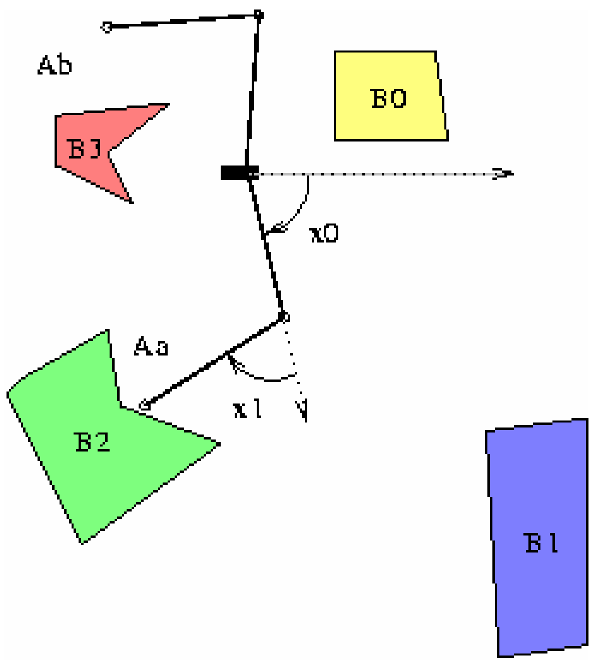
Path Planning Approaches

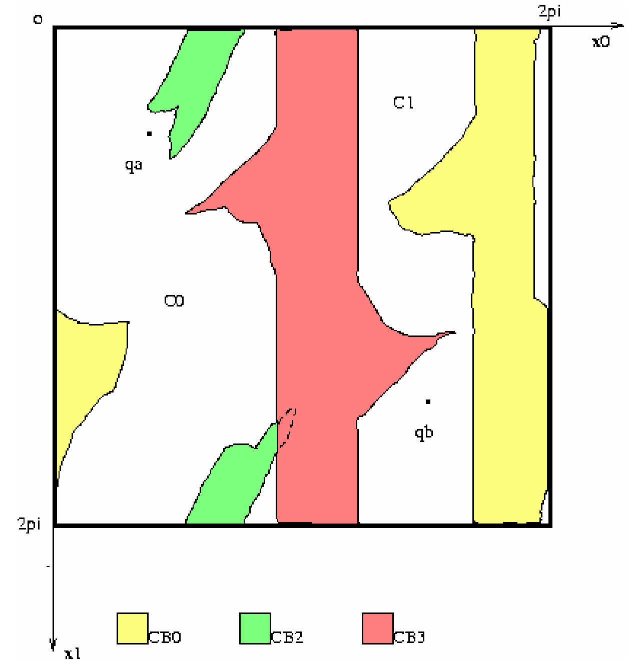
Dr. Thierry Fraichard

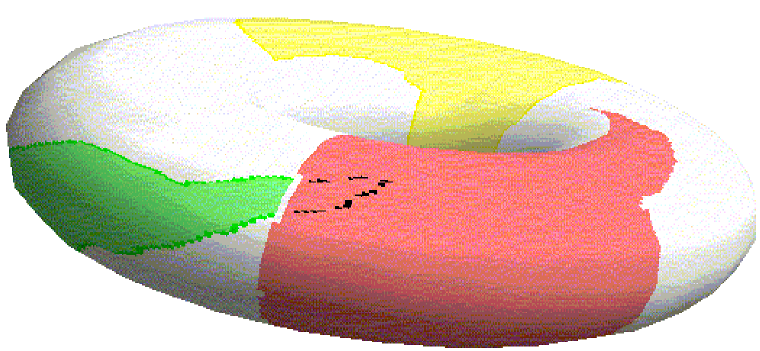
Path Planning Problem

W, A $\rightarrow$ *C, B*$_i$ $\rightarrow$ *CB*$_i$, *i = 1…b, q*$_s$, *q*$_g$

Goal: explore C_{free} to compute a collision-free path between q_s *and* q_g







Completeness Issue

Complete algorithm: finds a solution if one exists, reports failure if not

Complexity of complete path planning: strong evidence that it takes time exponential in *d*, the dimension of the configuration space *C*

Specific complete path planning algorithms have been implemented for *d* = 2, 3 or 4

Two complete general purpose path planning algorithms have been proposed *[Schwartz & Sharir 81, Canny 87]*, (resp. twice and singly exponential in *d*) but…

None has been implemented!

Complete algorithms: Theoretical interest mostly
In practice, difficult to implement and not robust

What to do then?

Completeness Issue (C'ed)

What can be done:

(1) Be practical, forget about completeness and be *heuristic*
$\Rightarrow$ Hopefully works well in most encountered situations, no performance guarantee

(2) Settle for a weaker notion of completeness:

Resolution completeness: based on a systematic discretization of *C*
Completeness is guaranteed for a given resolution level
(Does not work well when *d* is high)

Probabilistic completeness: the probability of finding a solution converges towards 1 when the algorithm is given infinite time
(Weaker property: if no solution is found within a finite time then what?)

Possible Classifying Criteria for Path Planning Methods

Is the method complete?

(a) *Exact* approaches	Complete
(b) *Approximate* approaches	Resolution complete
(c) *Randomized* approaches	Probabilistically complete
(d) *Heuristic* approaches	Uncomplete

Does the method explicitly compute the configuration space?

Does the method attempt to capture the topology of the configuration space?

Is the method designed to handle multiple path planning problems?

(a) Single query	Goal-dependent
(b) Multiple query	Goal-independent preprocessing

...

Families of Path Planning Methods

(1) Methods exploring a *search graph*

Attempt to capture the topology of the configuration space → Graph structure

Preprocessing of the configuration space independently of any goal (multiple query)

(2) Methods incrementally building a *search tree*

No attempt to capture the topology of the configuration space

Goal-dependent methods (single query)

(3) “Other” methods

Graph-Based Methods

Visibility graph *[Nilsson, 69]*

Retraction[-like]

- Voronoï diagram *[Dunlaing & Yap, 82]*
- Silhouette *[Canny, 88]*
- Generalized cylinders *[Brooks, 82]*

Cellular decomposition

- Exact
- Approximate

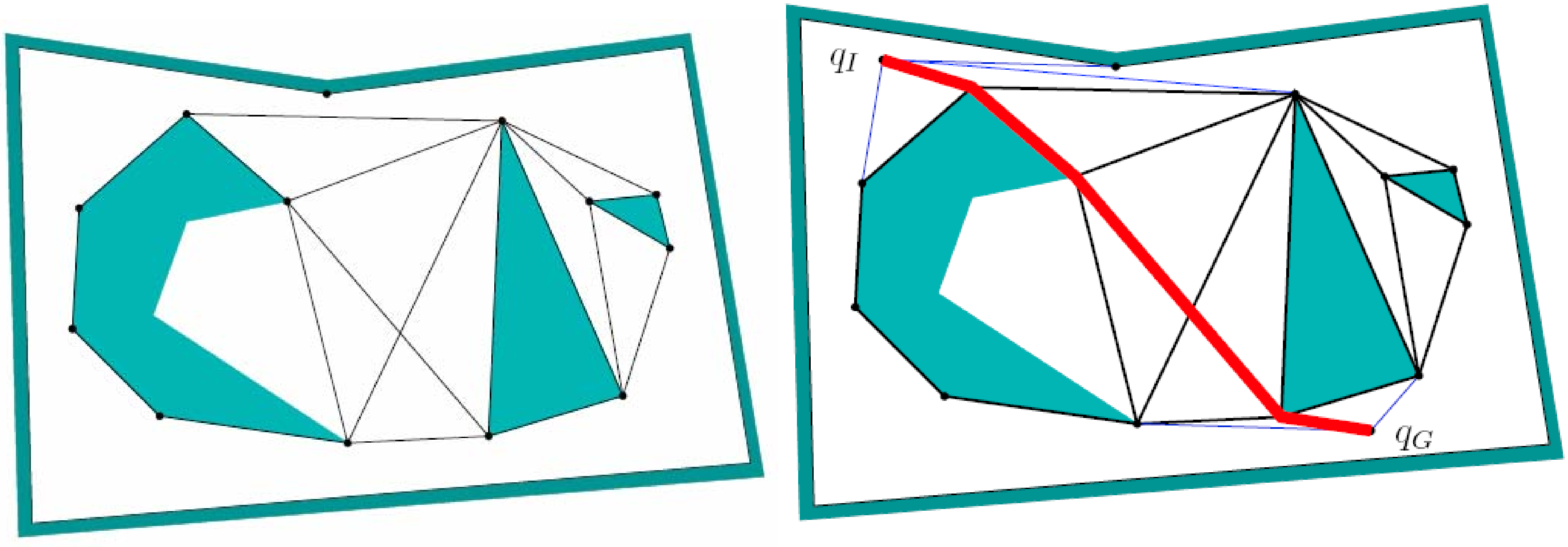
Probabilistic roadmap and its variants

Visibility Graph *[Nilsson, 69]*

Network of 1D curves capturing the topology of $C_{semifree}$, structured as a graph

Path planning: (1) connect q_s *and* q_g to the roadmap, (2) graph search

Shortest path in 2D space (no longer true in a 3D space)



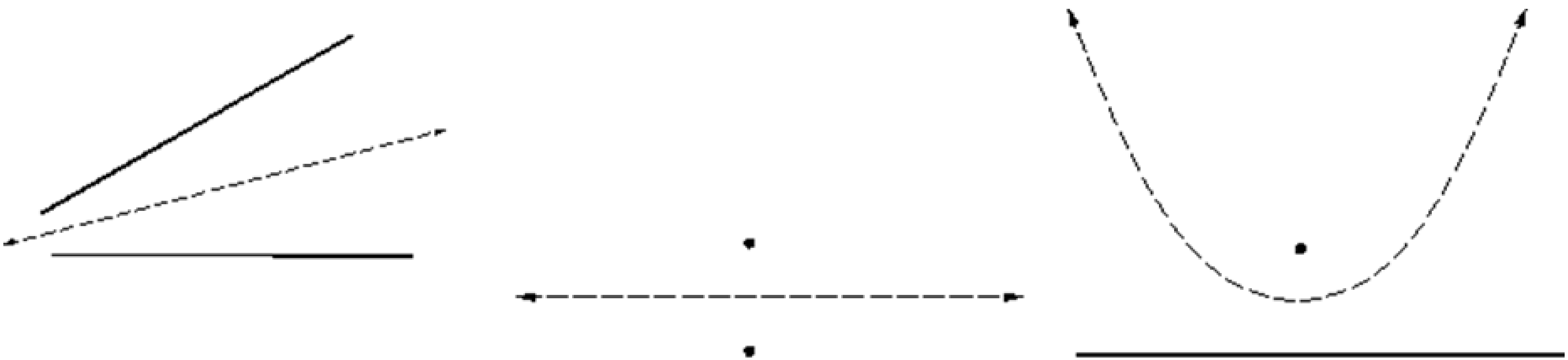
Voronoï Diagram *[Dunlaing & Yap, 82]*

Based on the topological notion of *retraction*: continuous surjective mapping (n to 1) of a topological space onto one of its subset (of lower dimensionality)

In addition, it should preserve the connectivity of the initial topological space

$C = R^2$, polygonal configuration obstacles regions

Voronoï Diagram: retraction defined as the set of points whose minimal distance to δC_{free} is achieved with more than one points of δC_{free}

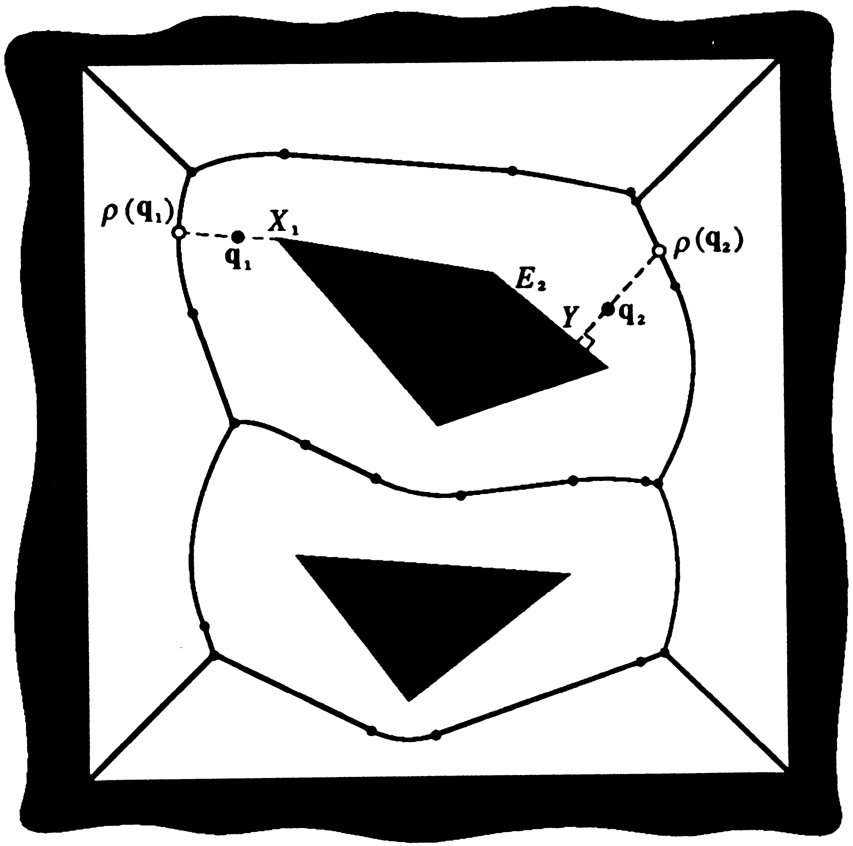


(a) (b) (c)

$\rightarrow$ 1D network of C_{free} curves: straight segments + parabolic arcs

Voronoï Diagram (C'ed)

Generate paths maximizing the clearance to the obstacles.
Applicable mostly to 2D spaces

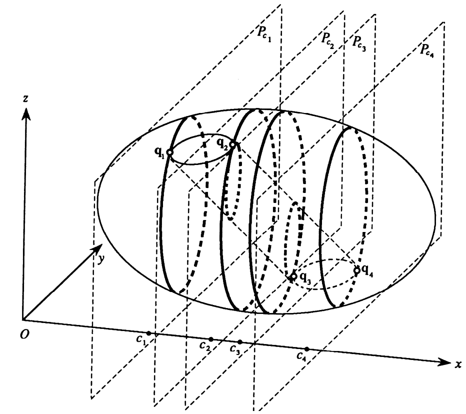


Silhouette *[Canny, 87]*

General (configuration space of arbitrary dimensionality *n*) and complete

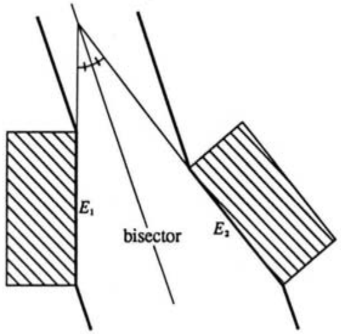
Single-exponential time complexity in *d* but…

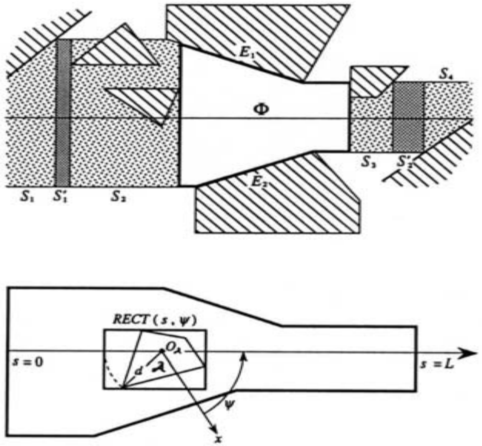
Never implemented! Theoretical interest only

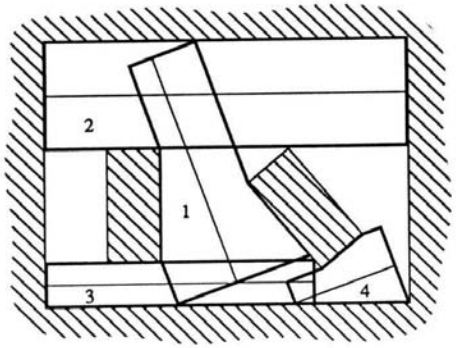


Generalized Cylinders *[Brooks, 82]*

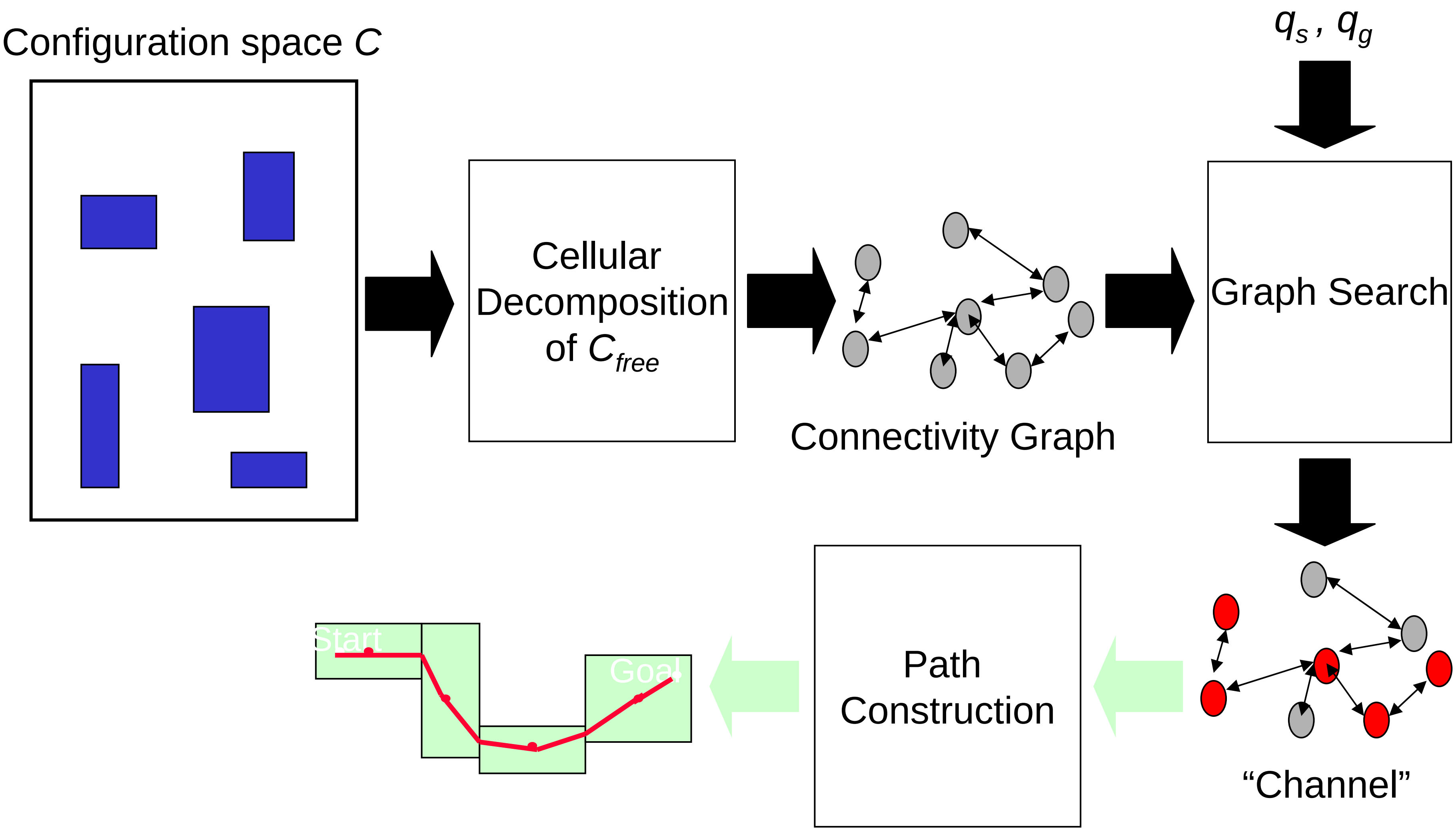
Approximation of the Voronoï diagram in the workspace







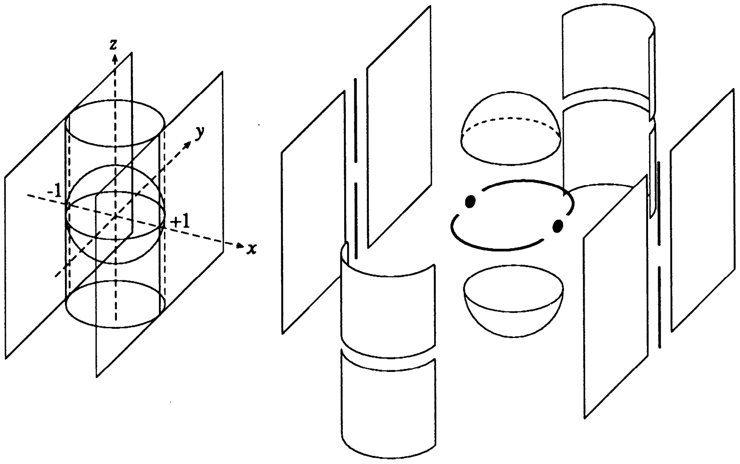
Cellular Decomposition



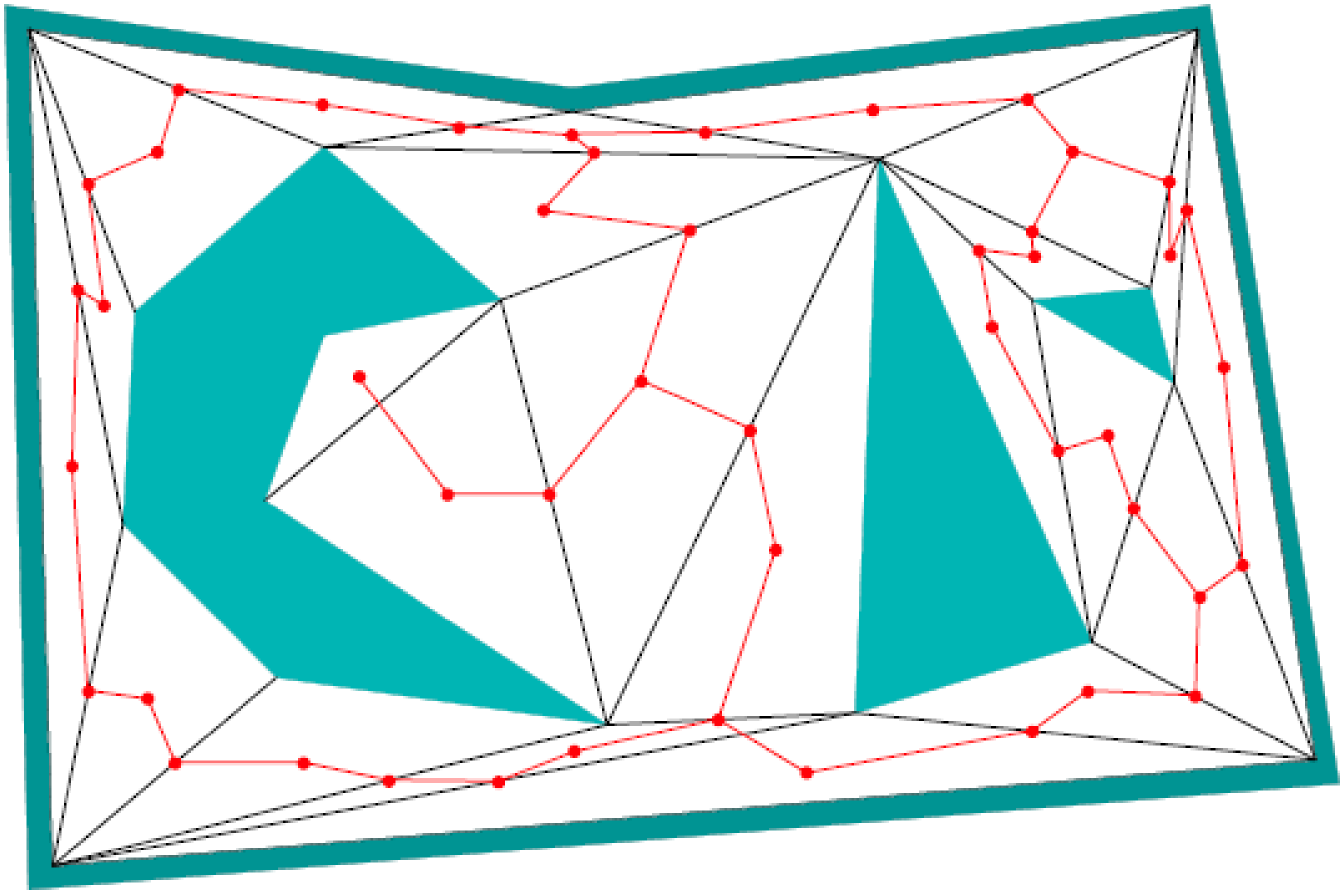
Exact Cellular Decomposition

Main features: Complete approach: U *cells* =C_{free}
Adapted cell shape
Reduced cell number
Increased decomposition complexity
Increased connectivity graph building complexity

e.g. Collins' cells decomposition for semi-algebraic sets *[Collins 75]* (used in *[Schwartz & Sharir 81]* to establish the decidability of the Generalized Piano Mover problem)

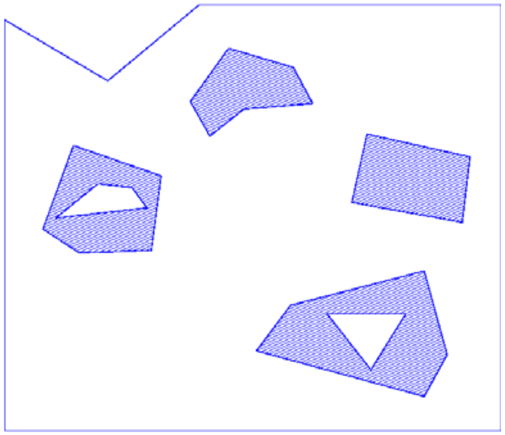


Convex Cell Decomposition

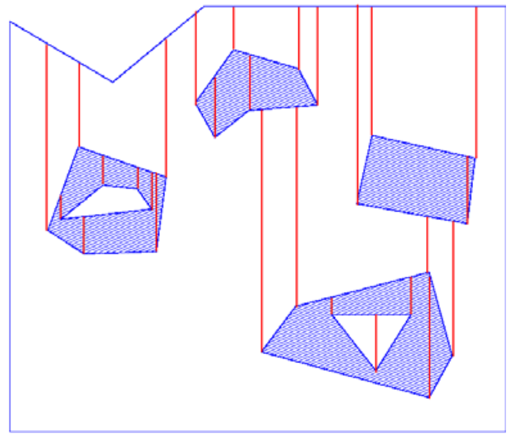


INRIA CNRS INP Grenoble Université Joseph Fourier

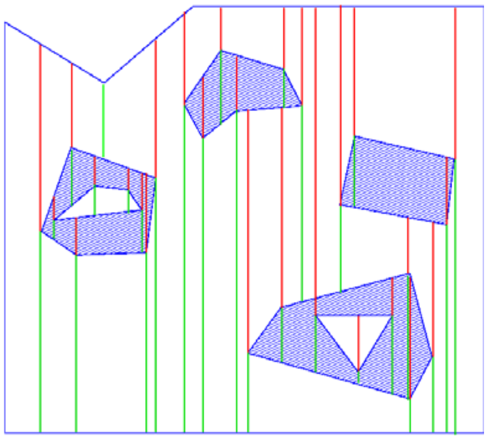
Trapezoidal Decomposition



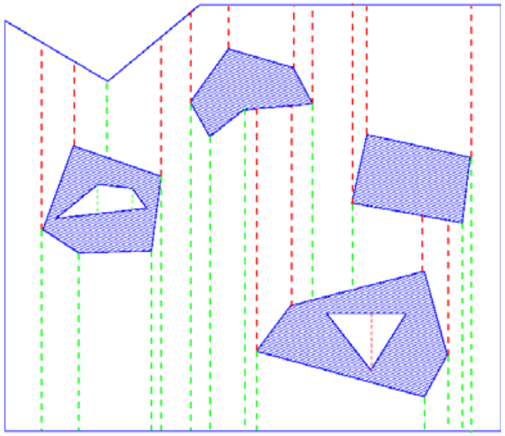
Configuration space



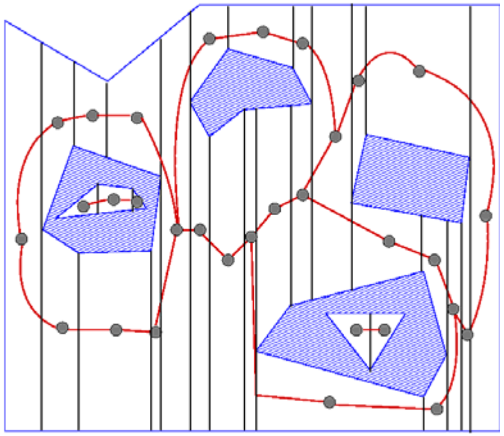
Upward extensions



Downward extensions



Deleting the trapezoids
within the obstacles

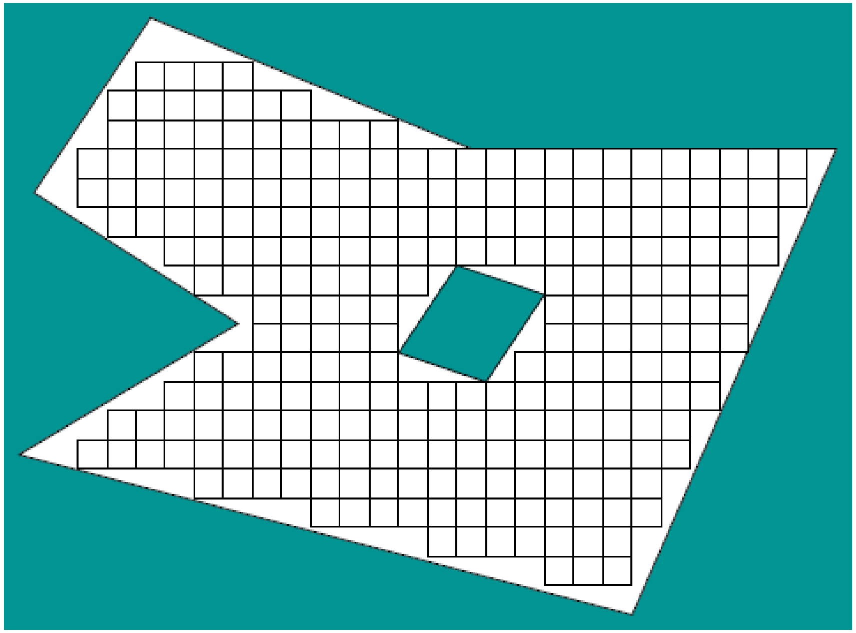


Building the
connectivity graph

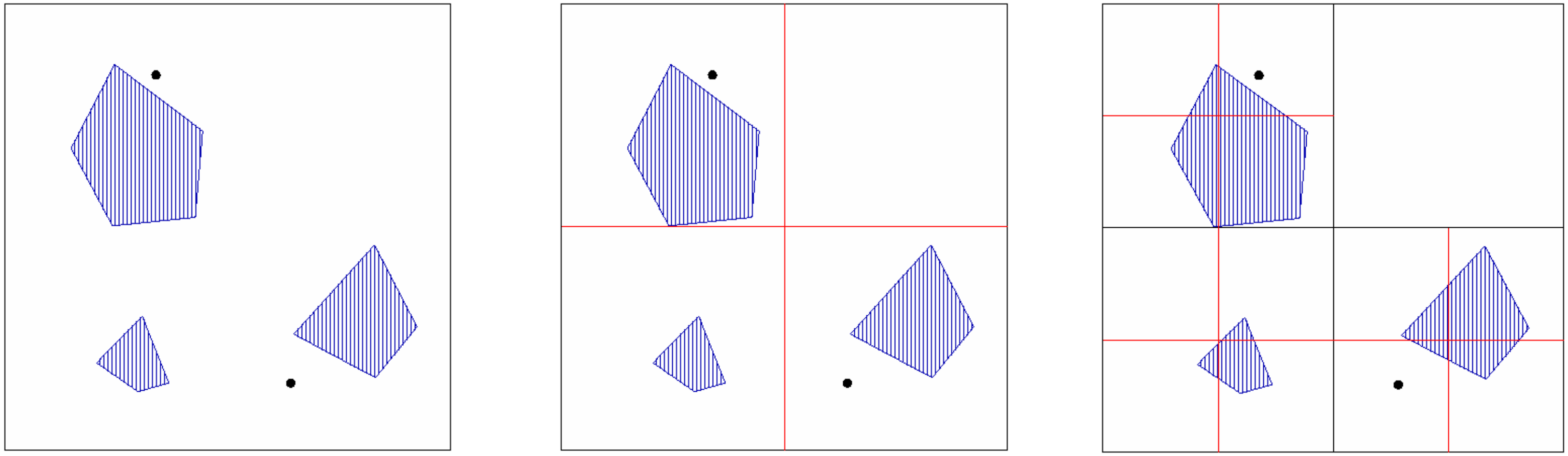
Approximate Cellular Decomposition

Main features: Resolution complete approach: U *cells* $\subset C_{free}$
Fixed cell shape
Large cell number
Reduced decomposition complexity
Reduced connectivity graph building complexity

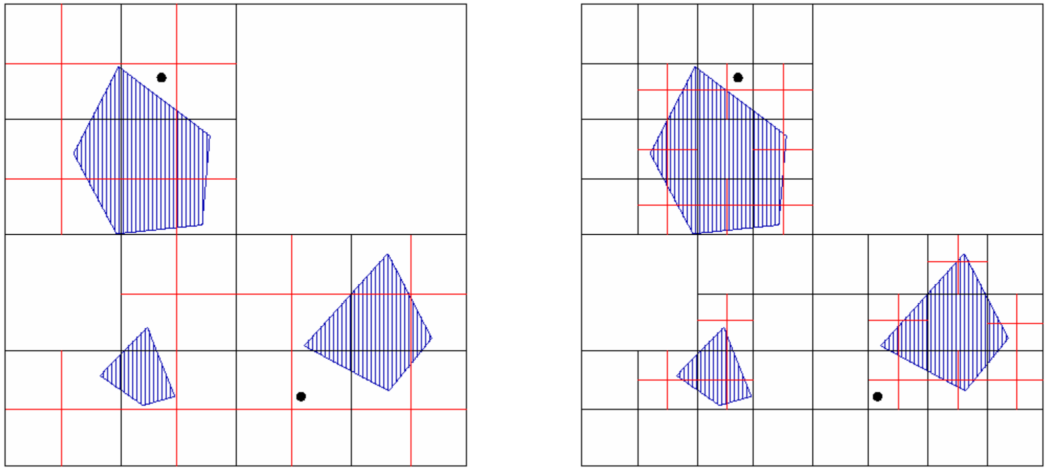
e.g. rectangular decomposition:



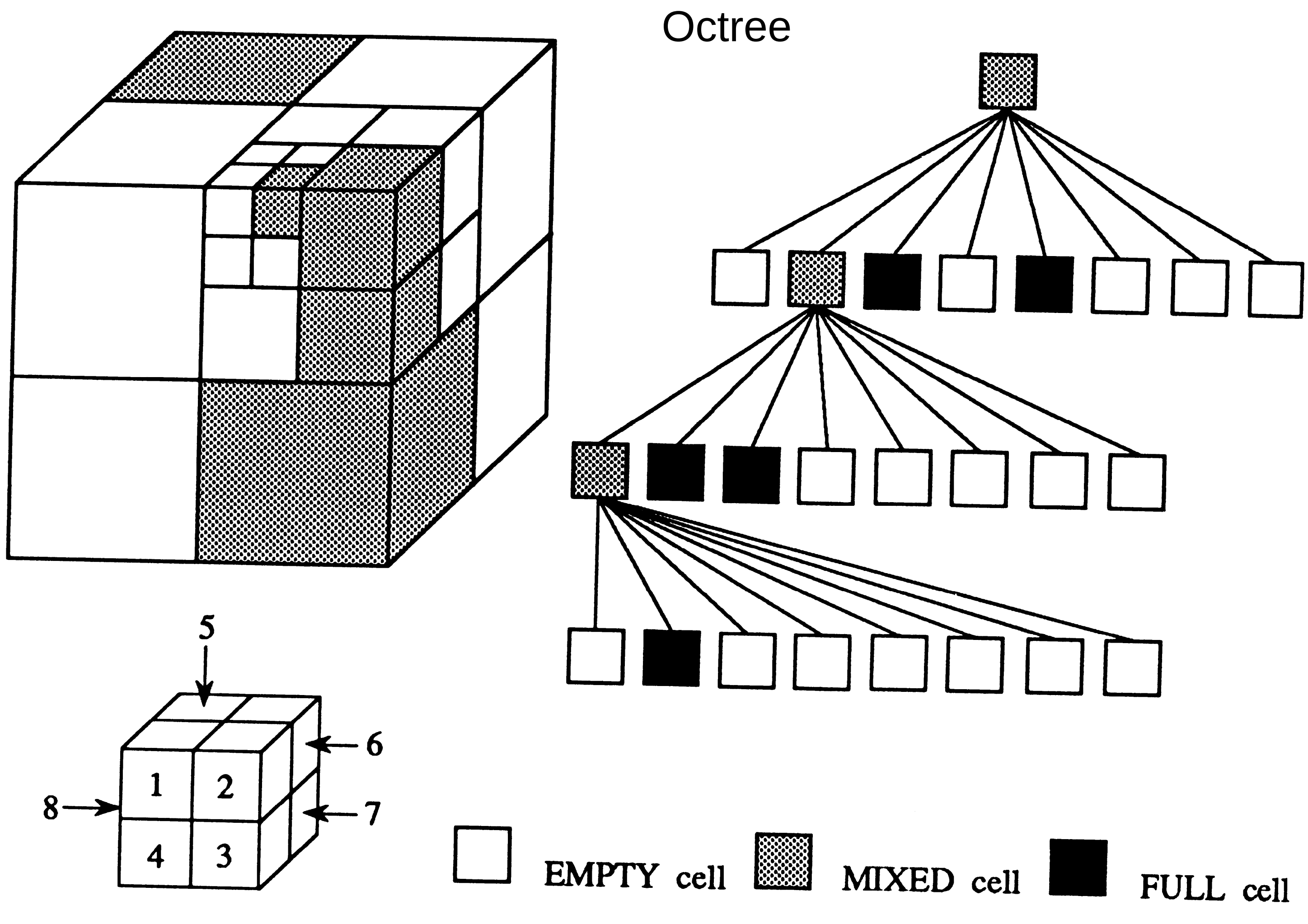
Hierarchical Cellular Decomposition



Quadtree

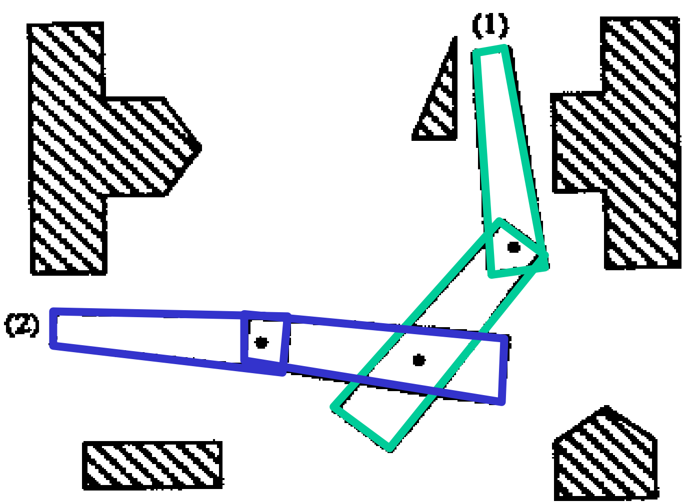


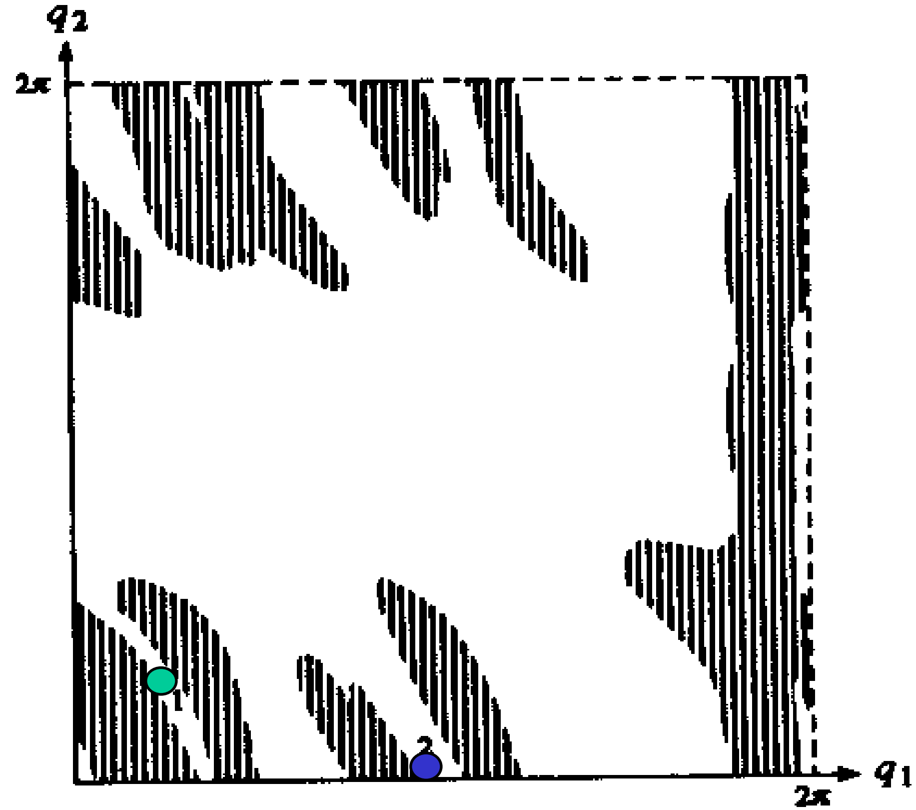
Hierarchical Cellular Decomposition



Probabilistic Roadmap
[Kavraki et al., *96; Svetska & Overmars, 96]*

Rationale: in general, computing C_{free} is too hard whereas checking whether a configuration or a path is collision-free can be done efficiently using recent collision-checking or distance computation techniques





INRIA

Probabilistic Roadmap Principle

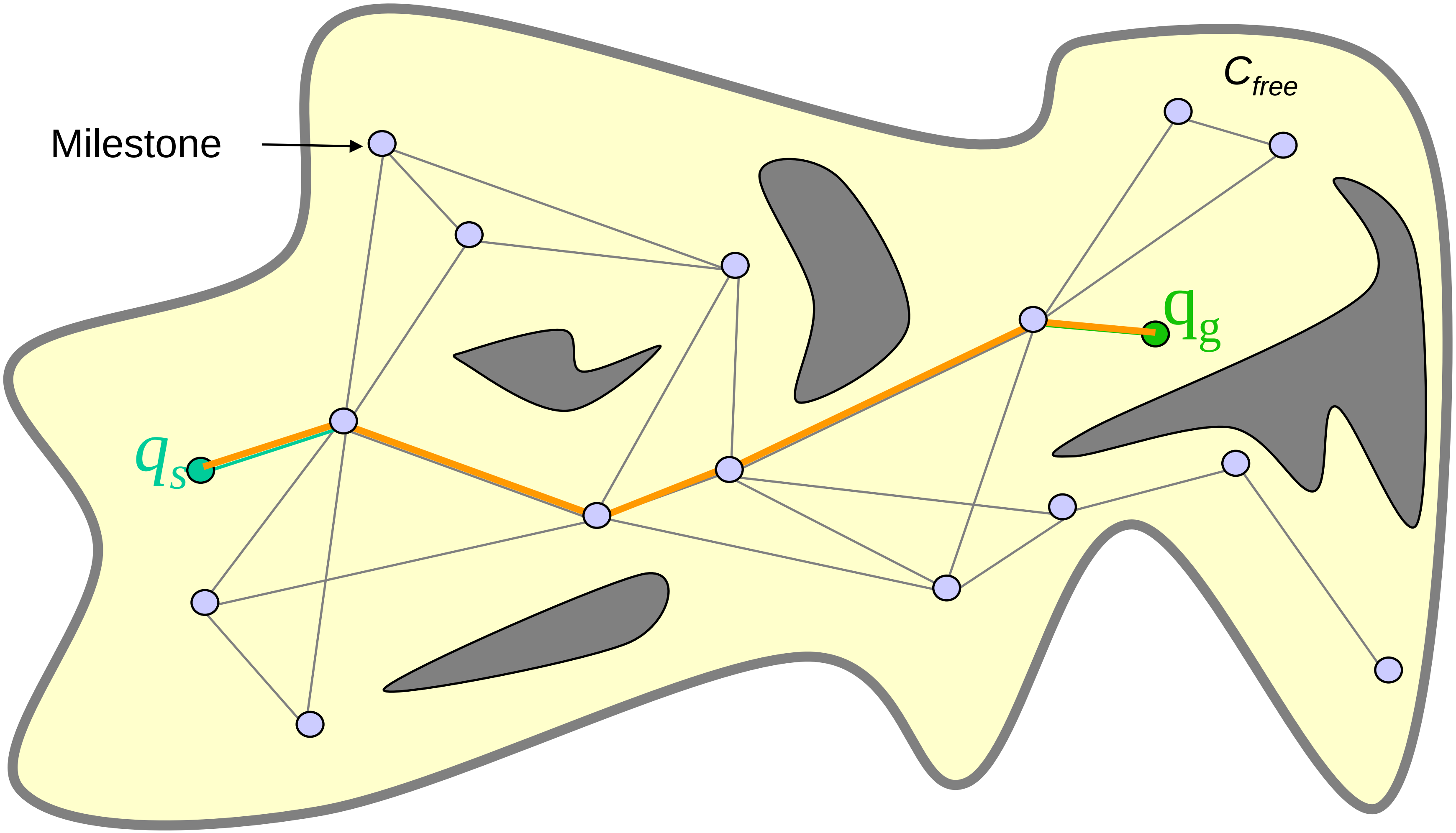
Key idea: approximate the free space by random sampling

Principle is very simple:

(1) Sample *C* randomly
(2) Keep the samples in C_{free} (*milestones*)
(3) Connect pair of milestones with simple paths

$\rightarrow$ Roadmap: network of 1D curves that approximate the connectivity of C_{free}

Probabilistic Roadmap Principle (C'ed)



“Good” Probabilistic Roadmap

Probabilistic completeness only

Main issue is to compute a “good” roadmap

Desirable properties:

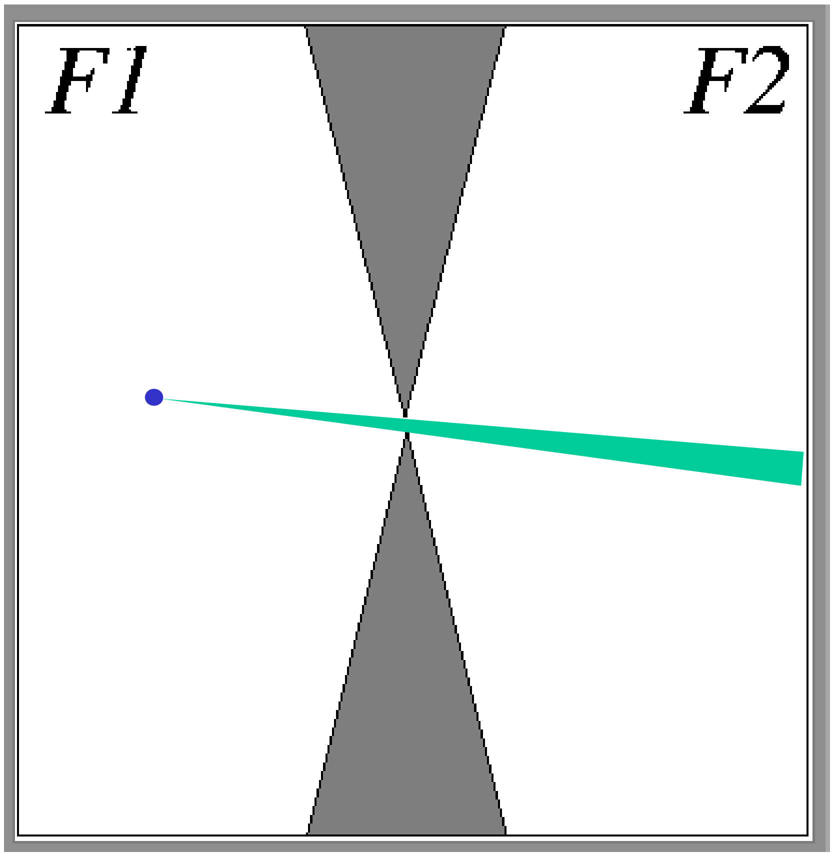
Coverage: the milestones should “see” most of C_{free} so as to guarantee that any start and goal configurations can be connected to the roadmap easily

$\rightarrow$ Concept of ε-*goodness* of C_{free}

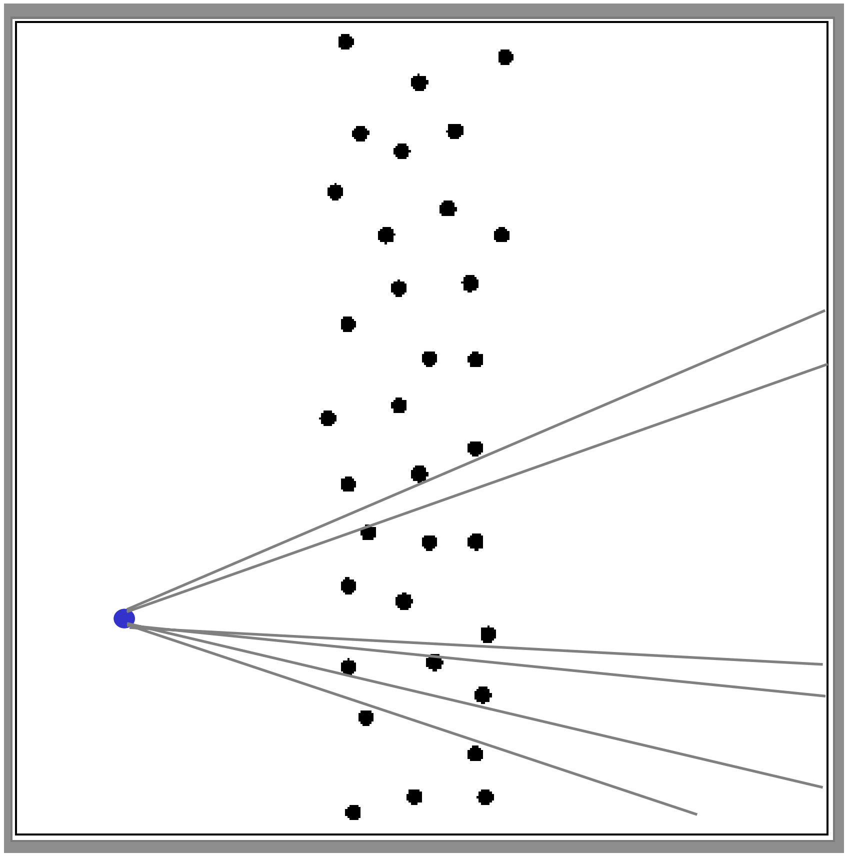
Connectivity: there should be a single-connected component of the roadmap in every connected component of C_{free}

$\rightarrow$ Concept of (α, β)-*expansiveness* of C_{free}

Narrow Passage Issue



Difficult



Easy

ε-Goodness

Let $\varepsilon \in (0, 1]$, $q \in C_{free}$ is ε-*good* if it sees an ε-fraction of $\mu(C_{free})$, the volume of C_{free}

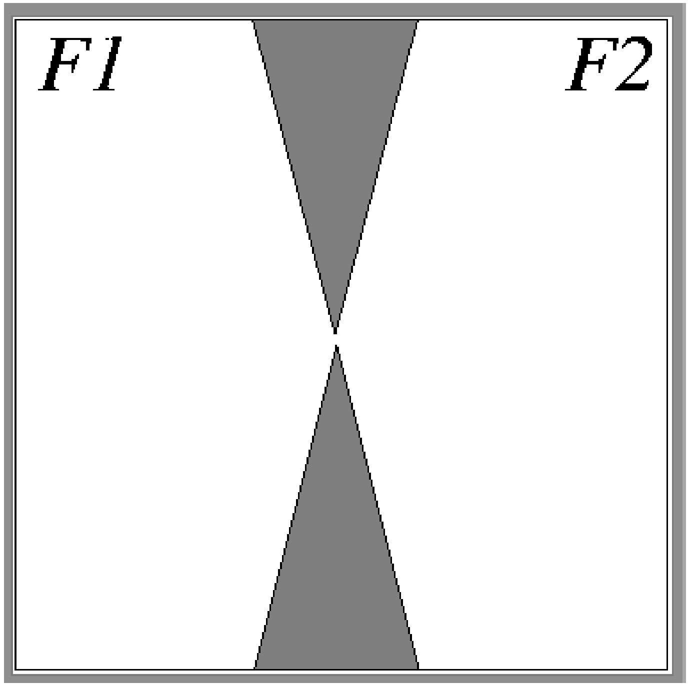
C_{free} is ε-*good* if every free configurations is ε-*good*

ε represents the smallest fraction of C_{free} visible from any configuration: $\varepsilon = \min \frac{\mu(V(q))}{\mu(C_{free})}$

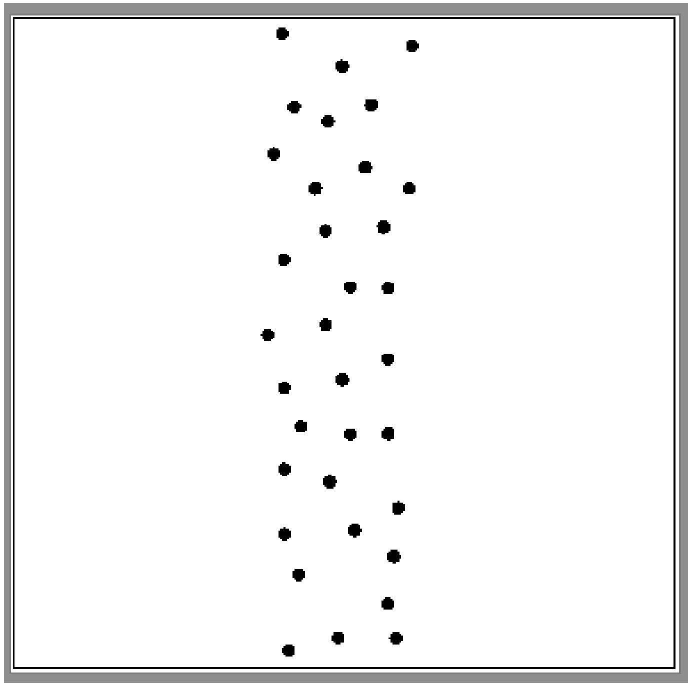
if C_{free} is ε-*good*, the volume of the subset of C_{free} not seen by any of s milestones picked uniformly at random has a probability proportional to e^{-s} of being greater than $\varepsilon\mu(C_{free})$ *[Kavraki et al., 95]*

INRIA CNRS INP Grenoble Université Joseph Fourier

Narrow Passage Issue



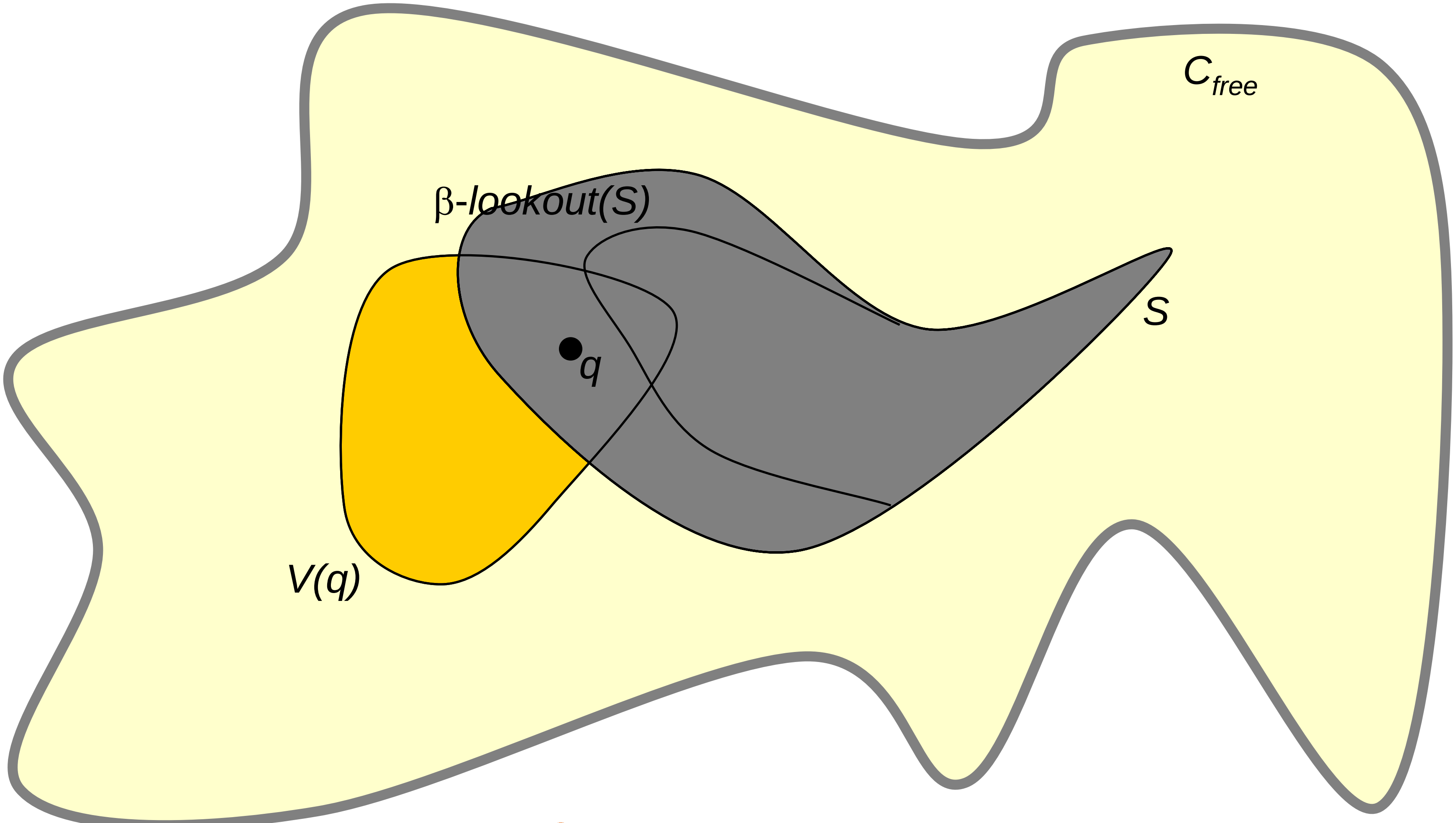
ε = 0.5



ε ≈ 1

β-*Lookout*

Let $\beta \in (0, 1]$, the β-*lookout* of an arbitrary subset S of C_{free} is the subset of the points of S that see a β-fraction of the volume $C_{free} \setminus S$



INRIA

(α, β)-Expansiveness

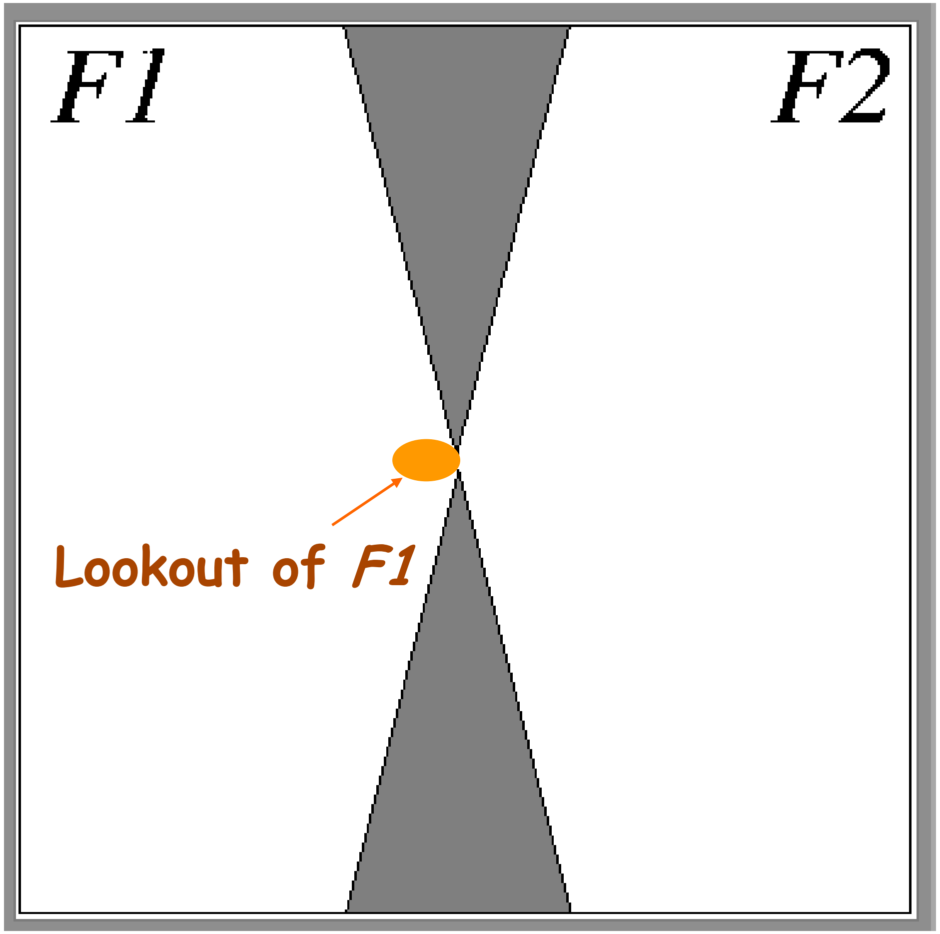
C_{free} is (α, β)-*expansive* if every subset S of C_{free} has a β -lookout of relative volume α

$$\alpha = \frac{\mu(\beta - lookout(S))}{\mu(S)}$$

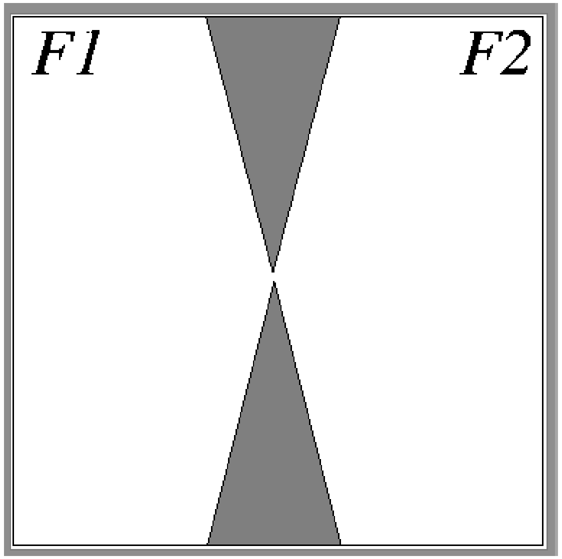
If C_{free} is expansive with large α and β then it is easy to sample new milestones that will expand the visibility region significantly (until C_{free} is completely covered)

[Hsu et al., 97] have established the relationship between (α, β), the number of milestones to sample and the probability that a connected component of C_{free} contains several roadmap components

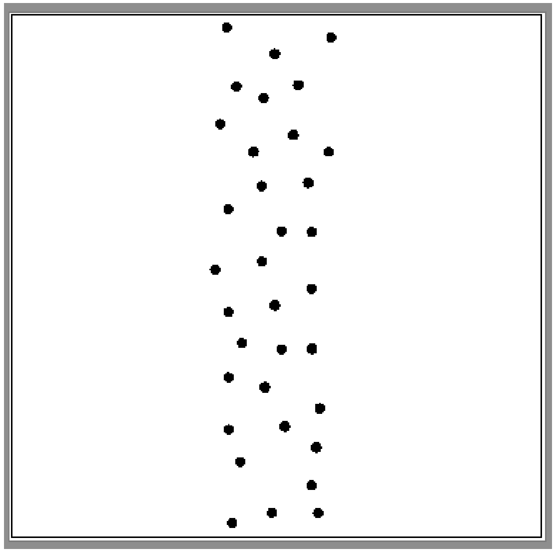
Narrow Passage Issue



Narrow Passage Issue



$\varepsilon = 0.5$
Poorly expansive



$\varepsilon \approx 1$
Expansive

ε-goodness and (α, β)-expansiveness are interesting results connecting the algorithm performance to *s* and ε or α and β, one problem though:
they are both defined in terms of *Cfree* that cannot be computed efficiently…

Importance of the sampling strategy, several were proposed: uniform, uniform with refinement in "difficult" regions, "push" non-free milestones in C_{free}, visibility-based...

Probabilistic Roadmap's Features

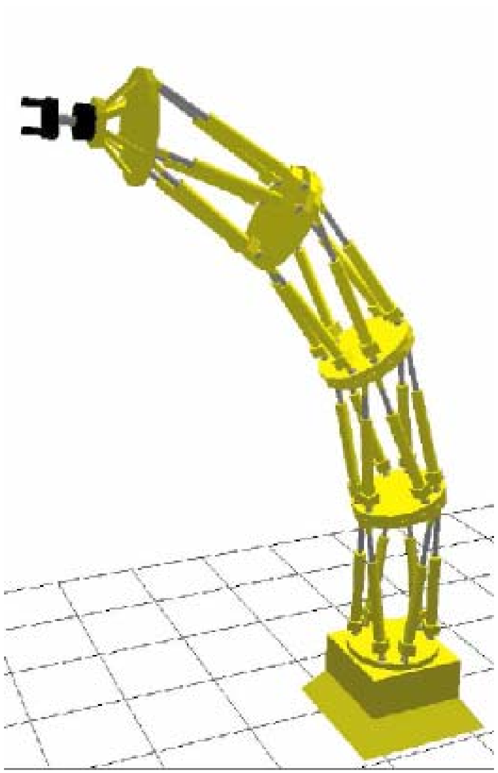
Proved to be an effective (easy to implement, fast, robust) computational framework to solve path planning problems in high-dimensional configuration spaces.

Successfully applied to different motion planning problems: moving obstacles, kinematic and dynamic constraints, manipulation…

Remaining issues:

How to obtain a good roadmap?

No rigorous termination criterion when no solution is found



97 dof

Tree-Based Methods

Grid-based methods

Dynamic programing

A* algorithm

Rapidly-exploring random trees *[LaValle, 98]*

Ariadne's Clew algorithm *[Ahuactzin, 94]*

Grid-Based Methods

Regular discretization of the configuration space (*grid*)
Adjacency relationship between the grid nodes (*neighbours*)

Starting from q_s, an exploration tree can be built and expanded until q_g is reached

Tree expansion techniques:

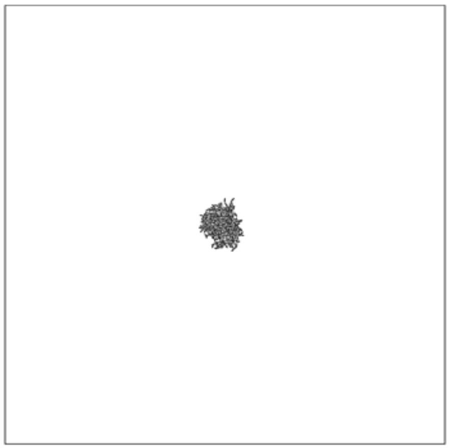
- Dynamic programming
  - Open nodes sorted by increasing c_{root}
- A* (c_{goal} = underestimate of the cost to q_g)
  - Open nodes sorted by increasing $c_{root} + c_{goal}$
- BF*
  - Open nodes sorted by increasing c_{goal} (no optimality then)
- …

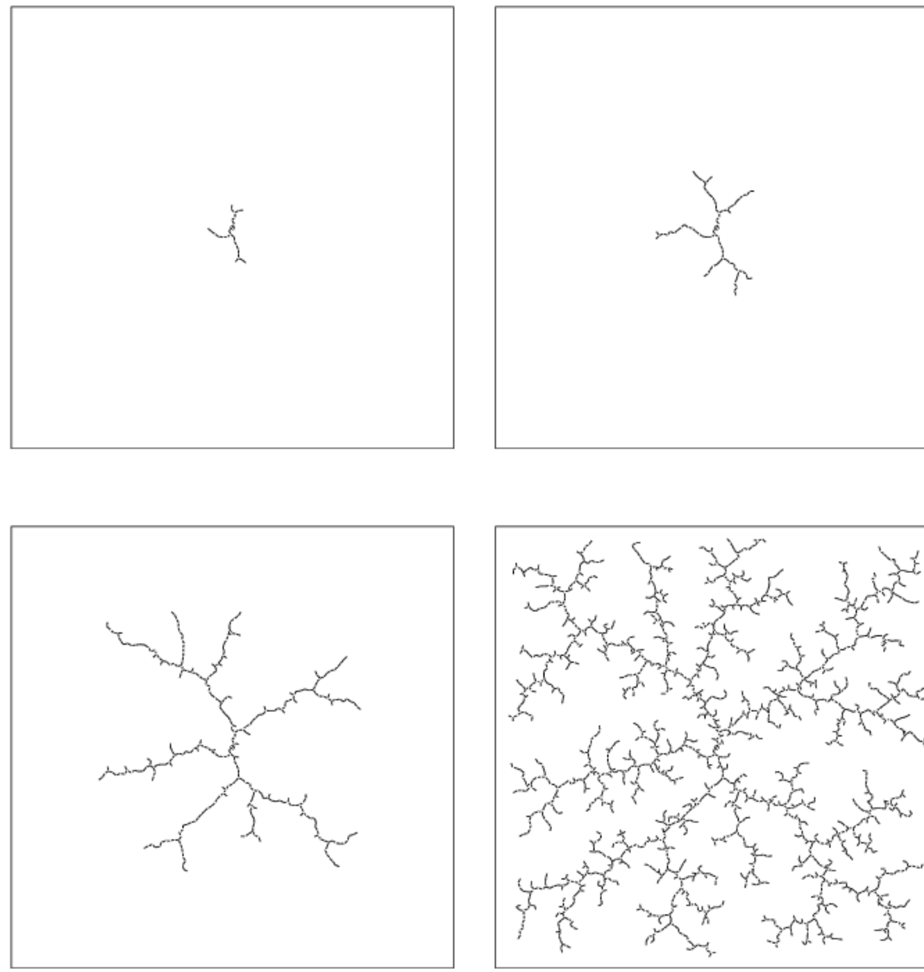
Variant: bi-directional search

Rapidly-Exploring Random Tree (RRT)

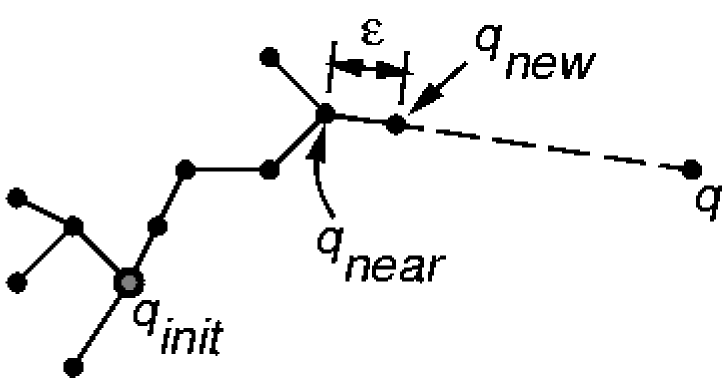
[LaValle, 98]

RRT = search tree grown from an initial state, expanded through incremental motion

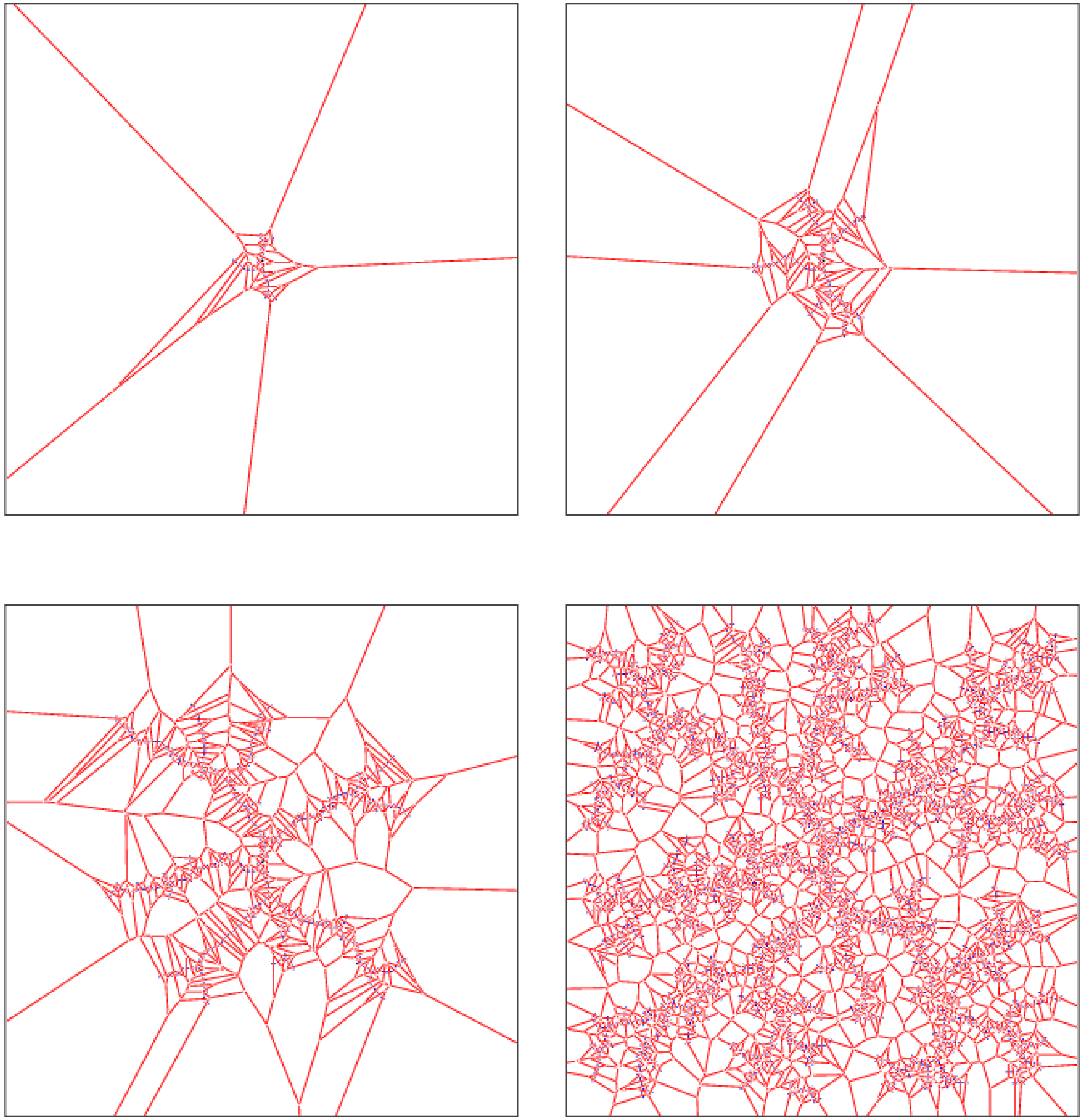




Naive random tree *vs*. RRT

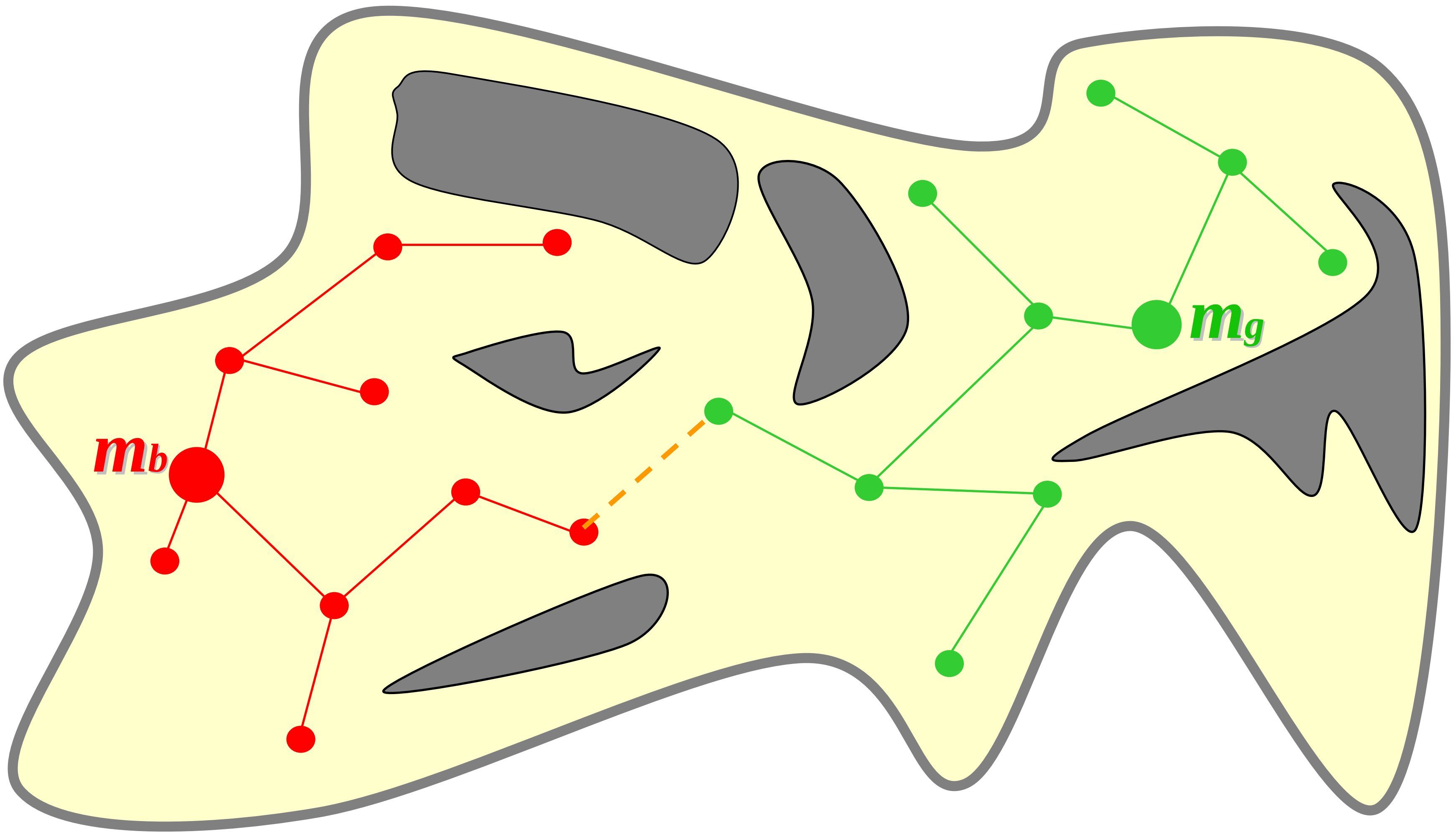


Voronoï Interpretation of RRT



Bias towards unexplored regions

Rapidly-Exploring Random Tree



RRT ' Features

Simple

Bias towards unexplored region

Eventually, uniform coverage

Probabilistic completeness

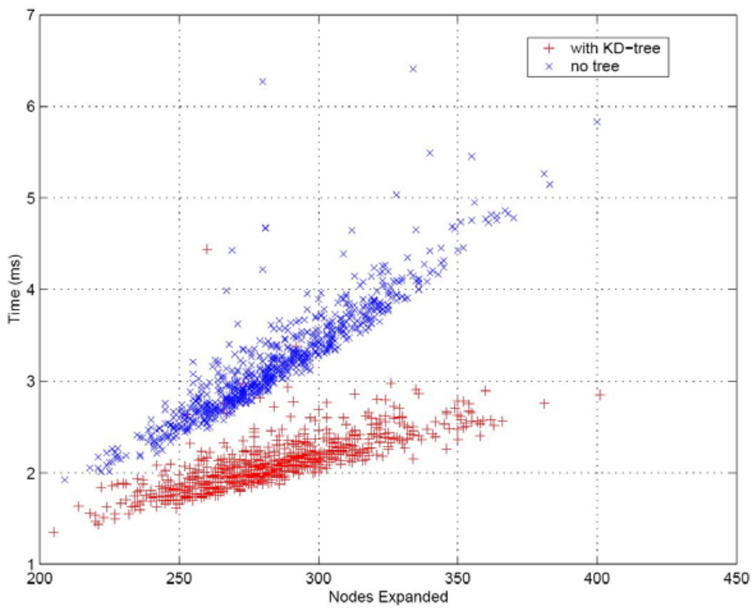
Performance depending on the metric

Rate of convergence?

Relationship to optimal paths?

Variants: single-tree vs. dual-tree

Relatively large standard deviation of planning time



Probabilistic Roadmap's Features

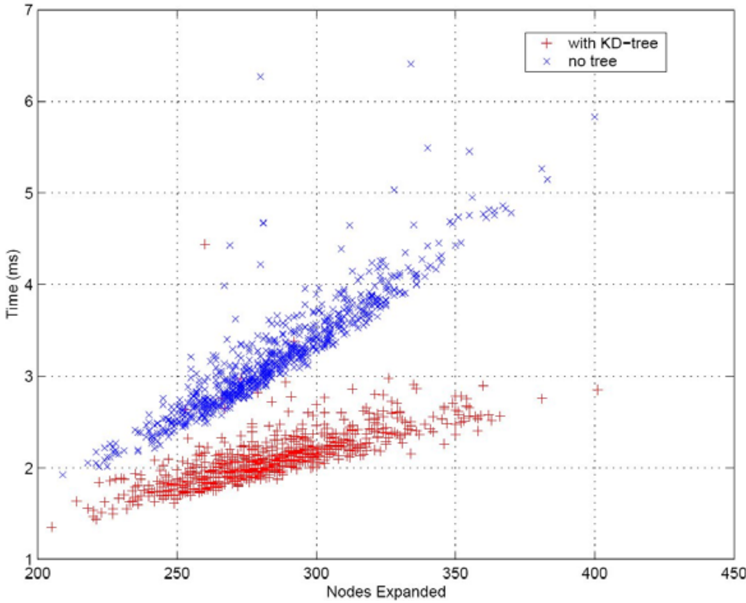
Proved to be an effective (easy to implement, fast, robust) computational framework to solve path planning problems in high-dimensional configuration spaces

Successfully applied to different motion planning problems: moving obstacles, kinematic and dynamic constraints, manipulation…

Remaining issues:

How to obtain good roadmaps?

No rigorous termination criterion when no solution is found



Ariadne's Clew Algorithm *[Ahuactzin, 94]*

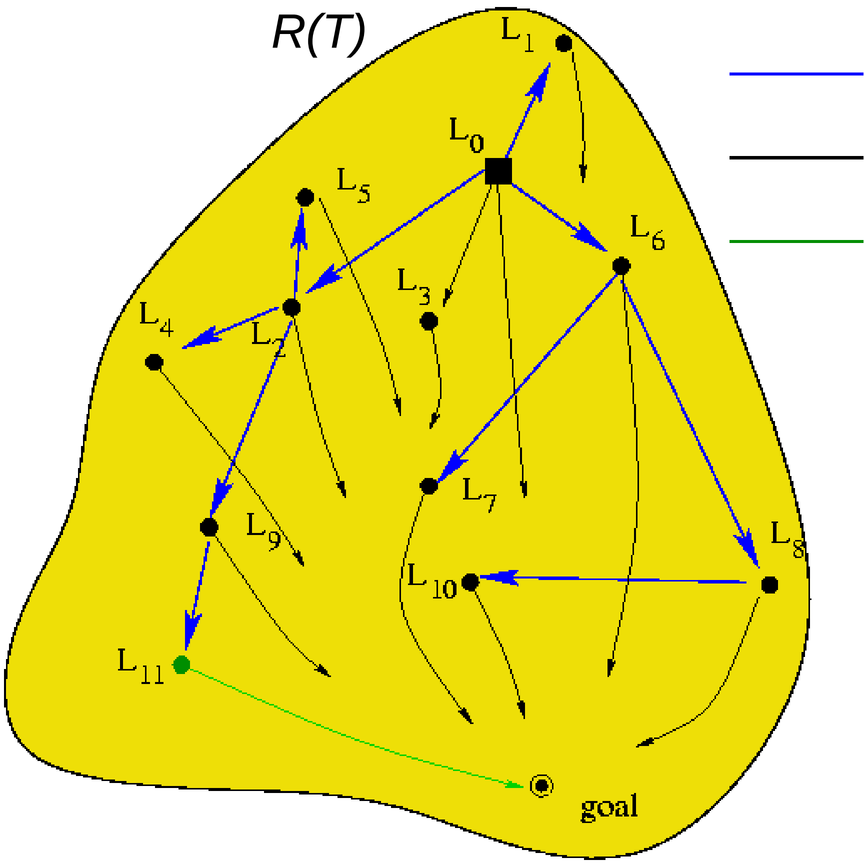
Tree *T* expanded from the start configuration

Local connecting function: *Search* (q_1, q_2)

Search defines a reachability set *R(T)*

Optimization procedure: *Explore* that selects a new node as far as possible from the other nodes of *T*

When a new node is selected, the algorithm tries to connect it to the goal configuration



Ariadne's Clew Algorithm (C 'ed)

Main property: relationship between the number of nodes *nb* and a scalar ε, *e.g*. a measure of the difficulty of the planning problem (size of a narrow passage)

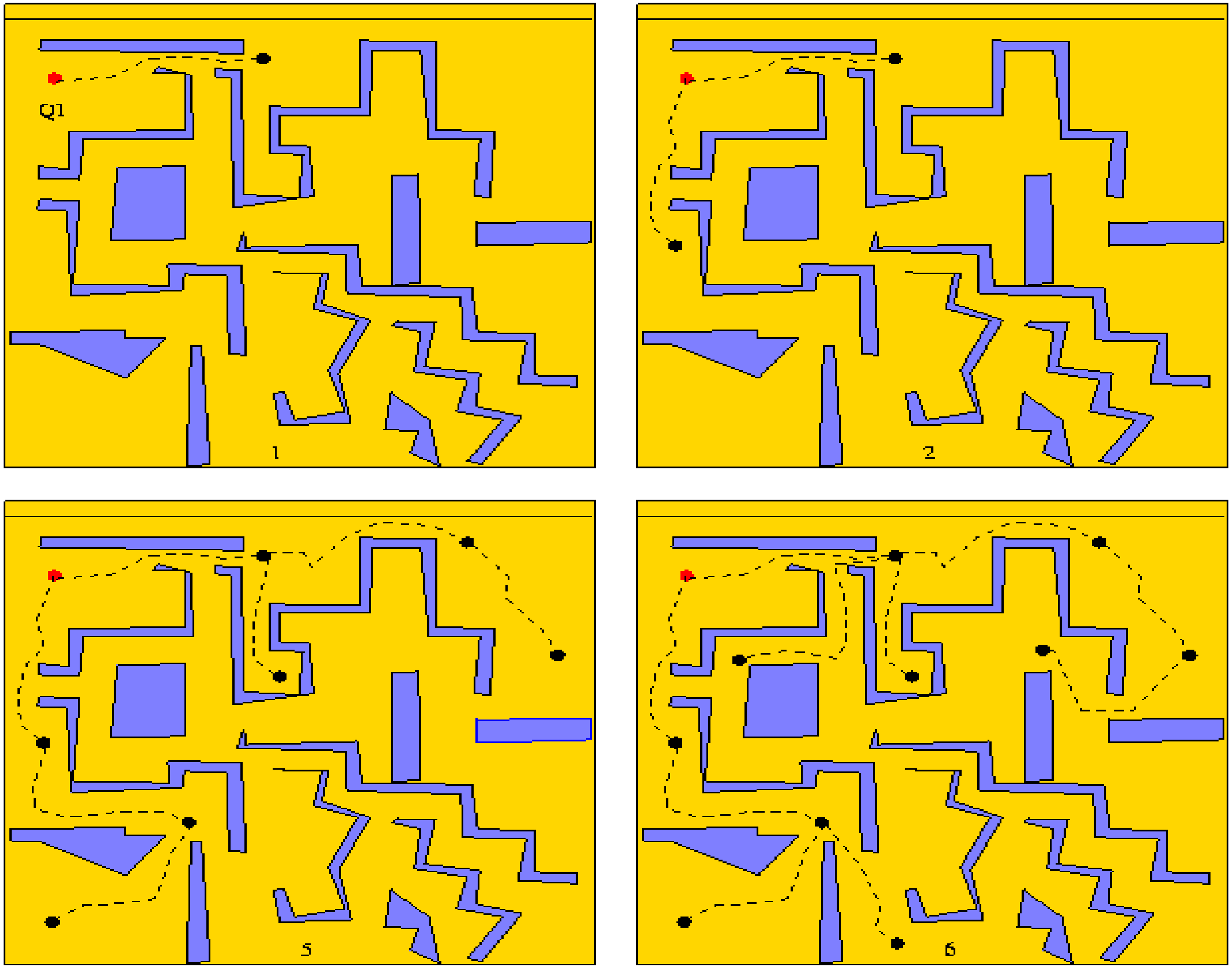
$$nb > \left\lfloor \frac{\sigma_n (size(C_{free}) + \varepsilon)^n}{J_n \left(\frac{\varepsilon}{2}\right)^n} \right\rfloor - 1 \Rightarrow d(T(nb), q_g) < \varepsilon$$

σ_n = Rogers' density, *i.e*. maximum % of *C* (of dimension *n*) that can be covered by *n*-balls

J_n = volume of a unit *n*-ball

$\Rightarrow$ Resolution completeness (and even completeness when q_g lies in a C_{free} ε -ball)

Ariadne's Clew Algorithm (C 'ed)



“Other” Methods

Navigation function

Path deformation

Navigation Function

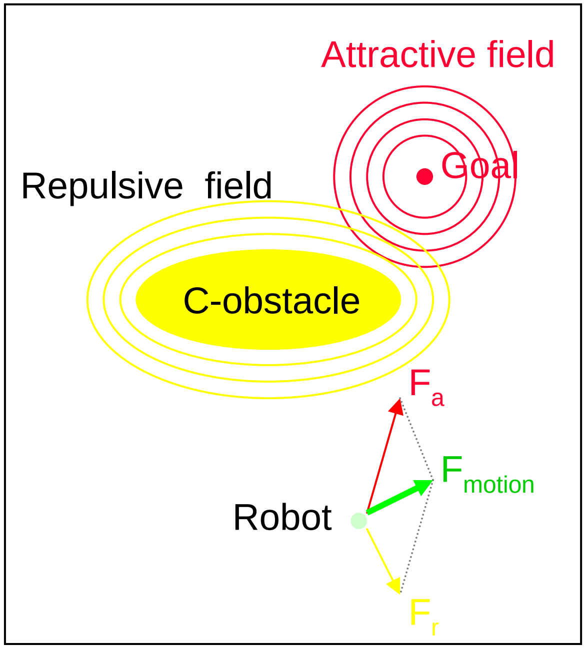
Aka Feedback motion planning

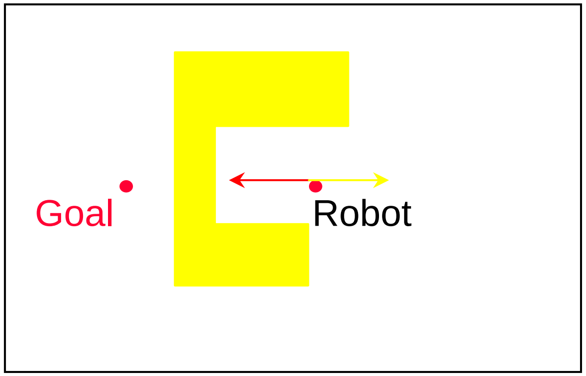
Navigation function: scalar function defined over the free configuration space

Incremental robot motion: negated gradient descent

Ideally, a navigation function should be smooth, with a global minimum at the goal, without any local minimum

Potential Field *[Khatib, 86]*



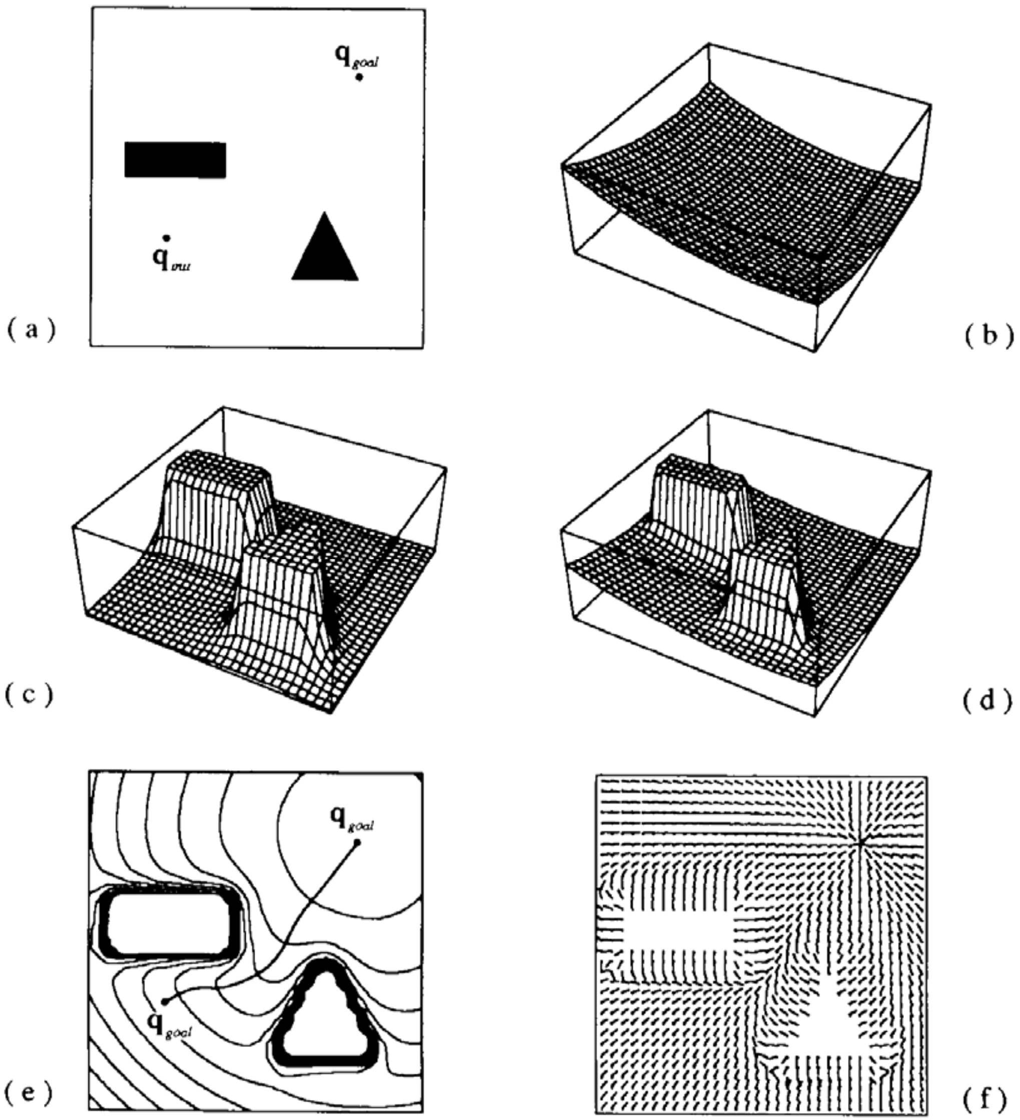


Local minimum

Force field analogy

Technique originally designed for real-time collision-avoidance

Potential Field-Based Navigation Function



$U(q) = U_{goal}(q) + U_{robstacles}(q)$

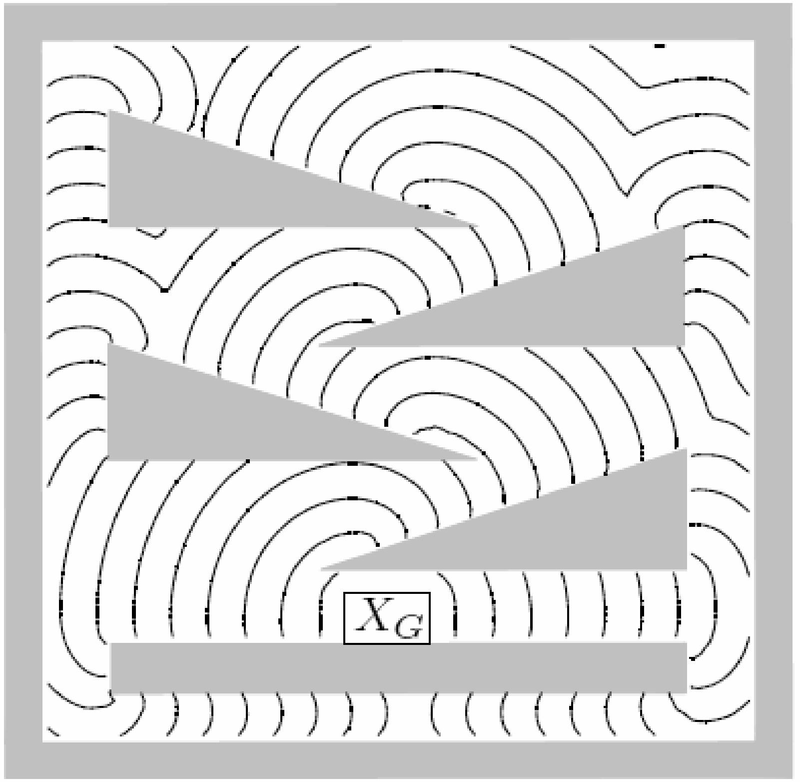
Negated gradient descent

Main issue: local minima
(how to design a good potential field?)

Optimal Navigation Function

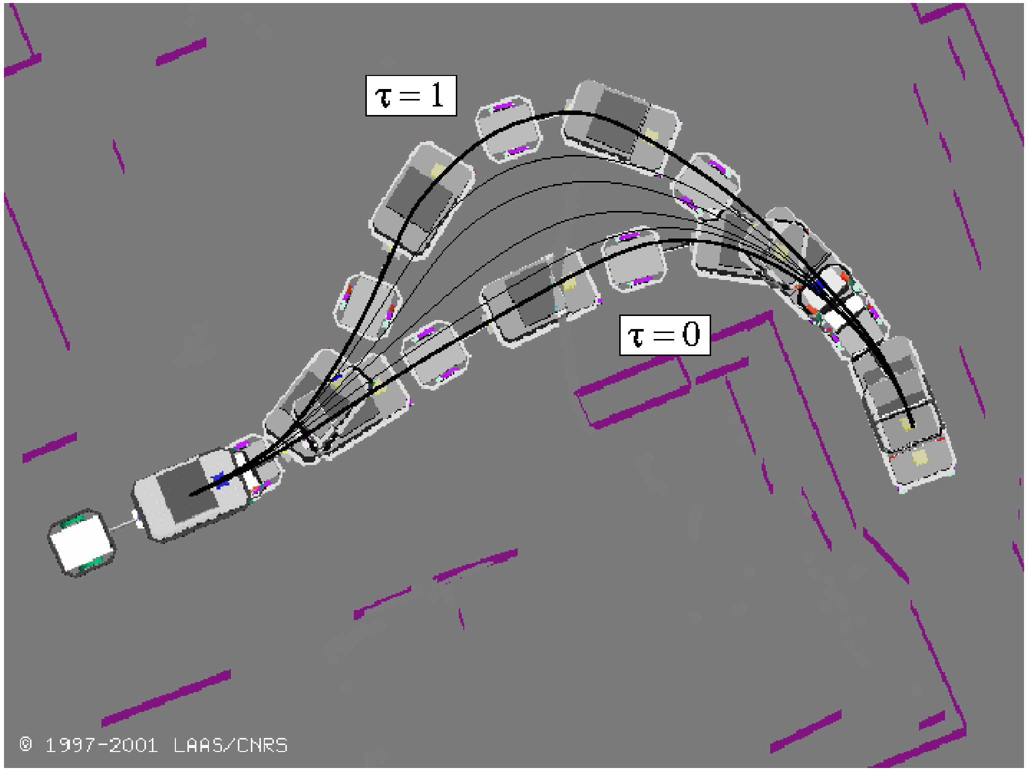
22	21	22	21	20	19	18	17	16	17
21	20							15	16
20	19							14	15
19	18	17	16	15	14	13	12	13	14
18	17	16	15	14	13	12	11	12	13
							10	11	12
							9	10	11
3	2	1	2	3			8	9	10
2	1	0	1	2			7	8	9
3	2	1	2	3	4	5	6	7	8

NF1



Path Deformation

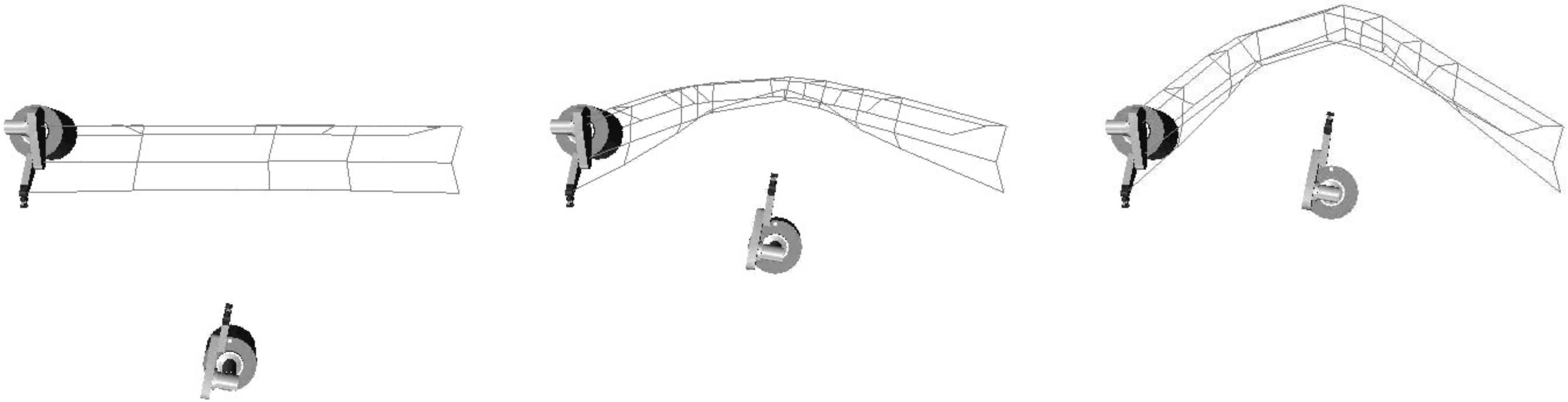
Nominal path, *on-line* deformation given updated world model



Variational aproaches *[Lamiraux 02]*

External + internal forces: elastic strip analogy *[Brock]*

Path Deformation



Loss of connectivty ⇒ local replanning