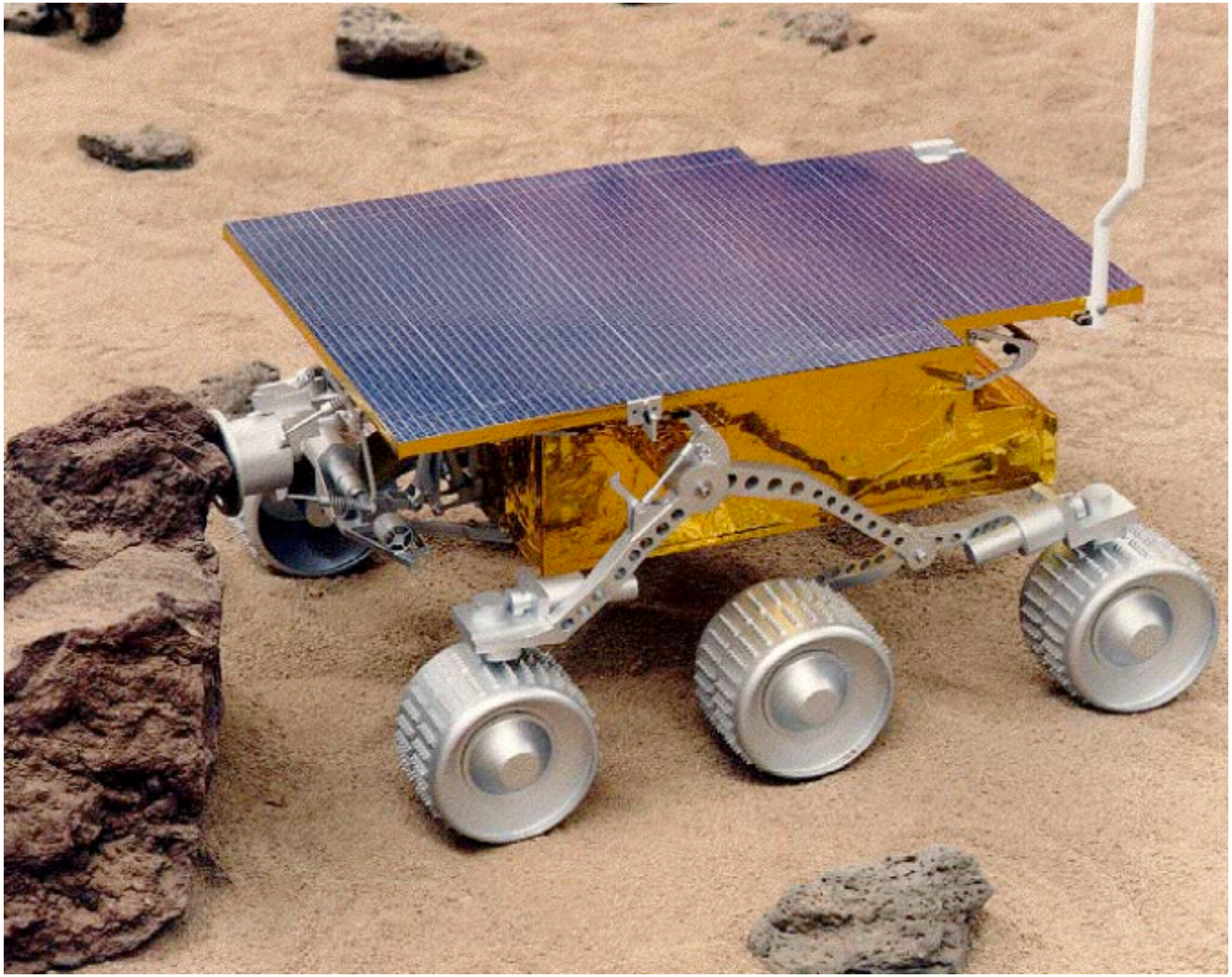
An Introduction to Motion Planning

Alan Kuntz

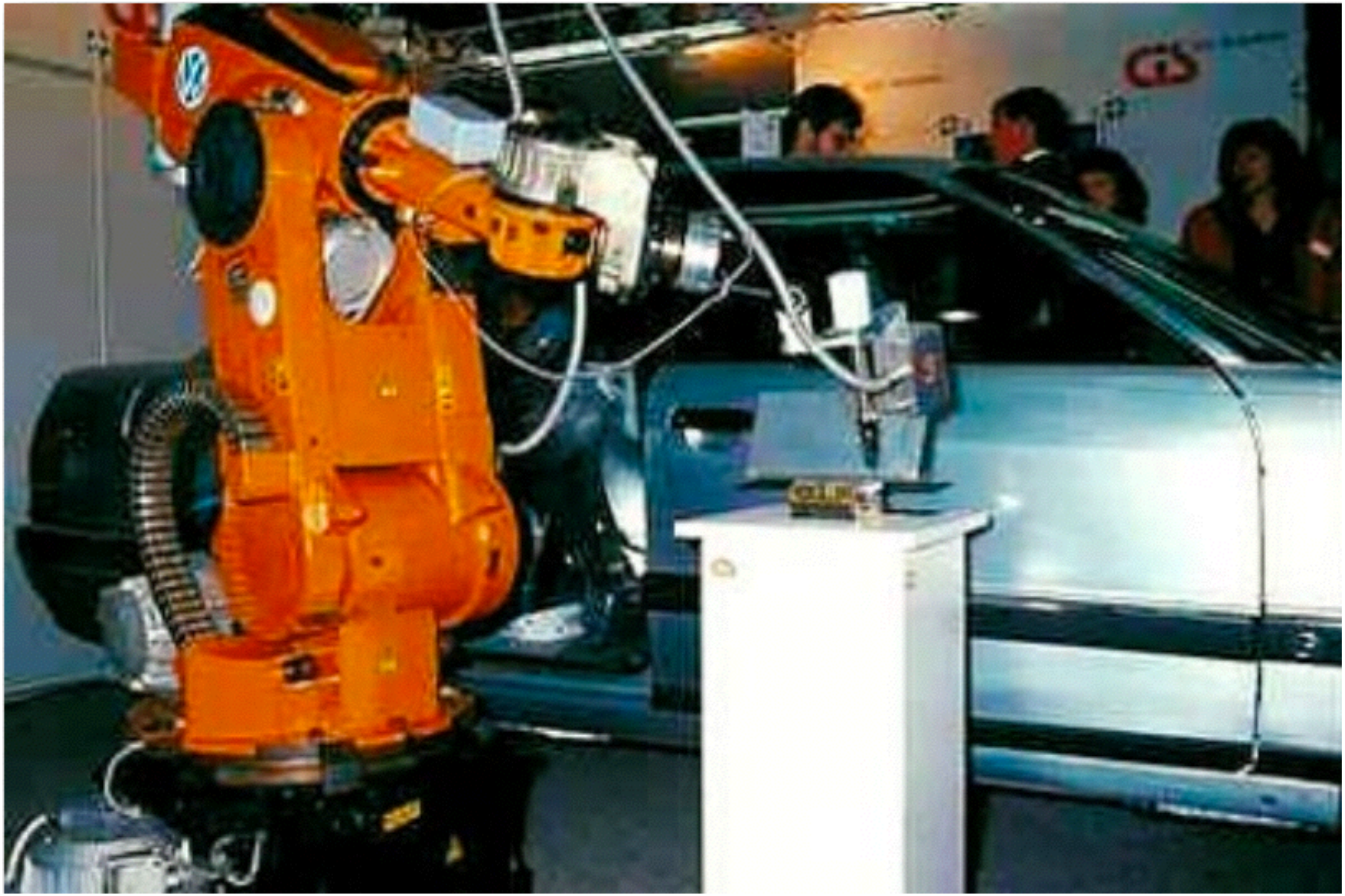
2/4/15

http://cs.unc.edu/~adkuntz/MotionPlanning.pdf

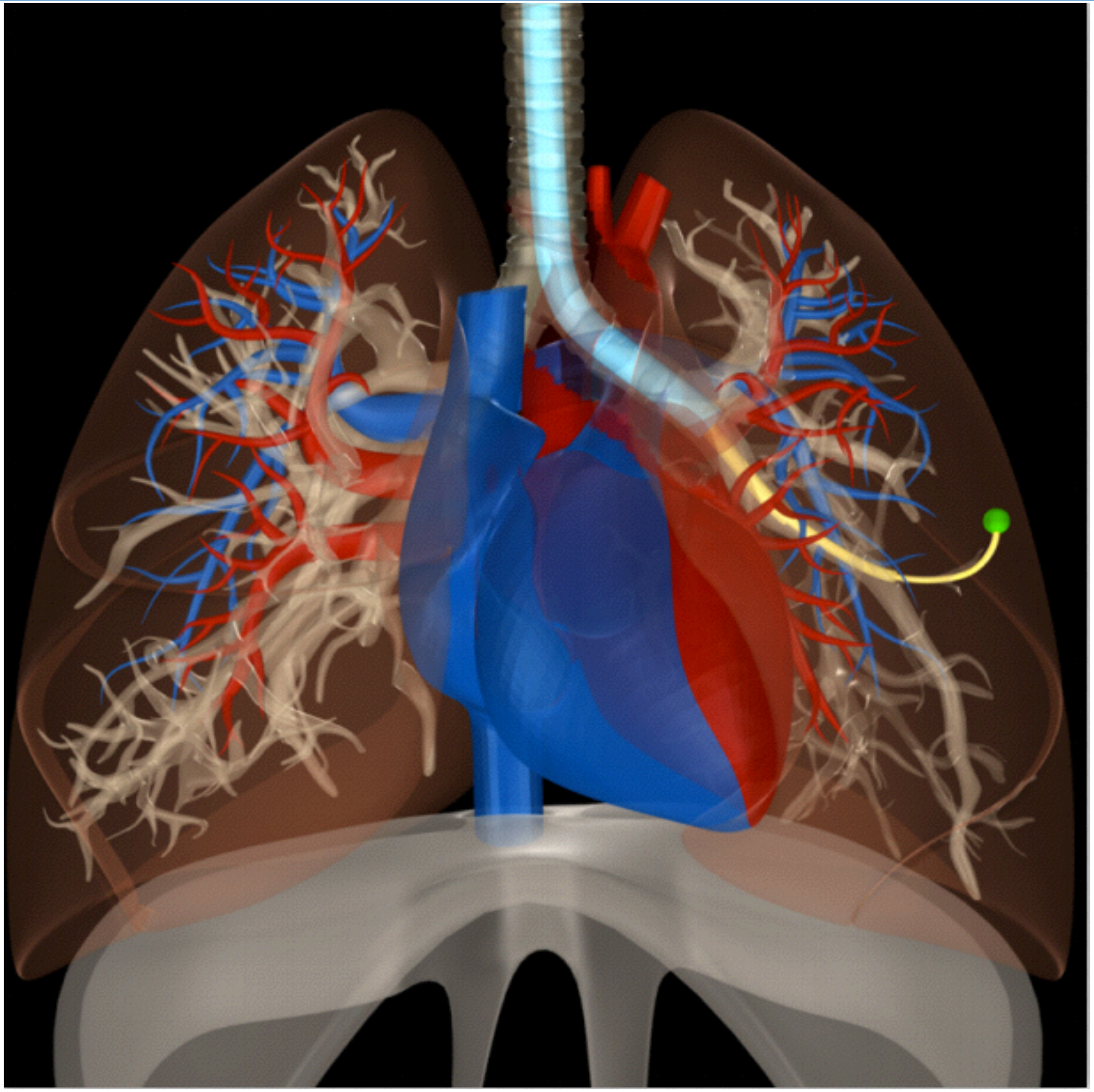
Applications



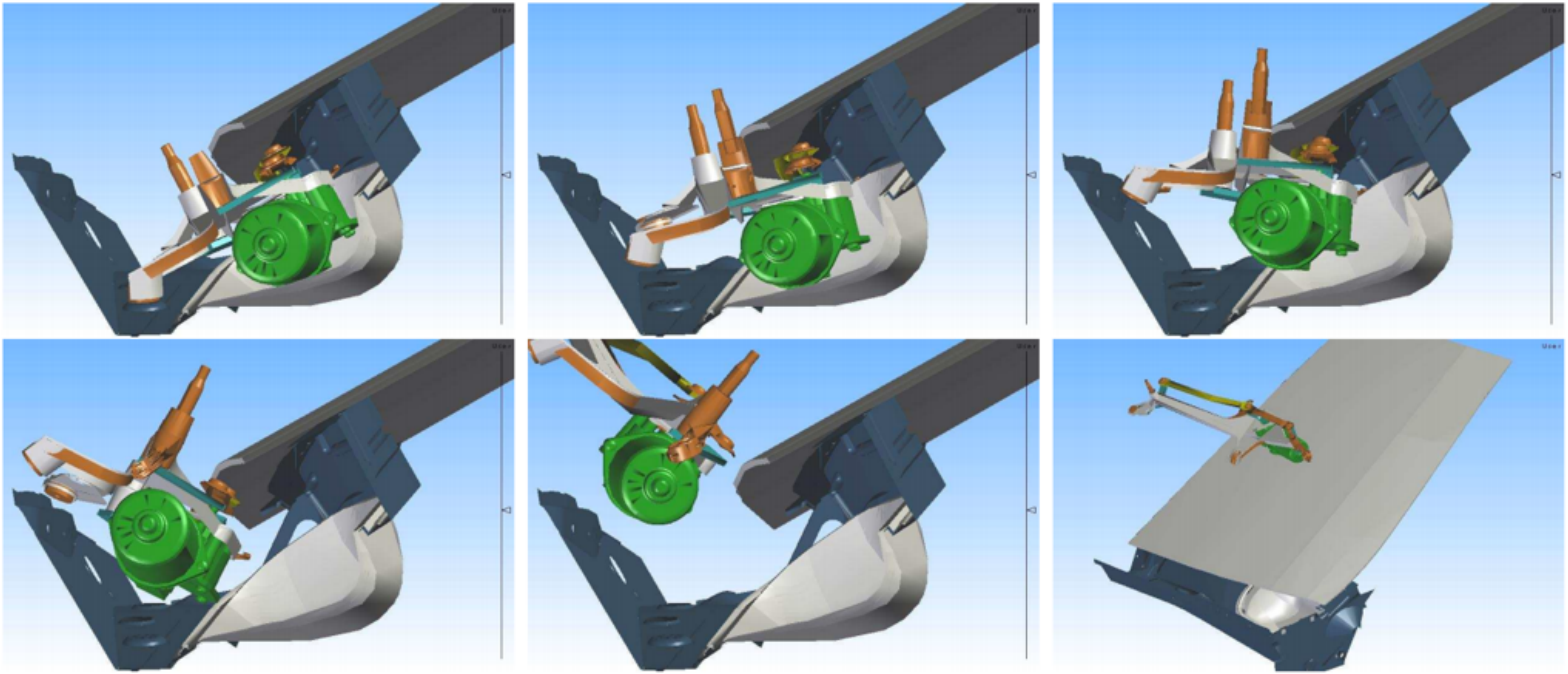
Applications



Applications

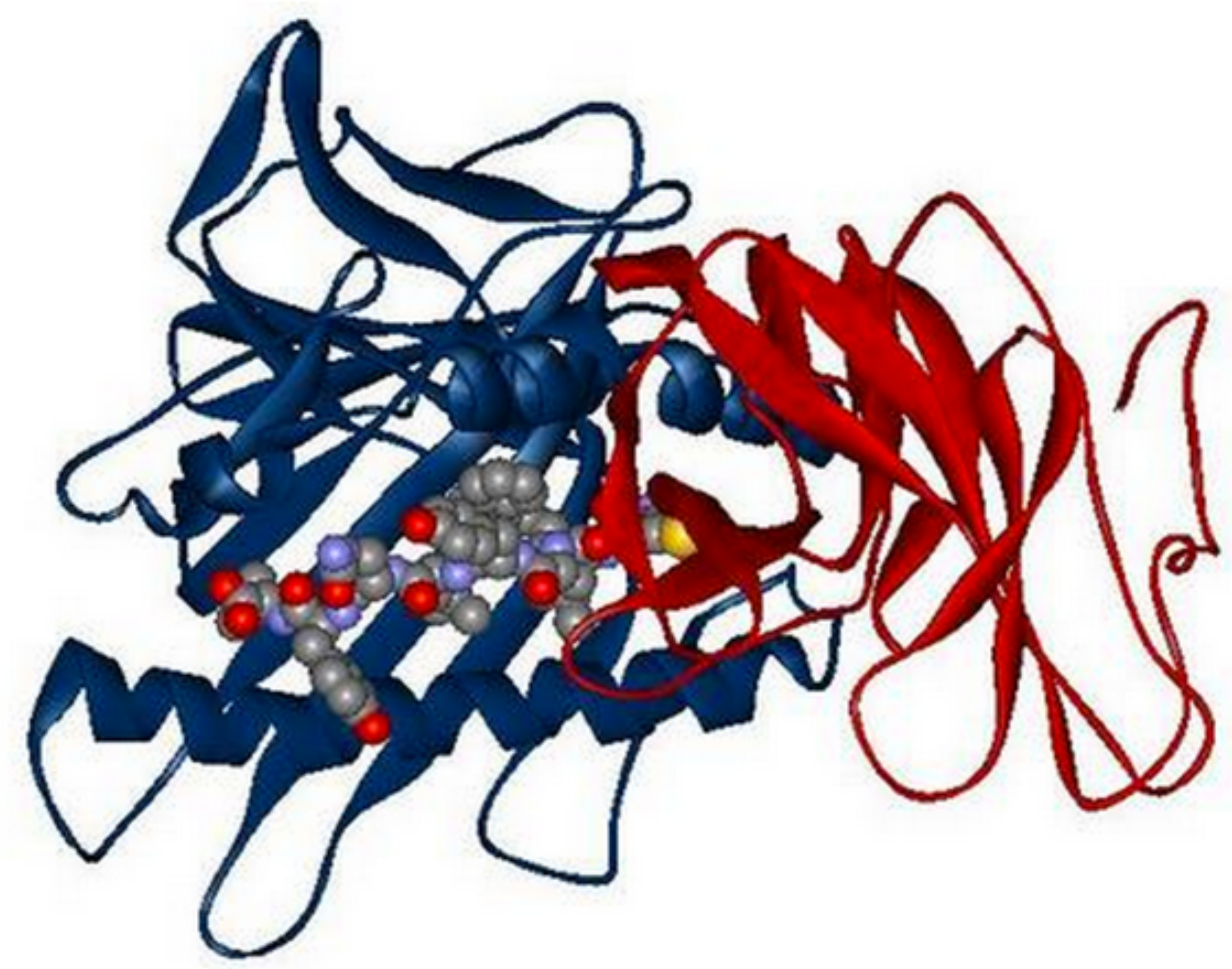


Applications



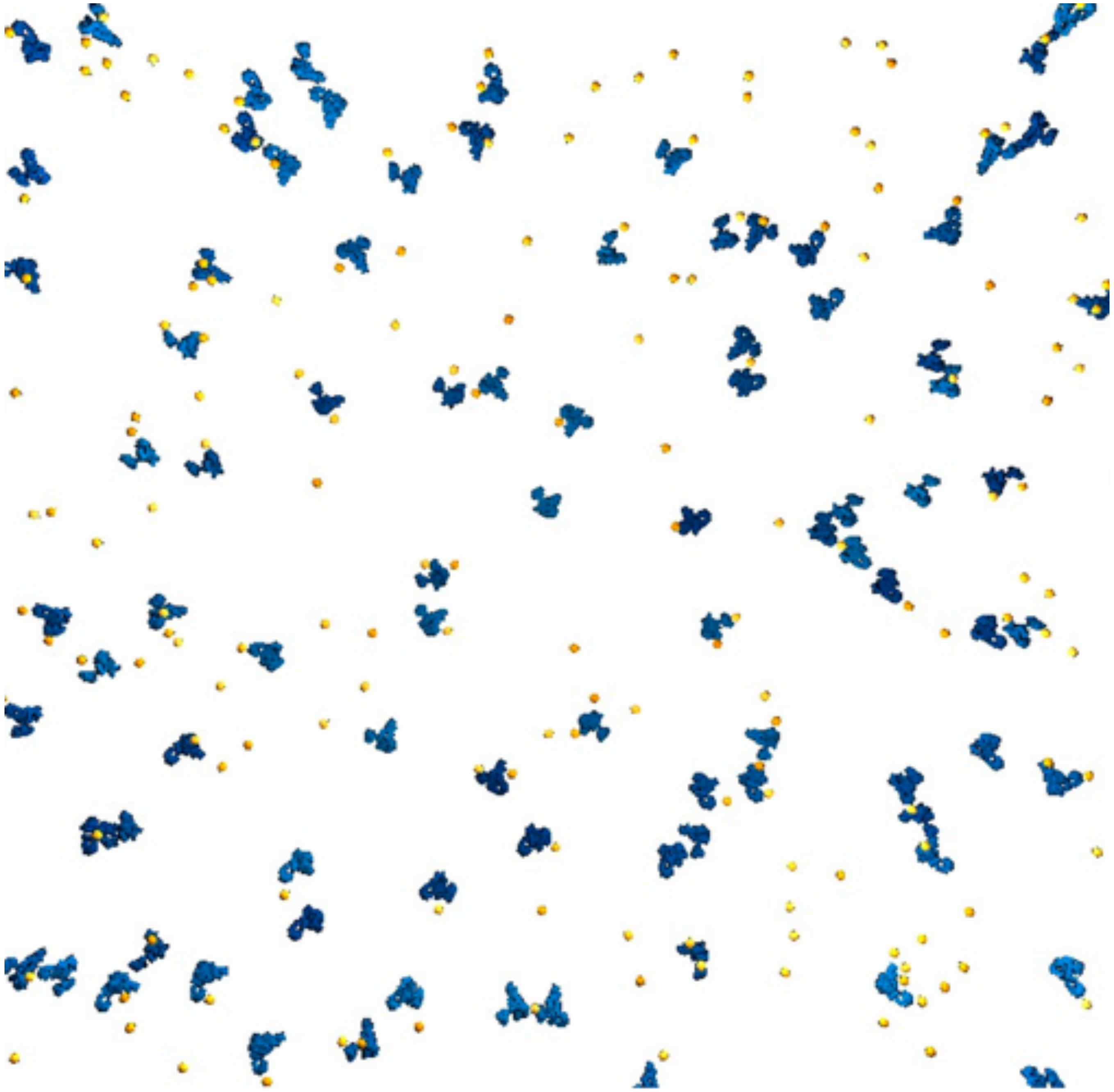
planning.cs.uiuc.edu

Applications



cs.unm.edu/amprg

Applications

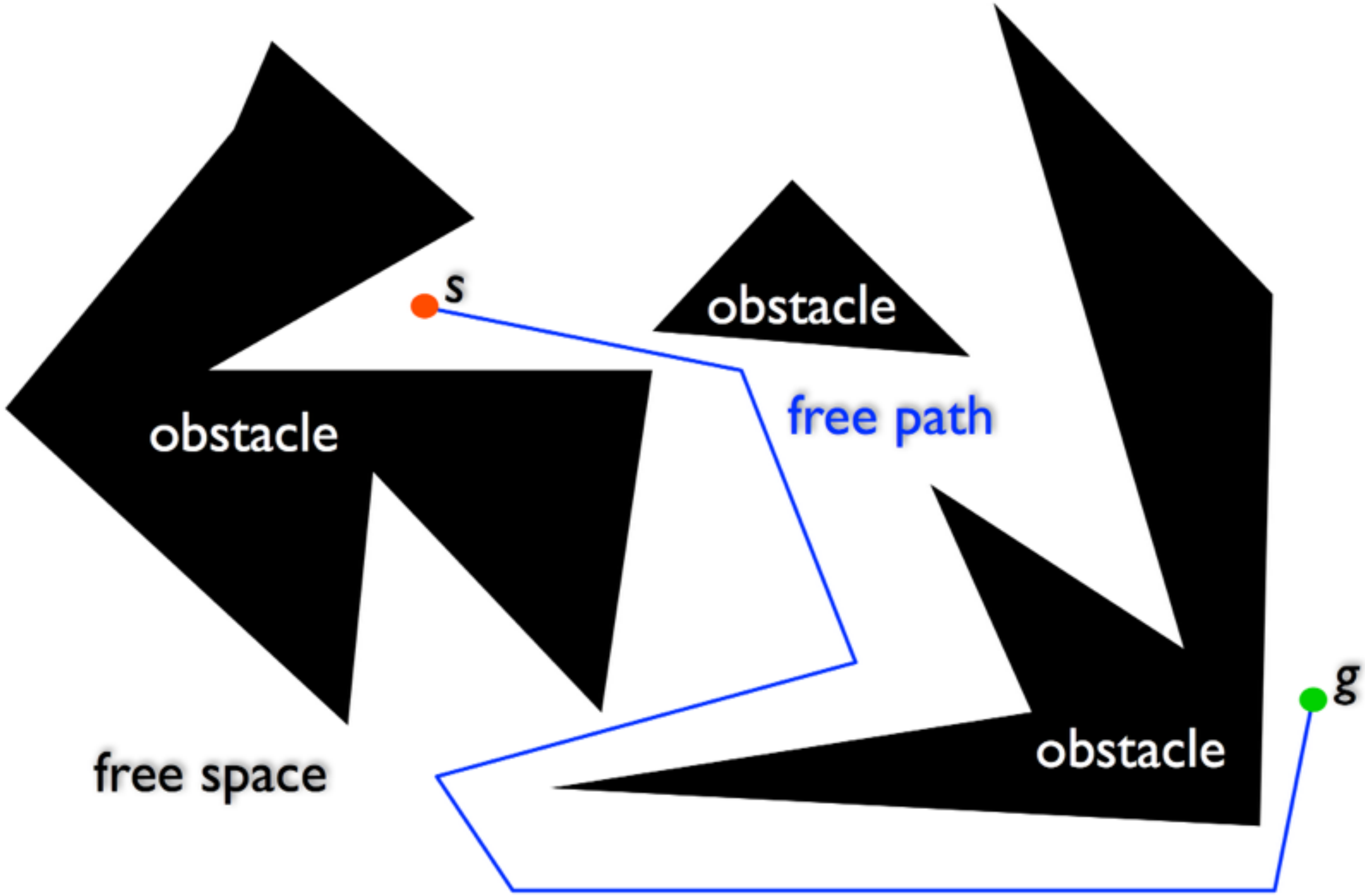


cs.unm.edu/amprg

Problem Definition

- Compute a collision-free path for the robot/agent from a start configuration to a goal configuration
- Inputs
  - Geometry of robot/agent
  - Geometry of environment
  - Start and goal configurations
- Outputs
  - Continuous sequence of robot/agent configurations connecting the start and goal configurations

Problem Definition



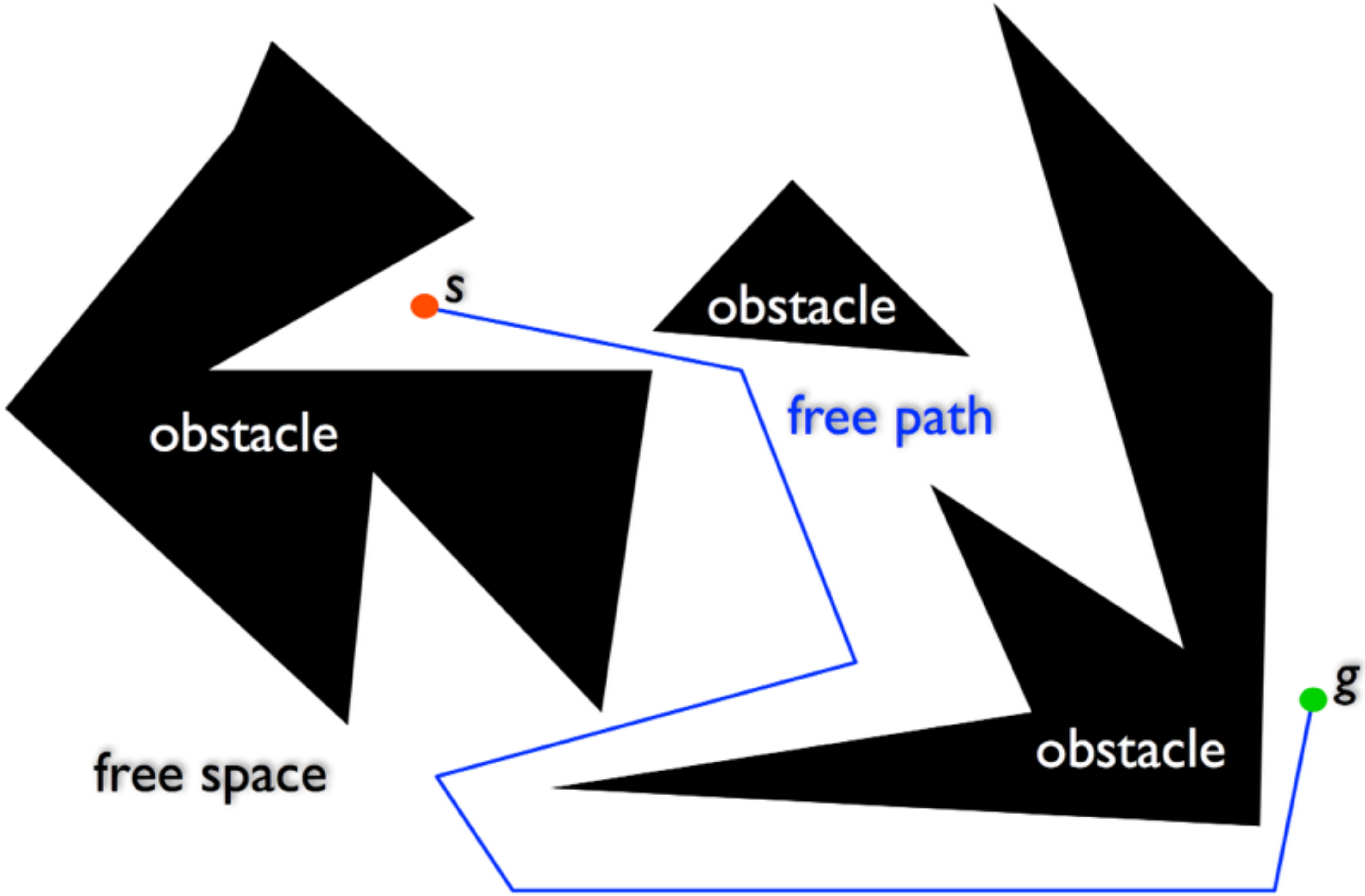
Problem Definition

- Complete - Always return a solution plan if one exists, otherwise indicate there isn't one
- Optimal - Always return the best solution plan under some value metric

Problem Definition

- Completeness - In more than 2D, PSPACE-hard
- Exponential in DOFs, number of obstacles, etc.
- May require computation of entire C-space.
- Doable in simple cases, like 2D with point robot. Easy because C-space is workspace

Problem Definition

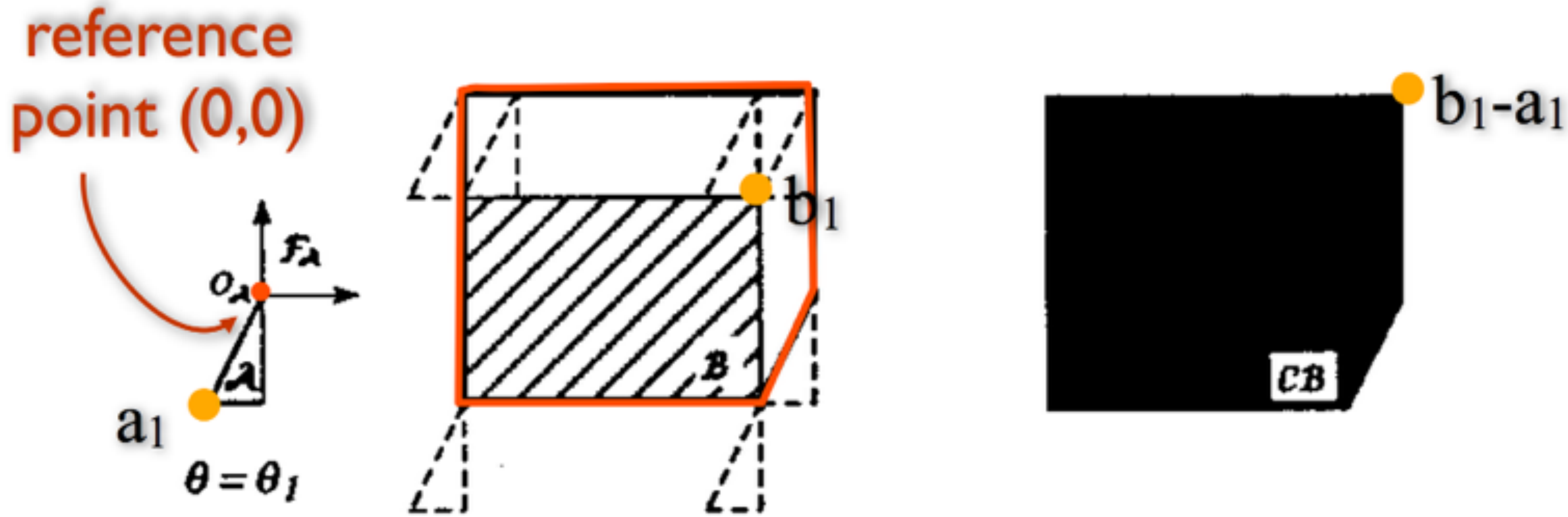


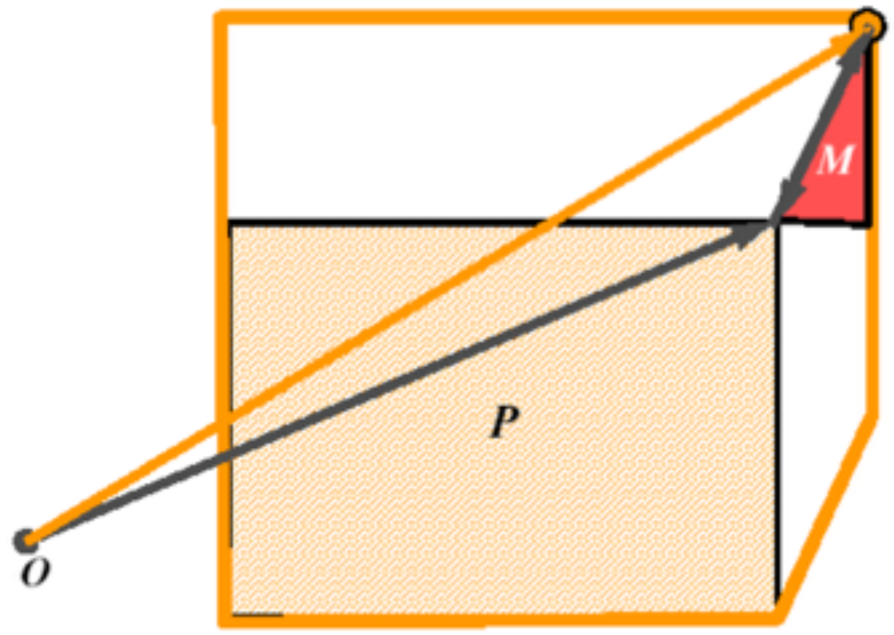
Problem Definition

- What about for something more complex than a point?
- Next most complex - Polygonal robot that translates but does not rotate.
- Can also be done relatively easy in 2D space through Minkowski Sums/Differences

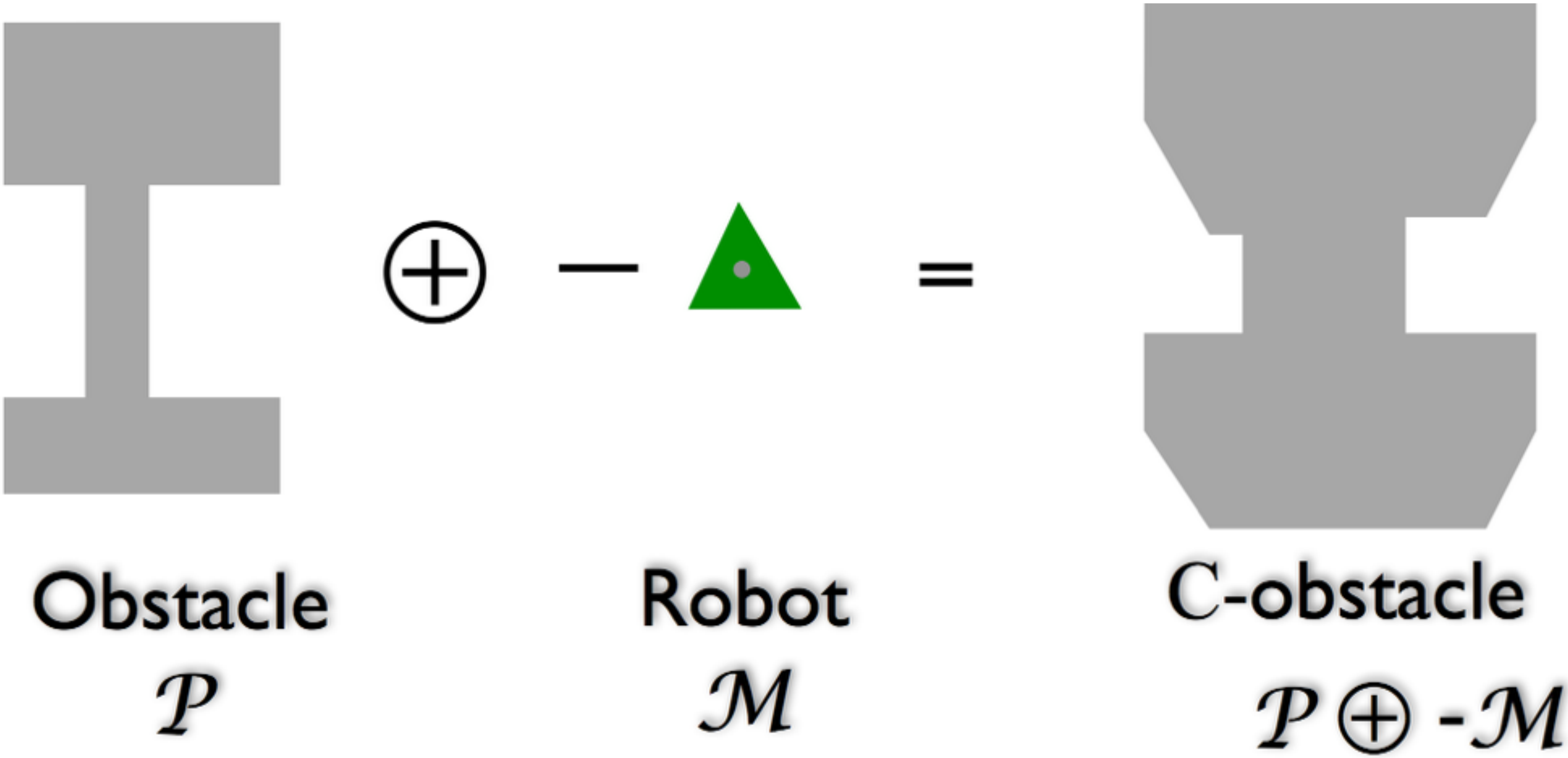
Minkowski Difference

$$CB = B \ominus A = \{b-a \mid a \in A, b \in B\}$$



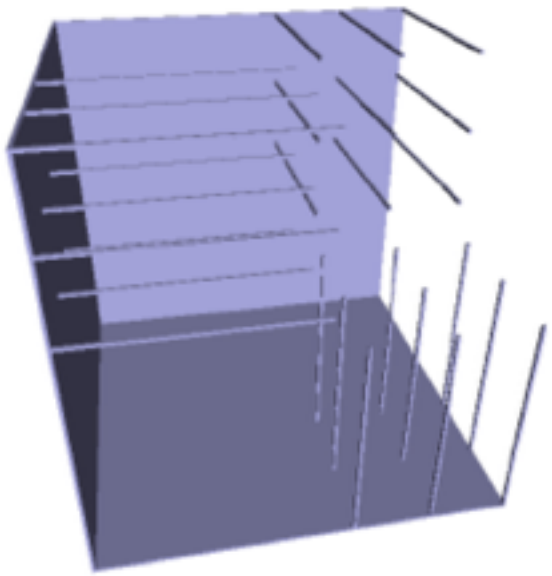


Minkowski Difference

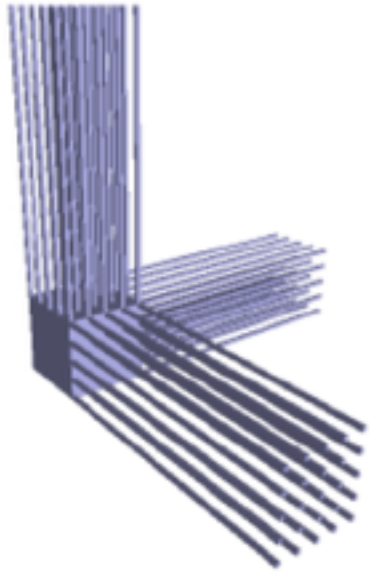


Classic result by Lozano-Perez and Wesley 1979

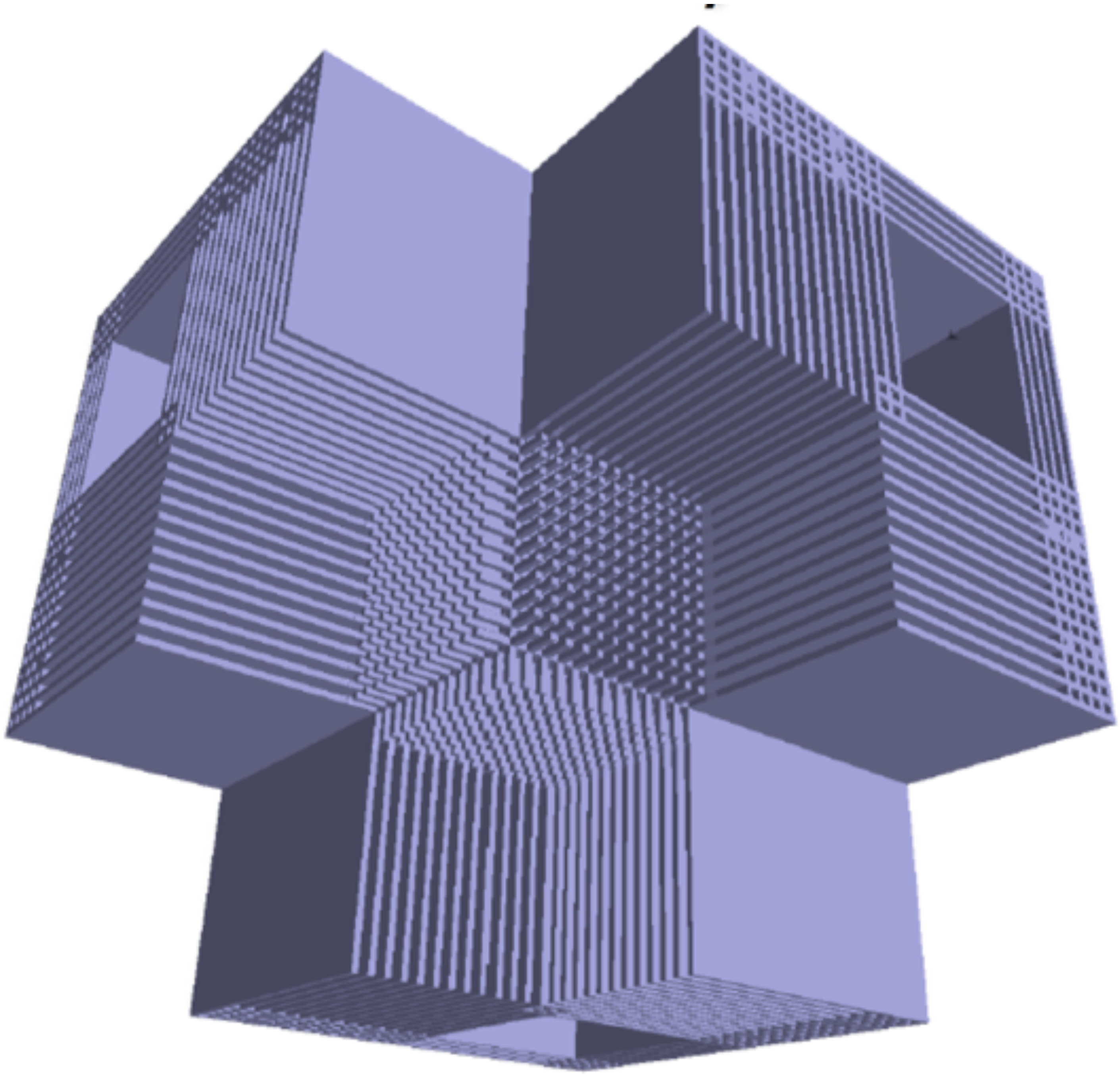
Minkowski Difference



Grate 1 (444 tris)

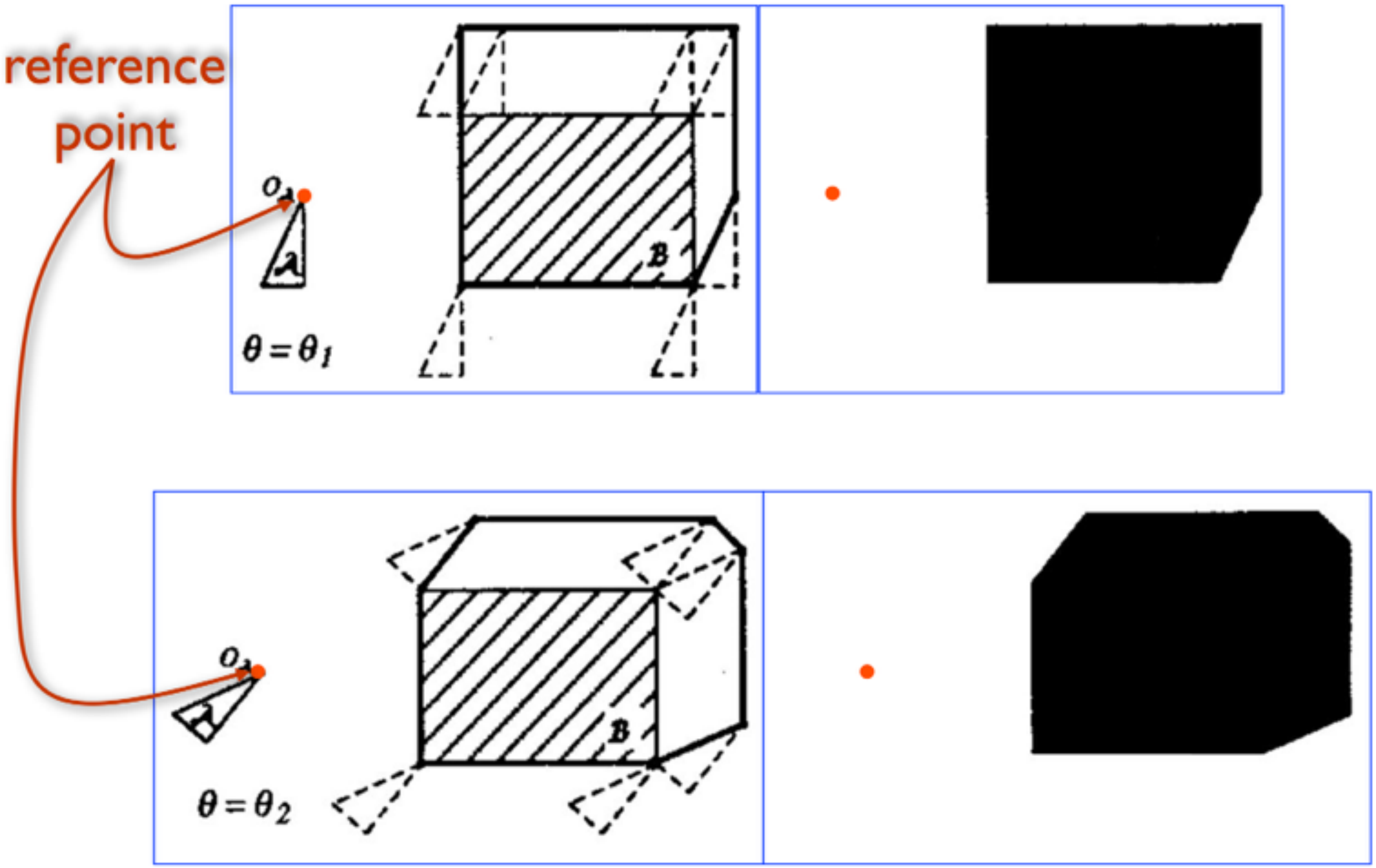


Grate 2 (1, 134 tris)

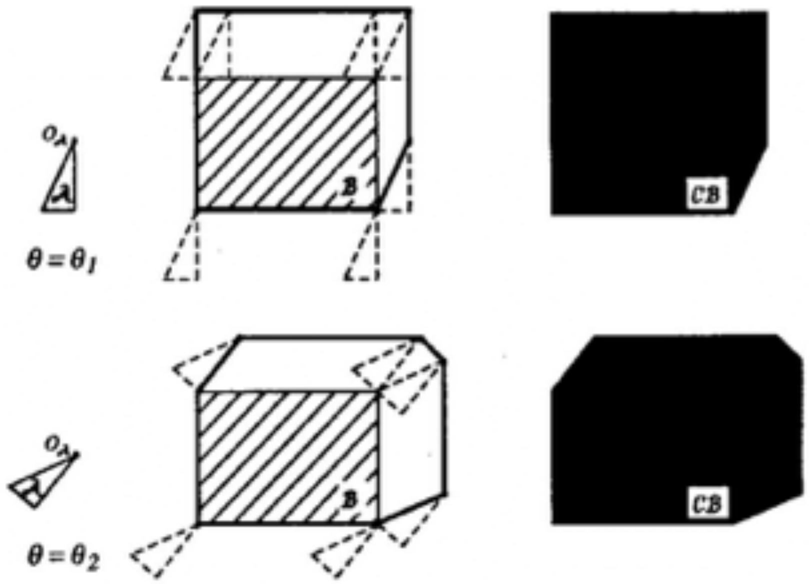


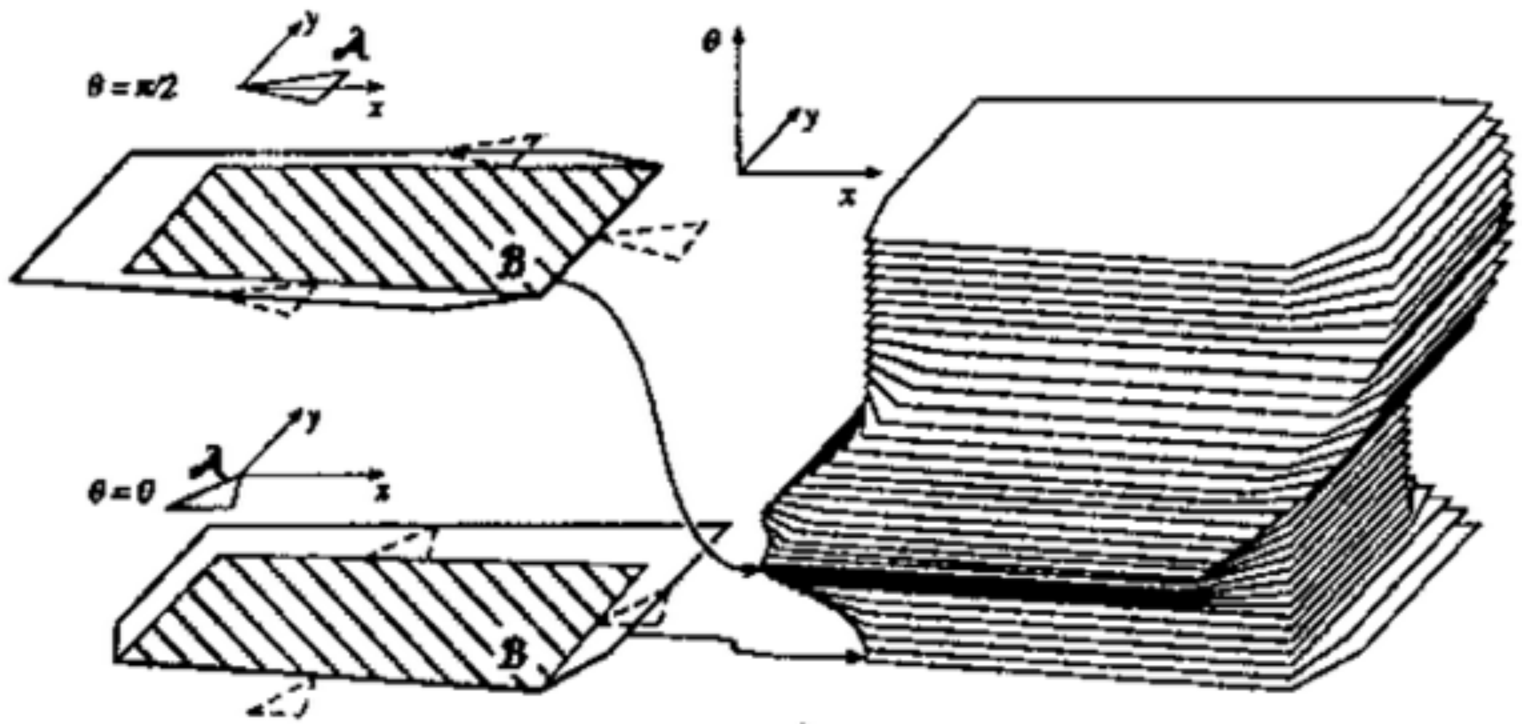
Grate 1 $\oplus$ Grate 2 (Union of 66, 667 prims, 358K tris)

Minkowski Difference



Minkowski Difference





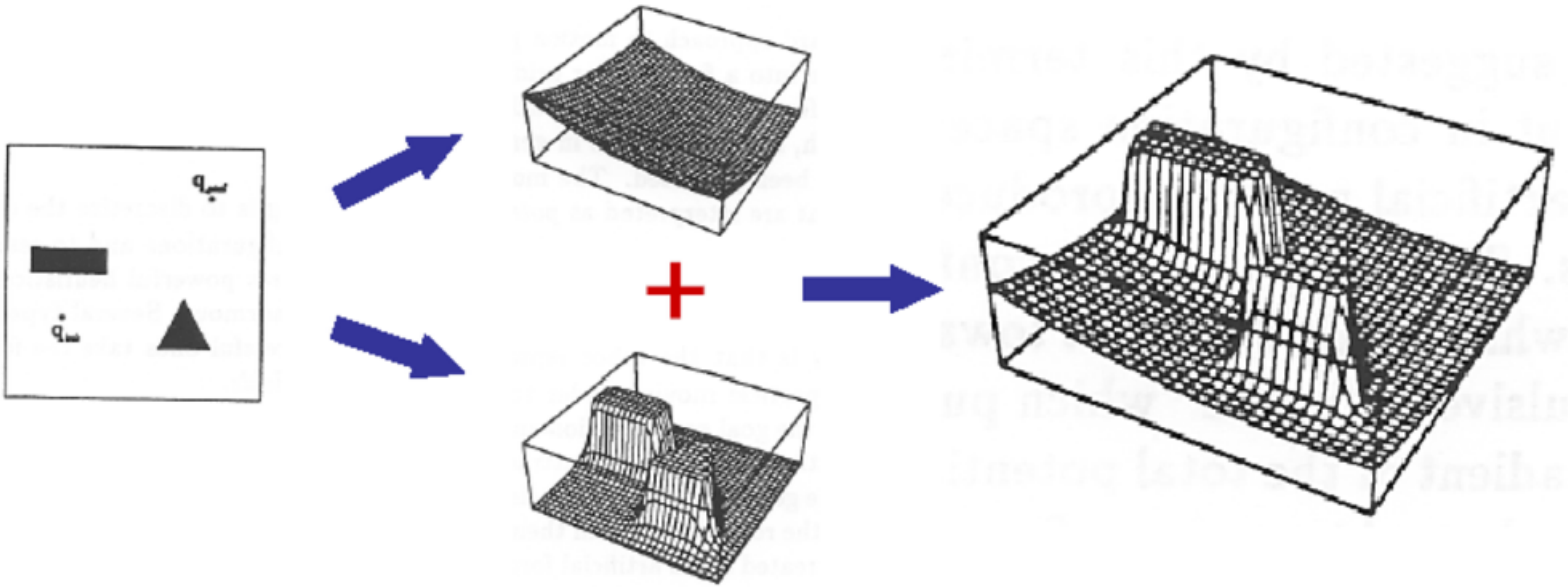
Minkowski Difference

- That's an obstacle in C-space for a mobile robot.
- The problem has now become to navigate a point through this higher dimensional space.

The Point

- A huge amount of motion planning concerns itself with navigating a point through some n-dimensional space.
- Why a point?
  - Points are easy.
    - Lines, Vectors, Graphs etc.
- Is this even useful?
  - Abstraction
  - Approximation
- How?

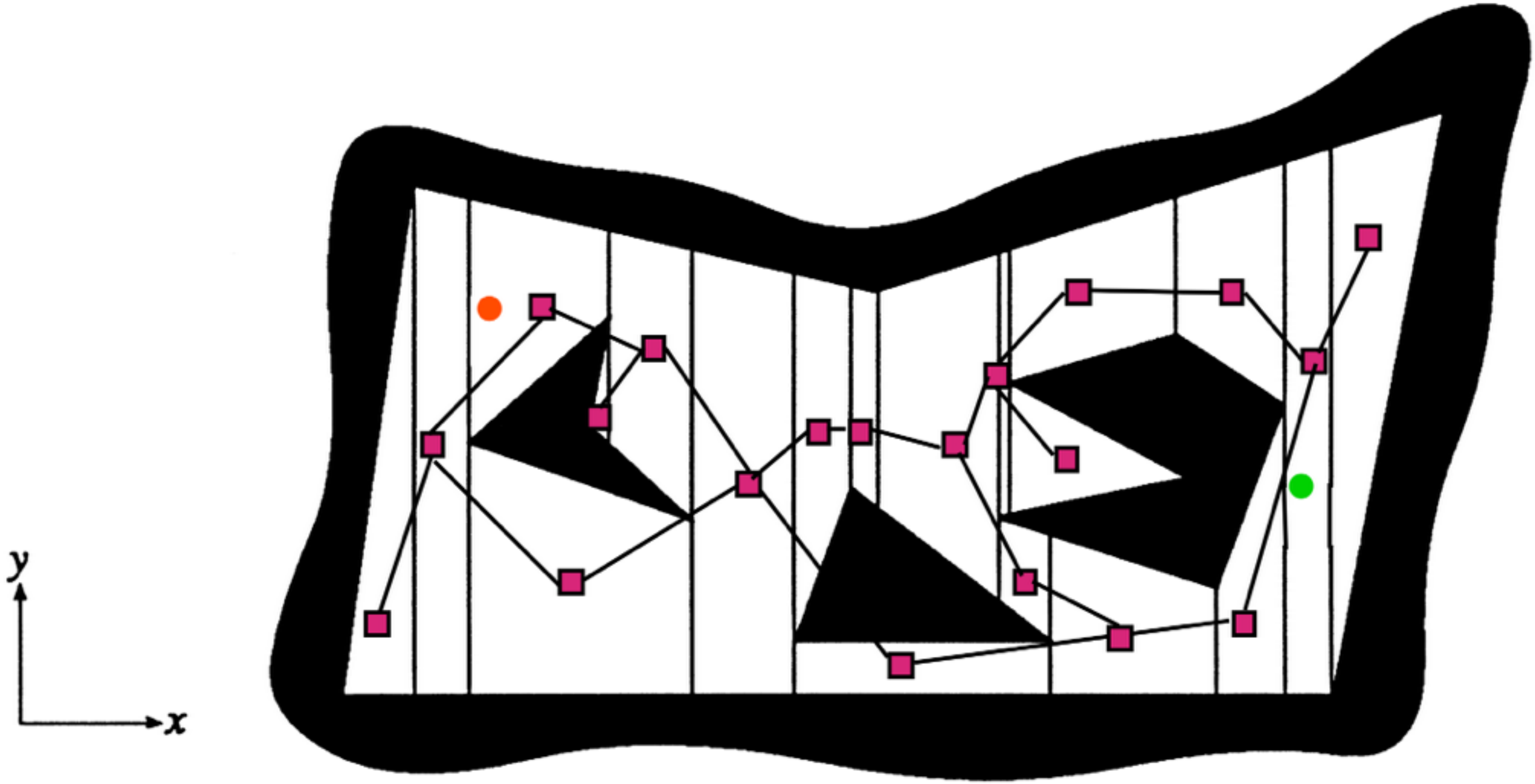
Potential Fields



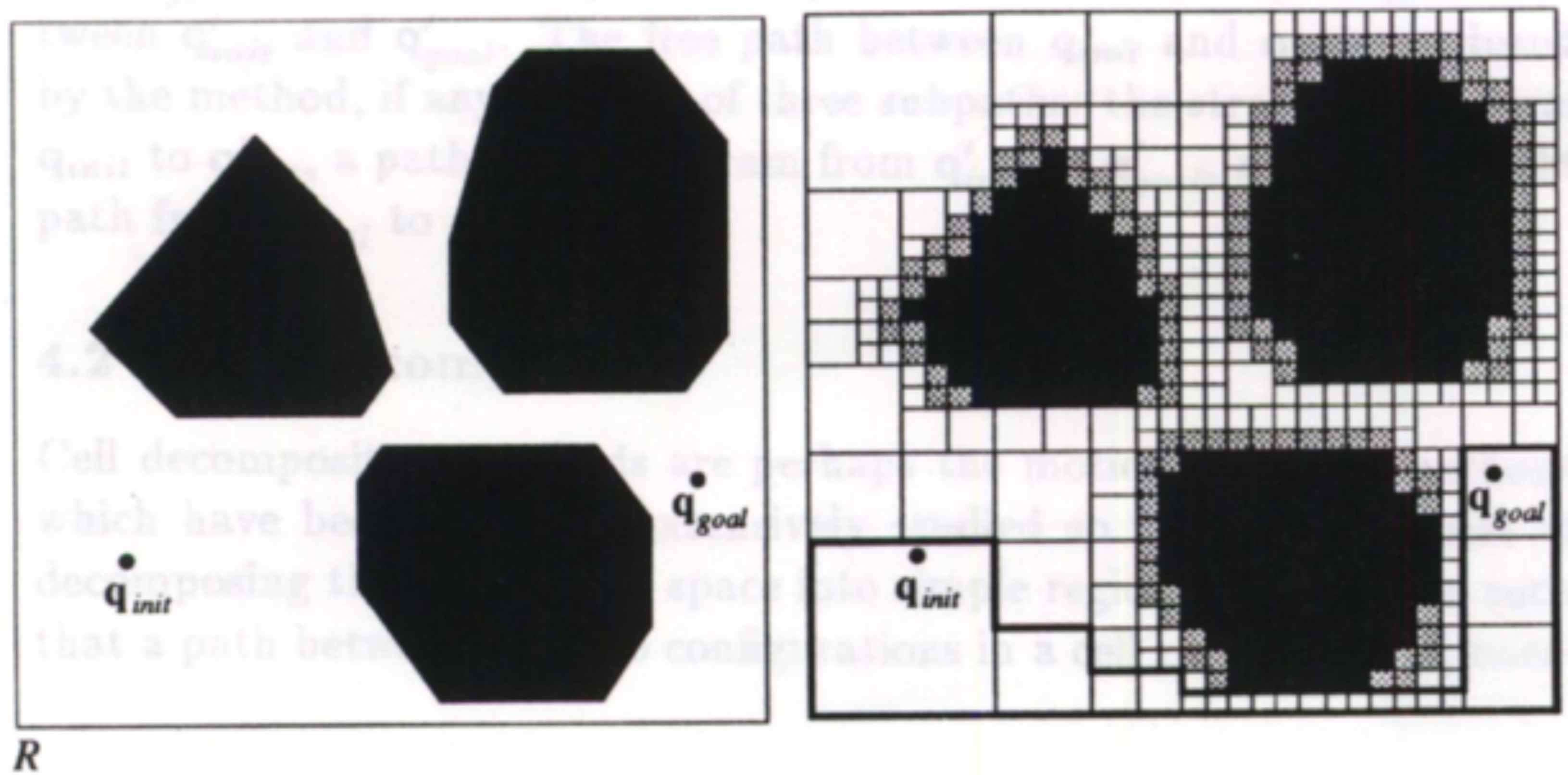
Back to low dimensions

- How to plan the motion of a point?
- Discretize the space, construct a graph, search the graph.

Trapezoidal Decomposition

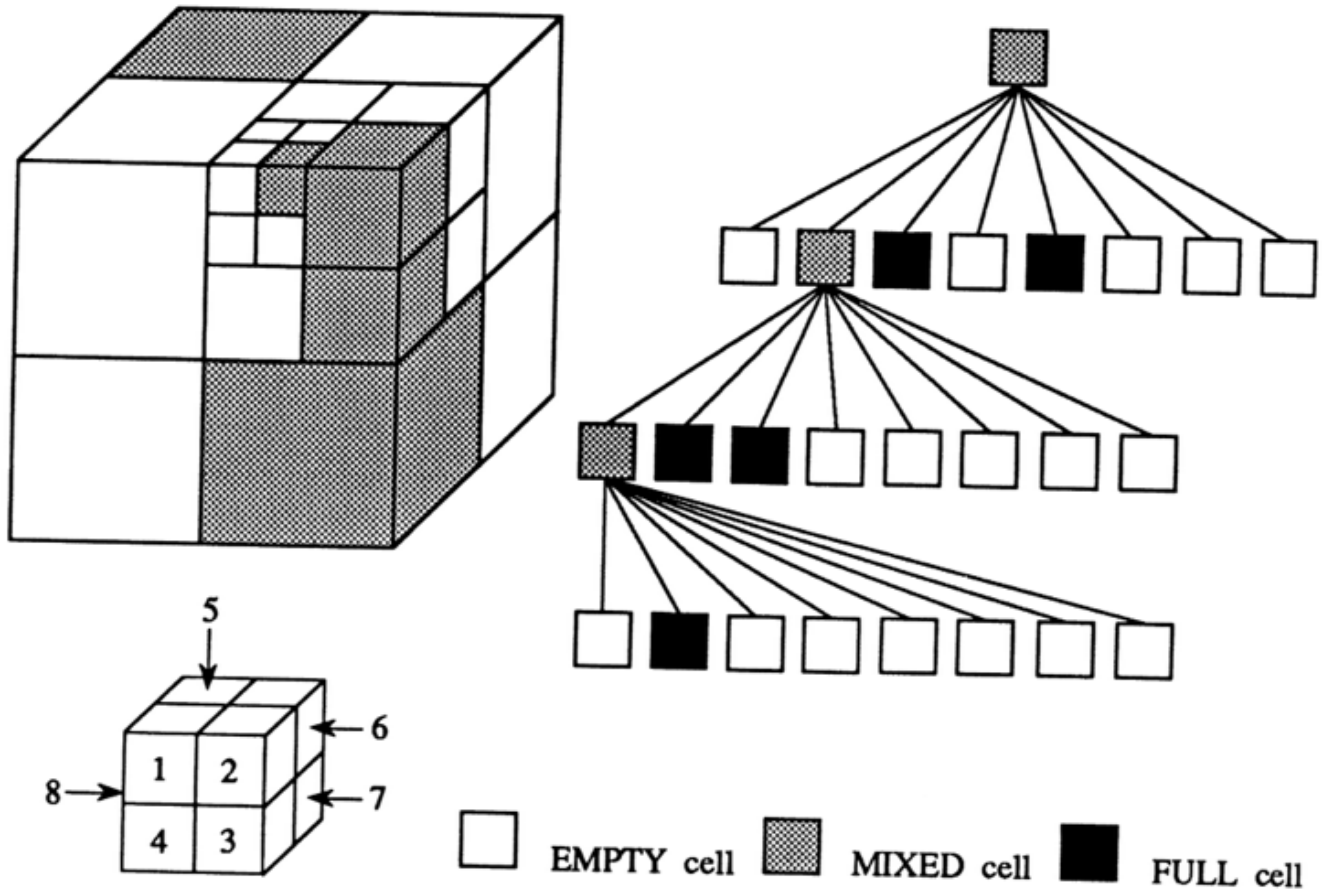


Quadtree Decomposition



(a) (b)

Octree Decomposition



The Problem

- Methods like these require a model of C-space
- These spaces becomes difficult/infeasible beyond three dimensions.
- How do we get around this?

The Point

- Describing the space is hard, but describing the state of a single point may not be.

Roadmaps

- Lets build a “roadmap” of the space, which requires much less evaluation.

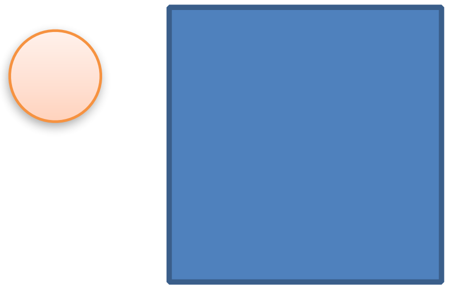
Probabilistic Road Maps - PRM

- Learning Phase
  - Sample free points
  - Link samples to learn connectivity
  - Precomputed
- Query Phase
  - Add start and goal to roadmap
  - Connect to nearest neighbor
  - Compute path from start to goal
  - Multiple queries per road map

Probabilistic Road Maps - PRM

- Interactive Demo: http://robotics.cs.unc.edu/interactive/prm.html

Rapidly Exploring Random Trees - RRT

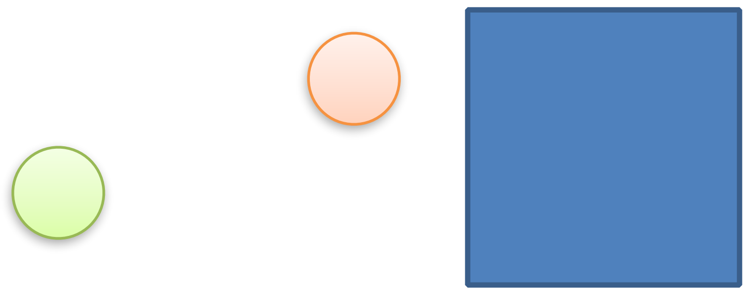


[LaValle, Kuffner 2001]

Rapidly Exploring Random Trees - RRT

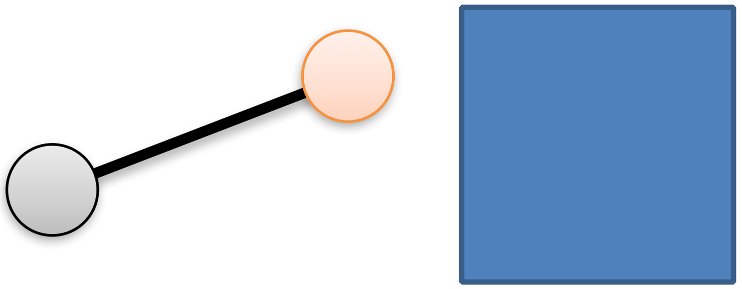
[LaValle, Kuffner 2001]

Rapidly Exploring Random Trees - RRT



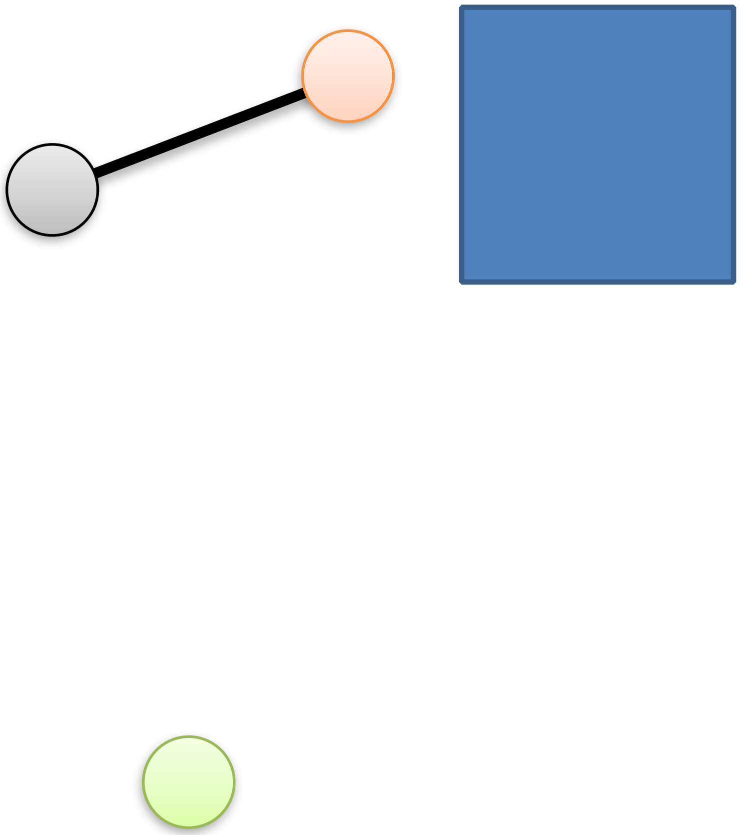
[LaValle, Kuffner 2001]

Rapidly Exploring Random Trees - RRT



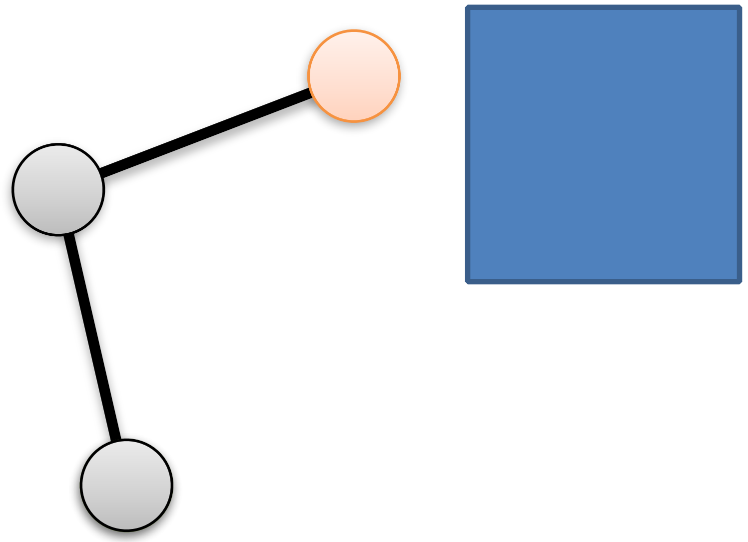
[LaValle, Kuffner 2001]

Rapidly Exploring Random Trees - RRT



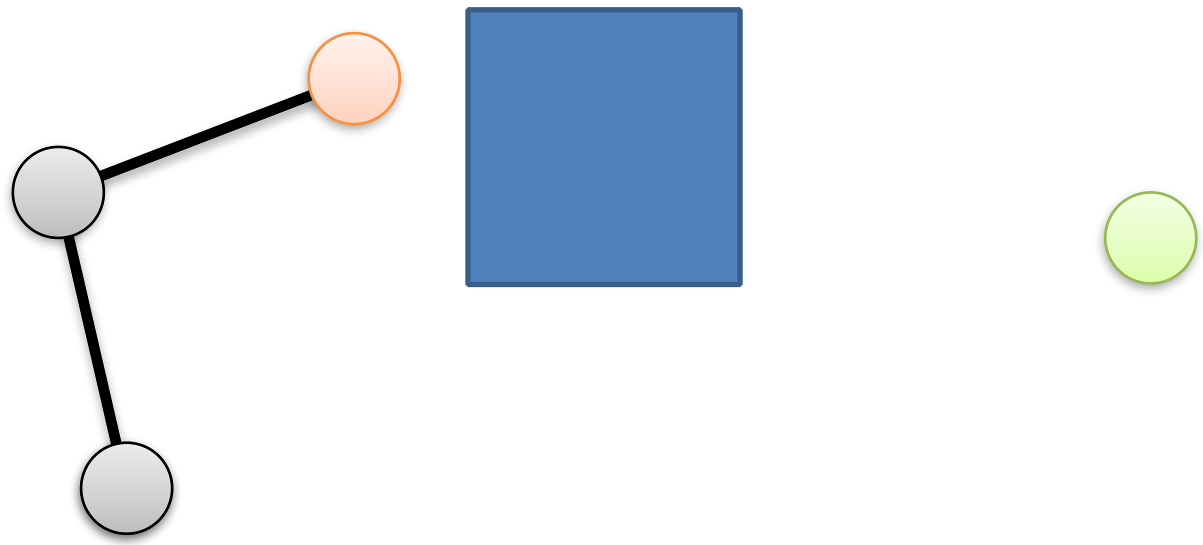
[LaValle, Kuffner 2001]

Rapidly Exploring Random Trees - RRT



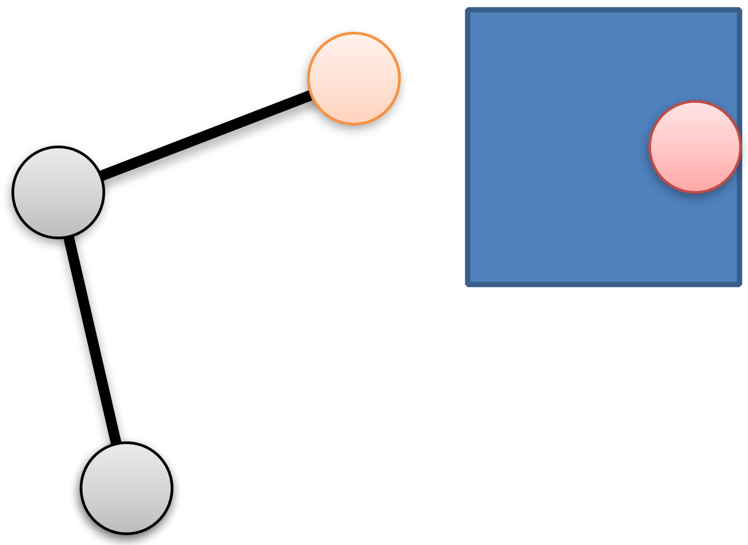
[LaValle, Kuffner 2001]

Rapidly Exploring Random Trees - RRT



[LaValle, Kuffner 2001]

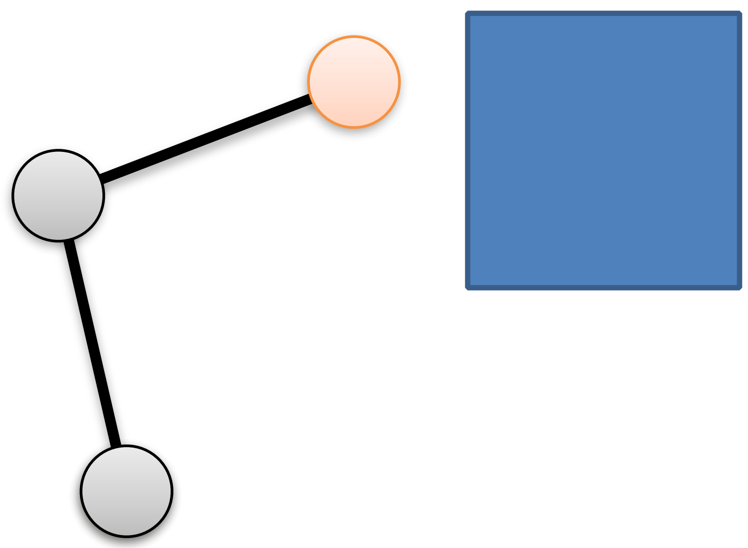
Rapidly Exploring Random Trees - RRT



THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

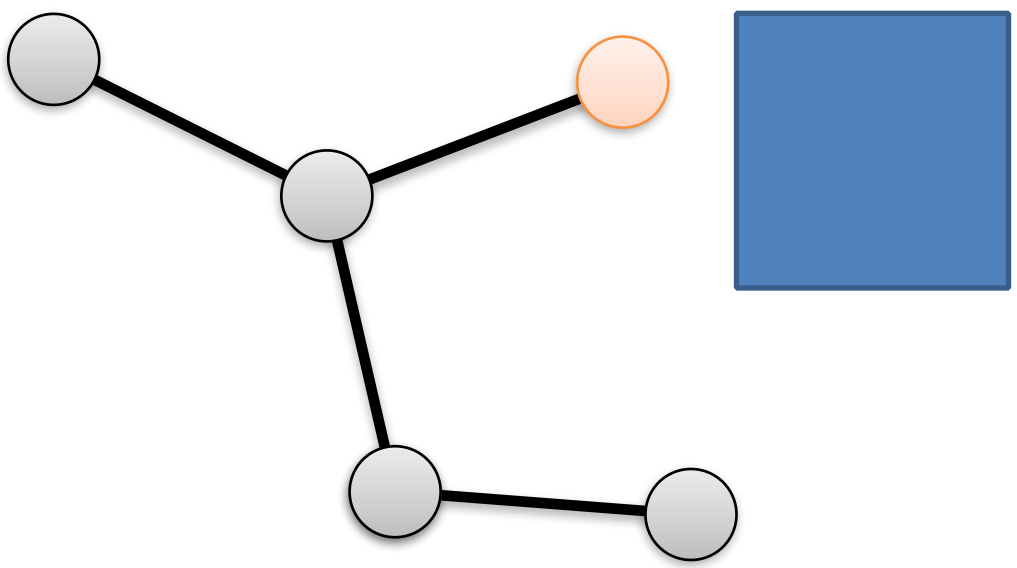
[LaValle, Kuffner 2001]

Rapidly Exploring Random Trees - RRT



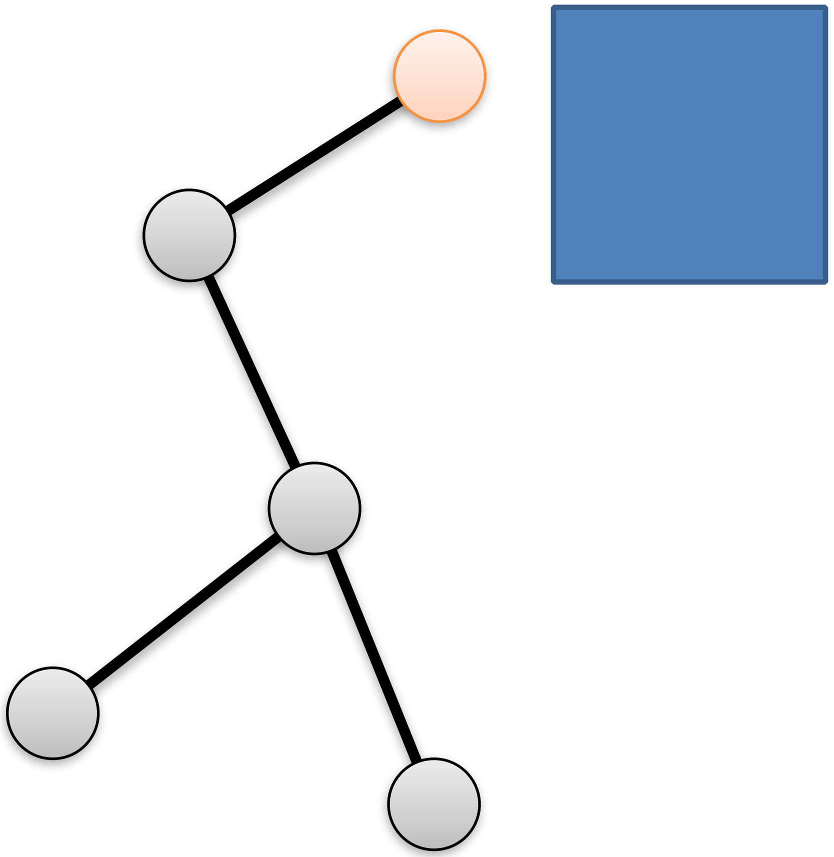
[LaValle, Kuffner 2001]

Rapidly Exploring Random Trees - RRT



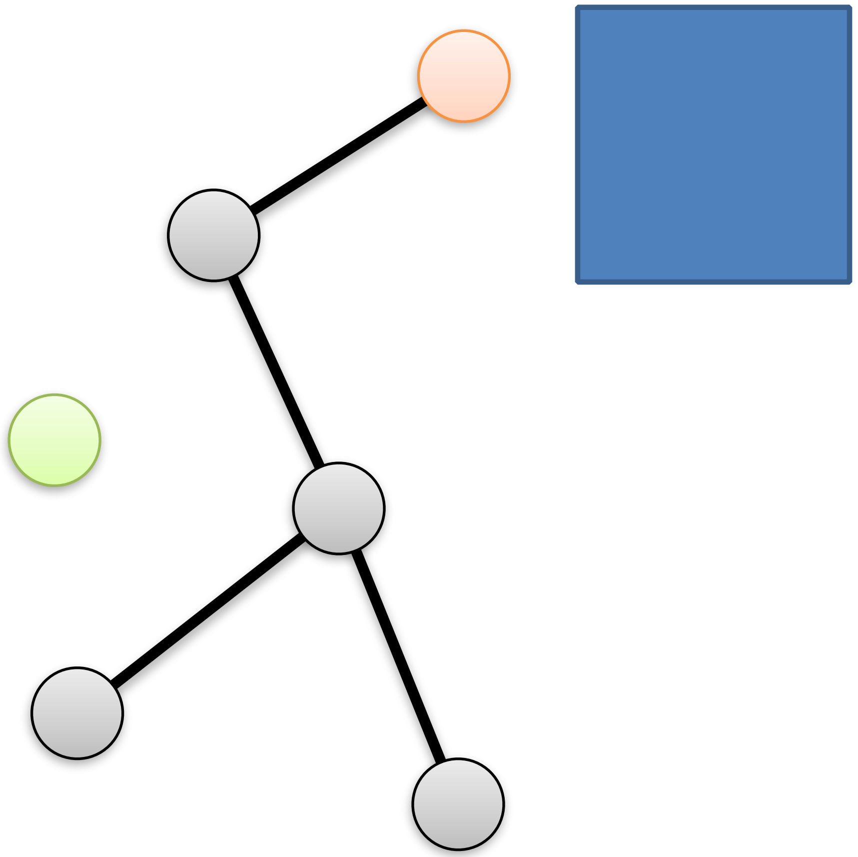
[LaValle, Kuffner 2001]

RRT*



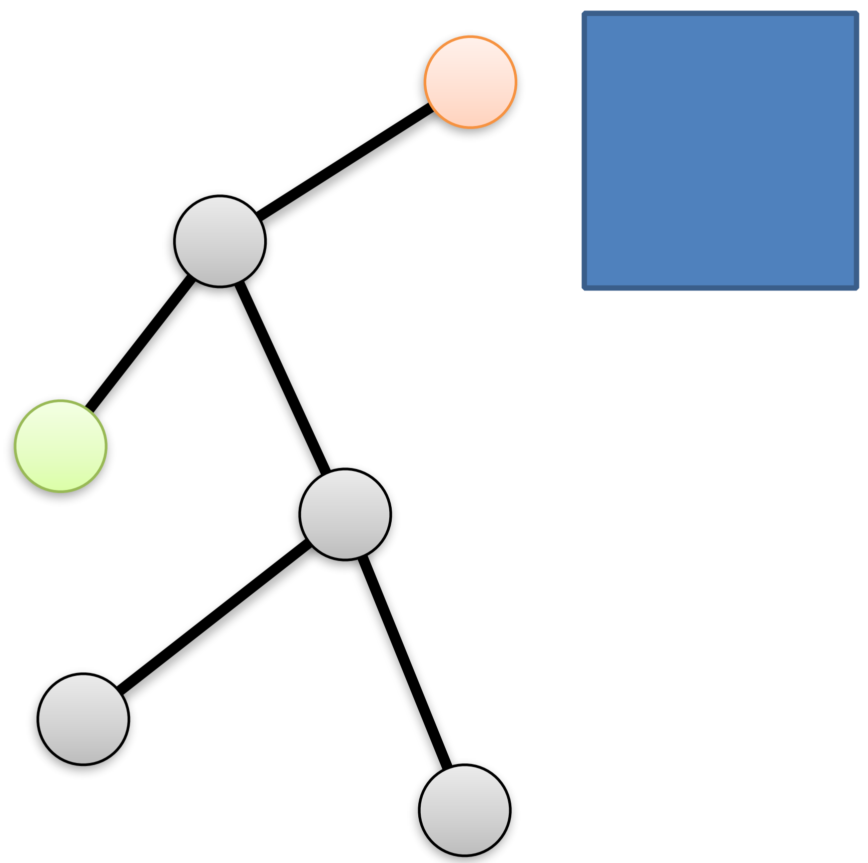
[Karaman, Frazzoli 2010]

RRT*



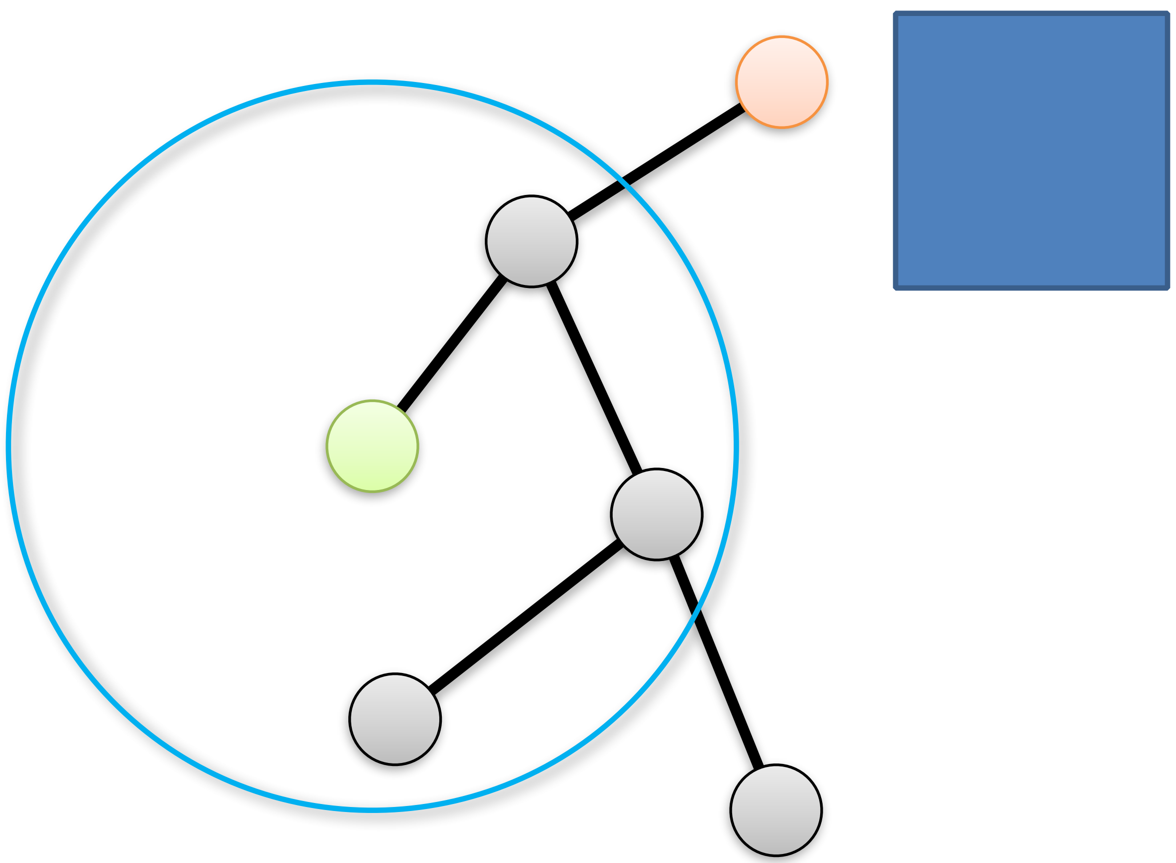
[Karaman, Frazzoli 2010]

RRT*



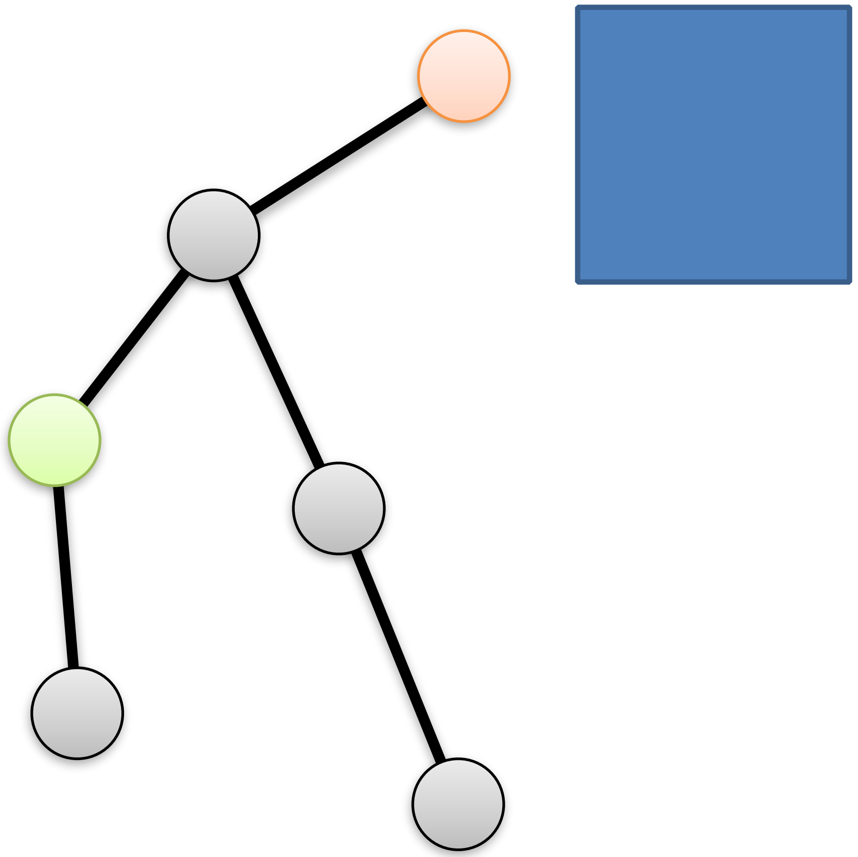
[Karaman, Frazzoli 2010]

RRT*



[Karaman, Frazzoli 2010]

RRT*



[Karaman, Frazzoli 2010]

Intuition

- Describe your system in terms of some high dimensional space
  - C-space
  - State space
  - Workspace
  - Trajectory space
  - A combination
- Plan a path through that space under some constraints

Space Choices

- Choice is frequently problem dependent
- Frequently require some approximation, so what model resolution is sufficient?
- May be influenced by the capabilities of your controller
  - One end of the spectrum, control propagation
  - Other end, maps.

Additional Considerations

- What space will allow for easy and effective implementation or adaptation of pre-existing algorithms?
- Space construction will affect topology, connectivity, obstacle definitions etc.

Increased Complexity

- Dynamic Environments
  - Ideas?
- Noisy Sensing/Actuation
  - Other Ideas?
- Nonholonomy
  - Even More Ideas?

Conclusion

- Many different classes of motion planning algorithms
- It is very difficult to generalize them
- The intuition gained from thinking about the abstractions will help you to understand the approaches as you encounter them.

Questions?

- Many images curtesy of Dr. Alterovitz' Robotics course.