AMIRKABIR WINTER SCHOOL

Minimalism in Robotics: From Sensing to Filtering to Planning

PART 4: PLANNING WITH PERFECT SENSING

Steven M. LaValle

March 3, 2012

Perfect Sensing

For this part, suppose $h : X \rightarrow Y$ is a bijective sensor.

Possible state spaces:

- $X = E \times S^1$
- $X \subset \mathbb{R}^2 \times S^1 \times \mathcal{E}$
- More generally, replace $SE(2)$ by any *configuration space*.

The information space becomes $\mathcal{I} = X$.
We are at the top of the sensor lattice at all times.
There is no uncertainty with respect to sensing.
Once h is given, sensing is *trivialized*!

Historical Perspective

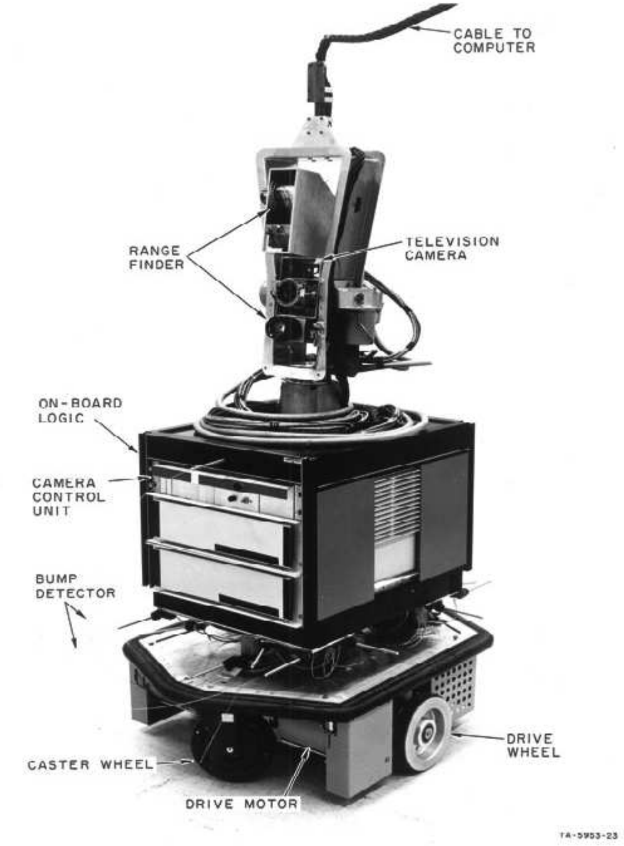
Nilsson, Stanford, late 1960s:

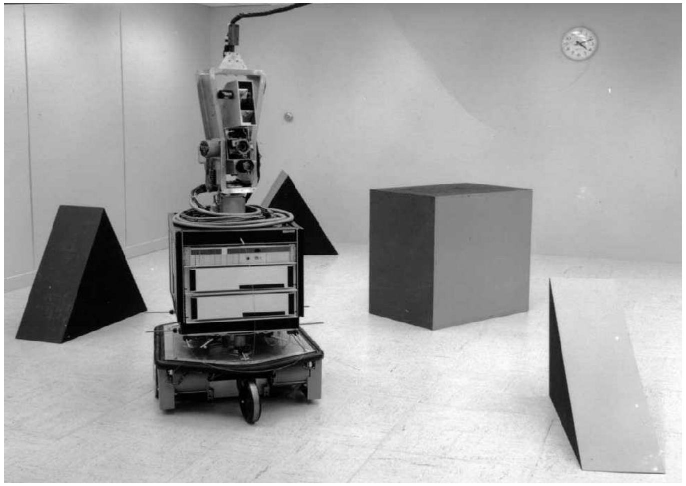
Shakey, A^*, visibility graphs, STRIPS

What is planning? Automated sequential decision making with heavy emphasis on algorithms and computation.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

In the beginning (1960s)...





Clear challenges: sensing, mapping, planning, ...

Algorithms Need Discretizations

The world is more or less continuous.

Computation is discrete.

- 1970s: Grids, logic-based planning
- 1980s: Combinatorial motion planning
- 1990s: Sampling-based motion planning

Planning problems are *implicitly* encoded.

Even with a complete model and perfect sensing, the space in which to search is much larger than the input representation.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

The Configuration Space

The *configuration space* (C-space) is the set of all geometric transformations that can be applied to a robot.

It is usually defined as a *topological manifold*, $\mathcal{C}$, which can be considered as an m dimensional surface embedded in $\mathbb{R}^n$ for some $m \leq n$.

The dimension of $\mathcal{C}$ corresponds to the number of *degrees of freedom* of the robot.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

The Configuration Space

For a planar mobile robot:

$\mathcal{C} = SE(2)$ or $\mathcal{C} = \mathbb{R}^2 \times S^1$.

$\mathcal{C}$ has three dimensions.

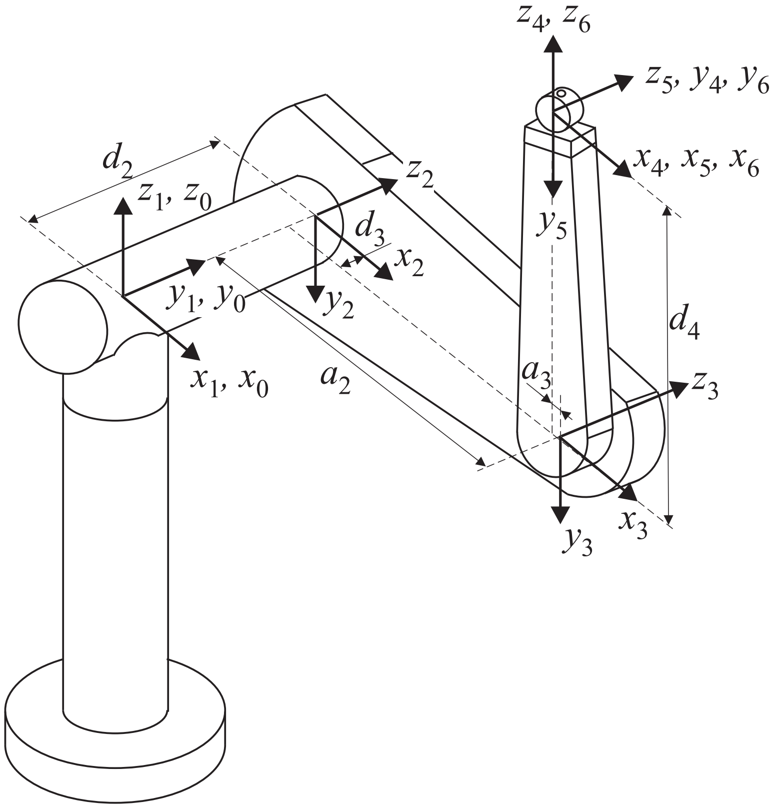
The Configuration Space

For a 3D rigid body:

$\mathcal{C} = SE(3)$ or $\mathcal{C} = \mathbb{R}^3 \times \mathbb{R}P^3$.

$\mathcal{C}$ has six dimensions.

For a manipulator based on revolute joints:



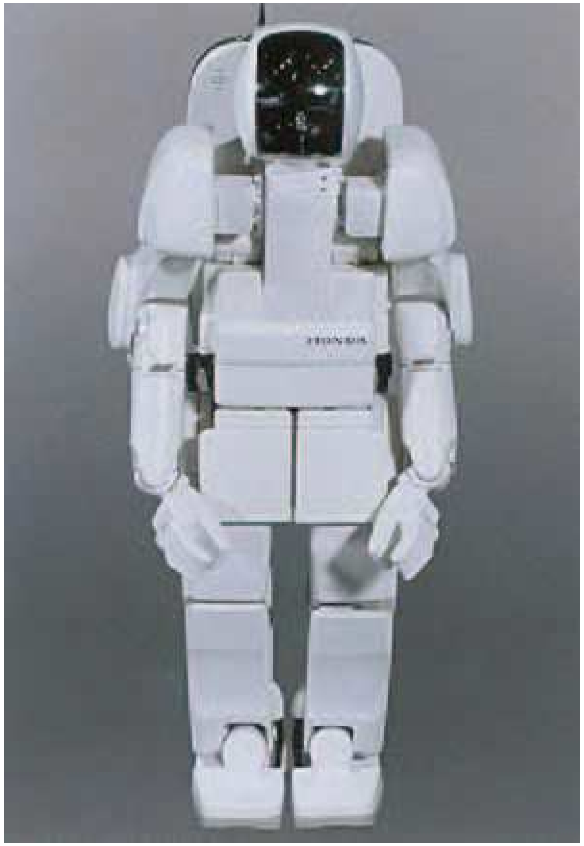
$\mathcal{C}$ is the Cartesian product of copies of $\mathbb{R}$ or S^1.

The dimension of $\mathcal{C}$ is the number of joints.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

The Configuration Space

For a humanoid robot:



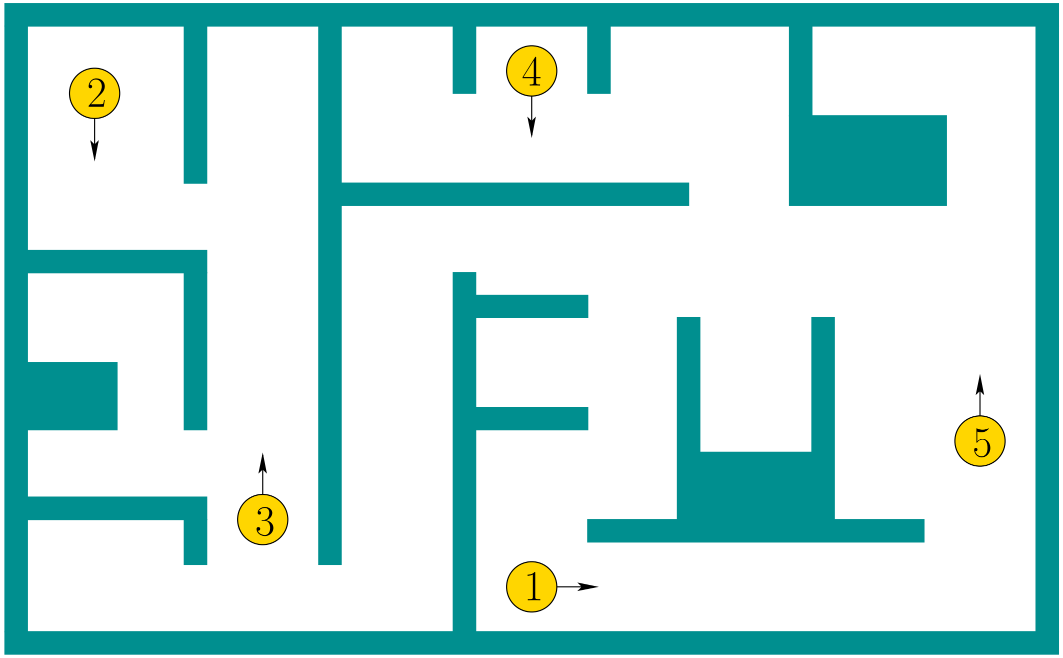
Components of $\mathcal{C}$ depend on joint types.

$\mathcal{C}$ has dozens of dimensions (for example, 80).

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

The Configuration Space

For multiple robots:

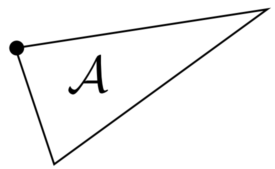


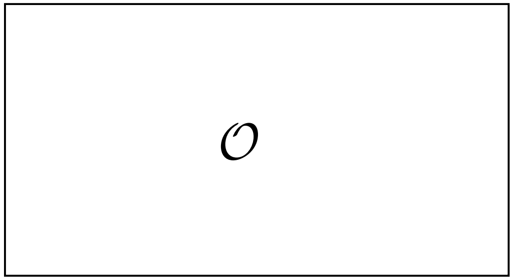
$\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_n$

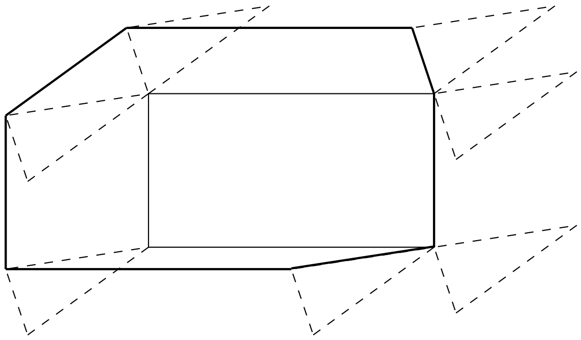
With n planar robots, the dimension of $\mathcal{C}$ is $3n$.

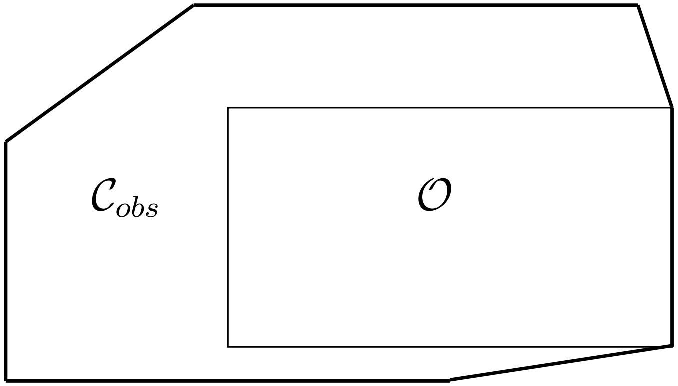
The C-Space Obstacles

Lozano-Perez, 1979 (based on Lagrangian mechanics ideas)





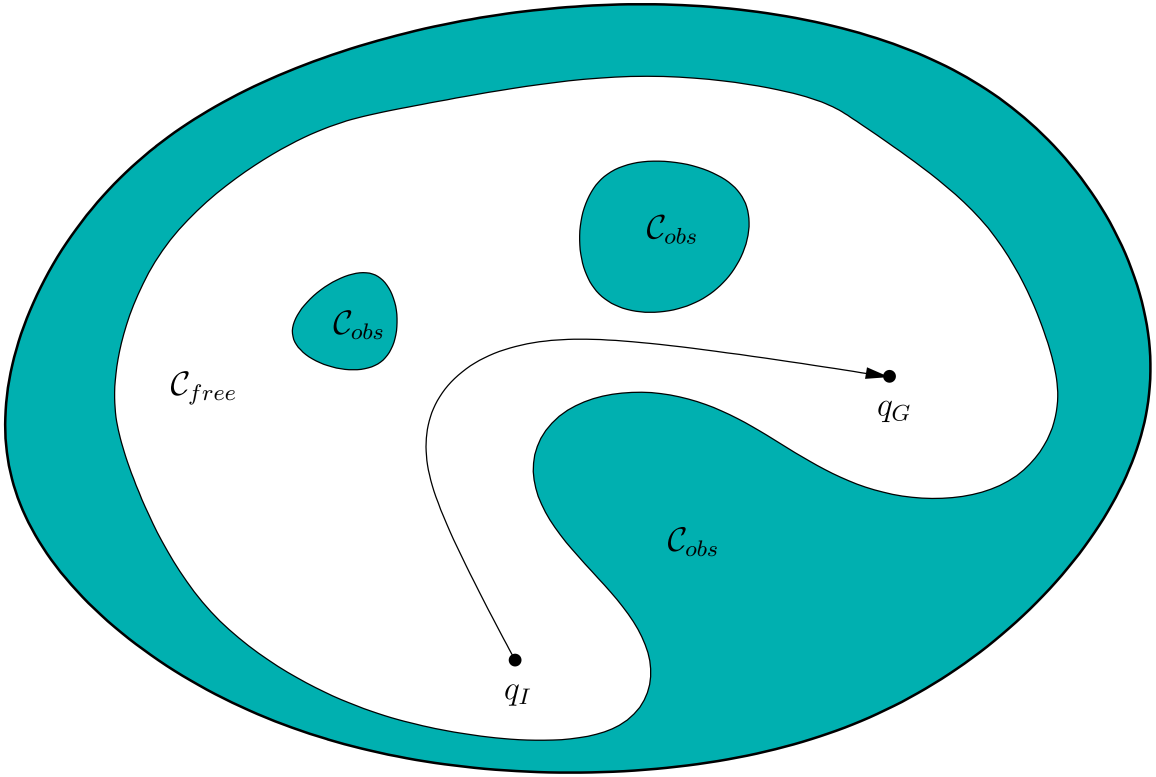




Reasoning about exact geometry

The C-Space Obstacles

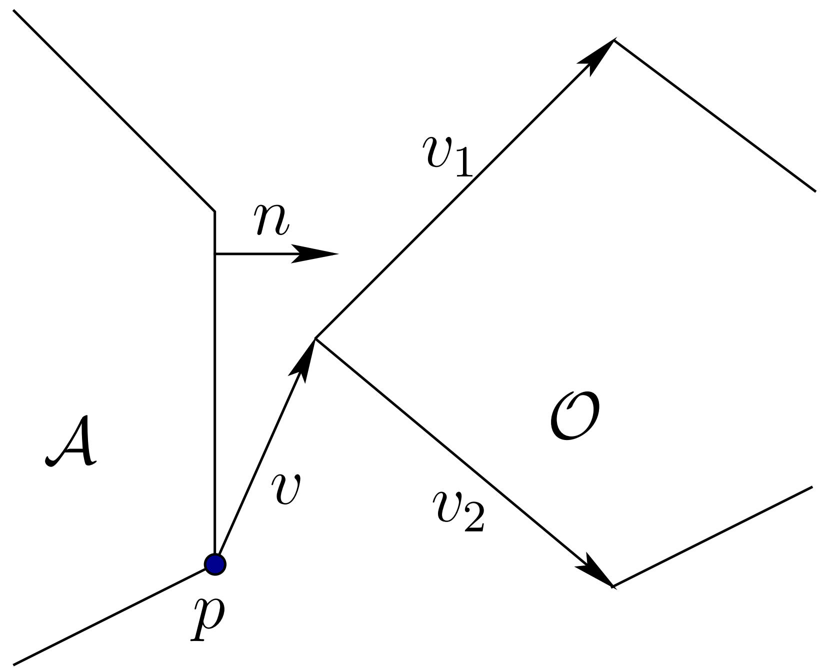
Think in the C-space...



Motion planning progressed after identifying the right spaces.

Exact Characterization: High Complexity

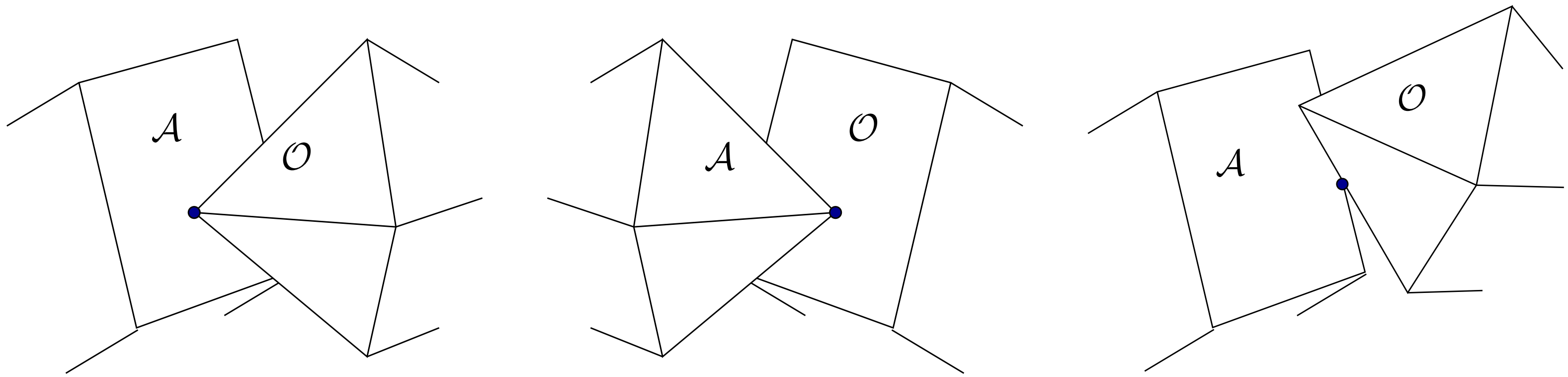
Contact conditions once rotation is allowed:



Piecewise algebraic surfaces describe the boundary of $\mathcal{C}_{obs}$.

Exact Characterization: High Complexity

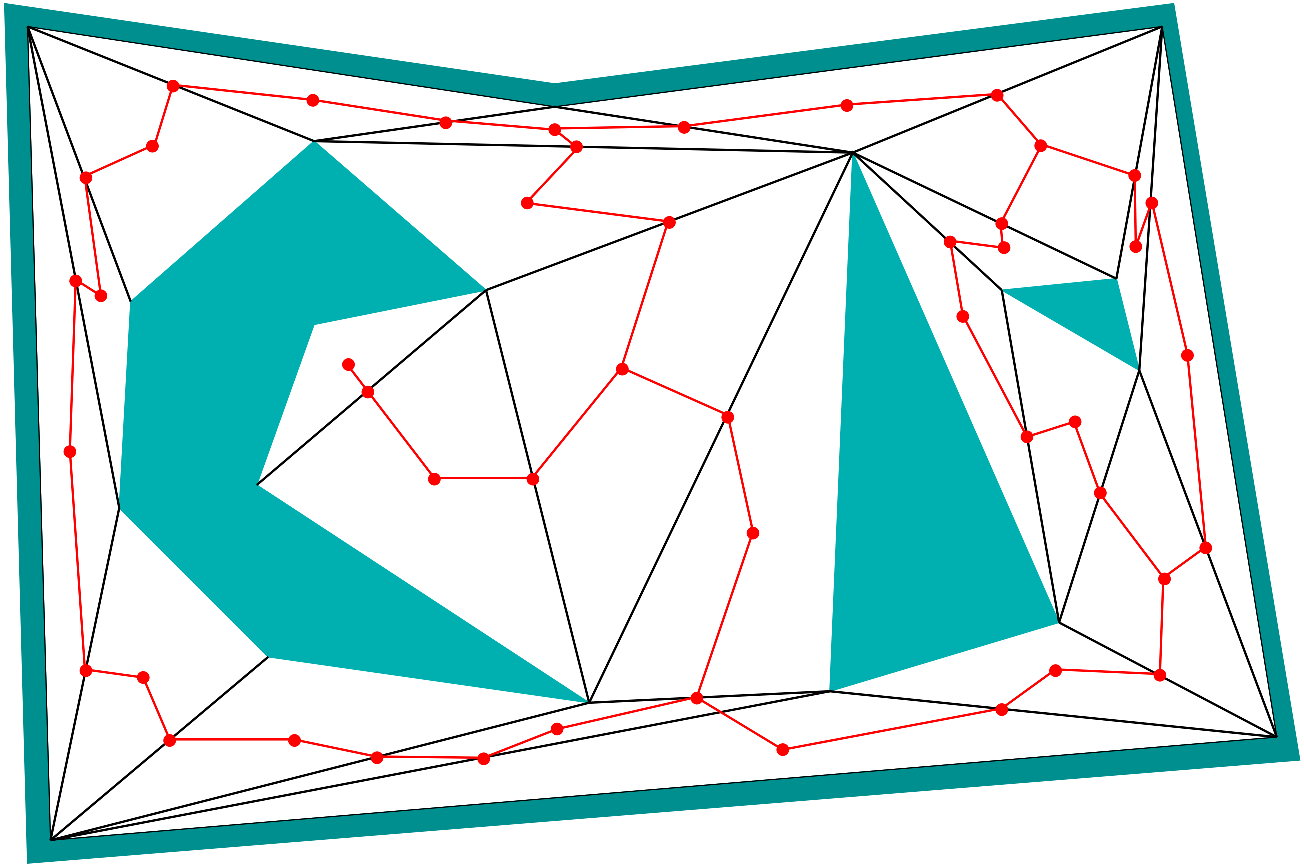
Worse for 3D rigid bodies:



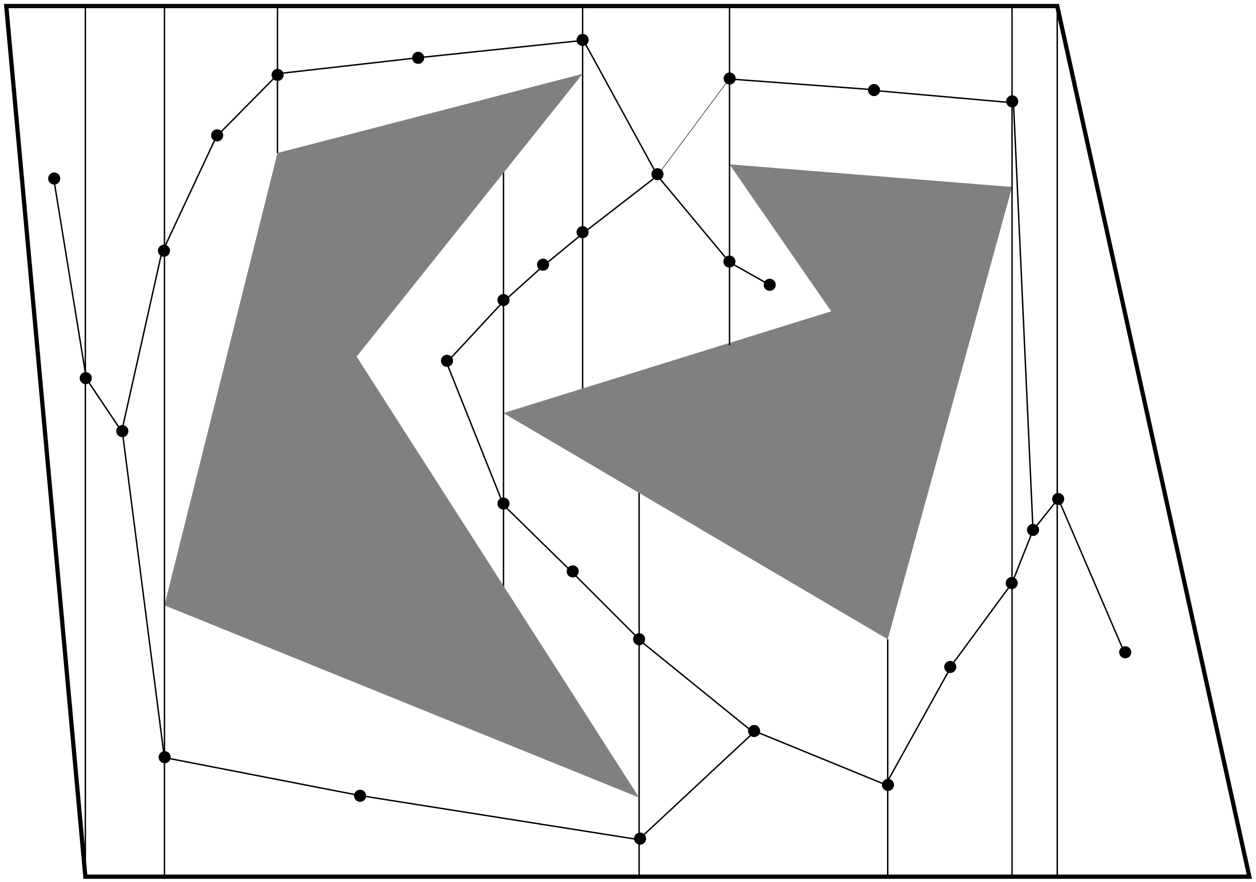
Type FV   Type VF   Type EE

Imagine what happens for humanoid robots!

Combinatorial Motion Planning

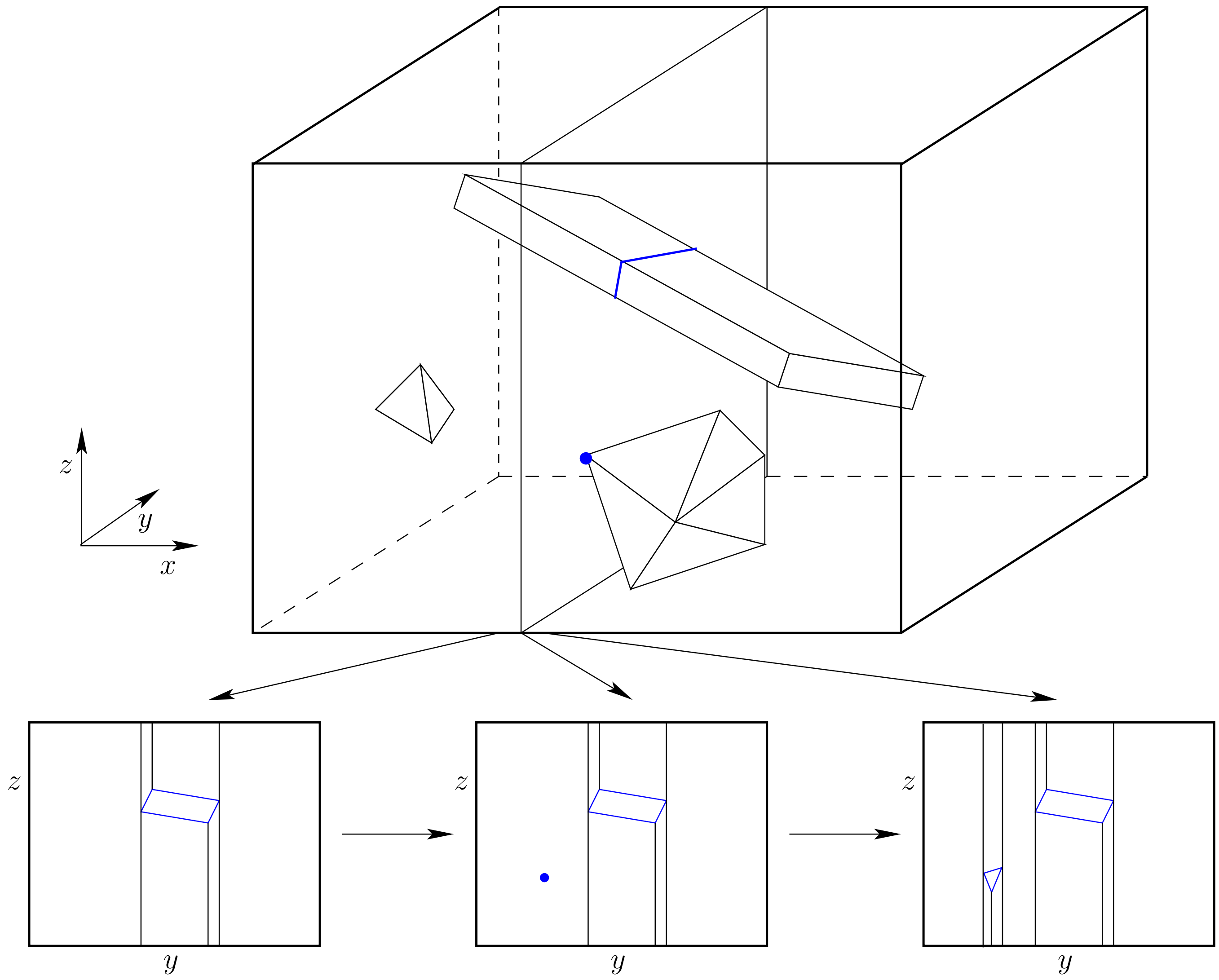


Triangulation

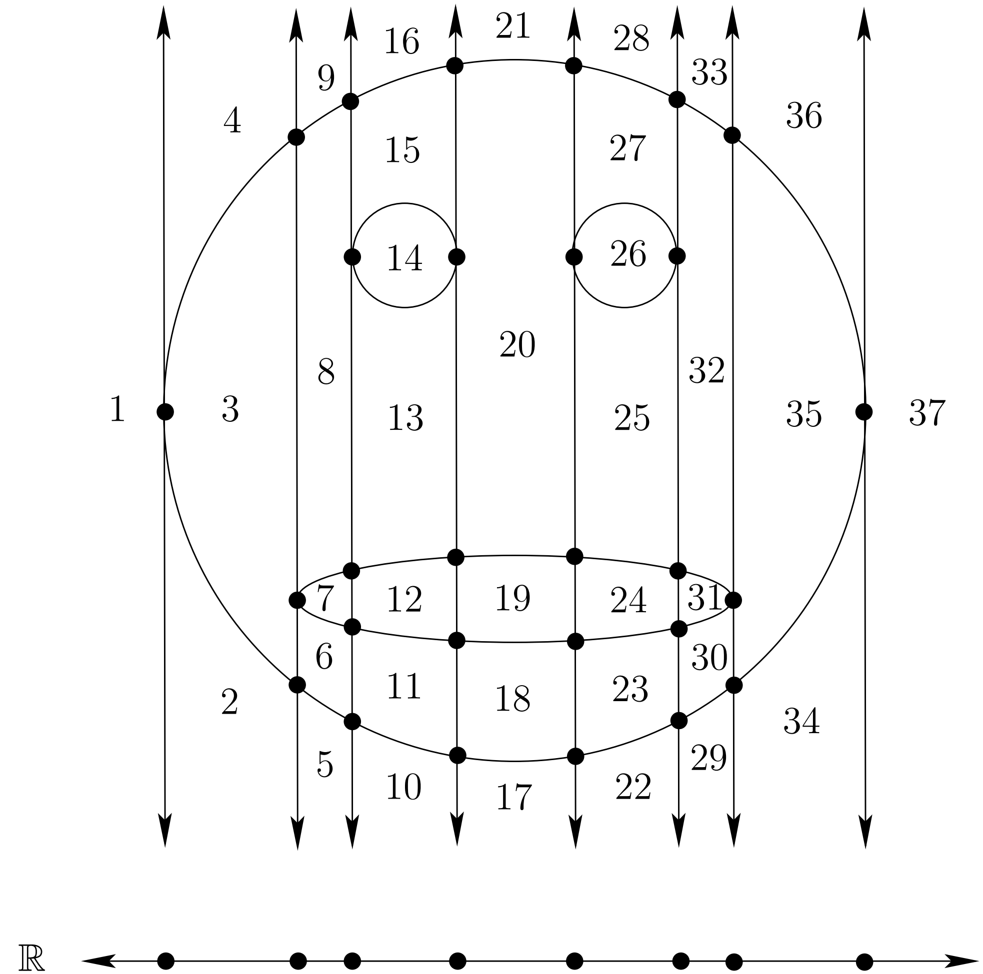


Vertical decomposition

3D vertical decomposition



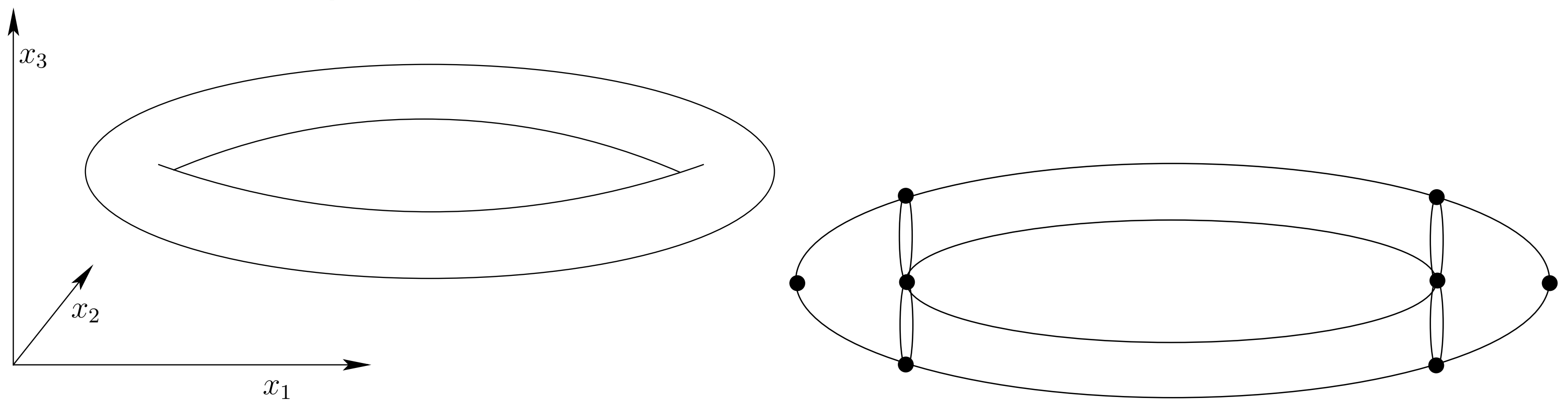
Cylindrical algebraic decomposition (Schwartz, Sharir, 1983)



Doubly exponentially many cells in dimension.

Combinatorial Motion Planning

Canny's roadmap algorithm, 1987:

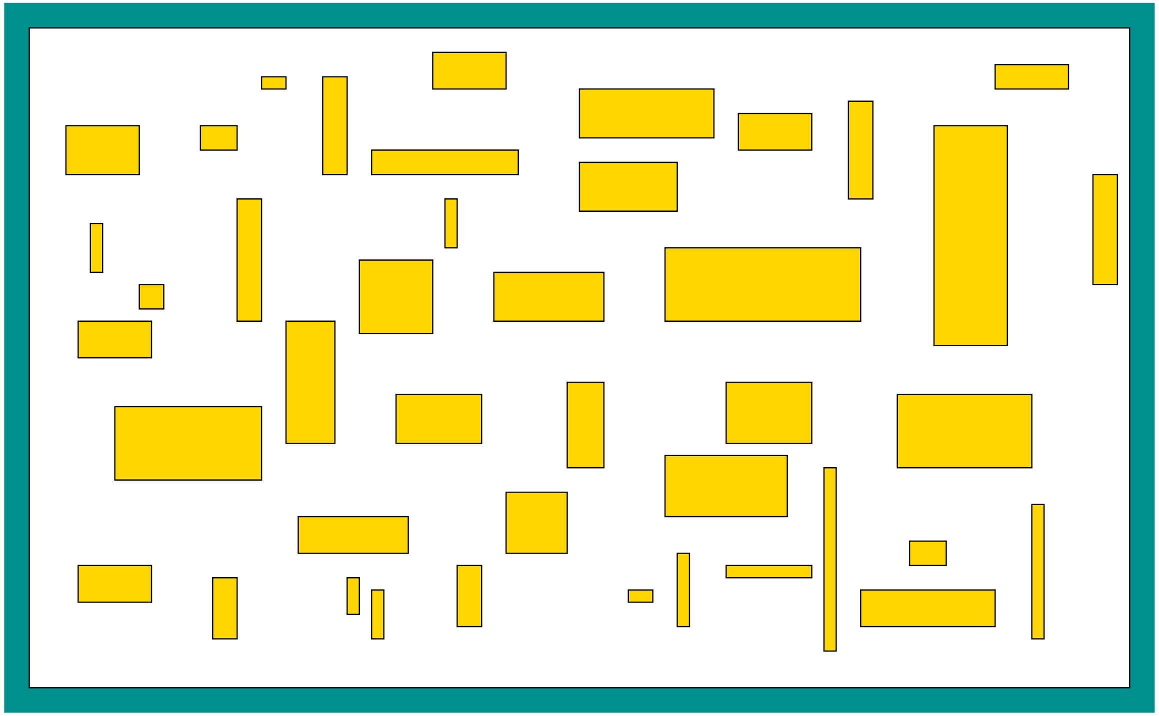


- Solves general motion planning problem.
- Complexity is close to optimal.
- Never implemented?

Combinatorial Motion Planning

Don't be harsh on combinatorial planning methods.

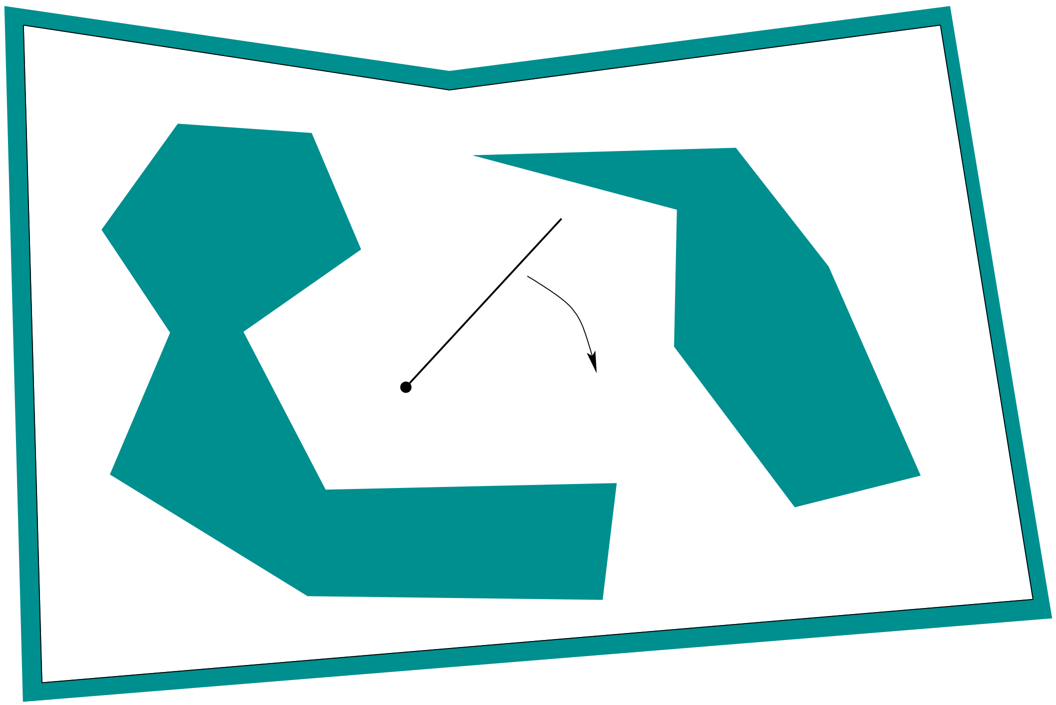
Reif, 1979; Hopcroft, Schwartz, Sharir, 1983: PSPACE-hardness



Even translating a bunch of rectangles inside of a rectangle is PSPACE-hard.

Combinatorial Motion Planning

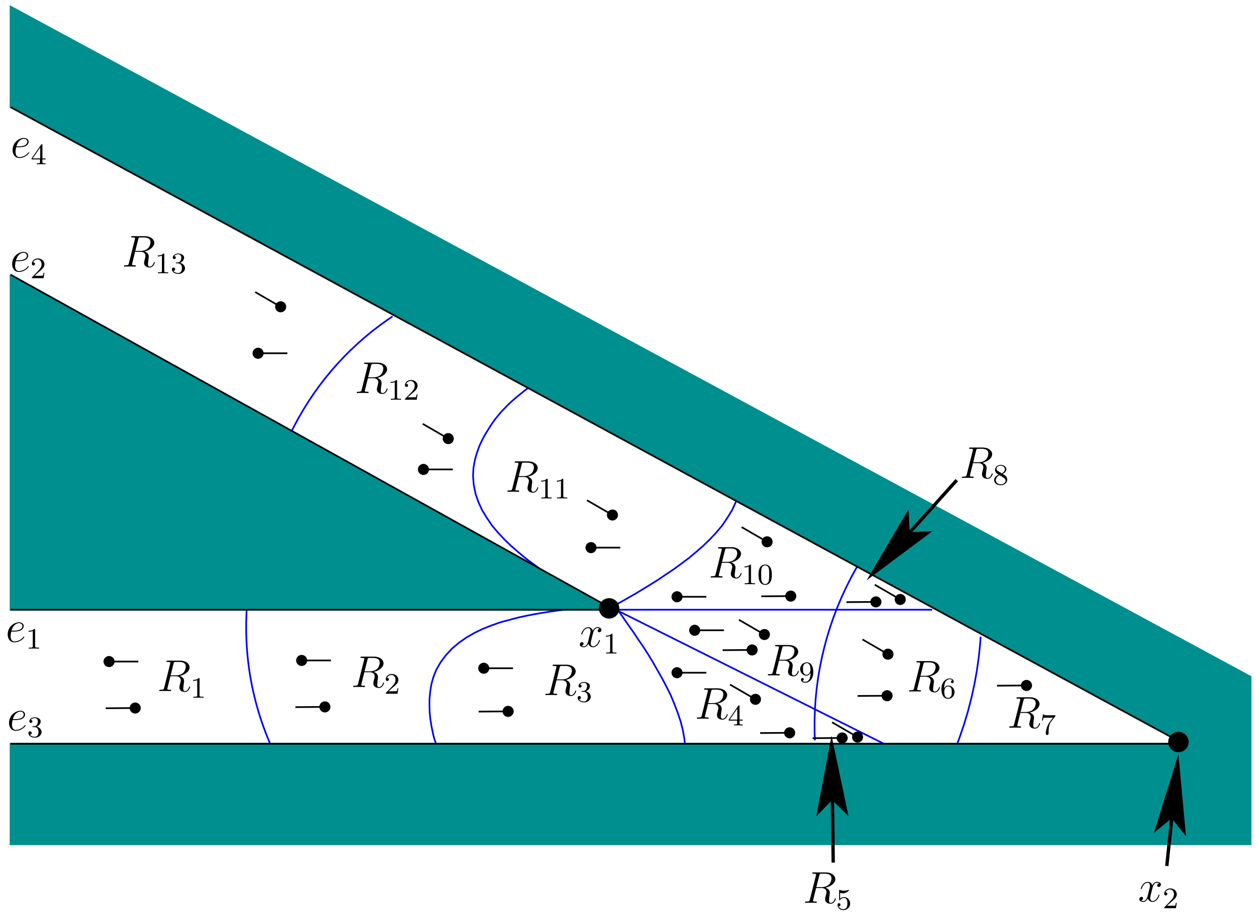
Careful! Some specific problems are easier:



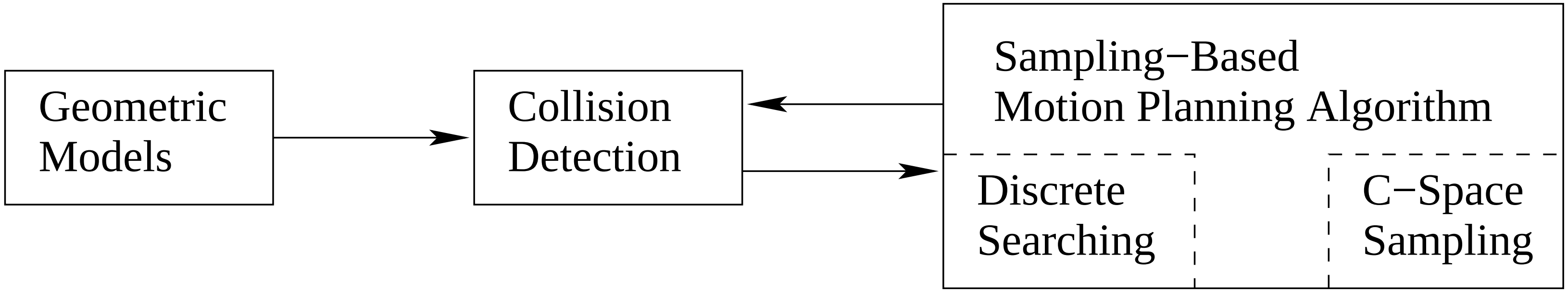
Motion planning for a “ladder”

Levin, Sharir, 1987; Ke, O’Roarke, 1988; Banon, 1990

A nice cell decomposition exists:



Sampling-Based Motion Planning



- Collision detection algorithms enabled a new abstraction.
- Incremental sampling and searching.
- Resolution or probabilistic complete.
- The methods are practical and widely used.

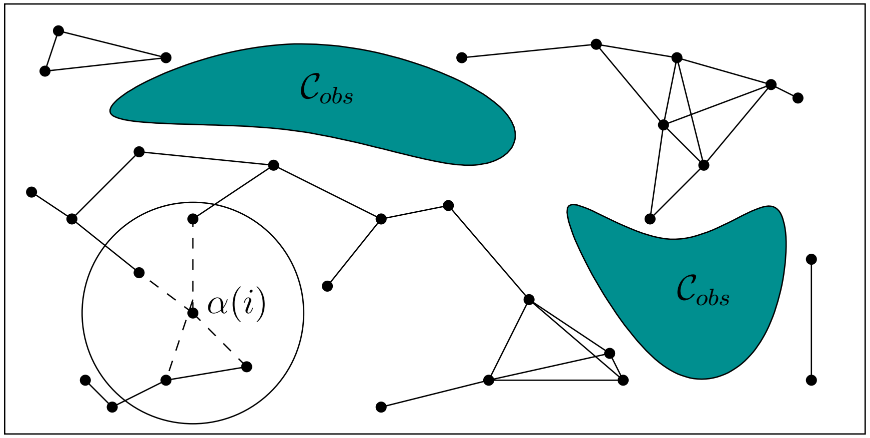
ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Sampling-Based Motion Planning

1984	Donald	grid search with heuristics based on C-constraints
1987	Faverjon, Tournassoud	distance computation, hierarchical CAD models
1989	Paden, Mees, Fisher	uses GTK algorithm, distance comp., 2^d tree
1989	Kondo	grid search, lazy collision checking
1990	Lengyel, Reichert, Donald, Greenberg	search bitmap of C-obstacles
1990	Barraquand, Latombe	randomized potential field, implicit grid
1990	Glavina	sample all of free space, connect with local planner
1992	Chen, Hwang	multiresolution grid search
1992	Mazer, Talbi, Ahuatzin, Bessiere	Ariadne's clew algorithm
1994	Kavraki, Svestka, Overmars, Latombe	Probabilistic Roadmaps (PRMs); multiple query
1997	Hsu, Latombe, Motwani	Expansive planner, single-query, tree search
1999	LaValle, Kuffner	Rapidly-exploring Random Trees (RRTs)

Collision detection: Gilbert, Johnson, Keerthi, 1988; Lin, Canny, 1991; Quinlan, 1994; Gottschalk, Lin and Manocha, 1996; Mirtich, 1997, etc.

Multiple Query: Sampling-Based Roadmaps (PRMs)



- Use sampling to build a roadmap
- Search roadmap for paths
- **PRM-based methods:** PRM (Kavraki, Latombe, Overmars, Svestka, 1994); Obstacle-Based PRM (Amato, Wu, 1996); Sensor-based PRM (Yu, Gupta, 1998); Gaussian PRM (Boor, Overmars, van der Stappen, 1999); Medial axis PRMs (Wilmarth, Amato, Stiller, 1999; Pisula, Hoff, Lin, Manocha, 2000; Kavraki, Guibas, 2000); Contact space PRM (Ji, Xiao, 2000); Closed-chain PRMs (LaValle, Yakey, Kavraki, 1999; Han, Amato 2000); Lazy PRM (Bohlin, Kavraki, 2000); PRM for changing environments (Leven, Hutchinson, 2000); Visibility PRM (Simeon, Laumond, Nissoux, 2000), ...

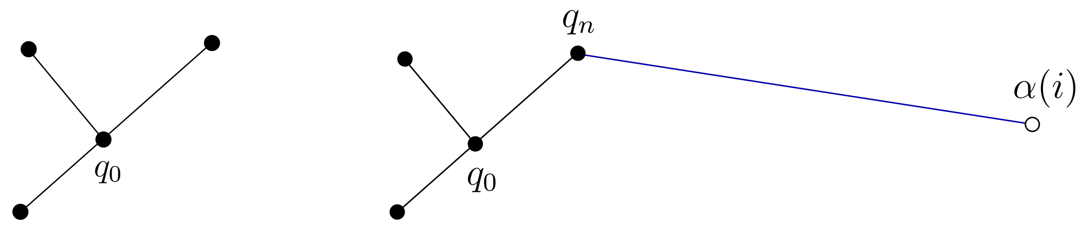
ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Single Query: Search Tree Methods

- Donald, 1984
  - Grid search over 6D C-space
  - Search guided by heuristics based directly on C-constraints
- Barraquand, Latombe, 1990 (randomized potential field)
  - Implicit grid search guided by potential field and random walks
  - Direct use of collision detector to validate motions
- Mazer, Talbi, Ahuatzin, Bessiere, 1992 (Ariadne's clew)
  - Search trees based on self avoidance
  - Node placement obtained by genetic algorithm
- Hsu, Latombe, Motwani, 1997 (expansive space planner)
  - Also based on self avoidance
  - Node placement biased toward low-density regions
- LaValle, Kuffner, 1999 (Rapidly-exploring Random Trees - RRTs)
  - Search tree based on Voronoi bias
  - Growth obtained by sampling and nearest-neighbor searching

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Rapidly Exploring Random Trees (RRTs)



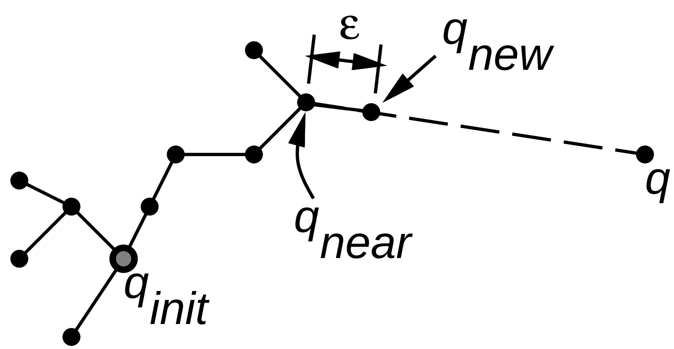
- Introduced by LaValle and Kuffner, 1999.
- Applied, adapted, and extended in many works: Frazzoli, Dahleh, Feron, 2000; Toussaint, Basar, Bullo, 2000; Vallejo, Jones, Amato, 2000; Strady, Laumond, 2000; Mayeux, Simeon, 2000; Karatas, Bullo, 2001; Li, Chang, 2001; Kuffner, Nishiwaki, Kagami, Inaba, Inoue, 2000, 2001; Williams, Kim, Hofbaur, How, Kennell, Loy, Ragno, Stedl, Walcott, 2001; Carpin, Pagello, 2002; ...
- Also, applications to biology, computational geography, verification, virtual prototyping, architecture, solar sailing, computer graphics, ...
- In IEEE ICRA 2011 Proceedings, “RRT” occurs 928 times.

The RRT Construction Algorithm

BUILD_RRT(q_{init})

$\mathbb{T}$.init(q_{init});
for $k = 1$ **to** K **do**
    $q_{rand} \leftarrow$ RANDOM_CONFIG$()$;
    EXTEND$(\mathbb{T}, q_{rand})$;

EXTEND$(\mathbb{T}, q_{rand})$

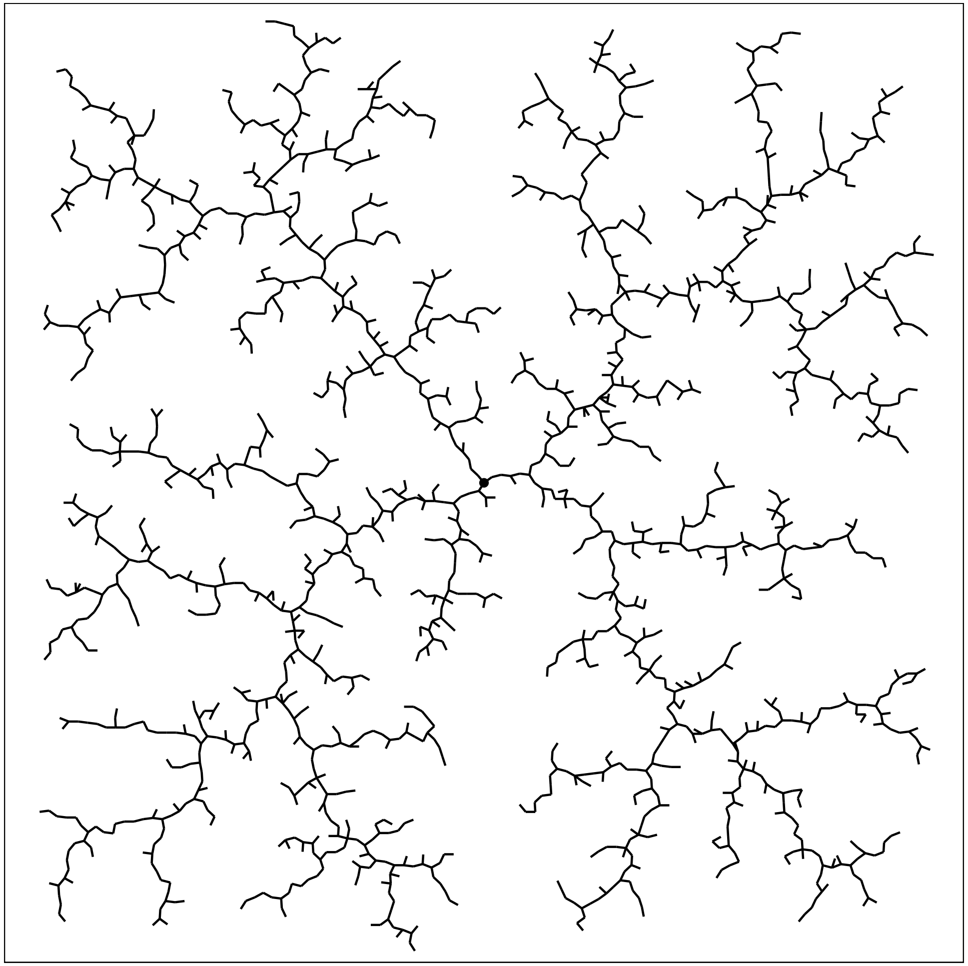


Metric on $\mathcal{C}$: $\rho : \mathcal{C} \times \mathcal{C} \rightarrow [0, \infty)$

Nearest neighbors: Yershova [Atramentov], LaValle, 2002; Arya, Mount, 1997

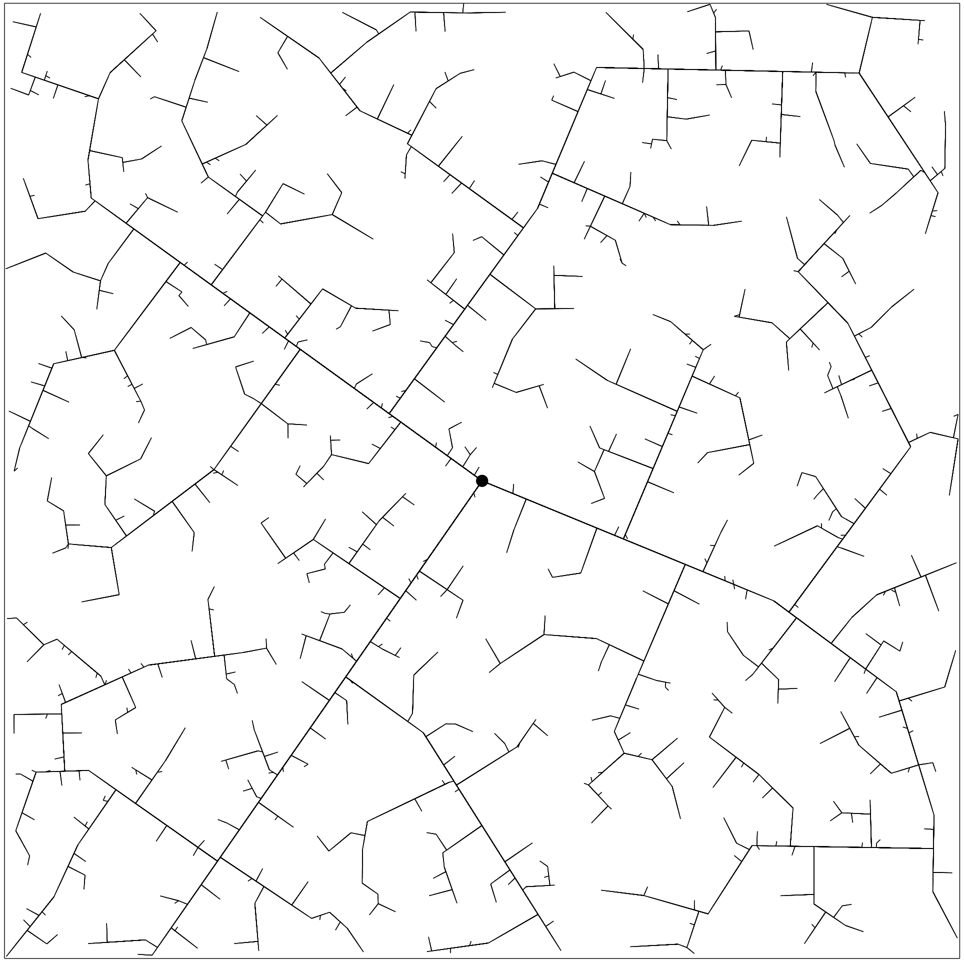
Incremental collision detection: Lin, Canny, 1991; Mirtich, 1997

A Rapidly-Exploring Random Tree (RRT)

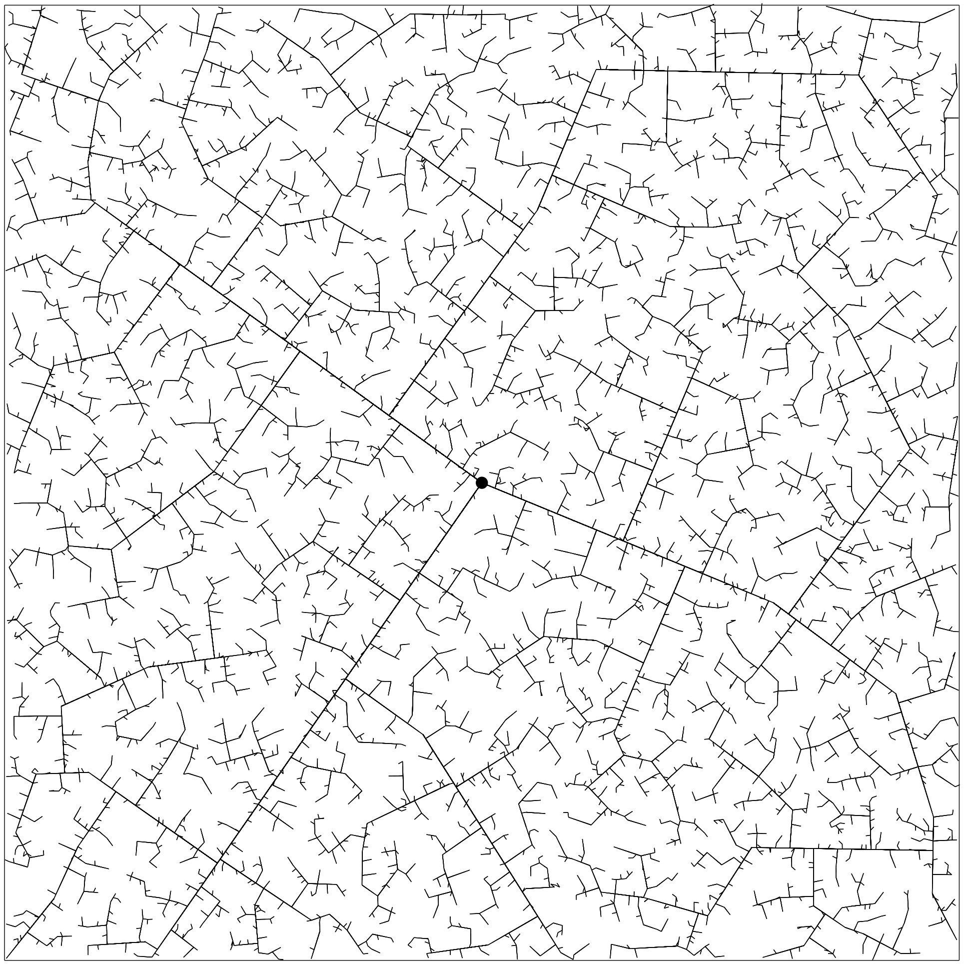


Segment-Based RRT

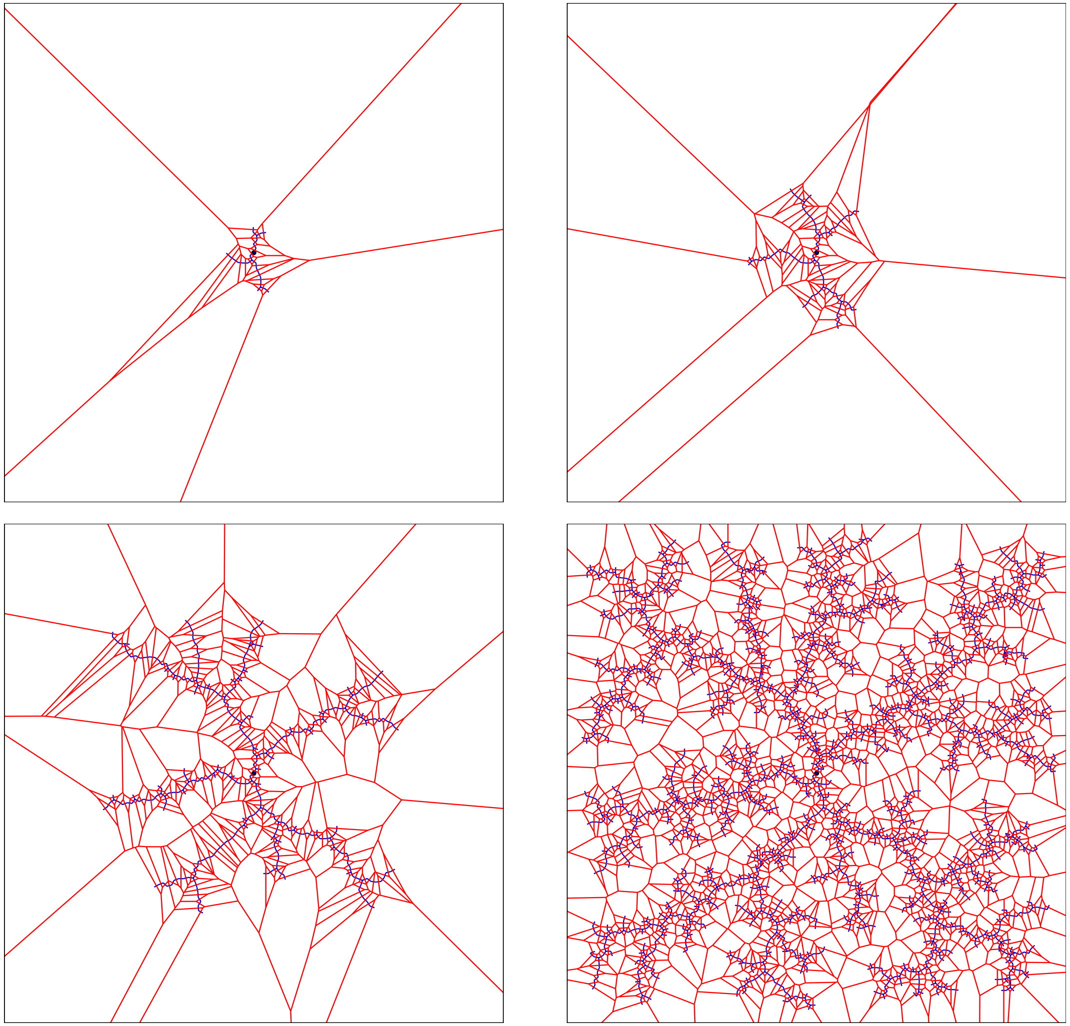
Segment-Based RRT



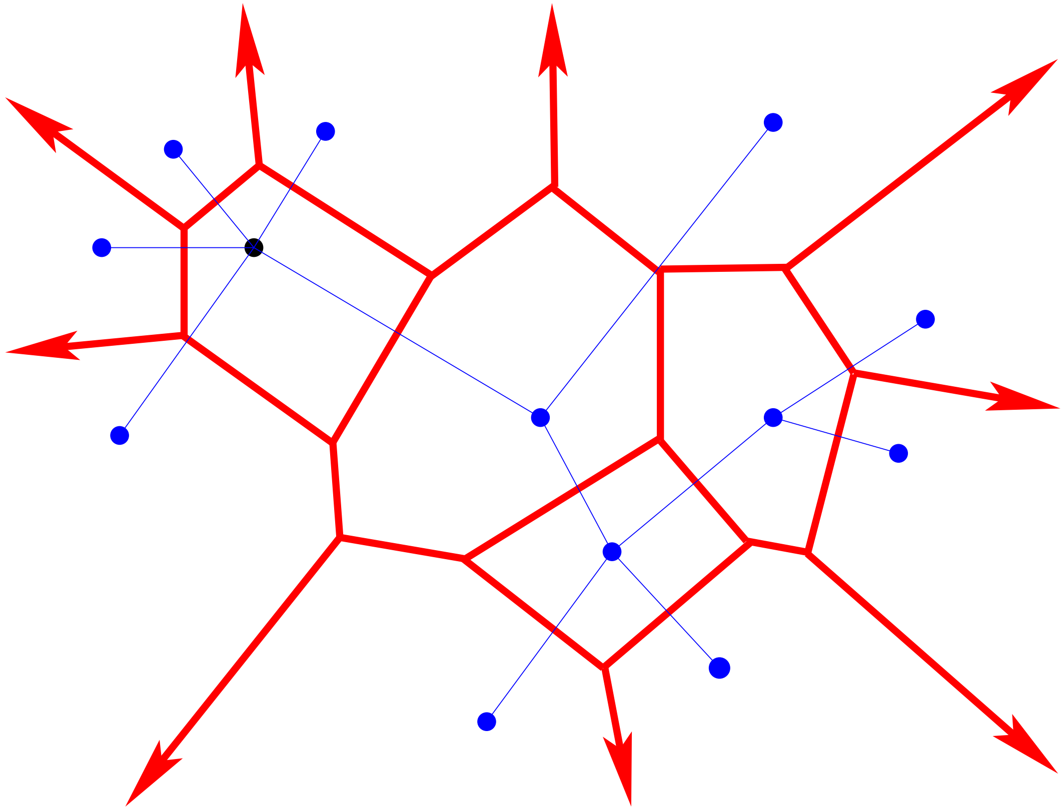
Segment-Based RRT



Voronoi-Biased Exploration

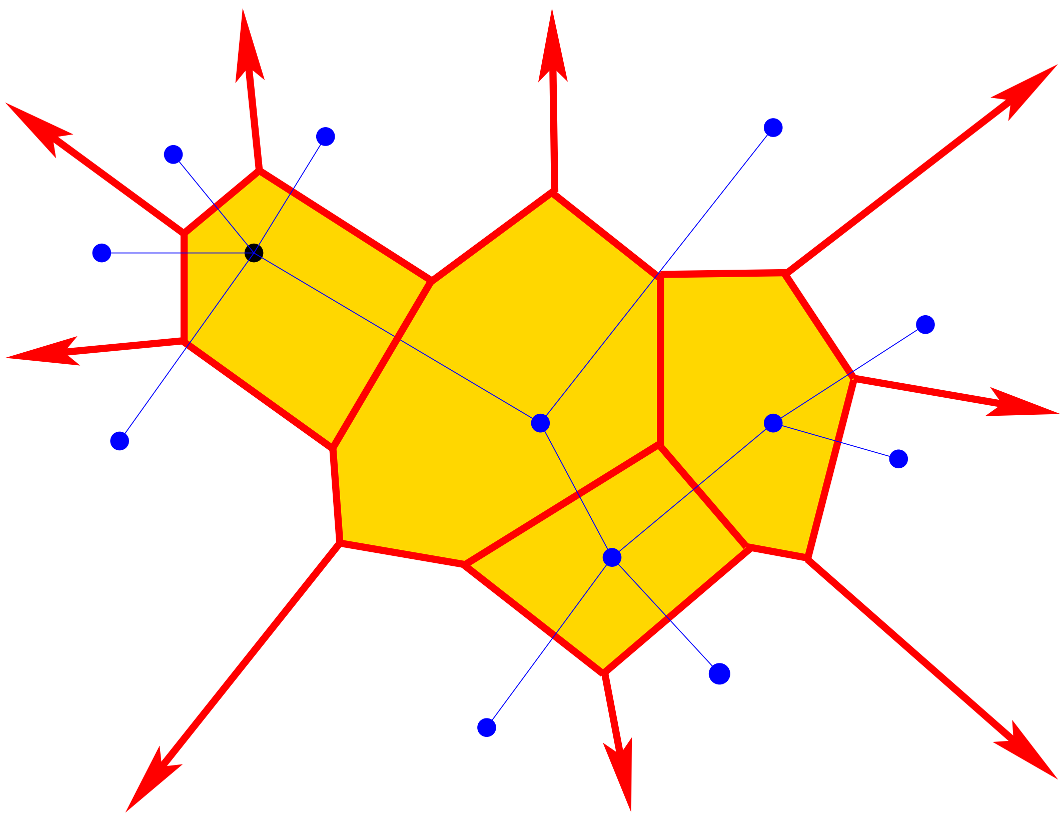


Voronoi Diagram in $\mathbb{R}^2$

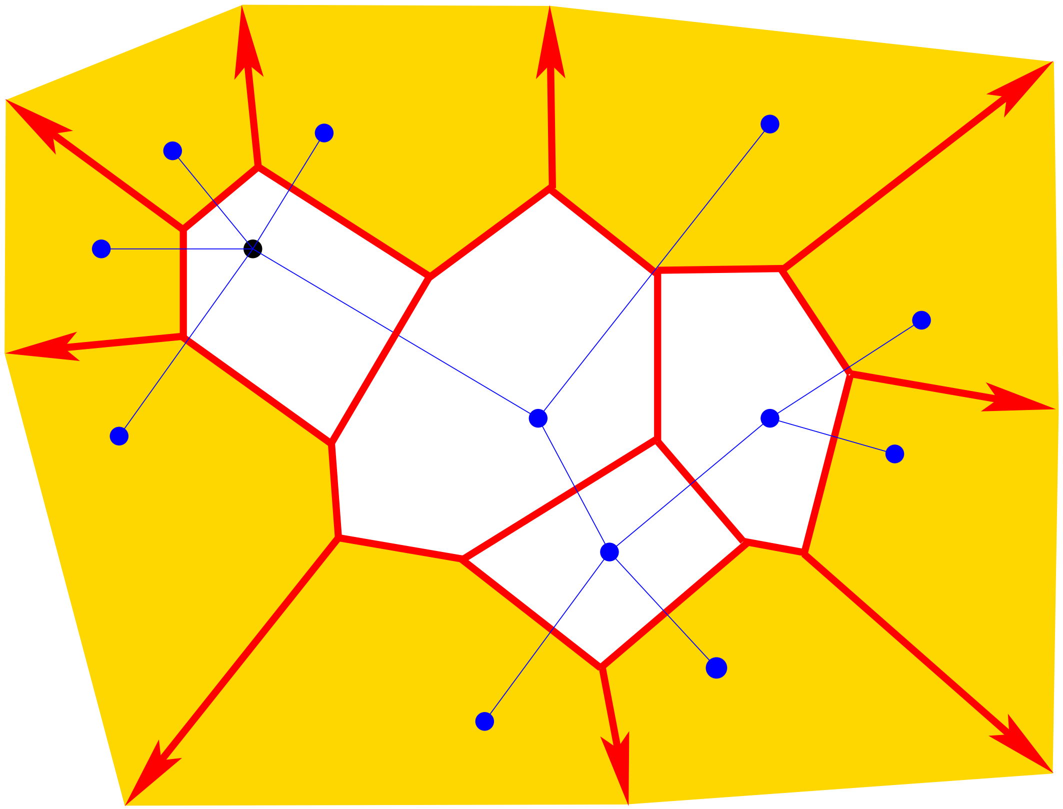


Think about: Where is the nearest subway station?

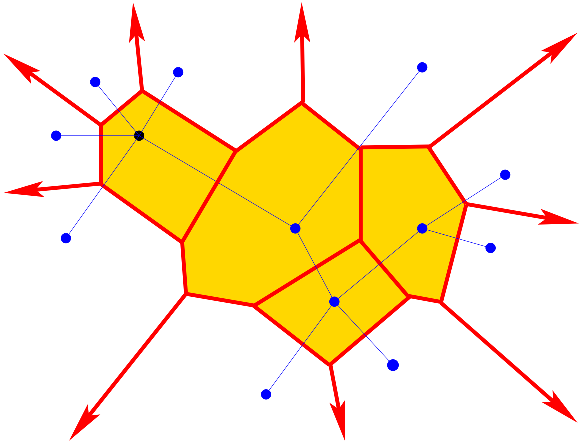
Voronoi Diagram in $\mathbb{R}^2$



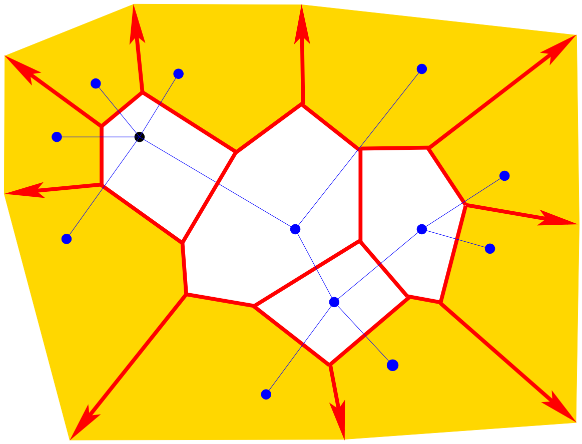
Voronoi Diagram in $\mathbb{R}^2$



Refinement vs. Expansion



Refinement

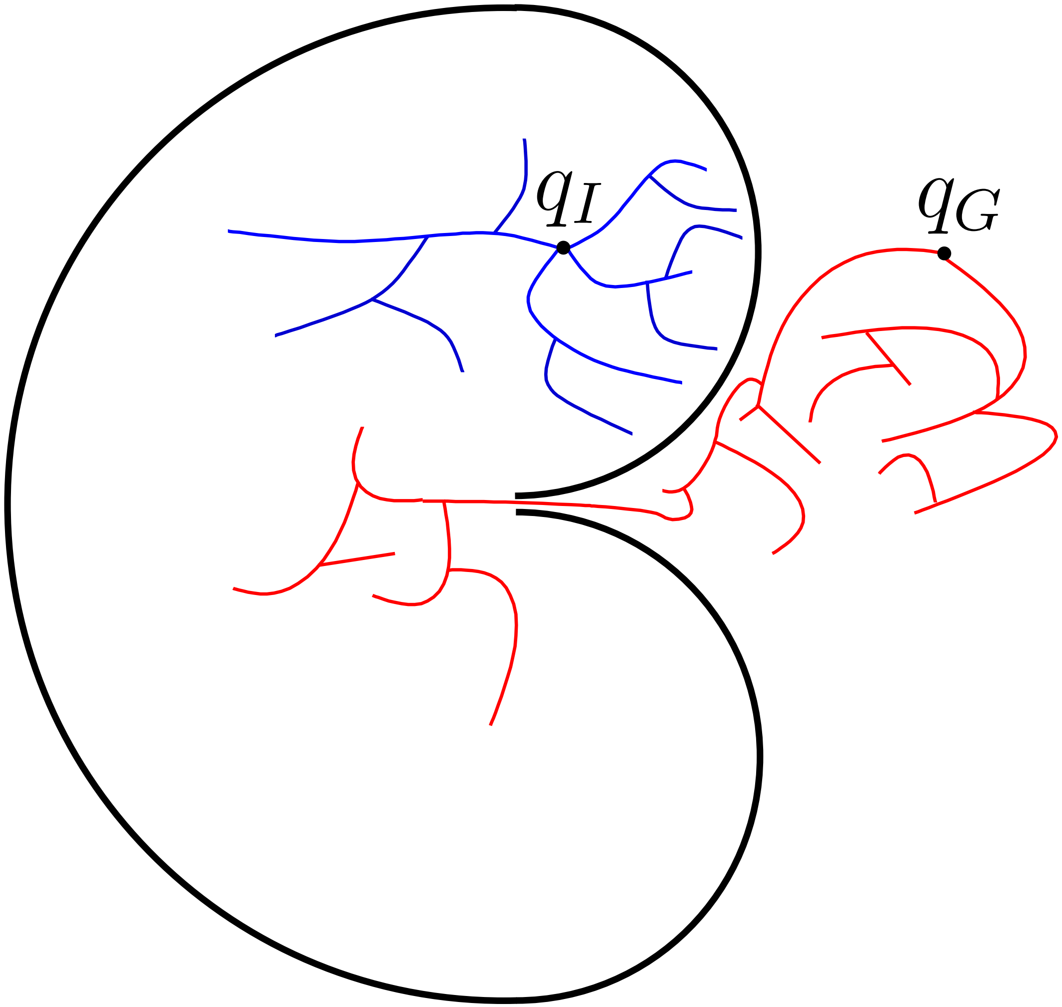


Expansion

Where will the random sample fall?

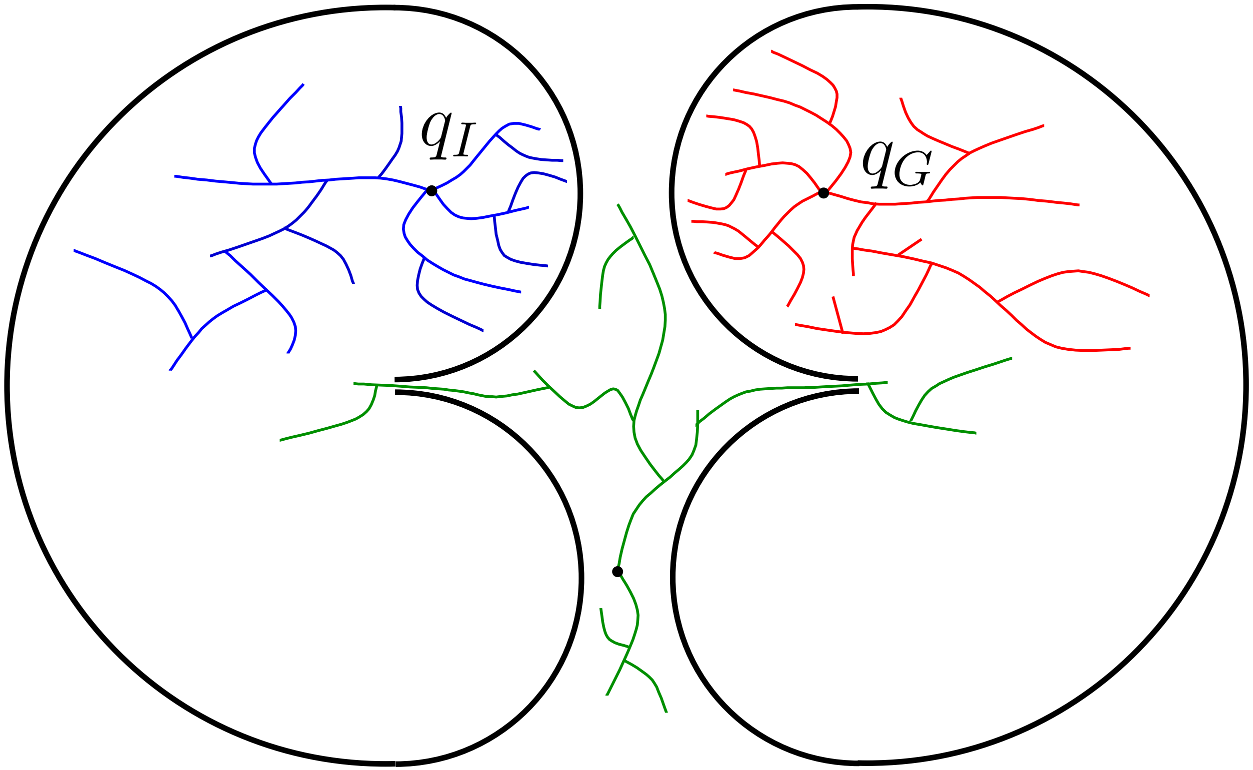
Bidirectional Search

Grow two RRTs, one from the initial and one from the goal.

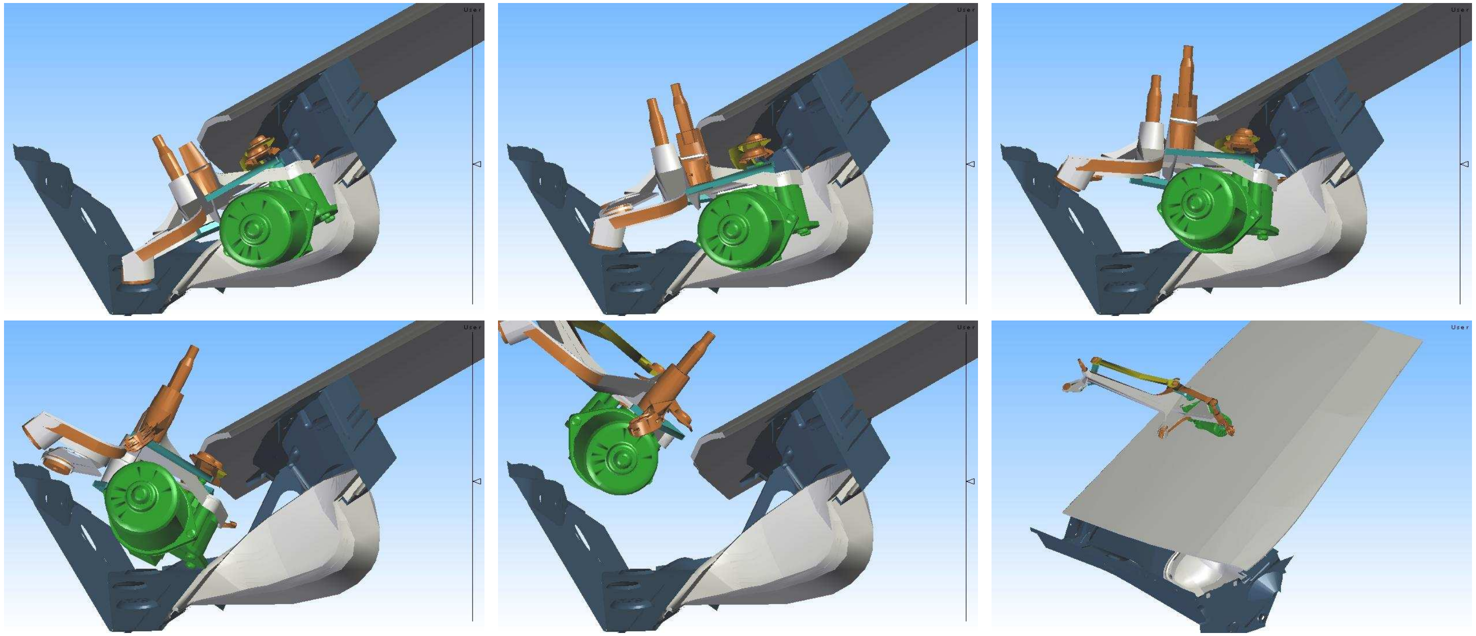


Spend some effort trying to connect them to each other.

Does not always work well...



Wiper Motor Assembly



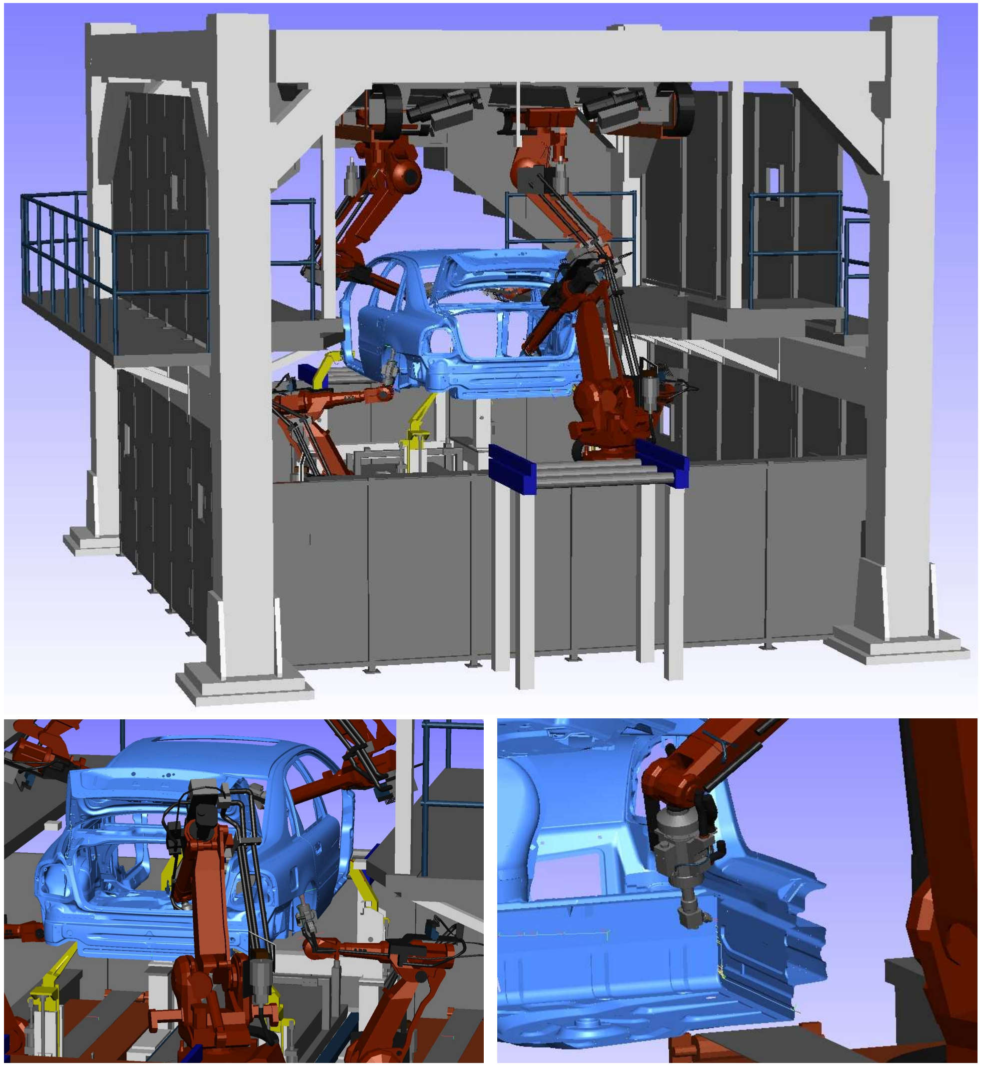
Kineo CAM and LAAS/CNRS, Toulouse, France

Integrated into Robcad (eM-Workplace)

Add-ons for 3D Studio Max, Solidworks

Direct users: Renault, Airbus, Ford, Optivus, ...

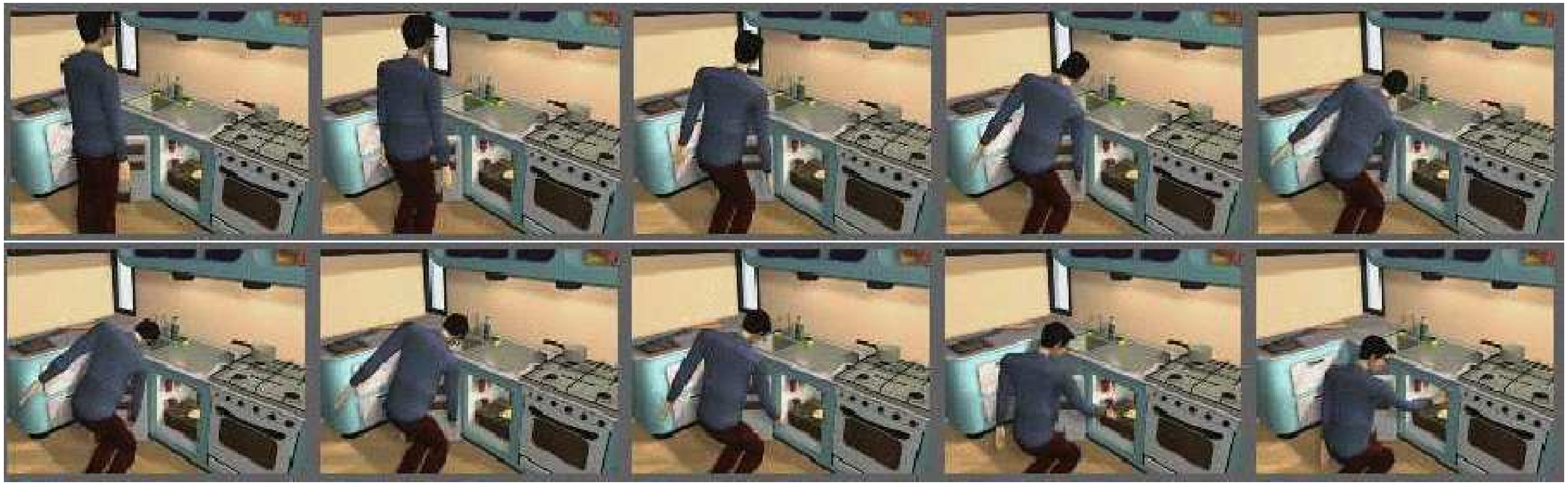
Sealing Cracks at Volvo Cars



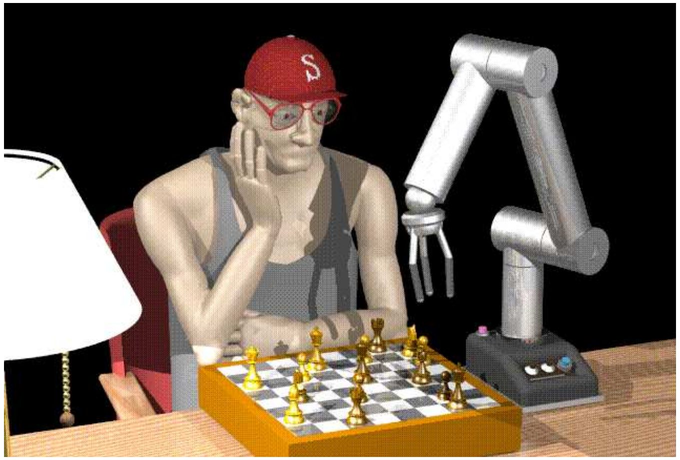
Fraunhofer Chalmers Centre and Volvo Cars, Sweden

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Virtual Humans

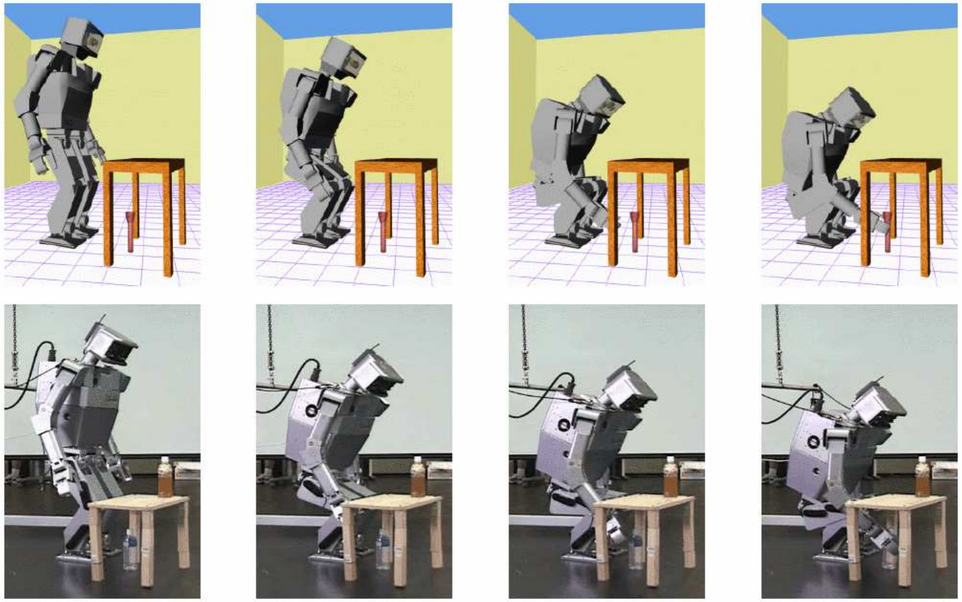


Marcelo Kallman, UC Merced



James Kuffner, CMU

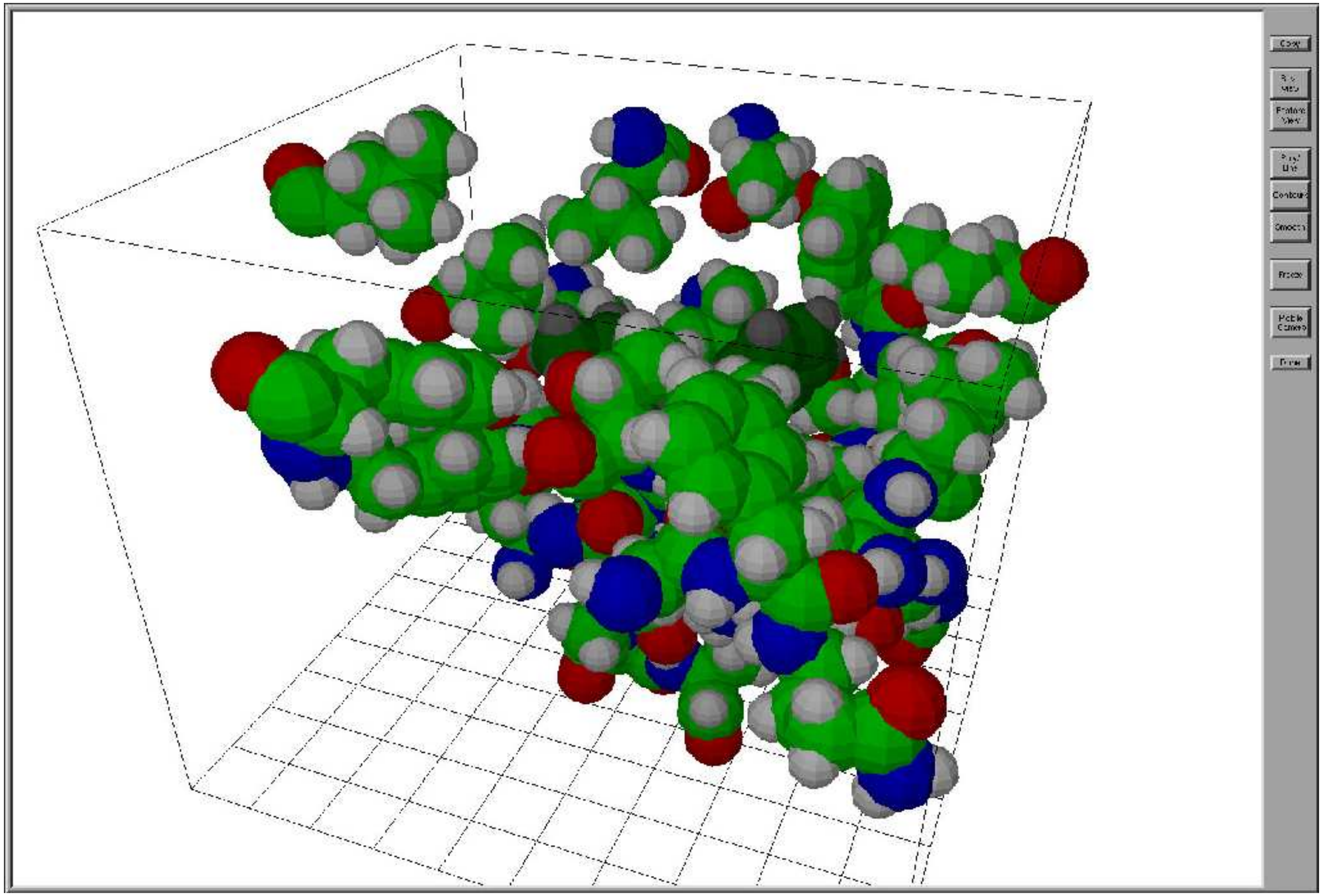
Kagami and H7



Planning

University of Tokyo and AIST

Molecules, Etc.



From Nic Simeon, LAAS/CNRS

Beyond Basic Path Planning

- Need to broaden the focus of planning.
- Better unification of ideas.
- Need to address important concerns: feedback, differential constraints, sensing, uncertainty.

Fundamental: Information comes from sensors, not the Turing tape.

- **Control theory:** analytical, continuous, differential, feedback, optimality
- **Motion planning:** algorithmic, continuous, paths, feasibility
- **AI planning:** algorithms, discrete, logic, feasibility

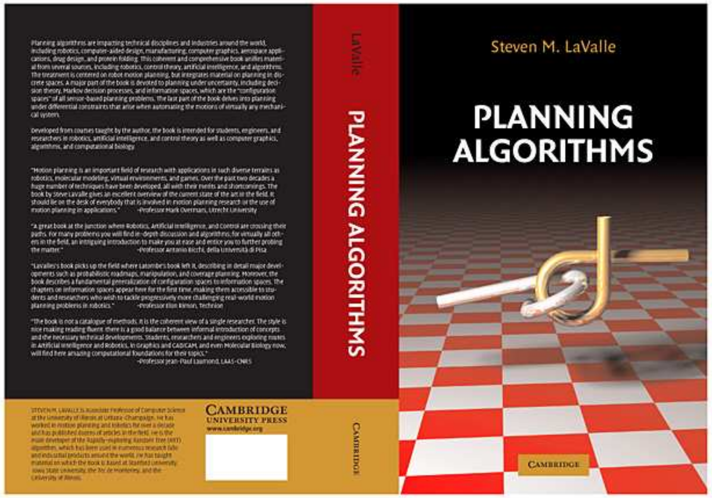
	Continuous	Discrete	Analytical	Algorithmic	Differential	Optimality	Feasibility	Feedback	Uncertainty
Control Theory	●	○	●	○	●	●	○	●	○
Motion Planning	●	○		●	○		●	○	○
AI Planning	○	●		●		○	●	○	○

These days, people are interested in the same issues.

Is it planning algorithms? Algorithmic control theory?

Three Main Places for Prospecting

After painfully putting together a landscape of literature, several enormous holes were visible.



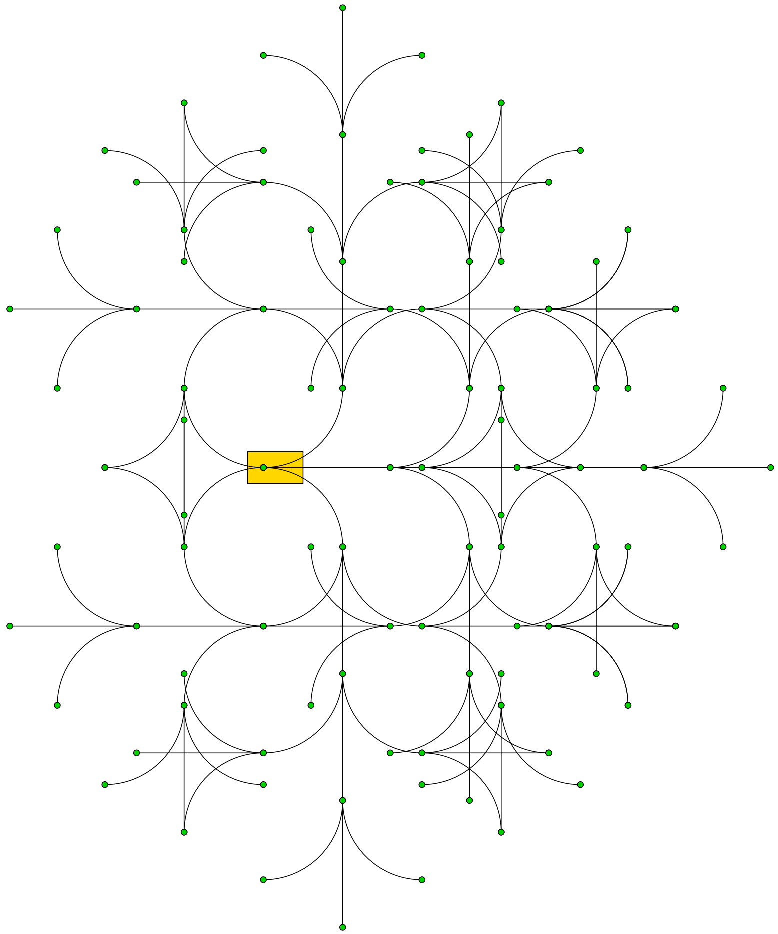
The hardest chapters to write:

- Motion planning under differential constraints (Ch 14)
- Feedback motion planning (Ch 8)
- Information spaces and sensing (Ch 11,12)

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Motion Planning with Differential Constraints

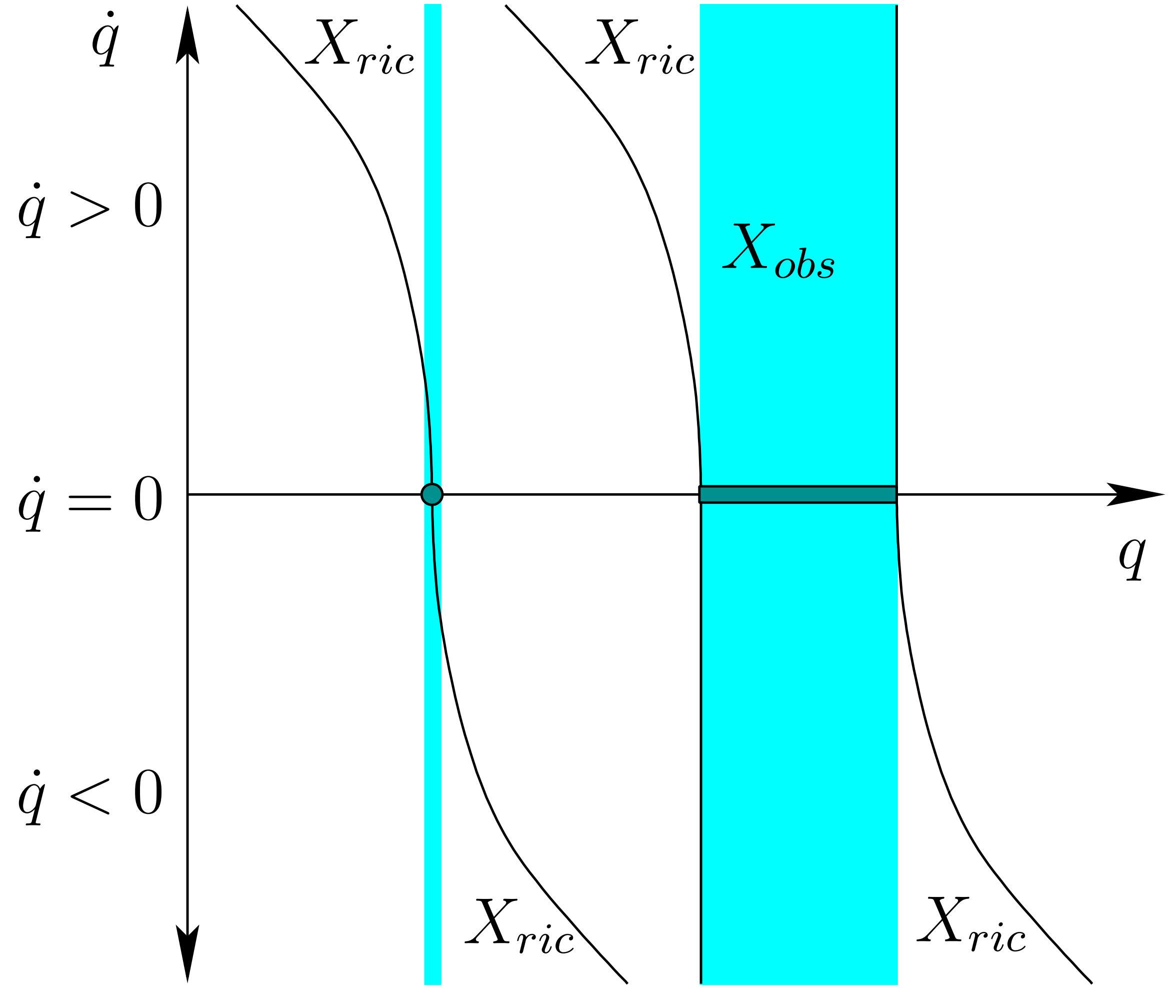
Reachable sets



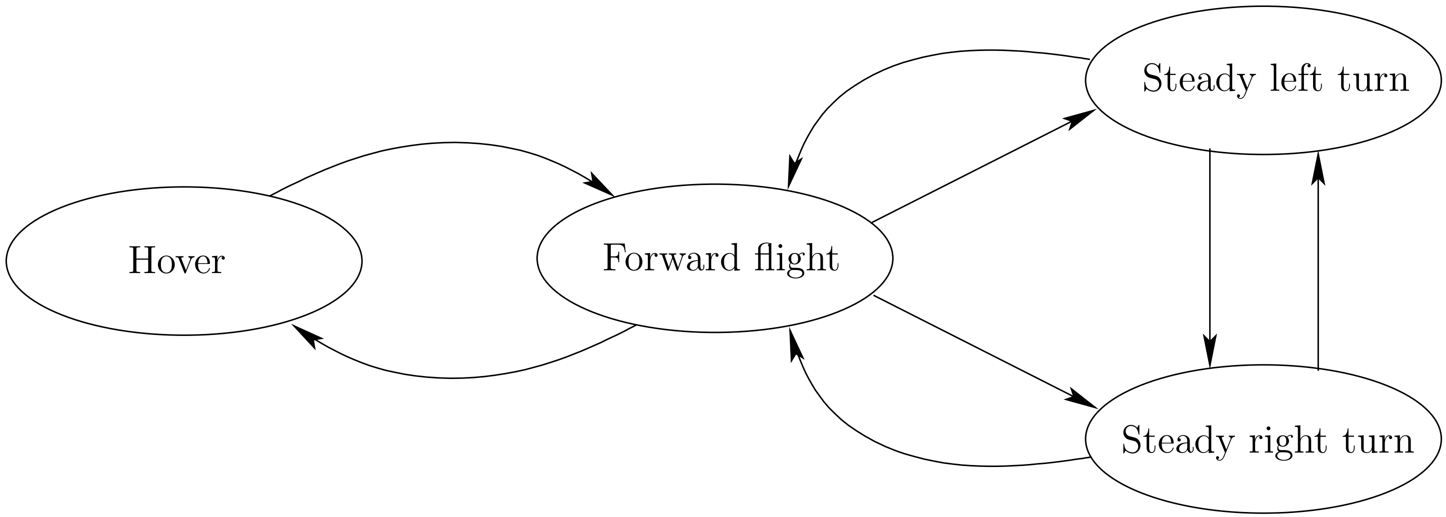
Dubins (forward-only) car example

Motion Planning with Differential Constraints

Obstacles in the phase space grow with speed.

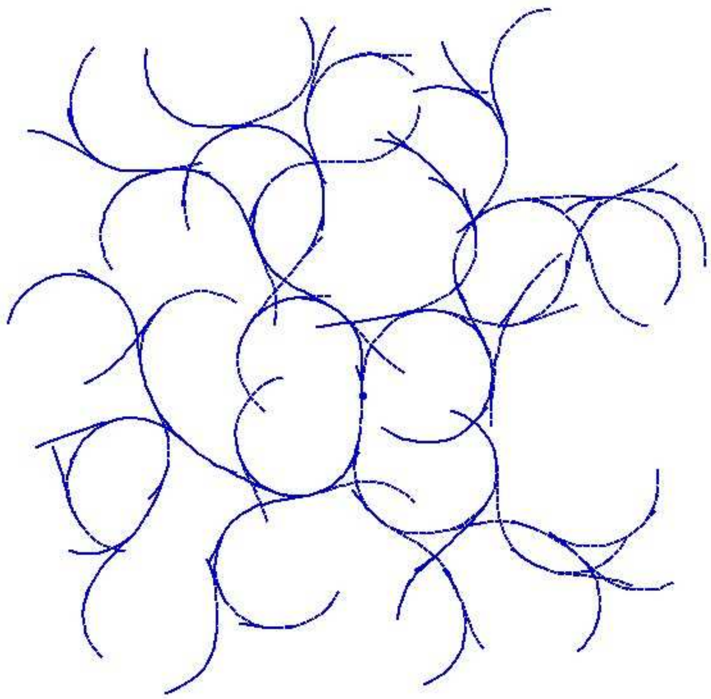


Motion Planning with Differential Constraints

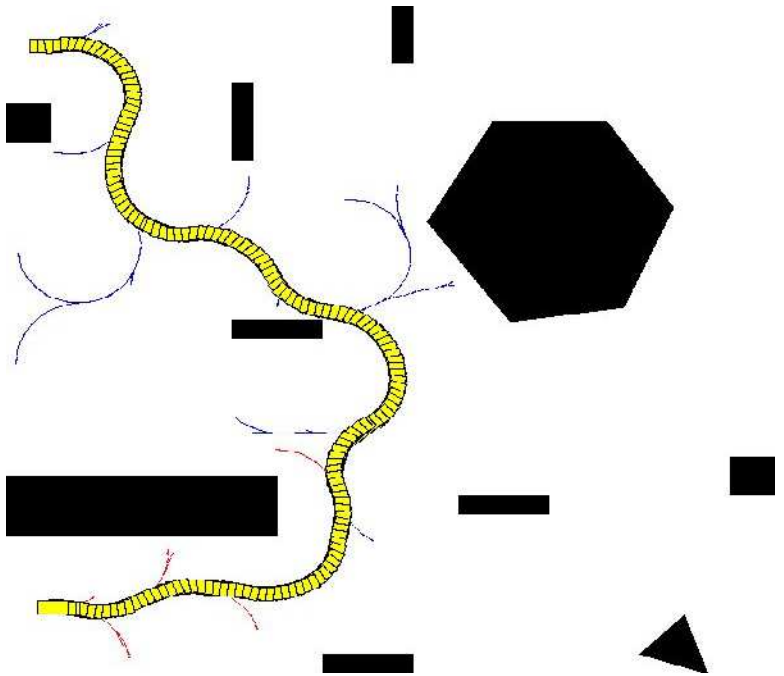


Motion primitive problem (Frazzoli, Egerstedt, Pappas, Murphey, Belta)

Adding Differential Constraints to RRTs

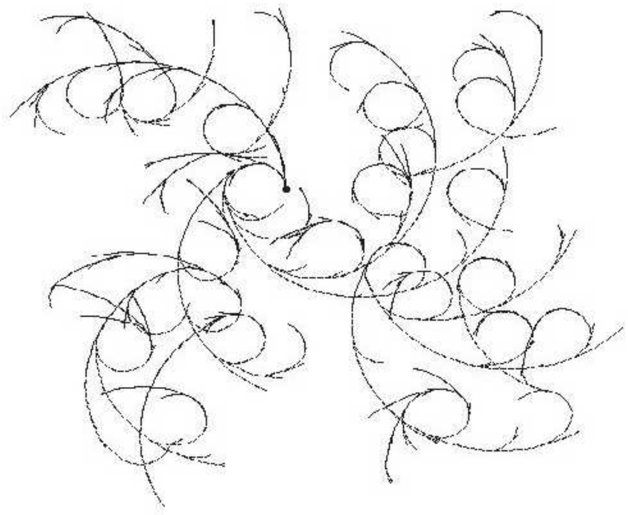


Smooth RRT

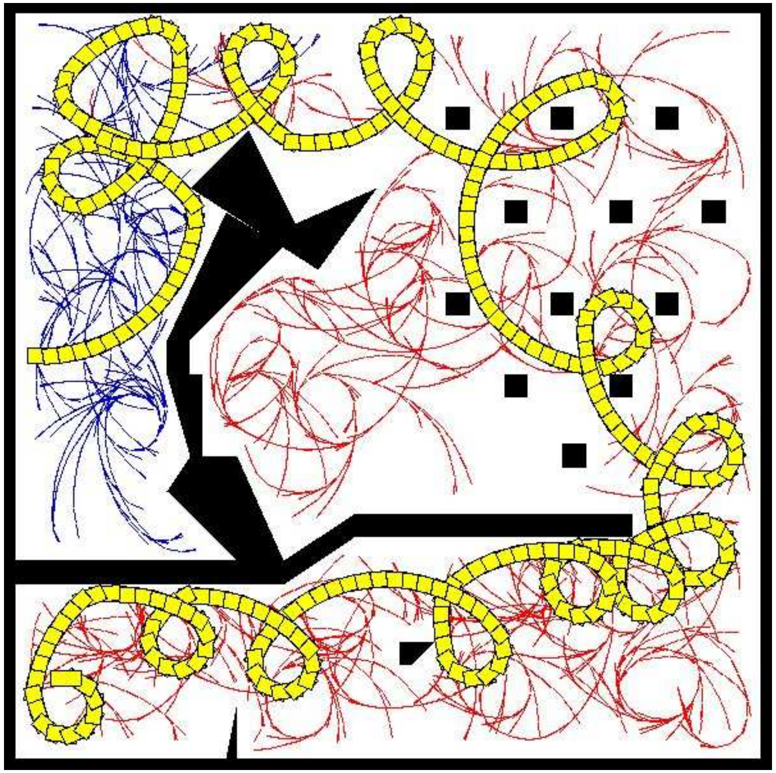


Bidirectional Search

The Left-Forward-Only Car



Left-Forward RRT



Bidirectional Search

Feedback Motion Planning

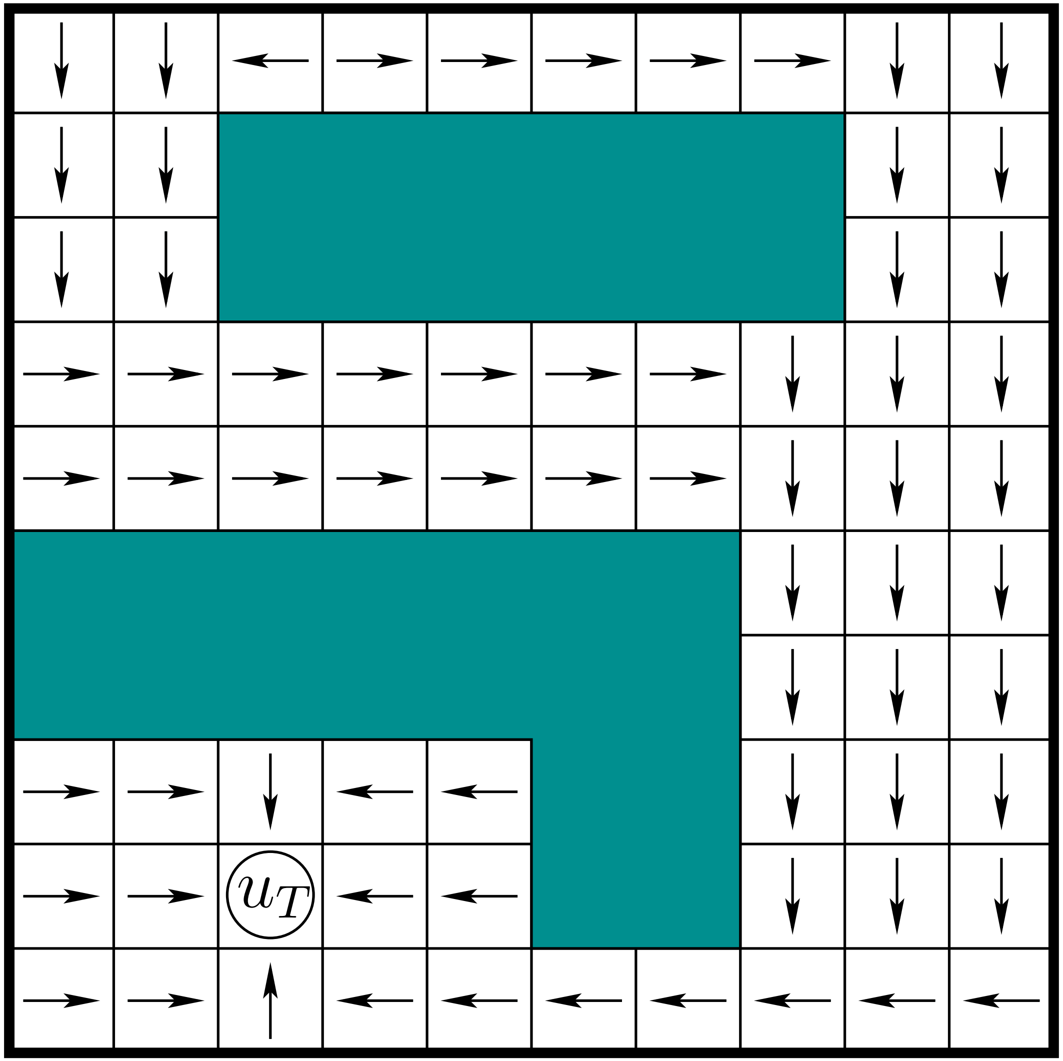
Future states (or configurations) are not necessarily predictable.

Need to compute a feedback plan $\pi : X \rightarrow U$
Here, U is a set of actions or system inputs.
Might have a control system: $\dot{x} = f(x(t), u(t))$

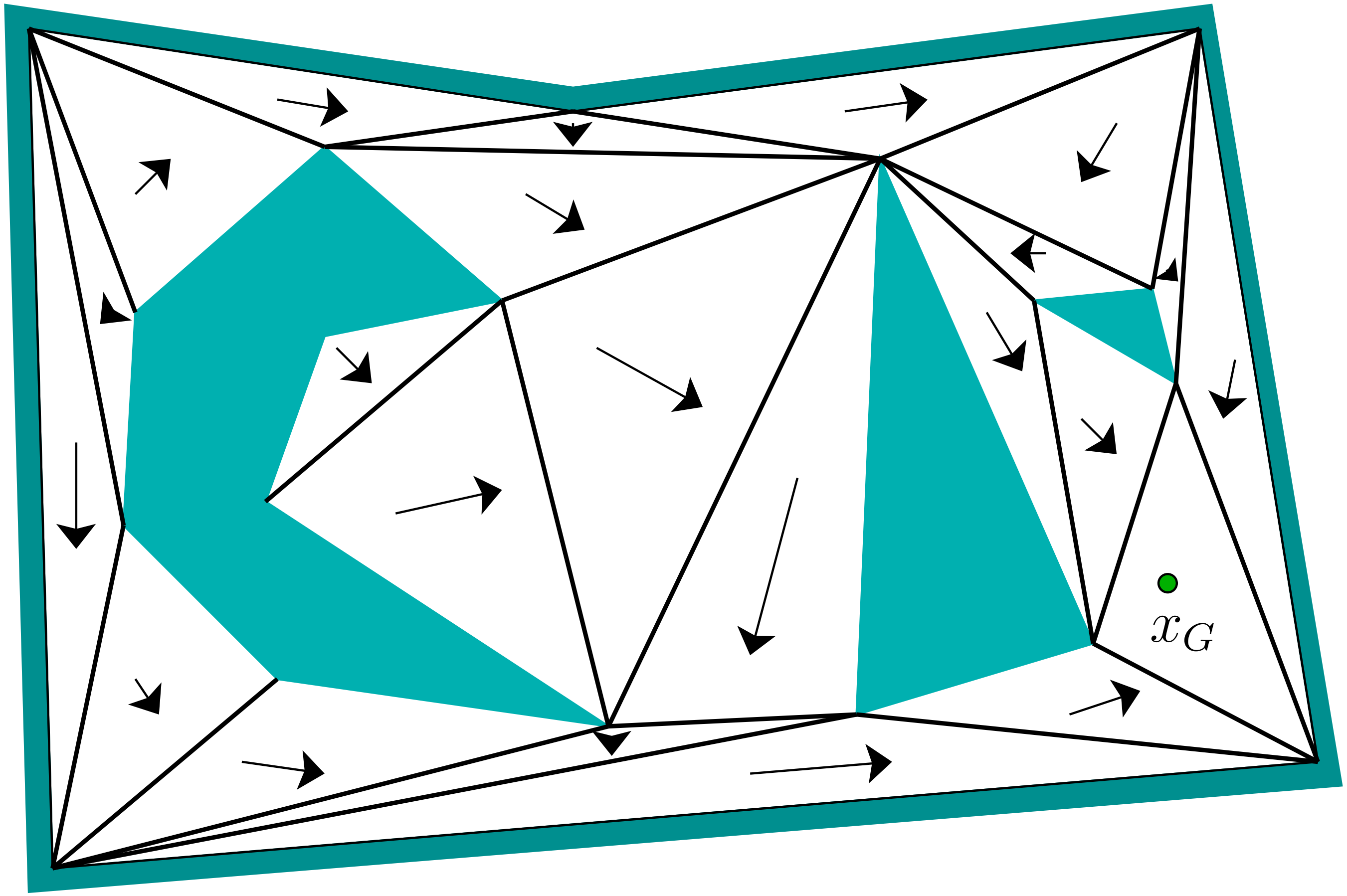
During execution, a sample path is generated.

Note: Powerful sensing is assumed because the state $x \in X$ is known at all times!

A discrete grid example:

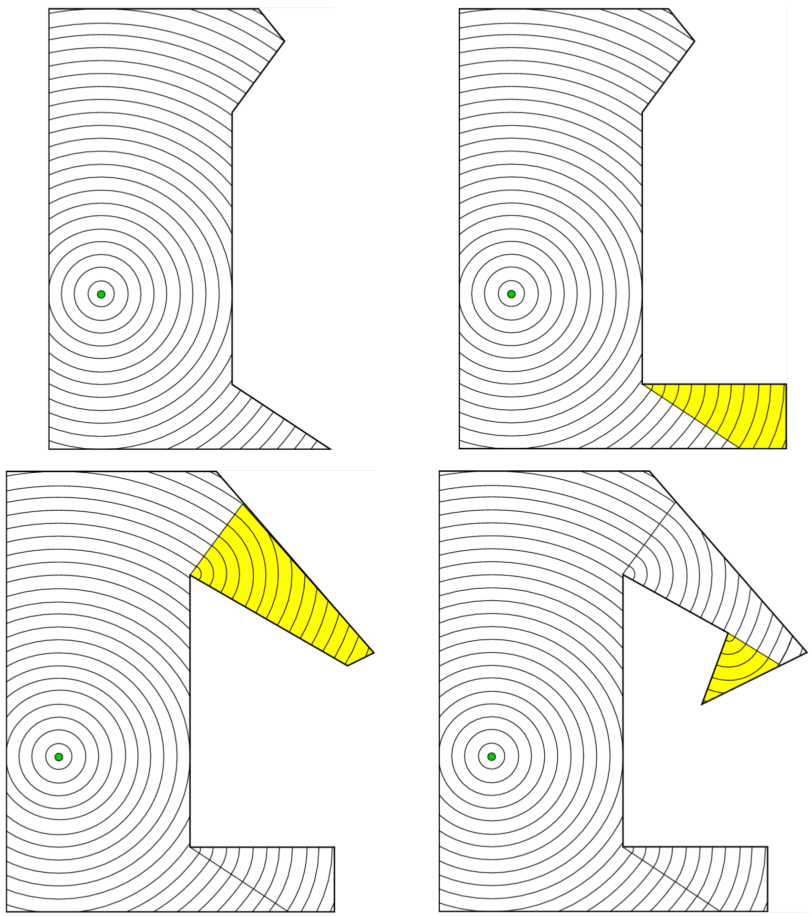


A nonsmooth continuous example:



Feedback Motion Planning

Navigation function:



Arrive in the goal by gradient descent.

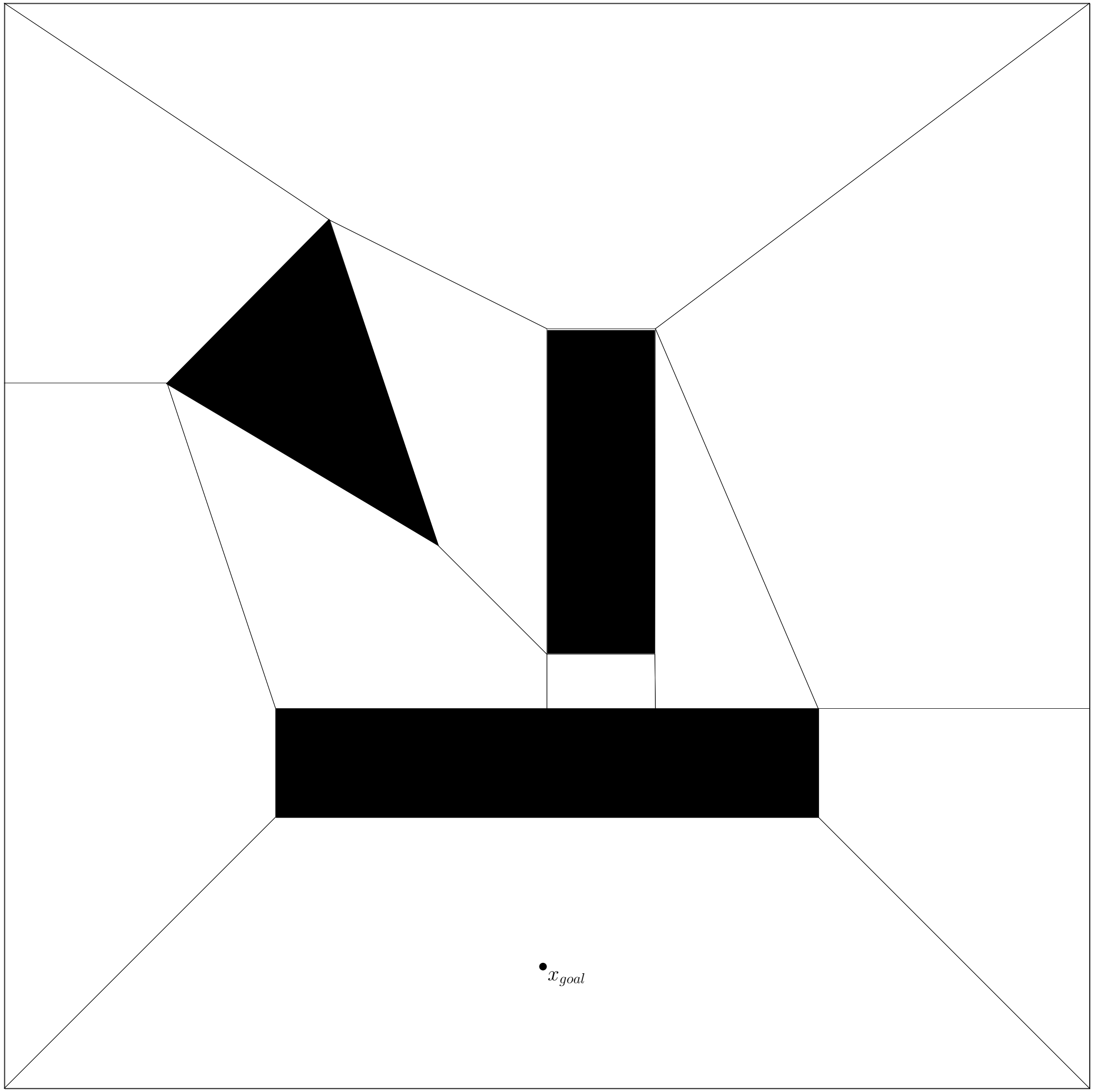
Our Approach

Lindemann, LaValle, IJRR 2009.

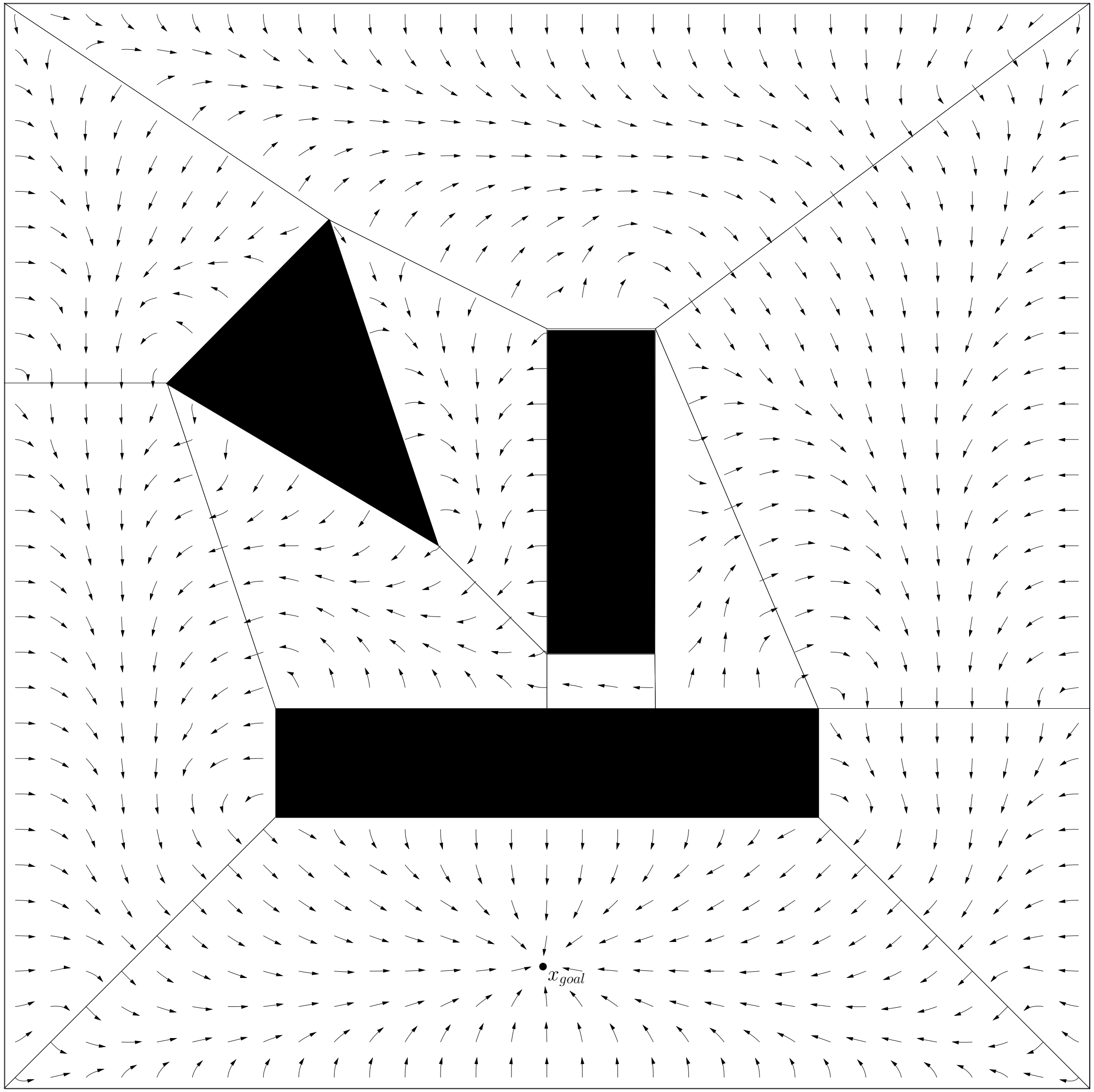
Instead of using the gradient of a navigation function as the vector field, we construct one directly. We do this as follows:

- Partition the space into simple cells.
- Use the cell connectivity graph to determine a high-level motion plan.
- Define local vector fields on each cell which are compatible with the motion plan.
- Appropriately blend the vector fields together to obtain a global vector field.

Decomposition



Computing Smooth Flows



I-Spaces: The Next Generation of C-Spaces

Problem! It may be expensive or impossible to accurately estimate $\tilde{x}(t)$ at all times.

Remember the previous parts: Start with the *task* and design the sensors and filters.
Planning naturally occurs in the resulting *information space*.

Maybe we need to develop:

- Formulations of sensor models, I-spaces
- Models of complexity, computation over I-spaces
- Sampling-based planning methods
- Combinatorial planning methods

For C-spaces, the early steps were already done (Lagrangian mechanics).