Intelligent Robotics

Navigation

Luís Paulo Reis, Armando Sousa, Nuno Lau

lpreis@fe.up.pt

Director of LIACC – Artificial Intelligence and Computer Science Lab.

Associate Professor at Faculty of Engineering, University of Porto, Portugal

President of APPIA – Portuguese Association for Artificial Intelligence

Background

- **Localization** – Where am I?
- **Mapping** – What are my (dynamic?) surroundings?
- **Navigation** – How do I get where I want to go?
- **SLAM –** Simultaneous Localization and Mapping

The Representation Problem

- Representation is the form in which information is stored or encoded in the robot (Mataric)
- Representation is more than memory
- It has a significant impact on robot control

What can the robot represent

- **Self**
  - Stored proprioception, self-limitations, goals, intentions, plans
- **Environment**
  - Navigable spaces, structures
- **Objects, people, other robots**
  - Detectable things in the world
- **Actions**
  - Outcomes of specific actions in the environment
- **Task**
  - What needs to be done, where, in what order, how fast, etc.

Navigation challenges

- **Path planning problem**
  - Robot has a map, knows own and target positions
- **Localization problem**
  - Robot has a map showing target, doesn’t know own position
- **Coverage problem**
  - Robot has a map, knows where it is, but doesn’t know where the target is
- **Mapping problem**
  - Robot does not have a map, may known own position
- **Simultaneous localization and mapping**
  - Robot does not have a map, and doesn’t know own position

Navigation questions

- **Where am I going?**
  - Usually defined by human operator or mission planner
- **What is the best way to get there?**
  - Path planning problem
- **Where have I been?**
  - Mapping problem
- **Where am I?**
  - Localization problem

Different types of representation

- **Maze navigator robot**
  - Exact path it has taken: “Go straight 2m, turn left 90 deg, go straight…”. This is an **odometric path**
  - Sequence of moves at particular landmarks: “Left at 1st junction, right at 2nd junction, straight…”. This is a **landmark-based path**
  - What to do at each landmark: “At the green/red junction go left, at the red/blue junction go right, …”. This is a **landmark-based map**
  - The map of the maze. This is a **metric map**

Metric Maps and Topological Maps

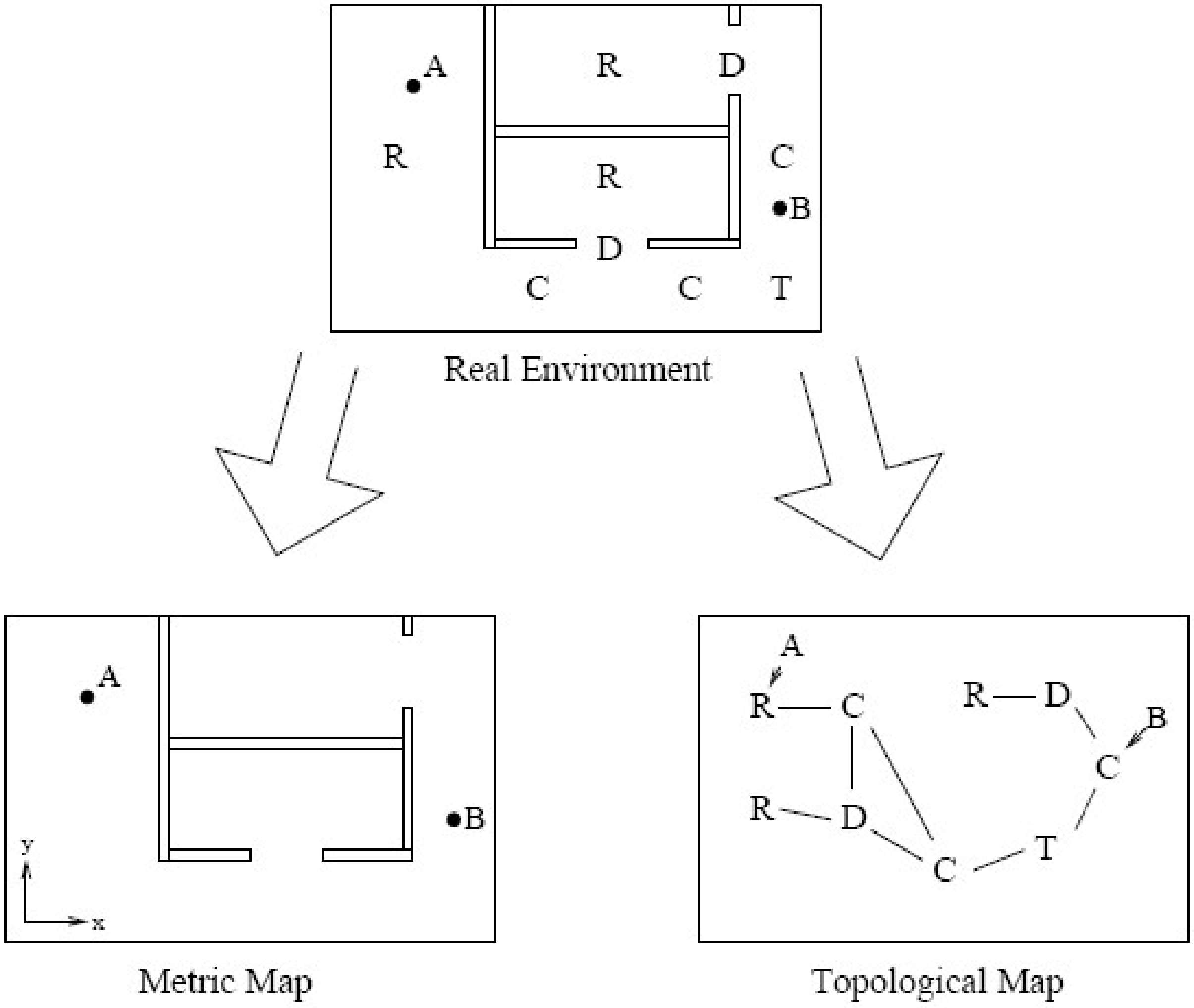


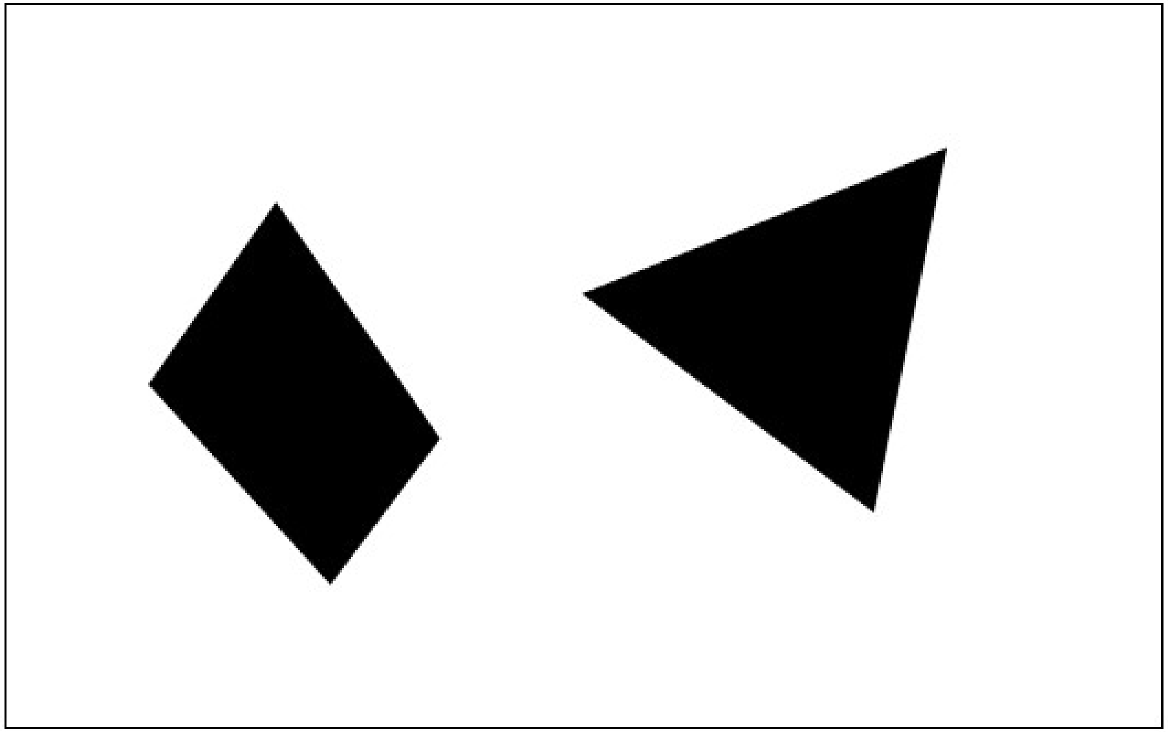
Figure from Meyer, “Map-based navigation in mobile robotics”, 2003, some other figures follow

Path Planning

- **Methodologies**
  - Roadmap
  - Cell decomposition
- **Roadmap**
  - Derive a graph from free space
  - Graph building
    - Visibility graph
    - Voronoi Diagram
- **Cell decomposition**
  - Free space is decomposed into simple regions (cells)
  - Path between two cells can be easily generated

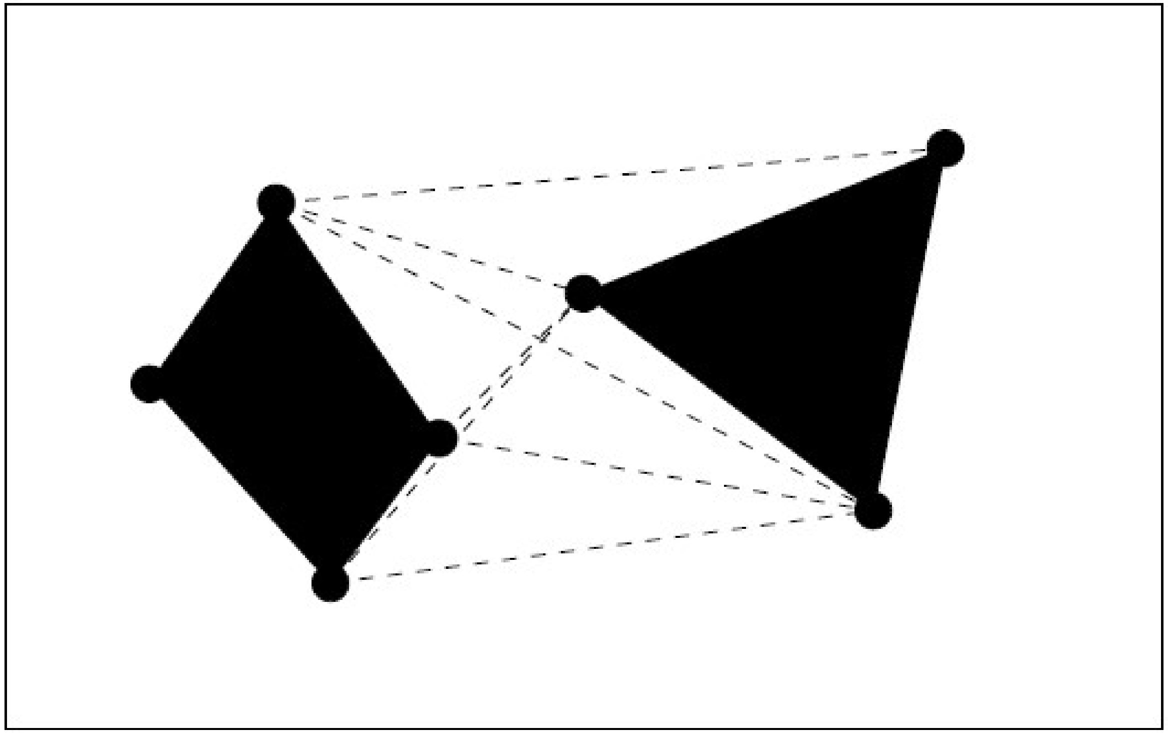
Visibility graph

- **Graph based representation**
  - Nodes are obstacles angles
  - Edges connect nodes that are visible from each other



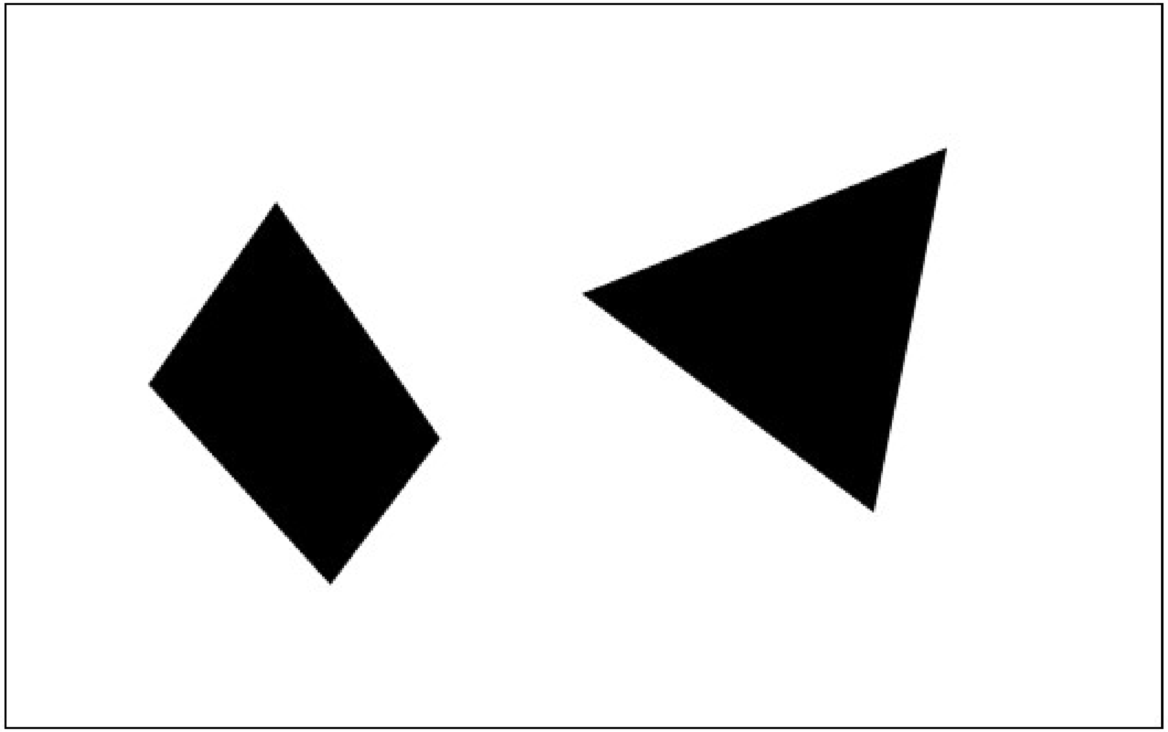
Visibility graph

- **Graph based representation**
  - Nodes are obstacles angles
  - Edges connect nodes that are visible from each other



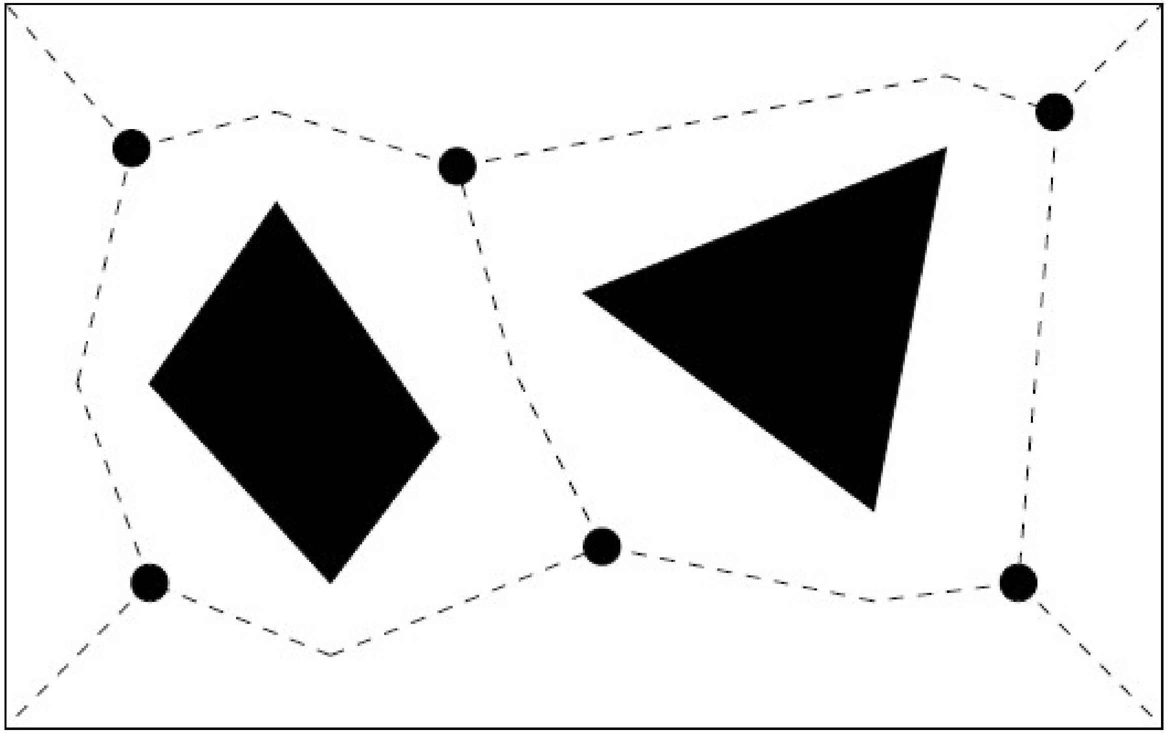
Voronoi diagram

- **Graph based representation**
  - Voronoi edges are equidistant to closest obstacles
  - Nodes are situated at the points where edges meet



Voronoi diagram

- **Graph based representation**
  - Voronoi edges are equidistant to closest obtacles
  - Nodes are situated at the points where edges meet



Graph based planning

- **Search the graph to find optimal path**
- **Which path is optimal?**
  - Minimal distance
  - Safest
  - Best view!
- **Searching algorithms**
  - Dikjstra and A* Algorithm
  - D*, Focused D*, D* Lite (https://www.youtube.com/watch?v=skK-3UfcXW0&ab_channel=CSMinute, http://idm-lab.org/bib/abstracts/papers/aaai02b.pdf, http://idm-lab.org/project-a.html)
  - RRT, RRT* (https://www.youtube.com/watch?v=Ob3BIJkQJEw&ab_channel=AaronBecker, https://www.youtube.com/watch?v=QR3U1dgc5RE&t=95s&ab_channel=MATLAB)

Dijkstra algorithm

1. **Init**
   - Assign starting node with a 0 distance, all other nodes with infinite distance, current = start, visited = {}
2. **Update minimum distances of neighbors to current node**
   - While updating minimum distance keep track of previous node in minimum path
3. **Add current to visited set**
4. **Current = minimum distance node AND not in visited**
5. **Repeat from step 2 until current = target**

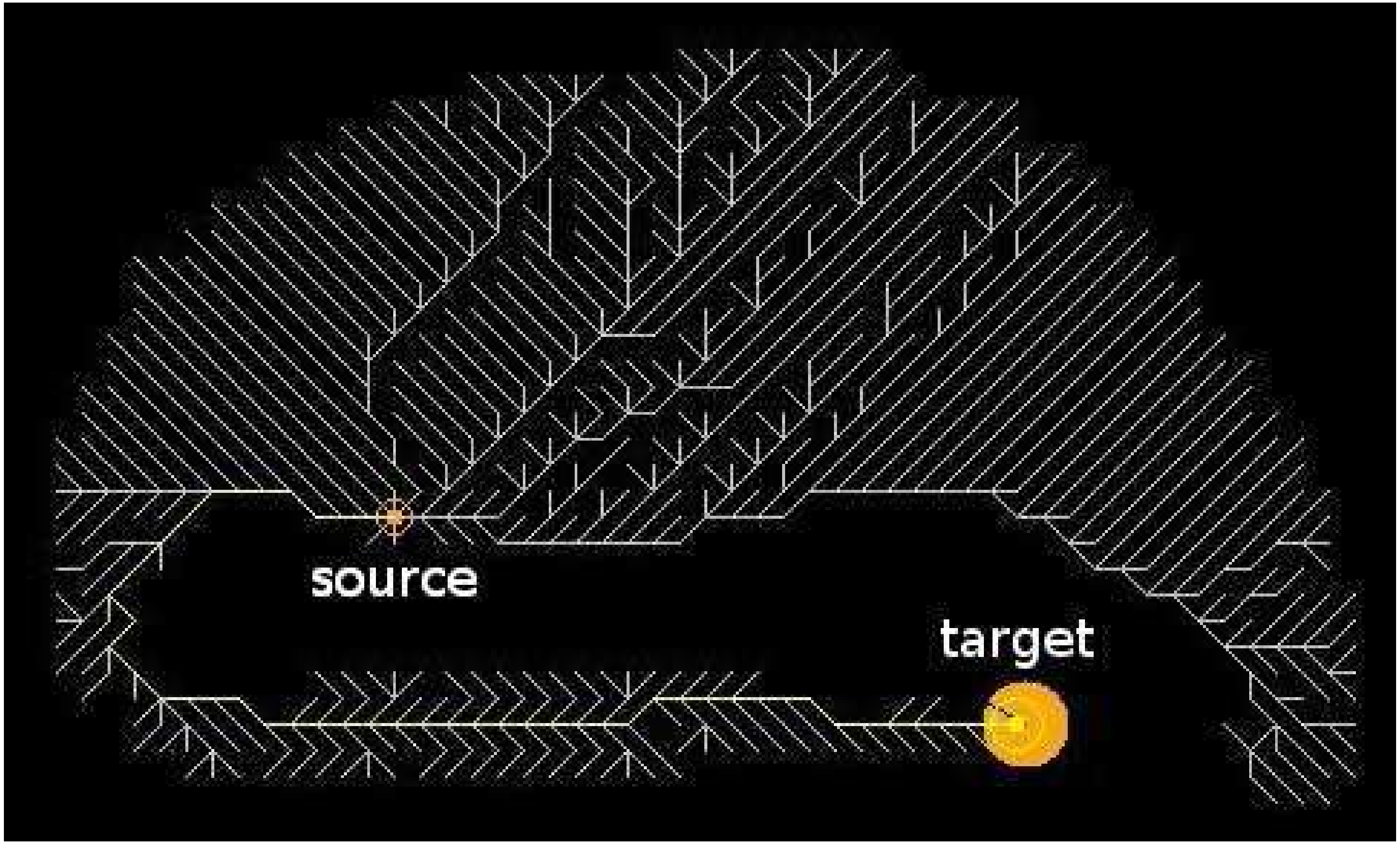
A* algorithm

Similar to Dijkstra but selection takes into distance to target into account:

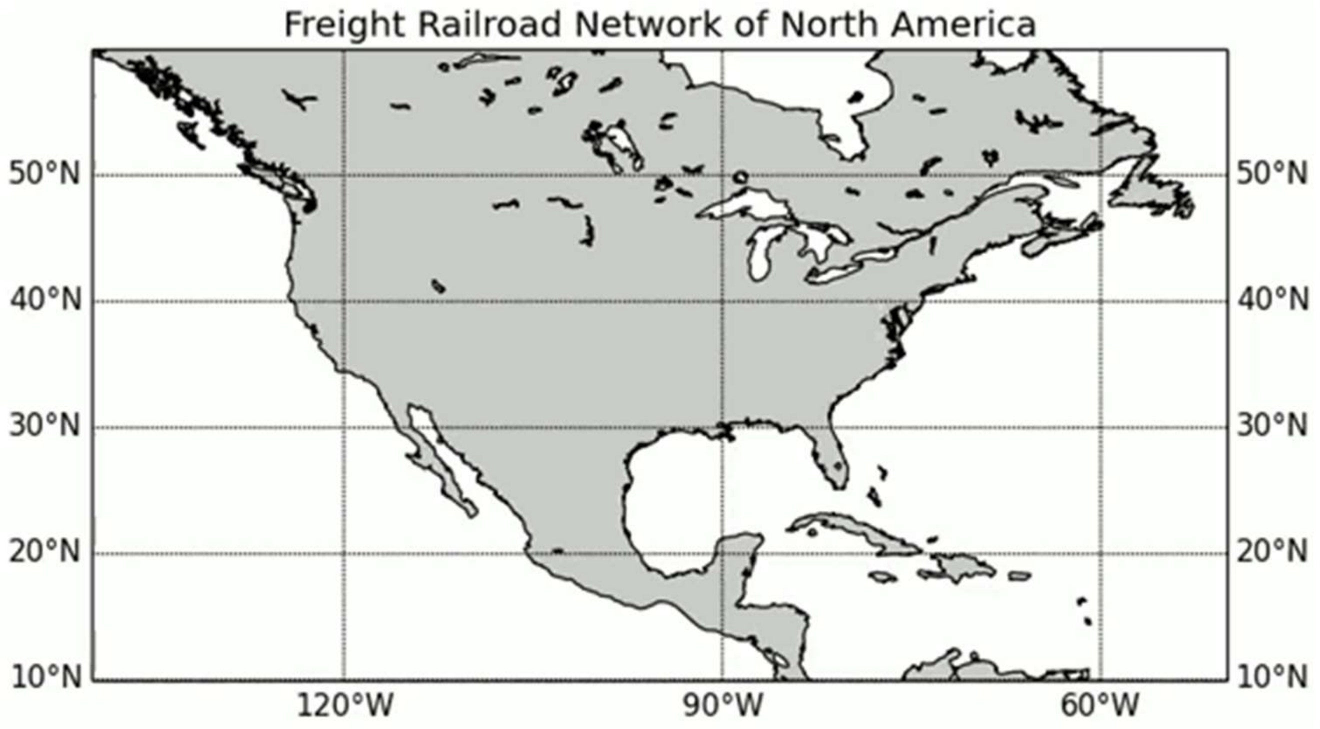
4. Current = minimum distance to start + euclidian distance to target node AND not in visited

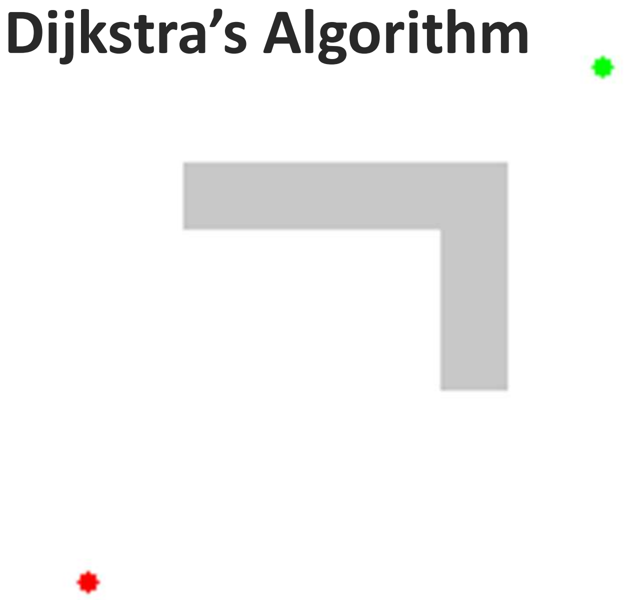
- Returns optimal path
- Tends to search in the direction of the target

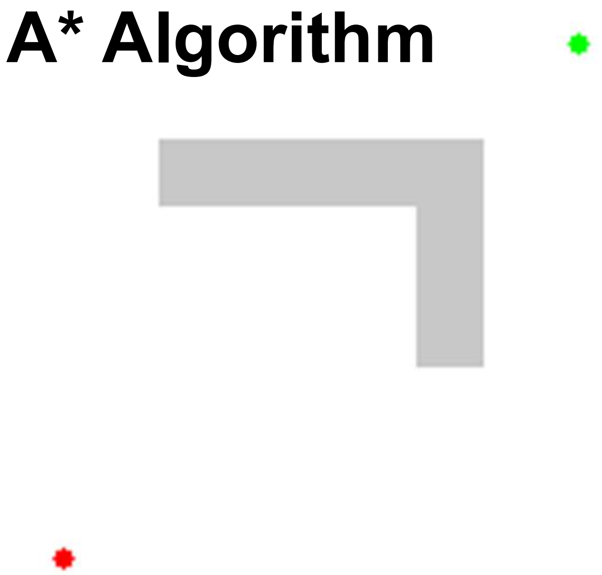
A* algorithm



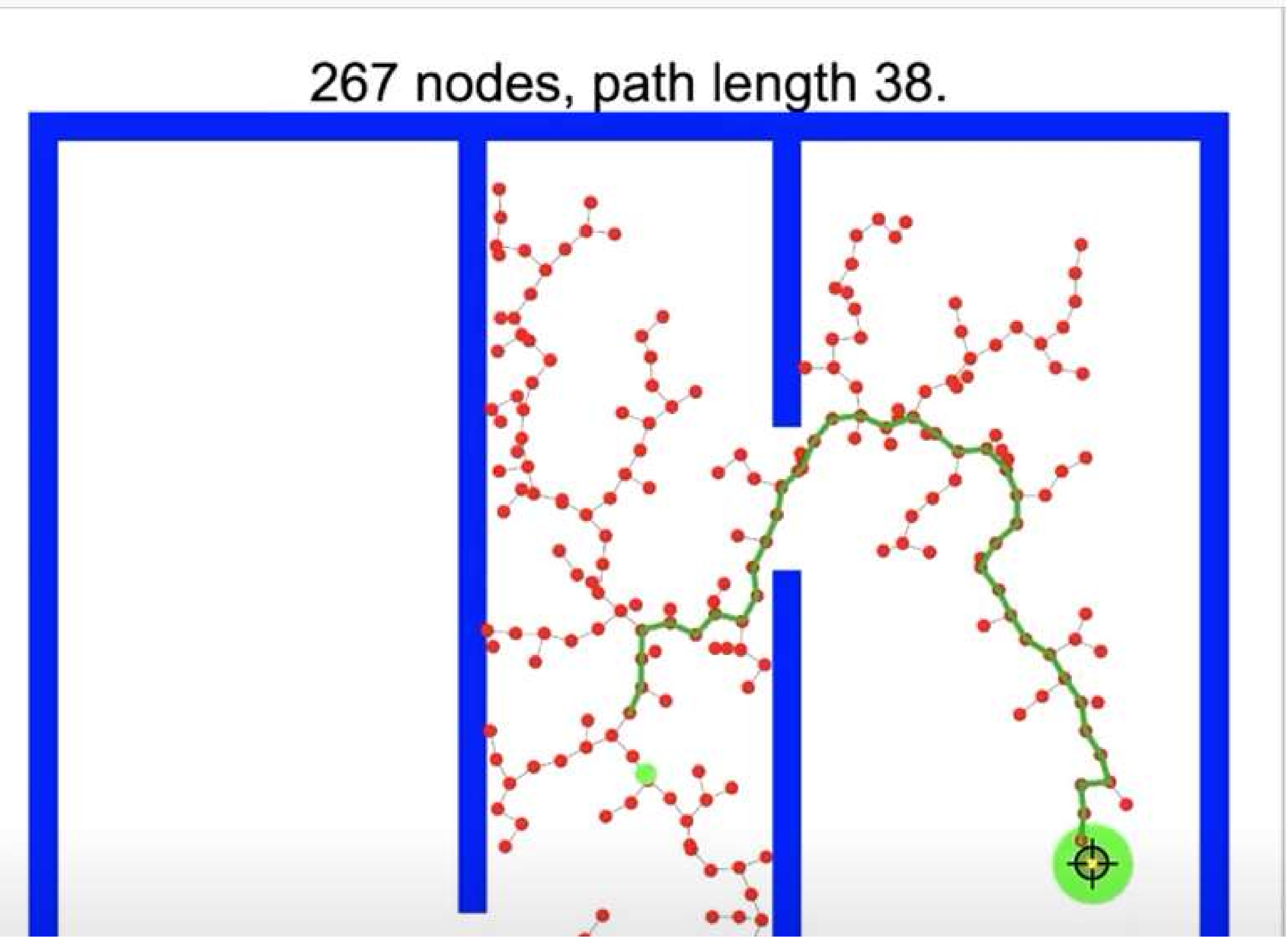
A* Algorithm Working



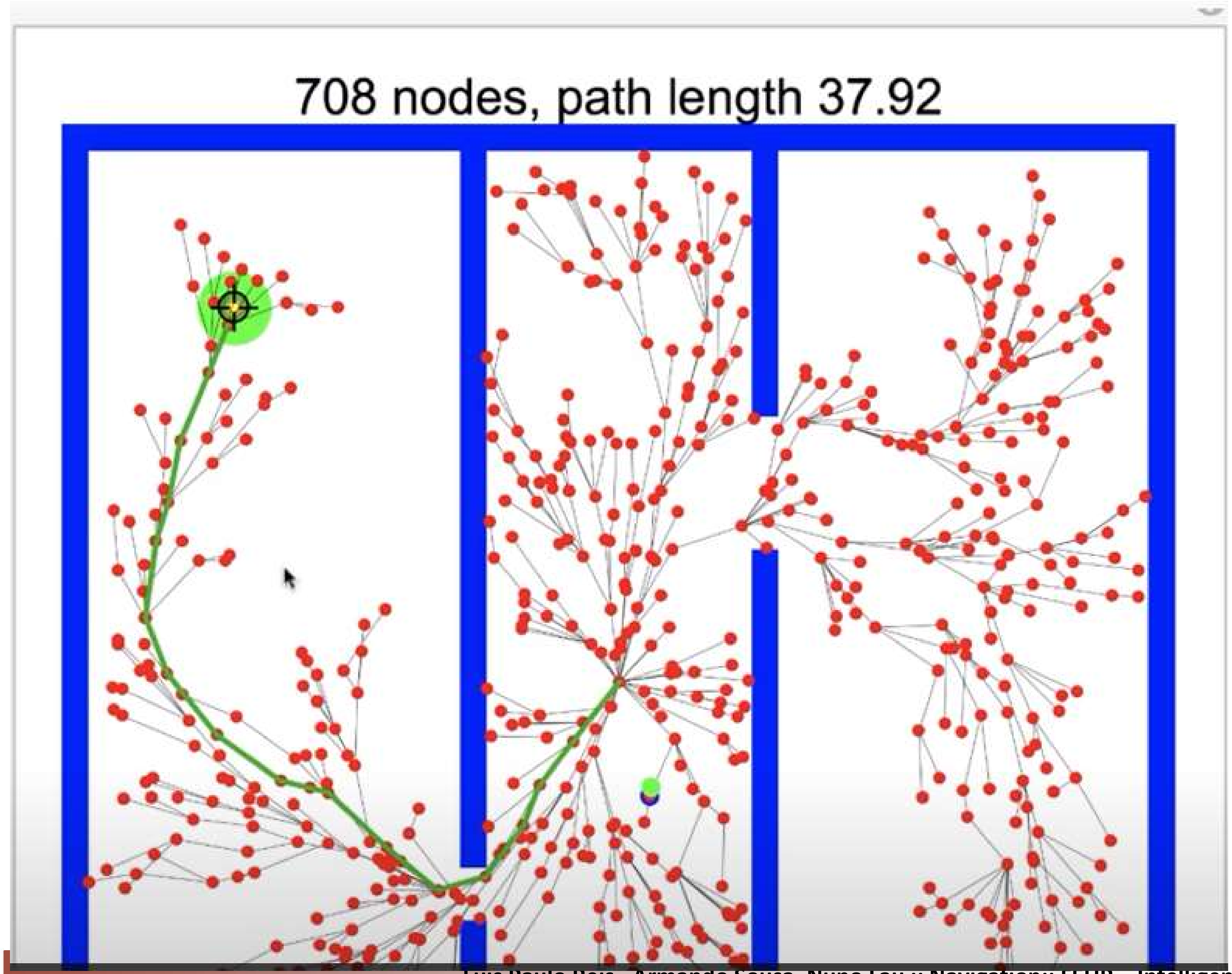




RRT - Rapidly-exploring Random Tree



RRT*



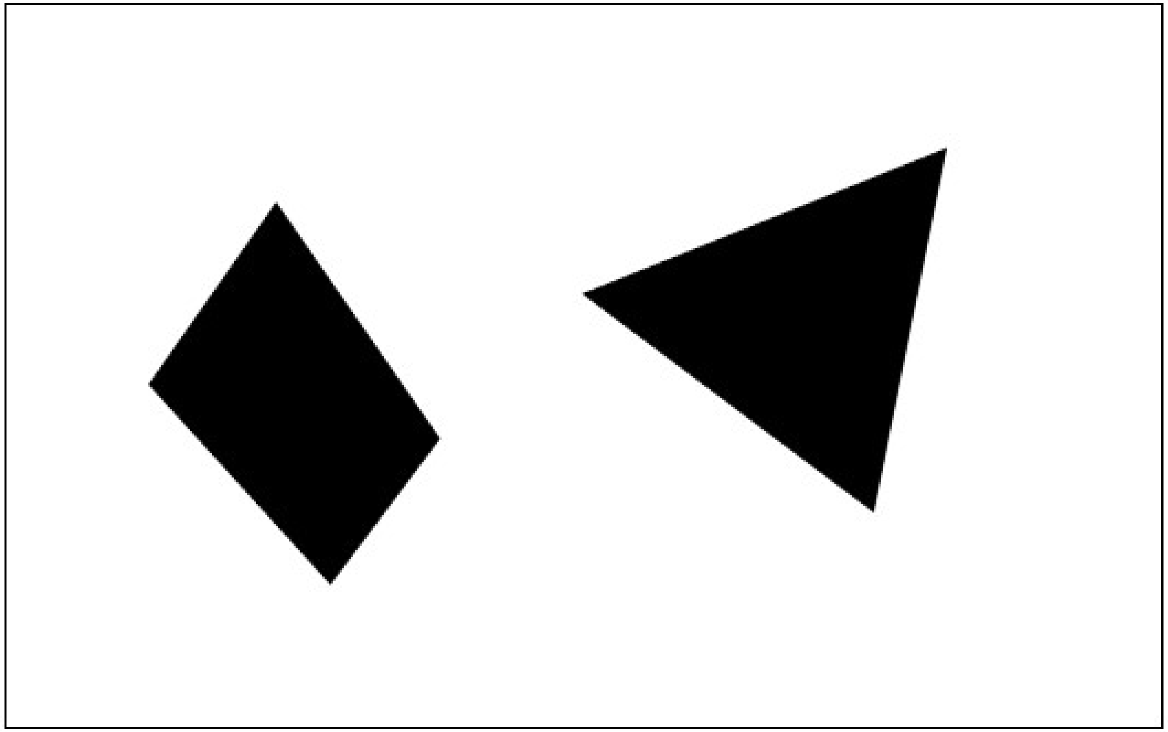
708 nodes, path length 37.92

Cell decomposition

- Exact cell decomposition
- Rectangular cell decomposition
- Regular cell decomposition
- Quadtree cell decomposition

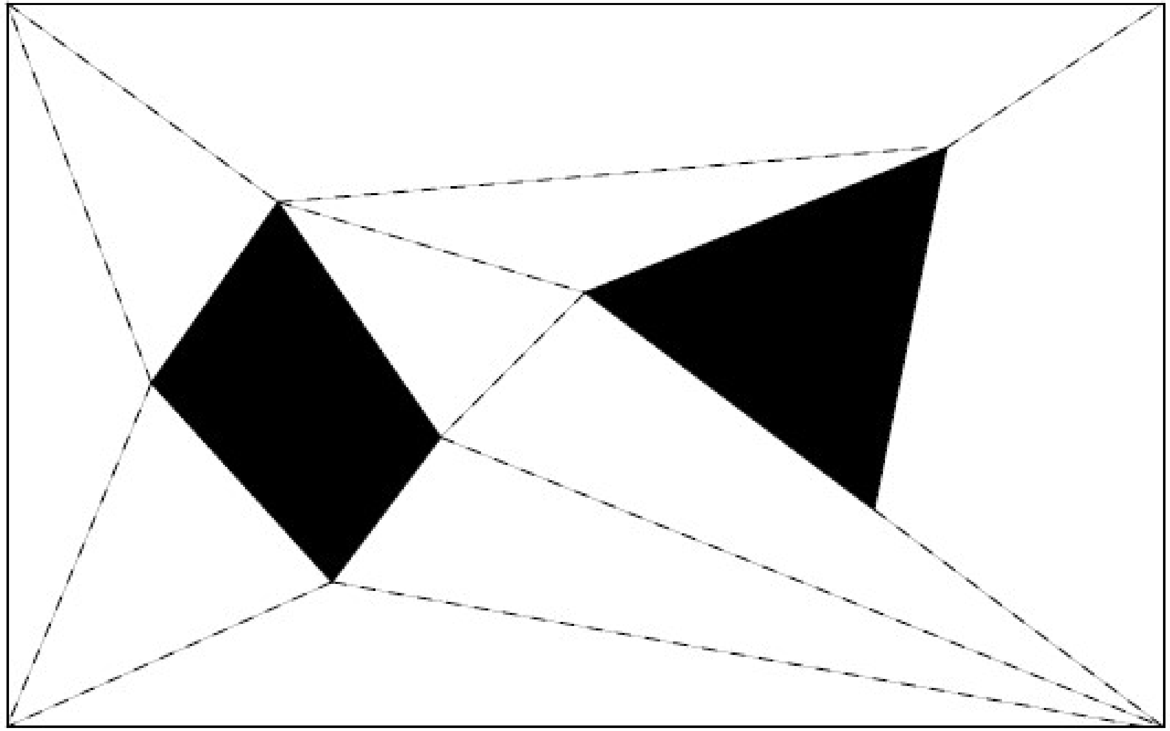
Cell decomposition

- **Exact cell decomposition**
  - Partition the free space into convex polygons



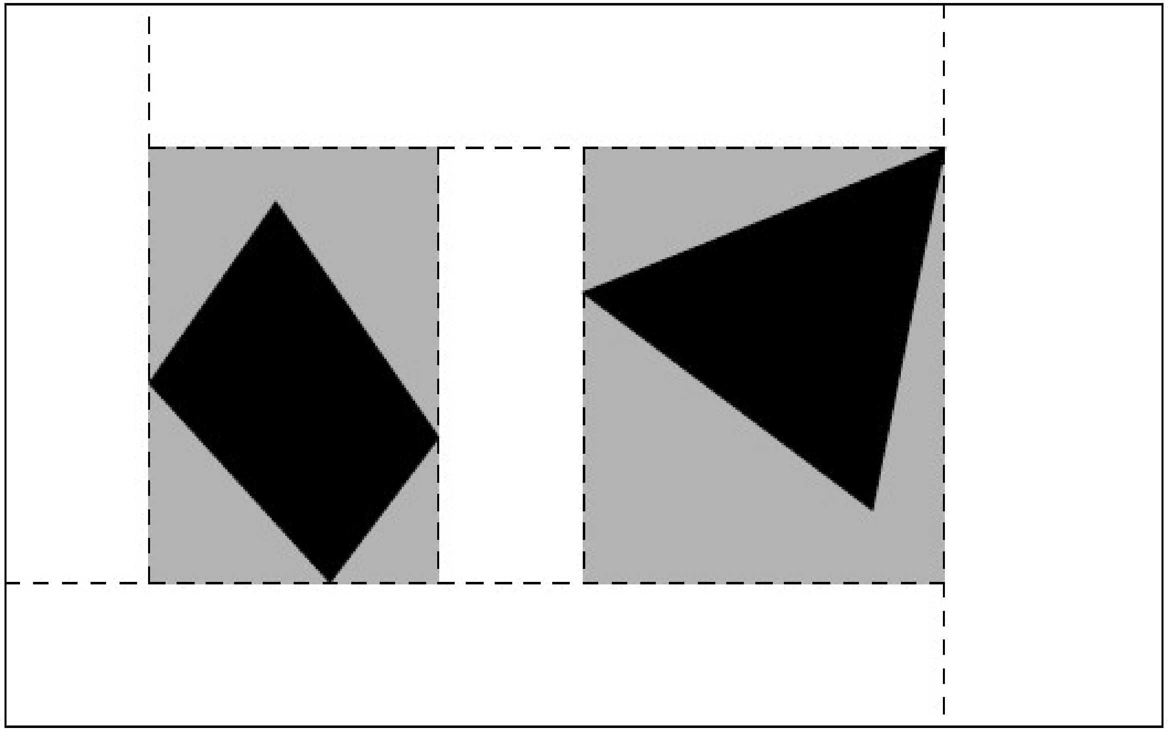
Cell decomposition

- **Exact cell decomposition**
  - Partition the free space into convex polygons



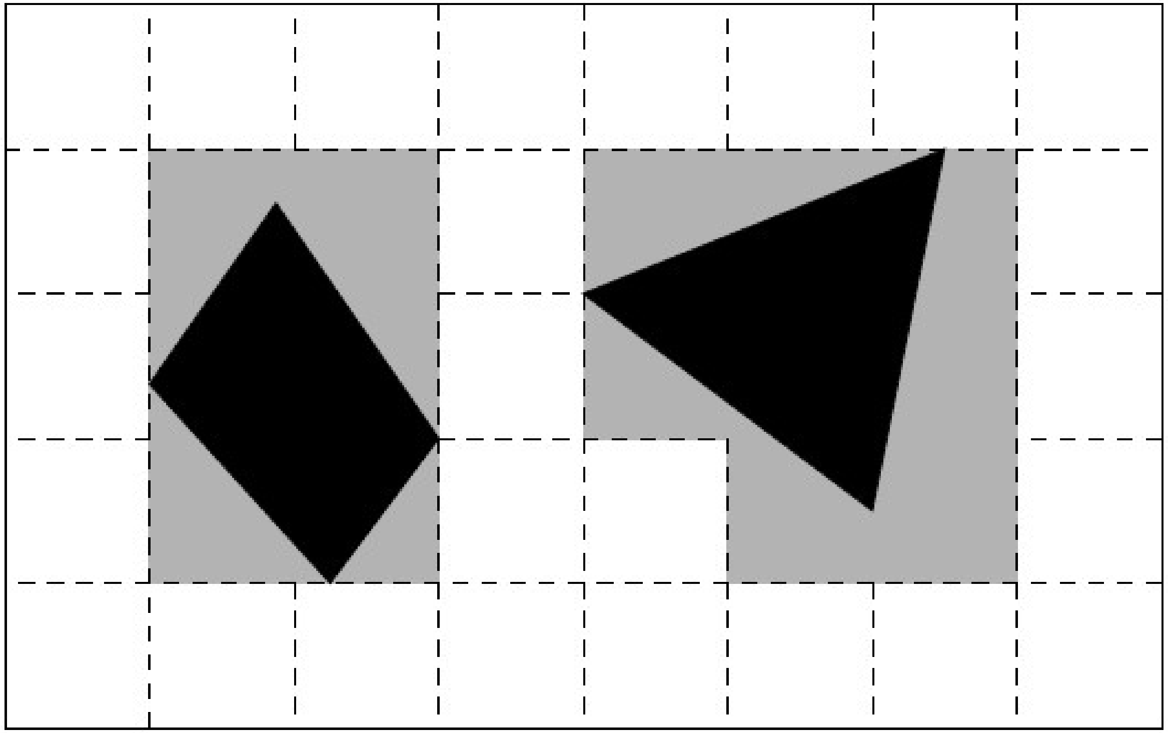
Cell decomposition

- **Rectangular cell decomposition**



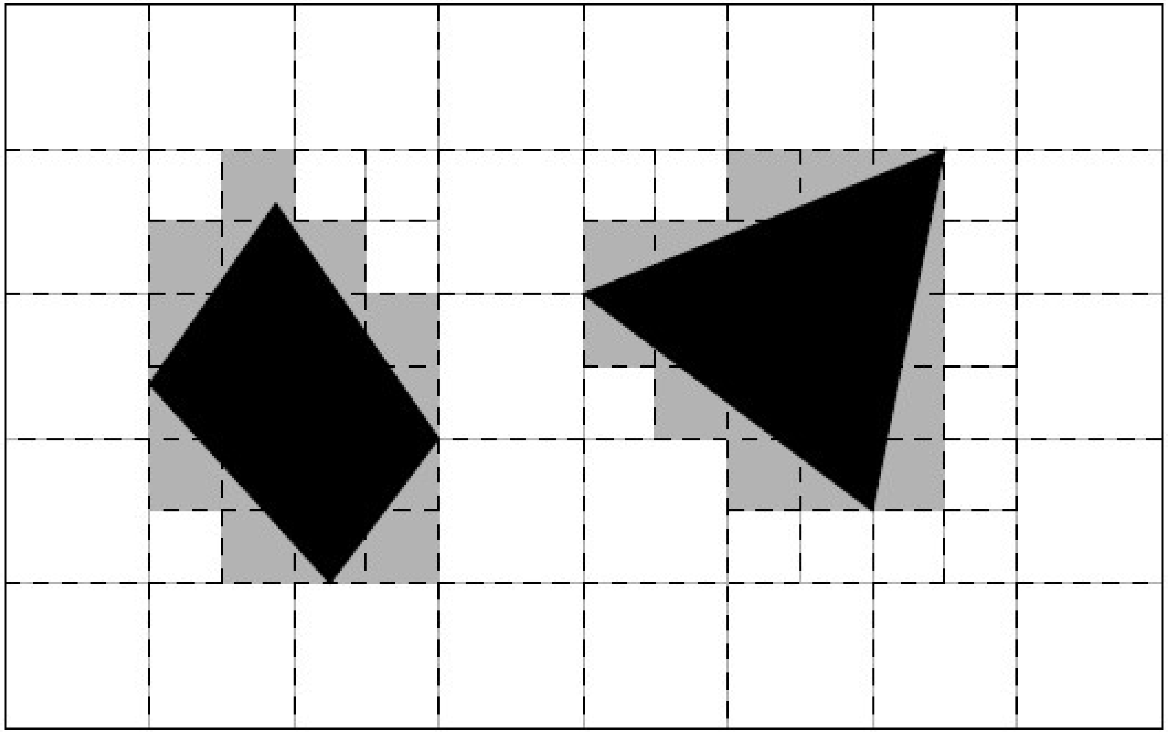
Cell decomposition

- **Regular cell decomposition**

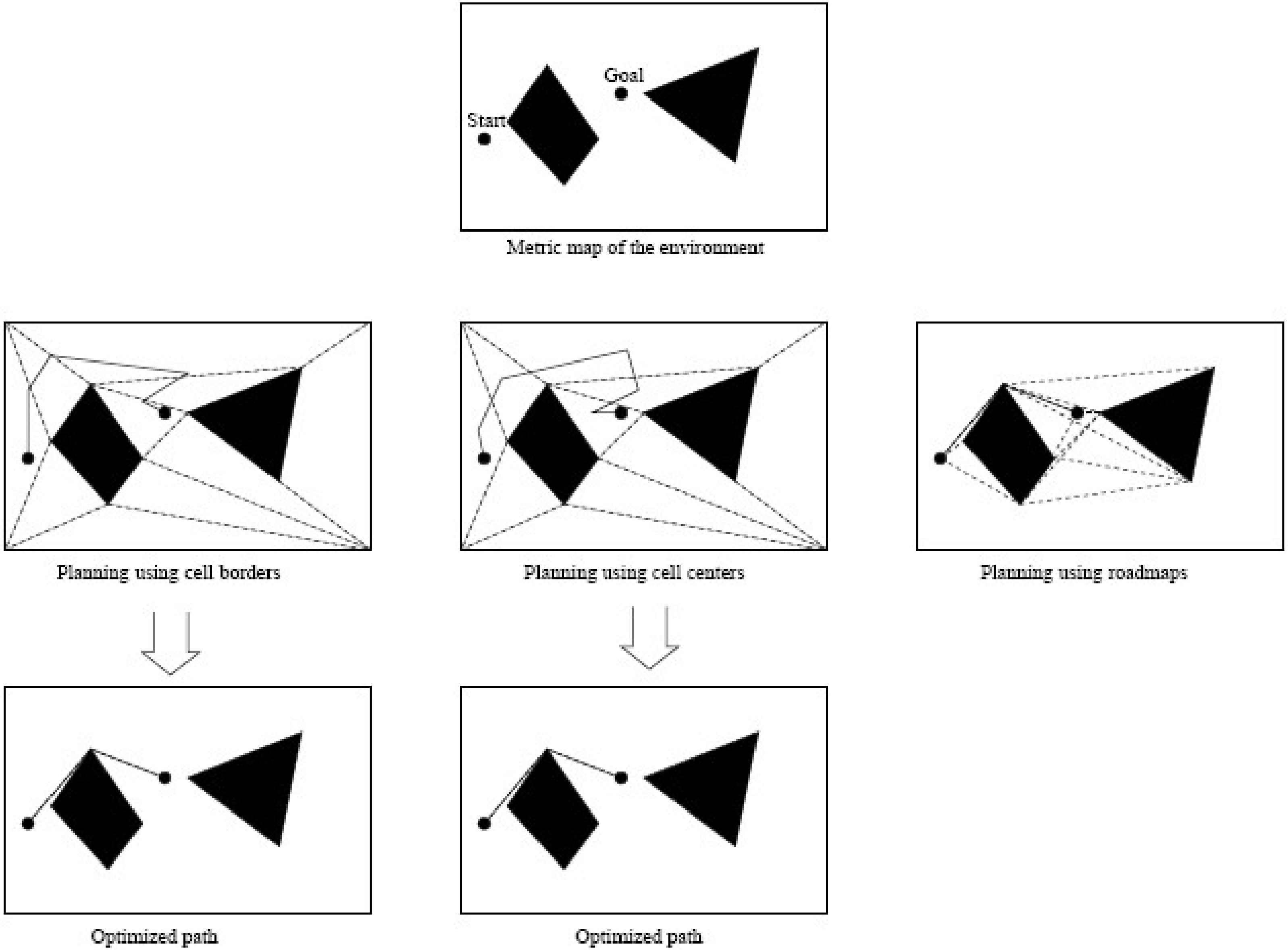


Cell decomposition

- **Quadtree cell decomposition**



Planning



Metric map of the environment

Planning using cell borders

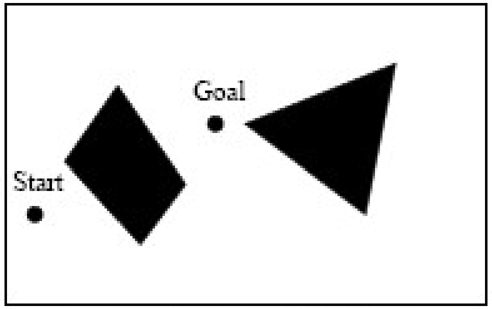
Planning using cell centers

Planning using roadmaps

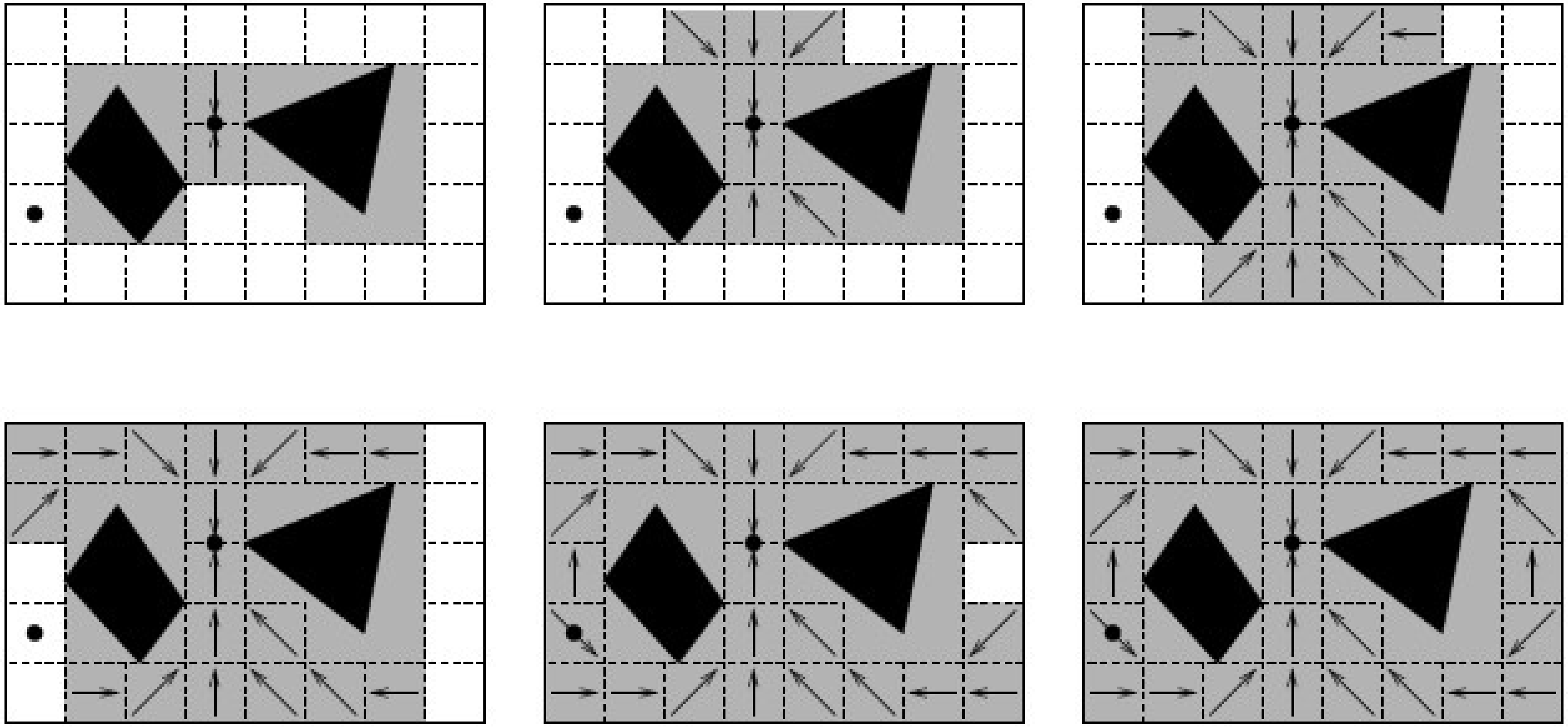
Optimized path

Optimized path

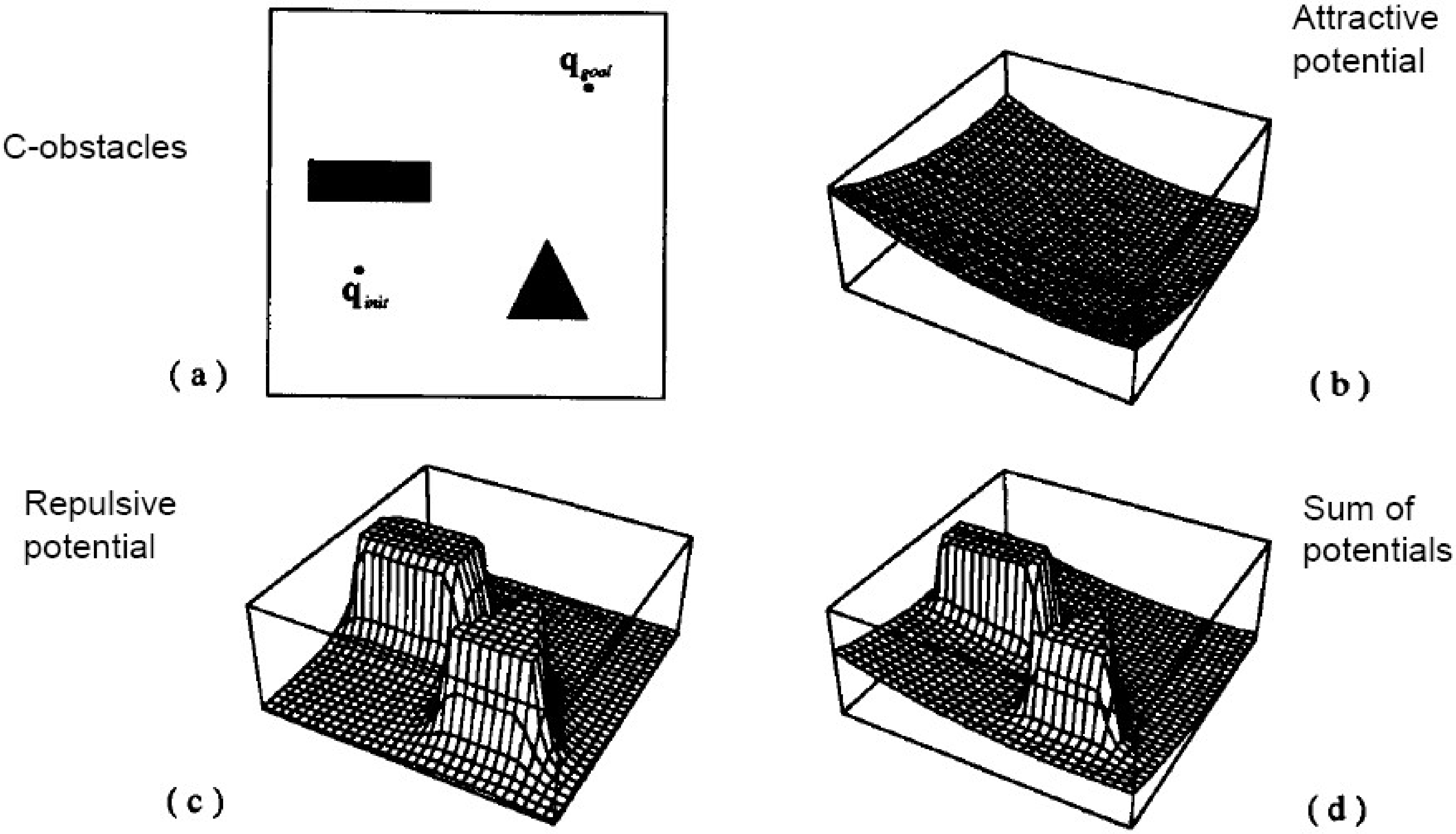
Wavefront planning



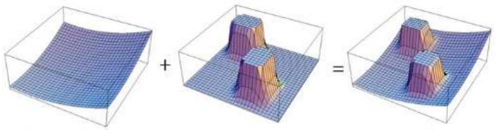
Metric map of the environment



Potential Field Local Planning

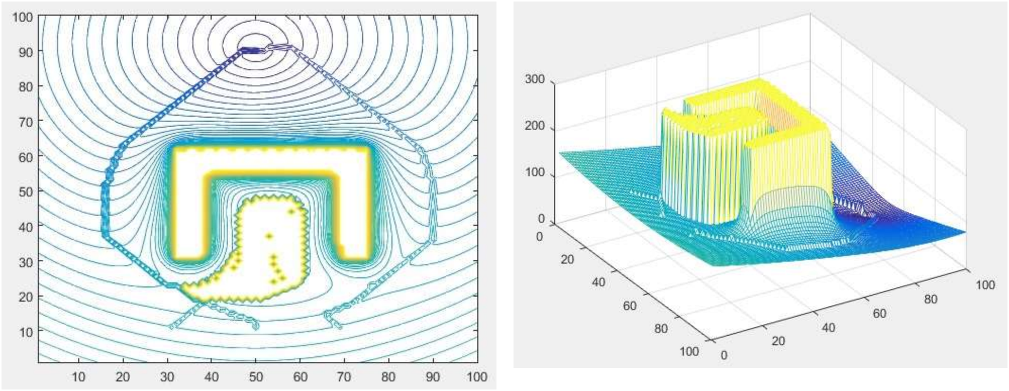


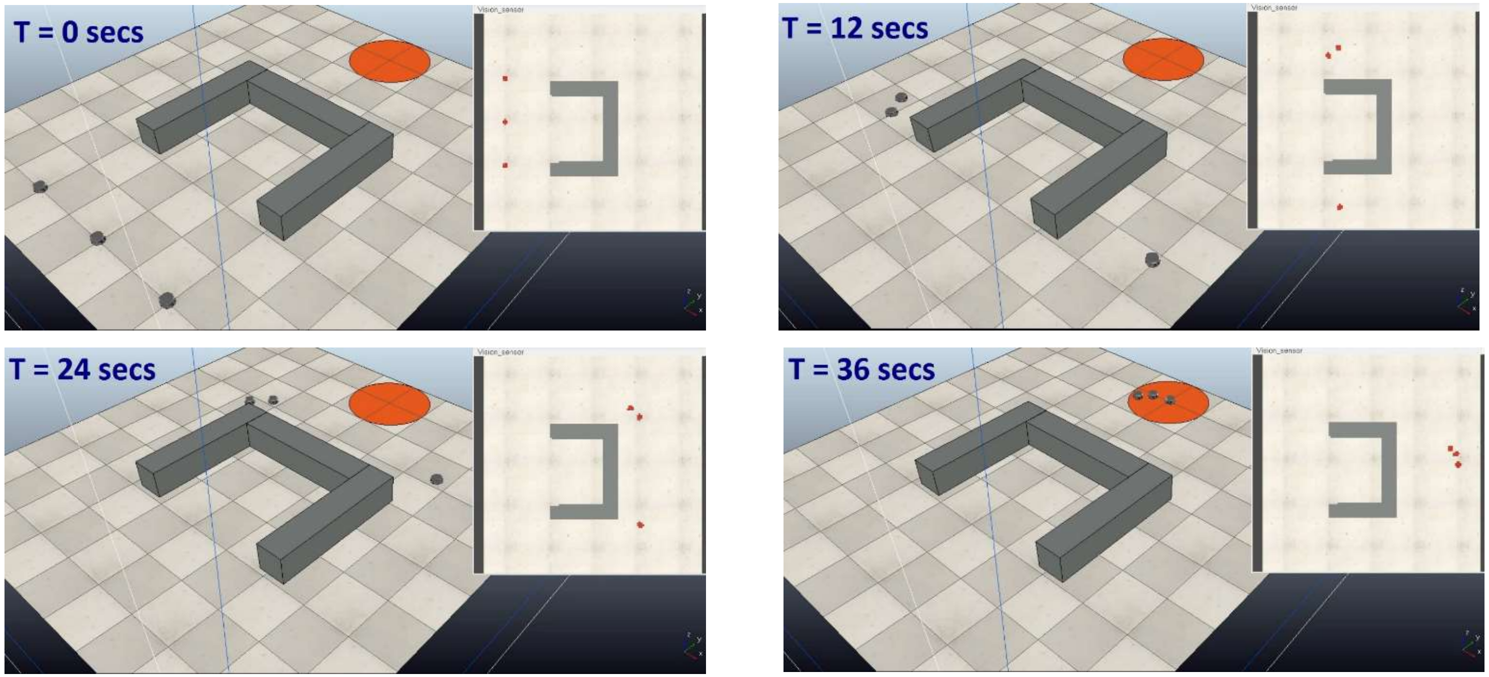
Potential Field (Virtual Obstacle)



Rimon, E., Koditschek, D.E.: Exact robot navigation using artificial potential functions. IEEE Transactions on robotics and automation 8(5), 501–518 (1992)

Ge, S.S., Cui, Y.J.: Dynamic motion planning for mobile robots using potential field method. Autonomous robots 13(3), 207–222 (2002)



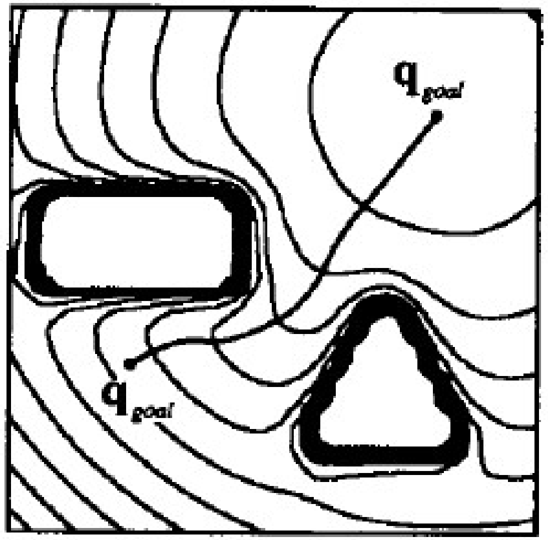


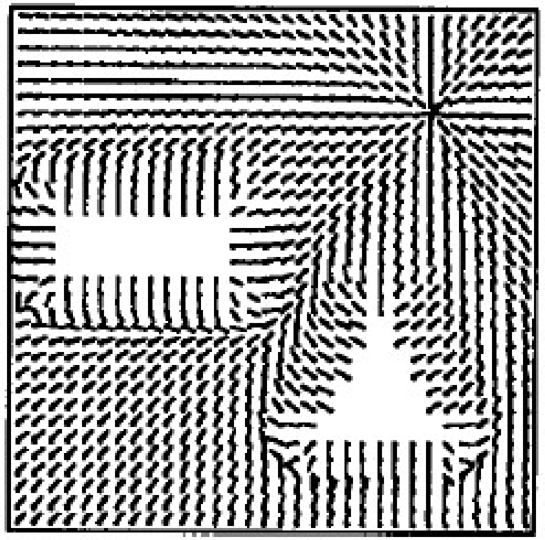
Abdelrahman M. Hassan 1 , Catherine M. Elias 1 , Omar M. Shehata 1 , Ahmed Hussein 2 , and Elsayed I. Morgan 1

Potential Field Local Planning

Equipotential contours

(e)





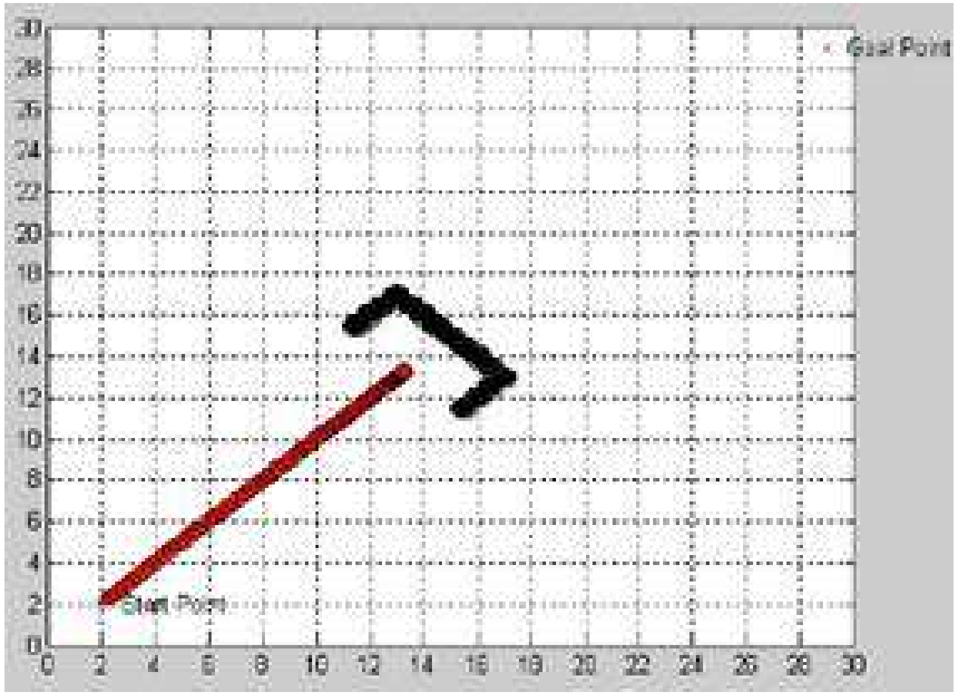
Negative gradient

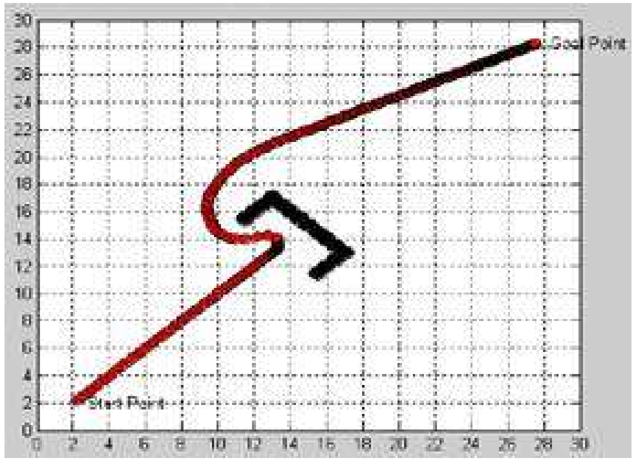
(f)

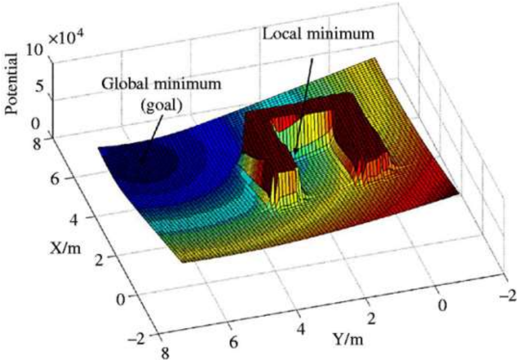
From
Robot Motion Planning
J.C. Latombe

Potential Field Planning

- http://www.cs.mcgill.ca/~hsafad/robotics/index.html



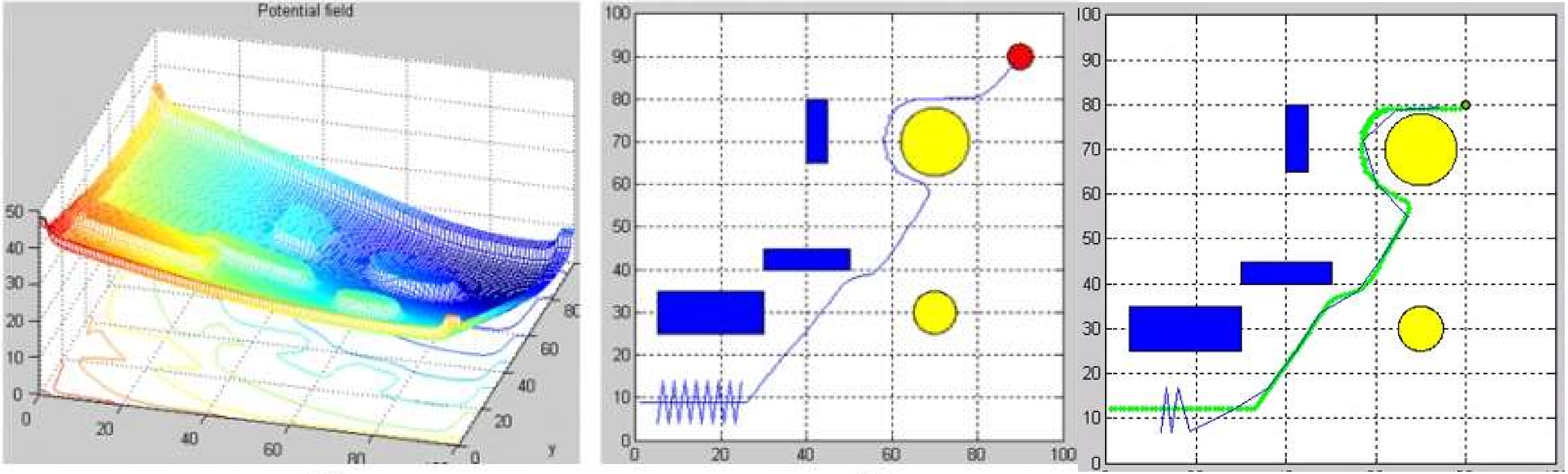




http://www.emeraldinsight.com/journals.htm?issn=0143-991X&volume=37&issue=4&articleid=1846407&show=pdf

Also Manipulators…

- http://taylorwang.files.wordpress.com/2012/04/potential-field1_robot.jpg
- http://taylorwang.wordpress.com/2012/04/06/collision-free-path-planning-using-potential-field-method-for-highly-redundant-manipulators/
- http://youtu.be/QTp1HRjXSSc



(a)

(b)

Also in Swarm

- http://youtu.be/r9FD7P76zJs

Intelligent Robotics

Navigation

Luís Paulo Reis, Armando Sousa, Nuno Lau

lpreis@fe.up.pt

Director of LIACC – Artificial Intelligence and Computer Science Lab.

Associate Professor at Faculty of Engineering, University of Porto, Portugal

President of APPIA – Portuguese Association for Artificial Intelligence