

ECE 484: Principles of Safe Autonomy (Fall 2026)
Lecture 12: Filtering and Localization

Professor: Huan Zhang
huanz@illinois.edu

Slides adapted from Prof. Sayan Mitra's slides for Spring 2025;
Reference: Probabilistic Robotics by Sebastian Thrun, Wolfram Burgard, and Dieter Fox
Some slides are from the book's website

Outline

Problem setting: Estimate x_t from observations $z_{1:t}$, applied controls $u_{0:t-1}$, and known map m.

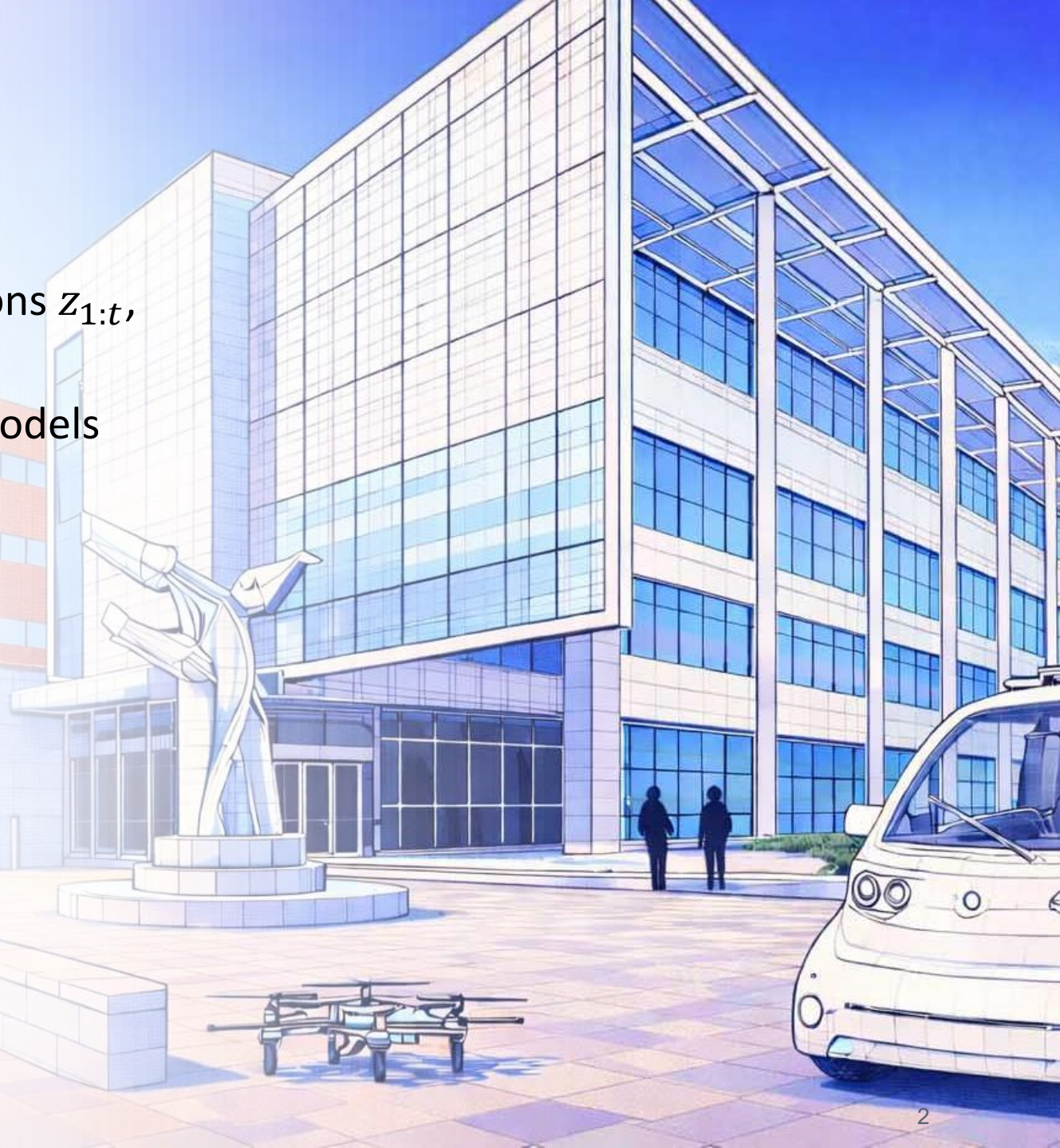
Review: motion models and measurement models

Today: discrete Bayes filter

Today: histogram filter and grid localization

Next lecture: particle filter

Timing: u_t moves the system from t to $t+1$.



Review: conditional probabilities and Bayes rule

Here X is **discrete**, with possible values $x_1, x_2, \ldots \in \mathbb{R}^n$.

Write $P(X = x) = P(x)$ and $P(X = x, Z = z) = P(x, z)$.

Conditional probability: $P(x|z) = \frac{P(x,z)}{P(z)}$, if $P(z) > 0$.

Bayes rule: $P(x|z) = \frac{P(z|x)P(x)}{P(z)}$.

The same joint event gives $P(x, z) = P(z|x)P(x)$.

Review: Markov chains

$\pi \in P(Q)$: distribution on finite state set Q; π is a **row vector**.

$\pi_i \geq 0, \sum_i \pi_i = 1$; $P(Q)$ is the set of such distributions.

Deterministic transition: $D: Q \to Q$.

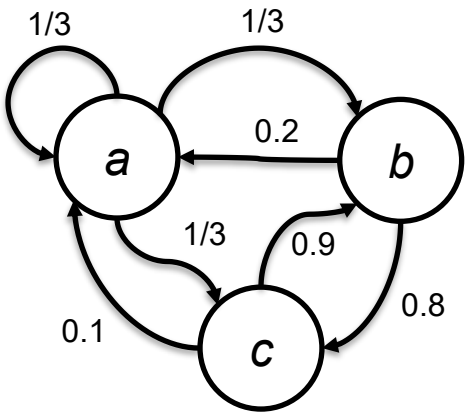
Probabilistic transition: $D: Q \to P(Q)$; $D(q)$ gives the next-state probabilities.

Example: $D(a) = [1/3,1/3,1/3]$; $D(b) = [1/5,0,4/5]$.

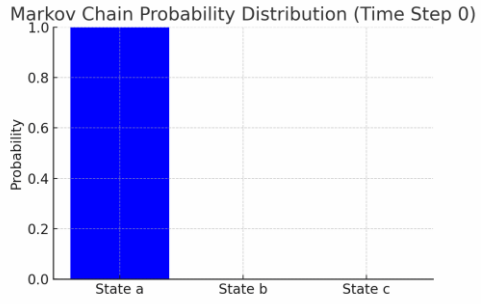
Markov chain: the current state summarizes the relevant past.

$D \in \mathbb{R}^{|Q|\times|Q|}$; $D_{ij} = P(i \to j)$, each row sums to 1.

Evolution: $\pi_{t+1} = \pi_t D$, starting from π_0.



$$\boldsymbol{D} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{5} & 0 & \frac{4}{5} \\ \frac{1}{10} & \frac{9}{10} & 0 \end{bmatrix}$$



Review: Markov Decision Process

Actions $u_t \in U$ change transitions: $D: Q \times U \to P(Q)$.

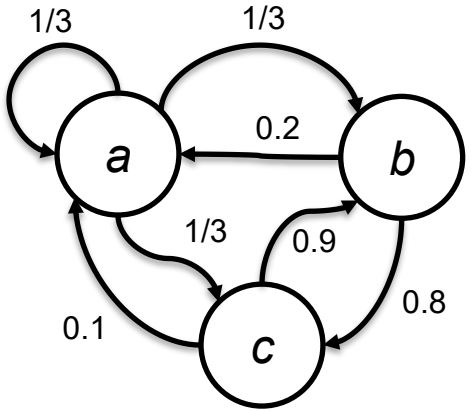
Example: $p_D(X_{t+1} = b | X_t = a, u_t = \text{red}) = 1/3$.

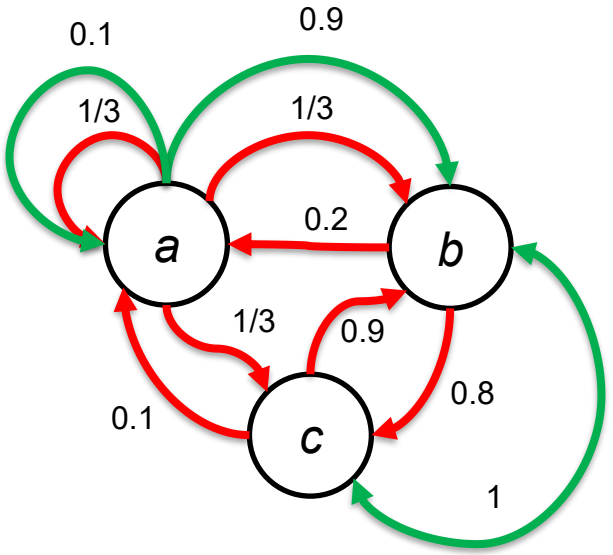
This is the **transition model** of a Markov decision process (MDP); a full MDP also includes rewards or costs.

$D(u)$ is an action-specific matrix; $D_{ij}(u) = P(i \to j | u)$.

Each row is nonnegative and sums to 1.

If **time invariant**, write $p_D(x' | x, u)$.





Review: the **motion model** under Markov assumption $p_D(x'|x,u)$

In general, motion can depend on earlier states, observations, and controls.

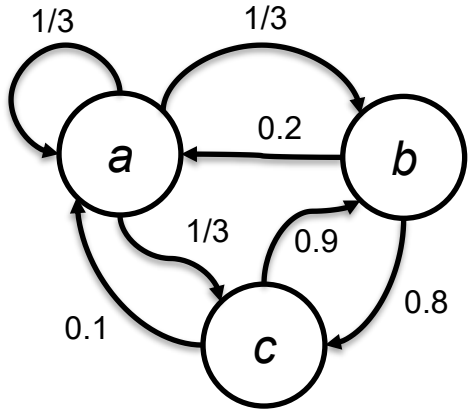
$$p_D(X_{t+1} = x'|X_0 = x_0, \dots, X_t = x_t, z_{1:t}, u_{0:t})$$

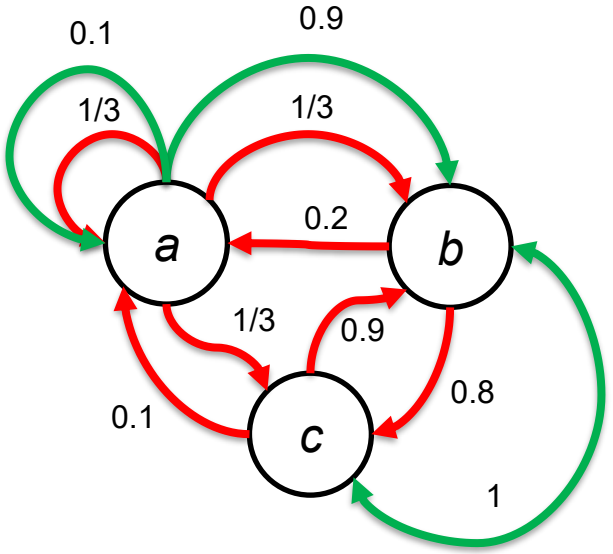
Shorthand: $p_D(x'|x_{0:t}, z_{1:t}, u_{0:t})$.

Controlled Markov assumption: the state includes the information needed for the next transition.

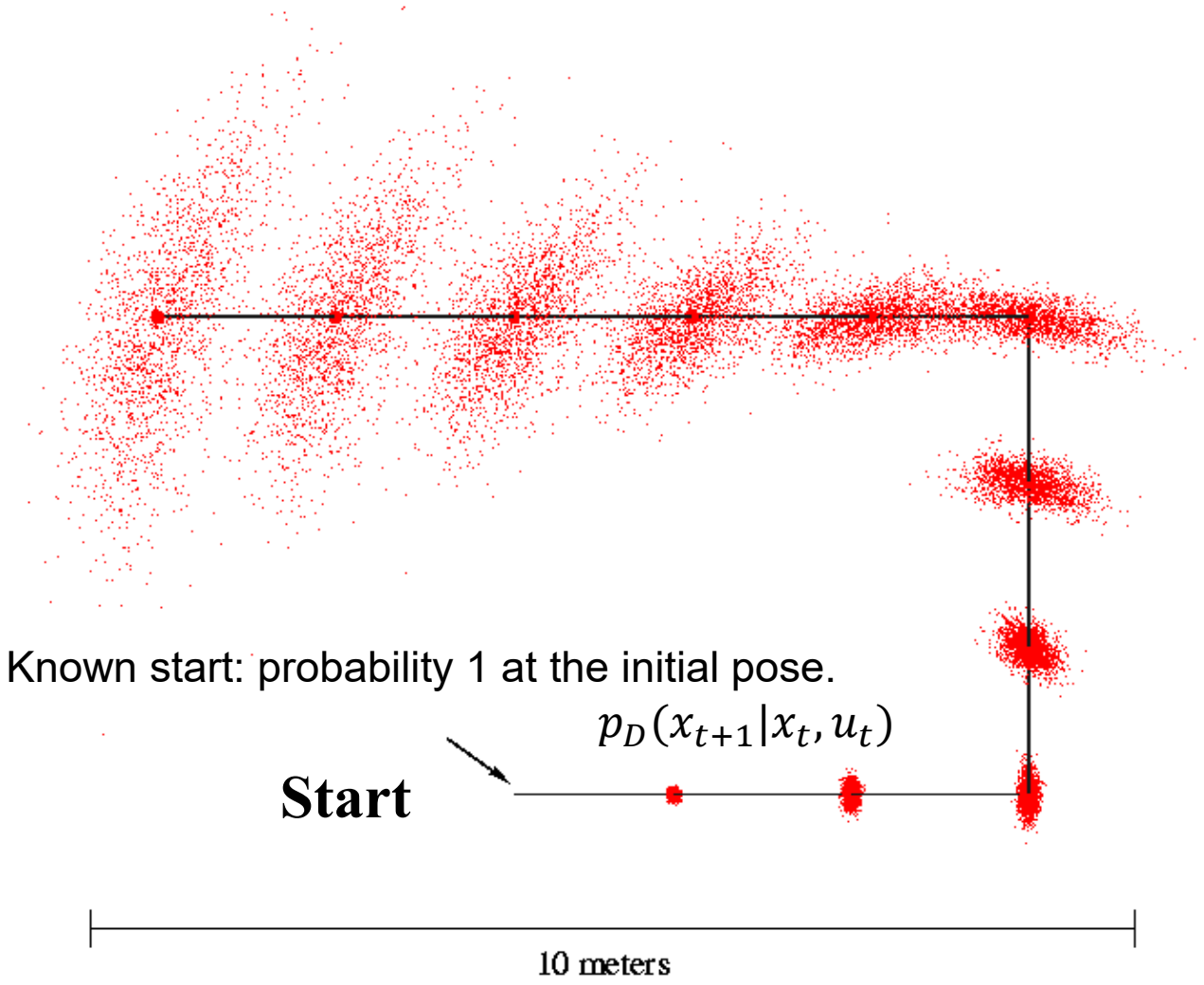
$p_D(x'|x_{0:t}, z_{1:t}, u_{0:t}) = p_D(x'|x_t, u_t)$.

For a time-invariant motion model: $p_D(x'|x,u)$.





Example Motion Model without measurements



One illustrative motion model:

$$x_{t+1} = f(x_t, u_t) + \varepsilon_t$$
$$\varepsilon_t \sim N(0, Q_t)$$

f: nominal motion; Q_t: noise covariance.

Review: probabilistic **measurement model** $p_M(z|x)$

A **measurement model** gives a distribution of sensor readings for each state.

Example: $p_M(z_t = 1|x_t = a) = 1/2$.

General history model: $p_M(z_t|x_{0:t}, z_{1:t-1}, u_{0:t-1})$.

Assume: the state is sufficient and sensor noise has no additional memory.

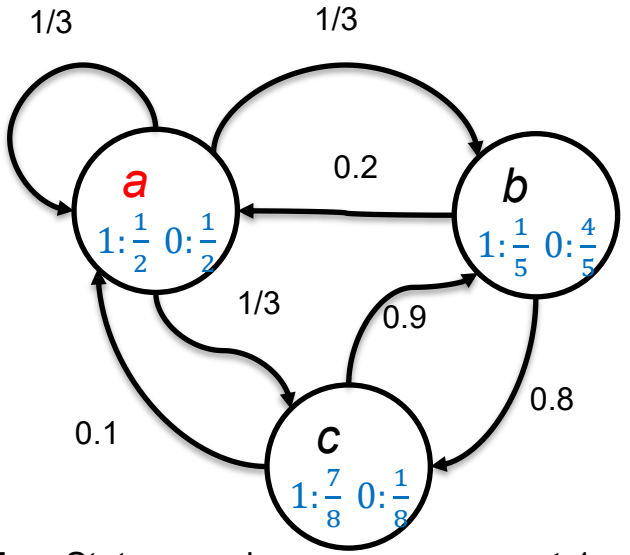
Then $p_M(z_t|x_{0:t}, z_{1:t-1}, u_{0:t-1}) = p_M(z_t|x_t)$.

For a **time-invariant** sensor: $p_M(z|x)$.

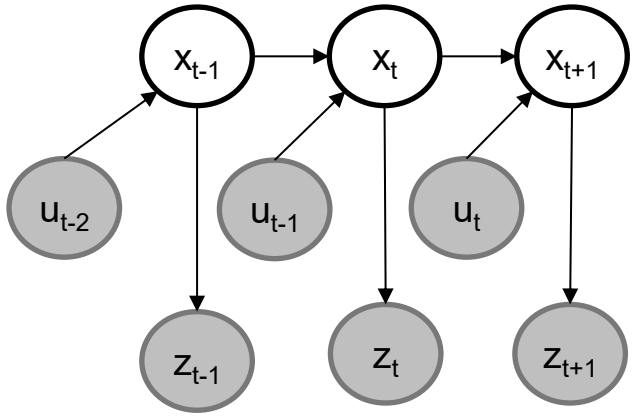
$$z_t = h(x_t) + \delta_t$$

h: nominal sensor model.

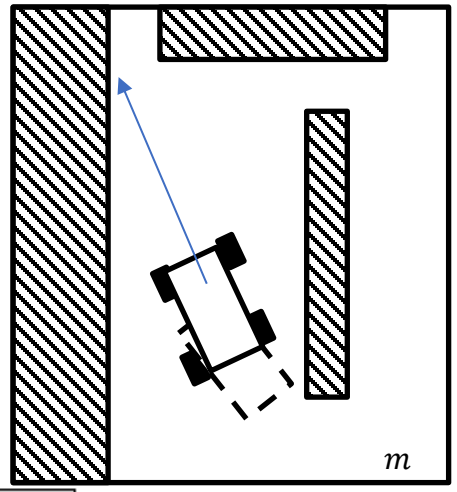
δ_t: sensor noise.



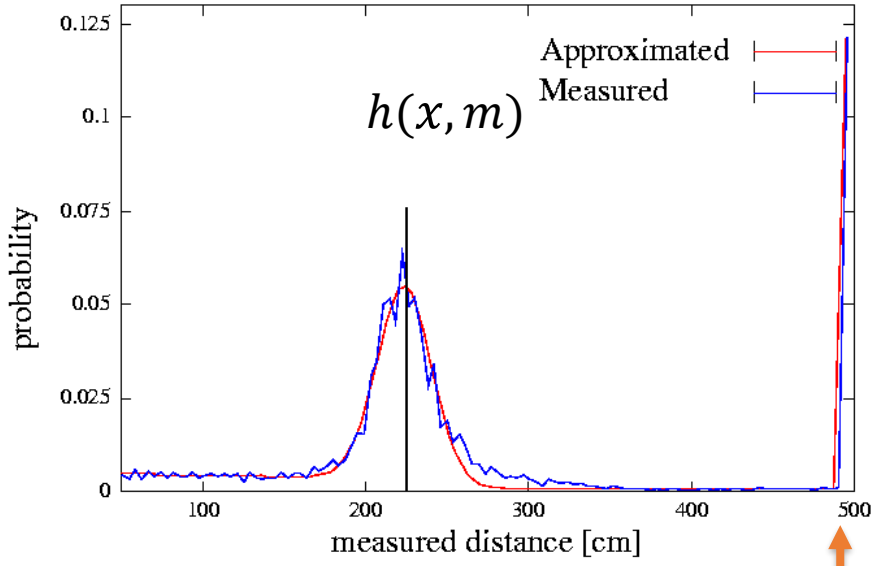
E.g., State a produces **measurement** 1 or 0 each with probability 0.5



Example Proximity Sensor Measurement Models

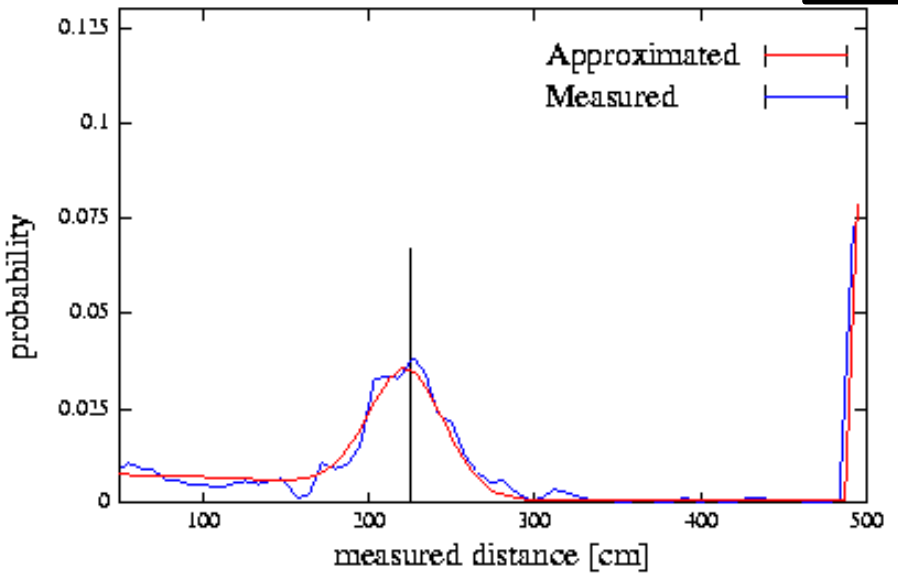


$p_M(z|x,m)$



Laser sensor

max-range spike



Sonar sensor

Summary: Motion and measurement models

Motion: general model $p_D(x_t|x_{0:t-1}, z_{1:t-1}, u_{0:t-1})$.

Assume the state summarizes the relevant motion history.

Then $p_D(x_t|x_{0:t-1}, z_{1:t-1}, u_{0:t-1}) = p_D(x_t|x_{t-1}, u_{t-1})$.

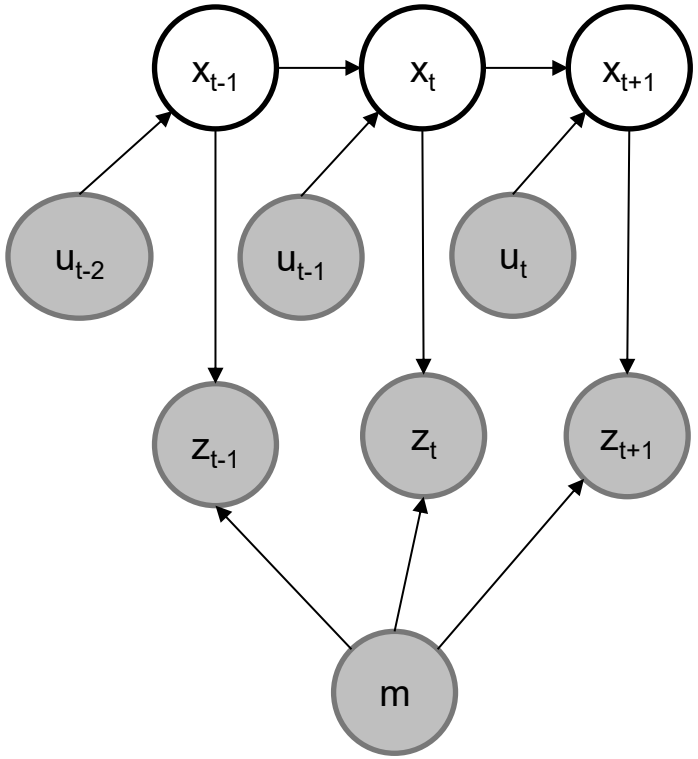
If time invariant: $p_D(x'|x, u)$.

Measurement: general model $p_M(z_t|x_{0:t}, z_{1:t-1}, u_{0:t-1}, m)$.

Assume the current state/map determine the sensor distribution; noise has no extra memory.

Then $p_M(z_t|x_{0:t}, z_{1:t-1}, u_{0:t-1}, m) = p_M(z_t|x_t, m)$.

If time invariant: $p_M(z|x, m)$.



Beliefs

Belief: a probability distribution over the possible state.

The true state x_t is hidden; measurements help infer it.

$bel(x_t) = p(x_t|z_{1:t}, u_{0:t-1})$.

Initially $bel(x_0) = \pi_0$; include any initial reading in π_0.

Bayes Filter: a two-step algorithm to update the belief:

Prediction: calculate $\overline{bel}(x_t) = p(x_t|z_{1:t-1}, u_{0:t-1})$ apply motion model

Correction: use z_t to turn $\overline{bel}(x_t)$ into $bel(x_t)$ apply measurement model

The overbar means "before the newest measurement."

Deriving prediction: From $bel(x_{t-1})$ to $\overline{bel}(x_t)$

Known history: $H = (z_{1:t-1}, u_{0:t-2})$; after the chosen action, $B = (H, u_{t-1})$.

x: old candidate ($X_{t-1} = x$); x': new candidate ($X_t = x'$); Q: finite state set.

Short notation: $bel(x) = P(x|H)$ and $\overline{bel}(x') = P(x'|B)$.

1. **Total probability**
Add disjoint old-state cases.

$$\overline{bel}(x') = P(x'|B) = \sum_{x \in Q} P(x', x|B)$$

2. **Product rule**
Definition of condition probability.

$$= \sum_{x \in Q} P(x'|x, B) P(x|B)$$

3. **Markov assumption**
No extra dependence on earlier history.
Our motion model appears here

$$= \sum_{x \in Q} p_D(x'|x, u_{t-1}) P(x|B)$$

$$= \sum_{x \in Q} p_D(x'|x, u_{t-1}) bel(x)$$

Deriving correction: apply Bayes' rule

Keep the same known background $B = (H, u_{t-1})$. Now observe the sensor reading z_t.

$\overline{bel}(r) = P(r|B)$; $bel(x') = P(x'|z_t, B)$. Here $r \in Q$ is any current candidate.

1. Bayes rule

Going from P(x|z) to P(z|x)

$$bel(x') = P(x'|z_t, B) = \frac{P(x', z_t|B)}{P(z_t|B)} = \frac{P(z_t|x', B)P(x'|B)}{P(z_t|B)}$$

2. Sensor assumption

Current state explains the reading. Drop B.

Measurement model appears.

$$= \frac{p_M(z_t|x')\overline{bel}(x')}{P(z_t|B)}$$

Same term

3. Total probability

To get the normalization factor, sum over current states; apply the product rule.

$$P(z_t|B) = \sum_{r \in Q} P(z_t, r|B) = \sum_{r \in Q} P(z_t|r, B)P(r|B)$$

$$= \sum_{r \in Q} p_M(z_t|r)\overline{bel}(r)$$

4. Normalize

One shared total.

$$\eta = \frac{1}{P(z_t|B)}, \quad bel(x') = \eta p_M(z_t|x')\overline{bel}(x')$$

Assume $P(z_t|B) > 0$.

Bayes Filter: Prediction and Correction

Bayes_filter$(bel(x_{t-1}), u_{t-1}, z_t)$ returns $bel(x_t)$.

Notation: Q: finite state set; x: old candidate; x': new candidate.

Prediction: $\overline{bel}(x') = \sum_{x \in Q} p_D(x'|x, u_{t-1}) bel(x)$. motion model

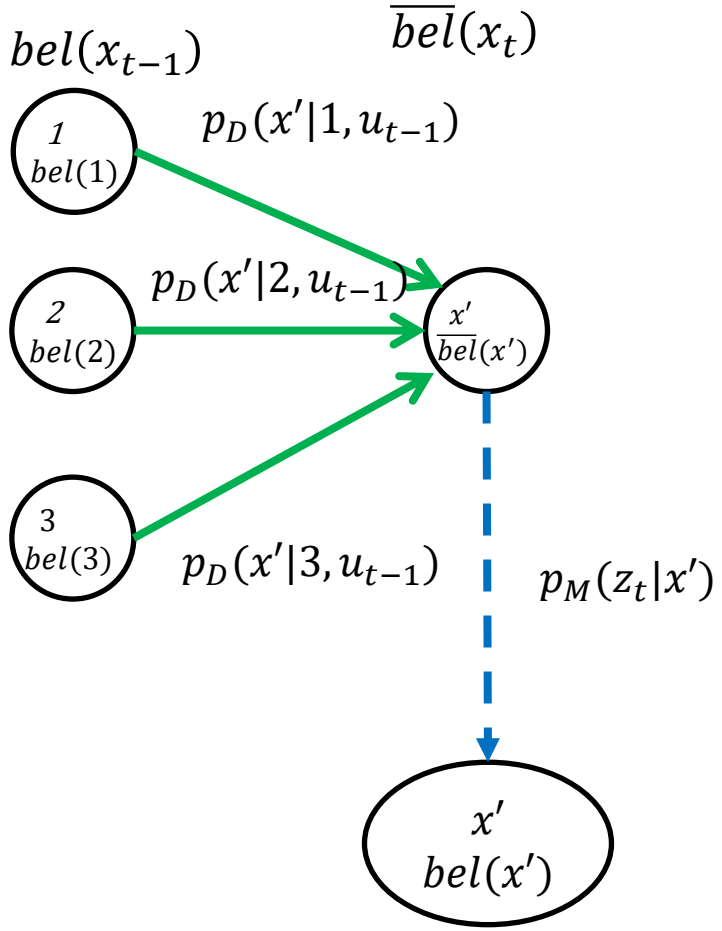
Intuition: all previous state x may have a probability to reach x', we must sum them up, weighted by probability $bel(x)$

Correction: $bel(x') = \eta p_M(z_t|x') \overline{bel}(x')$. measurement model

Normalization factor: $\eta^{-1} = \sum_{r \in Q} p_M(z_t|r) \overline{bel}(r) > 0$.

Intuition: if new observation align with believe, increase belief probability; otherwise decrease.

All old beliefs are at t − 1; predicted/current beliefs are at t.



Histogram Filter or Discrete Bayes Filter

Define: $Q = \{x_1, \dots, x_K\}$; $p_{k,t} = bel(X_t = x_k)$

Discrete_Bayes_filter$(\{p_{k,t-1}\}, u_{t-1}, z_t)$:

For each candidate k:

$\overline{p}_{k,t} = \sum_i p_D(x_k|x_i, u_{t-1}) p_{i,t-1}$

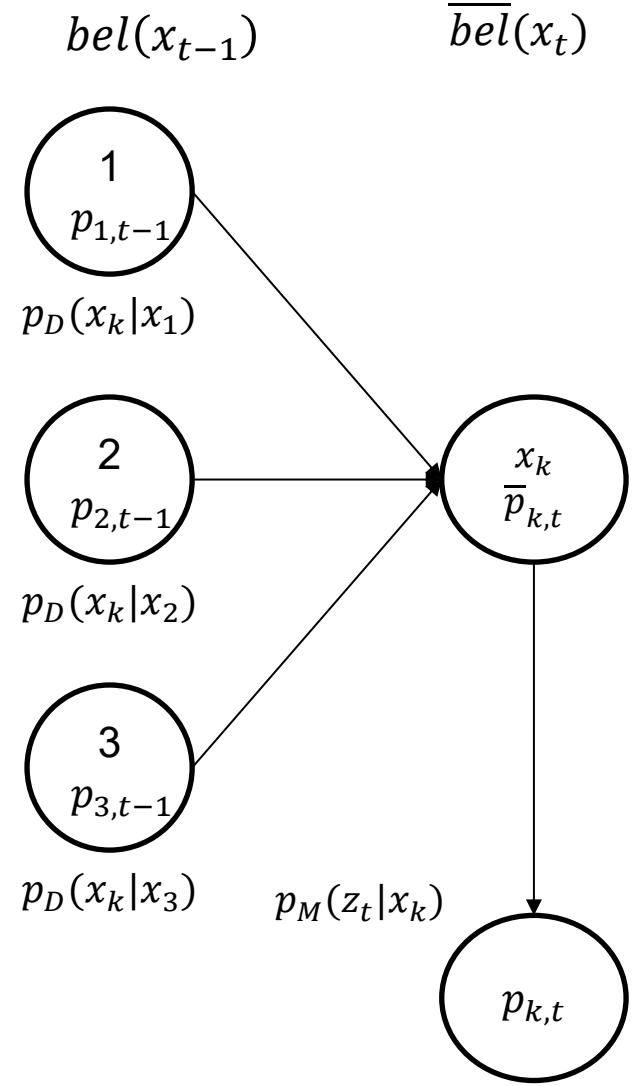
$w_{k,t} = p_M(z_t|x_k)\overline{p}_{k,t}$ **(unnormalized weights)**

End for

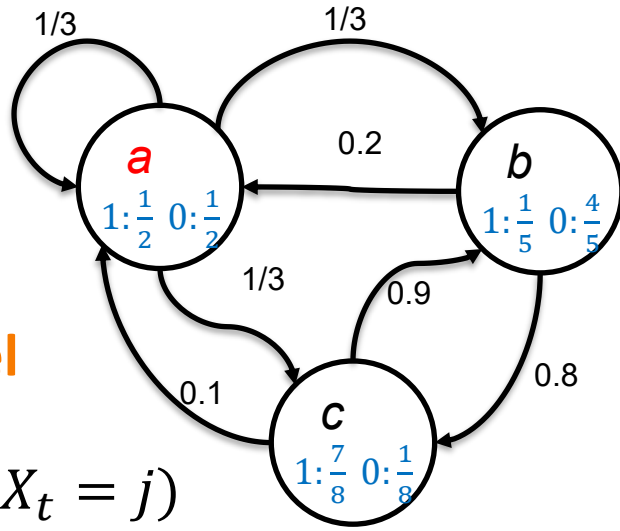
Compute $S = \sum_j w_{j,t}$; if $S > 0$, set $\eta = 1/S$. **(Normalization, shared by all)**

Return $p_{k,t} = \eta w_{k,t}$ for every k.

Fixed control u_{t-1} in this diagram.



Three-state example: all the probabilities



Hidden state $X_t \in \{a, b, c\}$; sensor reading $z_t \in \{0,1\}$.

Previous belief (before motion)

a b c

$p_{a,t-1} = 0.6$ $p_{b,t-1} = 0.3$ $p_{c,t-1} = 0.1$

Motion model = transition probabilities

$$D_{ij} = P(X_t = j | X_{t-1} = i)$$

from / to	a	b	c
a	$1/3$	$1/3$	$1/3$
b	0.2	0	0.8
c	0.1	0.9	0

Each row adds to 1. No control is used in this example.

Measurement model

$$p_M(z|j) = P(Z_t = z | X_t = j)$$

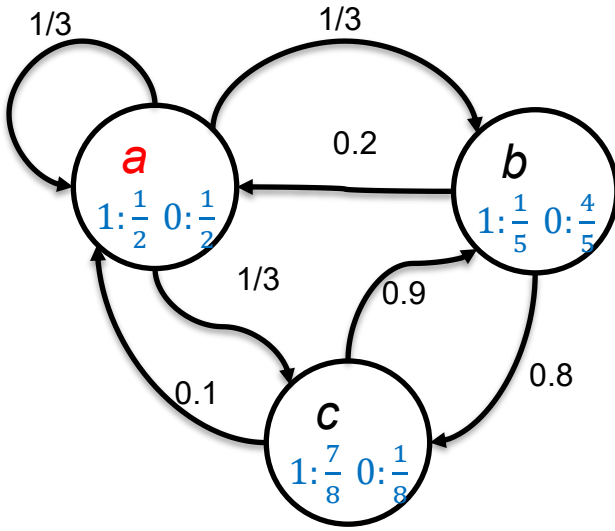
state	$z = 0$	$z = 1$
a	0.5	0.5
b	0.8	0.2
c	0.125	0.875

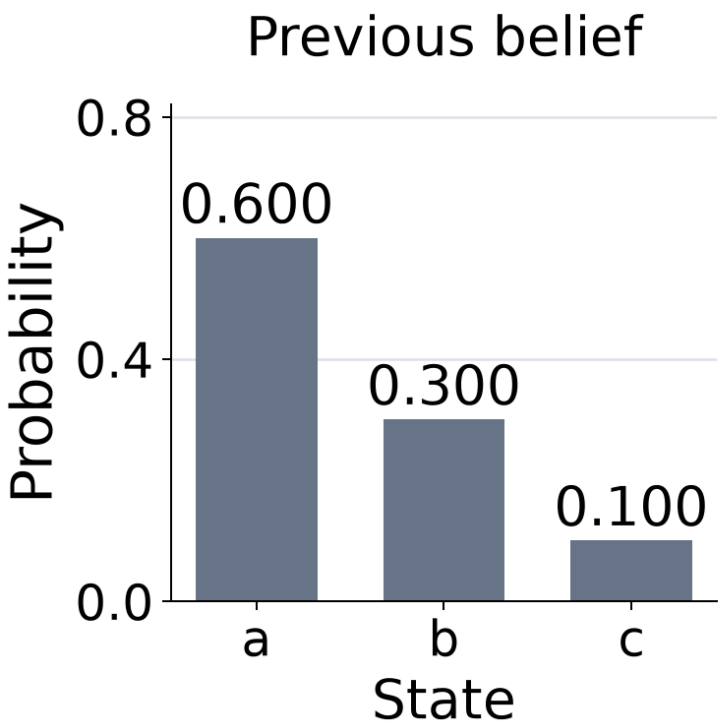
Each row adds to 1 across possible readings.

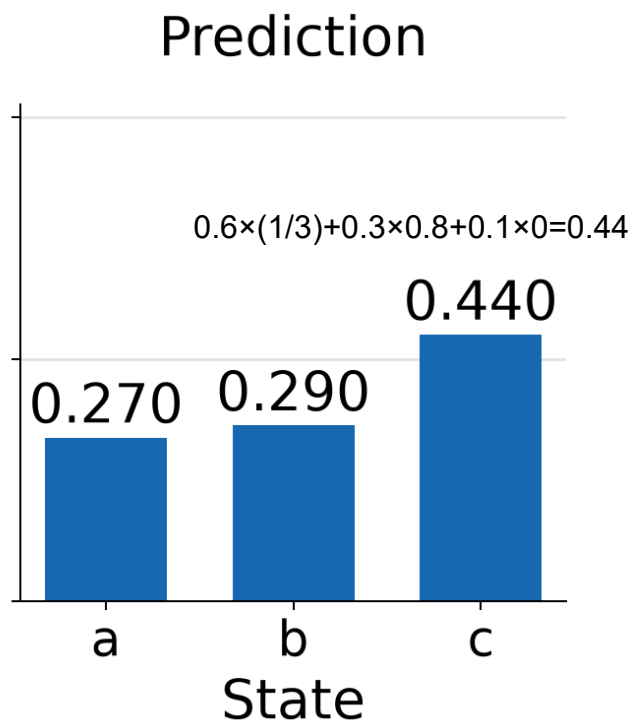
Next: move once; observe $z_t = 1$.

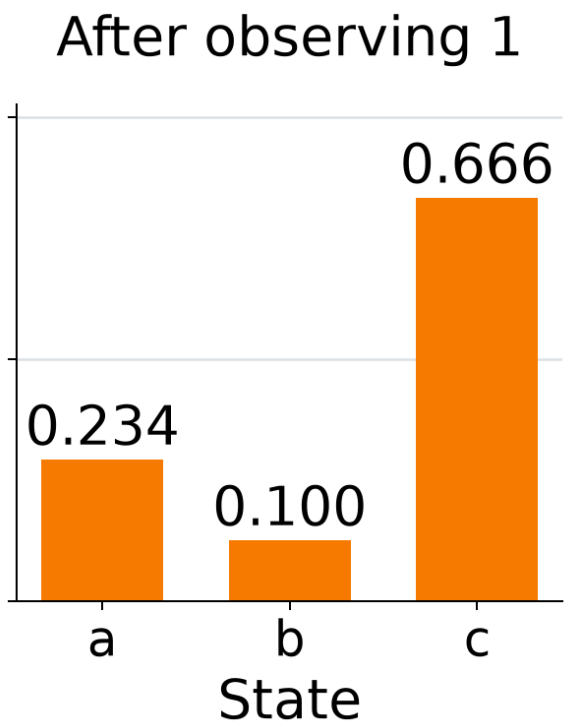
One Bayes-filter step, with numbers

Use the original three states a, b, c and their binary sensor.









Bayes Filter: Continuous Distributions

Bayes_filter$(bel(x_{t-1}), u_{t-1}, z_t)$:

For each candidate x_t (conceptually):

$$\overline{bel}(x_t) = \int p_D(x_t|x_{t-1}, u_{t-1}) bel(x_{t-1})\, dx_{t-1}$$

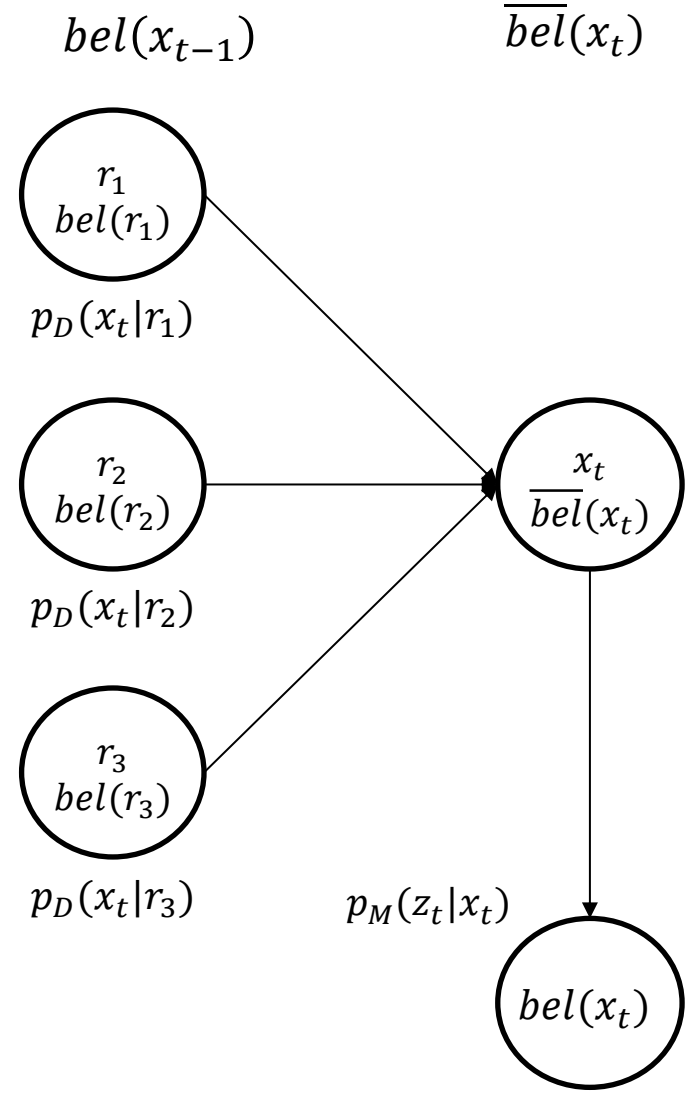
$$bel(x_t) = \eta p_M(z_t|x_t) \overline{bel}(x_t)$$

End for

$$\eta^{-1} = \int p_M(z_t|r) \overline{bel}(r)\, dr > 0.$$

Return bel; it is a **density**, with total integral 1.

Fixed control u_{t-1} in this diagram.



Grid Localization: Bayes Filter applied on grids

Same Bayes filter; each possible state is now a **pose cell** with discrete position and heading

Discrete Bayes filter



One probability per state

→

Grid localization



One heading layer; repeat for each heading bin.

One state = pose cell C_k
Center $c_k = (x_B, y_B, \theta_B)$

$p_{k,t}$ is the probability that the robot is in cell k

Predict

$$\overline{p}_{k,t} = \sum_i p_{i,t-1} T_{ik}(u_{t-1})$$

Reweight

$$w_{k,t} = \overline{p}_{k,t} \ell_{k,t}$$

Normalize

$$p_{k,t} = w_{k,t} / \sum_j w_{j,t}$$

T_{ik}: cell transition probability among the cells (motion model), $\sum_k T_{ik} = 1$
$\ell_{k,t} \approx p_M(z_t | c_k, m)$ (measurement model at the cell center).

Grid localization

Grid_localization$(\{p_{k,t-1}\}, u_{t-1}, z_t, m)$:

For each pose cell k with center c_k:

(The cells are the discrete states in Bayes filter)

$\overline{p}_{k,t} = \sum_i p_{i,t-1} T_{ik}(u_{t-1})$

$w_{k,t} = \overline{p}_{k,t}\, \text{measurement_model}(z_t, c_k, m)$

End for

Compute $S = \sum_j w_{j,t}$; if $S > 0$, return $p_{k,t} = w_{k,t}/S$.

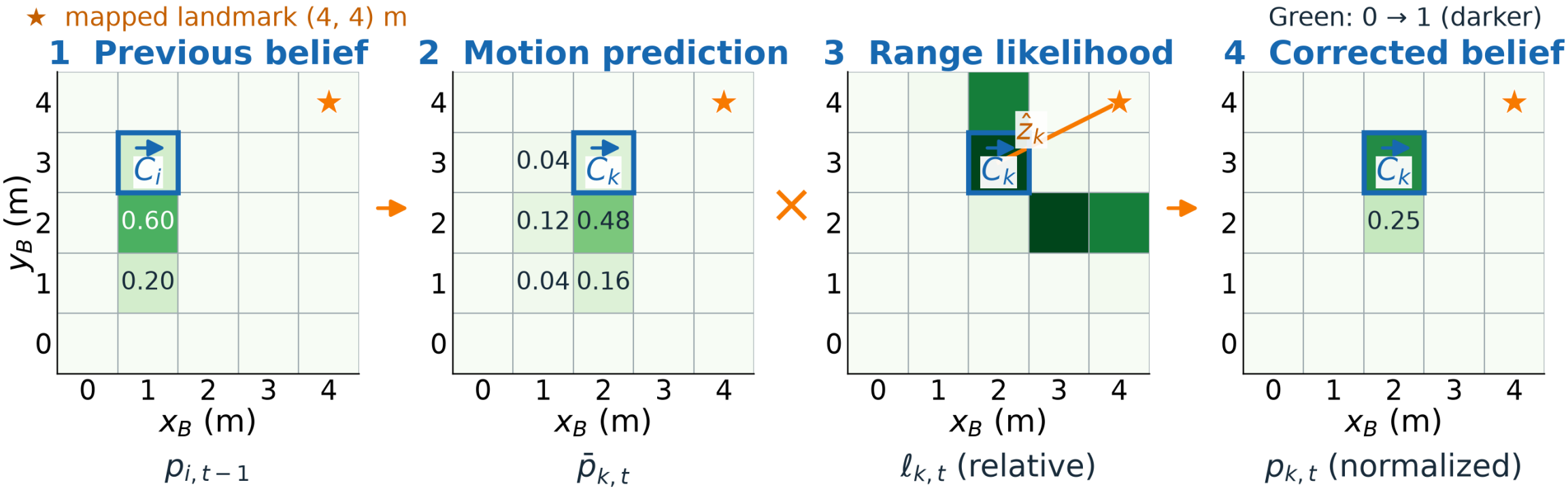
c_k: center of pose cell k; $p_{k,t}$: probability in that cell.

$T_{ik}(u) = \text{motion_model}(c_k, u, c_i)$: cell transition probability; $\sum_k T_{ik}(u) = 1$.

Evaluate likelihoods at centers only as a grid approximation.

Grid localization Example 1: bicycle motion + landmark range

Pose cell C_k has center $c_k = (x_B, y_B, \theta_B)$. Here $\theta_B = 0$ (east) is fixed.



Motion model: 0.8 moves one cell east; 0.2 stays. (A simple straight-motion grid model.)

Measurement (distant to landmark ★): $z_t = 2.2$ m; expected range $\hat{z}_k = \sqrt{(x_B - 4)^2 + (y_B - 4)^2}$.
At C_k: $\bar{p}_{k,t} = 0.16$, $\ell_{k,t} \approx 0.993$; after multiplying and normalizing, $p_{k,t} \approx 0.747$.

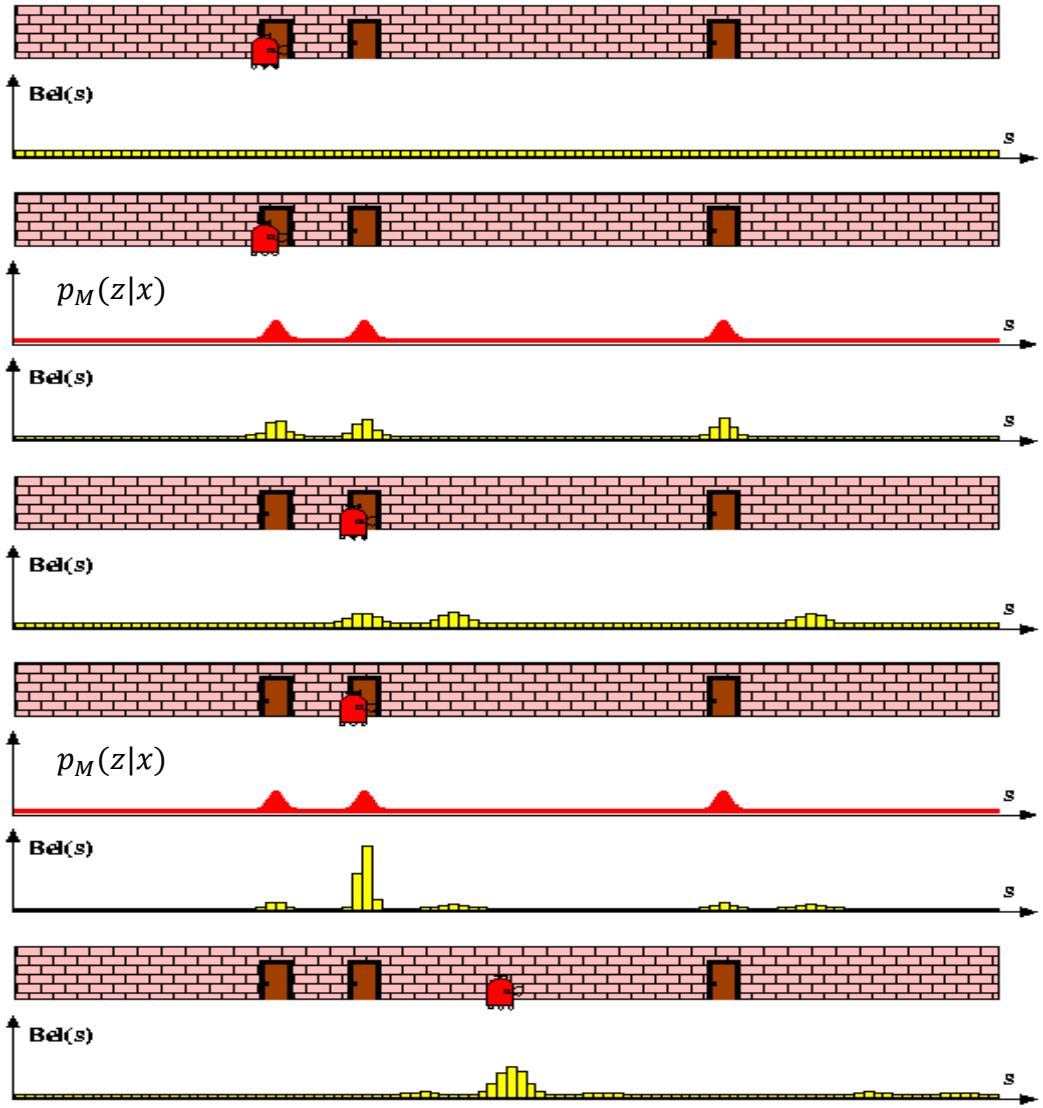
Grid localization Example 2: one-dimensional corridor

Correction: compare with the sensed doors.

Prediction: move all plausible positions.

A second reading can distinguish the earlier alternatives.

Here s is position along the corridor.



Initial

Correction

Prediction

Correction

Prediction

Try it yourself: https://colab.research.google.com/drive/1YyBCjQeIQQDxzZ-crB3Ss_KM-qA2QF7N?usp=sharing

Grid Localization Properties

A **histogram over poses** can retain several plausible locations during **global localization**.

Nonparametric: the posterior is not restricted to a fixed distribution family.

Can retain **multiple peaks**, unlike a single-Gaussian (Extended) Kalman Filter

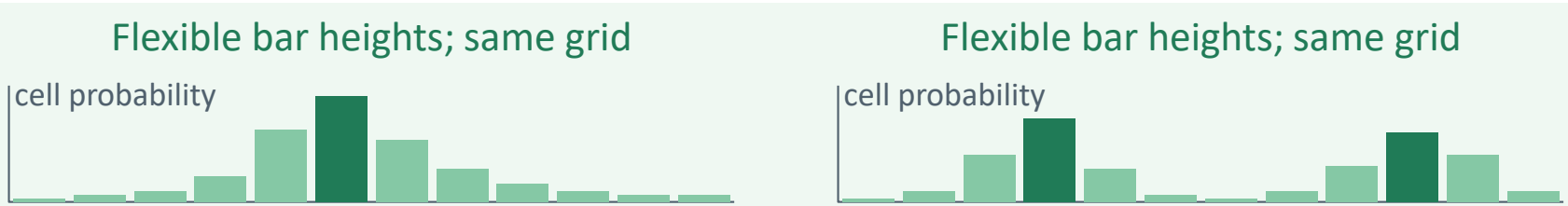
Parametric examples

Choose a fixed family;
set a few numbers.



Grid: nonparametric

Store a probability
for every cell.



Horizontal axis: corridor position s. Curves show density; bars show probability in each cell.

Easily use raw sensor data (e.g., laser/sonar ranges)**:** as long as we can easily evaluate the measurement model $p_M(z_t|c_k, m)$ at each cell center. No need to first identify features (e.g., landmarks) on a map.

Kidnapping recovery: possible, needs nonzero predicted belief at the new pose, or a reset.

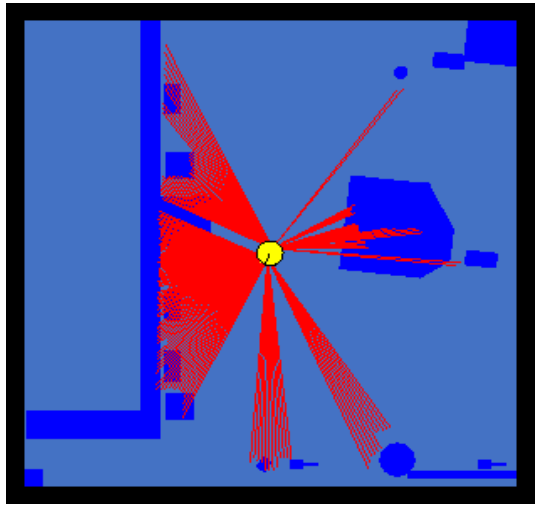
Grid-based Localization in practice

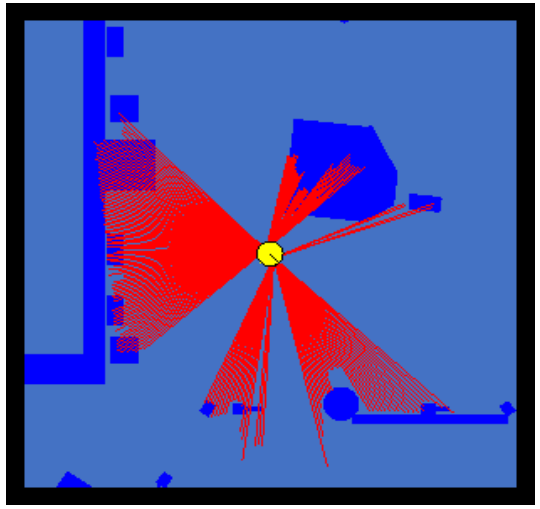
First scan

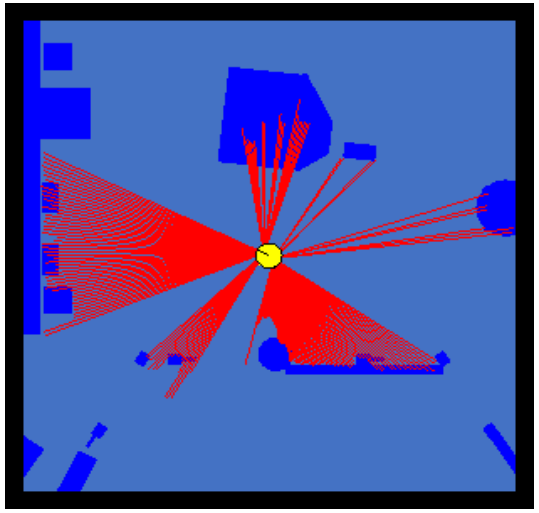
Second scan

Later scan

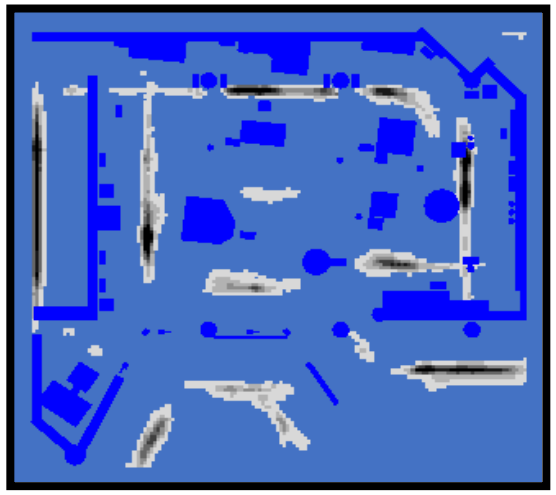
Scan



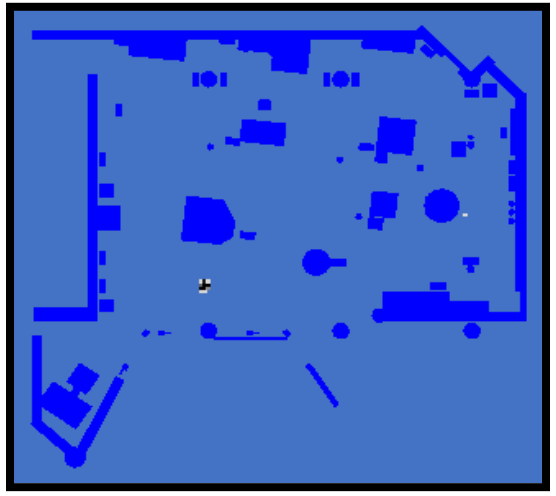




Belief

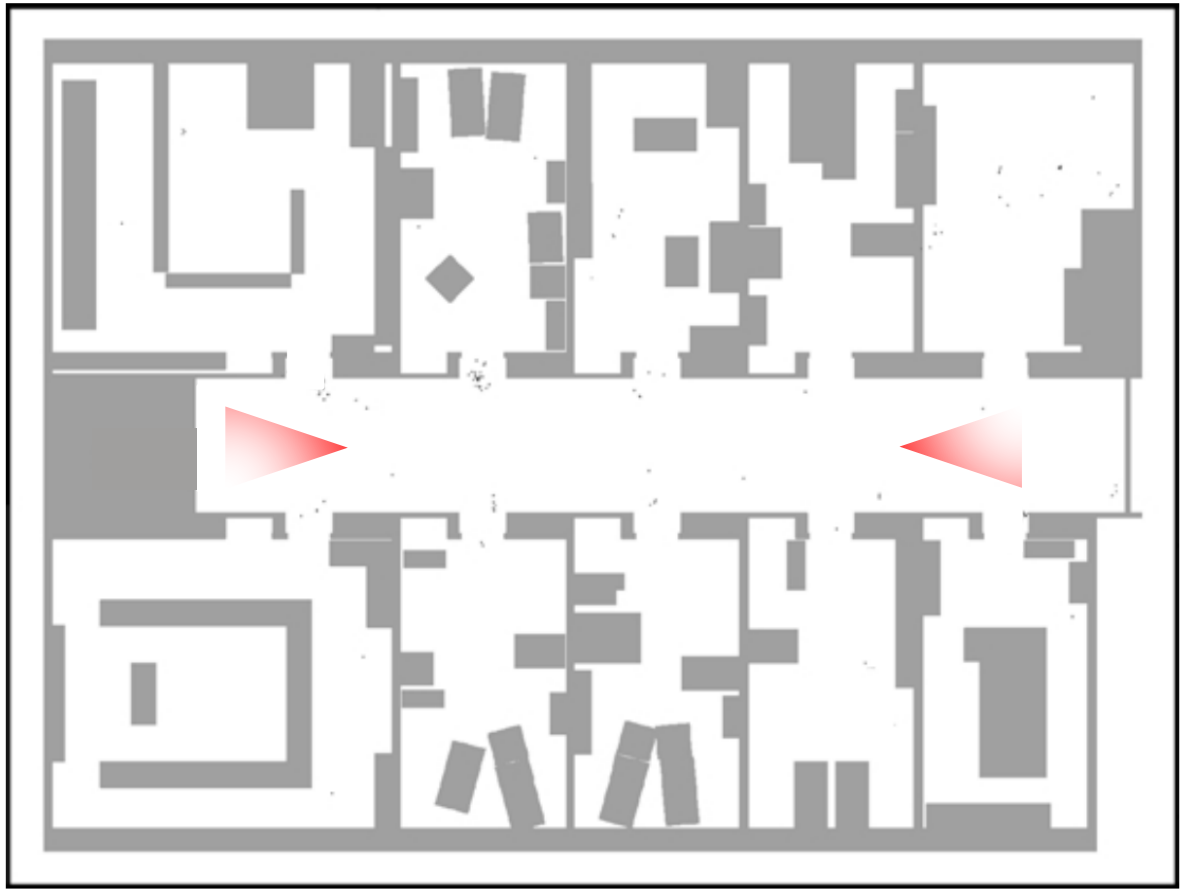




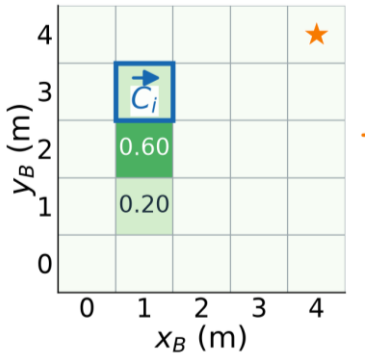


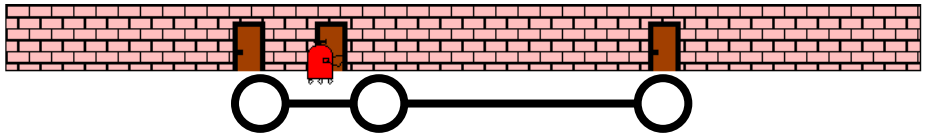
Global localization: two plausible poses

Question: what motion or observation could distinguish these two poses?



Grid localization: practical designs





Key choice: grid resolution.

Topological: group meaningful places. **Metric:** partition pose into fine cells.

A finer grid can improve approximation, with more memory and computation.

Coarse cells: should not discard small motions by repeatedly rounding to a cell center.

Example: 10 cm/s × 1 s = 0.1 m, only one tenth of a 1 m cell.

Computation: direct 3-D pose prediction sums over source/destination pairs: six coordinate loops.

With K cells: dense prediction $O(K^2)$; measurement update $O(K)$.

Fine grids can be costly; sparse transitions and caching (e.g., ray-cast ranges) help.