PATH AND MOTION PLANNING (FOR MOBILE ROBOTS)

BEST SUMMER SCHOOL 2013

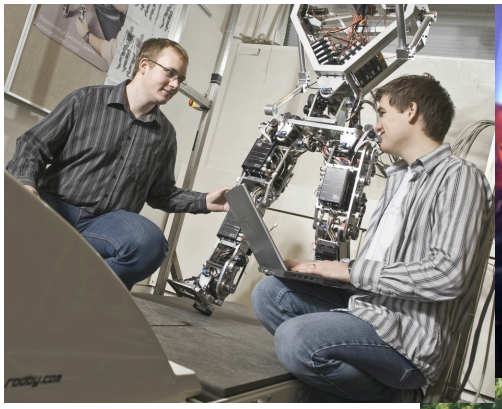
PROF. THOMAS BAK (TBA@ES.AAU.DK)

AALBORG UNIVERSITY
DENMARK

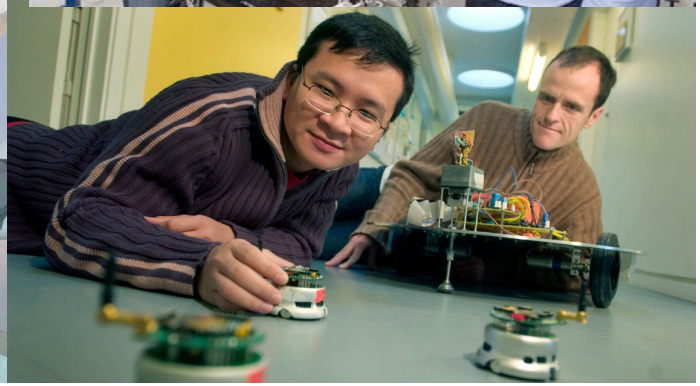
My robotics research

Gentler robots, ready to work with people

- Robots that sense, plan, and act in real and/or virtual worlds
- Algorithms and systems for representing, capturing, planning, controlling, and rendering motions
  - Motion planning in dynamic environments
  - Human-robot interaction




UNIVERSITY
AALBORG



Topics

- Applications and basic examples
  - Robotino
  - Segway RMP
- Motion planning context – deliberative/reactive robots
- The path and motion planning problem
- Robotino revisited
  - Obstacle avoidance
  - Path planning
- Planning methods – overview
- Rapidly-exploring Random Trees (RRT)

Applications

- Robot-assisted surgery
- Automated assembly plans
- Manufacturing
- Mobile robots
- Computational biology
- Digital actors
- Moving pianos around...

Examples

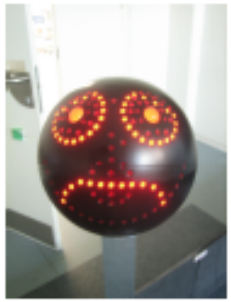
Robotino

Segway

Robotino – an experiment in human robot interaction

(a) Neutral

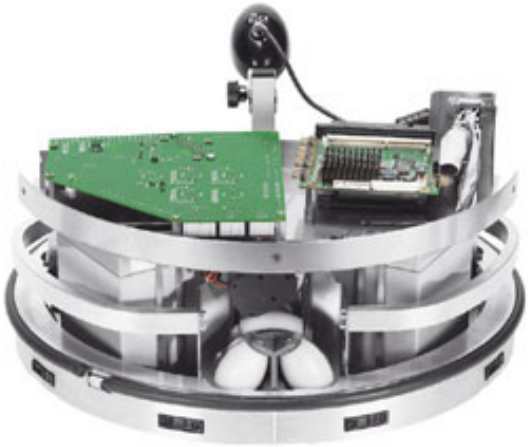
(b) Happy



(c) Sad

(d) Surprised

(e) Question mark



AALBORG UNIVERSITY
DENMARK

The SantaBot Experiment

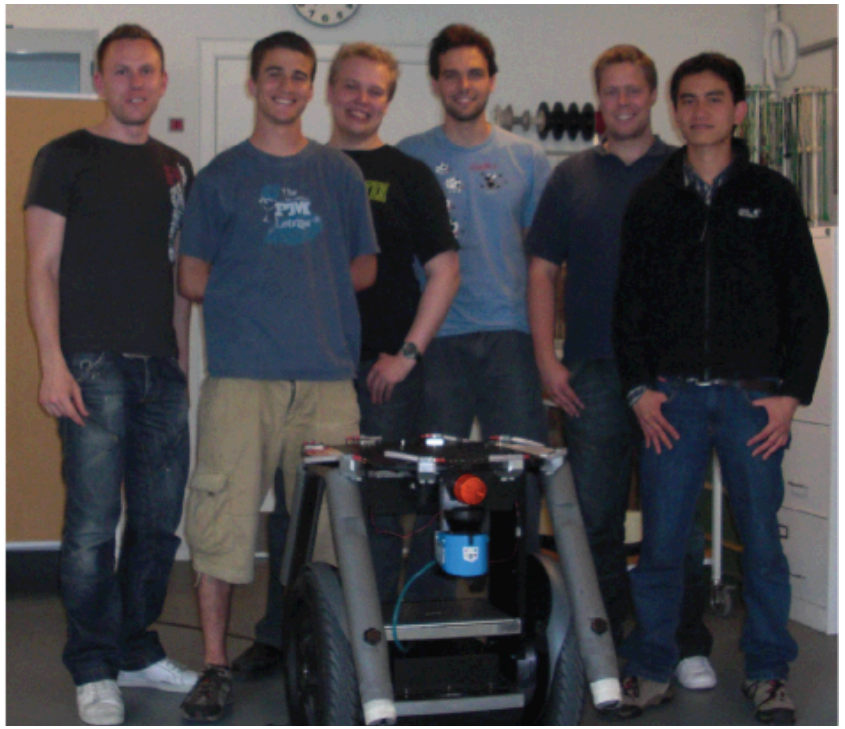
A Pilot Study of Human-Robot Interaction

Søren Tranberg, Center for Robot Technology, Danish Technological Institute
Mikael Svenstrup, Department of Electronic Systems, Aalborg Univeristy
Hans Jørgen Andersen, Department for Media Technology, Aalborg University
Thomas Bak, Department of Electronic Systems, Aalborg University
Ole B. Jensen, Department of Architecture & Design, Aalborg University

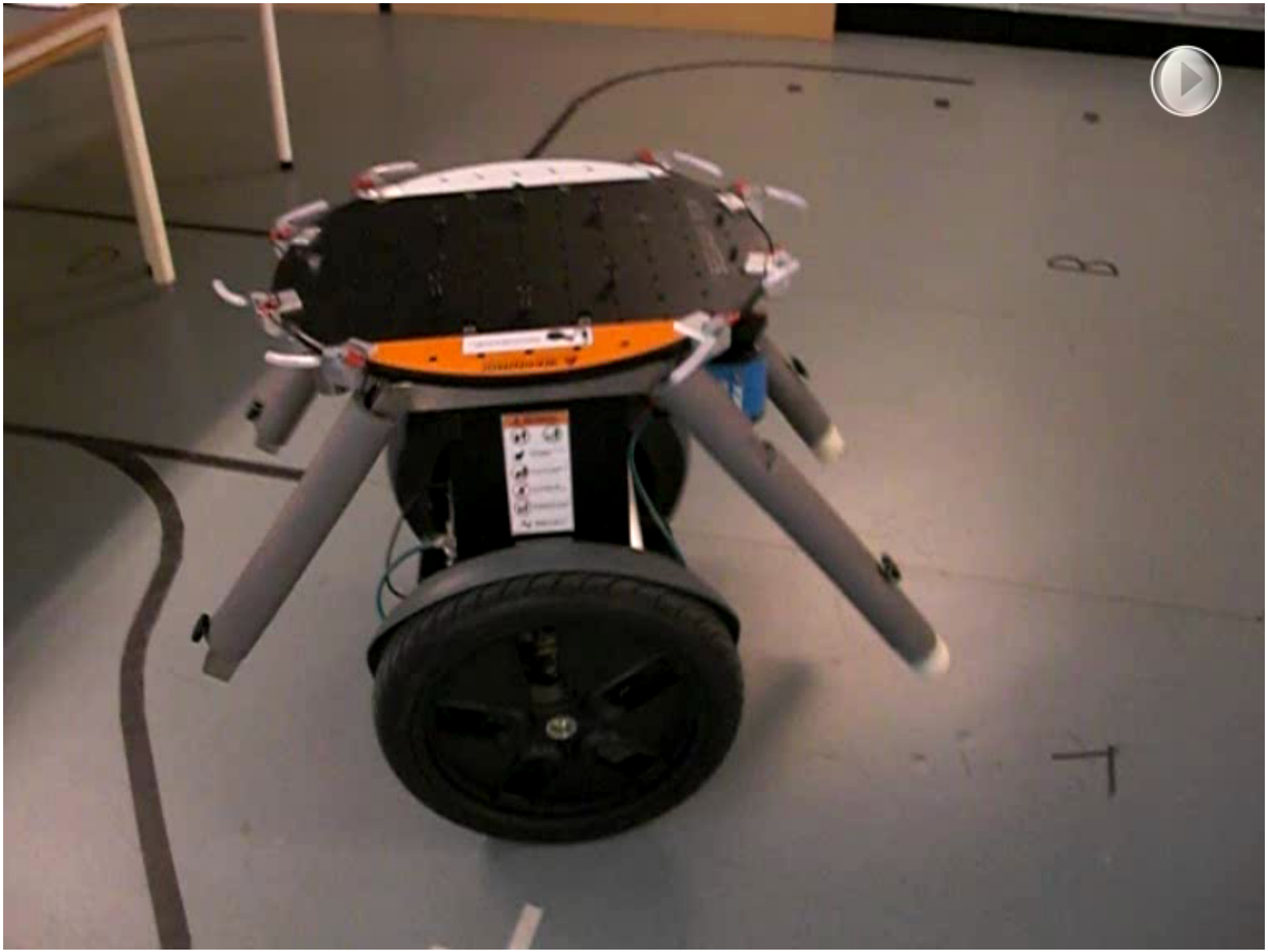
DANISH
TECHNOLOGICAL
INSTITUTE

AALBORG UNIVERSITY
DENMARK

Segway RPM



AALBORG UNIVERSITY
DENMARK

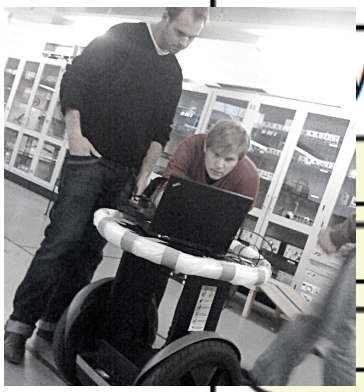


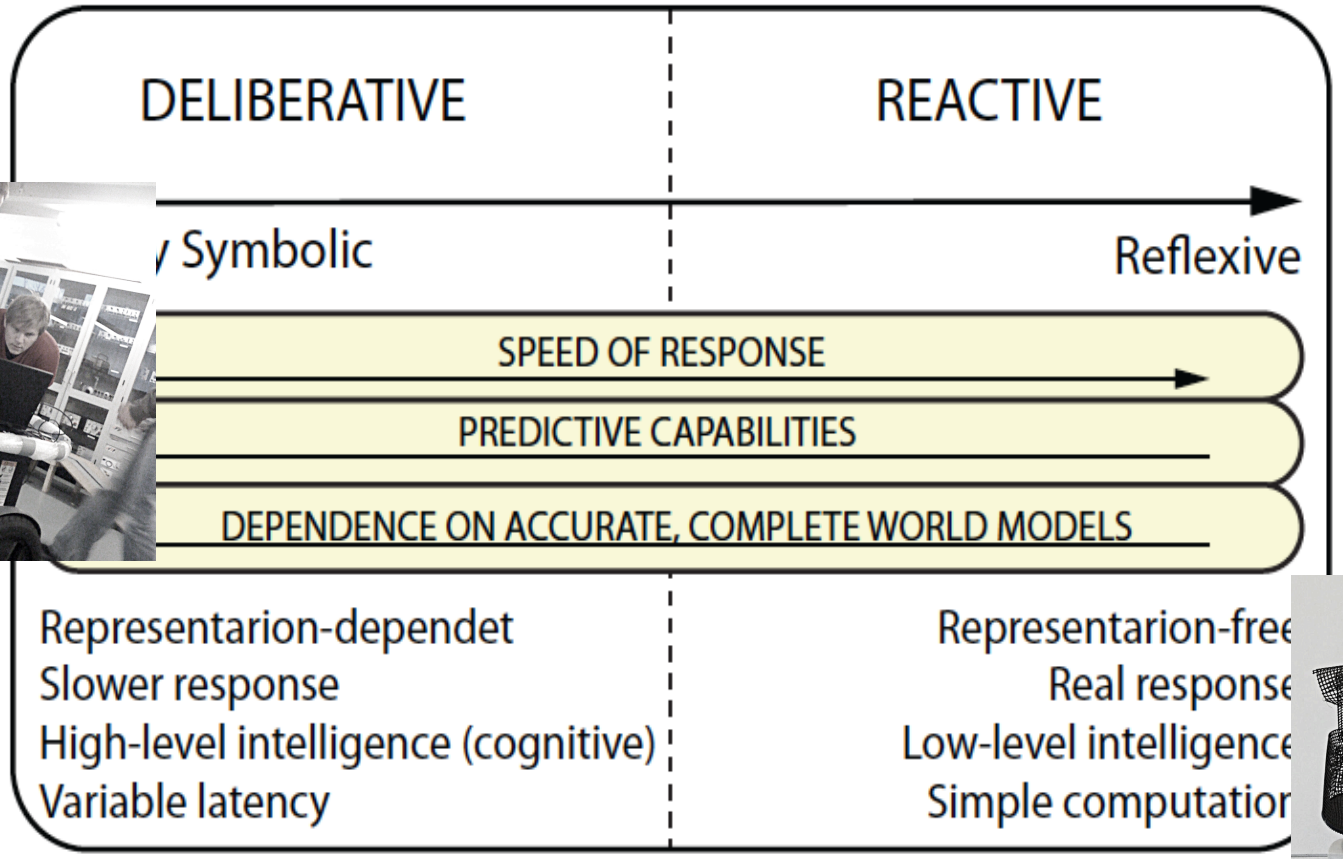
Motion and path planning context

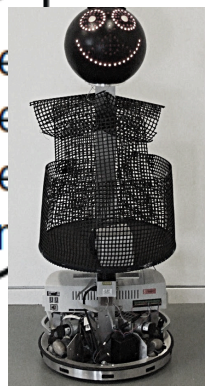
- Both robots have motion control (actuation of wheels)
- Both robots perceives the environment (via sensors)
- Both have localization:
  - Segway has a world model (global map)
  - Robotino is local (human centered)
- Both control orientation AND position (configuration)
- Motion and path planning is very different in the two cases

Problems

- How do we structure the software/hardware?
- How do we compute motion strategies?
- That help us achieve high-level goals such as
  - go to target without colliding with obstacles
  - build map of the environment

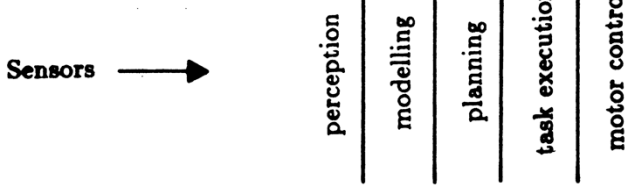




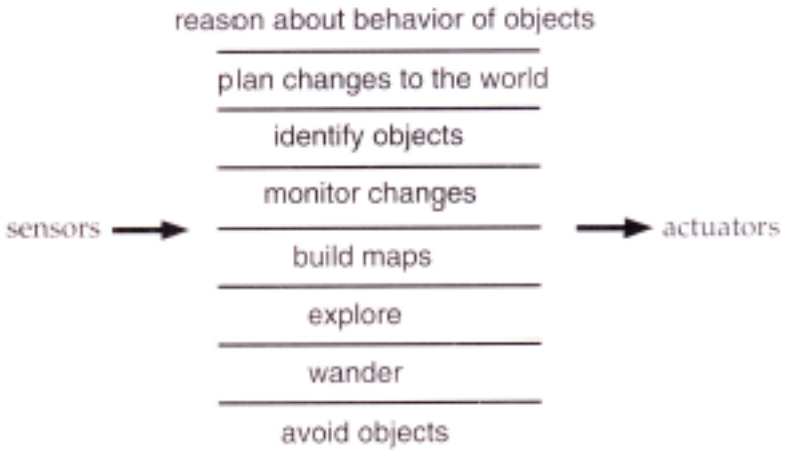


Arkin, 1998
Behaviour Based Robotics

Planning based vs. behaviour based decomposition



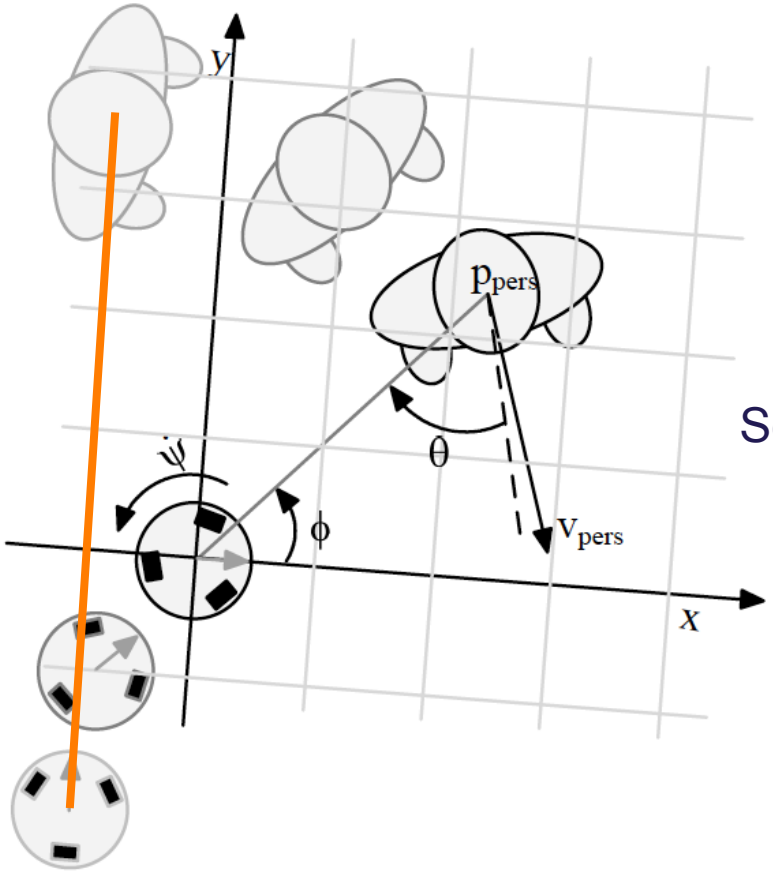
The traditional decomposition for an intelligent robot control system



The behaviour based decomposition approach

R. A Brooks 1986

Example – Robotino – reactive

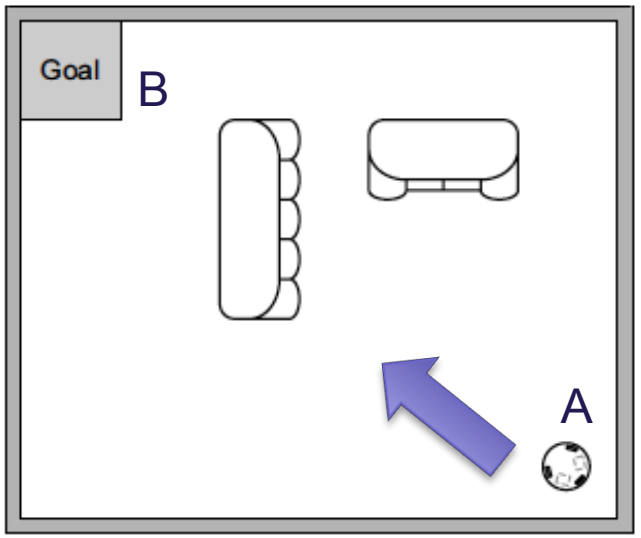


Sensors → Wander / Avoid obstacles / Rotate / Move closer /away → Actuators

Weighted sum of behaviors

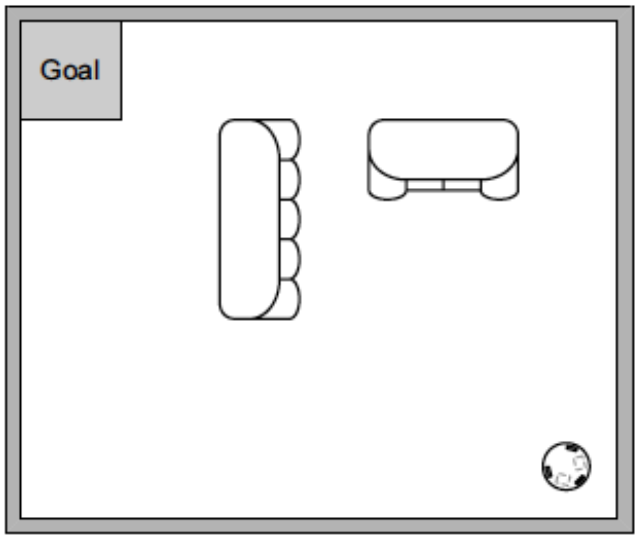
Basic problem – path and motion planning

- Compute a collision-free path from A (initial) to B (goal)
- Inputs:
  - Knowledge of robot and obstacles
  - Initial and goal robot configurations (position and orientation)
- Outputs:
  - Sequence of robot configurations connecting the initial and goal configurations



The world consists of …

- Obstacles
  - Already occupied spaces of the world
  - In other words, robots cannot go there
- Free Space
  - Unoccupied space within the world
  - Robots “might” be able to go here
  - To determine where a robot can go, we need to discuss what a Configuration Space is



Litterature

- Good overviews over planning algorithms can be found in the books
  - LaValle, S. (2006). Planning algorithms. Cambridge Univ Press
  - Thrun, S., Burgard, W., and Fox, D. (2005). Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). MIT Press
  - Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., and Thrun, S. (2005). Principles of Robot Motion: Theory, Algorithms, and Implementations. MIT Press
  - Siciliano, B. and Khatib, O. (2008). Handbook of Robotics. Springer-Verlag
  - J.C. Latombe, Robot Motion Planning, Kluwer Academic Publishers, 1991.

Goal of path and motion planning

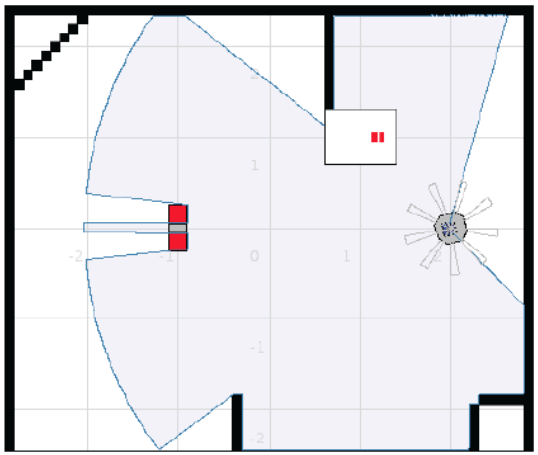
- Compute motion strategies:
  - geometric paths
  - time-parameterized trajectories
  - sequence of sensor-based motion commands
- To achieve high-level goals:
  - go to B without colliding with obstacles
  - assemble product P
  - build map of environment E
  - find object O

Extensions of the basic problem

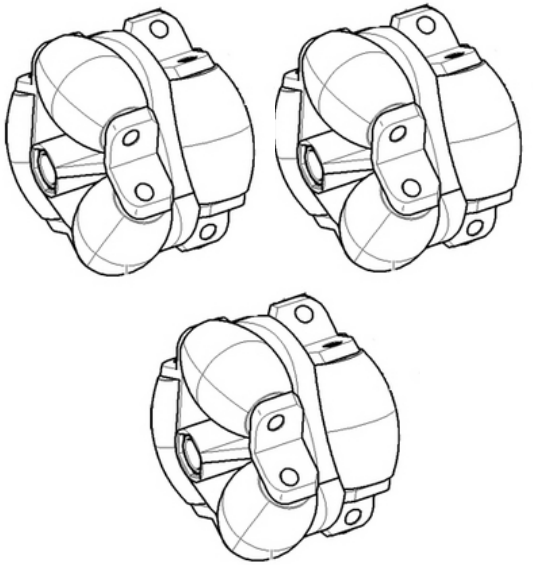
- More complex problems
  - Multiple robots
  - Higher dimensions
  - Obstacles and goals that move
  - Dynamic constraints (how sharp we can turn)
  - Physical models and deformable objects
  - Uncertainty in control of the robot

Example - Robotino

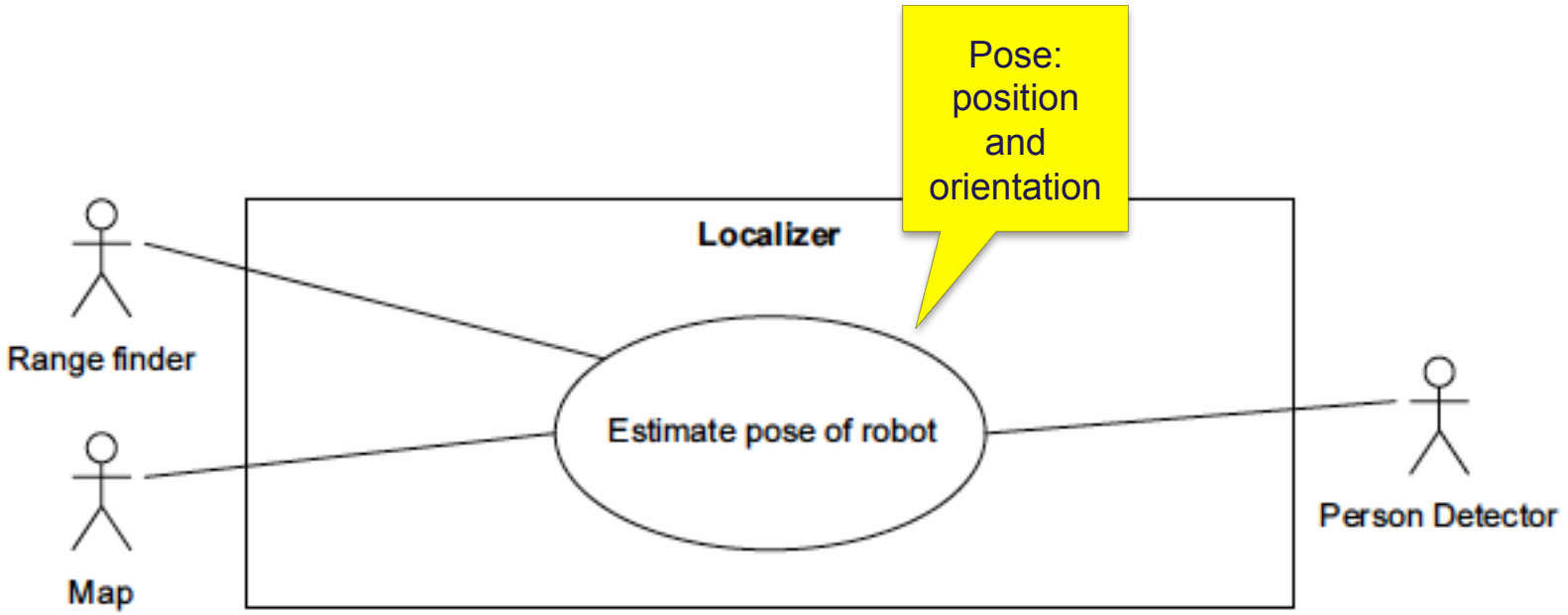
Sensors → Motion and path planning → Actuators



Localizer
Pilot (reactive)
Navigator
(deliberative functionality)

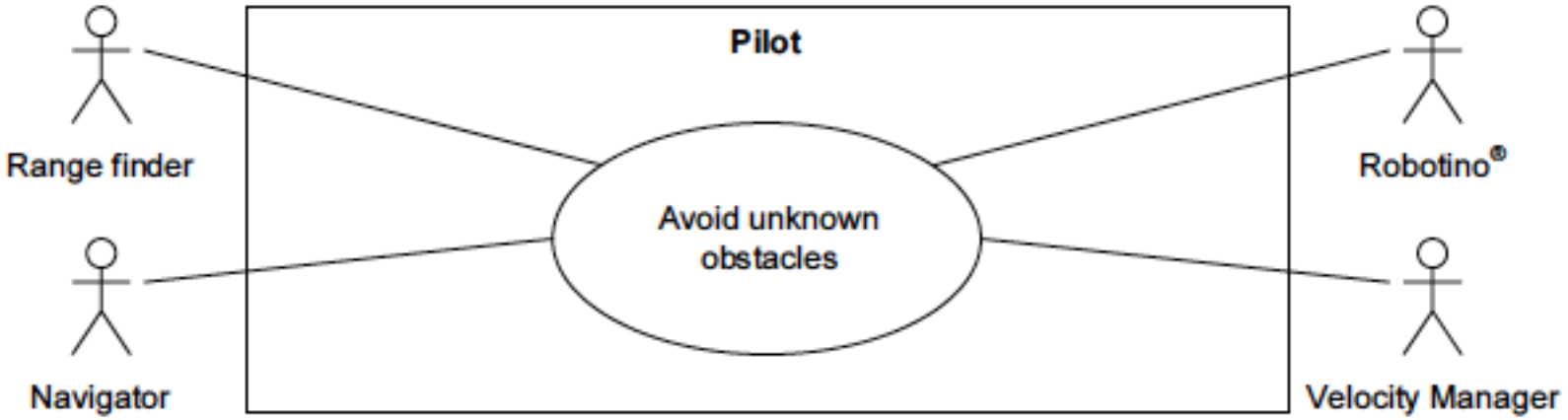


Based on: Simon Kracht and Carsten Nielsen (2007)
See report on
http://www.control.aau.dk/~tb/wiki/

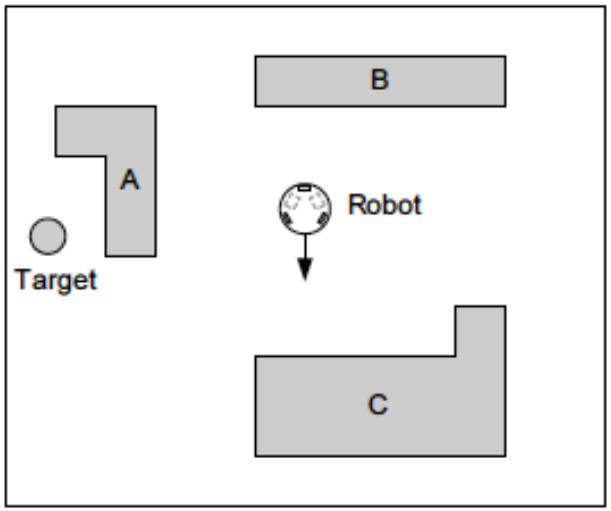


Adaptive Monte Carlo
Localization (AMCL), [Thrun et al., 2005].

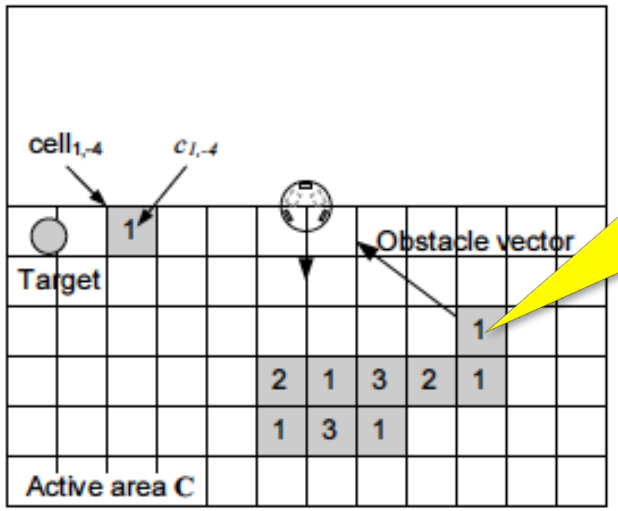
Obstacle avoidance



Vector Field Histogram+ (VFH+) algorithm



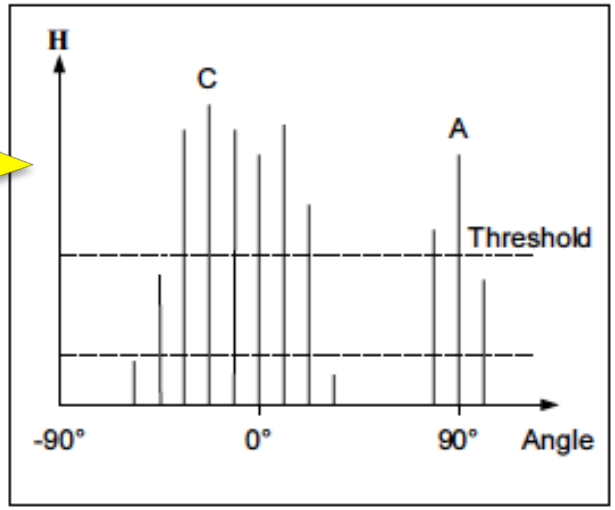
(a)



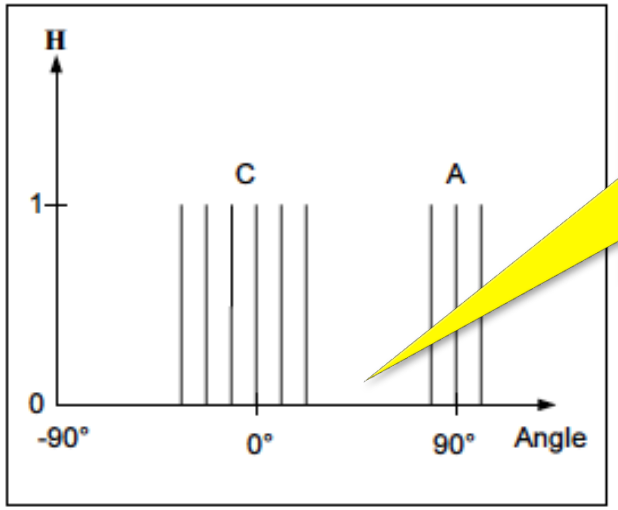
(b)

Certainty value, update a each detection

Polar Obstacle Density

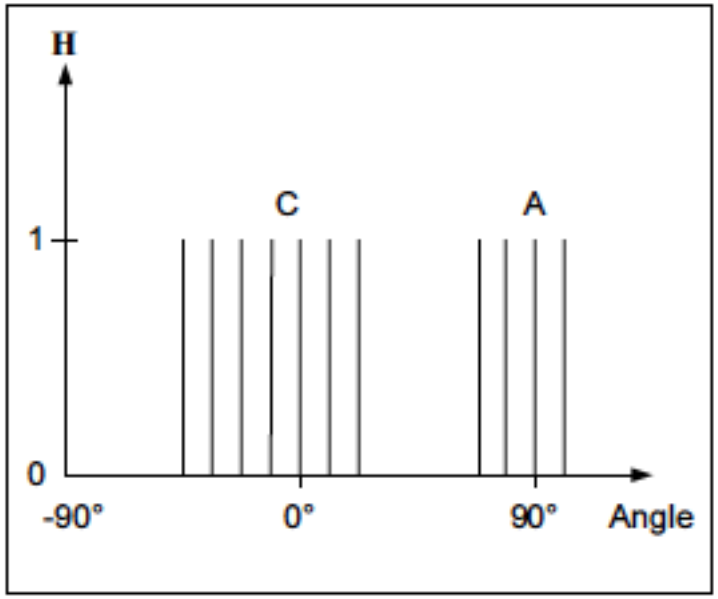


(c)

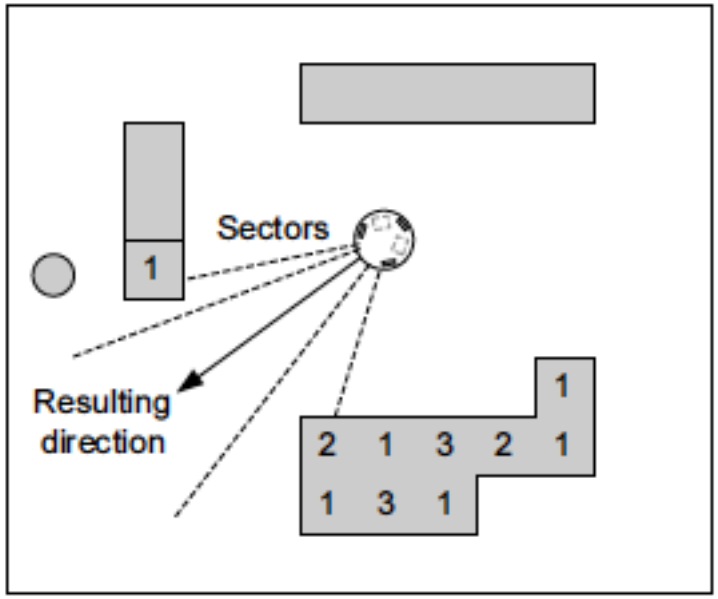


(d)

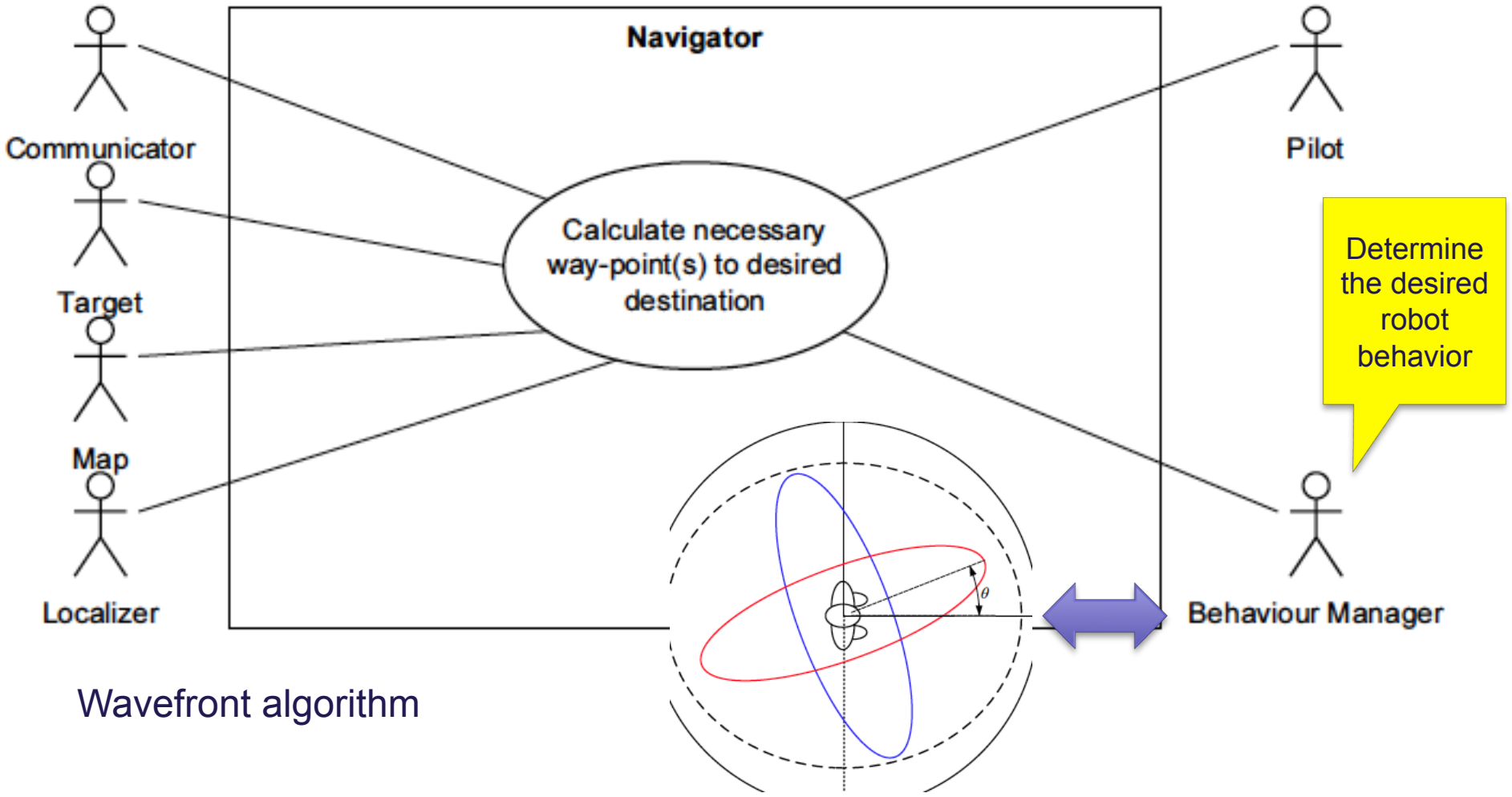
The valley, closest to the target direction


H
1
0
C
A
-90°
0°
90°
Angle

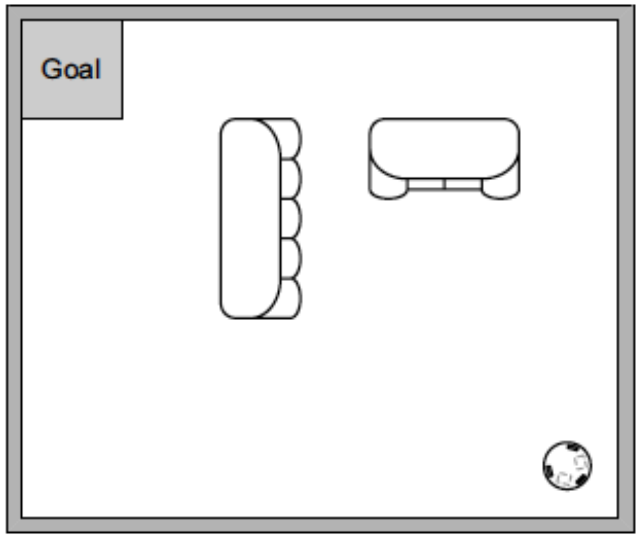
(e)


Sectors
1
Resulting
direction
1
2 1 3 2 1
1 3 1

(f)



Wavefront algorithm



(a)

2	0	0	0	0	0
0	0	1	1	1	0
0	0	1	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Occupancy grid

(b)

Incrementing – like wave from goal

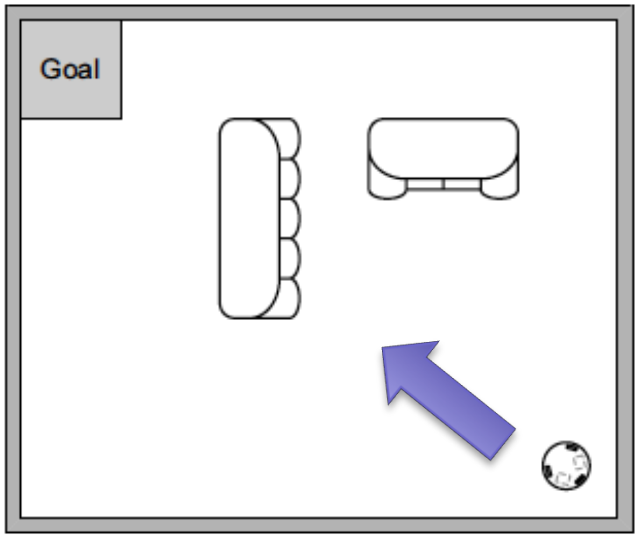
2	3	4	0	0	0
3	3	1	1	1	0
4	4	1	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

(c)

2 WP	3	4	5	6	7
3 WP	3	1	1	1	7
4	4	1	6	7	8
5	5	5	6	7	8
6	6	6	6 WP	7	8 WP

(d)

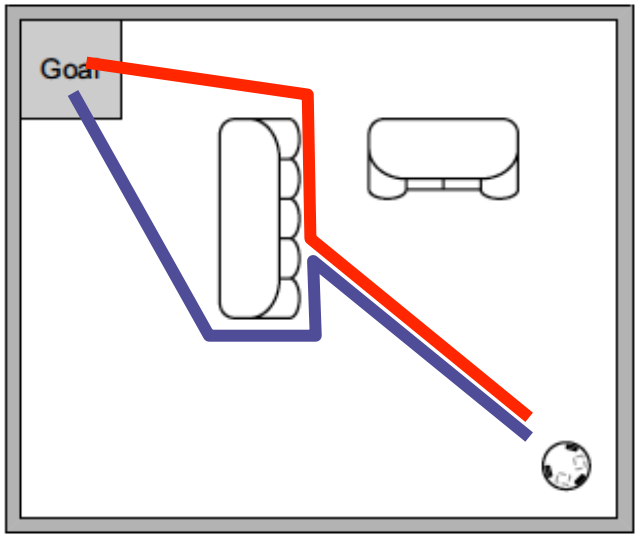
Alternative simple approach – the “bug” algorithm



(a)

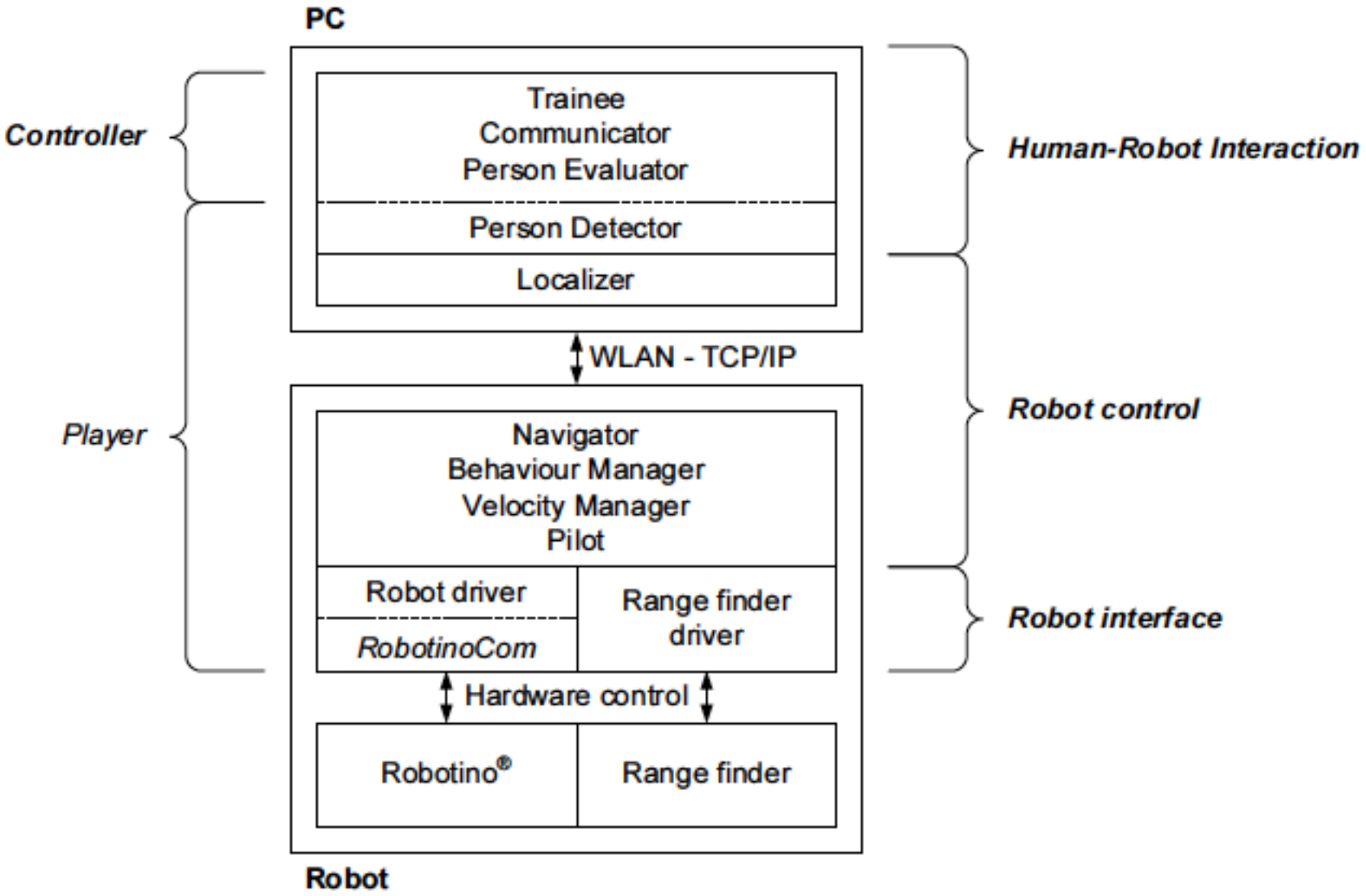
- Known direction to goal
- Only local sensing (walls/obstacle encoders)
- “a reasonable world”:
  - Finite obstacles in any finite range
  - A line will intersect an obstacle finite times

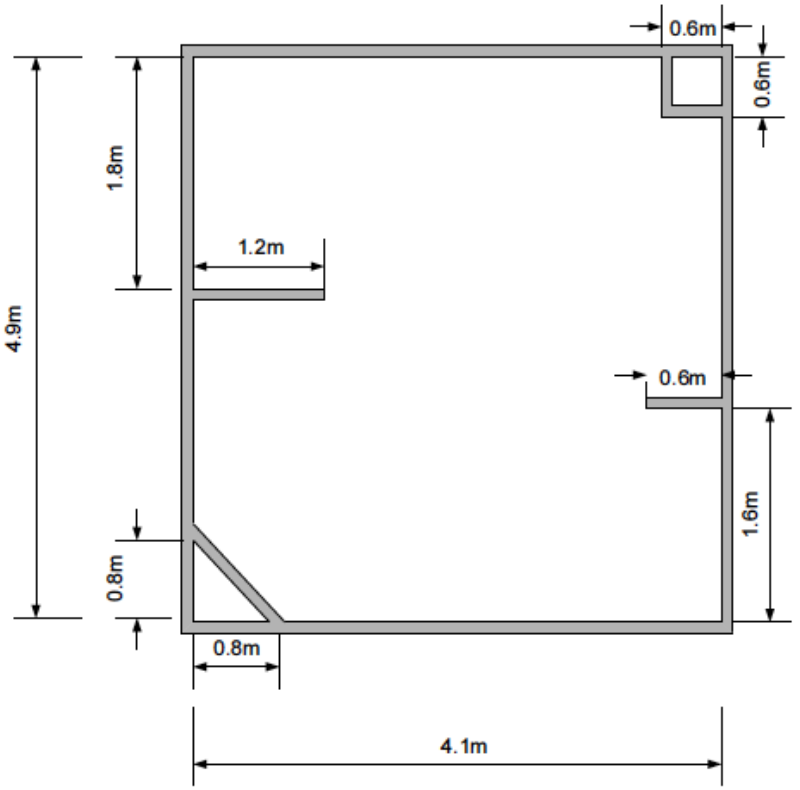
Alternative simple approach – the “bug” algorithm



(a)

- Head towards goal
- Follow obstacle until you can head towards goal again
- Continue


PC
Controller
Trainee
Communicator
Person Evaluator
Person Detector
Localizer
Human-Robot Interaction
WLAN - TCP/IP
Player
Navigator
Behaviour Manager
Velocity Manager
Pilot
Robot control
Robot driver
RobotinoCom
Range finder driver
Robot interface
Hardware control
Robotino®
Range finder
Robot


0.6m
0.6m
1.8m
1.2m
4.9m
0.6m
1.6m
0.8m
0.8m
4.1m

Planning methods

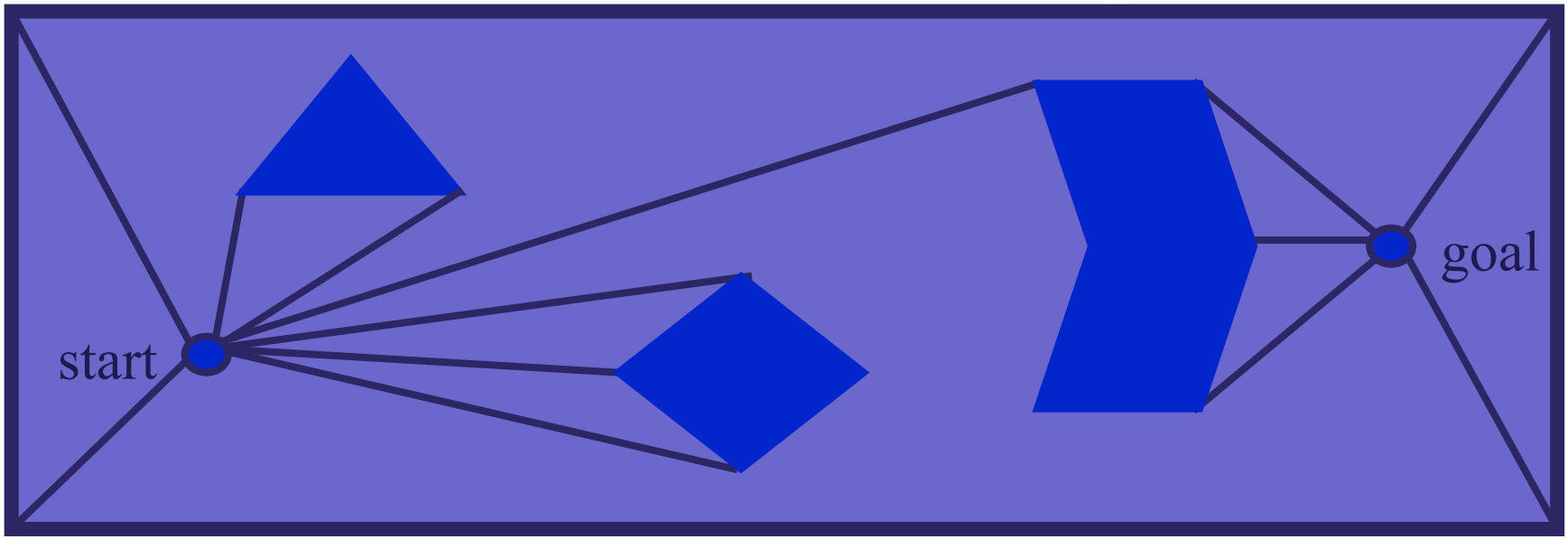
- Optimisation based approaches
- Grid based algorithms (**WAVEFRONT***)
- Graph based algorithms (**Visibility graph,** Voronoi diagramme)
- Sampling based algorithms
  - Rapidly-exploring Random Trees (**RRT**)

Converts the problem into a graph search. Dijkstra's algorithm O(N^2). N number of vertices in C-space

- Planning algorithms are often combined with local obstacle avoidance
  - Potential field methods
  - Vector Field Histogram (**VFH***)
  - Nearness Diagram (ND)
  - Dynamic Window Approach (DWA)
  - Velocity Obstacles (VO).

The Visibility Graph in Action (Part 1)

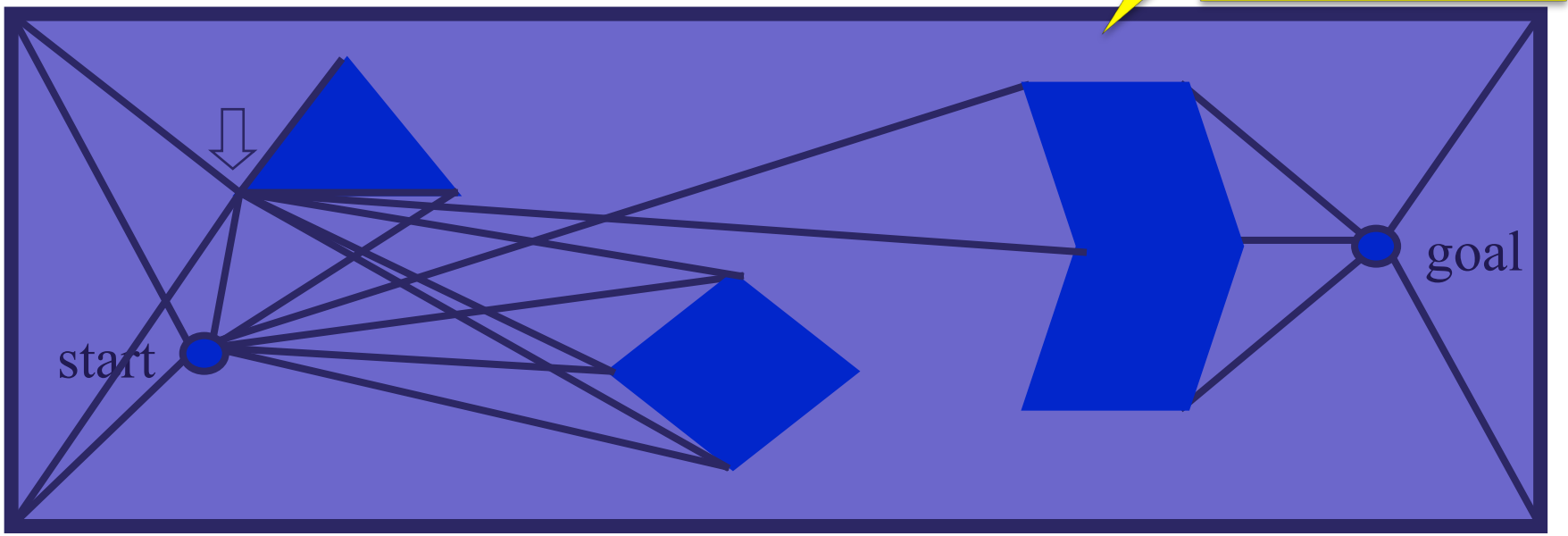
First, draw lines of sight from the start and goal to all "visible" vertices and corners of the world.



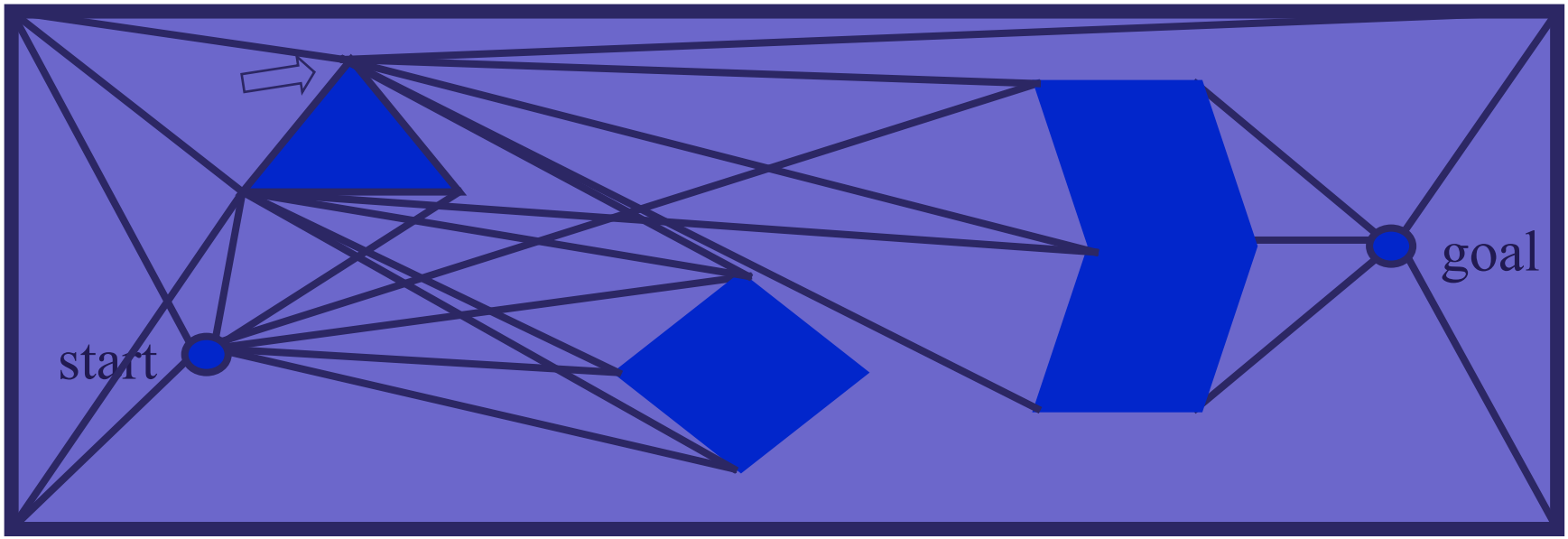
AALBORG UNIVERSITY
DENMARK

The Visibility Graph in Action (Part 2)

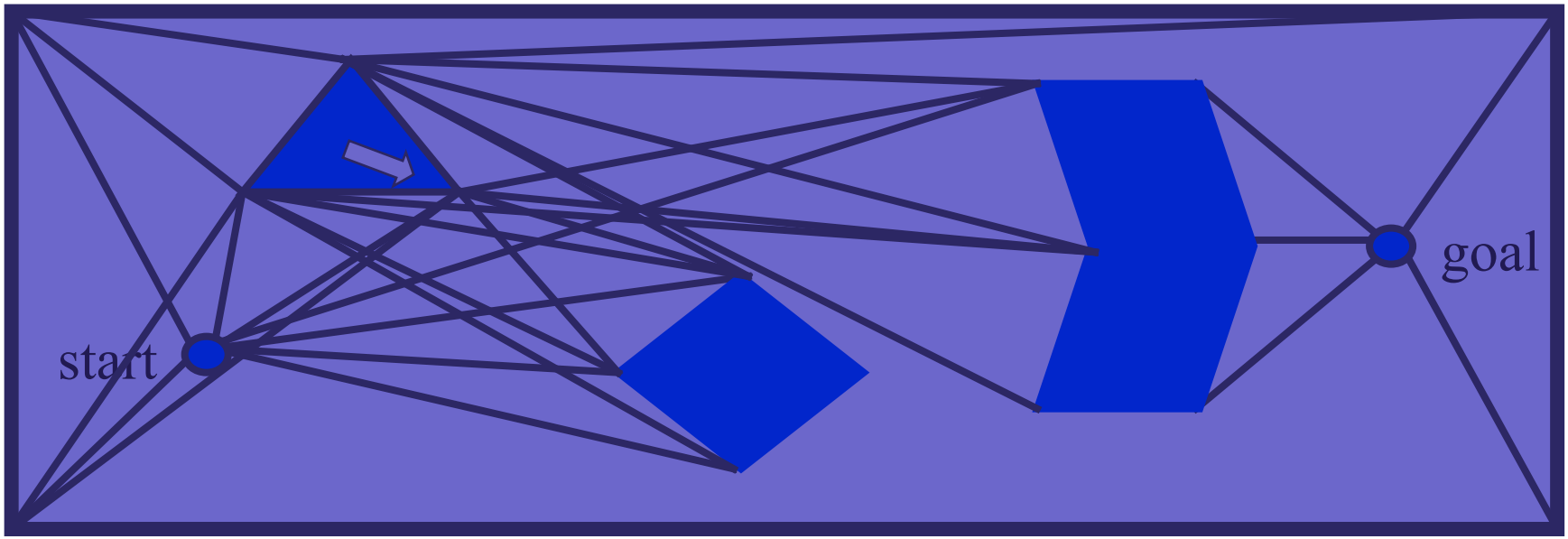
Second, draw lines of sight from every vertex of every obstacle like before. Remember lines along edges are also lines of sight.



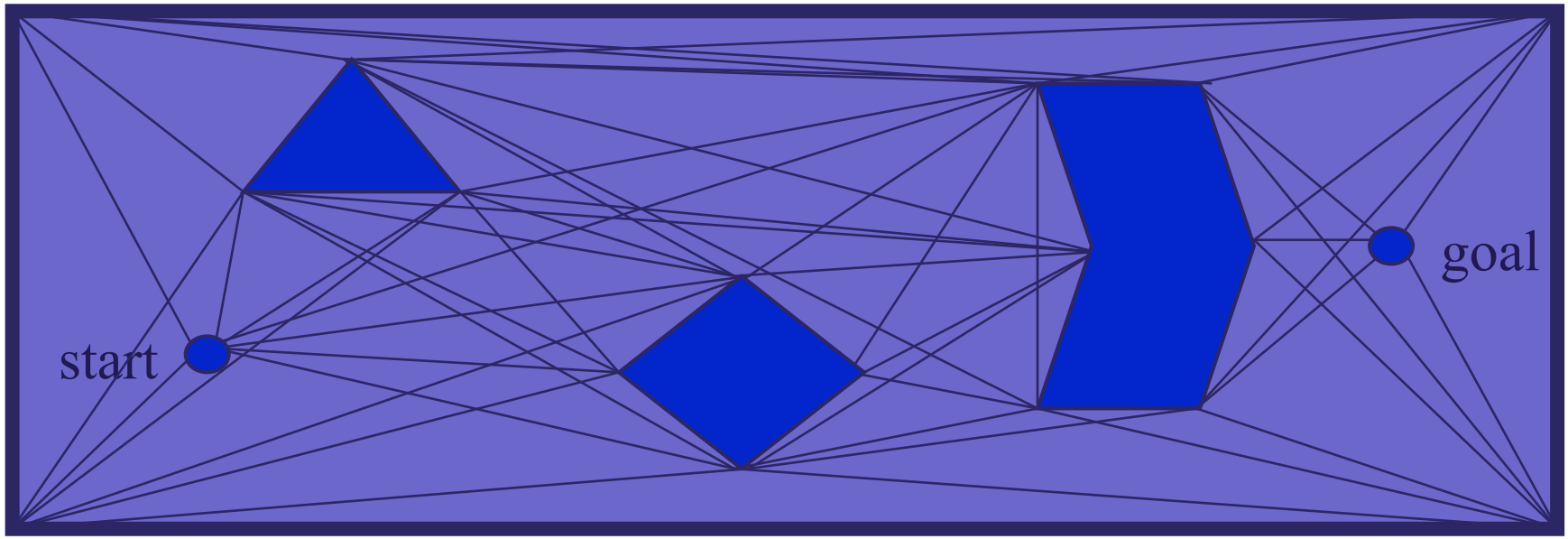
The Visibility Graph in Action (Part 3)



The Visibility Graph in Action (Part 4)

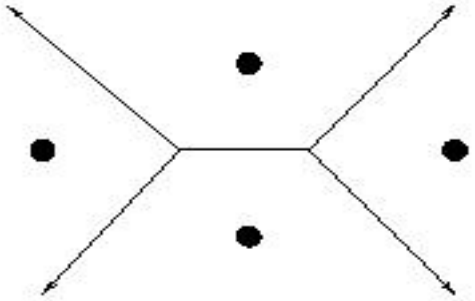


The Visibility Graph (Done)



Since the map was in C-space, each line potentially represents part of a path from the start to the goal.

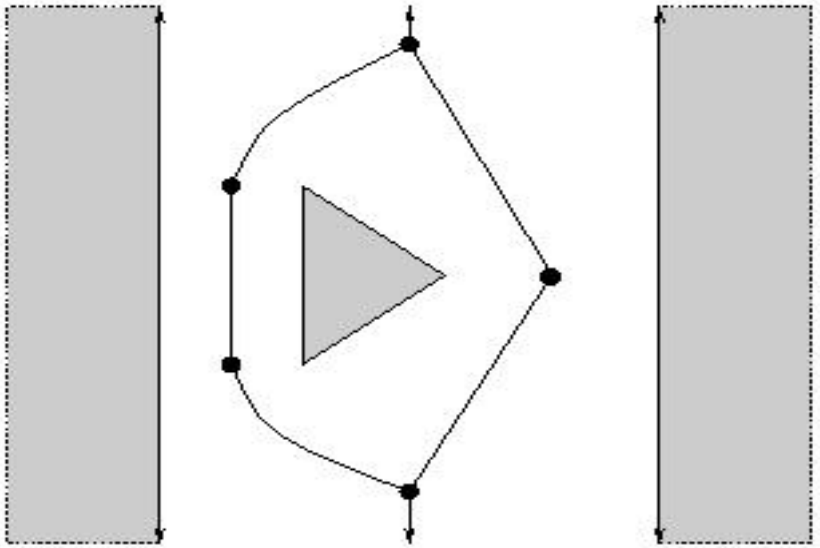
Roadmap: Voronoi diagrams



"official" Voronoi diagram

(line segments make up the **Voronoi diagram** isolates a set of points)

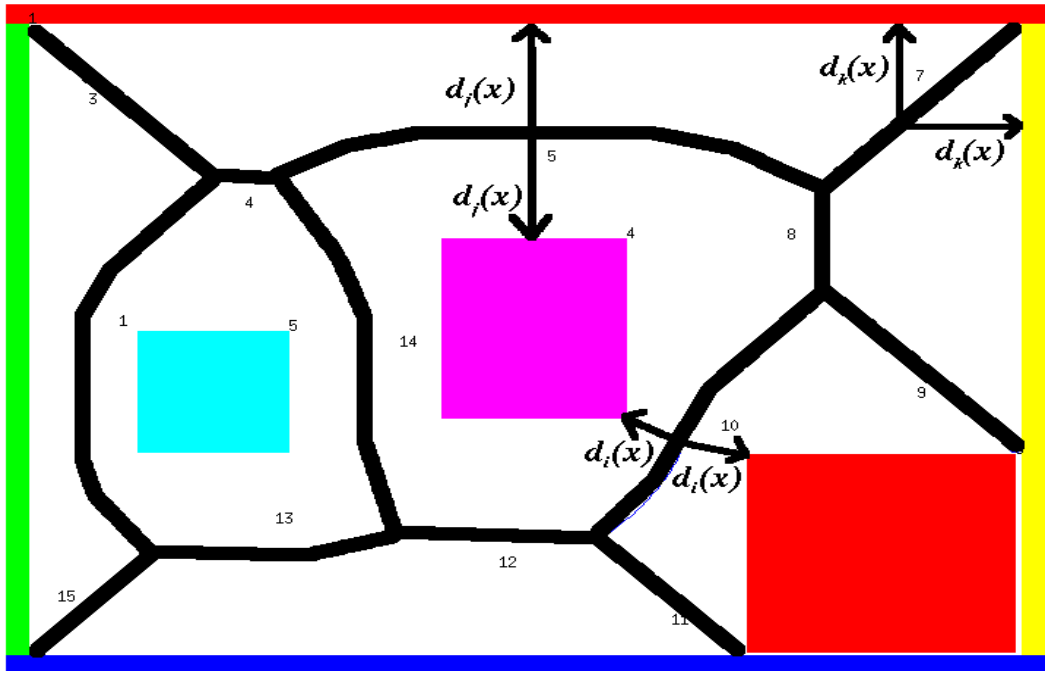
Property: maximizing the clearance between the points and obstacles.



Generalized Voronoi Graph (GVG): locus of points equidistant from the closest two or more obstacle boundaries, including the workspace boundary.

Roadmap: Voronoi diagrams

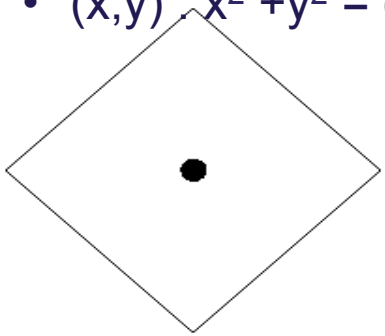
- GVG is formed by paths equidistant from the two closest objects
- maximizing the clearance between the obstacles.
- This generates a very safe roadmap



AALBORG UNIVERSITY
DENMARK

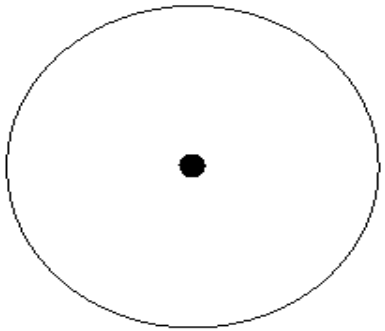
Voronoi Diagram: Metrics

- Many ways to measure distance; two are:
  - L1 metric
    - (x,y) : |x| + |y| = const
  - L2 metric
    - (x,y) : $x^2 + y^2$ = const



$\{(x,y) : |x| + |y| = \text{const}\}$

L1

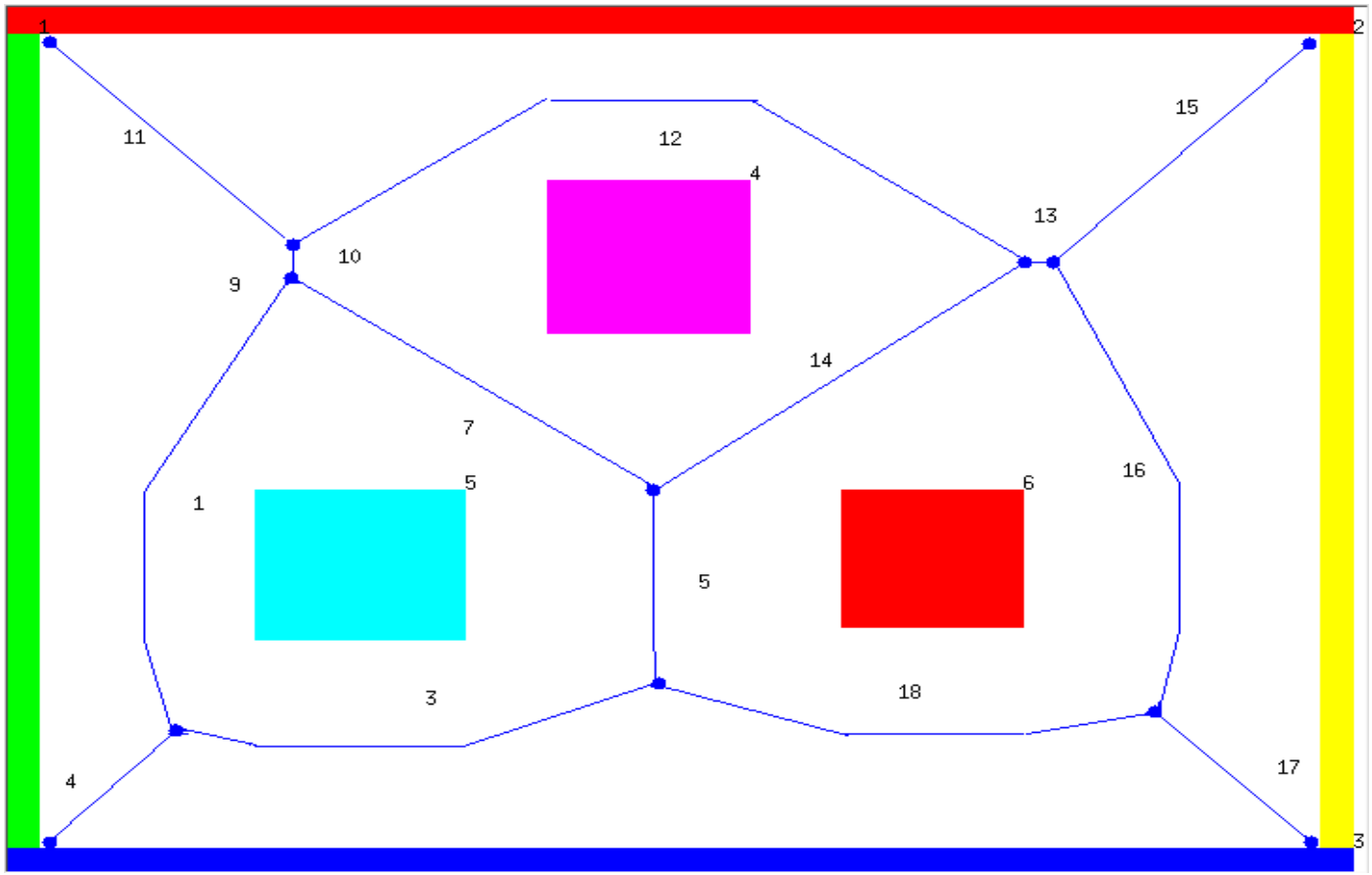


$\{(x,y) : x^2 + y^2 = \text{const}\}$

L2

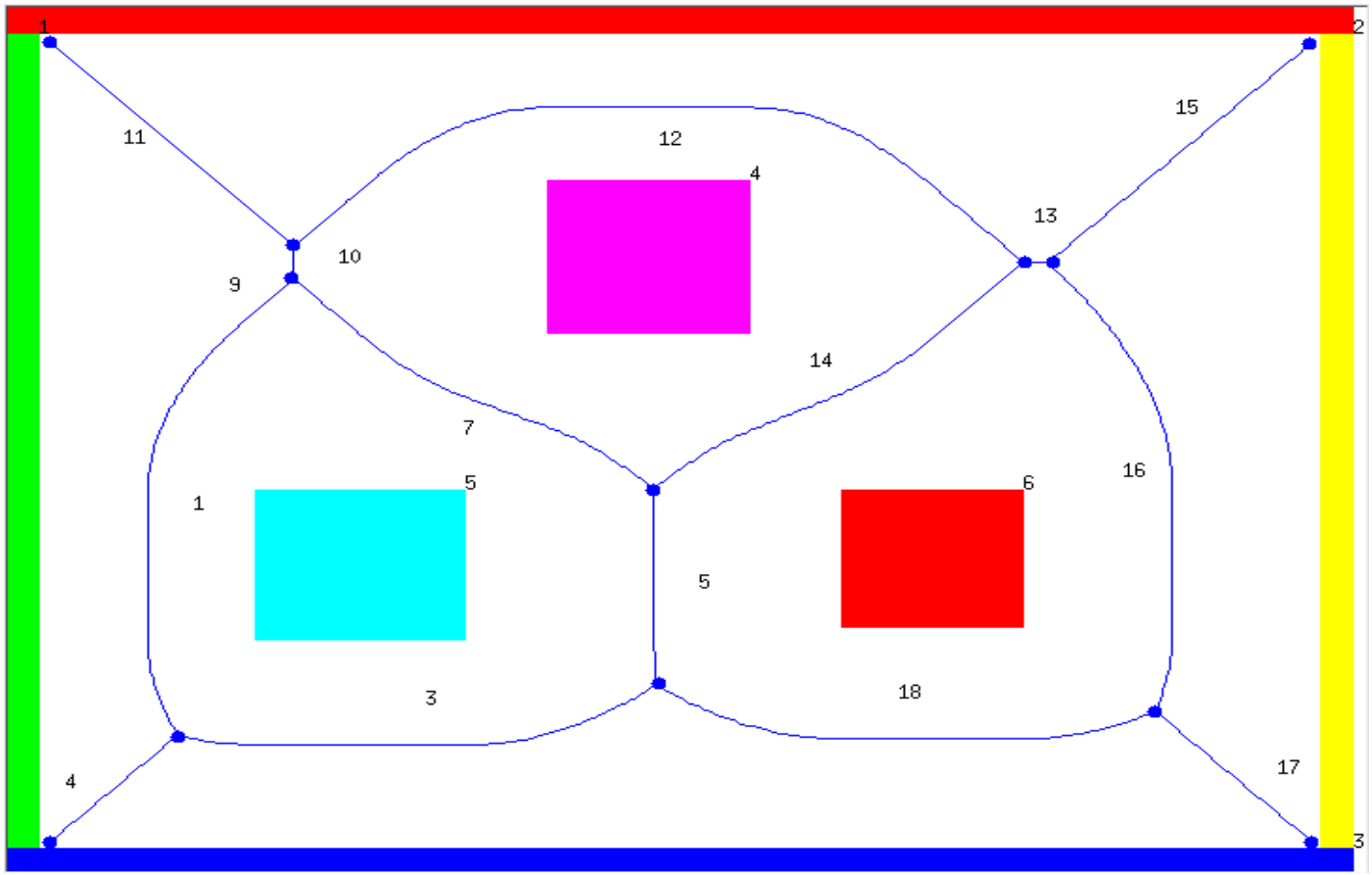
Voronoi Diagram (L1)

Note the lack of curved edges



Voronoi Diagram (L2)

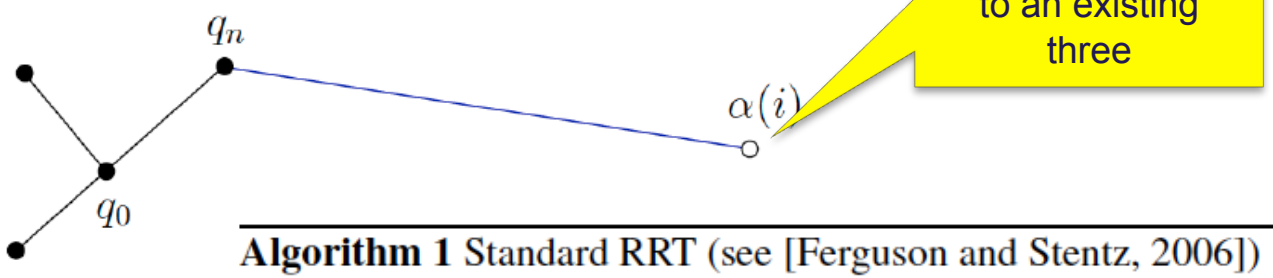
Note the curved edges



Rapidly-exploring Random Trees (RRT)

- Exploits the increasing computational power to test a lot of possible trajectories or configurations, instead of searching for the one optimal solution
- Not optimal, but probabilistic complete - the probability of finding a solution, (if one exists) goes to 1 as the number of samples goes towards infinity.
- A survey over sampling based motion planning and current issues can be found in [Lindemann and LaValle, 2005].
- RRT originally described in [LaValle and Kuffner Jr, 2001].

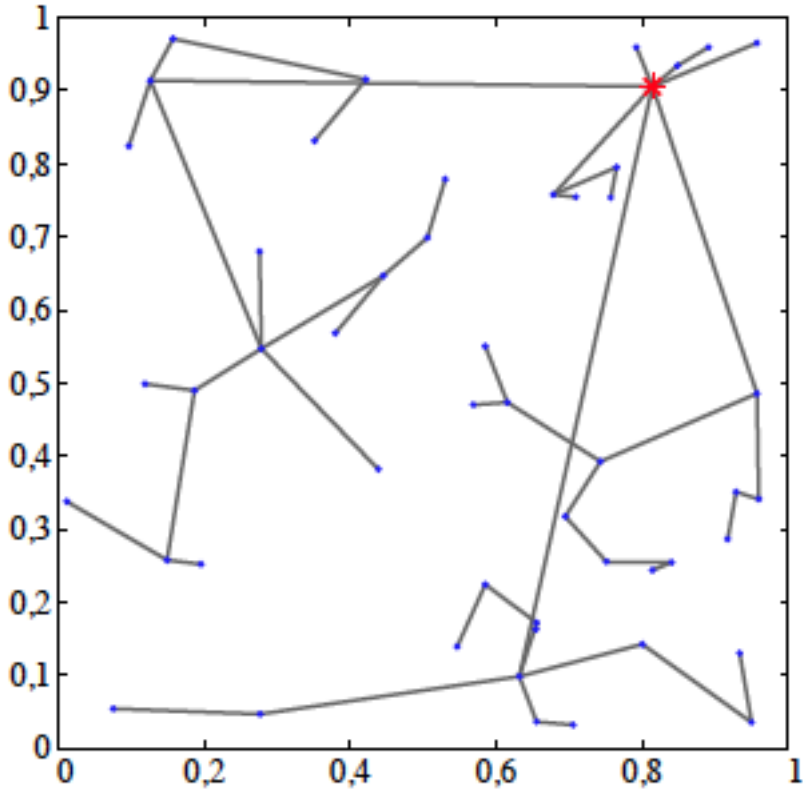
RRT



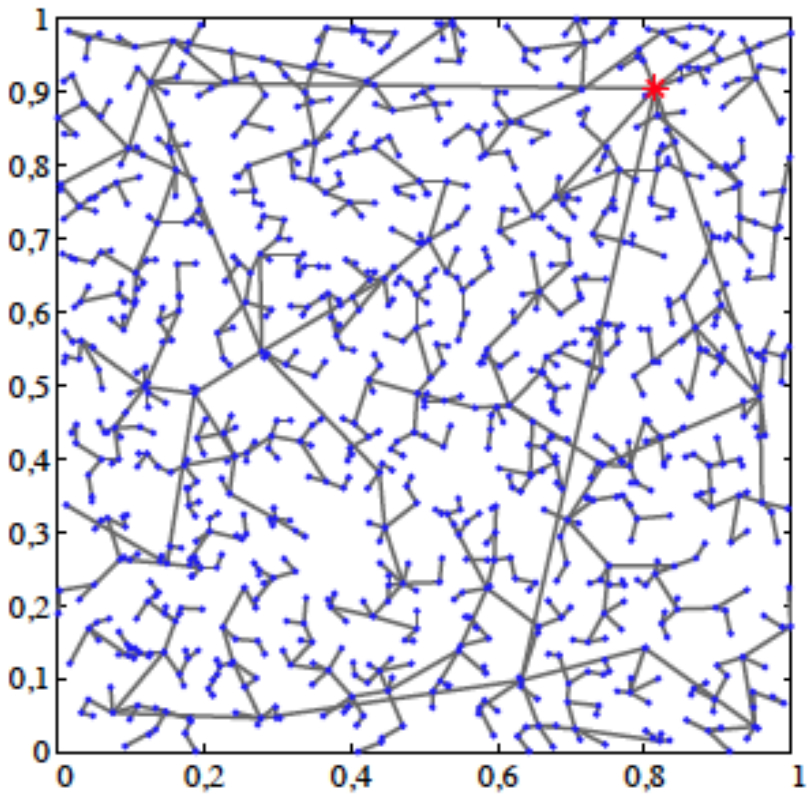
Algorithm 1 Standard RRT (see [Ferguson and Stentz, 2006])

RRTmain()

```
1: Tree = q.start
2: q.new = q.start
3: while Distance(q.new , q.goal) < ErrTolerance do
4:     q.target = SampleRandomTarget()
5:     q.nearest = NearestVertex(Tree , q.target)
6:     q.new = ExtendTowards(q.nearest,q.target)
7:     Tree.add(q.new)
8: end while
9: return  Trajectory(Tree,q.new)
```

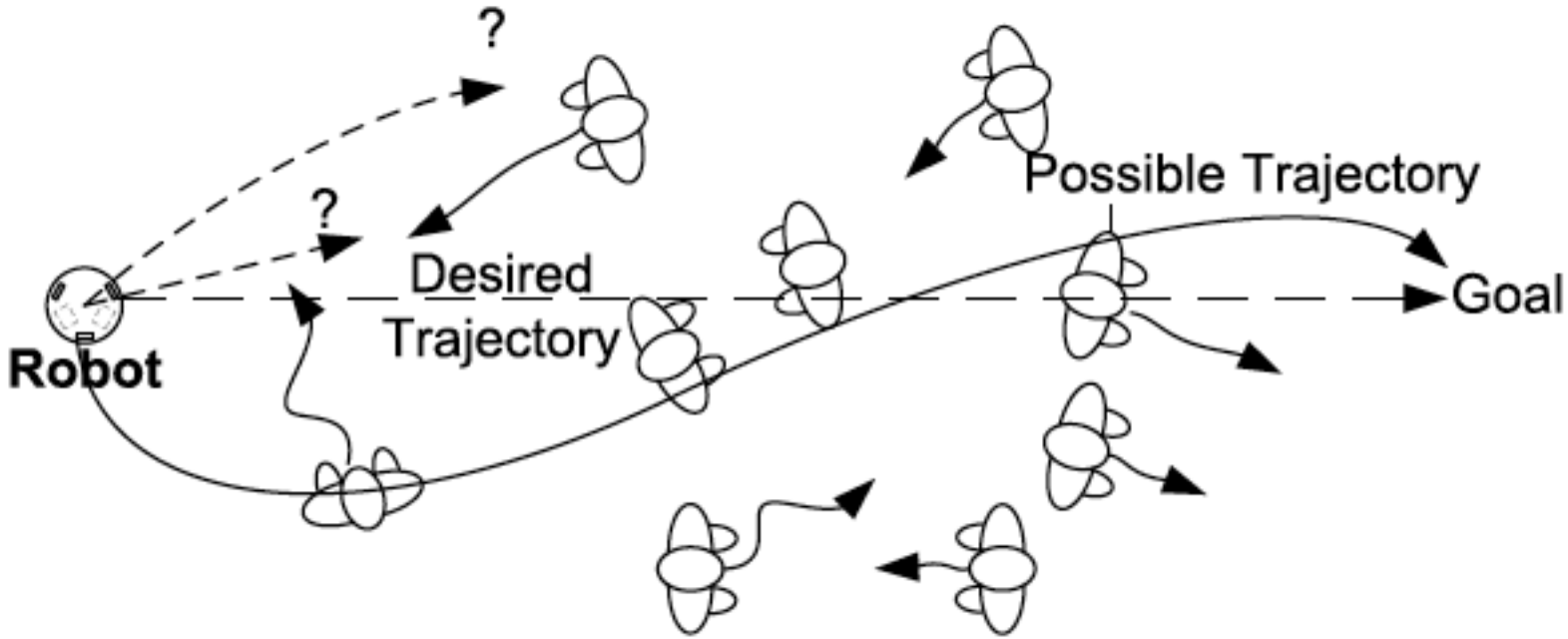


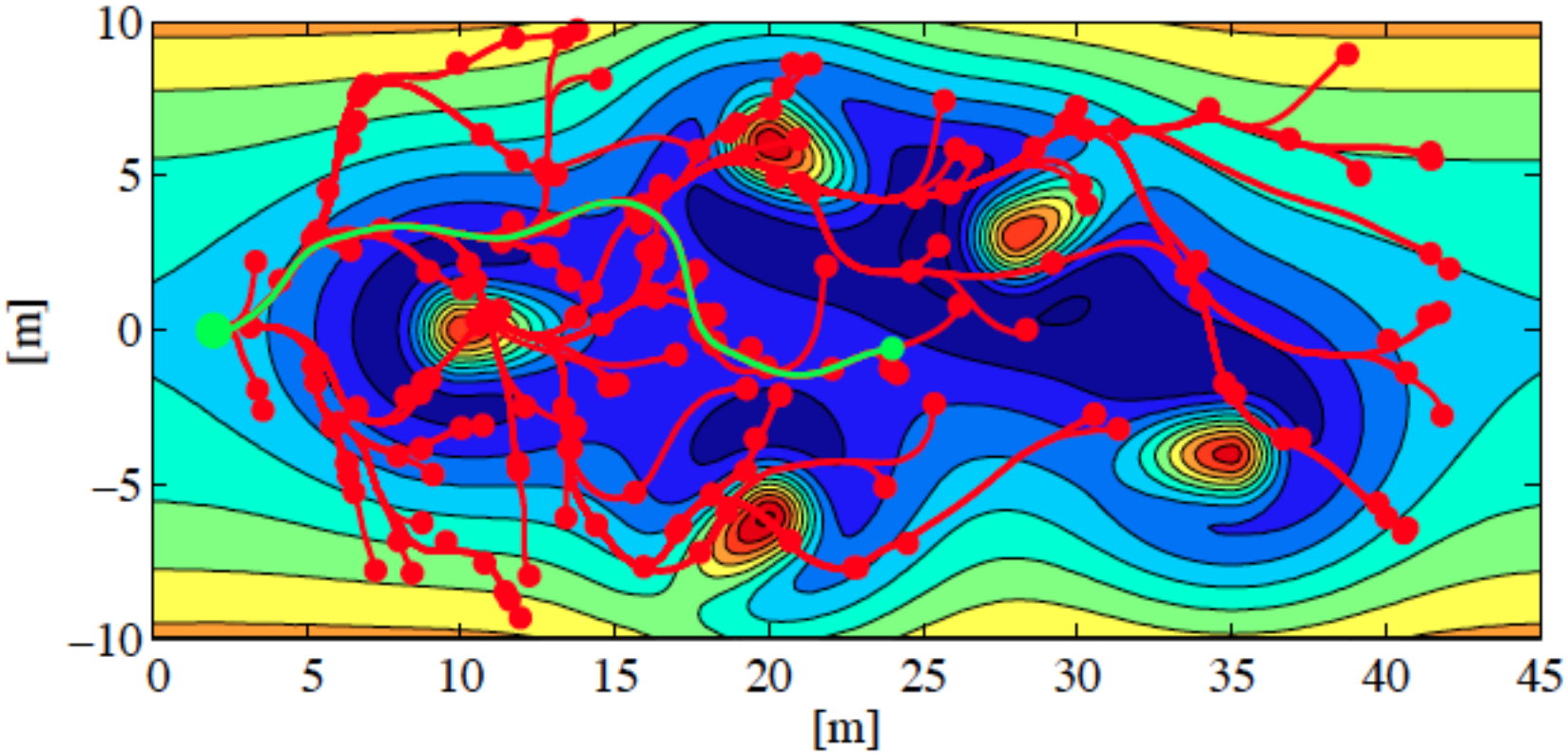
(a) A simple RRT with 50 vertices.



AALBORG UNIVERSITY
DENMARK

Trajectory Planning in Dynamic Human Environments





AALBORG UNIVERSITY
DENMARK

Thank you - http://www.control.aau.dk/~tb/wiki/

- Contributors:
  - Simon Kracht
  - Carsten Nielsen
  - Hans Jørgen Andersen
  - Mikael Svenstrup
  - Francesca Dedola
  - Søren T Hansen
  - André Ribeiro da Silva McGuire
  - Bernardo Sousa Machado Henriques
  - Hoang Chuong Ngo Nguyen
  - Kasper Fuglsang Jensen
  - Kasper Vinther
  - René Jespersen

AALBORG UNIVERSITY
DENMARK