

Robot Motion Planning

Rafael Murrieta Cid
CIMAT

Email: murrieta@cimat.mx

Web Page:
<http://www.cimat.mx/~murrieta/>

Slides were borrowed from
Profs. J.C. Latombe and J.P. Laumond

Goal of Motion Planning

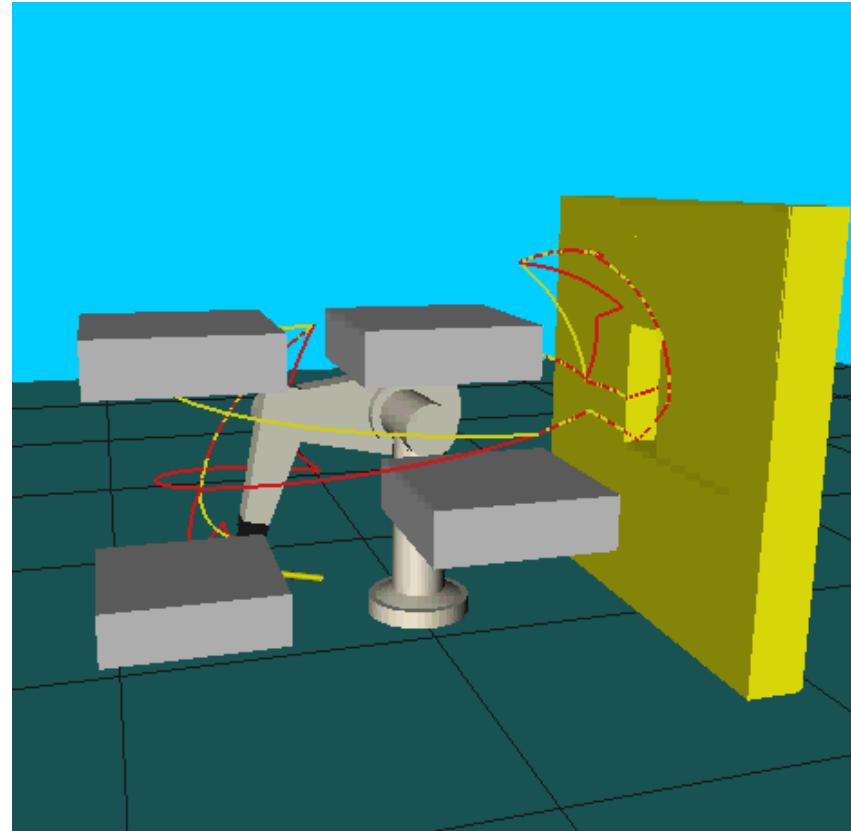
- Compute **motion strategies**, e.g.:
 - geometric paths
 - time-parameterized trajectories
 - sequence of sensor-based motion commands
- To achieve **high-level goals**, e.g.:
 - go to A without colliding with obstacles
 - assemble product P
 - build map of environment E
 - find object O

Basic Motion Planning Problem

Compute a **collision-free path** for a rigid or articulated object among static obstacles

- **Inputs:**
 - Geometry of moving object and obstacles
 - Kinematics of moving object (degrees of freedom)
 - Initial and goal **configurations** (placements)
- **Output:**

Continuous sequence of collision-free robot configurations connecting the initial and goal configurations

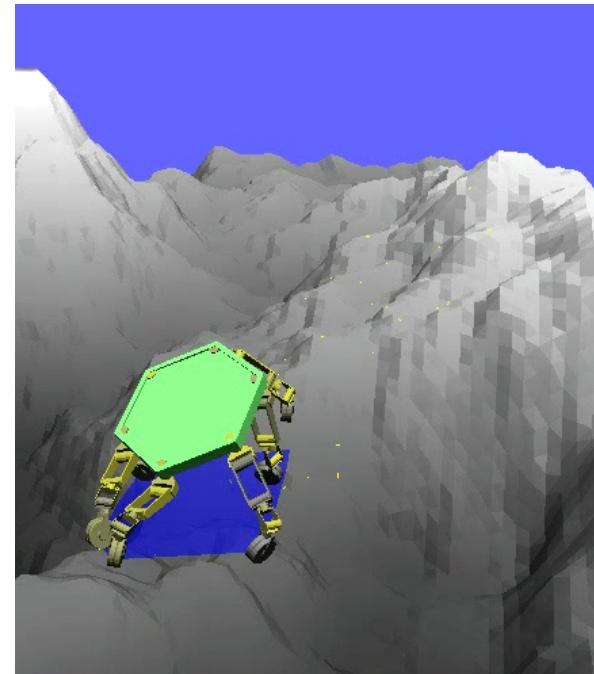
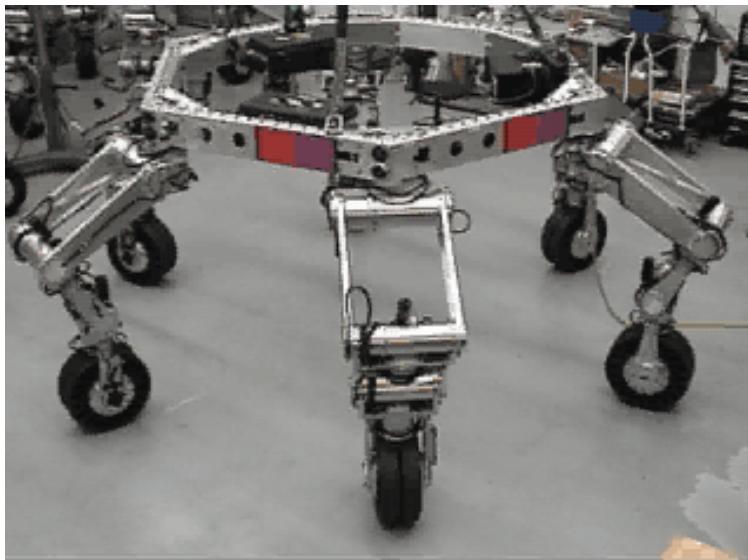
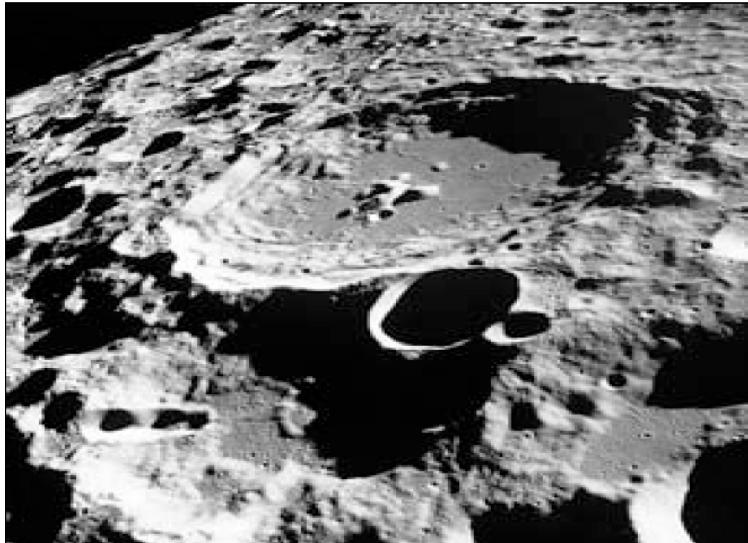


Extensions of Basic Problem

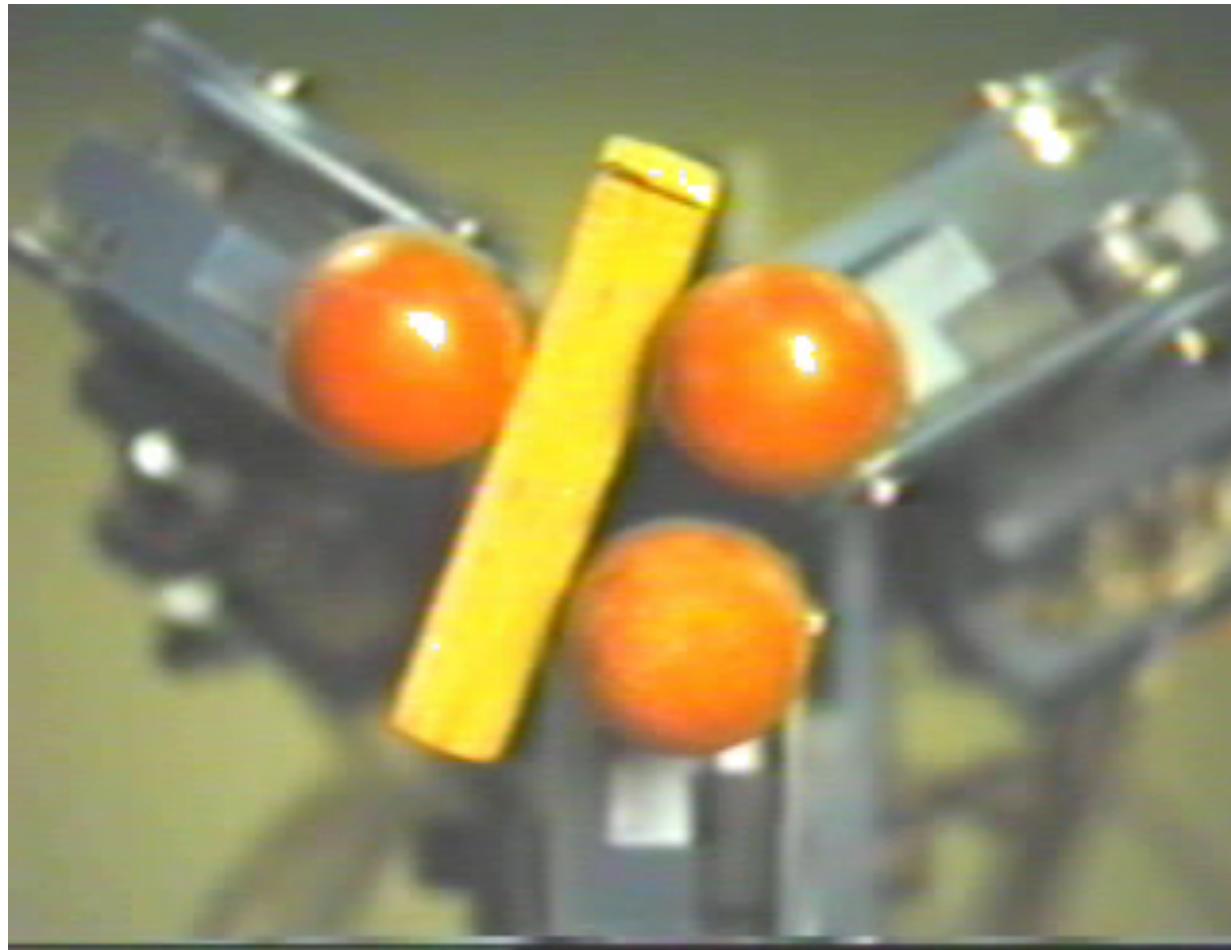
- Moving obstacles
- Multiple robots
- Movable objects
- Assembly planning
- Goal is to acquire information by sensing
 - Model building
 - Object finding/tracking
 - Inspection
- Nonholonomic constraints
- Dynamic constraints
- Stability constraints
- Optimal planning
- Uncertainty in model, control and sensing
- Exploiting task mechanics (sensorless motions, underactuated systems)
- Physical models and deformable objects
- Integration of planning and control
- Integration with higher-level planning

Some Applications

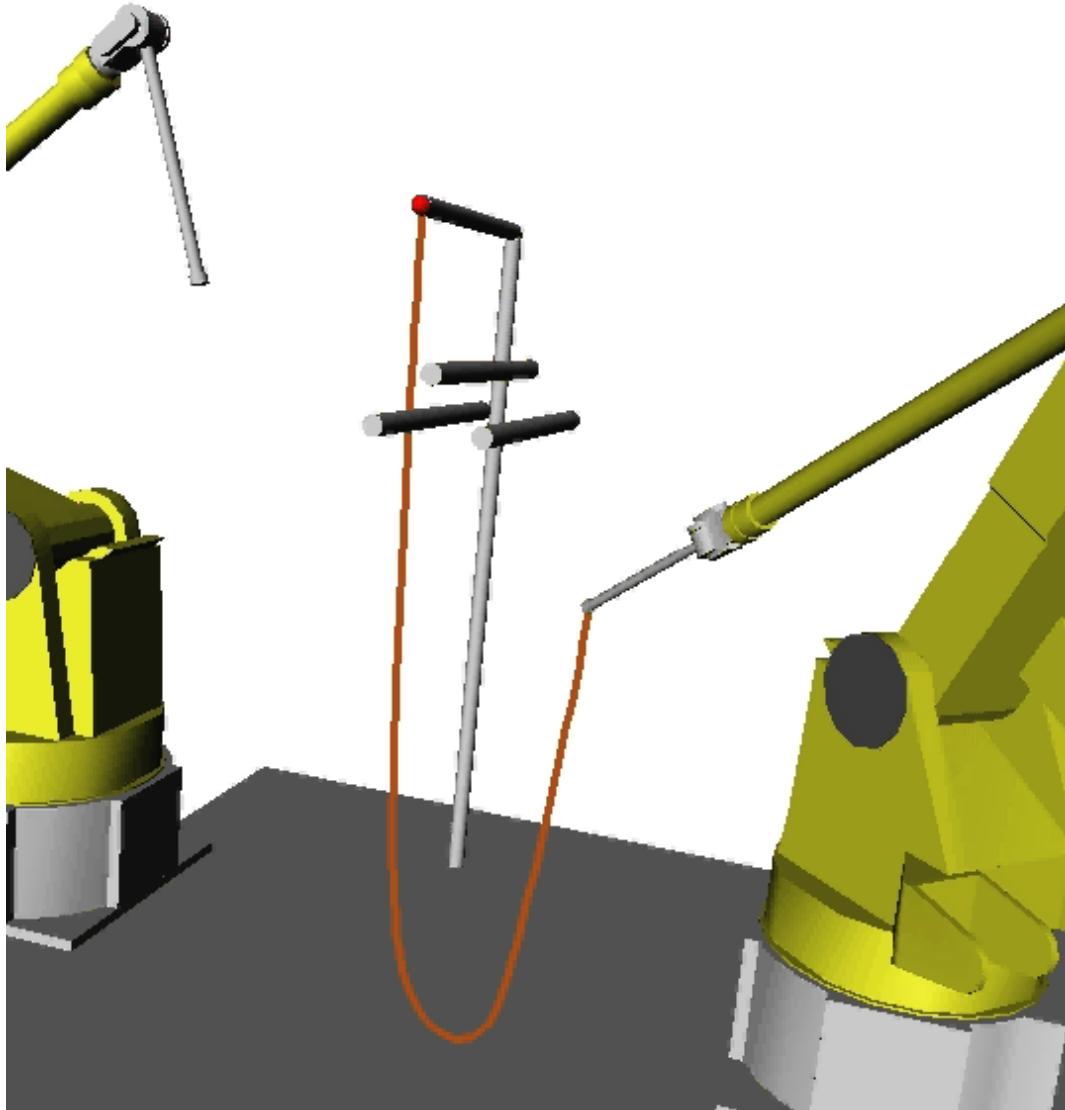
Lunar Vehicle (ATHLETE, NASA/JPL)



Dexterous Manipulation

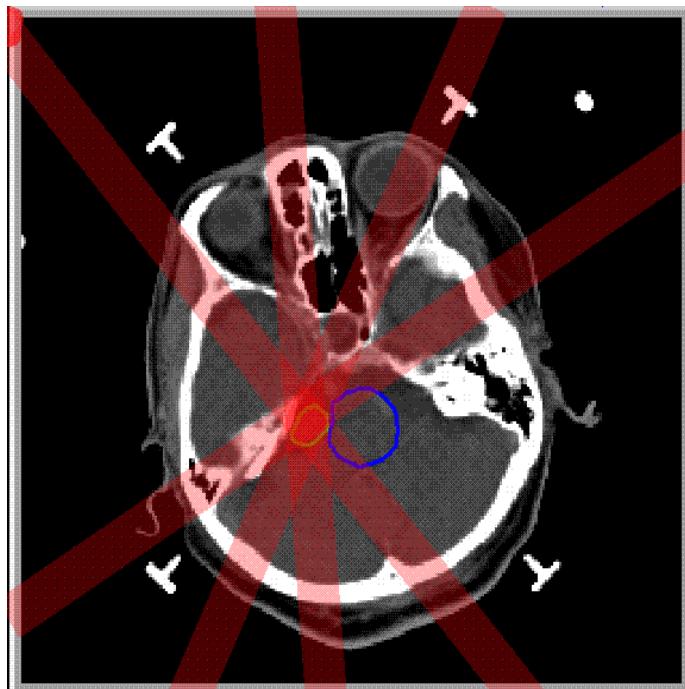


Manipulation of Deformable Objects



Topologically
defined goal

Radiosurgical Planning



CyberKnife (Accuracy)

INTEGRATION OF TWO REVOLUTIONARY TECHNOLOGIES

Proprietary Image-Guidance System
Locates and monitors tumor location to enable automatic beam penumbra for tumor movement.

Multi-Jointed Robotic Arm
Enables access to previously unreachable tumors and reduces damage to surrounding normal structures.

Integration of these unique technologies allows physicians to treat complex-shaped tumors with clinically proven accuracy that has been demonstrated to be comparable, if not superior, to frame-based radiosurgical systems.¹

Simple Outpatient Treatment Process

Planning: CT scanning and customized treatment planning can be completed in minutes.

Positioning: The patient lies on a table with only a face mask or body mold used for immobilization.

Verification: The image-guidance system verifies tumor location and compares it to previously stored data.

Targeting: When tumor movement is detected, the robotic arm is repositioned within a fraction of a second.

Reports: This verification process is repeated prior to delivery of each radiation beam.

Treatments: Beams of finely collimated radiation beams deliver precision radiosurgery to the tumor.

Completion: Following CyberKnife[®] treatment, the patient goes home. There is zero recovery time.

CyberKnife[®] T³ Radiosurgery
A new standard in IMRT radiosurgery

100% freedom
Ability to achieve stereotactic accuracy.²

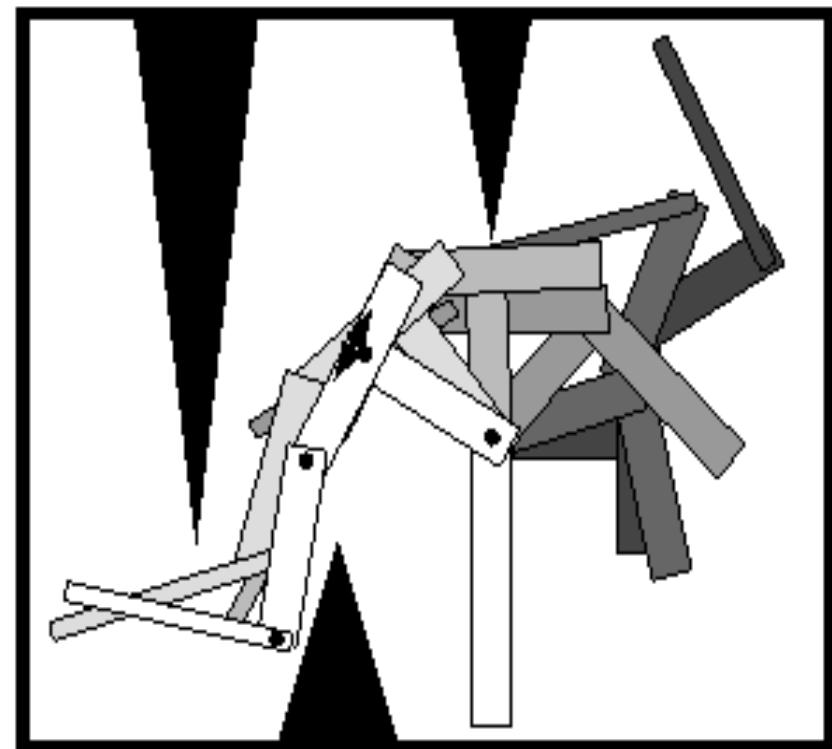
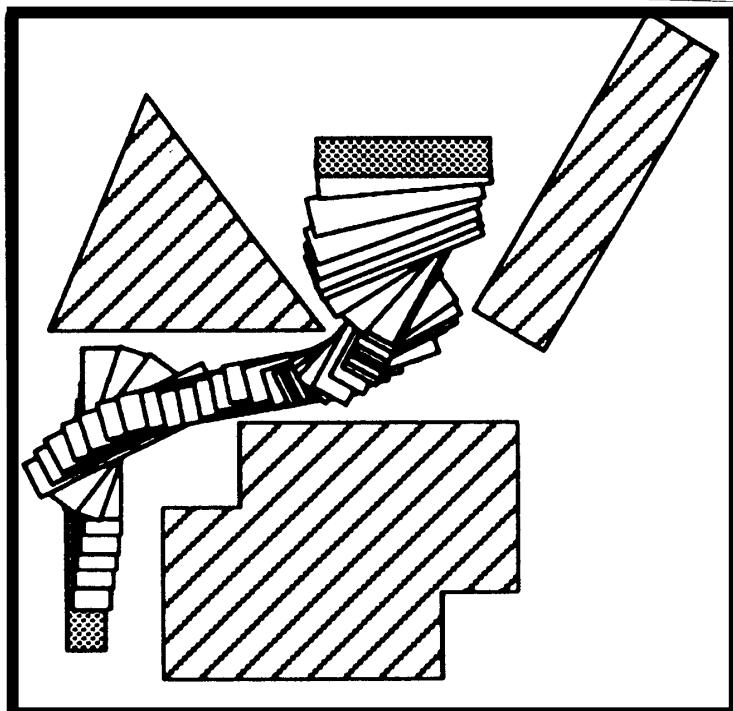
Isometric or non-isometric treatment planning
Up to 12X per second beam switching

1.04 1.04

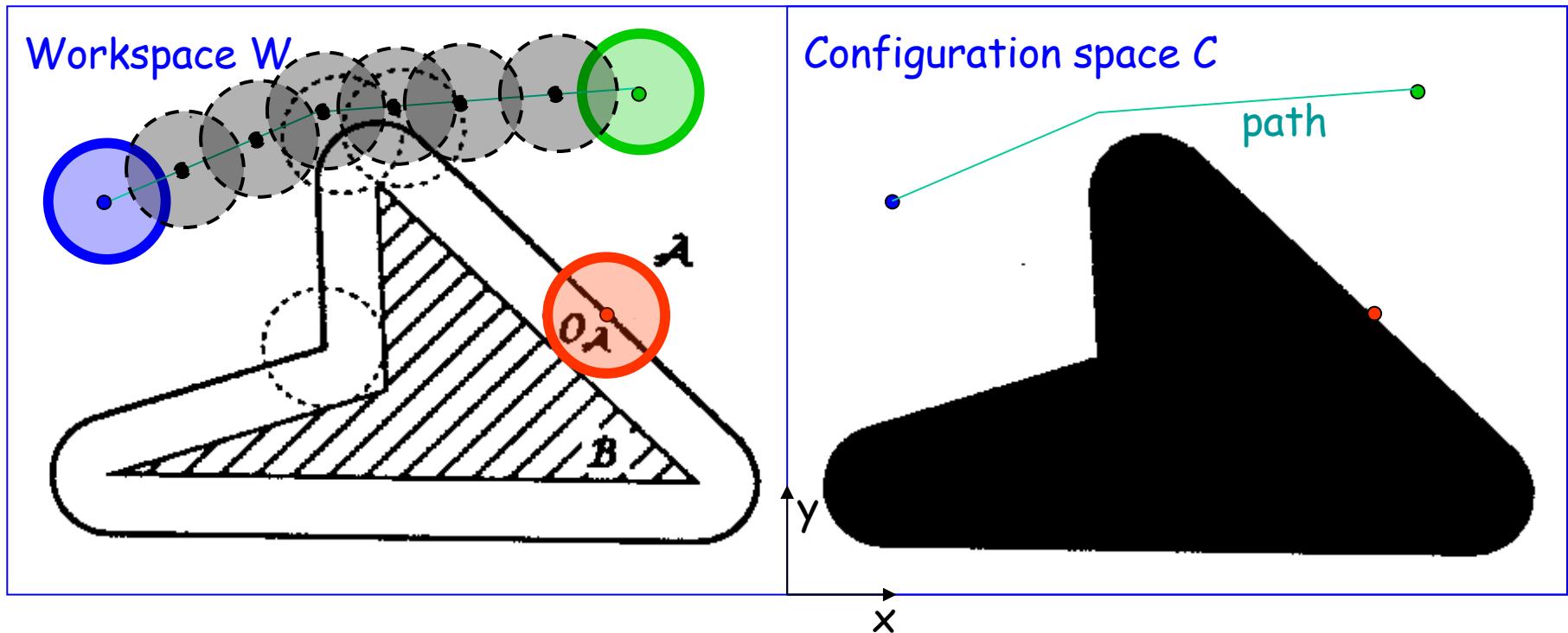
Transportation of A380 Fuselage through Small Villages



Paths



Disc Robot in 2-D Workspace

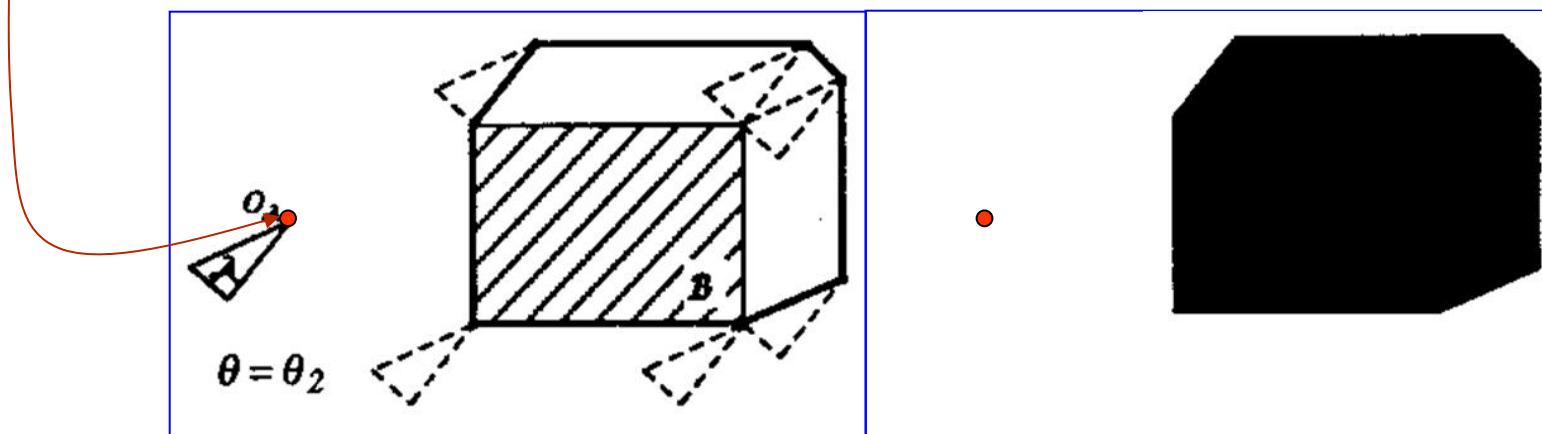
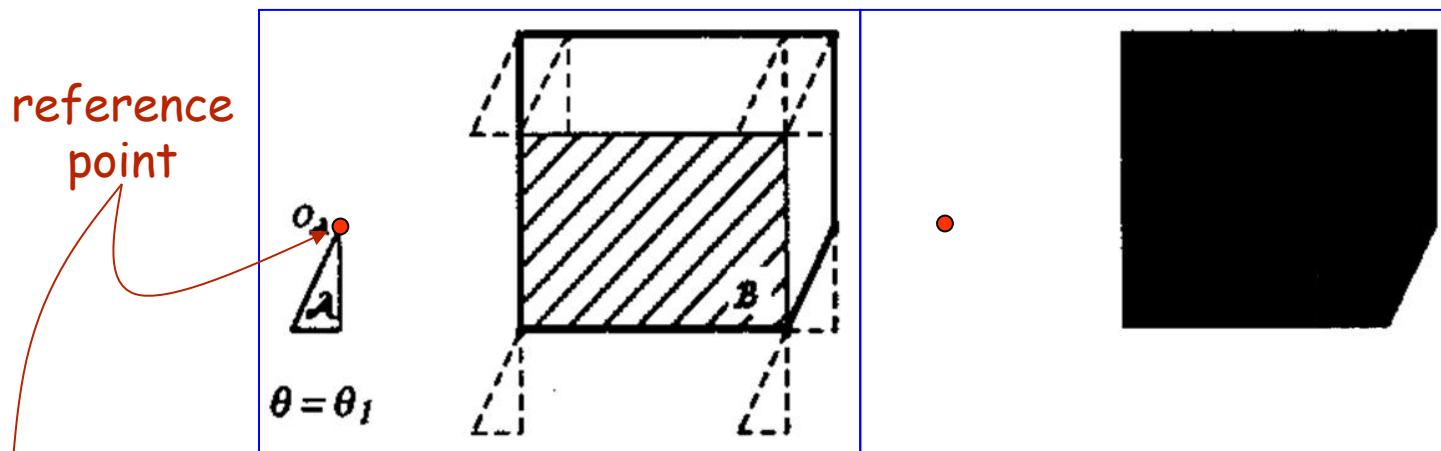


configuration = coordinates (x, y) of robot's center

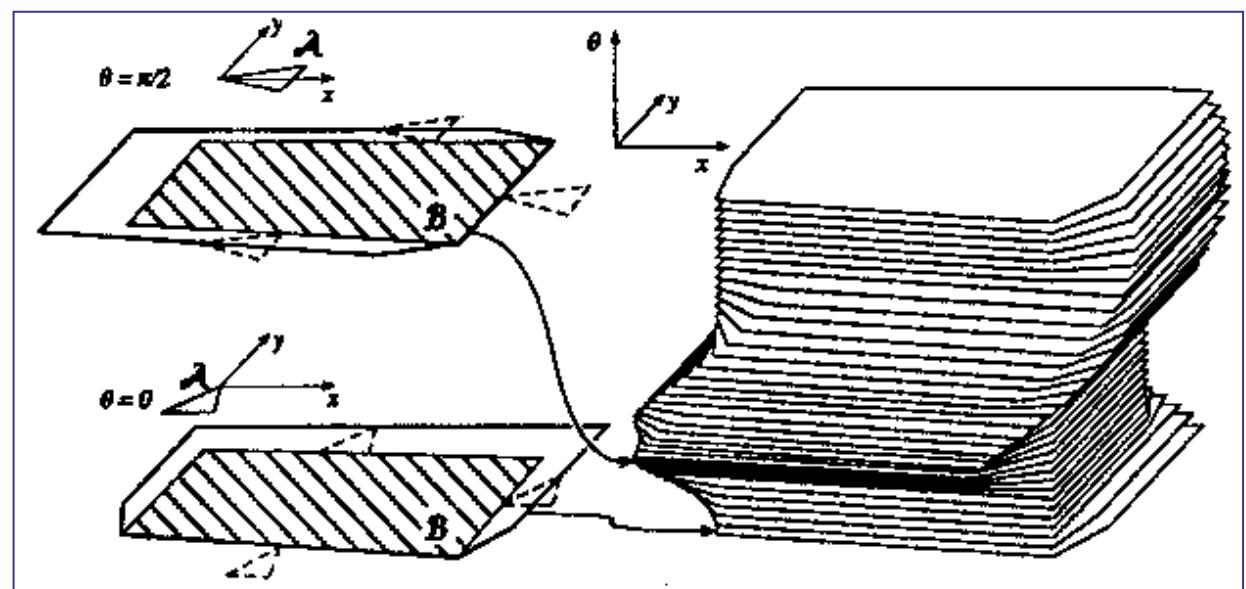
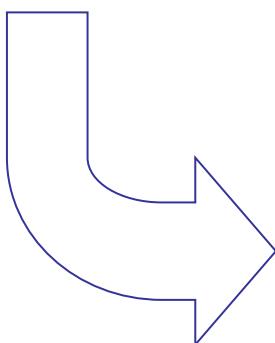
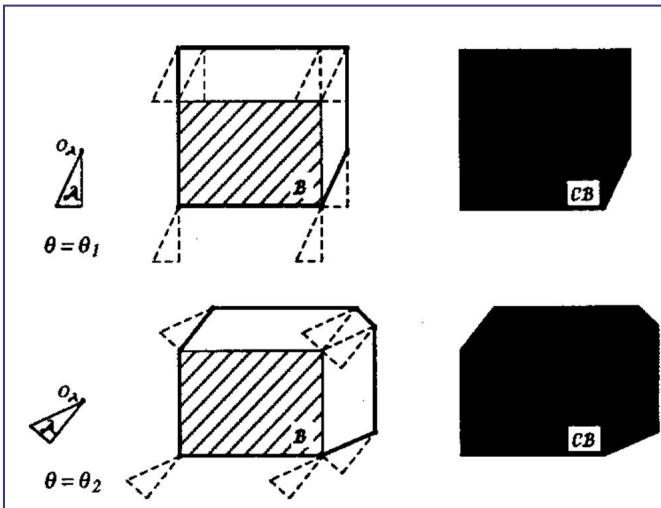
configuration space $C = \{(x, y)\}$

free space F = subset of collision-free configurations

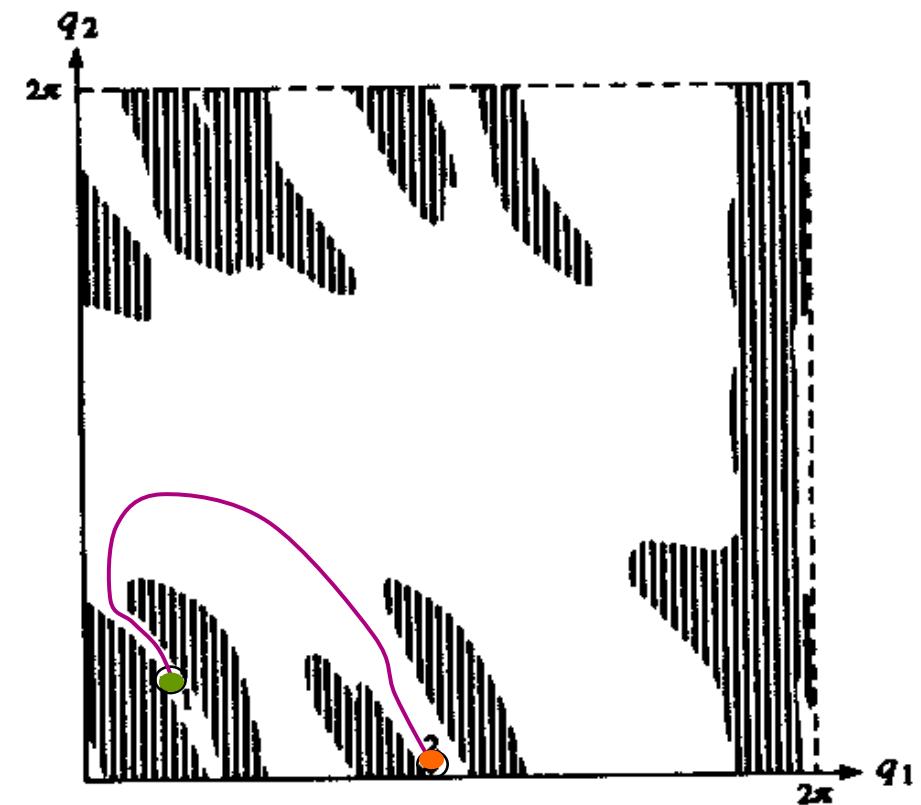
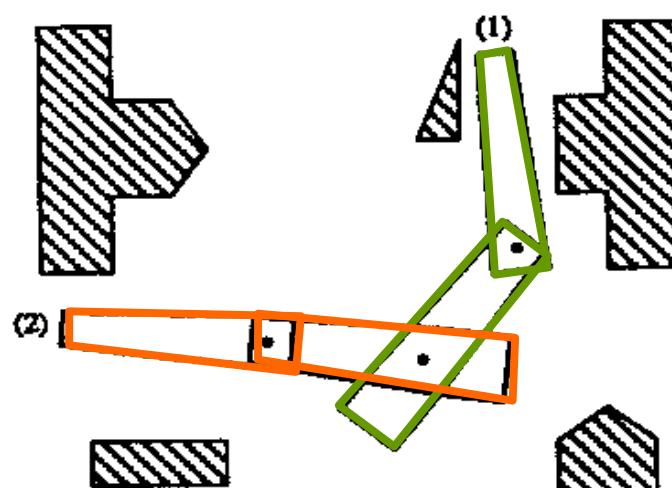
Translating Polygon in 2-D Workspace



Translating & Rotating Polygon in 2-D Workspace

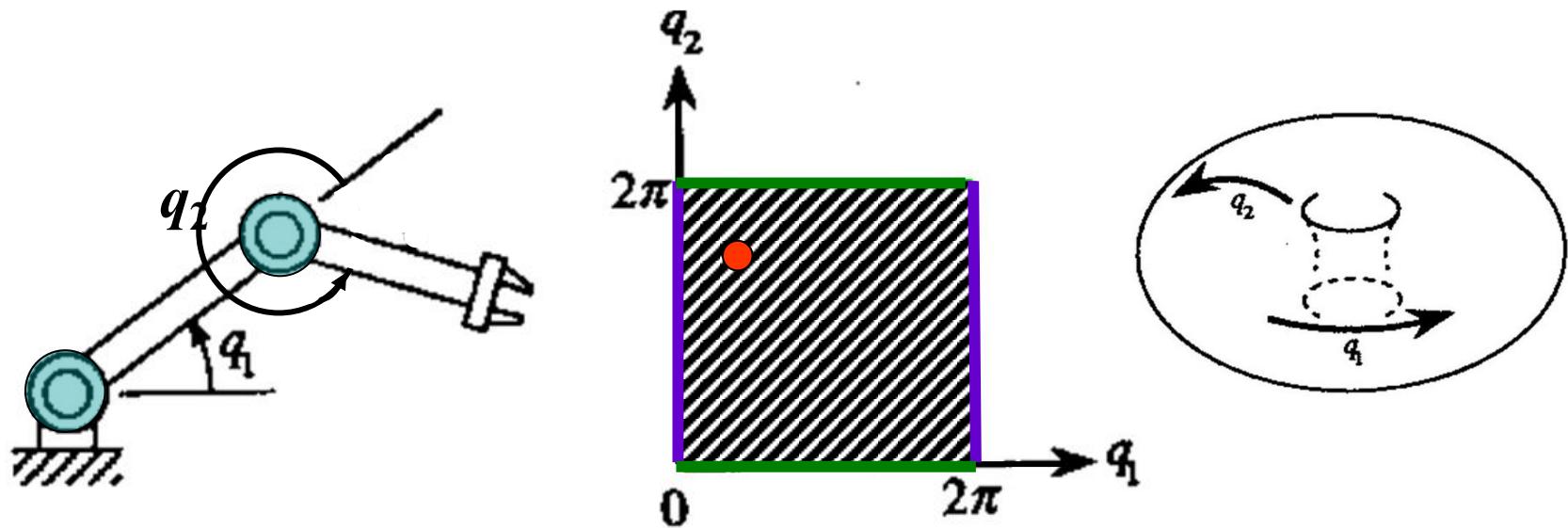


Tool: Configuration Space (C-Space C)

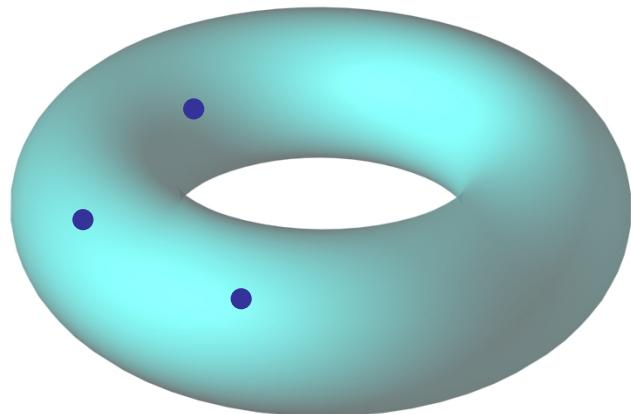
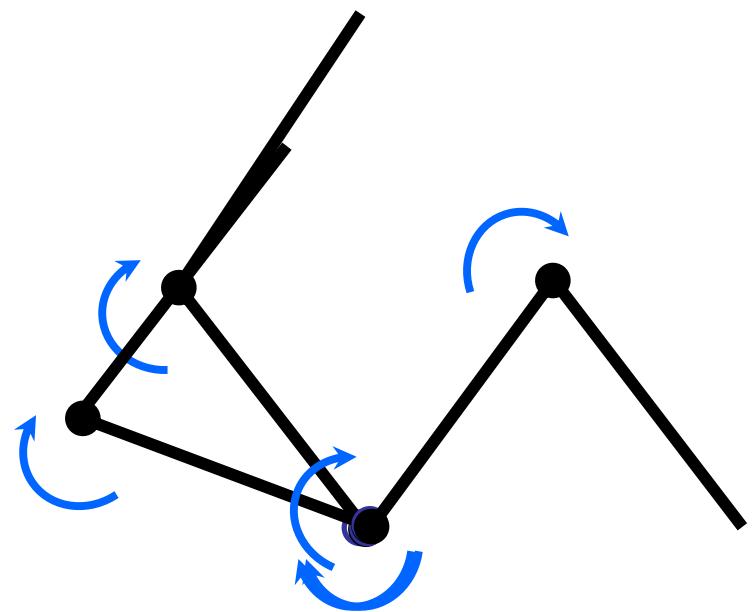


Configuration Space

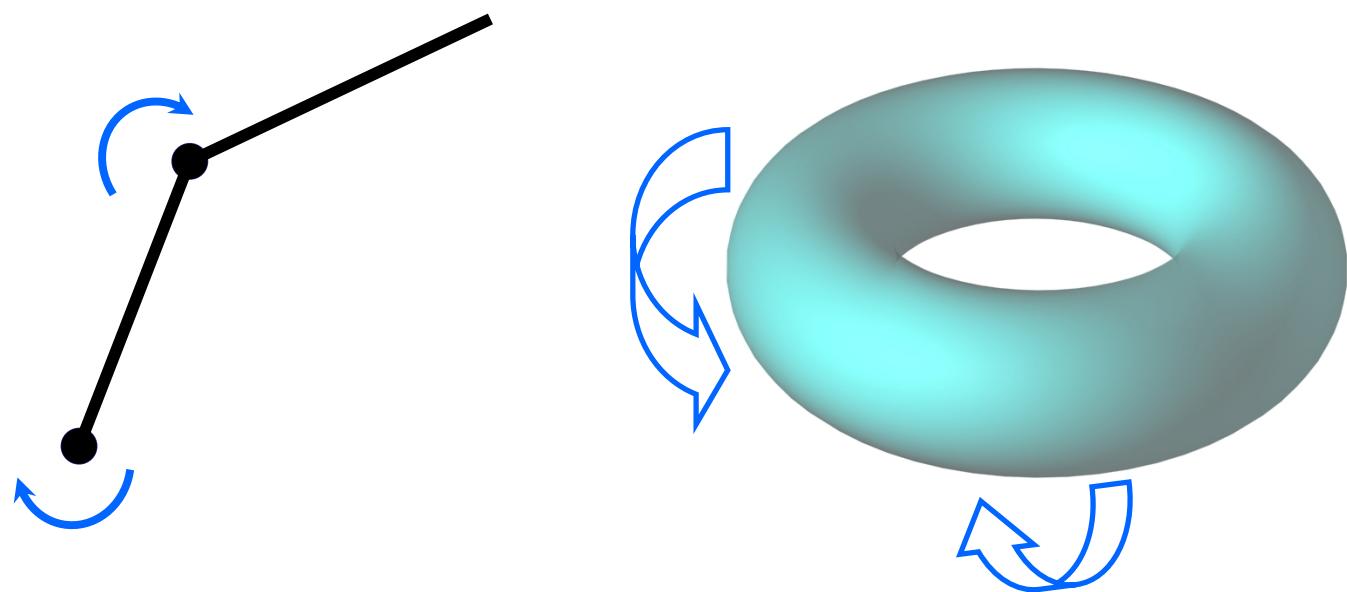
- Space of all its possible configurations
- But the topology of this space is usually not that of a Cartesian space



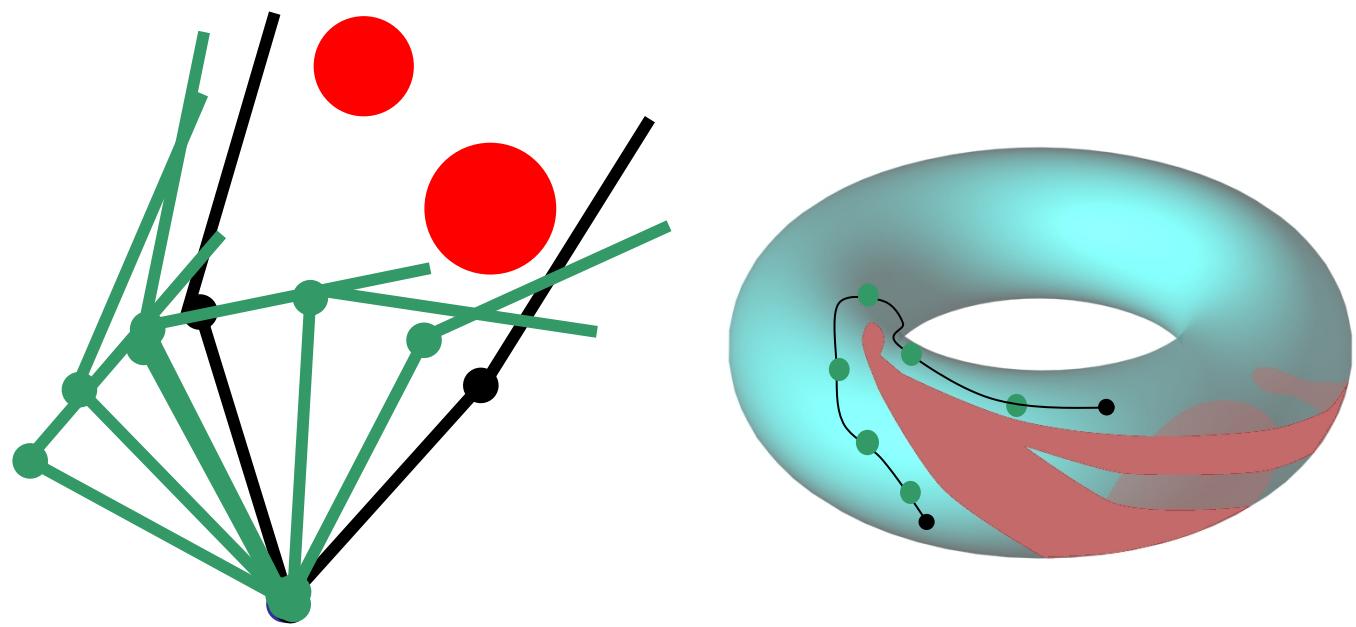
Move



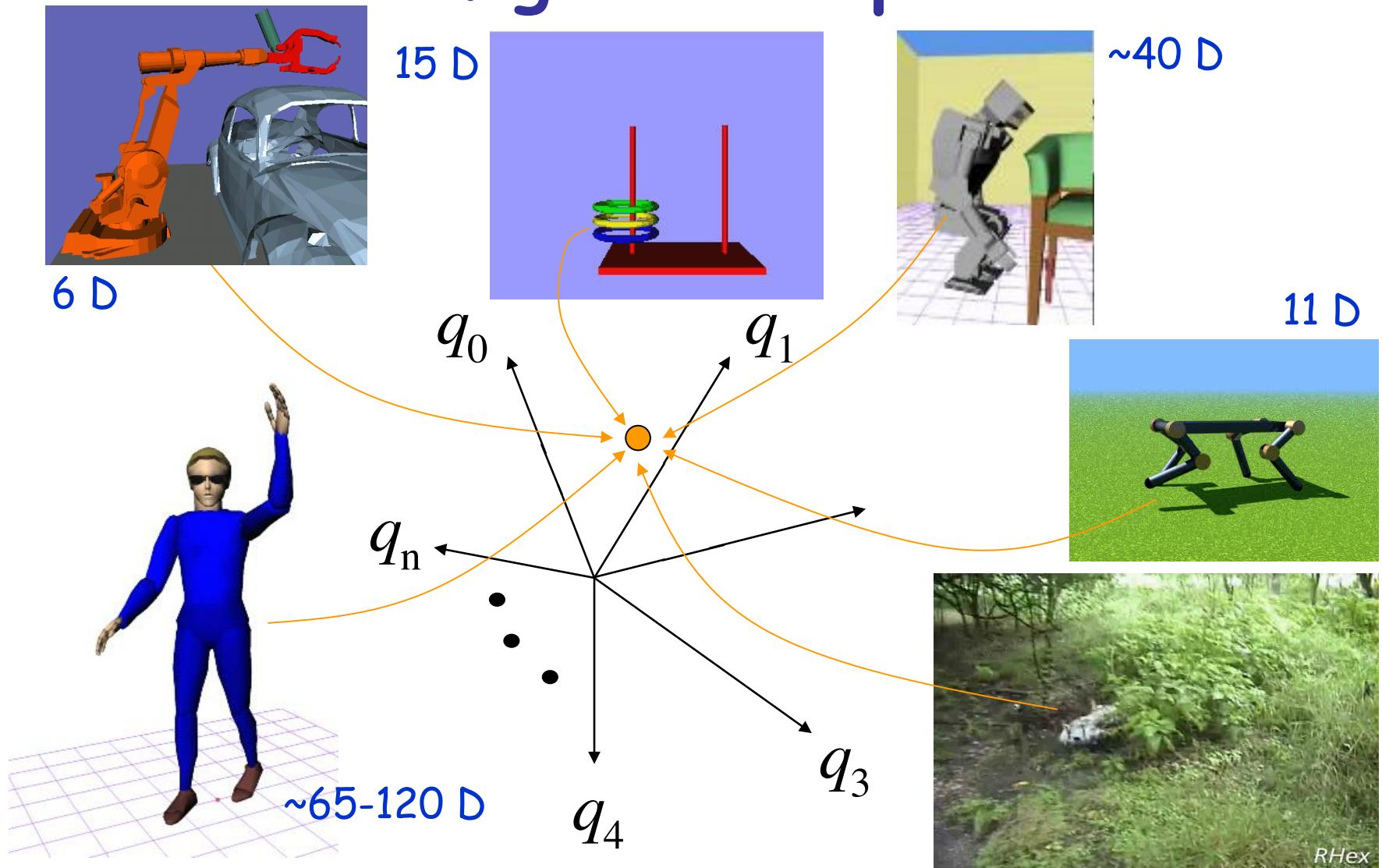
Configuration Space



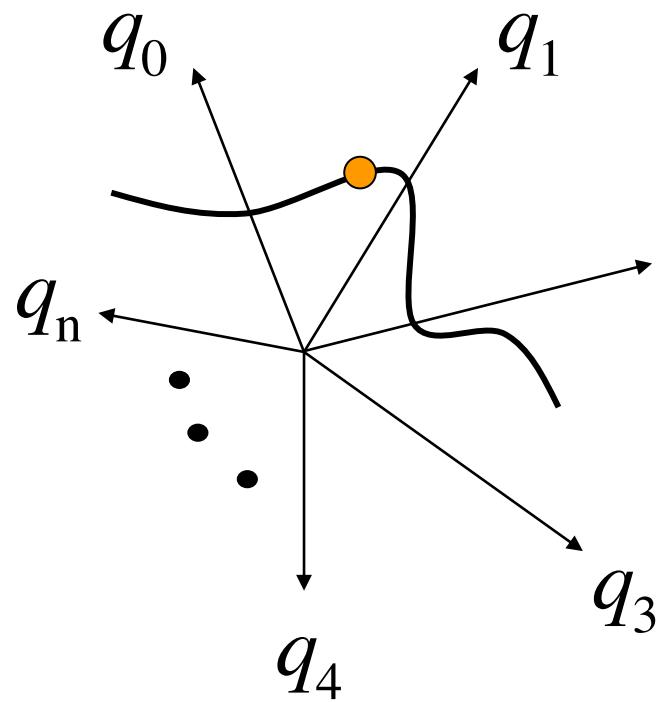
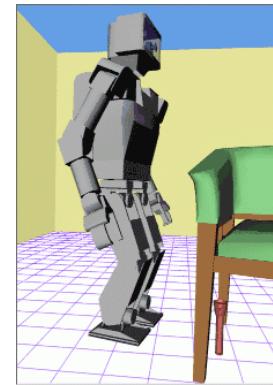
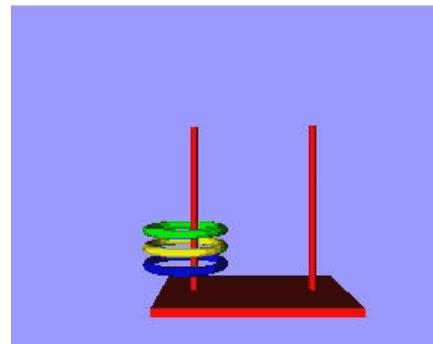
Configuration Space



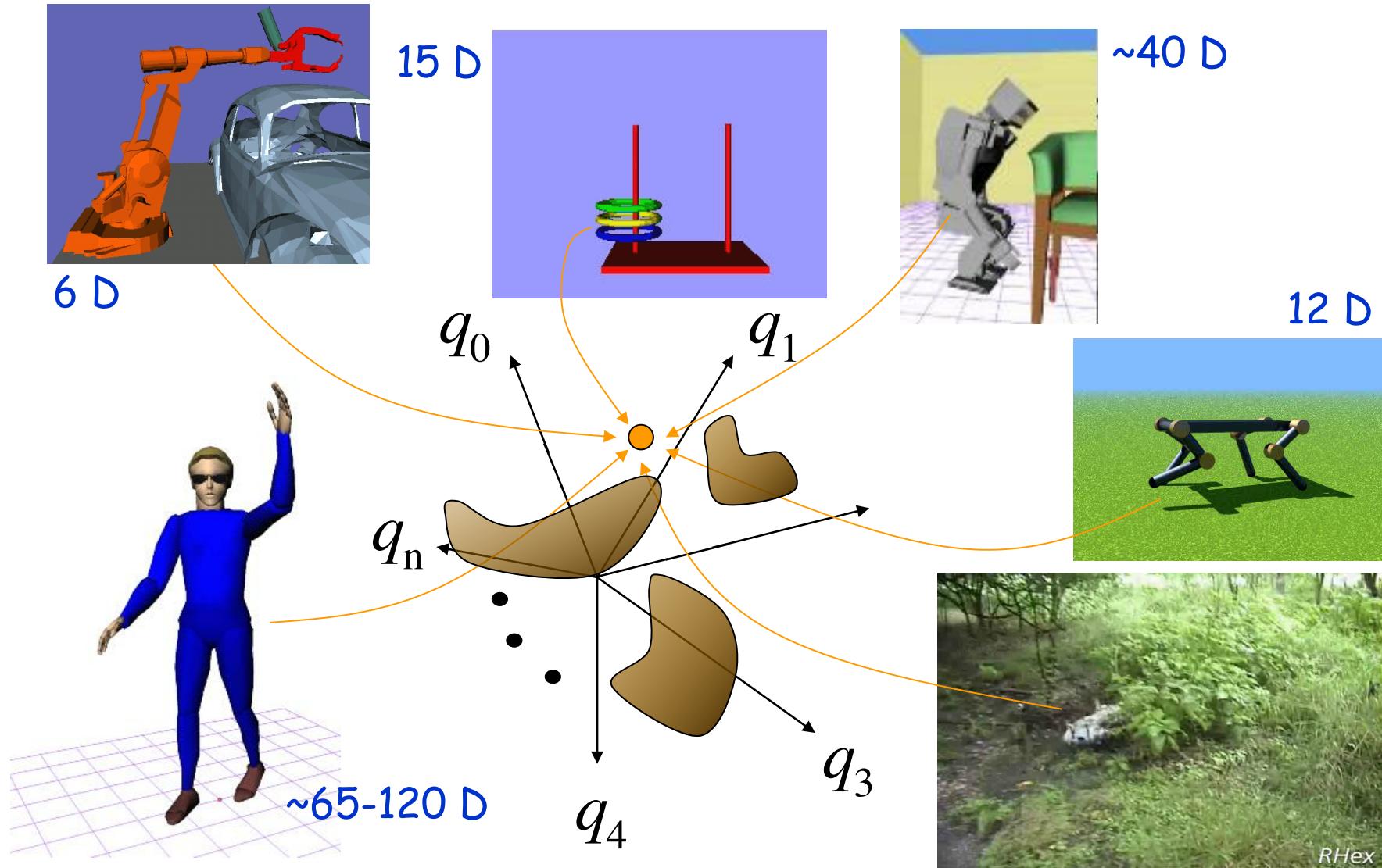
Every robot maps to a point in its configuration space ...



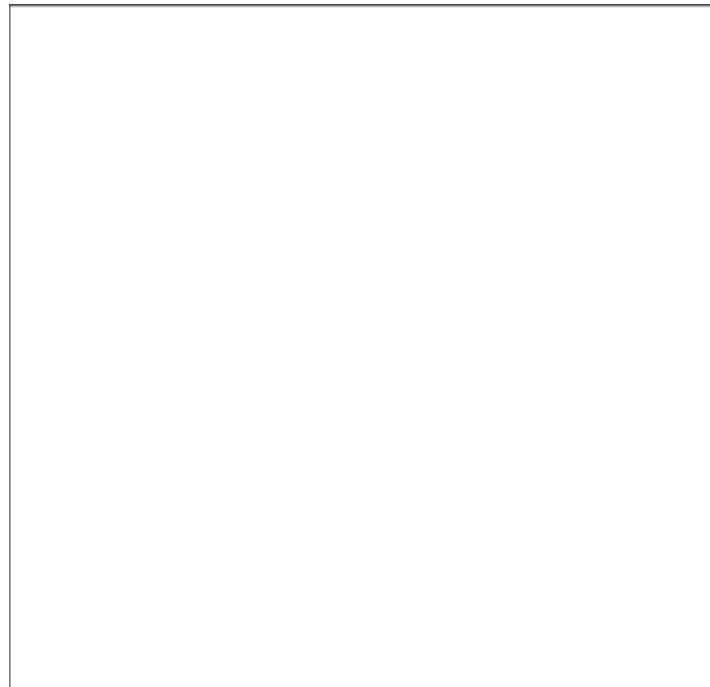
*... and every robot path is a curve
in configuration space*



But how do obstacles (and other constraints) map in configuration space?



RRTs

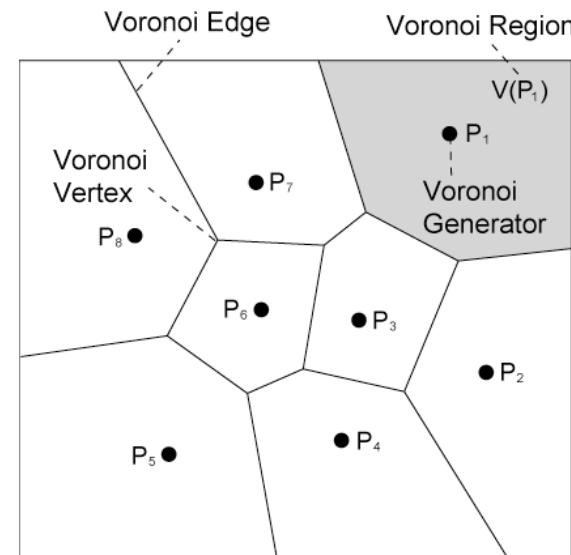
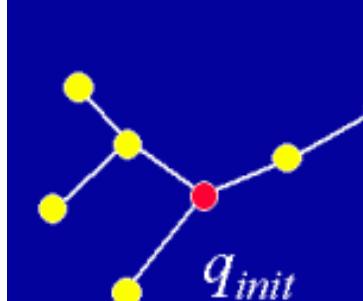


[LaValle, Kuffner, IJRR02]

- RRT is a data structure and algorithm that is designed for efficiently searching non convex high-dimensional spaces.
- RRT can be considered as a Monte-Carlo way of biasing search into largest Voronoi Regions.

Basic construction

Existing RRT is “grown” as follows...



RRT Algorithm

```
BUILD_RRT( $x_{init}$ )
```

- 1 $\mathcal{T}.\text{init}(x_{init})$;
 - 2 **for** $k = 1$ **to** K **do**
 - 3 $x_{rand} \leftarrow \text{RANDOM_STATE}()$;
 - 4 EXTEND(\mathcal{T}, x_{rand});
 - 5 Return \mathcal{T}
-

```
EXTEND( $\mathcal{T}, x$ )
```

- 1 $x_{near} \leftarrow \text{NEAREST_NEIGHBOR}(x, \mathcal{T})$;
 - 2 **if** NEW_STATE($x, x_{near}, x_{new}, u_{new}$) **then**
 - 3 $\mathcal{T}.\text{add_vertex}(x_{new})$;
 - 4 $\mathcal{T}.\text{add_edge}(x_{near}, x_{new}, u_{new})$;
 - 5 **if** $x_{new} = x$ **then**
 - 6 Return *Reached*;
 - 7 **else**
 - 8 Return *Advanced*;
 - 9 Return *Trapped*;
-

Sampling-based Algorithms for Optimal Motion Planning

RRT*-Map Specs

$$\mu(X_{free}) = 0.92$$

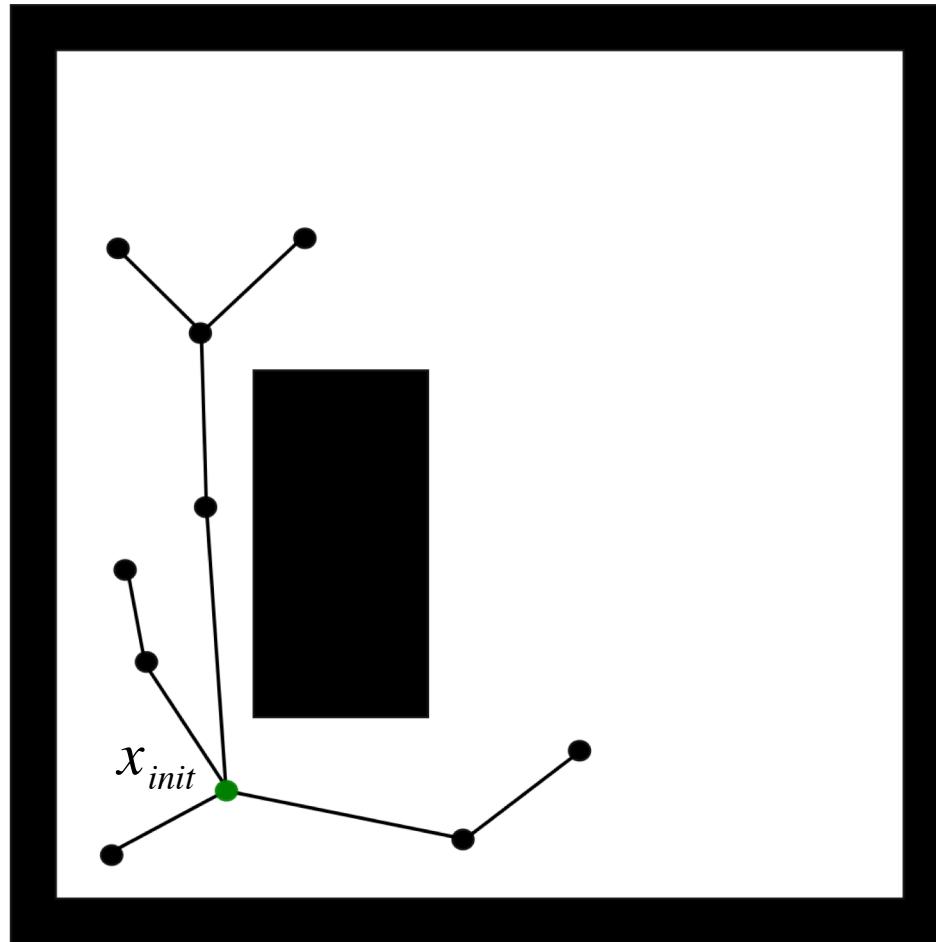
$$d = 2$$

$$\zeta_d = \pi$$

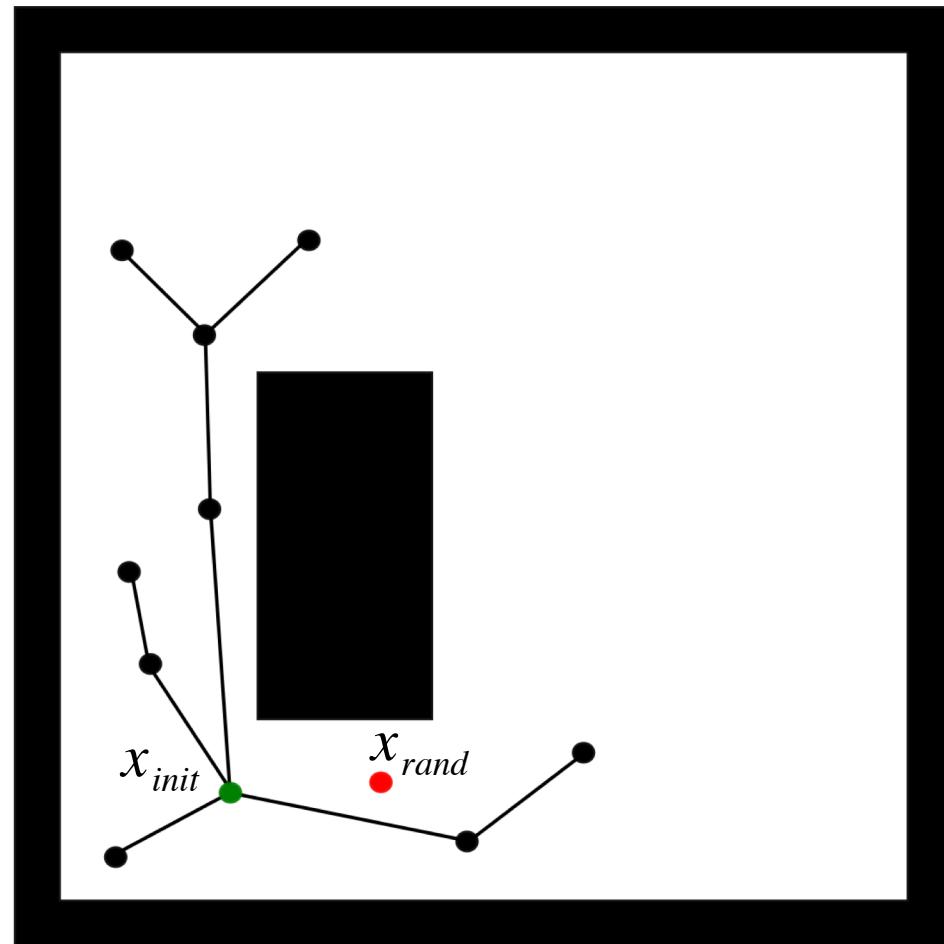
$$\gamma_{RRT} > \left(2(1 + 1/d)\right)^{1/d} \left(\frac{\mu(X_{free})}{\zeta_d}\right)^{1/d} \approx 0.9373$$

$$r_n = \gamma_{RRT} \left(\frac{\log(n)}{n}\right)^{1/d} \quad n = 10 \\ r_{10} > 0.4497$$

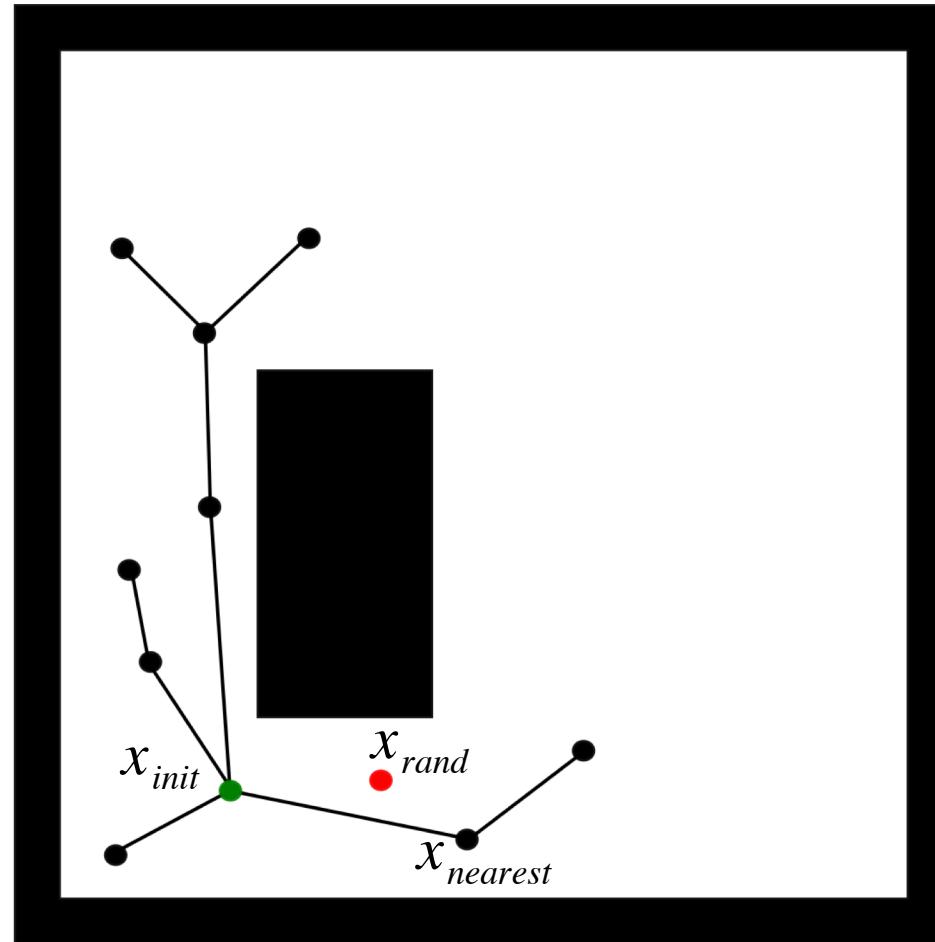
RRT*-Tree after iteration 9



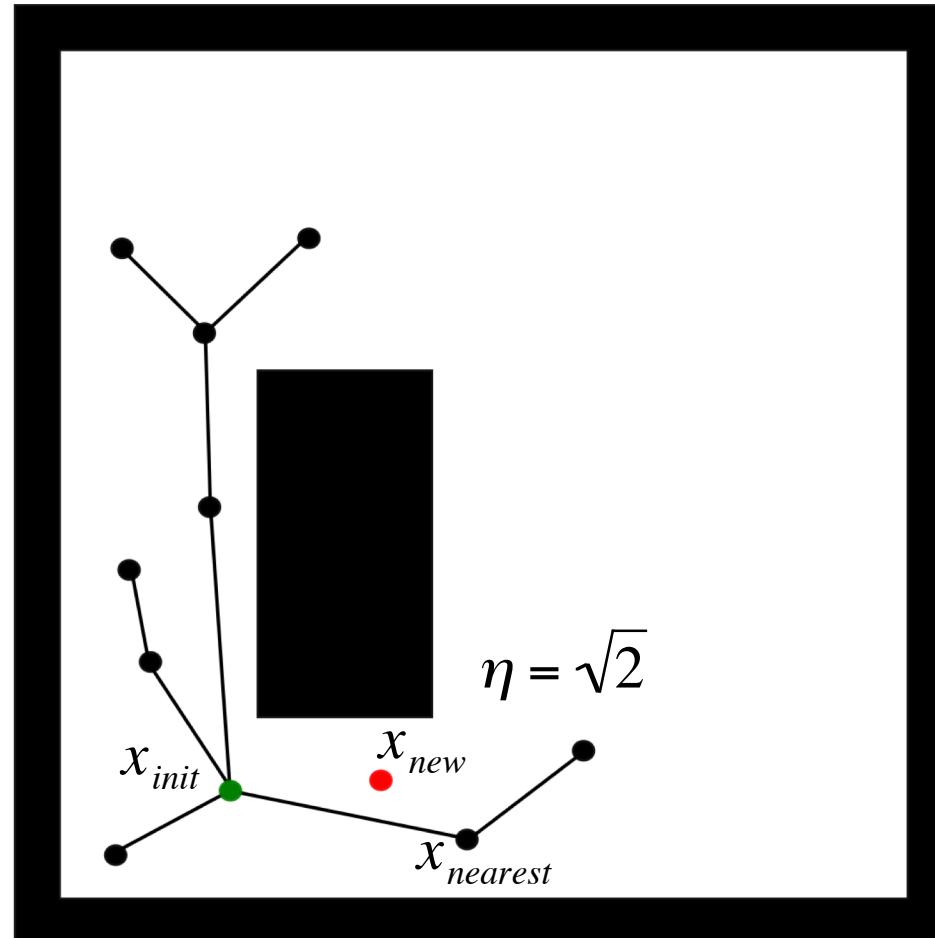
RRT*-New Sample



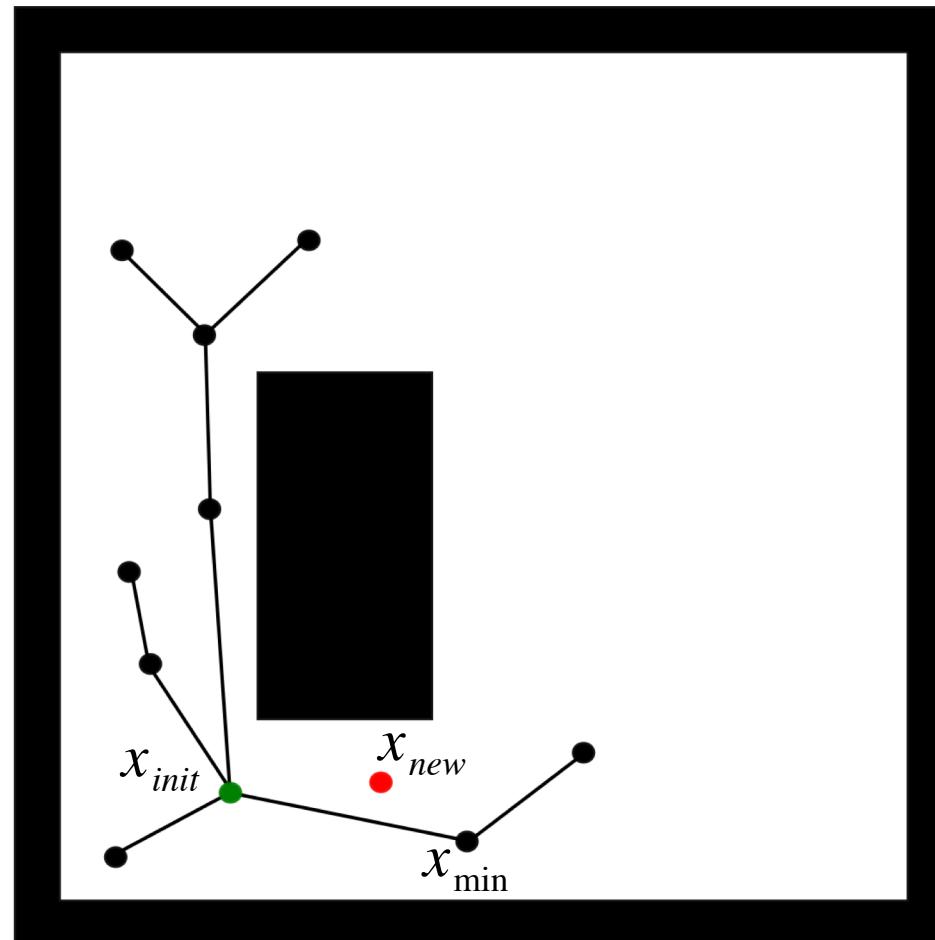
RRT*-New Sample



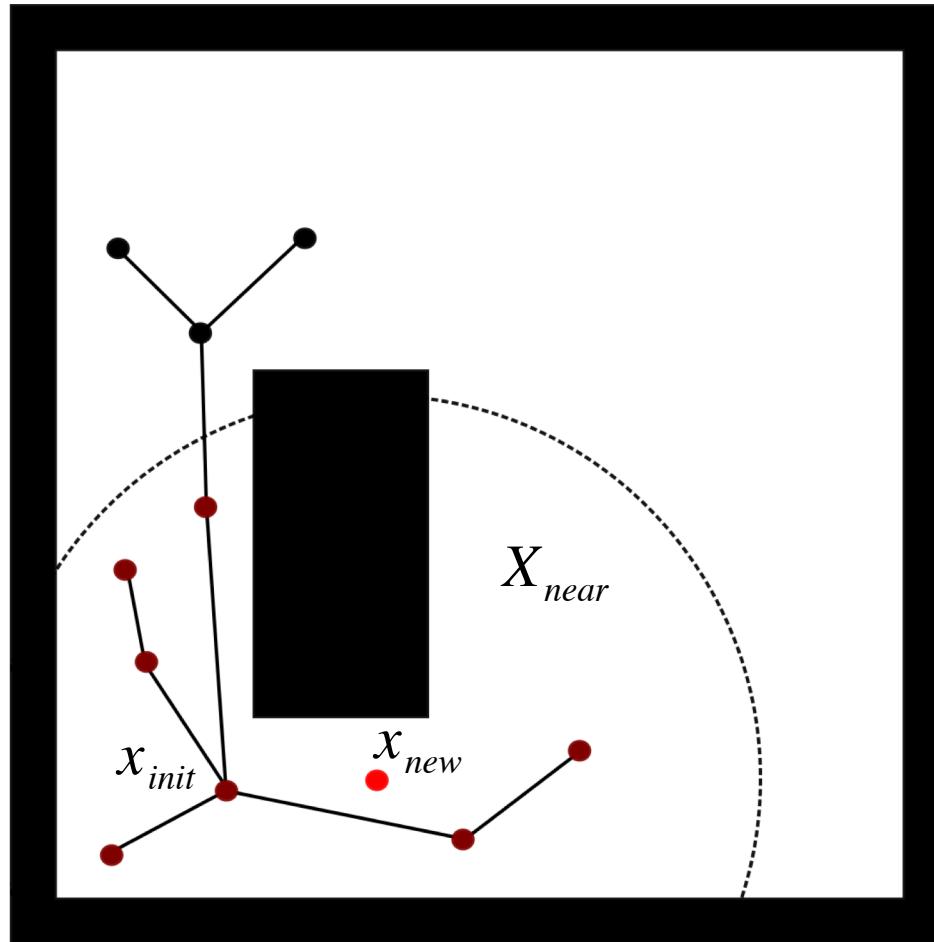
RRT*-New Sample



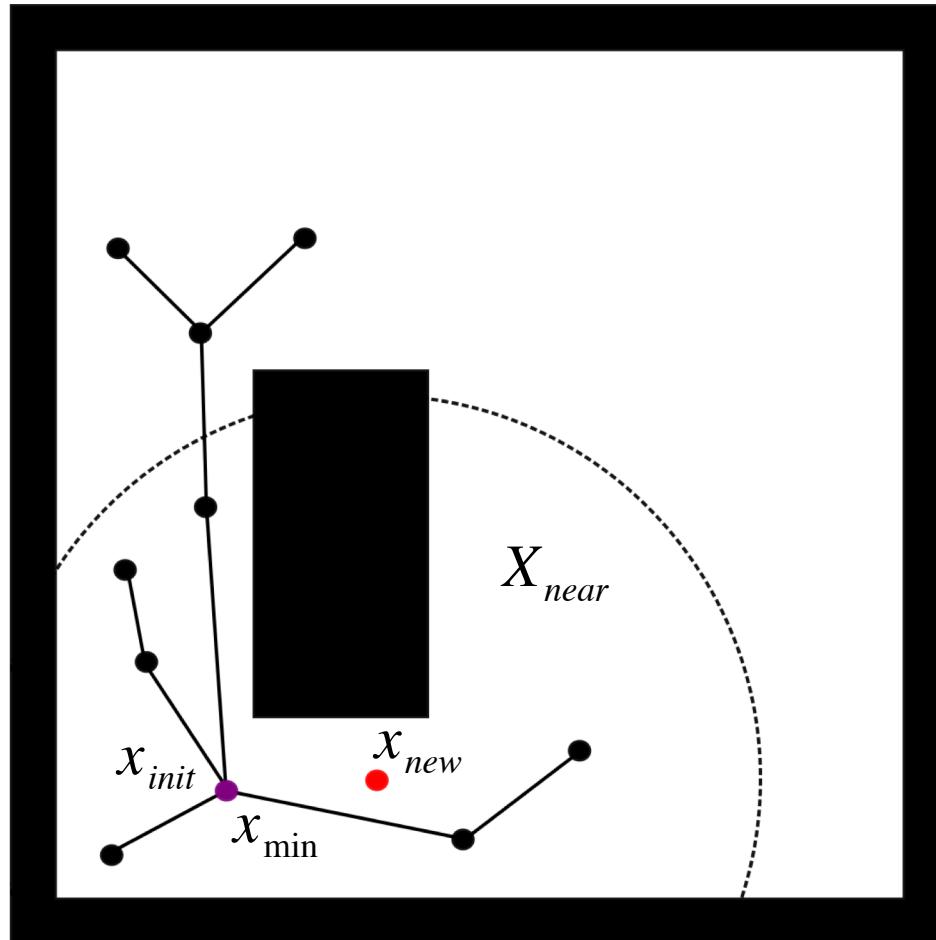
RRT*-New Sample



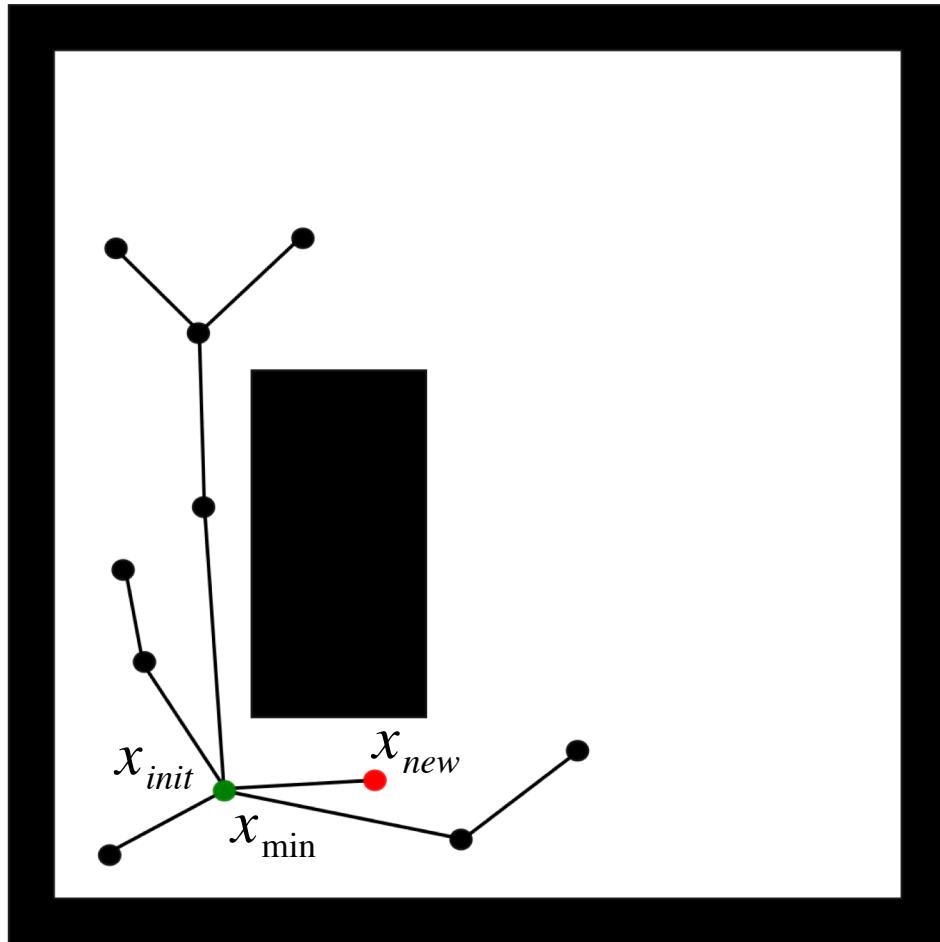
RRT*-Connect Min Cost Path



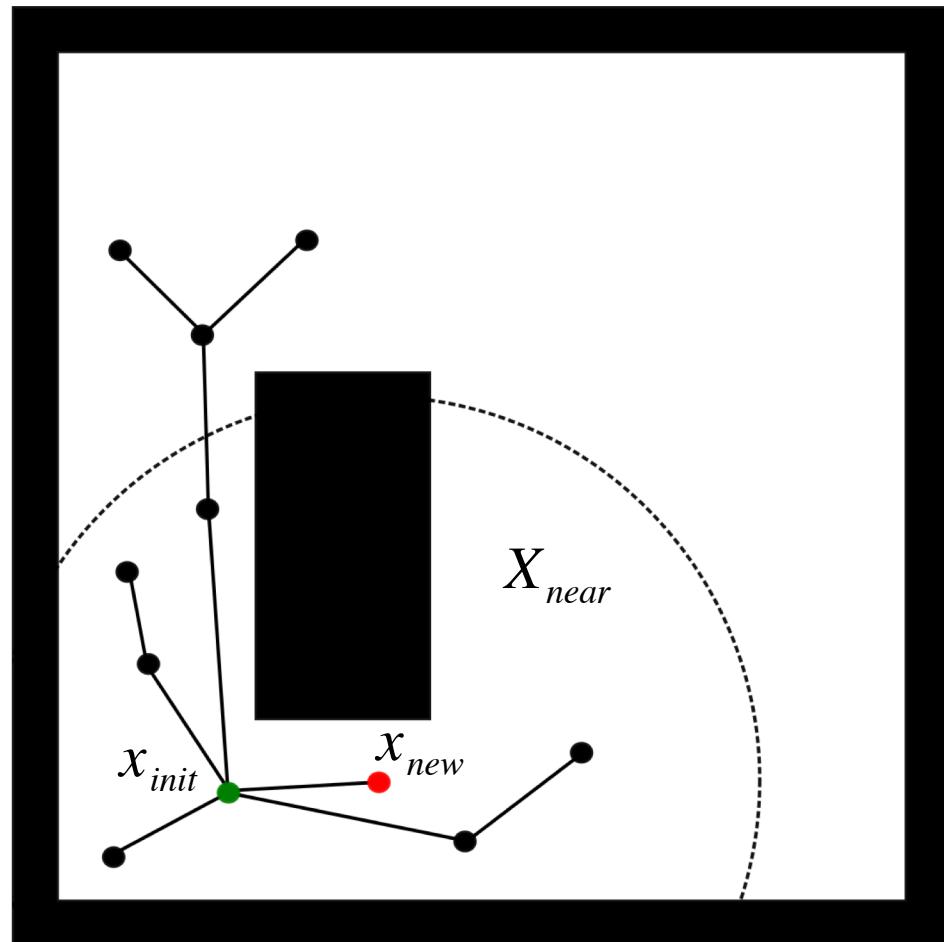
RRT*-Connect Min Cost Path



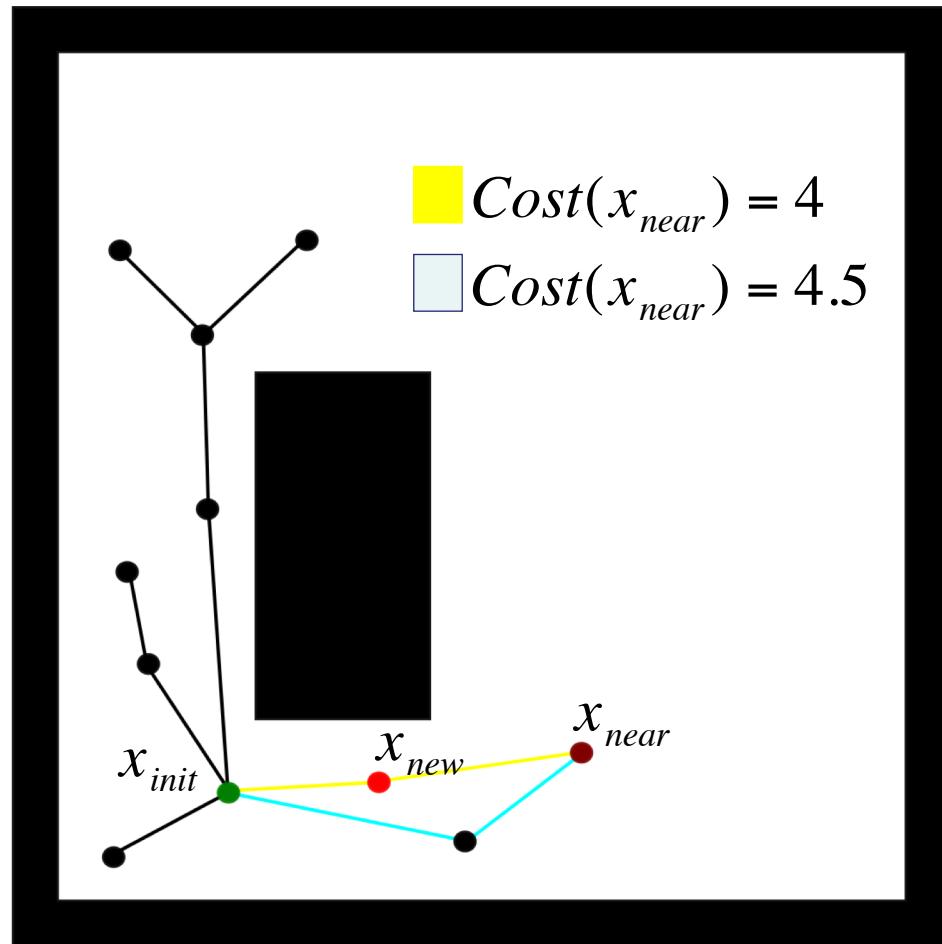
RRT*-Connect Min Cost Path



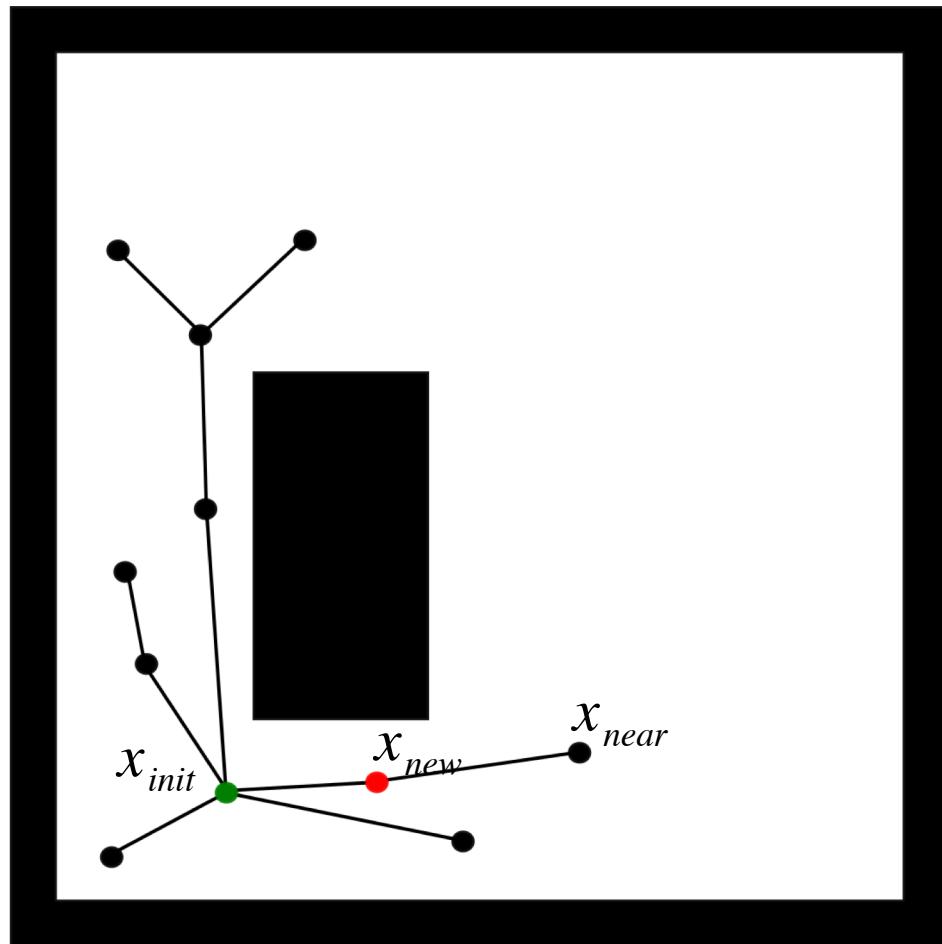
RRT*-Rewire



RRT*-Rewire



RRT*-Rewire



RRT algorithm

Algorithm 3: RRT

```
1  $V \leftarrow \{x_{\text{init}}\}; E \leftarrow \emptyset;$ 
2 for  $i = 1, \dots, n$  do
3    $x_{\text{rand}} \leftarrow \text{SampleFree}_i;$ 
4    $x_{\text{nearest}} \leftarrow \text{Nearest}(G = (V, E), x_{\text{rand}});$ 
5    $x_{\text{new}} \leftarrow \text{Steer}(x_{\text{nearest}}, x_{\text{rand}}) ;$ 
6   if  $\text{ObstacleFree}(x_{\text{nearest}}, x_{\text{new}})$  then
7      $V \leftarrow V \cup \{x_{\text{new}}\}; E \leftarrow E \cup \{(x_{\text{nearest}}, x_{\text{new}})\} ;$ 
8 return  $G = (V, E);$ 
```

RRT* algorithm

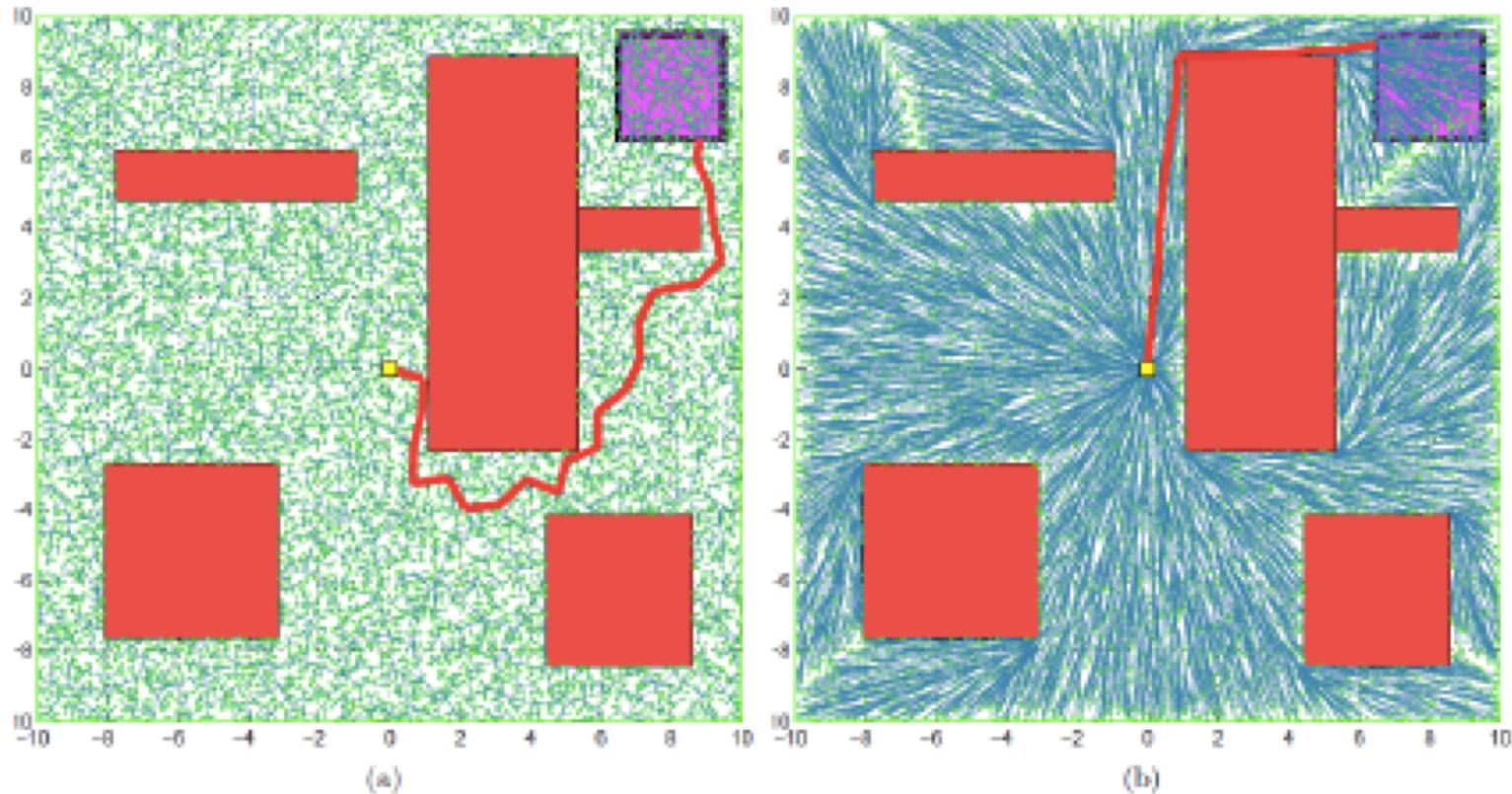
Algorithm 6: RRT*

```

1  $V \leftarrow \{x_{\text{init}}\}; E \leftarrow \emptyset;$ 
2 for  $i = 1, \dots, n$  do
3    $x_{\text{rand}} \leftarrow \text{SampleFree}_i;$ 
4    $x_{\text{nearest}} \leftarrow \text{Nearest}(G = (V, E), x_{\text{rand}});$ 
5    $x_{\text{new}} \leftarrow \text{Steer}(x_{\text{nearest}}, x_{\text{rand}}) ;$ 
6   if  $\text{ObstacleFree}(x_{\text{nearest}}, x_{\text{new}})$  then
7      $X_{\text{near}} \leftarrow \text{Near}(G = (V, E), x_{\text{new}}, \min\{\gamma_{\text{RRT}^*}(\log(\text{card}(V)) / \text{card}(V))^{1/d}, \eta\}) ;$ 
8      $V \leftarrow V \cup \{x_{\text{new}}\};$ 
9      $x_{\text{min}} \leftarrow x_{\text{nearest}}; c_{\text{min}} \leftarrow \text{Cost}(x_{\text{nearest}}) + c(\text{Line}(x_{\text{nearest}}, x_{\text{new}}));$ 
10    foreach  $x_{\text{near}} \in X_{\text{near}}$  do           // Connect along a minimum-cost path
11      if  $\text{CollisionFree}(x_{\text{near}}, x_{\text{new}}) \wedge \text{Cost}(x_{\text{near}}) + c(\text{Line}(x_{\text{near}}, x_{\text{new}})) < c_{\text{min}}$  then
12         $x_{\text{min}} \leftarrow x_{\text{near}}; c_{\text{min}} \leftarrow \text{Cost}(x_{\text{near}}) + c(\text{Line}(x_{\text{near}}, x_{\text{new}}))$ 
13
14     $E \leftarrow E \cup \{(x_{\text{min}}, x_{\text{new}})\};$ 
15    foreach  $x_{\text{near}} \in X_{\text{near}}$  do           // Rewire the tree
16      if  $\text{CollisionFree}(x_{\text{new}}, x_{\text{near}}) \wedge \text{Cost}(x_{\text{new}}) + c(\text{Line}(x_{\text{new}}, x_{\text{near}})) < \text{Cost}(x_{\text{near}})$ 
17      then  $x_{\text{parent}} \leftarrow \text{Parent}(x_{\text{near}});$ 
18       $E \leftarrow (E \setminus \{(x_{\text{parent}}, x_{\text{near}})\}) \cup \{(x_{\text{new}}, x_{\text{near}})\}$ 
19
20 return  $G = (V, E);$ 

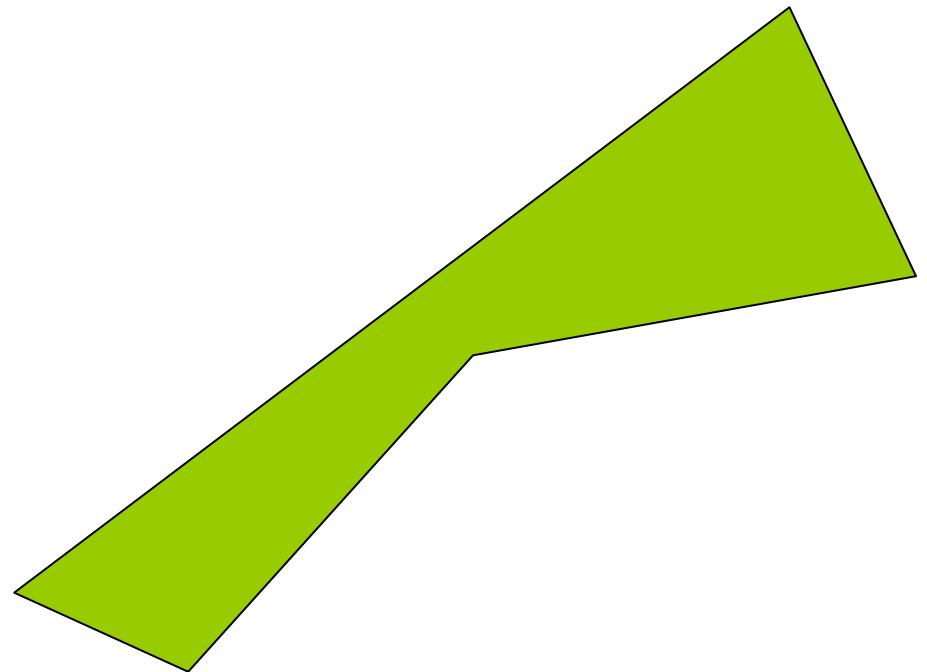
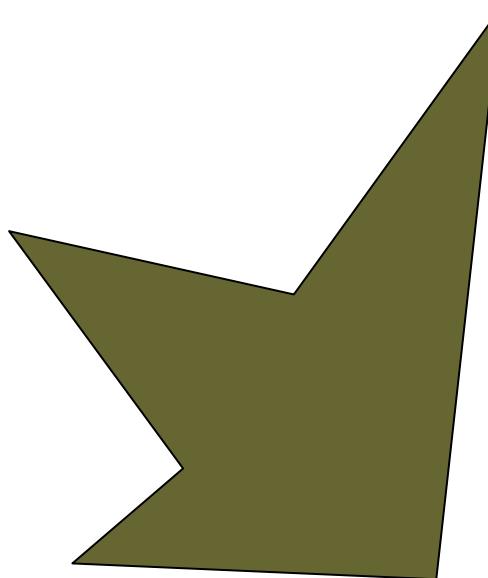
```

Results of RRT vs RRT*



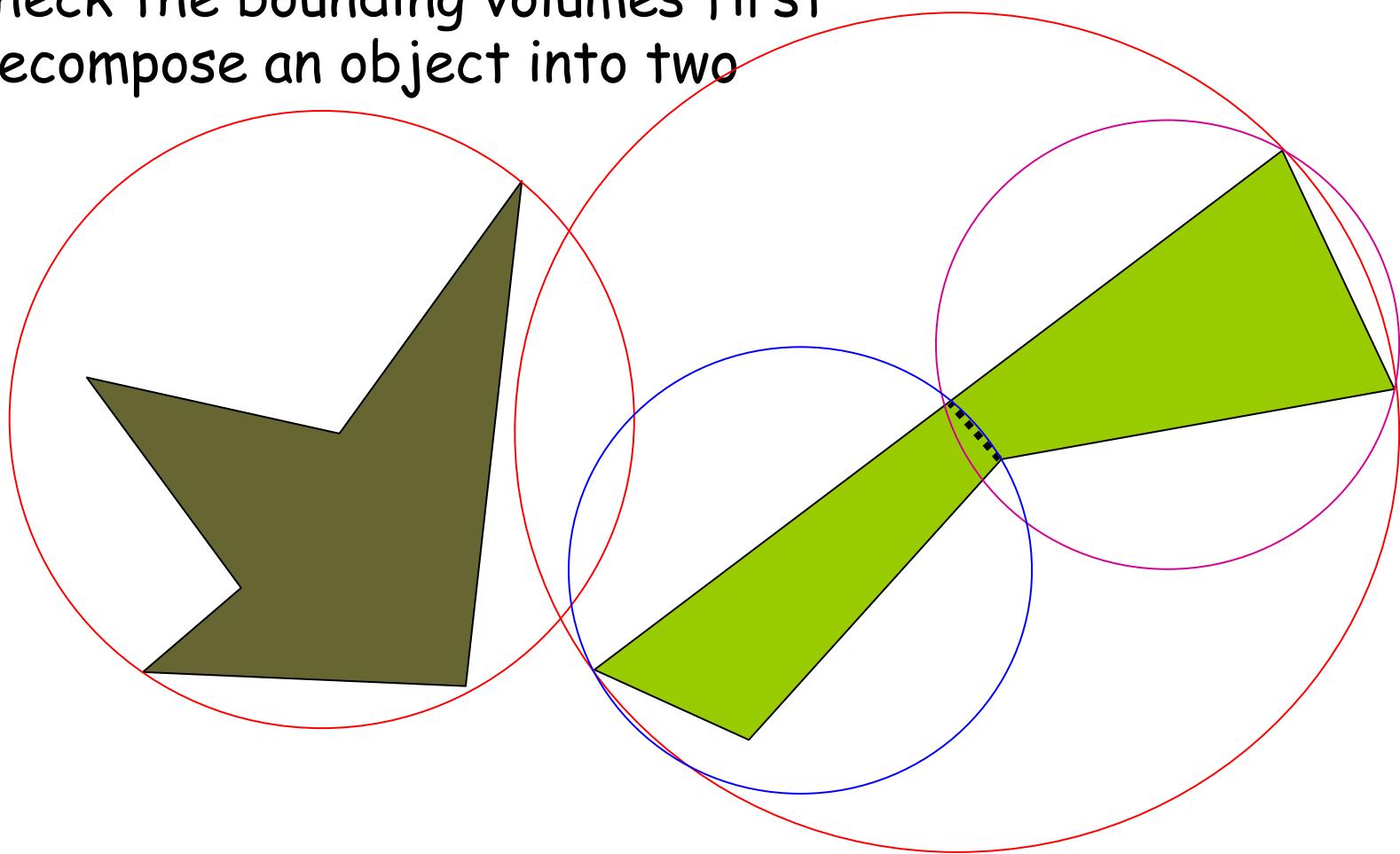
Collision Checking

- Check if objects overlap



Hierarchical Collision Checking

- Enclose objects into bounding volumes (spheres or boxes)
- Check the bounding volumes first
- Decompose an object into two



BVH of a 3D Triangulated Cat



Bibliography

- Robot motion planning, Kluwer,
Jean-Claude Latombe, 1991.
- Robot motion planning and control, Springer,
Jean-Paul Laumond, 1998.
- Planning algorithms, Cambridge, Steven M.
LaValle, 2006.
- Sampling-based algorithms for optimal motion
planning, Sertac Karaman and Emilio Frazzoli,
Int. J. Robotics Research, 2011.