

AA 274

Principles of Robotic Autonomy

Localization III: Markov localization and EKF-localization



Stanford
University

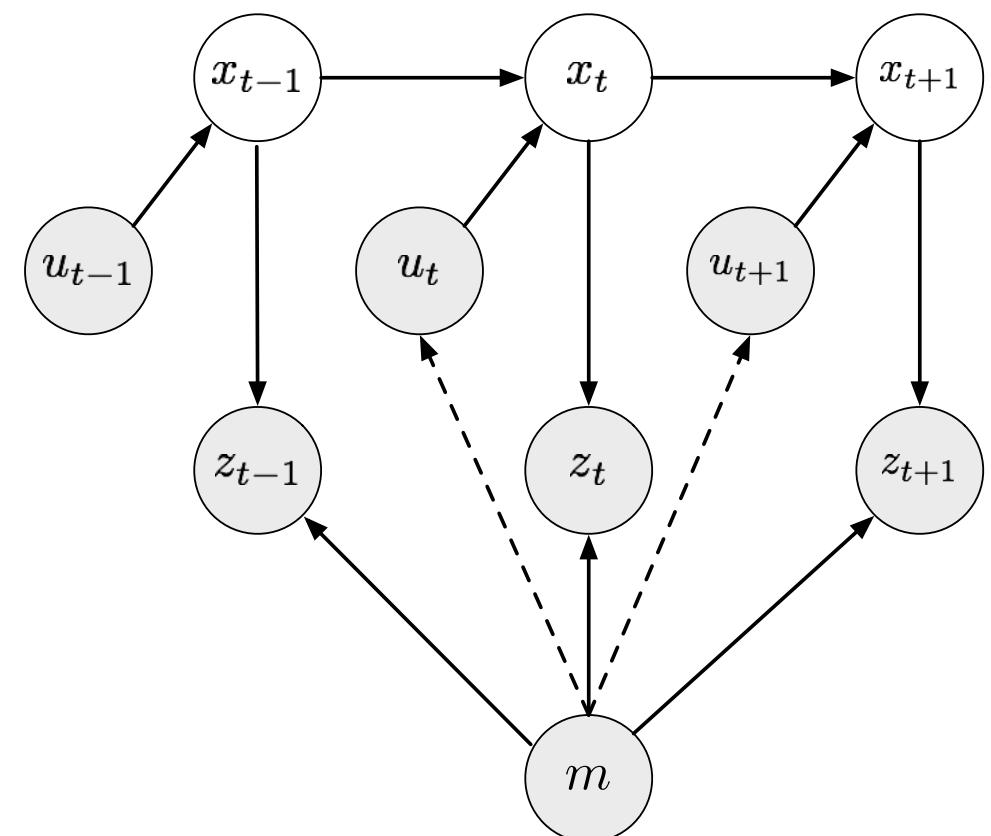


Today's lecture

- Aim
 - Learn about Markov localization, with an emphasis on EKF and non-parametric localization
- Readings
 - SNS: 5.6.4 – 5.6.8, and 5.7
 - S. Thrun, W. Burgard, and D. Fox. Probabilistic robotics. MIT press, 2005. Sections 7.2 – 7.6, 8.3

Mobile robot localization

- **Problem:** determine pose of a robot relative to a *given* map
- Localization can be interpreted as the problem of establishing correspondence between the map coordinate system and the robot's local coordinate frame
- This process requires integration of data over time



Local versus global localization

- **Position tracking** assumes that the initial pose is known -> *local* problem well-addressed via Gaussian filters
- In **global localization**, the initial pose is unknown -> *global* problem best addressed via non-parametric, multi-hypothesis filters
- In **kidnapped robot** localization, initial pose is unknown and during operation robot can be “kidnapped” and “teleported” to some other location -> *global* problem best addressed via non-parametric, multi-hypothesis filters

Static versus dynamic environments

- **Static environments** are environments where the only variable quantity is the pose of the robot
- **Dynamic environments** possess objects (e.g., people) other than the robot whose locations change over time -> addressed via either state augmentation or outlier rejection

Passive versus active localization

- In **passive localization**, localization module only *observes* the robot; i.e., robot's motion is not aimed at facilitating localization
- In **active localization**, robot's actions are aimed at minimizing the localization error
- Hybrid approaches are possible

Single-robot versus multi-robot

- In **single-robot localization**, a single, individual robot is involved in the localization process
- In **multi-robot localization**, a team of robots is engaged with localization, possibly cooperatively (or even adversarially!)

In this class we will focus on **local & global, static** (or quasi-static), **passive, single-robot** localization problems

Casting the localization problem within a Bayesian filtering framework

- State x_t , control u_t and measurements z_t have the same meaning as in the general filtering context
- For a differential drive robot equipped with a laser range-finder (returning a set of range r_i and bearing ϕ_i measurements)

$$x_t = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad u_t = \begin{pmatrix} v \\ \omega \end{pmatrix} \quad z_t = \left\{ \begin{pmatrix} r_i \\ \phi_i \end{pmatrix} \right\}_i$$

Casting the localization problem within a Bayesian filtering framework

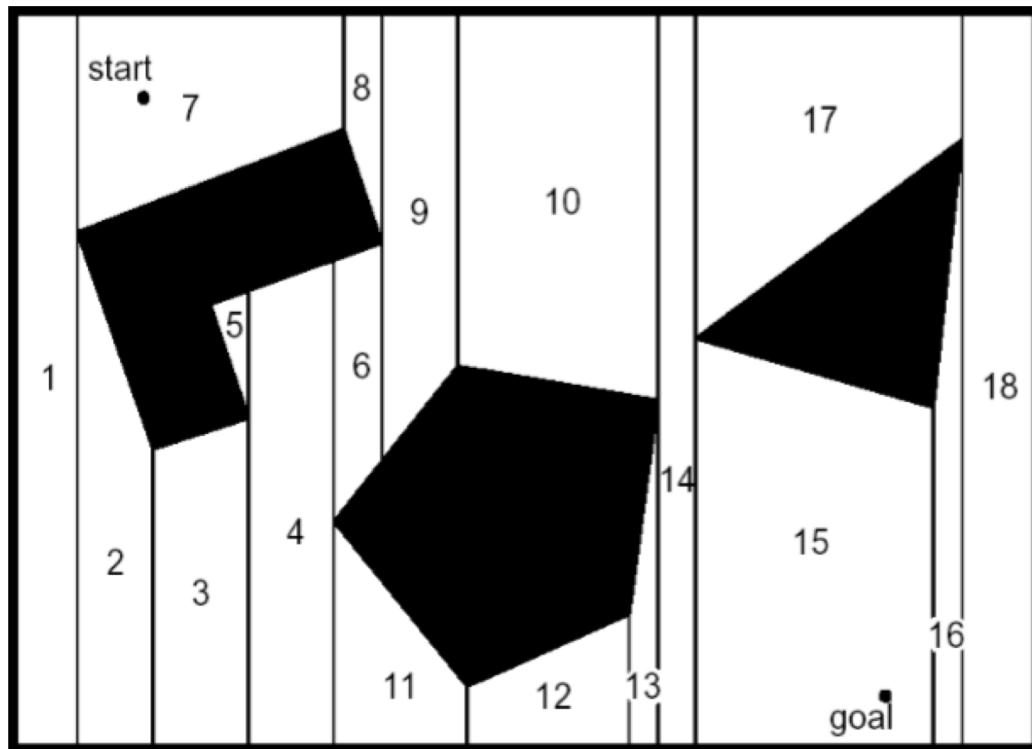
- A map m is a list of objects in the environment along with their properties

$$m = \{m_1, m_2, \dots, m_N\}$$

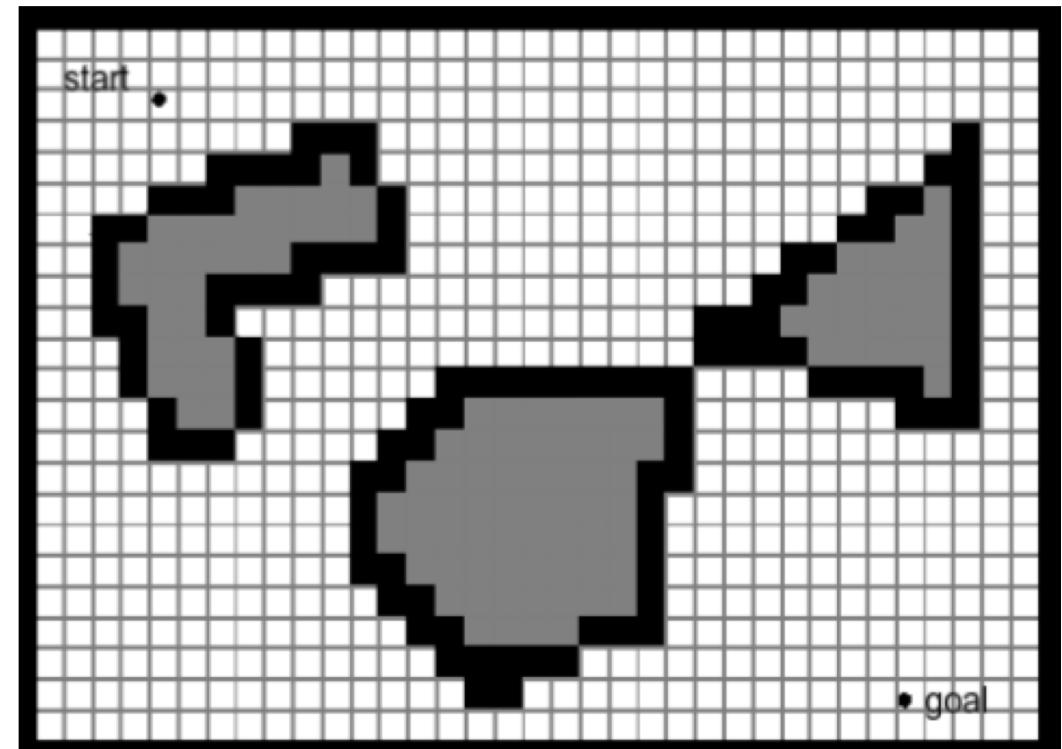
- Maps can be
 - *Location-based*: index i corresponds to a specific location (hence, they are volumetric)
 - *Feature-based*: index i is a feature index, and m_i contains, next to the properties of a feature, the Cartesian location of that feature

Location-based maps

Vertical cell decomposition

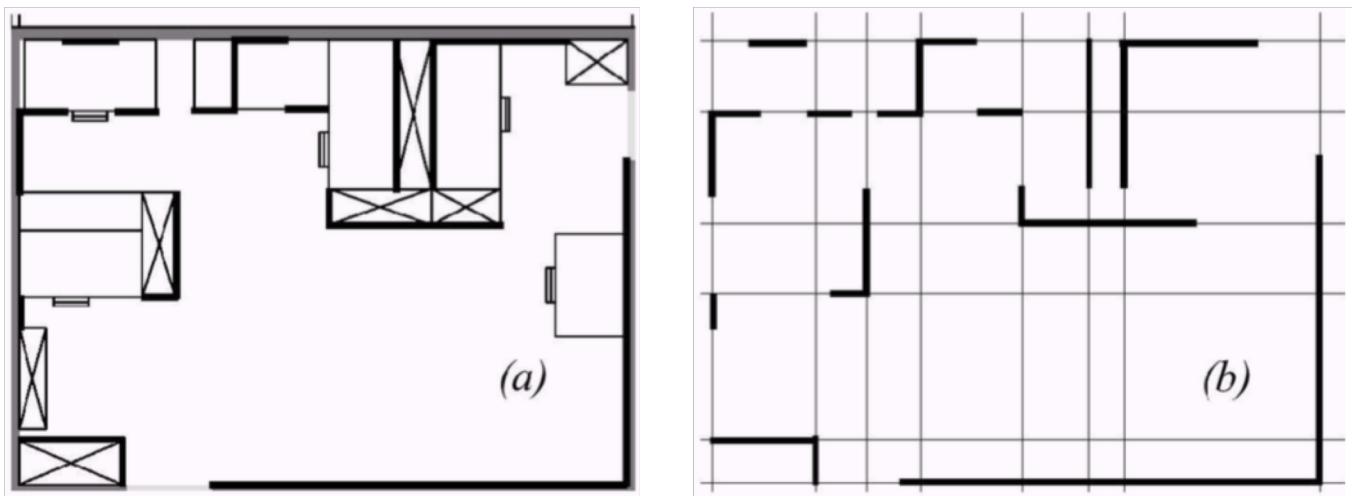


Fixed cell decomposition (occupancy grid)

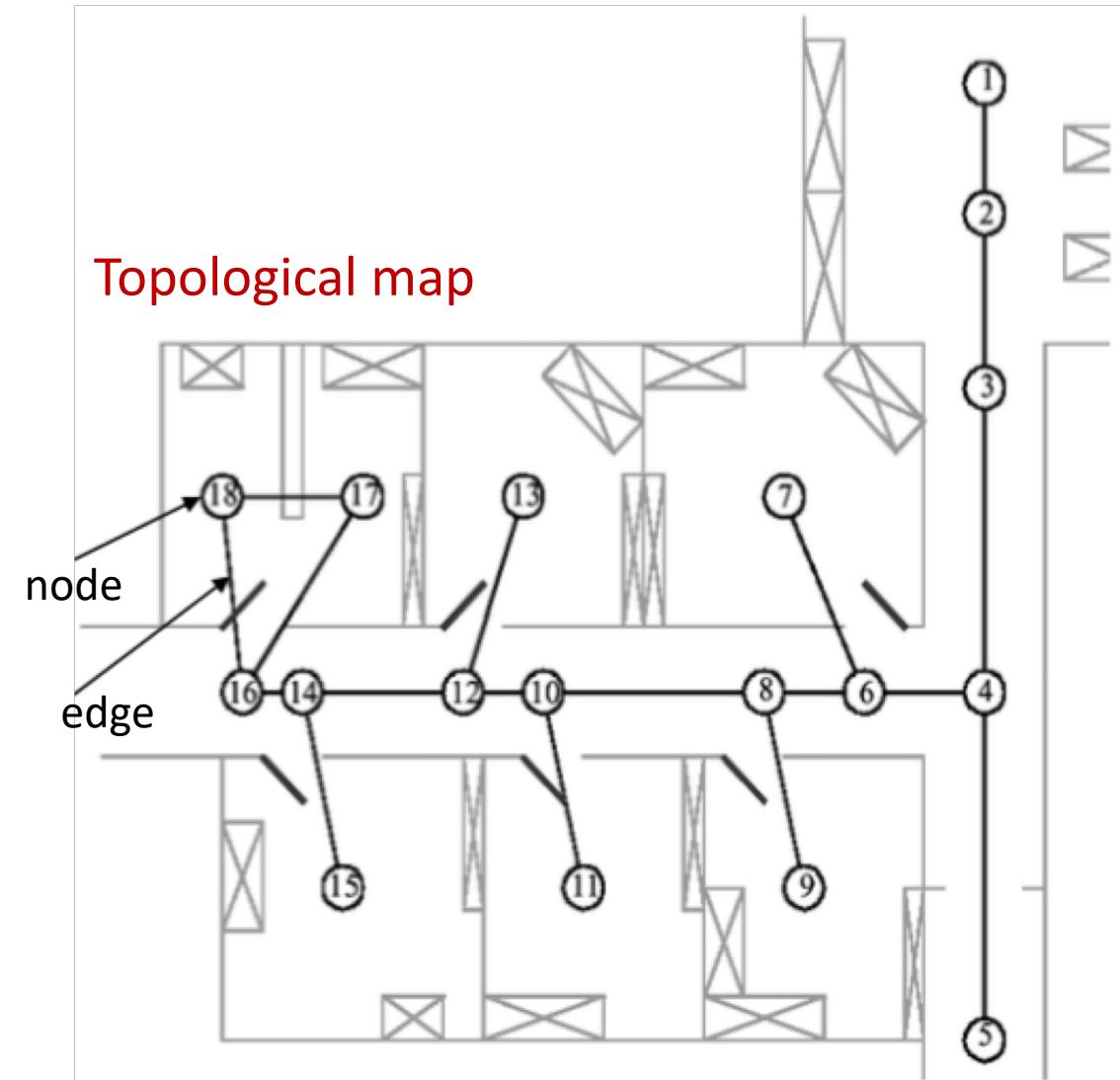


Feature-based maps

Line-based map

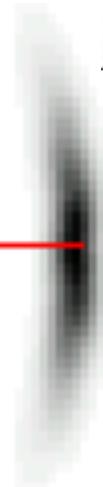


Topological map



Casting the localization problem within a Bayesian filtering framework

- Motion model is probabilistic

$$p(x_t | u_t, x_{t-1})$$


- Key fact: $p(x_t | u_t, x_{t-1}) \neq p(x_t | u_t, x_{t-1}, m)$
- Useful approximation (tight at high update rates)

$$p(x_t | u_t, x_{t-1}, m) \approx \eta \frac{p(x_t | u_t, x_{t-1}) p(x_t | m)}{p(x_t)}$$

Consistency of state
 x_t with map m

Uses approximation
 $p(m | x_t, u_t, x_{t-1}) \approx p(m | x_t)$

Casting the localization problem within a Bayesian filtering framework

- Measurement model is probabilistic

$$p(z_t | x_t, m)$$

- Sensors usually generate more than one measurement when queried

$$z_t = \{z_t^1, \dots, z_t^K\}$$

- Typically, independence assumption is made

$$p(z_t | x_t, m) = \prod_{k=1}^K p(z_t^k | x_t, m)$$

Markov localization

- Straightforward application of Bayes filter
- Requires a map m as input
- Addresses:
 - Global localization
 - Position tracking
 - Kidnapped robot problem

Data: $bel(x_{t-1}), u_t, z_t, \textcolor{red}{m}$

Result: $bel(x_t)$
foreach x_t **do**

$$\left| \begin{array}{l} \overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}, \textcolor{red}{m}) bel(x_{t-1}) dx_{t-1}; \\ bel(x_t) = \eta p(z_t | x_t, \textcolor{red}{m}) \overline{bel}(x_t); \end{array} \right.$$

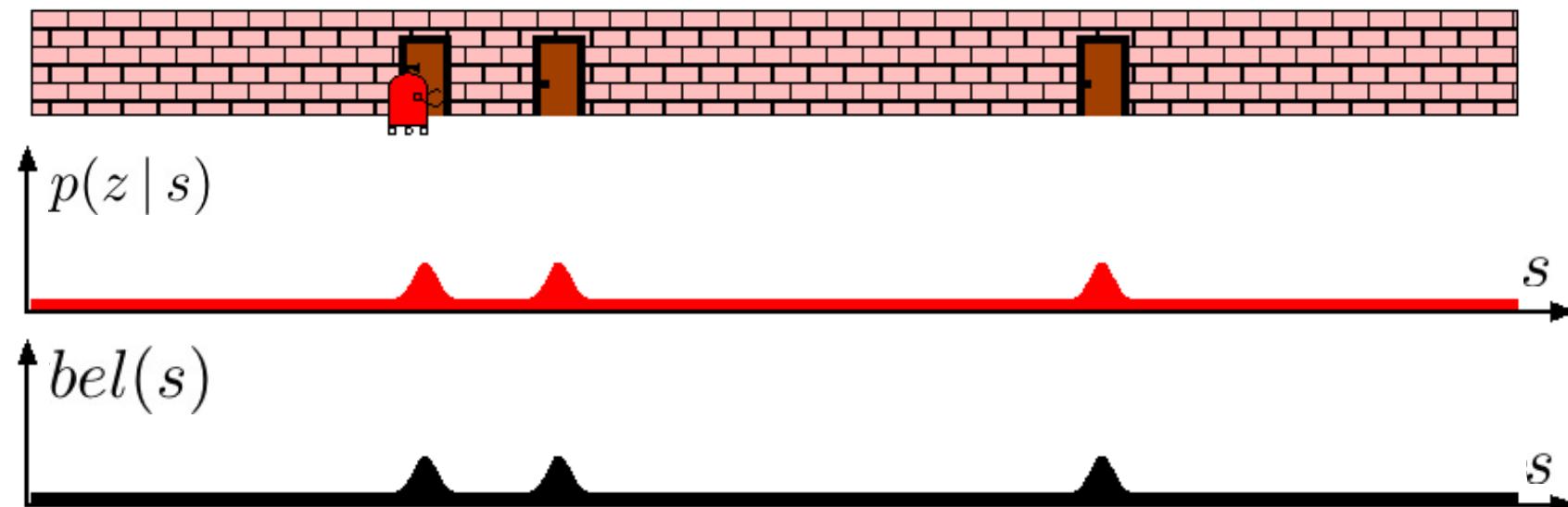
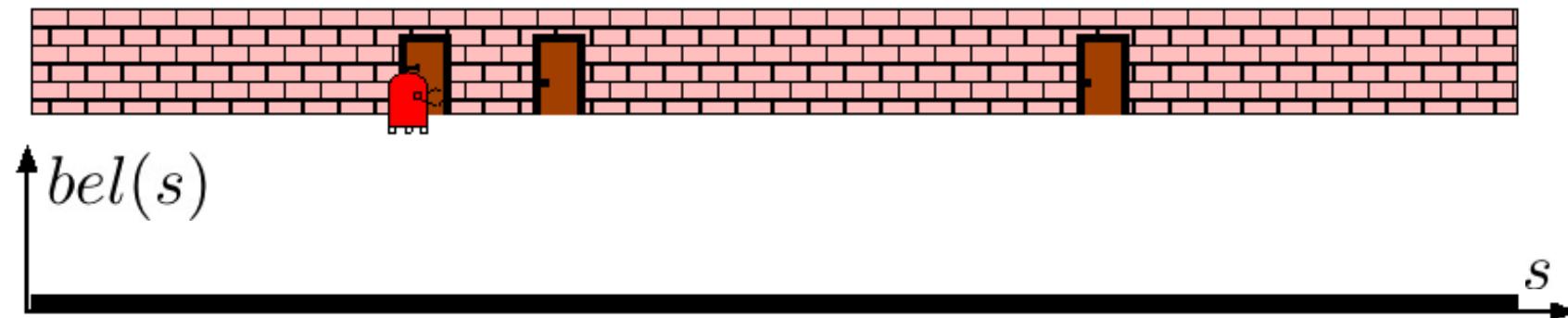
end

Return $bel(x_t)$

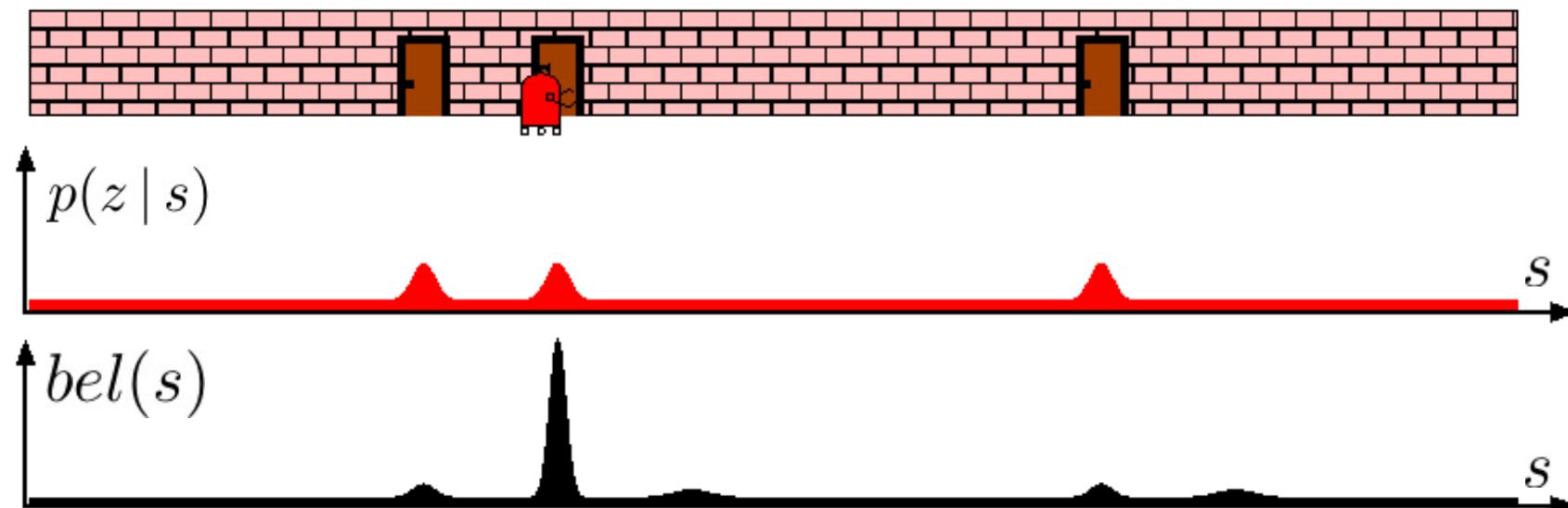
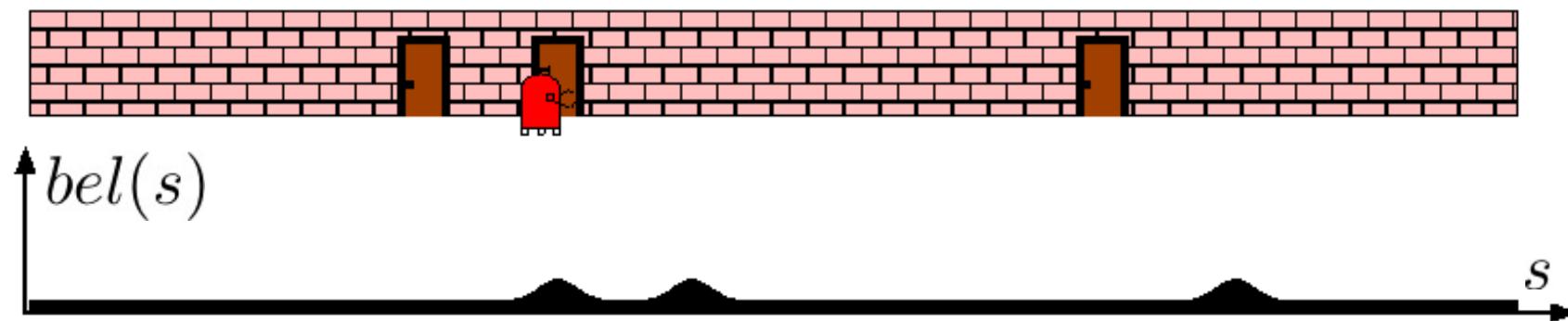
Markov localization: typical choices for initial belief

- Initial belief, $bel(x_0)$ reflects initial knowledge of robot pose
- For position tracking
 - If initial pose is known, $bel(x_0) = \begin{cases} 1 & \text{if } x_0 = \bar{x}_0 \\ 0 & \text{otherwise} \end{cases}$
 - If partially known, $bel(x_0) \sim \mathcal{N}(\bar{x}_0, \Sigma_0)$
- For global localization
 - If initial pose is unknown, $bel(x_0) = 1/|X|$

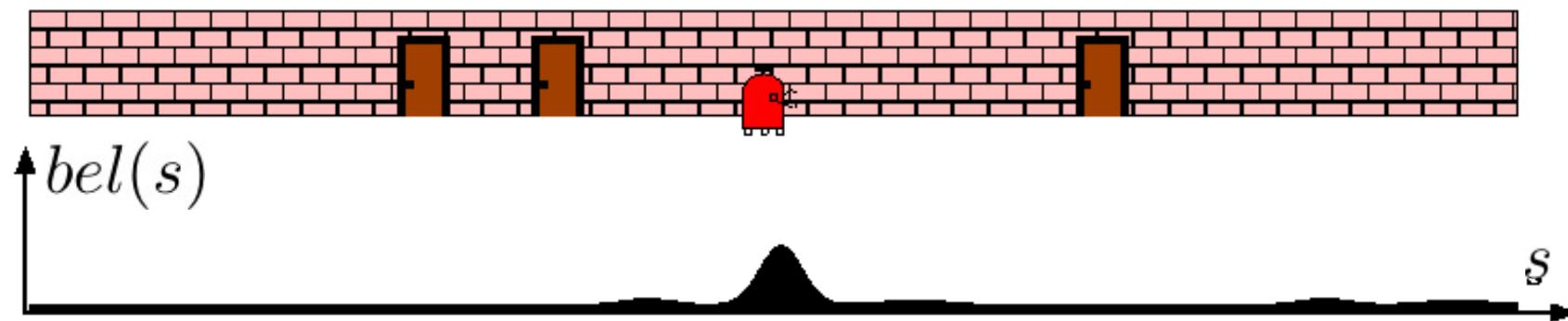
Markov localization: example



Markov localization: example



Markov localization: example



Instantiation of Markov localization

- To make algorithm tractable, we need to add some structure to the representation of $bel(x_t)$
 1. Gaussian representation
 2. Particle filter representation

Extended Kalman filter (EKF) localization

- **Key idea:** represent belief $bel(x_t)$ by its first and second moment, i.e., μ_t and Σ_t
- We will develop the EKF localization algorithm under the assumptions that:

1. A **feature-based map** is available, consisting of point landmarks

$$m = \{m_1, m_2, \dots\}, \quad m_i = (m_{i,x}, m_{i,y}) \xleftarrow{\text{Location of the landmark in the global coordinate frame}}$$

2. There is a sensor that can measure **the range r and the bearing ϕ** of the landmarks relative to the robot's local coordinate frame

- Key concepts carry forward to other map / sensing models

Range and bearing sensors

- Range & bearing sensors are common: features extracted from range scans and stereo vision come with range r and bearing ϕ information
- At time t , a **set** of features is measured (assumed independent)

$$z_t = \{z_t^1, z_t^2, \dots\} = \{(r_t^1, \phi_t^1), (r_t^2, \phi_t^2), \dots\}$$

- Assuming that the i -th measurement at time t corresponds to the j -th landmark in the map, the measurement model is

$$\begin{pmatrix} r_t^i \\ \phi_t^i \end{pmatrix} = \underbrace{\begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ \text{atan2}(m_{j,y} - y, m_{j,x} - x) - \theta \end{pmatrix}}_{=h(x_t, j, m)} + \mathcal{N}(0, Q_t)$$

Gaussian noise 

The issue of data association

- **Data association problem:** uncertainty may exists regarding the identity of a landmark
- Formally, we define a *correspondence variable* between measurement z_t^i and landmark m_j in the map as (assume N landmarks)
$$c_t^i \in \{1, \dots, N + 1\}$$
 - $c_t^i = j \leq N$ if i -th measurement at time t corresponds to j -th landmark
 - $c_t^i = N + 1$ if a measurement does not correspond to any landmark
- Two versions of the localization problem
 1. Correspondence variables are known
 2. Correspondence variables are not known (usual case)

EKF localization with known correspondences

- Algorithm is derived from EKF filter
- Assume motion model (in our case, differential drive robot)

$$x_t = g(u_t, x_{t-1}) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, R_t), \quad G_t := J_g(u_t, \mu_{t-1})$$

- Assume range and bearing measurement model

$$z_t^i = h(x_i, j, m) + \delta_t, \quad \delta_t \sim \mathcal{N}(0, Q_t), \quad H_t^i := \frac{\partial h(\bar{\mu}_t, j, m)}{\partial x_t}$$

$$\frac{\partial h(\bar{\mu}_t, j, m)}{\partial x_t} = \begin{pmatrix} \frac{\partial r_t^i}{\partial \bar{\mu}_{t,x}} & \frac{\partial r_t^i}{\partial \bar{\mu}_{t,y}} & \frac{\partial r_t^i}{\partial \bar{\mu}_{t,\theta}} \\ \frac{\partial \phi_t^i}{\partial \bar{\mu}_{t,x}} & \frac{\partial \phi_t^i}{\partial \bar{\mu}_{t,y}} & \frac{\partial \phi_t^i}{\partial \bar{\mu}_{t,\theta}} \end{pmatrix} = \begin{pmatrix} -\frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{(m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2}} & -\frac{m_{j,y} - \bar{\mu}_{t,y}}{\sqrt{(m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2}} & 0 \\ \frac{m_{j,y} - \bar{\mu}_{t,y}}{(m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2} & -\frac{m_{j,x} - \bar{\mu}_{t,x}}{(m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2} & -1 \end{pmatrix}$$

$$Q_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix}$$

EKF localization with known correspondences

- Main difference with EKF filter: multiple measurements are processed at the same time
- We exploit conditional independence assumption

$$p(z_t | x_t, c_t, m) = \prod_i p(z_t^i | x_t, c_t^i, m)$$

- Such assumption allows us to incrementally add the information, as if there was zero motion in between measurements

Data: $(\mu_{t-1}, \Sigma_{t-1}), u_t, z_t, \mathbf{c}_t, \mathbf{m}$
Result: (μ_t, Σ_t)

$$\bar{\mu}_t = g(u_t, \mu_{t-1}) ;$$
$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t;$$

foreach $z_t^i = (r_t^i, \phi_t^i)^T$ **do**

$$j = c_t^i;$$
$$\hat{z}_t^i = \begin{pmatrix} \sqrt{(m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2} \\ \text{atan2}(m_{j,y} - \bar{\mu}_{t,y}, m_{j,x} - \bar{\mu}_{t,x}) - \bar{\mu}_{t,\theta} \end{pmatrix};$$
$$S_t^i = H_t^i \bar{\Sigma}_t [H_t^i]^T + Q_t;$$
$$K_t^i = \bar{\Sigma}_t [H_t^i]^T [S_t^i]^{-1};$$
$$\bar{\mu}_t = \bar{\mu}_t + K_t^i (z_t^i - \hat{z}_t^i);$$
$$\bar{\Sigma}_t = (I - K_t^i H_t^i) \bar{\Sigma}_t;$$

end

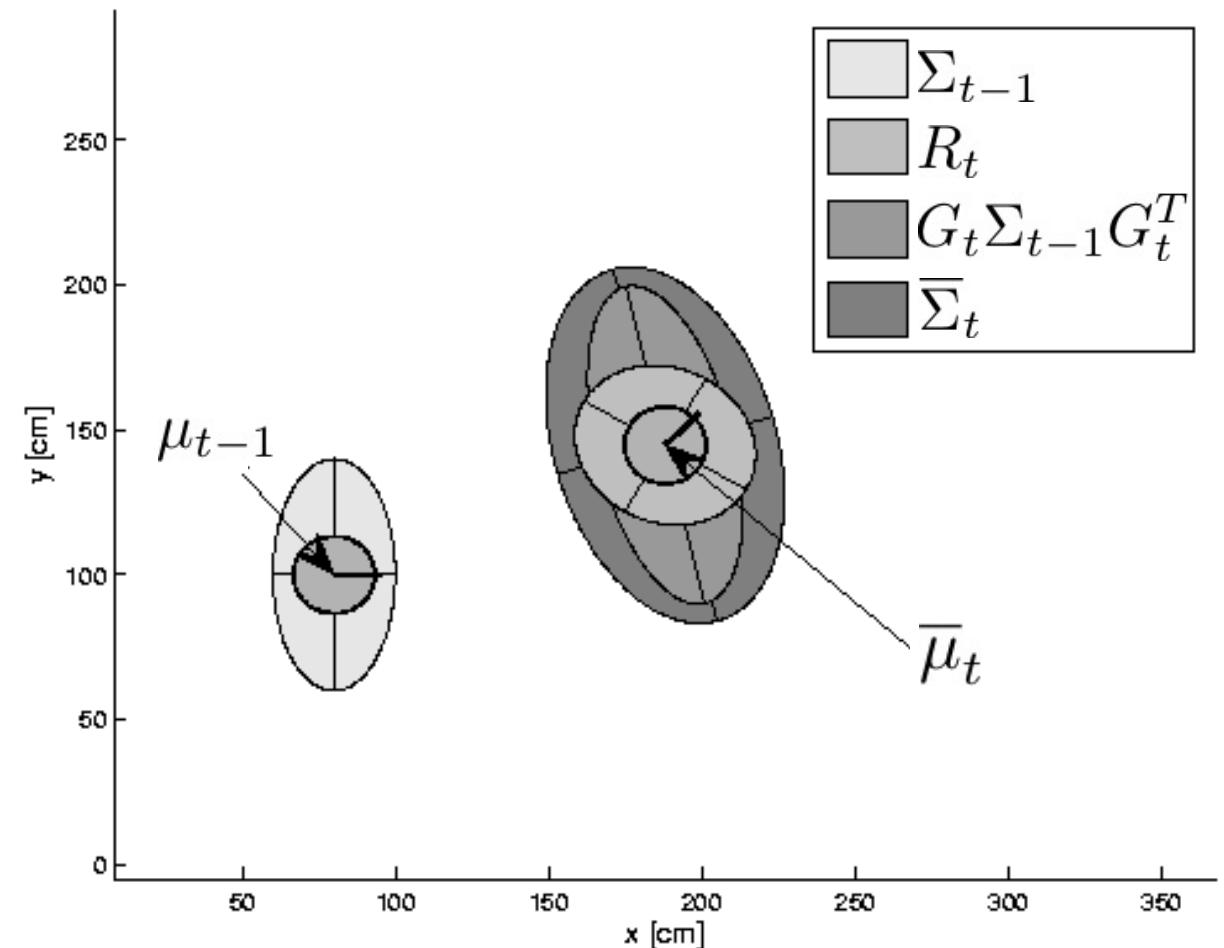
$$\mu_t = \bar{\mu}_t \text{ and } \Sigma_t = \bar{\Sigma}_t;$$

Return (μ_t, Σ_t)

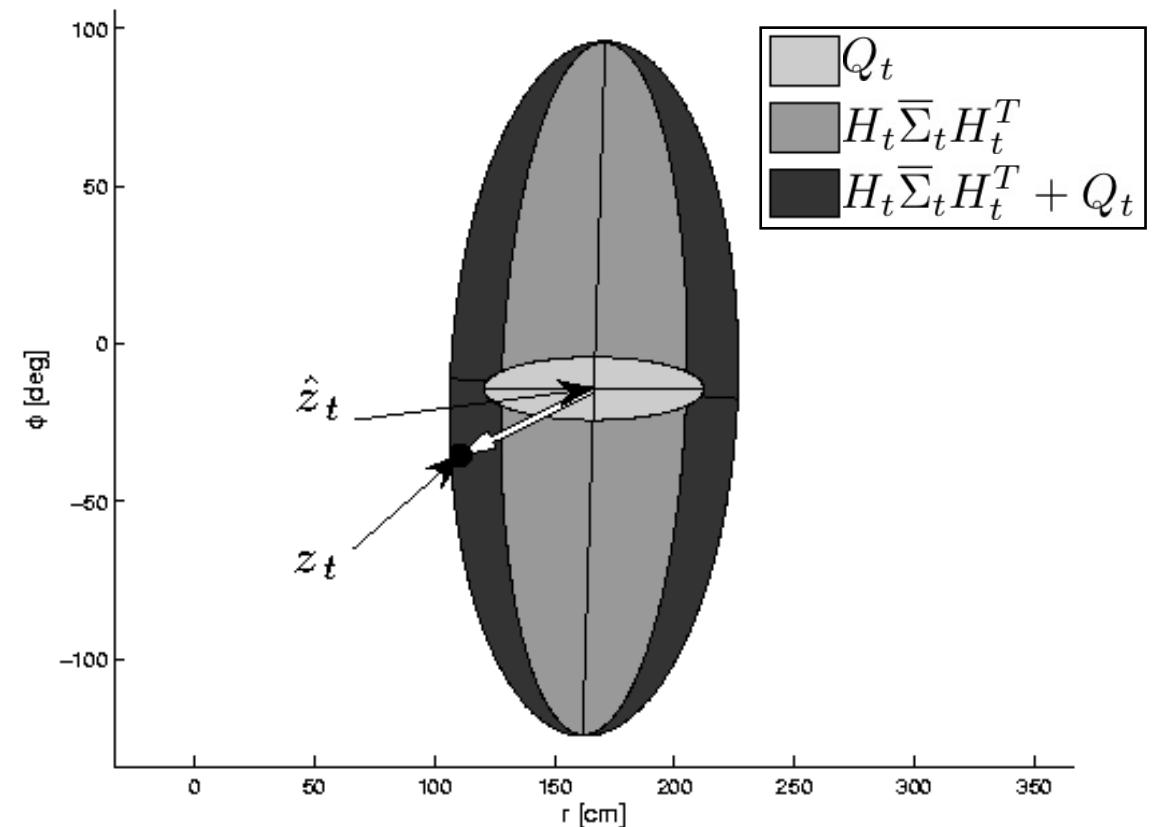
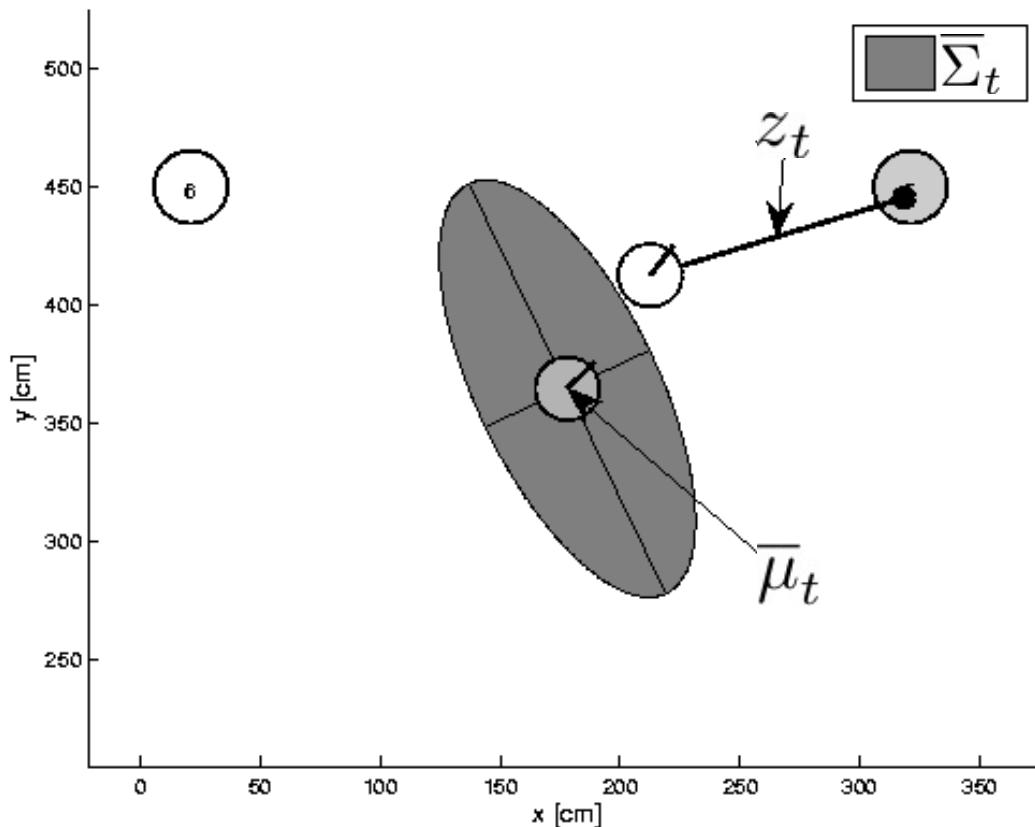
Innovation covariance

Example of EKF-localization: prediction step

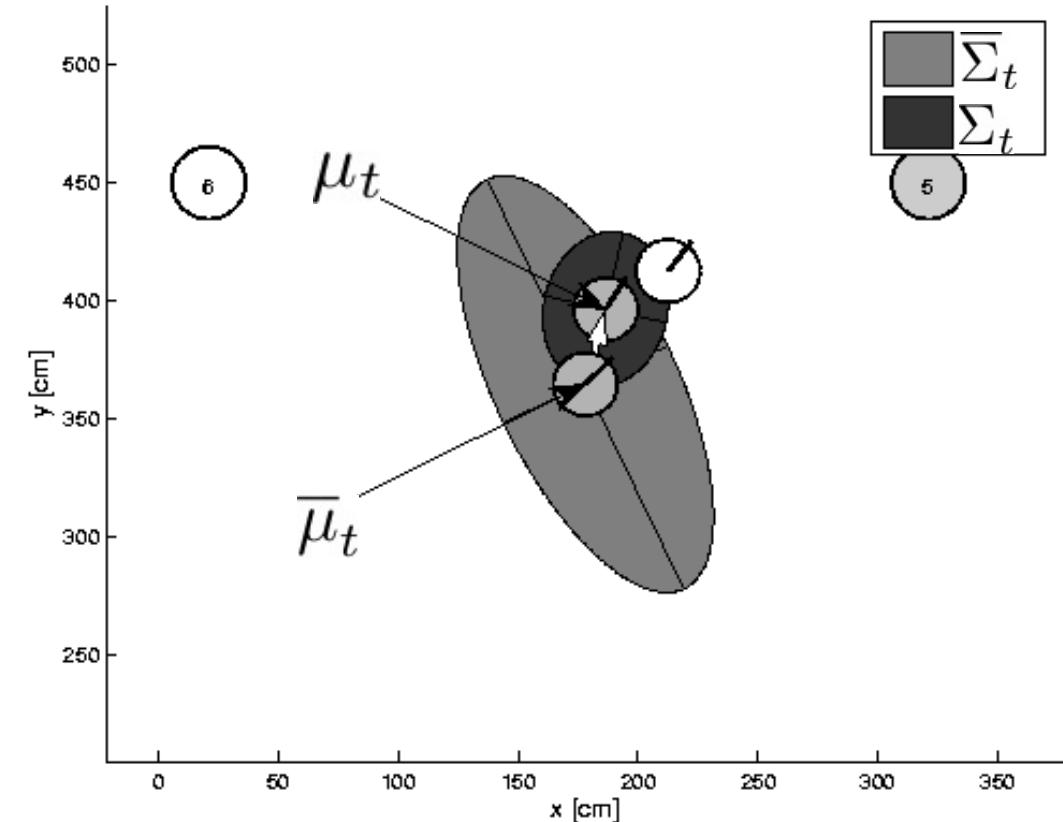
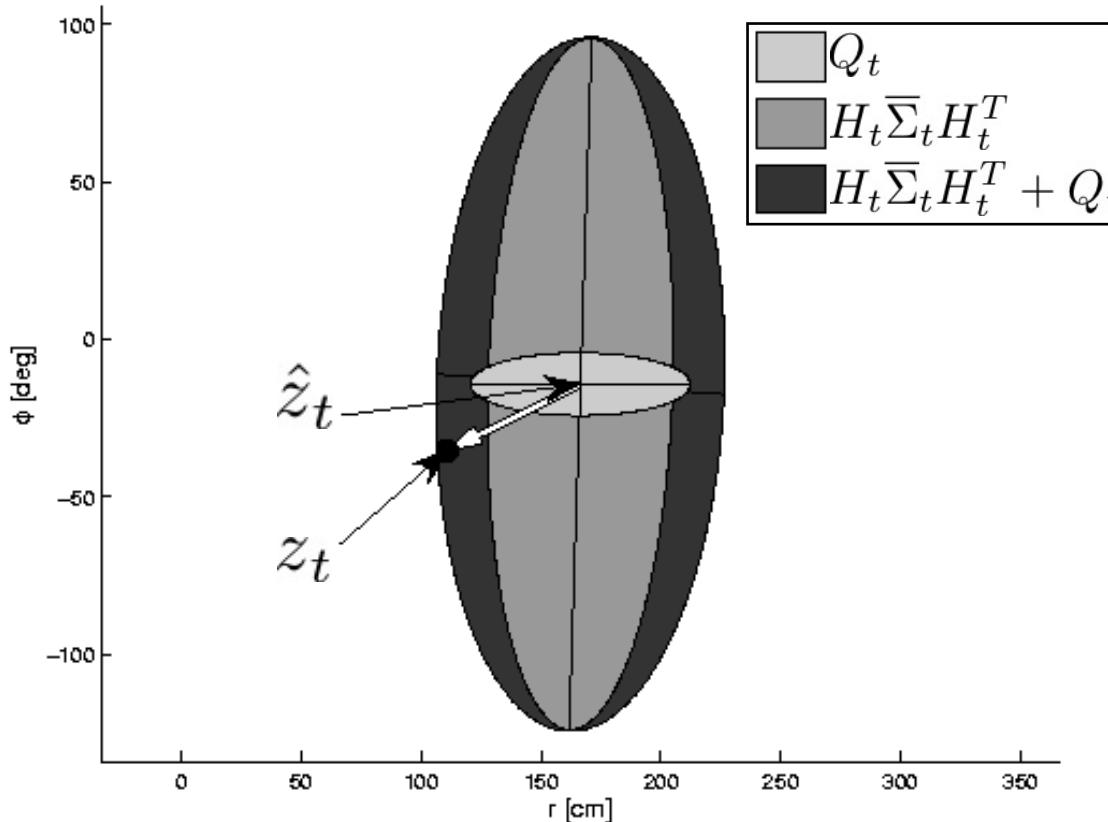
- Observations measure relative distance and bearing to a marker
- For simplicity, we assume that the robot detects only one marker at a time



Example of EKF-localization: measurement prediction step



Example of EKF-localization: correction step



EKF localization with unknown correspondences

- **Key idea:** determine the identity of a landmark during localization via maximum likelihood estimation, whereby one first determines the most likely value of c_t , and then takes this value for granted
- Formally, the maximum likelihood estimator determines the correspondence that maximizes the data likelihood

$$\hat{c}_t = \arg \max_{c_t} p(z_t | c_{1:t}, m, z_{1:t-1}, u_{1:t})$$

- Challenge: there are exponentially many terms in the maximization above!
- Solution: perform maximization *separately* for each z_t^i

Estimating the correspondence variables

- Step #1: find

$$p(\mathbf{z}_t^i \mid c_{1:t}, m, z_{1:t-1}, u_{1:t})$$

- Derivation (sketch)

$$\begin{aligned} p(z_t^i \mid c_{1:t}, m, z_{1:t-1}, u_{1:t}) &= \int p(z_t^i \mid x_t, c_{1:t}, m, z_{1:t-1}, u_{1:t}) p(x_t \mid c_{1:t}, m, z_{1:t-1}, u_{1:t}) dx_t \\ &= \int p(z_t^i \mid x_t, c_t^i, m) \cdot \overline{bel}(x_t) dx_t \\ &\sim \mathcal{N}(h(x_t, c_t^i, m), Q_t) \quad \text{---} \quad \overline{bel}(x_t) \sim \mathcal{N}(\bar{\mu}_t, \bar{\Sigma}_t) \\ &\approx \mathcal{N}(h(\bar{\mu}_t, c_t^i, m) + H_t^i(x_t - \bar{\mu}_t), Q_t) \end{aligned}$$

Estimating the correspondence variables

- Performing the algebraic calculations

$$p(z_t^i \mid c_{1:t}, m, z_{1:t-1}, u_{1:t}) \approx \mathcal{N}(h(\bar{\mu}_t, c_t^i, m), H_t^i \bar{\Sigma}_t [H_t^i]^T + Q_t)$$

- Step #2: estimate correspondence as

$$\begin{aligned}\hat{c}_t^i &= \arg \max_{c_t^i} p(z_t^i \mid c_{1:t}, m, z_{1:t-1}, u_{1:t}) \\ &\approx \arg \max_{c_t^i} \mathcal{N}(z_t^i; h(\bar{\mu}_t, c_t^i, m), H_t \bar{\Sigma}_t H_t^T + Q_t)\end{aligned}$$

EKF localization with unknown correspondences

- Same as before, plus the inclusion of a maximum likelihood estimator for the correspondence variables

Data: $(\mu_{t-1}, \Sigma_{t-1}), u_t, z_t, m$
Result: (μ_t, Σ_t)

$$\bar{\mu}_t = g(u_t, \mu_{t-1}) ;$$
$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t;$$

foreach $z_t^i = (r_t^i, \phi_t^i)^T$ **do**

foreach landmark k in the map **do**

$\hat{z}_t^k = \begin{pmatrix} \sqrt{(m_{k,x} - \bar{\mu}_{t,x})^2 + (m_{k,y} - \bar{\mu}_{t,y})^2} \\ \text{atan2}(m_{k,y} - \bar{\mu}_{t,y}, m_{k,x} - \bar{\mu}_{t,x}) - \bar{\mu}_{t,\theta} \end{pmatrix};$

$S_t^k = H_t^k \bar{\Sigma}_t [H_t^k]^T + Q_t;$

end

$j(i) = \arg \max_k \mathcal{N}(z_t^i; \hat{z}_t^k, S_t^k)$

$K_t^i = \bar{\Sigma}_t [H_t^{j(i)}]^T [S_t^{j(i)}]^{-1};$

$\bar{\mu}_t = \bar{\mu}_t + K_t^i (z_t^i - \hat{z}_t^{j(i)});$

$\bar{\Sigma}_t = (I - K_t^i H_t^{j(i)}) \bar{\Sigma}_t;$

end

$\mu_t = \bar{\mu}_t$ and $\Sigma_t = \bar{\Sigma}_t$;

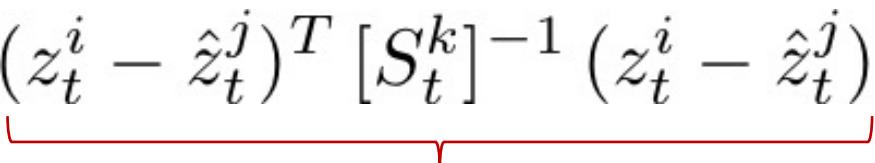
Return (μ_t, Σ_t)

Correspondence estimation

Comments

- Alternative approach to estimate correspondences is to use a *validation gate*:

Match landmark j with measurement i if $(z_t^i - \hat{z}_t^j)^T [S_t^k]^{-1} (z_t^i - \hat{z}_t^j) \leq \gamma$



Mahalanobis distance

- A more general approach to deal with data association is the multi-hypothesis tracking filter, where a belief is represented by a mixture of Gaussians (each tracking a sequence of data association decisions)
- UKF localization is another popular approach for feature-based localization

Comments

- Other popular features include lines, corners, distinct patterns
- In the case of lines, an observation would be

$$z_t^i = \begin{bmatrix} r_t^i \\ \alpha_t^i \end{bmatrix}$$

Monte Carlo localization (MCL)

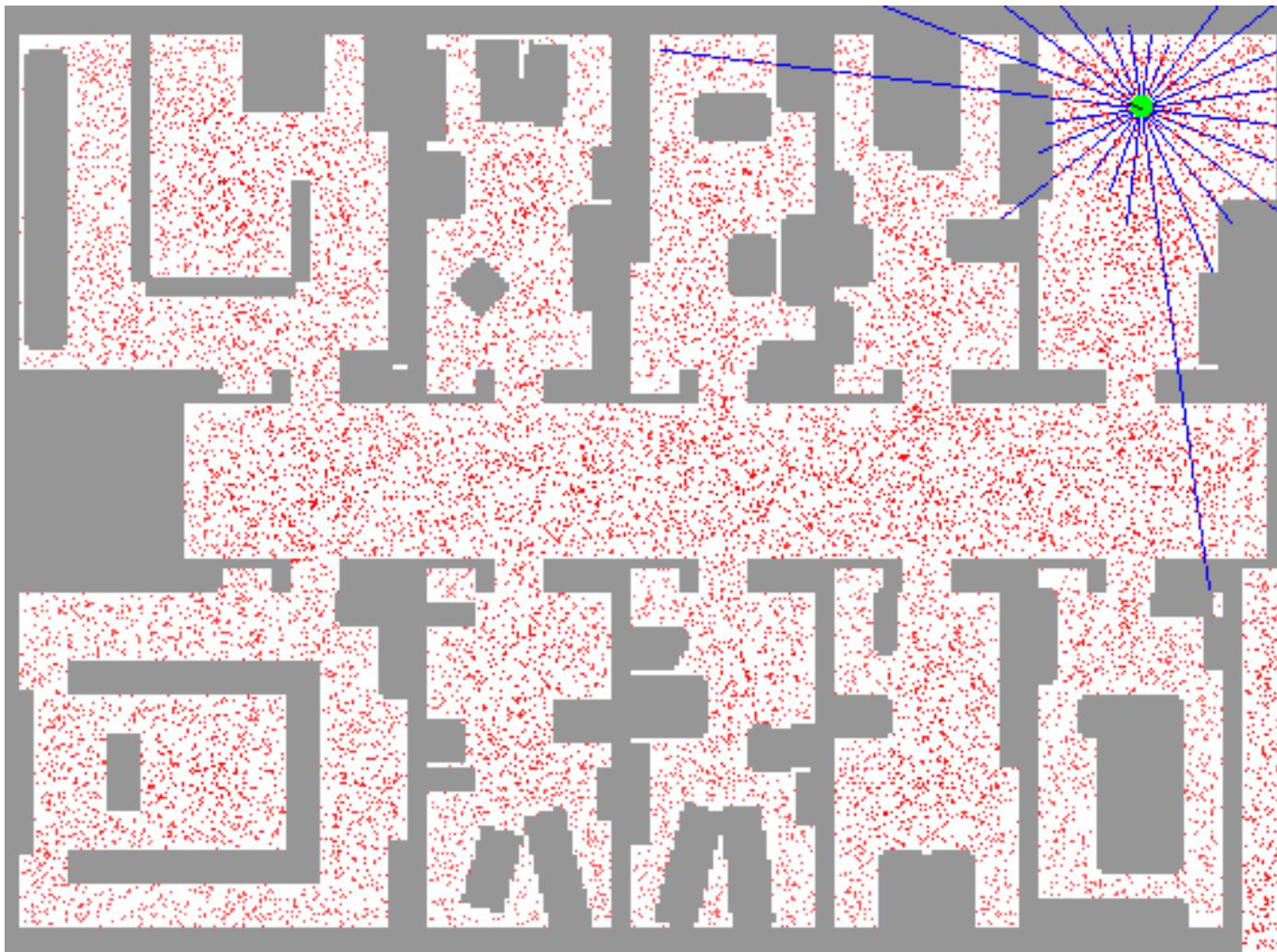
- **Key idea:** represent belief $bel(x_t)$ by a set of M particles

$$\mathcal{X}_t = \{x_t^{[1]}, x_t^{[2]}, \dots, x_t^{[M]}\}$$

- Requires a map m as input
- Addresses:
 - Global localization
 - Position tracking
 - Kidnapped robot problem (by injecting random particles)
- Can handle dynamic environments via outlier rejection

Data: $\mathcal{X}_{t-1}, u_t, z_t, \textcolor{red}{m}$
Result: \mathcal{X}_t
 $\overline{\mathcal{X}}_t = \mathcal{X}_t = \emptyset;$
for $i = 1$ **to** M **do**
 | Sample $x_t^{[m]} \sim p(x_t | u_t, x_{t-1}^{[m]}, \textcolor{red}{m})$;
 | $w_t^{[m]} = p(z_t | x_t^{[m]}, \textcolor{red}{m})$;
 | $\overline{\mathcal{X}}_t = \overline{\mathcal{X}}_t \cup (x_t^{[m]}, w_t^{[m]})$;
 | **end**
 | **for** $i = 1$ **to** M **do**
 | | Draw i with probability $\propto w_t^{[i]}$;
 | | Add $x_t^{[i]}$ to \mathcal{X}_t ;
 | **end**
 | Return \mathcal{X}_t

MCL: example



Next time

