

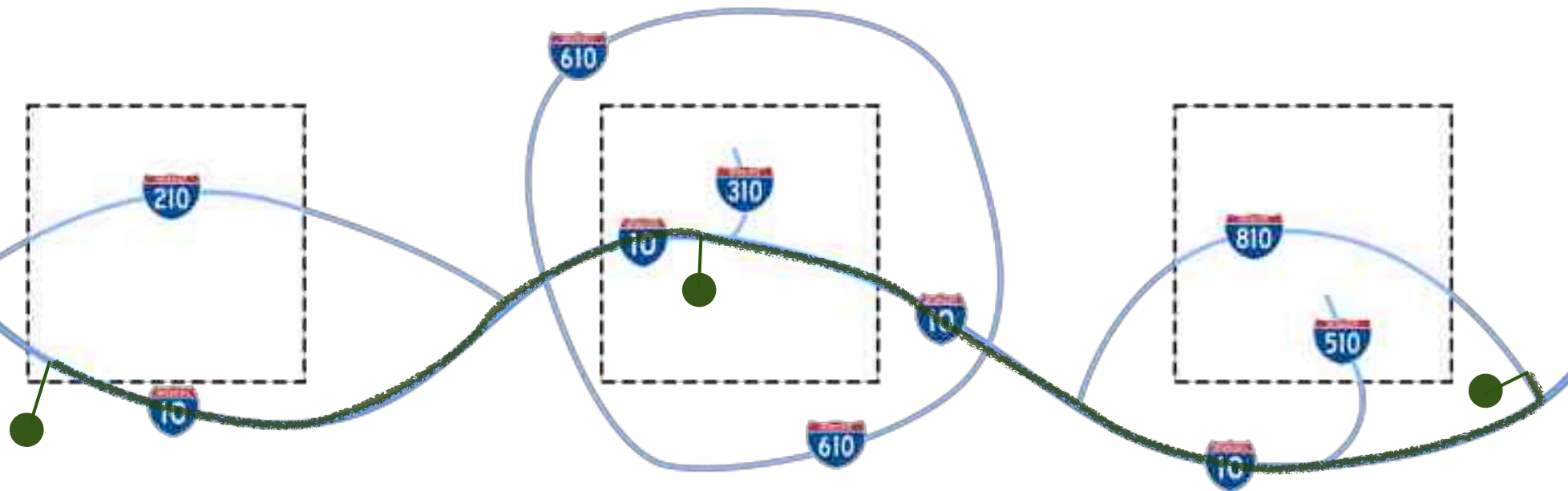
Topic 8

Planning with Roadmaps

CITY A

CITY B

CITY C

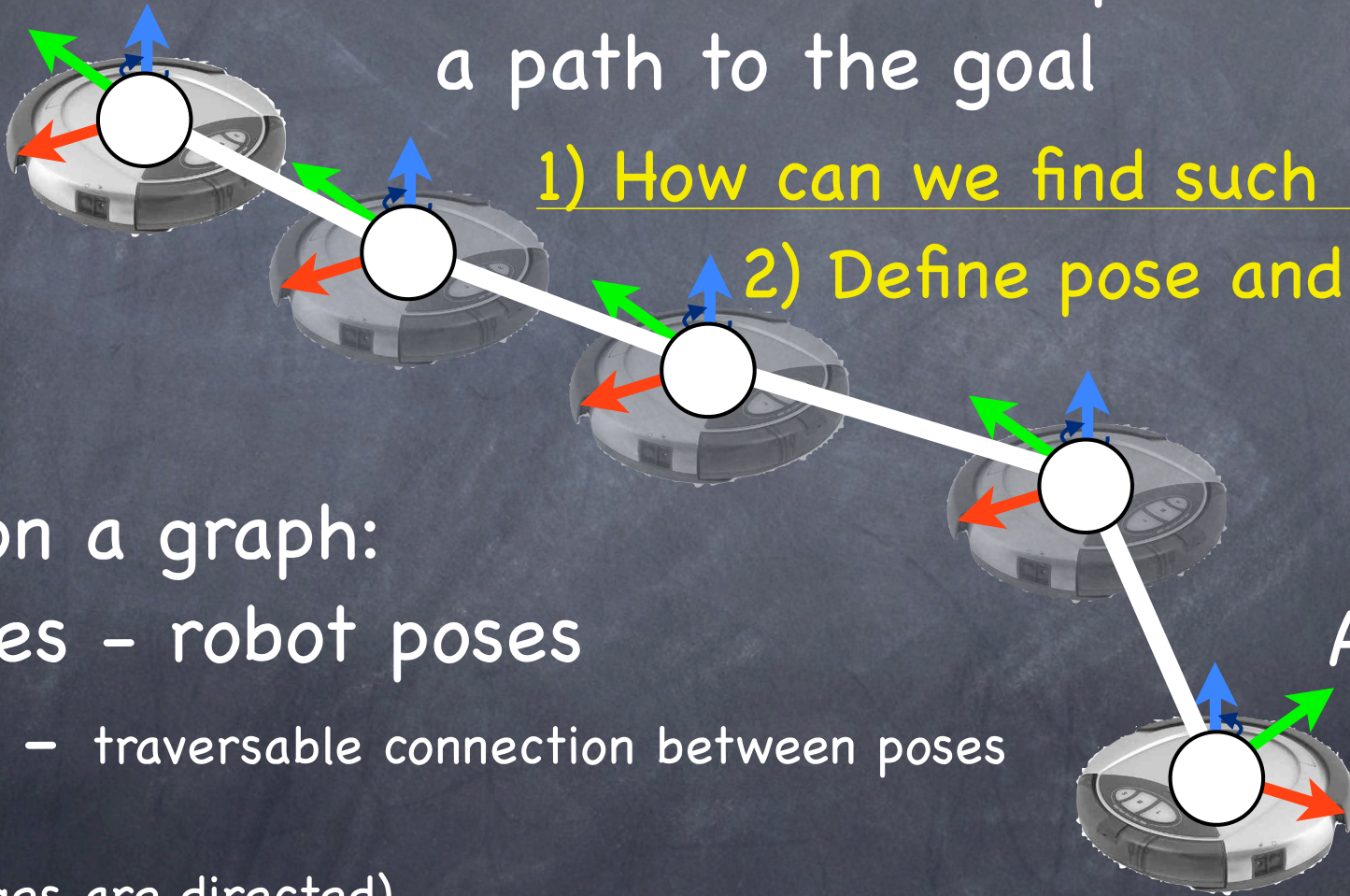


Building a path to goal

Path Planning

B: Goal

Find intermediate poses forming a path to the goal



1) How can we find such paths?

2) Define pose and controls?

Path on a graph:

vertices - robot poses

edges - traversable connection between poses

(note edges are directed)

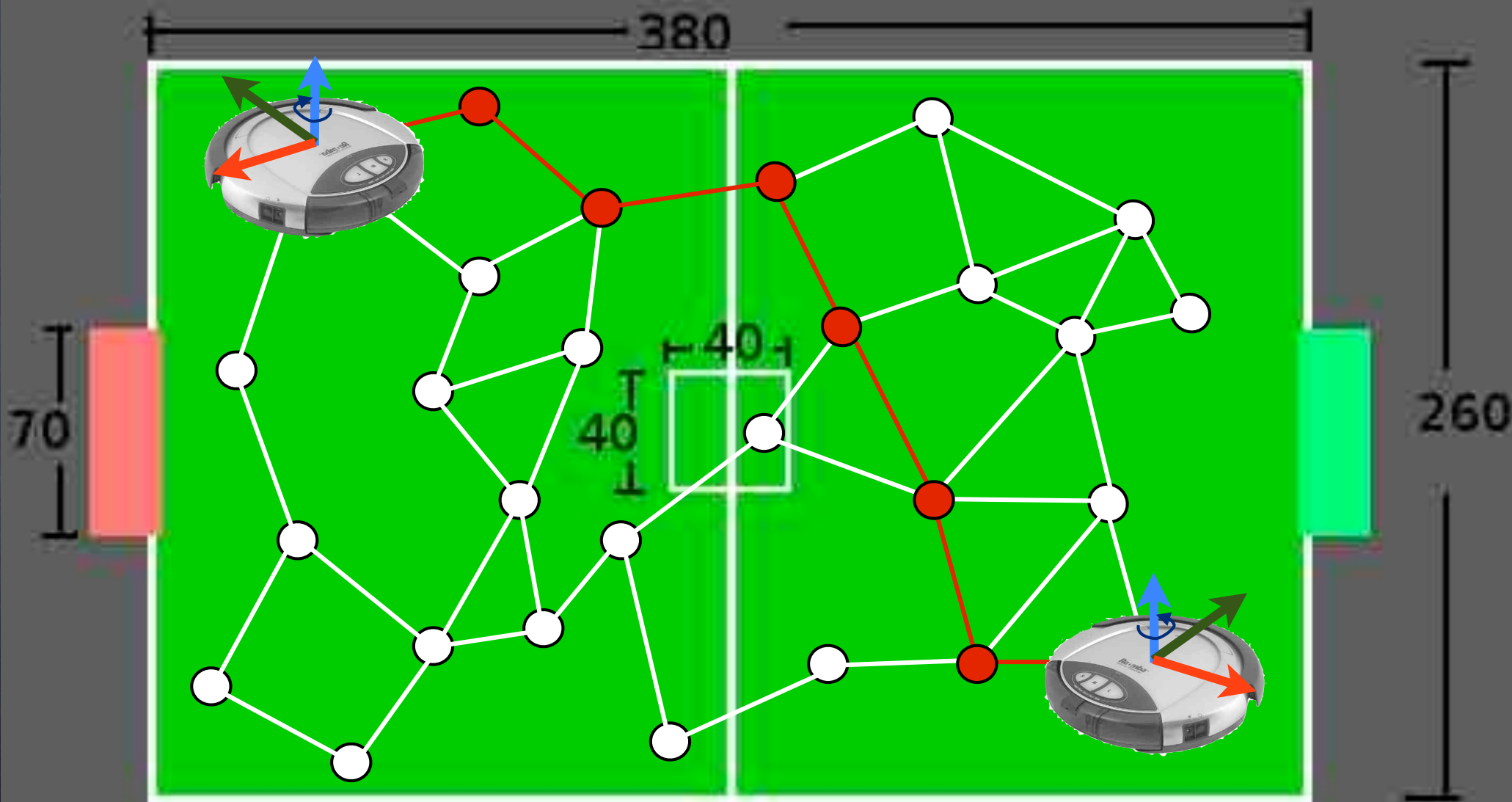
A: Start

Approaches to path planning

- Search (fixed graph)
 - DFS, BFS, Dijkstra, A*
- Search (explore graph):
 - Probabilistic Road Maps
 - Rapidly-exploring Random Trees
- Optimization (local search):
 - Potential fields, gradient descent

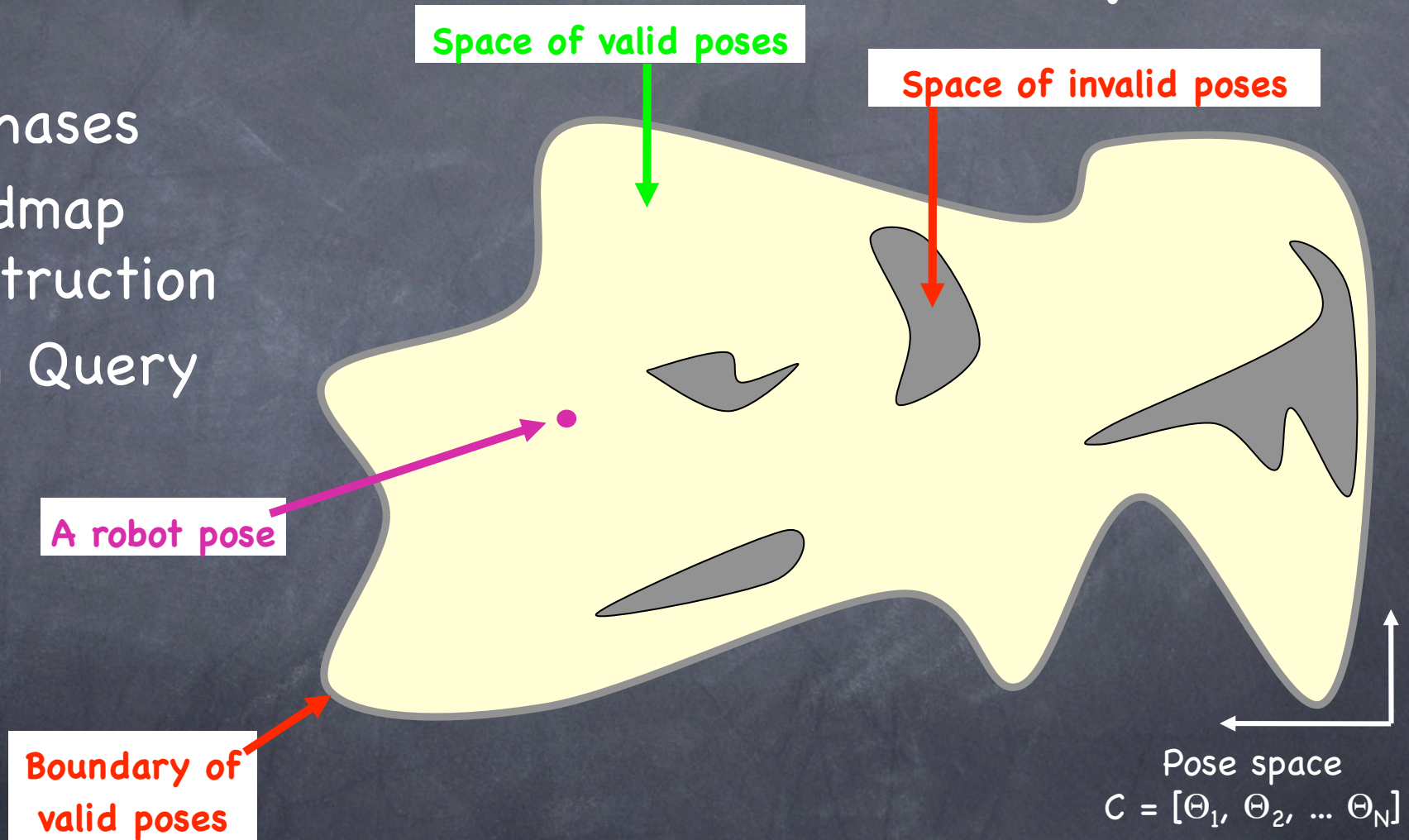
PRMs/ RRT

Explore poses and connectivity;
Find shortest path in built graph



Probabilistic road maps

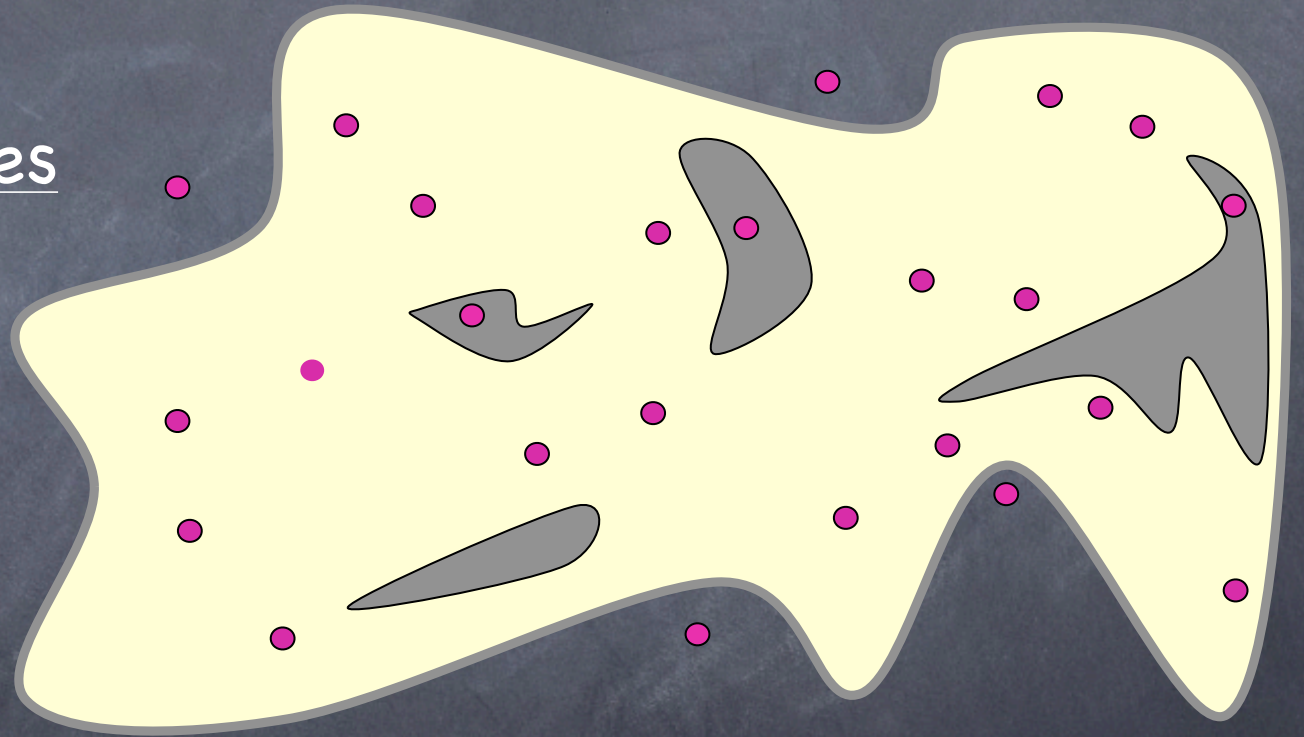
- Two phases
 - Roadmap construction
 - Path Query



[Kavraki, Svetska, Latombe, Overmars, 95]

PRM: construction phase

- Select sample poses at random
- Eliminate invalid poses
- Connect neighboring poses



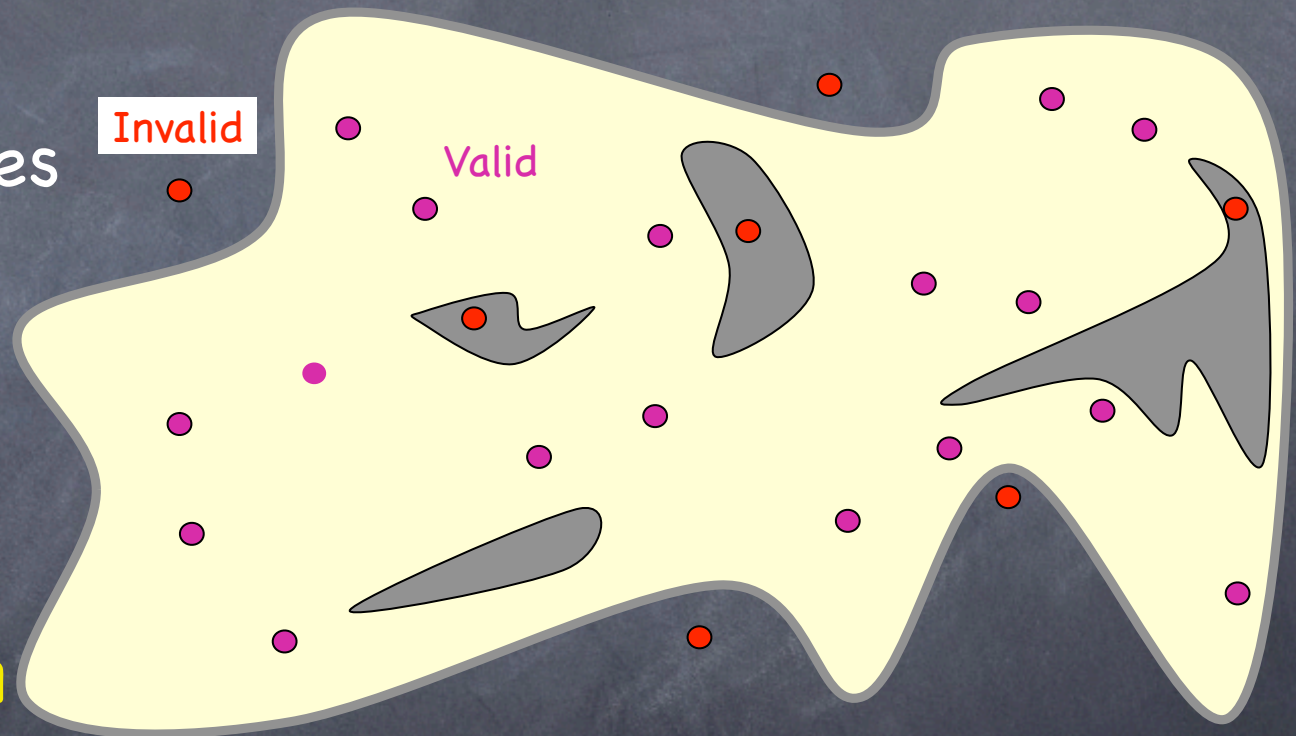
PRM: construction phase

- Select sample poses at random

- Eliminate invalid poses

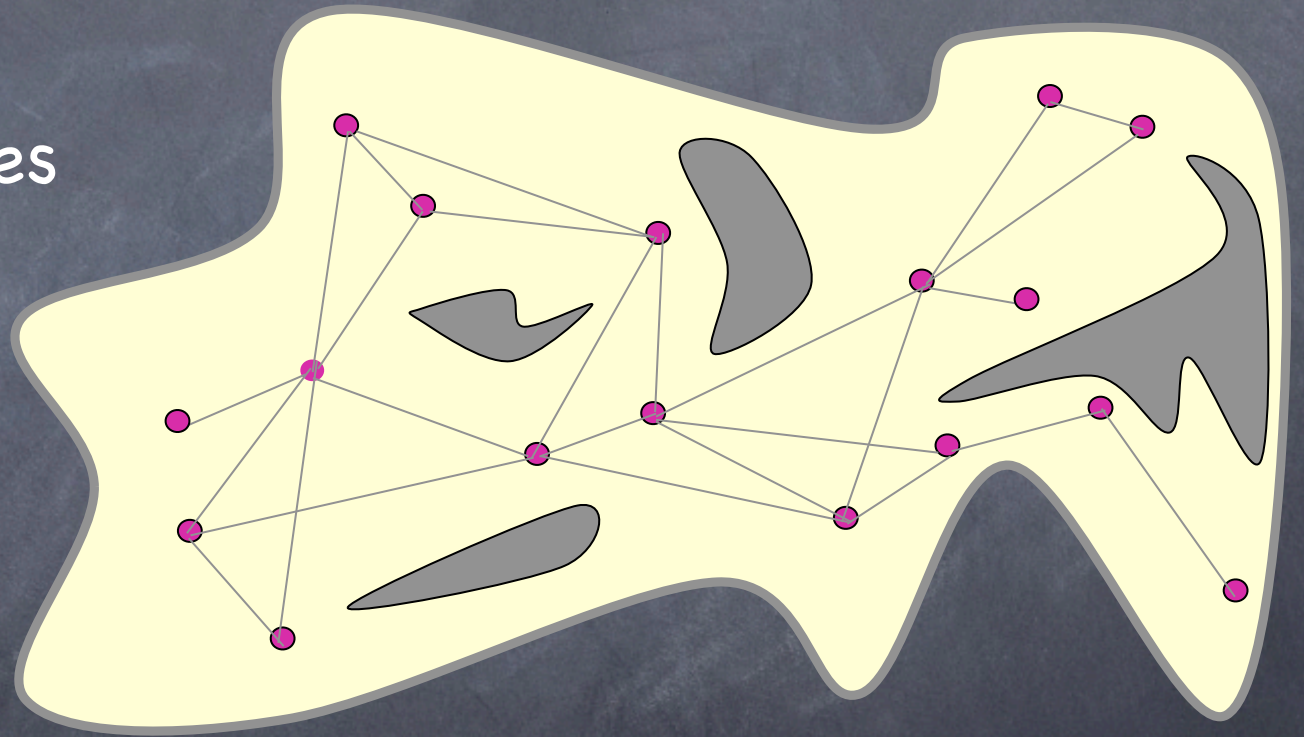
Collision detection

- Connect neighboring poses



PRM: construction phase

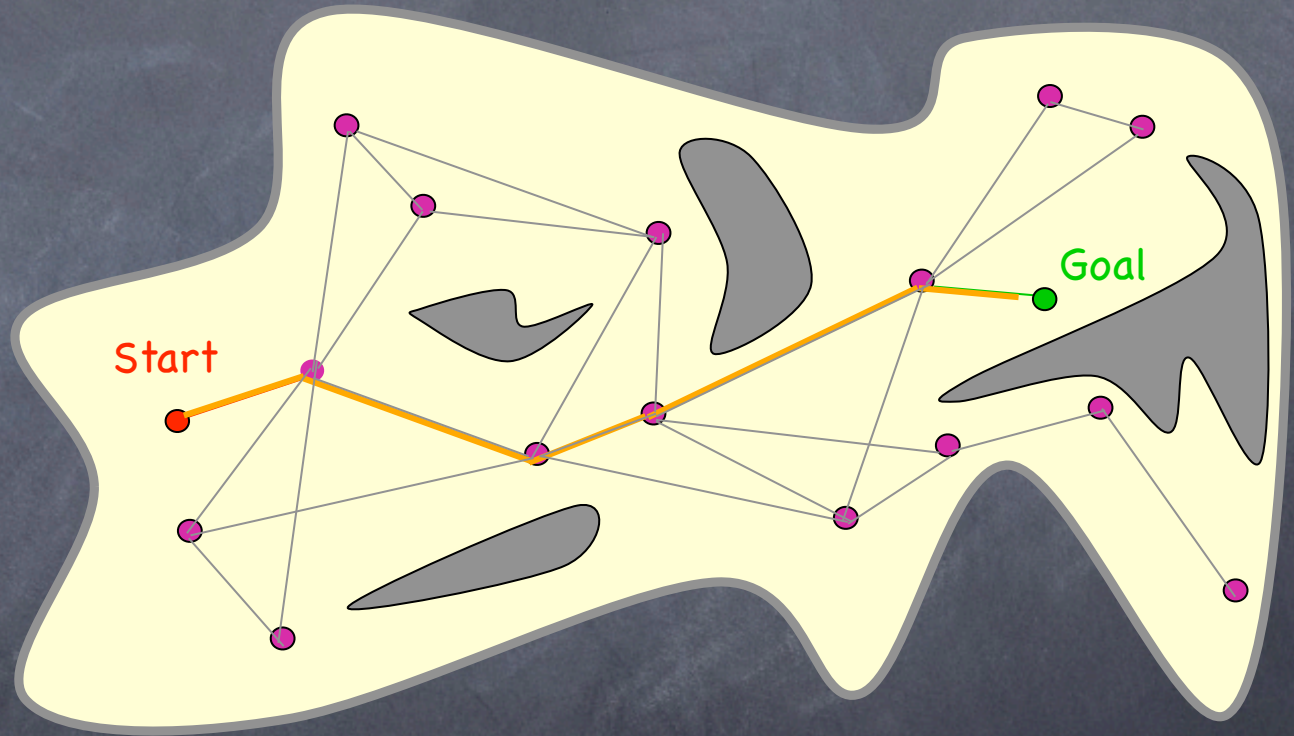
- Select sample poses at random
- Eliminate invalid poses
- Connect neighboring poses



Threshold neighborhood radius or population

PRM: query phase

- Given constructed roadmap
- Find path in roadmap between two poses
- Search on an undirected graph



Rapidly-exploring Random Trees

Explore incrementally along
straight line path to goal

```
BUILD_RRT( $q_{init}$ )  
1  $T$ .init( $q_{init}$ );  
2 for  $k = 1$  to  $K$  do  
3    $q_{rand} \leftarrow$  RANDOM_CONFIG();  
4   EXTEND( $T, q_{rand}$ );  
5 Return  $T$ 
```

```
EXTEND( $T, q$ )  
1  $q_{near} \leftarrow$  NEAREST_NEIGHBOR( $q, T$ );  
2 if NEW_CONFIG( $q, q_{near}, q_{new}$ ) then  
3    $T$ .add_vertex( $q_{new}$ );  
4    $T$ .add_edge( $q_{near}, q_{new}$ );  
5   if  $q_{new} = q$  then  
6     Return Reached;  
7   else  
8     Return Advanced;  
9 Return Trapped;
```

Figure 2: The basic RRT construction algorithm.

[Kuffner, LaValle 2000]

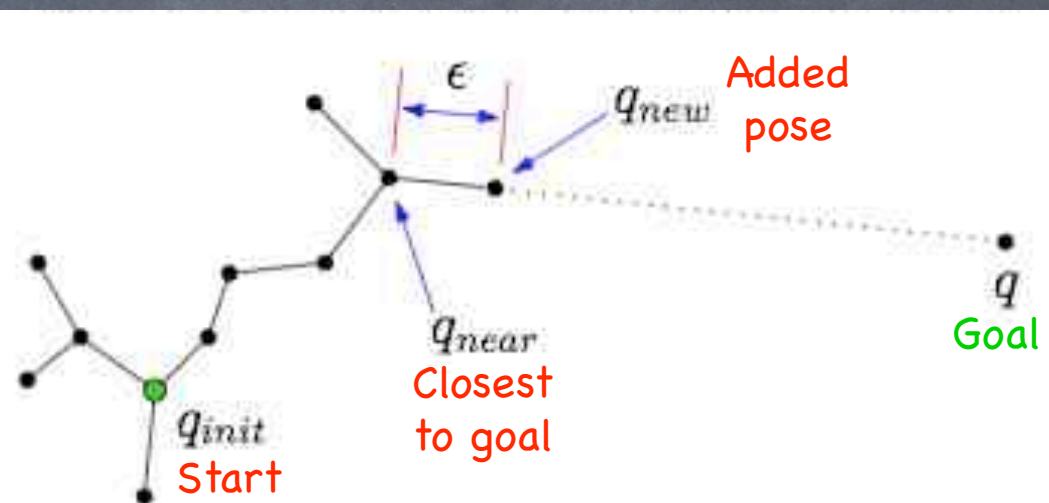


Figure 3: The EXTEND operation.

Single vs. multiple query

Kuffner's RRT Animations

Existing RRT is "grown" as follows...

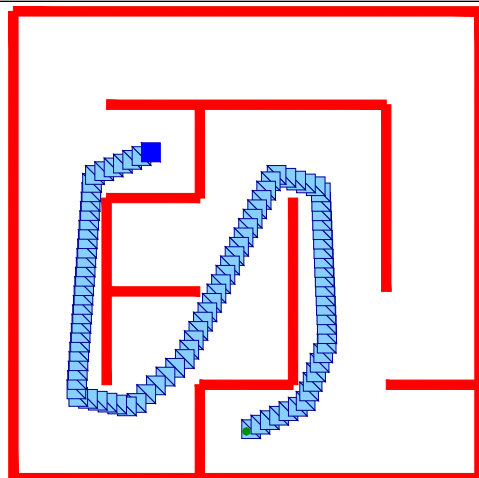
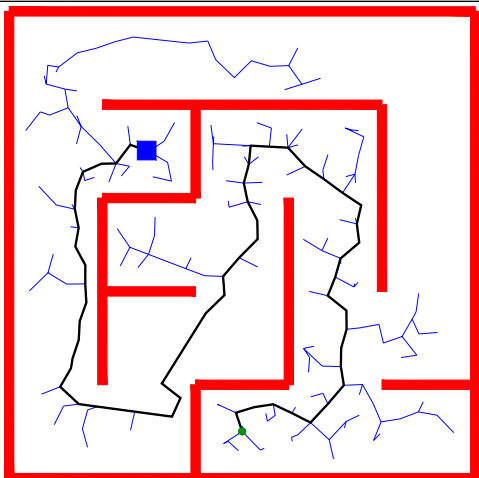


Extending roadmap

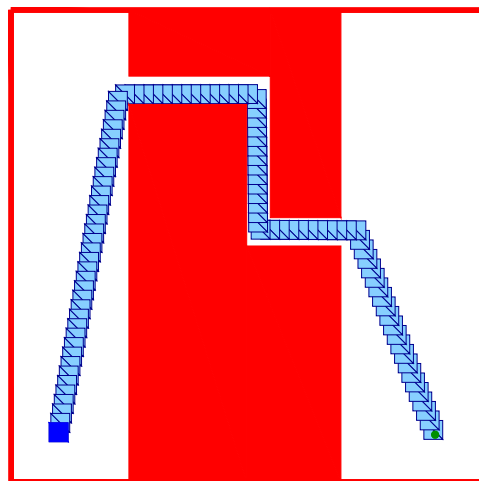
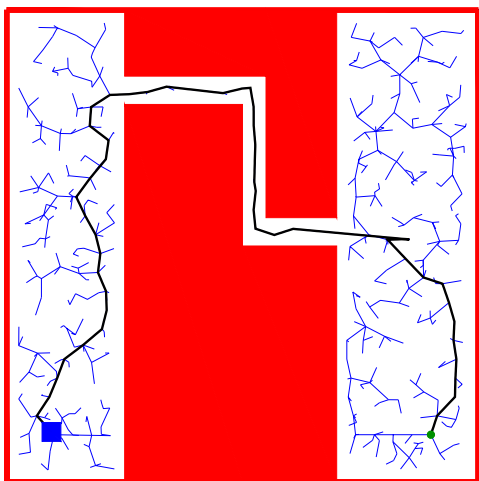
Bidirectional exploration

A single RRT-Connect iteration...

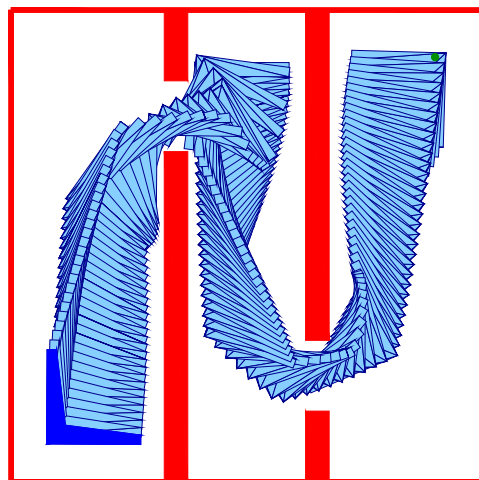
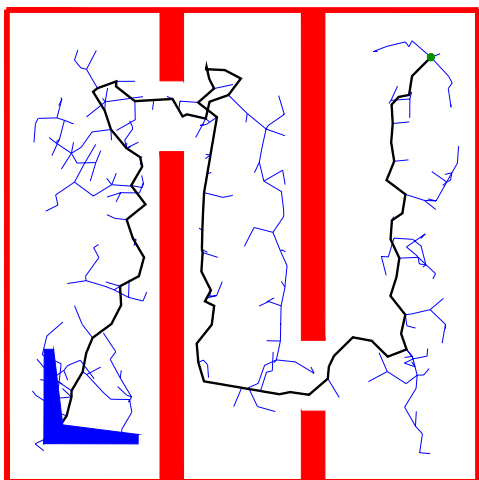




2 DOF maze



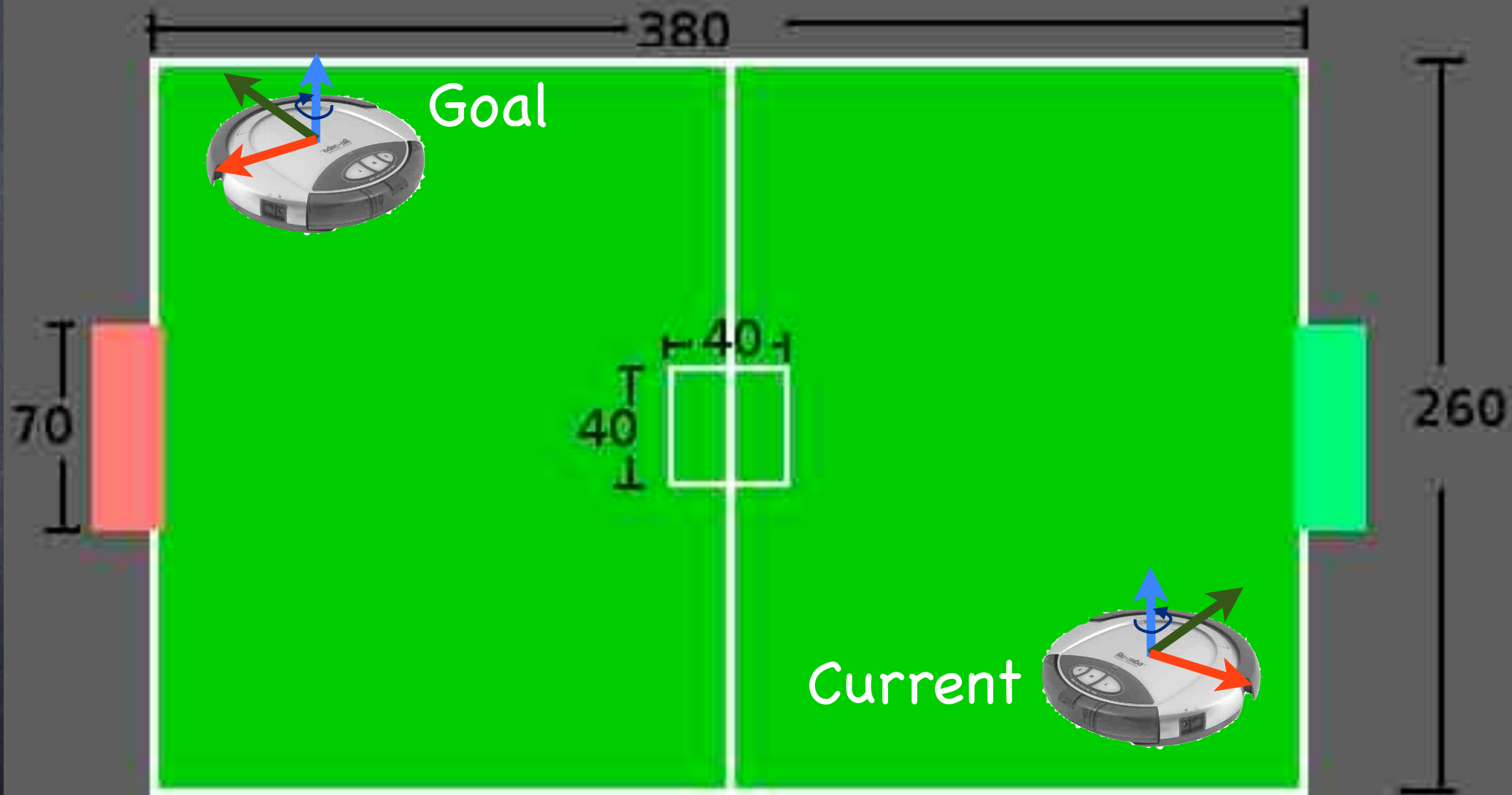
2 DOF single passway



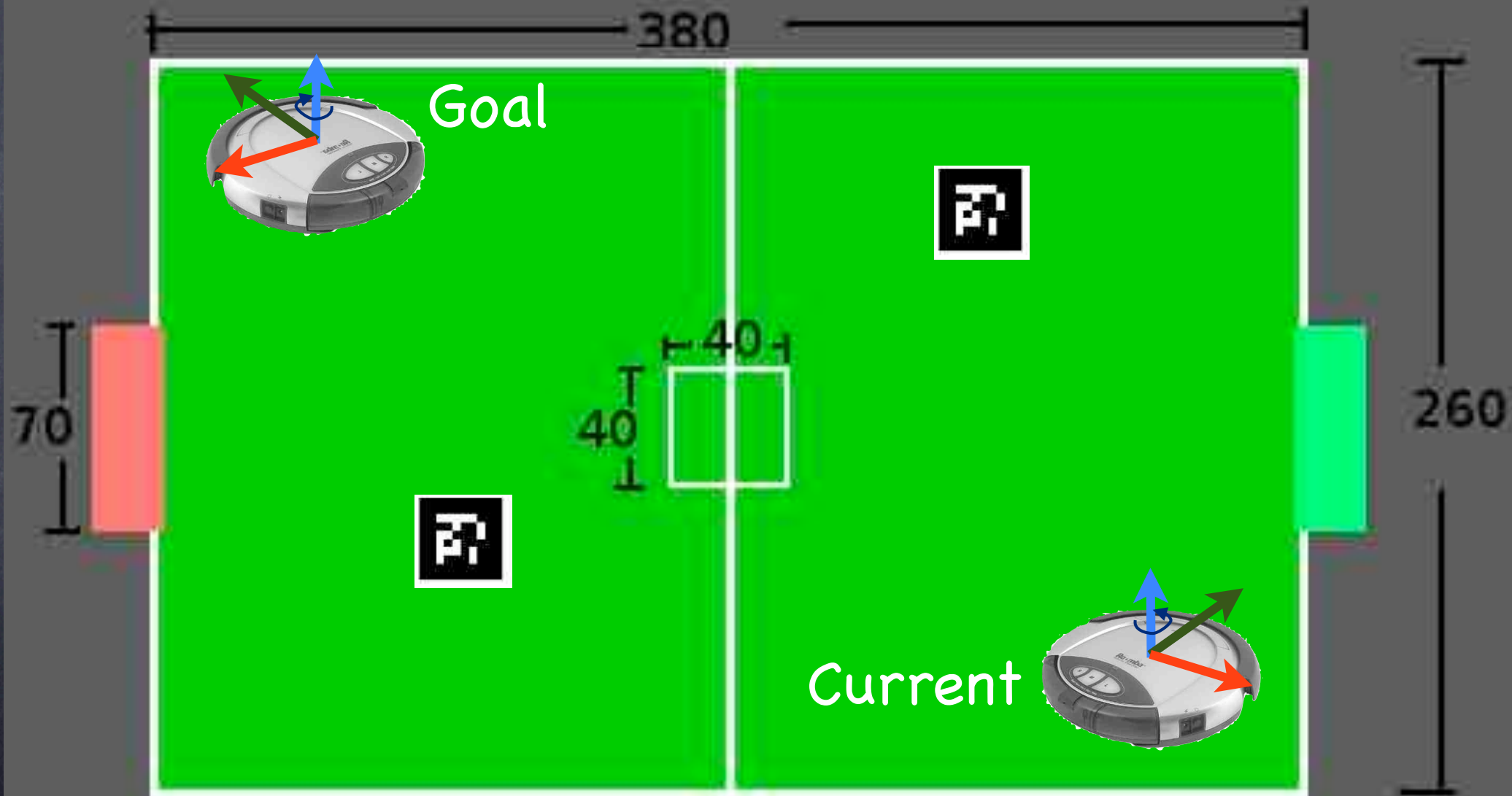
3 DOF single passway

RRT Examples

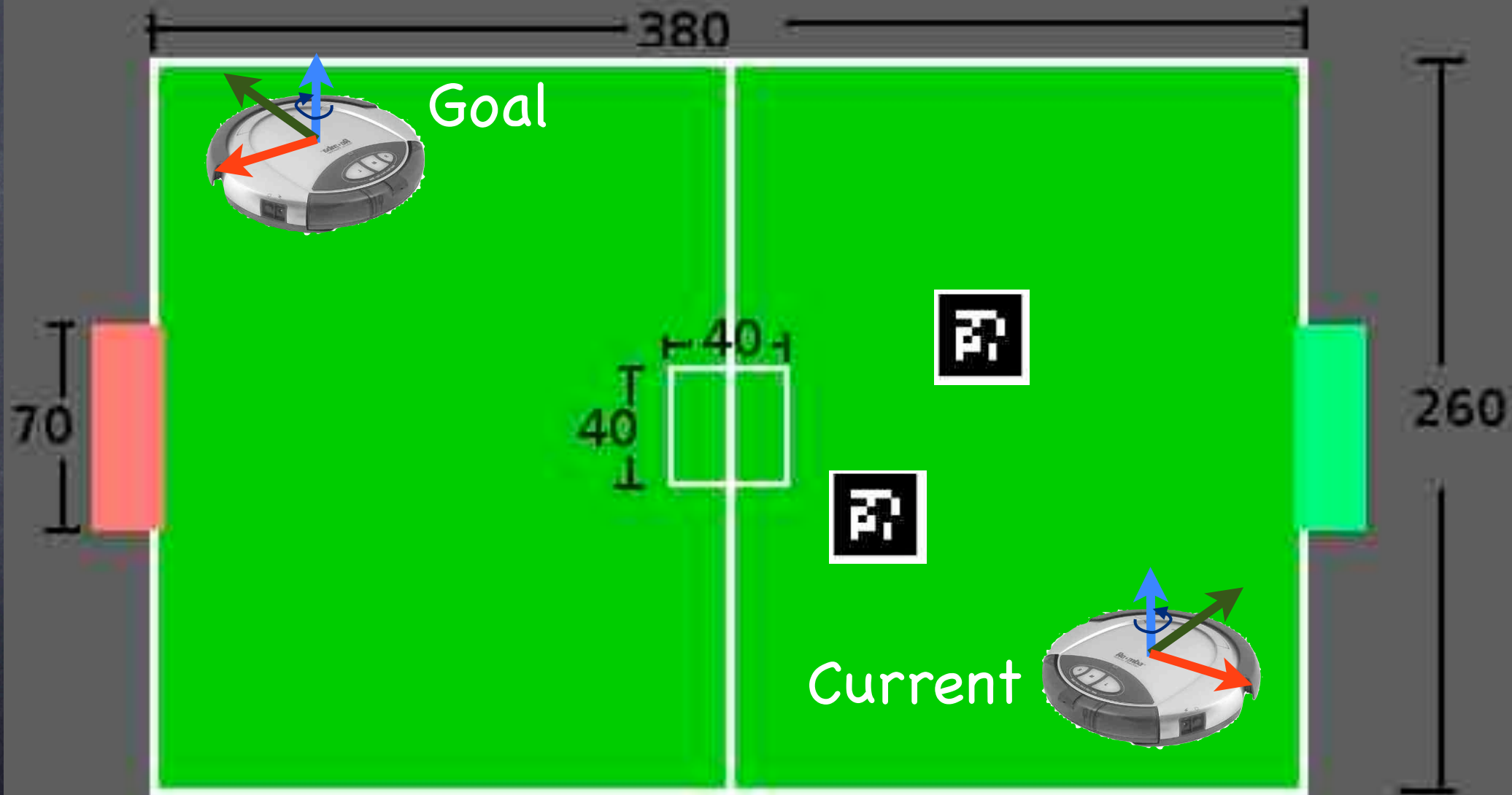
describe performance for this case



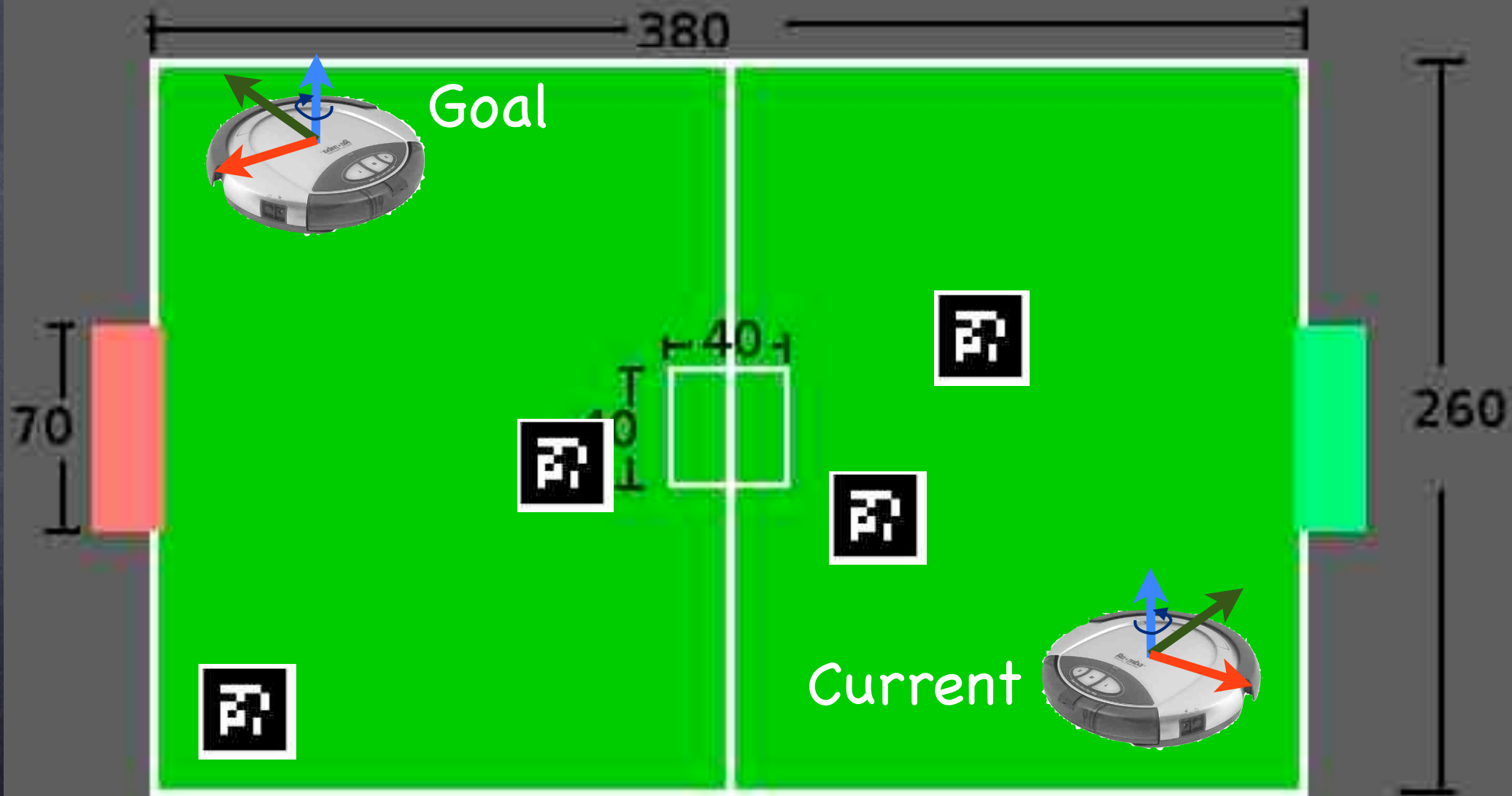
describe performance for this case



describe performance for this case



describe performance for this case



describe performance for this case

