# Algorithmic Motion Planning Primer

이화여자대학교 컴퓨터공학

김영준

http://graphics.ewha.ac.kr/

**To download updated tutorial slides**

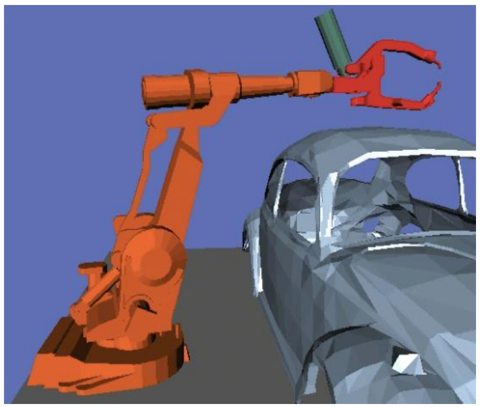
# Slide Credits

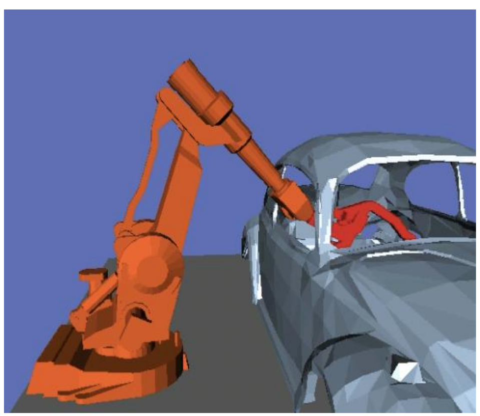
- J.-C. Latombe's lectures at Stanford

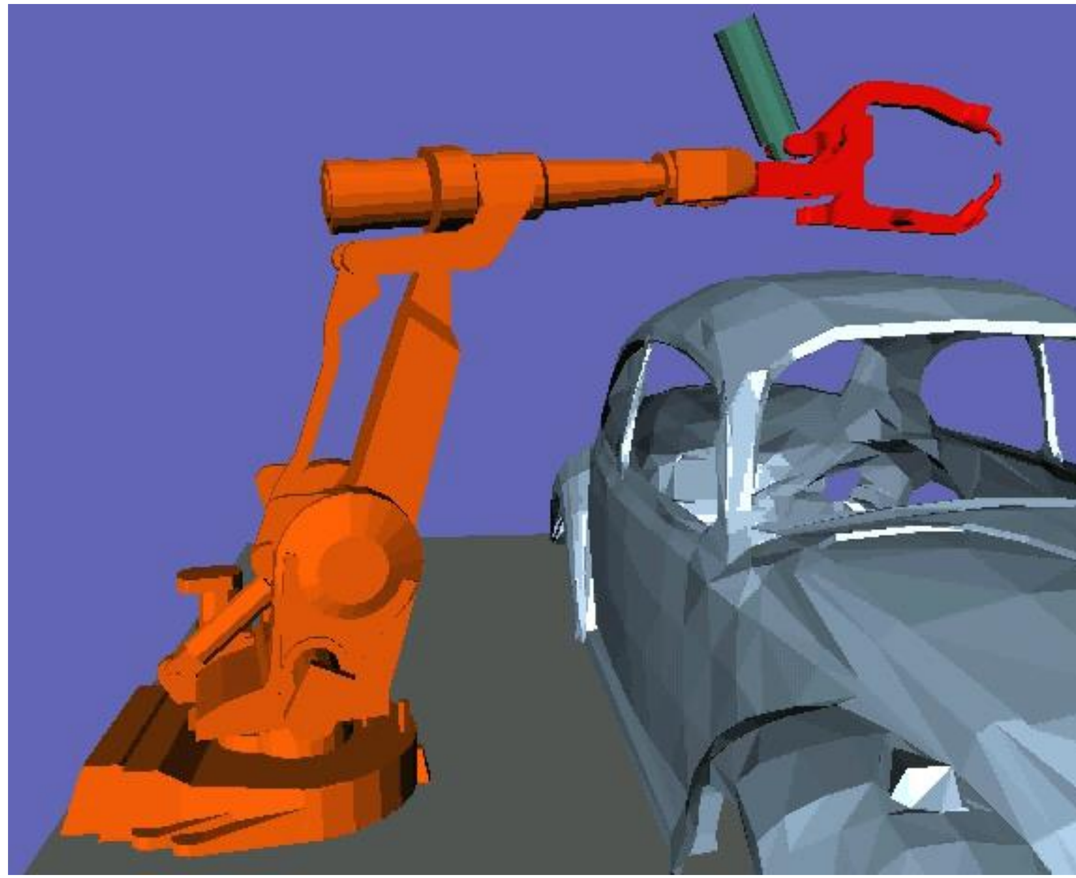
# Robot Control Paradigm


[Aledebaran Robotics]

# Motion Planning


Initial configuration

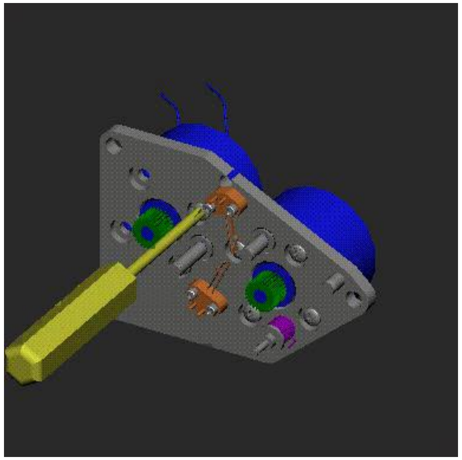

Final configuration


Collision-free Motion

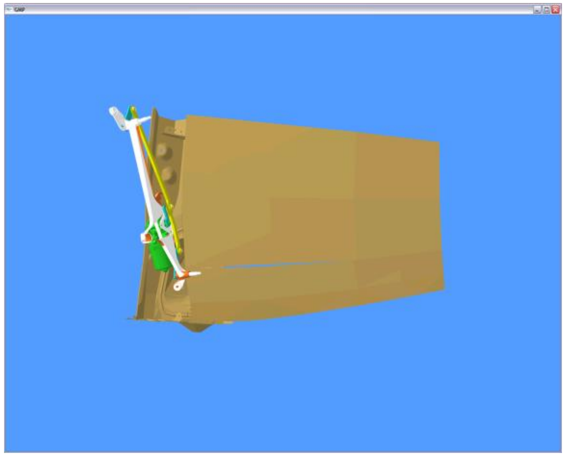
# Goal of Motion Planning

- Compute **motion strategies**, e.g.:
  - geometric paths
  - time-parameterized trajectories
  - sequence of sensor-based motion commands
- To achieve **high-level goals**, e.g.:
  - go to A without colliding with obstacles
  - assemble product P
  - build map of environment E
  - find object O

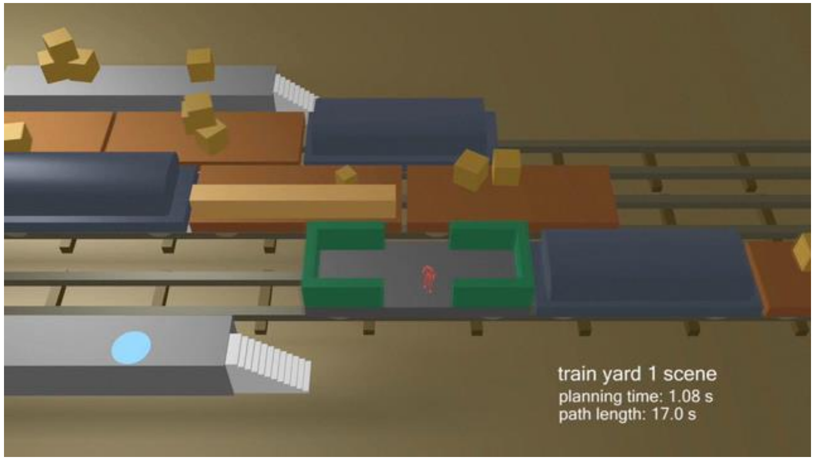
# Applications



Assembly

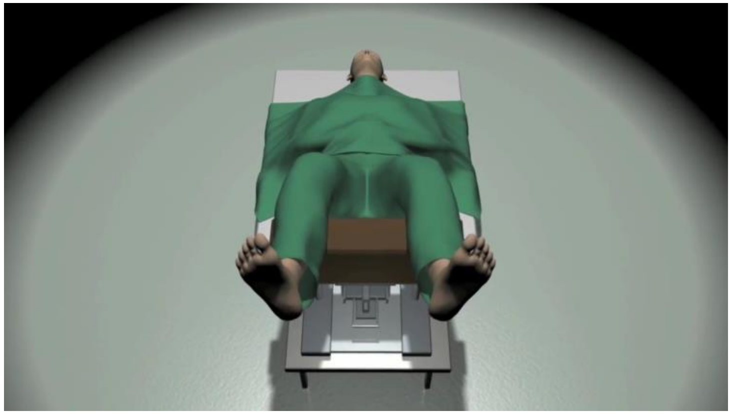


Disassembly

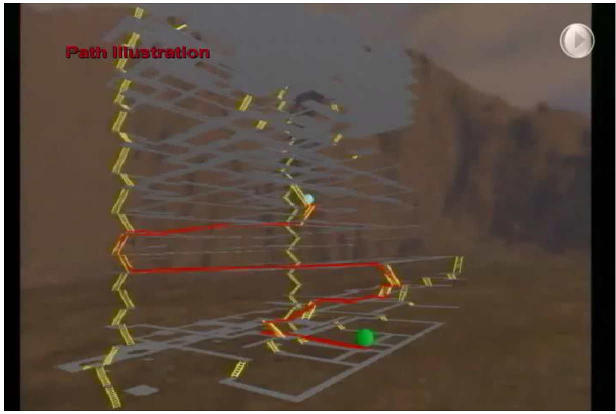



Computer Animation

Autonomous Vehicle

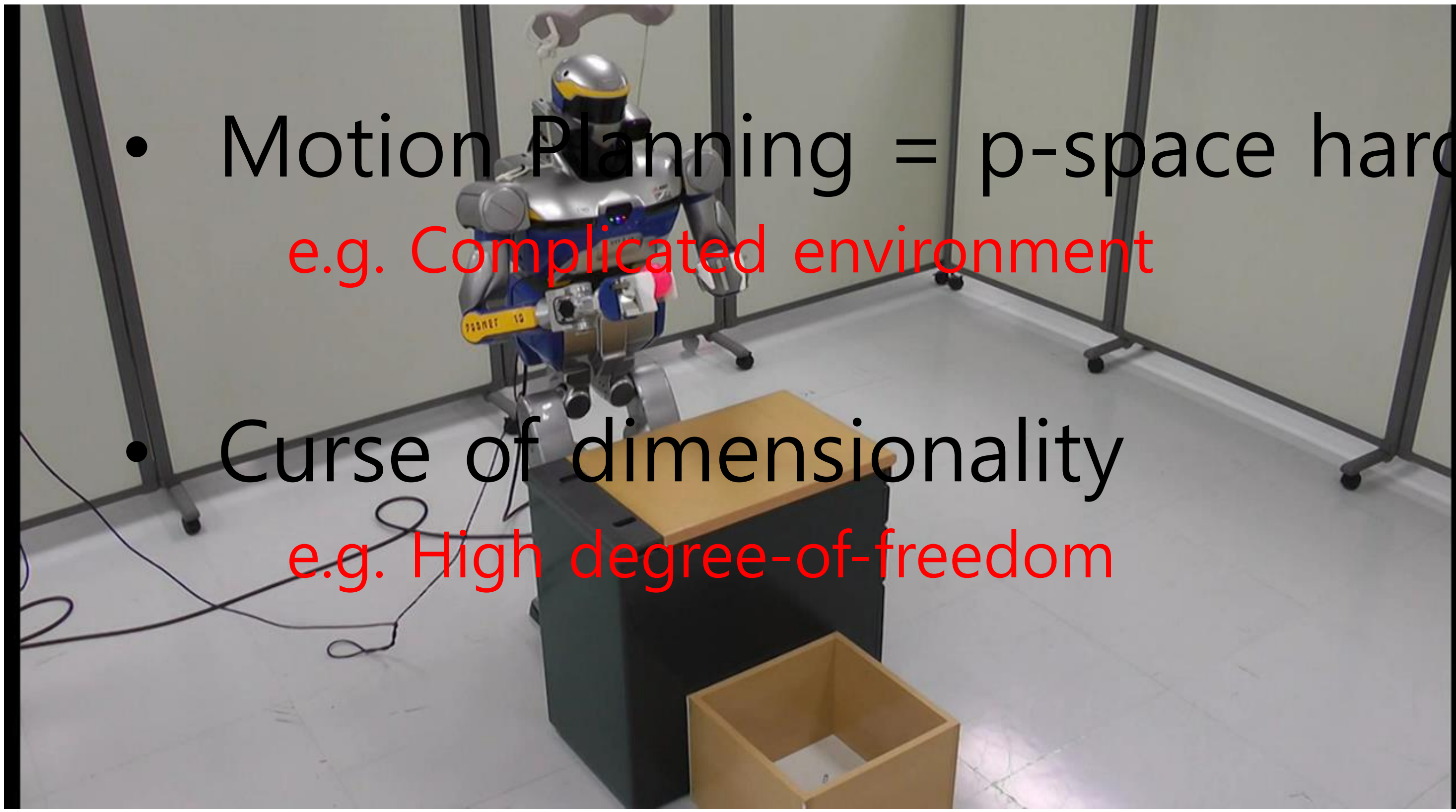


Surgical Planning




Navigation

# Challenges in Motion Planning



- Motion Planning = p-space hard
  - e.g. Complicated environment
- Curse of dimensionality
  - e.g. High degree-of-freedom

Optimization-based Motion Planning for Humanoids : hours ~ days

[A. Kheddar]

# Overview

- Classification
- Basic concepts
- Sampling-based algorithms
- Motion planning research @ Ewha

# CLASSIFICATION OF MOTION PLANNING ALGORITHMS

# Methodology

- Combinatorial planning
- Sampling-based planning
- Potential field-based planning
- Optimization-based planning

# Completeness

| | Complete | Semi-complete | Resolution complete | Probabilistically complete |
|---|---|---|---|---|
| When a path exists | Find a path | Find a path | Find a path | Find a path with a high probability |
| When no path exists | Report path non-existence | No termination | Report path non-existence in given resolution | No termination |

# Constraints

| | Holonomic | Non holonomic | Kino dynamic |
|---|---|---|---|
| Geometric | O | O | O |
| Kinematic | O | O | O |
| Quasistatic | | O | O |
| Dynamic | | | O |

# Methodology

- Combinatorial planning
- Sampling-based planning
- Potential field-based planning
- Optimization-based planning

# Completeness

| | Complete | Semi-complete | Resolution complete | Probabilistically complete |
|---|---|---|---|---|
| When a path exists | Find a path | Find a path | Find a path | Find a path with a high probability |
| When no path exists | Report path non-existence | No termination | Report path non-existence in given resolution | No termination |

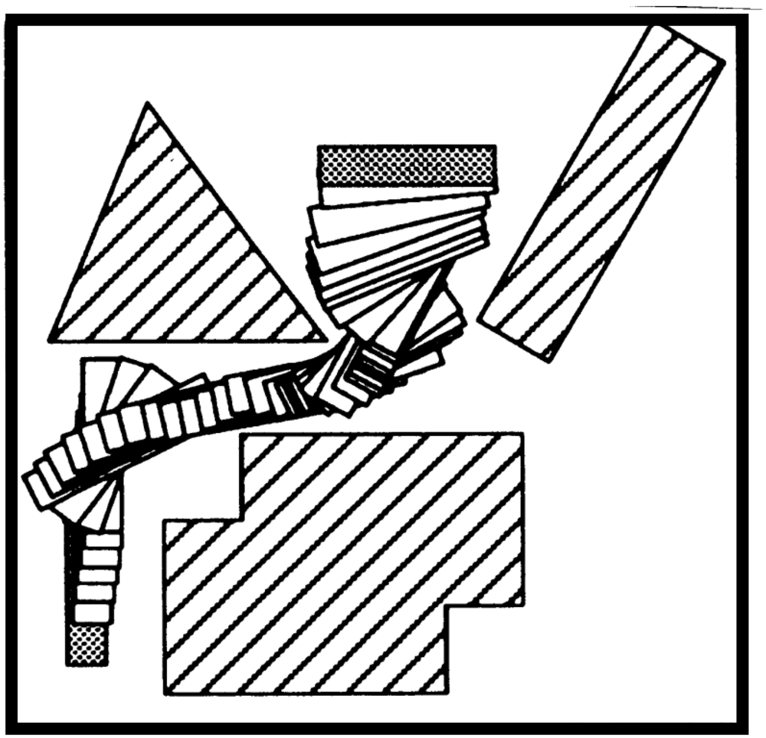
# Constraints

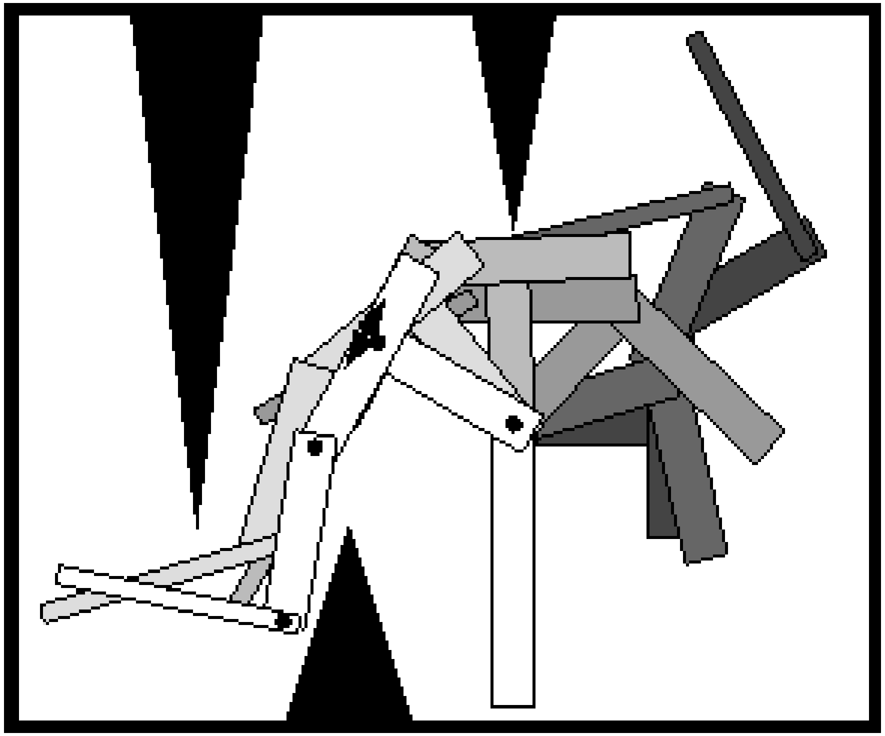
| | Holonomic | Non holonomic | Kino dynamic |
|---|---|---|---|
| Geometric | ○ | ○ | ○ |
| Kinematic | ○ | ○ | ○ |
| Quasistatic | | ○ | ○ |
| Dynamic | | | ○ |

# BASIC CONCEPTS IN MOTION PLANNING

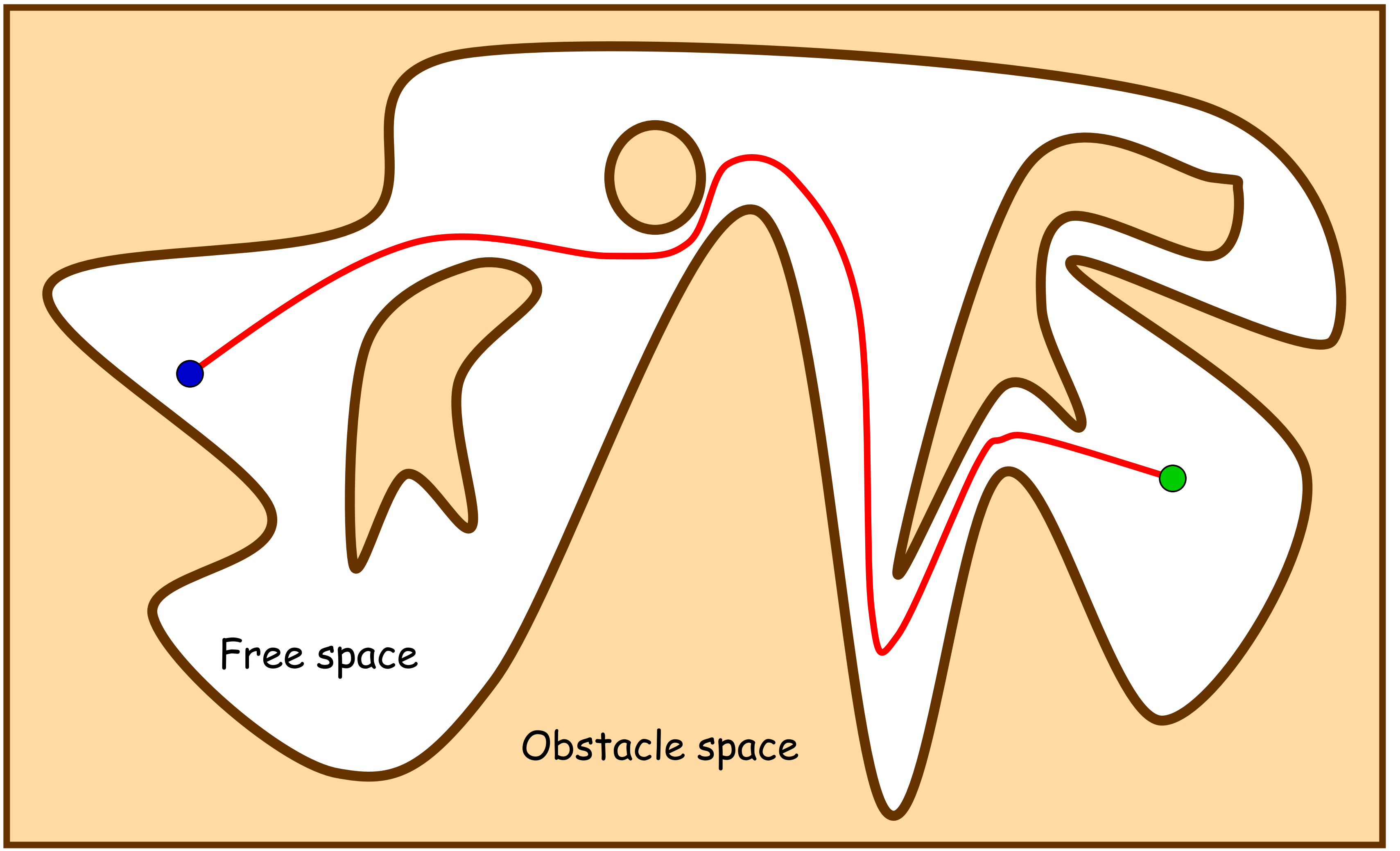
[J.-C. Latombe]

# Robots have different shapes and kinematics! What is a path?

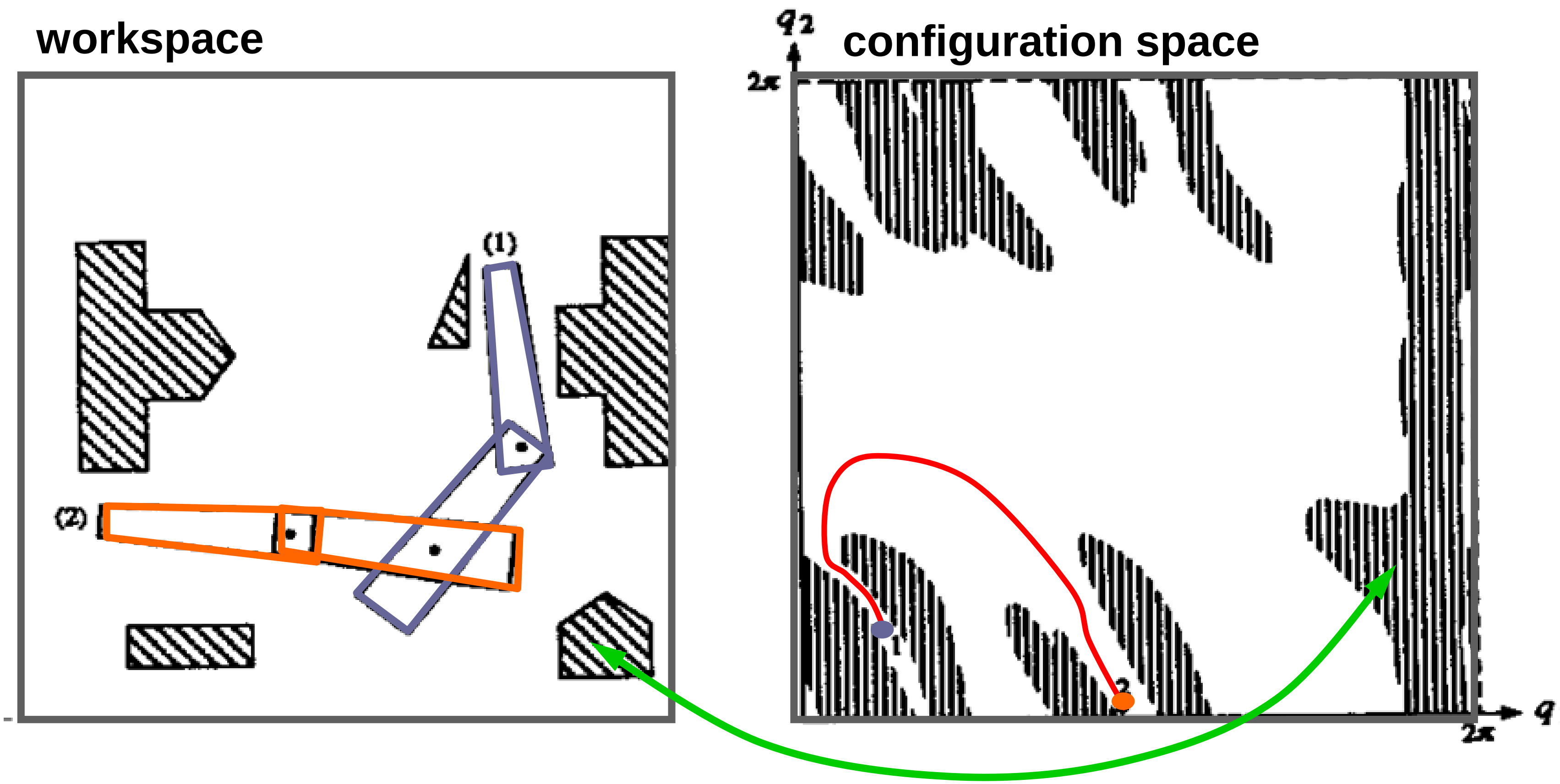




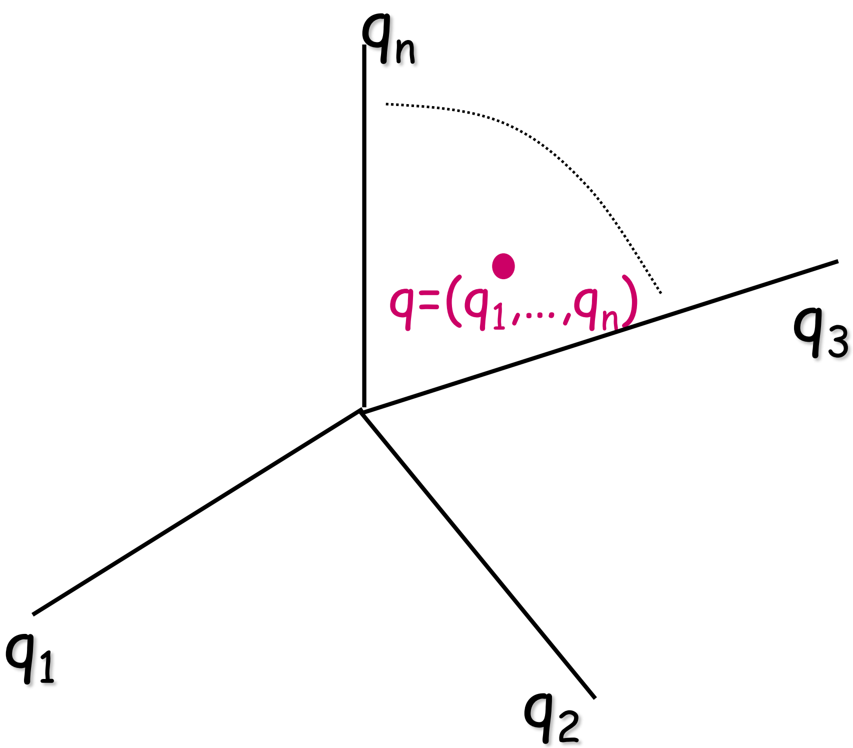
# Idea: Reduce robot to a point → Configuration Space



# Mapping from the workspace to the configuration space

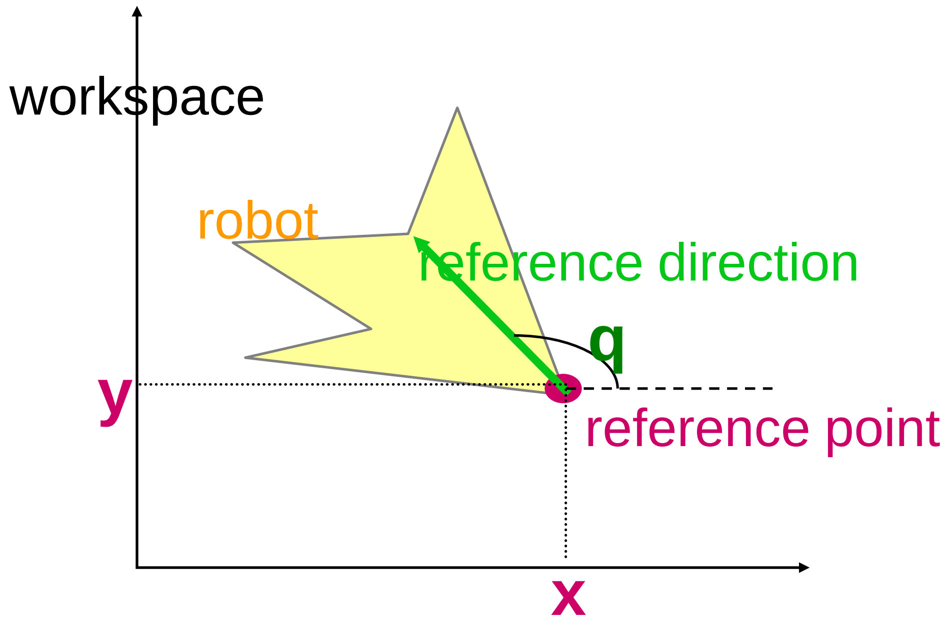


# Configuration Space



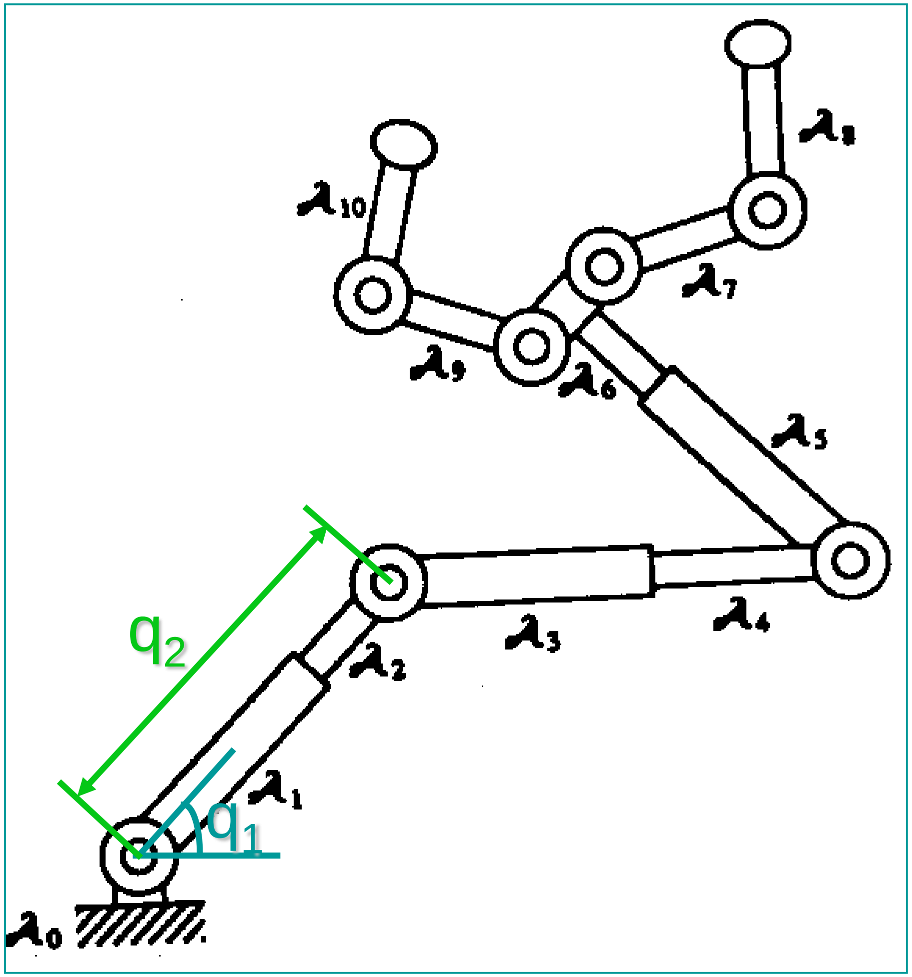

- A robot **configuration** is a specification of the positions of all robot points relative to a fixed coordinate system

# Rigid Robot




- 3-parameter representation: q = (x,y,q)
- In a 3-D workspace q would be of the form (x,y,z,α,β,γ)

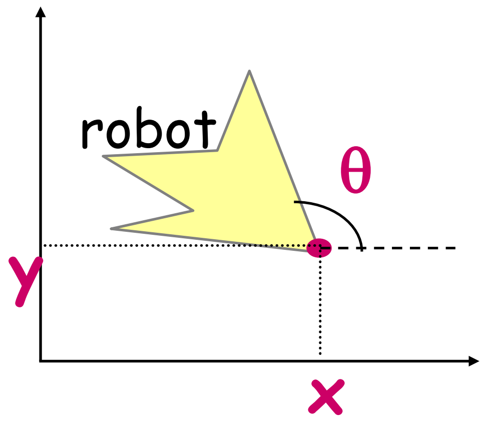
# Articulated Robot

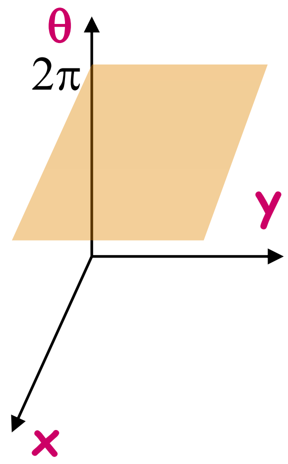


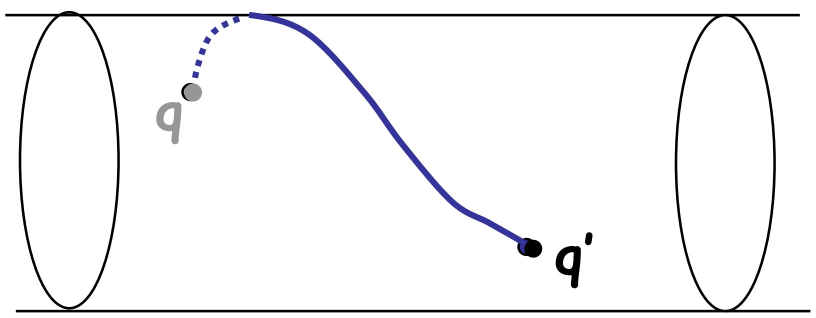

$$q = (q_1, q_2, \ldots, q_{10})$$

# Configuration Space

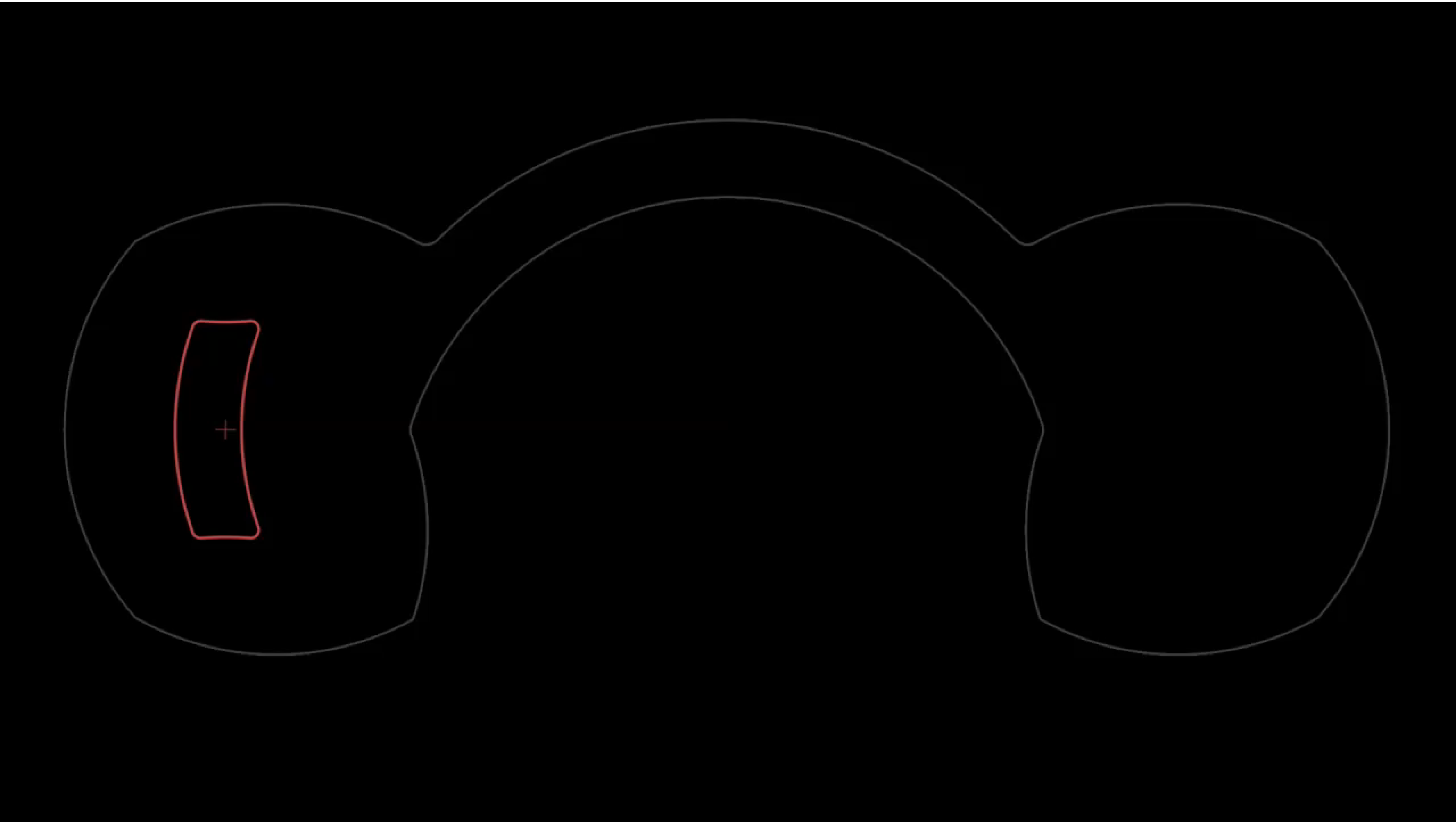
- Space of all its possible configurations
- The topology of configuration space is usually not that of a Cartesian space







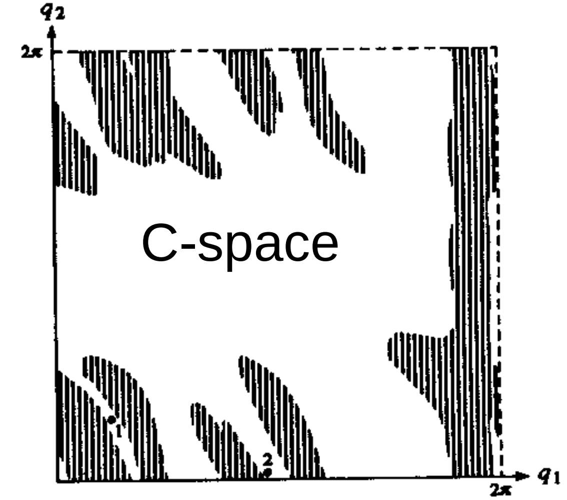
# Configuration Space Visualization

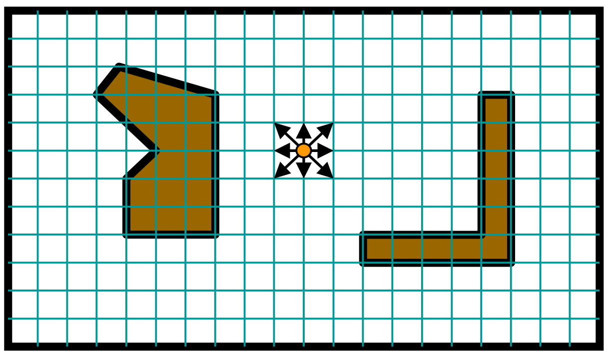


[Stoecker and Milenkovic]

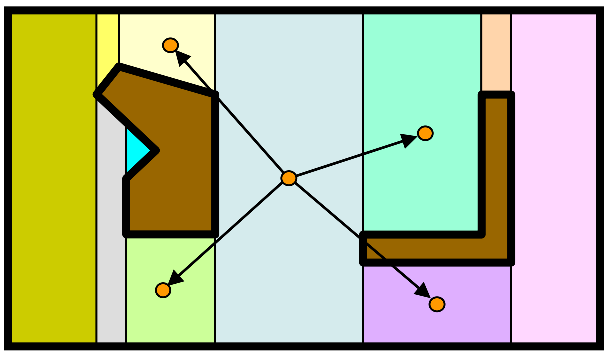
# Continuous space

# Discretization



Sampling-based



Criticality-based

# Search

# SAMPLING-BASED MOTION PLANNING

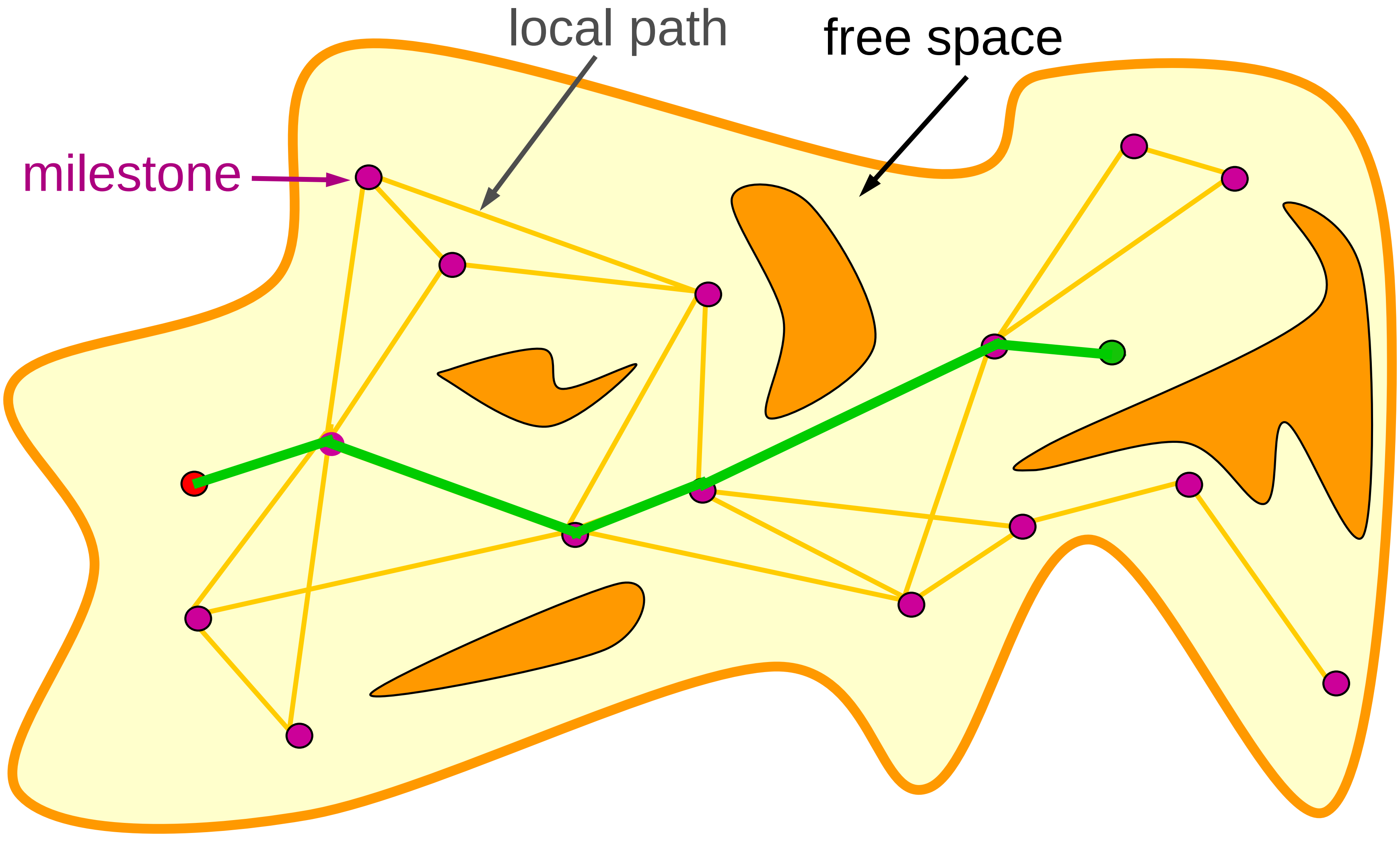
[J.-C. Latombe]

Probabilistic Roadmaps for Path Planning in High-dimensional Configuration Spaces

[Kavraki et al. IEEE TRO 1996]

# PROBABILISTIC ROADMAP (PRM)

# Probabilistic Roadmap (PRM)



# Requirements of PRM Planning

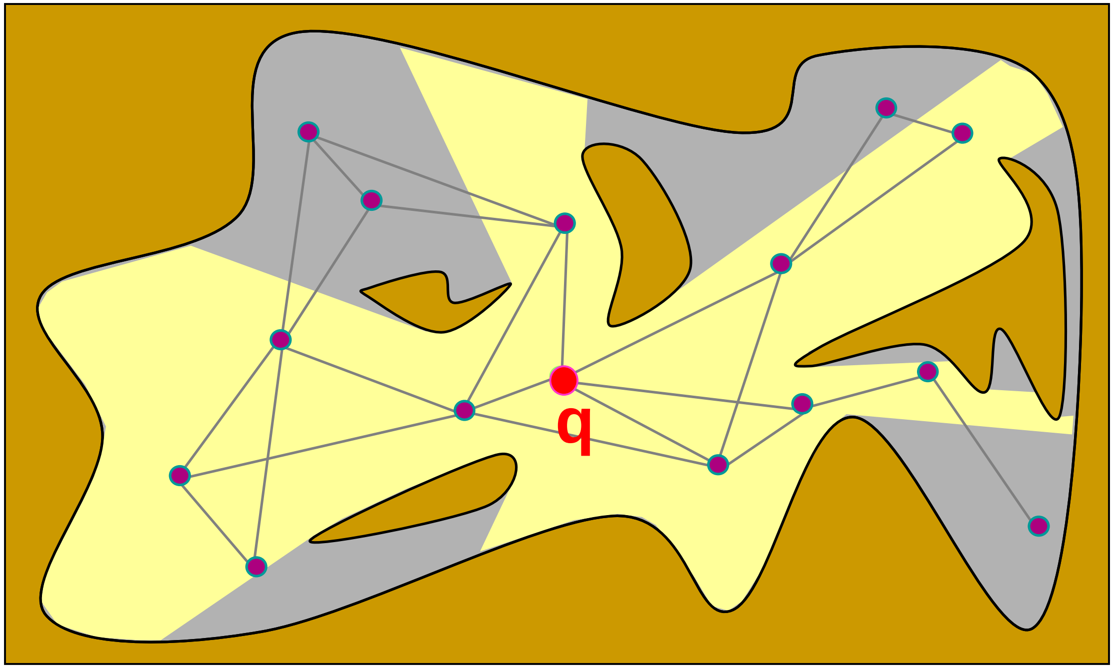
1. Checking sampled configurations and connections between samples for collision can be done efficiently.
   → Collision detection (PQP, FCL, $C^2A$, PolyDepth)

2. A relatively small number of milestones and local paths are sufficient to capture the connectivity of the free space.
   → Non-uniform sampling strategies

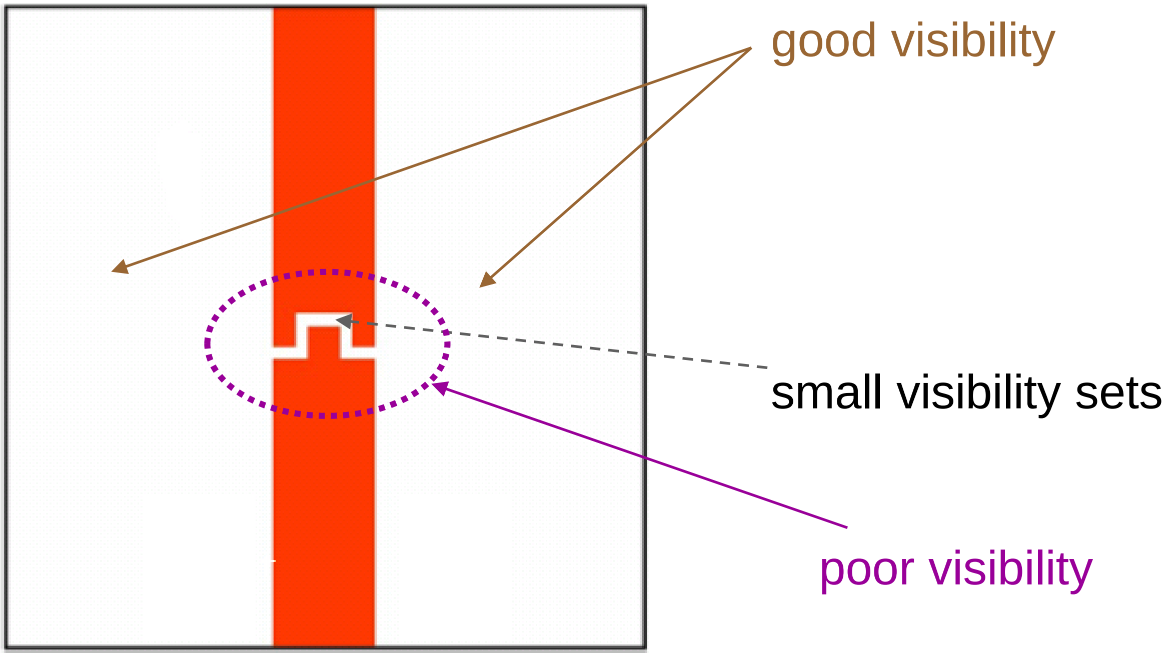
# Visibility in Free Space

- The visibility set of q is a set of all q's such that q and q' see each other

$$\{q' \mid \textbf{path}(q,q') \in \text{Free space}\}$$



- Visibility is usually not uniformly favorable across free space

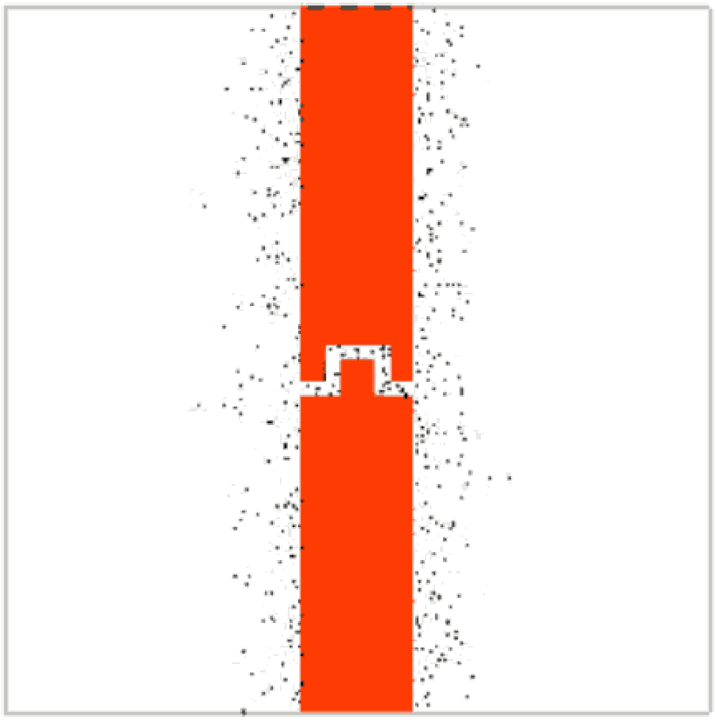



- Regions with poorer visibility should be sampled more densely (more connectivity information can be gained there)

- Workspace-guided strategies
  Identify narrow passages in the workspace and map them into the configuration space
- Filtering strategies
  Sample many configurations, find interesting patterns, and retain only promising configurations
- Adaptive strategies
  Adjust the sampling distribution (p) on the fly
- Deformation strategies
  Deform the free space, e.g., to widen narrow passages

# Gaussian Sampling

[Boor et al. ICRA 99]



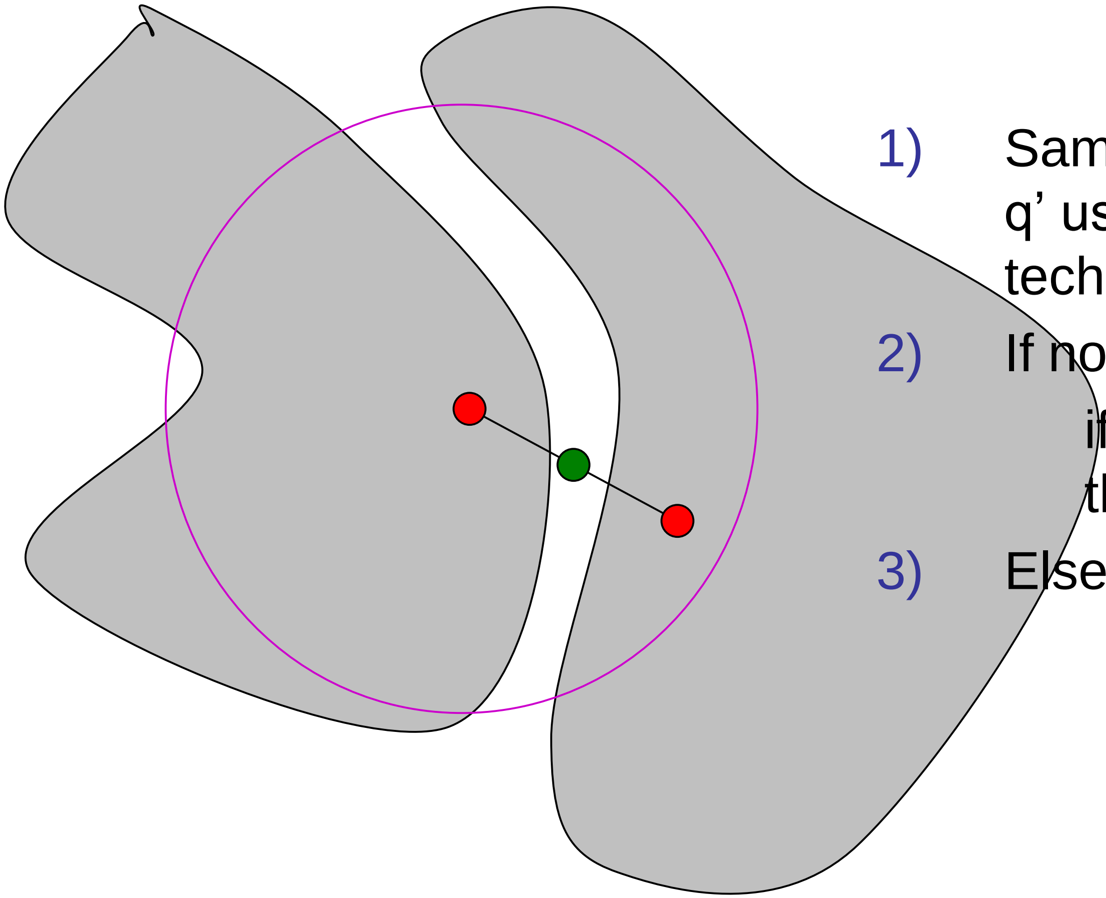
1) Sample a configuration q uniformly at random from configuration space

2) Sample a real number x at random with Gaussian distribution $\mathbf{N}_{[0,s]}(x)$

3) Sample a configuration q' in the ball B(q,|x|) uniformly at random

4) If only one of q and q' is in free space, retain the one in free space as a node; else retain none

# Bridge Test [Sun et al. IEEE TRO 05]

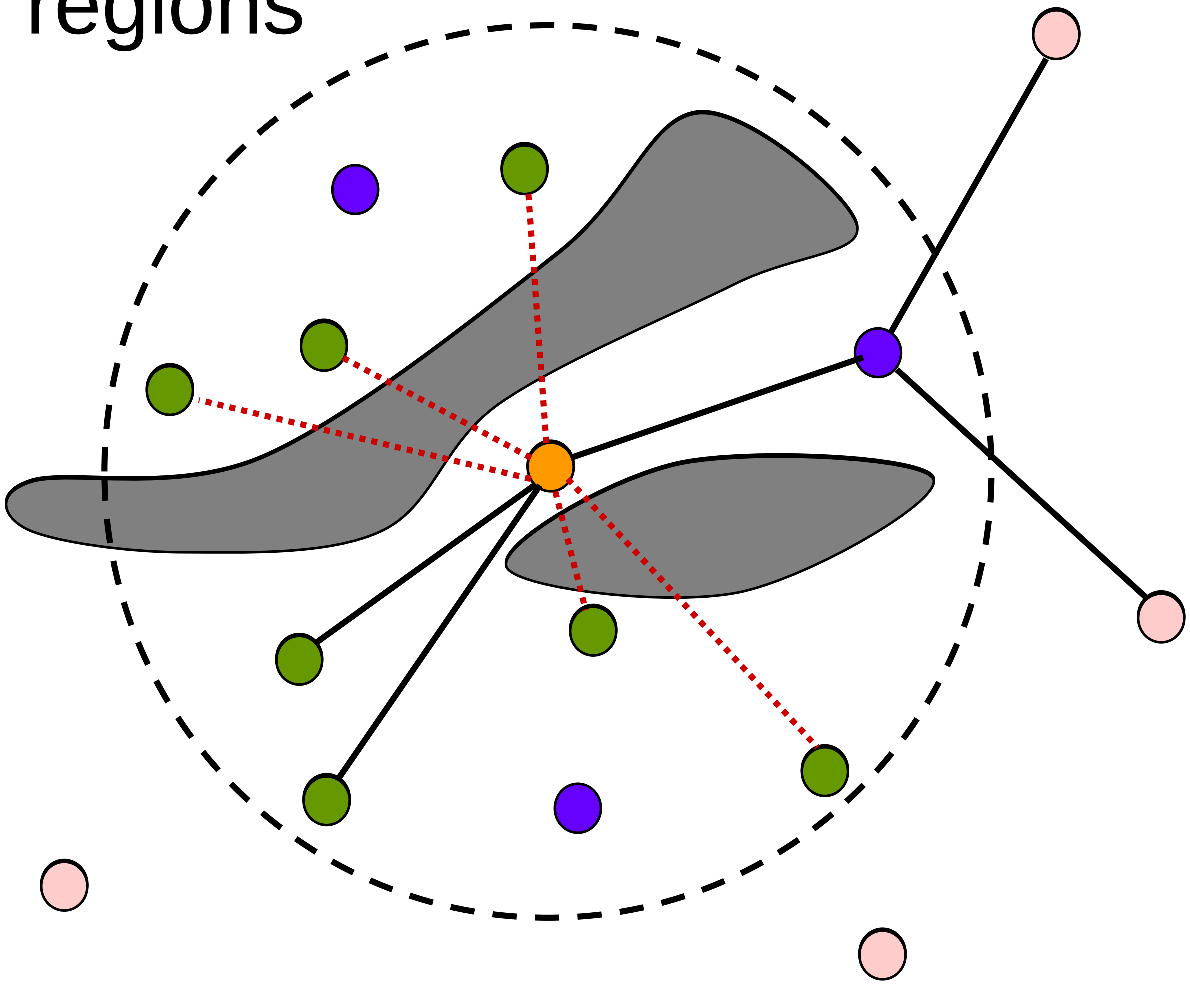


1) Sample two conformations q and q’ using Gaussian sampling technique
2) If none is in free space, then
   if $q_m = (q+q')/2$ is in free space, then retain $q_m$ as a node
3) Else retain none

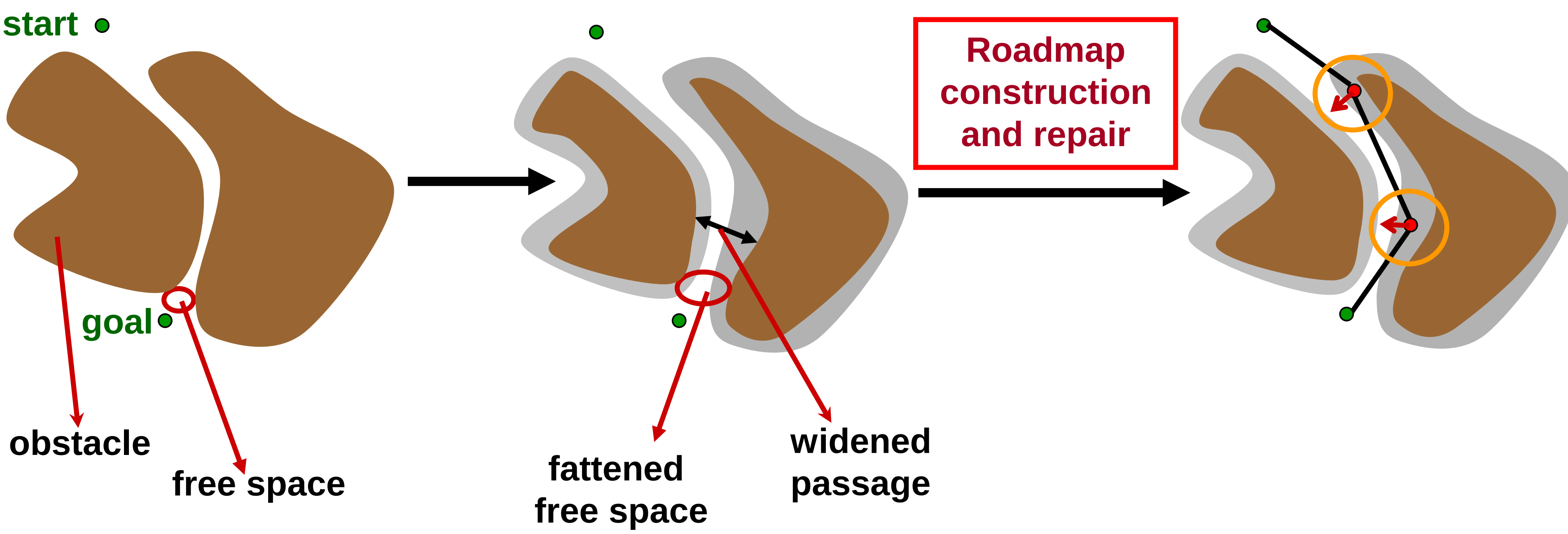
[Kavraki, ICRA 94]

# Connectivity Expansion

Use work already done to detect low-visibility regions



# Free Space Dilatation



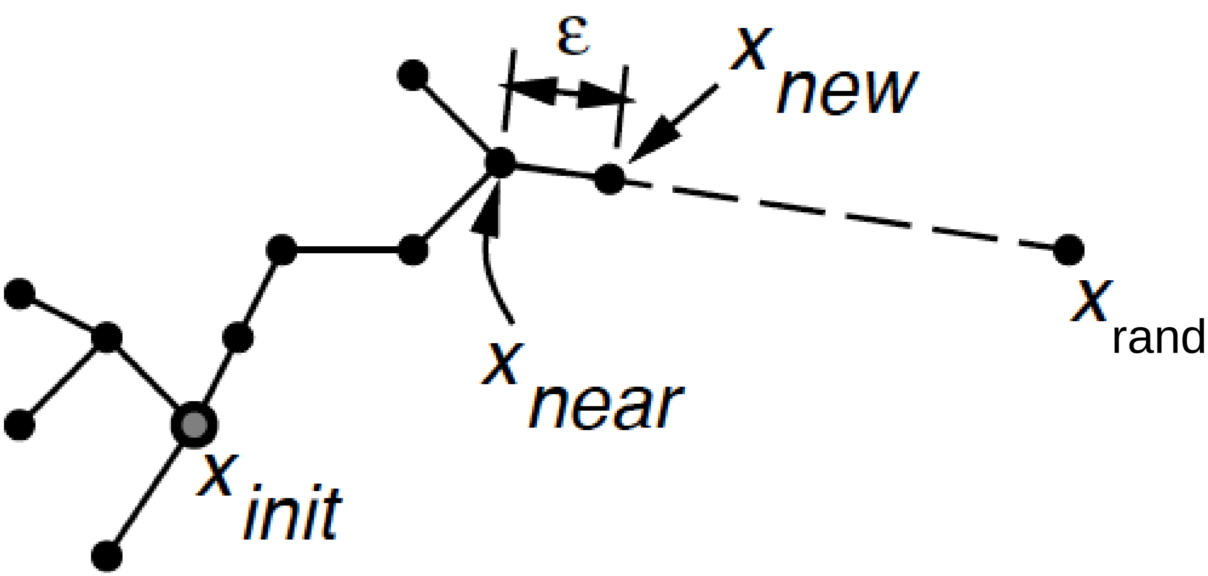
Randomized Kinodynamic Planning [LaValle and Kuffner, ICRA 99]

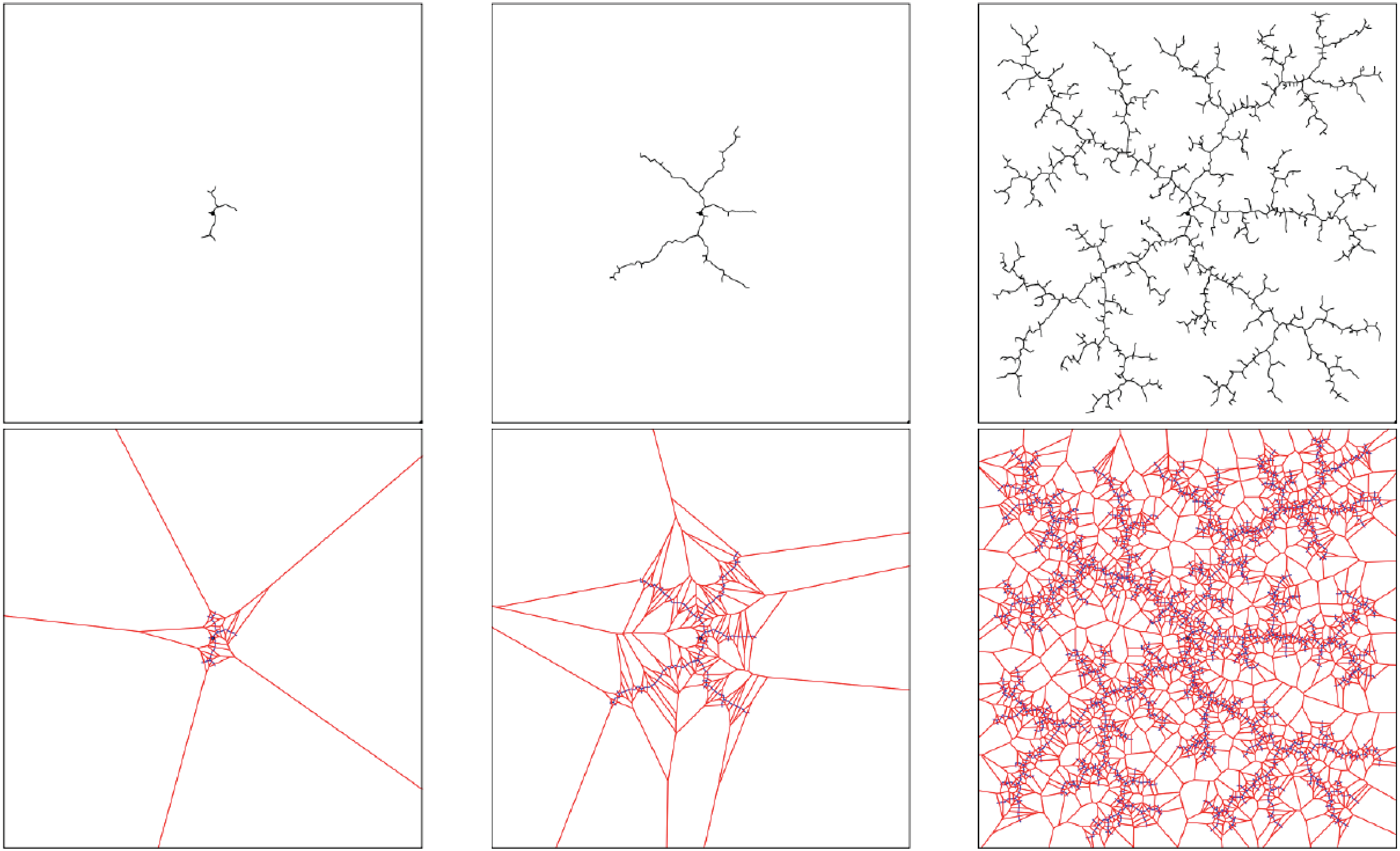
# RAPIDLY EXPLORING RANDOM TREES (RRT)

$x_{rand}$ can be either RANDOM_STATE, or $x_{goal}$ with a small probability (e.g. 0.05)

GENERATE_RRT($x_{init}, K, \Delta t$)

$\mathcal{T}$.init($x_{init}$);<br>
**for** $k = 1$ **to** $K$ **do**<br>
    $x_{rand} \leftarrow$ RANDOM_STATE();<br>
    $x_{near} \leftarrow$ NEAREST_NEIGHBOR($x_{rand}, \mathcal{T}$);<br>
    $u \leftarrow$ SELECT_INPUT($x_{rand}, x_{near}$);<br>
    $x_{new} \leftarrow$ NEW_STATE($x_{near}, u, \Delta t$);<br>
    $\mathcal{T}$.add_vertex($x_{new}$);<br>
    $\mathcal{T}$.add_edge($x_{near}, x_{new}, u$);<br>
Return $\mathcal{T}$

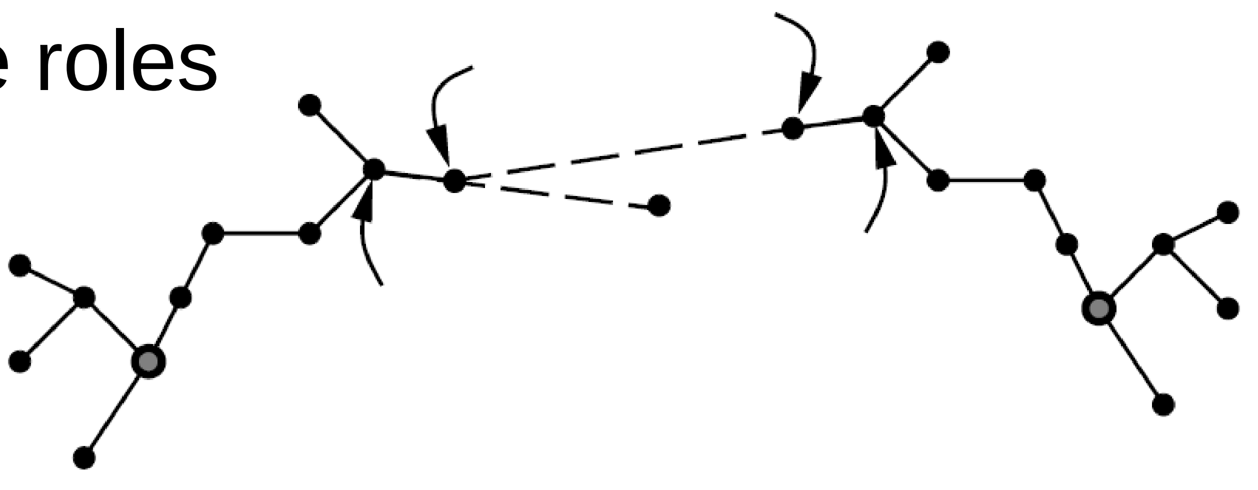




# Bidirectional-RRT

1. One tree is extended, and try to connect the nearest vertex of the other tree to the new vertex

2. Swap the roles

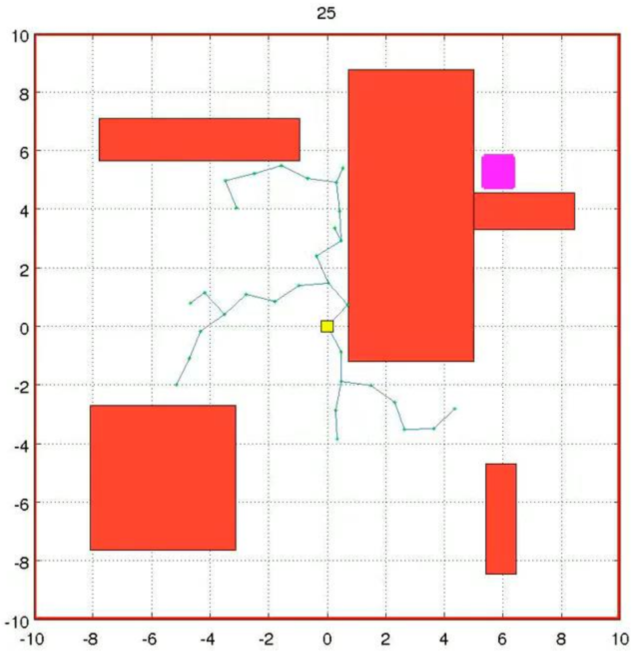


Sampling-based Algorithm for Optimal Motion Planning

[Karaman and Frazzoli IJRR 2014]

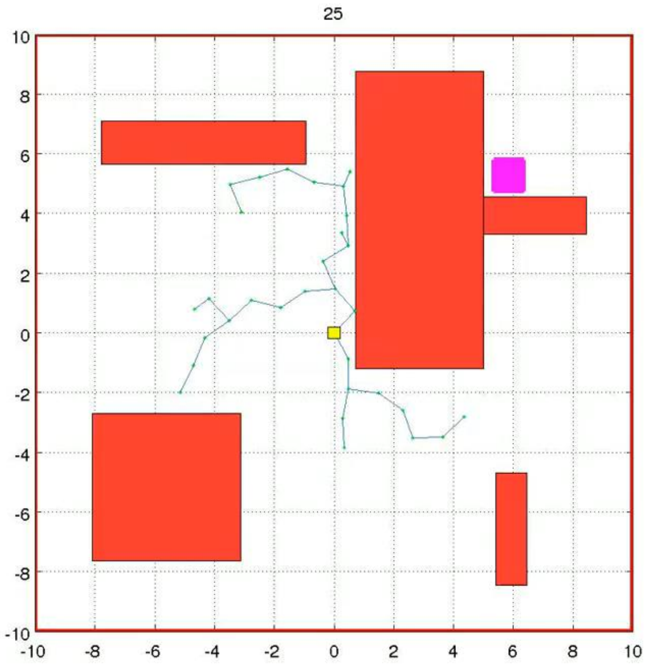
# RRT*

# RRT




No path optimality (quality)

# RRT*




Asymptotic optimality

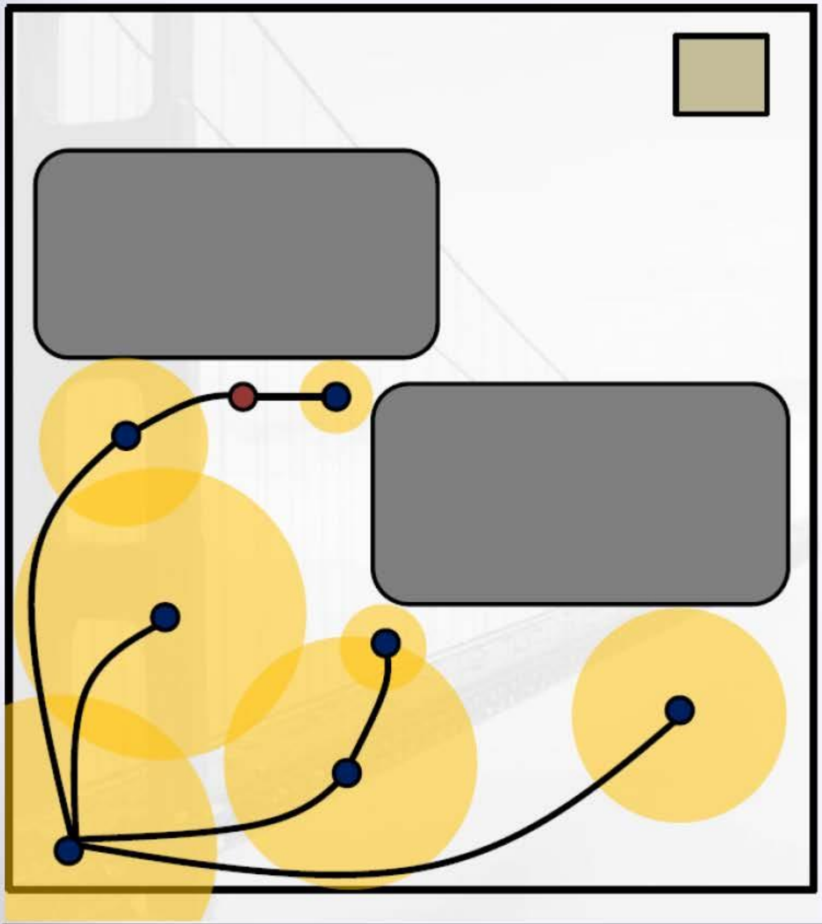
[S. Karaman]

# RRT vs RRT*

- RRT
  - Connect to the nearest node

- RRT*
  - Identify neighbor nodes within some $\varepsilon$ −ball, and re-wire them to optimize the path

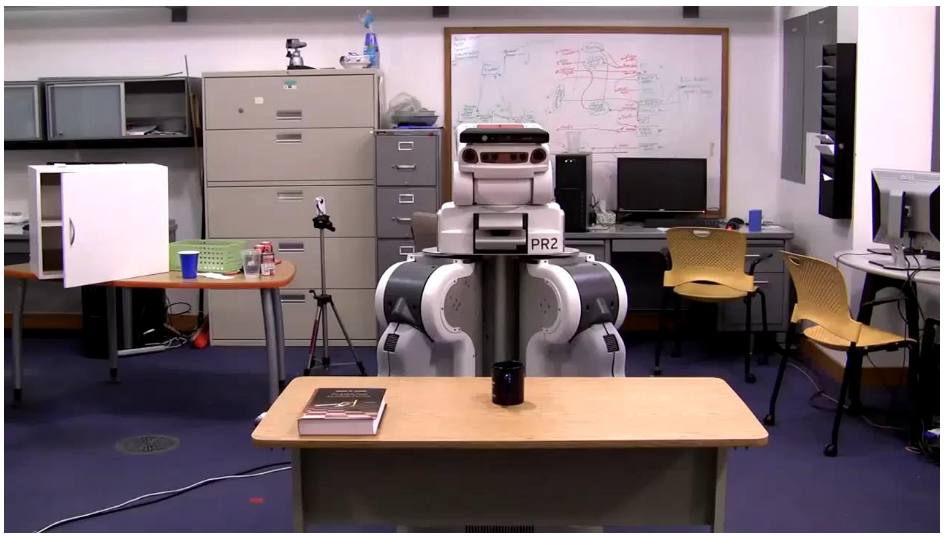
# Example: Re-Wiring Operation



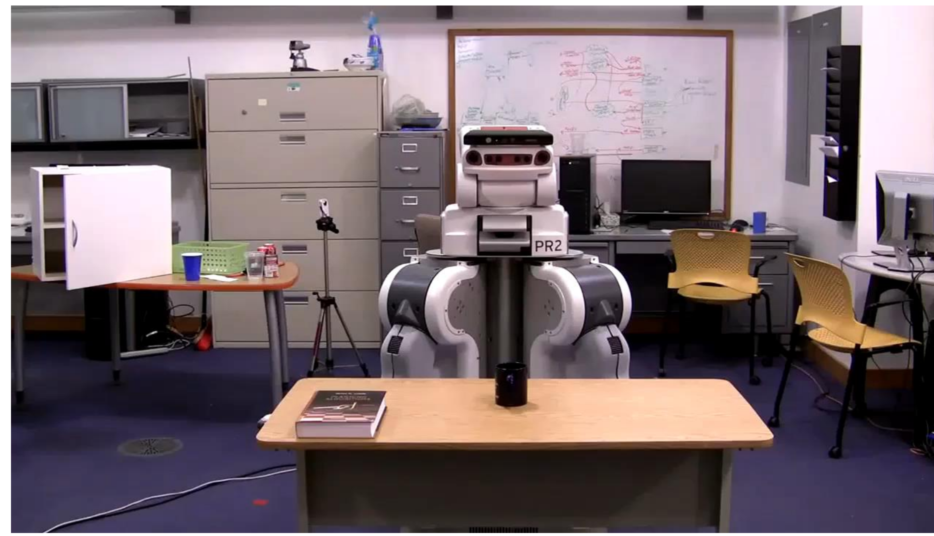
Video showing benefits with real robot

From ball tree paper

# RRT




# RRT*




[S. Karaman]

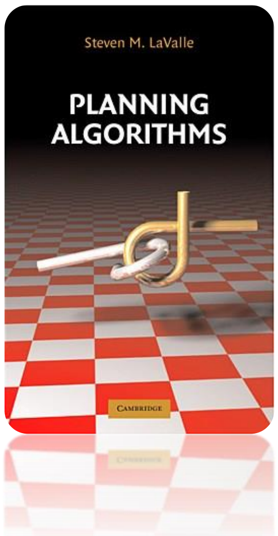
# Summary

- Basic concepts

- Classification

- Sampling-based algorithms

# Recommended Readings





Planning Algorithms

Steven LaValle

Cambridge Press, 2006

http://planning.cs.uiuc.edu/

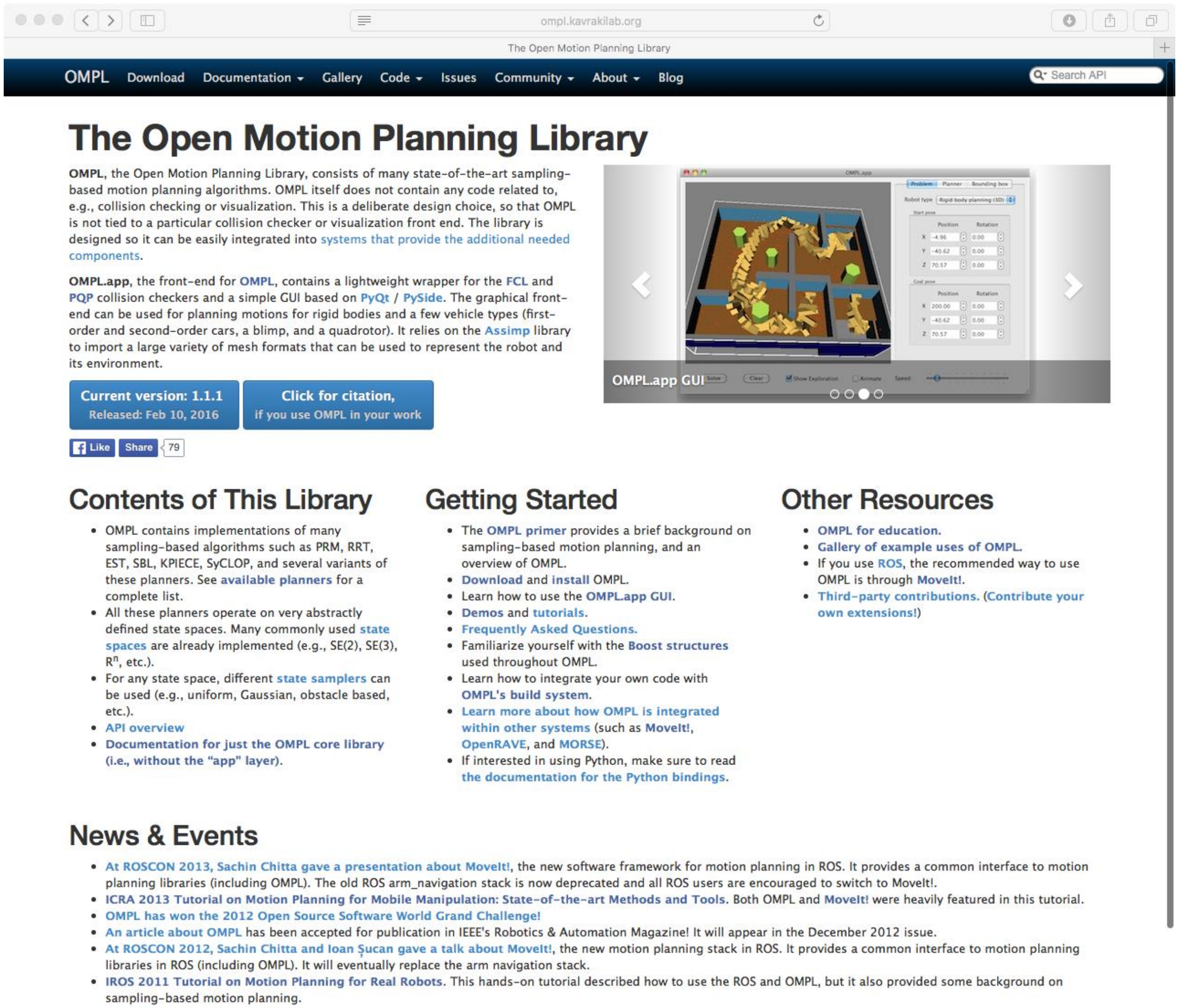
Robot Motion Planning

Jean Claude Latombe

Springer, 1990

http://books.google.com/books?id=Mbo_p4-46-cC

# Open Software - OMPL



ompl.kavrakilab.org

The Open Motion Planning Library

OMPL Download Documentation Gallery Code Issues Community About Blog

## The Open Motion Planning Library

**OMPL**, the Open Motion Planning Library, consists of many state-of-the-art sampling-based motion planning algorithms. OMPL itself does not contain any code related to, e.g., collision checking or visualization. This is a deliberate design choice, so that OMPL is not tied to a particular collision checker or visualization front end. The library is designed so it can be easily integrated into systems that provide the additional needed components.

**OMPL.app**, the front-end for OMPL, contains a lightweight wrapper for the FCL and PQP collision checkers and a simple GUI based on PyQt / PySide. The graphical front-end can be used for planning motions for rigid bodies and a few vehicle types (first-order and second-order cars, a blimp, and a quadrotor). It relies on the Assimp library to import a large variety of mesh formats that can be used to represent the robot and its environment.

OMPL.app GUI

Current version: 1.1.1
Released: Feb 10, 2016

Click for citation,
if you use OMPL in your work

Like Share 79

### Contents of This Library

- OMPL contains implementations of many sampling-based algorithms such as PRM, RRT, EST, SBL, KPIECE, SyCLOP, and several variants of these planners. See available planners for a complete list.
- All these planners operate on very abstractly defined state spaces. Many commonly used state spaces are already implemented (e.g., SE(2), SE(3), $R^n$, etc.).
- For any state space, different state samplers can be used (e.g., uniform, Gaussian, obstacle based, etc.).
- API overview
- Documentation for just the OMPL core library (i.e., without the "app" layer).

### Getting Started

- The OMPL primer provides a brief background on sampling-based motion planning, and an overview of OMPL.
- Download and install OMPL.
- Learn how to use the OMPL.app GUI.
- Demos and tutorials.
- Frequently Asked Questions.
- Familiarize yourself with the Boost structures used throughout OMPL.
- Learn how to integrate your own code with OMPL's build system.
- Learn more about how OMPL is integrated within other systems (such as MoveIt!, OpenRAVE, and MORSE).
- If interested in using Python, make sure to read the documentation for the Python bindings.

### Other Resources

- OMPL for education.
- Gallery of example uses of OMPL.
- If you use ROS, the recommended way to use OMPL is through MoveIt!.
- Third-party contributions. (Contribute your own extensions!)

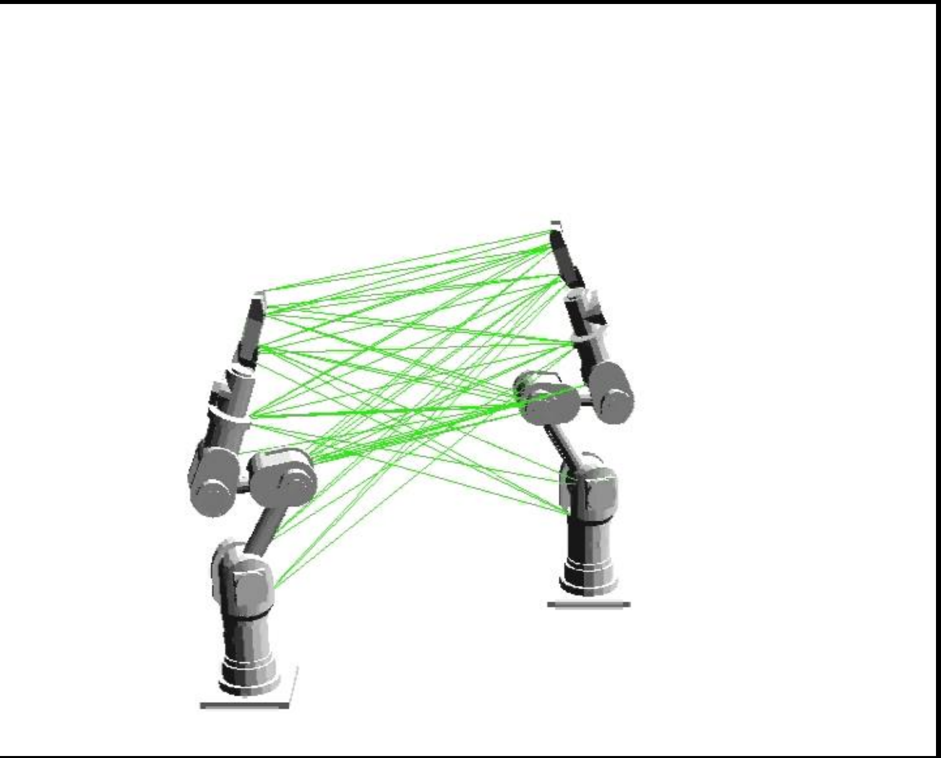
### News & Events

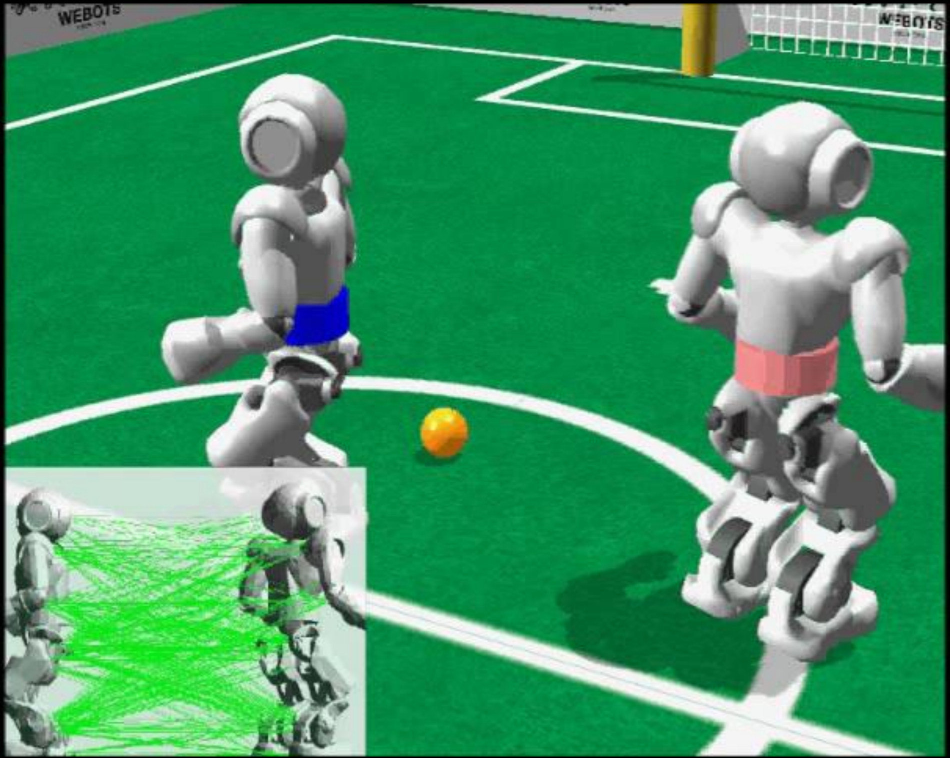
- At ROSCON 2013, Sachin Chitta gave a presentation about MoveIt!, the new software framework for motion planning in ROS. It provides a common interface to motion planning libraries (including OMPL). The old ROS arm_navigation stack is now deprecated and all ROS users are encouraged to switch to MoveIt!.
- ICRA 2013 Tutorial on Motion Planning for Mobile Manipulation: State-of-the-art Methods and Tools. Both OMPL and MoveIt! were heavily featured in this tutorial.
- OMPL has won the 2012 Open Source Software World Grand Challenge!
- An article about OMPL has been accepted for publication in IEEE's Robotics & Automation Magazine! It will appear in the December 2012 issue.
- At ROSCON 2012, Sachin Chitta and Ioan Şucan gave a talk about MoveIt!, the new motion planning stack in ROS. It provides a common interface to motion planning libraries in ROS (including OMPL). It will eventually replace the arm navigation stack.
- IROS 2011 Tutorial on Motion Planning for Real Robots. This hands-on tutorial described how to use the ROS and OMPL, but it also provided some background on sampling-based motion planning.

# ROBOT MOTION PLANNING RESEARCH AT EWHA

# Distance Tracking and Collision Detection

(shown in green lines)



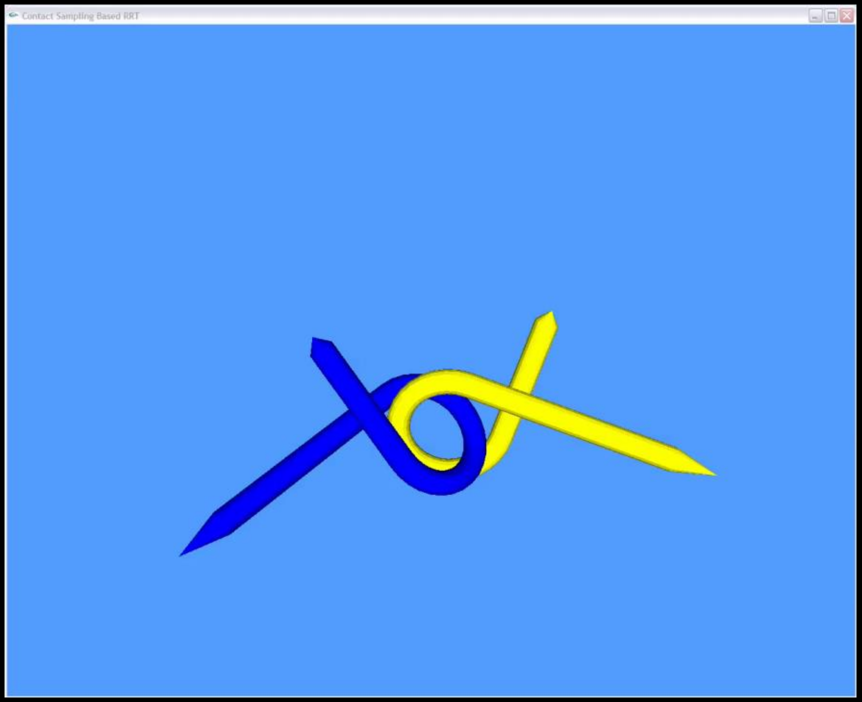



Zhang and Kim, k-IOS: intersections of spheres, IEEE ICRA 12

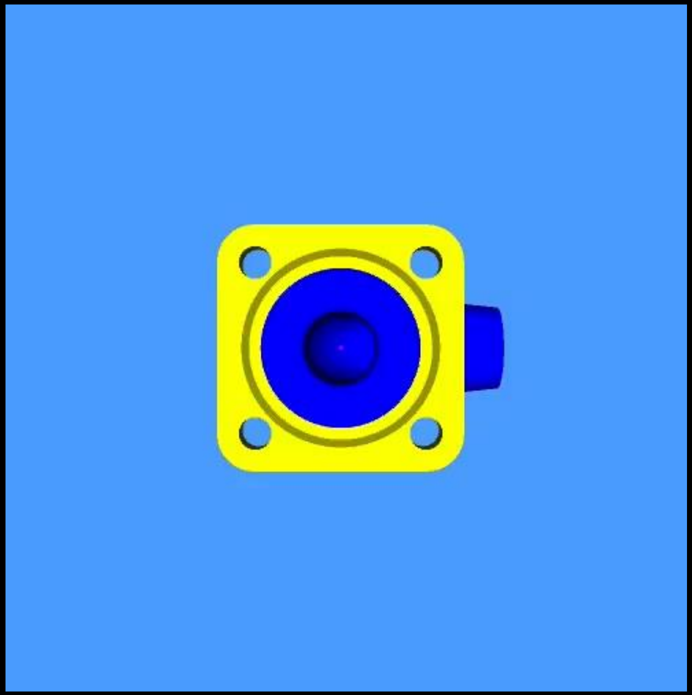
# 6 DoF Haptic Rendering


Yi et al., Six-degree-of-freedom haptic rendering using translational and generalized penetration Depth Computation, IEEE WHC 13

# Retraction-based Motion Planning



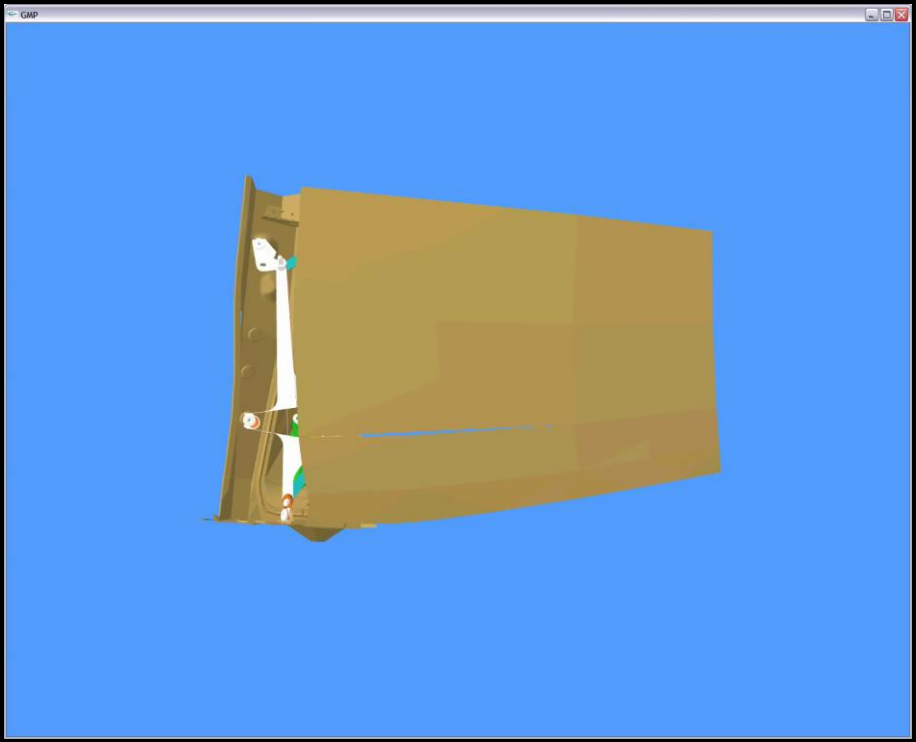

Alpha Puzzle



Flange

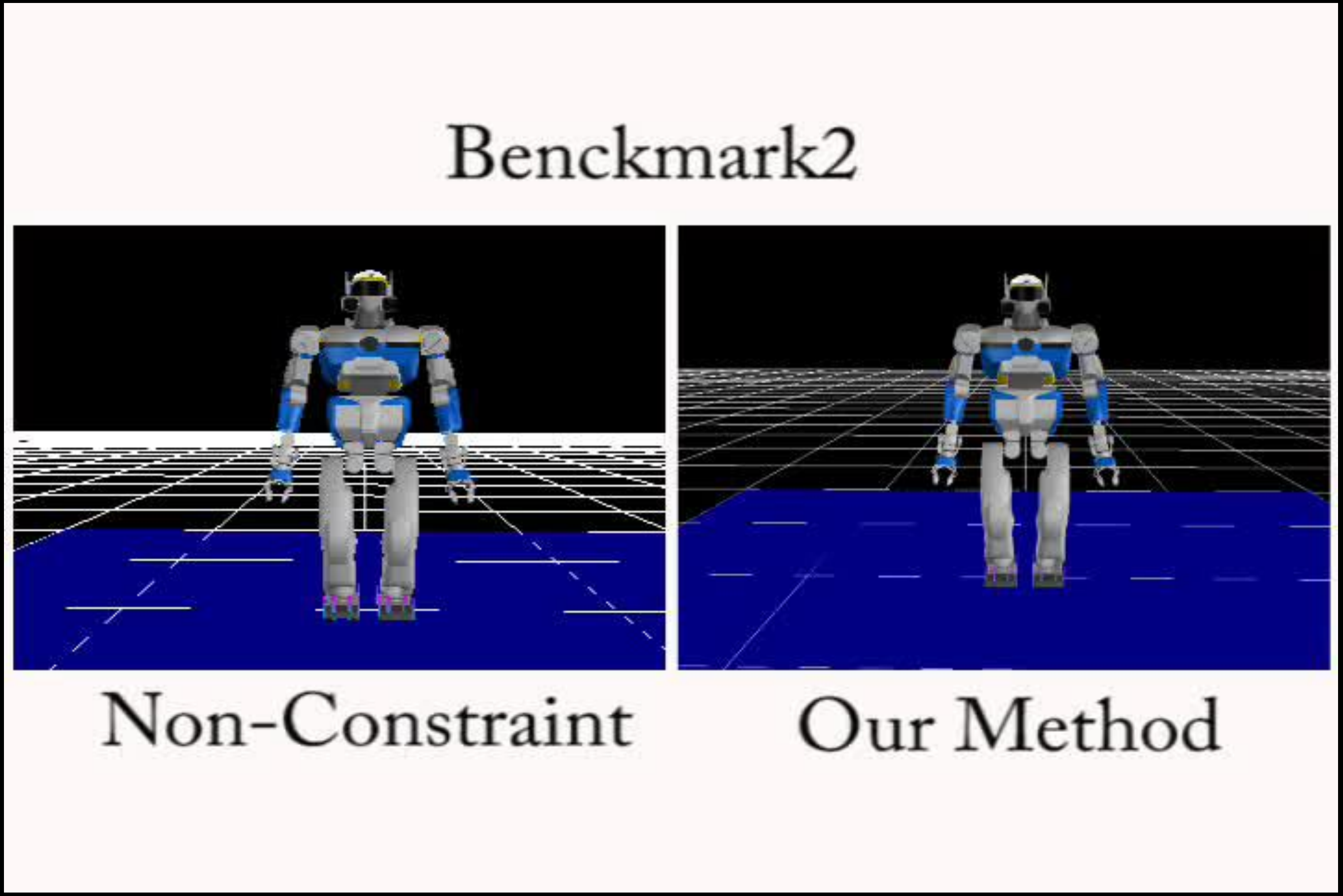
Zhang et al., D-plan: efficient collision-free path computation for part removal and disassembly, CAD&A 08


# Car seat Removal
# (240K triangles)



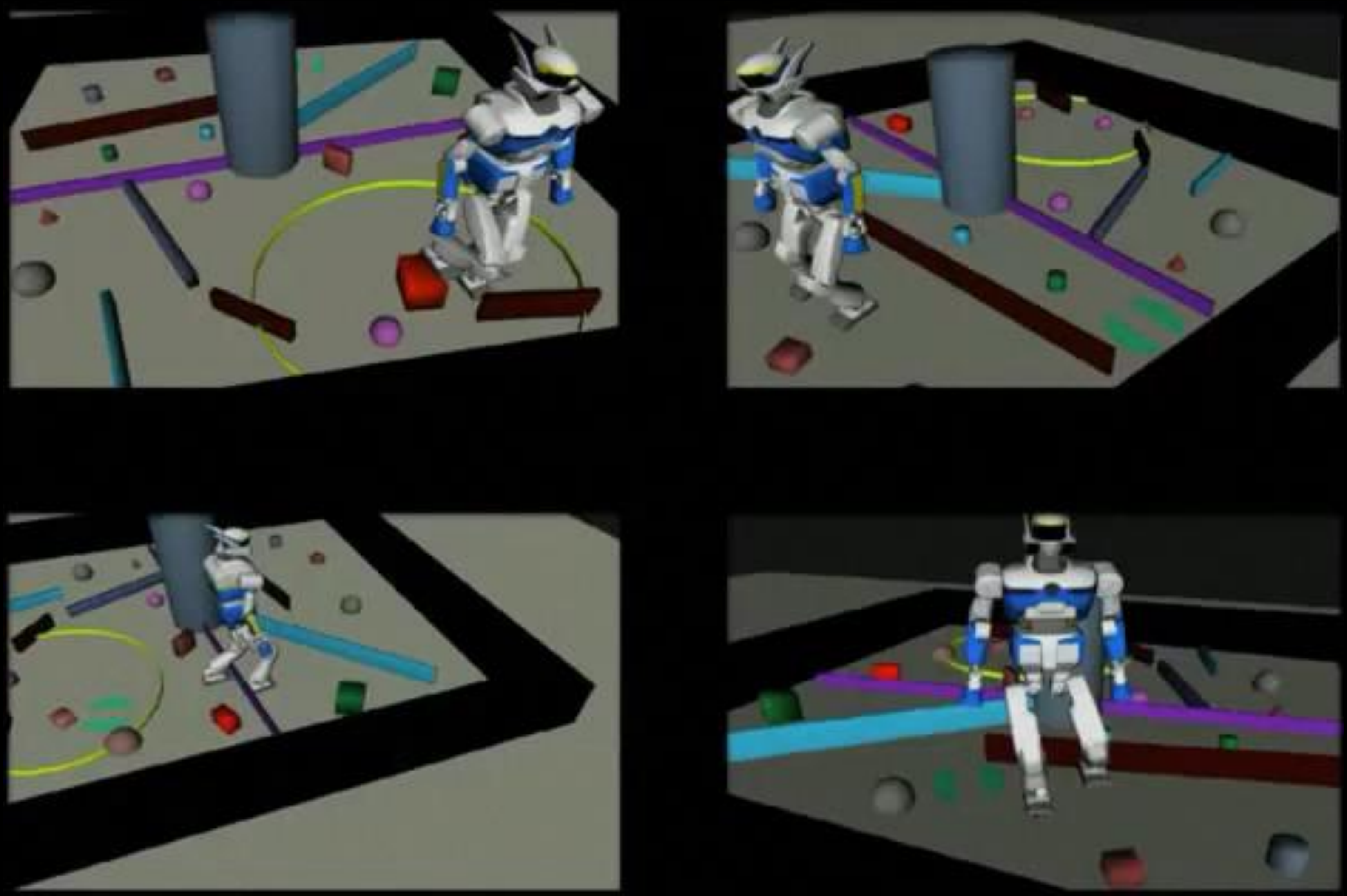

# Wiper Removal
# (27K triangles)

# Optimization-based Motion Planning



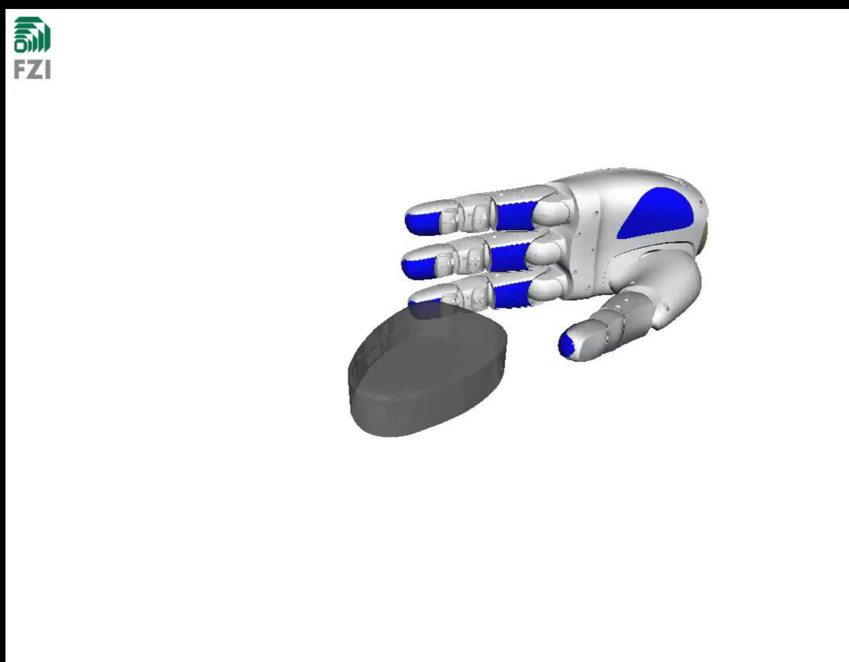

Lee et al., Accurate evaluation of a distance function for optimization based motion planning, IEEE IROS 2012

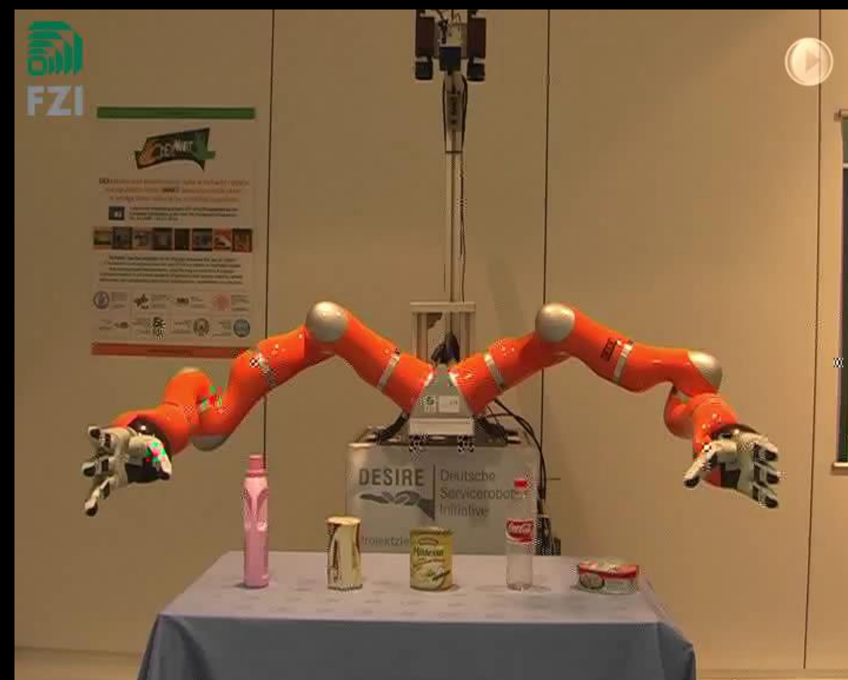
# Real-time Footstep Planning



Perrin et al., Real-time footstep planning for humanoid robots among 3d obstacles using a hybrid bounding box, IEEE ICRA 2012

# Grasp Planning



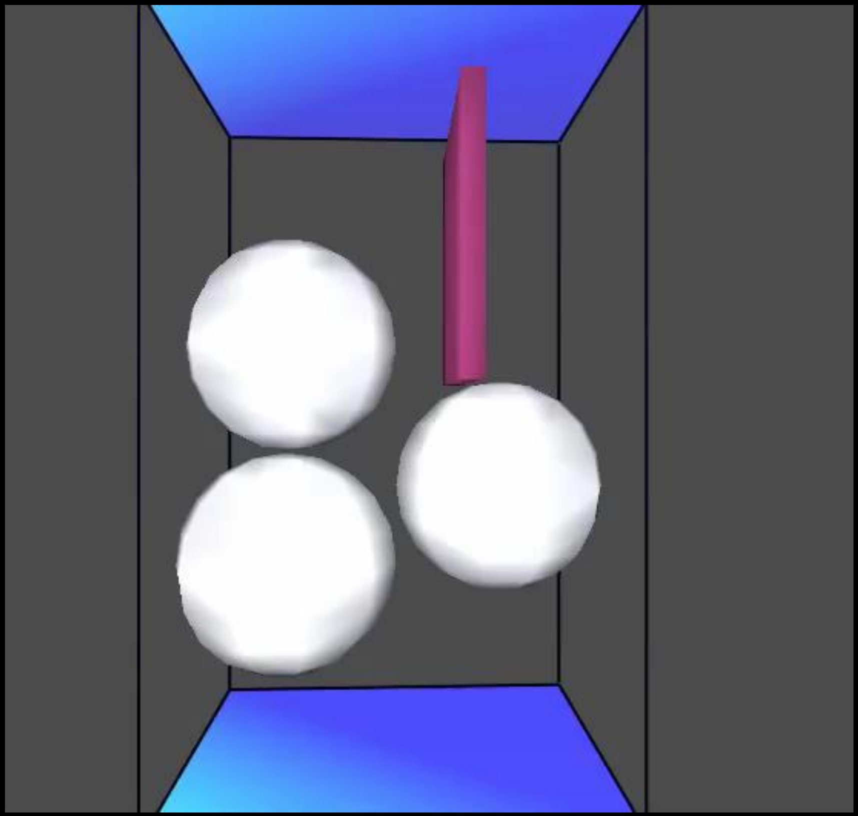




Courtesy of Zhixing Xue, FZI

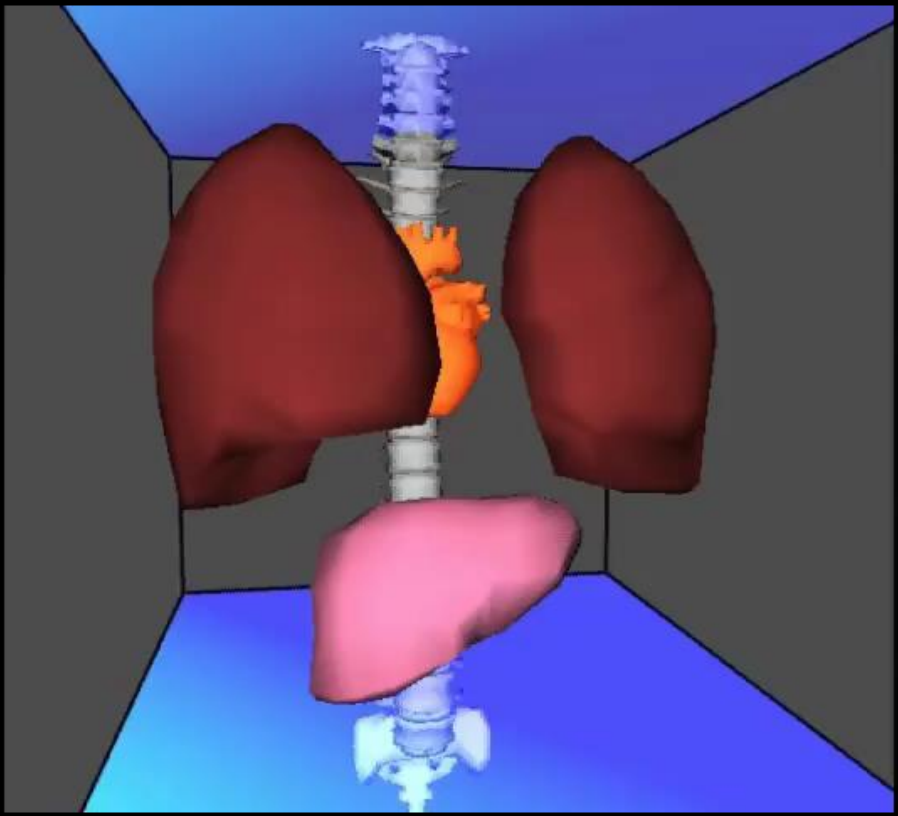

FZI
Grasp Planning
Zhixing Xue, FZI
3x
MANGO

# KUKA IIWA R800

# Deformable Planning



Bar/Sphere
(636 triangles)



Human Organs
(14K triangles)

Tang et al., Continuous collision detection for non-rigid contact computation using local advancement, IEEE ICRA 2012

# Thank you!

http://graphics.ewha.ac.kr/