

CS-417 INTRODUCTION TO ROBOTICS AND INTELLIGENT SYSTEMS

Localization

Fundamental Problems In Robotics

- How to Go From A to B ? (**Path Planning**)
- What does the world looks like? (**mapping**)
 - sense from various positions
 - integrate measurements to produce map
 - assumes perfect knowledge of position
- Where am I in the world? (**localization**)
 - Sense
 - relate sensor readings to a world model
 - compute location relative to model
 - assumes a perfect world model
- Together, the above two are called **SLAM**
(Simultaneous Localization and Mapping)



Localization

- Tracking: Known initial position
- Global Localization: Unknown initial position
- Re-Localization: Incorrect known position
 - (kidnapped robot problem)



Uncertainty

- Central to any real system!



Localization

Initial state
detects nothing:



Moves and
detects landmark:



Moves and
detects nothing:



Moves and
detects landmark:



Sensors

- **Proprioceptive Sensors**

(monitor state of vehicle-propagate)

- IMU (accels & gyros)
- Wheel encoders
- Doppler radar ...
 - Noise



- **Exteroceptive Sensors**

(monitor environment-update)

- Cameras (single, stereo, omni, FLIR ...)
- Laser scanner
- MW radar
- Sonar
- Tactile...
 - Uncertainty



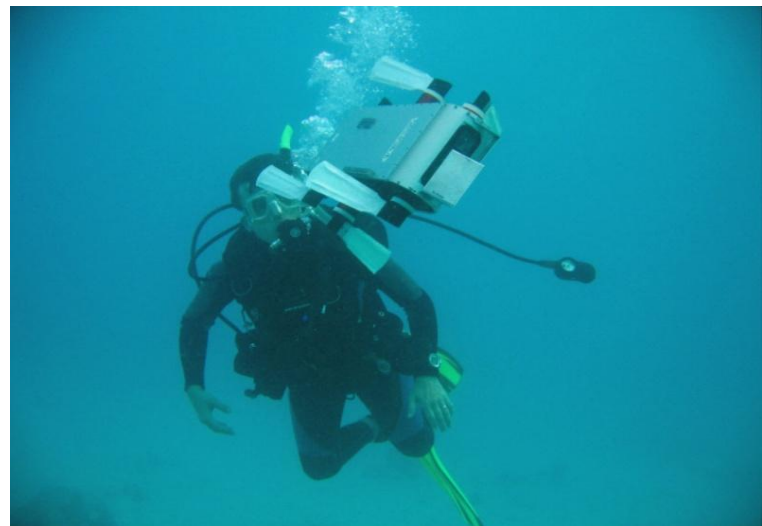
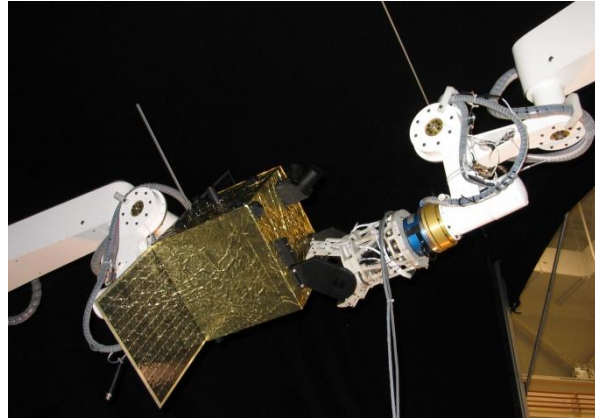
Bayesian Filter

- "**Filtering**" is a name for combining data.
- Nearly all algorithms that exist for spatial reasoning make use of this approach
 - If you're working in robotics, you'll see it over and over!
- Efficient state estimators
 - Recursively compute the robot's current state based on the previous state of the robot



State Estimation

- *What is the robot's state?*
- Depends on the robot
 - Indoor mobile robot
 - $\mathbf{x}=[x, y, \theta]$
 - 6DOF mobile vehicle
 - $\mathbf{x}=[x, y, z, \varphi, \psi, \theta]$
 - Manipulators



Bayesian Filter

- Estimate state x from data Z
 - *What is the probability of the robot being at x ?*
- x could be robot location, map information, locations of targets, etc...
- Z could be sensor readings such as range, actions, odometry from encoders, etc...)
- This is a general formalism that does not depend on the particular probability representation
- Bayes filter **recursively** computes the posterior distribution:

$$Bel(x_T) = P(x_T | Z_T)$$



Derivation of the Bayesian Filter

Estimation of the robot's state given the data:

$$Bel(x_t) = p(x_t | Z_T)$$

The robot's data, Z , is expanded into two types: observations o_i and actions a_i

$$Bel(x_t) = p(x_t | o_t, a_{t-1}, o_{t-1}, a_{t-2}, \dots, o_0)$$

Invoking the Bayesian theorem

$$Bel(x_t) = \frac{p(o_t | x_t, a_{t-1}, \dots, o_0) p(x_t | a_{t-1}, \dots, o_0)}{p(o_t | a_{t-1}, \dots, o_0)}$$



Derivation of the Bayesian Filter

Denominator is constant relative to x_t

$$\eta = 1 / p(o_t | a_{t-1}, \dots, o_0)$$

$$Bel(x_t) = \eta p(o_t | x_t, a_{t-1}, \dots, o_0) p(x_t | a_{t-1}, \dots, o_0)$$

First-order Markov assumption shortens first term:

$$Bel(x_t) = \eta p(o_t | x_t) p(x_t | a_{t-1}, \dots, o_0)$$

Expanding the last term (theorem of total probability):

$$Bel(x_t) = \eta p(o_t | x_t) \int p(x_t | x_{t-1}, a_{t-1}, \dots, o_0) p(x_{t-1} | a_{t-1}, \dots, o_0) dx_{t-1}$$



Reminder: Bayes Rule

- Conditional probabilities

$$p(o \wedge S) = p(o | S) p(S)$$

- Bayes rule relates conditional probabilities

$$p(o | S) = \frac{p(S | o) p(o)}{p(S)} \quad \text{Bayes rule}$$

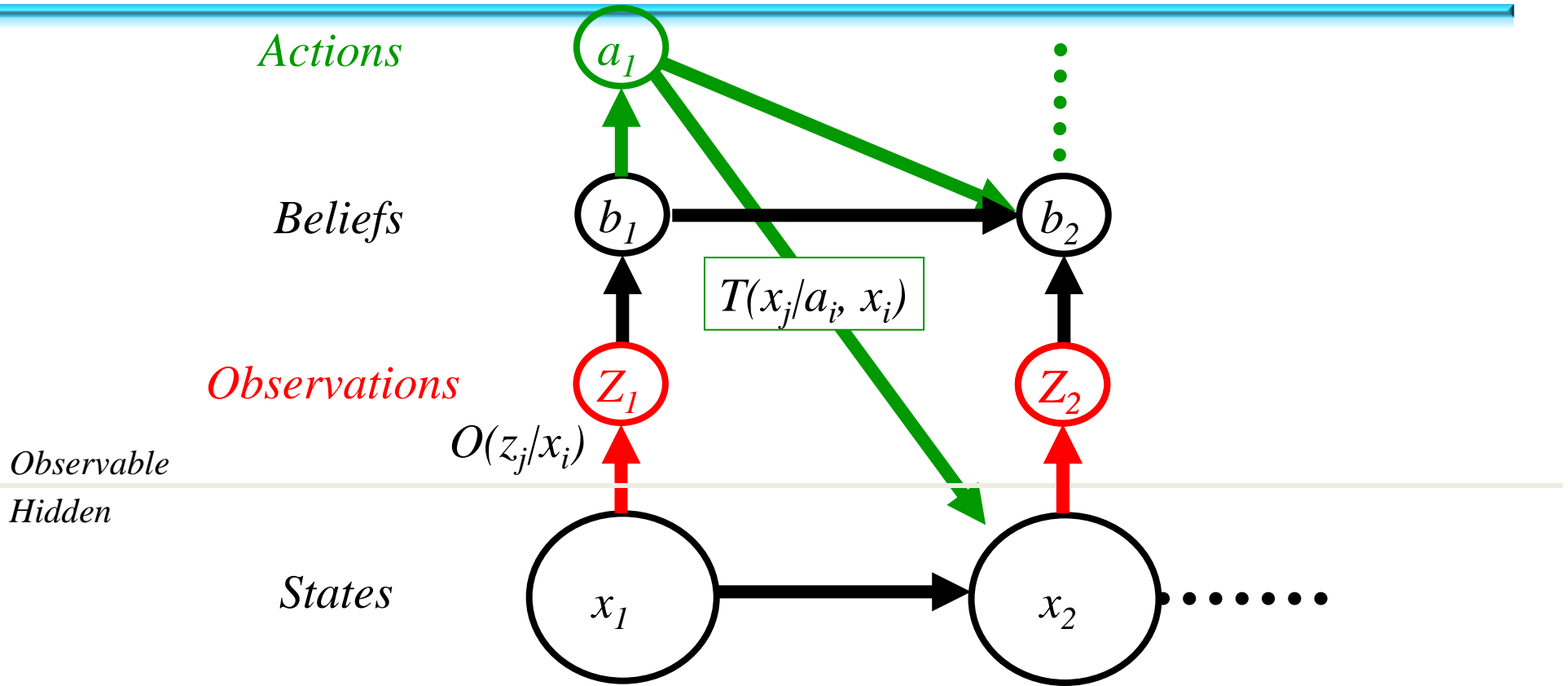
- So, what does this say about $\text{odds}(o | S_2 \wedge S_1)$?

Can we update easily ?

$$p(a | b, c) = \frac{p(b | a, c) p(a | c)}{p(b | c)}$$



Graphical Models, Bayes' Rule and the Markov Assumption



Bayes rule:
$$p(x | y) = \frac{p(y | x) p(x)}{p(y)}$$

Markov:
$$p(x_t | x_{t-1}, a_t, a_0, z_0, a_1, z_1, \dots, z_{t-1}) = p(x_t | x_{t-1}, a_t)$$



Derivation of the Bayesian Filter

First-order Markov assumption shortens middle term:

$$Bel(x_t) = \eta \boxed{p(o_t | x_t)} \int p(x_t | x_{t-1}, a_{t-1}) p(x_{t-1} | a_{t-1}, \dots, o_0) dx_{t-1}$$

Finally, substituting the definition of $Bel(x_{t-1})$:

$$Bel(x_t) = \eta p(o_t | x_t) \int p(x_t | x_{t-1}, a_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

The above is the probability distribution that must be estimated from the robot's data



Iterating the Bayesian Filter

- Propagate the motion model:

$$Bel_{-}(x_t) = \int P(x_t | a_{t-1}, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

Compute the current state estimate before taking a sensor reading by integrating over all possible previous state estimates and applying the motion model

- Update the sensor model:

$$Bel(x_t) = \eta P(o_t | x_t) Bel_{-}(x_t)$$

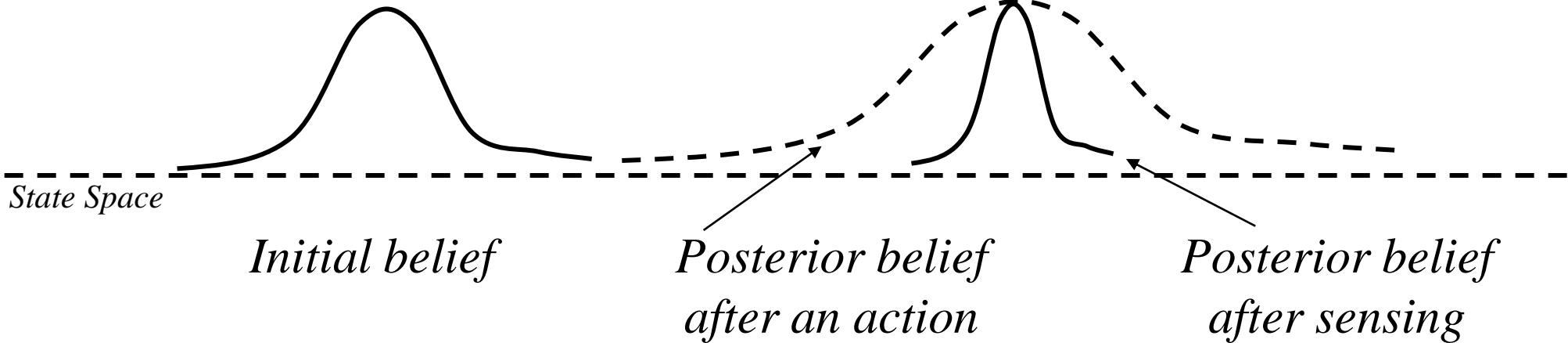
Compute the current state estimate by taking a sensor reading and multiplying by the current estimate based on the most recent motion history



Bayes Filter



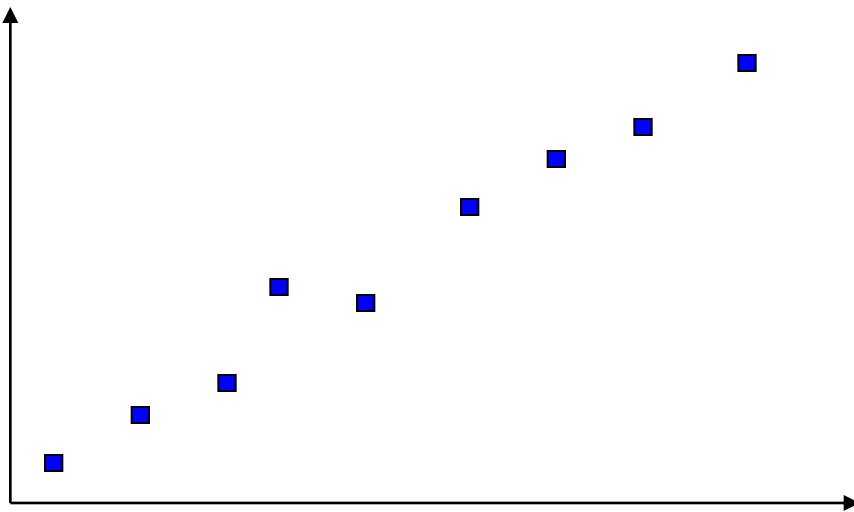
*An action
is taken*



Representation of the Belief Function

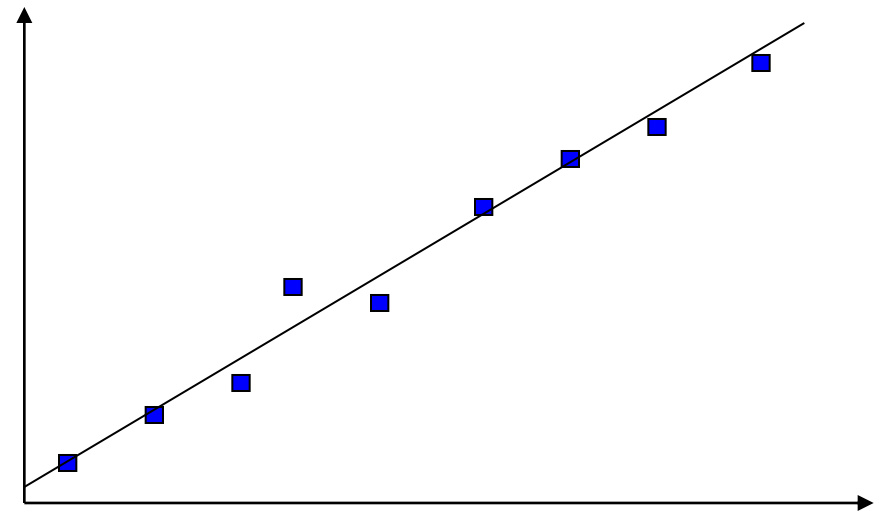
Sample-based
representations

e.g. Particle filters



$(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)$

Parametric
representations



$y = mx + b$



Different Approaches

Kalman filters (Early-60s?)

- Gaussians
- approximately linear models
- position tracking

Extended Kalman Filter

Information Filter

Unscented Kalman Filter

Multi-hypothesis ('00)

- Mixture of Gaussians
- Multiple Kalman filters
- Global localization, recovery

Discrete approaches ('95)

- Topological representation ('95)
- Uncertainty handling (POMDPs)
- occas. global localization, recovery
- Grid-based, metric representation ('96)
- global localization, recovery

Particle filters ('98)

- Condensation (Isard and Blake '98)
- Sample-based representation
- Global localization, recovery
- Rao-Blackwellized Particle Filter



Bayesian Filter : Requirements for Implementation

- Representation for the belief function
- Update equations
- Motion model
- Sensor model
- Initial belief state

