ELEC 3210
Introduction to Mobile Robotics
Lecture 16

(Machine Learning and Infomation Processing for Robotics)

Huan YIN

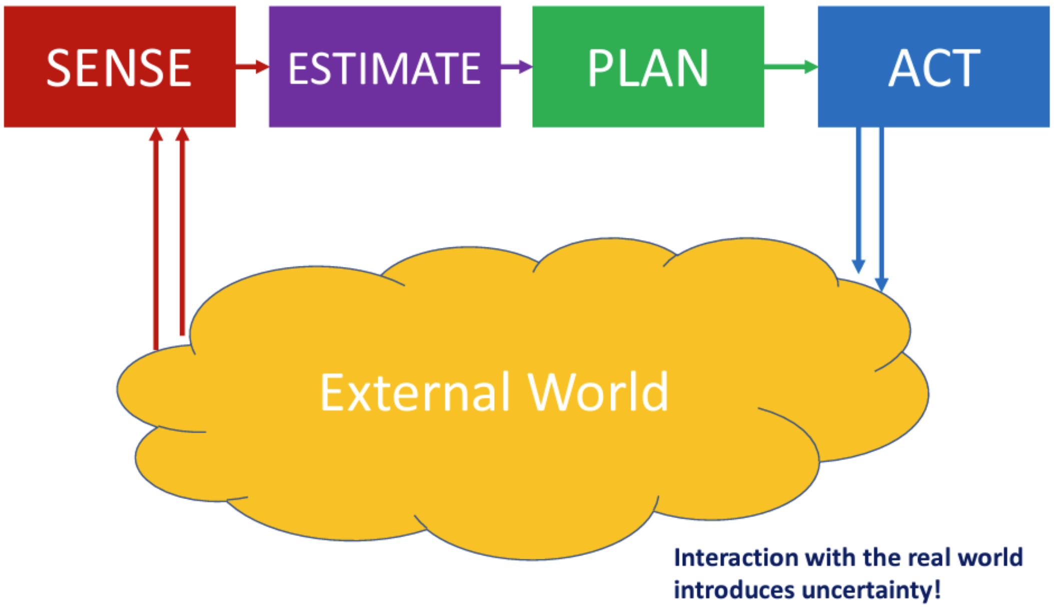
Research Assistant Professor, Dept. of ECE

eehyin@ust.hk




Robot Navigation Paradigm

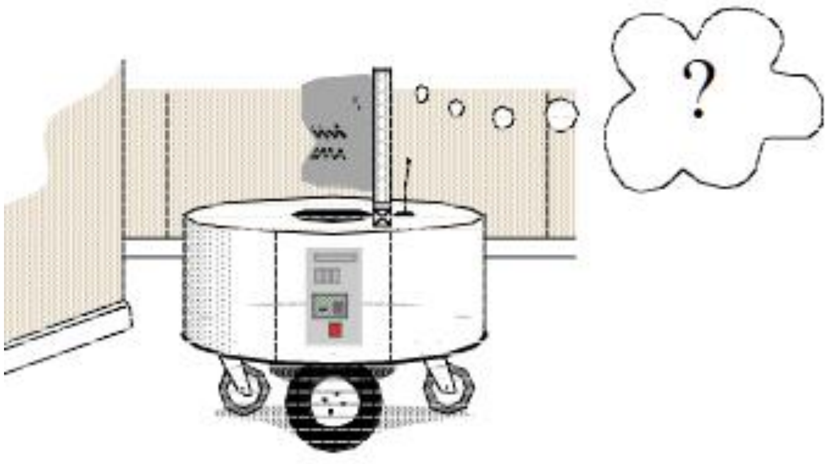
- Sensing&Estimation - Estimate current and past robot pose
- Planning - Generate future robot pose
- Control - Stabilize robot pose



Courtesy: Nikolay Atanasov, PyPose Video

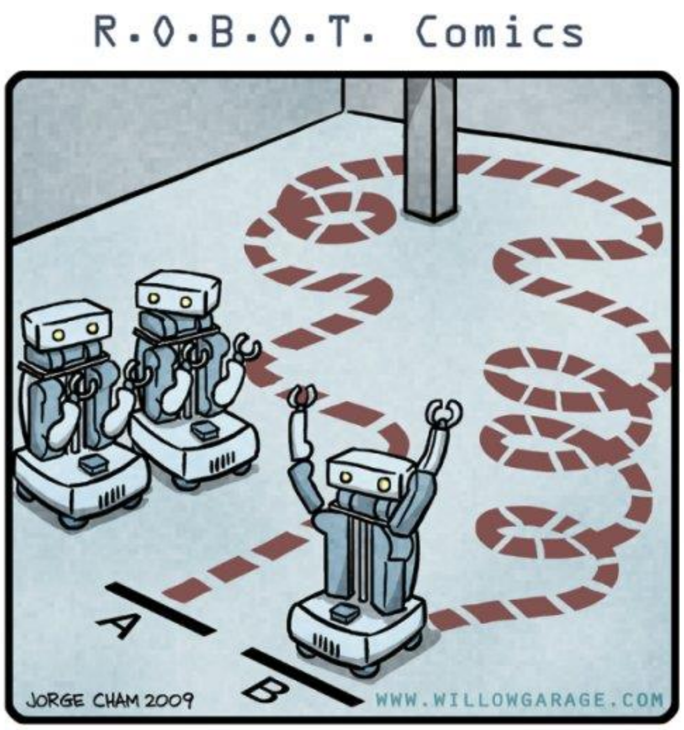
Three Questions

- Where am I ? (Sensing/Estimation) ☺
- Where am I going ? (Planning)
- How do I get there ? (Control)



Motion Planning This Week

- Mainly on concepts and classical algorithms on wheeled robot
- Guest Lecturer for Drones, next week



"HIS PATH-PLANNING MAY BE SUB-OPTIMAL, BUT IT'S GOT FLAIR."

Motion Planning - Concepts

What we need

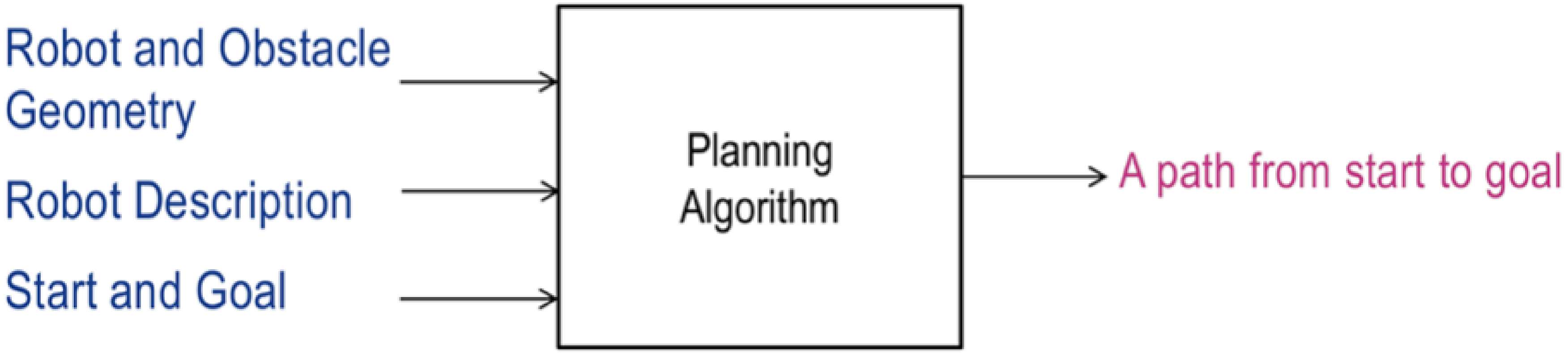
- Assume the estimated pose is reliable and accurate
- Move from place A to place B
    - collision-free
    - less distance
    - less energy
    - robot can move
    - etc.

- Constraints?

Courtesy:

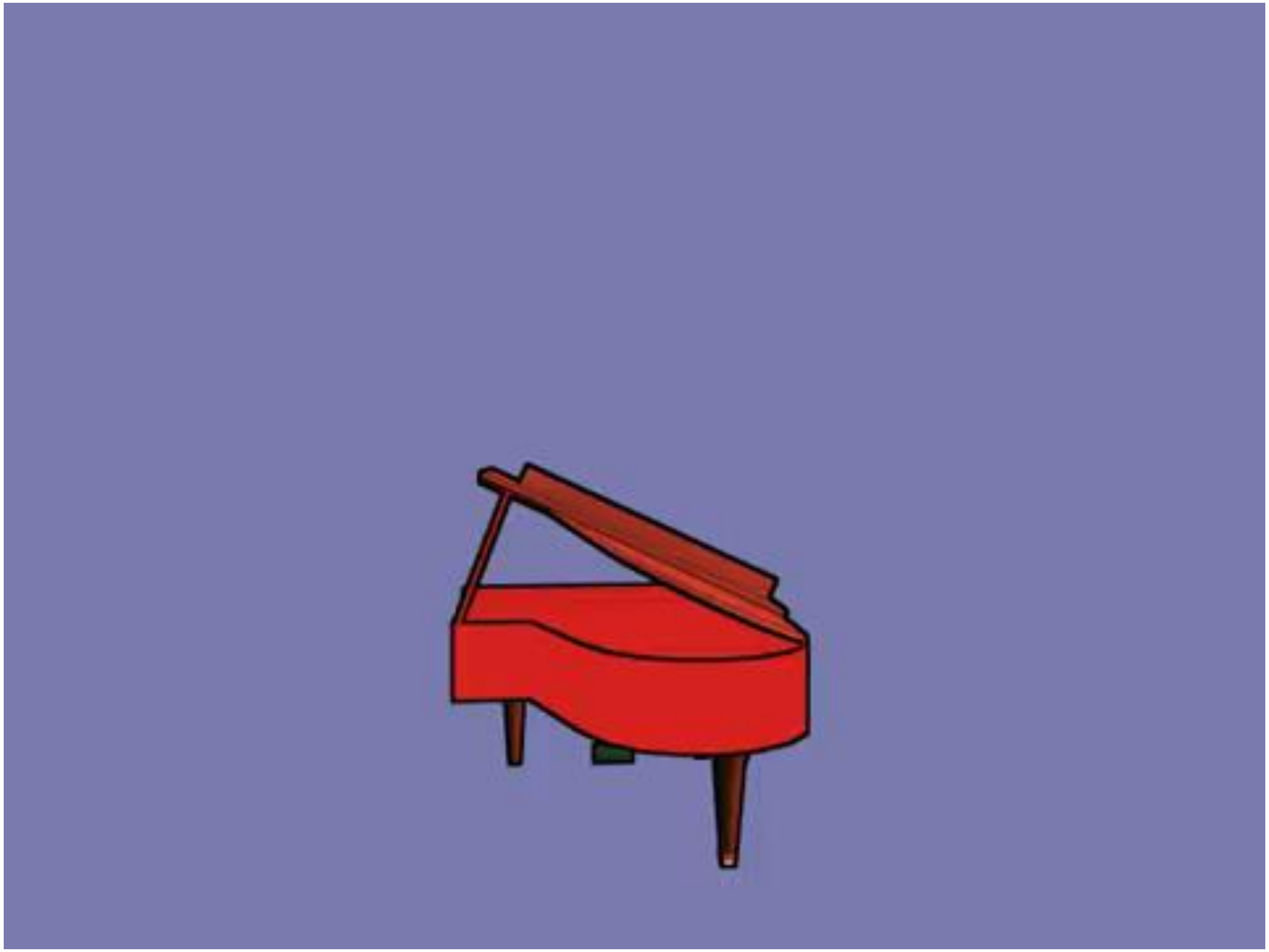
Constraints of Motion Planning

- Constraints
  - environment constraints (e.g., obstacles)
  - kinematics/dynamics of the robot



Courtesy: Nikolay Atanasov

Piano Mover's Problem



Courtesy: YouTube

Piano Mover’s Problem

Balancing Exploration and Exploitation in Sampling-Based Motion Planning

"The piano mover's problem"

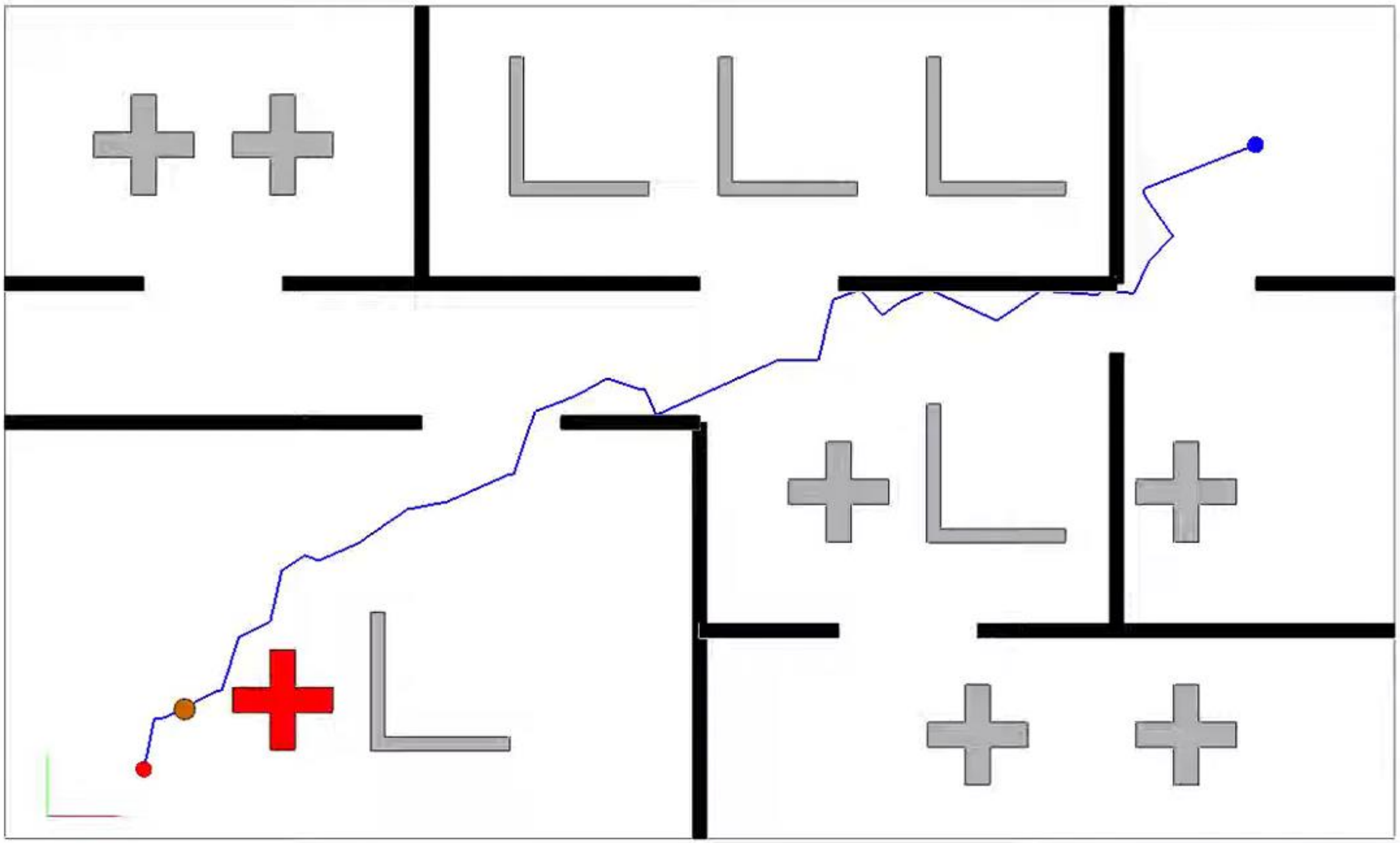
Markus Rickert, Arne Sieverling, and Oliver Brock

fortiss GmbH, An-Institut Technische Universität München, München, Germany
Robotics and Biology Laboratory, Technische Universität Berlin, Berlin, Germany

IEEE Transactions on Robotics

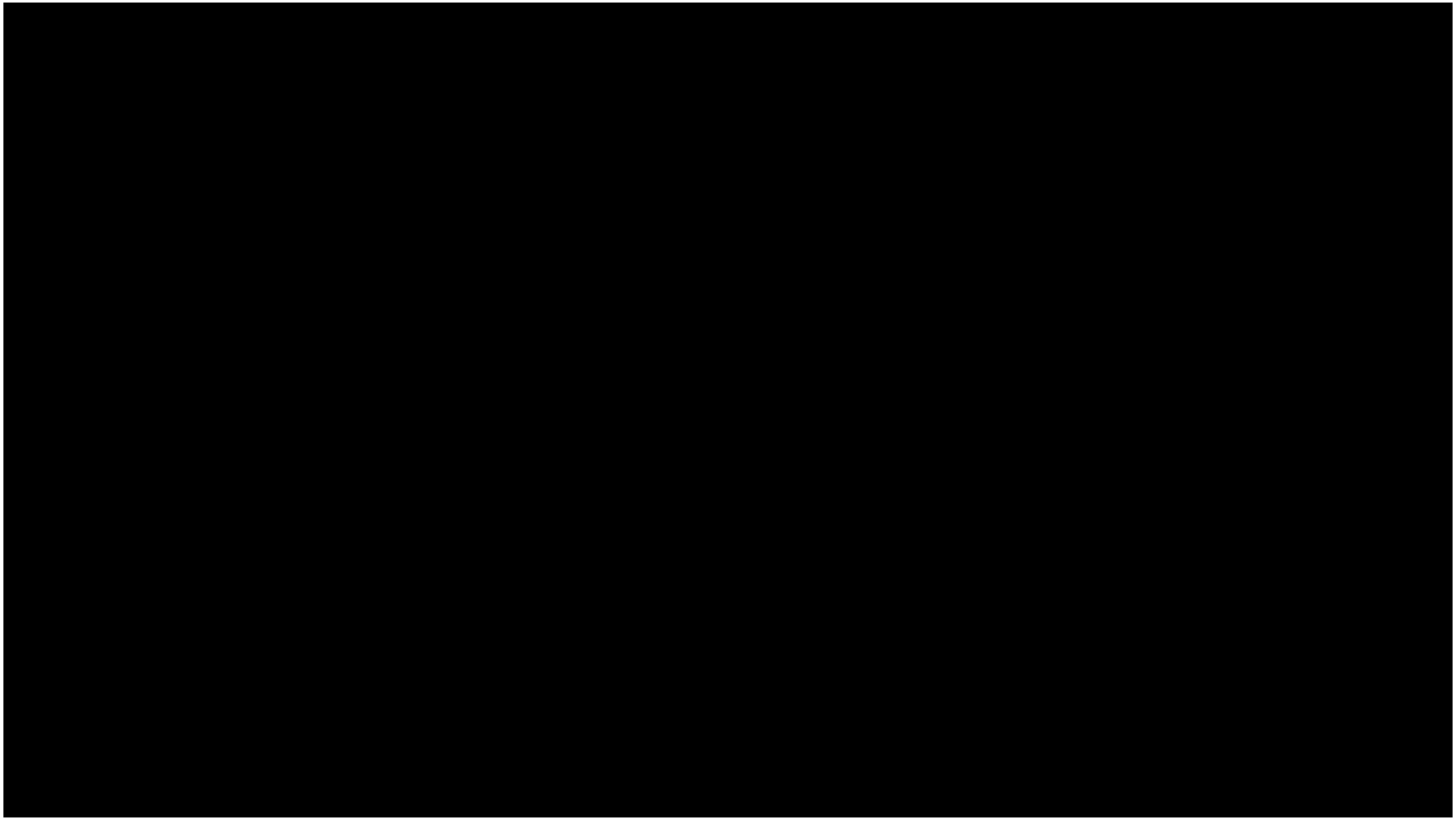
Courtesy: Robotics Library

Planning in Dynamic Environments



Courtesy: YouTube

Motion Planning for Swarms



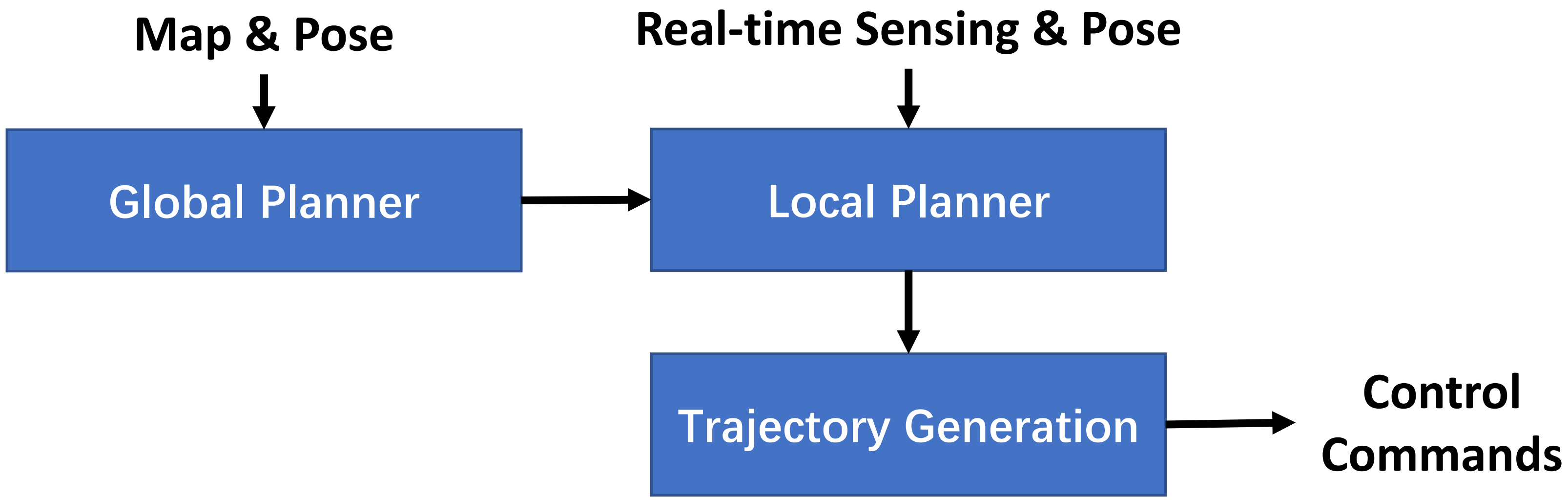
Courtesy: ZJU FAST LAB

Task Decomposition

- When you walk from the school gate to the classroom
- From coarse to fine Manner
- Motion Planning
  - **Global Planner**
    - Global path planning
    - find a path from point A to point B
    - low frequency
  - **Local Planner**
    - local path planning based on the global path
    - for obstacle avoidance
    - high frequency
  - **Trajectory Generation**
    - Convert path to trajectory or control commands that robot can move
    - higher frequency

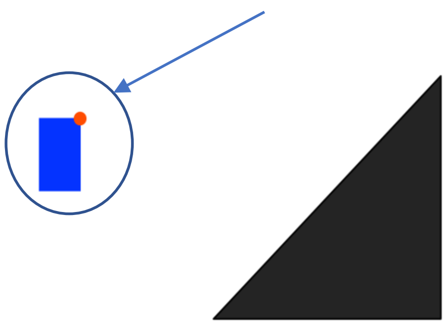
Task Decomposition

- A typical full pipeline for autonomous vehicles
- Partial pipeline could also work well in many cases



Planning in Workspace

- Workspace: 2D or 3D Euclidean space where the robot operates
- Planning in *workspace*
  - Robot has different shape and size
  - Collision detection requires knowing the robot geometry - time consuming and hard



(1) Rectangular mobile robot

(2) Circular mobile robot

Configuration

- A configuration is a specification of the position of every point on a robot body
- A configuration q is expressed as a vector of the degrees of freedom (DOF) of the robot:

$$\mathbf{q} = (q_1, \ldots, q_n)$$

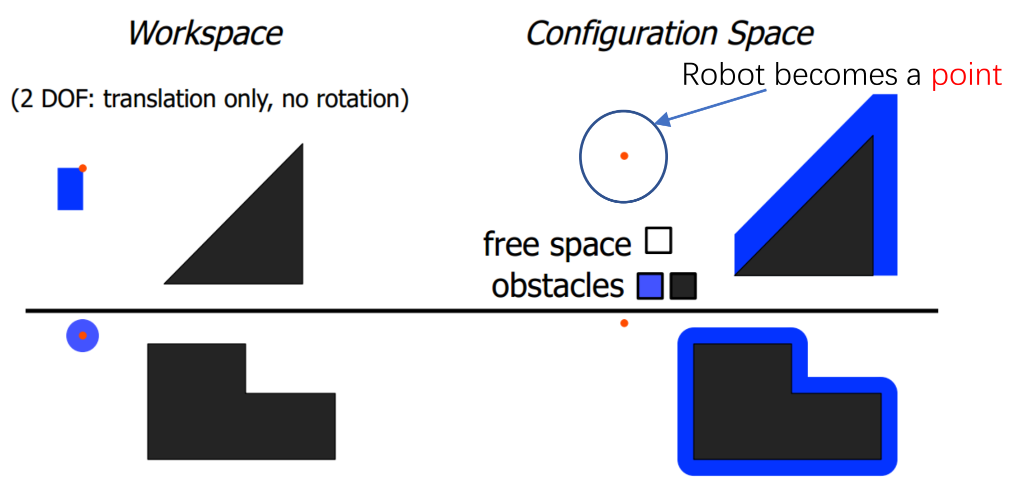
- 3 DOF: differential drive robot
- 6 DOF: rigid body motion with pose
- 7 DOF: 7-link manipulator (humanoid arm)

- **Configuration space C-Space:** set of all possible robot configurations.
- Each robot pose is a point in the C-space

Courtesy: Nikolay Atanasov

Planning in Configuration Space

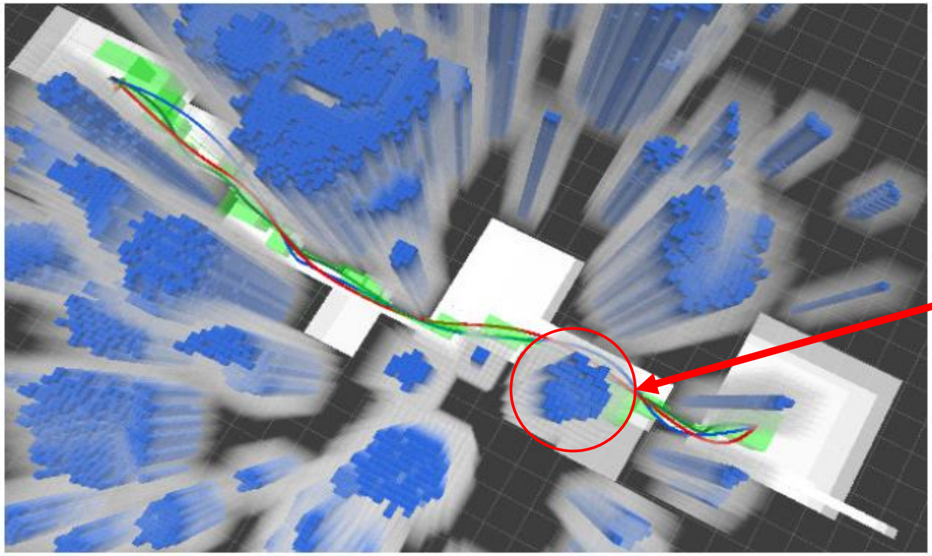
- Planning in *configuration space*: C-space
  - Robot is represented by a point in C-space, e.g. position (a point in $\mathbb{R}^3$), pose (a point in $SE(3)$) , etc.
  - Obstacles need to be represented in configuration space (one-time work prior to motion planning), called configuration space obstacle, or C-obstacle
  - *C-space = (C-obstacle)* ∪ *(C-free)*
  - The path planning is finding a path between start point q_{start} and goal point q_{goal} within C-free



Courtesy: Shaojie Shen

Workspace and Configure Space

- In *workspace*
  - Robot has shape and size (i.e. hard for motion planning)
- In *configuration space*: C-space
  - Robot is a point (i.e. easy for motion planning)
  - Obstacle are represented in C-space prior to motion planning
- Representing an obstacle in C-space can be extremely complicated. So approximated (but more conservative) representations are used in practice.



If we model the robot conservatively as a ball with radius δ_r,
then the C-space can be constructed by inflating obstacle at all directions by δ_r.

Courtesy: Shaojie Shen

Graph Construction for Planning

Planning as Graph Search Problem

- Motion planning as a deterministic shortest path problem on a graph
  - Pre-compute the C-Space (e.g., inflate the obstacles with the robot radius)
  - Construct a graph (road map) representing the planning problem
  - Search the graph for a (close-to)optimal path

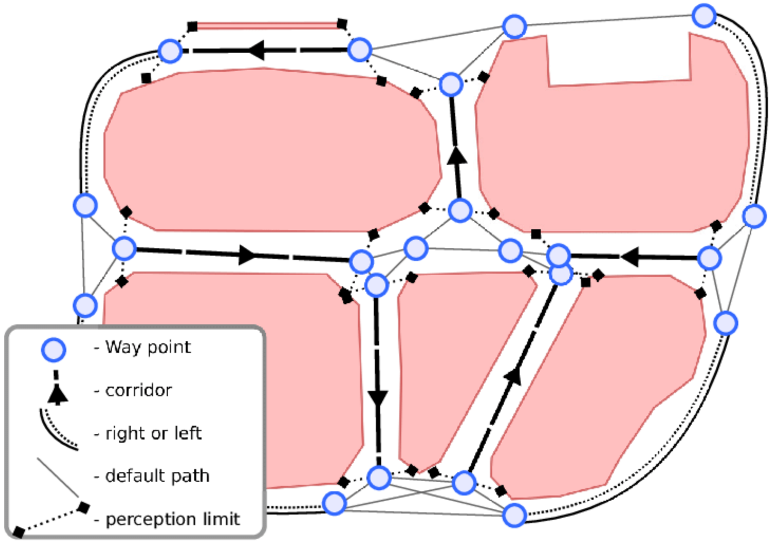
- Often collision checking, graph construction, and planning are all interleaved and performed on the fly (in short time)

- How to construct a graph as a road map?
  - different applications in different environments
  - different map representations

Courtesy:

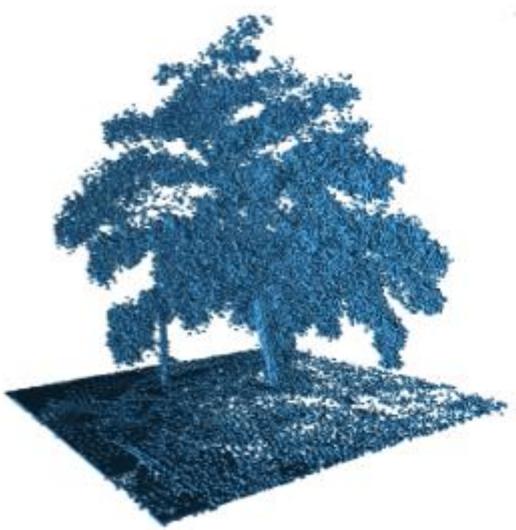
Recap L6 - Map Organizations

- We can directly apply search methods on topological map
- Node (Pose) + Edge

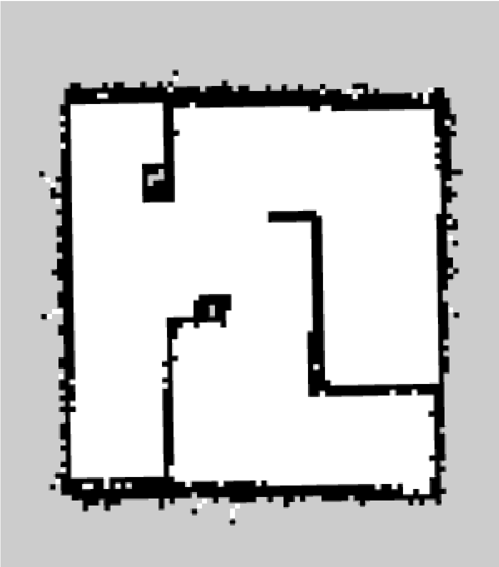


On Global Map?

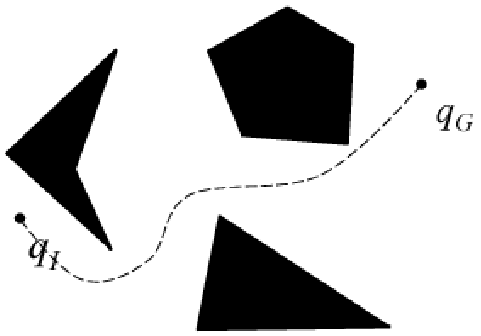
- Dense or object-level representations
- Generally, we need convert it to sparse graph first



3D Grid Map



2D Grid Map



Polygonal World (Object Map)

Courtesy:

C-Space Discretizations

- Continuous terrain needs to be discretized for path planning
- There are two general approaches to discretize C-spaces:
- Combinatorial planning
  - Characterizes C-free explicitely by capturing the connectivity of Cfree into a graph and finds solutions using search
  - Resolution completeness: the planner is guaranteed to find a path if the resolution of an underlying grid is fine enough.

- Sampling-based planning
  - Uses collision-detection to probe and incrementally search the C-space for solution
  - Probabilistic completeness: more "work" is performed, the probability that the planner fails to find a path asymptotically approaches zero.

Courtesy: Wolfram Burgard

Combinatorial Planning

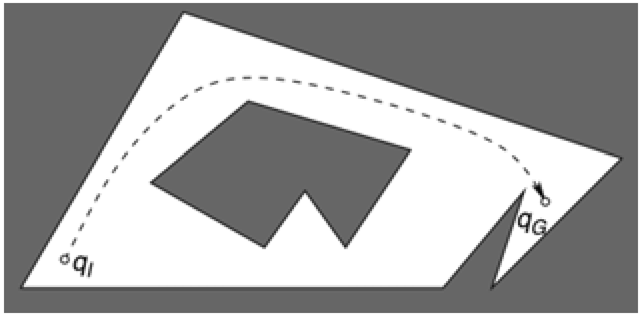
- We will first look at four combinatorial planning methods
  - Visibility graphs
  - Voronoi diagrams
  - Exact cell decomposition
  - Approximate cell decomposition

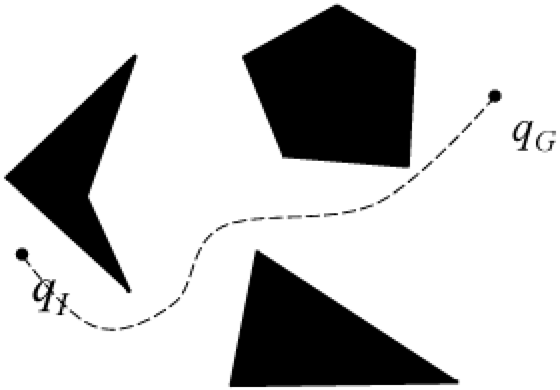
- They all produce a road map
  - **A road map is a graph** in *C-free* in which each vertex is a configuration in C-free and each edge is a collision-free path through C-free

Courtesy: Wolfram Burgard

Combinatorial Planning

- Without loss of generality, we will consider a problem in two-dimensional world with a point robot that cannot rotate.
- We further assume a polygonal world

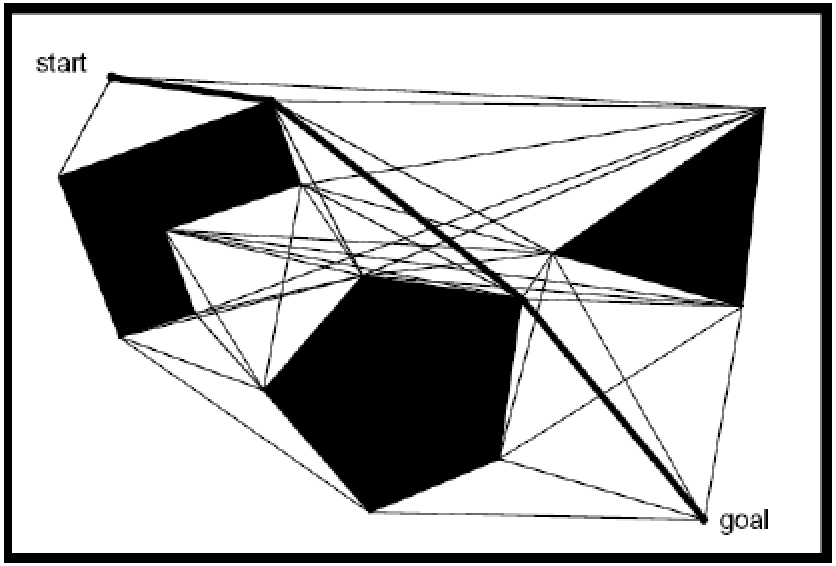




Courtesy: Wolfram Burgard

Visibility Graph

- **Idea:** construct a path as a polygonal line connection through vertices of obstacles
- One of the earliest path planning methods



Courtesy: Wolfram Burgard

Visibility Graph

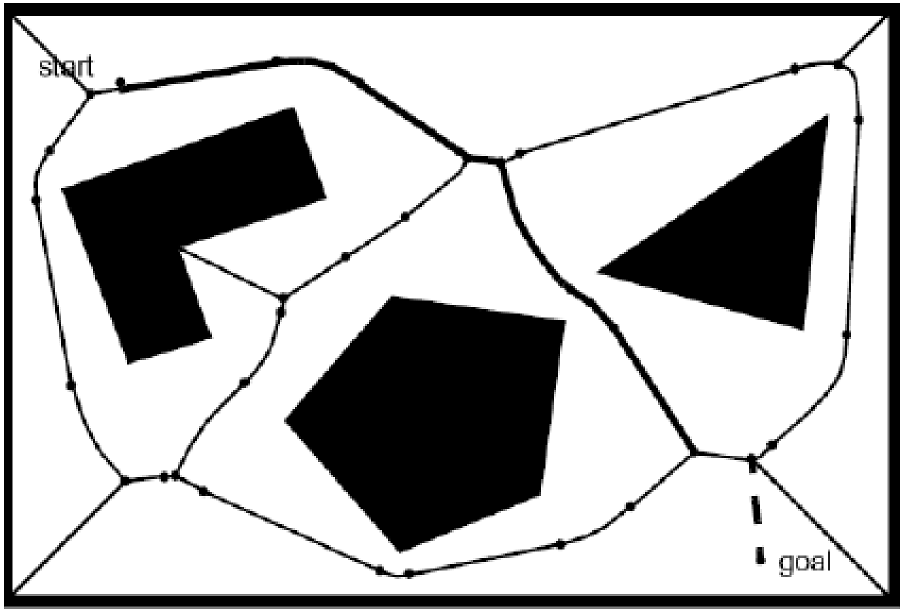
- **Pros**
- Very simple, especially when the environment is described by polygons for objects.
- Existence proof for such paths, **optimality**

- **Cons**
- The resulting path is too close to obstacles, which is not safe.

Courtesy: Wolfram Burgard

Voronoi Diagram

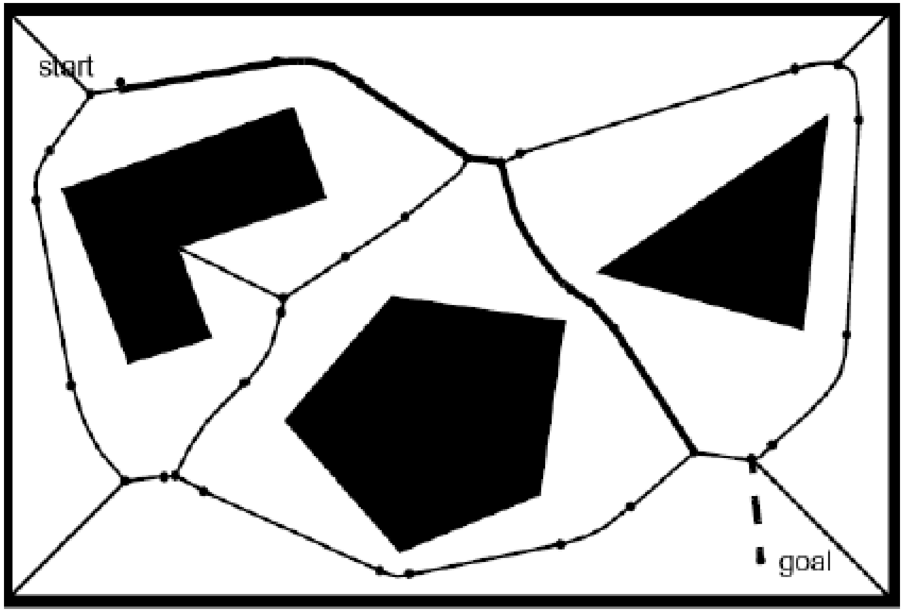
- **Idea:** Take the midpoint between obstacles to maximize the distance between the robot and the obstacles.



Courtesy: Wolfram Burgard

Voronoi Diagram - How

- For every point in C-free, calculate its distance to the nearest obstacle
- Represent the distance from the point to the obstacle by height on the plane
- When a point is equidistant from two or more obstacles, a peak appears at its distance point, and the Voronoi diagram is composed of edges connecting these peak points.



Courtesy: Wolfram Burgard

Voronoi Diagram

- Voronoi diagrams have been well studied for (reactive) mobile robot path planning

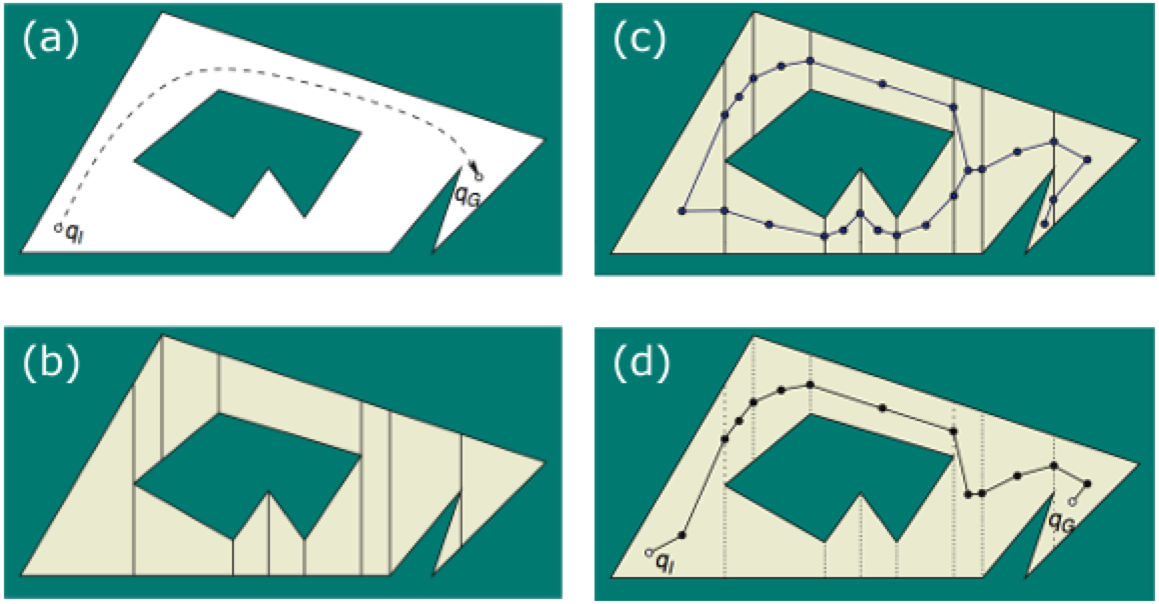
- **Pros**
- High safety
- Maximize clearance is a good idea for an uncertain robot

- **Cons**
- The calculation is complex
- Not suitable for short-distance robotic sensors

Courtesy: Wolfram Burgard

Exact Cell Decomposition

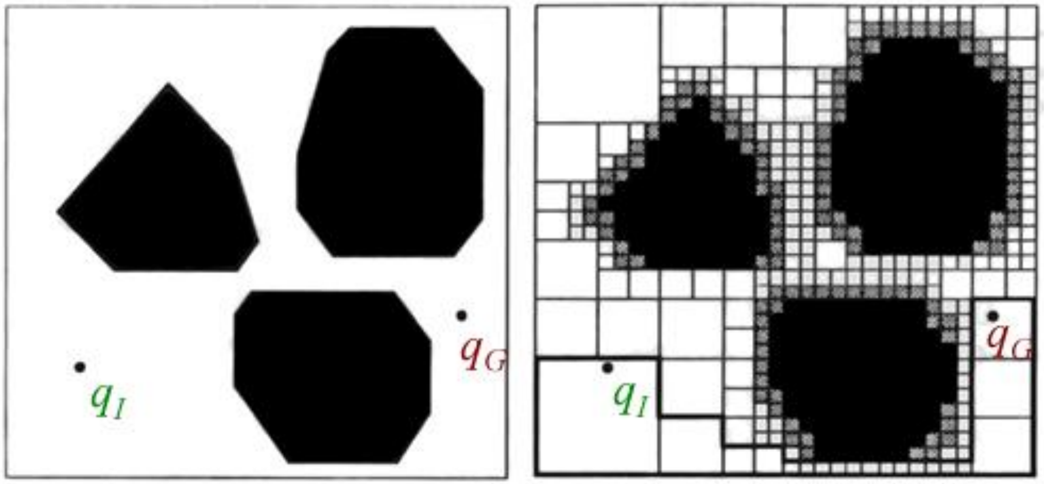
- **Idea:** decompose C-fee into non-overlapping cells, construct connectivity graph torepresent adjacencies, then search
- Need pre-defined decomposition rule



Courtesy: Wolfram Burgard

Approximate Cell Decomposition

- Exact decomposition methods can be involved and inefficient for complex problems
- **Idea:** Approximate decomposition uses cells with the same simple predefined shape



Quadtree decomposition

Courtesy: Wolfram Burgard

Approximate Cell Decomposition

- **Pros**
- Iterating the same simple computations
- Simpler to implement
- Can be made complete

- **Cons**
- Requires large storage space (in large-scale environments)

Courtesy: Wolfram Burgard, Rong Xiong

Combinatorial Planning

- Wrap Up (Short Summary)
  - Combinatorial planning techniques are elegant and complete (they find a solution if it exists, report failure otherwise)
  - But: become quickly intractable when C-space dimensionality increases
  - When rotations bring in non- linearities and make C-free a nontrivial manifold

- Use sampling-based planning Weaker guarantees but more efficient

Courtesy: Wolfram Burgard

Sampling-based Planning

- Abandon the concept of explicitly characterizing *C-free* and *C-Obs* and leave the algorithm in the dark when exploring *C-free*
- The only light is provided by a **collision detection algorithm**, that probes *C* to see whether some configuration lies in *C-free*

- We will have a look at
  - Probabilistic road maps (PRM)
  - Rapidly exploring random trees (RRT)

Courtesy: Wolfram Burgard

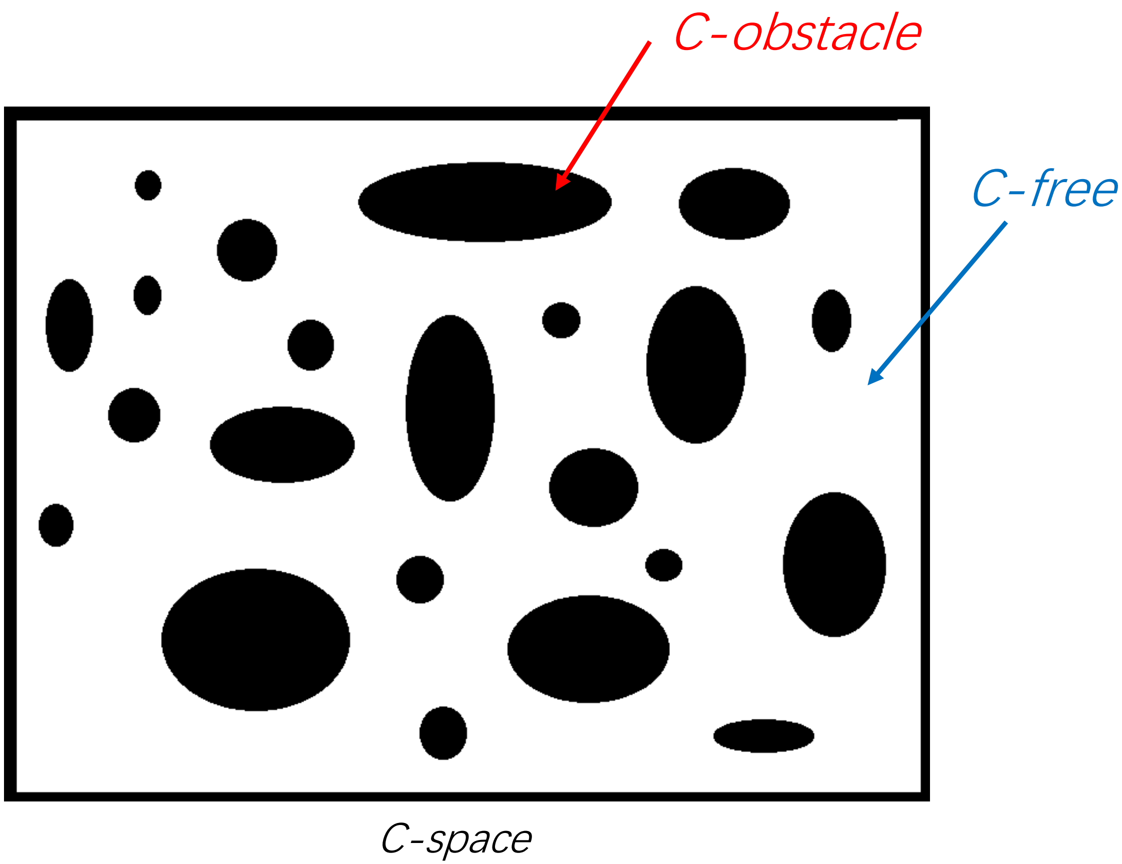
Probabilistic Roadmap (PRM)

- Basic Idea
  - Build a **graph** to characterizes the free configuration space in probabilistic manner, and then use graph search algorithm to find a path
- Algorithm
  - Initialize set of points with q_{start} and q_{goal}
  - Randomly sample points in configuration space
  - Connect nearby points if they can be reached from each other
  - Find path from q_{start} to q_{goal} in the graph
- Step by step illustration as follows

Courtesy: Shaojie Shen

Probabilistic road maps (PRM)

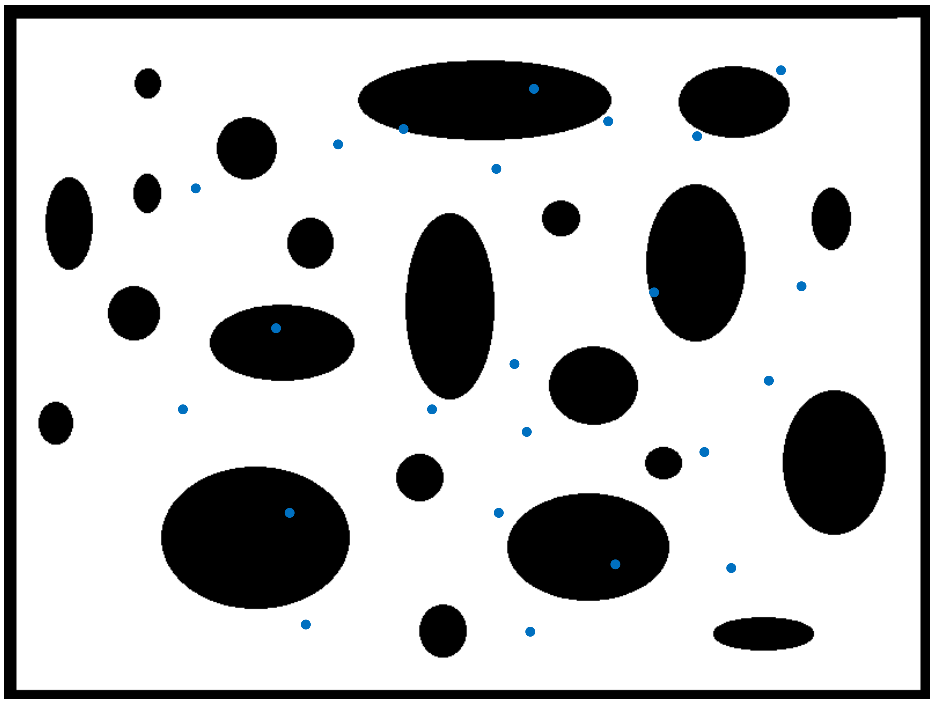
- Free space and obstacle space



Courtesy: Shaojie Shen

PRM

- Configurations are sampled by picking each coordinate at random.

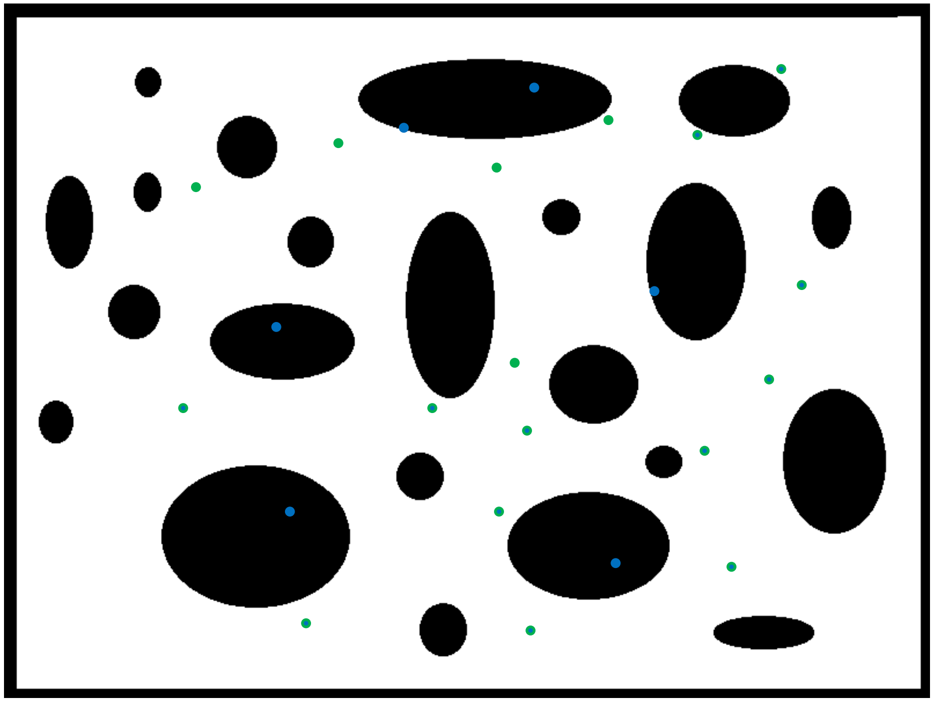


C-space

Courtesy: Shaojie Shen

PRM

- Sampled configurations are tested for collision.

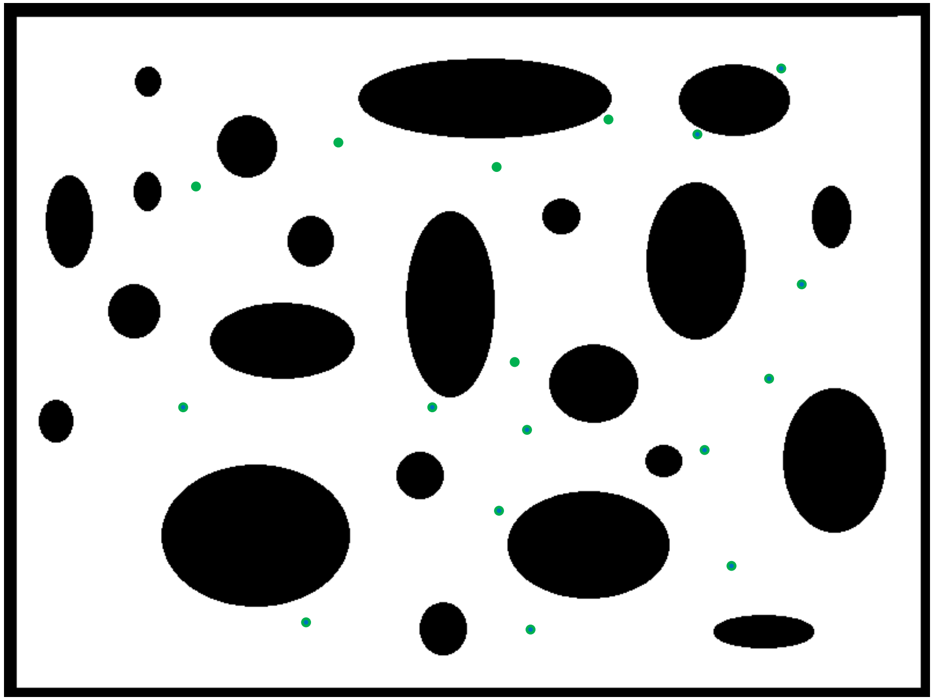


C-space

Courtesy: Shaojie Shen

PRM

- The collision-free configurations are retained as milestones.

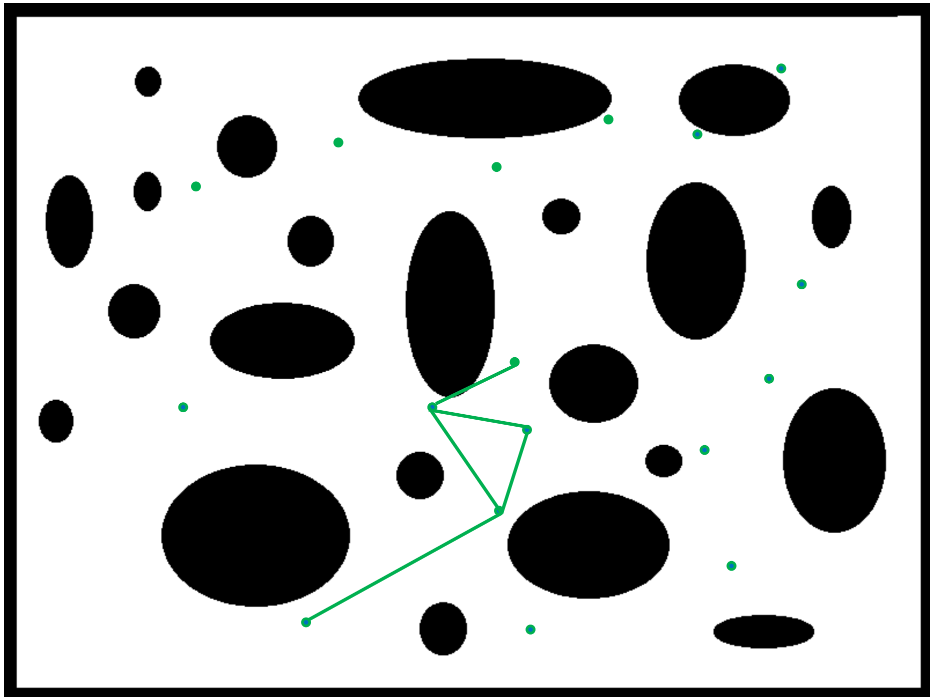


C-space

Courtesy: Shaojie Shen

PRM

- Each milestone is linked by straight paths to its nearest neighbors.

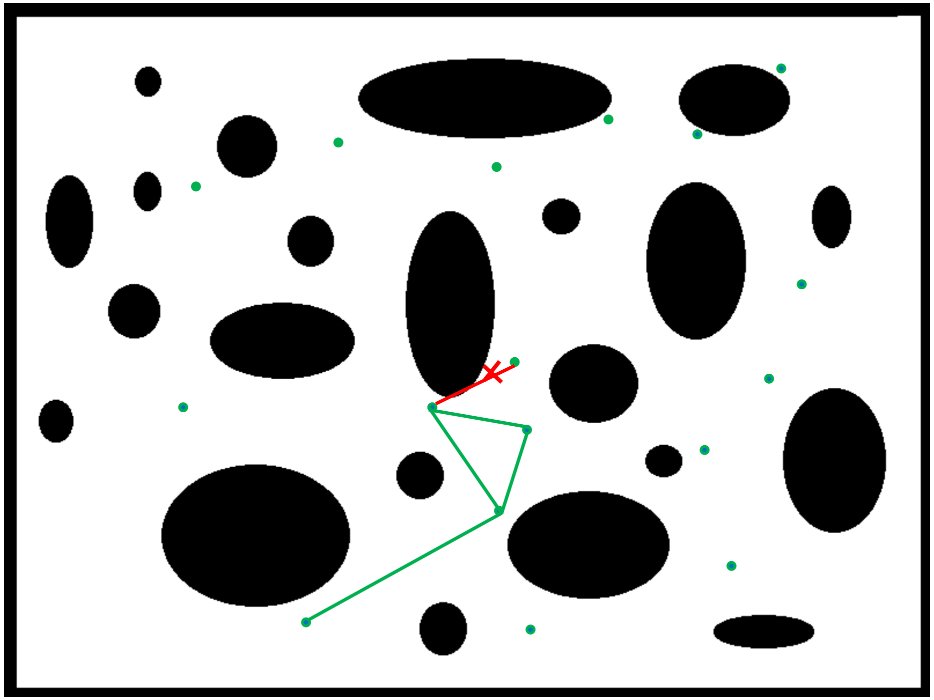


C-space

Courtesy: Shaojie Shen

PRM

- Eliminate collision links.

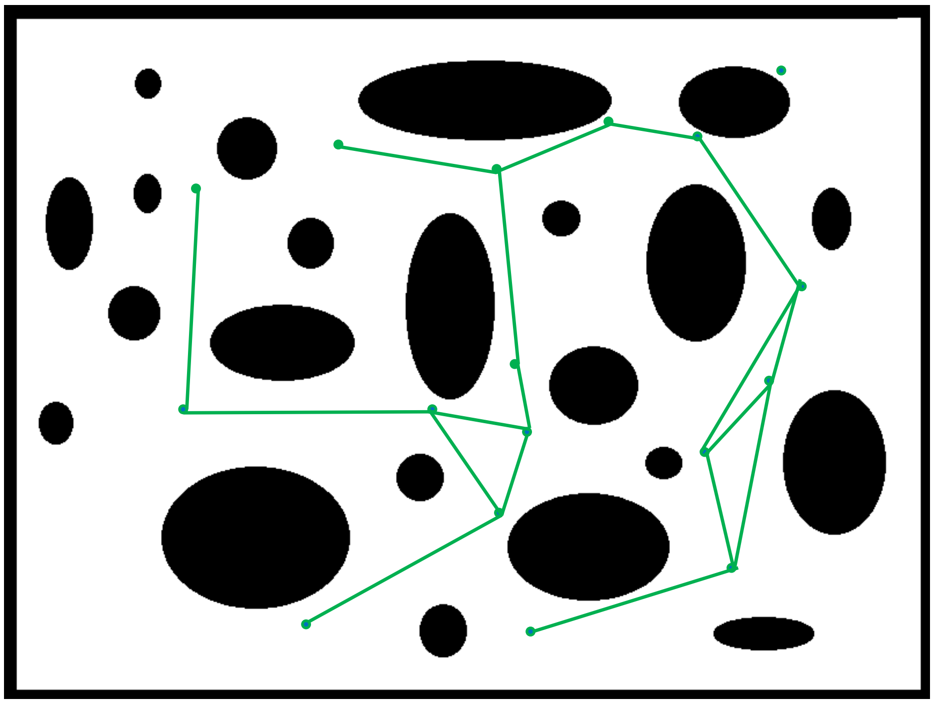


C-space

Courtesy: Shaojie Shen

PRM

- The collision-free links are retained as local paths to form the PRM.

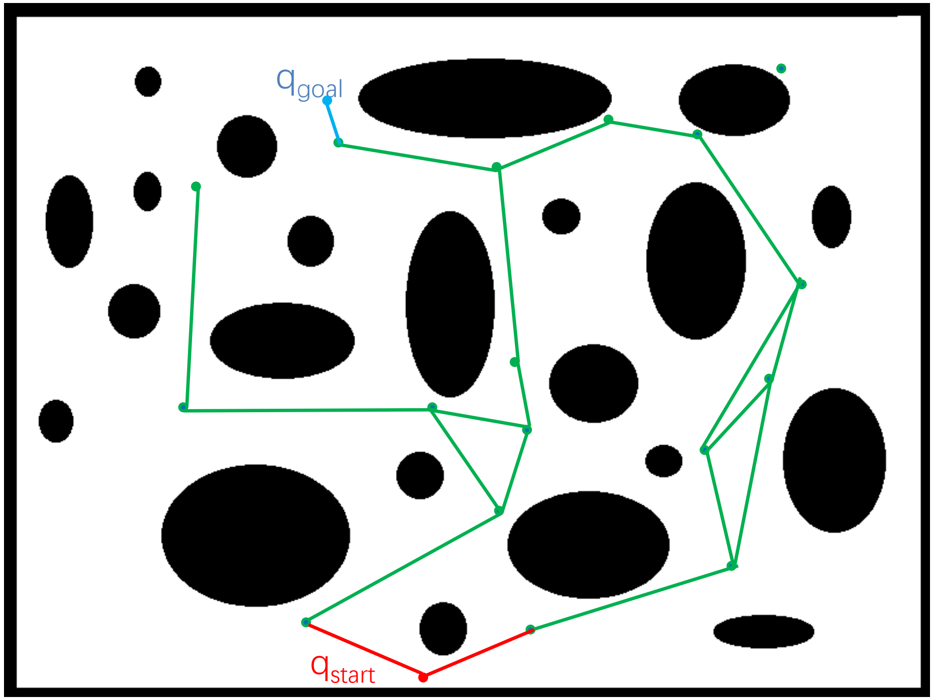


C-space

Courtesy: Shaojie Shen

PRM

- Connect the start and goal point to the roadmap.

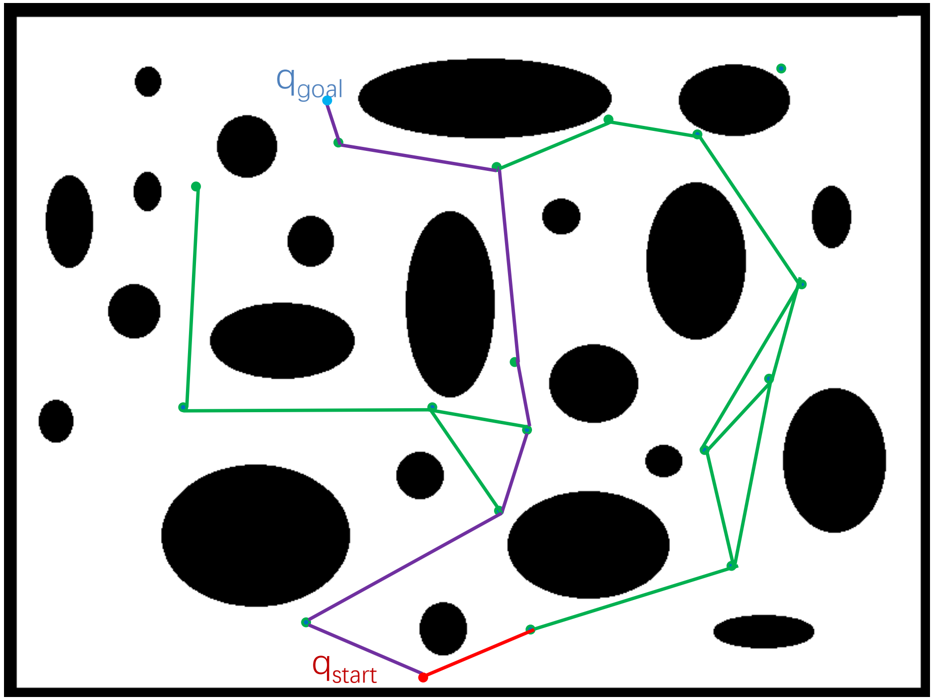


C-space

Courtesy: Shaojie Shen

PRM

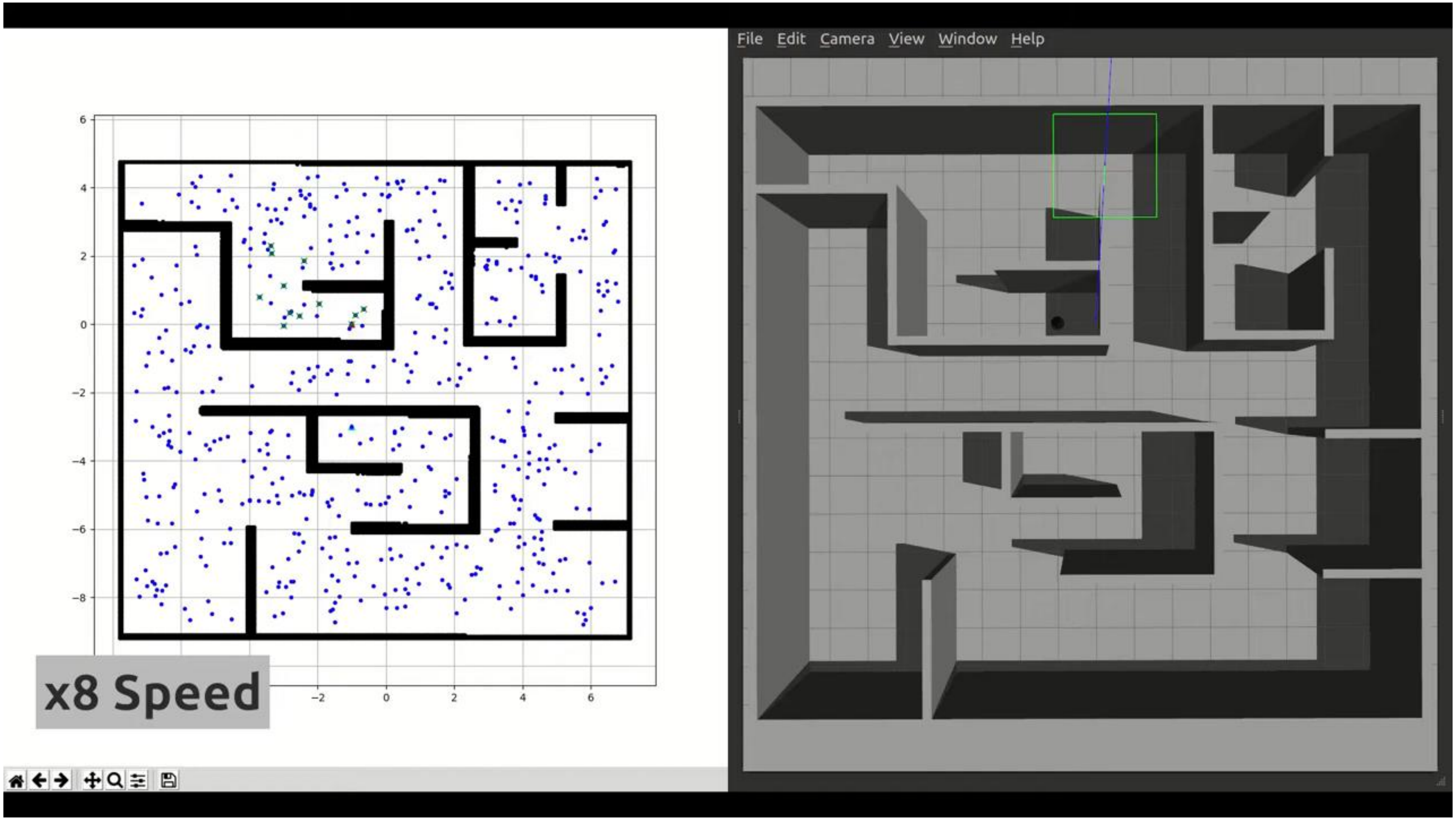
- Search the roadmap for a path from start to goal point (e.g. A* algorithm).



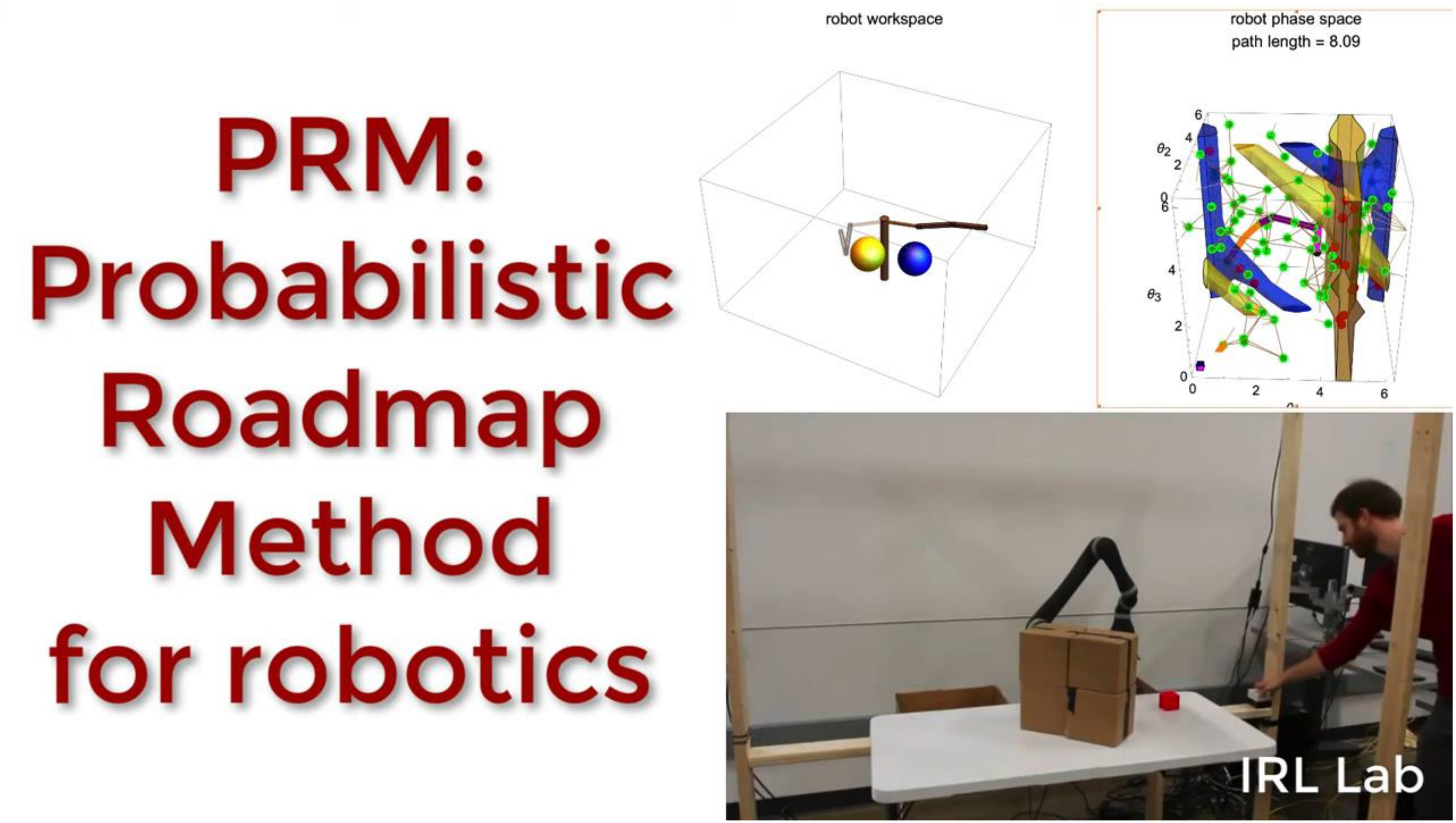
C-space

Courtesy: Shaojie Shen

PRM



PRM for Robotic Manipulator



Courtesy: YouTube

PRM’s Pros and Cons

- Pros:
  - Probabilistically complete: i.e., with probability one, if run for long enough the graph will contain a solution path if one exists.
  - Can cope with high-dimensional system

- Cons:
  - Collision detection takes majority of time
  - Suboptimal solution if only limited samples are given
  - Need the whole C-space as prior condition

Today's Summary

- Basic concepts of Motion Planning
  - workspace space
  - configuration space
  - global/local planner (path planning) + trajectory planning
- Planning as graph search problem
- Combinatorial Planning
  - Visibility Graph
  - Voronoi Diagram
  - Exact/ Approximate Cell Decomposition
- Sampling-based Planning
  - Probabilistic road maps (PRM)

Next Lecture

- Another Sampling-based Planning
  - Rapidly exploring random trees (RRT)
- Search on the road map
  - Dijkastra
  - A*