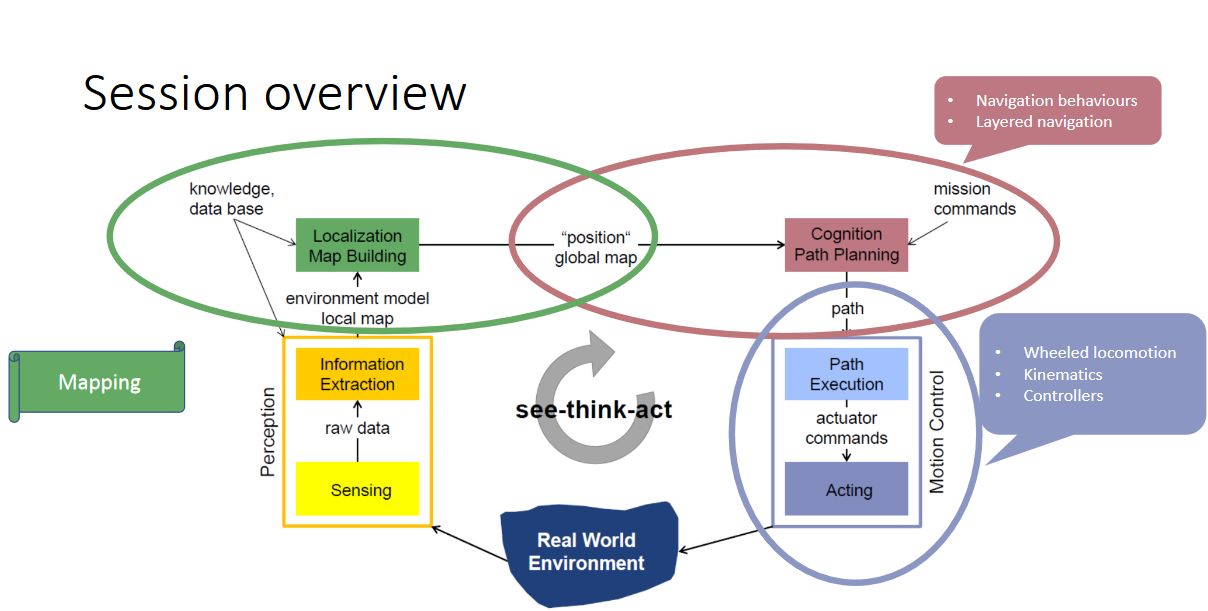
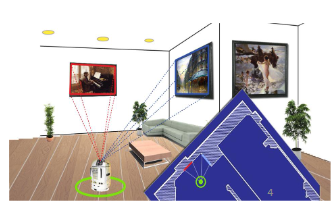
Foundations of Robotics - Lecture 11 Mapping

Gautham Das

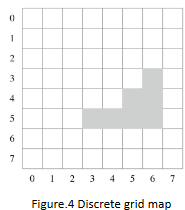


Localisation

* Given a map of the environment, what is the pose of a robot  
  For a ground robot –Pose = 𝑥,𝑦,𝜃
* We need a map of the environment to localise a robot
* Landmark based localisation
  + Angle and distance to landmarks
  + Triangulation
  + Markov localisation

In this session, we will see:

* How to represent a map
* How can we create a map –Exploration (Assuming perfect motion)
* How can we address simultaneous localisation and mapping (SLAM)
* Mapping and localisation are complementary

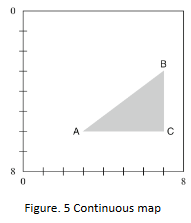
Discrete Maps

Simple representation (grid map)

* 0 – Unoccupied cell
* 1 – Occupied cell

Memory use grow with map size and resolution

Can be inaccurate depending on resolution

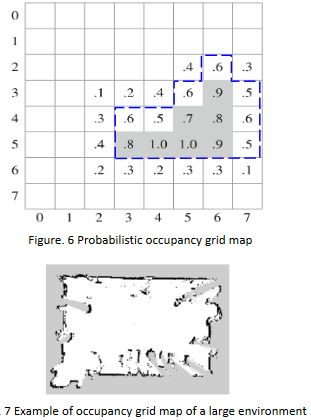
Continuous Maps

Represents environment and objects

* Square environment with sides of length 8 units
* A triangular object with corners at position A, B and C

Good for few/ simple objects

* Complexity grows with number of objects, complexity of shapes and boundary which is not straight lines

Probabilistic Map

Each cell is assigned a probability to be occupied.

Cells without numbers are assumed to be 0

A threshold p\* can be set to decide whether there is an obstacle or not

e.g. p\* >= 0.7 (grey obstacle in Figure.6) or p\* >= 0.5(blue obstacle in figure.6)

Drawbacks discussed for discrete maps are applicable.

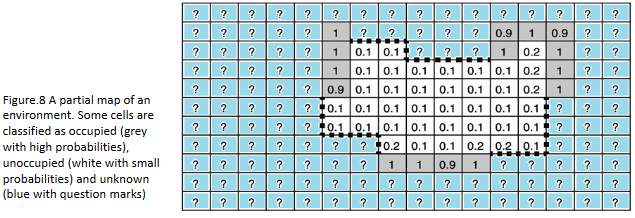
Mapping – Exploration

A robot with a range sensor and odometry is used for mapping, it follows these series of steps:  
Explore the environment

* Start from an arbitrary location
* Move to a new position –known from odometry w.r.t. to previous pose
* Note down the sensory observations

Occupancy mapping based on sensory observations

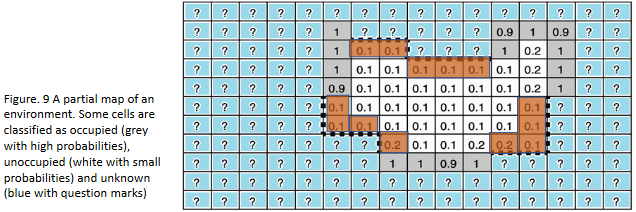
* Occupied –High probability (depending percentage of the cell being occupied)
* Unoccupied –Low probability
* Unknown –Haven’t explored yet

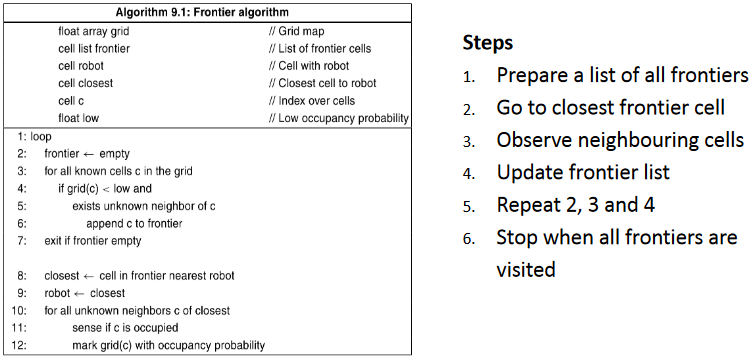


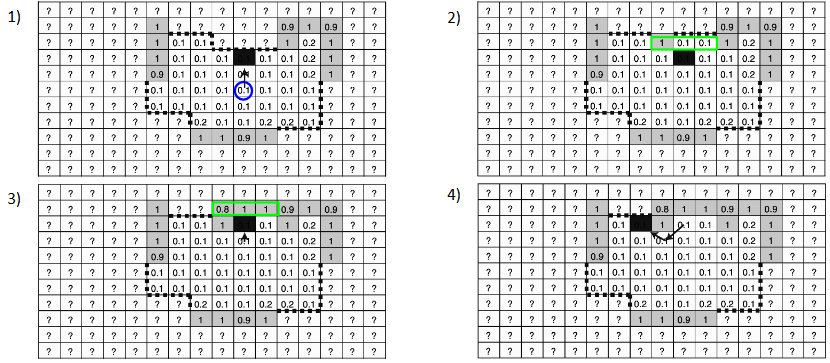
Frontier Algorithm

Frontier Algorithm –structured way to explore  
Frontier cells:

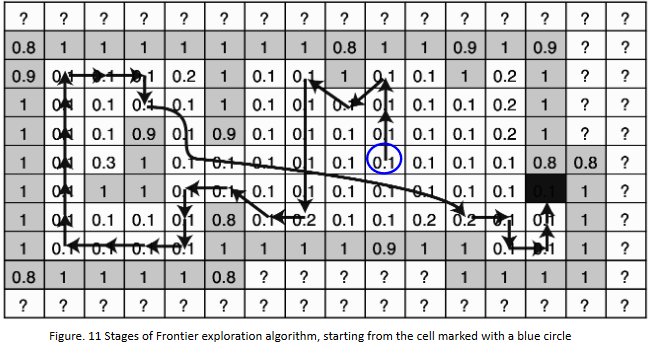
* An unoccupied cell with one or more unknown (?) neighbours is a frontier cell

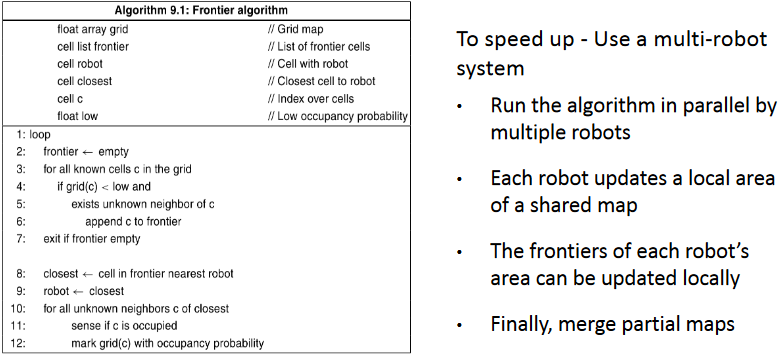




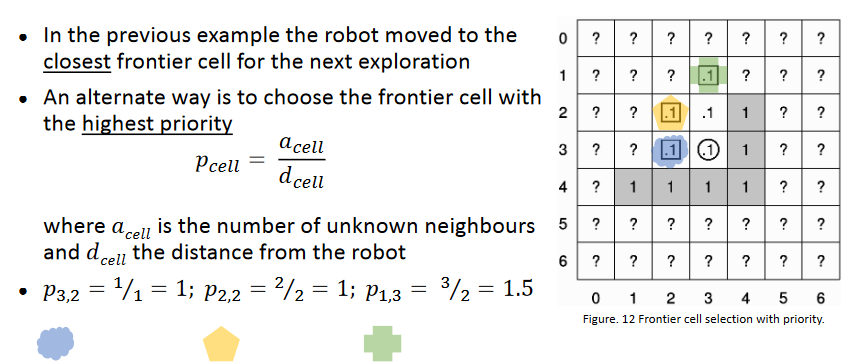








Frontier algorithm with priority

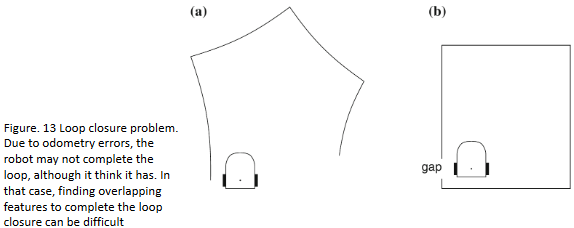


Map Correction

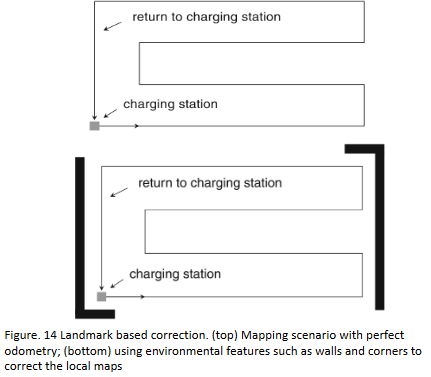
We assumed perfect Odometry for the mapping

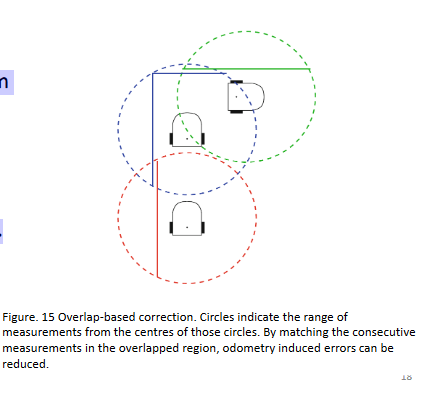
But, there are problems due to Odometry

Known landmarks and distinct features from the maps can help



Map Correction – Landmark based

• Example: A robotic lawnmower   
mowing an area and returning to   
charging station  
• Let’s say it builds a map while mowing  
• Relying on Odometry alone to build a   
map can be tricky (loop closure)  
• Instead, make use of some landmarks   
such as boundaries and corners to   
correct the map

Map Correction – Overlap based

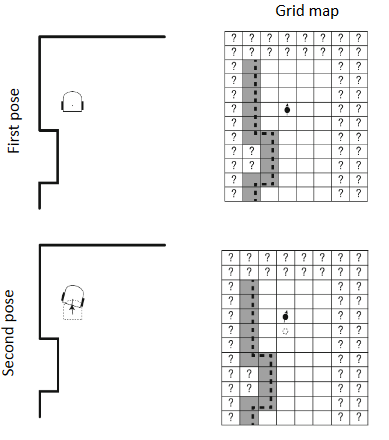
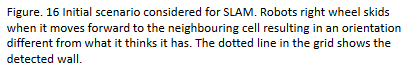
• An alternate approach is to correct   
local maps by matching features from   
overlapped regions  
• Sensors should have long range   
• Good overlap of consecutive   
perceptions between the local maps.

SLAM

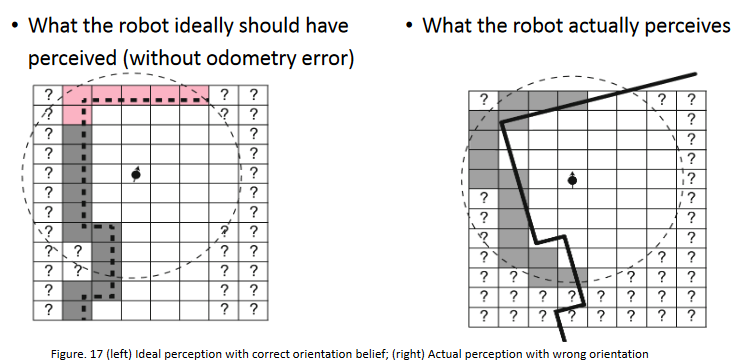
● Simultaneous Localization And Mapping  
● Chicken-and-egg problem:

* to correct the robot pose we need a good map
* to get a map we need a good robot pose

For a robot moving up one cell but its right wheel skidded, the robot thinks it is pointing up but in reality it is pointing slightly right.

SLAM – Percieved Map



SLAM – Expected Map

