Motion Planning

METR 4202: **Robotics** & Automation

Dr Surya Singh -- Lecture # 10

October 5, 2016

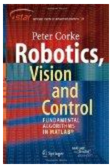
metr4202@itee.uq.edu.au

http://robotics.itee.uq.edu.au/~metr4202/ [http://**metr4202.com**]

Schedule of Events

Week	Date	Lecture (W: 12:05-1:50, 50-N202)
1	27-Jul	Introduction
2	3-Aug	Representing Position & Orientation & State (Frames, Transformation Matrices & Affine Transformations)
3	10-Aug	Robot Kinematics Review (& *Ekka Day*)
4	17-Aug	Robot Inverse Kinematics & Kinetics
5	24-Aug	Robot Dynamics (Jacobeans)
6	31-Aug	Robot Sensing: Perception & Linear Observers
7	7-Sep	Robot Sensing: Single View Geometry & Lines
8	14-Sep	Robot Sensing: Feature Detection
9	21-Sep	Robot Sensing: Multiple View Geometry
	28-Sep	*Study break*
10	**5-Oct**	**Motion Planning**
11	12-Oct	Probabilistic Robotics: Localization & SLAM
12	19-Oct	Probabilistic Robotics: Planning & Control
13	26-Oct	State-Space Automation (Shaping the Dynamic Response/LQR) + Course Review

Follow Along Reading:



Robotics, Vision & Control
by Peter Corke

Also online: SpringerLink

UQ Library eBook:
364220144X

Today

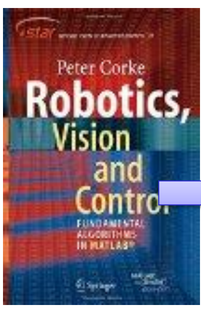
➔Planning & Control 🡐

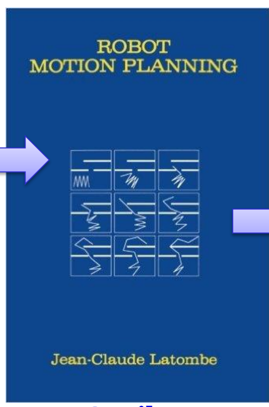
- Planning (**Global** Motion)
  - pp. 91-103
    (Yup! That's all Peter Corke has to say! Yet there is a Chapter [15] on Visual Servoing, a **local** motion method that can't handle obstacles).

- SLAM
  - pp. 123-4
    (§6.4-6.5)

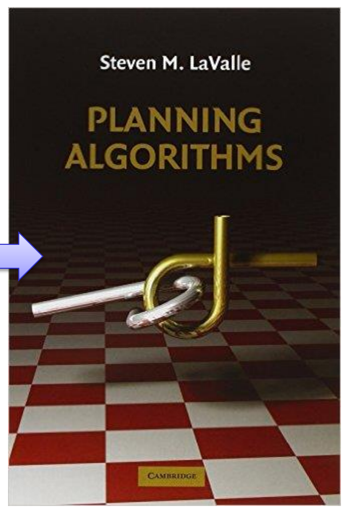
Next Time

Reference Material





UQ Library
(TJ211.4 .L38 1991)



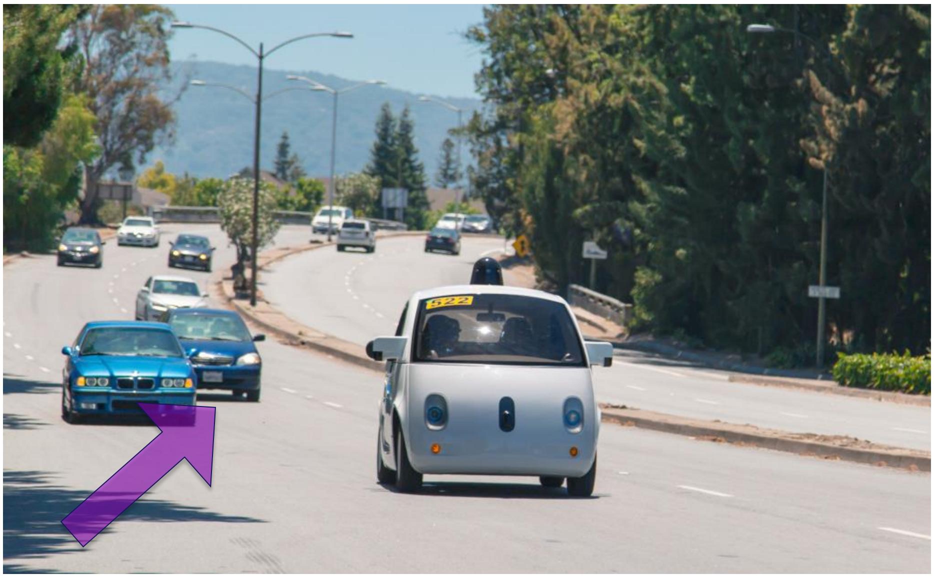
UQ Library / Online (PDF)

(Kinematic) Motion Planning

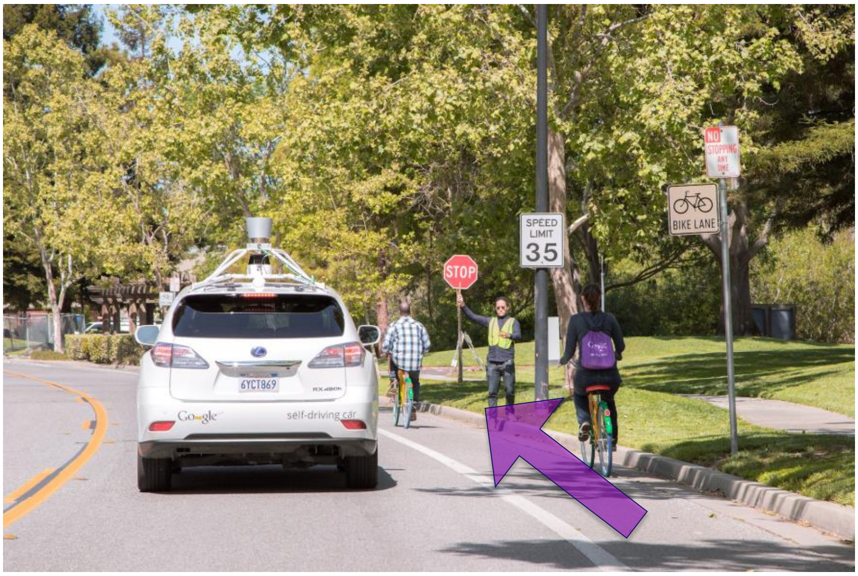
Motion Planning? Let's Get Moving…

Motion Planning? Let's Get Moving?

Motion Planning? The clutter can not be "ignored"



Motion Planning: Processing the Limits

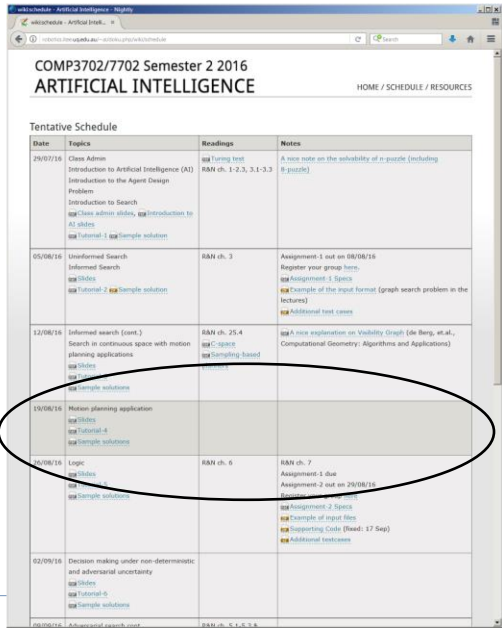


Path-Planning Approaches

- Roadmap
  Represent the connectivity of the free space by a network of 1-D curves
- Cell decomposition
  Decompose the free space into simple cells and represent the connectivity of the free space by the adjacency graph of these cells
- Potential field
  Define a function over the free space that has a global minimum at the goal configuration and follow its steepest descent

Slide from Latombe, CS326A

See Also: http://robotics.itee.uq.edu.au/~ai/



COMP3702/7702 Semester 2 2016
ARTIFICIAL INTELLIGENCE

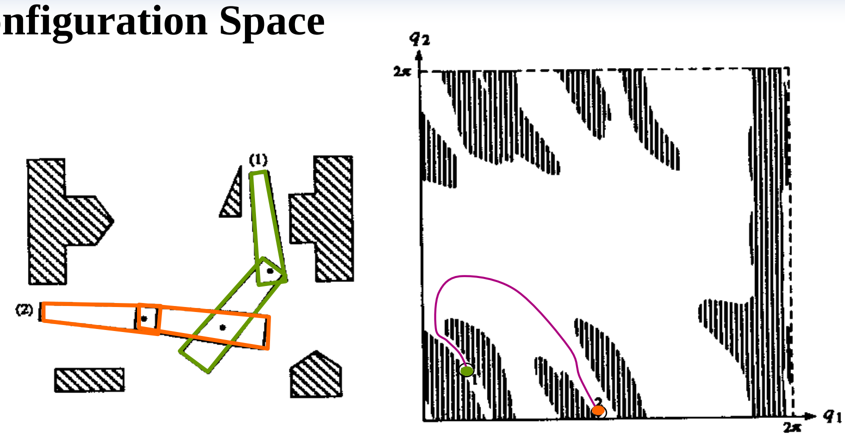
HOME / SCHEDULE / RESOURCES

Tentative Schedule

Date	Topics	Readings	Notes
29/07/16	Class Admin Introduction to Artificial Intelligence (AI) Introduction to the Agent Design Problem Introduction to Search Class admin slides, Introduction to AI slides Tutorial-1 Sample solution	Turing test R&N ch. 1-2.3, 3.1-3.3	A nice note on the solvability of n-puzzle (including 8-puzzle)
05/08/16	Uninformed Search Informed Search Slides Tutorial-2 Sample solution	R&N ch. 3	Assignment-1 out on 08/08/16 Register your group here. Assignment-1 Specs Example of the input format (graph search problem in the lectures) Additional test cases
12/08/16	Informed search (cont.) Search in continuous space with motion planning applications Slides [illegible] Sample solutions	R&N ch. 25.4 C-space Sampling-based [illegible]	A nice explanation on Visibility Graph (de Berg, et.al., Computational Geometry: Algorithms and Applications)
19/08/16	Motion planning application Slides Tutorial-4 Sample solutions		
26/08/16	Logic Slides [illegible] Sample solutions	R&N ch. 6	R&N ch. 7 Assignment-1 due Assignment-2 out on 29/08/16 Register [illegible] here Assignment-2 Specs Example of input files Supporting Code (fixed: 17 Sep) Additional testcases
02/09/16	Decision making under non-deterministic and adversarial uncertainty Slides Tutorial-6 Sample solutions		

External Configuration Is Important …

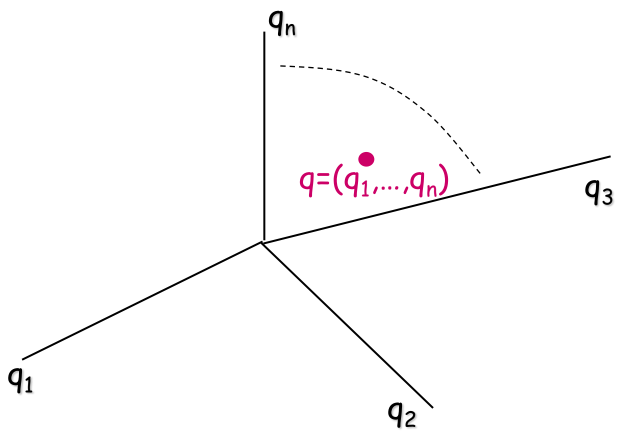
Configuration Space



- A robot configuration is a specification of the positions of all robot points relative to a fixed coordinate system
- Usually a configuration is expressed as a "vector" of position/orientation parameters

Slide from Latombe, CS326A

Motion Planning in C-Space

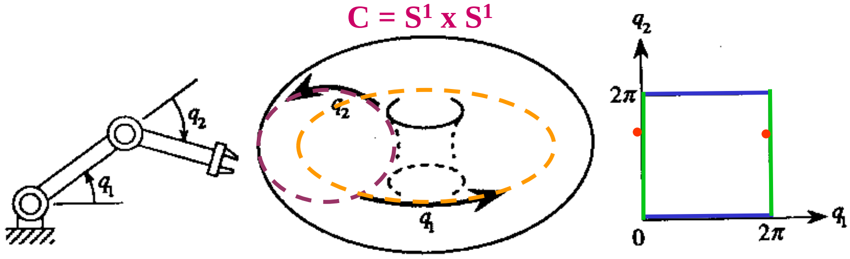


Slide from Latombe, CS326A

Configuration Space of a Robot

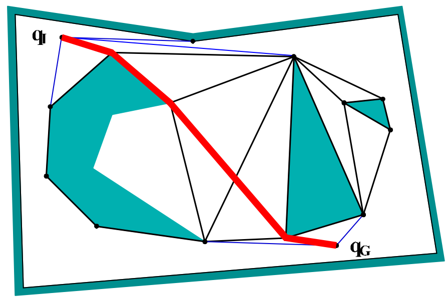
- Space of all its possible configurations
- But the topology of this space is usually not that of a Cartesian space

$C = S^1 \times S^1$



Slide from Latombe, CS326A

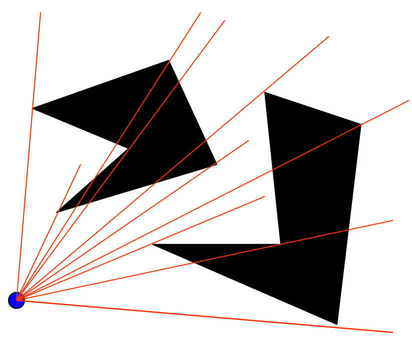
Geometric Planning Methods



- Several Geometric Methods:
  - Vertical (Trapezoidal) Cell Decomposition
  - **Roadmap Methods**
    - **Cell (Triangular) Decomposition**
    - Visibility Graphs
    - Veroni Graphs

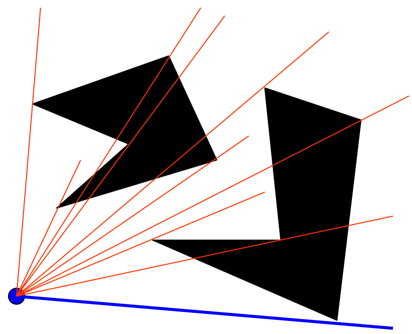
Artwork from LaValle, Ch. 6

I. Rotational Sweep



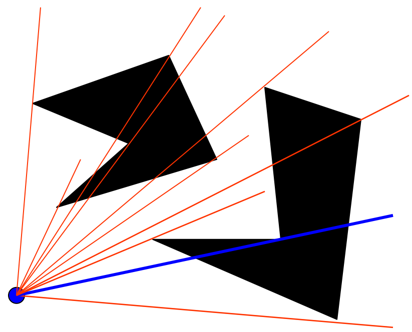
Slide from Latombe, CS326A

Rotational Sweep



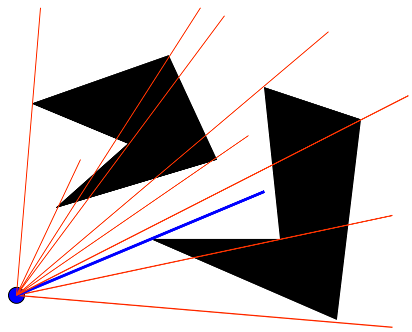
Slide from Latombe, CS326A

Rotational Sweep



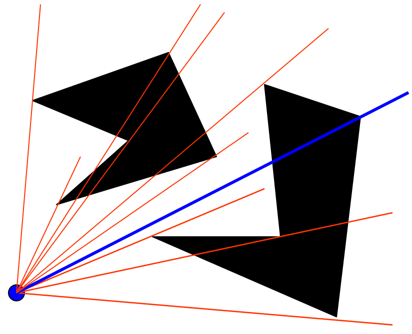
Slide from Latombe, CS326A

Rotational Sweep



Slide from Latombe, CS326A

Rotational Sweep



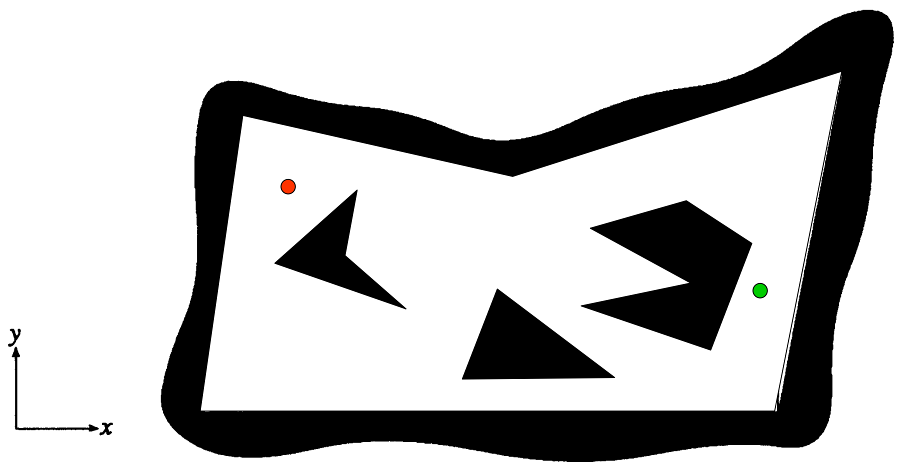
Slide from Latombe, CS326A

II. Cell-Decomposition Methods

Two classes of methods:

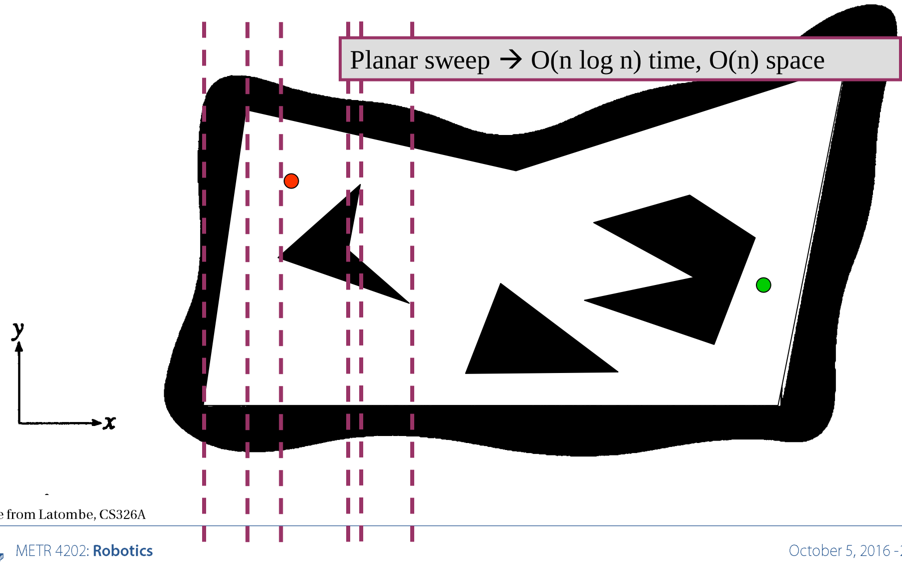
- Exact cell decomposition
  - The free space **F** is represented by a collection of non-overlapping cells whose union is exactly **F**
  - Example: trapezoidal decomposition
- Approximate cell decomposition
  - **F** is represented by a collection of non-overlapping cells whose union is contained in **F**
    Examples: quadtree, octree, 2n-tree

Trapezoidal decomposition



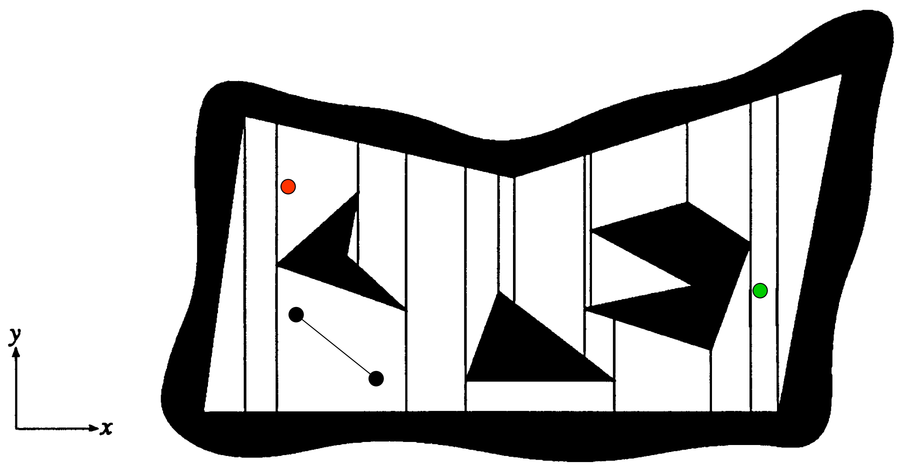
Slide from Latombe, CS326A

Trapezoidal decomposition



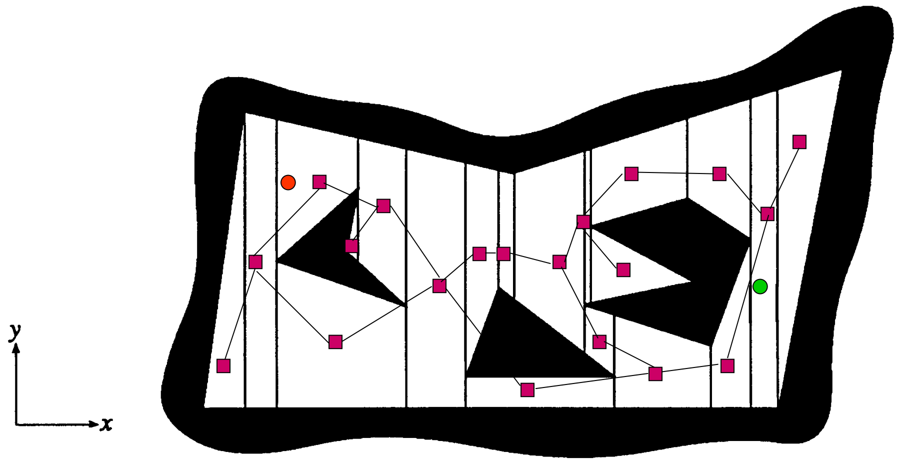
Slide from Latombe, CS326A

Trapezoidal decomposition



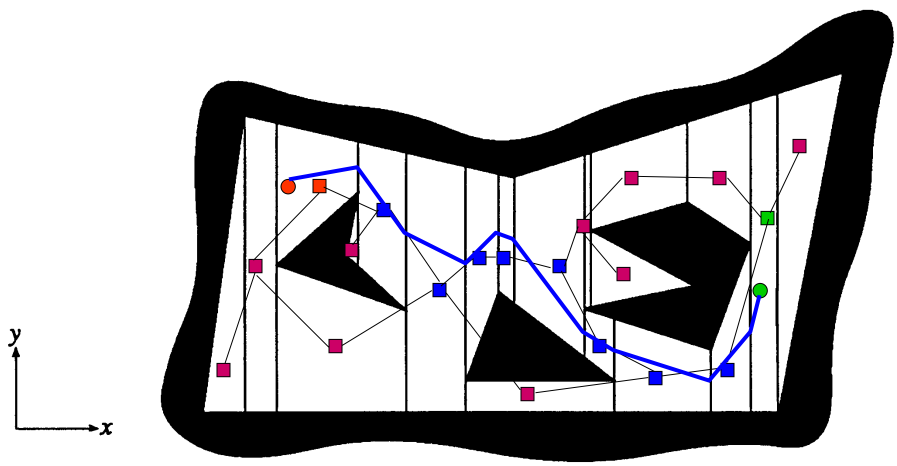
Slide from Latombe, CS326A

Trapezoidal decomposition



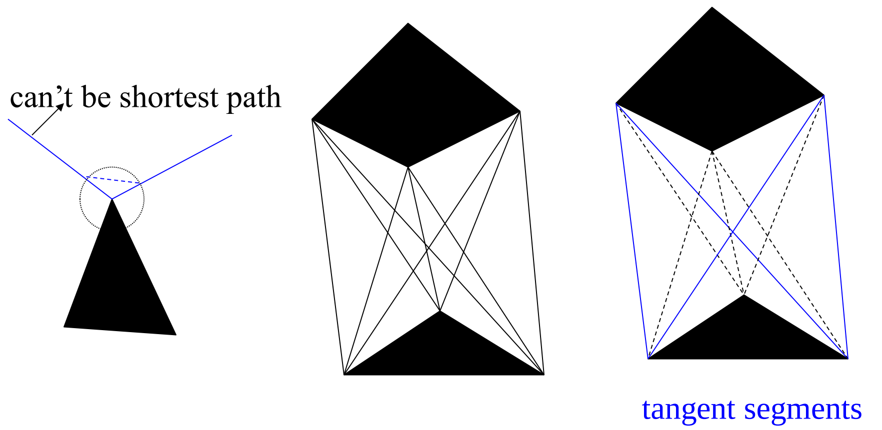
Slide from Latombe, CS326A

Trapezoidal decomposition



Slide from Latombe, CS326A

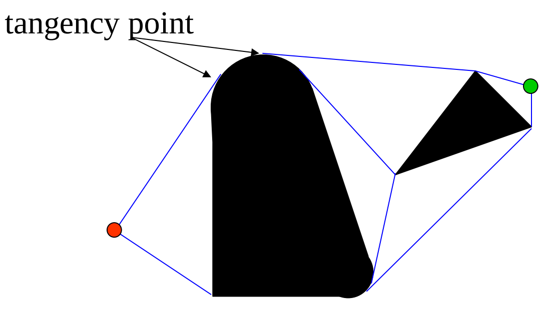
II.Visibility Graph



→ Eliminate concave obstacle vertices

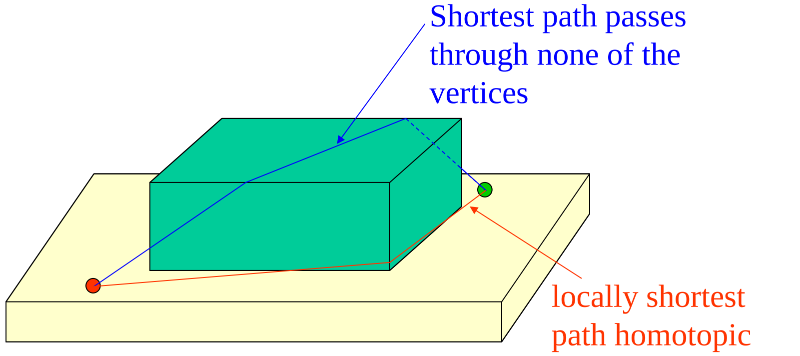
Slide from Latombe, CS326A

Generalized (Reduced) -- Visibility Graph



Slide from Latombe, CS326A

Three-Dimensional Space

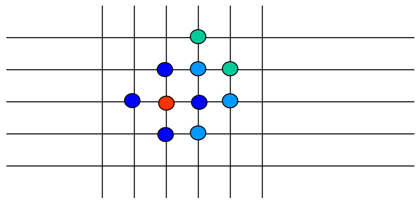


Computing the shortest collision-free path in a polyhedral space is NP-hard

Slide from Latombe, CS326A

Sketch of Grid Algorithm (with best-first search)

- Place regular grid G over space
- Search G using best-first search algorithm with potential as heuristic function



Slide from Latombe, CS326A

Simple Algorithm (for Visibility Graphs)

- Install all obstacles vertices in VG, plus the start and goal positions
- For every pair of nodes u, v in VG

    If segment(u,v) is an obstacle edge then

        insert (u,v) into VG

    else

    for every obstacle edge e

        if segment(u,v) intersects e

            then go up to segment

    insert (u,v) into VG
- Search VG using A*

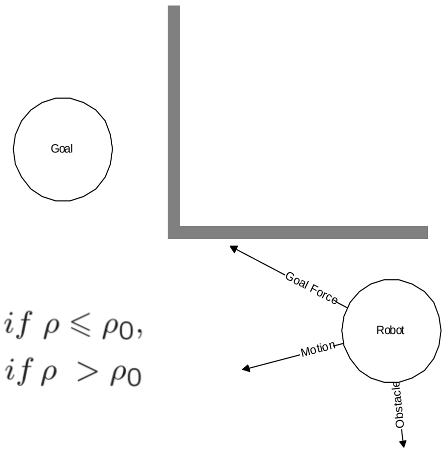
Slide based on Latombe, CS326A

III. Potential Field Methods

- Approach initially proposed for real-time collision avoidance [Khatib, 86]

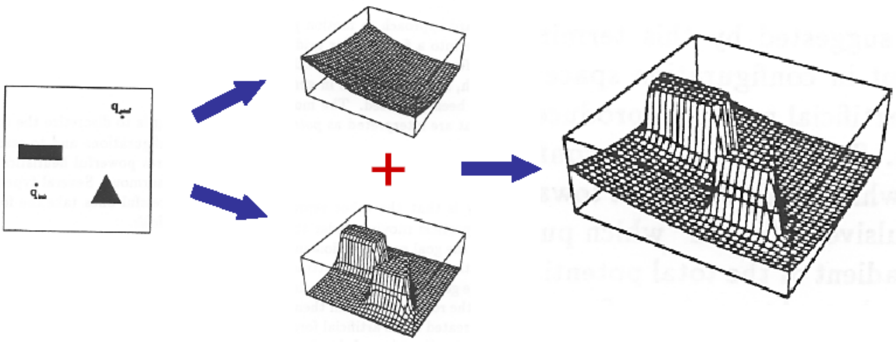
$$F_{Goal} = -k_p(x - x_{Goal})$$

$$F_{Obstacle} = \begin{cases} \eta \left(\frac{1}{\rho} - \frac{1}{\rho_0}\right) \frac{1}{\rho^2}\frac{\partial \rho}{\partial x} & if\ \rho \leqslant \rho_0, \\ 0 & if\ \rho > \rho_0 \end{cases}$$



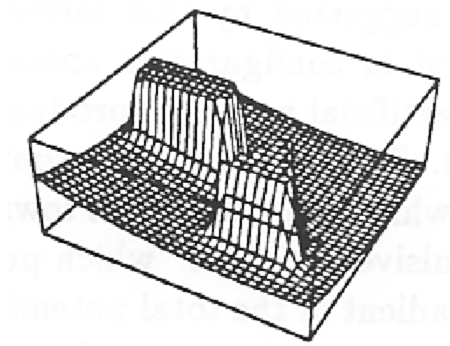
Slide based on Latombe, CS326A

Attractive and Repulsive fields



Slide from Latombe, CS326A

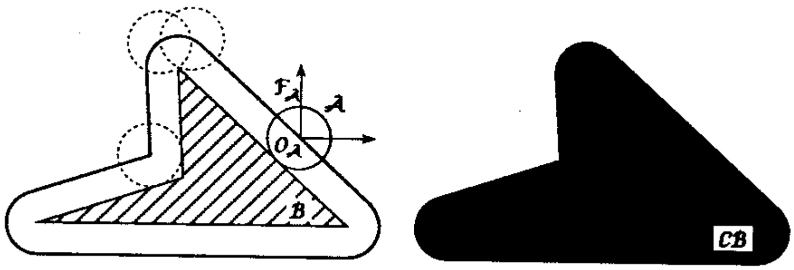
Local-Minimum Issue



- Perform best-first search (possibility of combining with approximate cell decomposition)
- Alternate descents and random walks
- Use local-minimum-free potential (navigation function)

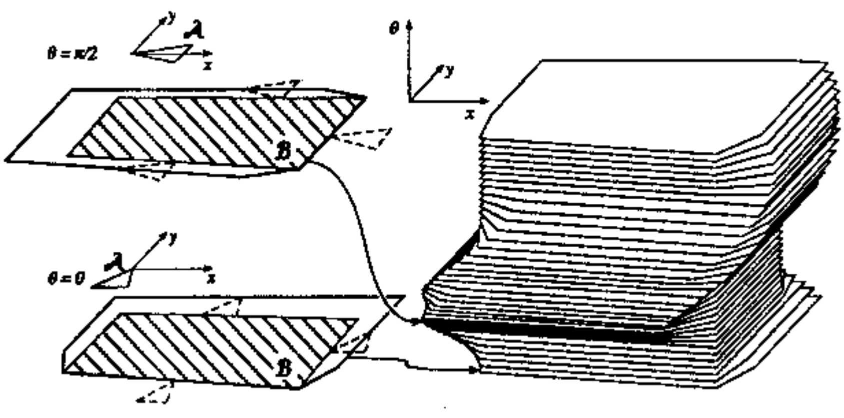
Slide from Latombe, CS326A

Disc Robot in 2-D Workspace



Slide from Latombe, CS326A

Rigid Robot Translating and Rotating in 2-D



Slide from Latombe, CS326A

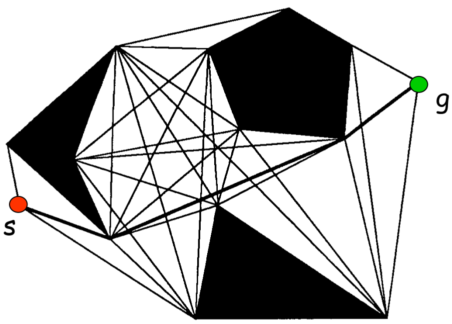
IV. Roadmap Methods

- **Visibility graph**
- **Voronoi diagram**
- Silhouette
  First complete general method that applies to spaces of any dimension and is singly exponential in # of dimensions [Canny, 87]
- **Probabilistic roadmaps (PRMS) and Rapidly-exploring Randomized Trees (RRTs)**

Slide from Latombe, CS326A

Roadmap Methods

- **Visibility graph**
  Introduced in the Shakey project at SRI in the late 60s. Can produce shortest paths in 2-D configuration spaces

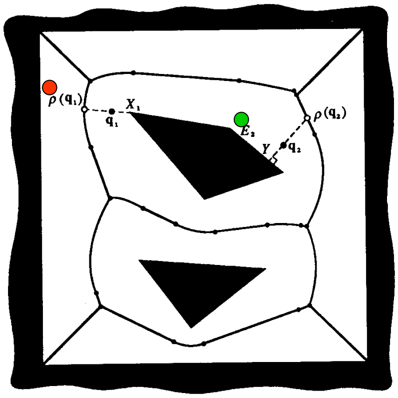


Slide from Latombe, CS326A

Roadmap Methods

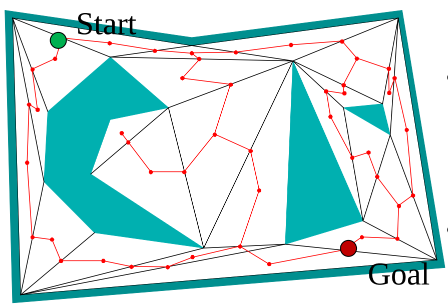
- Voronoi diagram
  Introduced by Computational Geometry researchers. Generate paths that maximizes clearance.

  O(n log n) time
  O(n) space



Slide from Latombe, CS326A

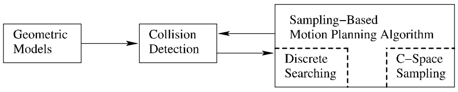
Limits of Geometric Planning Methods



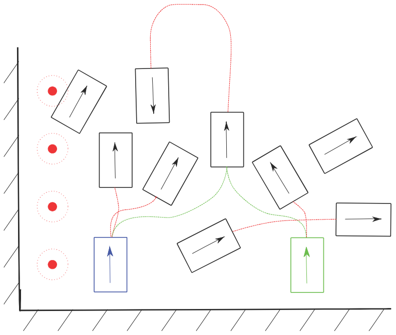
- How does this scale to high degrees of freedom?
- What about "dynamic constraints"?
- What about optimality?
- How to tie this to learning and optimization

Artwork from LaValle, Ch. 6

Sample-Based Motion Planning

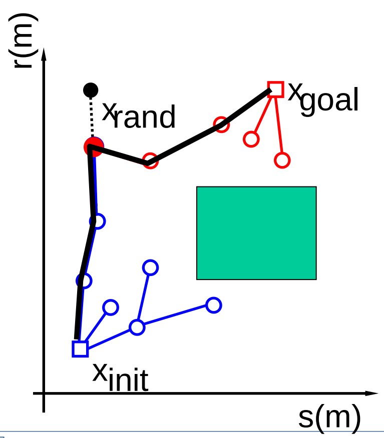


- PRMs
- RRTs

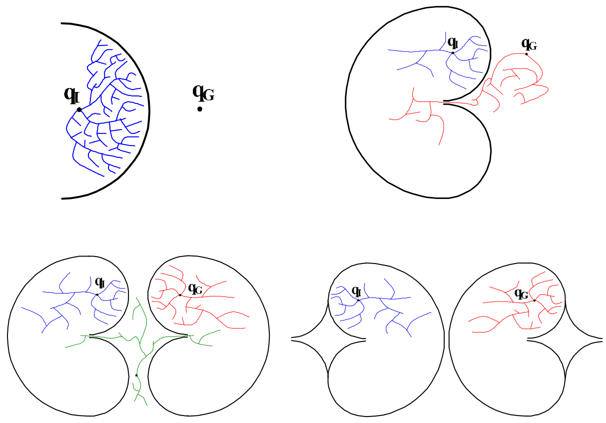


Artwork based on LaValle, Ch. 5

Rapidly Exploring Random Trees (RRT)

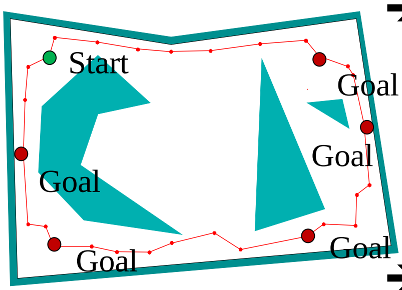


Sampling and the "Bug Trap" Problem



Artwork based on LaValle, Ch. 5

Multiple Points & Sequencing



- Sequencing
  - Determining the "best" order to go in

➔ Travelling Salesman Problem

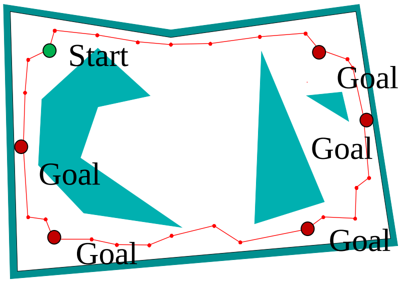
A salesman has to visit each city on a given list exactly once. In doing this, he **starts** from his home city and in the **end he has to return to his home** city. It is plausible for him to select the order in which he visits the cities so that the **total of the distances travelled** in his tour is as small as possible.

➔ Multi-Goal Problem

A salesman has to visit each city on a given list exactly once. In doing this, he **starts** from his home city and in the **end he has to return to his home** city. It is plausible for him to select the order in which he visits the cities so that the **total of the distances travelled** in his tour is as small as possible.

Artwork based on LaValle, Ch. 6

Travelling Salesman Problem

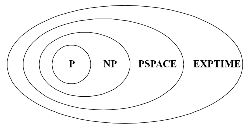


- Given a $n \times n$ distance matrix $\mathbf{C}=(c_{ij})$

- Minimize:

$$c(\pi) = \sum_{i=1}^{n} c_{i\pi(i)}$$

- Note that this problem is NP-Hard

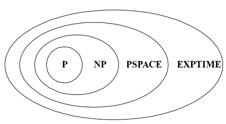


➔ BUT, Special Cases are Well-Solvable!

Artwork based on LaValle, Ch. 6

Travelling Salesman Problem [2]

- This problem is NP-Hard



➔ BUT,
Special Cases are
Well-Solvable!

For the Euclidean case
(where the points are on the 2D Euclidean plane) :

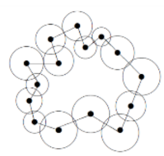
- The shortest TSP tour does not intersect itself, and thus geometry makes the problem somewhat easier.
- If all cities lie on the boundary of a convex polygon, the optimal tour is a cyclic walk along the boundary of the polygon (in clockwise or counterclockwise direction).

***The k*-line TSP**

- The a special case where the cities lie on *k* parallel (or almost parallel) lines in the Euclidean plane.
- EG: Fabrication of printed circuit boards
- Solvable in $\mathbf{O}(n^3)$ time by Dynamic Programming (Rote's algorithm)

The necklace TSP

- The special Euclidean TSP case where there exist *n* circles around the *n* cities such that every cycle intersects exactly two adjacent circles



Cool Robotics Share