Robotics
Robot Navigation (3)

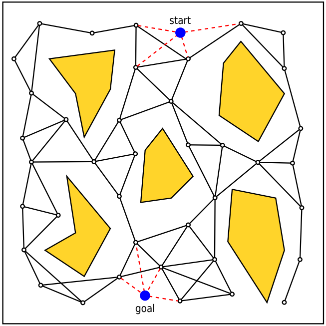
Tullio Facchinetti
<tullio.facchinetti@unipv.it>

Wednesday 30th October, 2024

http://robot.unipv.it/toolleeo

Roadmap

a roadmap M is a sequence of unidimensional curves connecting a starting point p_{start} and a goal p_{goal}



properties

1. accessibility: a path exists from p_{start} to some point $p'_s \in M$
2. departability: a path exists from a point $p'_g \in M$ to the goal point p_{goal}
3. connectivity: a path exists between p'_s and p'_g made of nodes belonging to M

Probabilistic methods

when the dimension of the configuration space becomes high, some approaches are needed to reduce the complexity of the search: this is the objective of probabilistic methods

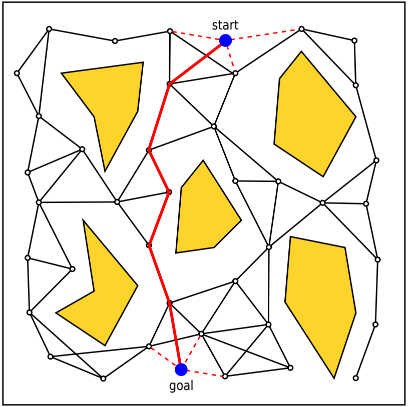
- we will consider the so-called "sampling-based methods"
- the father of all sampling-based methods is the Probabilistic RoadMap (PRM) trajectory planner

PRM: Probabilistic RoadMap

it leverages the idea that it is effortless to decide whether a point belongs to the free configuration space or not

- first, a map of the free space is built
- the planner randomly generates points in the configuration space
- it keeps only those point that belong to the free space
- rough generation of point
- accurate planning of trajectories between collected points

Probabilistic methods: example



Operations of a probabilistic planner

learning phase

- a map of the configuration space is built
- a graph is generated to connect points in the free space

query phase

- check whether p_{start} and p_{goal} can be connected to the map, to nodes p'_s and p'_g respectively
- generate the shortest path between p'_s and p'_g (e.g. using A^*)

Learning phase

1. a set of V points in the configuration space is generated
2. for each point $q \in V$, the k closest points are considered, composing the V_k set
3. for each point $p \in V_k$ a path is found to connect p and q, if p and q are not yet connected

key issues

- how the points in V are generated (randomly, using some models of obstacles, addressing the narrow passage problem, etc.)
- considering the distance between points in V_k (the computation time increases with such a distance, since more points are involved)
- how to connect pairs of points ("local planner": straight lines, splines, etc.)

Query phase

for both the starting point p_{start} and the goal p_{goal}:

1. the k closest points of the map to p_{start} and p_{goal} are considered
2. the possiblity to connect p_{start} to p'_s and p_{goal} to p'_g is checked
3. the shortest path between p'_s and p'_g is searched

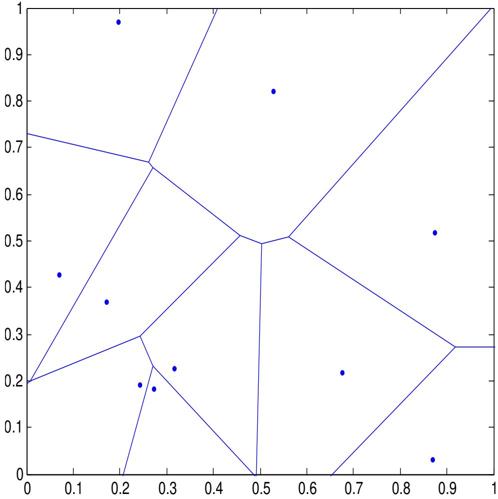
key issues

- how many nodes k shall be considered
- how to connect pairs of points ("local planner": straight lines, splines, etc.)

Voronoi diagrams

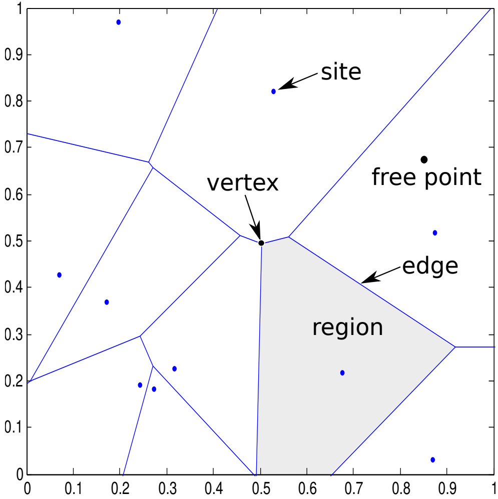
also known as:

- Voronoi tessellation
- Voronoi decomposition
- Voronoi partition
- Dirichlet tessellation



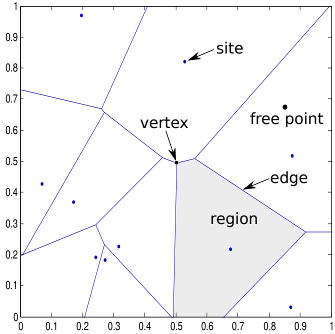
Notation

- p_i : sites
- q : free points
- e : (Voronoi) edge
- v : (Voronoi) vertex
- F_i : region or cell

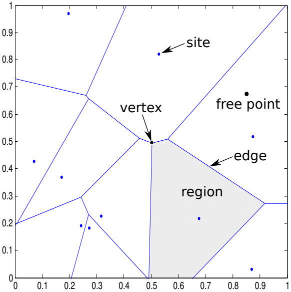


Usage for path planning

- sites correspond to obstacles
- the edges of the Voronoi graph constitute the roadmap
- the roadmap can be modeled as a graph connecting the Voronoi vertexes
- start and goal points are connected to the roadmap, and the path is searched in the graph
- the Voronoi map guarantees to move as far as possible from the closest obstacles



Voronoi region



set of sites:

$$P = \{p_i\}, 1 \leq i \leq n$$

Voronoi diagram of $Vor(P)$:

- division of the plane into n regions
- there is a region for each site

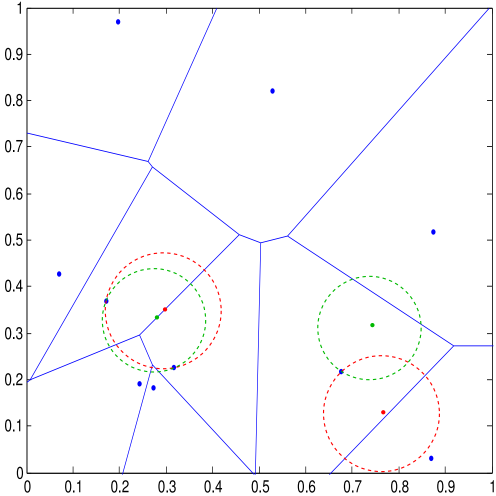
definition of the i-th region:

$$F_i = \{q \in Q_{\text{free}} : d(p_i, q) \leq d(p_j, q) \forall i \neq j\}$$

Voronoi regions are bounded by line segments

Features of vertices

1. the locus of the center of a largest empty circles passing through only a pair of points $p_i, p_j \in P$ defines an edge
2. the locus of the center of largest empty circles passing through only one points in P defines a region



Construction of a Voronoi map

sensor-based

- uses the robot onboard sensor to build the map
- the map can be built online

polygonal spaces

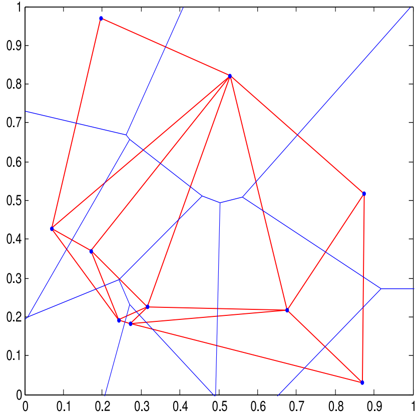
- a full representation of the environment is needed
- obstacles must be polygons

grid map

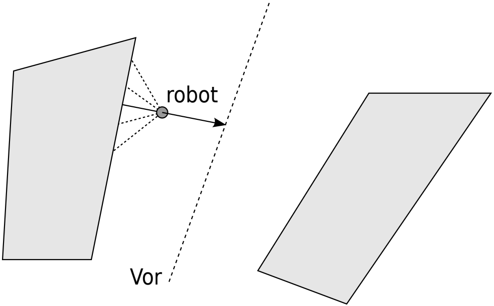
- a full representation of the environment is needed
- easy and fast to compute

Delaunay triangulation

- Delaunay triangulation is complementary to the Voronoi tessellation
- each triangle's edge is orthogonal to a Voronoi edge
- it can be leveraged to construct the Voronoi partition
  - an algorithm is applied to build the Delauney triangulation
  - the Voronoi graph is derived from the triangulation

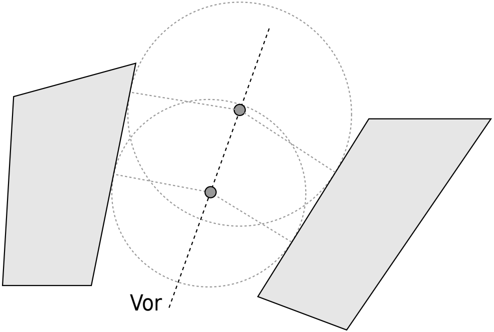


Sensor-based construction



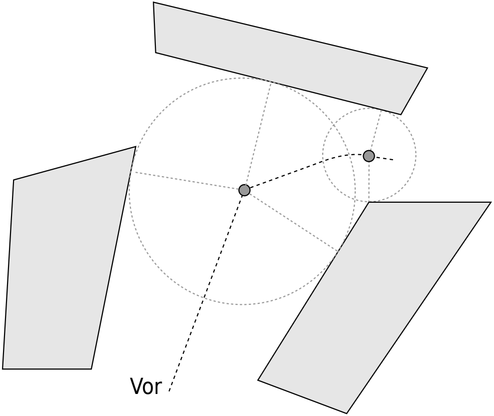
- the robot reaches the *Vor* with gradient ascent
- motion along the direction that maximise $\nabla d(q_i, q)$ from the closest obstacle

Sensor-based construction



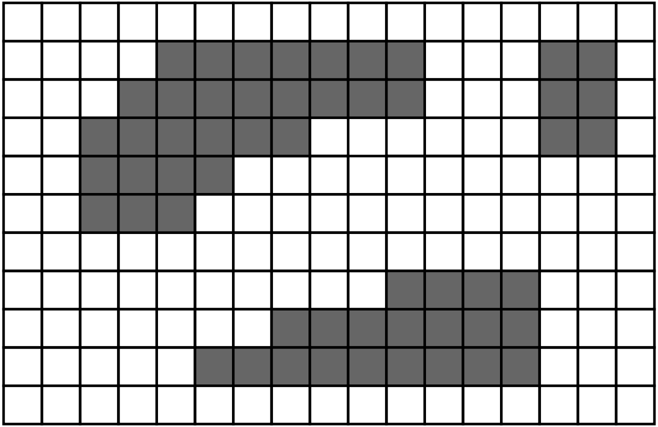
- the robot moves along a line that keep $d(q_i, q) = d(q_j, q)$
- q_i and q_j are the closest points belonging to the closest obstacles

Sensor-based construction



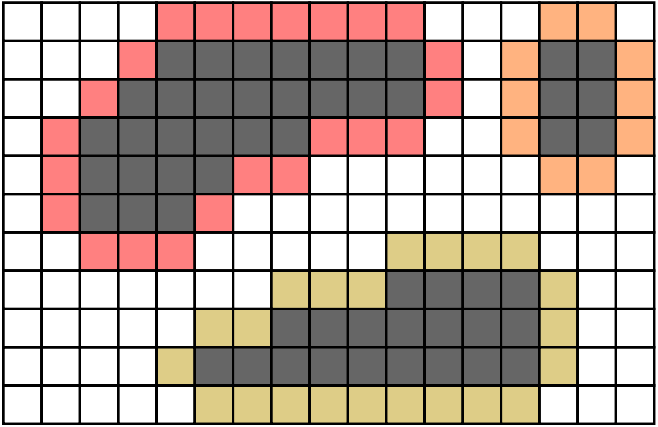
- meet points are detected from changes in the closest obstacle

Grid-based construction



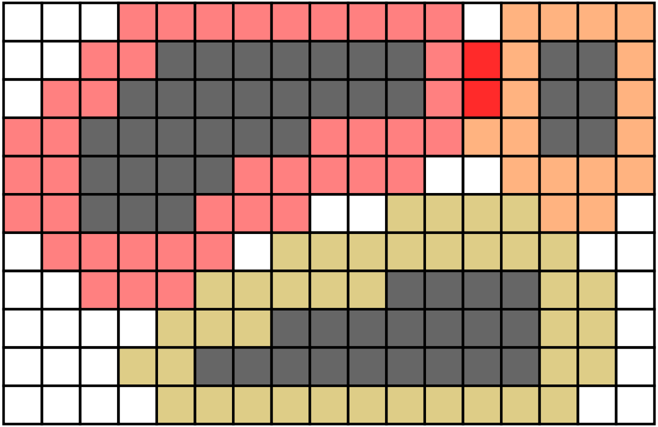
- the space is represented by means of a grid

Grid-based construction



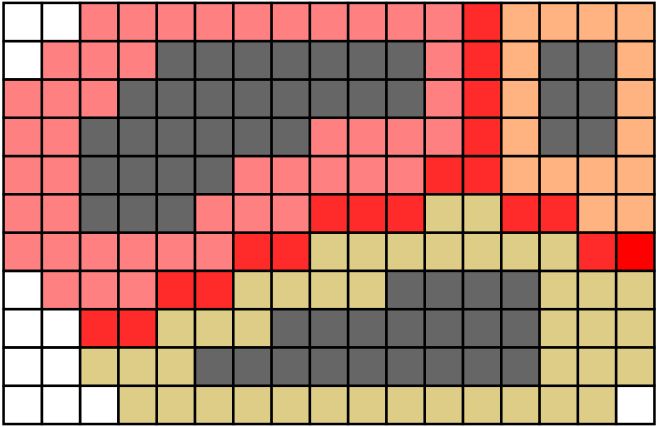
- a wave-front grow is started from each obstacle

Grid-based construction



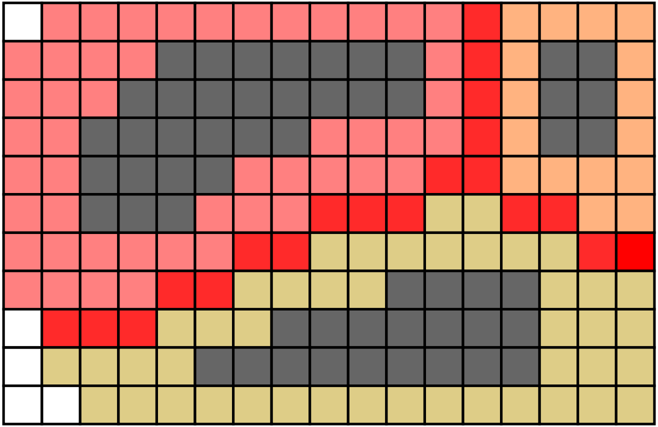
- a collision between two waves detects the presence of a *Vor* edge

Grid-based construction



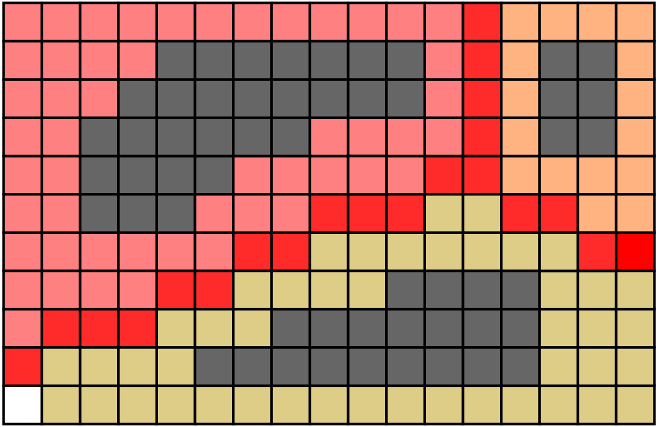
- where there is a collision "on the boundary", the selection of the grid cell to mark is up to the algorithm

Grid-based construction



- the algorithm continues until...

Grid-based construction

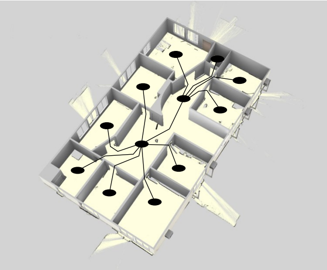


- ...the grid is fully divided into Voronoi regions

Cell decomposition

exact cell decomposition

the free space is represented by the union of adjacent cells (regions)



- cell boundaries are often associated to physical features of the free space
  - change in the line of sight
  - change in the closest obstacle
  - intersection with vertices
- adjacent cells share a common boundary

Cell decomposition

adjacency graph

it encapsulates the relationships between adjacent cells

- a node is associated to a cell or an edge (boundary)
- graph edges connect adjacent cells

path planning

- the planner finds the cell containing the start and goal points
- the adjacency graph is used as a roadmap

Trapezoidal decomposition

- made by two-dimensional cells having trapezoidal shape
- triangular cells are possible
  - they can be seen as degenerated trapezoid
  - one of the parallel edges is collapsed into a 0-length edge

assumptions

- planar configuration space
- obstacles are polygonal
- the free space is bounded by a polygon

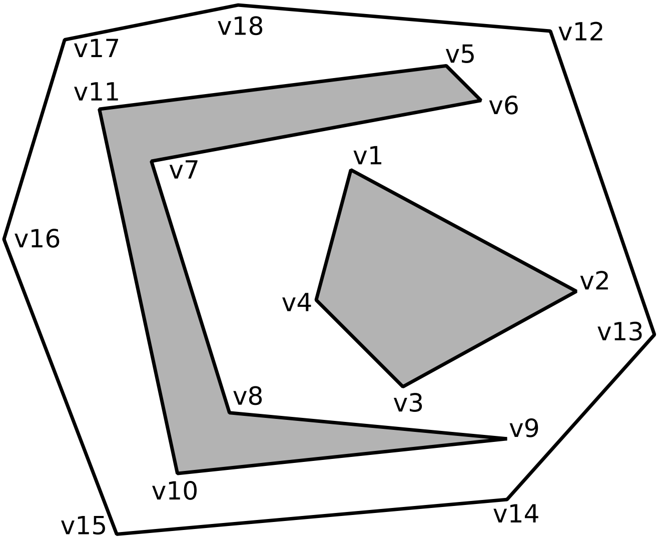
Build the trapezoidal decomposition

- for each vertex v_i, perform the following procedure
- draw two segments originating from v_i
- the extension of each segment stops when it first intersects an edge
- segments can be drawn in any direction, but they must be parallel

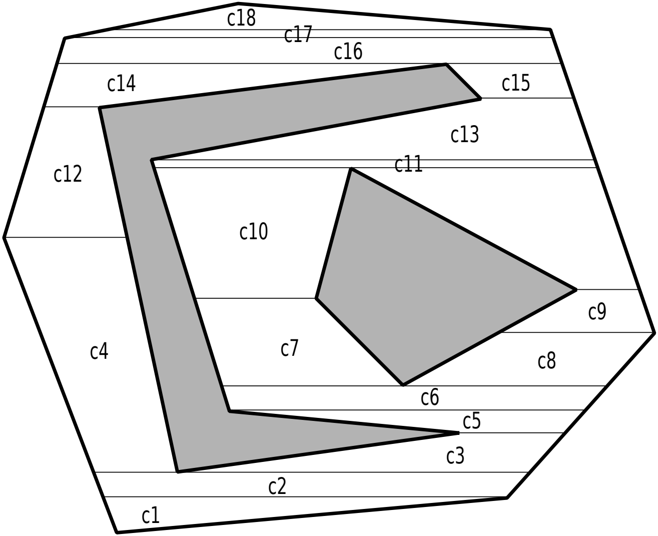
note:

- some vertices may generate only one or even no segment

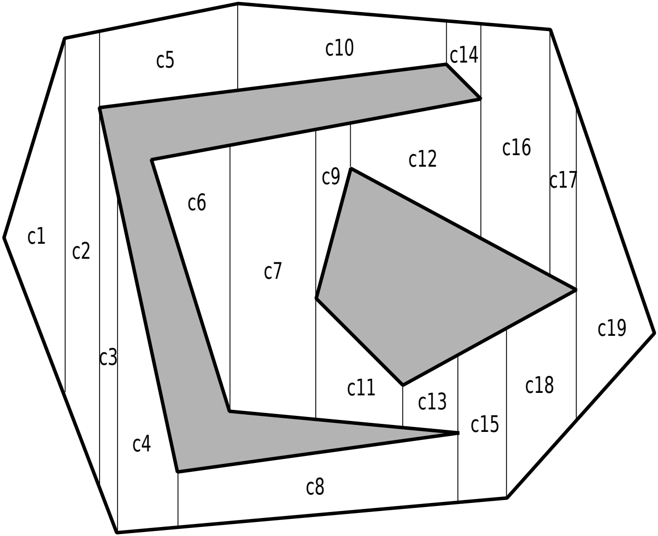
Build the trapezoidal decomposition: example



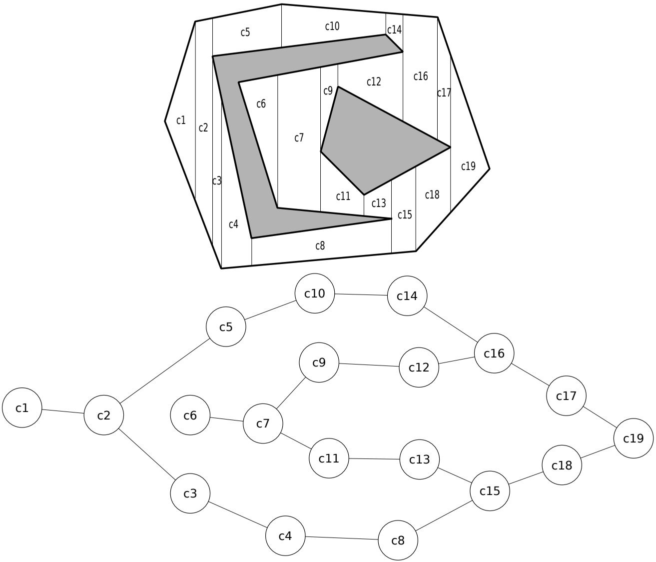
Build the trapezoidal decomposition: example



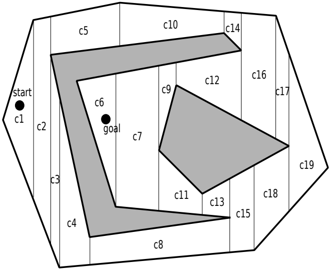
Build the trapezoidal decomposition: example

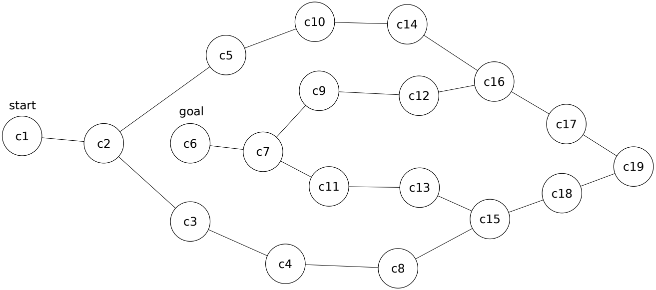


Build the trapezoidal decomposition: example

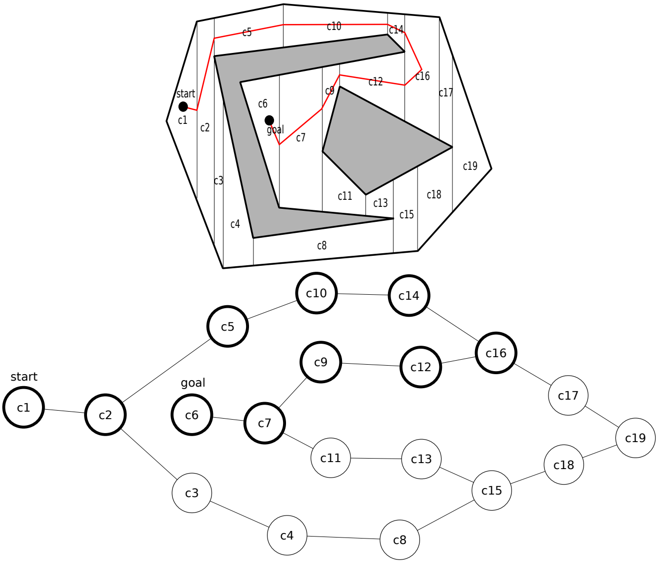


Build the trapezoidal decomposition: example





Build the trapezoidal decomposition: example



Path planning

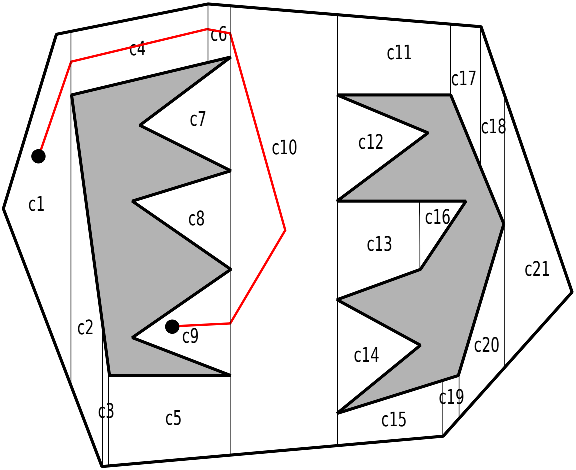
- the cells containing the start ang goal points are identified
- the adjacency graph is searched for a path connecting the two cells
- the sequence of nodes in the graph must be translated into waypoints in the free space

Path planning

trapezoids are convex
therefore the centroid can be connected with every boundary segment by a straight line

1. beginning from the start point...
2. each point is connected with the mid-point of the segment associated with the next graph node
3. the mid-point is connected to the centroid of the trapezoid
4. repeat from step 2 until the cell containing the goal point is reached
5. connect the last mid-point with the goal point

Properties of trapezoid



- many boundary segments are related with cell $c10$
- the approach to reach the centroid allows to easily reach every boundary segment

Coverage based on cell decomposition

coverage

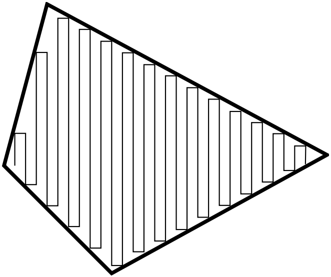
the ability to pass a sensor or effector over all points composing the free space

applications:

- exploration
- survellaince
- manufacturing (e.g., painting, polishing, smoothing)

Coverage based on cell decomposition

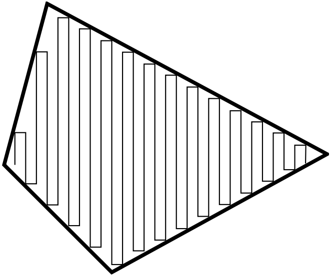
- in general, the structure of a cell is simple
- after the free space is divided into cells, simple algorithms can be used to cover each cell
- once all cells are visited, coverage is obtained



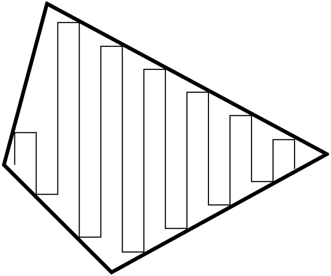
back-and-forth coverage of a cell

Coverage based on cell decomposition

- the horizontal offset determines the trade-off between coverage accuracy and execution time
- wider offset leads to quicker but more inaccurate coverage
- shorter offset leads to more accurate coverage but takes more time



shorter offset



wider offset