Unit 4: Localization Part 1
Maps, Beliefs, and Probability Review

Computer Science 4766/6912

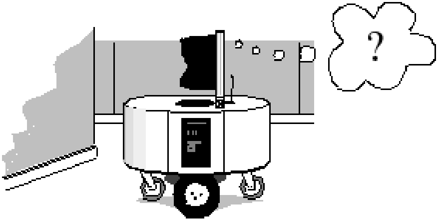
Department of Computer Science
Memorial University of Newfoundland

June 13, 2018

1. Introduction
   - Navigation
   - Why is Localization Difficult?
   - Issues
2. Map Representation
3. Belief Representation
4. Probability Review

Navigation

- Providing robots with the requisite sensors, actuators, and algorithms to allow them to navigate in general environments is a big job; Four major subtasks are...
  - Perception: extract meaningful data from sensors
  - Localization: determine the robot's position
  - Cognition / Planning: decide on what actions to take
  - Motion control: control the actuators
- Localization answers the question "Where am I?"



Why is Localization Difficult?

Doesn't odometry solve the localization problem?

- Localization by odometry (i.e. integrating wheel rotation) is subject to cumulative error

Doesn't GPS solve the localization problem?

- GPS is enormously helpful in some applications, but has disadvantages for others:
  - GPS will be unavailable in a variety of situations:
    - Indoors
    - In obstructed areas
    - Underwater
    - In space (or on other worlds)
    - If the U.S. decides to make it unavailable
  - Insufficiently accurate for localizing smaller robots (such as the "body-navigating nanorobots of the future" [Siegwart and Nourbakhsh, 2004])

Issues

- Sensor noise:
  - Noise from proprioceptive sensors such as wheel encoders makes localization w.r.t. last known position difficult
  - Noise from exteroceptive sensors makes localization w.r.t. a map difficult
- **Perceptual aliasing**: the sensor data obtained at one location is indistinguishable from the data obtained at another location
- Noise introduced by movement:
  - Systematic errors: e.g. Differences between wheel radii
  - Non-Systematic errors: e.g. Wheel slippage

Map Representation

- **In this part of the course we will assume the existence of a map**
- Map used to compare with the robot's current sensory state
  - Can be supplied to the robot or,
  - Built autonomously — SLAM (Simultaneous Localization and Mapping)
- Choice of map representation depends on precision required, available sensors, and computational constraints

A **continuous representation** represents all mapped objects in continuous-space.

e.g. represent map as the set of infinite lines through object boundaries [Tomatis et al., 2003]

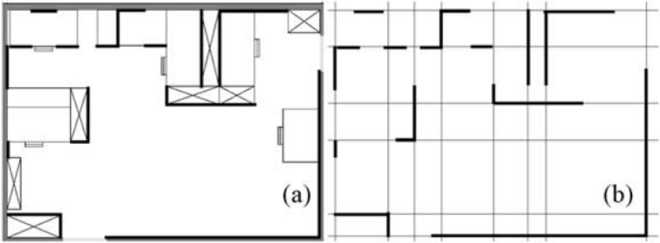
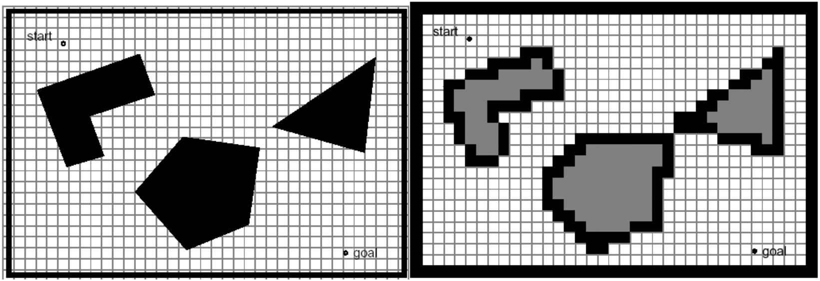


Fig. 4. An office of the institute (a) and the lines representing it in the local metric map (b). The black segments permit to see the correspondence between the two figures.

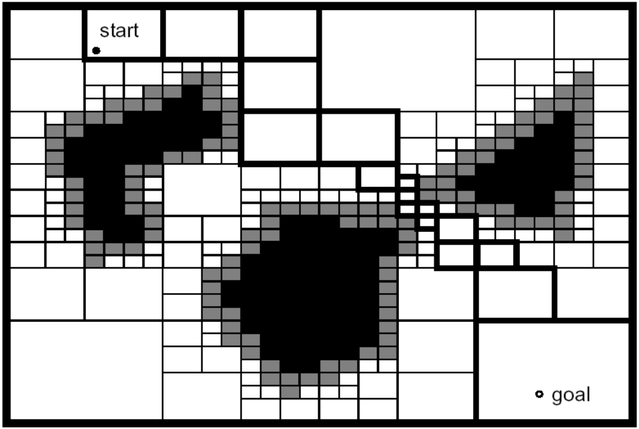
Cons: Requires storage proportional to the number of objects

An **occupancy grid** imposes a grid upon the world, with grid cells corresponding to free space left empty; and those for objects filled



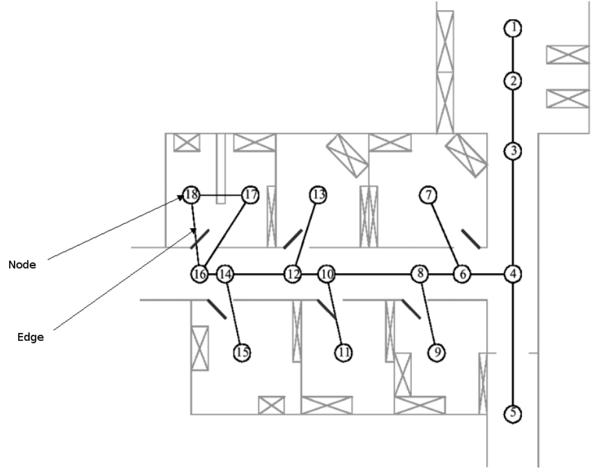
Cons: inexact, size of map grows quickly

An adaptive decomposition can improve the storage efficiency of an occupancy grid



Obtained by recursively splitting occupied grid cells into sub-cells

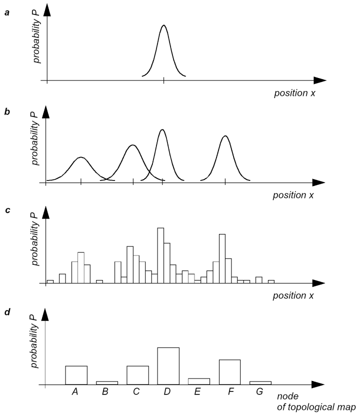
A **topological representation** represents the world as a graph, with *nodes* used to represent positions, and *edges* passable paths between positions



- Topological representations have two main requirements:
  1. A method to detect current position (node)
  2. A method of travelling between nodes (e.g. beacon aiming, corridor centreing, wall following, visual homing,...)
- Visual features particularly useful for (1)
- pros:
  - The *local navigation strategies* required for (2) can be quite simple
  - This representation is lightweight (low memory cost) and well-suited for planning
- cons:
  - Difficulty in maintaining a consistent density of nodes
  - Subject to perceptual aliasing problems

Belief Representation

- We will represent a robot's belief in its position as a probability distribution
- There are a number of ways to represent probability distributions
  - Continuous or discrete
  - Single-hypothesis or multiple-hypothesis
    - Single-hypothesis: The robot believes it is at one position with some margin for error
    - Multiple-hypothesis: The robot can represent the belief that it could be at a number of possible positions



(a) Continuous, single-hypothesis (Gaussian)

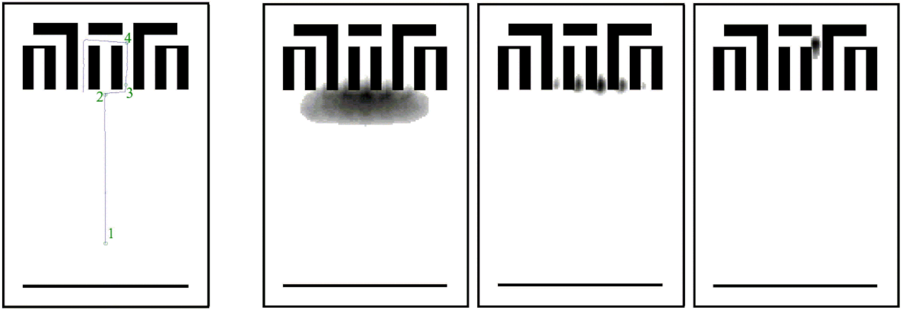
(b) Continuous, multiple-hypothesis (mixture of Gaussians)

(c) Discrete, multiple-hypothesis

(d) Discrete, multiple-hypothesis (for a topological map)

- Single-hypothesis belief:
    - (a) in previous slide
    - Very efficient to update
    - May not be suitable if starting from a position of global uncertainty (i.e. no clue as to where you are)
- Multiple-hypothesis beliefs:
    - (b), (c), or (d) on previous slide
    - A sampled representation of a continuous distribution is also possible (Monte Carlo Localization)
    - Representation is more powerful, but updates can be very expensive

Example of multiple-hypothesis belief tracking using a discrete representation,



Path of the robot *Belief states at positions 2, 3 and 4*

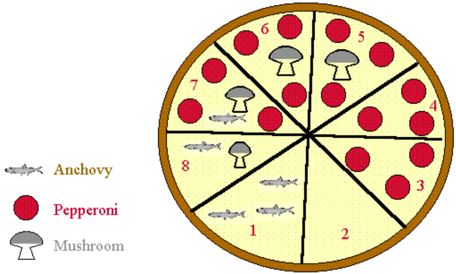
Large initial uncertainty is reduced by new observations

Probability Review

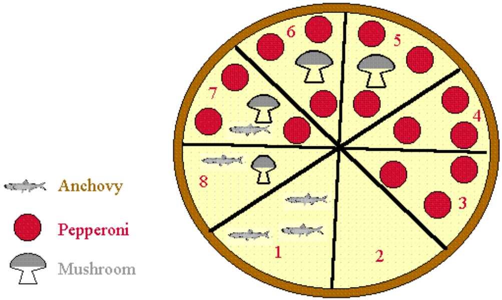
To understand probabilistic localization we will have to review some concepts from probability. We start with **conditional probability**.

Given that some event B has occurred, the probability $p(A|B)$ gives the conditional probability that A has also occurred.

To illustrate, consider the following probability pizza:



Assume that we pick a random slice of pizza with uniform probability...



$p(Pepperoni) = 5/8$

$p(Anchovies) = 3/8$

$p(Anchovies|Pepperoni) = 1/5$

The definition of conditional probability:

$$p(A|B) = \frac{p(A \wedge B)}{p(B)}$$

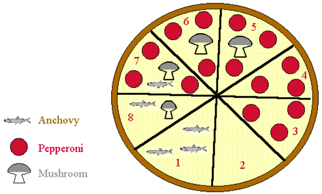
Bayes Rule

If we write the definition of conditional probability as $p(B|A)$ we can obtain the following well-known rule (DERIVATION COVERED ON BOARD),

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

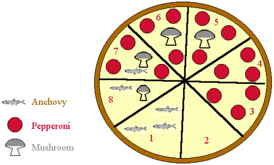
This is called Bayes rule and allows us to calculate a conditional probability (a.k.a. prior probability) from its "inverse"

e.g. What is $p(Pepperoni|Anchovies)$?



$$p(Pe|An) = \frac{p(An|Pe)p(Pe)}{p(An)} = \frac{1}{3}$$

Theorem of Total Probability



$p(Mushrooms) = p(Mu) = 1/2$

If $A_1, \ldots, A_n$ are mutually exclusive and exhaustive events, the theorem of total probability says that:

$$p(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

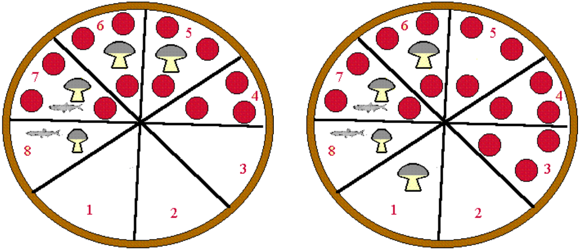
$$\begin{aligned} p(Mu) &= p(Mu|Slice1)p(Slice1) + p(Mu|Slice2)p(Slice2) + \cdots \\ &= 0 \cdot 1/8 + 0 \cdot 1/8 + 0 \cdot 1/8 + 0 \cdot 1/8 + \\ &\quad 1 \cdot 1/8 + 1 \cdot 1/8 + 1 \cdot 1/8 + 1 \cdot 1/8 = 1/2 \end{aligned}$$

Independence

Two events A and B are independent **iff** their joint probability is equal to the product of their individual probabilities

$$p(A \wedge B) = p(A)p(B)$$

For the pizza on the left, events Pe and An are independent: COVERED ON BOARD



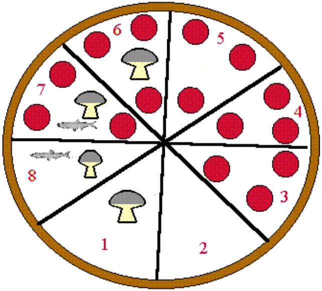
However, for the pizza on the right these events are **not** independent: COVERED ON BOARD

Conditional Independence

Sometimes two events can be independent, but only if some other event is known to have occurred. This is known as **conditional independence** and is expressed by the following relation,

$$p(A \wedge B|C) = p(A|C)p(B|C)$$

For this pizza, if we know that a slice contains mushrooms, the event that it contains anchovies is independent from the event that it contains pepperoni,



References

Siegwart, R. and Nourbakhsh, I. (2004).
Introduction to Autonomous Mobile Robots.
MIT Press.

Tomatis, N., Nourbakhsh, I., and Siegwart, R. (2003).
Hybrid simultaneous localization and map building: a natural integration of topological and metric.
Robotics and Autonomous Systems, 44:3–14.