LECTURE 6-2

Metric Path Planning

Introduction to AI Robotics (Sec. 10.1 – 10.3)

Quote of the Week

“Making realistic robots is going to polarize the market, if you will. You will have some people who love it and some people who will really be disturbed.”

David Hanson, CNN.com, 11/23/06

ANNOUNCEMENTS

- Quiz 12 on Sec. 10.1 – 10.3, Lec. 6-2 on *Monday, 4/26/10*
- Lab 7 due on *Tuesday, 4/27/10*
- Lab 7 memo and code is due on Angel by midnight on *Thursday, 4/29/10*

OBJECTIVES

Upon completion of this lecture the student should be able to:

- Define Cspace, path relaxation, digitization bias, subgoal obsession, termination condition
- Explain the difference between graph and wavefront planners
- Represent an indoor environment with a generalized Voronoi graph, a regular grid, or a quadtree, and create a graph suitable for path planning
- Apply wavefront propagation to a regular grid
- Explain the differences between continuous and event-driven replanning

GLOBAL PATH PLANNING

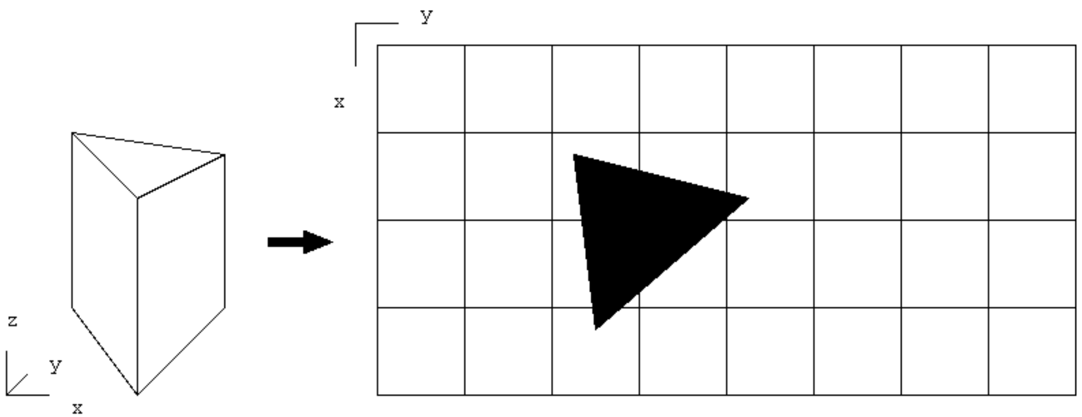
- The robot's environment representation can range from a continuous geometric description to a decomposition-based geometric map or a topological map
- Assumption: there exists a good enough map of the environment for navigation.
- Three general strategies for decomposition
  - *road map* - identify a set of routes within the free space
  - *cell decomposition* – discriminate between free and occupied cells
  - *potential field* – impose a mathematical function over the space

METRIC PATH PLANNING

- *Metric path planning*, or *quantitative navigation* is the use of metric methods to produce an optimal path to a specified goal
- Metric paths decompose a path into subgoals or *waypoints* instead of landmarks or gateways like topological navigation
- Metric path planners have two components:
  - Representation
  - Algorithm
- Representation stores the world as salient features or navigationally relevant objects and this is called the *configuration space*

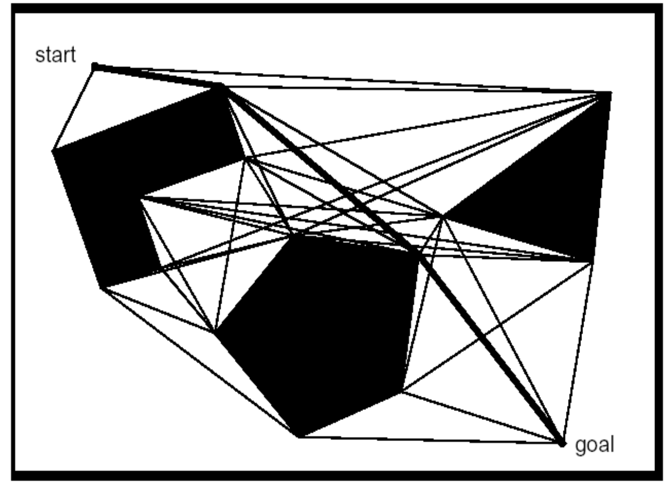
CONFIGURATION SPACE (CSPACE)

- *Cspace* transforms three dimensional space to 2 dimensional space suitable for robots, this is a simplifying assumption
- This is more amenable for storage in computer and for rapid execution of algorithms
- There are several types of Cspace representations including Voronoi diagrams, regular grids, quadtrees (octrees), vertex graphs, and hybrid free space/vertex graphs



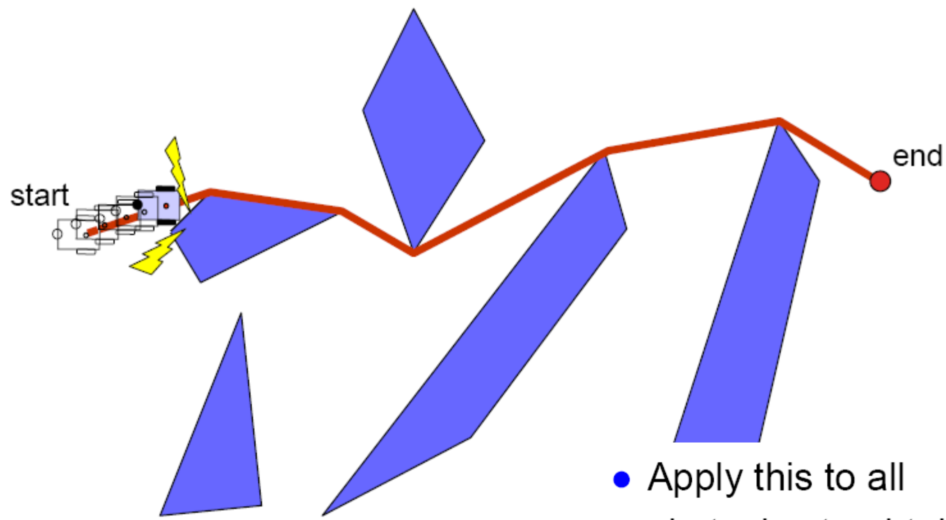
VISIBILITY GRAPH

- the *visibility graph* consists of all edges joining vertices that can see each other
- objects in the environment are polygons in either discrete or continuous space
- the size of the representation and the number of edges and nodes increase with the number of polygons
- paths take the robot as close as possible to obstacles on the way to the goal

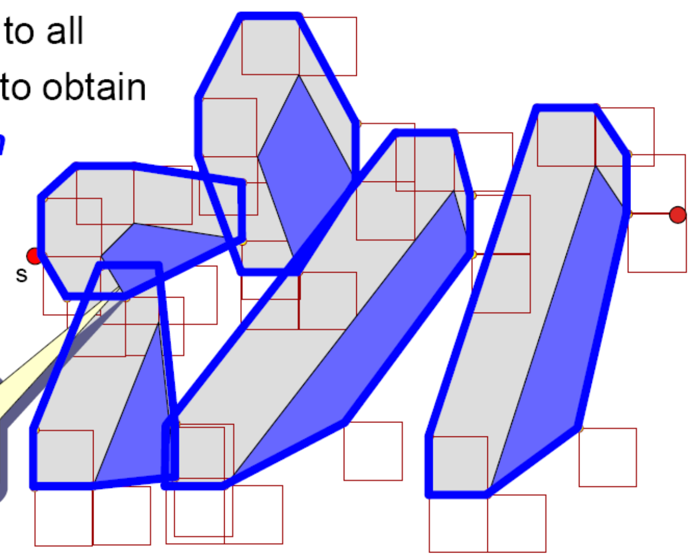


- the length of the solution path is *optimal*
- sense of safety from obstacles is sacrificed for this optimality
- one solution is to grow obstacles by the robot's radius or modify the solution path

VISIBILITY GRAPH PATHS

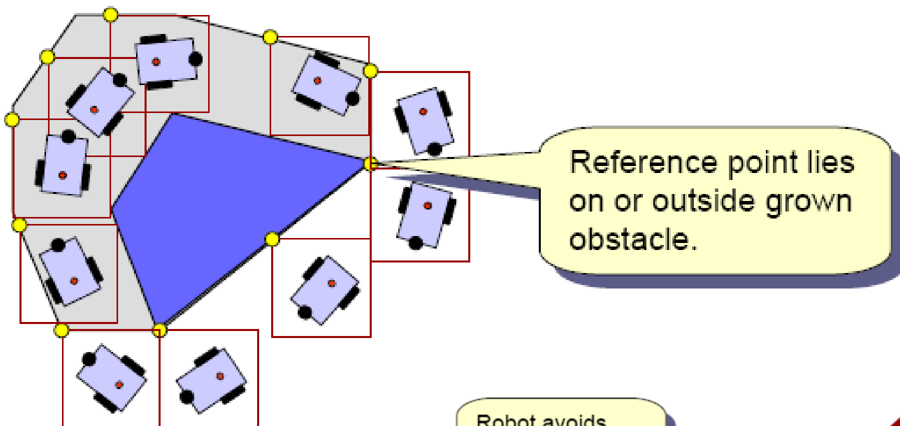


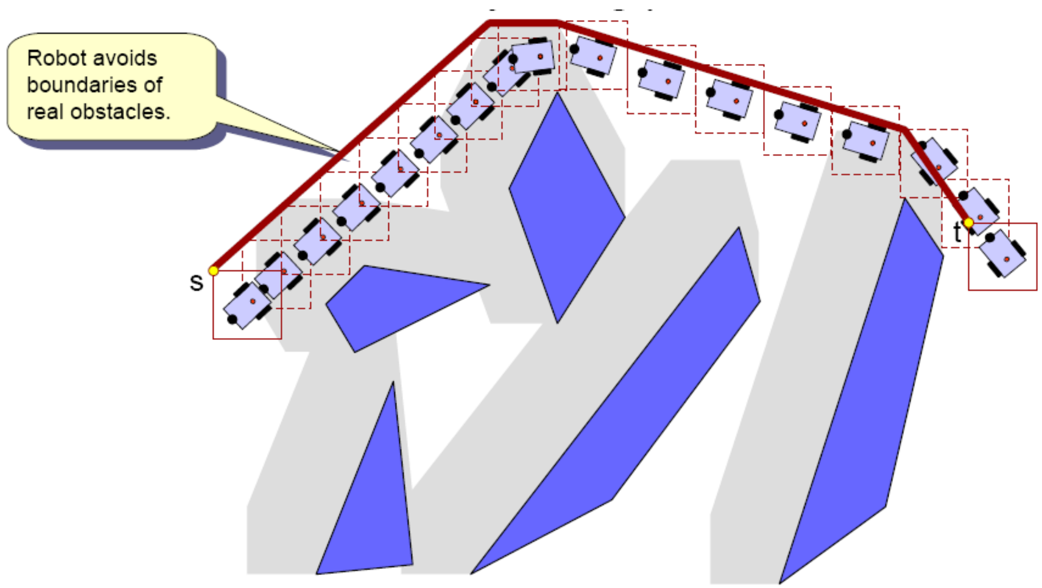
- Apply this to all obstacles to obtain the ***grown obstacle space***.



Grown obstacles may overlap … indicating that robot cannot travel safely in between.

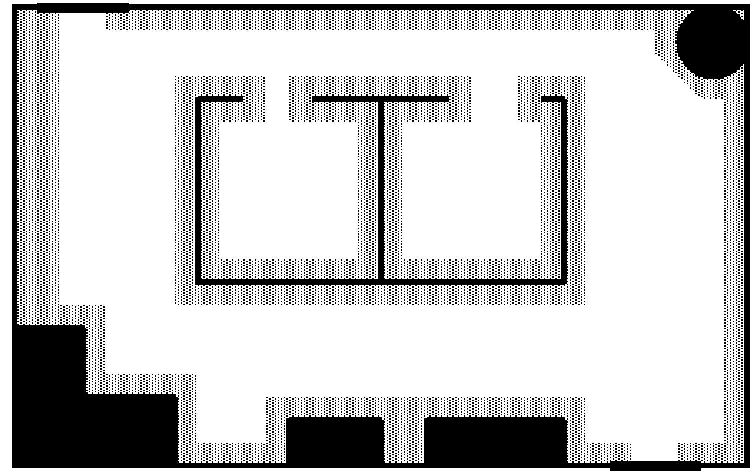
VISIBILITY GRAPH PATHS

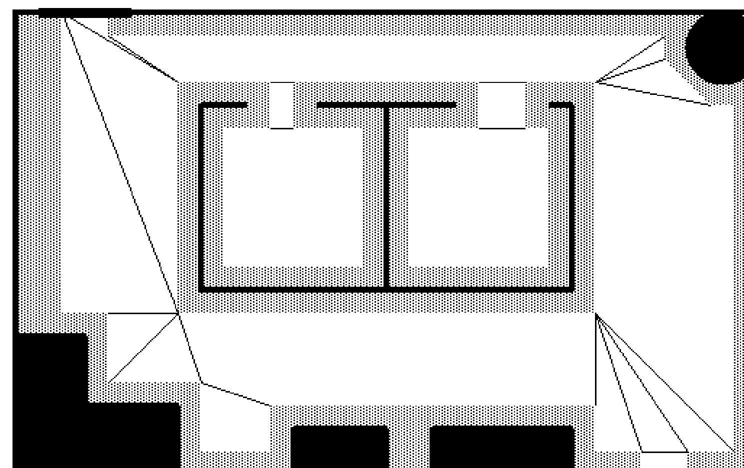




MEADOW MAPS

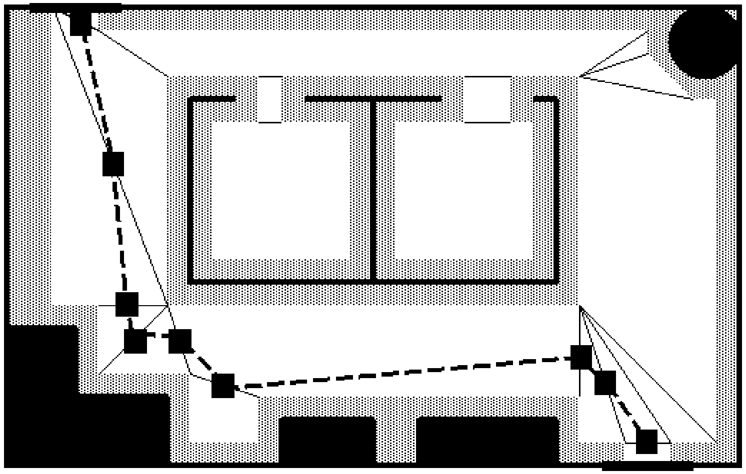
- A ***meadow map*** or hybrid vertex-graph free-space model transforms free space to convex polygons
- The polygon represents a safe region for the robot to traverse
- The first step is to grow obstacles to be the size of the robot and treat the robot as a point
- Construct convex polygons between pairs of corners or edges

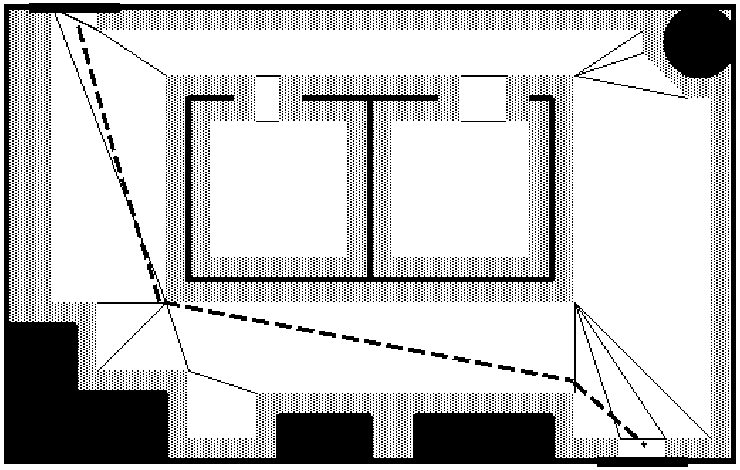




MEADOW MAPS, CONT.

- Convert the polygons to relational graphs
- Some of the challenges are that there is not a unique set of polygons
- You can't create this map with sensor data
- The robot cannot recognize corners and edges or the middle for navigation
- Path relaxation may help with some of these challenges



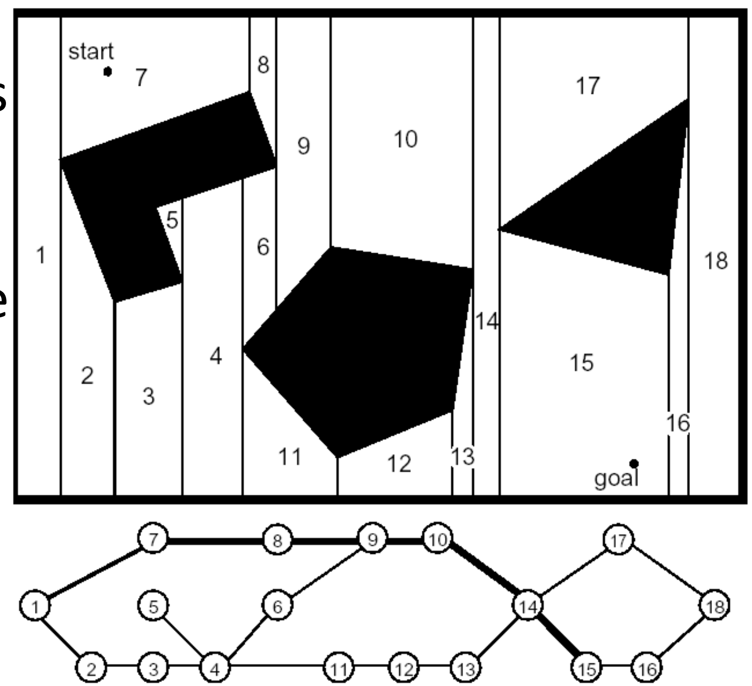


CELL DECOMPOSITION PATH PLANNING

- Use cell decomposition to discriminate between geometric areas, or cells that are free and those that are occupied by objects
- Divide space into simple, connected regions called ***cells***
- Determine which open cells are adjacent and construct a ***connectivity graph***
- Find cells in which the initial and goal configuration (state) lie and search for a path in the connectivity graph to join them.
- From the sequence of cells found with an appropriate search algorithm, compute a path within each cell.
  - e.g. passing through the midpoints of cell boundaries or by sequence of wall following movements.

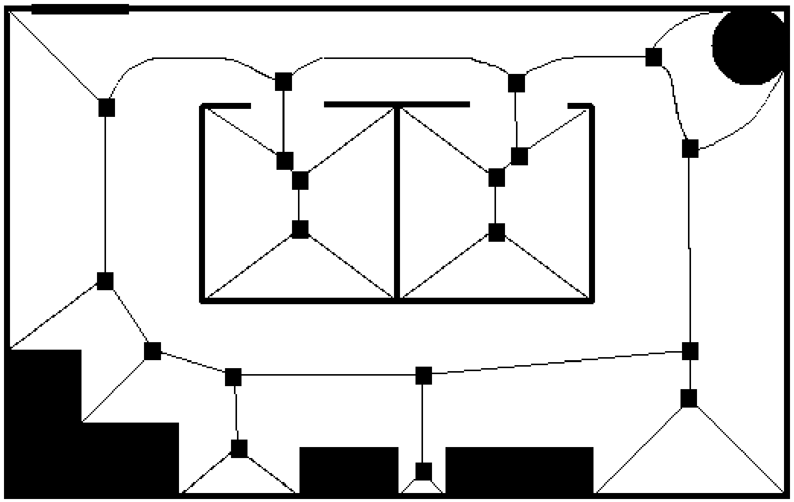
CELL DECOMPOSITION PATH PLANNING

- An important aspect of **cell decomposition** is the placement of the boundaries between the cells
- if the boundaries are placed as a function of the structure of the environment then the method is ***exact cell decomposition***
- if the decomposition is an approximation of the actual map, the system is an ***approximate cell decomposition***



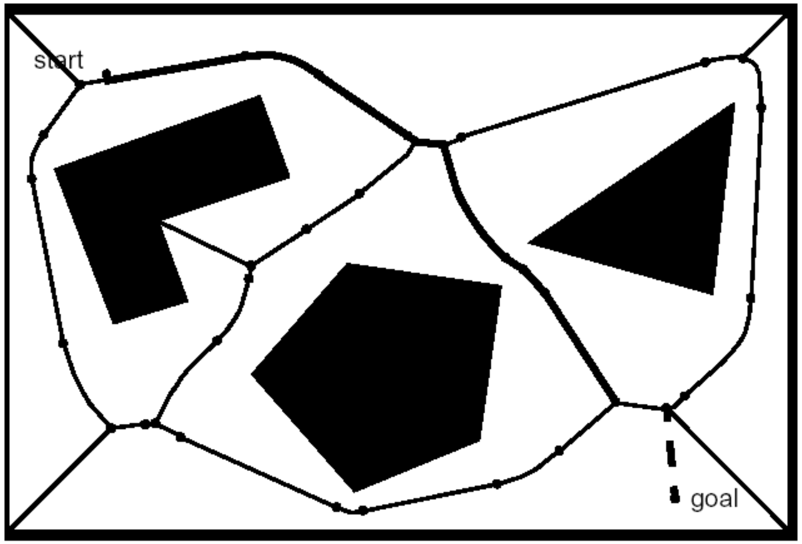
GENERALIZED VORONOI GRAPHS

- A ***GVG*** can be created as a robot enters a new environment as a topological map
- A ***Voronoi edge*** is equidistant from all points
- The point where Voronoi edges meets is known as the ***Voronoi vertex***
- a ***Voronoi diagram*** is a complete road map method that tends to maximize the distance between the robot and obstacles
- paths on the **Voronoi** diagram are usually far frorm optimal in the sense of the total path length
- the ***Voronoi diagram*** has the advantage in ***executability***

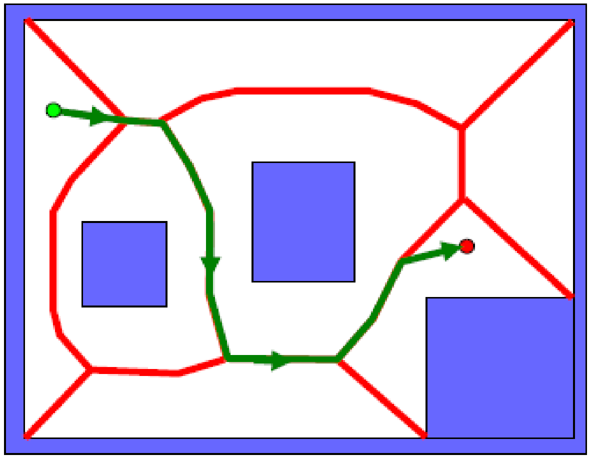


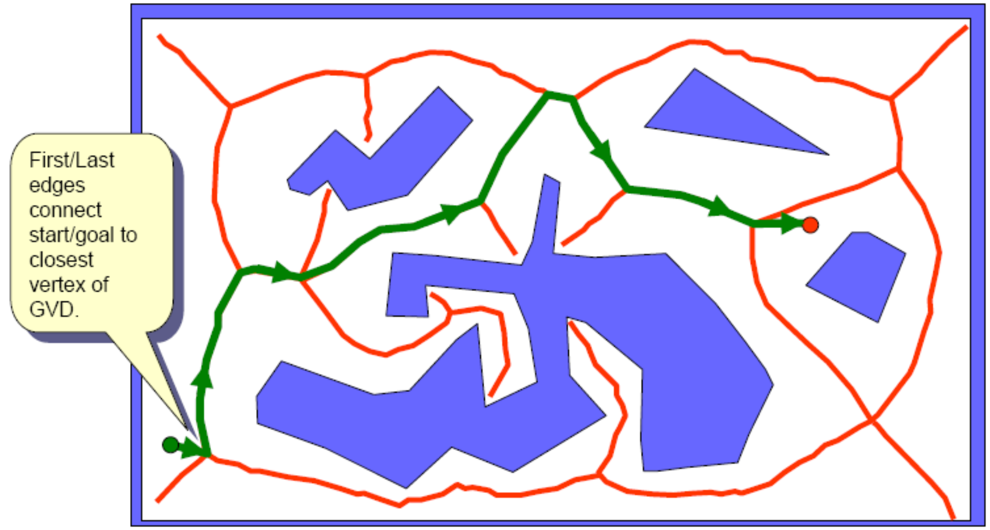
VORONOI DIAGRAM CONT.

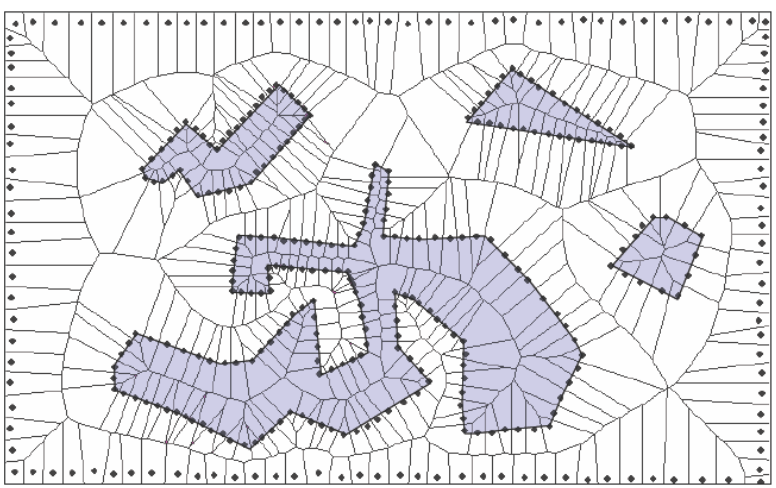
- This is easy for a robot to follow because the implicit local control strategy is to stay equidistant from all obstacles
- If the robot follows the Voronoi edge it will not collide with any obstacles, there is no need to grow the obstacles
- one important weakness is in the case of limited range localization sensors.
- this has been used to conduct ***automatic mapping*** by finding and moving on unknown Voronoi edges and then constructing a consistent Voronoi map of the environment

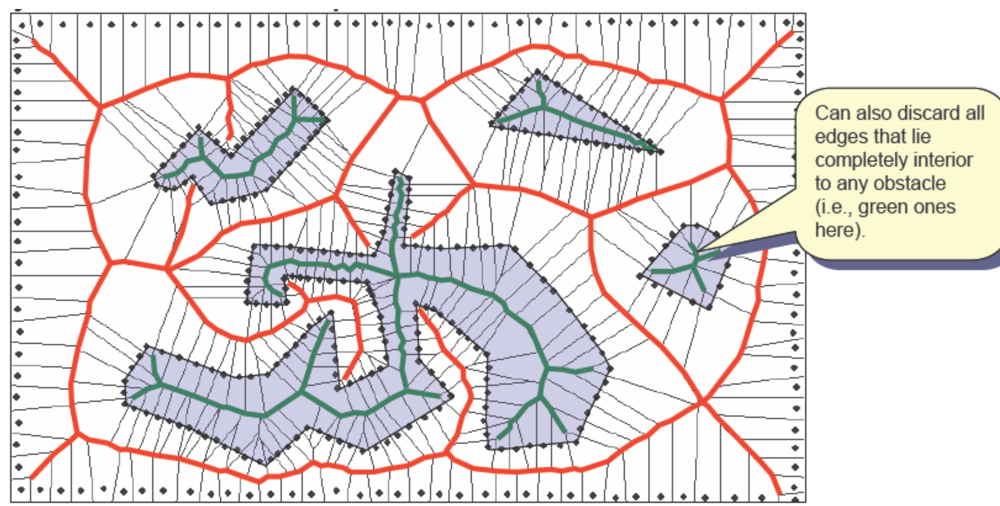


VORONOI DIAGRAM

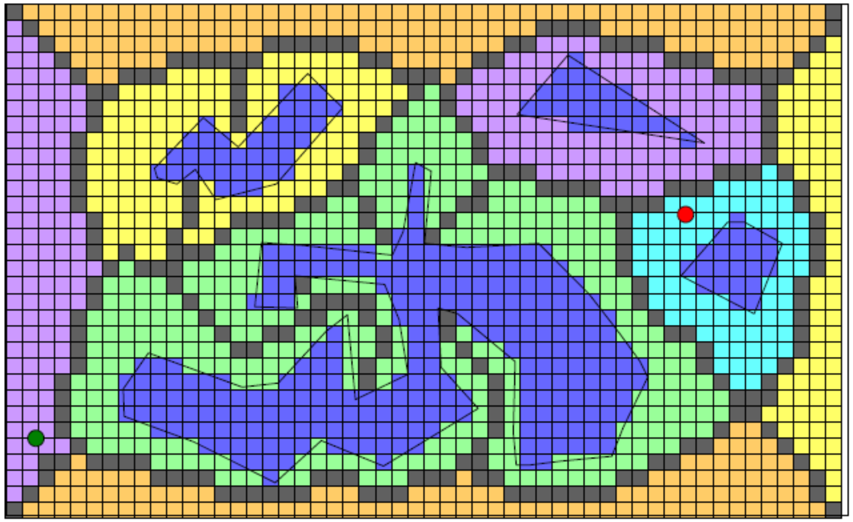


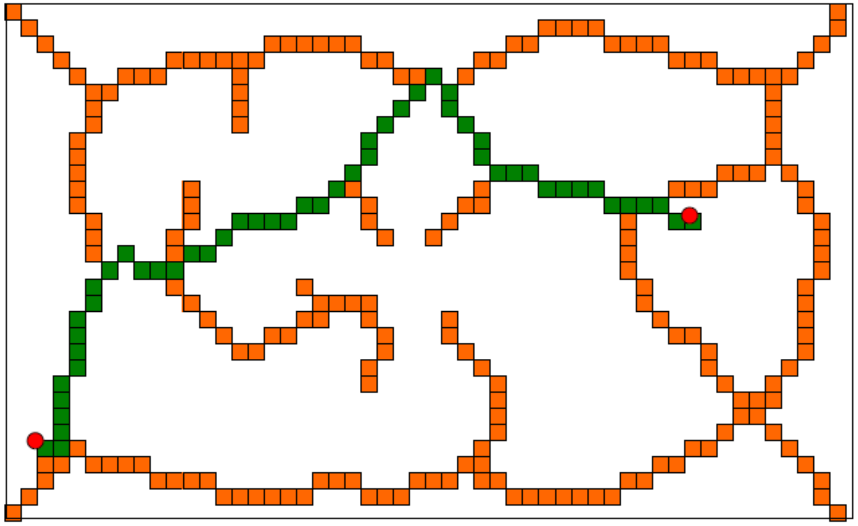






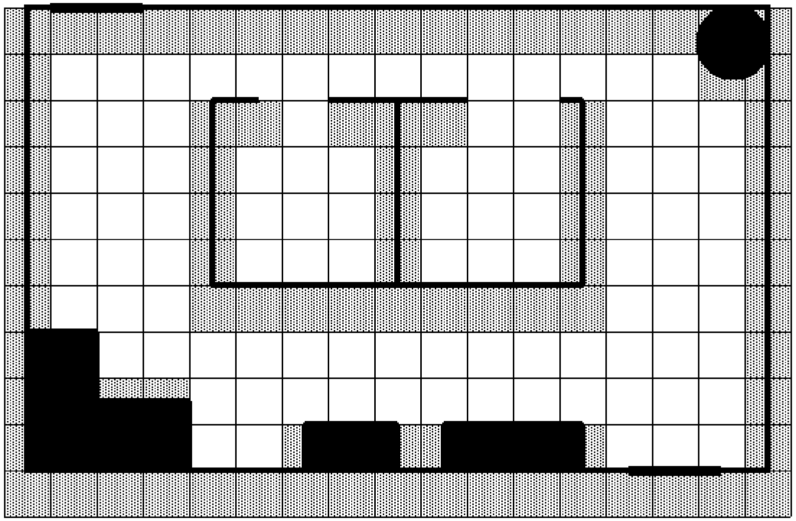
DISCRETIZED VORONOI DIAGRAM





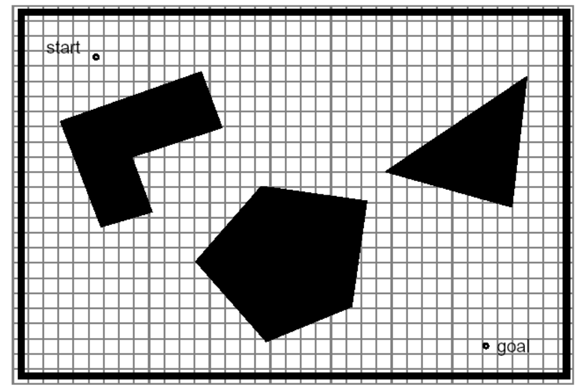
REGULAR GRIDS

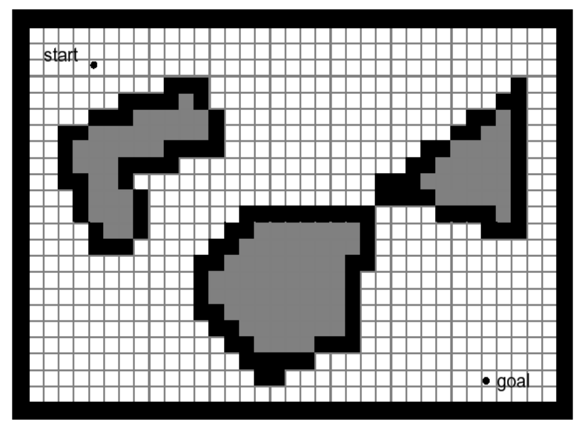
- A ***regular grid*** superimposes a 2D Cartesian grid on the world space
- If there is an object in an area, that element is marked as occupied
- This is also called an ***occupancy grid***
- Grids can be ***4-connected*** or ***8-connected***
- Regular grids suffer from ***digitization bias*** because if an object falls into any part of a grid (cell) element it is marked as occupied



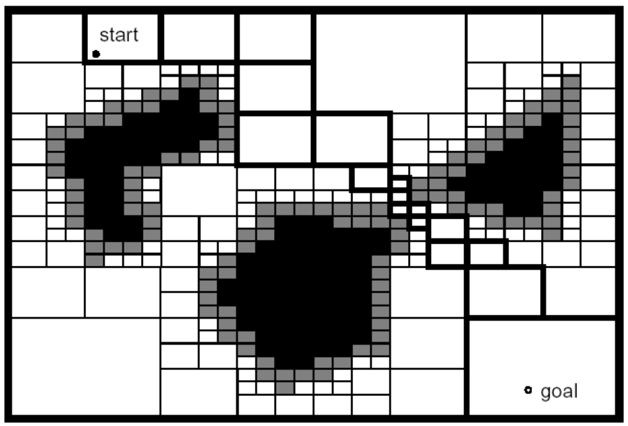
EXACT CELL DECOMPOSITION

- the boundaries of cells is based on geometric criticality
- the cells are completely free or occupied
- what matters is the robot's ability to traverse from each free cell to adjacent free cells
- efficient computation in that case of large, sparse environment
- used rarely in mobile robot applications due to complexities in implementation





ADAPTIVE CELL DECOMPOSITION QUADTREE



- ***Quadtrees*** avoid wasted space because it is a recursive grid
- If an object falls into part of a grid it divides the element into four
- A 3D ***quadtree*** is called an ***octree***

- one of the most popular techniques for mobile robot path planning
- cell size is not dependent upon objects in an environment so narrow passageways may be lost
- low computational complexity for path planning
- the fundamental cost is memory because the grid must be represented in entirety
- sparse environments contain few cells consuming dramatically less memory