Introduction

Jane Li

Assistant Professor

Mechanical Engineering & Robotics Engineering

http://users.wpi.edu/~zli11

Practical Issues

- Office hours are available by appointment
- See course website for most current information:

  https://sites.google.com/site/muyimolin/teaching/rbe550

  (Link to this site is on my homepage)
- Course capacity = 35
- For those not yet enrolled, keep coming to class to see if spots open up

What you can expect to get from this course

- An understanding of modern motion planning methods
- Programming experience
- Hands-on experience working with motion planning
- *** Paper-reading, presentation, and research skills ***

Prerequisites

- Undergraduate Linear Algebra
- Experience with 3D geometry
- Significant programming experience
  - We will use python and C++ in this course
  - You don’t need to be an expert but I expect you to pick it up

Python Example

```
    #get the second TSR's offset from the handle to the hand
    Tw1_e = linalg.inv(T0_w0*Tw0_e)*T0_RH1

    #define bounds to allow rotation of the hand about z axis of the handle from -pi
    Bw1 = mat([0, 0,   0, 0,   0, 0,   0, 0,   0, 0,   -pi/2, 0])

    #serialize the TSRs
    TSRstring1 = SerializeTSR(0,'NULL',T0_w0,Tw0_e,Bw0)
    TSRstring2 = SerializeTSR(0,'NULL',mat(eye(4)),Tw1_e,Bw1) #note the T0_w value h
    TSRChainString = SerializeTSRChain(0,0,1,2,TSRstring1 + ' ' + TSRstring2,'kitche

    #set the desired amount to open the door
    door_rot = pi/1.5
    #set the desired amount to rotate the hand about the handle at the goal
    handle_rot = -pi/2.1

#keep track of how many DOF of the mimic body are being mimiced
numTSRChainMimicDOF = 1;

#get goal transform
Tdoor_rot = MakeTransform(rodrigues([0, 0, door_rot]),mat([0, 0, 0]).T)
Thandle_rot = MakeTransform(rodrigues([0, 0, handle_rot]),mat([0, 0, 0]).T)
T0_doornew = T0_w0*Tdoor_rot;
T0_RH2 = T0_doornew*Tw0_e*Thandle_rot*linalg.inv(T0_doornew*Tw0_e)* T0_doornew*linal

goalik = str2num(probs_cbirrt.SendCommand('DoGeneralIK exec nummanips 1 maniptm 0 %s
robot.SetActiveDOFValues(goalik)

#set robot to starting configuration
```

C++ example

```
    bool bGotSolution = false;
    for(int i = 0; i < qGuesses.size(); i++)
    {
        _planner->ClearTSRChainValues();
        // if a guess is valid, this will confirm it and return quickly, if not, it wil
        if(ConstrainedNewConfig(qGuesses[i], q_near, _planner->vTSRChainValues_temp,fal
        {

            RrtNode * pikNode = new RrtNode(_planner->GetNumDOF(),ptree->GetFromGoal(),
            pikNode->SetConfig(&qGuesses[i][0]);
            pikNode->SetTSRChainValues(_planner->vTSRChainValues_temp);

            ptree->AddNode(*pikNode); //this will automatically increment the size
            pikNode->Print();
            delete pikNode;
            //return true;
            bGotSolution = true;
        }
    }
    if(bGotSolution)
        RAVELOG_INFO("Constrained IK solution(s) found!\n");
    else
        RAVELOG_INFO("No Constrained IK solution found\n");

    return bGotSolution;
}

bool CBirrtPlanner::_CheckCollision(std::vector<dReal>& pConfig)
{
    collisioncheckcalls++;
#ifdef RECORD_TIMES
    double starttime = timeGetThreadTime();
```

Math expectations

- Linear algebra proficiency is a **MUST**
  - Matrix operations
  - Dot products, cross products, etc.
  - A review of linear algebra:
    - http://cs229.stanford.edu/section/cs229-linalg.pdf
- Be able to read math notation
  - Some examples from the textbook:

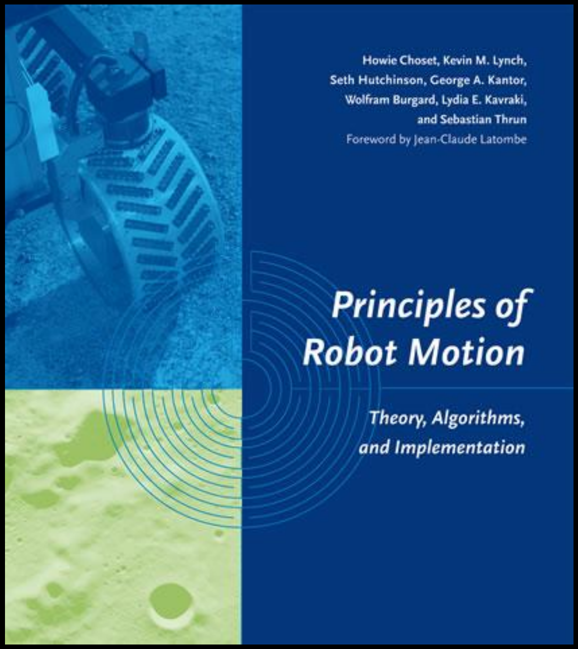
$$X_{obs} = \left(\bigcup_{i=1}^{m} X^{i}_{obs}\right) \cup \left(\bigcup_{ij,\ i\neq j} X^{ij}_{obs}\right)$$

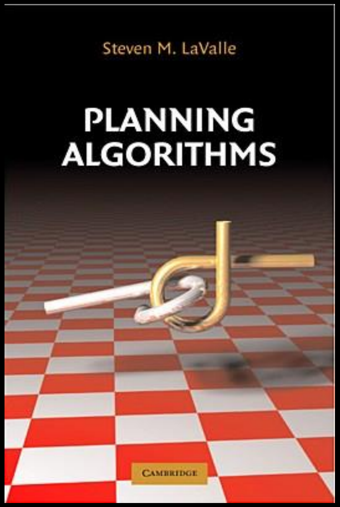
$$X^{ij}_{obs} = \{(s_1, \ldots, s_m) \in X \mid \mathcal{A}^i(\tau_i(s_i)) \cap \mathcal{A}^j(\tau_j(s_j)) \neq \emptyset\}$$

$$\mathcal{C}^{p}_{obs} = \{(q^a, q^p) \in \mathcal{C} \mid \operatorname{int}(\mathcal{P}(q^p)) \cap \mathcal{O} \neq \emptyset\}$$

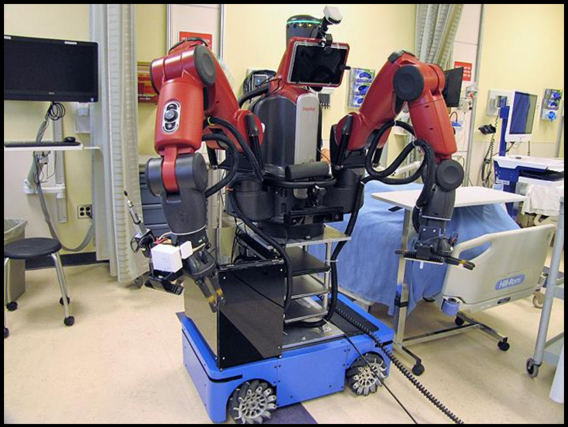
Readings

- Textbook
  - Principles of Robot Motion (by Howie Choset)
  - Referred as “**Principles**” in this course
  - Available online via library
- Week 1-7 ~ Textbook
- Week 8-14 ~ Research papers
- Students will present and discuss research papers
- Other reference:
  - Planning algorithms by S. M. LaValle
  - Available online





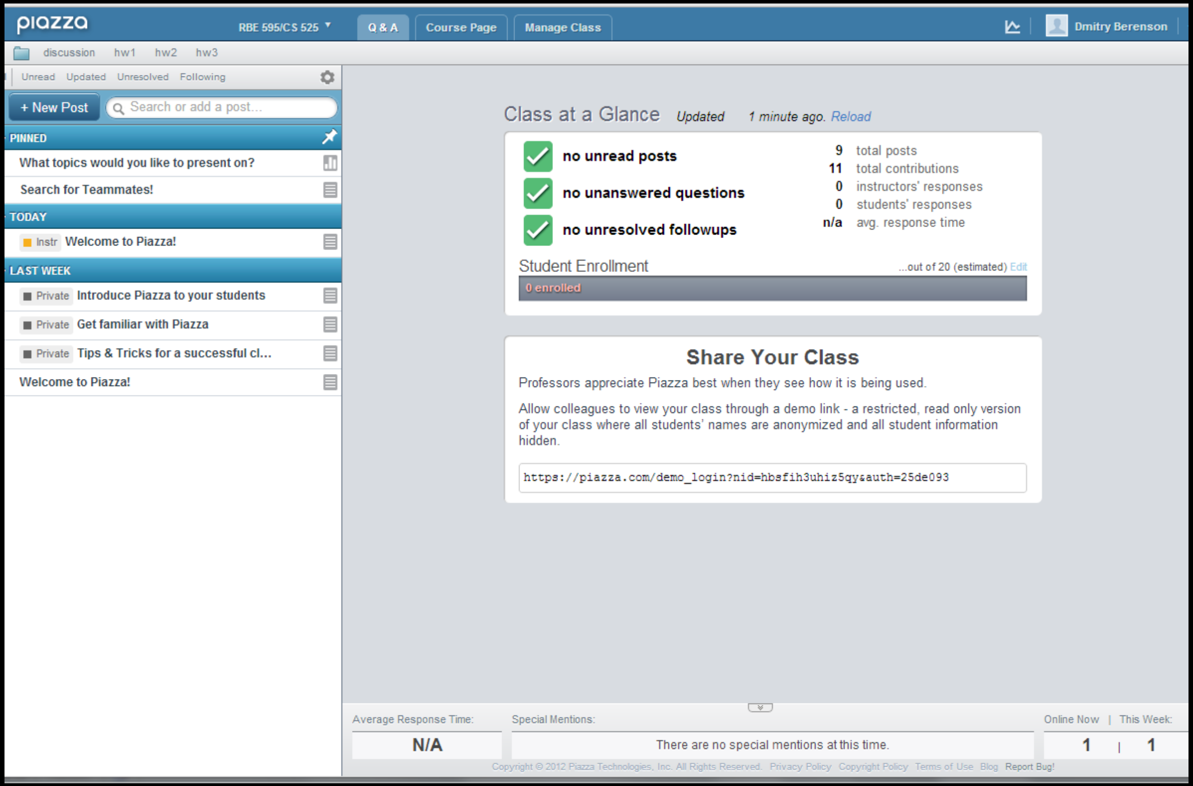
Syllabus



- No exams!
- Three homeworks
  - Individual (no groups)
  - Learn to implement algorithms
  - Use openrave open-source motion planning framework
- Final Project
  - Human/robot motion planning
  - Individual project encouraged (tailor to your research), teams of 2 or 3 also allowed

Piazza

- https://piazza.com/wpi/spring2017/rbe550/home
- **Piazza.com** for all class communication (question/answer about homeworks, paper discussion, etc.)



Grading

- In-Class Participation and Preparation 15%
  - Attending class and asking questions during lecture
  - Participating in paper discussions
  - Participating in discussions on Piazza
- Research Paper Presentations 15%
- Homeworks 20%
- Final Project Proposal 10%
- Final Project Report 25%
- Final Project Presentation 15%

Schedule

- Latest schedule is on course website: https://sites.google.com/site/muyimolin/teaching/rbe550

	A	B	C	D	E	F	G
1	Date	Topic	Slides	Reading			Assignment
2	1/13/2017	L: Introduction to Motion Planning	link	Principles: CH 1, 5	Graph Review		Sign up for course on Piazza
3	1/18/2017	L: Planning for point robots	link				Enter presentation topic preferances into Piazza
4	1/20/2017	L: Configuration Space	link				Homework 1 Out
5	1/25/2017	L: Discrete Motion Planning	link				
6	1/27/2017	L: Transformations	link				
7	2/1/2017	L: Collision Checking	link	Probability Review			Homework 1 Due
8	2/3/2017	L: Sampling-based Planning 1	link	How to read a paper			Homework 2 Out
9	2/8/2017	L: Sampling-based Planning 2	link	Non-holonomic motion planning guide			
10	2/10/2017	L: Non-holonomic motion planning	link	Paper Presentation Guidelines	Presentation Grading Sheet	Reviewing Guidelines	
11	2/15/2017	Presentation		ANA*	Adaptive Anytime Motion Planning For Robust Robot Navigation In Natural Environments	LEARCH	
12	2/17/2017	Presentation		Adaptive workspace biasing for sampling-based planners	Demonstration-guided motion planning	RRT*	Homework 2 Due
13	2/22/2017						
14	2/24/2017	Presentation		Randomized Kinodynamic Planning	A Sampling-Based Tree Planner for Systems with Complex Dynamics	Simulation-Based LQR-Trees with Input and State Constraints	Homework 3 Out
15	3/1/2017	L: Grasping	link				Submit Project Proposal (recommended)
16	3/3/2017	SPRING BREAK					
17	3/8/2017	SPRING BREAK					
18	3/10/2017	SPRING BREAK					
19	3/15/2017	Presentation		Data Driven Grasp Synthesis using Shape Matching and Task-Based Pruning	Grasping Objects with Holes: A Topological Approach	Probabilistic Collision Checking With Chance Constraints	Project Proposal Due

Presentations

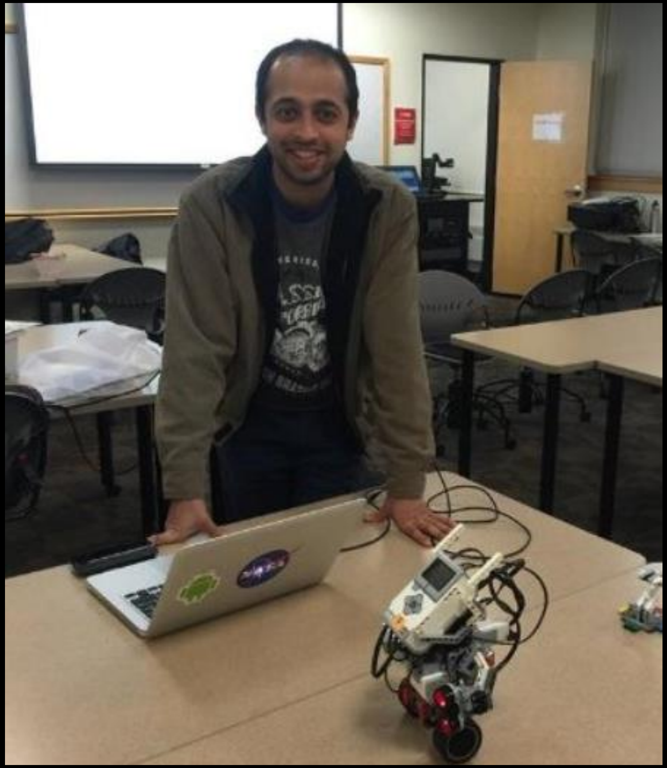
- In-depth understanding of a paper
- Tentatively 15 minutes long + 2 minutes of questions
  - Like a standard conference talk
- Evaluated on
  - Depth of understanding
  - Clarity of presentation
  - Presentation skill (don't run out of time!)

Final Project

- Schedule meeting with me to discuss project topic and expectation
- Preparation (now ~ March 1)
  - Literature survey
  - Acquire necessary skills
  - Work on project proposal -- clear timeline
- Project proposal due March 15
- Final project progress report due April 12
- Final project presentation + Final report due on April 28

First person to ask for help – TA Aditya Bhat

- Email - abhat@wpi.edu
- Office: 85 Prescott 224



Introduction to Motion Planning

Jane Li

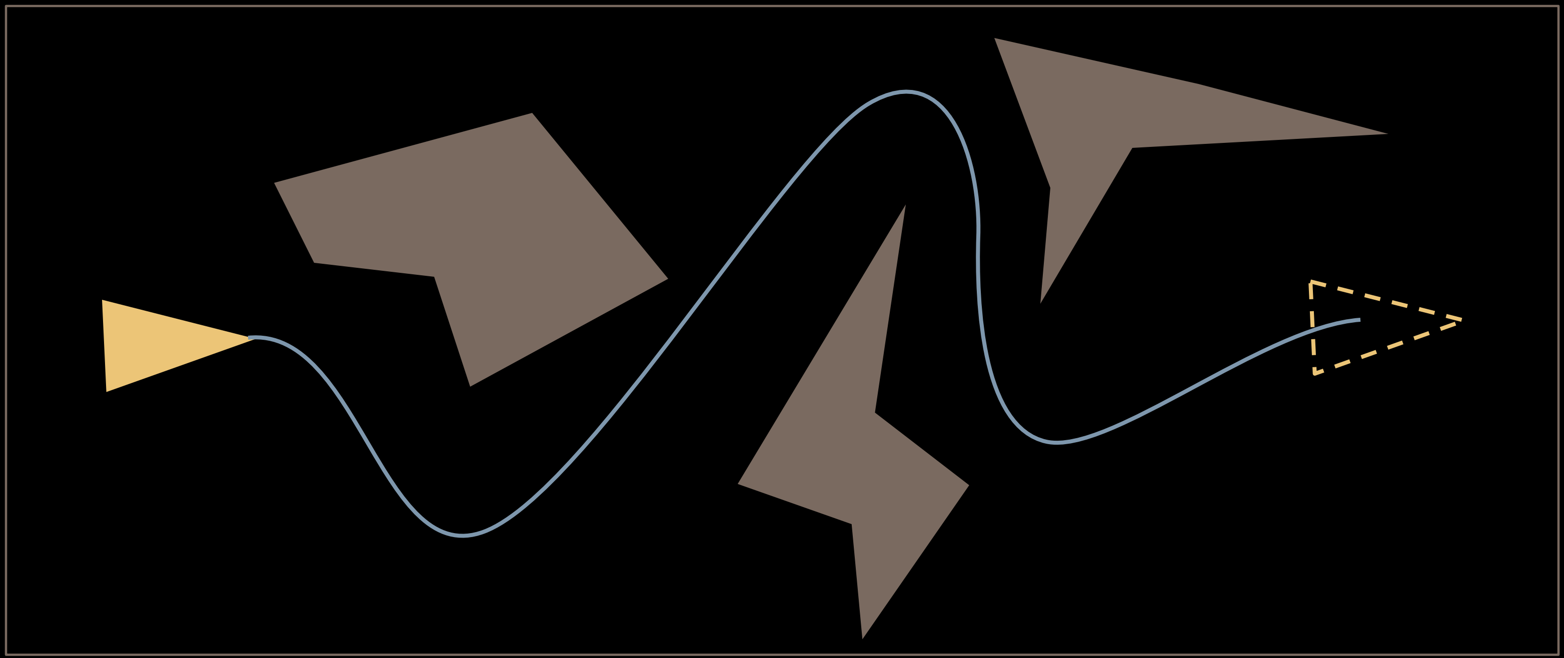
Assistant Professor

Mechanical Engineering & Robotics Engineering

http://users.wpi.edu/~zli11

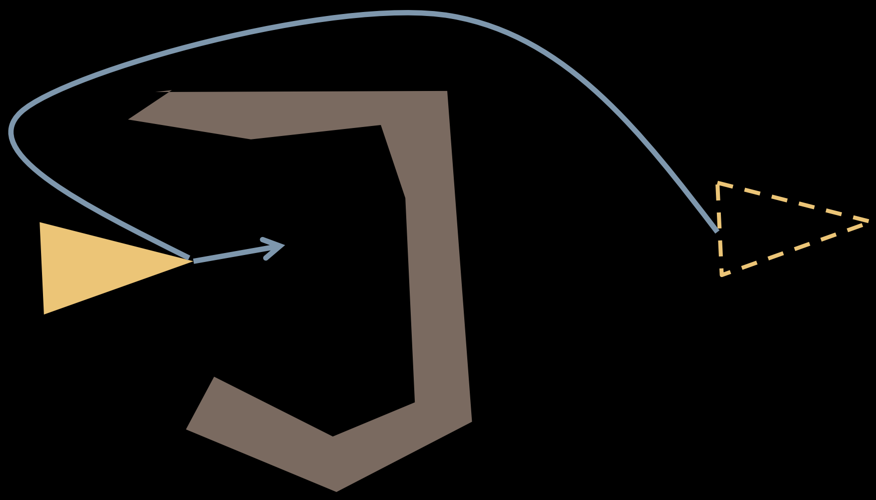
What is motion planning?

- The automatic generation of motion
  - Path + velocity and acceleration along the path

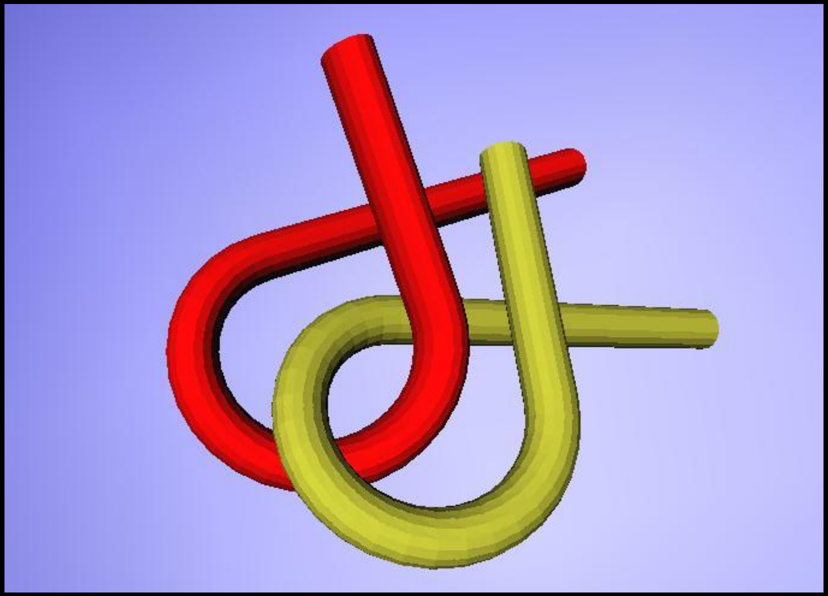


Why Motion Planning Instead of Obstacle Avoidance?

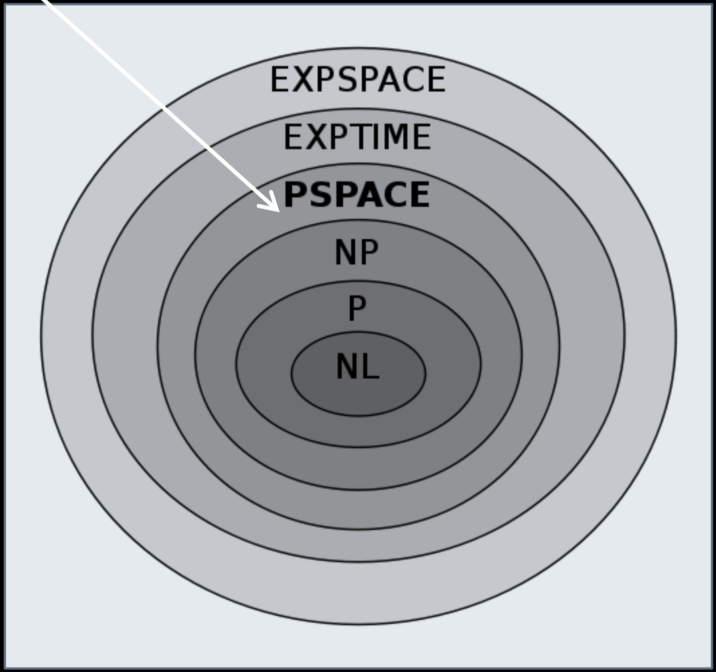
- Path planning
  - low-frequency, time-intensive search method for global finding of a (optimal) path to a goal
- Obstacle avoidance (aka "local navigation")
  - fast, reactive method with local time and space horizon
- Distinction: Global vs. local reasoning



Is motion planning hard?



Basic Motion
Planning Problems

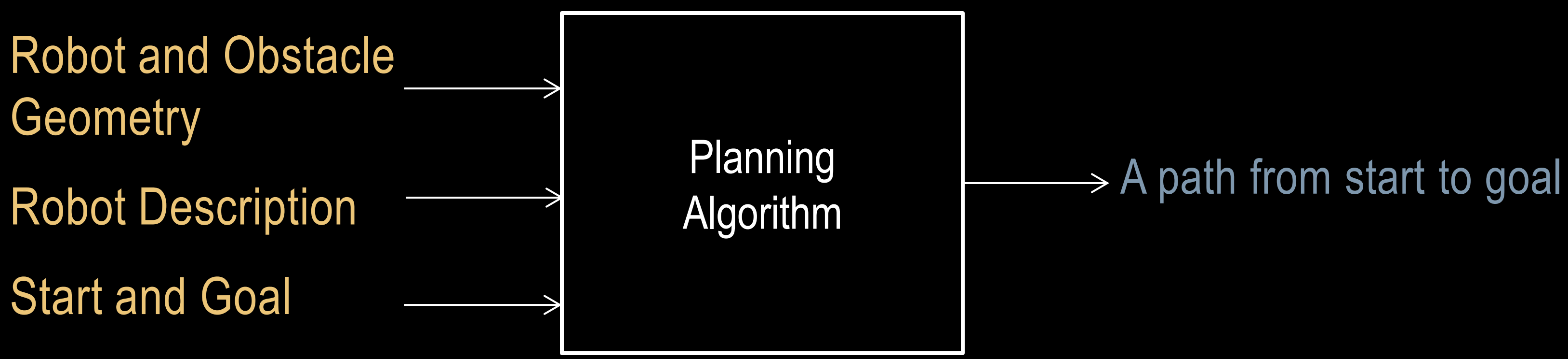


Related Fields and Contents

- Intersection of three fields
  - **Robotics**
  - Control theory
  - AI
- Topics
  - Planning in **discrete space**
  - Planning in **continuous space**
  - Planning under **uncertainty**
  - Planning under **differential constraints**

Basic Problem Statement

- Motion planning in robotics
  - Automatically compute a path for an object/robot that does not collide with obstacles.



Basic Motion Planning Problem

- Examples
  - Piano mover's problem – 3D
  - Sofa mover's problem – 2D
  - Generalized mover's problem

- Key to the problem
  - Make sure not point on the robot hits an obstacle
  - All **kinematic** motion planning

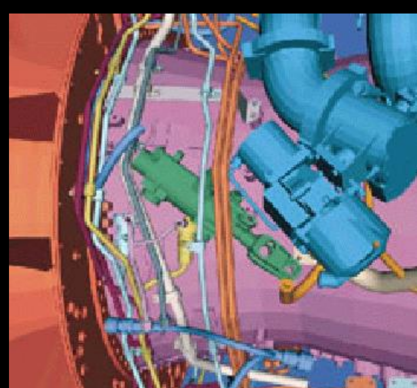
More than kinematic motion planning …

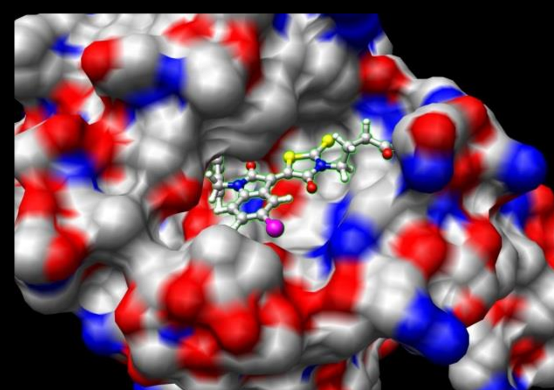
- Trajectory planning (speed & acceleration along the path)
- Nonholonomic constraints (e.g. parallel parking)
- Sensor-based motion planning (deal with uncertainty)
- SLAM (e.g. planetary exploration)
- Coverage path planning (e.g. vacuum robot, demining)
- Hyper-redundant robot (e.g. snake robot for search and rescue)
- Human-like motion (e.g. animated characters)
- Drug design (e.g. protein as linked rigid bodies)

Applications

- Automatically generate motion

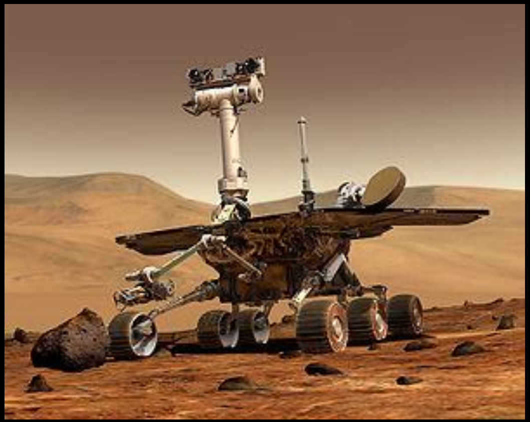
- Automatically validate





Applications: Mobile Robots

Roomba Create



Mars Rovers

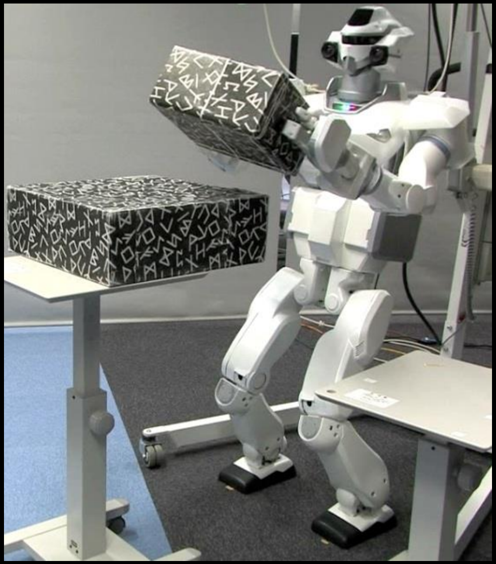
DARPA Urban Challenge

Google Self-Driving Car

Applications: Robotic Manipulation

Factory Automation

Personal Robots

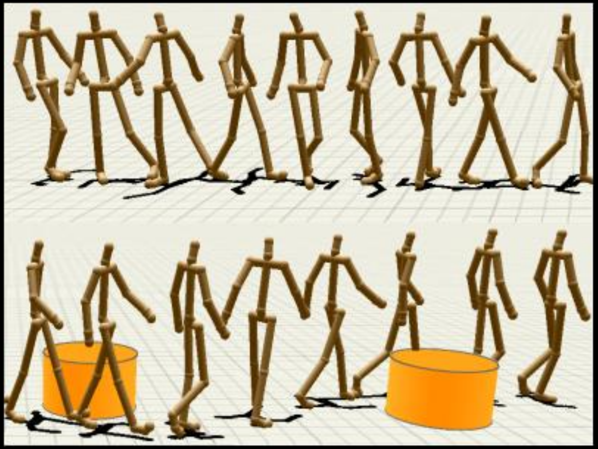


Humanoid Robots

Personal Robots

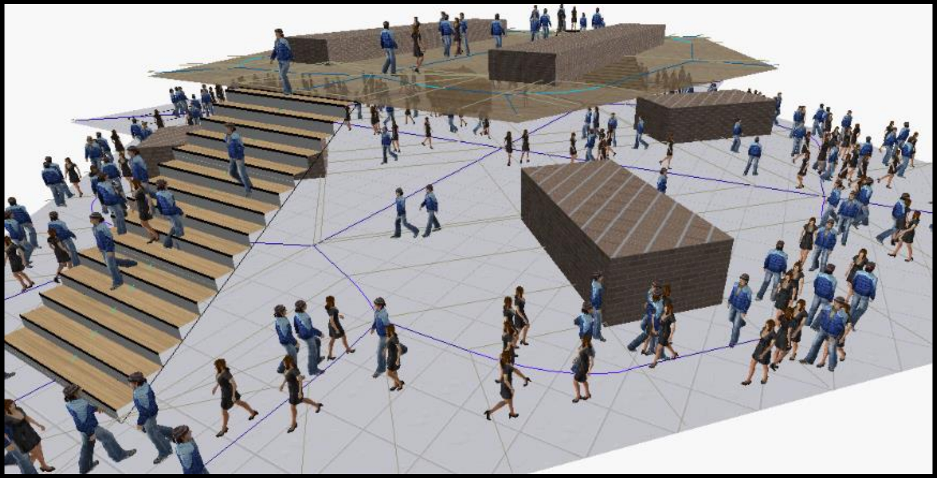
Applications: Computer Games/Graphics

Path Finding in Games



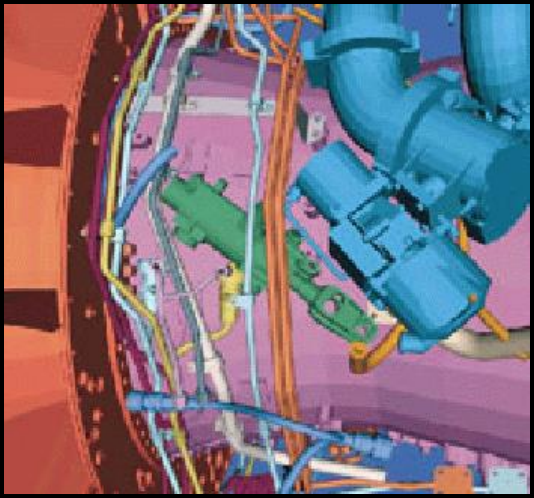
Character Animation

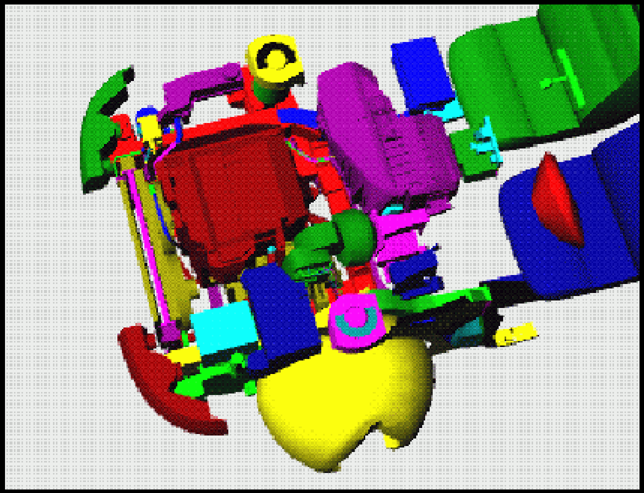
Retargeting Motion Capture

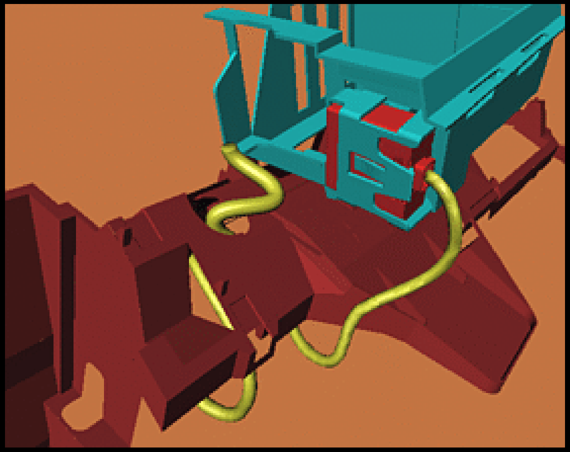


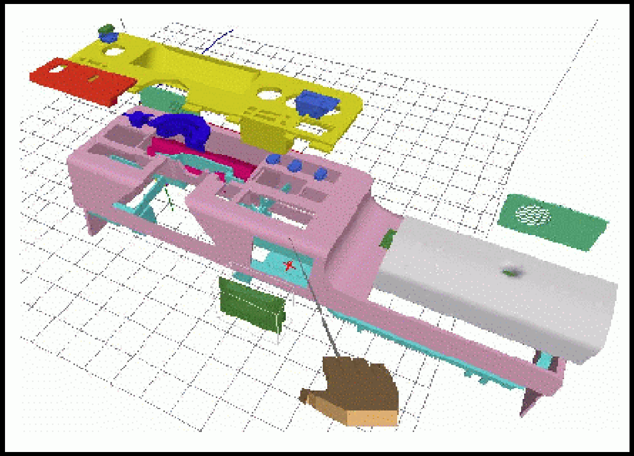
Animation of Crowds

Applications: Assembly Planning

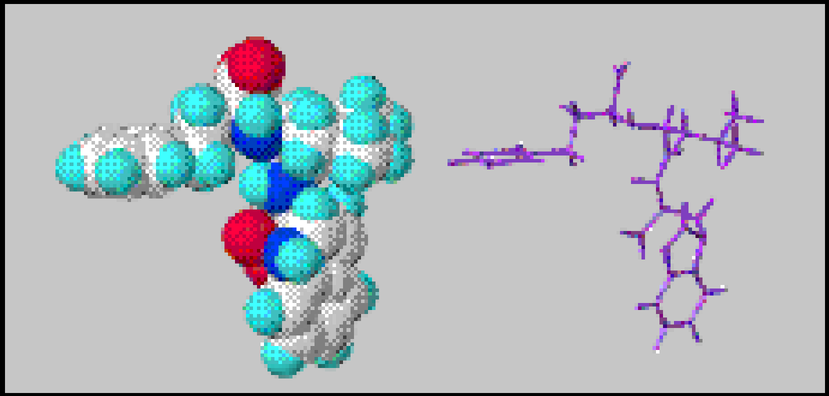


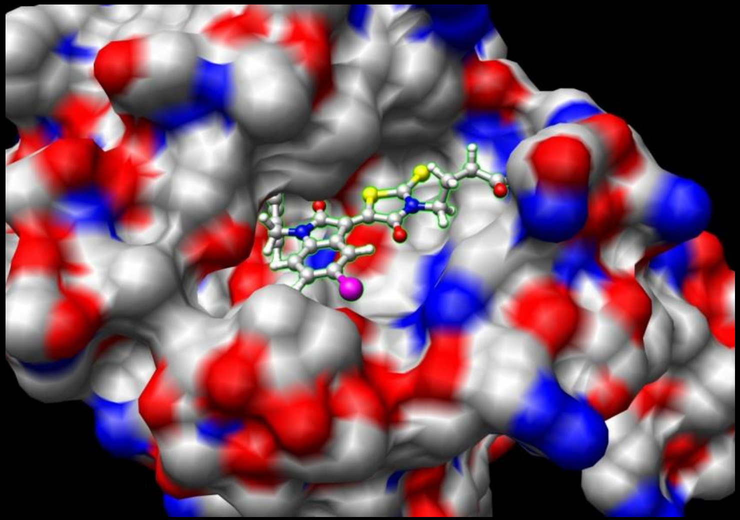


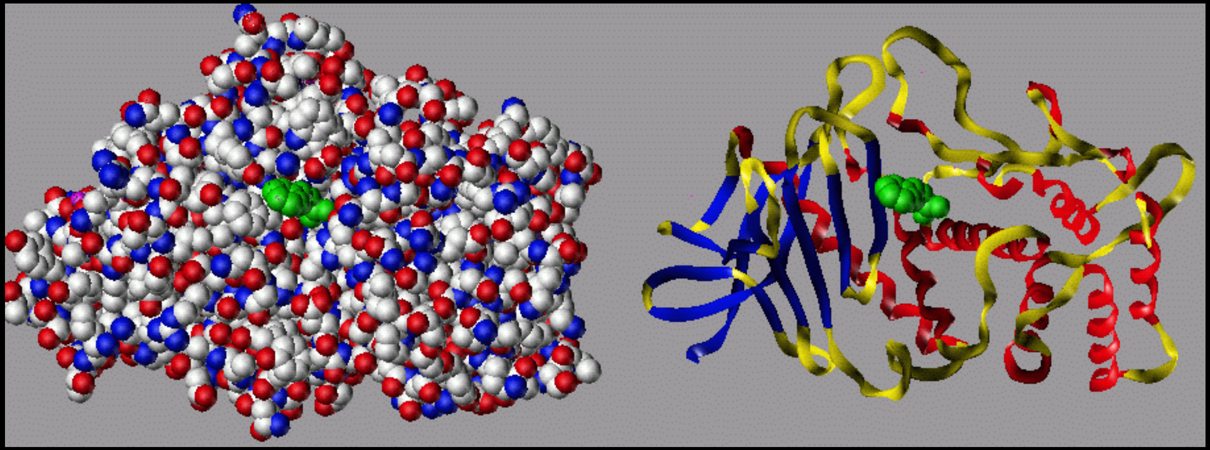




Applications: Computational Biology

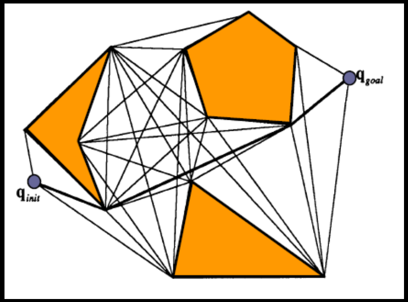




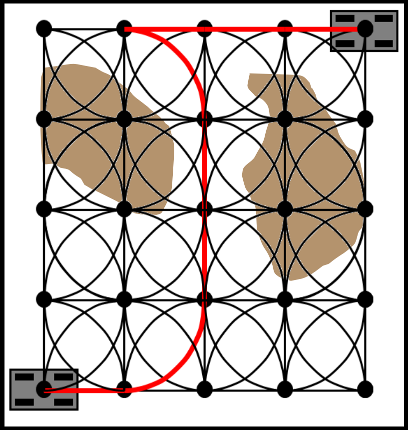


Approaches

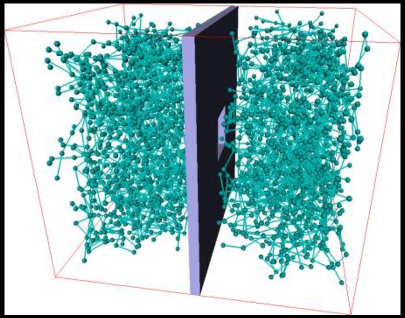
- Exact algorithms
  - Either find a solution or prove none exists
  - Very computationally expensive
  - Unsuitable for high-dimensional spaces



- Discrete Search
  - Divide space into a grid, use A* to search
  - Good for vehicle planning
  - Unsuitable for high-dimensional spaces



- Sampling-based Planning
  - Sample the C-space, construct path from samples
  - Good for high-dimensional spaces
  - Weak completeness and optimality guarantees



What matters?

- Motion planning algorithms are judged on
  - Completeness
  - Optimality
  - Speed (AKA efficiency)
  - Generality
- These vary in importance depending on the application

What matters: Completeness

- Will the algorithm solve all solvable problems?
- Will the algorithm return no solution for unsolvable problems?
- What if the algorithm is probabilistic?

- For what application(s) is completeness very important?
- For what application(s) is completeness not important?

What matters: Optimality

- Will the algorithm generate the shortest path?
- Will the algorithm generate the least-cost path (for an arbitrary cost function)?
- Do we need optimality or is feasibility enough?

- For what application(s) is completeness very important?
- For what application(s) is completeness not important?

What matters: Speed (AKA Efficiency)

- How long does it take to generate a path for **real-world** problems?
- How does the run-time scale with dimensionality of the problem and complexity of models?
- Is there a quality vs. computation time tradeoff?

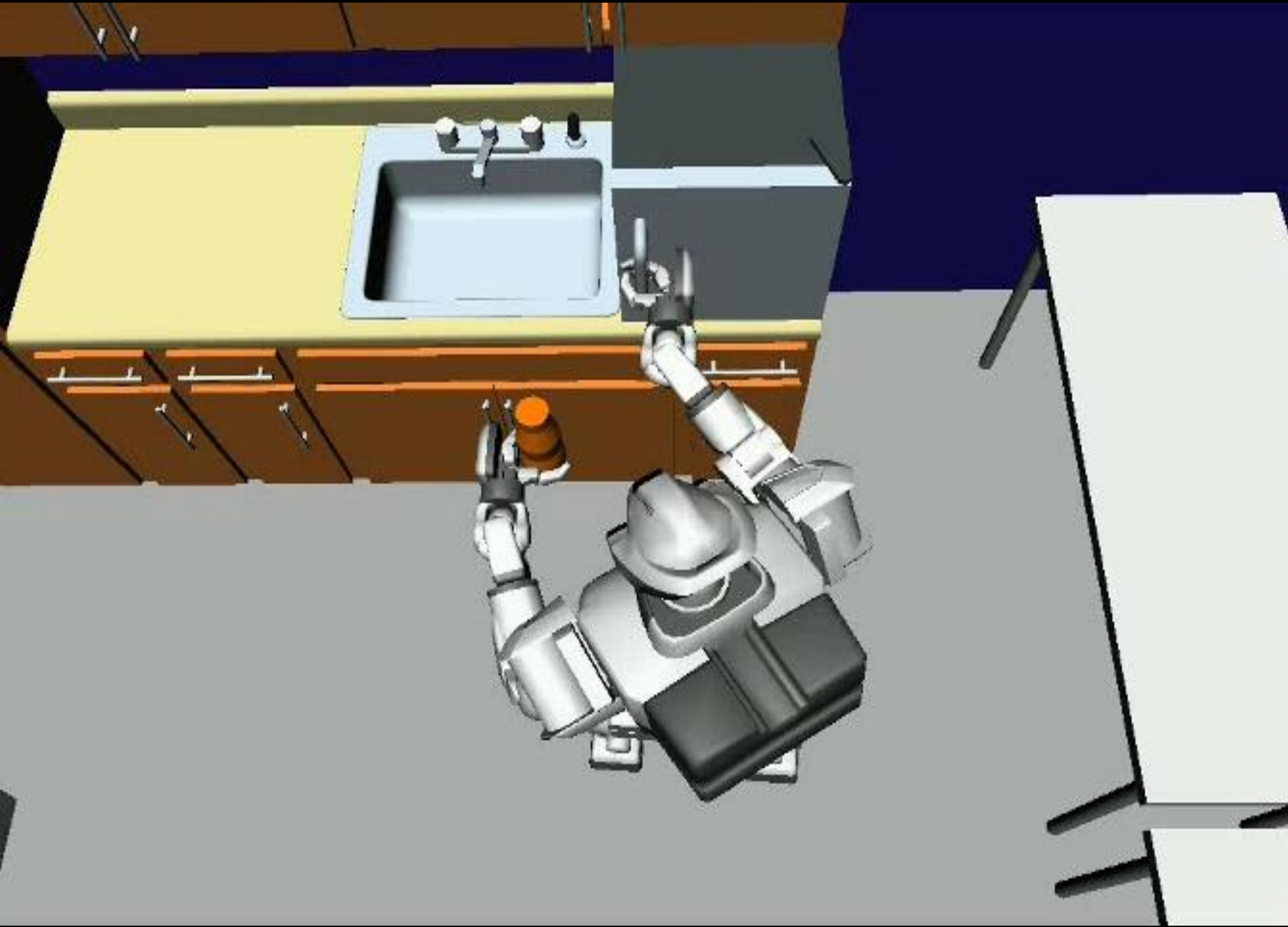
- For what application(s) is completeness very important?
- For what application(s) is completeness not important?

What matters: Generality

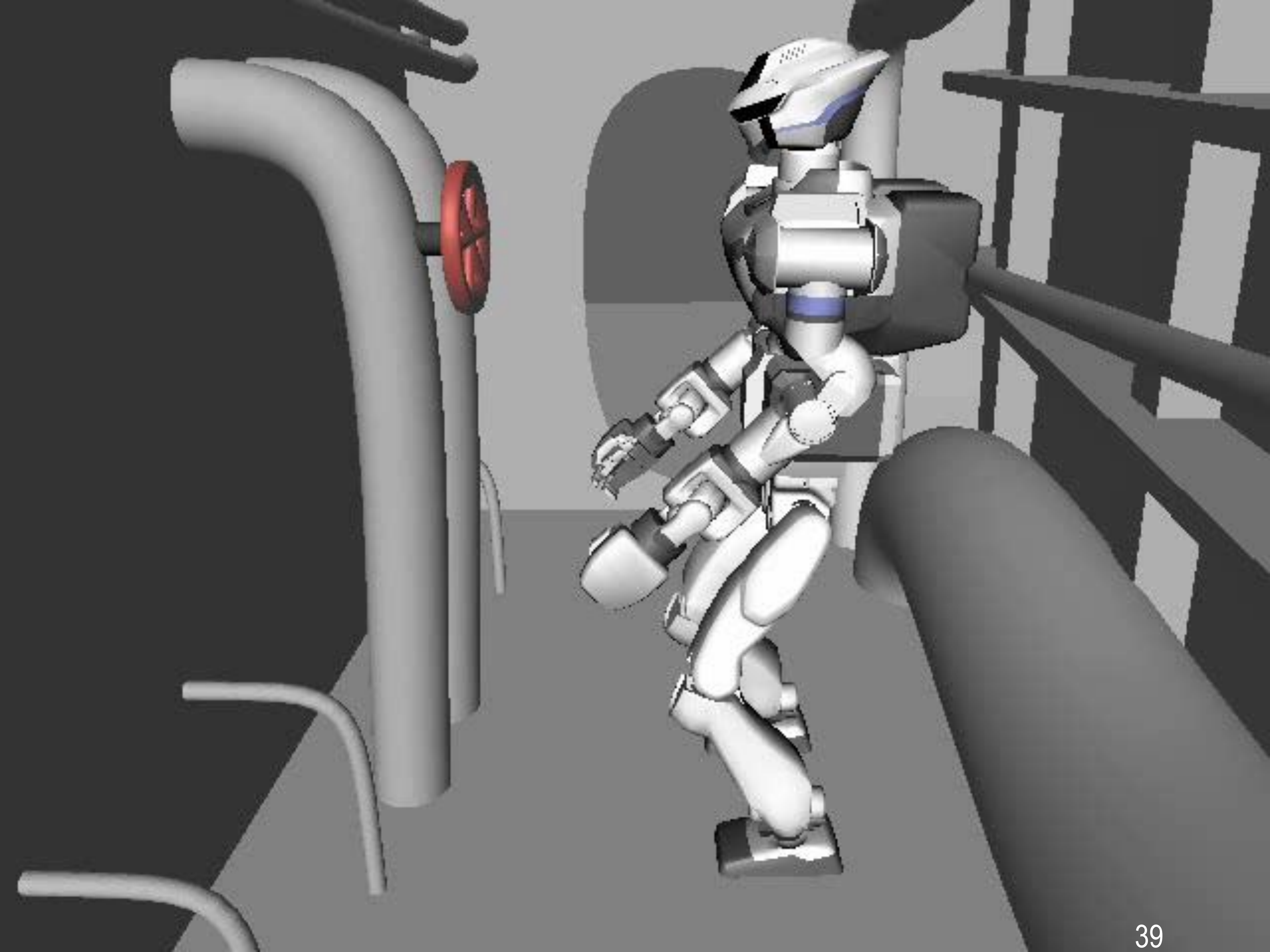
- What types of problems can it solve?
- What types of problems can't it solve?

- For what application(s) is completeness very important?
- For what application(s) is completeness not important?

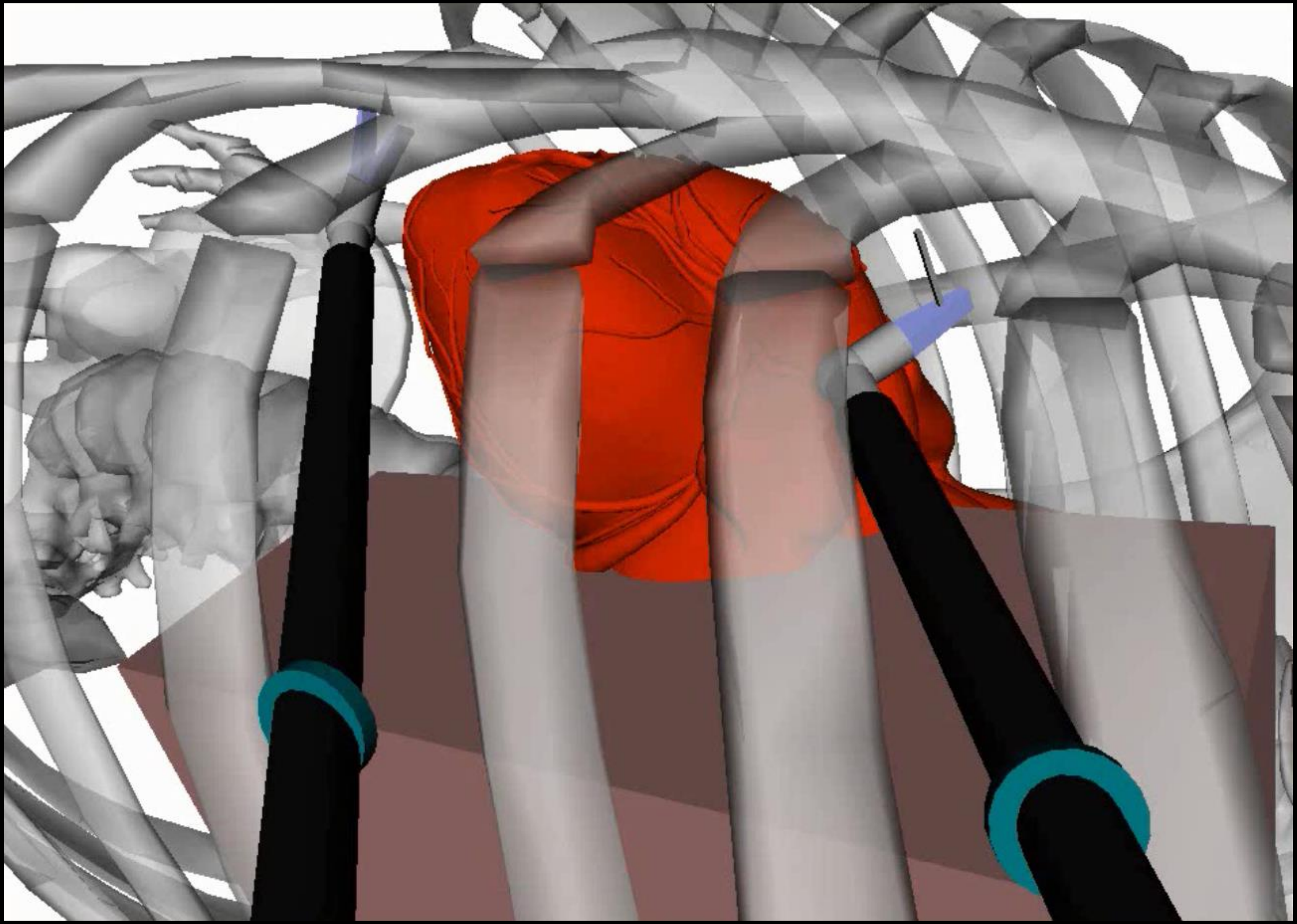

w = 4.98kg
intel



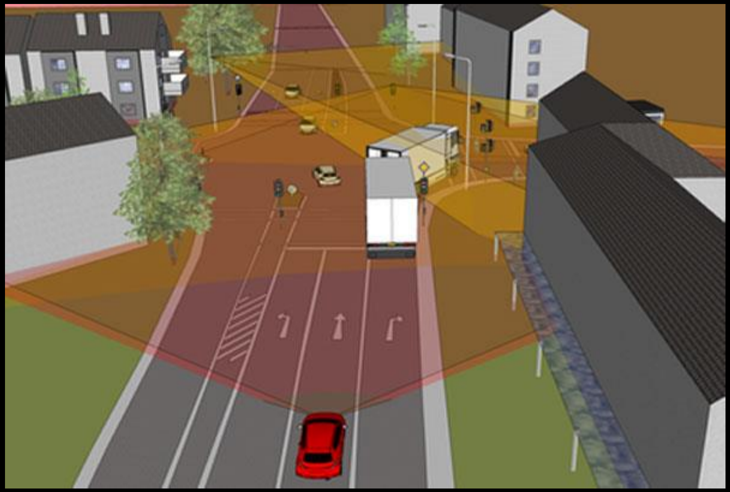

x3

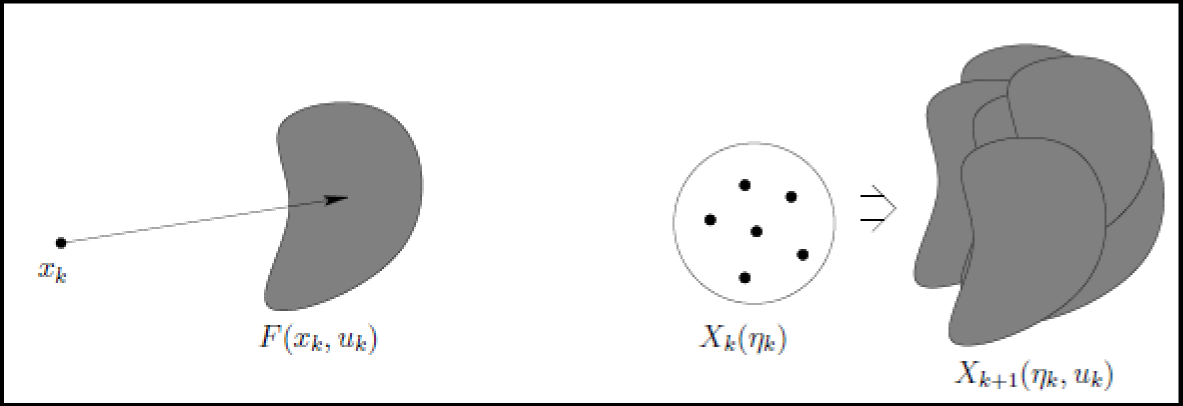


Robotic Surgery



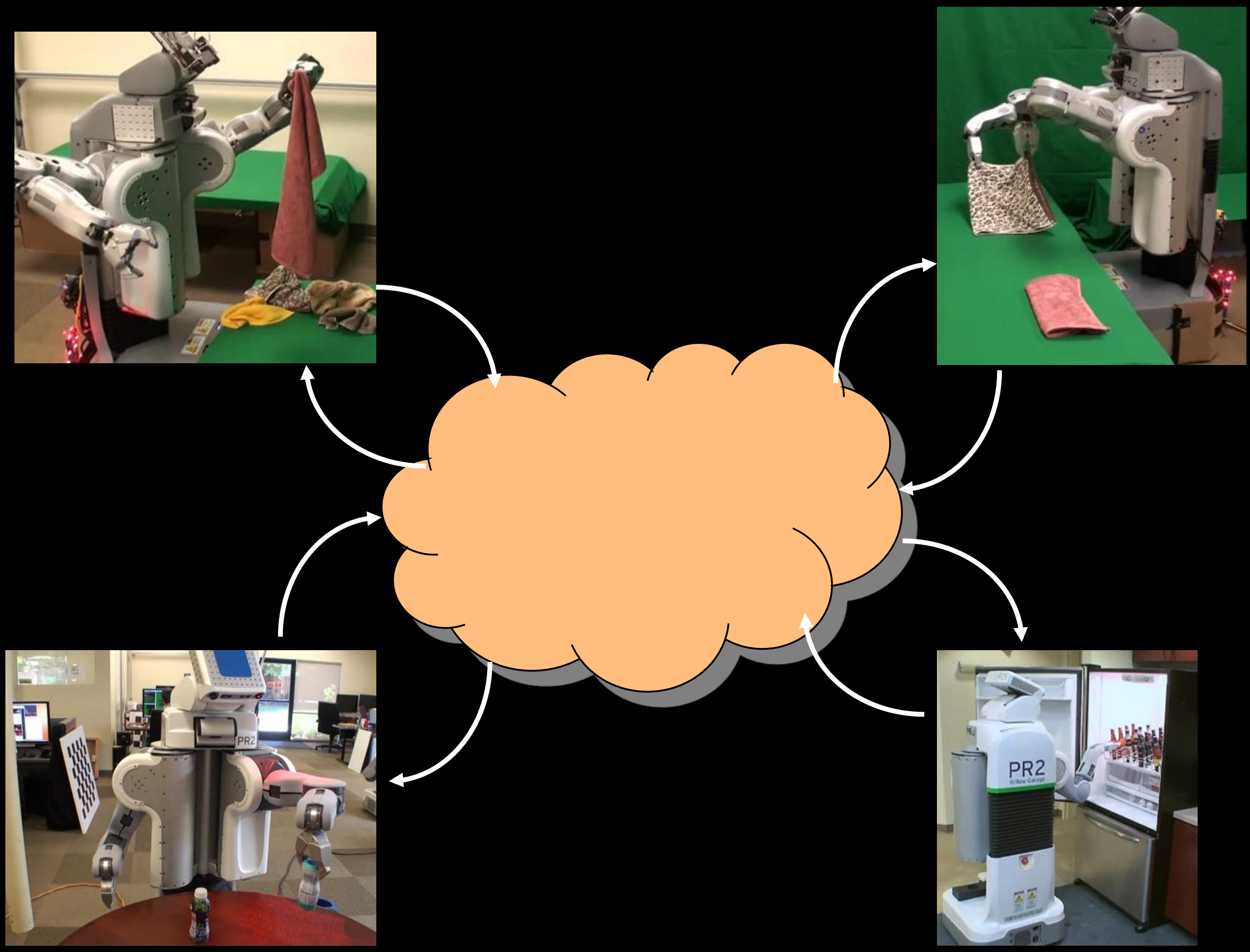
Information Space and Sensor Uncertainty





Consider multiple states of the world

Planning with the Cloud

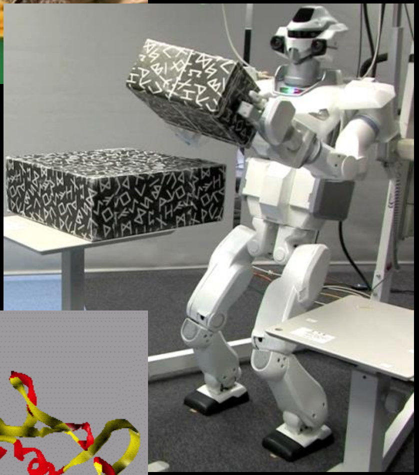
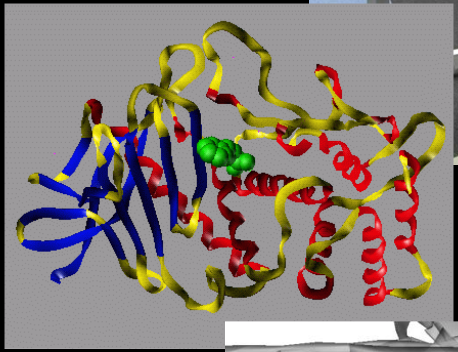
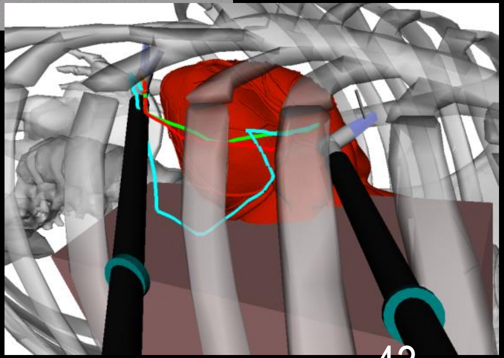


Summary

- Robot motion planning is used for…
  - Vehicles
  - Robots
  - Digital Characters
  - Molecules
  - Design verification

- Types of planning methods
  - Exact algorithms
  - Discrete Search
  - Sampling-based planners

- *Many frontiers to explore*

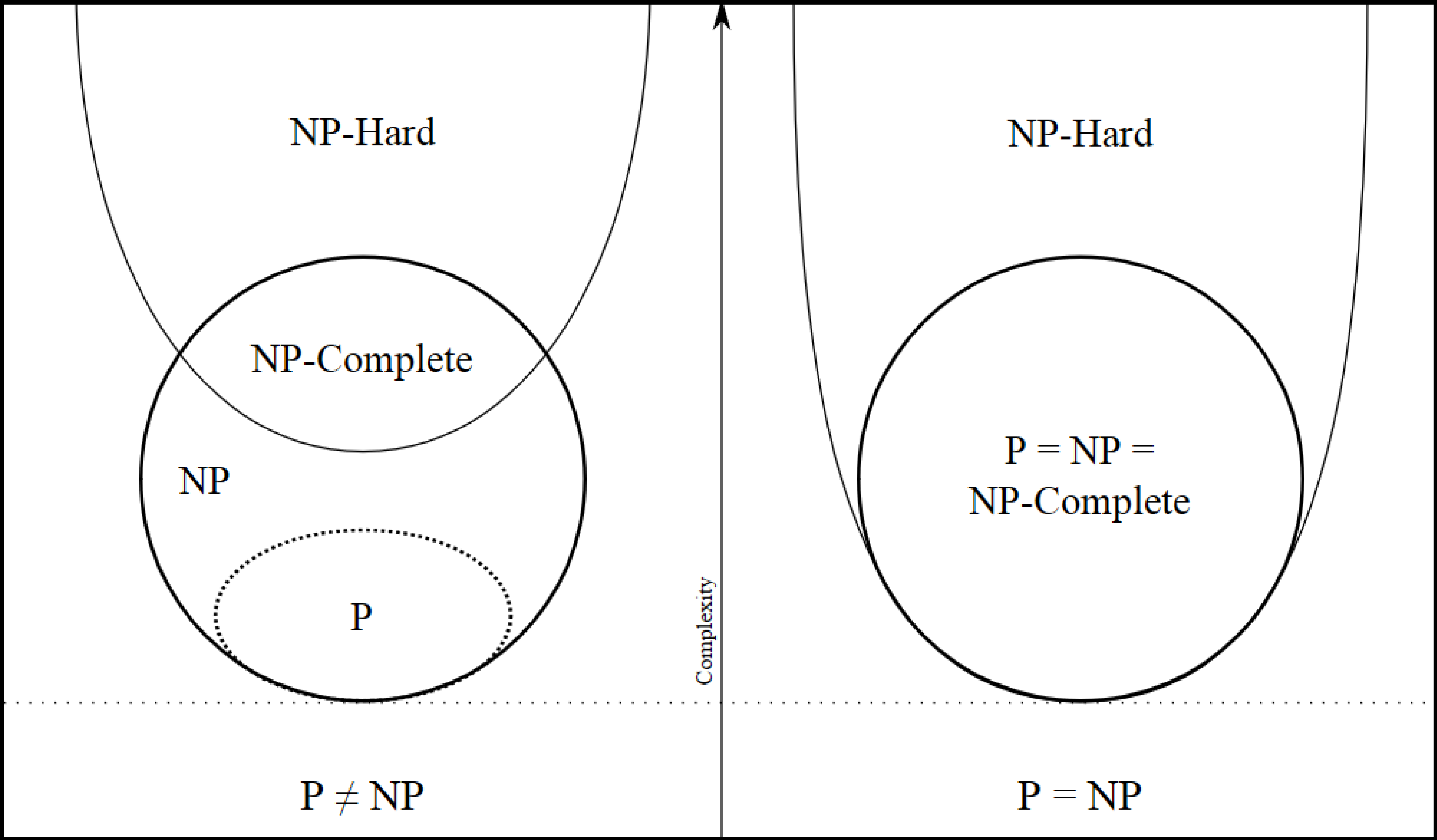





Homework

- Reading
  - Recap:
    - CH 1 - Introduction
    - Principles Appendix G - Analysis of Algorithm and Complexity Classes
  - Next class:
    - Graph Review (short)
    - Principles CH 4.0-4.1, 4.4 - Potential Fields
    - Principles CH 5.0-5.2 - Roadmaps, Voronoi diagram
    - Principles CH 6.0-6.1 - Cell decomposition
- Sign up for course on Piazza

End

NP Complete


NP-Hard
NP-Complete
NP
P
P ≠ NP
Complexity
NP-Hard
P = NP =
NP-Complete
P = NP