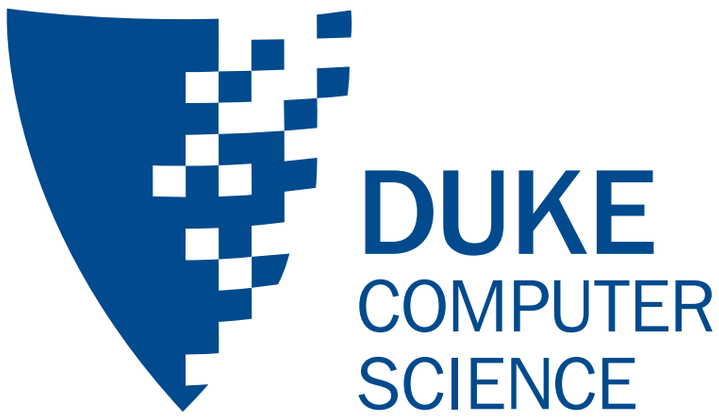
Robot Motion Planning

George Konidaris
gdk@cs.duke.edu



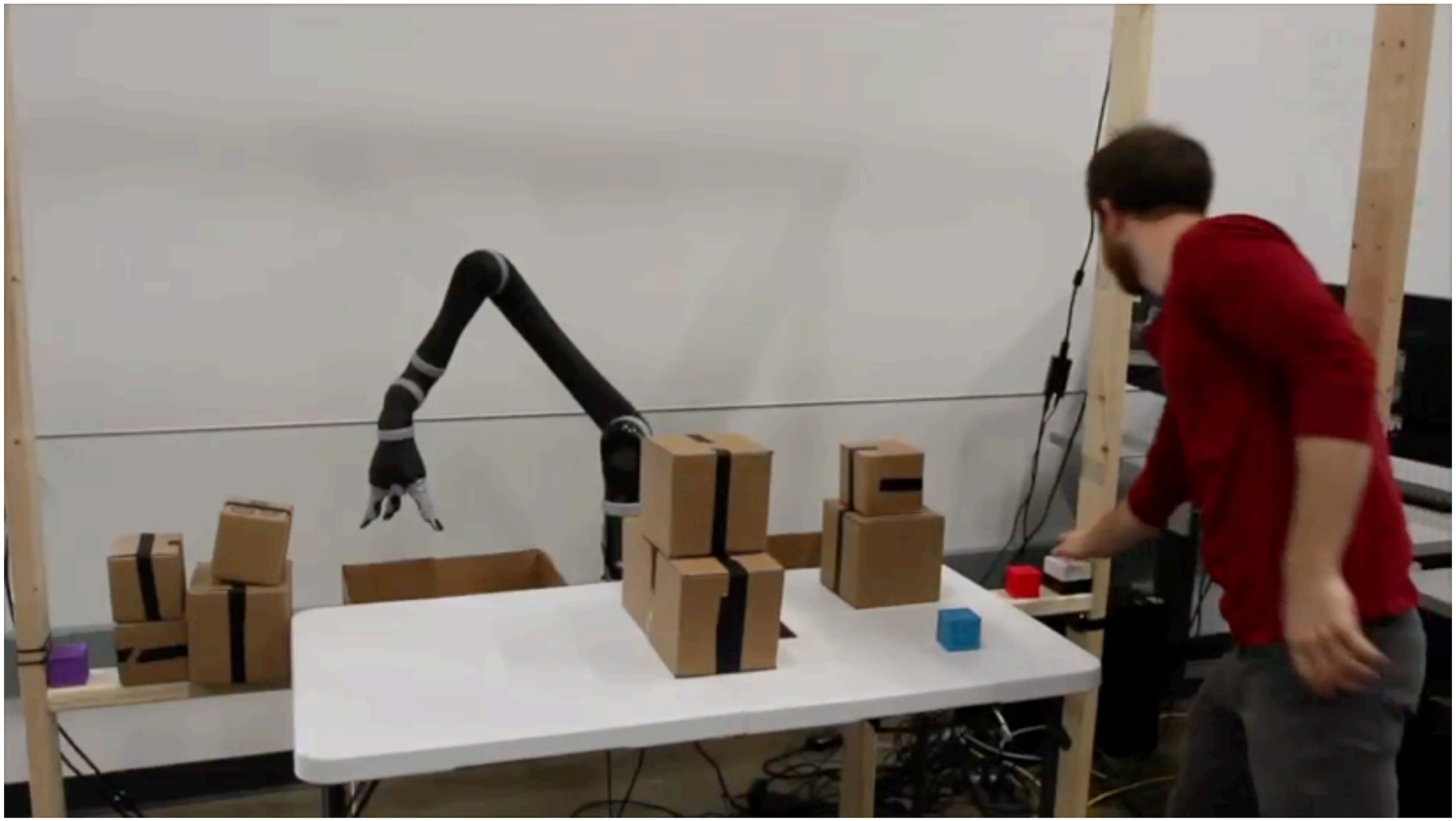
Spring 2016

Robot Motion Planning

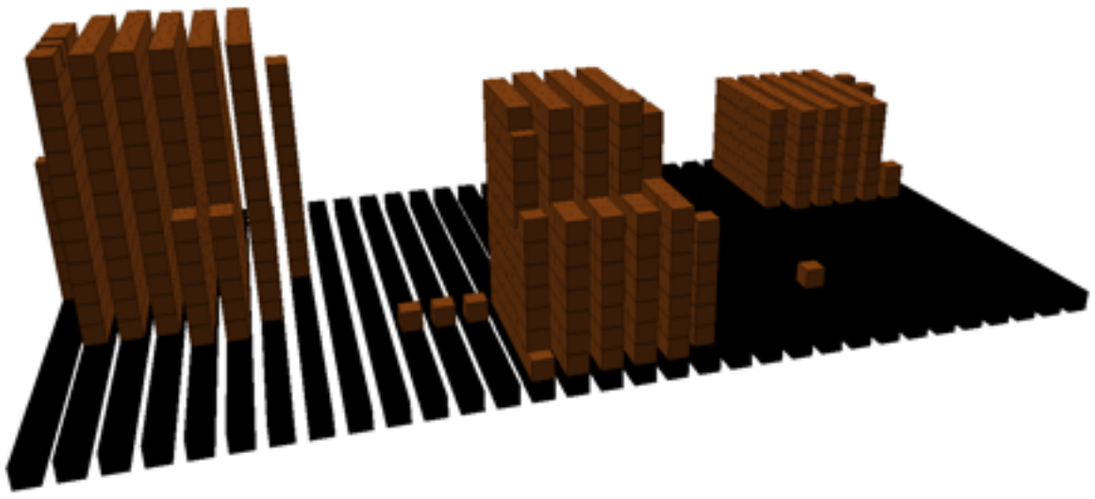
DUKE
COMPUTER
SCIENCE

Motion Planning

DUKE
COMPUTER
SCIENCE



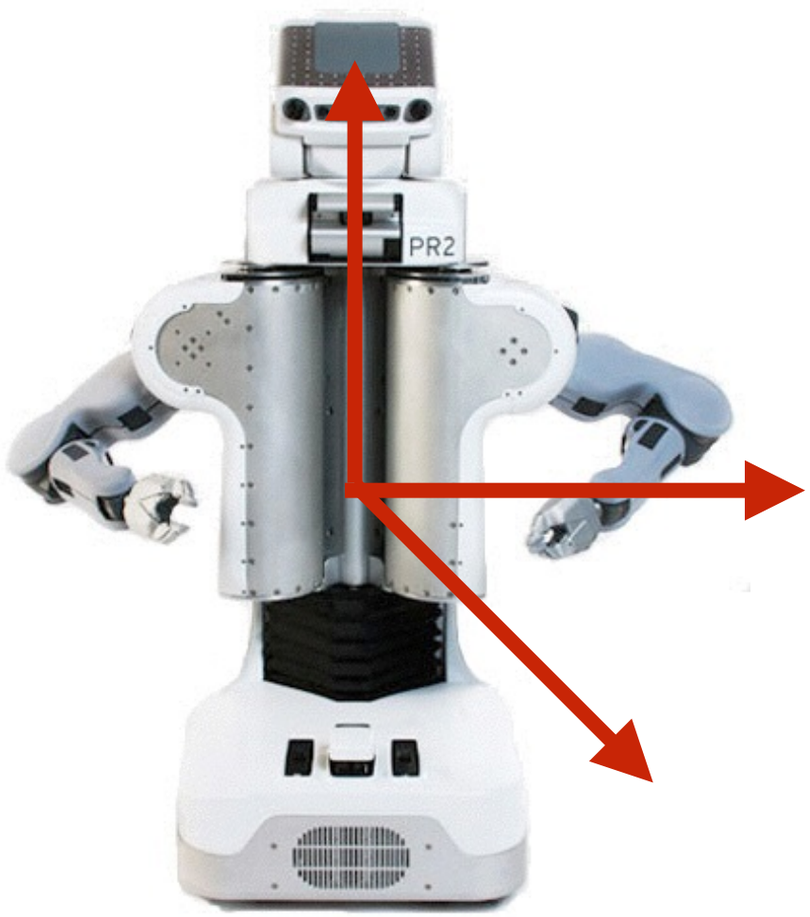
Motion Planning



Problem Definition

Robot has a **configuration space (C-space**

- Values for each joint
- Overall pose of reference frame



Start and end points

- Start point
- Goal region

Set of obstacles

- Dense regions of 3D-space (*Also* regions of C-space)

Objective: obstacle-free path through configuration space from start to goal.

Configuration Spaces

Each joint is a ***dimension*** of the configuration space.

Let's say we have a robot with an arm with two revolute joints.

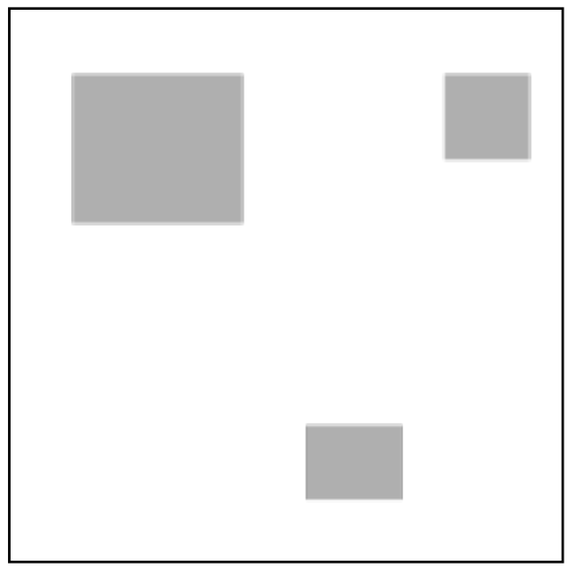
Configuration space:

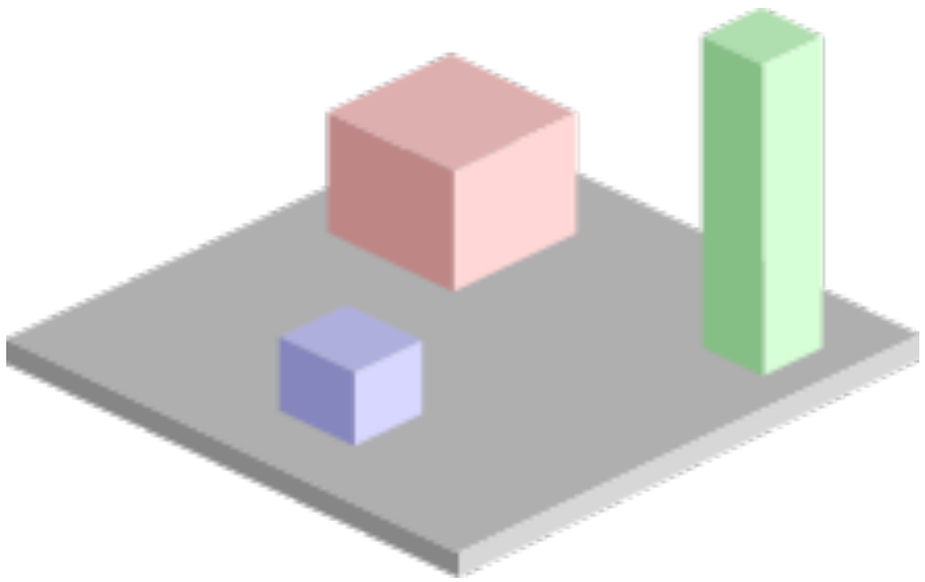
- *x, y, z* of *base frame*
- pitch, roll, yaw, of *base frame*
- angle of first joint
- angle of second joint

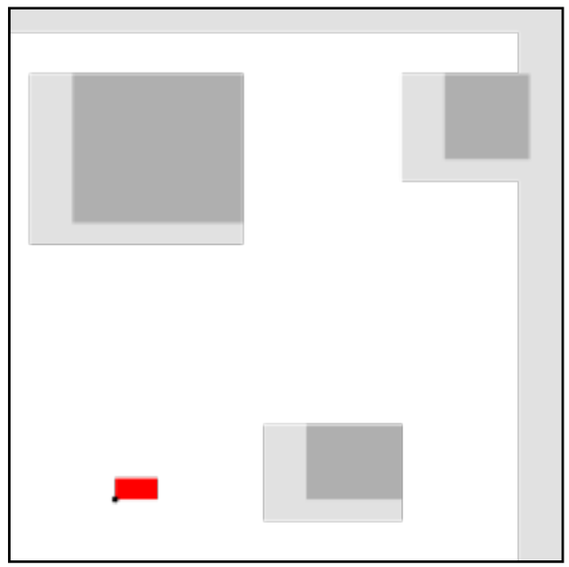
A configuration is a *setting of values* to these 8 variables.
Configuration space is the *space of all such settings*.

Configuration Space

Obstacles are no-go regions of configuration space.





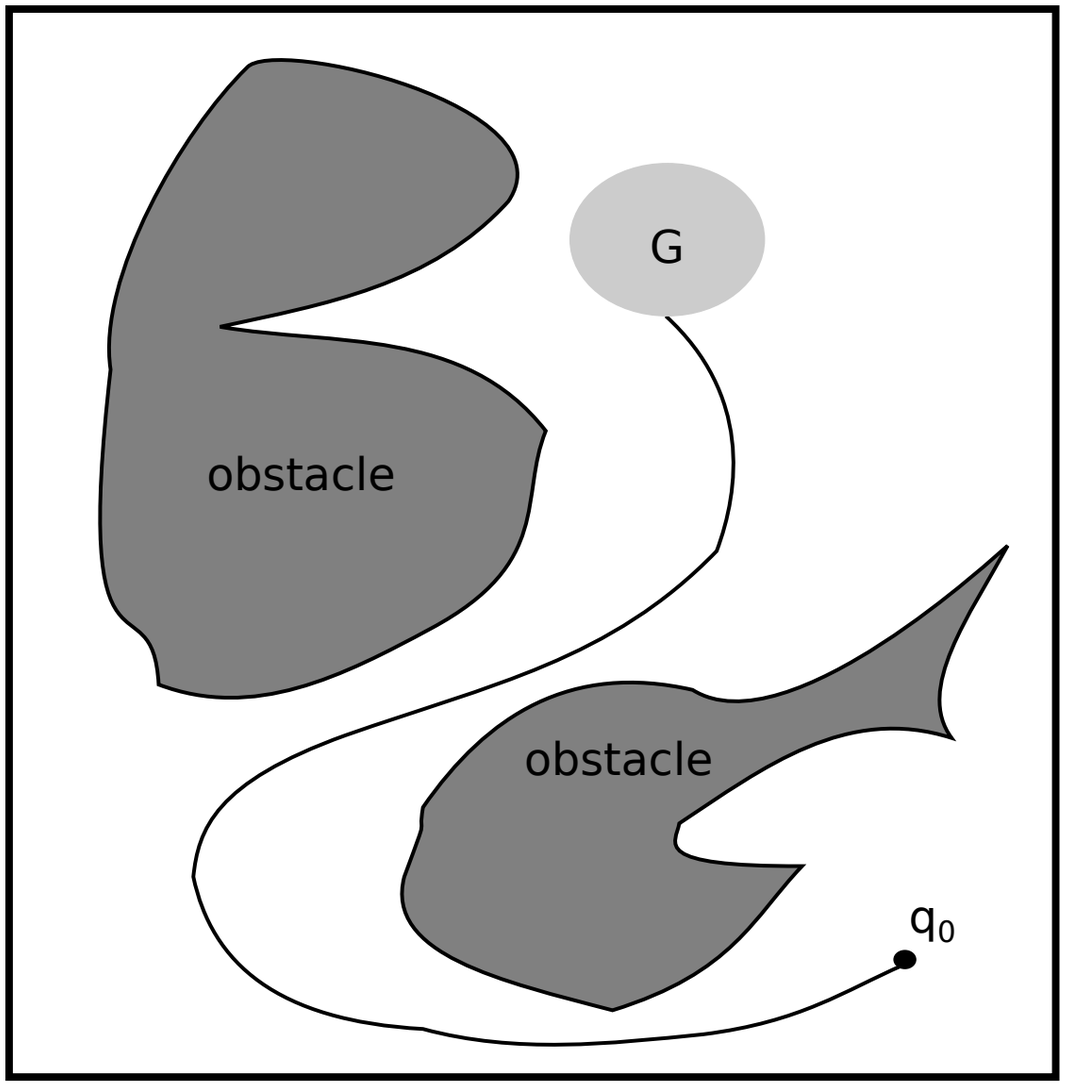


(images from Wikipedia)

Planning

We wish to find a path through configuration space such that:

- Path feasible
- No collisions

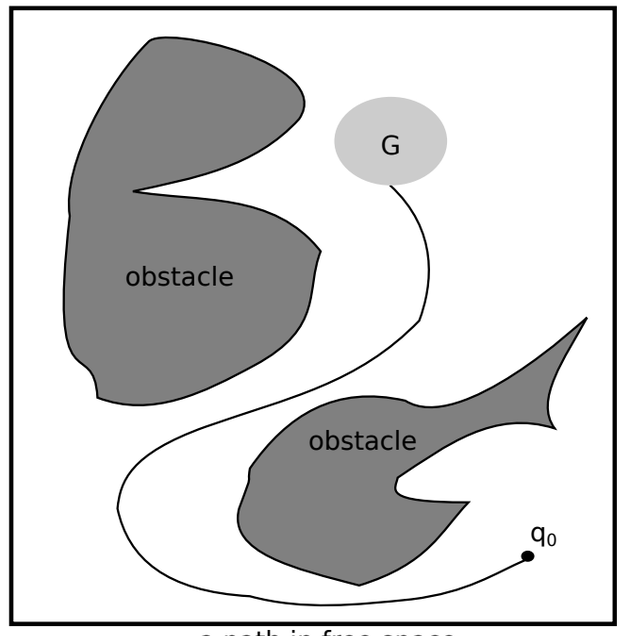


a path in free space

Paths

Simple definition of a path:

- Sequence of points $p = \{p_1, \ldots, p_n\}$
- "Easy" to go between p_i and p_{i+1}.



a path in free space

Solution - path such that:

- p_1 = start
- p_n inside goal
- No collision between any p_i and p_{i+1}.

Broad Approach: Visibility Graphs

DUKE COMPUTER SCIENCE

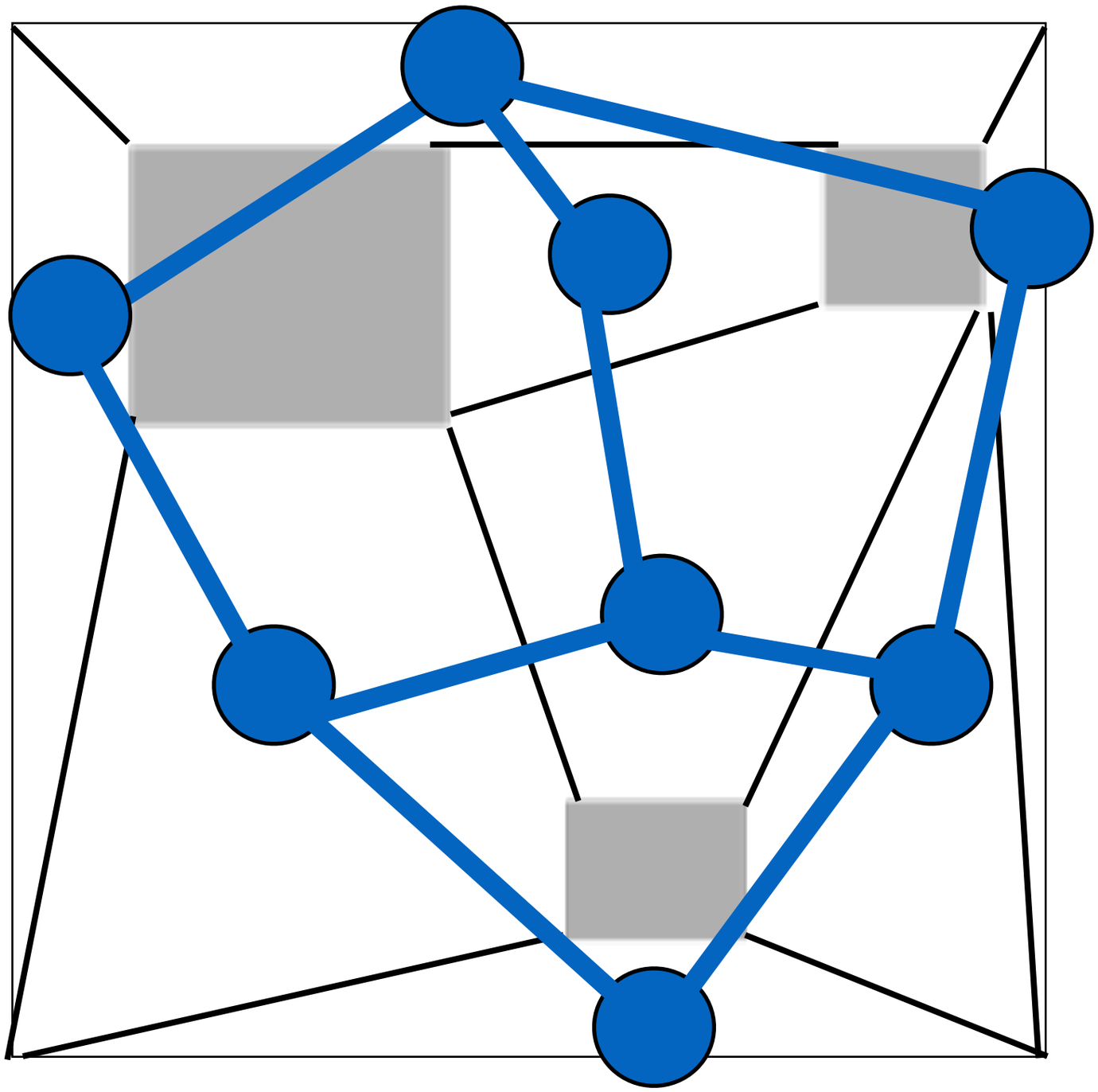
1. Break *C-space* up into convex regions.

2. We know we can get from anywhere in a convex region to anywhere else with a straight line.

3. Build a graph: each node convex region, edge when they share a face.

4. Do search on the graph.

Visibility Graphs



Issues

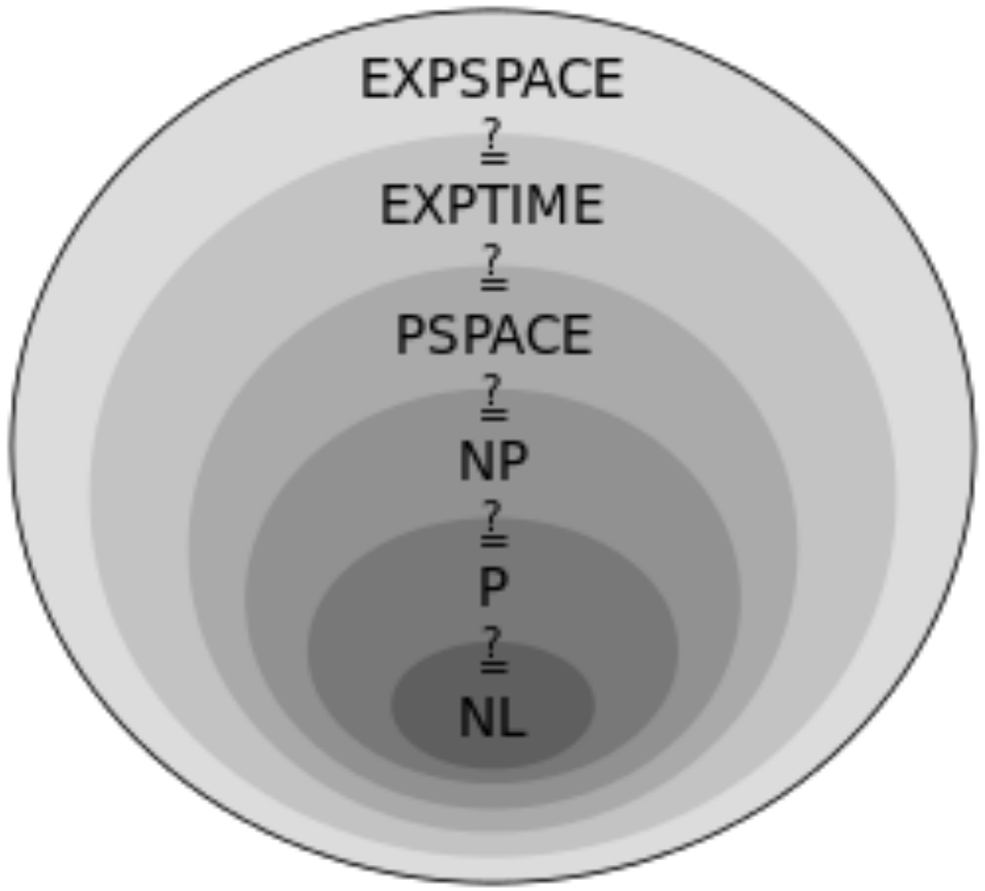
These are hard to use:

- Convex region numbers grow exponentially with dimension.
- Need analytical model of each obstacle *in C-space*.
- Need analytical model of C-space!

Consequently, these methods only used for very low-*d* problems.

Complexity

Issue: motion planning is P-SPACE complete (Reif, 1979).



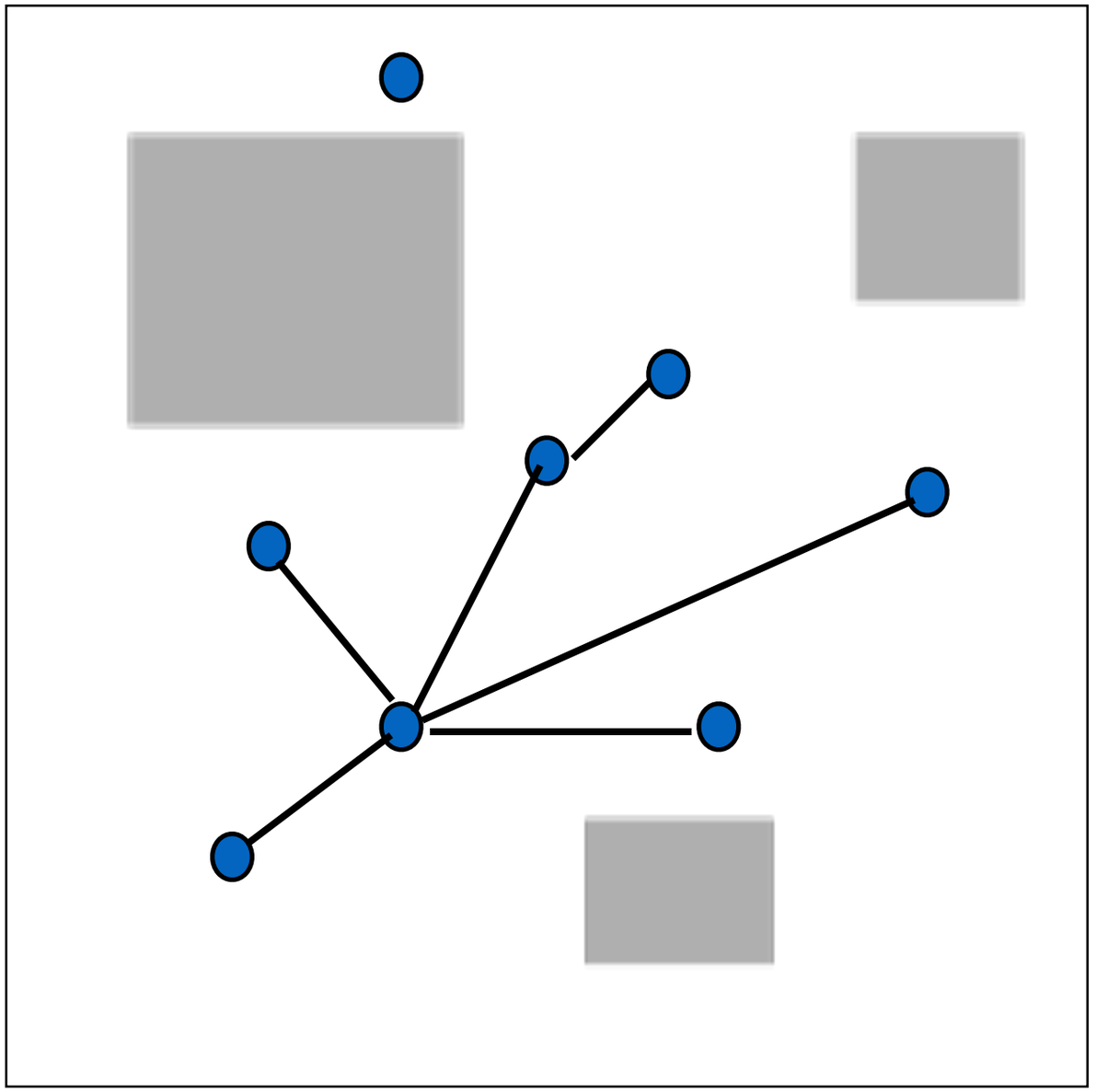
Solution Family: RRTs

DUKE COMPUTER SCIENCE

Alternative solution:

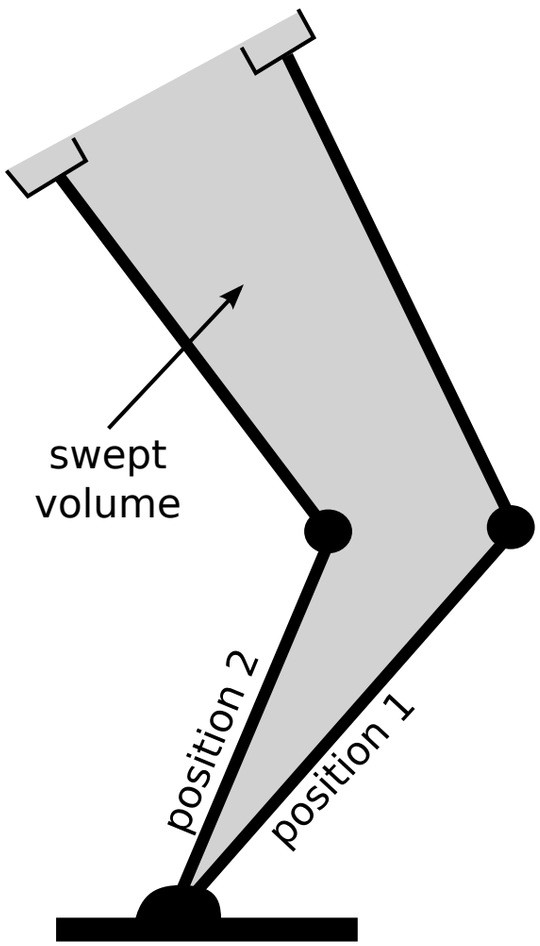
- Rely on randomized algorithms.
- Build *randomized trees* starting from the start state.
- High processing costs but probabilistic guarantees.

RRTs



RRTs

What does an edge between two nodes mean?

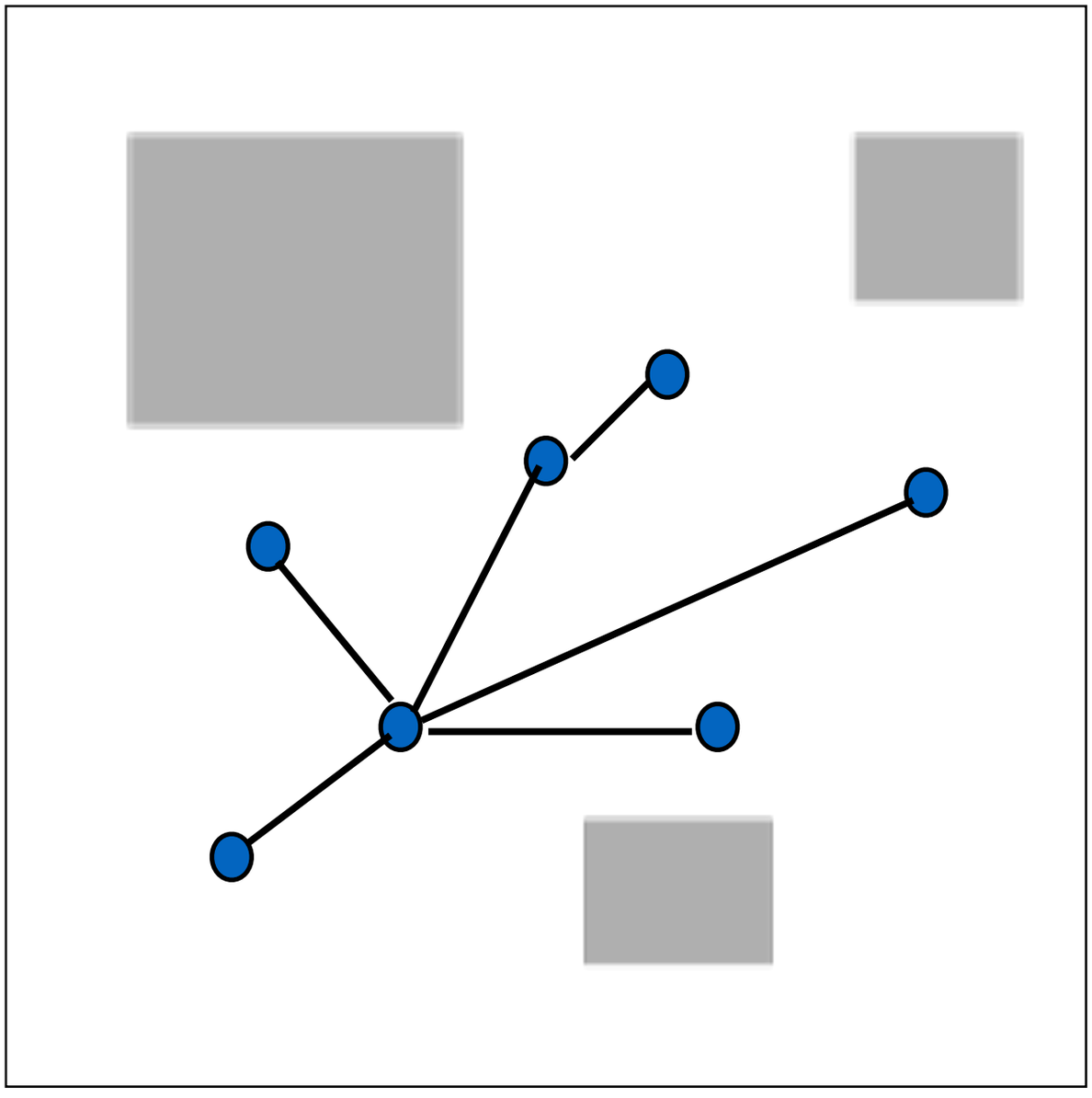


Must test: collision between *obstacle* and *swept volume*.
This can be done in 3-space.

RRT

Property: the tree *rapidly expands* to fill free space.

Why?



Robot Motion Planning

DUKE COMPUTER SCIENCE

Absolutely critical for using robots in **unstructured** environments.

But:

- Fundamentally hard.
- Very well studied
- No real-time solutions.

(Watch this space.)