Principles of Robot Autonomy I

Occupancy grid mapping , Frontier Exploration,

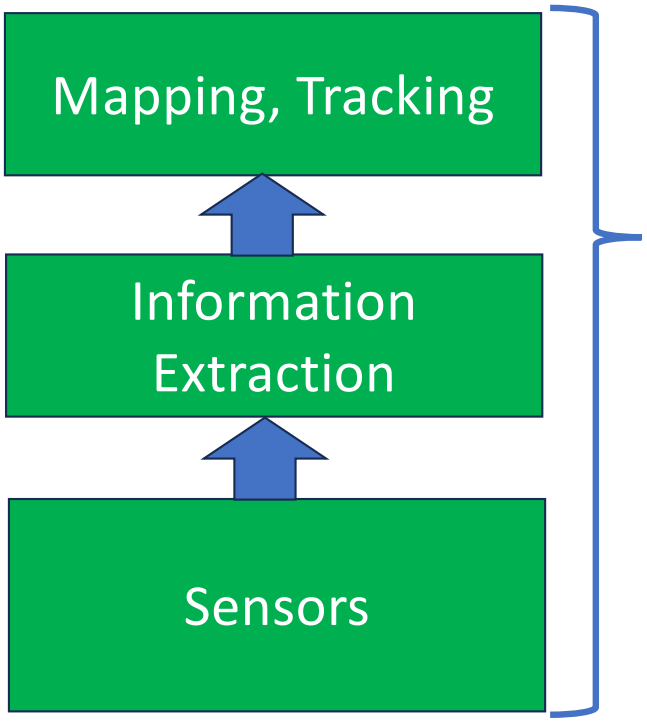
Bayesian estimation and filtering,

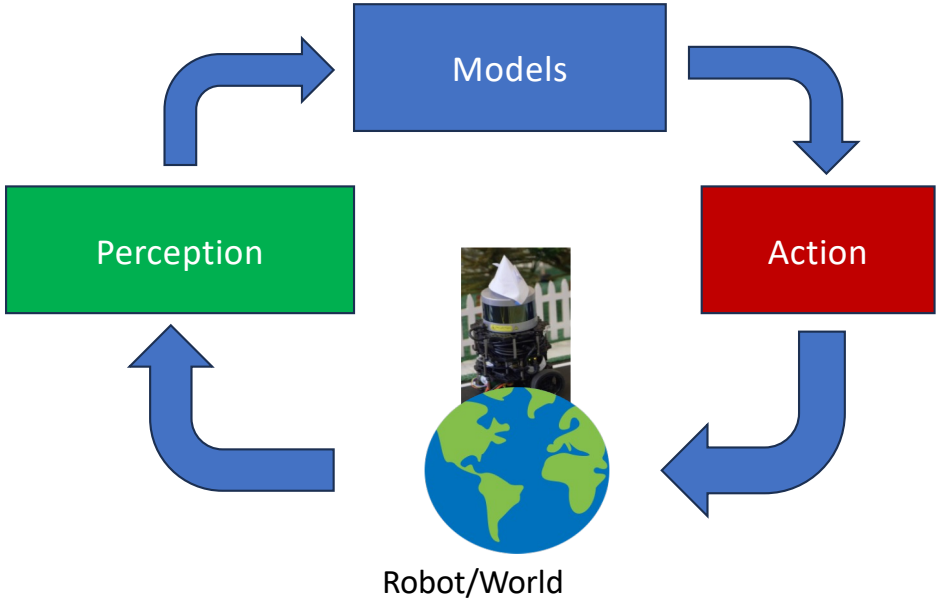
Logistics

- Homework 4: due Thurs 11/13
- Homework 5: out Thurs 11/13, Due 12/2 (last one!)
- Guest Lecture: Dr. Vincent Vincent Vanhoucke, Distinguished Engineer, Waymo, Thurs 11/20
- Midterm grades out soon
- Lecture 14:
  - Occ grid mapping
  - Frontier exploration, information gathering
  - General Bayesian filter

Robot Perception

Perception Stack



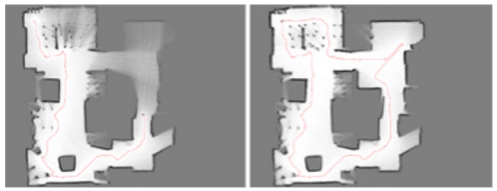


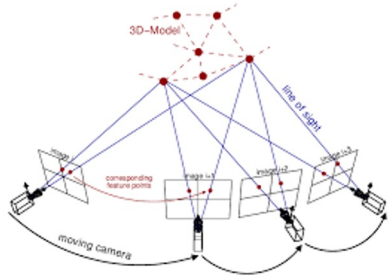
Perception Stack: Computer vision, filtering, SLAM

Use sensor data to update the models

Localization, Mapping, Tracking:

- EKF/Monte Carlo localization
- Occupancy grid mapping
- Factor graphs/SLAM
- Tracking (EKF and Particle Filter)
- **AA273: Filtering (Schwager)**
- **AA275: Navigation (Gao)**



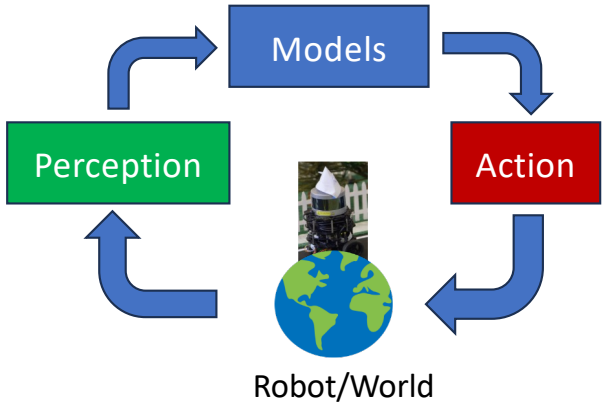


Information extraction

- Computer vision: features, correspondences, Structure from Motion (SfM), depth
- Lidar scan matching, ICP
- **CS231A: Comp Vision**

Sensors:

- RGB Cameras, RGB-D/stereo cameras, Lidar
- IMU, GPS, wheel encoders

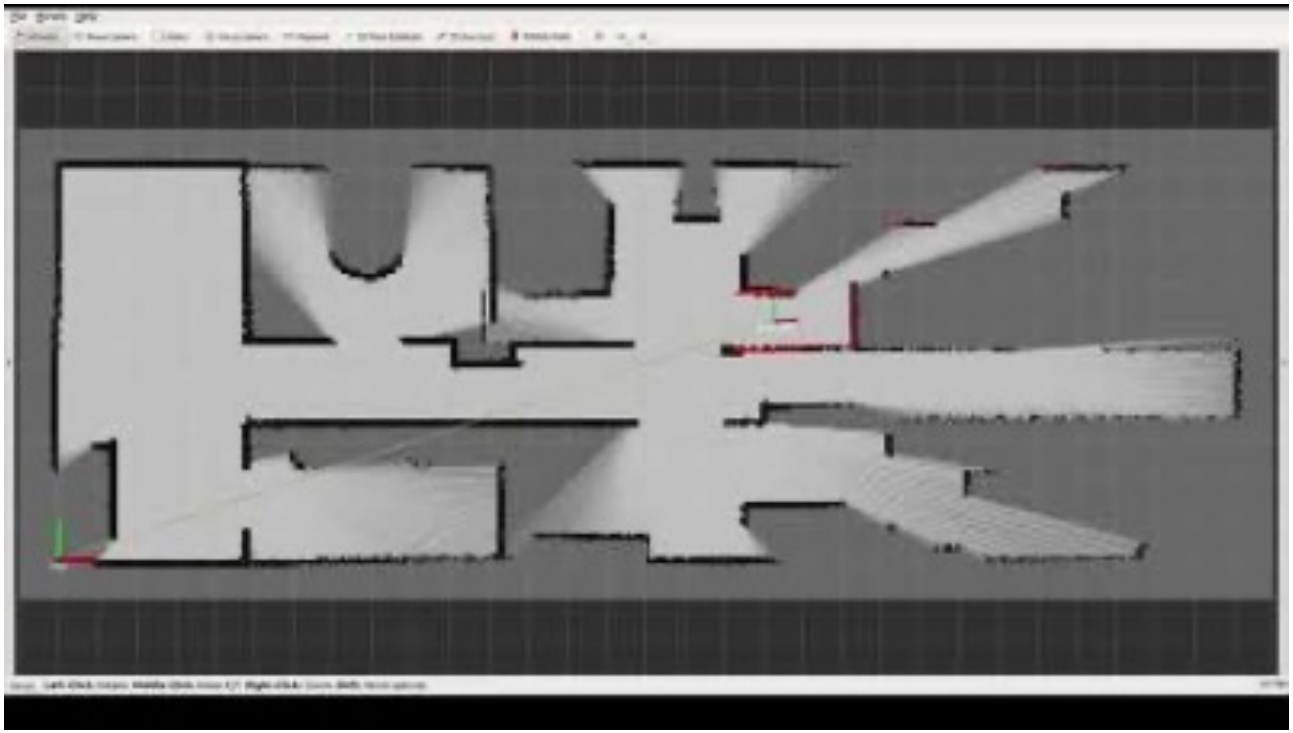


Today's lecture

- Aim
  - Learn basic concepts about Bayesian filtering and apply to occupancy grid mapping and Frontier Exploration.

- Readings
  - S. Thrun, W. Burgard, and D. Fox. Probabilistic robotics. MIT press, 2005. Chapter 2, Chapter 9

Example: Occupancy grid mapping



Probabilistic occupancy grid mapping

Occupancy map: $m \in \{0,1\}^{M \times N}$

Single grid cell: $m_{i,j} \in \{0,1\}$

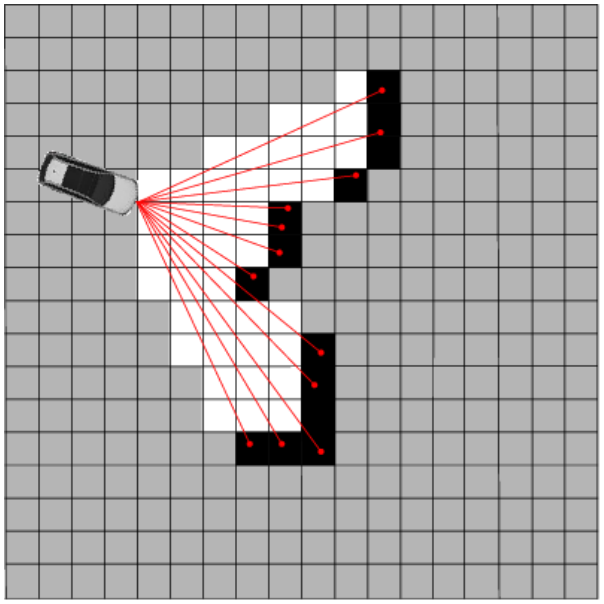
Measurement model (Lidar, stereo, vision):

See point in cell — Cell occupied

$$p(z_t(i,j) \mid m(i,j)) = \begin{bmatrix} p(0 \mid 0) & p(0 \mid 1) \\ p(1 \mid 0) & p(1 \mid 1) \end{bmatrix}$$

True negative — False negative

False positive — True positive



Cell (i,j) only gets a measurement if it is inside the footprint of sensor

Footprint depends on robot pose (assumed known)

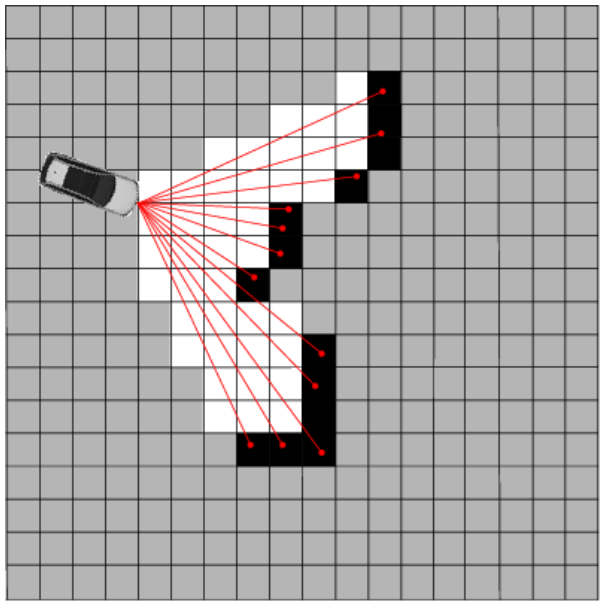
Probabilistic occupancy grid mapping

E.g.

$$p(z(i,j) \mid m(i,j)) = \begin{bmatrix} 0.9 & 0.05 \\ 0.1 & 0.95 \end{bmatrix}$$

True negative — False negative

False positive — True positive



Goal:

Find probability distribution over each occupancy cell given sequence of measurements

$$p(m(i,j) \mid z_{1:t}(i,j))$$

Sequence of measurements from 1 to t

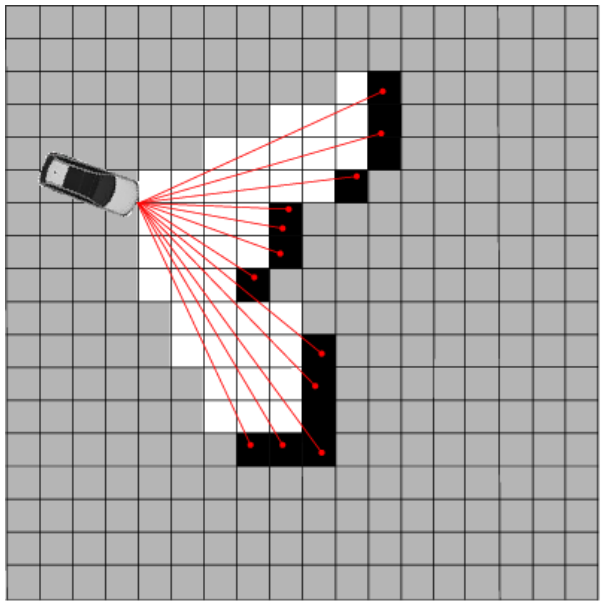
Probabilistic occupancy grid mapping

Recall:

$$p(m(i,j) = 1 \mid z_{1:t}(i,j)) \in [0,1]$$

Called a binary random variable,
or a Bernoulli random variable

Each cell in the map is a *probability*
Ground truth map is binary, but unknown.



Bayes' rule and probabilistic inference

- Assume *x* is a quantity we would like to infer from *y*
- Bayes rule allows us to do so through the inverse probability, which specifies the probability of data *y* assuming that *x* was the cause

Posterior probability distribution

Prior probability distribution

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{\int p(y \mid x')p(x')\,dx'}$$

Data

Normalizer, does not depend on $x := \eta^{-1}$

- Notational simplification

$$p(x \mid y) = \eta\, p(y \mid x)p(x)$$

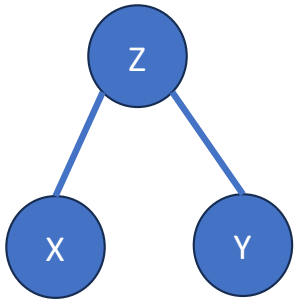
Independence vs Conditional Independence

- The following are equivalent: X and Y are independent

$$p(x, y) = p(x)p(y)$$
$$p(x \mid y) = p(x)$$
$$p(y \mid x) = p(y)$$

- *X* and *Y* are conditionally independent given *Z*

$$p(x, y \mid z) = p(x \mid z)p(y \mid z)$$
$$p(x \mid y, z) = p(x \mid z)$$
$$p(y \mid x, z) = p(y \mid z)$$



- "Puddles" and "umbrellas" given "raining"

Sequential Bayes' with Conditionally Indepenedent Measurements

Sequence of measurements: $z_{1:t}$

Unknown map state: x

Conditionally Indep. measurements: $p(z_{1:t} \mid x) = \prod_{\tau=1}^{t} p(z_\tau \mid x)$

Bayes' Rule: $p(x \mid z_{1:t}) = \eta_t \prod_{\tau=1}^{t} p(z_\tau \mid x) p(x)$

Sequential Bayes': $p(x \mid z_{1:t}) = \frac{\eta_t}{\eta_{t-1}} p(z_t \mid x) p(x \mid z_{1:t-1})$

Occupancy Grid Bayesian Update for Cell (i,j)

Occ Prob at t (unnormalized)

Occ Prob at t-1

$$\bar{p}(m(i,j) \mid z_{1:t}(i,j)) = p\big(z_t(i,j) \mid m(i,j)\big)p\big(m(i,j) \mid z_{1:t-1}(i,j)\big)$$

See point in cell

Cell occupied

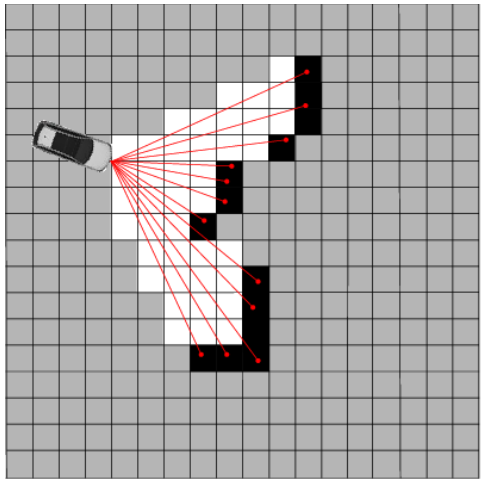
True negative

False negative

$$p(z_t(i,j) \mid m(i,j)) = \begin{bmatrix} p(0 \mid 0) & p(0 \mid 1) \\ p(1 \mid 0) & p(1 \mid 1) \end{bmatrix}$$

False positive

True positive



Occ Prob at t (normalized)

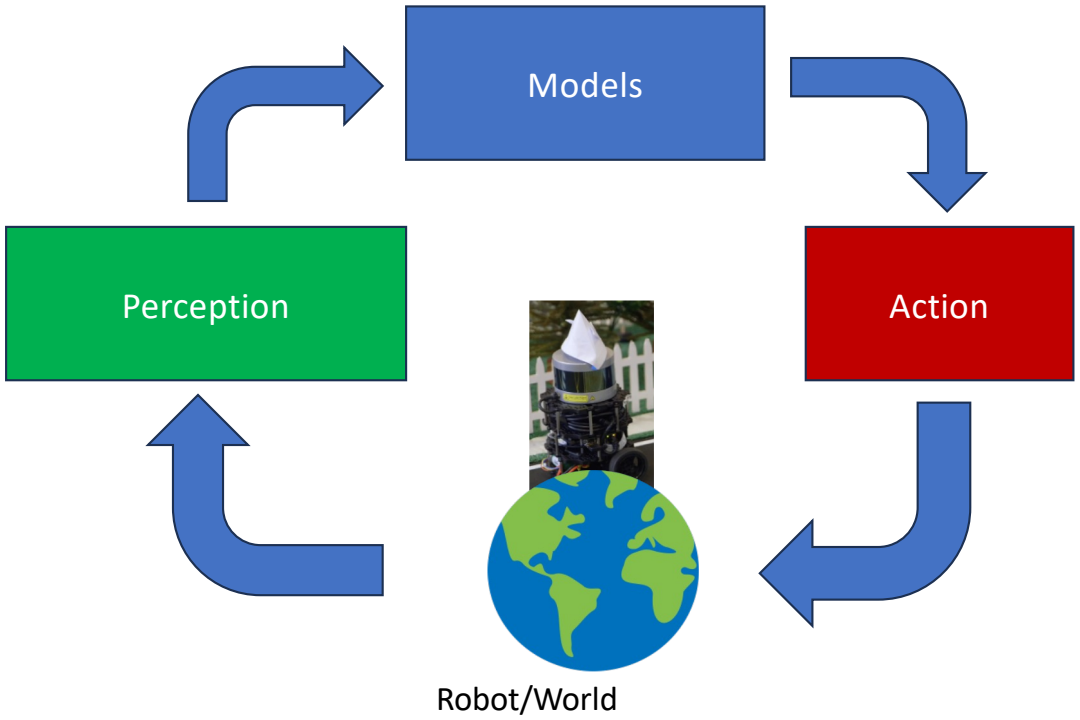
$$p(m(i,j) \mid z_{1:t}(i,j)) = \frac{\bar{p}\big(m(i,j) \mid z_{1:t-1}(i,j)\big)}{\bar{p}\big(m(i,j) = 0 \mid z_{1:t-1}(i,j)\big) + \bar{p}\big(m(i,j) = 1 \mid z_{1:t-1}(i,j)\big)}$$

Expectation of a RV

- Expectation for discrete RVs: $E[X] = \sum_x x\, p(x)$
- Expectation for continuous RVs: $E[X] = \int x\, p(x)\, dx$
- Expectation is a linear operator: $E[aX + b] = a\, E[X] + b$
- Expectation of a vector of RVs is simply the vector of expectations
- Covariance

$$\text{cov}(X, Y) = E[(X - E[X])(Y - E[Y])^T] = E[XY^T] - E[X]E[Y]^T$$

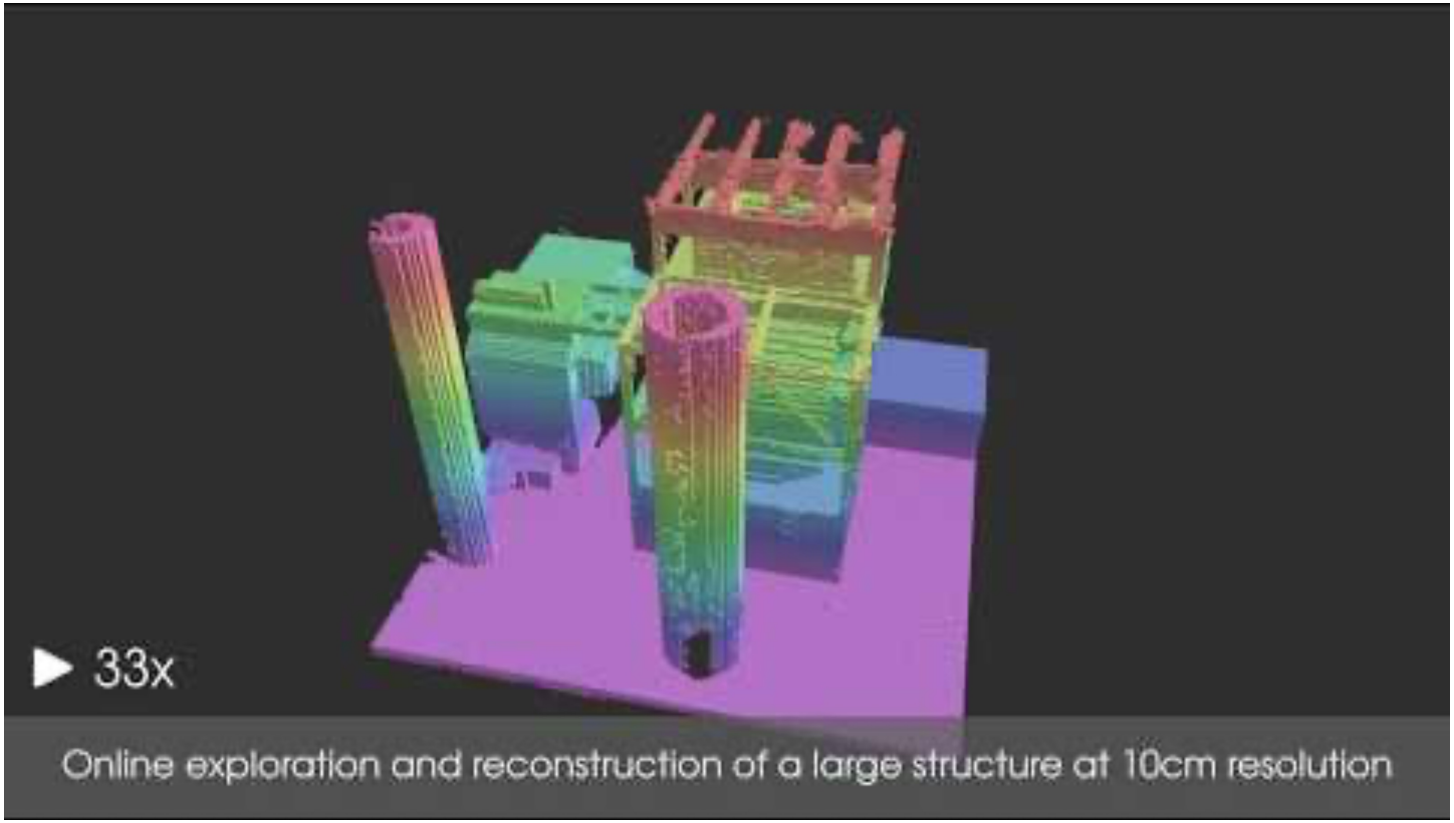
Active Perception as a Planning Objective



Active Perception:
Take actions to improve perception performance

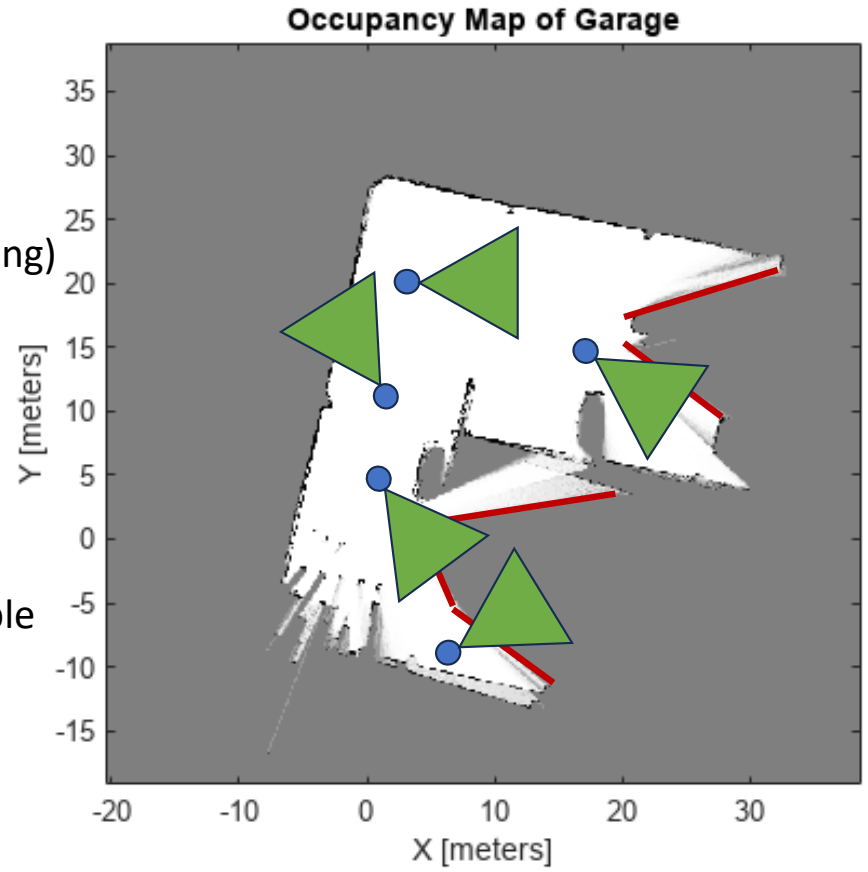
- Exploration (to improve a map)
- Informative planning (to improve robot pose estimate)
- Target following (to improve target pose estimate)

Frontier Exploration



Frontier Exploration

- Frontier points:
  - High uncertainty (prob occ ~0.5)
  - Neighboring other cells with low uncertainty
- Consider robot pose candidates (e.g. from sampling)
- Score poses based on:
  - Number of frontier cells in footprint
  - Length of path to get there
- Pick the best and execute
- Called next-best-view planning, myopic planning, Greedy planning
- Other methods solve an optimization over multiple future view points
- “Orienteering problem”



Information gathering, informative planning, active sensing, belief space planning

- Probabilistic approach
- Quantify “information gain” from a Bayesian estimate
- Can be
  - A map
  - Target state
  - Ego state of the robot
- Turn into trajectory optimization problem: find sequence of control inputs to maximize expected information gain over a time horizon
- Main advantage: general unifying approach
- Main drawback: tends to be computationally expensive

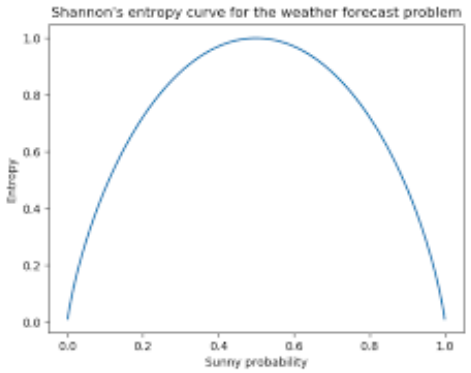
Information gathering, informative planning, active sensing, belief space planning

Shannon Entropy: measure of uncertainty in a RV

Discrete: $$H(X) = -\sum_x \log_2(p(x))p(x)$$

Continuous: $$H(X) = -\int_x \ln(p(x))p(x)dx$$

E.g., binary RV:



E.g., Gaussian RV:

$$H(X) = C + \frac{1}{2}\ln\det(\Sigma)$$

Information gathering, informative planning, active sensing, belief space planning

Conditional Entropy: Expected entropy after taking a measurement

$$H(X \mid Y) = \mathbb{E}[H(X) \mid Y] = \int_y \left(- \int_x p(x \mid y) \ln(p(x \mid y))\, dx \right) p(y)\, dy$$

$$= - \int_y \int_x p(y \mid x) p(x) \ln\left(\frac{p(y \mid x) p(x)}{p(y)}\right) dx dy$$

Measurement likelihood
Depends on robot state

Prior

Mutual information: Expected entropy reduction after taking a measurement

$$I(X, Y) = H(X) - H(X \mid Y)$$

Informative planning: Plan trajectory for robot to maximize mutual information

$$\max_{u_{1:T}} \sum_{t=1}^{T} I(X, Y; s_t) - u_t^T R u_t$$

$$\text{s.t.} \quad s_{t+1} = f(s_t, u_t) \quad \text{Dynamics}$$

$$g(s_t) \leq 0 \quad \text{Collision constraints}$$

$$h(u_t) \leq 0 \quad \text{Input constraints}$$

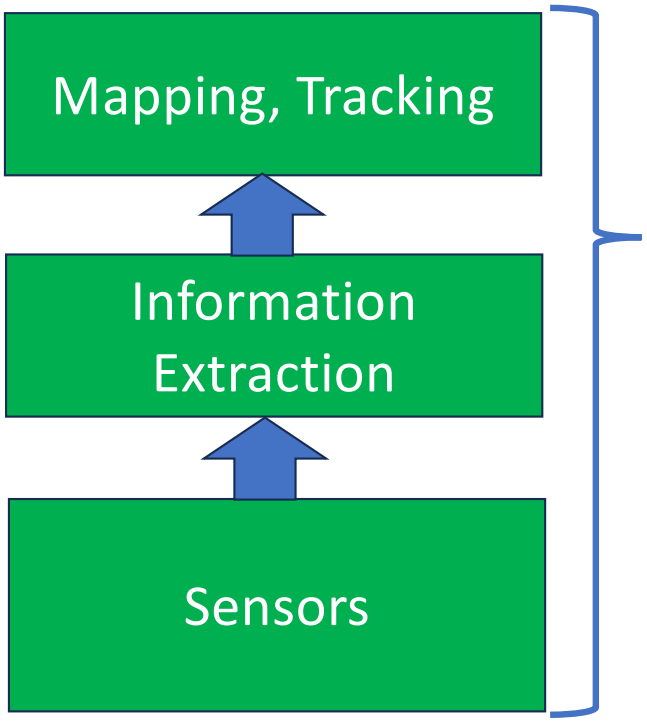
Challenging to solve in practice!

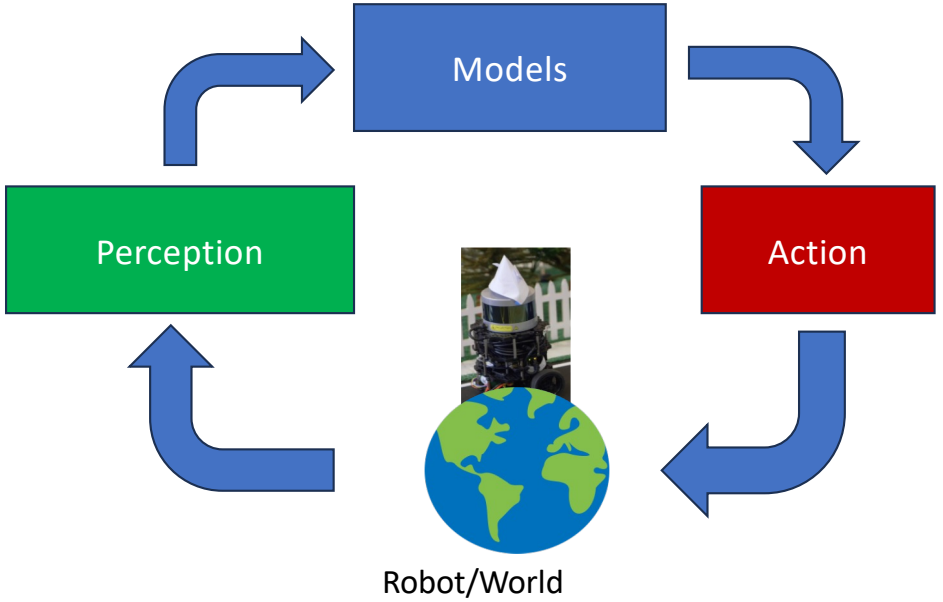
Information gathering, informative planning, active sensing, belief space planning

- Highly versatile and general
- Works for any problem posed as a Bayesian filter
  - Special variants for occ maps, EKFs, particle filters, UKFs
  - Special variants for map exploration, robot ego state estimation, active target tracking and target search, active SLAM
- Drawback: usually requires approximations and computational shortcuts

Robot Perception

Perception Stack





Model for robot-environment interaction

- Two fundamental types of robot-environment interactions: the robot can influence the state of its environment through control actions, and gather information about the state through measurements
- State x_t: collection at time *t* of all aspects of the robot and its environment that can impact the future
  - Map (location of fixed objects in environment)
  - Robot pose (e.g., robot location and orientation)
  - Robot velocity
  - State of dynamics objects in environment
- Useful notation: $x_{t_1:t_2} := x_{t_1}, x_{t_1+1}, x_{t_1+2}, \ldots, x_{t_2}$
- A state x_t is called *complete* if no variables prior to x_t can influence the evolution of future states → Markov property

Measurement and control data

- Measurement data z_t: information about state of the environment at time *t*; useful notation

$$z_{t_1:t_2} := z_{t_1}, z_{t_1+1}, z_{t_1+2}, \ldots, z_{t_2}$$

- Control data u_t: information about the change of state at time *t*; useful notation

$$u_{t_1:t_2} := u_{t_1}, u_{t_1+1}, u_{t_1+2}, \ldots, u_{t_2}$$

- Key difference: measurement data tends to increase robot's knowledge, while control actions tend to induce a loss of knowledge

State equation

- General probabilistic generative model

Convention: first take control action and then take measurement

$$p(x_t \mid x_{0:t-1},\, z_{1:t-1},\, u_{1:t})$$

- Key assumption: state is complete (i.e., the Markov property holds)

State transition probability

$$p(x_t \mid x_{0:t-1},\, z_{1:t-1},\, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$$

- In other words, we assume *conditional independence*, with respect to conditioning on x_{t-1}
- Special case (typical dynamics model):

$$x_t = f(x_{t-1}, u_{t-1}) + w_{t-1}, \quad w_{t-1} \sim \mathcal{N}(0, Q_{t-1})$$

Measurement equation and overall stochastic model

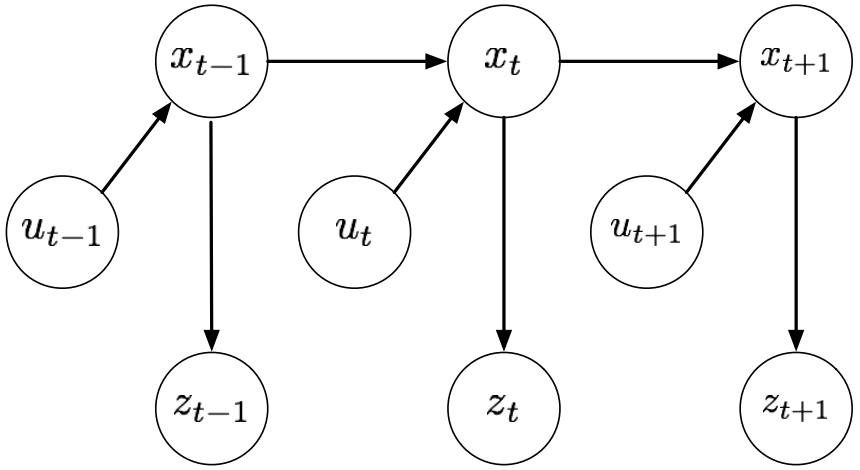
- Assuming x_t is complete

$$p(z_t \mid x_{0:t},\ z_{1:t-1},\ u_{1:t}) = p(z_t \mid x_t)$$

Measurement probability

- Overall dynamic Bayes network model (also referred to as hidden Markov model)
- Special case (typical measurement model):

$$z_t = g(x_t, u_t) + v_t, \quad v_t \sim \mathcal{N}(0, R_t)$$



Belief distribution

- Belief distribution: reflects internal knowledge about the state
- A belief distribution assigns a probability to each possible hypothesis about the true state
- Formally, belief distributions are posterior probabilities over state variables conditioned on the available data

$$bel(x_t) := p(x_t \mid z_{1:t}, u_{1:t})$$

- Similarly, the *prediction* distribution is defined as

$$\overline{bel}(x_t) := p(x_t \mid z_{1:t-1}, u_{1:t})$$

- Calculating $bel(x_t)$ from $\overline{bel}\ (x_t)$ is called correction or measurement update

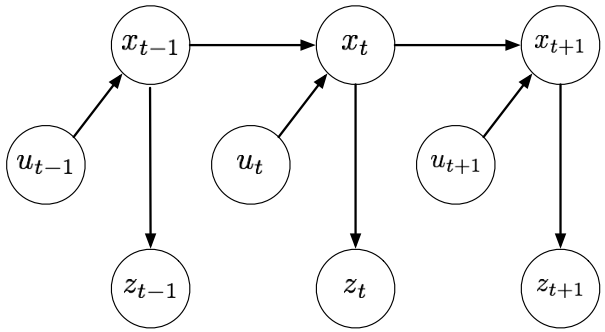
Bayes filter algorithm

- Bayes' filter algorithm: most general algorithm for calculating beliefs
- Key assumption: state is complete

- Recursive algorithm
  - Step 1 (prediction): compute $\overline{bel}\ (x_t)$
  - Step 2 (measurement update): compute $bel(x_t)$
- Algorithm initialized with $bel(x_0)$ (e.g., uniform or points mass)

Data: $bel(x_{t-1}), u_t, z_t$
Result: $bel(x_t)$
foreach x_t **do**
  $\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\, bel(x_{t-1})\, dx_{t-1}$;
  $bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t)$;
end
Return $bel(x_t)$

Derivation: measurement update



$$bel(x_t) = p(x_t \mid z_{1:t}, u_{1:t})$$

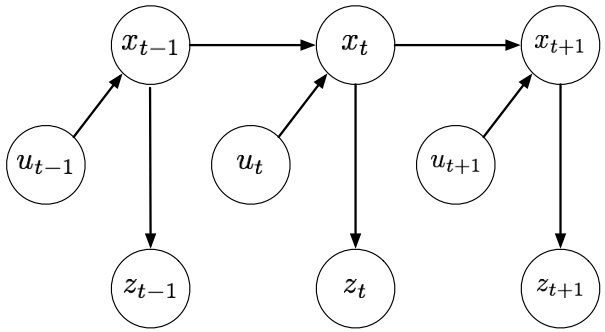
$$= \frac{p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t})}{\underbrace{p(z_t \mid z_{1:t-1}, u_{1:t})}_{:=\eta^{-1}}}$$

Bayes rule

$$= \eta\, p(z_t \mid x_t)\, \underbrace{p(x_t \mid z_{1:t-1}, u_{1:t})}_{=\overline{bel(x_t)}}$$

Markov property

Derivation: prediction step



$$\overline{bel}(x_t) = p(x_t \mid z_{1:t-1},\, u_{1:t})$$

$$= \int p(x_t \mid x_{t-1},\, z_{1:t-1},\, u_{1:t})\, p(x_{t-1} \mid z_{1:t-1},\, u_{1:t})\, dx_{t-1}$$

Total probability

$$= \int p(x_t \mid x_{t-1},\, u_t)\, p(x_{t-1} \mid z_{1:t-1},\, u_{1:t})\, dx_{t-1}$$

Markov

$$= \int p(x_t \mid x_{t-1},\, u_t)\, p(x_{t-1} \mid z_{1:t-1},\, u_{1:t-1})\, dx_{t-1}$$

For general output feedback policies, u_t does not provide additional information on x_{t-1}

$$= \int p(x_t \mid x_{t-1},\, u_t)\, bel(x_{t-1})\, dx_{t-1}$$

Discrete Bayes' filter

- Discrete Bayes' filter algorithm: applies to problems with *finite* state spaces

- Belief $bel(x_t)$ represented as pmf $\{p_{k,t}\}$

Data: $\{p_{k,t-1}\}, u_t, z_t$
Result: $\{p_{k,t}\}$
foreach k **do**
  $\bar{p}_{k,t} = \sum_i p(X_t = x_k \mid u_t, X_{t-1} = x_i)\, p_{i,t-1}$;
  $p_{k,t} = \eta\, p(z_t \mid X_t = x_k)\, \bar{p}_{k,t}$;
end
Return $\{p_{k,t}\}$

Simple example – Belief & Measurement Model

Adapted from [PR]

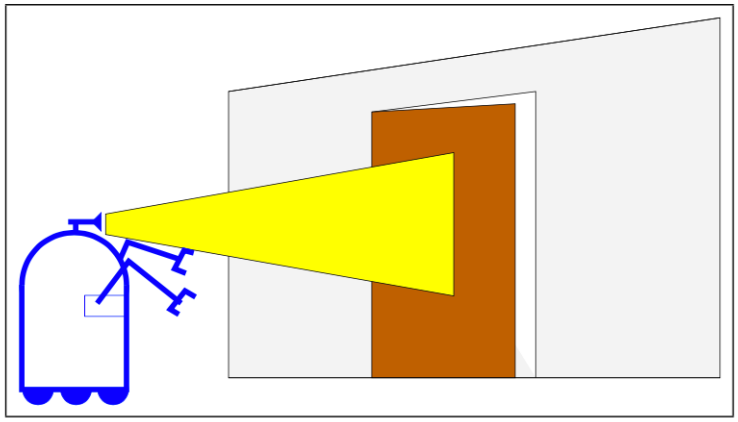


Figure 2.2 A mobile robot estimating the state of a door.

$$bel(X_0 = \textbf{open}) = 0.5$$
$$bel(X_0 = \textbf{closed}) = 0.5$$

$$p(Z_t = \textbf{sense\_open} \mid X_t = \textbf{is\_open}) = 0.6$$
$$p(Z_t = \textbf{sense\_closed} \mid X_t = \textbf{is\_open}) = 0.4$$

$$p(Z_t = \textbf{sense\_open} \mid X_t = \textbf{is\_closed}) = 0.2$$
$$p(Z_t = \textbf{sense\_closed} \mid X_t = \textbf{is\_closed}) = 0.8$$

Simple example – Transition Model

Adapted from [PR]

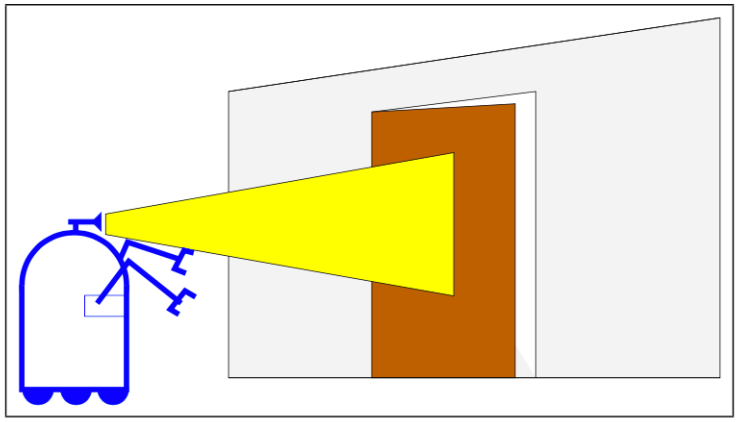


Figure 2.2 A mobile robot estimating the state of a door.

$$
\begin{aligned}
p(X_t = \textbf{is\_open} \mid U_t = \textbf{push}, X_{t-1} = \textbf{is\_open}) &= 1 \\
p(X_t = \textbf{is\_closed} \mid U_t = \textbf{push}, X_{t-1} = \textbf{is\_open}) &= 0 \\
p(X_t = \textbf{is\_open} \mid U_t = \textbf{push}, X_{t-1} = \textbf{is\_closed}) &= 0.8 \\
p(X_t = \textbf{is\_closed} \mid U_t = \textbf{push}, X_{t-1} = \textbf{is\_closed}) &= 0.2
\end{aligned}
\qquad
\begin{aligned}
p(X_t = \textbf{is\_open} \mid U_t = \textbf{do\_nothing}, X_{t-1} = \textbf{is\_open}) &= 1 \\
p(X_t = \textbf{is\_closed} \mid U_t = \textbf{do\_nothing}, X_{t-1} = \textbf{is\_open}) &= 0 \\
p(X_t = \textbf{is\_open} \mid U_t = \textbf{do\_nothing}, X_{t-1} = \textbf{is\_closed}) &= 0 \\
p(X_t = \textbf{is\_closed} \mid U_t = \textbf{do\_nothing}, X_{t-1} = \textbf{is\_closed}) &= 1
\end{aligned}
$$

Simple Example – Update Step

Adapted from [PR]

- Is door open or not?

$$bel(X_0 = \textbf{open}) = 0.5$$
$$bel(X_0 = \textbf{closed}) = 0.5$$

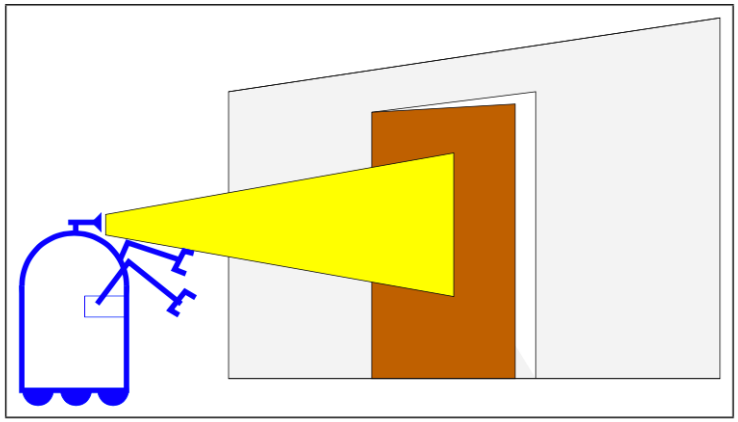


Figure 2.2 A mobile robot estimating the state of a door.

Measurement Model

$$p(Z_t = \textbf{sense\_open} \mid X_t = \textbf{is\_open}) = 0.6$$
$$p(Z_t = \textbf{sense\_closed} \mid X_t = \textbf{is\_open}) = 0.4$$

$$p(Z_t = \textbf{sense\_open} \mid X_t = \textbf{is\_closed}) = 0.2$$
$$p(Z_t = \textbf{sense\_closed} \mid X_t = \textbf{is\_closed}) = 0.8$$

Transition Model for do_nothing

$$p(X_t = \textbf{is\_open} \mid U_t = \textbf{do\_nothing}, X_{t-1} = \textbf{is\_open}) = 1$$
$$p(X_t = \textbf{is\_closed} \mid U_t = \textbf{do\_nothing}, X_{t-1} = \textbf{is\_open}) = 0$$
$$p(X_t = \textbf{is\_open} \mid U_t = \textbf{do\_nothing}, X_{t-1} = \textbf{is\_closed}) = 0$$
$$p(X_t = \textbf{is\_closed} \mid U_t = \textbf{do\_nothing}, X_{t-1} = \textbf{is\_closed}) = 1$$

Simple Example – Update Step

Adapted from [PR]

- Is door open or not?

$$bel(X_0 = \textbf{open}) = 0.5$$
$$bel(X_0 = \textbf{closed}) = 0.5$$

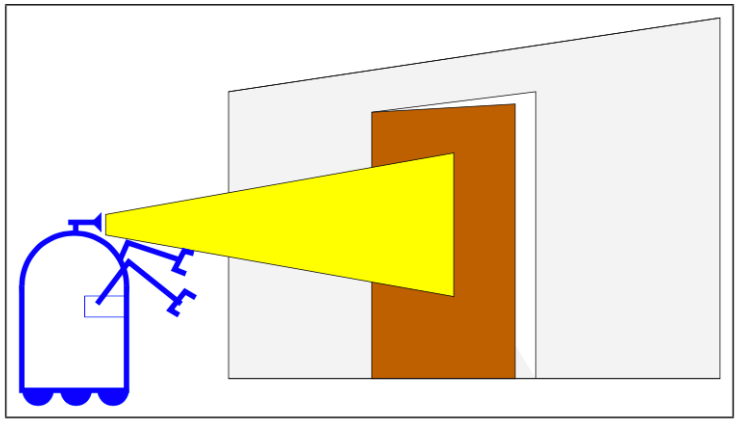


Figure 2.2 A mobile robot estimating the state of a door.

Received Sensor Measurement:

$$bel(x_1) = \eta\, p(Z_1 = \textbf{sense\_open} \mid x_1)\, \overline{bel}(x_1)$$

Measurement Model

$$p(Z_t = \textbf{sense\_open} \mid X_t = \textbf{is\_open}) = 0.6$$
$$p(Z_t = \textbf{sense\_closed} \mid X_t = \textbf{is\_open}) = 0.4$$

$$p(Z_t = \textbf{sense\_open} \mid X_t = \textbf{is\_closed}) = 0.2$$
$$p(Z_t = \textbf{sense\_closed} \mid X_t = \textbf{is\_closed}) = 0.8$$

Transition Model for do_nothing

$$p(X_t = \textbf{is\_open} \mid U_t = \textbf{do\_nothing}, X_{t-1} = \textbf{is\_open}) = 1$$
$$p(X_t = \textbf{is\_closed} \mid U_t = \textbf{do\_nothing}, X_{t-1} = \textbf{is\_open}) = 0$$
$$p(X_t = \textbf{is\_open} \mid U_t = \textbf{do\_nothing}, X_{t-1} = \textbf{is\_closed}) = 0$$
$$p(X_t = \textbf{is\_closed} \mid U_t = \textbf{do\_nothing}, X_{t-1} = \textbf{is\_closed}) = 1$$

$\boldsymbol{bel(x_t) = (0.75, 0.25)}$ Don't forget normalization!

Next time