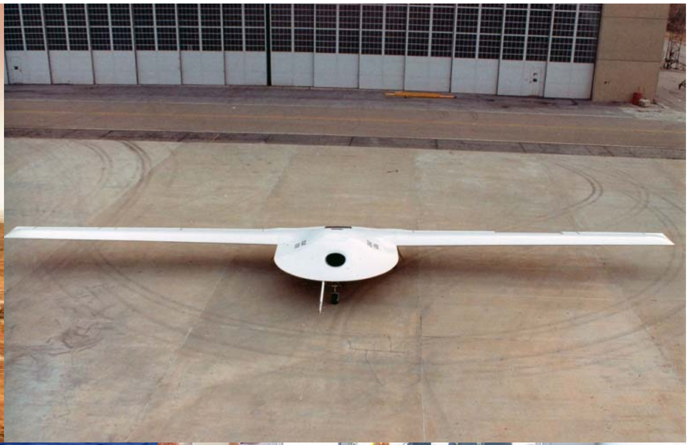
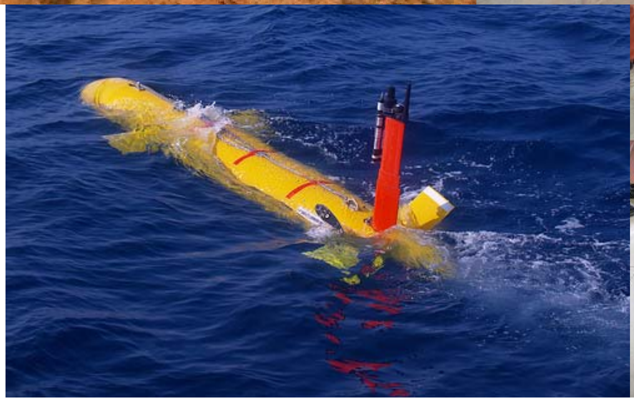
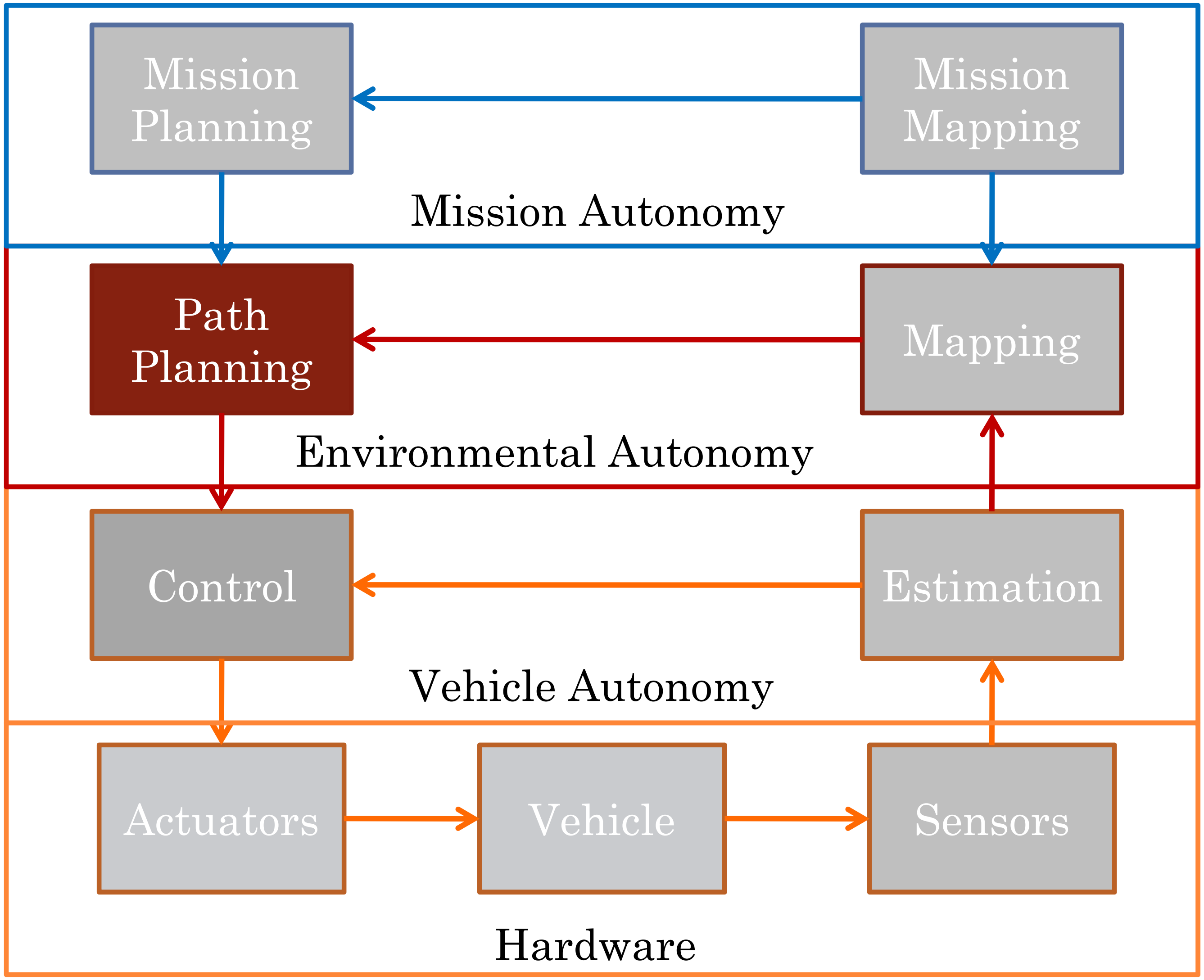


ME 597: Autonomous Mobile Robotics
Section 8 – Planning II

Prof. Steven Waslander

Components



Mission Planning
Mission Mapping
Mission Autonomy
Path Planning
Mapping
Environmental Autonomy
Control
Estimation
Vehicle Autonomy
Actuators
Vehicle
Sensors
Hardware

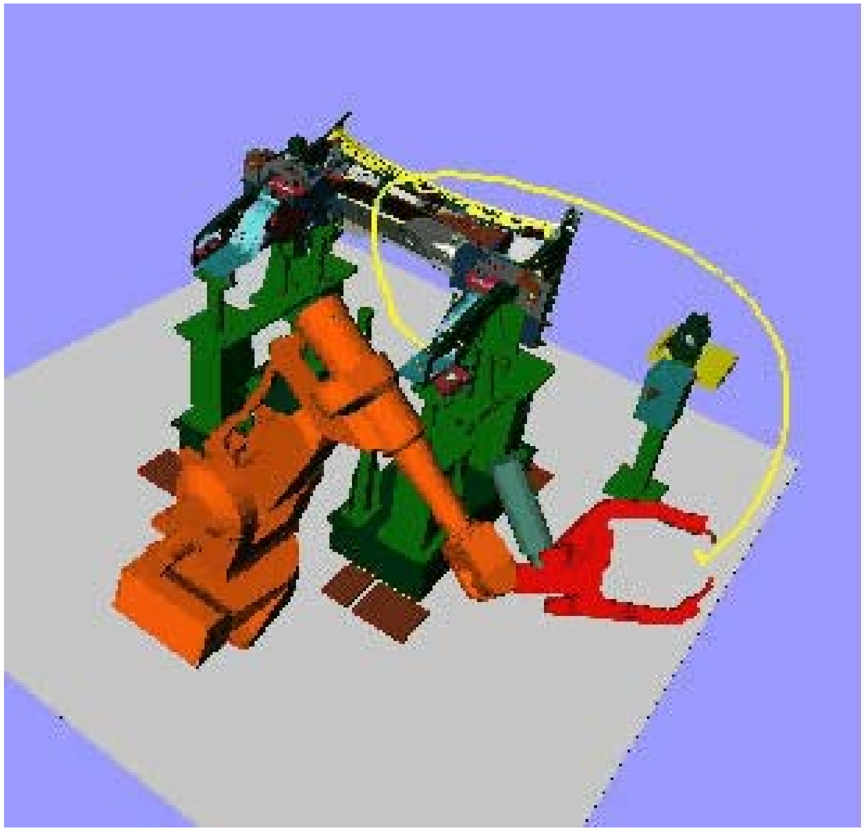
OUTLINE

- Probabilistic Graph Based Planning
  - Complex Planning Examples
  - Probabilistic Roadmaps
  - Collision Detection
  - Sampling Strategies
  - Nonholonomic Path Planning

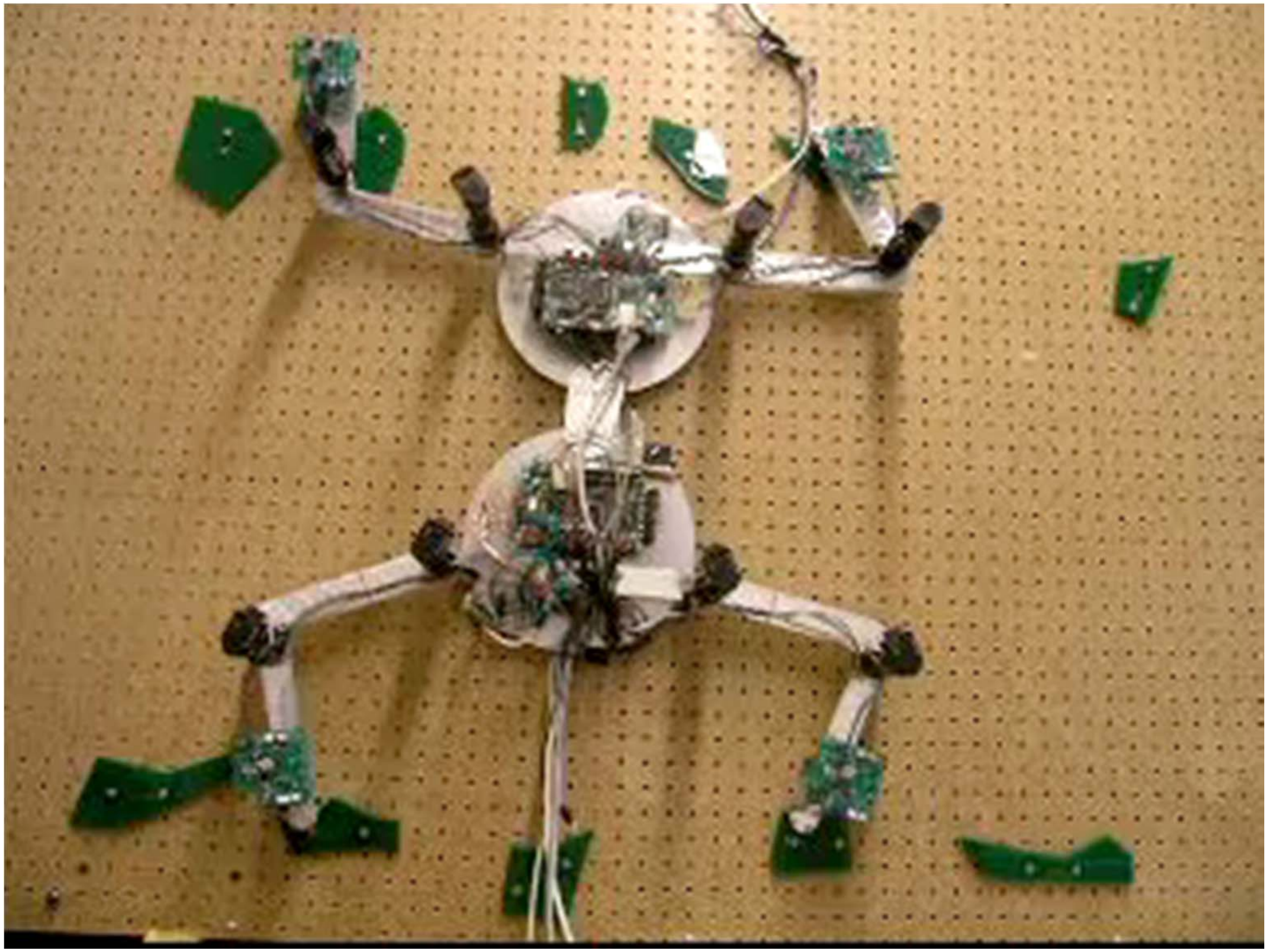
- Most slides courtesy of J.-C. Latombe

Planning in Complex Environments

- The complexity of real-world planning problems can overwhelm all the methods described so far
  - Industrial Robotics



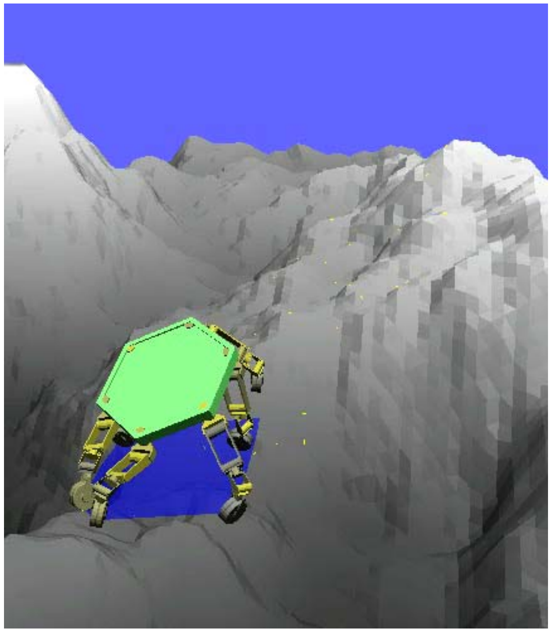
Planning in Complex Environments



DARPA ROBOTICS CHALLENGE

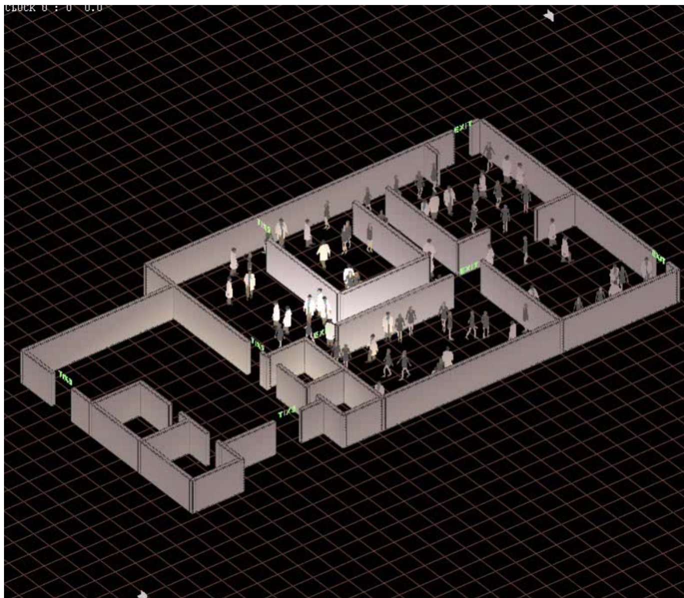
Planning in Complex Environments

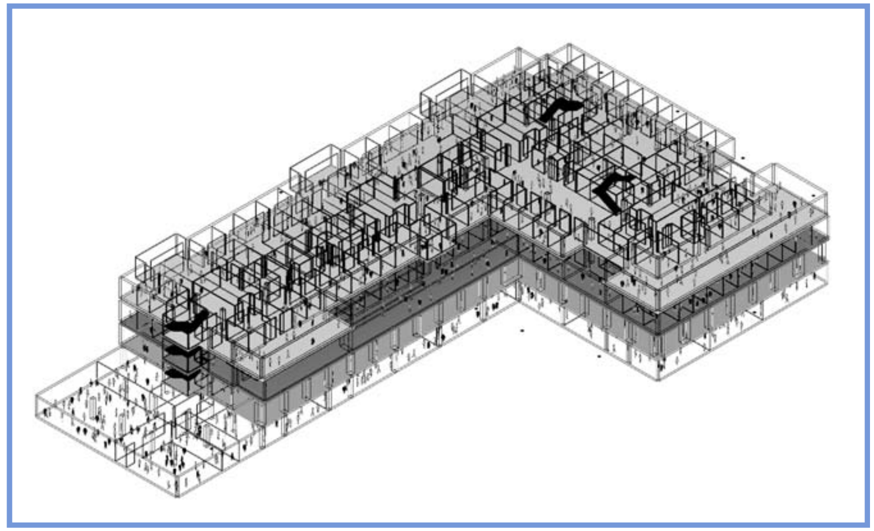
- NASA Athlete

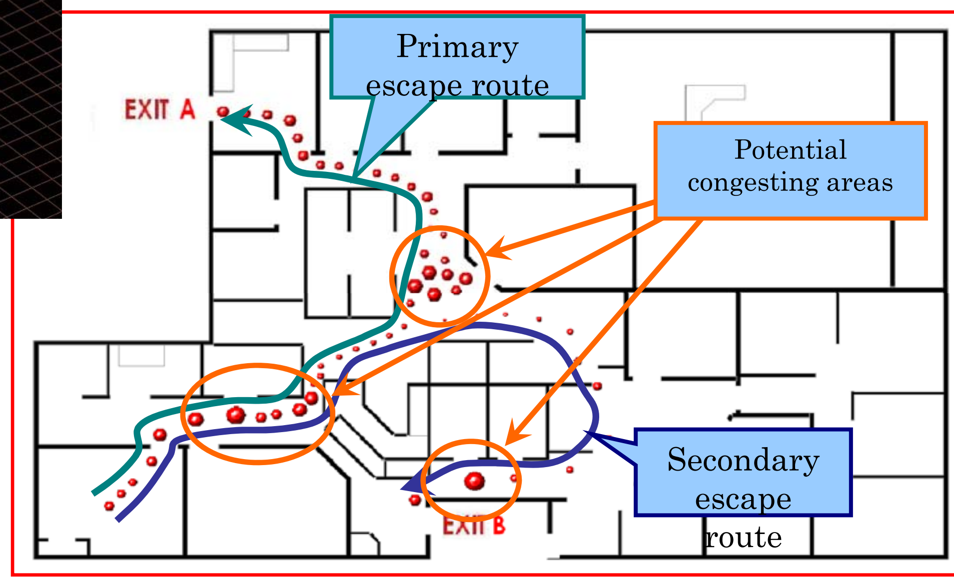


Planning in Complex Environments

- 3D Path Planning for egress





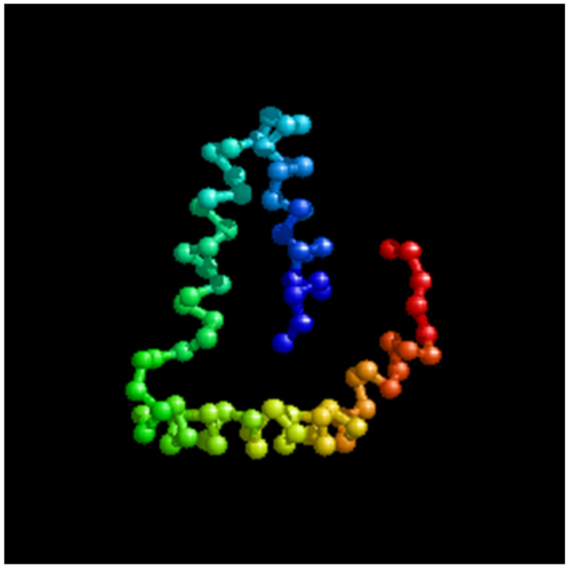


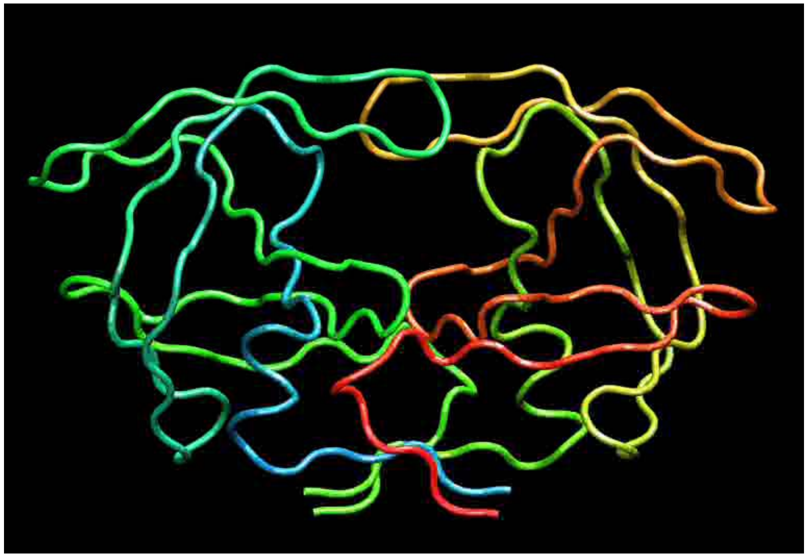
Planning in Complex Environments

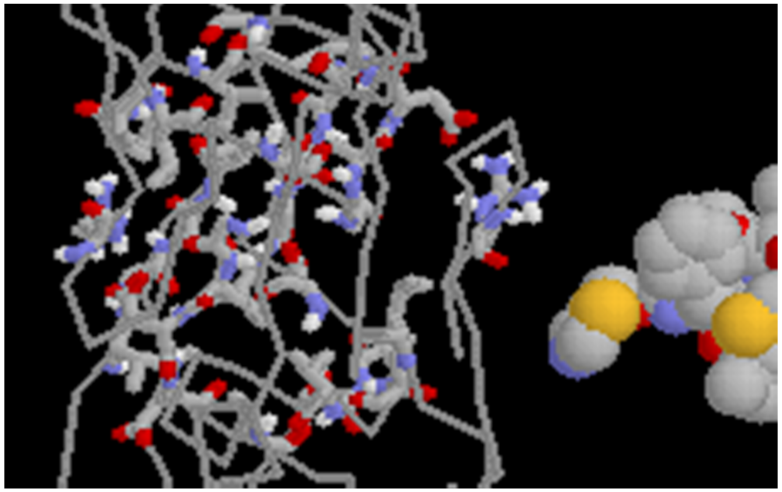
- Transport of A380 Sections through small French villages.

PLANNING IN COMPLEX ENVIRONMENTS

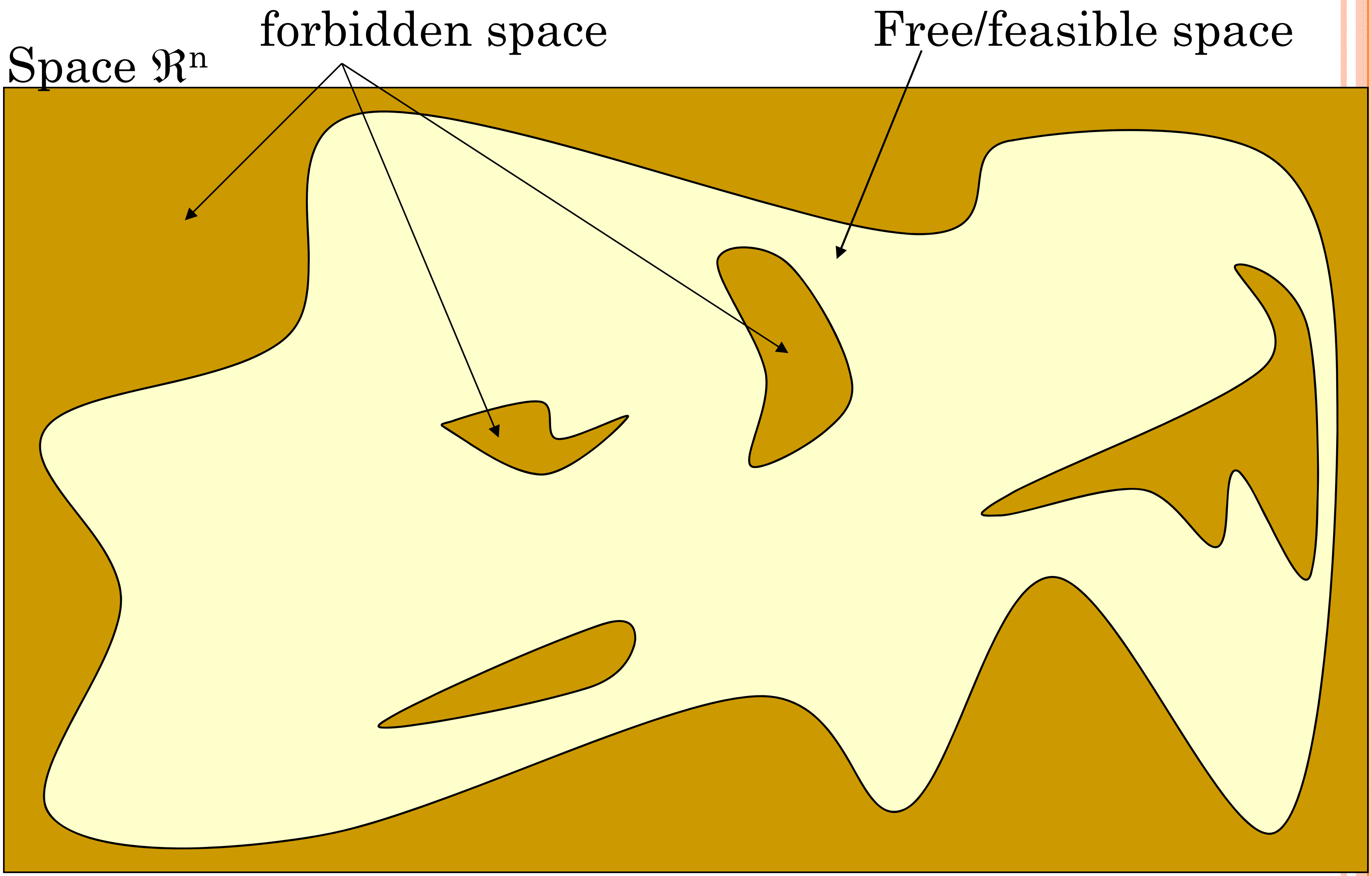
- Simulation of Protein Folding





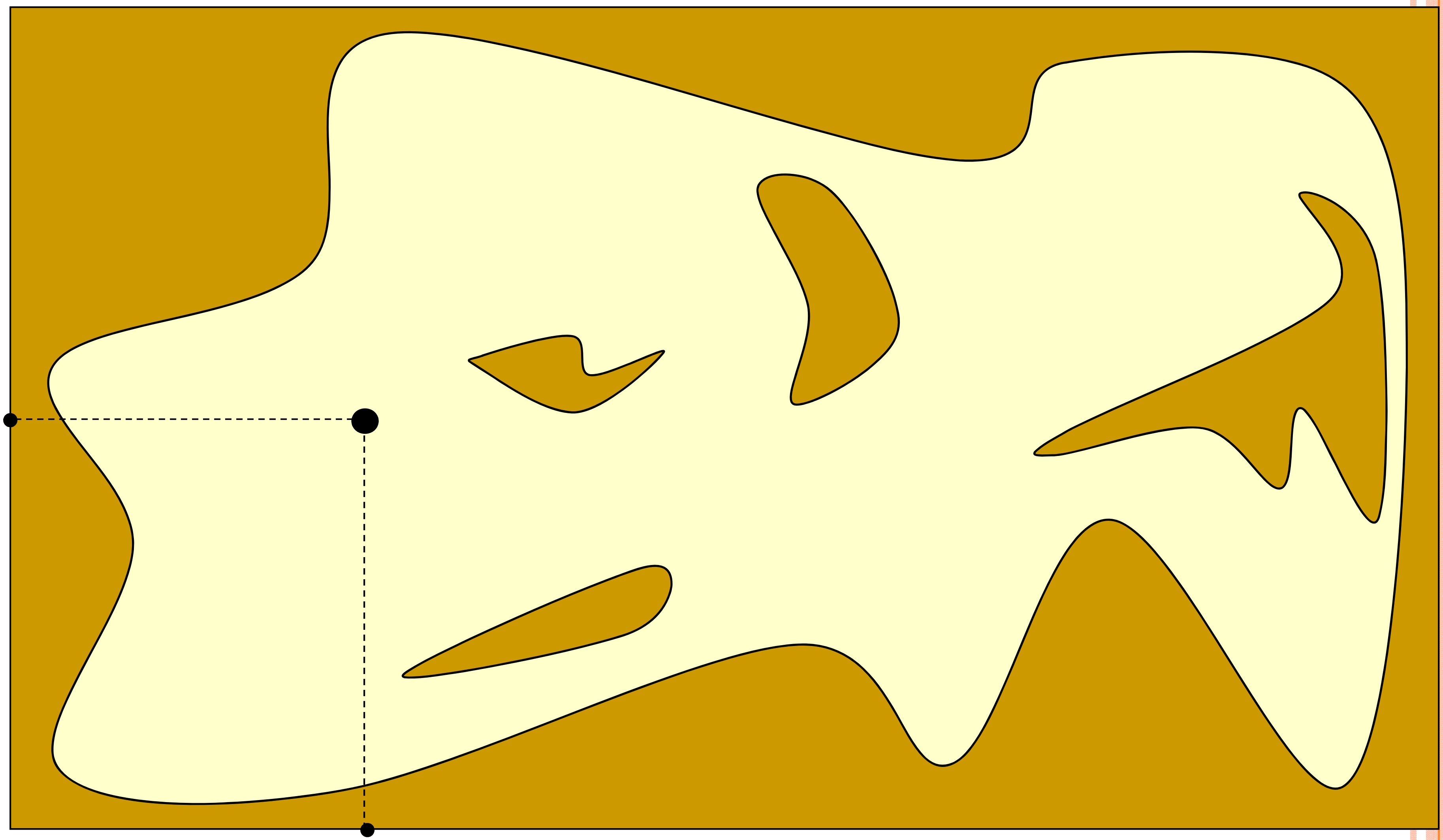


Probabilistic Roadmap (PRM)



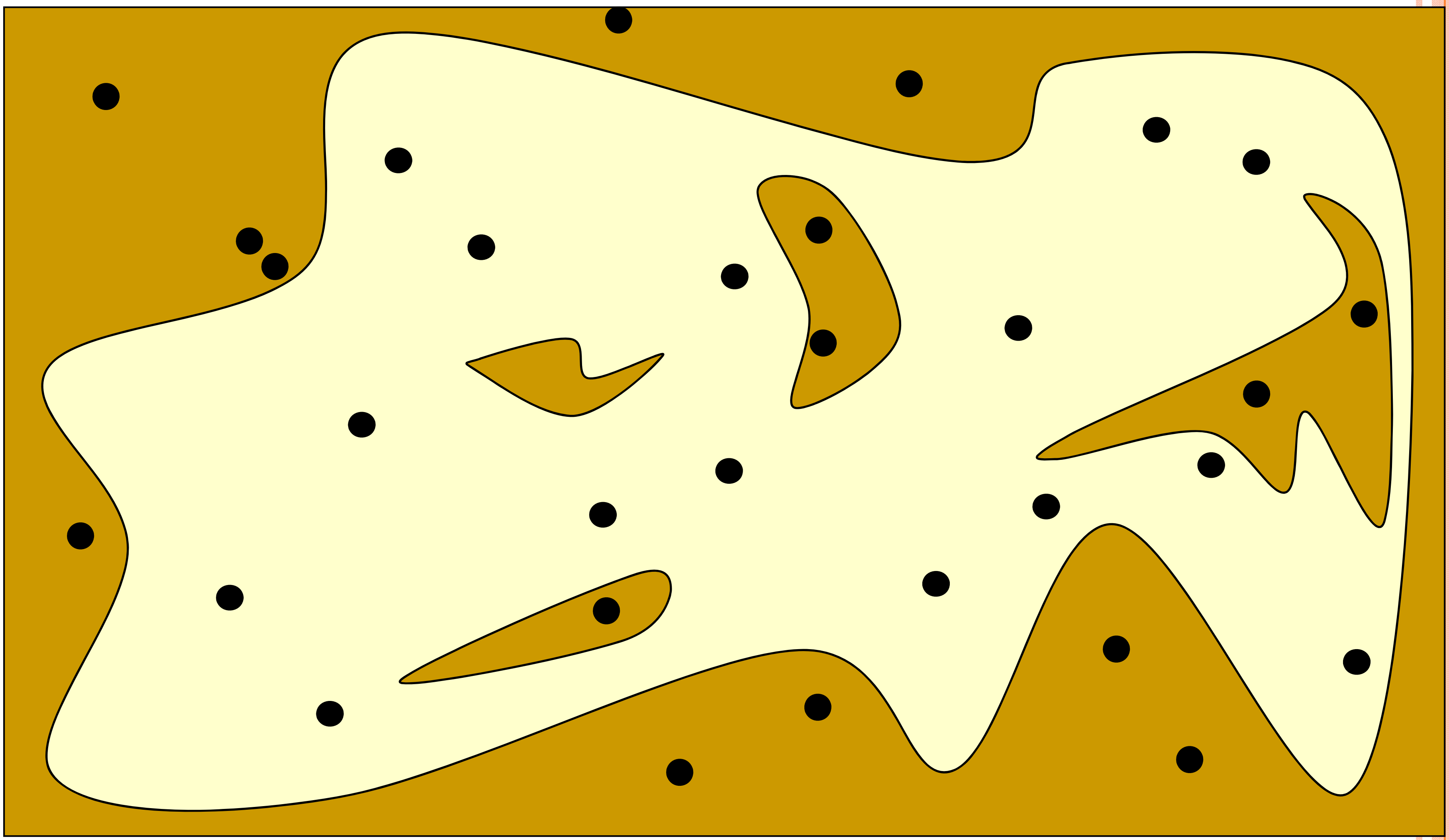
Probabilistic Roadmap (PRM)

Configurations are sampled by picking coordinates at random



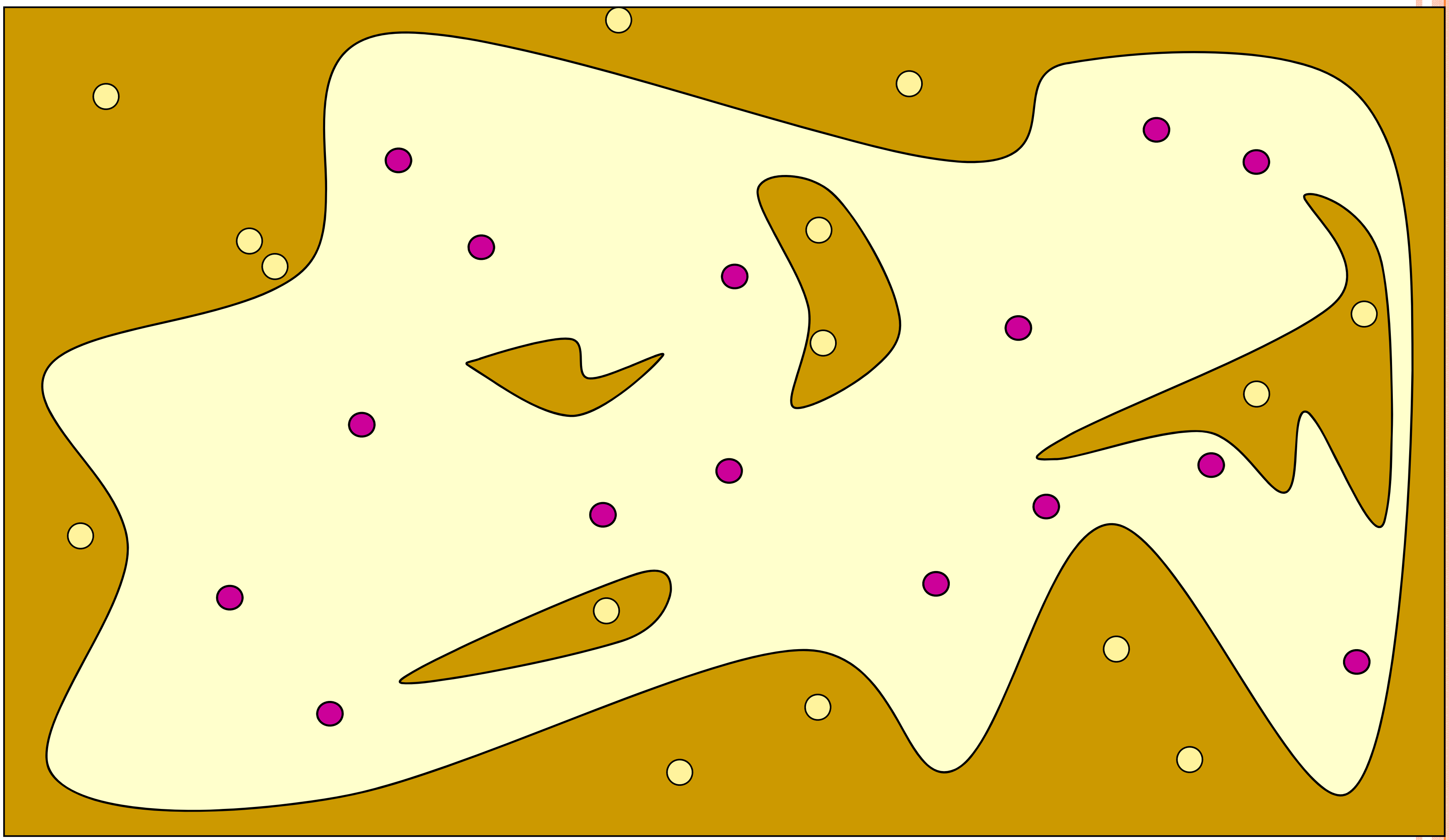
PROBABILISTIC ROADMAP (PRM)

Configurations are sampled by picking coordinates at random



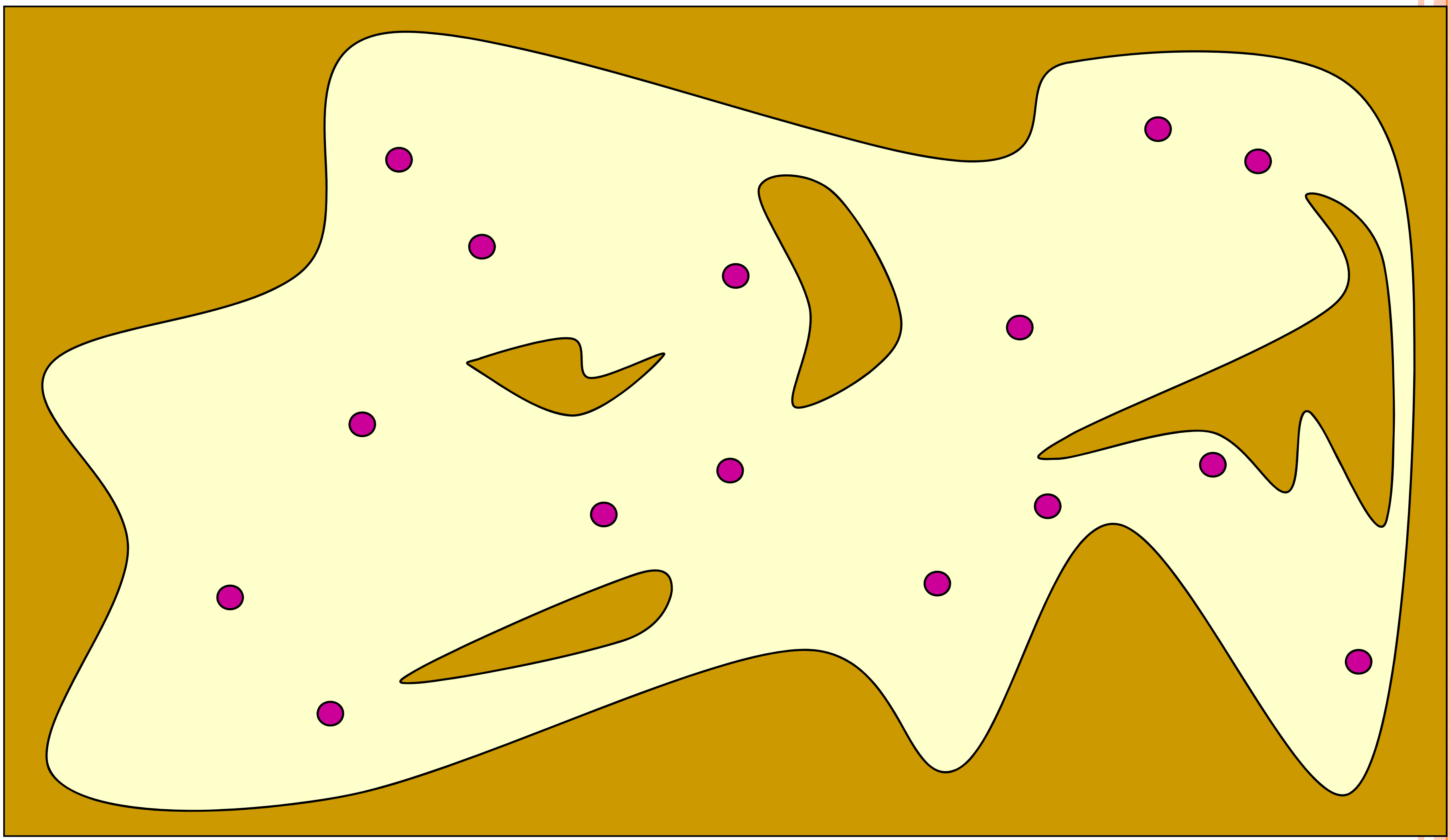
Probabilistic Roadmap (PRM)

Sampled configurations are tested for collision



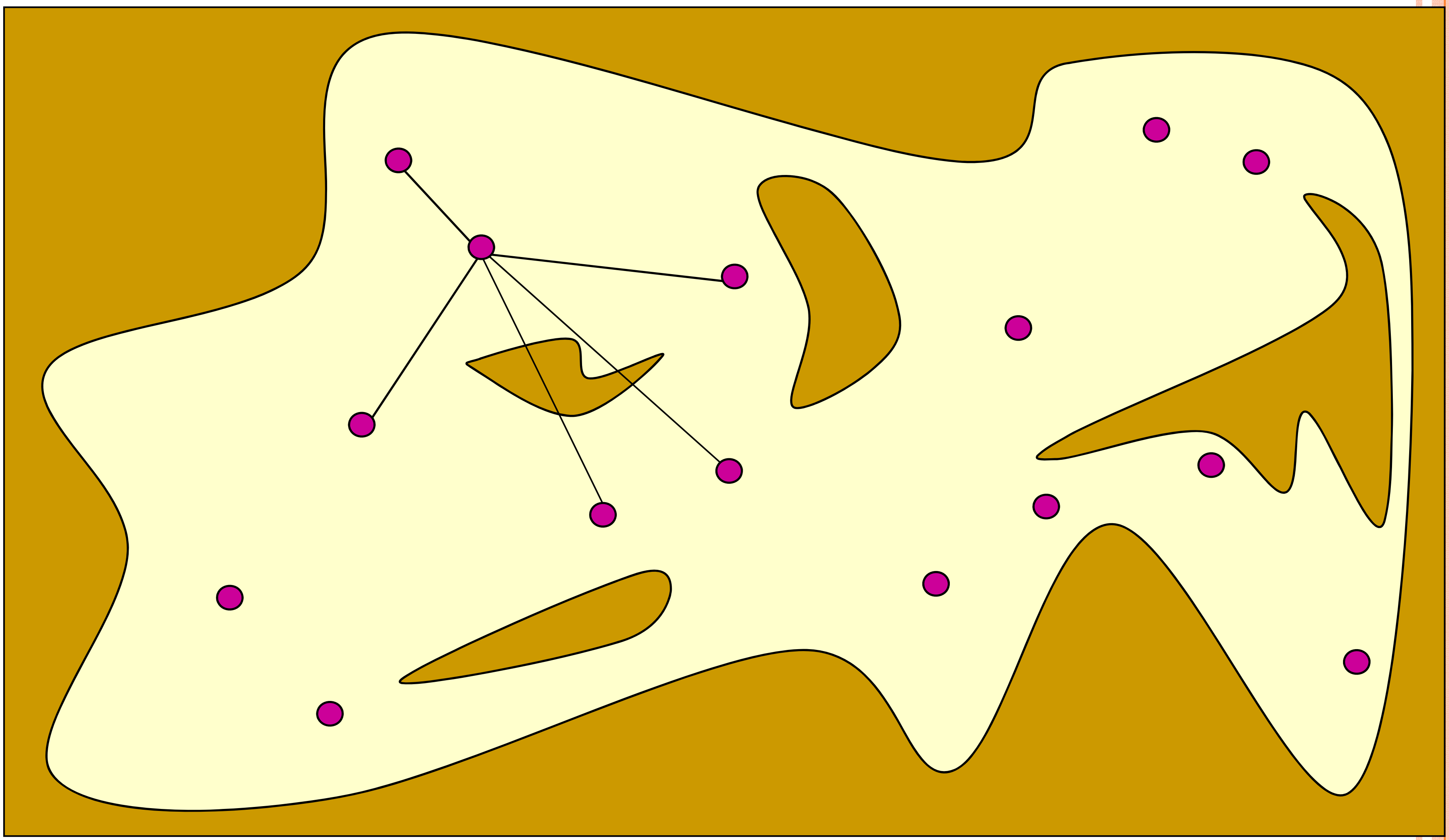
Probabilistic Roadmap (PRM)

The collision-free configurations are retained as milestones



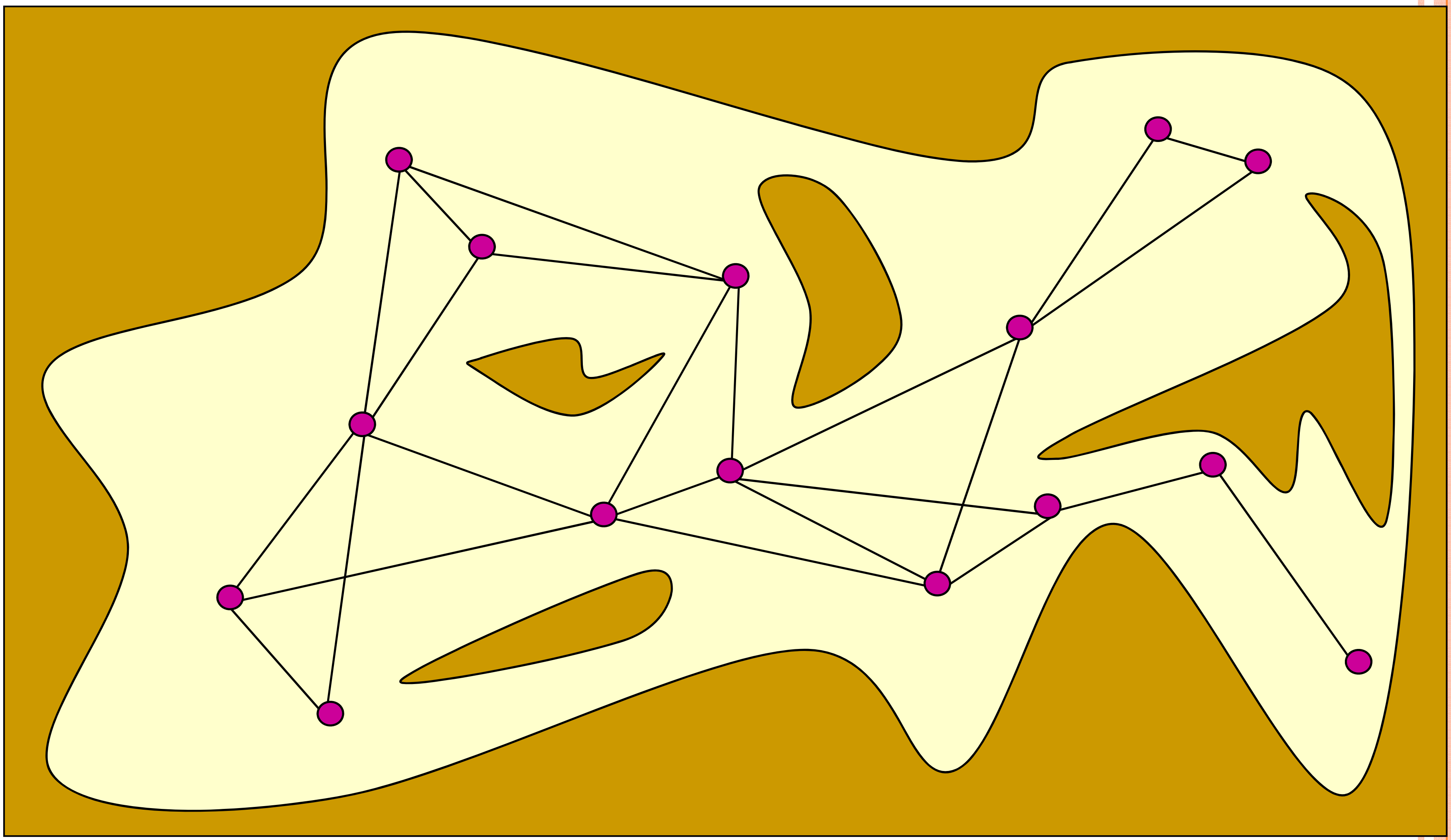
Probabilistic Roadmap (PRM)

Each milestone is linked by straight paths to its nearest neighbors



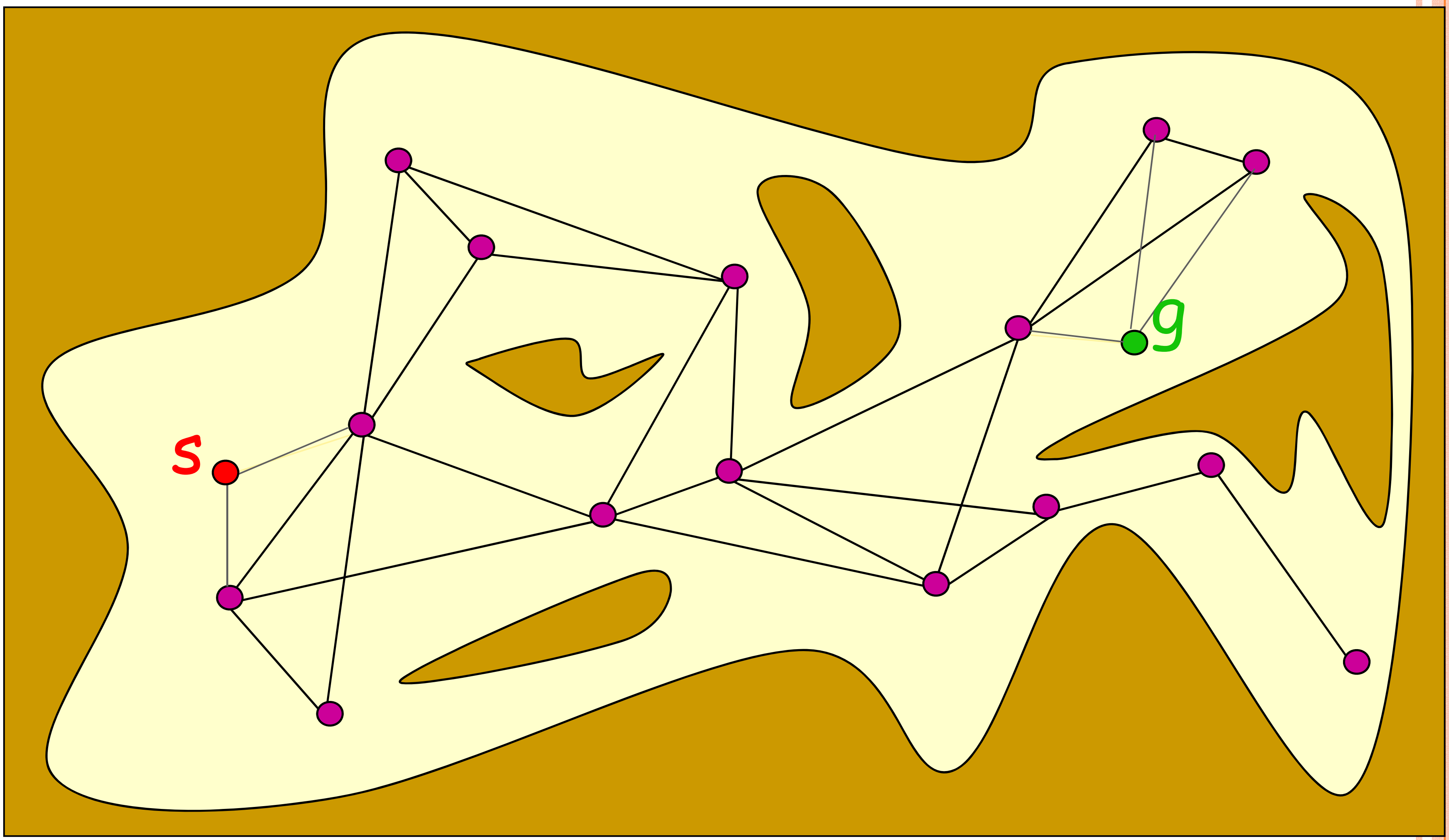
Probabilistic Roadmap (PRM)

The collision-free links are retained as local paths to form the PRM



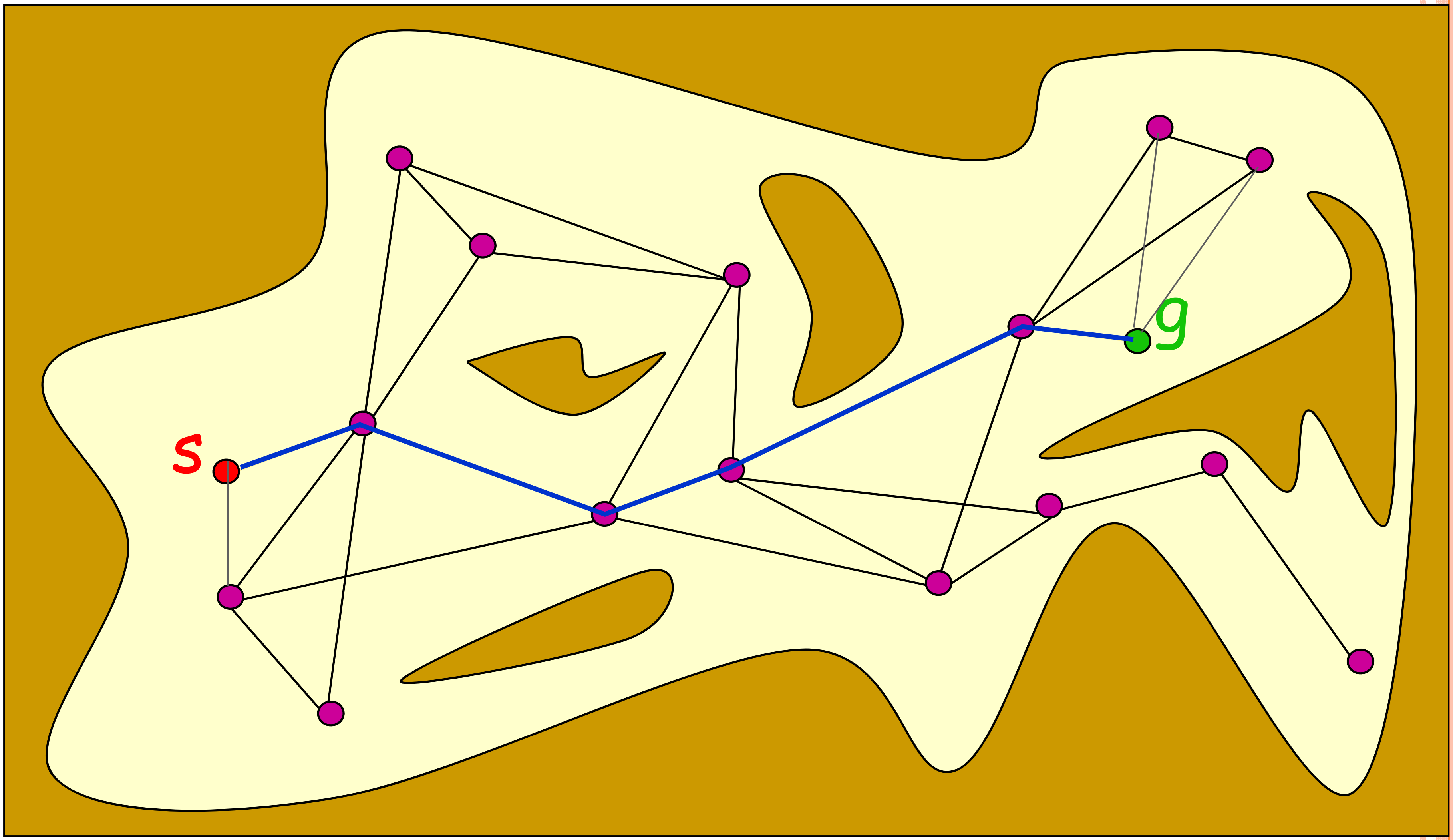
Probabilistic Roadmap (PRM)

The start and goal configurations are included as milestones



Probabilistic Roadmap (PRM)

The PRM is searched for a path from s to g



Multi- vs. Single-Query PRMs

- Multi-query roadmaps
  - Pre-compute roadmap
  - Re-use roadmap for answering queries

- Single-query roadmaps
  - Compute a roadmap from scratch for each new query

PRM Algorithm

1. Initialize the roadmap R with two nodes, s and g
2. Repeat:
   a. Sample a configuration q from C with probability p
   b. If $q \in F$ then add q as a new milestone of R
   c. For milestones v in R such that $v \neq q$ do

      If path $(q,v) \in F$ then add (q,v) as a new edge of R

   Until s and g are in the same connected component of R or R contains $N+2$ nodes
3. If s and g are in the same connected component of R then

   Return a path between them
4. Else

   Return no path

Requirements of PRM Planning

- Checking sampled configurations and connections between samples for collision can be done efficiently.
  - Hierarchical collision detection

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

Why PRMs Work

- By abstracting full configuration space to a graph representation, the PRM greatly reduces the search space for a feasible path

- Only effective if graph connects desired start and end goals
  - Dictated by the ability to find milestones in narrow passages and connect them to the rest of the roadmap

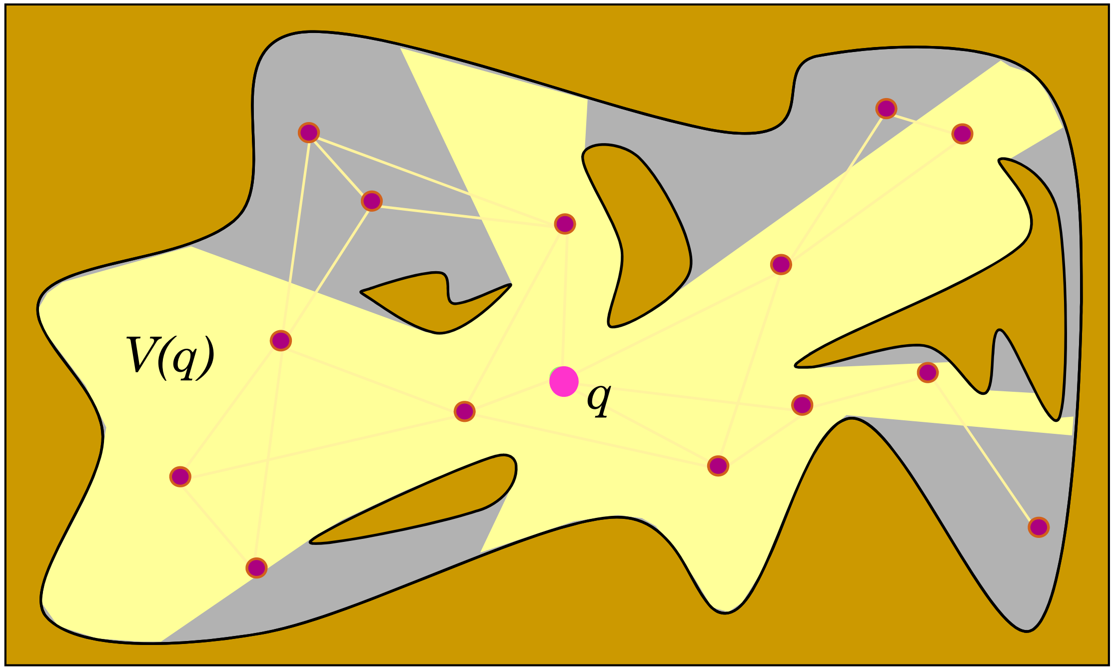
VISIBILITY IN F

- Two configurations *q* and *q'* see each other if path

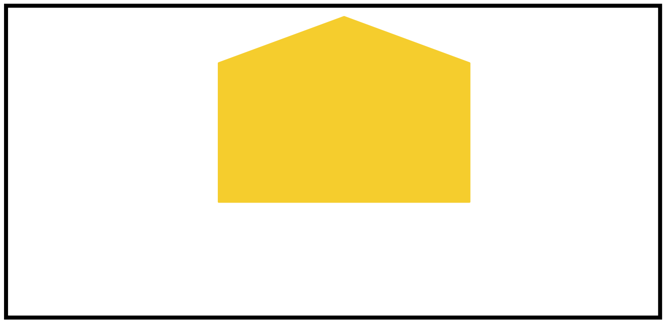
$$(q,q') \in F$$

- The visibility set of *q* is

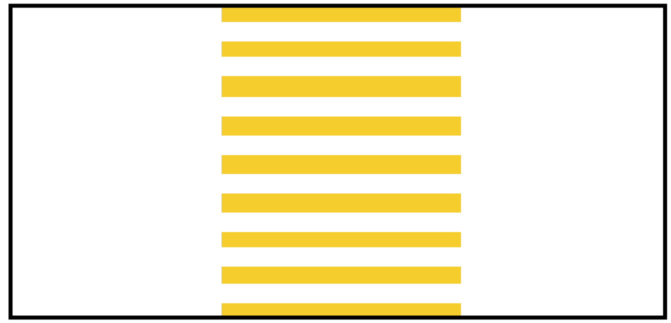
$$V(q) = \{q' \mid (q,q') \in F\}$$



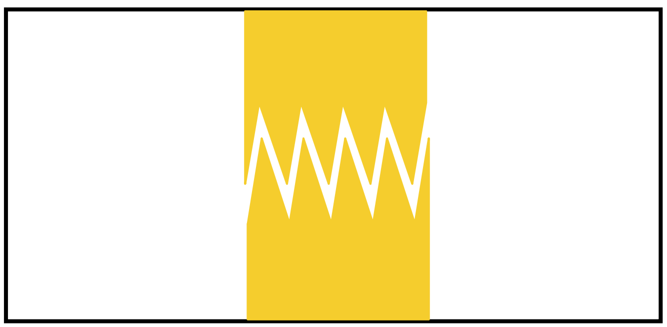
EXPANSIVENESS



Thanks to the wide passage at the bottom this space favorably expansive



Many narrow passages might be better than a single one

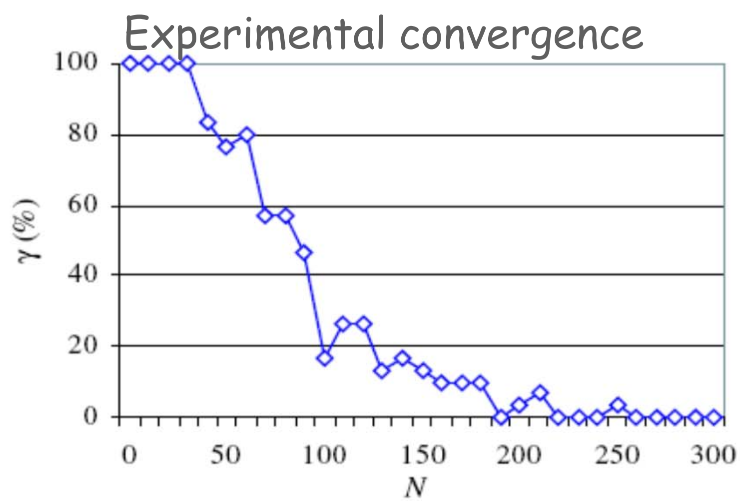


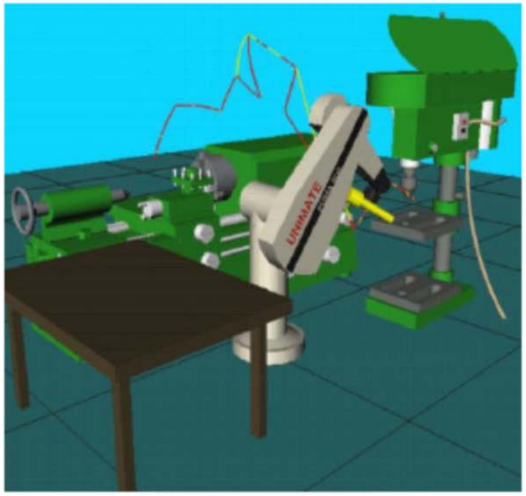
This space's expansiveness is worse than if the passage was straight

A convex set is maximally expansive.

Solutions will be Found

- It is possible to prove that:
  - With probability converging to 1 exponentially in the number of milestones
  - A feasible path will be found if one exists
  - Requires formal definition of expansiveness



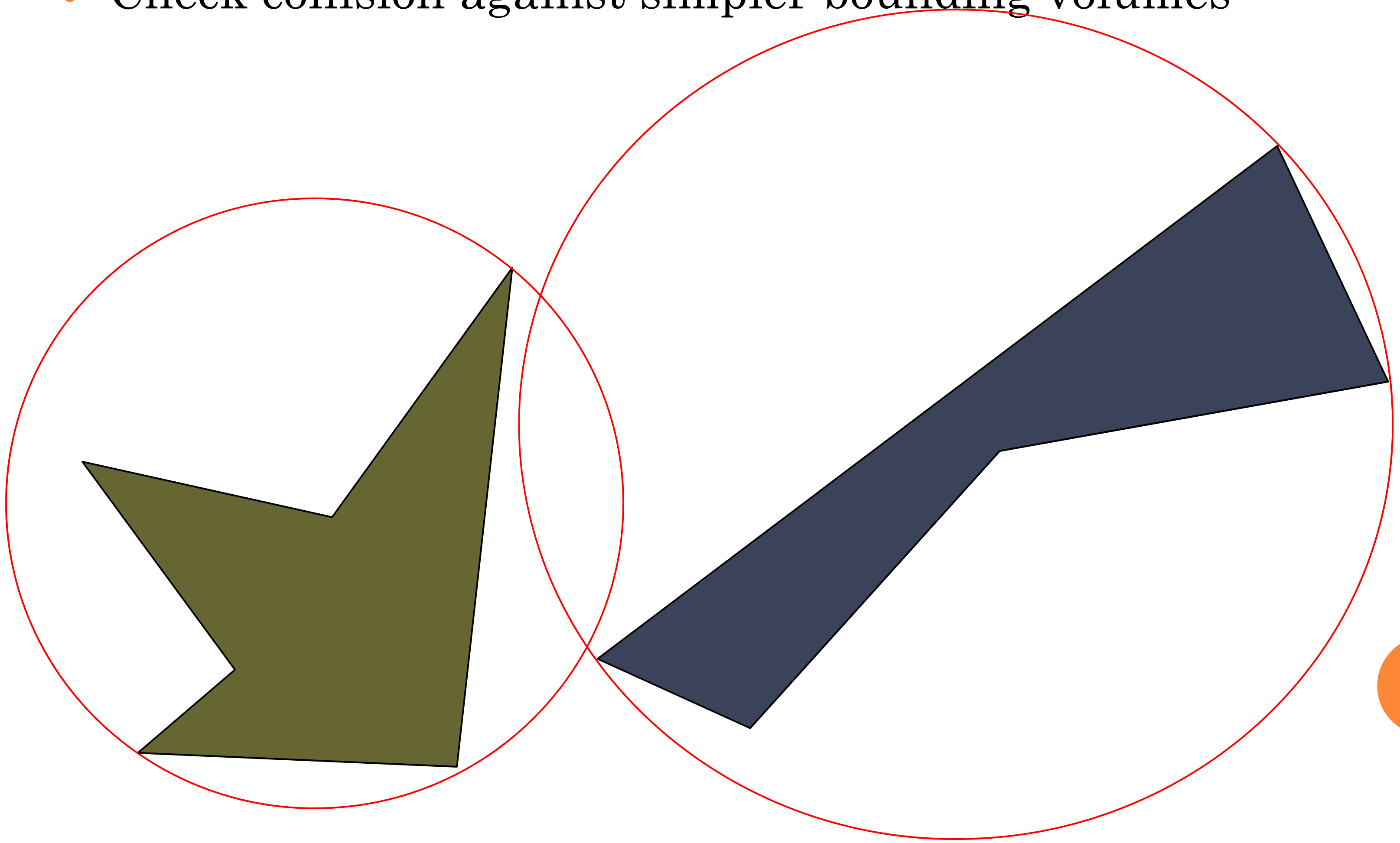


Why PRMs Work

- In practice, most planning problems result in favourably expansive configuration spaces
  - Even though constraints are challenging, nonlinear, high dimensional
  - Straight line connection of configurations works
- Benefits depend highly on two key technologies
  - Fast collision checking along paths
  - Fruitful sampling of configuration space to generate connected roadmap

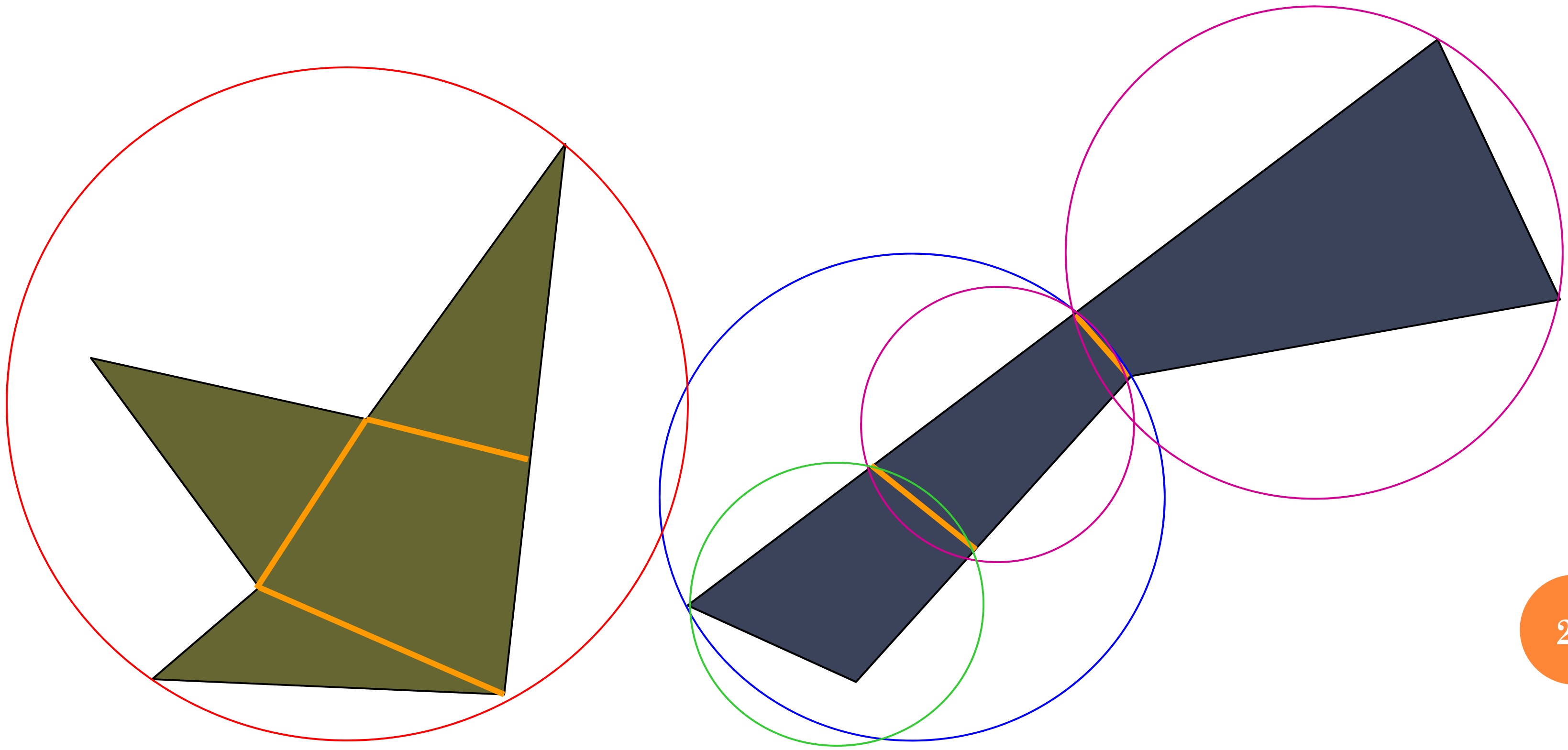
Collision Checking

- Bounding Volume Hierarchy Method
  - Enclose objects into bounding volumes
  - Check collision against simpler bounding volumes



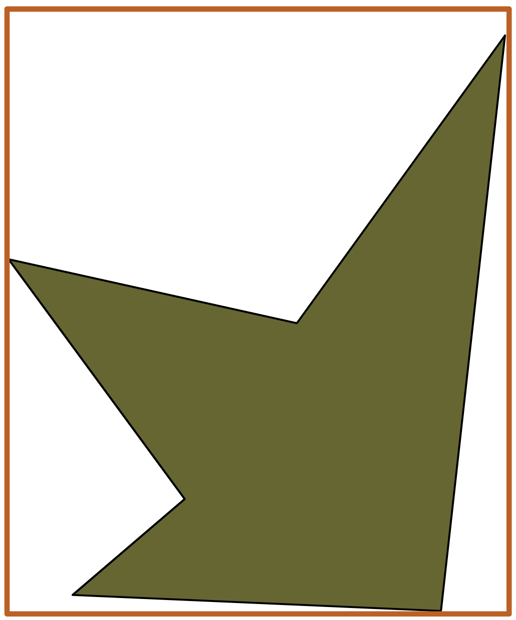
Bounding Volume Hierarchy Method

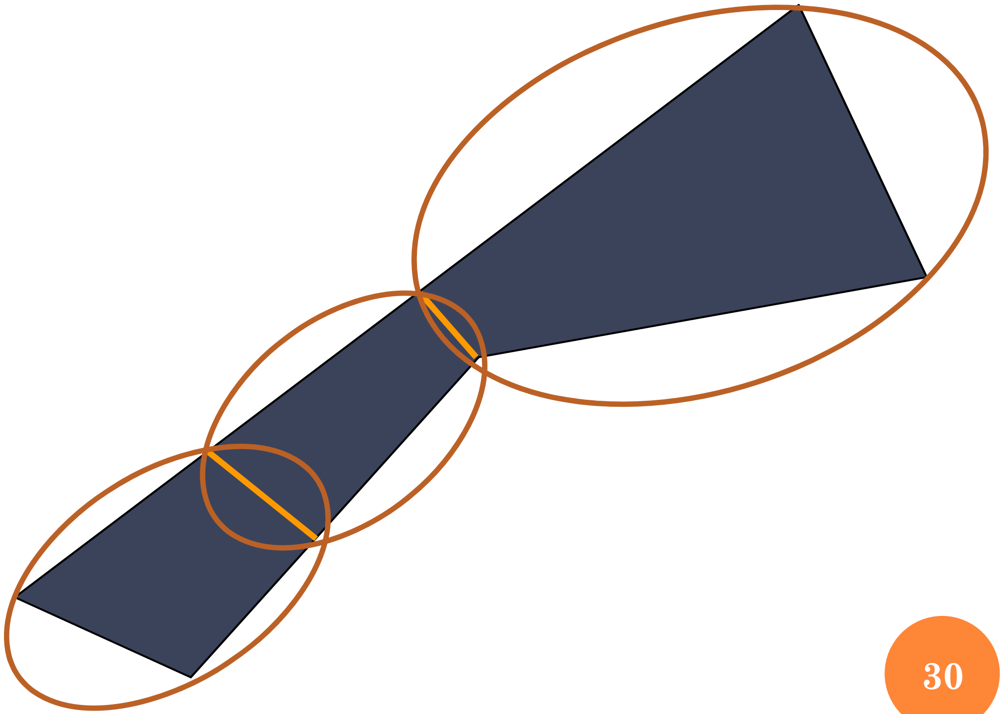
- Bounding Volume Hierarchy Method
  - If collision with bounding object occurs
    - Split object into pieces and create tighter bounds
    - Check collision against tighter bounding volumes



Bounding Volume Hierarchy Method

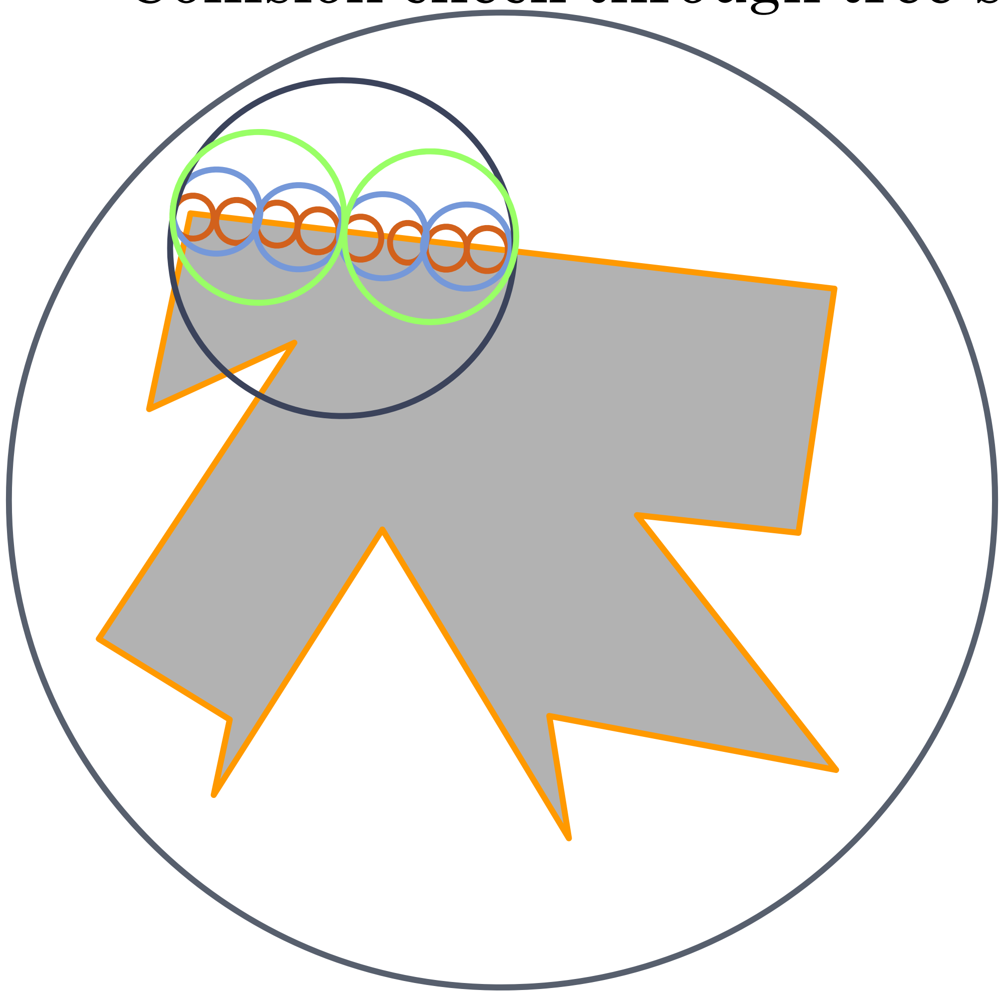
- Bounding Volume Hierarchy Method
  - Boxes, ellipses can also be used
    - Utility depends on shape, simplicity of distance calculation

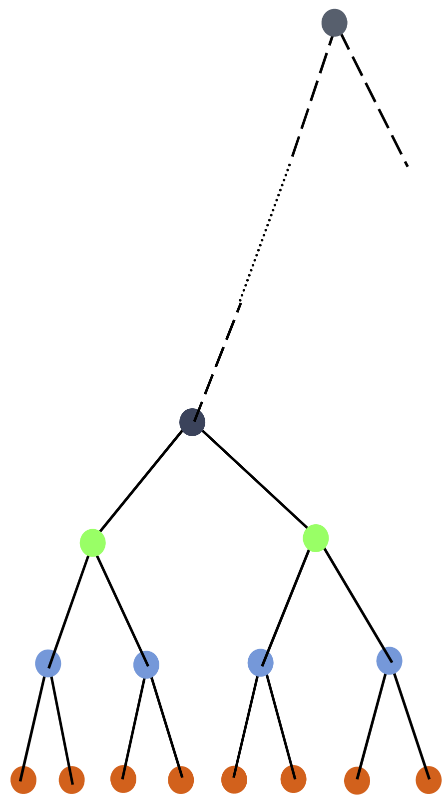




Bounding Volume Hierarchy Method

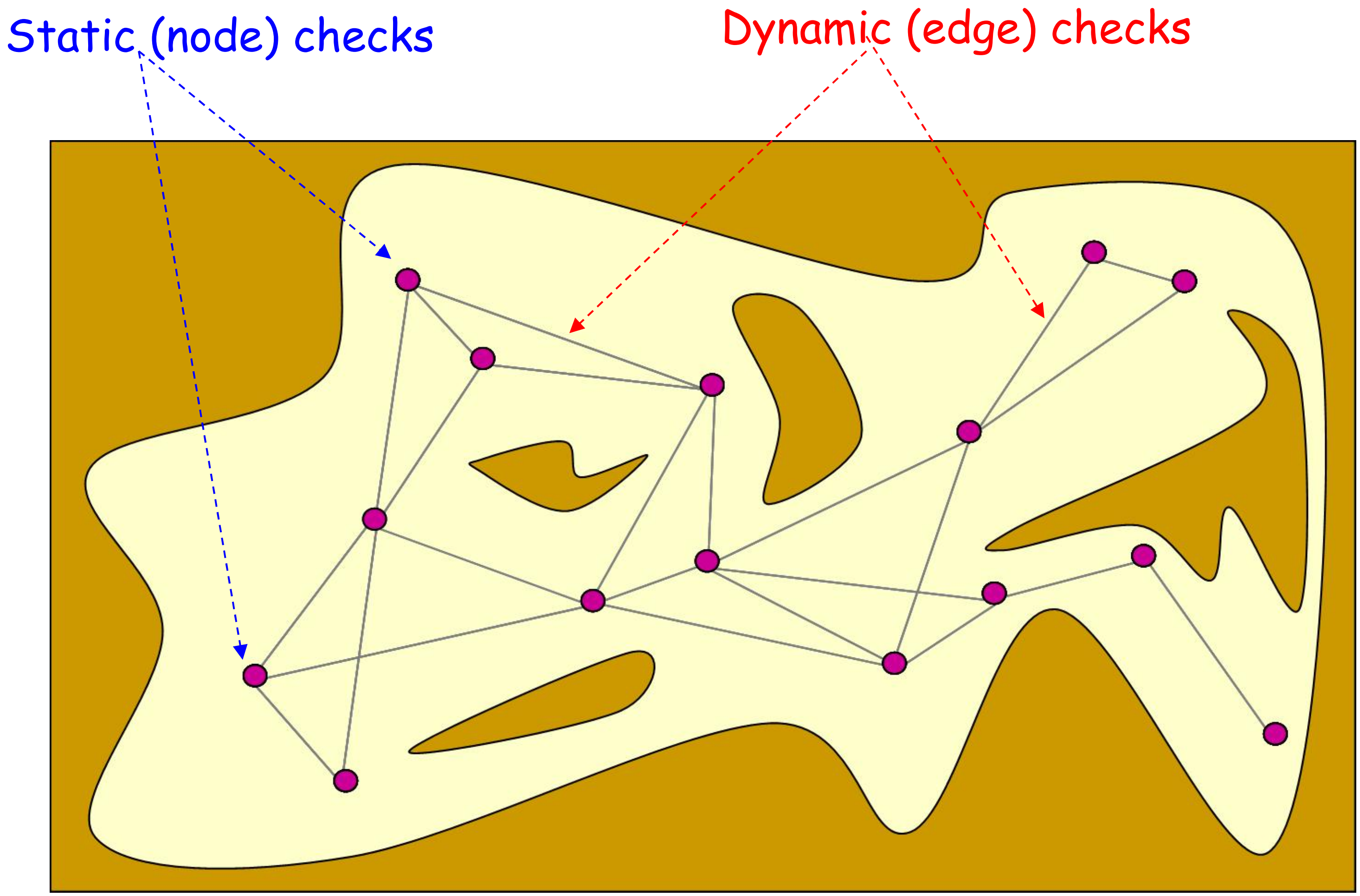
- BVH is pre-computed for each object
- Collision check through tree structure





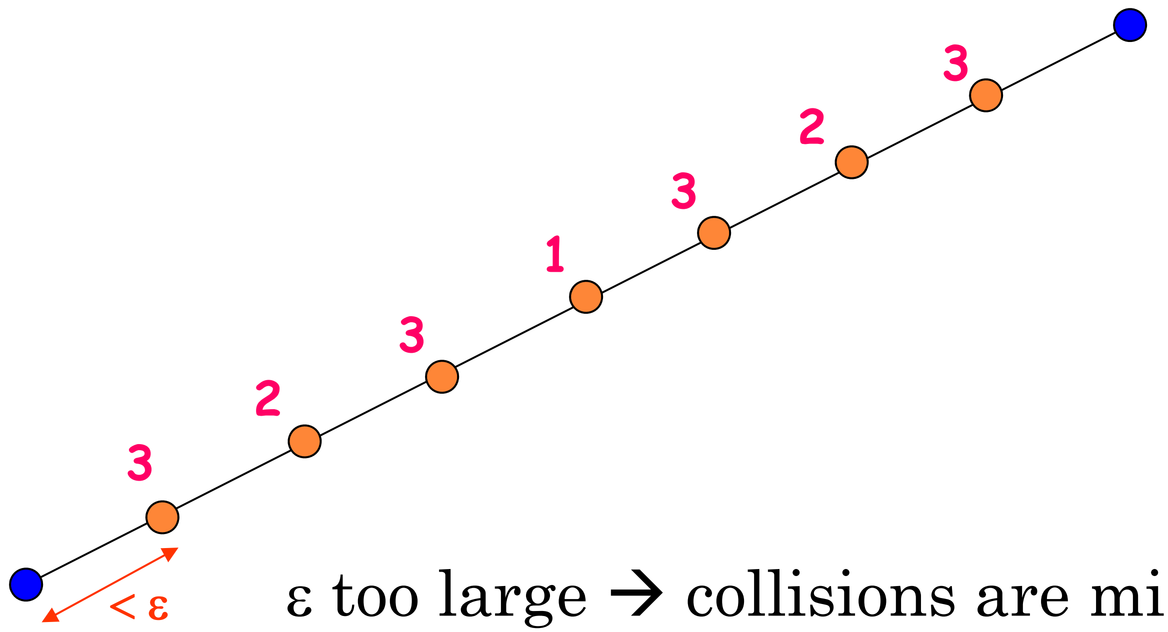
BVH in 3D

Static vs. Dynamic Collision Detection



Usual Approach to Dynamic Checking

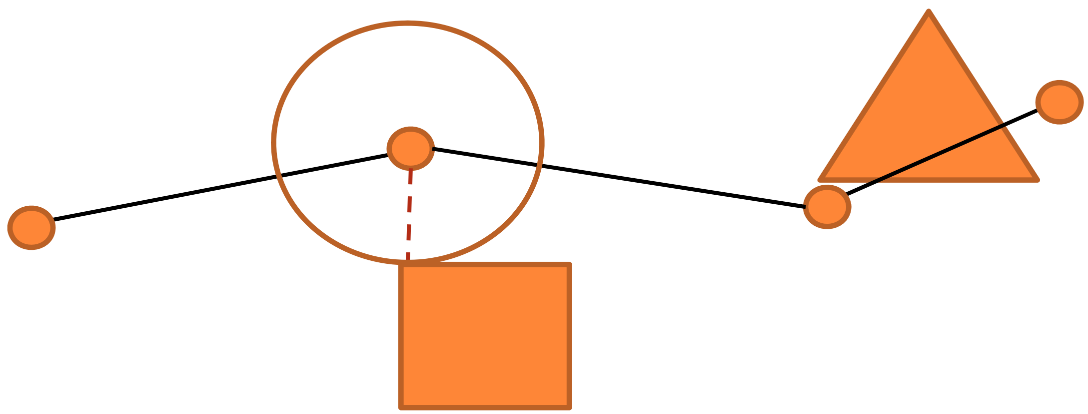
1) Discretize path at some finite resolution *e*, using bisection
2) Test statically each intermediate configuration



ε too large → collisions are missed
ε too small → slow test of local paths

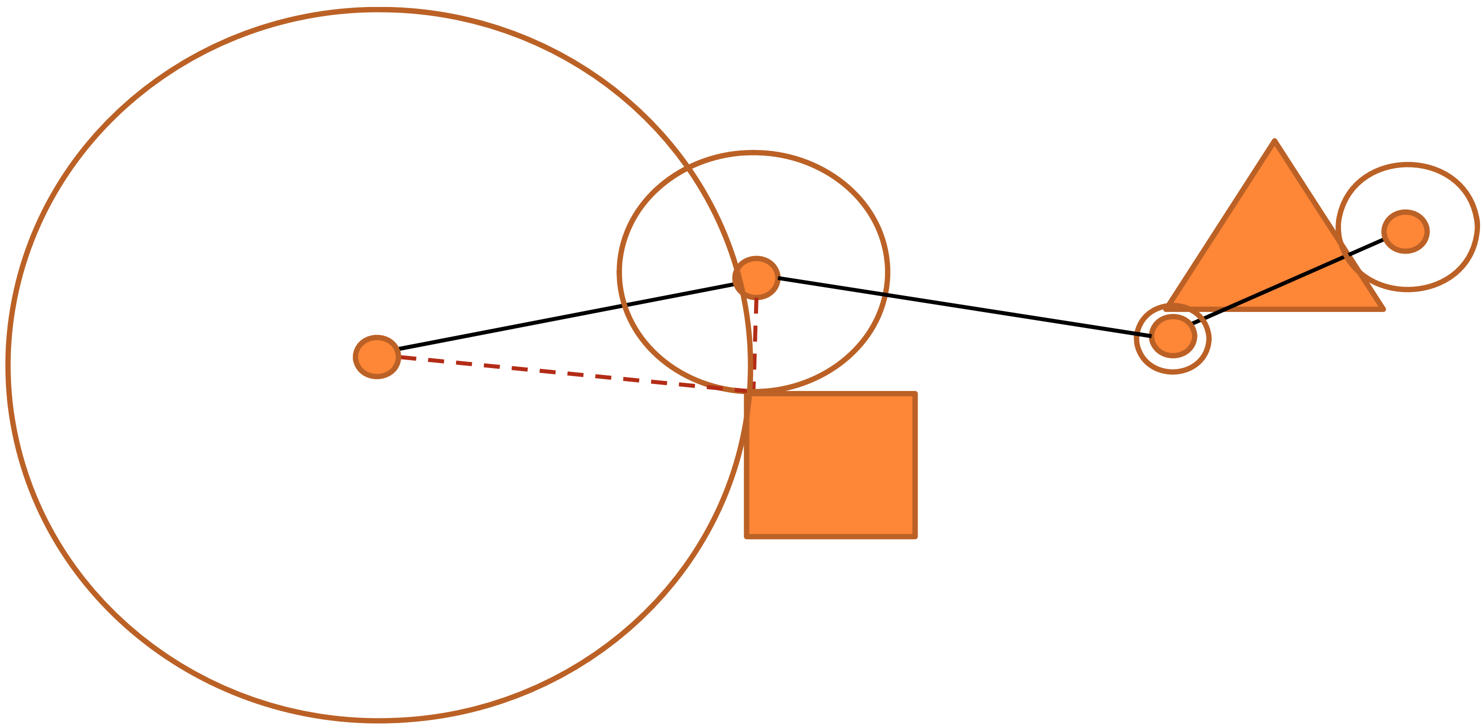
Adaptive Approach

- Since we are picking paths, motion constraint definitions are unnecessary
  - Find distance to closest obstacle at end points
  - Each point checked eliminates a section of the path



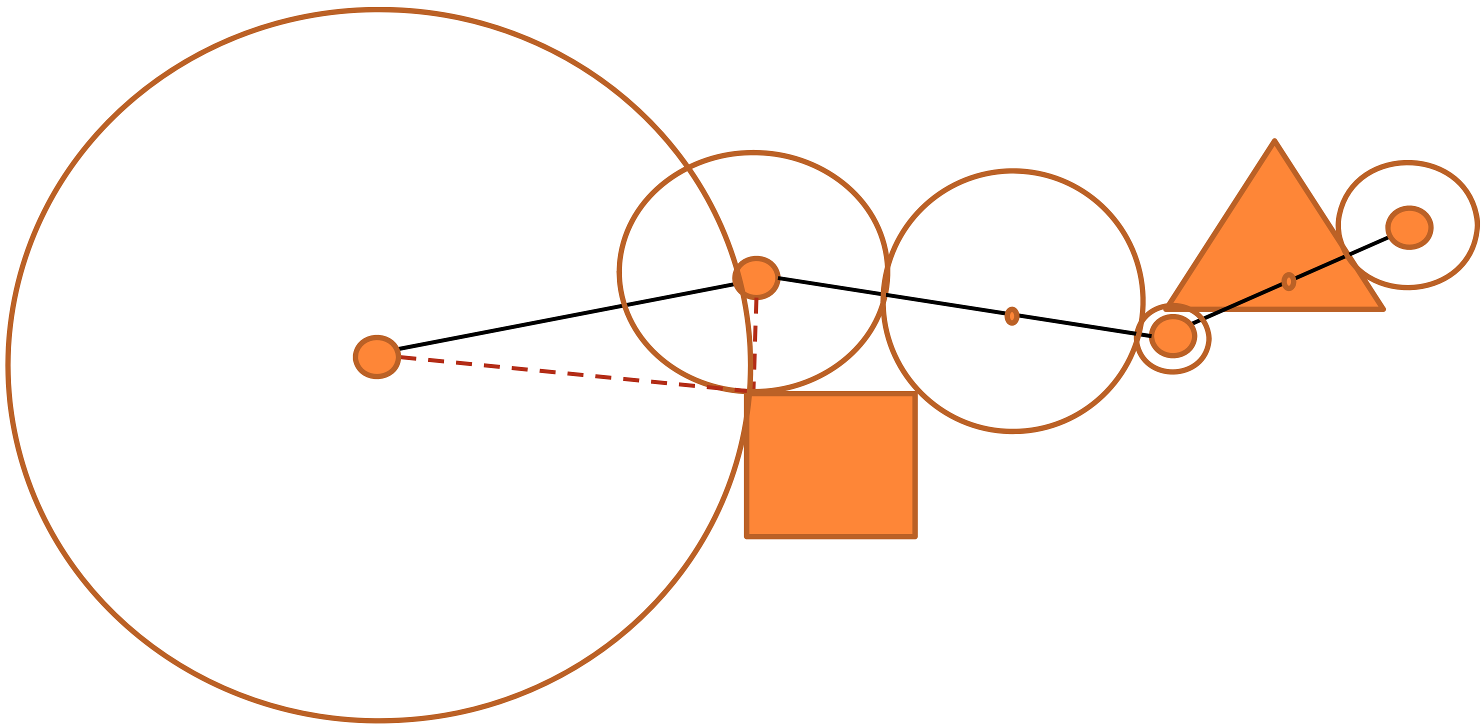
ADAPTIVE APPROACH

- Since we are picking paths, motion constraint definitions are unnecessary
  - Find distance to closest obstacle at end points
  - Each point checked eliminates a section of the path



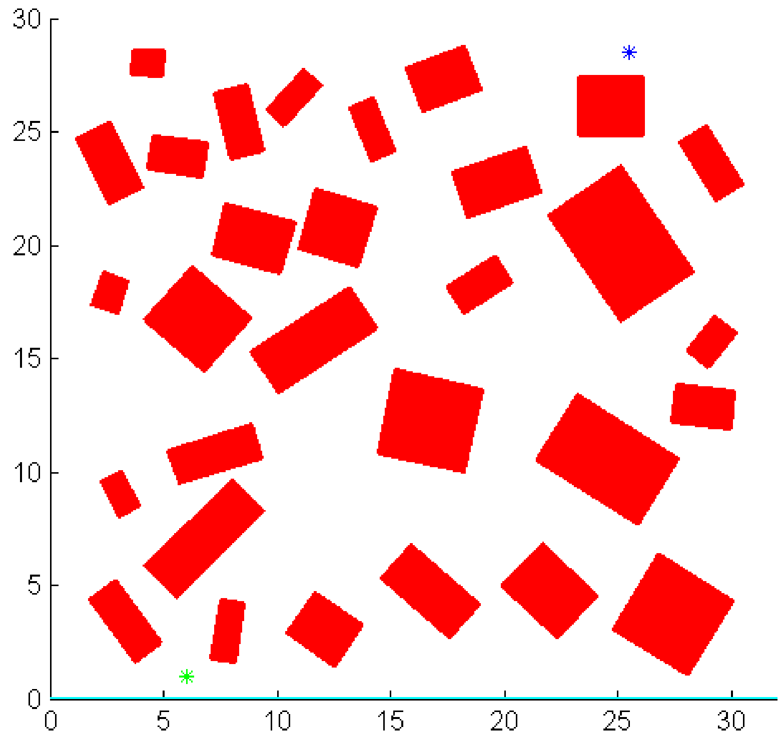
Adaptive Approach

- Since we are picking paths, motion constraint definitions are unnecessary
  - Bisect remaining path length and check



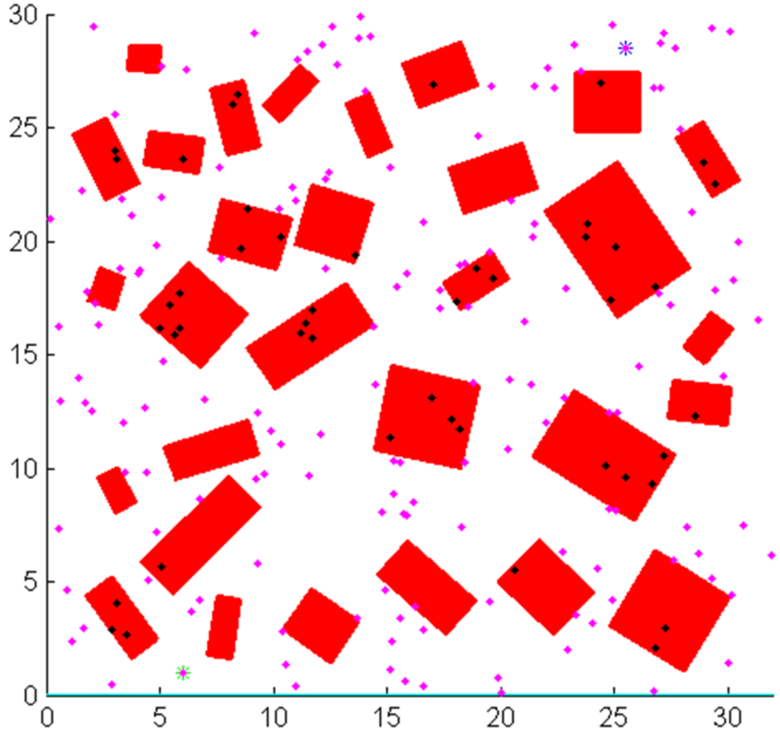
PROBABILISTIC ROADMAPS

- Example – 2D path planning
  - To keep things simple, focus on finding a path through a 2D environment with many obstacles



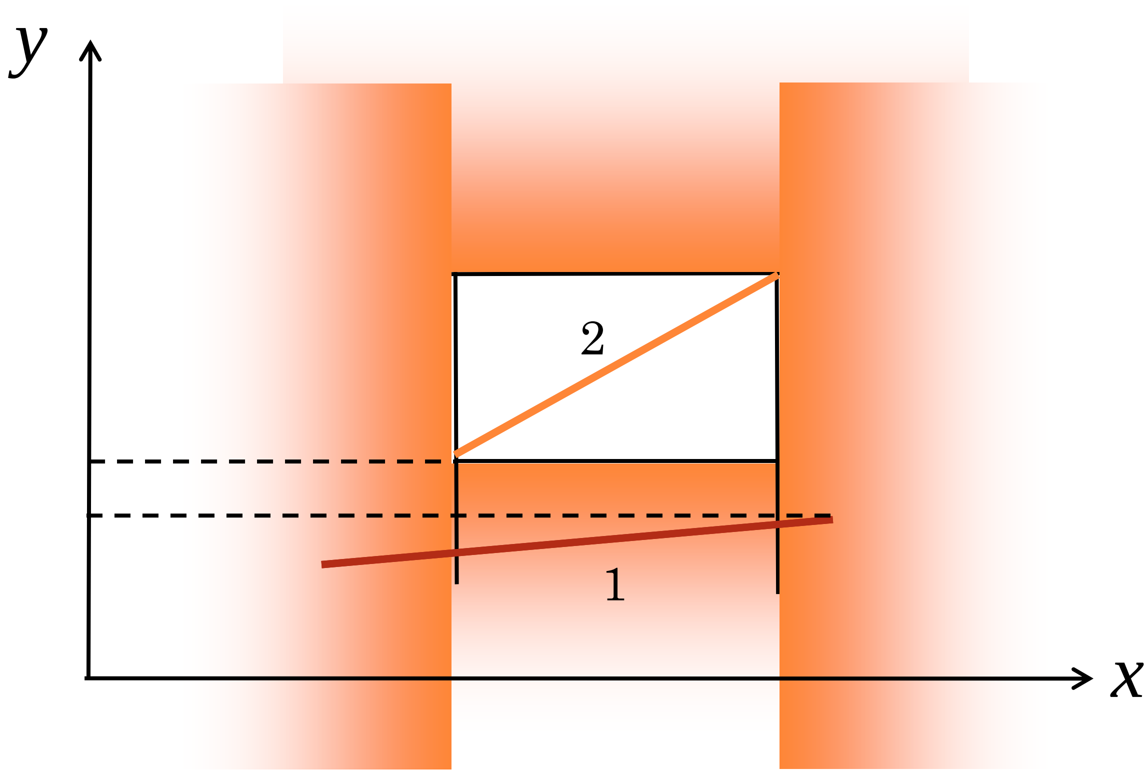
PROBABILISTIC ROADMAPS

- Example – 2D path planning
  - Collision checking
    - Points using inpolygon function in Matlab
      - Can evaluate a single point relative to entire environment very rapidly
      - Used for milestone selection



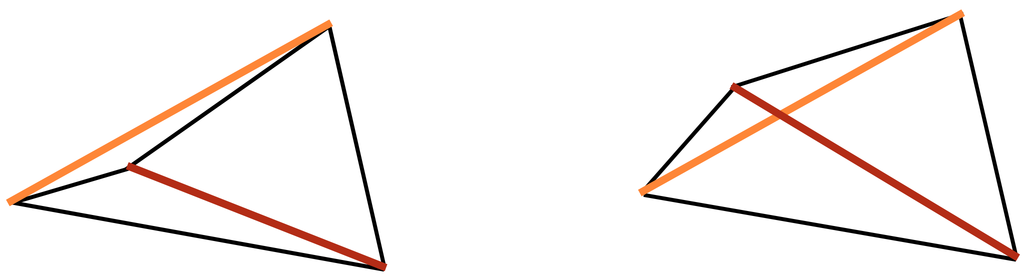
Probabilistic Roadmaps

- Example – 2D path planning
  - Collision checking
    - Edges: Use technique specific to 2D line segments
      - Step 1: If max y of edge 1 < min y of edge 2, no collision
        - All four permutations of this are checked



Probabilistic Roadmaps

- Example – 2D path planning
  - Collision checking
    - Edges: Use technique specific to 2D line segments
      - Step 2: Find shortest distance between two lines
        - If 0, collision
        - Requires four cross products and four if statements

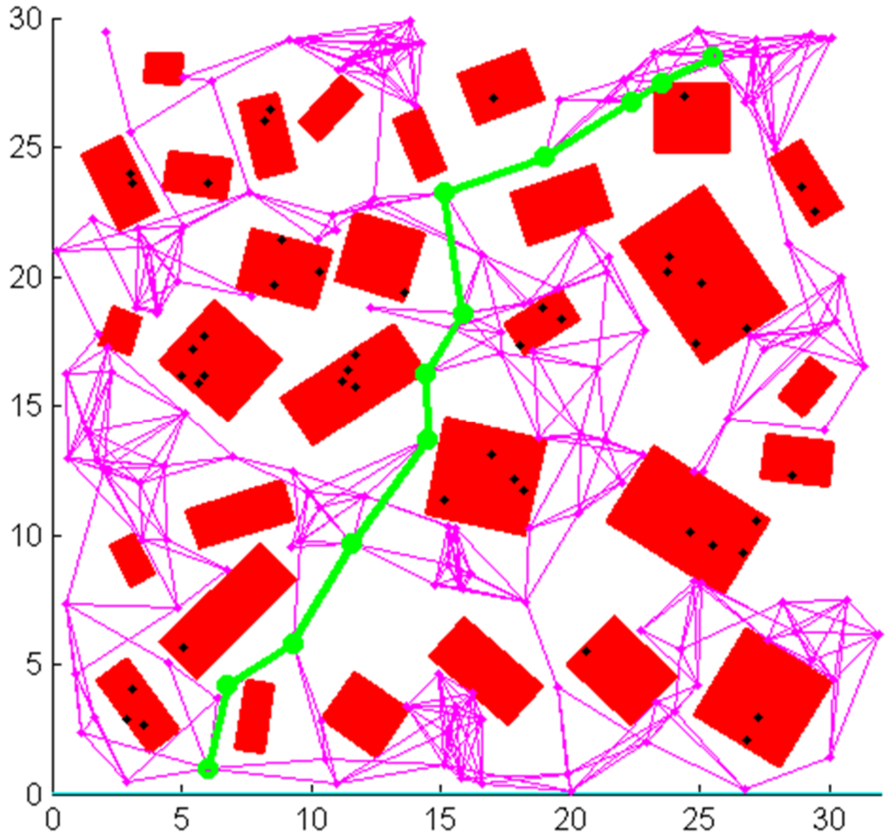


Probabilistic Roadmap

- Example – path planning
  - Batch execution
    - 200 samples yields 156 milestones
    - Attempt to connect each milestone to its closest 8 neighbours
      - Requires 1248 edge collision checks
      - Yields 504 edges
    - Find shortest path using A* search

Probabilistic Roadmaps

- Example – 2D path planning
  - Total run time : 1.71 s
  - Path length: 37.44



Probabilistic Roadmap

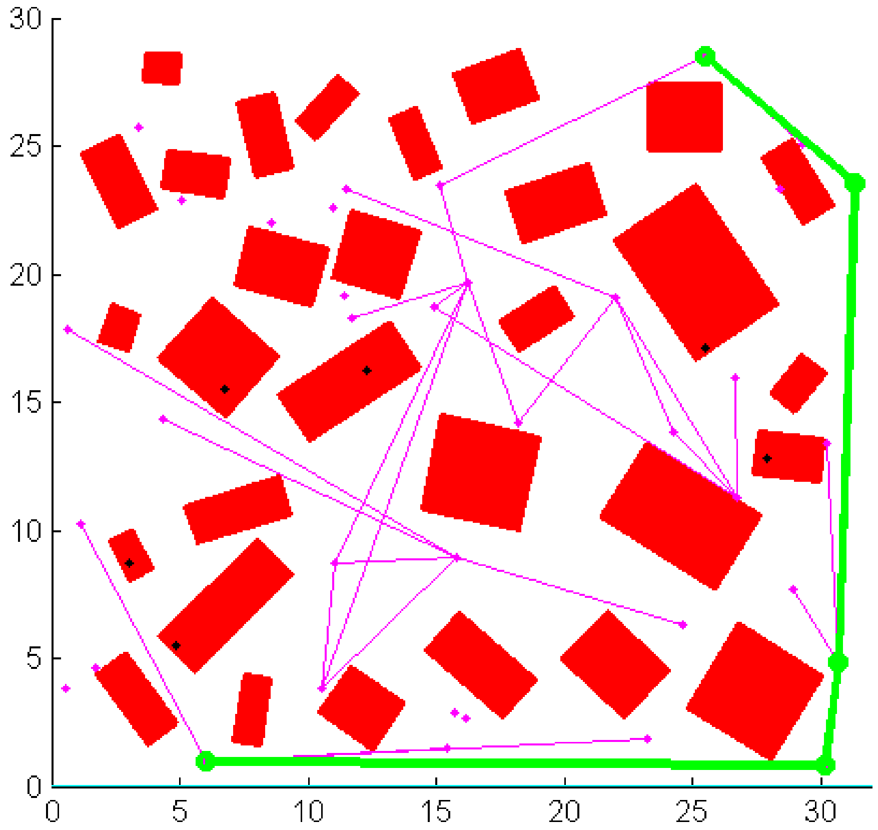
- Example – 2D path planning
  - Timing breakdown:
    - Generation of milestones: 0.02 s
    - Edge collision checking: 1.67 s
    - Shortest path: 0.02 s
  - The edge collision checking component is the biggest contributor to runtime
  - Picking edges to check wisely makes a big difference
  - Batch algorithm is fragile: need to guess correct number of links to add, correct number of samples to use to cover the space, expand if no path found

Probabilistic Roadmaps

- Example – 2D path planning
  - Online execution
    - Initialize with start and end node
      - Select a new milestone to add
      - Try to connect to n closest existing milestones
      - Stop as soon as a shortest path exists
  - Seeks to reduce total computation time
  - Sacrifices optimality
  - Single-query PRM, or RRT

PROBABILISTIC ROADMAPS

- Example – 2D path planning
  - Runtime: 0.67 s
  - Path length: 54.52

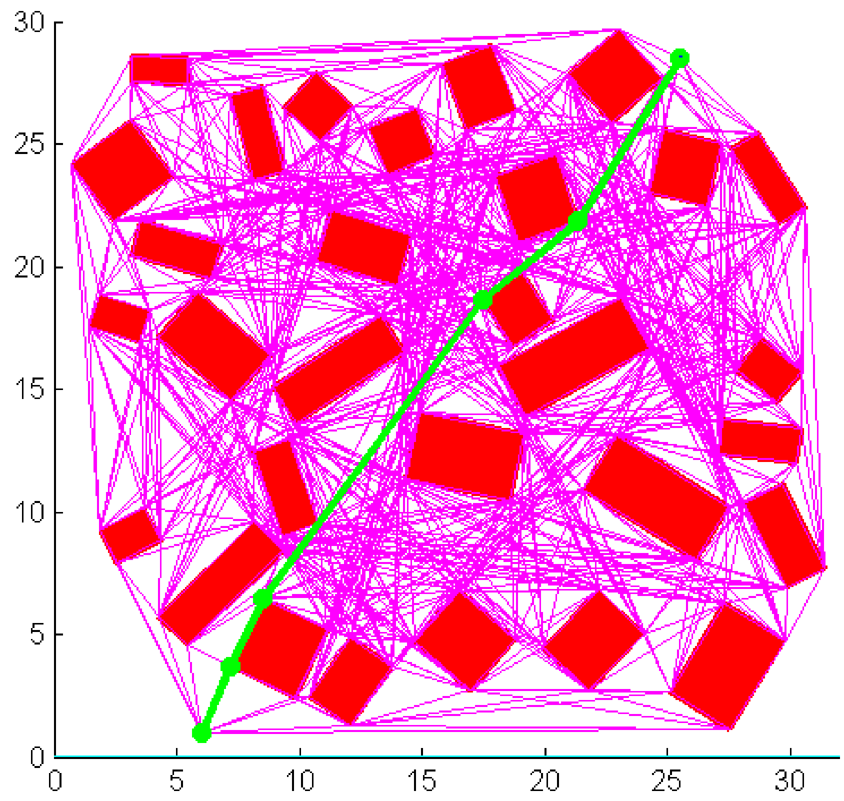


Probabilistic Roadmap

- Example – 2D path planning
  - Timing breakdown:
    - Generation of milestones: 0.03 s
    - Edge collision checking: 0.62 s
    - Shortest path: 0.02s
  - Total edges checked is significantly lower
    - Online: 259 vs Batch: 504
  - Path length is signficantly worse
    - Online: 55.52 vs Batch: 37.44
  - Online algorithm searches quickly, but checking connections between sparse milestones is a disadvantage
  - Looks for outside route (long, straight lines)

PROBABILISTIC ROADMAP

- Example – 2D path planning visibility graph
  - Runtime: 30 s
  - Path length: 34.03

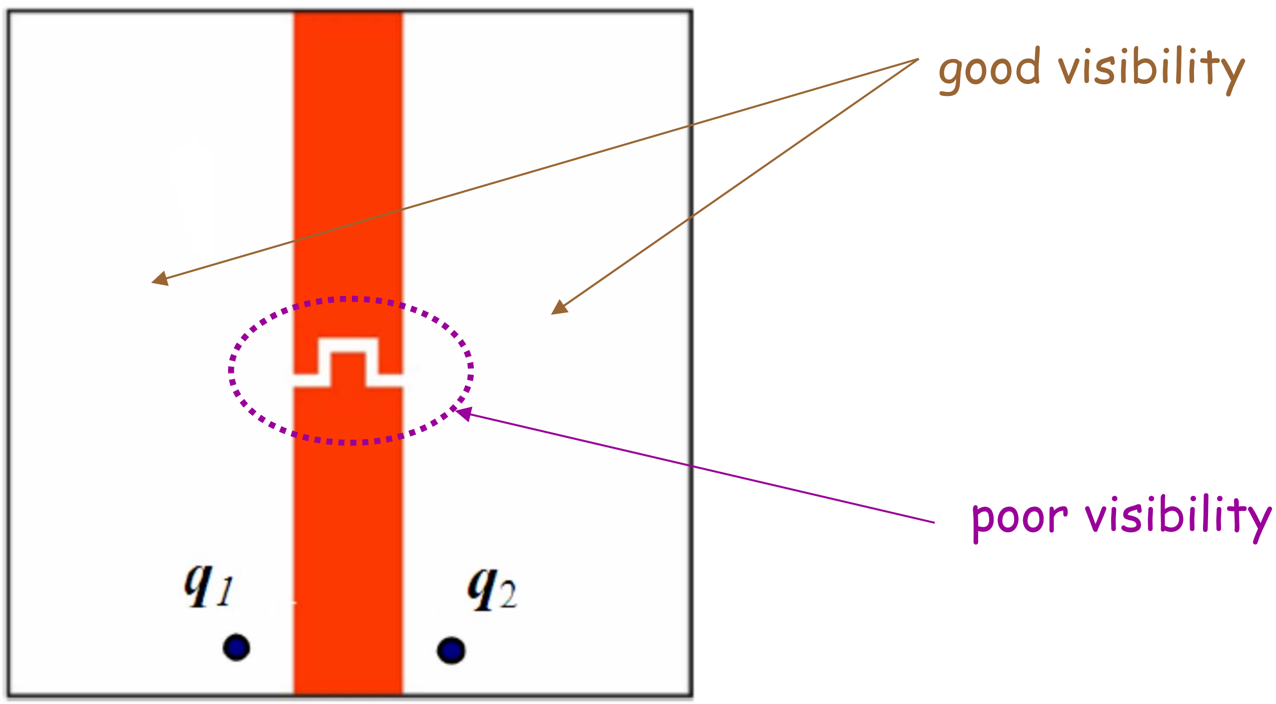


Probabilistic Roadmaps

- Making the right choices in PRMs
  - How to generate node samples
    - Sampling strategy
  - Which milestones to connect
    - Connection strategy
- Goal: Minimize the roadmap size to find feasible path to end configuration

Probabilistic Roadmaps

- Why non-uniform sampling?
  - Visibility is not uniformly favorable in free space



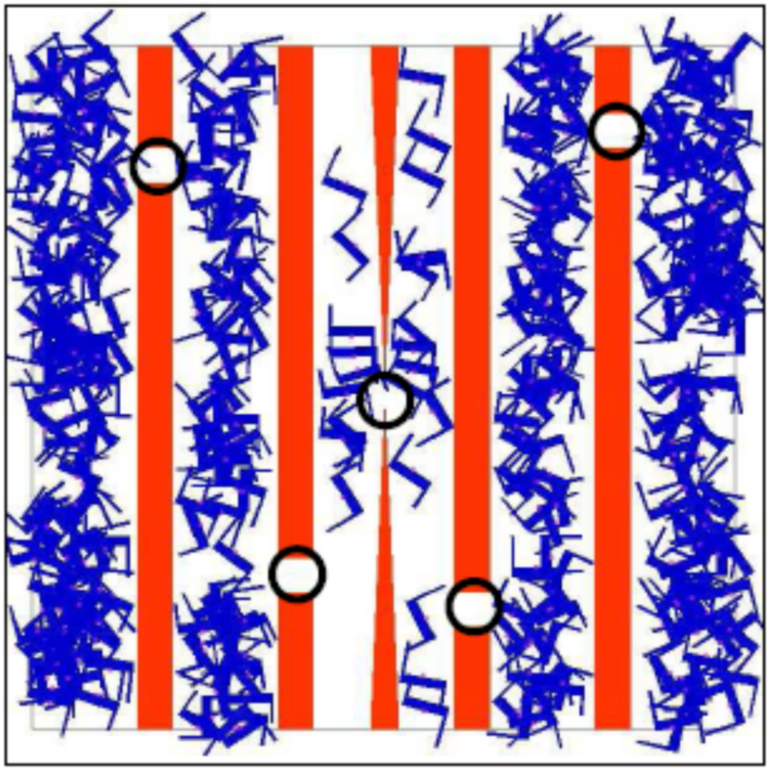
  - Regions with poorer visibility should be more densely sampled

Non-Uniform Sampling

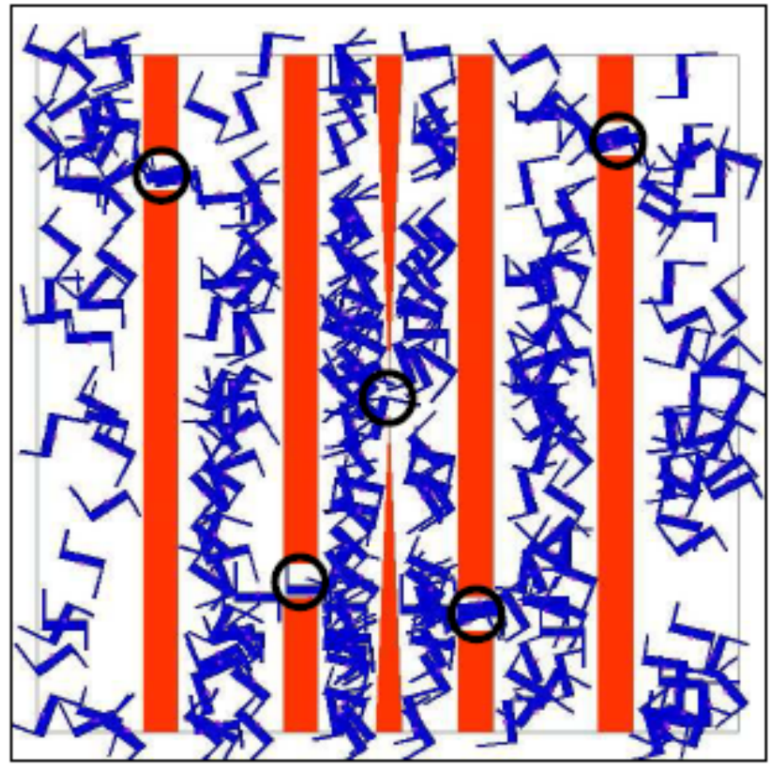
- But how to identify low visibility regions?
  - Workspace-guided strategies
    - Identify narrow passages in the workspace and map them into the configuration space
  - Filtering strategies
    - Sample many configurations, find interesting patterns, and retain only promising configurations
  - Adaptive strategies
    - Adjust the sampling distribution (p) on the fly, by considering collisions
  - Deformation strategies
    - Deform the free space, e.g., to widen narrow passages
    - Morph resulting path to account for expansion

NON-UNIFORM SAMPLING

- Workspace Guided Strategies



Uniform sampling



Workspace-guided sampling

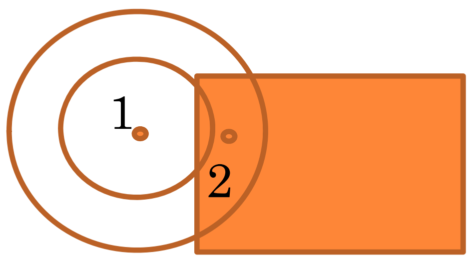
- Fails when robot configuration is complex relative to workspace

Non-Uniform Sampling

- Filtering strategies
  - Because point sampling is cheap, sample many configurations and only keep interesting ones
  - Remove the clutter from easy to navigate regions
  - Two methods, each of which tests the properties of two samples
    - Gaussian
    - Bridge

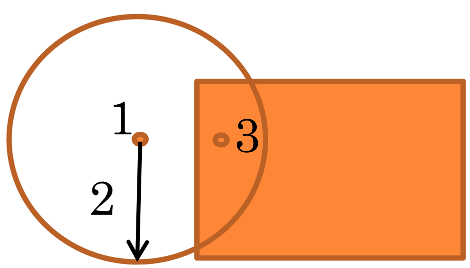
Non-Uniform Sampling

- Gaussian Sampling (1 - Lavalle)
    1. Sample a configuration q uniformly at random from configuration space
    2. Sample a configuration q' at random with Gaussian distribution $\boldsymbol{N}_{[0,s]}(x)$
    3. If only one of q and q' is in free space, retain the one in free space as a node; else retain none



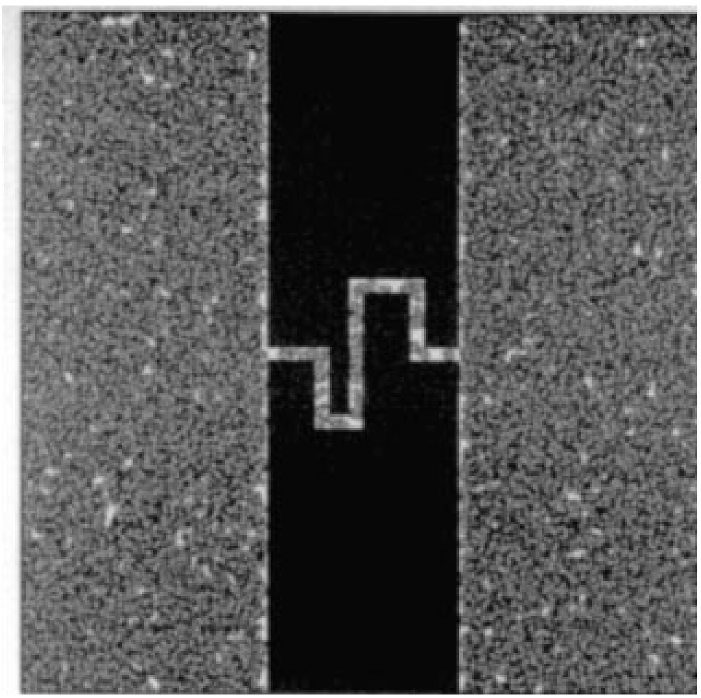
Non-Uniform Sampling

- Gaussian Sampling (2 - Latombe)
  1. Sample a configuration q uniformly at random from configuration space
  2. Sample a real number x at random with Gaussian distribution $N_{[0,s]}(x)$
  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

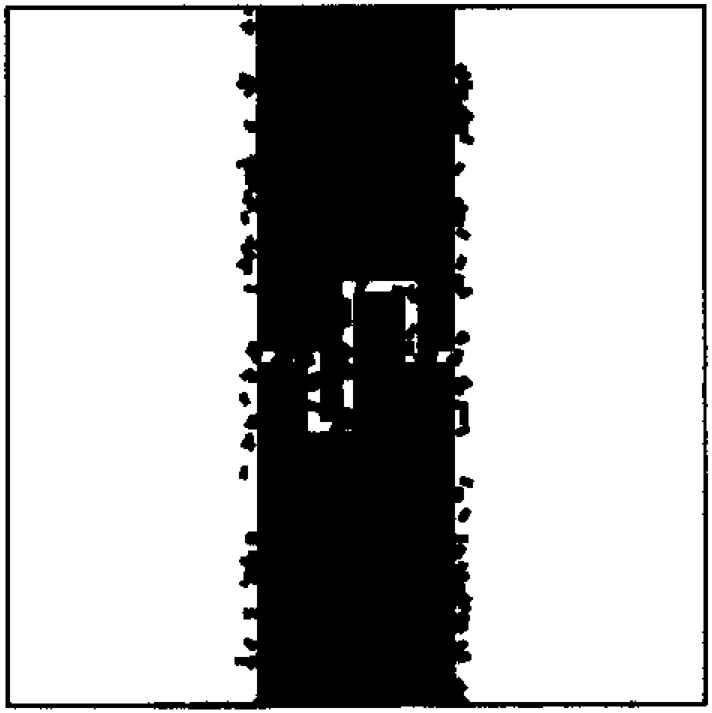


Non-Uniform Sampling

- Uniform vs Gaussian Sampling (2)
  - The benefit lies in fewer samples to connect, similar or more samples tried and rejected.



Milestones (13,000) created by uniform sampling before the narrow passage was adequately sampled

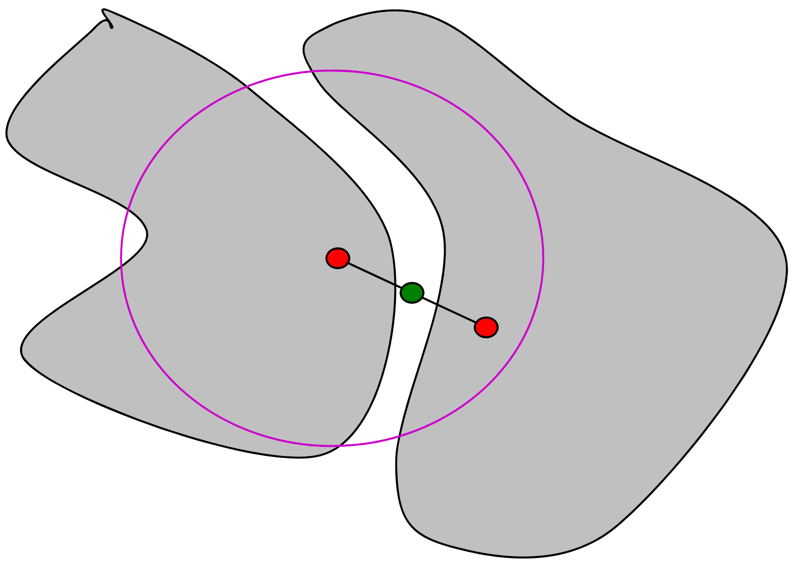


Milestones (150) created by Gaussian sampling

Non-Uniform Sampling

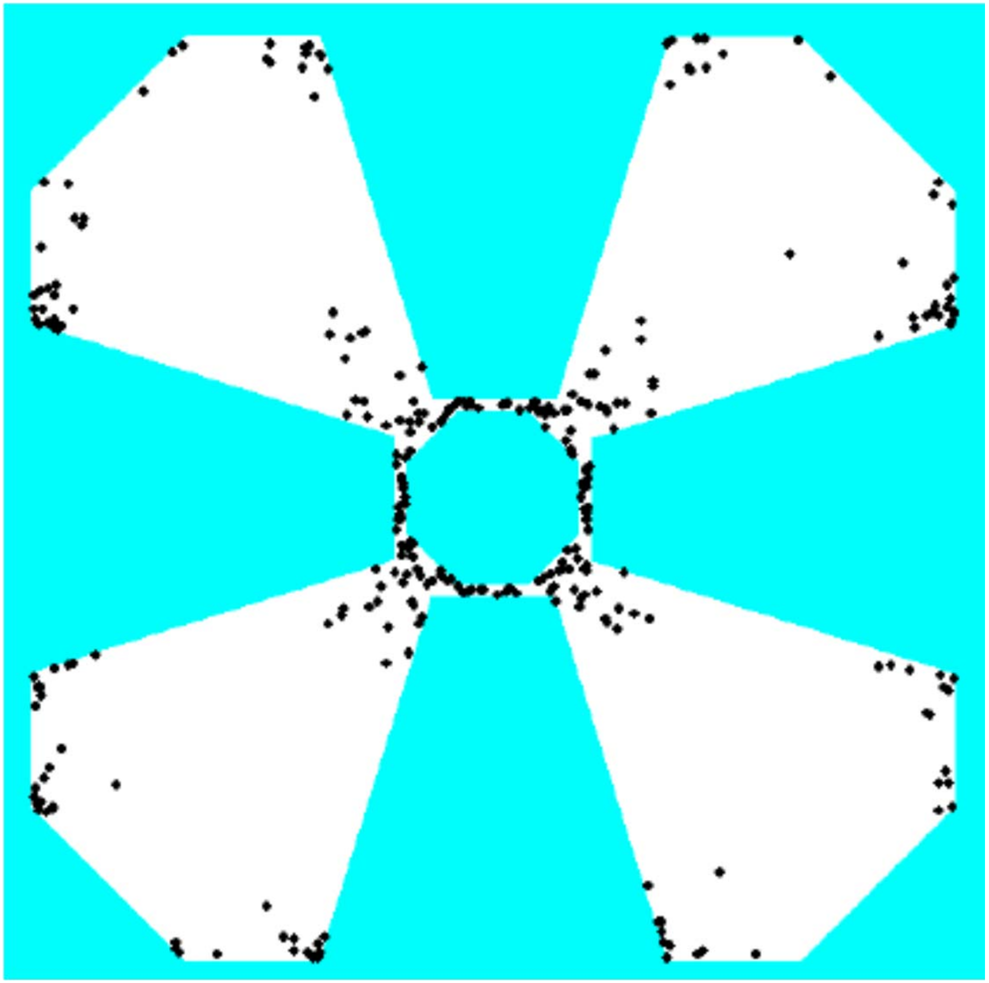
- Bridge sampling
  - Altered end check from Gaussian sampling

1. Sample two configurations *q* and *q'* using Gaussian sampling technique (1 or 2)
2. If neither is in free space, then
   1. if $q_m = (q+q')/2$ is in free space, then retain q_m
3. Else retain none



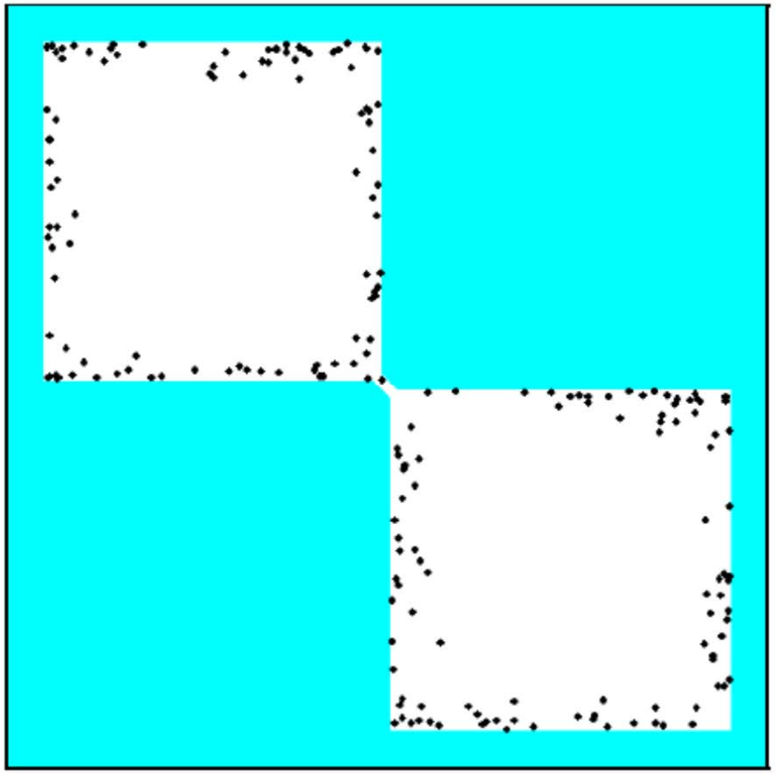
Non-Uniform Sampling

- Example of Bridge test sampling

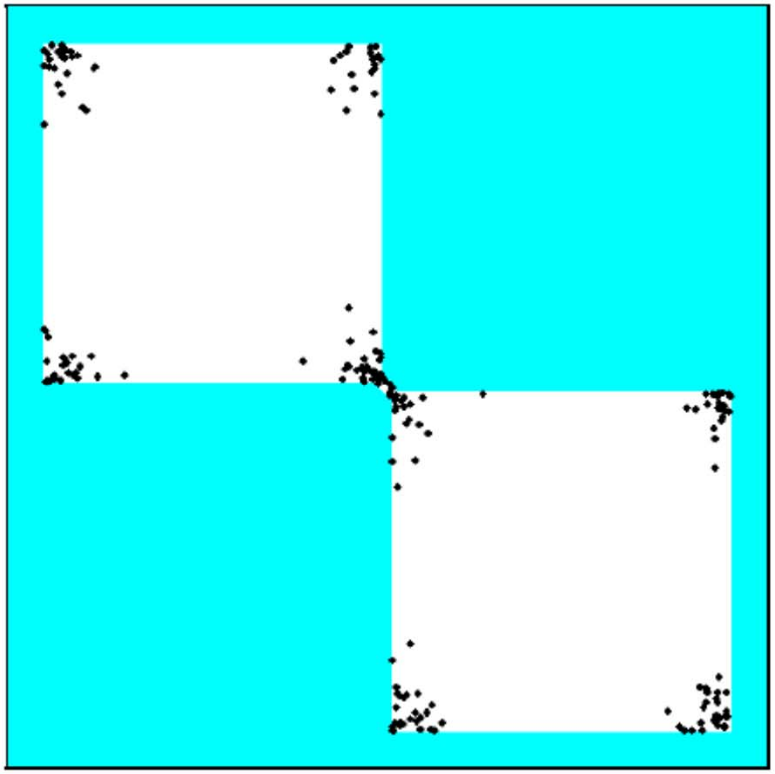


Non-Uniform Sampling

- Comparison of Gaussian and Bridge sampling



Gaussian (2)



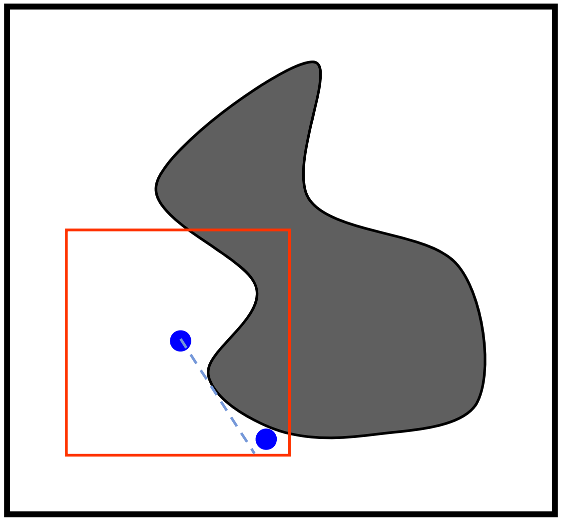
Bridge test

NON-UNIFORM SAMPLING

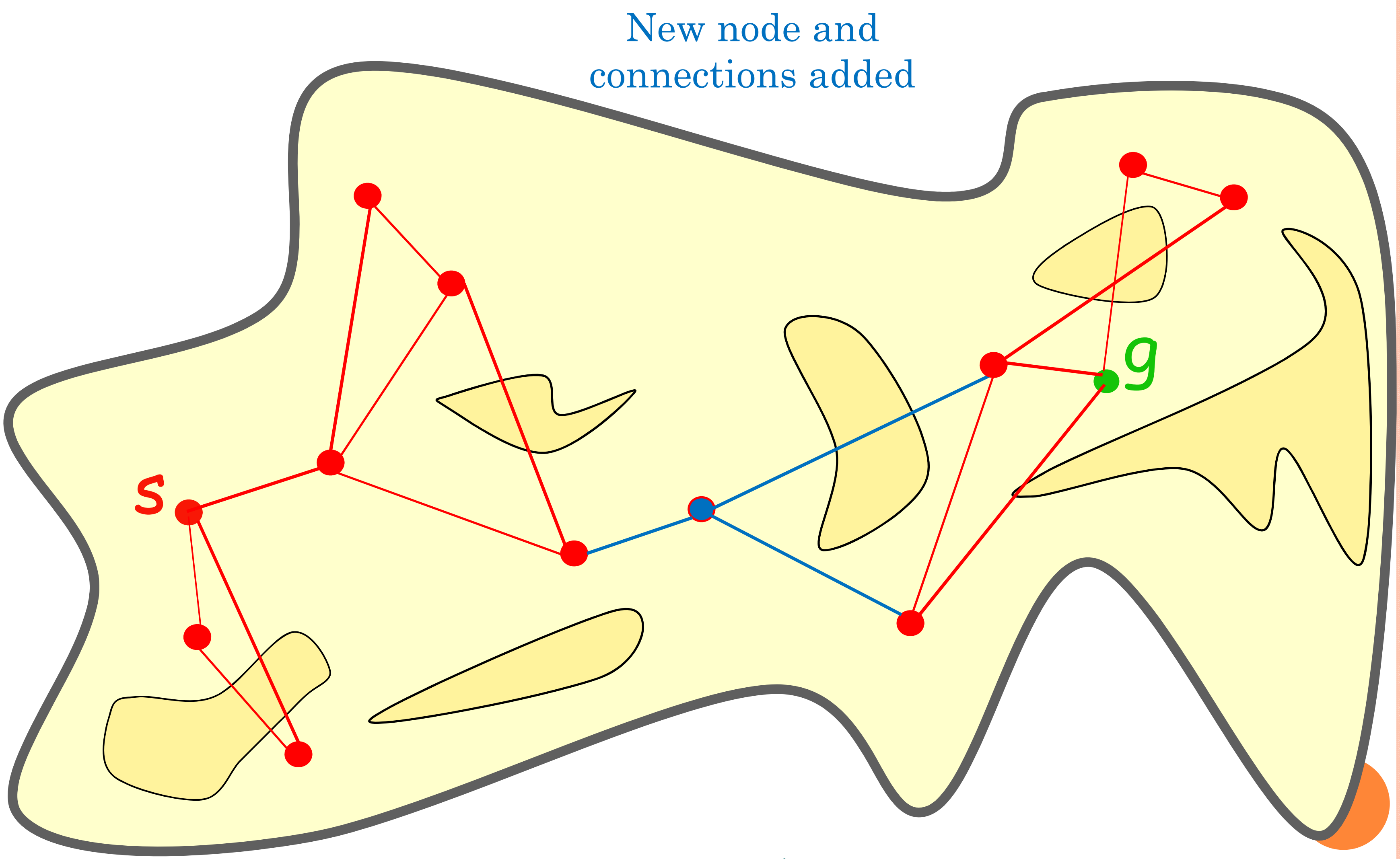
- Connection Sampling
  - Limit number of connections:
    - Nearest-neighbor strategy
  - Delay costly computation:
    - Lazy collision checking [Sanchez-Ante, 02]

Lazy Collision Checking

- Connections between close milestones have high probability of being free of collision
- Most of the time spent in collision checking is done to test connections
- Most collision-free connections will not be part of the final path
- Testing connections is more expensive for collision-free connections
- Hence: Postpone the tests of connections until they are absolutely needed

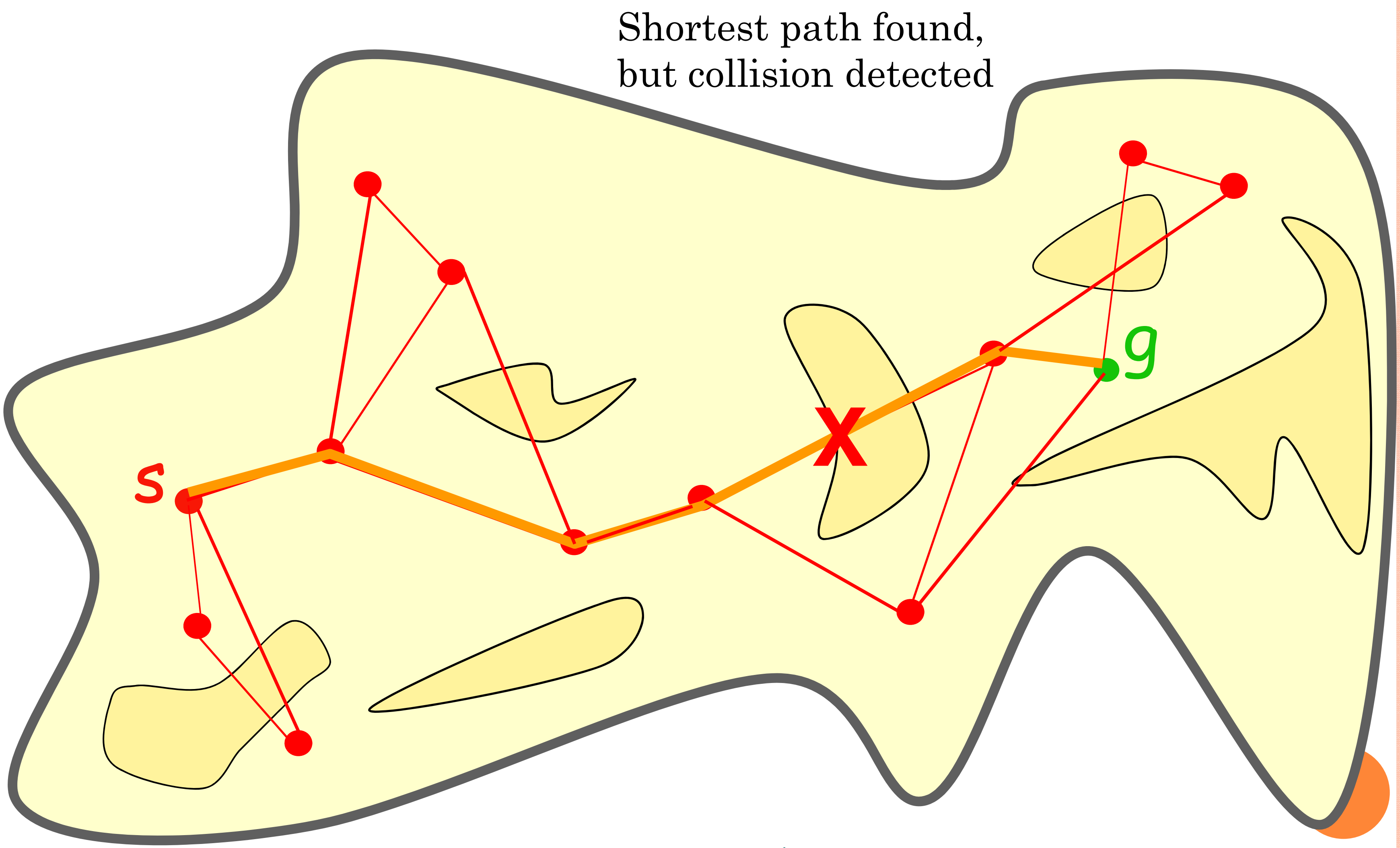


Lazy Collision Checking



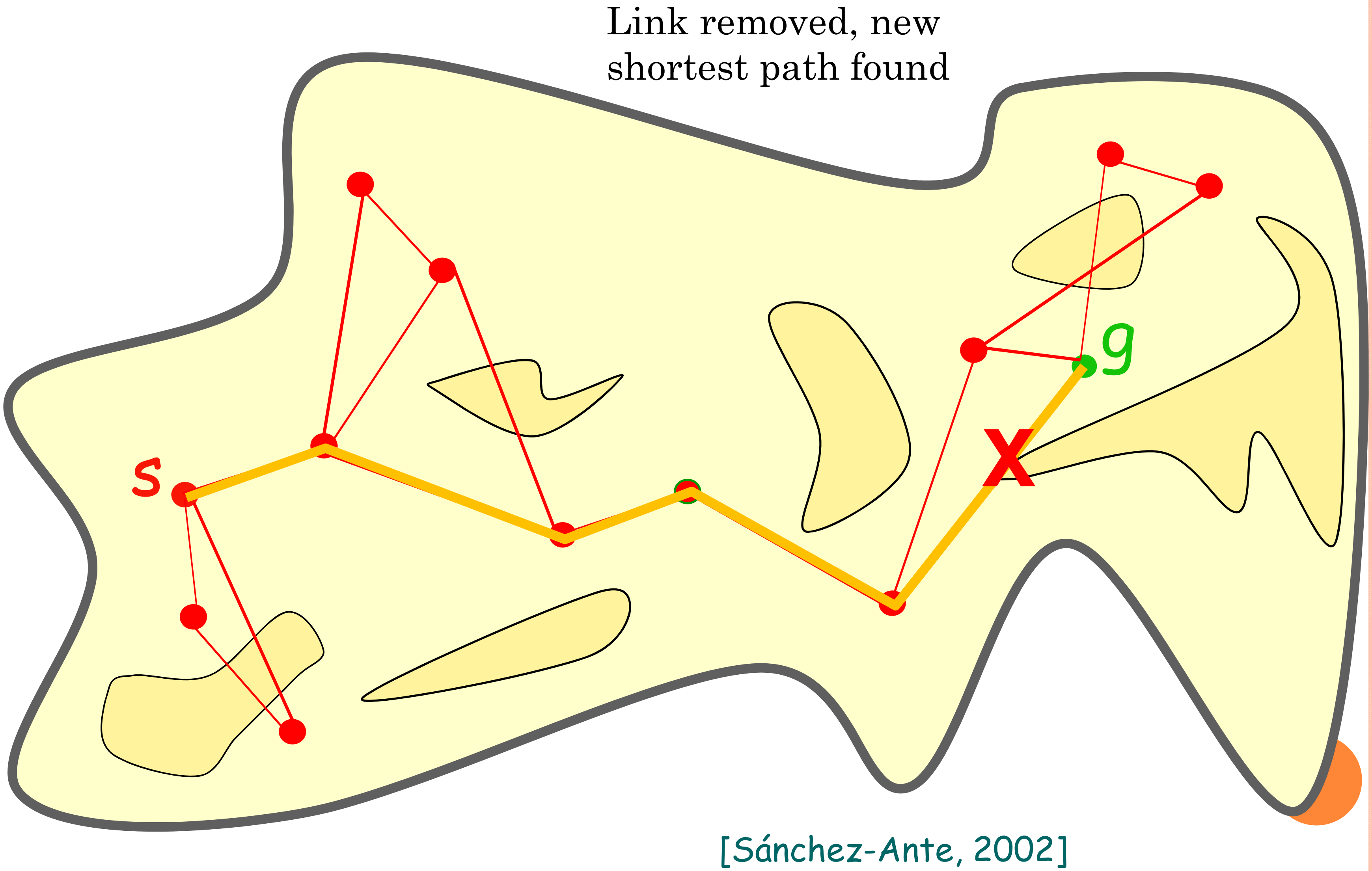
[Sánchez-Ante, 2002]

Lazy Collision Checking



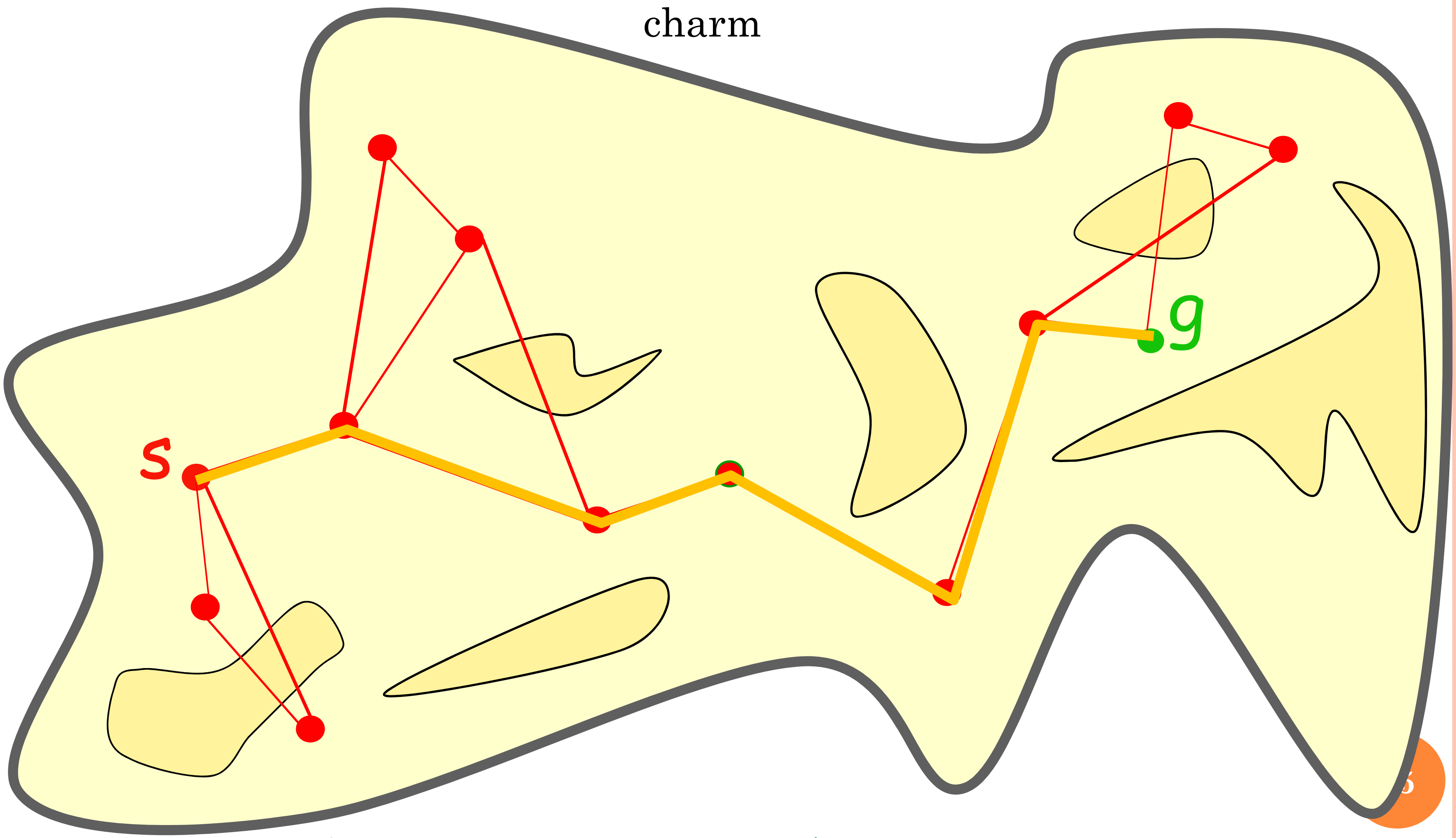
[Sánchez-Ante, 2002]

Lazy Collision Checking



LAZY COLLISION CHECKING

Third time's the charm



Potential 10x speedup

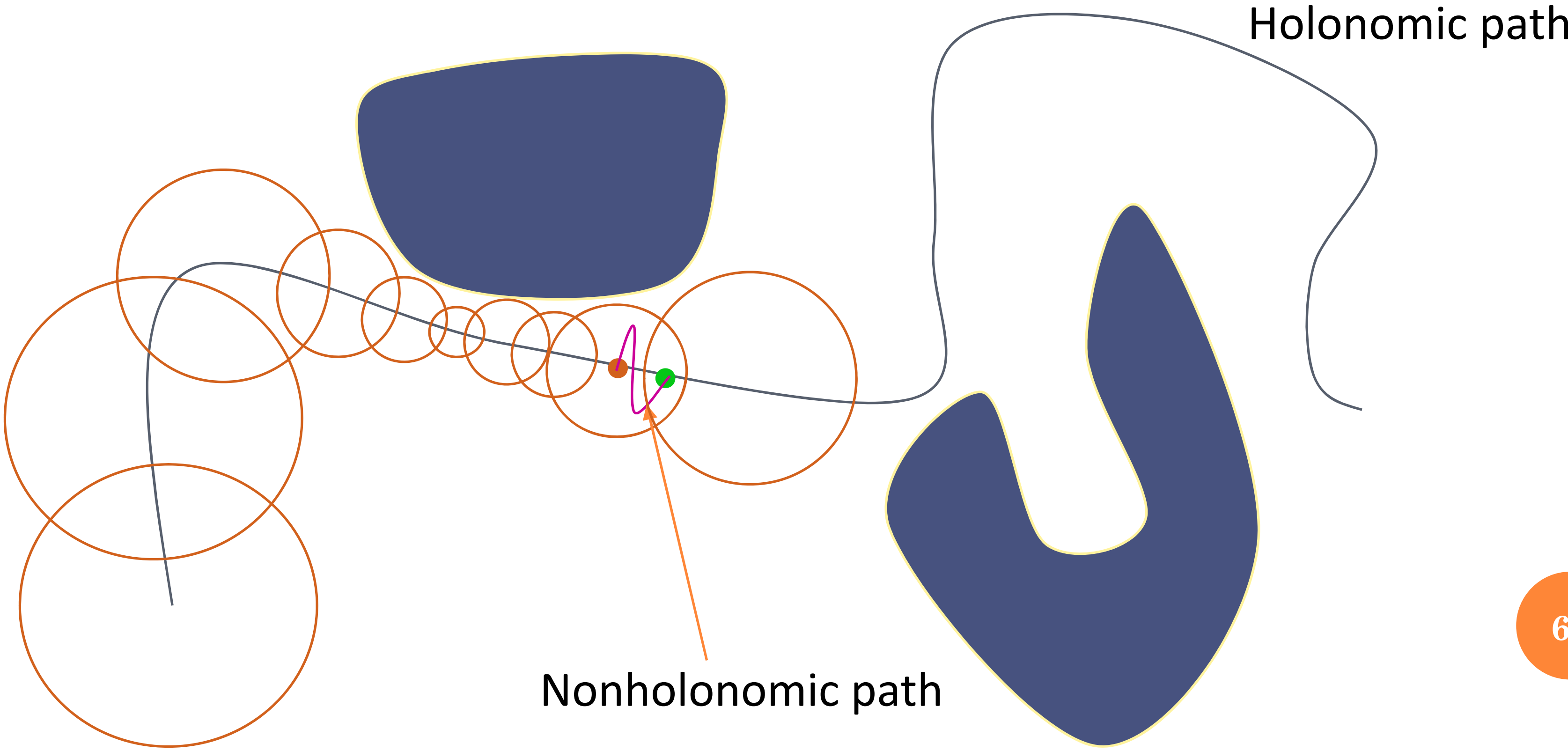
[Sánchez-Ante, 2002]

Nonholonomic Path Planning

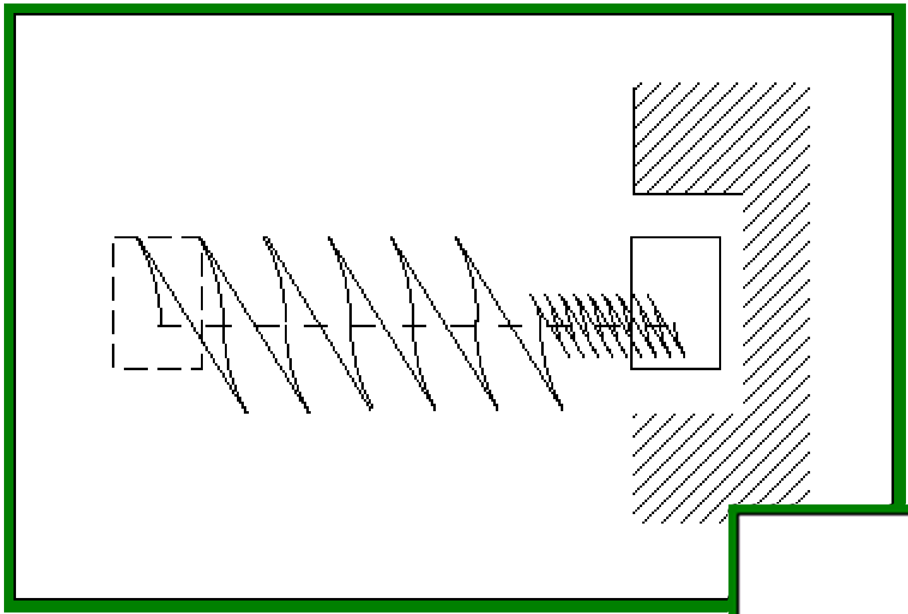
- Two-phase planning (path deformation):
  - Compute collision-free path ignoring nonholonomic constraints
  - Transform this path into a nonholonomic one
  - Efficient
  - Need for a "good" set of maneuvers
- Direct planning (control-based sampling):
  - Use "control-based" sampling to generate a tree of milestones until one is close enough to the goal (deterministic or randomized)
  - Applicable to high-dimensional c-spaces

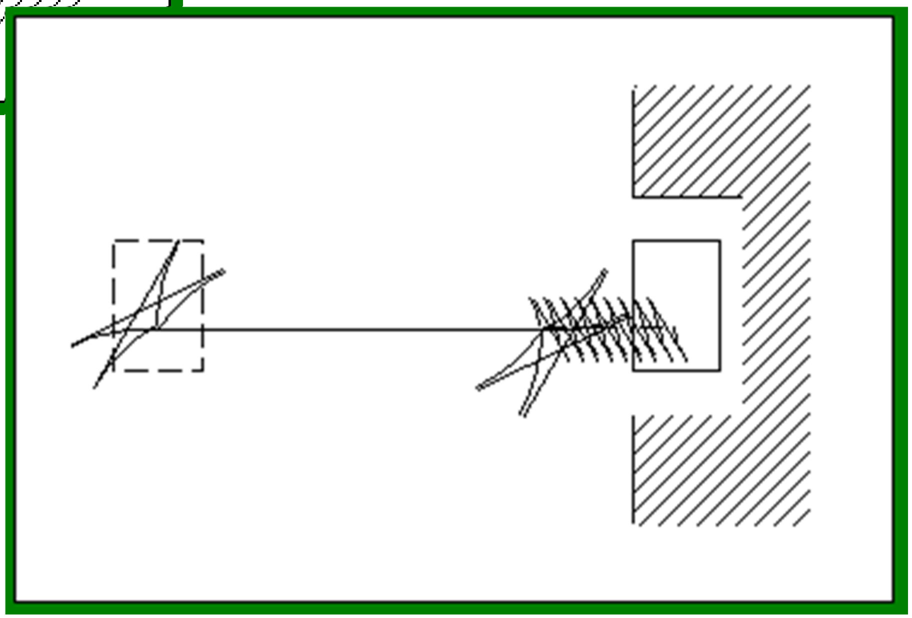
Path Deformation

- Identify holonomic path and minimum distance to obstacles
- Select nonholonomic maneuver from library of moves to execute holonomic path segment



Path Examples – Parking a Car

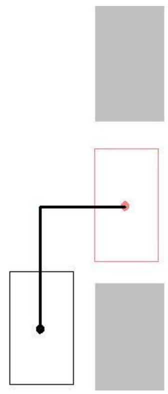




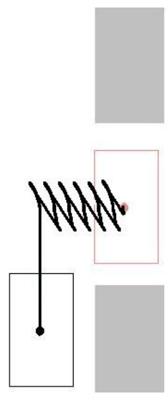
Drawbacks of Two-phase Planning

- Final path can be far from optimal

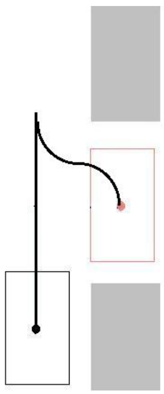
Holonomic Solution



Deformed Path

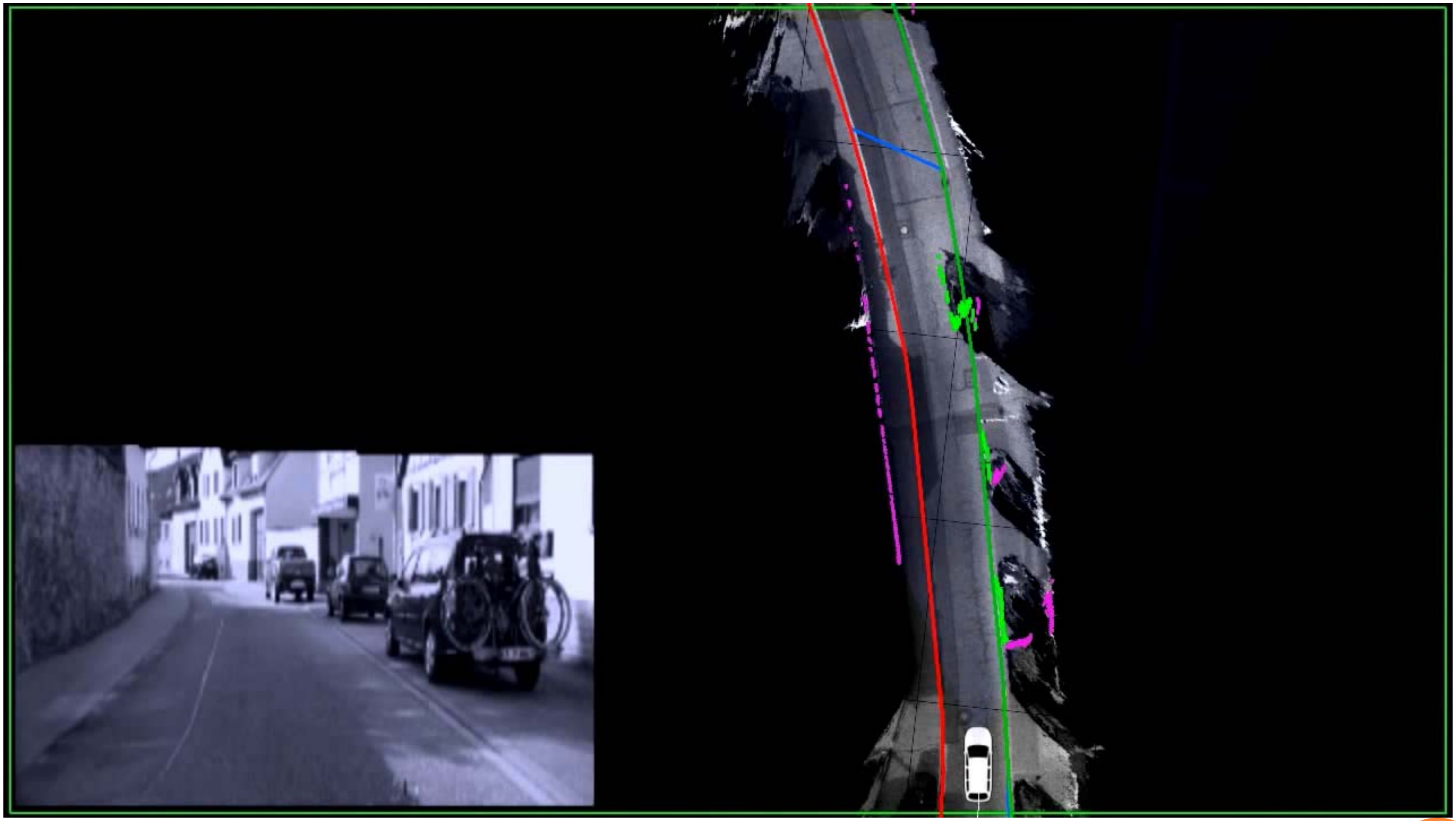


Nonholonomic Optimum



- Must create a library of maneuvers that can get everywhere in the workspace

Autonomous Driving in Mercedes



DIRECT PLANNING

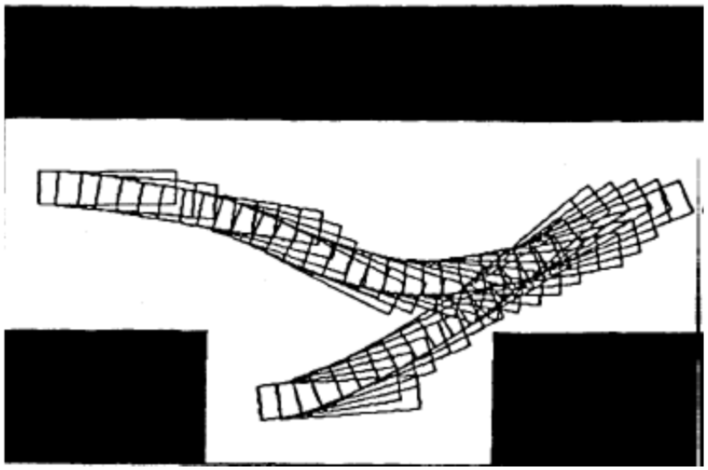
- Sometimes referred to as Kinodynamic planning
  - Implies acceleration and velocity constraints on motion

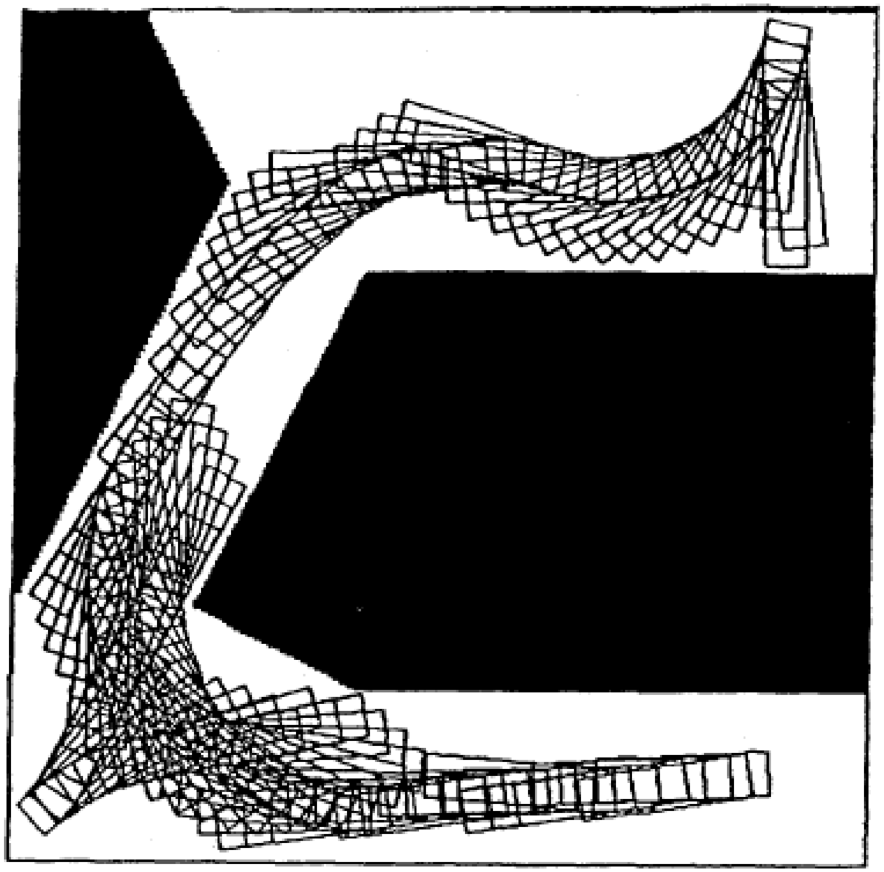
- Control-based sampling (trajectory rollout):
  1. Pick control vector (at random or not)
  2. Integrate equation of motion over short duration (picked at random or not)
  3. If the motion is collision-free, then the endpoint is the new milestone

  - Tree-structured roadmaps
    - Rapidly-expanding Random Trees (RRTs)
    - Need for endgame regions

Direct Planning

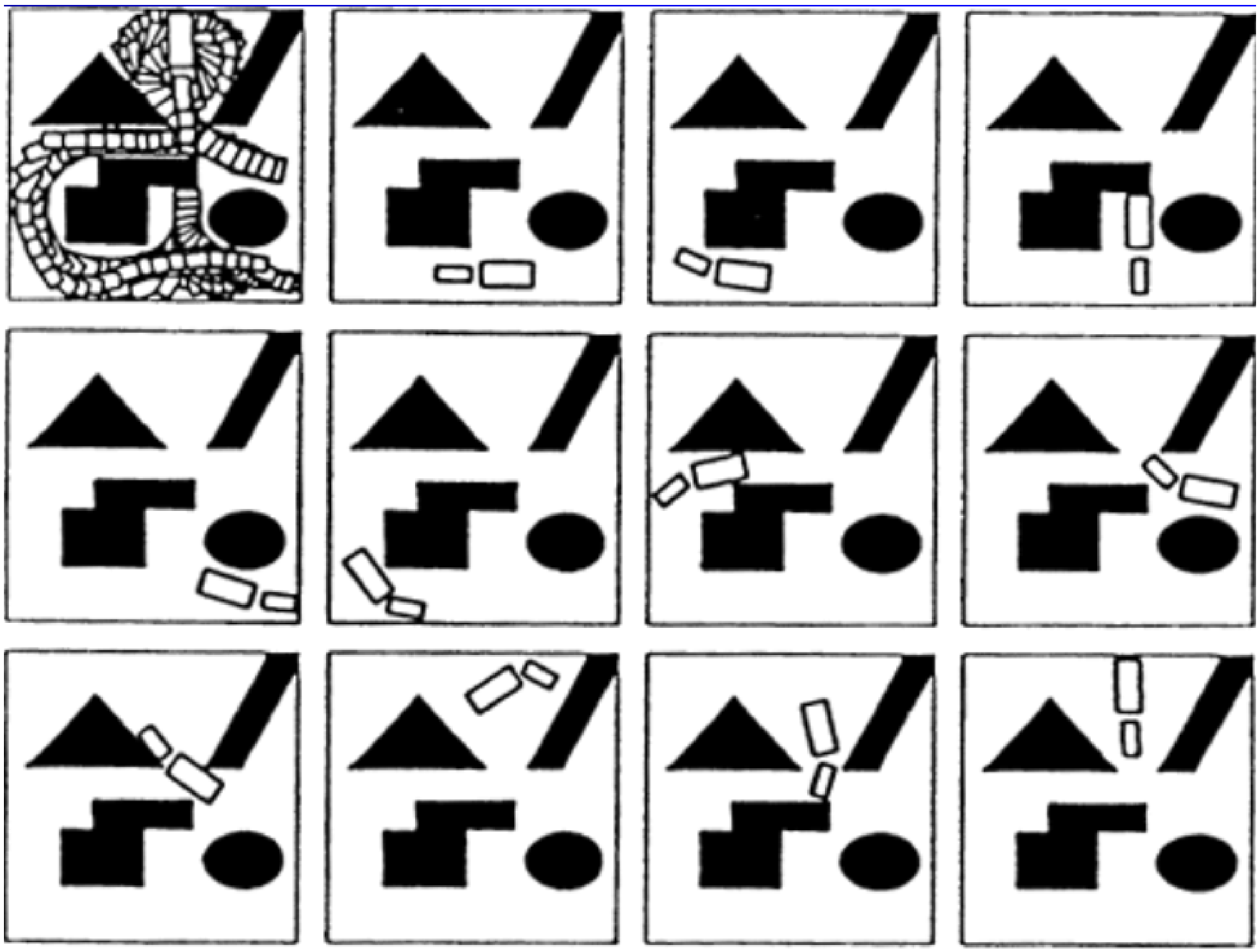
- Some great examples



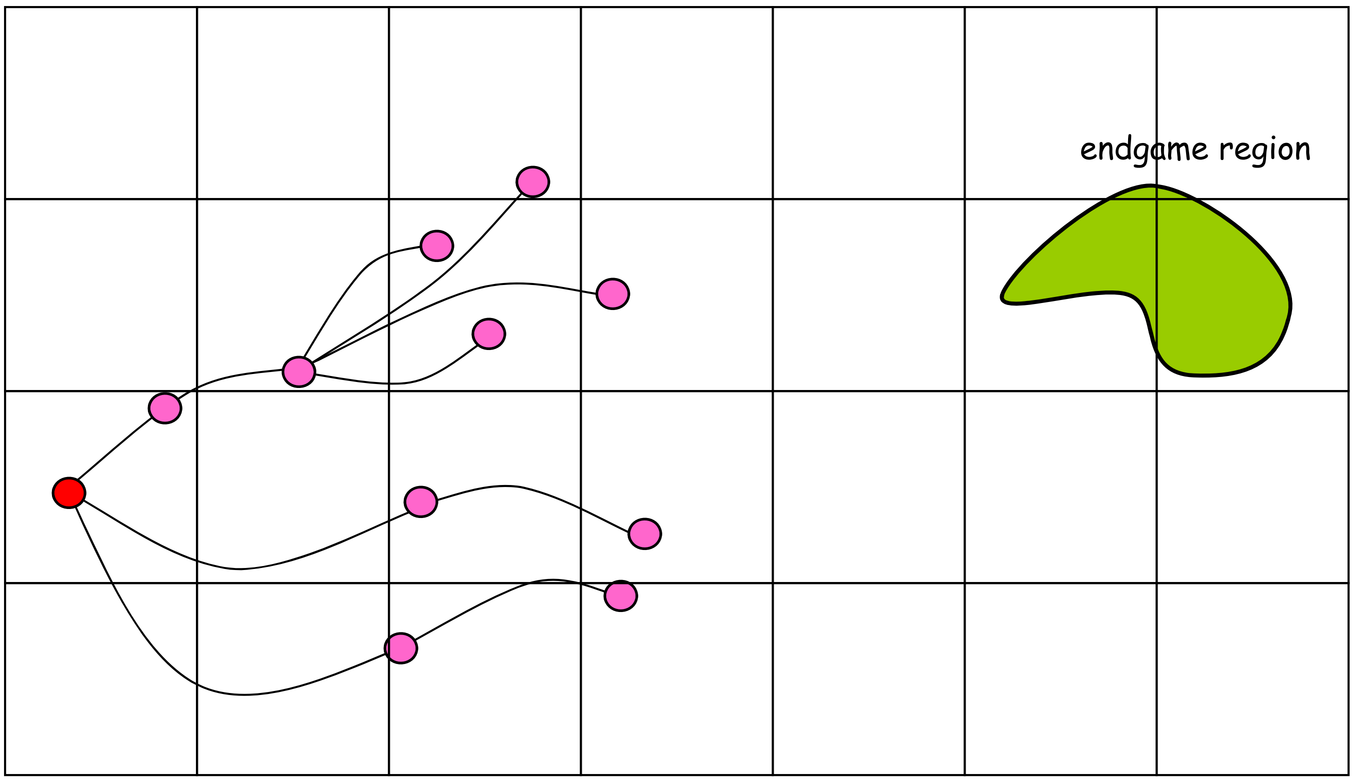


Direct Planning

- Examples – tractor trailer limited to 45 degree turns

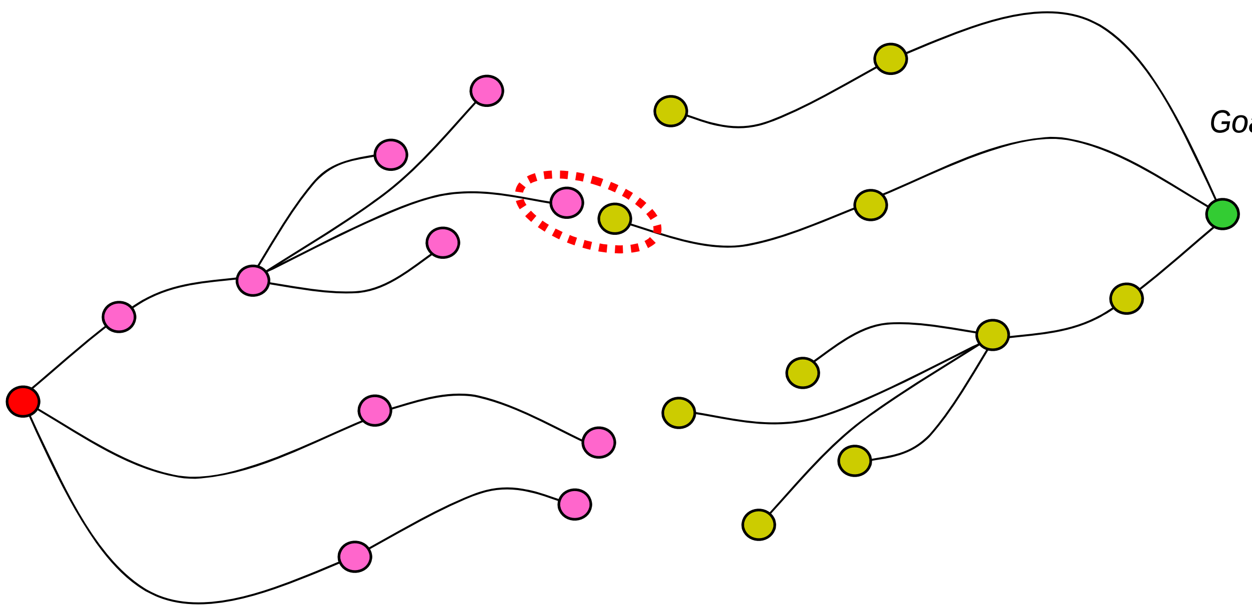


Sampling Strategy



$$p(m_i) \propto \frac{1}{\rho(m_i)}$$

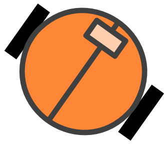
Bi-Directional Search: Forward & Backward Integration



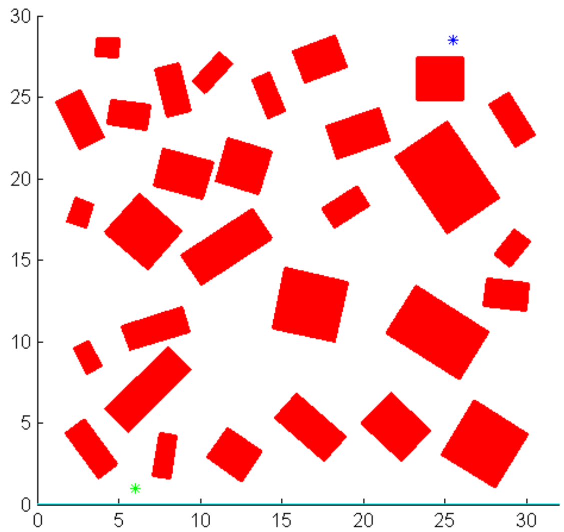
Direct PRM

- Example – The two wheeled robot

$$\begin{bmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \end{bmatrix} = g(x_{t-1}, u_t) = \begin{bmatrix} x_{1,t-1} + u_{1,t} \cos x_{3,t-1} dt \\ x_{2,t-1} + u_{1,t} \sin x_{3,t-1} dt \\ x_{3,t-1} + u_{2,t} dt \end{bmatrix}$$



- Environment – the 30 obstacle maze

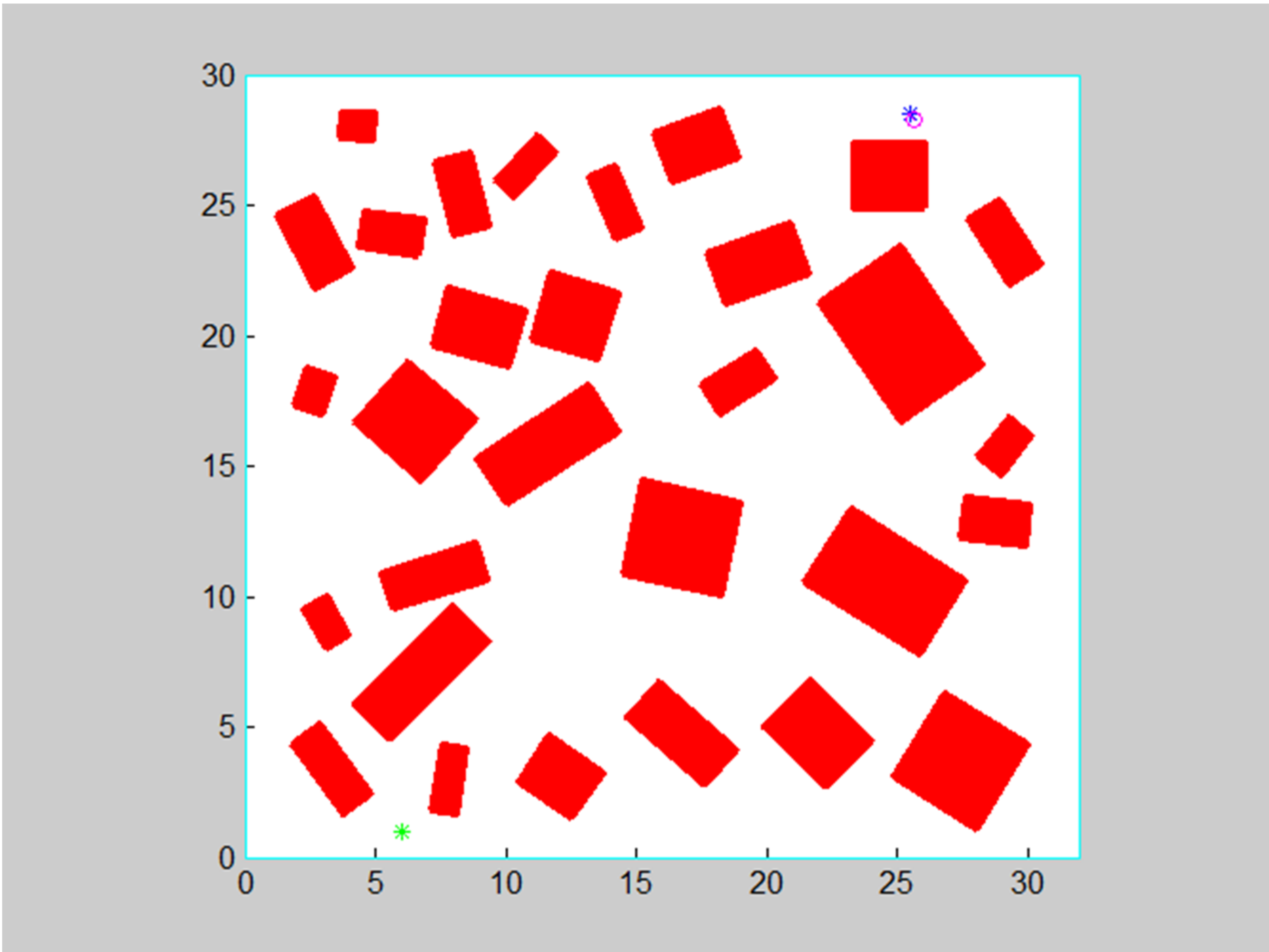


Direct Planning

- Control Based Sampling Algorithm
  1. Initialize with start node as only milestone
     1. Choose a milestone to expand
        1. Sort milestones in order of distance to goal
        2. Assign weights with exponential decay
        3. Sample using weighted sampling technique
     2. Expand the chosen milestone
        1. Select random number of integration timesteps from 30-100
        2. Select random control inputs within feasible range
        3. Propagate dynamics to generate trajectory
        4. Check for collision and repeat if not valid path
     3. Add endpoint as a new milestone
     4. Test end condition
        1. If new milestone is within 0.5 of end point, terminate

Direct PRM

The algorithm in practice

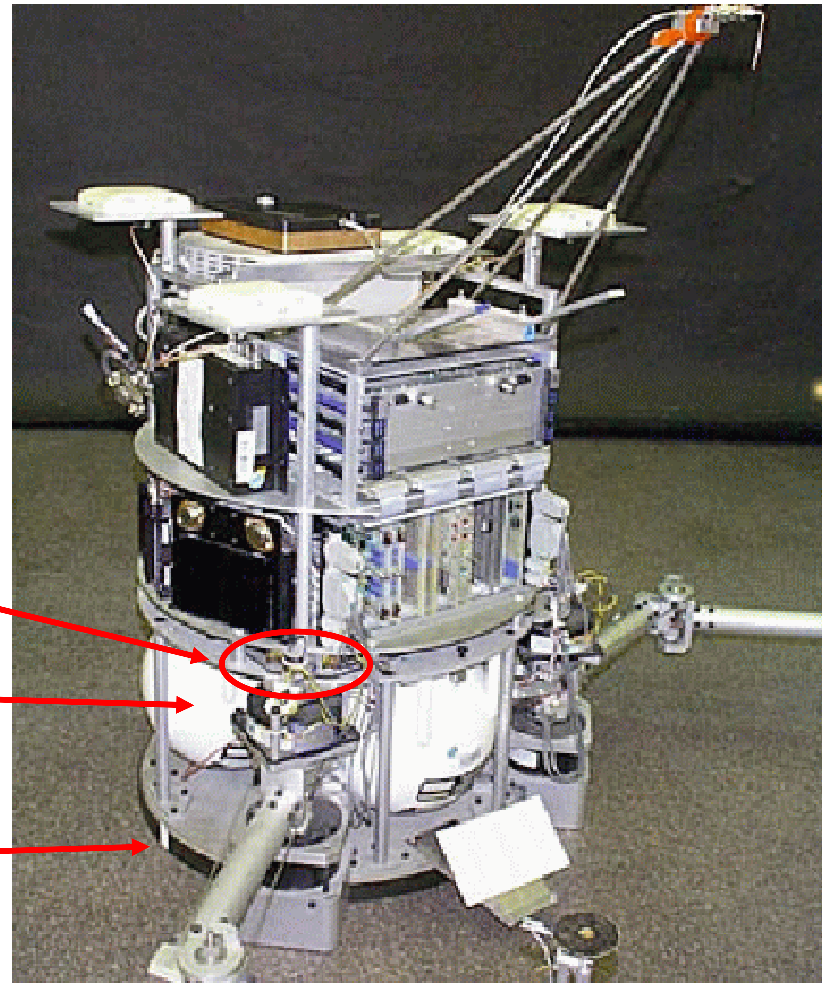


Direct Planning

- Statistics
  - Total runtime: 10.6s
  - Number of milestones: 417
  - Number of milestones on path: 44
  - Approx length of path: 45.0
  - Visibility path length: 34

Example: Space Robot

Robot created to study issues in robot control and planning in zero-gravity space environment



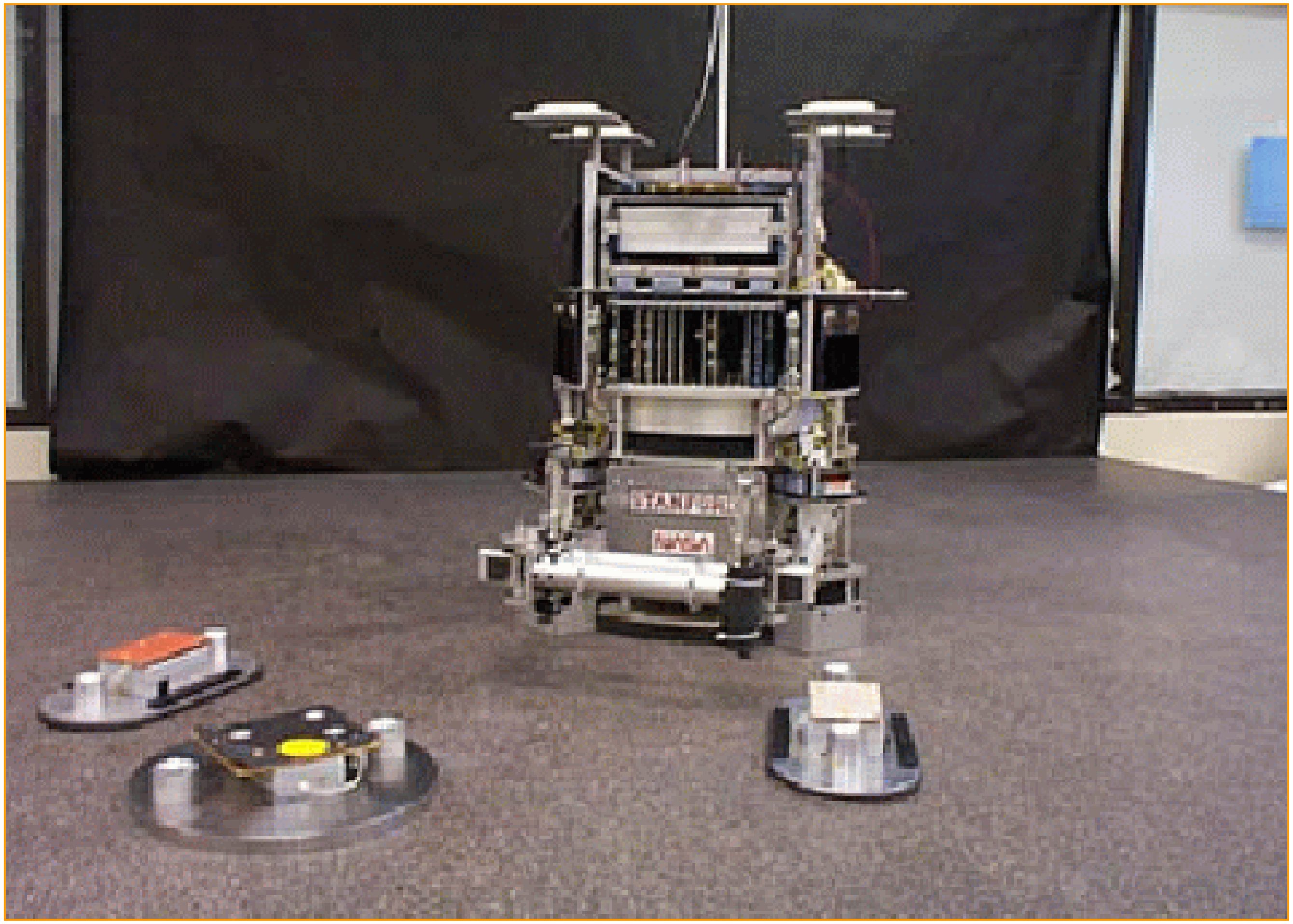
air thrusters

gas tank

air bearing

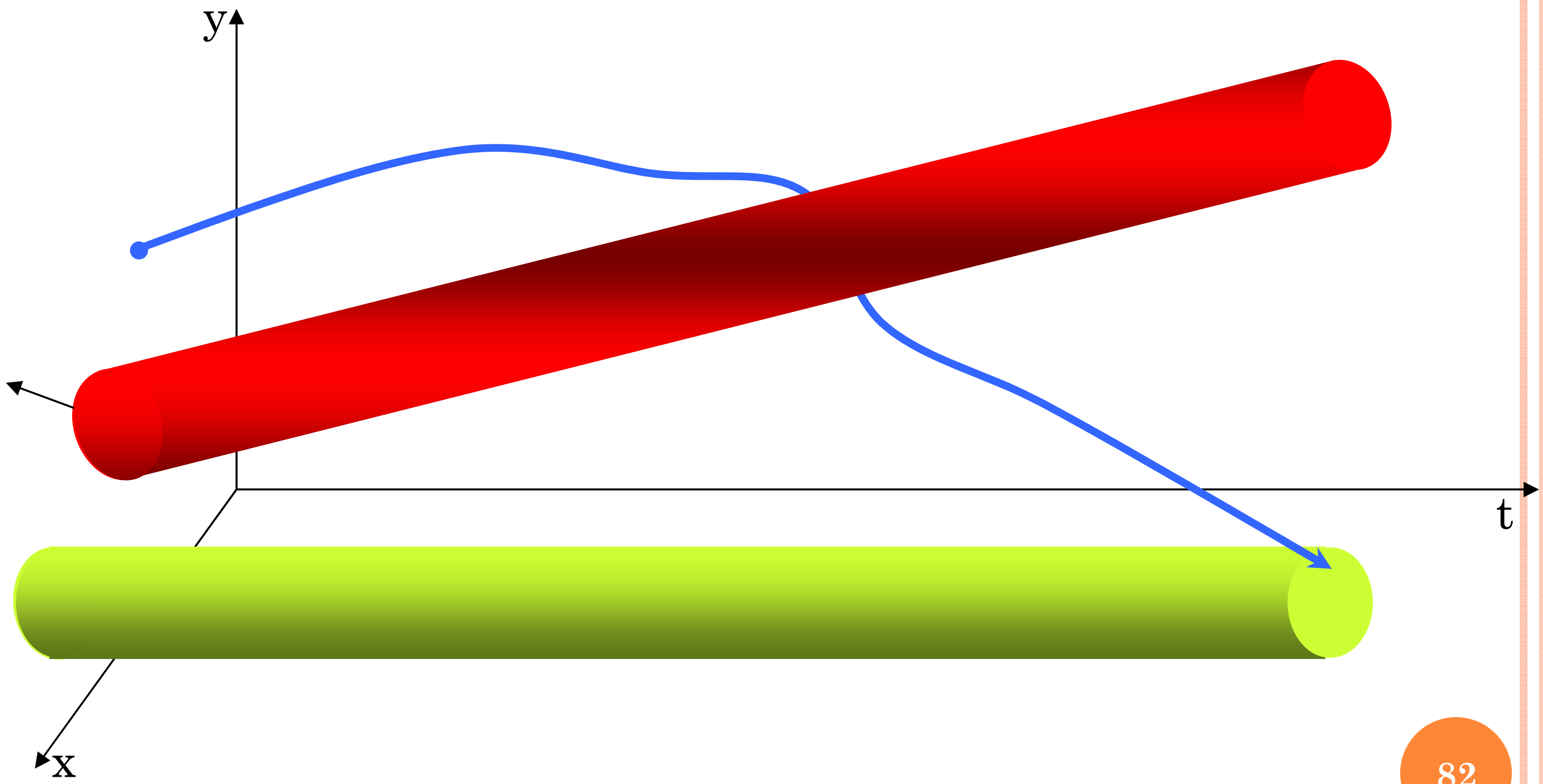
80

Navigation Among Moving Obstacles

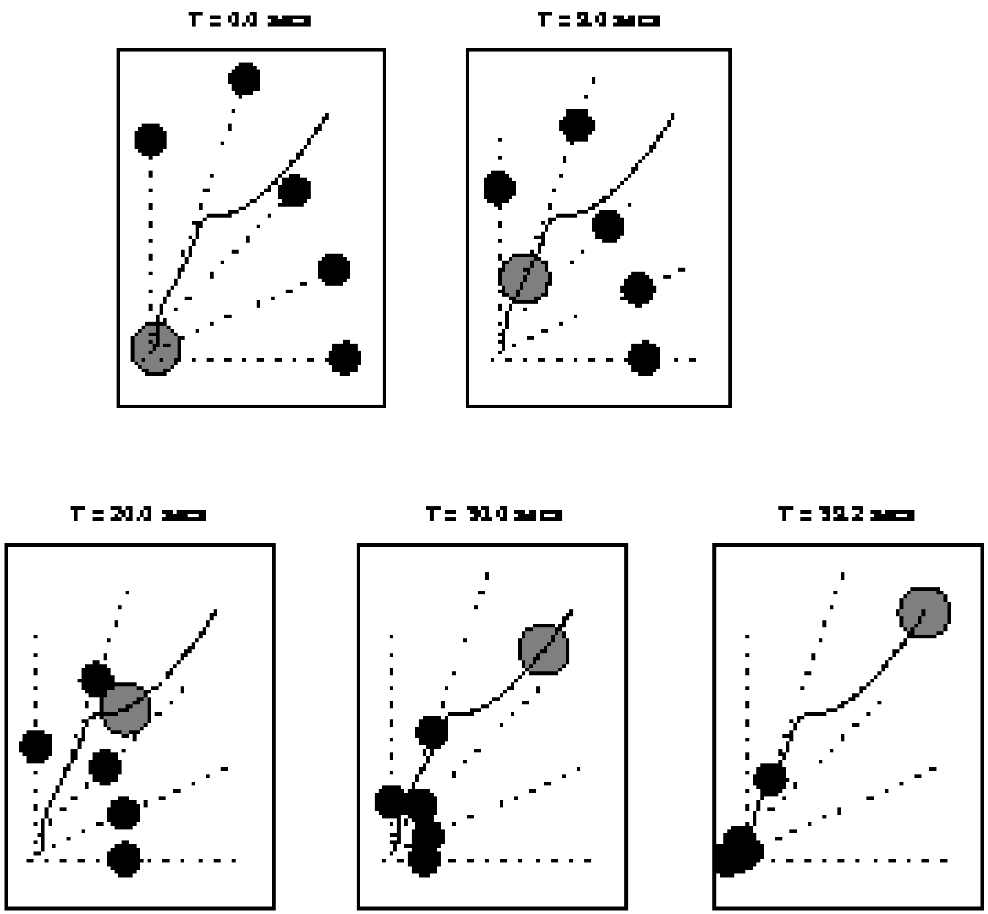


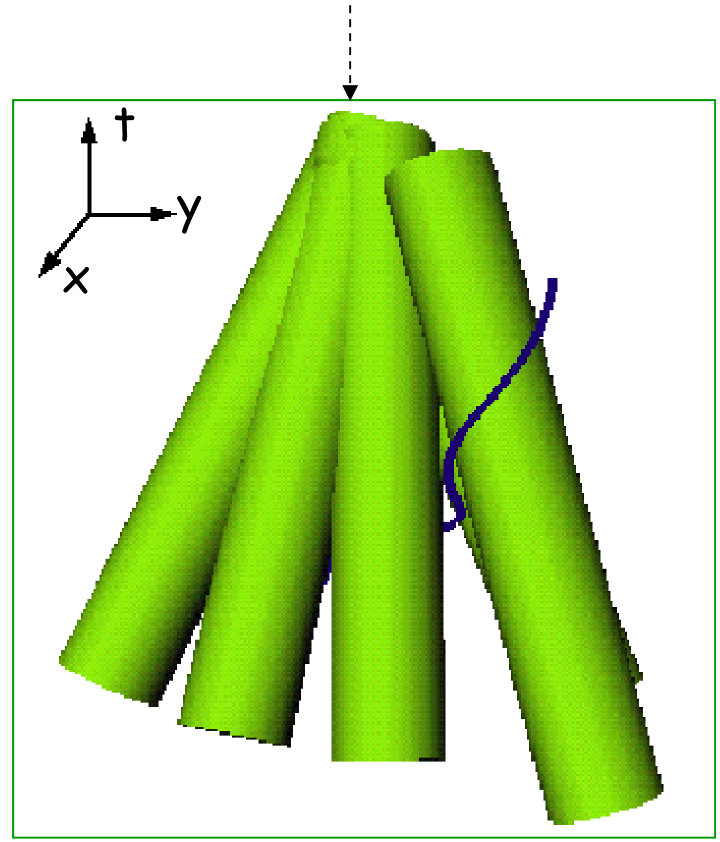
81

Moving Obstacles in Configuration x Time Space



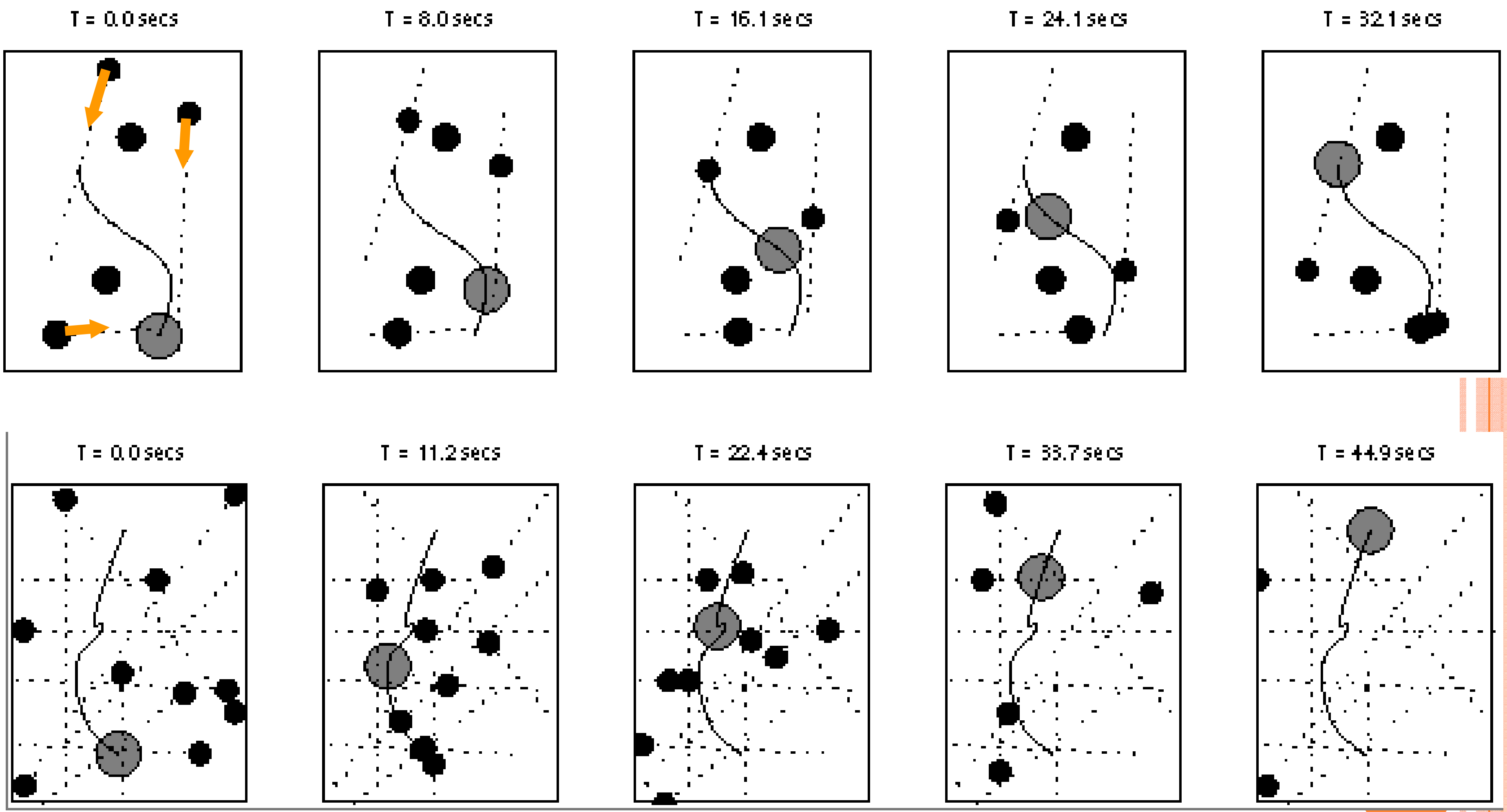
EXAMPLE RUN





Obstacle map to cylinders in configuration×time space

Other Examples



Direct Planning

- Potential improvements
  - Control sampling strategy
    - Deterministic, or combination
  - Milestone selection for expansion
    - Distance to goal restricts exploration
    - Can avoid oversampling in an area by keeping track of spatial location of milestones (histogram binning)
  - Trajectory collision checking strategy
    - Currently checking all points individually, could switch to path covering approach

DIRECT PLANNING

- Improvements
  - It is also possible to select intermediate goal points in the work space to help push the trajectory to exploration.
    - Sample a location per PRM sampling strategies
    - Sample and apply inputs until goal is achieved
  - This is known as the Rapidly-expanding Random Tree (RRT)
    - Improved ability to search space with kinodynamic constraints on vehicle motion

Direct Planning

- Example of RRT approach
  - Gaussian goal selection

$$\sim N\left(0, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right)$$

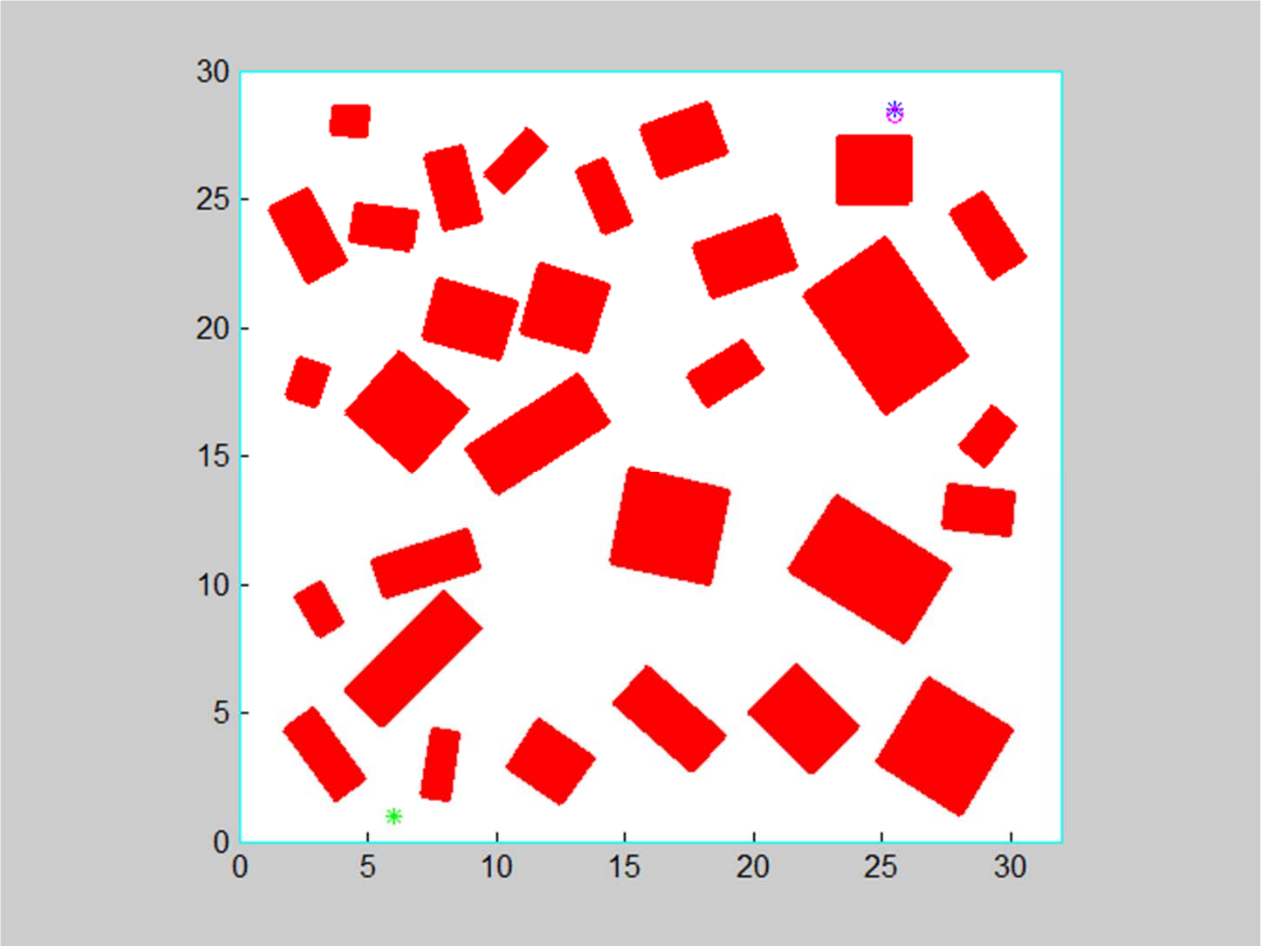
    - Also tried uniform 4x4 square

  - Random input selection until goal is achieved

  - Remaining elements of PRM are identical
    - Expand milestones with weights based on distance to goal
      - Exponential

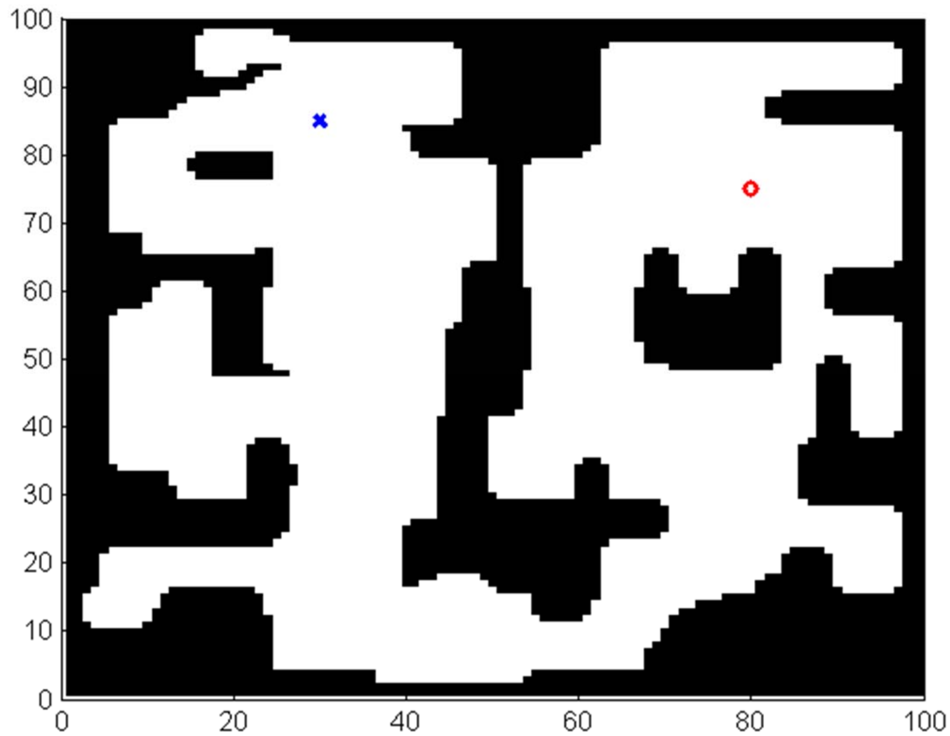
DIRECT PLANNING

- RRT Example
  - Link to video



2009 Exam Question #3

- Apply the PRM to the following cistern

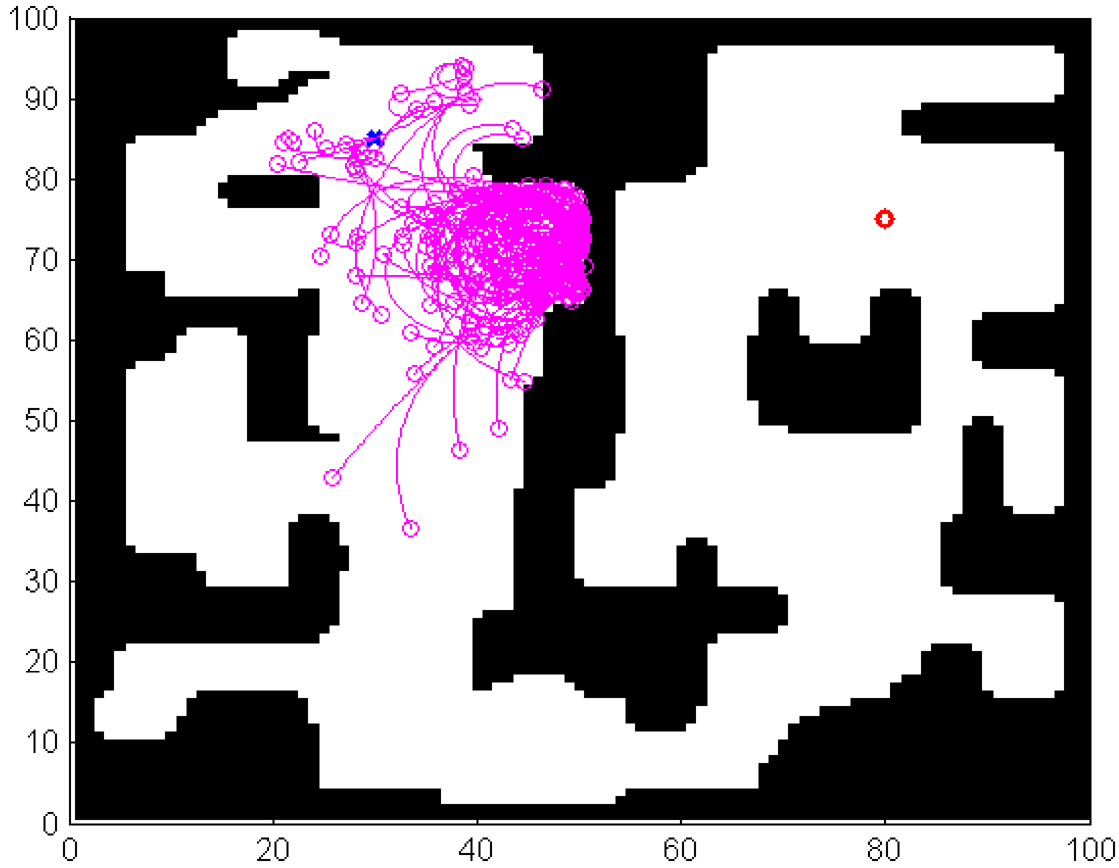


2009 Exam Question #3

- Solution
  - The stock PRM code needed to be adapted in only a couple of ways to get a basic answer
    - Remove environment generation and add in map of cistern
    - Change vehicle capabilities
    - Change collision checking to work with occupancy grid map

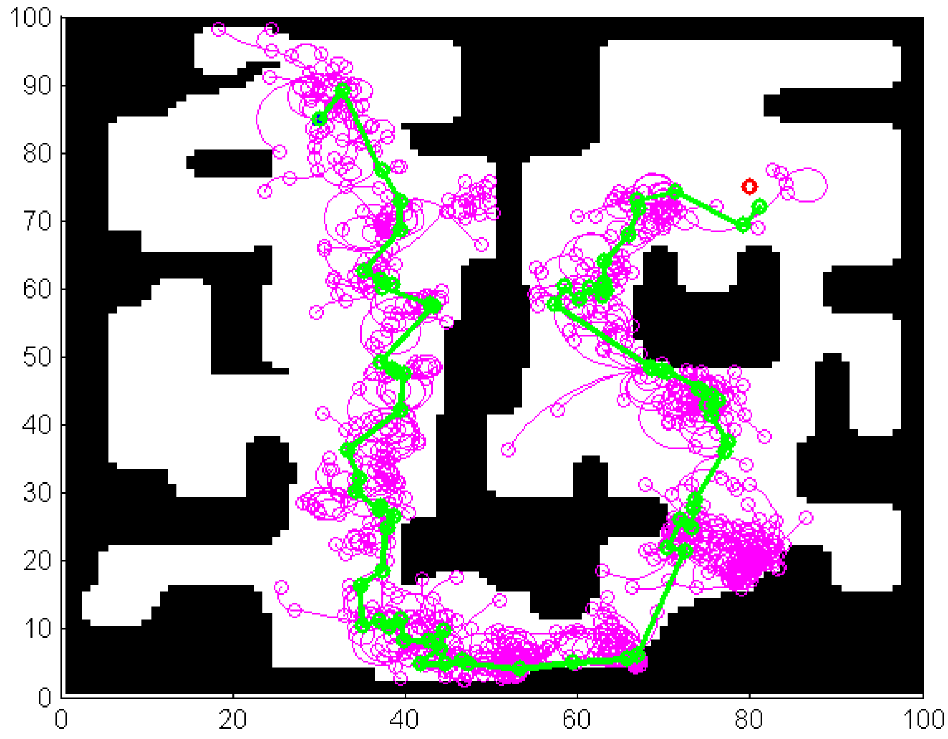
2009 Exam Question #3

- The result was this



2009 Exam Question #3

- The fun part was modifying the node selection component to better explore the environment
  - Peiyi Chen's solution
    - Use knowledge of the shape of the environment to change direction of node selection (hmmmmmm).



Recent Work At U of Toronto