# CS123

Programming Your Personal Robot

Part 3: Reasoning Under Uncertainty

# This Week (Week 2 of Part 3)

- Part 3-3
  - Basic Introduction of Motion Planning
  - Several Common Motion Planning Methods
  - Plan Execution
  - Planning Under Uncertainty
  - HW #3-2
- Part 3-4
  - Other Motion Planning Methods
  - Search (in particular A*)
  - Opportunity for Student Demo (Race?)
  - Talk about Final Project
    - Logistics (form team, submit proposal)
    - Two Suggested Projects

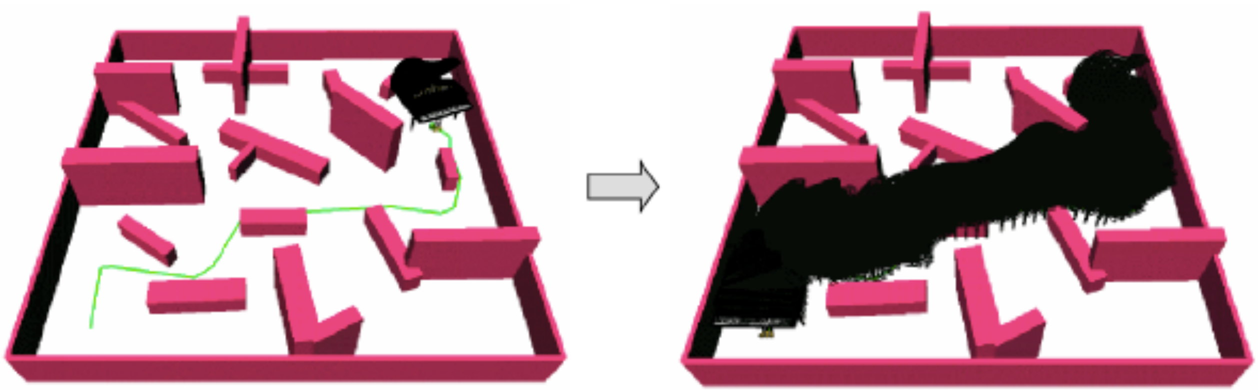
# 3.3 Robot Motion Planning and Control Under Uncertainty

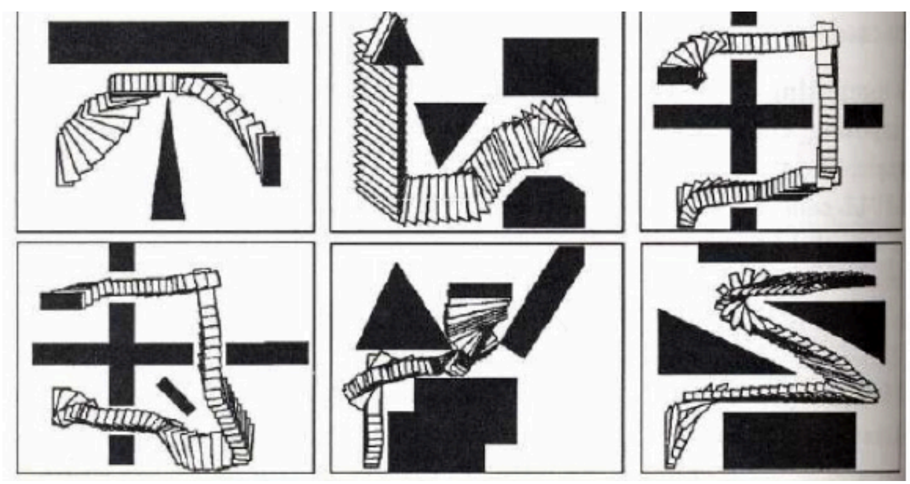
# Topics

- Introduction to Robot Motion Planning
  - Configuration Space (C-Space) Approach
  - Basic Motion Planning Methods
- Plan Execution (Control)
  - Virtual World (Perfect Control)
  - Real World (Uncertainty in control)
- Planning Under Uncertainty
- Homework Assignment Part # 3-2

# What is Motion Planning

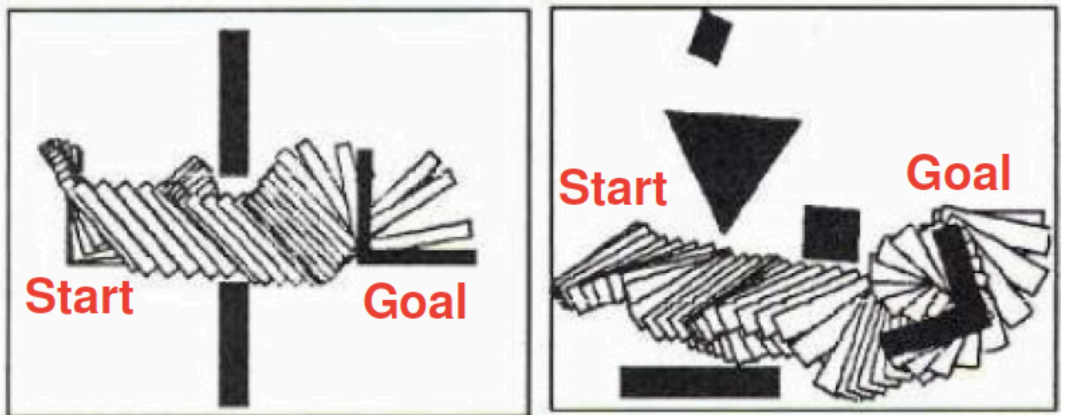
- Also known as the **Piano Mover**'s Problem





# Problem Formulation

- The problem of motion planning can be stated as follows
  - A start pose of the robot
  - A desired goal pose
  - A geometric description of the robot
  - A geometric description of the world
- Find a path that moves the robot
  - from start to goal while
  - never touching any obstacle



# Why Motion Planning For Robot

- A robot needs to move to accomplish task. Such movement should be "purposeful" (with respect to a given task or goal)"
- A robot's ability to plan its movement is critical for it to be autonomous
- A vast research area
- A lot has been accomplished, but still a lot more to be done


J.-C. Latombe (1991):

"...eminently necessary since, by definition, a robot accomplishes tasks by moving in the real world."

# Examples of Motion Planning

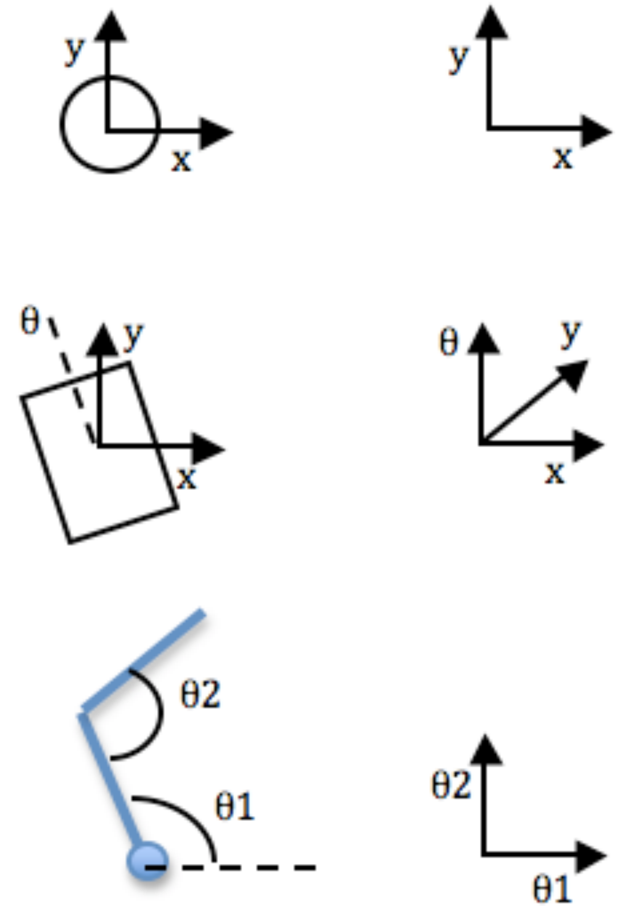
- Mobile Robots
- Manipulator (Arms)


# Formal Definition of Motion Planning

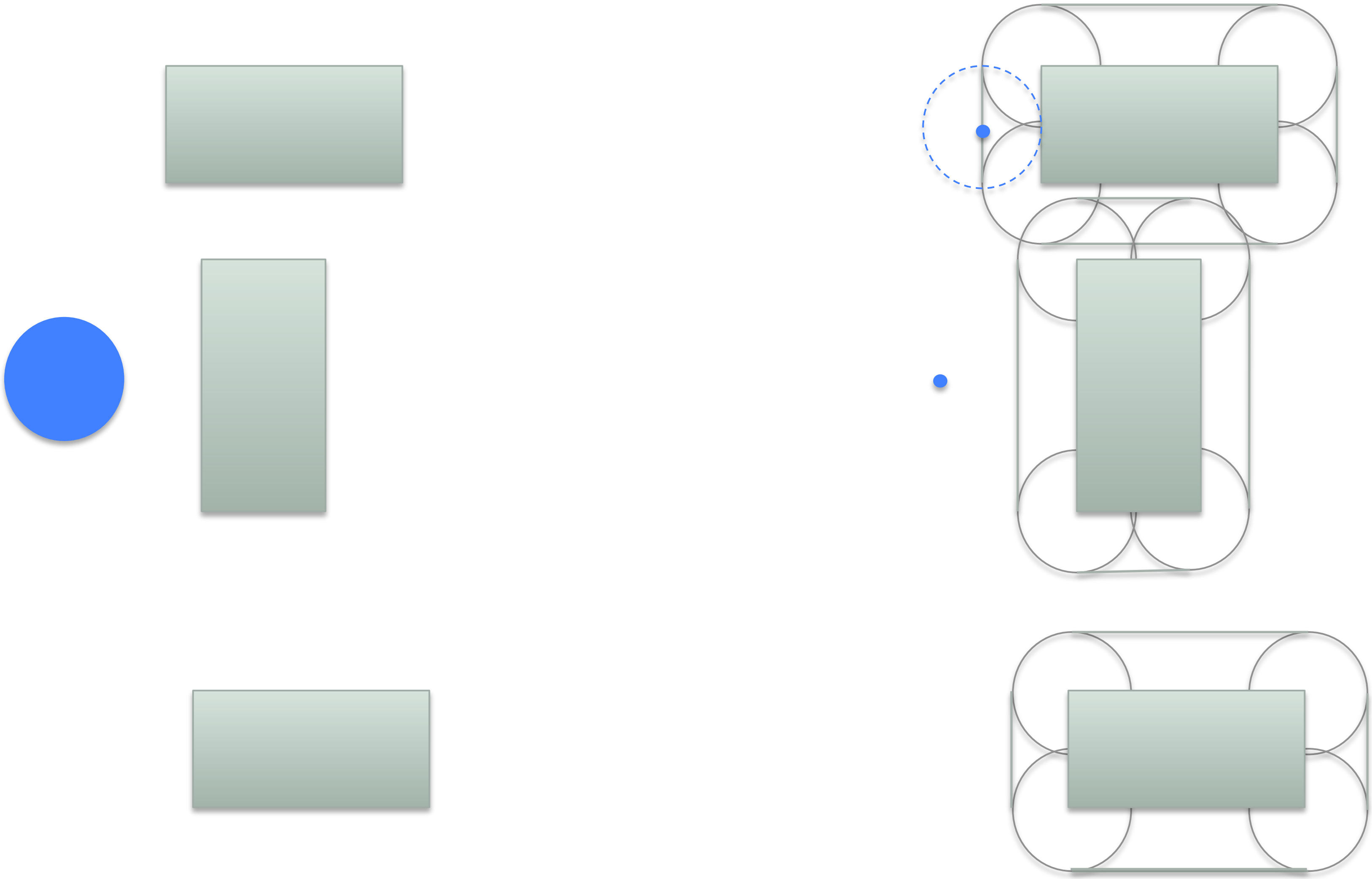
- **Configuration Space (C-Space) Approach** – Mapping the geometry of the task into configuration space allows us to transform the problem of planning the motion of a dimensioned object into that of planning the motion of a point, P
  - First Introduced by Lozano Perez (MIT), 1980
- Robot reduced to a **point** in C-Space – Obstacles mapped into C-Obstacles (in C-Space)
- Finding a path in the free C-Space that connects the Start Configuration to the Goal Configuration

# Configuration of A Robot

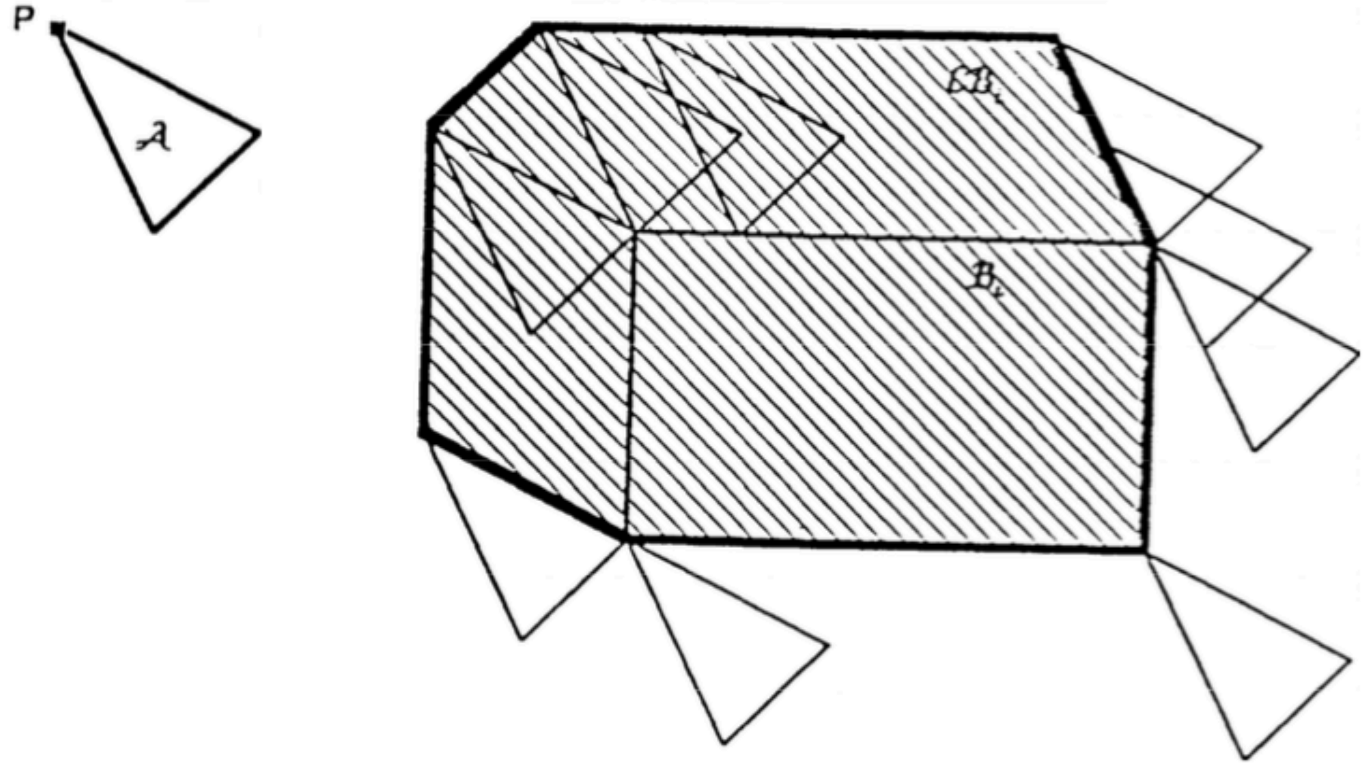
- Configuration of a robot: It is the precise specification of all of the robot's degrees of freedom (DOFs).



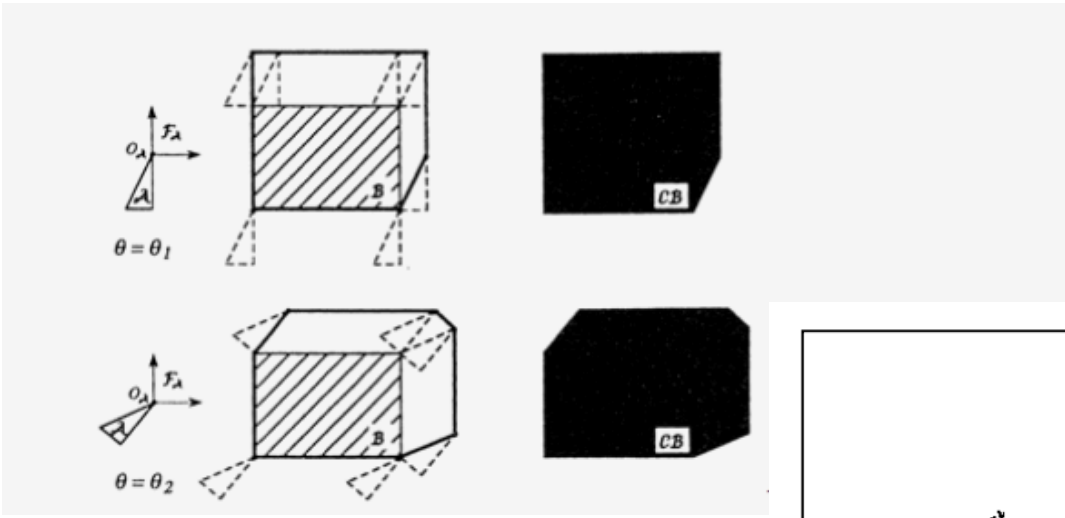
# Example of 2D Circular Robot

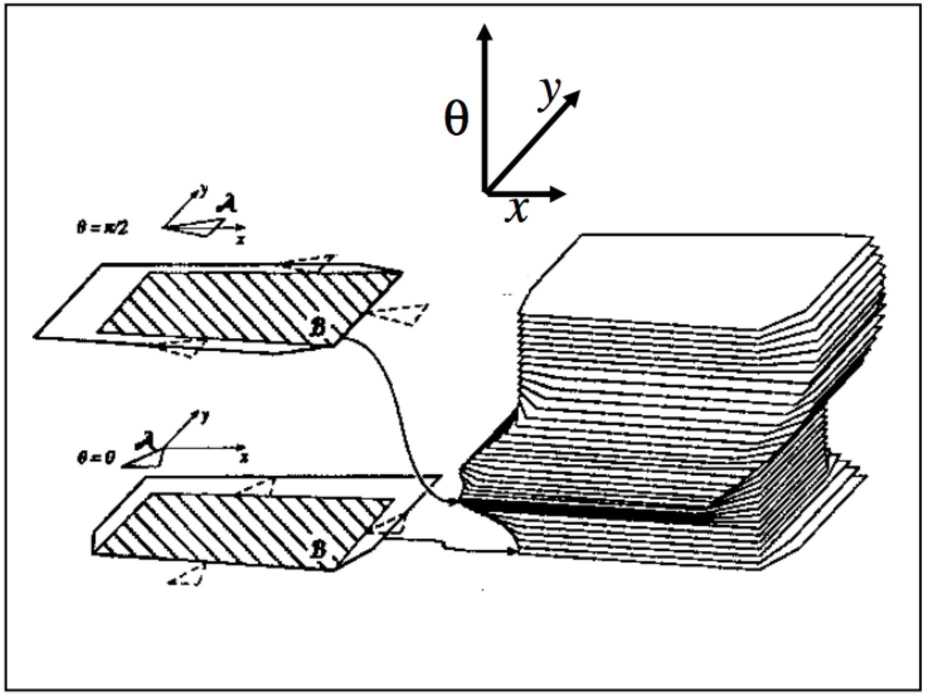


# 2D Polygonal Object without Rotation

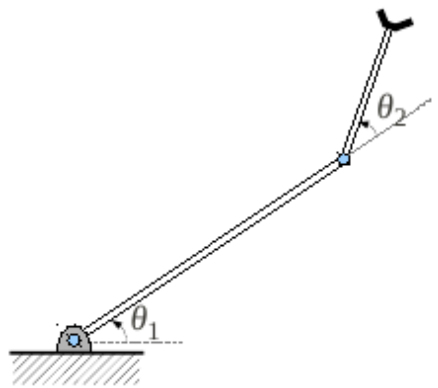


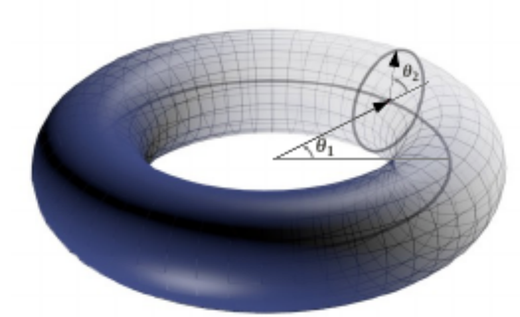
# 2D Polygonal Object with Rotation

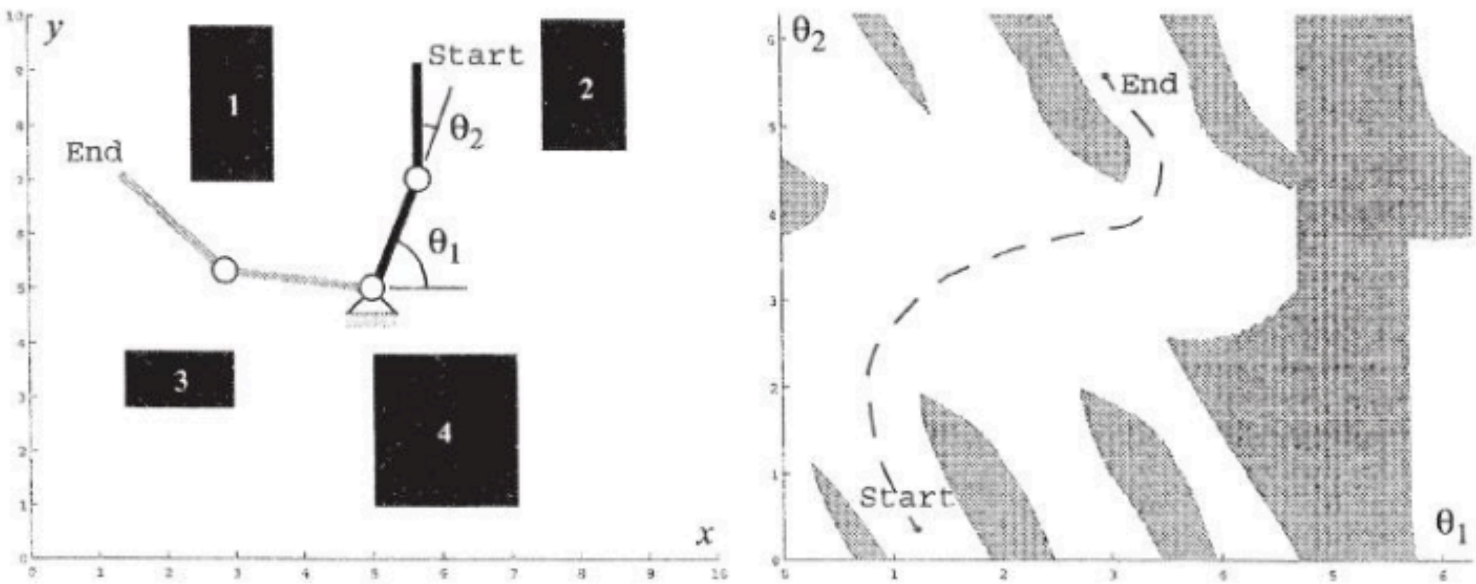




# 2D Arm (2-links)



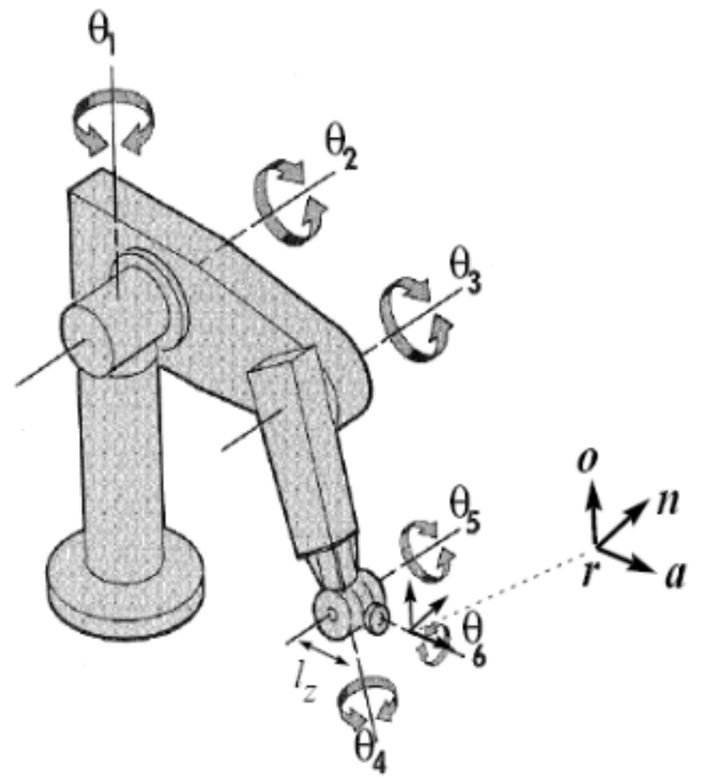




a) b)

# Challenges of Motion Planning

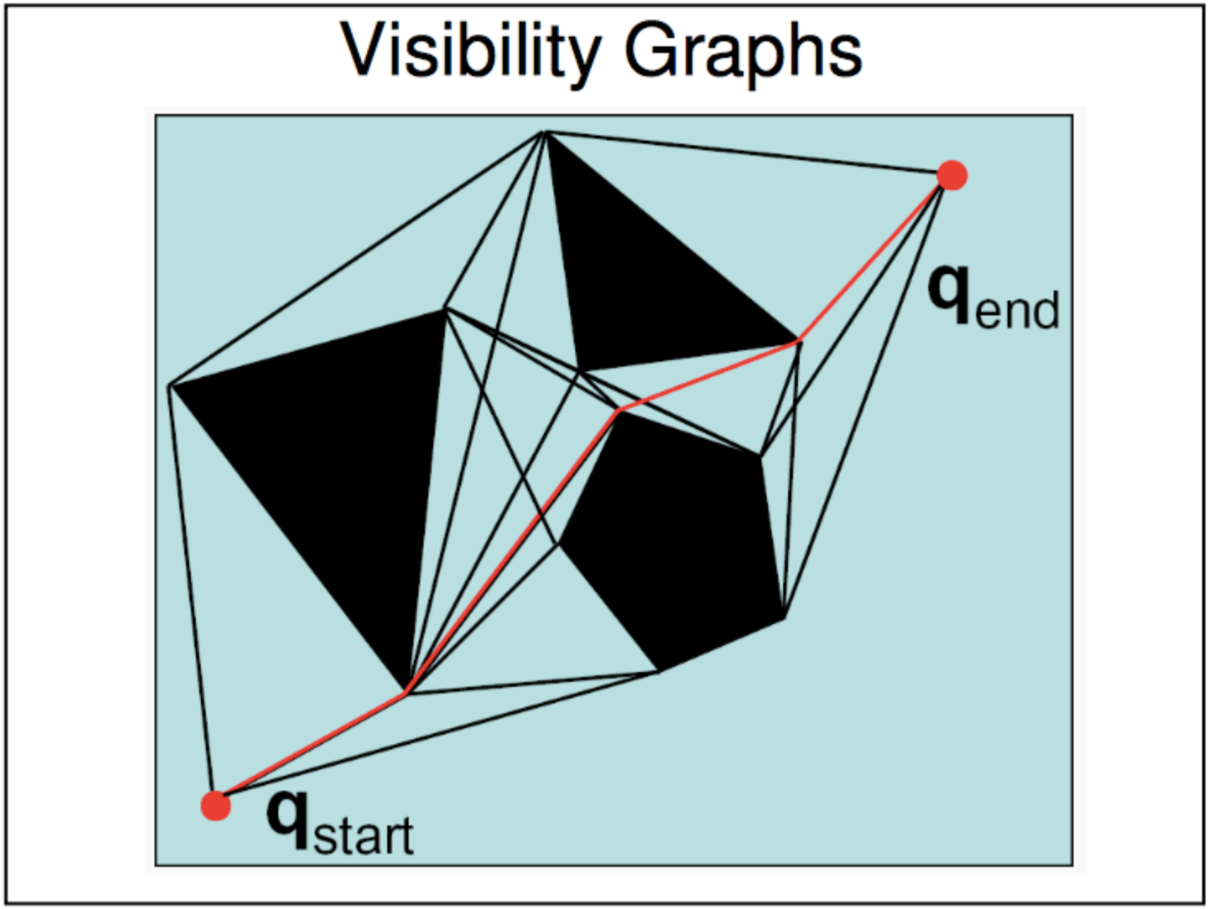
- Continuous space and high DOF's
- 3D "free-flying" rigid object : 6 DOF
- Puma Arm : 6 DOF



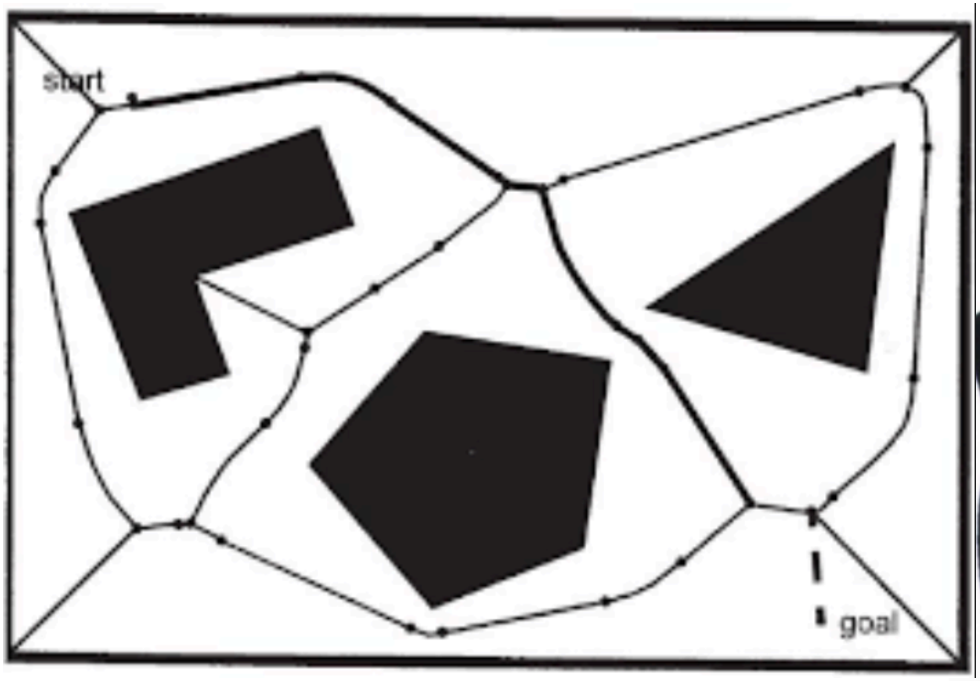
# Motion Planning Methods

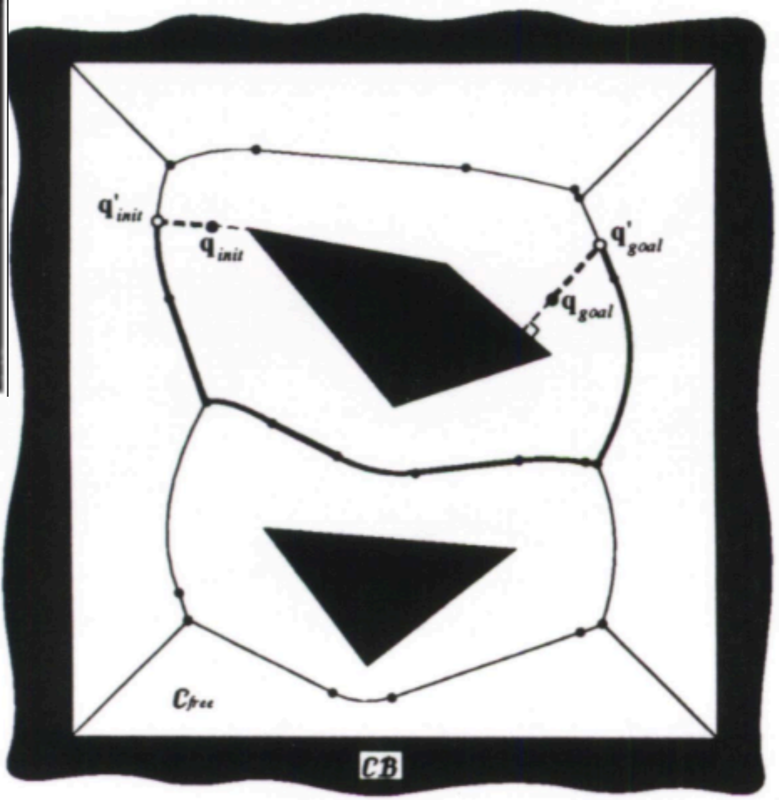
- Converting a “continuous” space problem into a discrete graph search problem (discretization of C-space)
- Decouple “independent” DoF
  - mobile vs. manipulatiom
- We will focus on planning problem of mobile robots
- Visibility Graph
- Voronoi Diagrams
- Cell Decomposition
  - Exact
  - Approximate

# Visibility Graph

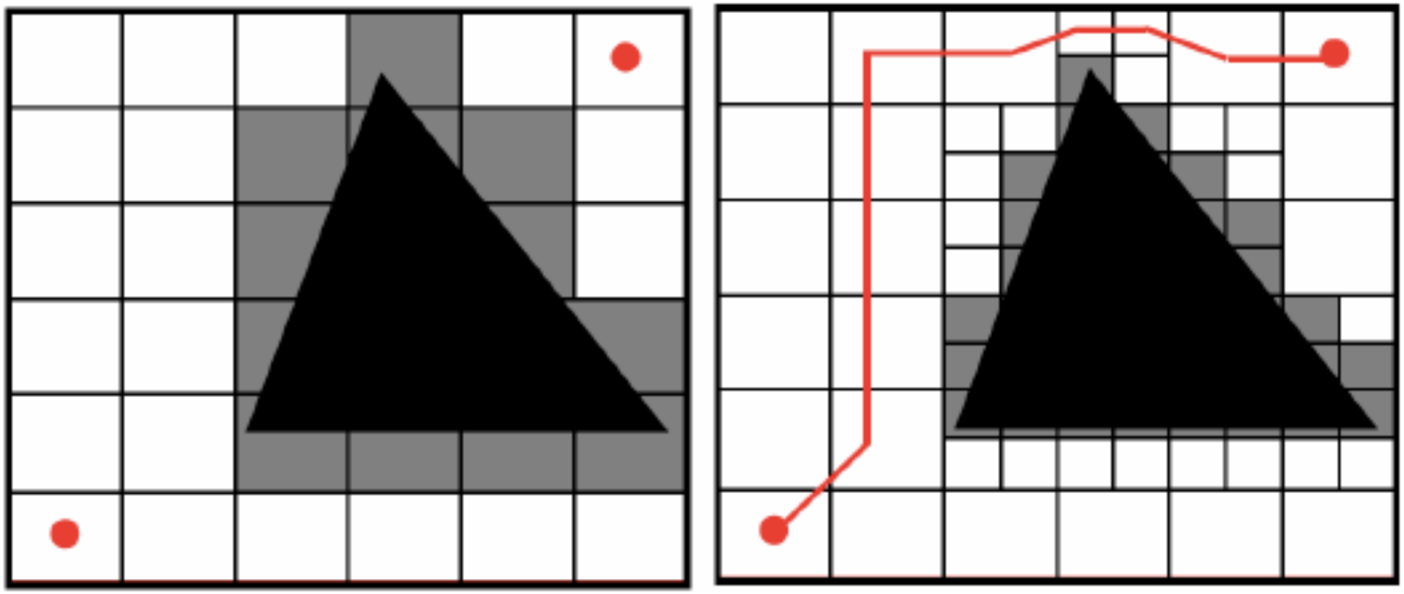


# Voronoi Diagrams

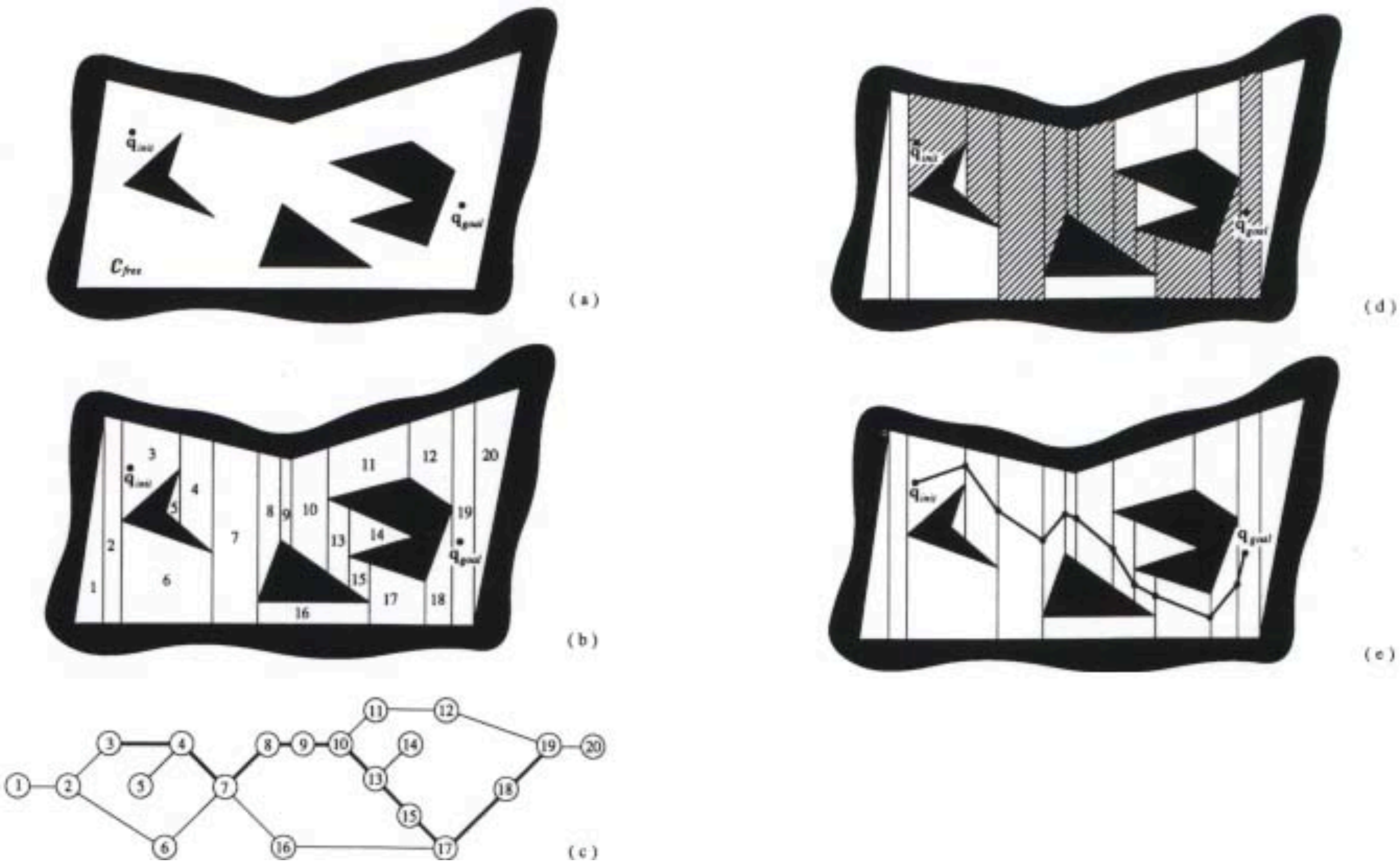


# Cell Decomposition : Approximate



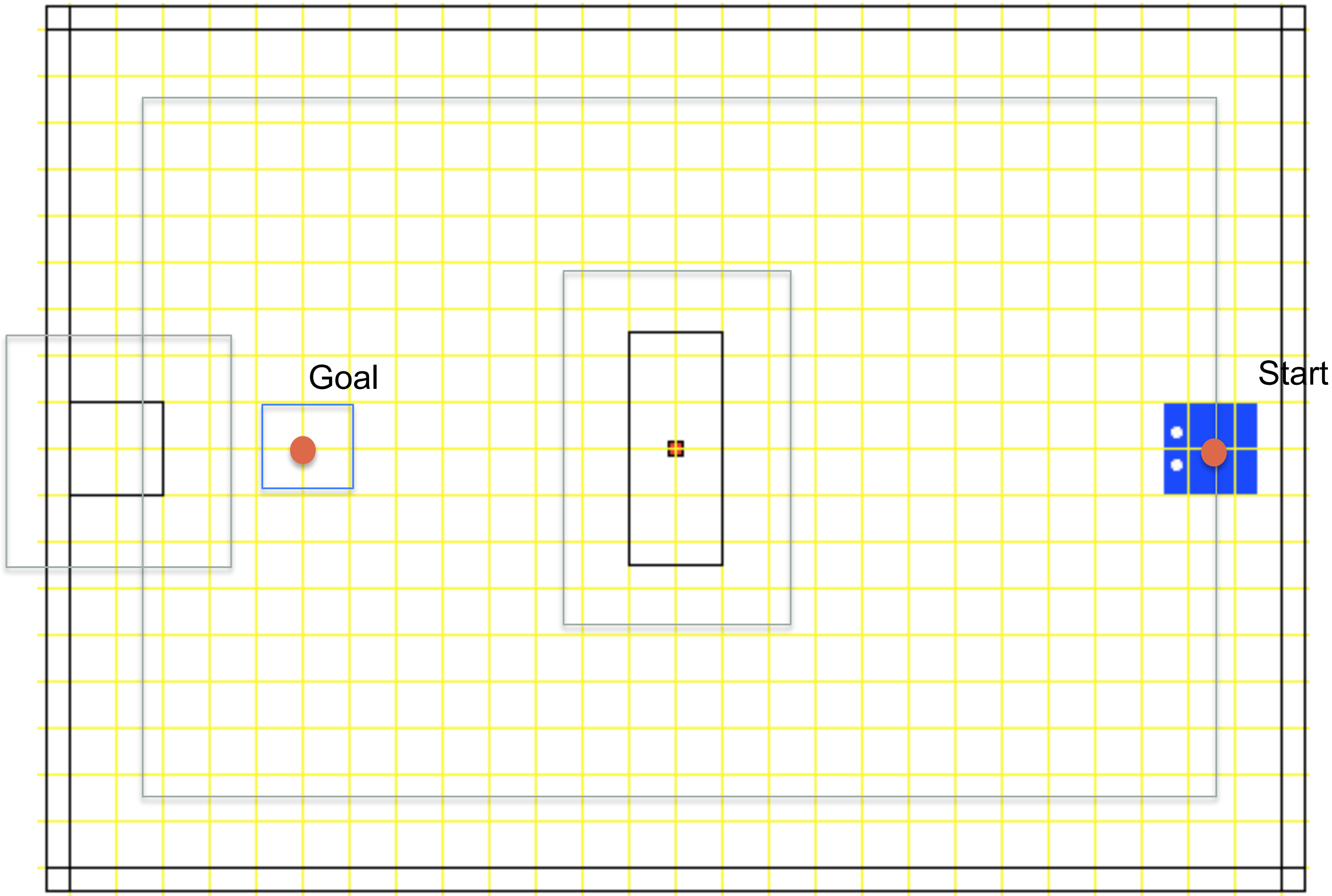
# Cell Decomposition : Exact



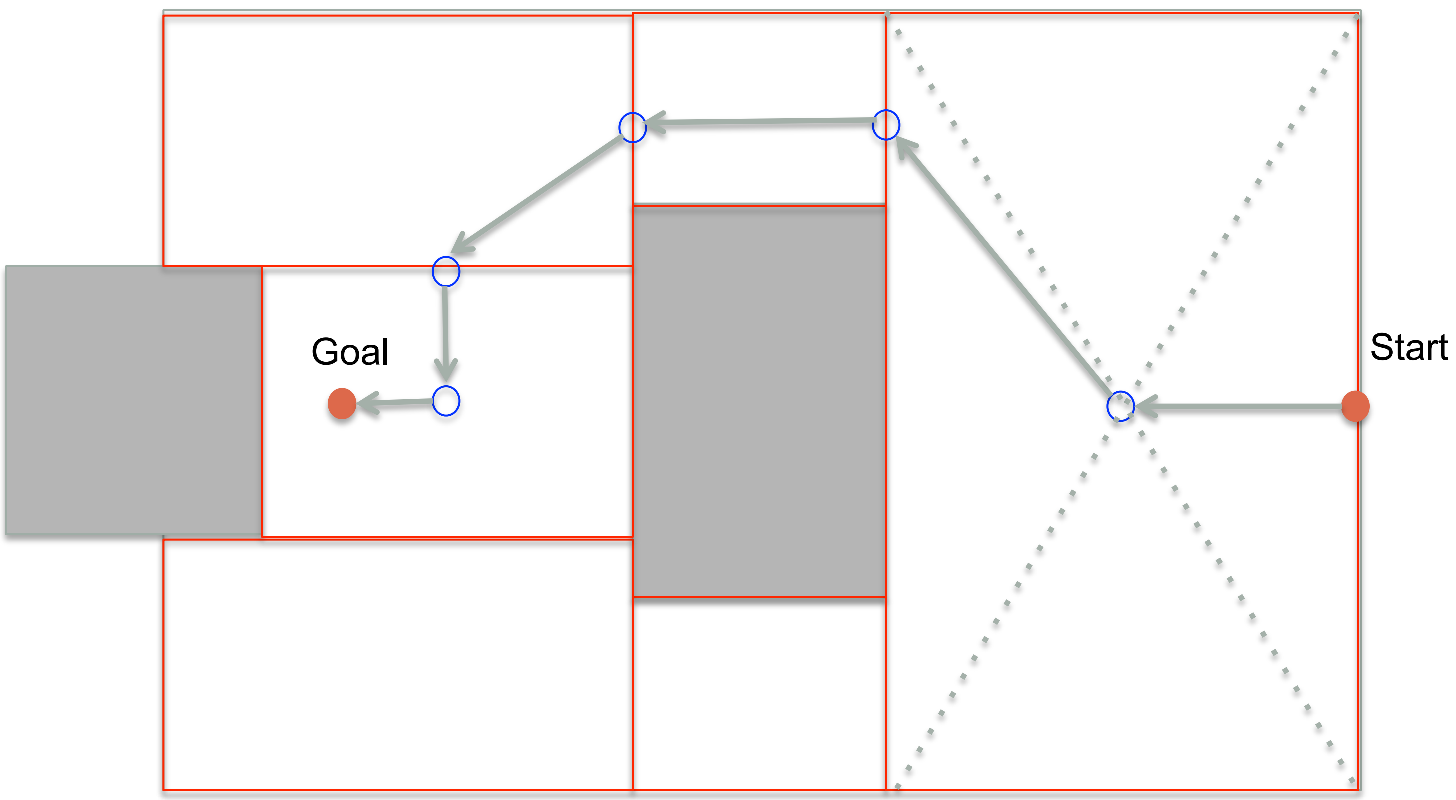
# Simplify Hamster's Simple World

- We approximate Hamster as its Circumscribing Circle (we assume Hamster is a 40mm x 40 mm Square)
- Approximate the C-space obstacles by their bounding rectangle

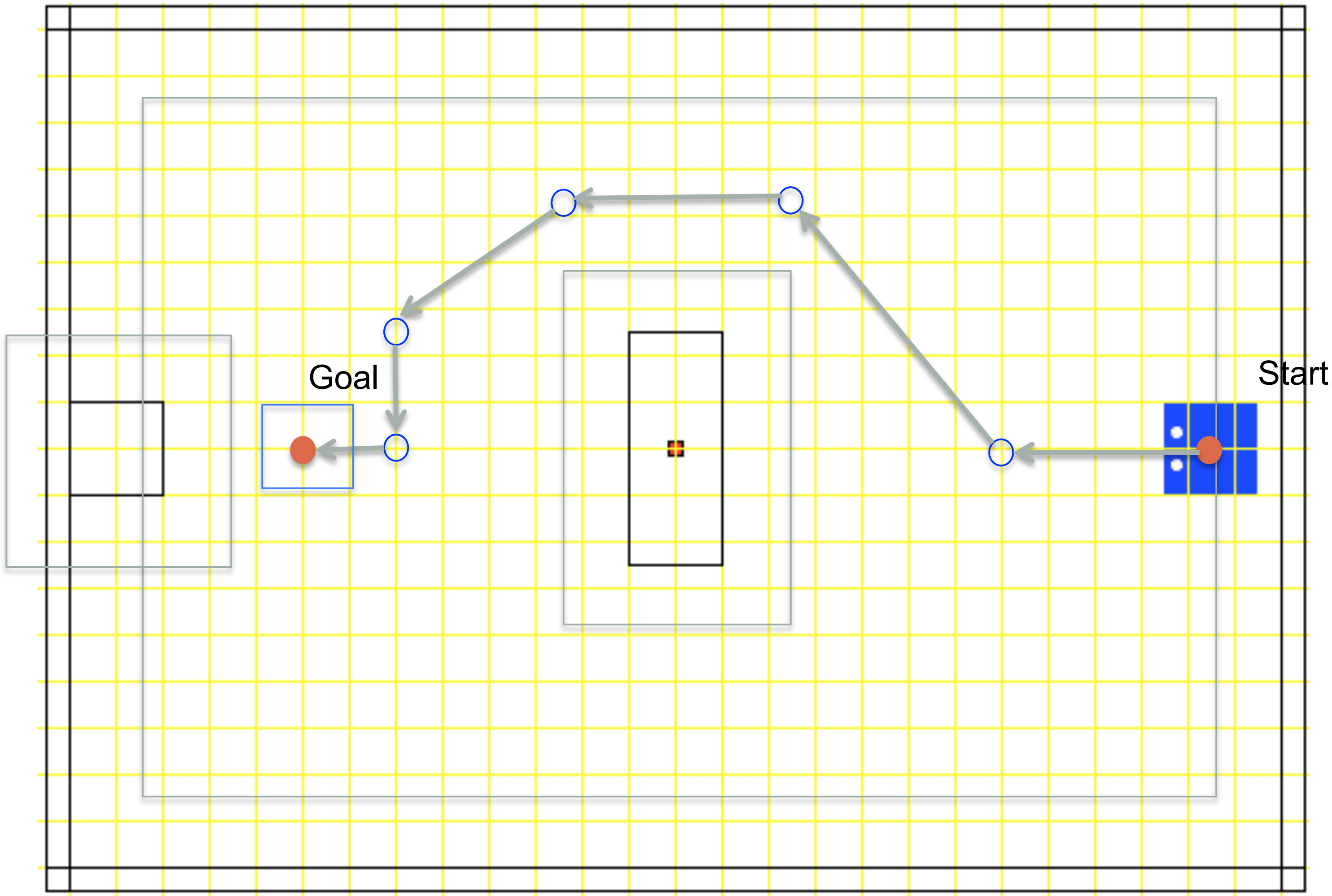
# A Simple Work Space / C-space



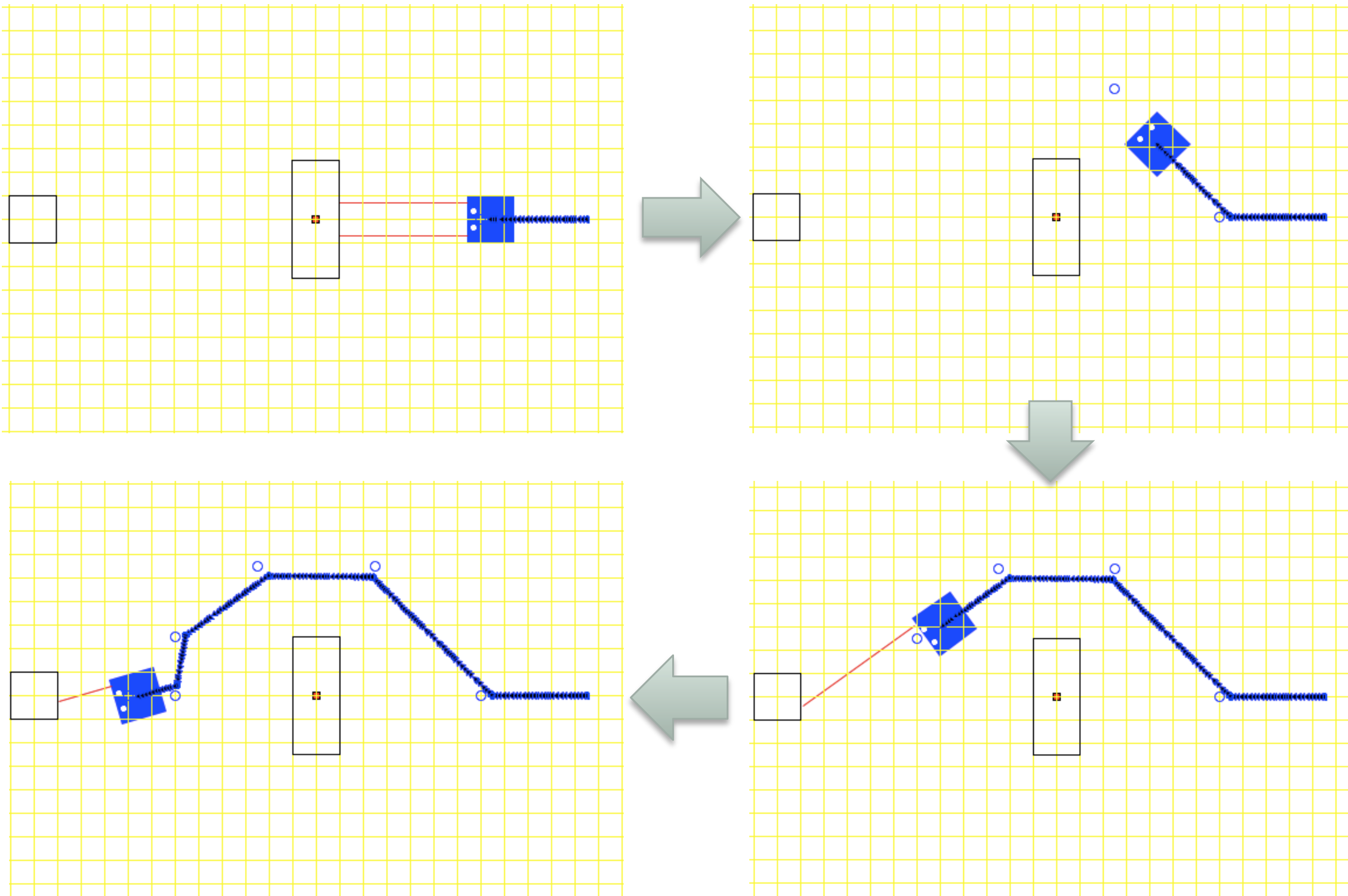
# Simple Motion Plan For Hamster Using Exact Cell Decomposition



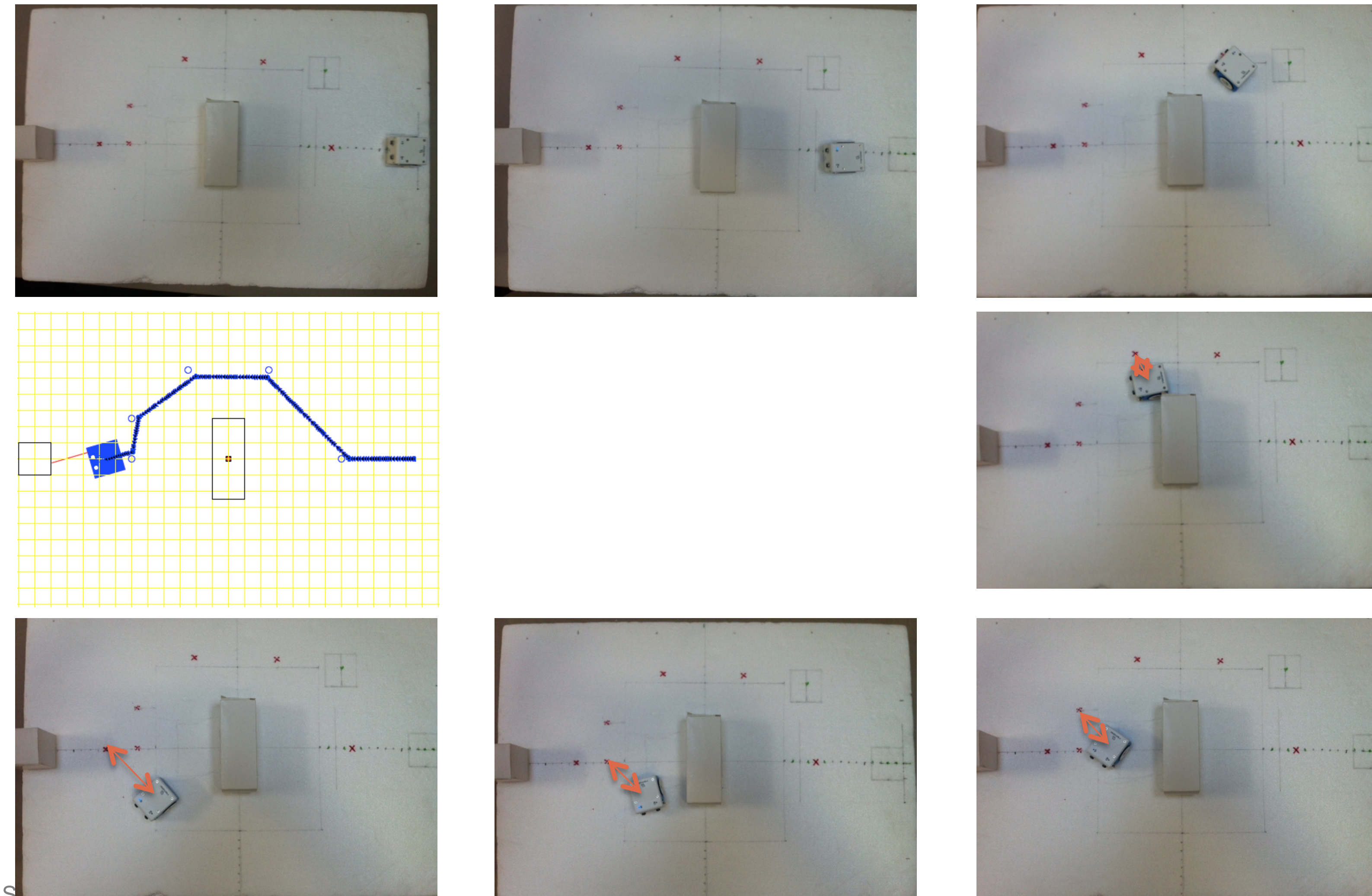
# Path in Work Space



# Plan Execution In A Perfect (Virtual)  World



# Plan Execution In Real World

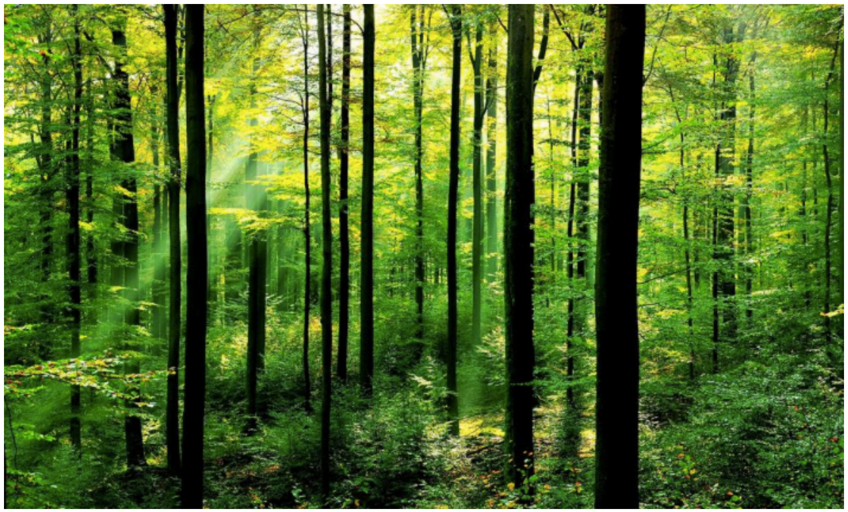


# Control Error Propagation

# Motion Planning With Uncertainty

- Classical path planning methods, which use simple geometric models while assuming null uncertainty, are clearly insufficient.
- Taking uncertainty into account at planning time is essential when potential control errors are comparable to or larger than the tolerances allowed by the task.

# Importance of “Landmarks”



# “Landmarks” Helps Navigation

Landmarks are every where and we (human) use landmarks extensively for our navigation often without realizing it

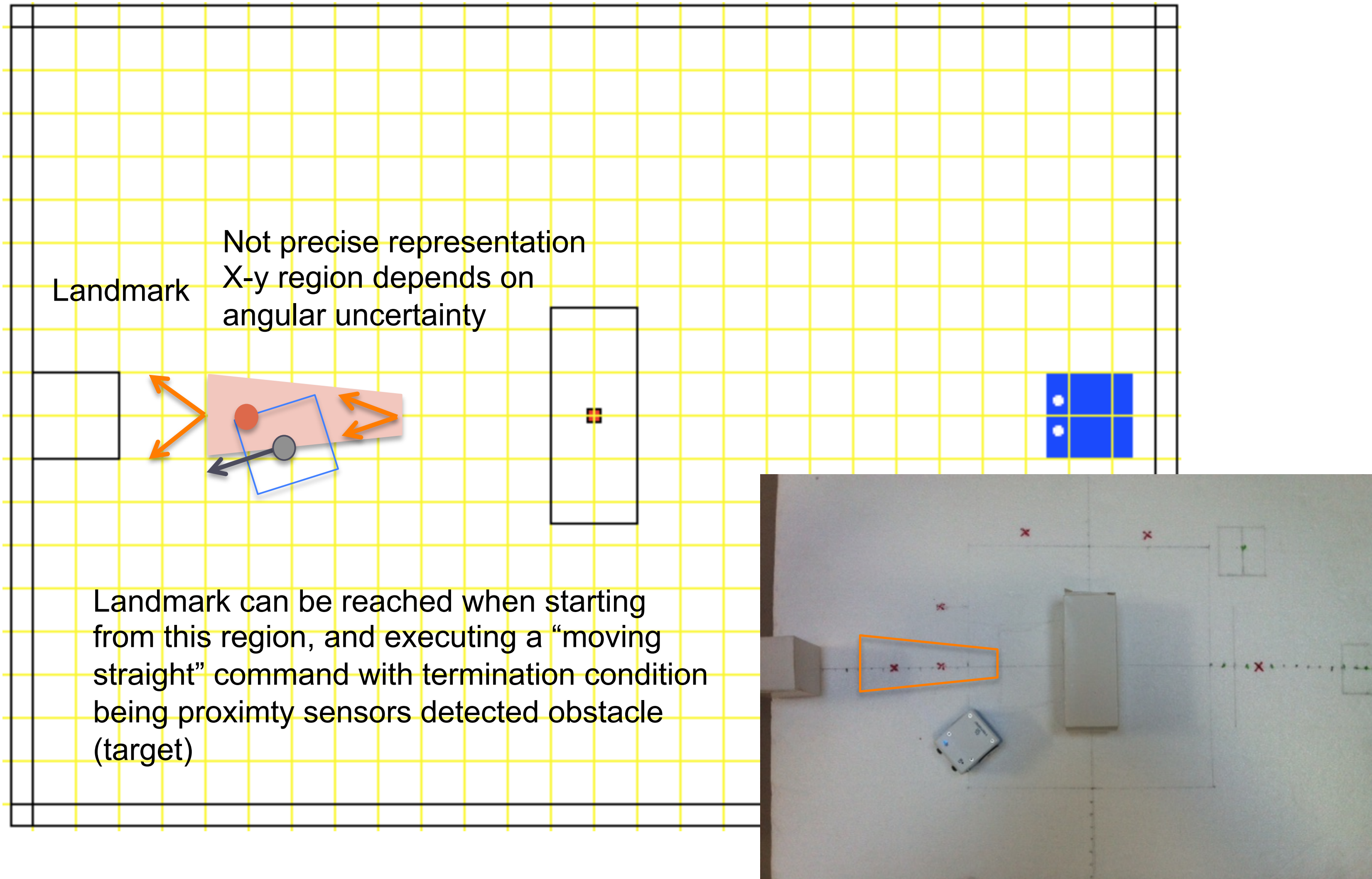



# Use of Landmarks

- Must get “close enough” to a landmark
- And has a method to “search” for the landmark


First get to the block where the restaurant is, and then walk up/down street looking for the name/sign

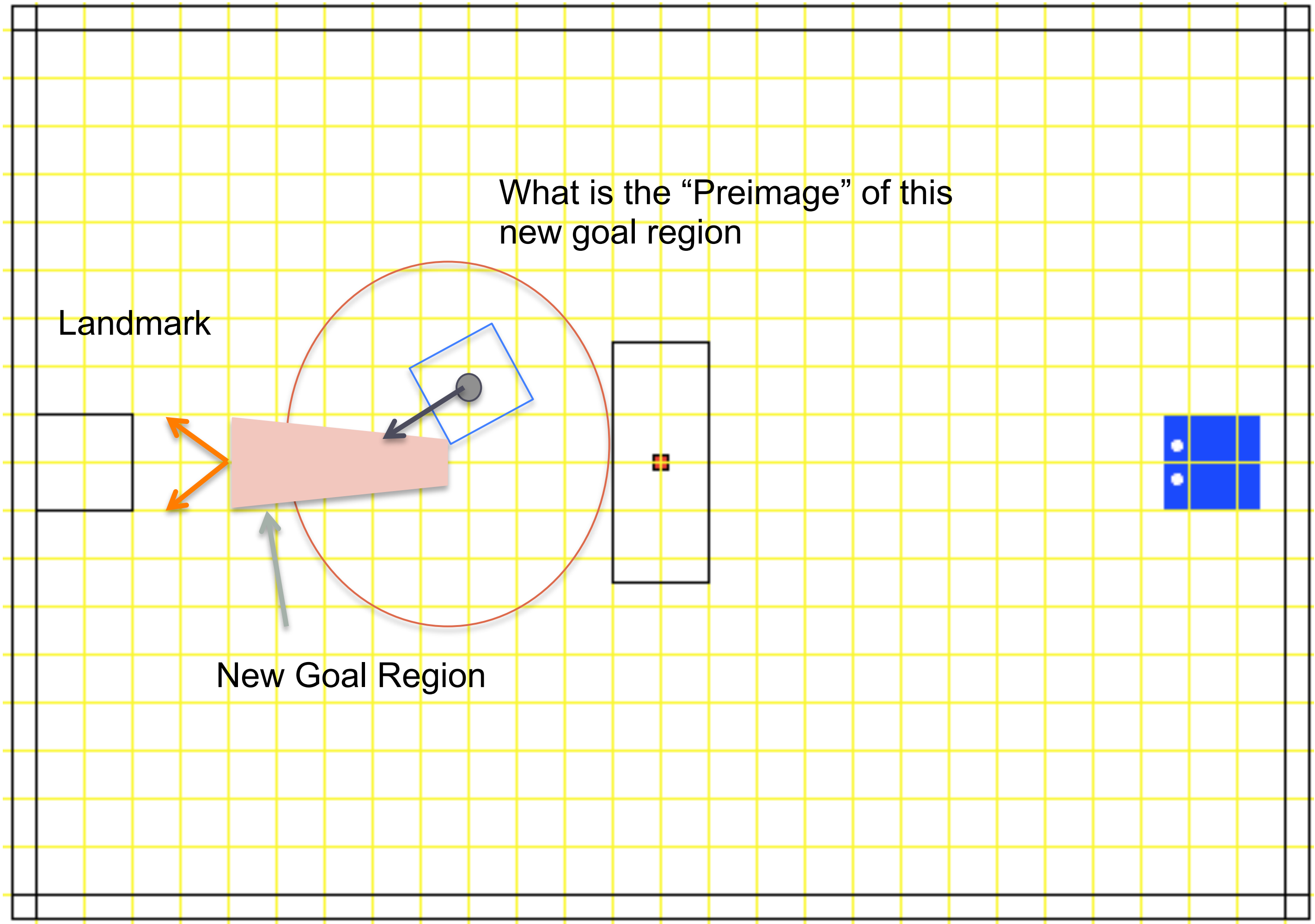
# Use Of Landmark: Region Where Landmark Can be Reached



# Concept of "Preimage"

- The "preimage" of a goal region for a given motion command M = (d, TC) is the set of all points in the robot's configuration space such that if the robot starts executing the command from any one of these points, it is guaranteed to reach the goal and stop in it.
- We are only using the concept in a high level. For rigorous treatment of this topic, please see reading list

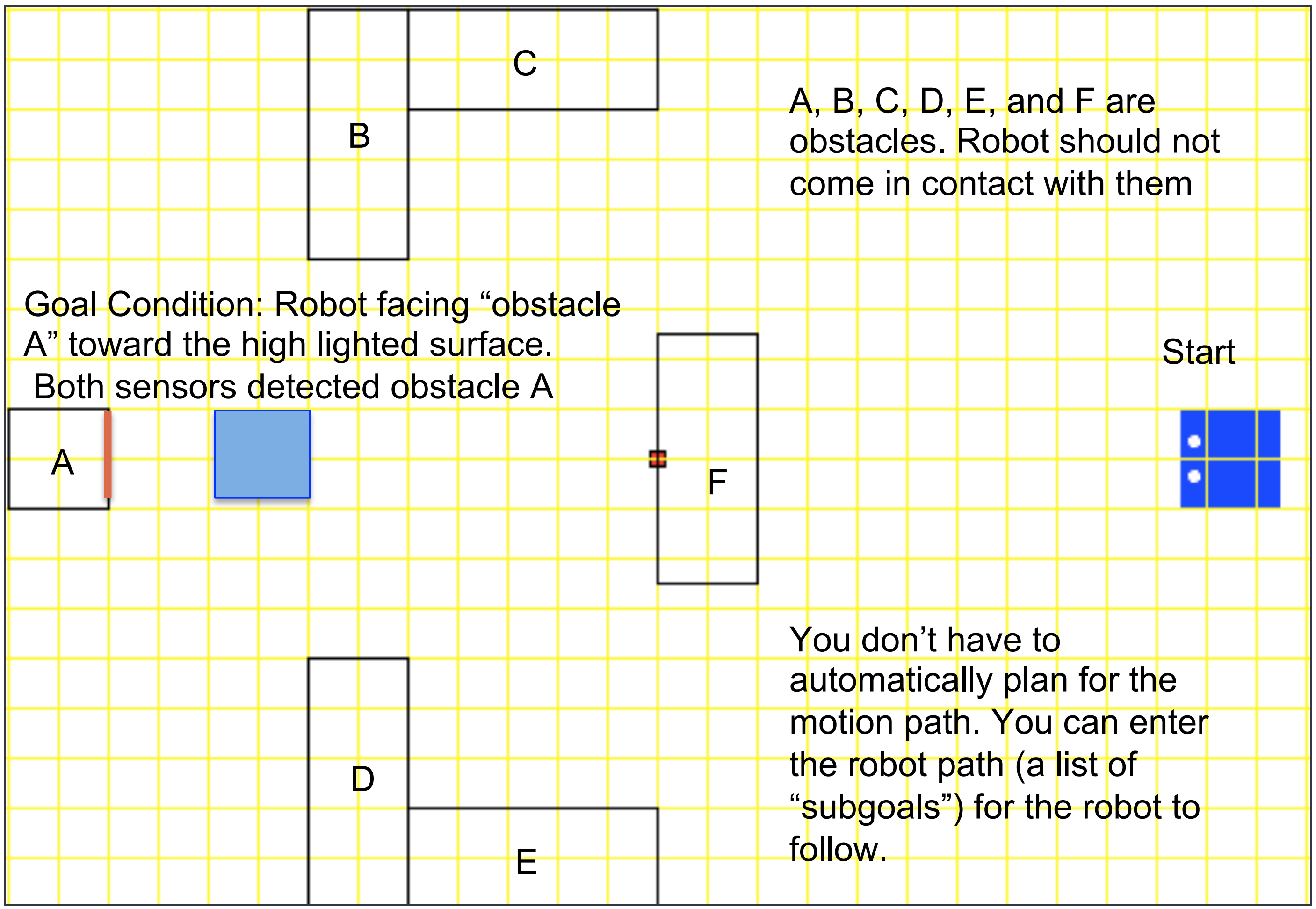
# How to Guarantee Reaching New Goal Region



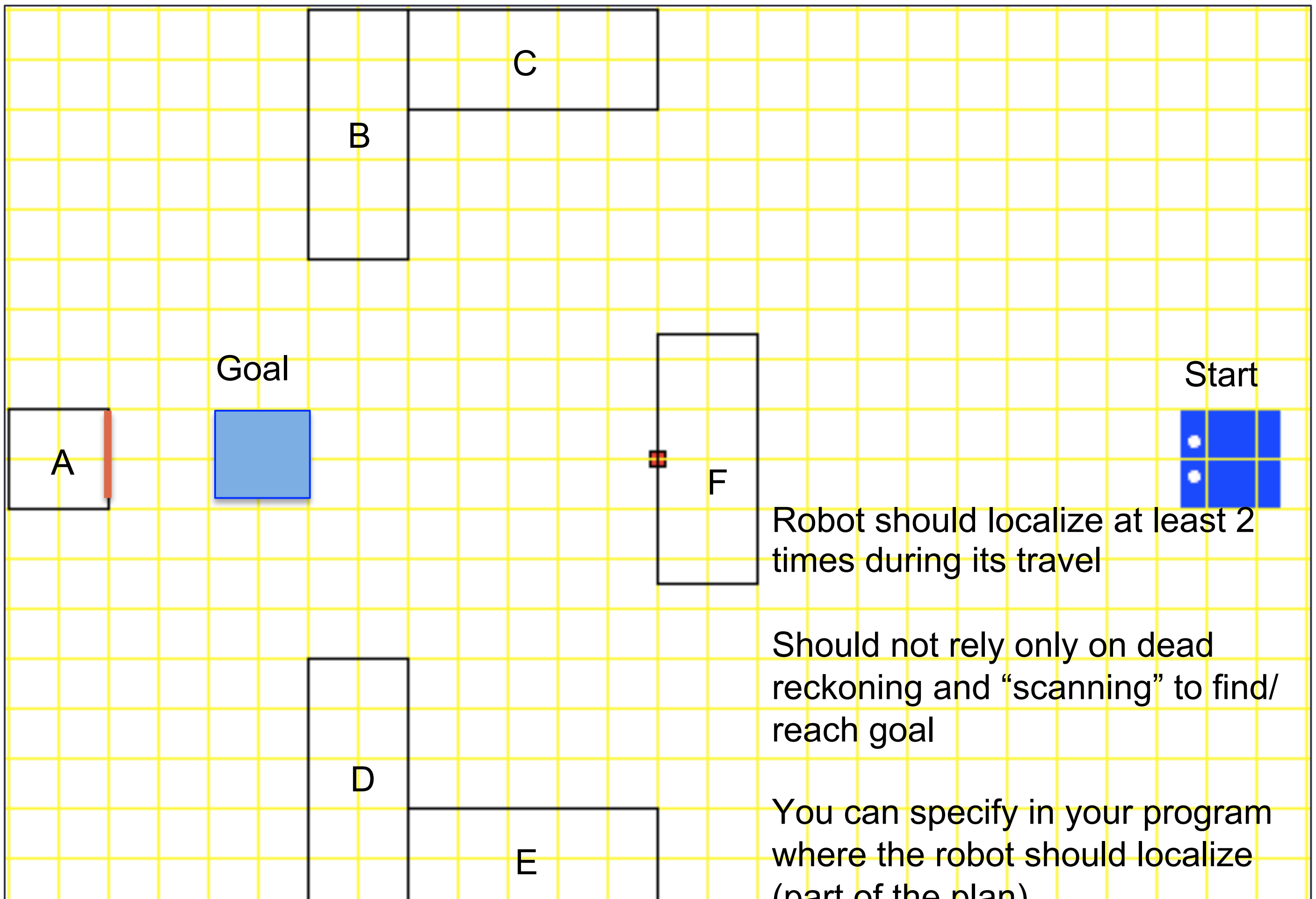
# Pre-image “Backchaining”

- Preimage backchaining consists of constructing a sequence of motion commands Mi, i= l,... ,n, such that, if P,, is the preimage of the goal for M,, P,_t the preimage of P,, for M,,_t, and so on, then Pt contains the initial region.

# Home Work Part #3-2



# Home Work Part #3-2




Robot should localize at least 2 times during its travel

Should not rely only on dead reckoning and "scanning" to find/ reach goal

You can specify in your program where the robot should localize (part of the plan)