Search, Navigate, and Actuate

Quantative Navigation

University of Amsterdam

Path Planning

- *What's the Best Way There?* depends on the representation of the world
- A robot's world representation and how it is maintained over time is its *spatial memory*
- Two forms
  - *Route (or qualitative / topology)*
  - *Map (or quantitative / metric)*
- Map leads to Route, but not the other way

University of Amsterdam

Spatial memory

- World representation used by robot
- Provides methods and data structures for processing and storing information derived from sensors
- Organized to support methods that extract relevant expectations about a navigational task

University of Amsterdam

Basic functions of spatial memory

- Attention:
  What features, landmarks to look for next?
- Reasoning:
  E.g., can I fit through that door?
- Path Planning:
  What is the best way through this building?
- Recognition:
  What does this place look like? Have I ever seen it before? What has changed since I was here before?

Quantitative spatial memory

- Express space in terms of physical distances of travel
- Bird’s eye view of the world
- Not dependent upon the perspective of the robot
- Independent of orientation and position of robot
- Can be used to generate qualitative (route) representations

University of Amsterdam

Thinning

- Amsterdam Oxford Joint Rescue Foreces

University of Amsterdam

Metric Maps

- Motivation for having a metric map is often *path planning* (others include reasoning about space…)

- Determine a path from one point to goal
  - Generally interested in “best” or “optimal”
  - What are measures of best/optimal?
  - Relevant: occupied or empty

- Path planning assumes an *a priori* map of relevant aspects
  - Only as good as last time map was updated

University of Amsterdam

Metric Path Planning

- Objective: determine a path to a specified goal
- Metric methods:
  - Tend to favor techniques that produce an optimal path
  - Usually decompose path into subgoals called waypoints

University of Amsterdam

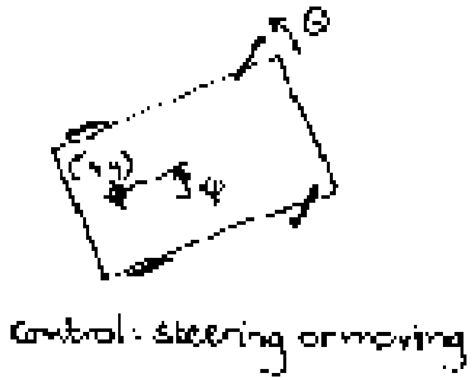
Configuration Space

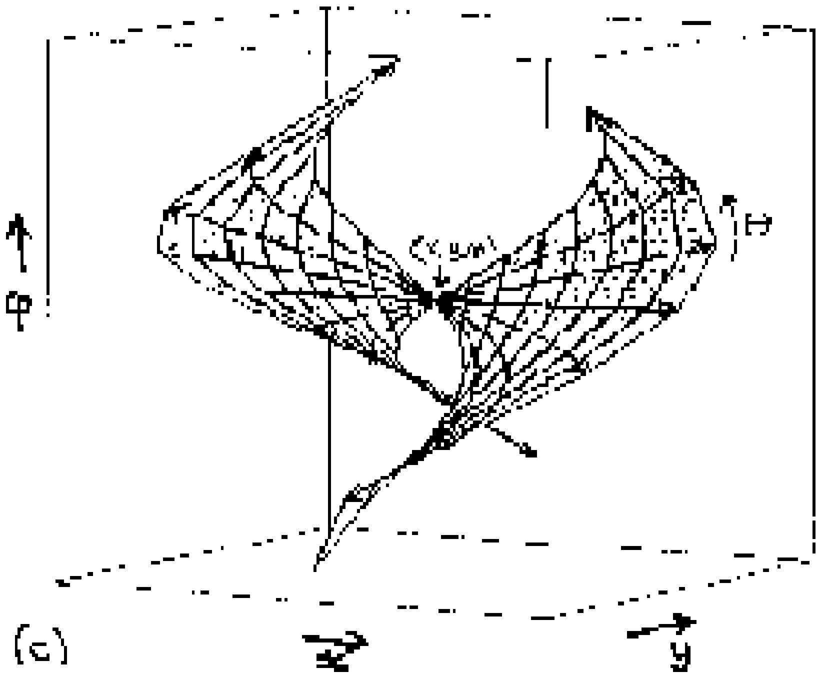
- Data structure that allows robot to specify position and orientation of objects and robot in the environment
- Reduces # of dimensions that a planner has to deal with
- Typically, for indoor mobile robots:
  - Assume 2 DOF for representation
  - Assume robot is round, so that orientation doesn't matter
  - Assumes robot is holonomic (i.e., it can turn in place)
  - Classify "occupied" and "free" space

University of Amsterdam

Non-holonomic mobile robot

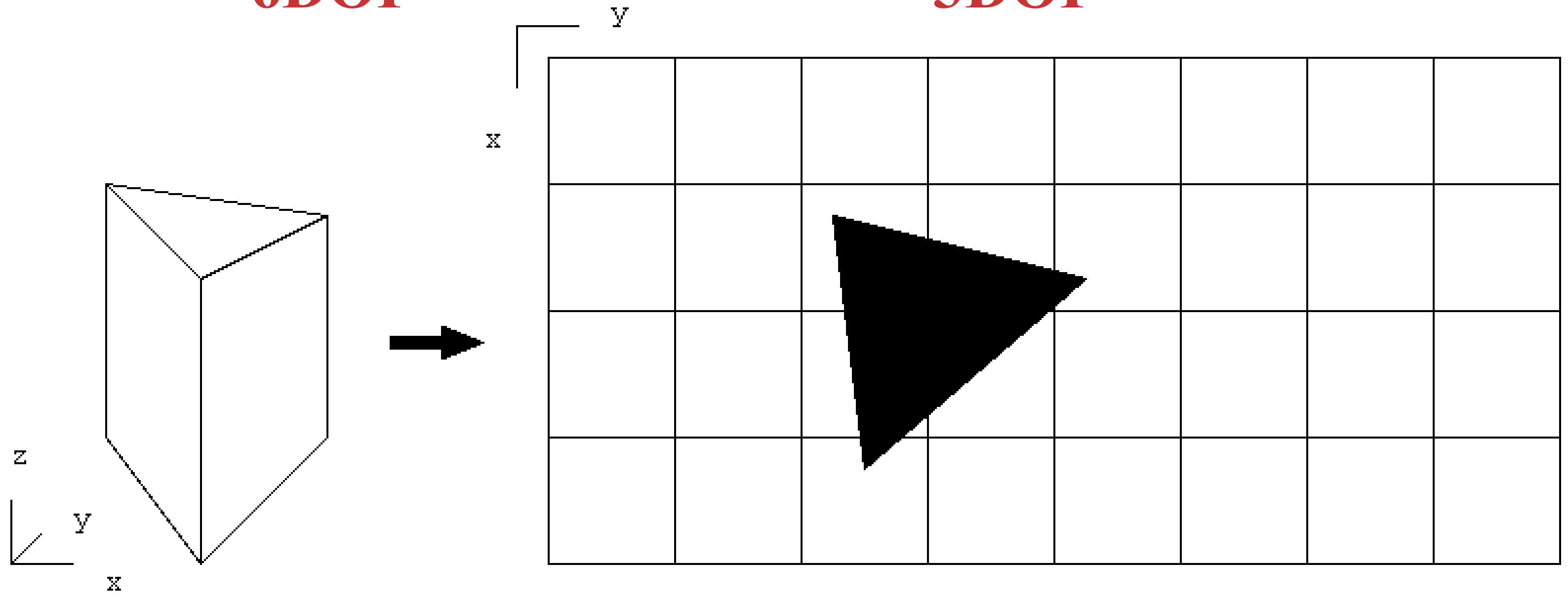
- 2D-surface in 3D configuration space





Reduction and Discretization

6DOF **3DOF**



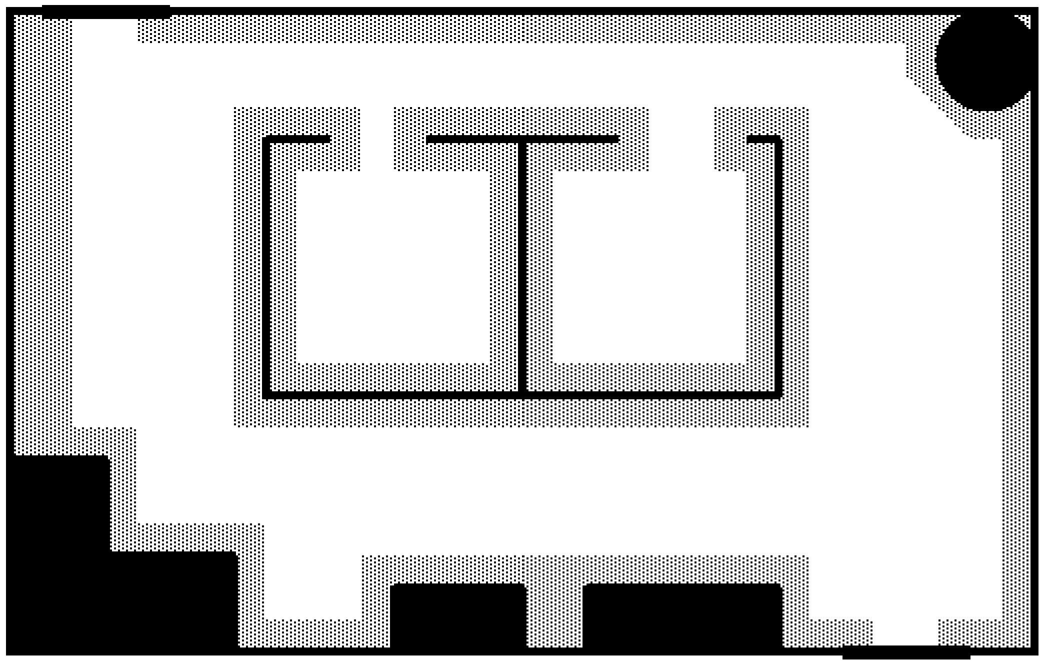
University of Amsterdam

Object Growing

- Since we assume robot is round, we can “grow” objects by the width of the robot and then consider the robot to be a point
- Greatly simplifies path planning
- New representation of objects typically called “forbidden region”

University of Amsterdam

Dilation of geometrical objects

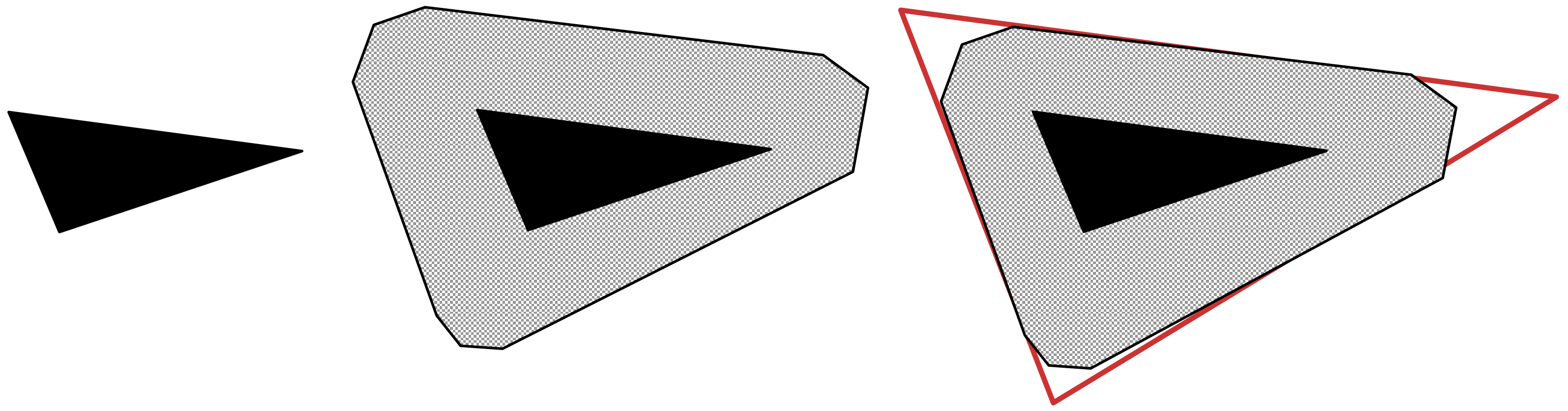


- Combination of straight walls and a round robot

University of Amsterdam

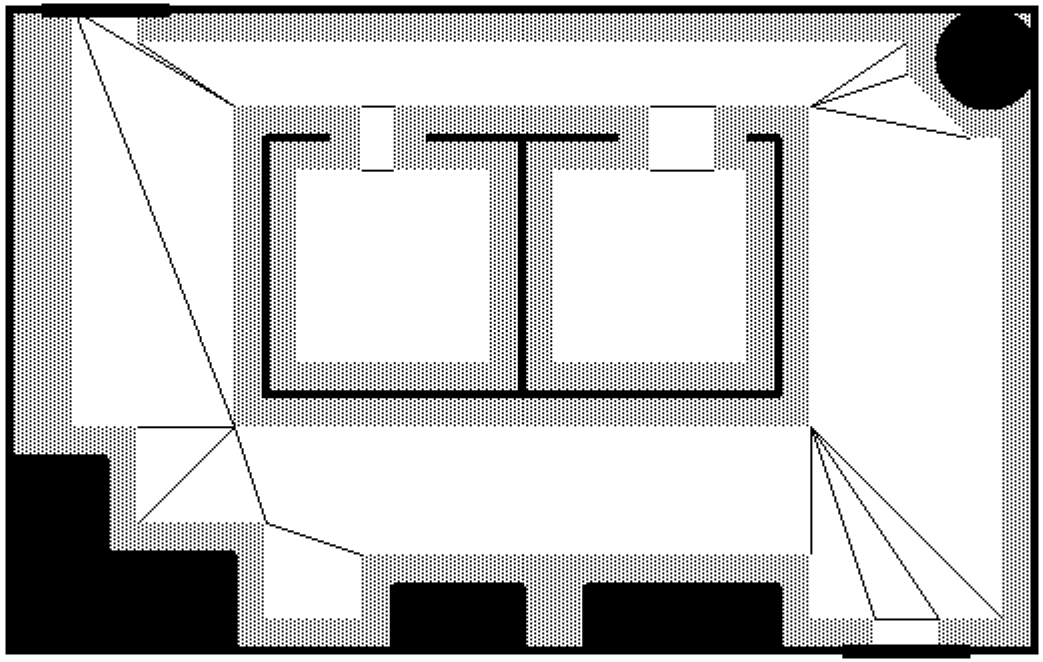
Minkowski sum

Take the union of the current (bit)map with a set of copies of a mask (circle for round robot) placed on the boundaries (edges) of the current shape.



Object dilation can make an environment description much more complex by making it smooth!

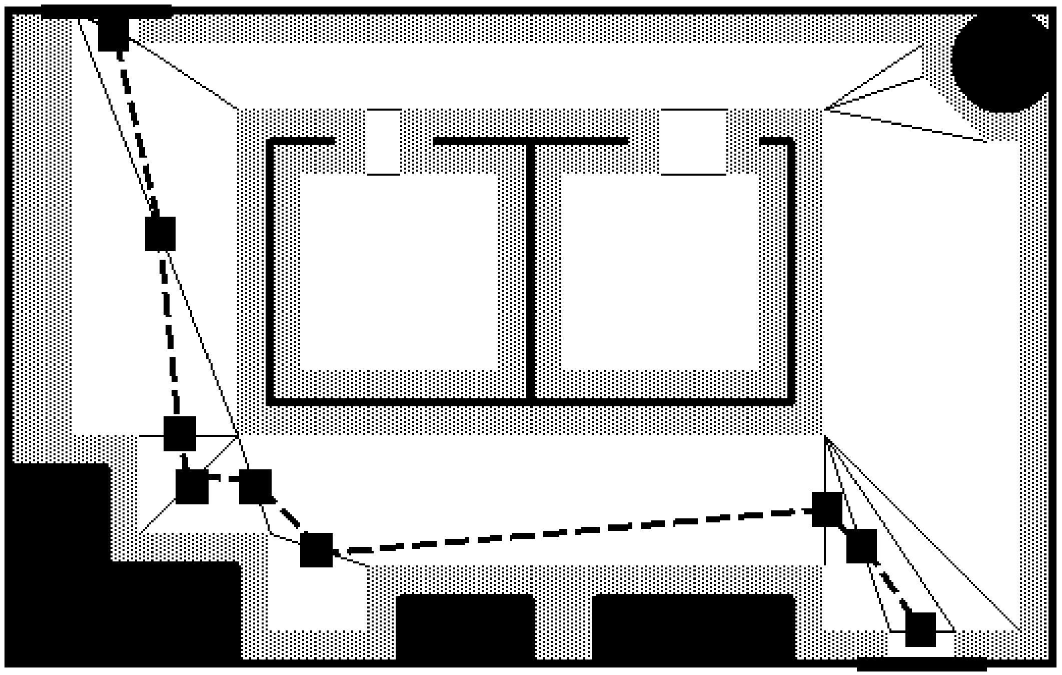
Partitioning of free space



Decompose *C*-space into smaller (polygonal) cells

University of Amsterdam

Meadow map



A *connectivity graph* can be made cells, and a save path can be generated by travel the midpoint of the edges

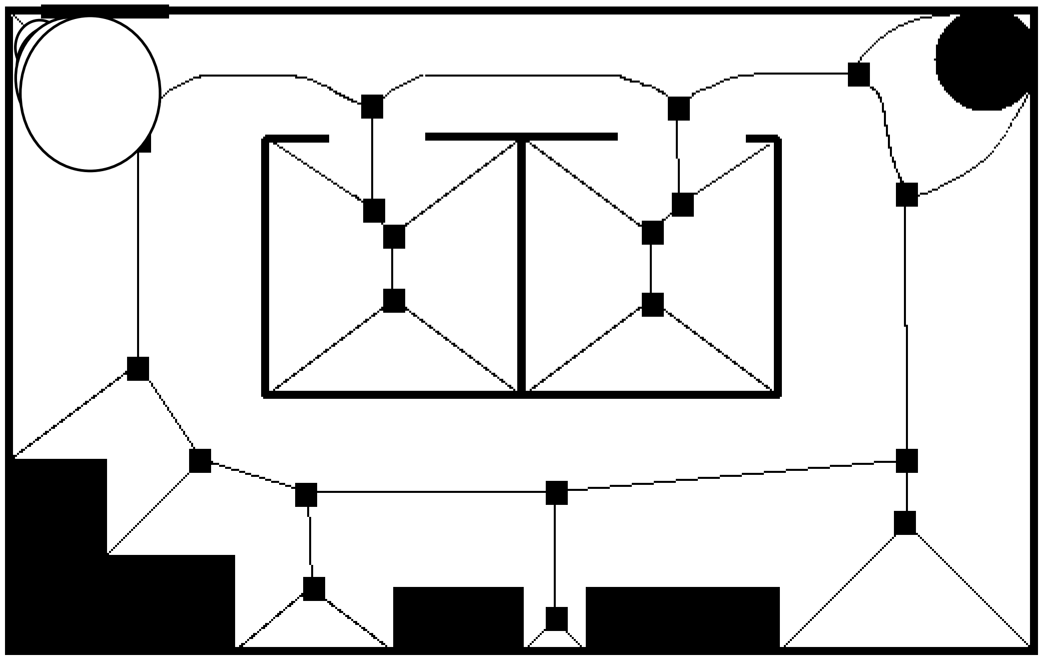
Problems with Meadow Maps

- Not unique generation of polygons
- Could you actually create this type of map with sensor data, instead of *a-priori maps*?
- Uses artifacts like edges and corners of the map to determine polygon boundaries, rather than things that can be sensed
- In principle, one likes to decompose in large and stable cells

University of Amsterdam

Voronoi diagram

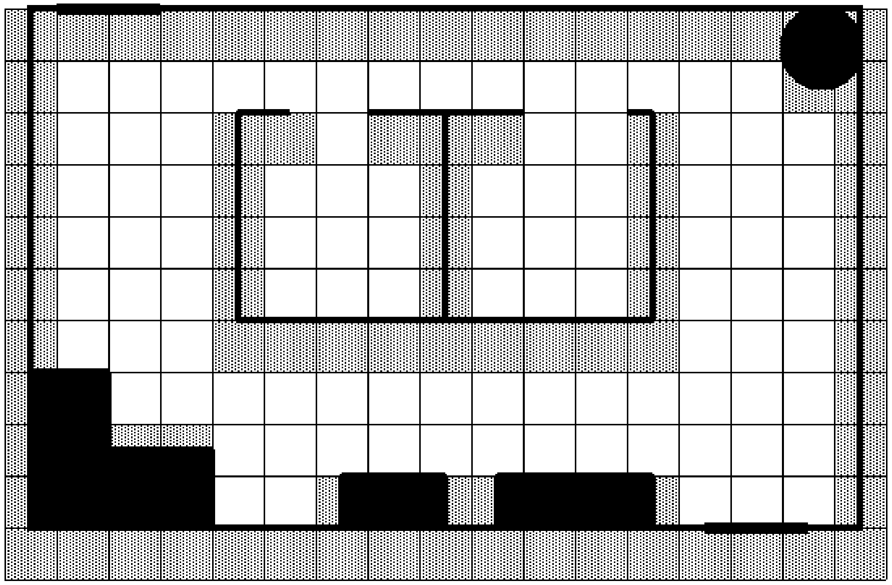
A *Voronoi diagram* is generated by a *retract*, leading to a (n-1)-dimensional hypersurface



The paths have *maximal clearance*, and are not *optimal*

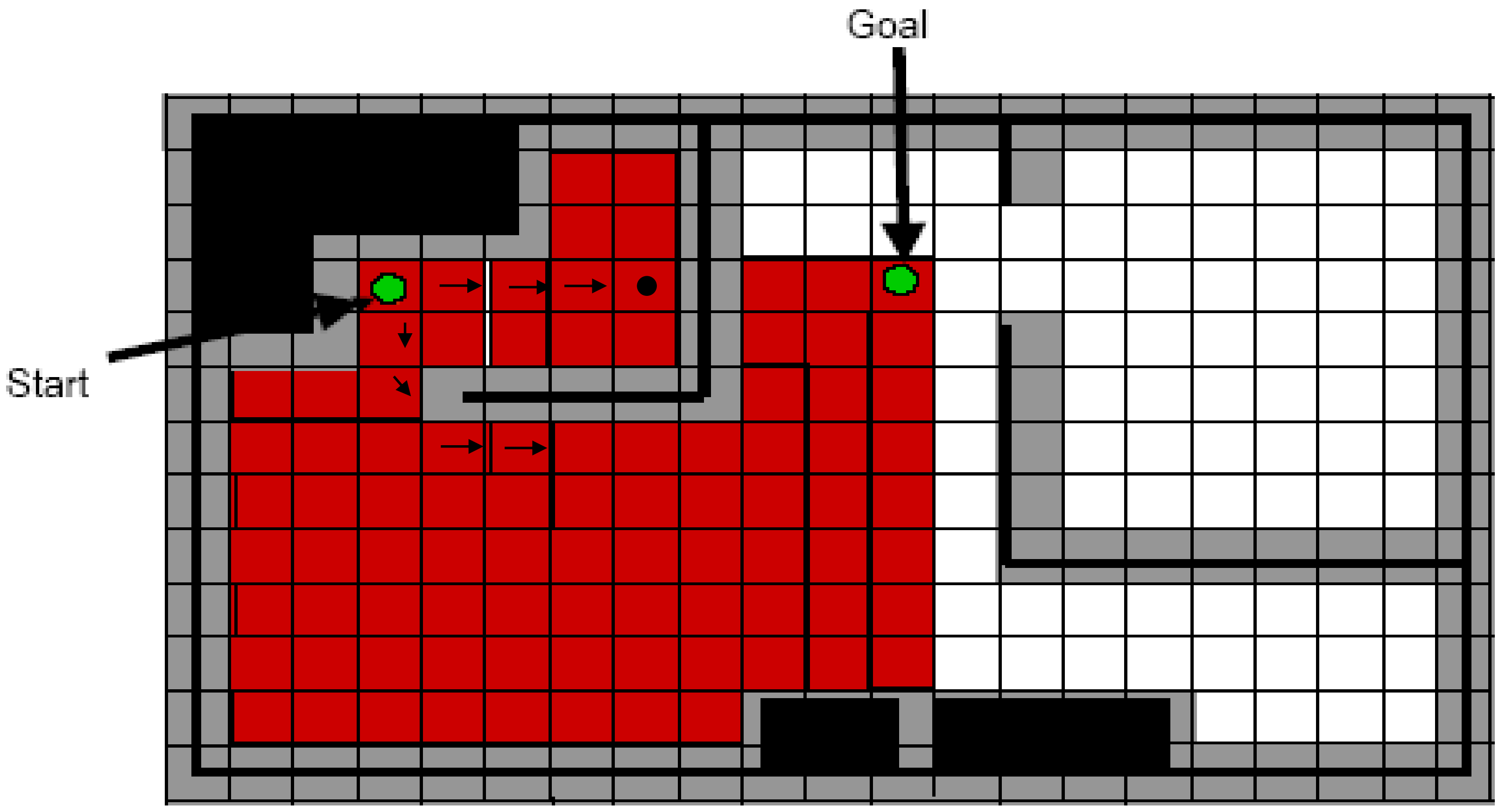
Uniform cells

In bounded low dimensional C-space, one can overlay the map with a grid (*sampling*).



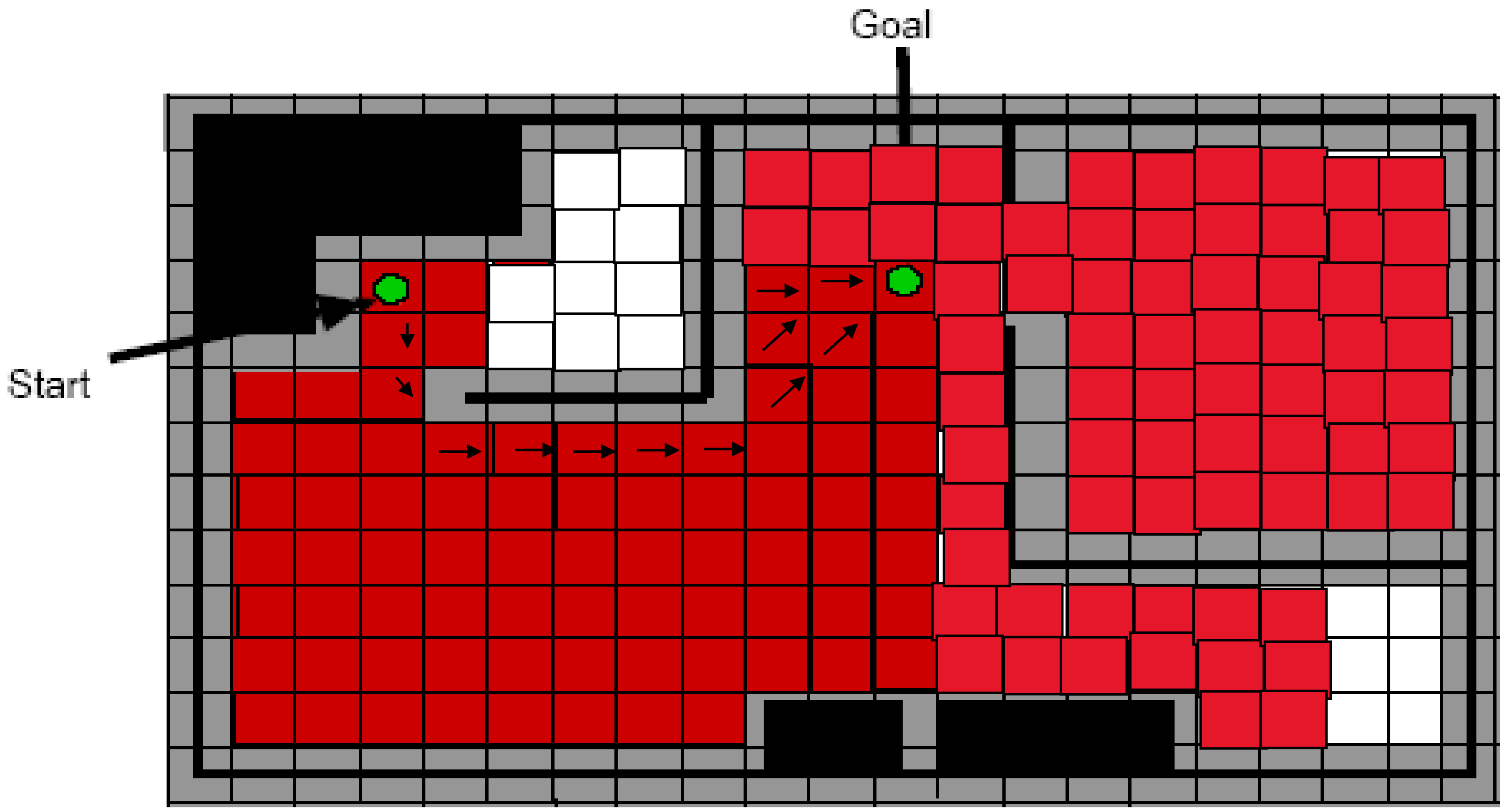
- Make a graph by each seeing each *coxel* as a node, connecting neighbors (4-connected, 8-connected)

Path Search algorithms



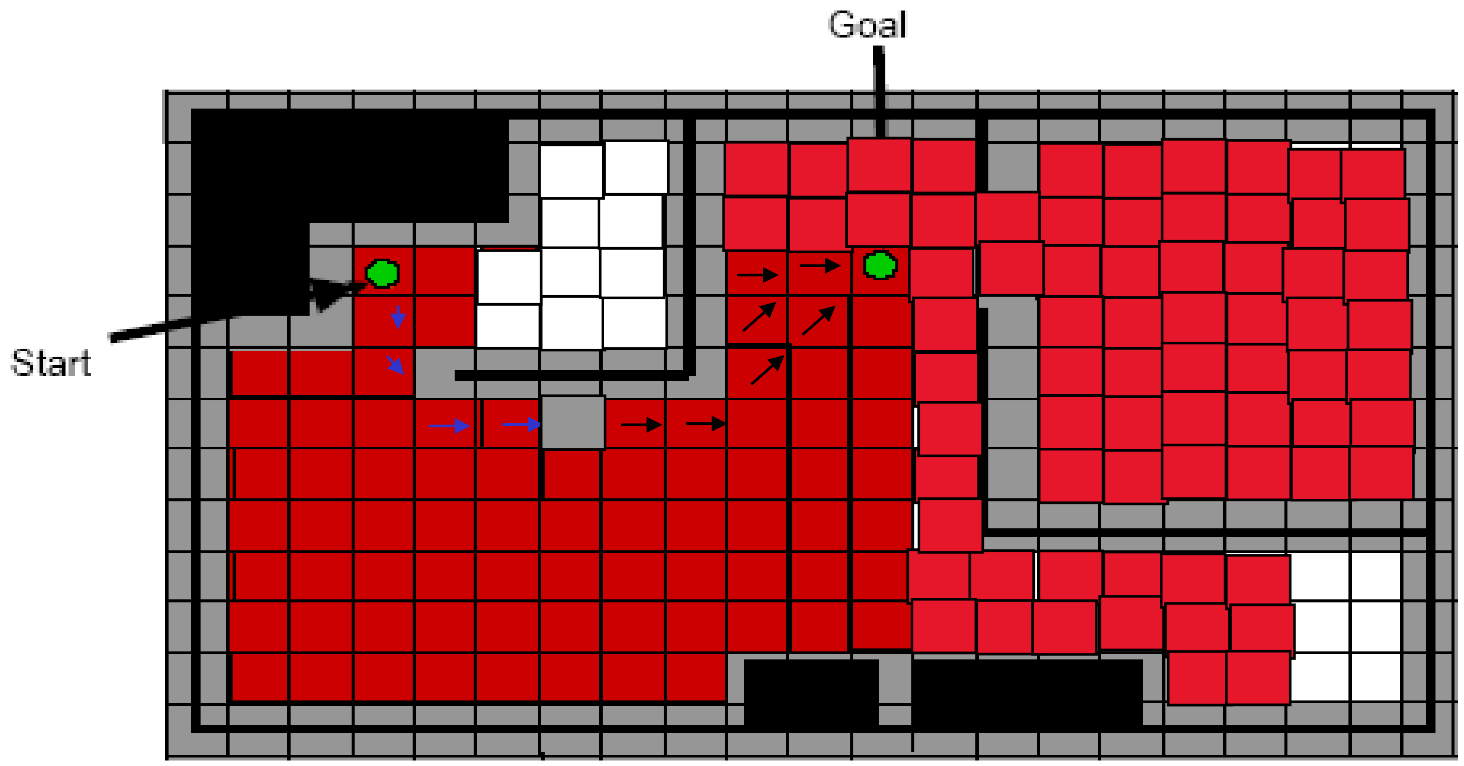
Wave-propagation ~ breath first

Search backwards



Wave-propagation ~ force field

Differential A*



Keep the costs and arrows, redo only nodes that are 'pointing into' new forbidden areas

Conclusions

- Mobile robot has to model the free space to plan paths
- After discretization into cells with connectivity, graph search techniques can be applied
- Before discretization, the parameter-space has to be reduced to degrees of freedom of the robot
- The shape of the robot can be transferred to the environment by *stamping*
- *Heuristics* are handy for search techniques, but branches have their value (*do not prune*)