Day 23

Representing and Reasoning About Space (Chapter 6)

Introduction

- representing the environment
  - affects how the robot reasons about its environment
- how should space be represented?
  - map representation, how to map, planning on a map
- how to represent the robot?
  - configuration space
- how the robot can reason with respect to its representation of space

Representing Space

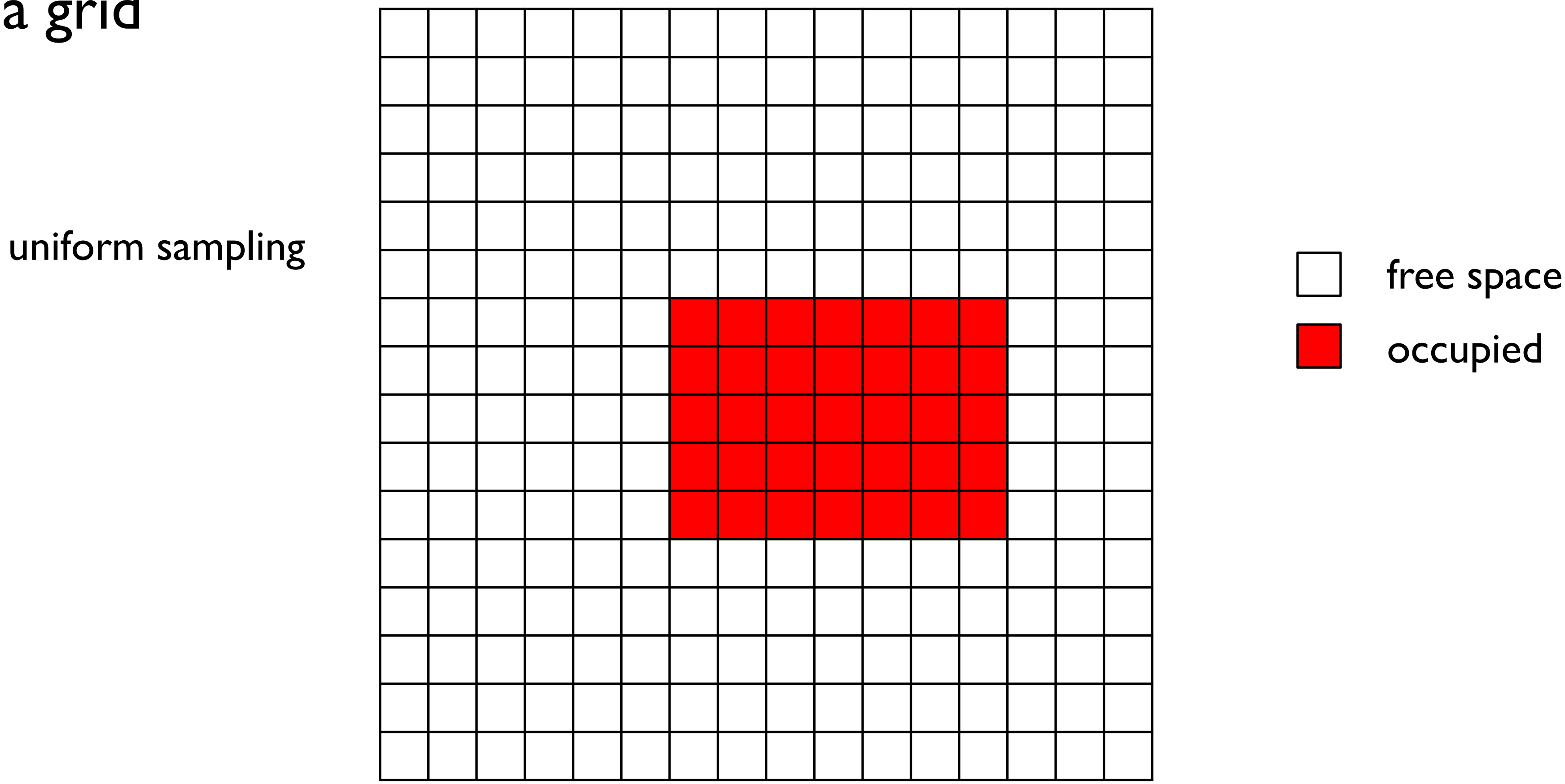
- many tasks require a representation of the robot's environment (a map)
  - but many complex tasks can be accomplished without an explicit map (e.g., Roomba)
- in addition to representing places in the environment, the map can include other information
  - "Here there be dragons!"

Representing Space

- at least 3 different classes of task typically require a map
  1. to establish what parts of the environment are free for navigation
     - called the free space
     - path planning
  2. to recognize regions or locations
  3. to recognize specific objects

Spatial Decomposition

- represent space itself, rather than the objects in it, using discrete samples
- many ways to perform the sampling, but the simplest is to use a grid



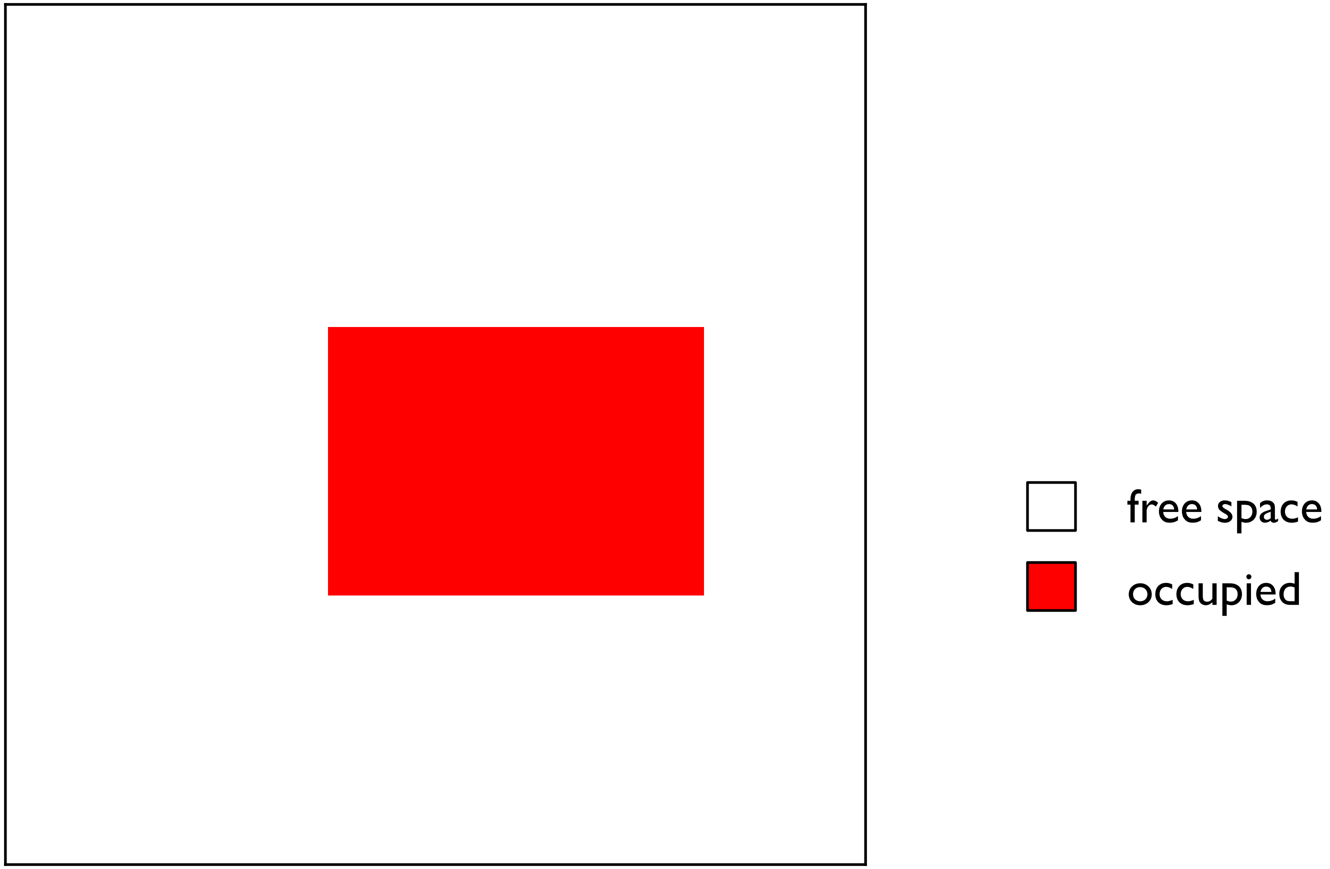
Uniform Sampling

- very general representation
  - grid locations can represent anything
- if something moves then the representation does not change dramatically
- limited by grid resolution
  - large cell sizes give a coarse representation
  - small cell sizes are storage intensive
    - football pitch (soccer field) at $1cm^2$ resolution
      - 105m x 68m x 100 x 100 = 71,400,000 cells
    - 3D is much worse

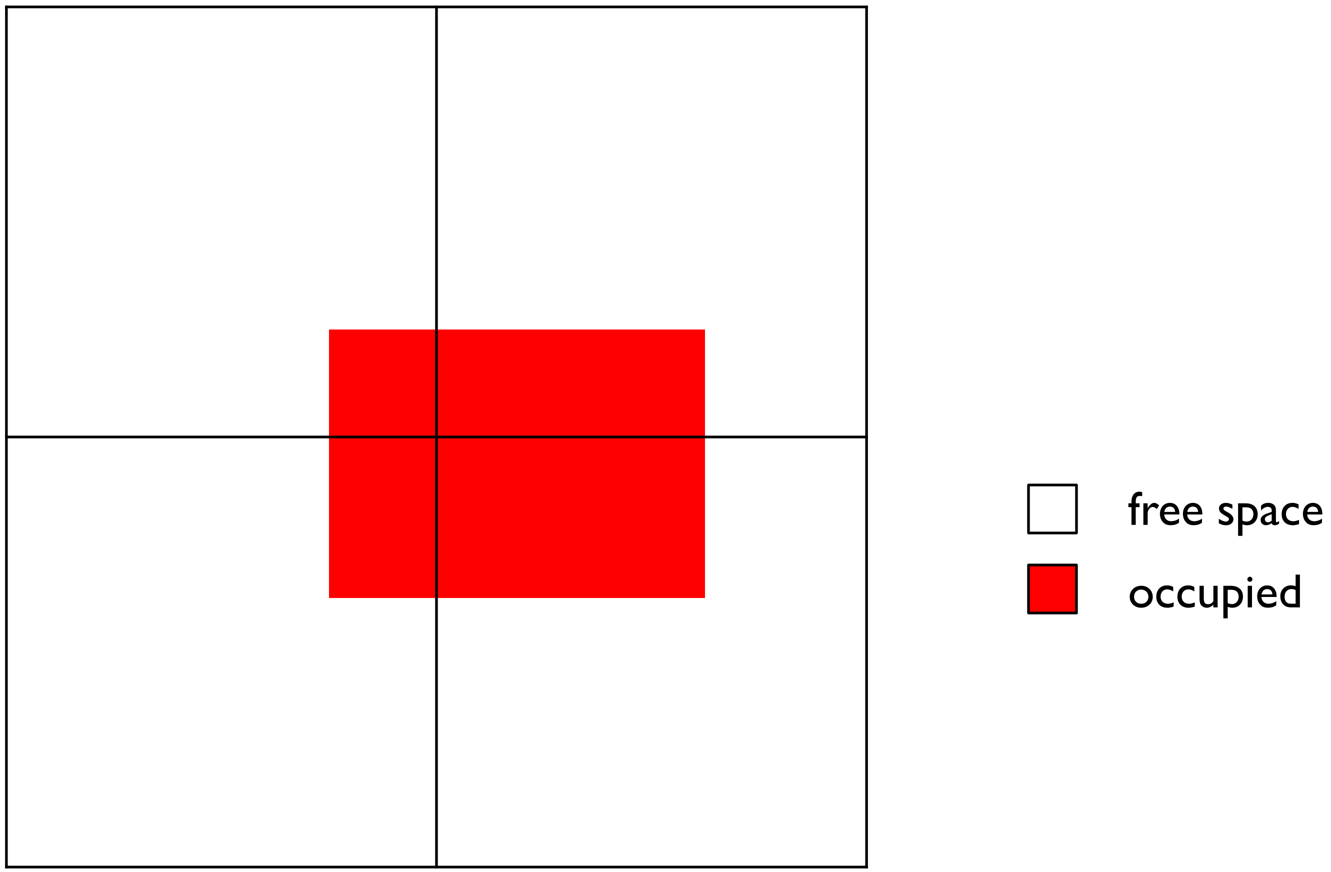
Recursive Hierarchical Representations

- storage space can be conserved by observing that free space cells and occupied cells tend to cluster
  - group the clusters into larger cells
- quadtree
  - recursively subdivided space into 4 equal-sized cells until every cell is either uniformly free or uniformly occupied
    - or some threshold resolution is achieved

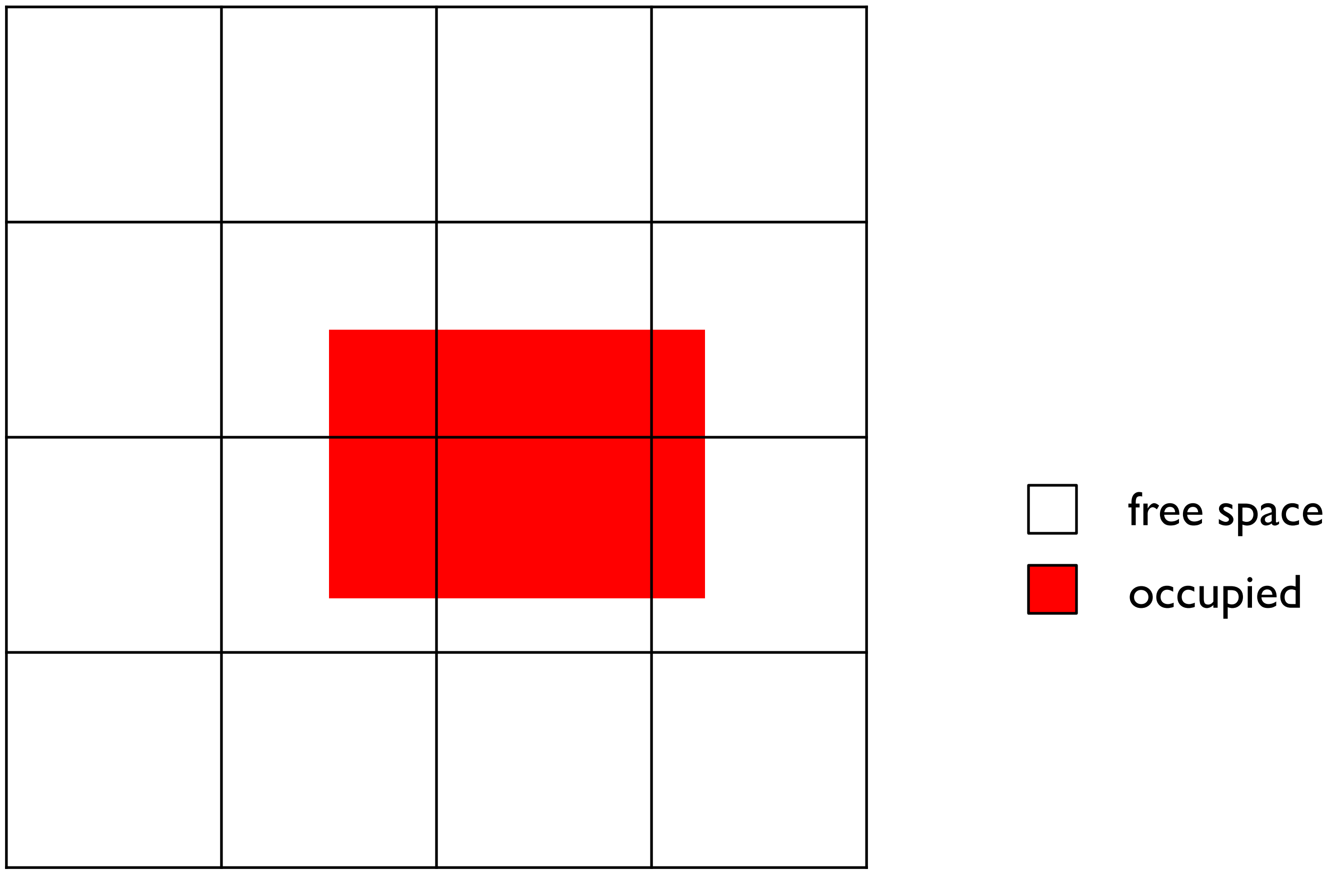
Quadtree Decomposition



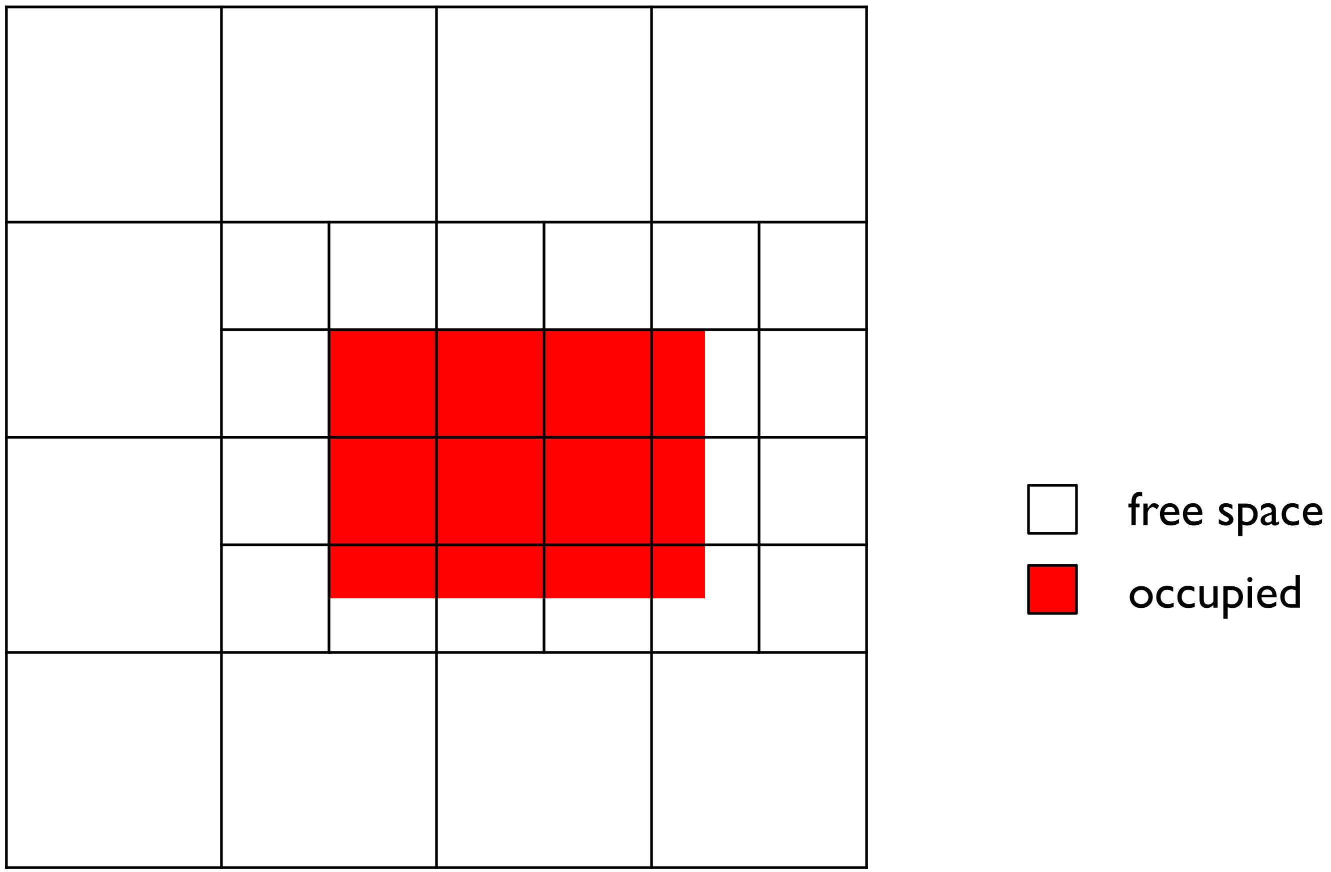
Quadtree Decomposition



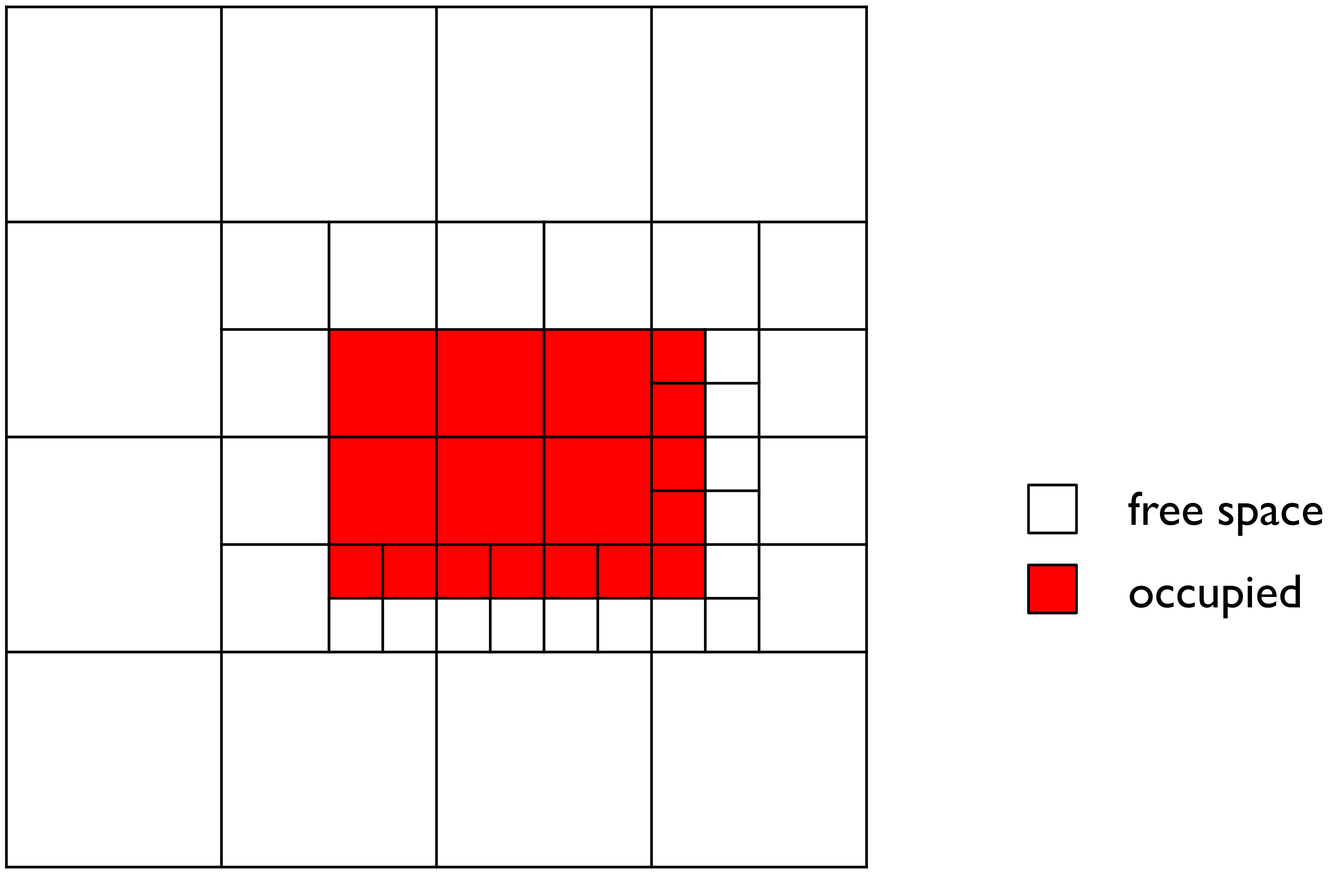
Quadtree Decomposition



Quadtree Decomposition



Quadtree Decomposition



Quadtree Decomposition

- worst case performance
  - same as uniform subdivision
- if most of the space is occupied or freespace then the representation is compact
- generalizes to N dimensions
- representation can change dramatically if objects move even a small amount

Geometric Representations

- discrete geometric primitives
  - points
  - lines, line segments, polylines
  - circles, ellipses
  - polyhedra
  - splines

Representing the Robot

- to create motion plans for the robot, we must account for the position and size of the robot
  - we must be able to specify the location of every point on the robot
- in robotics, the configuration space is a fundamental concept in motion planning

Configuration Space

- configuration
  - a complete specification of the position of every point of the robot
- configuration space (C-space)
  - the space of all possible configurations of the robot

Example: Point Robot

- consider a point robot that can translate (but not rotate) in the infinite plane
- configuration
  - just the location

$$q = (x, y)$$

- configuration space
  - R^2 (the Cartesian plane)

Example: Circular Robot

- consider a circular robot of radius R that can translate (but not rotate) in the infinite plane
- configuration?
  - suppose the center of the robot has position (x, y)
  - then the points on the robot are given by

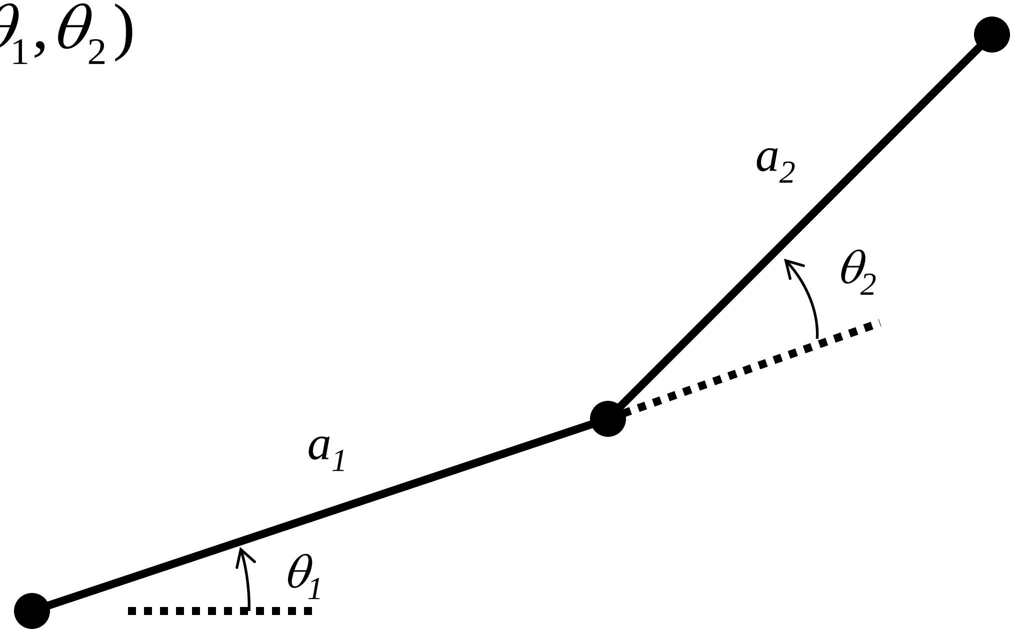
$$R(x,y) = \{(x',y') \mid (x-x')^2 + (y-y')^2 \le r^2\}$$

  - so (x, y) is sufficient to describe the configuration of the robot
- configuration space
  - R^2 (the Cartesian plane)

Example: 2-Link Planar Arm

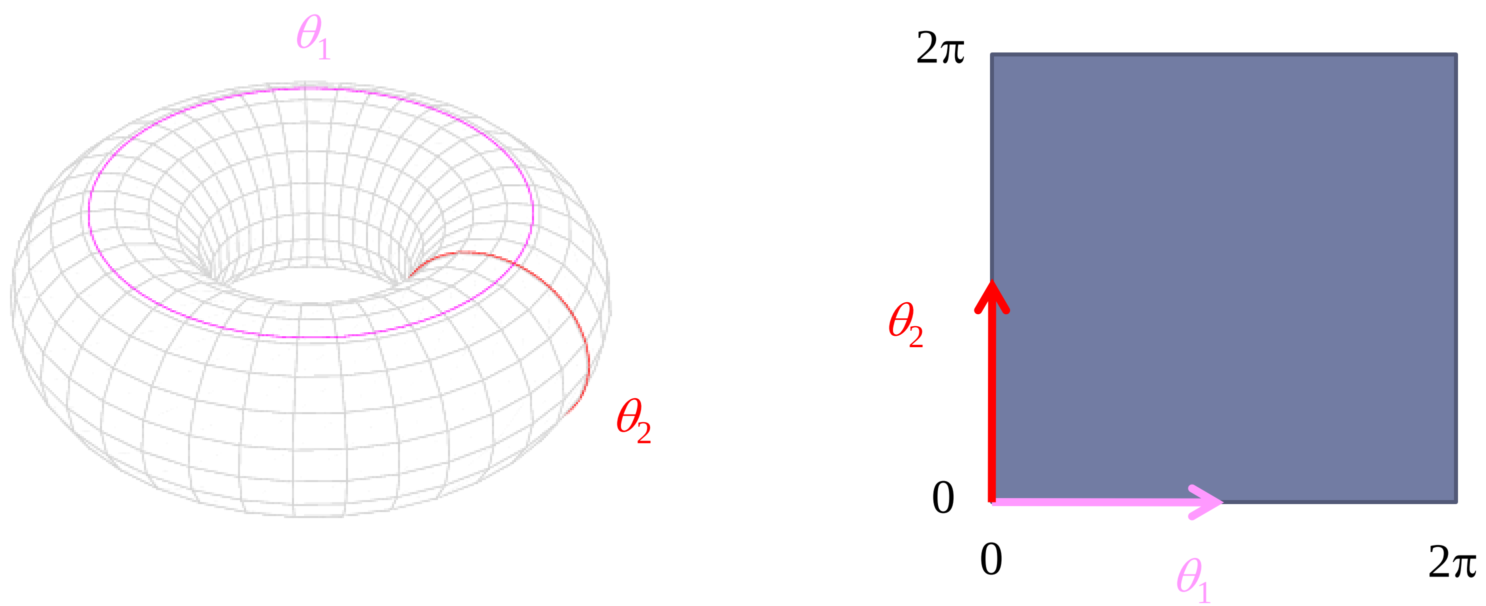
- consider a 2-link planar arm with no joint angle limits
- configuration
  - joint angles

$$q = (\theta_1, \theta_2)$$



Example: 2-Link Planar Arm

- configuration space
  - each angle corresponds to a point on a unit circle
    - configuration space is a torus, which can be cut and flattened onto the plane



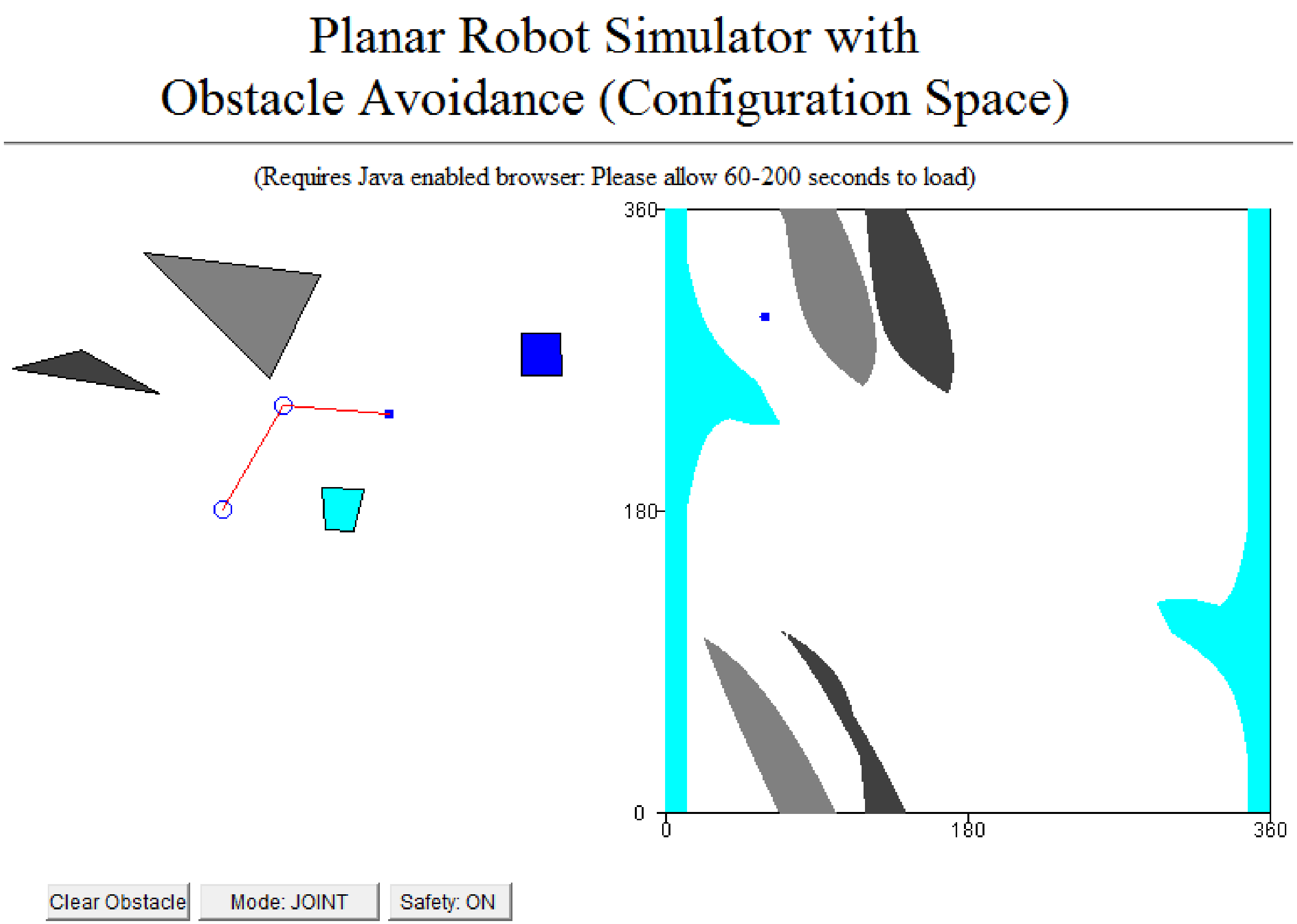
Obstacles in C-Space

- obstacles in the environment may limit the set of possible configurations of the robot
  - let B_i be an obstacle in the environment
  - let C be the configuration space of the robot
  - let $A(q)$ be the portion of space occupied by the robot when it is in configuration q
  - then CB_i is the configuration space representation of the obstacle (C-obstacle) defined as

$$CB_i = \{q \in C \mid A(q) \cap B_i \neq 0\}$$

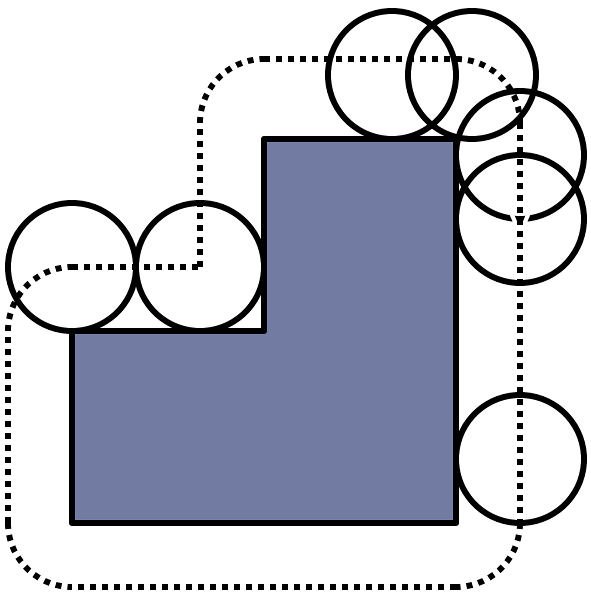
Example: 2-Link Planar Robot

- http://ford.ieor.berkeley.edu/cspace

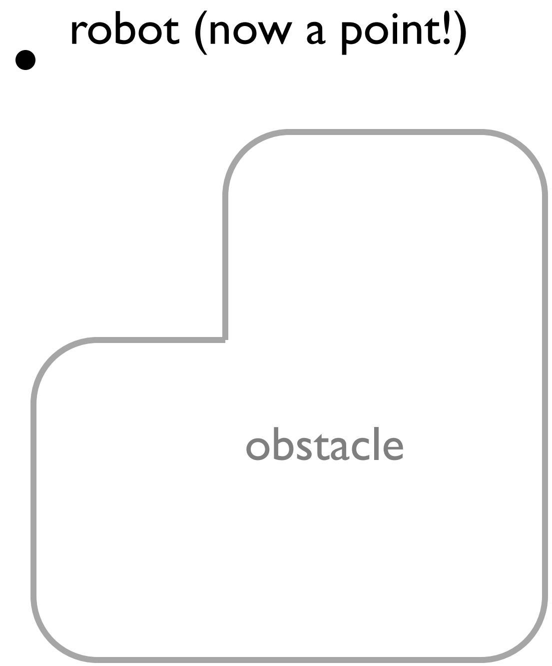


Example: Circular Robot

- the free configuration space for a circular robot can be found by tracing the obstacles in the workspace with the robot



workspace



C-space

Example: Circular Robot