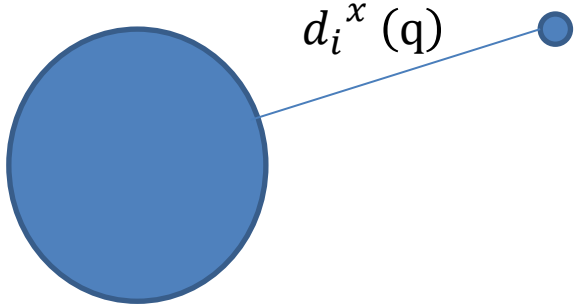
Basic Definitions

- Point to Obstacle Distance

$$d_i{}^x(\mathrm{q}) = \min_{c \in QO_i} d(q, c)$$

- Gradient

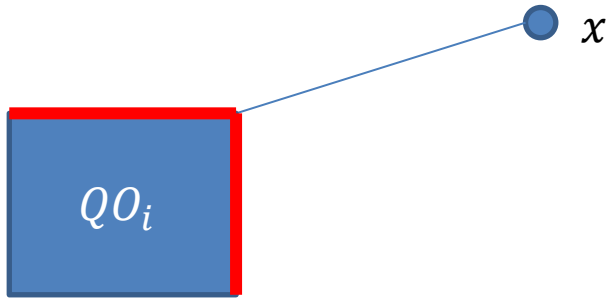
$$\nabla d_i{}^x(\mathrm{q}) = \frac{q - c}{d(q, c)}$$



Basic Definitions

- Points within line of sight

$$C_i(\mathrm{x}) = \{\mathrm{c} \in QO_i : \forall t \in [0,1], x(1-t) + c\,t \in Q_{free}\}$$



- Visible distance function:

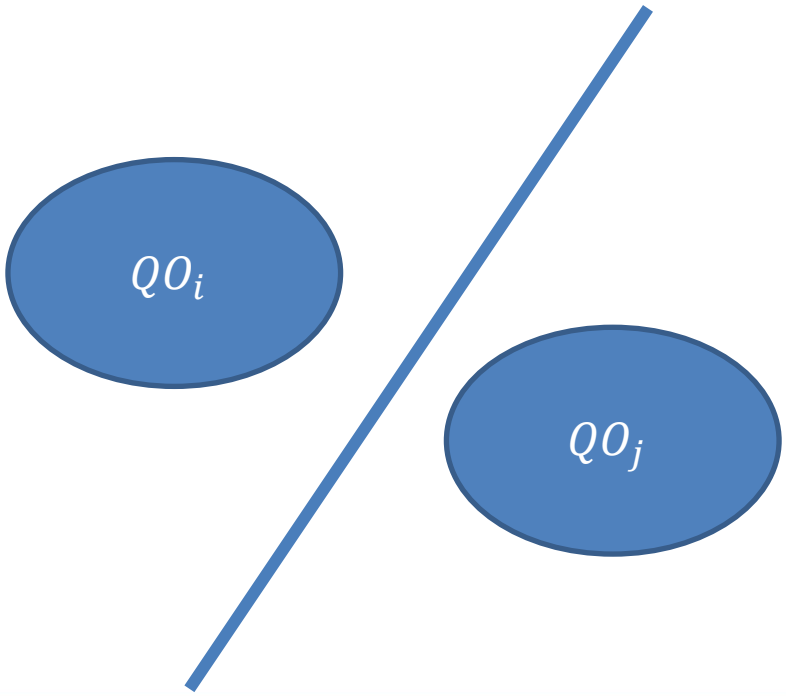
$$d_i(x) = \begin{cases} \min\limits_{c \in QO_i} d(x,c) \;\; if \;\; c \in C_i(x) \\ \infty \; otherwise \end{cases}$$

- Multiple objects

$$D(x) = \min_i d_i(x)$$

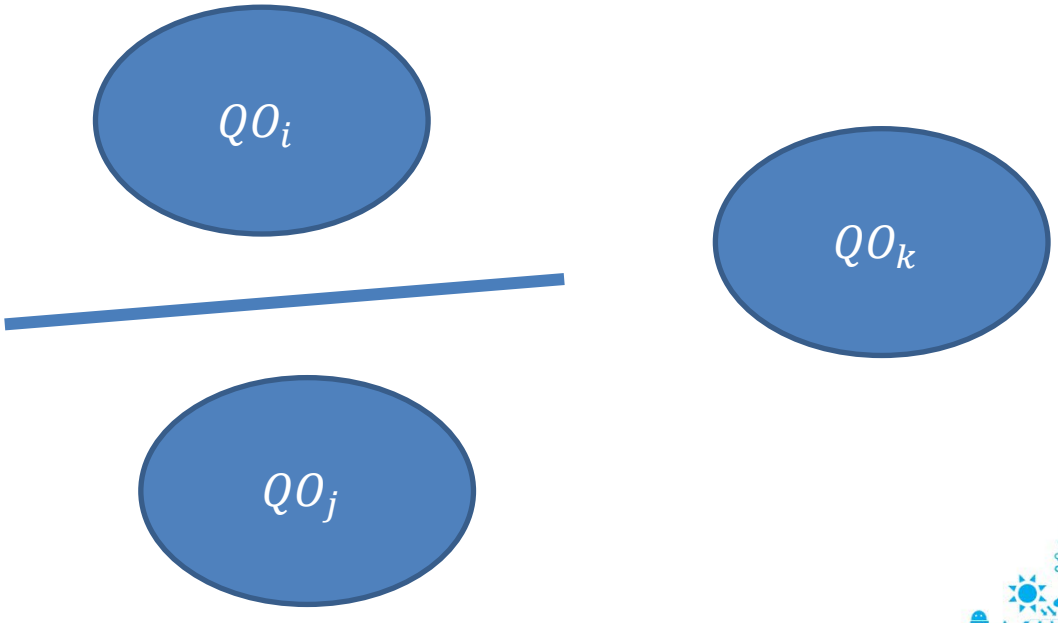
Equidistant for Two Objects

- Equidistant surface
- $S_{ij} = \{q \in Q_{free} | d_i(q) - d_j(q) = 0\}$



Equidistant for with Multiple Objects

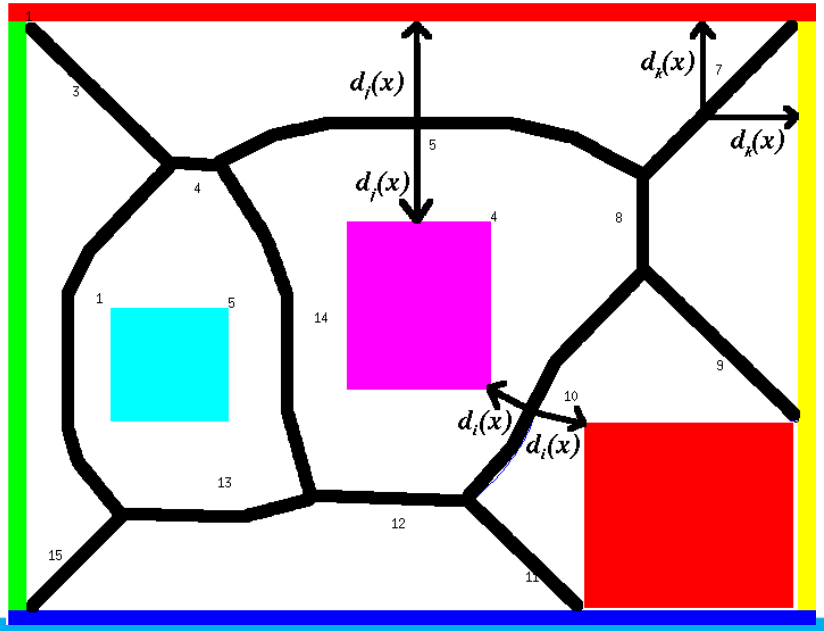
- In case of multiple obstacles
- $F_{ij} = \{q \in S_{ij} | d_i(q) \leq d_k(q) \forall k\}$



Generalized Voronoi Diagram

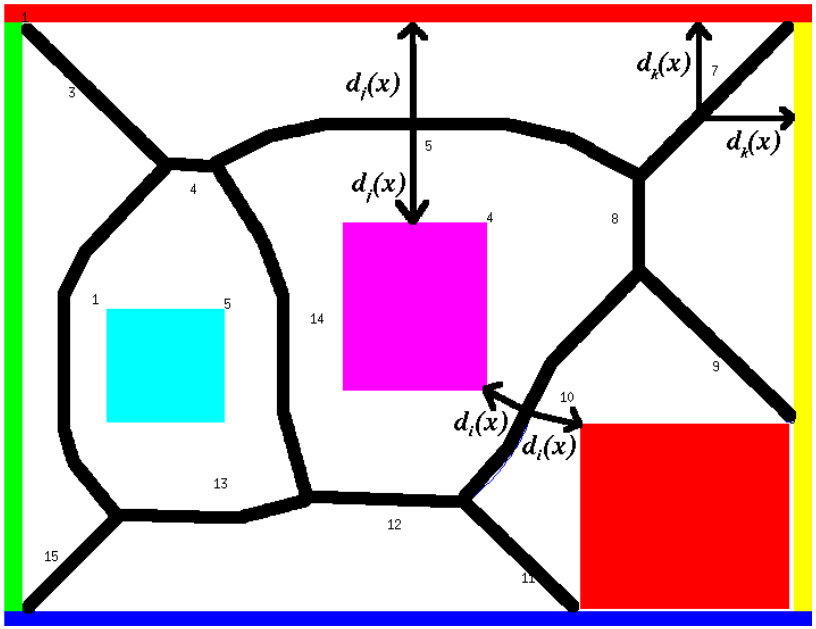
- Definition of Generalized Voronoi Diagram

$$GVD = \bigcup_i \bigcup_j F_{ij}$$



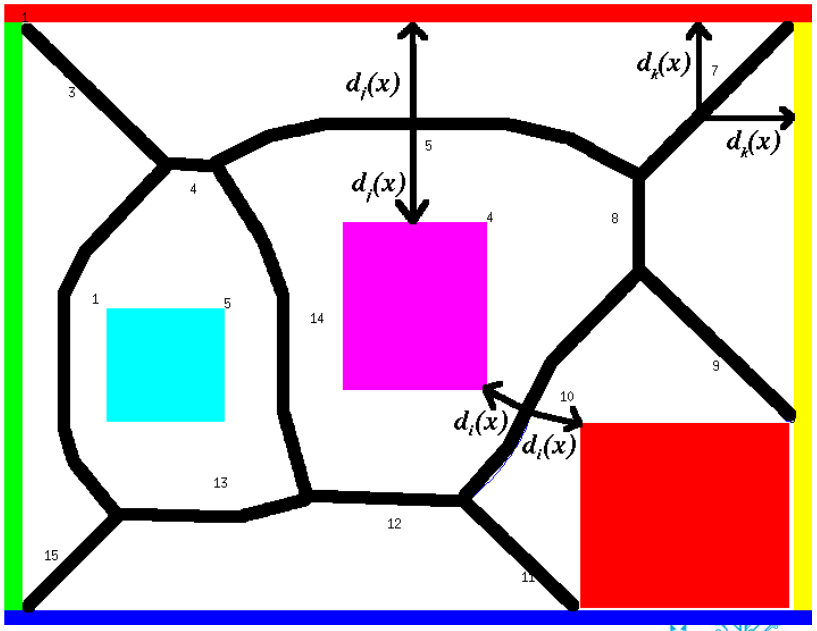
Roadmaps with Generalized Voronoi Diagrams

- How to interpret a Voronoi Diagram as a roadmap?



Roadmaps with Generalized Voronoi Diagrams

- Will the roadmap satisfy the requirements of
  - Accessibility
  - Departability
  - Connectivity



Relation between the Brushfire Algorithm and the Voronoi Diagram

1	1				1	1	1		
1	1				1	1	1		
1	1				1	1	1		
1	1								
1	1						1	1	
1	1						1	1	

Relation between the Brushfire Algorithm and the Voronoi Diagram

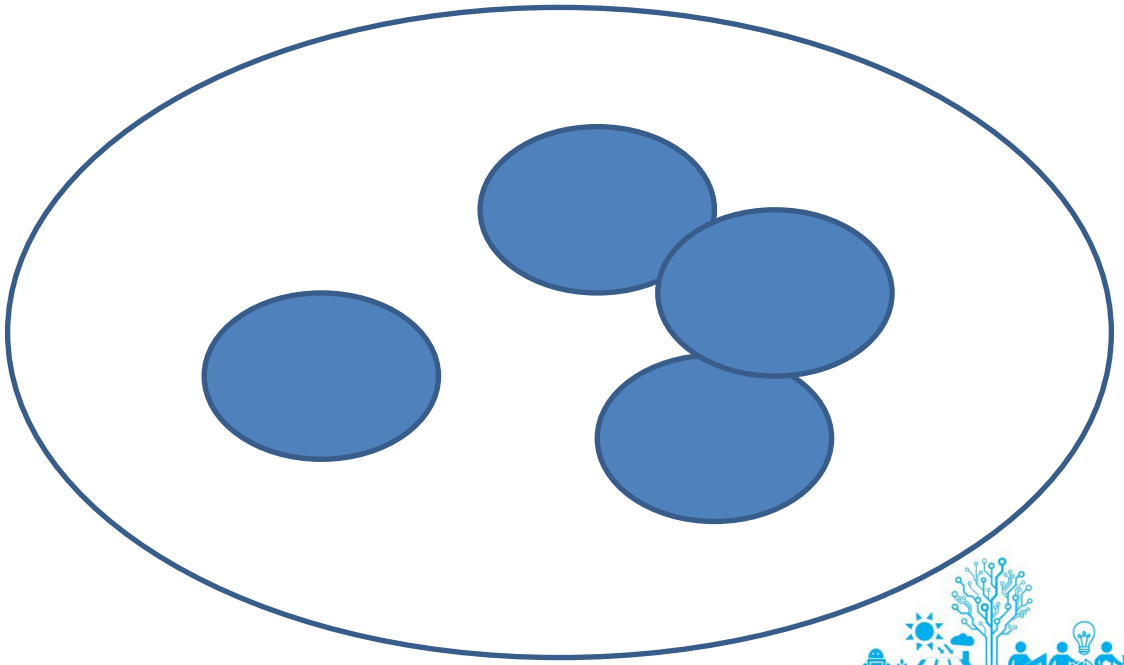
2	2	2	3	2	2	2	2	2	3
1	1	2	3	2	1	1	1	2	3
1	1	2	3	2	1	1	1	2	3
1	1	2	3	2	1	1	1	2	3
1	1	2	3	2	2	2	2	2	2
1	1	2	3	3	3	2	1	1	2
1	1	2	3	4	3	2	1	1	2
2	2	2	3	4	3	2	2	2	2
3	3	3	3	4	3	3	3	3	3

Building a Roadmap

- Grid sampling
- Visibility Graph
- Voronoi Diagram
- **Cannys roadmap method**

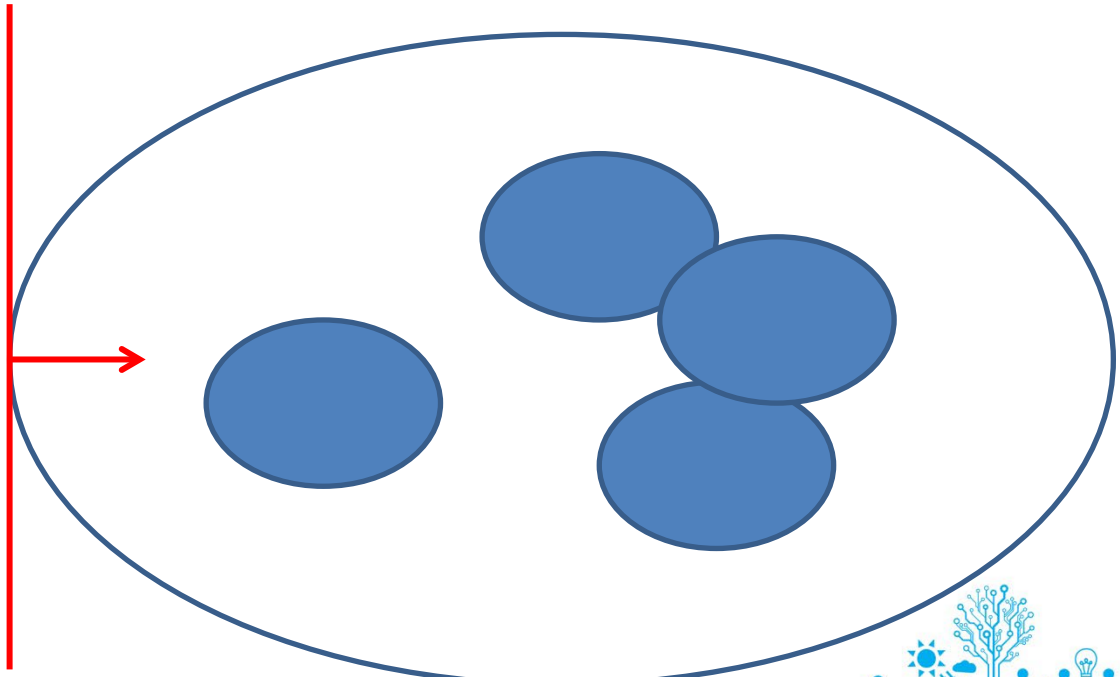
Silhouette Methods

- Canny's Roadmap Algorithm
  - Works for obstacles defined a polynomials



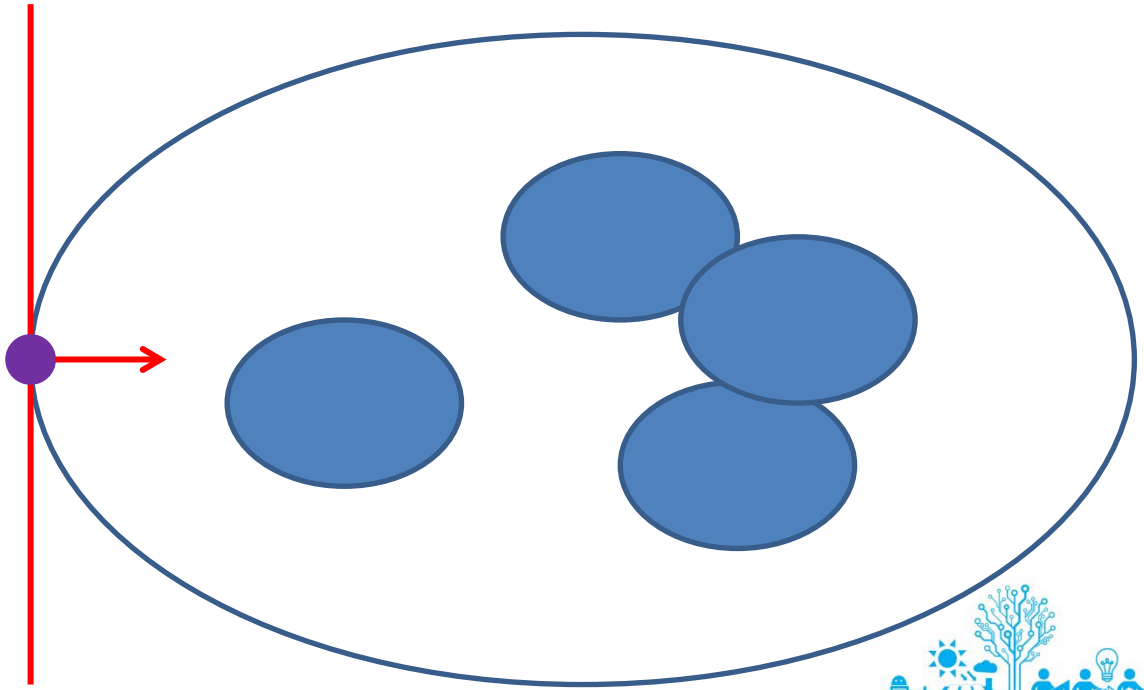
Canny's Roadmap Algorithm

- Define a direction for a sweeping slice



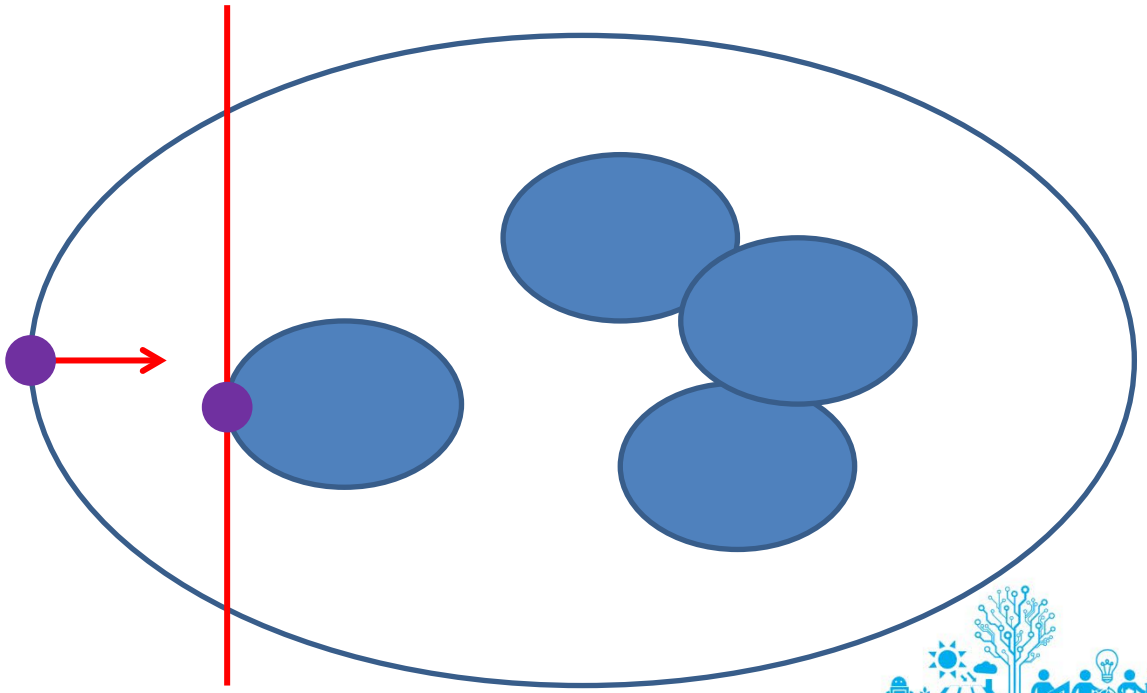
Canny's Roadmap Algorithm

- Mark all points where the number of contours intersecting changed



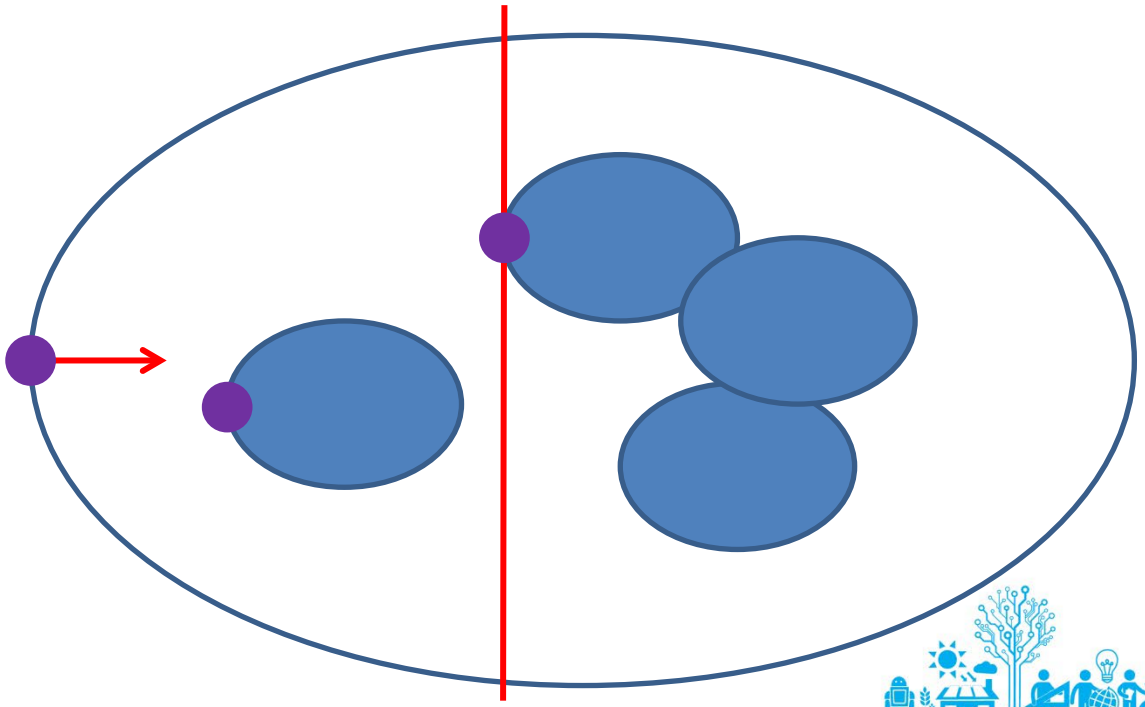
Canny's Roadmap Algorithm

- Mark all points where the number of contours intersecting changed



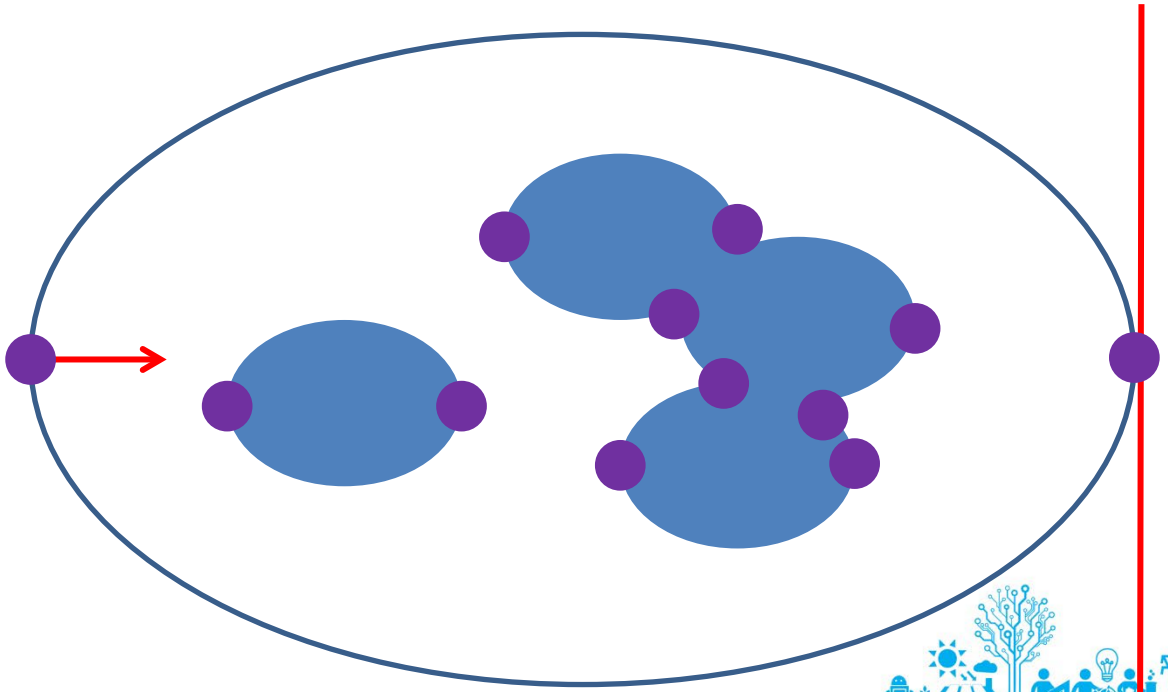
Canny's Roadmap Algorithm

- Mark all points where the number of contours intersecting changed



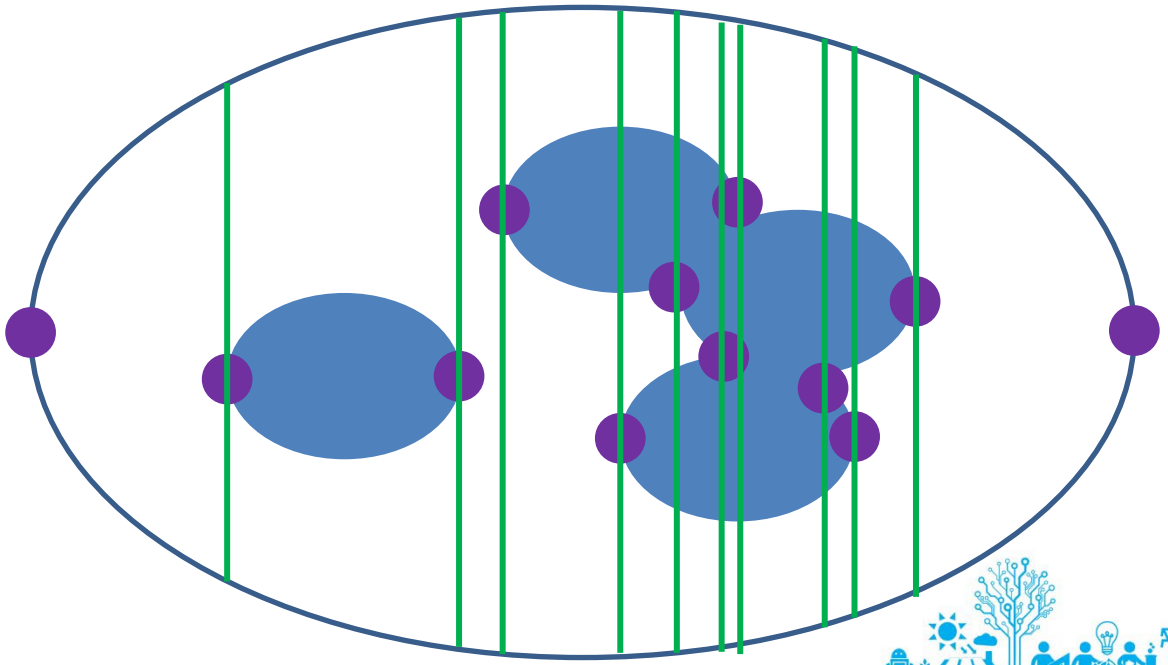
Canny's Roadmap Algorithm

- Mark all "critical points" where the number of contours intersecting changed



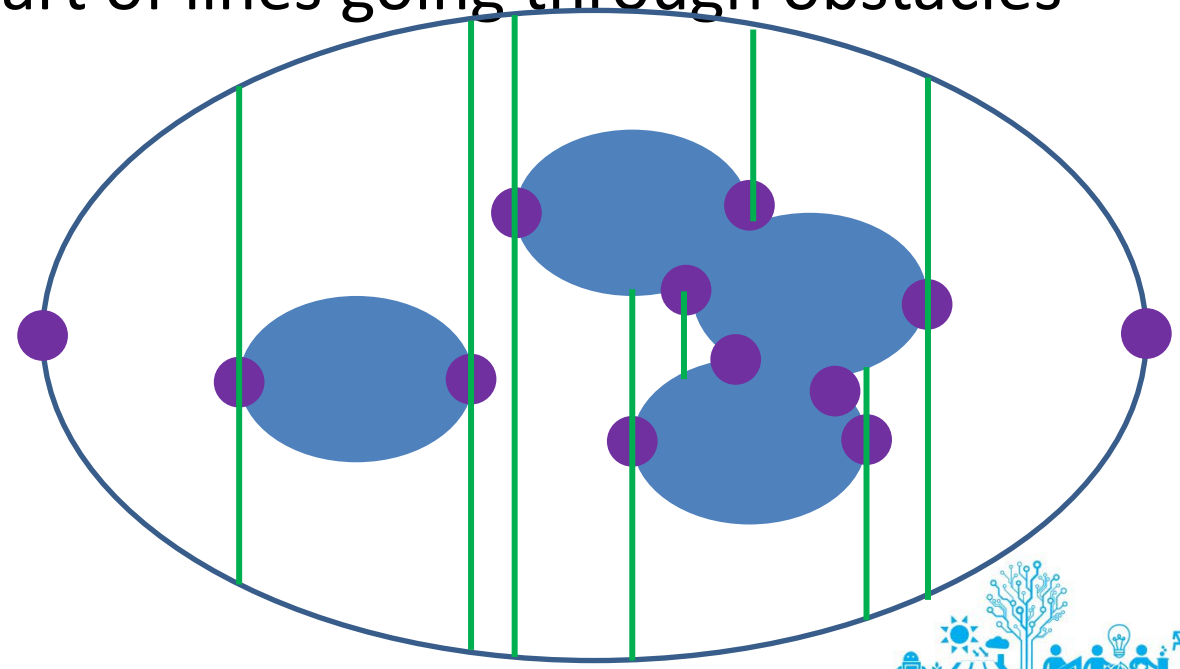
Canny's Roadmap Algorithm

- Create connections to the outer boundary through the critical points.



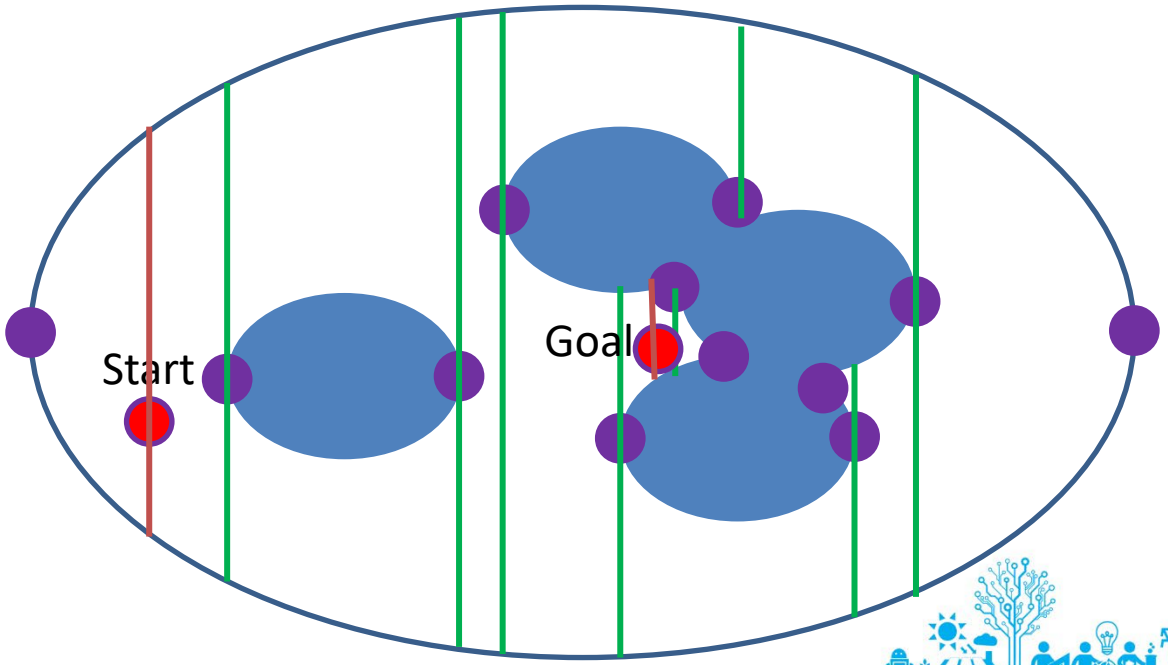
Canny's Roadmap Algorithm

- Keep the lines which
  - Are not colliding with obstacles (only tangent to)
  - The inner part of lines going through obstacles



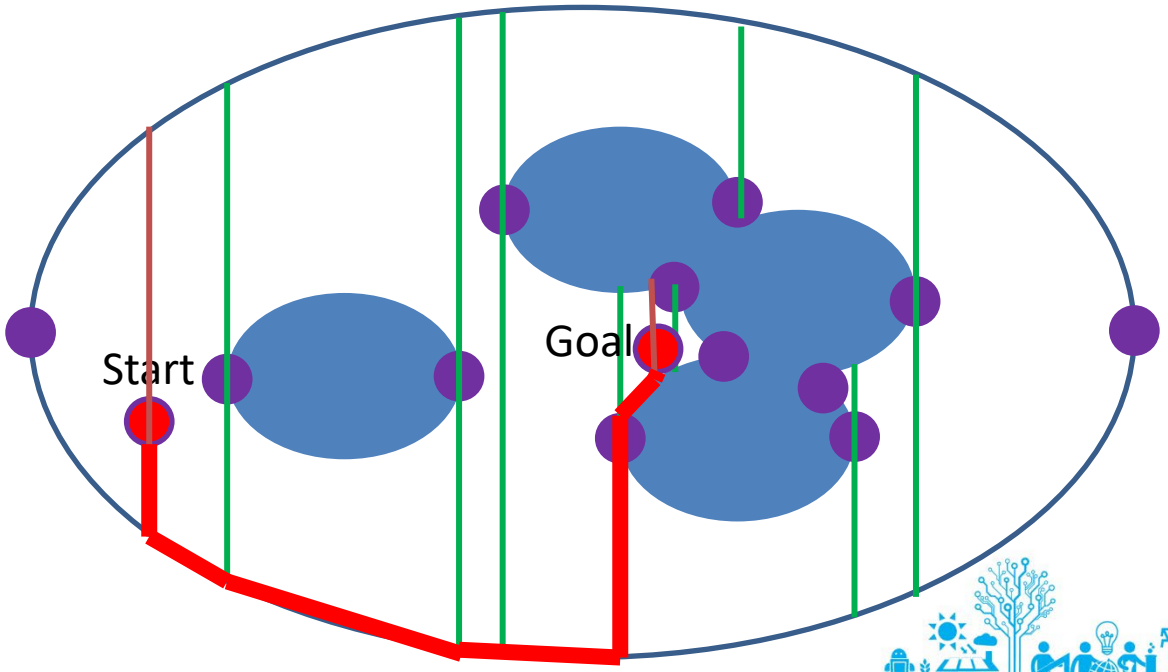
Canny's Roadmap Algorithm

- To plan add the start and goal configurations and add lines for these



Canny's Roadmap Algorithm

- Search the roadmap for a path from start to goal



Pros and Cons

- Quite fast to compute roadmap
- Long detours
- Follows many obstacles/boundaries
- Obstacles as polynomials
- Works in higher dimensions

Summary

- Roadmaps
  - Nodes correspond to collision-free configurations
  - Edges correspond to collision-free paths
  - Requirements
    - Accessibility
    - Connectivity
    - Departability