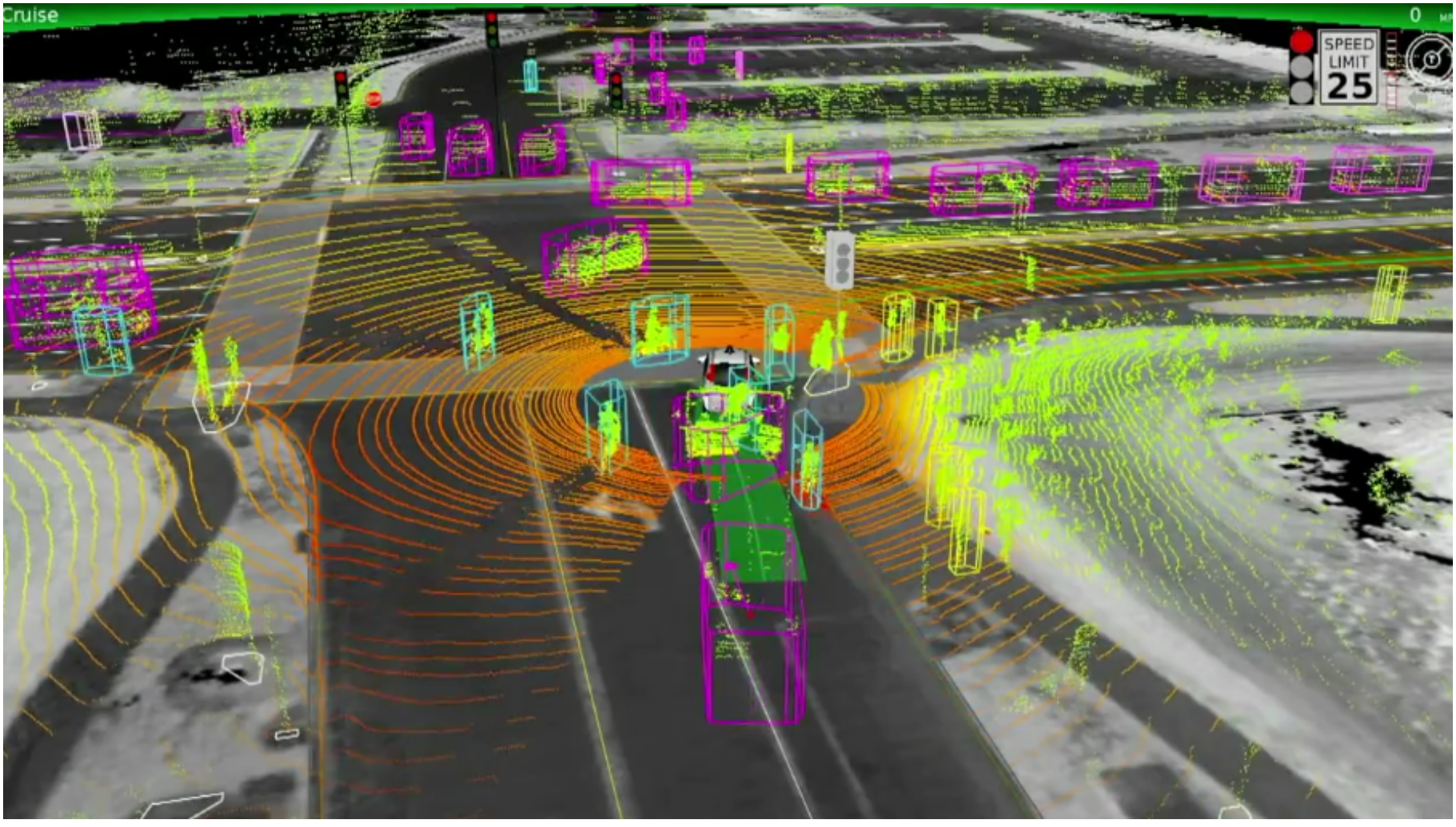
Robotics

Mobile Robotics

State estimation, Bayes filter, odometry, particle filter, Kalman filter, SLAM, joint Bayes filter, EKF SLAM, particle SLAM

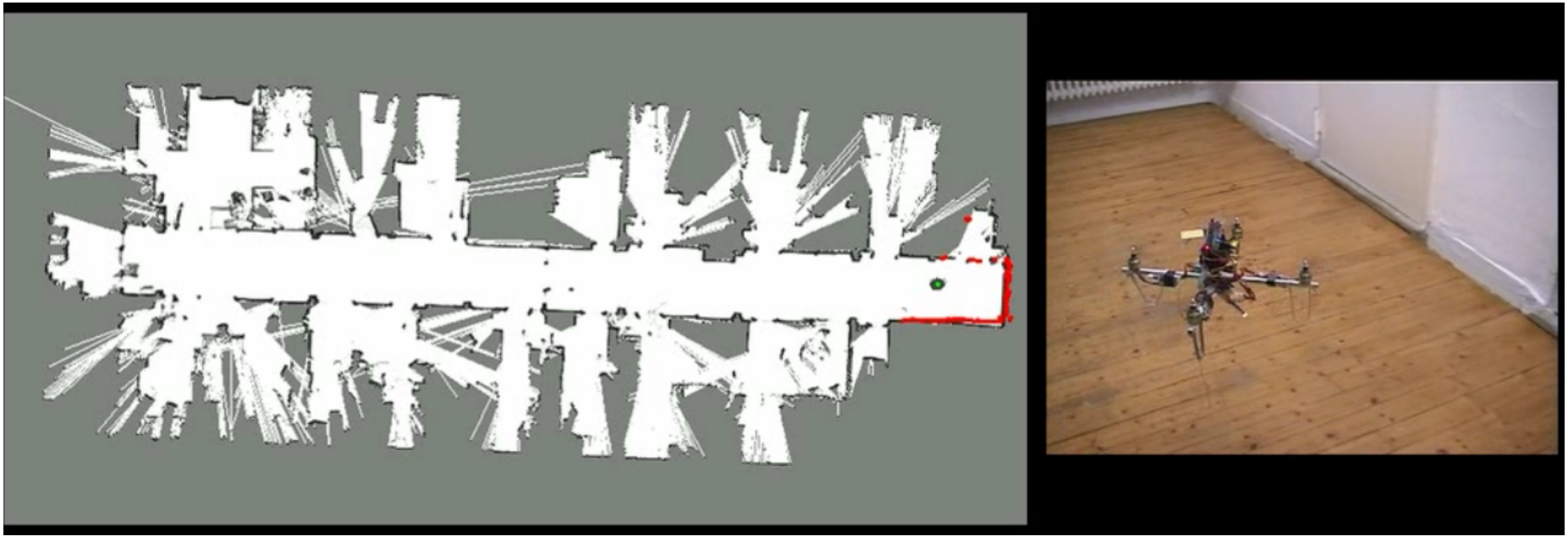
Marc Toussaint
University of Stuttgart
Winter 2016/17

Lecturer: Peter Englert



https://www.youtube.com/watch?v=tiwVMrTLUWg

Google Autonomous Car



http://www.slawomir.de/publications/grzonka09icra/grzonka09icra.pdf

Quadcopter Indoor Localization

http://stair.stanford.edu/multimedia.php

Outline

- PART I:
  A core challenge in mobile robotics is **state estimation**
  $\rightarrow$ Bayesian filtering & smoothing
  particle filter, Kalman filter

Outline

- PART I:
  A core challenge in mobile robotics is **state estimation**
  $\rightarrow$ Bayesian filtering & smoothing
  particle filter, Kalman filter

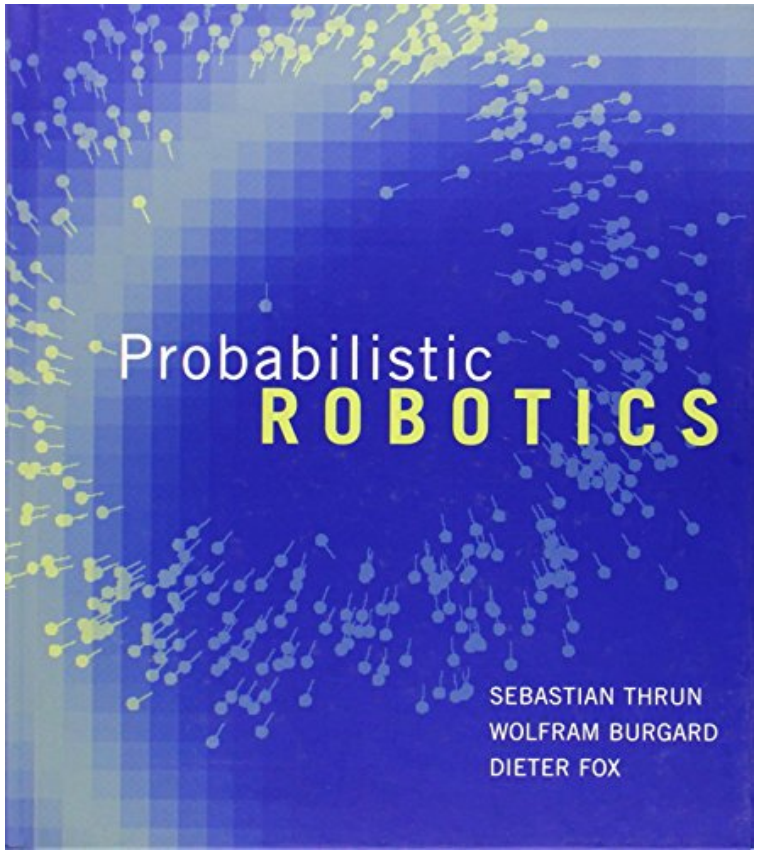
- PART II:
  Another challenge is to **build a map** while exploring
  $\rightarrow$ SLAM (simultaneous localization and mapping)

Outline

- PART I:
  A core challenge in mobile robotics is **state estimation**
  $\rightarrow$ Bayesian filtering & smoothing
  particle filter, Kalman filter

- PART II:
  Another challenge is to **build a map** while exploring
  $\rightarrow$ SLAM (simultaneous localization and mapping)

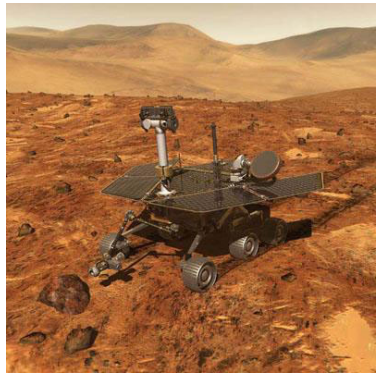
How to represent our knowledge? $\rightarrow$ **Probabilities**



Sebastian Thrun, Wolfram Burghard, Dieter Fox: Probabilistic Robotics

`probabilistic-robotics.org`

Types of Robot Mobility



- Each type of robot mobility corresponds to a
  system equation $x_t = x_{t-1} + \tau f(x_{t-1}, u_{t-1})$
  or, if the dynamics are stochastic,

$$P(x_t \mid x_{t-1}, u_{t-1}) = \mathcal{N}(x_t \mid x_{t-1} + \tau f(x_{t-1}, u_{t-1}), \Sigma)$$

- Each type of robot mobility corresponds to a
  system equation $x_t = x_{t-1} + \tau f(x_{t-1}, u_{t-1})$
  or, if the dynamics are stochastic,

$$P(x_t \,|\, x_{t-1}, u_{t-1}) = \mathcal{N}(x_t \,|\, x_{t-1} + \tau f(x_{t-1}, u_{t-1}), \Sigma)$$

- We considered control, path finding, and trajectory optimization

  For this we always assumed to know the state x_t of the robot (e.g., its posture/position)!

- Each type of robot mobility corresponds to a
  system equation $x_t = x_{t-1} + \tau f(x_{t-1}, u_{t-1})$
  or, if the dynamics are stochastic,

$$P(x_t \mid x_{t-1}, u_{t-1}) = \mathcal{N}(x_t \mid x_{t-1} + \tau f(x_{t-1}, u_{t-1}), \Sigma)$$

- We considered control, path finding, and trajectory optimization

  For this we always assumed to know the state x_t of the robot (e.g., its posture/position)!

- Now we assume to observe sensor readings y_t (e.g., laser scan, image) and to have a sensor/observation model $P(y_t \mid x_t)$.

State Estimation

State Estimation Problem

- Our sensory data does not provide sufficient information to determine our location.

- Given the local sensor readings y_t, the current state x_t (location, position) is *uncertain*.

  – which hallway?
  – which door exactly?
  – which heading direction?

State Estimation Problem

- What is the probability of being in front of room 154, given we see what is shown in the image?
- What is the probability given that we were just in front of room 156?
- What is the probability given that we were in front of room 156 and moved 15 meters?

Recall Bayes' theorem

$$P(X|Y) = \frac{P(Y|X)\ P(X)}{P(Y)}$$

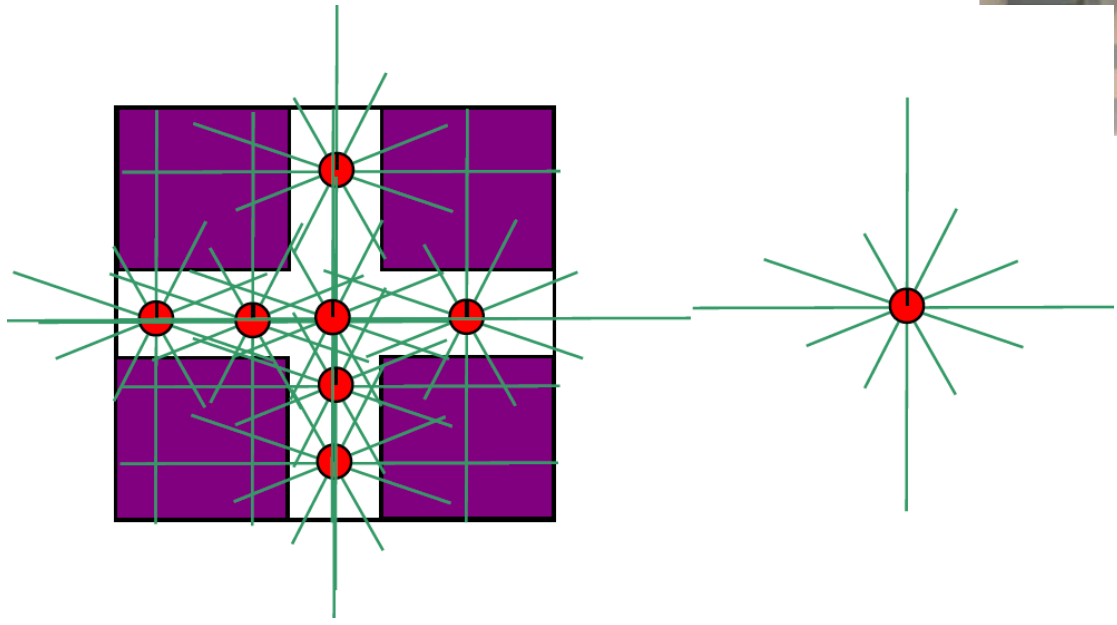
$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{(normalization)}}$$

- How can we apply this to the State Estimation Problem?

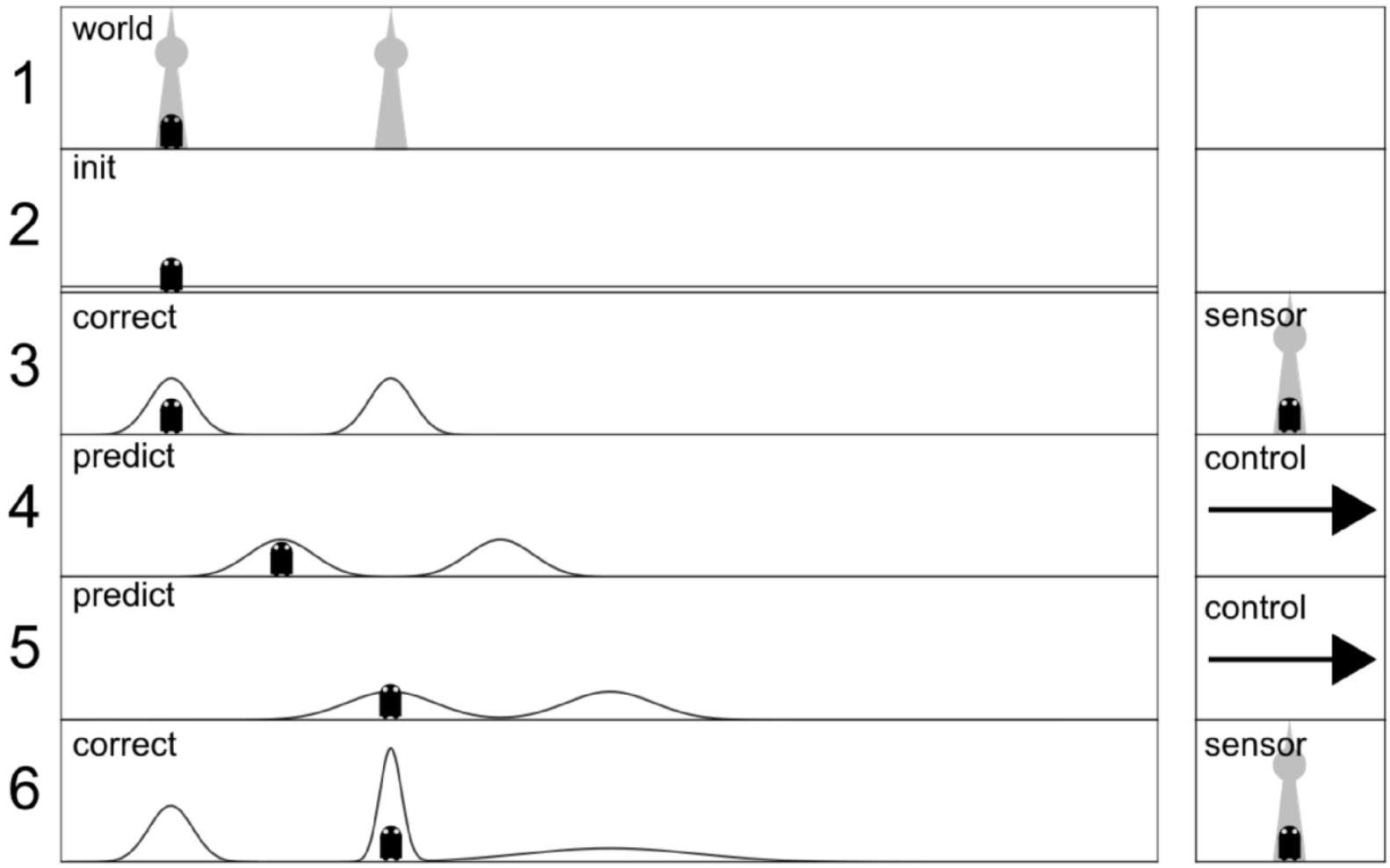
- How can we apply this to the State Estimation Problem?

Using Bayes Rule:

$$P(\text{location} \mid \text{sensor}) = \frac{P(\text{sensor} \mid \text{location}) P(\text{location})}{P(\text{sensor})}$$



Bayes Filter: 1d example



Bayes Filter

x_t = state (location) at time t
y_t = sensor readings at time t
$u_{t\text{-}1}$ = control command (action, steering, velocity) at time t-1
transition model $P(x_t|u_{t-1}, x_{t-1})$
observation model $P(y_t|x_t)$

- Given the history $y_{0:t}$ and $u_{0:t\text{-}1}$, we want to compute the probability distribution over the state at time t

$$p_t(x_t) := P(x_t \mid y_{0:t}, u_{0:t\text{-}1})$$

Bayes Filter

$$p_t(x_t) := P(x_t \mid y_{0:t}, u_{0:t\text{-}1})$$

Bayes Filter

$$
\begin{aligned}
p_t(x_t) &:= P(x_t \mid y_{0:t}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_t \mid y_{0:t\text{-}1}, u_{0:t\text{-}1})
\end{aligned}
$$

using Bayes rule $P(X|Y,Z) = c\, P(Y|X,Z)\; P(X|Z)$ with some normalization constant c_t

Bayes Filter

$$
\begin{aligned}
p_t(x_t) &:= P(x_t \mid y_{0:t}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_t \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t) \; P(x_t \mid y_{0:t\text{-}1}, u_{0:t\text{-}1})
\end{aligned}
$$

uses conditional independence of the observation on past observations and controls

Bayes Filter

$$
\begin{aligned}
p_t(x_t) &:= P(x_t \mid y_{0:t}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_t \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t) \; P(x_t \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t) \int_{x_{t\text{-}1}} P(x_t, x_{t\text{-}1} \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1}
\end{aligned}
$$

by definition of the marginal

Bayes Filter

$$
\begin{aligned}
p_t(x_t) &:= P(x_t \mid y_{0:t}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_t \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t) \; P(x_t \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t) \int_{x_{t\text{-}1}} P(x_t, x_{t\text{-}1} \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1} \\
&= c_t \; P(y_t \mid x_t) \int_{x_{t\text{-}1}} P(x_t \mid x_{t\text{-}1}, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_{t\text{-}1} \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1}
\end{aligned}
$$

by definition of a conditional

Bayes Filter

$$
\begin{aligned}
p_t(x_t) &:= P(x_t \mid y_{0:t}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_t \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t) \; P(x_t \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t) \int_{x_{t\text{-}1}} P(x_t, x_{t\text{-}1} \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1} \\
&= c_t \; P(y_t \mid x_t) \int_{x_{t\text{-}1}} P(x_t \mid x_{t\text{-}1}, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_{t\text{-}1} \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1} \\
&= c_t \; P(y_t \mid x_t) \int_{x_{t\text{-}1}} P(x_t \mid x_{t\text{-}1}, u_{t\text{-}1}) \; P(x_{t\text{-}1} \mid y_{0:t\text{-}1}, u_{0:t\text{-}2}) \; dx_{t\text{-}1}
\end{aligned}
$$

given $x_{t\text{-}1}$, x_t depends only on the controls $u_{t\text{-}1}$ (Markov Property)

Bayes Filter

$$
\begin{aligned}
p_t(x_t) &:= P(x_t \mid y_{0:t}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_t \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t) \; P(x_t \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \mid x_t) \int_{x_{t\text{-}1}} P(x_t, x_{t\text{-}1} \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1} \\
&= c_t \; P(y_t \mid x_t) \int_{x_{t\text{-}1}} P(x_t \mid x_{t\text{-}1}, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_{t\text{-}1} \mid y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1} \\
&= c_t \; P(y_t \mid x_t) \int_{x_{t\text{-}1}} P(x_t \mid x_{t\text{-}1}, u_{t\text{-}1}) \; P(x_{t\text{-}1} \mid y_{0:t\text{-}1}, u_{0:t\text{-}2}) \; dx_{t\text{-}1} \\
&= c_t \; P(y_t \mid x_t) \int_{x_{t\text{-}1}} P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1}) \; p_{t\text{-}1}(x_{t\text{-}1}) \; dx_{t\text{-}1}
\end{aligned}
$$

- A Bayes filter updates the posterior belief $p_t(x_t)$ in each time step using the:
  - observation model $P(y_t \mid x_t)$
  - transition model $P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1})$
  - old estimate $p_{t-1}(x_{t-1})$

Bayes Filter

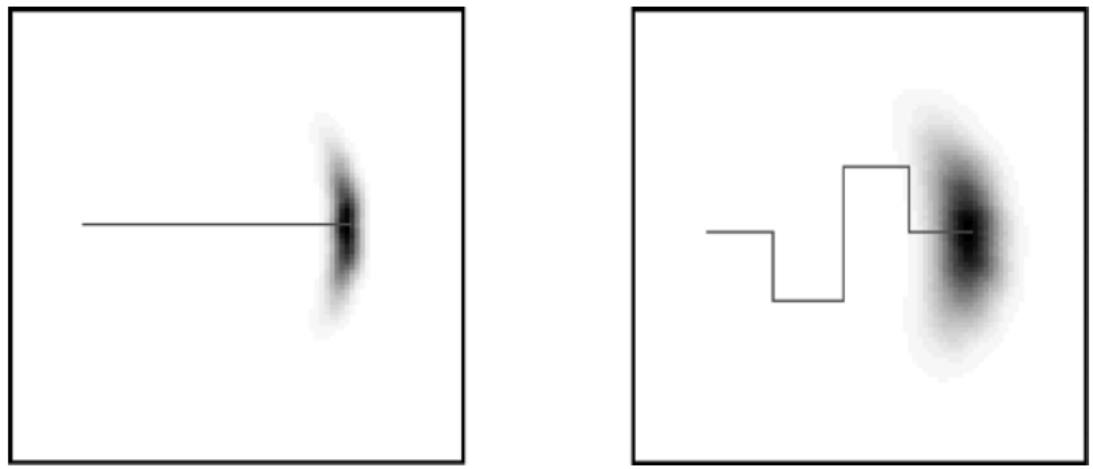
$$p_t(x_t) \;\propto\; \underbrace{P(y_t \mid x_t)}_{\text{new information}} \underbrace{\int_{x_{t\text{-}1}} P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1}) \; \underbrace{p_{t\text{-}1}(x_{t\text{-}1})}_{\text{old estimate}} \; dx_{t\text{-}1}}_{\text{predictive estimate } \hat{p}_t(x_t)}$$

1. We have a belief $p_{t\text{-}1}(x_{t\text{-}1})$ of our previous position

2. We use the motion model to predict the current position
$$\hat{p}_t(x_t) \propto \int_{x_{t\text{-}1}} P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1}) \; p_{t\text{-}1}(x_{t\text{-}1}) \; dx_{t\text{-}1}$$

3. We integetrate this with the current observation to get a better belief
$$p_t(x_t) \propto P(y_t \mid x_t) \; \hat{p}_t(x_t)$$

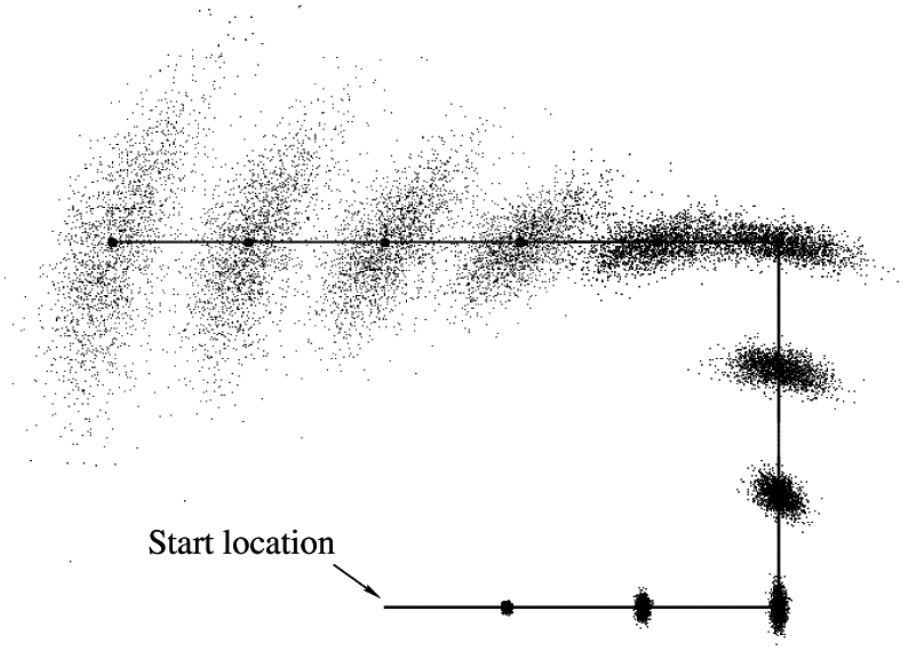
- Typical transition model $P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1})$ in robotics:



(from *Robust Monte Carlo localization for mobile robots* Sebastian Thrun, Dieter Fox, Wolfram Burgard, Frank Dellaert)

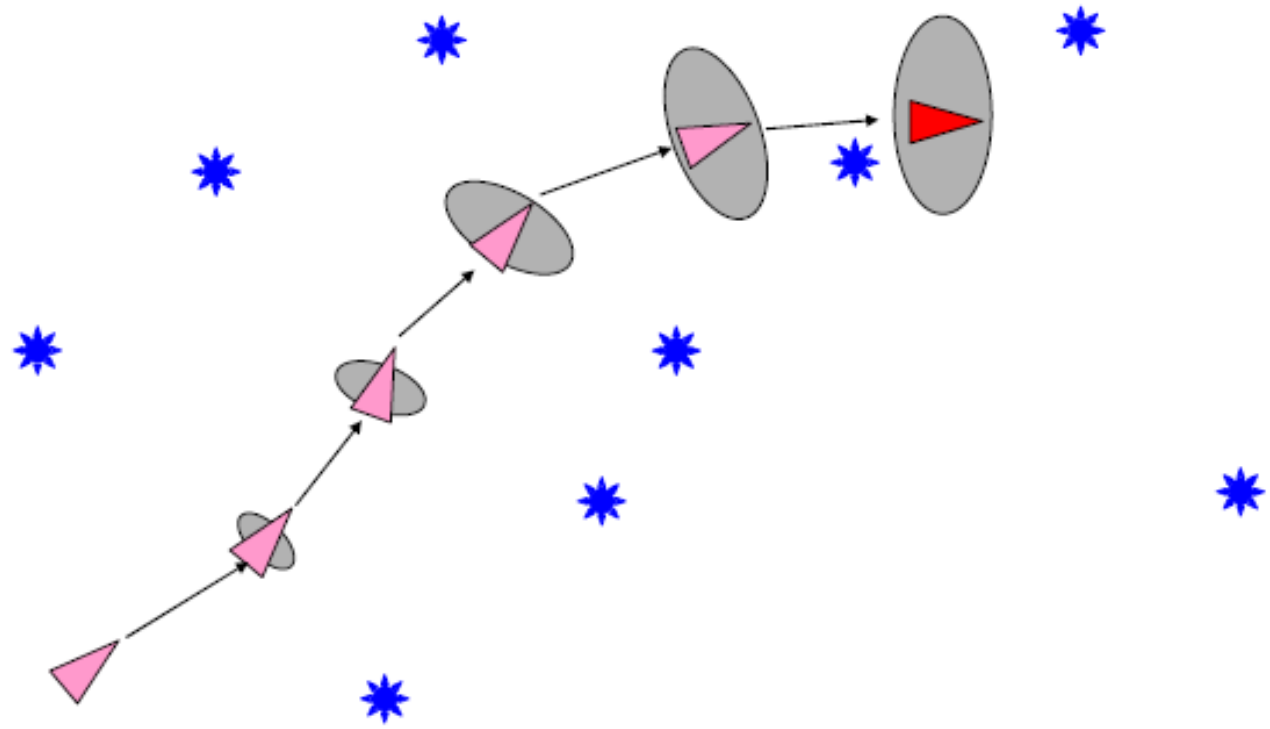
Odometry ("Dead Reckoning"): Filtering without observations

- The predictive distributions $\hat{p}_t(x_t)$ *without integrating observations* (removing the $P(y_t|x_t)$ part from the Bayesian filter)

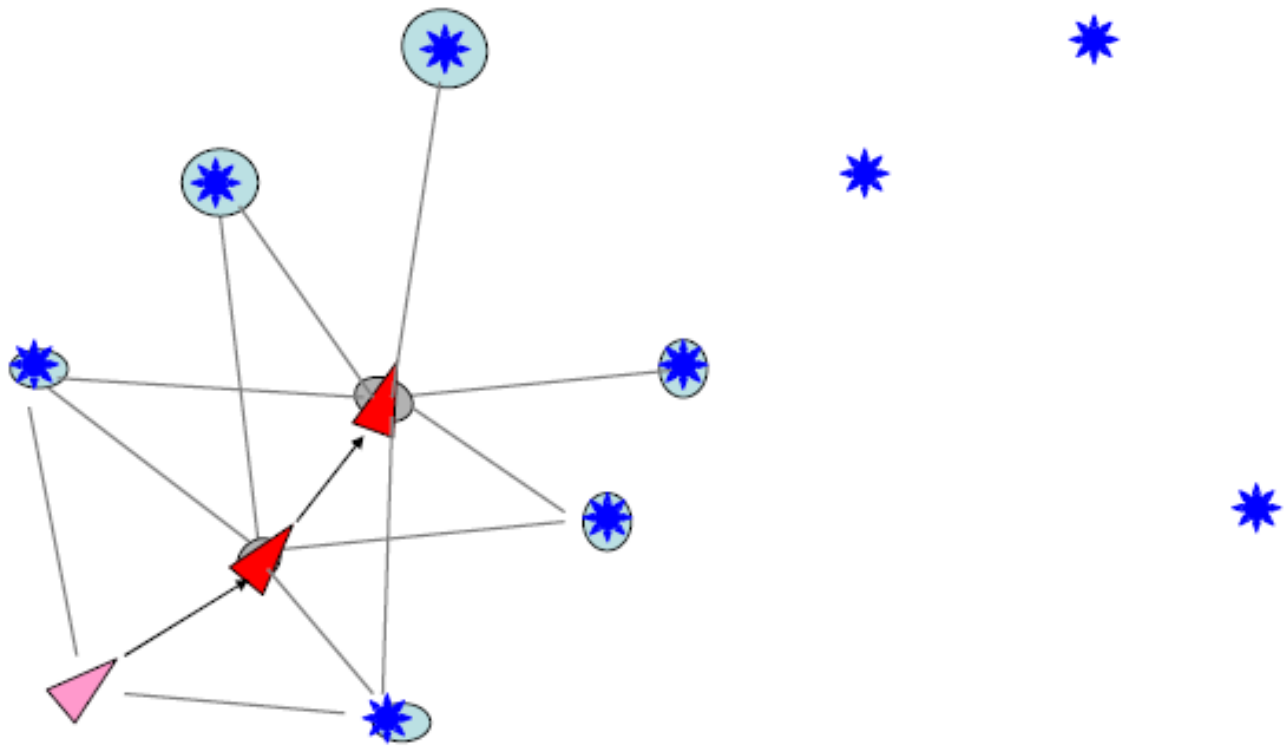


(from *Robust Monte Carlo localization for mobile robots* Sebastian Thrun, Dieter Fox, Wolfram Burgard, Frank Dellaert)

Again, predictive distributions $\hat{p}_t(x_t)$ *without integrating landmark observations*

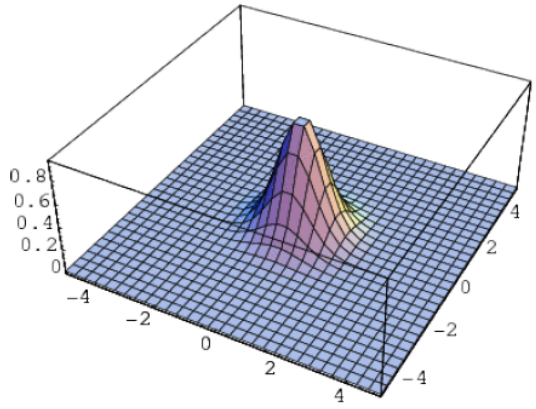


The Bayes-filtered distributions $p_t(x_t)$ integrating landmark observations



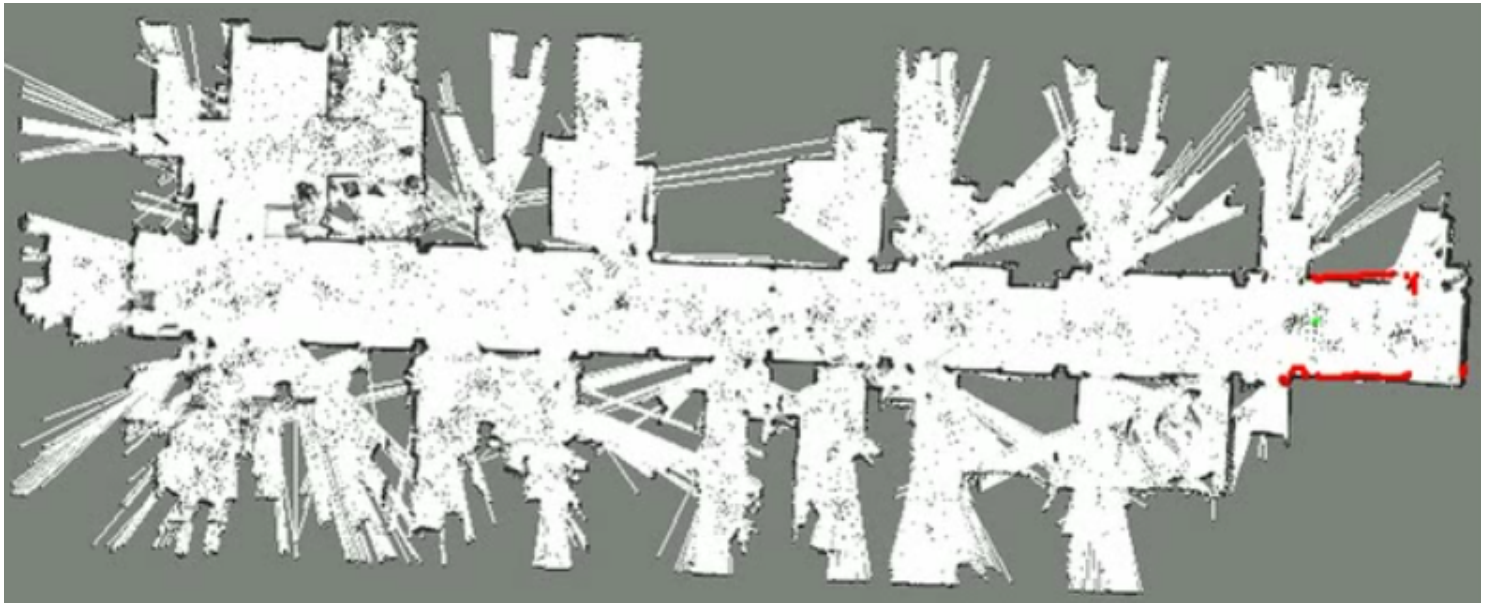
Bayesian Filters

- How to represent the belief $p_t(x_t)$:
- Gaussian



$$\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

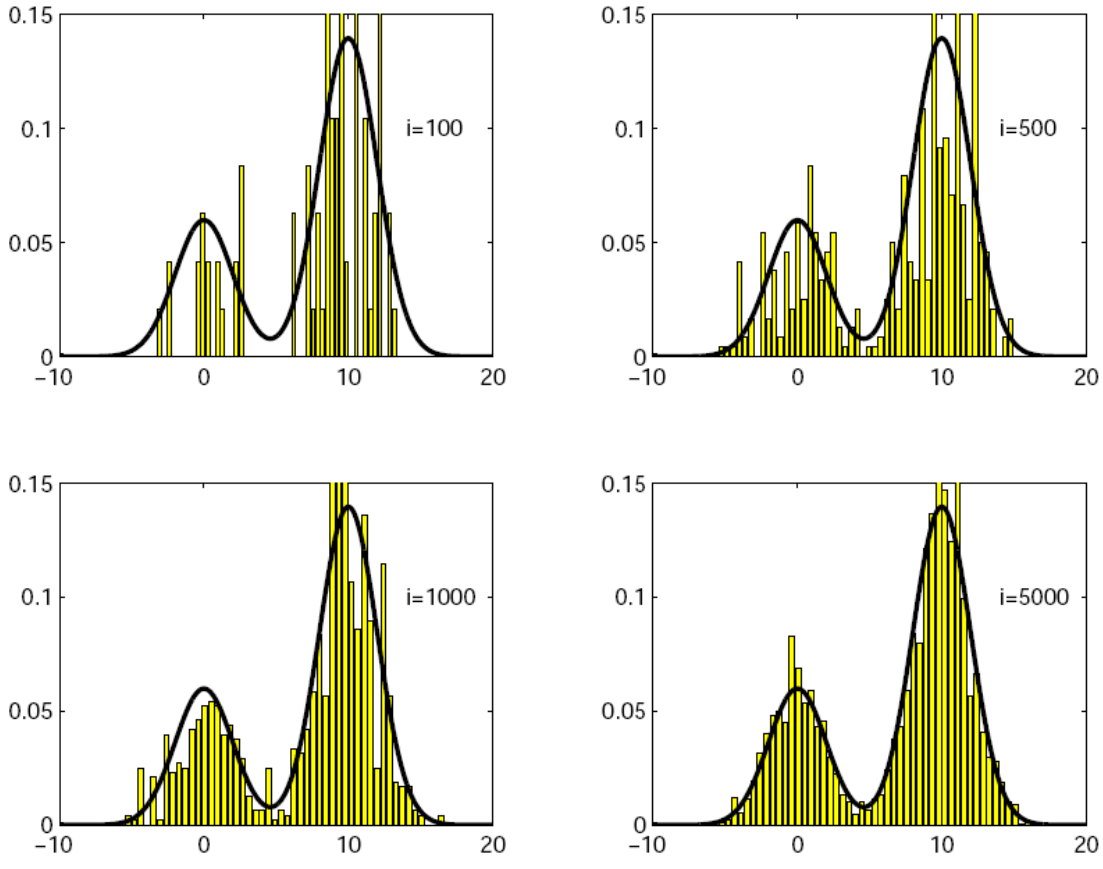
- Particles



Recall: Particle Representation of a Distribution

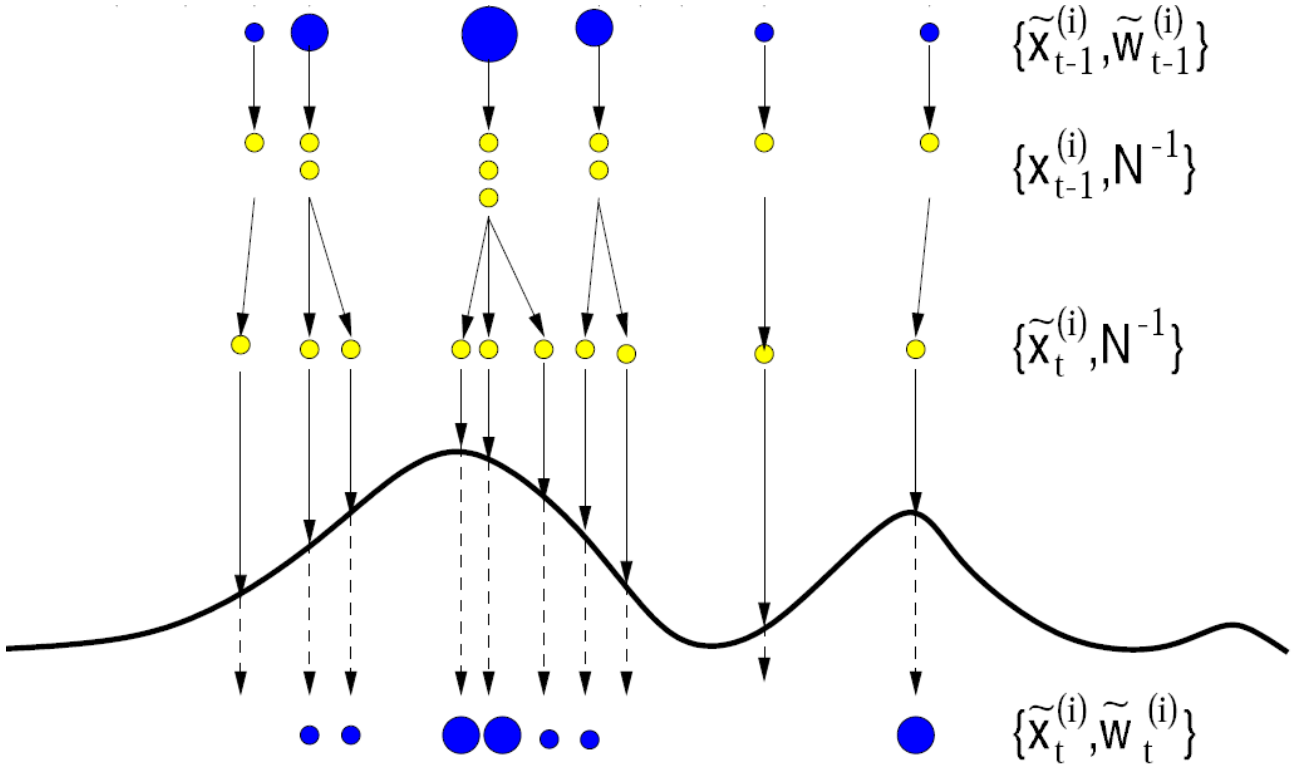
- Weighed set of N particles $\{(x^i, w^i)\}_{i=1}^N$

$$p(x) \approx q(x) := \sum_{i=1}^N w^i \delta(x, x^i)$$



Particle Filter := Bayesian Filtering with Particles

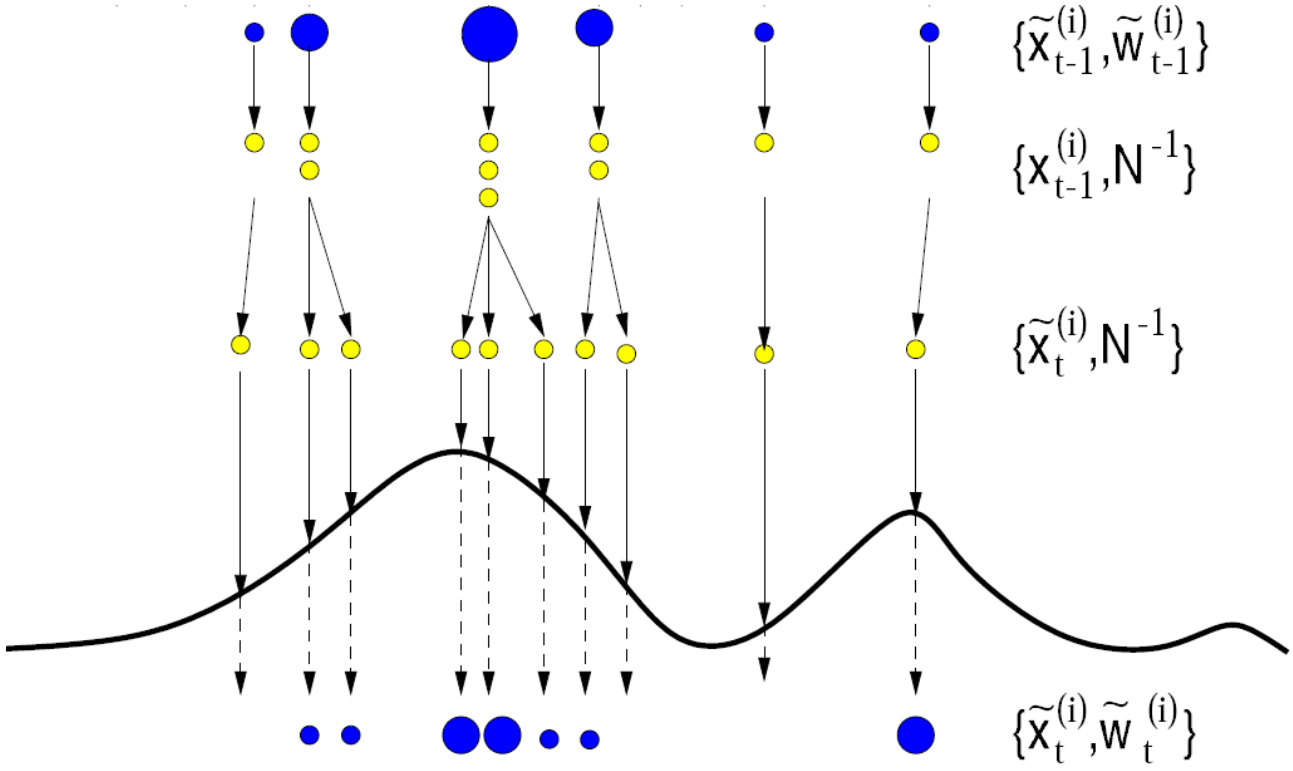
(Bayes Filter: $p_t(x_t) \;\propto\; P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})\; p_{t\text{-}1}(x_{t\text{-}1})\; dx_{t\text{-}1}$)



1. Start with N particles $\{(x^i_{t\text{-}1}, w^i_{t\text{-}1})\}_{i=1}^N$

Particle Filter := Bayesian Filtering with Particles

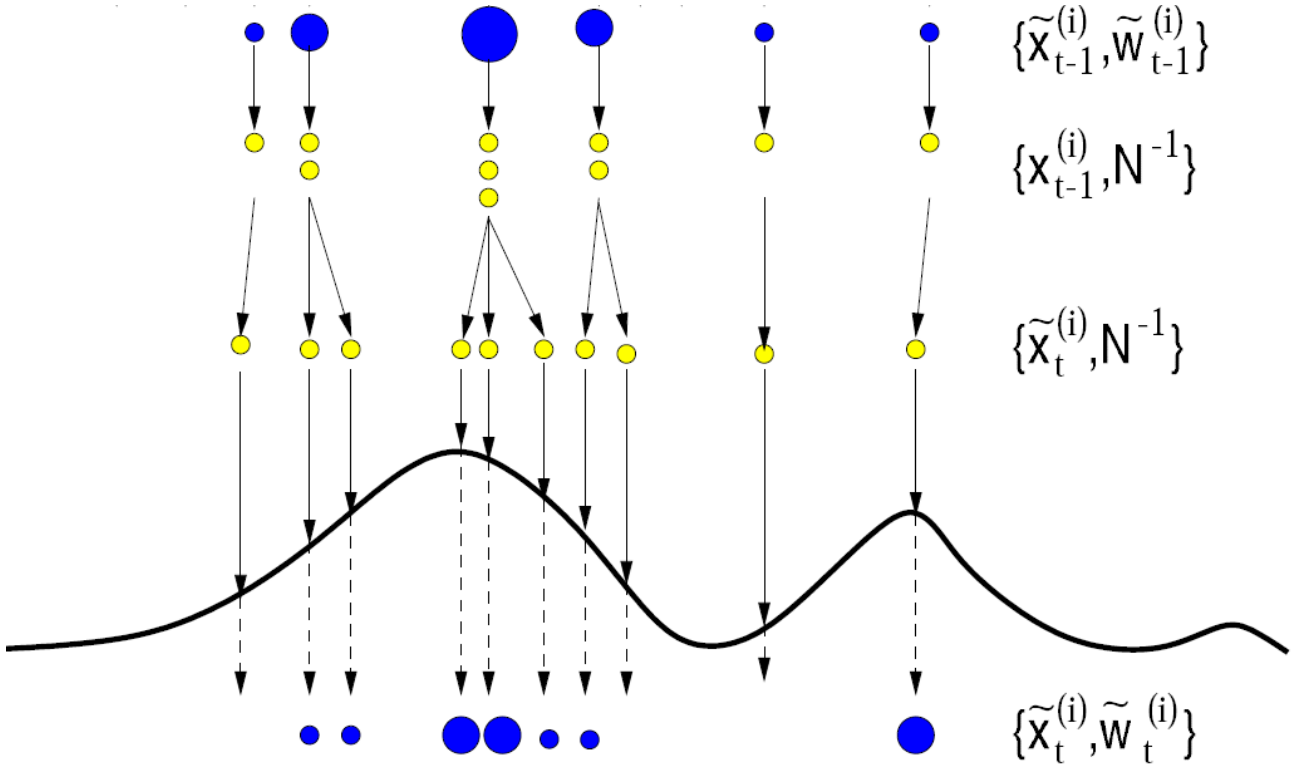
(Bayes Filter: $p_t(x_t) \;\propto\; P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})\; p_{t\text{-}1}(x_{t\text{-}1})\; dx_{t\text{-}1}$)



1. Start with N particles $\{(x^i_{t\text{-}1}, w^i_{t\text{-}1})\}_{i=1}^N$
2. Resample particles to get N weight-1-particles: $\{\hat{x}^i_{t\text{-}1}\}_{i=1}^N$

Particle Filter := Bayesian Filtering with Particles

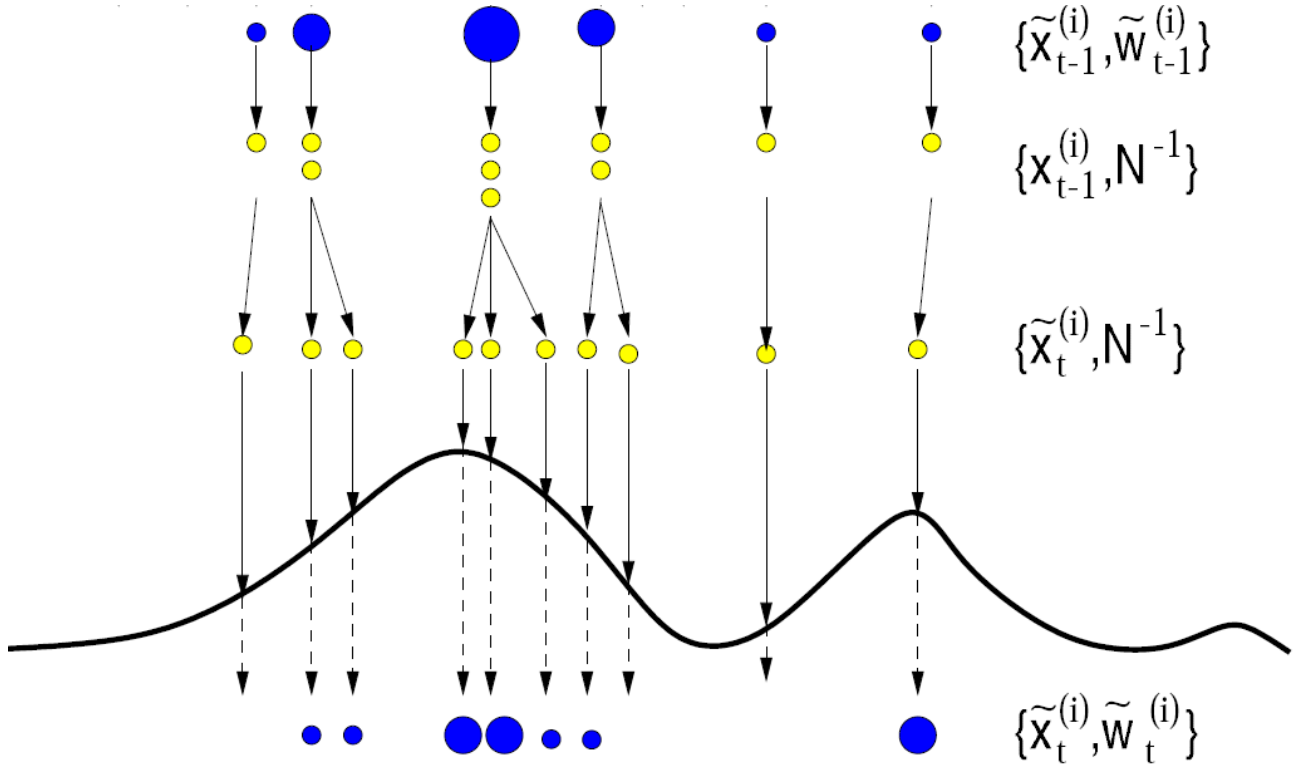
(Bayes Filter: $p_t(x_t) \;\propto\; P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})\; p_{t\text{-}1}(x_{t\text{-}1})\; dx_{t\text{-}1}$)



1. Start with N particles $\{(x^i_{t\text{-}1}, w^i_{t\text{-}1})\}_{i=1}^N$
2. Resample particles to get N weight-1-particles: $\{\hat{x}^i_{t\text{-}1}\}_{i=1}^N$
3. Use motion model to get new "predictive" particles $\{x^i_t\}_{i=1}^N$
   for each $x^i_t \sim P(x_t \,|\, u_{t\text{-}1}, \hat{x}^i_{t\text{-}1})$

Particle Filter := Bayesian Filtering with Particles

(Bayes Filter: $p_t(x_t) \;\propto\; P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})\; p_{t\text{-}1}(x_{t\text{-}1})\; dx_{t\text{-}1}$)



1. Start with N particles $\{(x^i_{t\text{-}1}, w^i_{t\text{-}1})\}_{i=1}^N$
2. Resample particles to get N weight-1-particles: $\{\hat{x}^i_{t\text{-}1}\}_{i=1}^N$
3. Use motion model to get new "predictive" particles $\{x^i_t\}_{i=1}^N$
   for each $x^i_t \sim P(x_t \,|\, u_{t\text{-}1}, \hat{x}^i_{t\text{-}1})$
4. Use observation model to assign new weights $w^i_t \propto P(y_t \,|\, x^i_t)$

Practical Considerations

- Given a complex multi-modal distribution: *where are we?*
  - particle with the heighest weight?
  - which *cluster*?
  - kernel density estimation: cluster with highest density?

Practical Considerations

- Given a complex multi-modal distribution: *where are we?*
  - particle with the heighest weight?
  - which *cluster*?
  - kernel density estimation: cluster with highest density?

- Particle deprivation: $\{x_t^i\}_{i=1}^N$ have very low observation likelihood $P(y_t \mid x_t^i)$: “particle die over time”
  - random samples
  - *sensor reset*: samples from $P(y_t|x_t)$
  - ...

Practical Considerations

- Given a complex multi-modal distribution: *where are we?*
  - particle with the heighest weight?
  - which *cluster*?
  - kernel density estimation: cluster with highest density?

- Particle deprivation: $\{x_t^i\}_{i=1}^N$ have very low observation likelihood $P(y_t \mid x_t^i)$: “particle die over time”
  - random samples
  - *sensor reset*: samples from $P(y_t|x_t)$
  - ...

- How many particles? $\rightarrow$ adaptive techniques

Practical Considerations

- Given a complex multi-modal distribution: *where are we?*
  - particle with the heighest weight?
  - which *cluster*?
  - kernel density estimation: cluster with highest density?

- Particle deprivation: $\{x_t^i\}_{i=1}^N$ have very low observation likelihood $P(y_t \mid x_t^i)$: “particle die over time”
  - random samples
  - *sensor reset*: samples from $P(y_t|x_t)$
  - ...

- How many particles? $\rightarrow$ adaptive techniques

- Resampling is important: many variations exist!
  Liu & Chen (1998): Sequential Monte Carlo Methods for Dynamic Systems.
  Douc, Cappé & Moulines: Comparison of Resampling Schemes for Particle Filtering.

Particle Filter: Summary

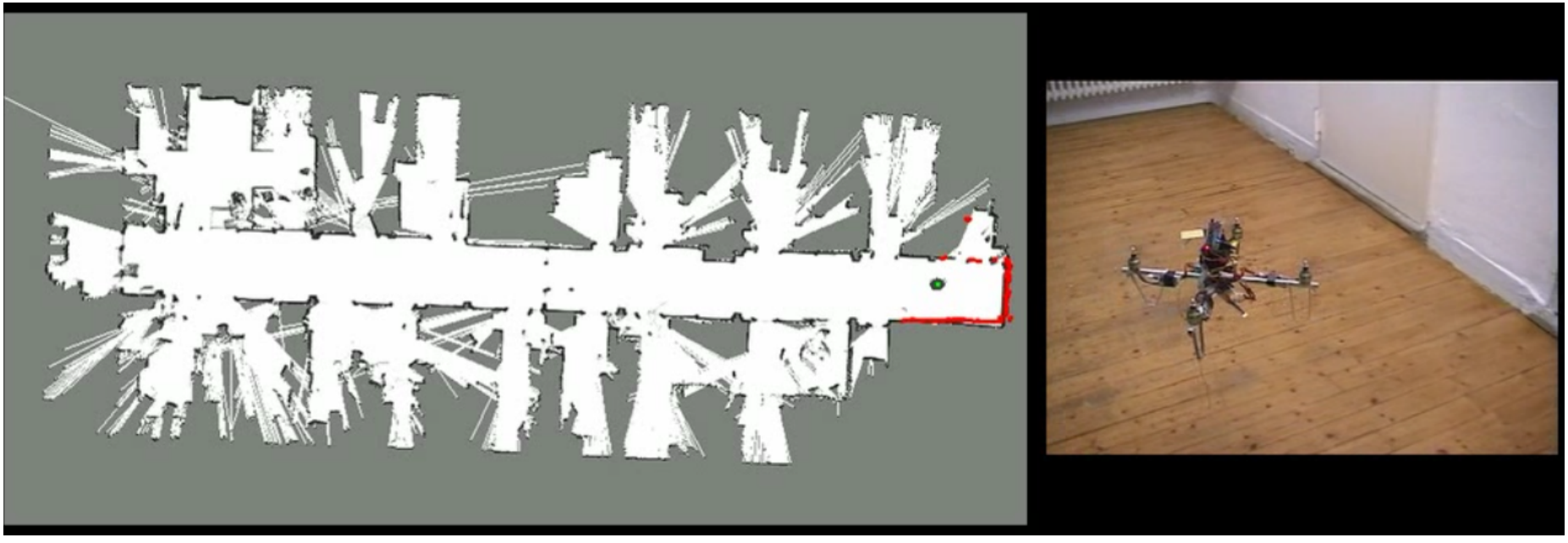
- conceptually simple
- approximates the true distribution $p(x)$
- non-parametric
- can deal with non-linear transformations

- "Particle Filter" in different communities

  - *Monte Carlo Localization* in the mobile robotics community

  - *Condensation Algorithm* in the vision community

Example: Quadcopter Localization



http://www.slawomir.de/publications/grzonka09icra/grzonka09icra.pdf

Quadcopter Indoor Localization

Kalman filter := Bayesian Filtering with Gaussians

Bayes Filter: $p_t(x_t) \;\propto\; P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})\; p_{t\text{-}1}(x_{t\text{-}1})\; dx_{t\text{-}1}$

- Can be computed analytically for linear-Gaussian observations and transitions:
  $P(y_t \,|\, x_t) = \mathcal{N}(y_t \,|\, C x_t + c, W)$
  $P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1}) = \mathcal{N}(x_t \,|\, A(u_{t\text{-}1})\; x_{t\text{-}1} + a(u_{t\text{-}1}), Q)$

  Defition:
  $\mathcal{N}(x \,|\, a, A) = \frac{1}{|2\pi A|^{1/2}}\; \exp\{-\frac{1}{2}(x\text{-}a)^\top\, A^{\text{-}1}\; (x\text{-}a)\}$
  Product:
  $\mathcal{N}(x \,|\, a, A)\; \mathcal{N}(x \,|\, b, B) = \mathcal{N}(x \,|\, B(A{+}B)^{\text{-}1}a + A(A{+}B)^{\text{-}1}b, A(A{+}B)^{\text{-}1}B)\;\; \mathcal{N}(a \,|\, b, A + B)$
  "Propagation":
  $\int_y \mathcal{N}(x \,|\, a + Fy, A)\; \mathcal{N}(y \,|\, b, B)\; dy = \mathcal{N}(x \,|\, a + Fb, A + FBF^\top)$
  Transformation:
  $\mathcal{N}(Fx + f \,|\, a, A) = \frac{1}{|F|}\; \mathcal{N}(x \,|\; F^{\text{-}1}(a - f),\; F^{\text{-}1}AF^{\text{-}\top})$

  (more identities: see "Gaussian identities"
  `http://ipvs.informatik.uni-stuttgart.de/mlr/marc/notes/gaussians.pdf`)

Kalman filter derivation

$$\begin{aligned}
\text{state:} \quad & p_t(x_t) = \mathcal{N}(x_t \mid s_t, S_t) \\
\text{transition model:} \quad & P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1}) = \mathcal{N}(x_t \mid Ax_{t\text{-}1} + a, Q) \\
\text{observation model:} \quad & P(y_t \mid x_t) = \mathcal{N}(y_t \mid Cx_t + c, W)
\end{aligned}$$

$$p_t(x_t) \propto P(y_t \mid x_t) \int_{x_{t\text{-}1}} P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1})\, p_{t\text{-}1}(x_{t\text{-}1})\, dx_{t\text{-}1}$$

Kalman filter derivation

$$\begin{aligned}
\text{state:} \quad & p_t(x_t) = \mathcal{N}(x_t \mid s_t, S_t) \\
\text{transition model:} \quad & P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1}) = \mathcal{N}(x_t \mid A x_{t\text{-}1} + a, Q) \\
\text{observation model:} \quad & P(y_t \mid x_t) = \mathcal{N}(y_t \mid C x_t + c, W)
\end{aligned}$$

$$\begin{aligned}
p_t(x_t) &\propto P(y_t \mid x_t) \int_{x_{t\text{-}1}} P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1})\, p_{t\text{-}1}(x_{t\text{-}1})\, dx_{t\text{-}1} \\
&= \mathcal{N}(y_t \mid C x_t + c, W) \int_{x_{t\text{-}1}} \mathcal{N}(x_t \mid A x_{t\text{-}1} + a, Q)\, \mathcal{N}(x_{t\text{-}1} \mid s_{t\text{-}1}, S_{t\text{-}1})\, dx_{t\text{-}1}
\end{aligned}$$

insert definitions of state, observation and transition model

Kalman filter derivation

$$\begin{aligned}
\text{state:} \quad & p_t(x_t) = \mathcal{N}(x_t \,|\, s_t, S_t) \\
\text{transition model:} \quad & P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1}) = \mathcal{N}(x_t \,|\, Ax_{t\text{-}1} + a, Q) \\
\text{observation model:} \quad & P(y_t \,|\, x_t) = \mathcal{N}(y_t \,|\, Cx_t + c, W)
\end{aligned}$$

$$\begin{aligned}
p_t(x_t) &\propto P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})\, p_{t\text{-}1}(x_{t\text{-}1})\, dx_{t\text{-}1} \\
&= \mathcal{N}(y_t \,|\, Cx_t + c, W) \int_{x_{t\text{-}1}} \mathcal{N}(x_t \,|\, Ax_{t\text{-}1} + a, Q)\, \mathcal{N}(x_{t\text{-}1} \,|\, s_{t\text{-}1}, S_{t\text{-}1})\, dx_{t\text{-}1} \\
&= \mathcal{N}(y_t \,|\, Cx_t + c, W)\, \mathcal{N}(x_t \,|\, \underbrace{As_{t\text{-}1} + a}_{=:\hat{s}_t},\ \underbrace{Q + AS_{t\text{-}1}A^\top}_{=:\hat{S}_t})
\end{aligned}$$

"Propagation": $\int_y \mathcal{N}(x \,|\, a + Fy, A)\, \mathcal{N}(y \,|\, b, B)\, dy = \mathcal{N}(x \,|\, a + Fb, A + FBF^\top)$

Kalman filter derivation

$$\begin{aligned}
\text{state:} \quad & p_t(x_t) = \mathcal{N}(x_t \,|\, s_t, S_t) \\
\text{transition model:} \quad & P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1}) = \mathcal{N}(x_t \,|\, Ax_{t\text{-}1} + a, Q) \\
\text{observation model:} \quad & P(y_t \,|\, x_t) = \mathcal{N}(y_t \,|\, Cx_t + c, W)
\end{aligned}$$

$$\begin{aligned}
p_t(x_t) &\propto P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})\, p_{t\text{-}1}(x_{t\text{-}1})\, dx_{t\text{-}1} \\
&= \mathcal{N}(y_t \,|\, Cx_t + c, W) \int_{x_{t\text{-}1}} \mathcal{N}(x_t \,|\, Ax_{t\text{-}1} + a, Q)\, \mathcal{N}(x_{t\text{-}1} \,|\, s_{t\text{-}1}, S_{t\text{-}1})\, dx_{t\text{-}1} \\
&= \mathcal{N}(y_t \,|\, Cx_t + c, W)\, \mathcal{N}(x_t \,|\, \underbrace{As_{t\text{-}1} + a}_{=:\hat{s}_t}, \underbrace{Q + AS_{t\text{-}1}A^\top}_{=:\hat{S}_t}) \\
&= \mathcal{N}(Cx_t + c \,|\, y_t, W)\, \mathcal{N}(x_t \,|\, \hat{s}_t, \hat{S}_t)
\end{aligned}$$

Use symmetry of Gaussian: $\mathcal{N}(x \,|\, a, A) = \mathcal{N}(a \,|\, x, A)$

Kalman filter derivation

$$\begin{aligned}
\text{state:} \quad & p_t(x_t) = \mathcal{N}(x_t \,|\, s_t, S_t) \\
\text{transition model:} \quad & P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1}) = \mathcal{N}(x_t \,|\, A x_{t\text{-}1} + a, Q) \\
\text{observation model:} \quad & P(y_t \,|\, x_t) = \mathcal{N}(y_t \,|\, C x_t + c, W)
\end{aligned}$$

$$\begin{aligned}
p_t(x_t) &\propto P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})\, p_{t\text{-}1}(x_{t\text{-}1})\, dx_{t\text{-}1} \\
&= \mathcal{N}(y_t \,|\, C x_t + c, W) \int_{x_{t\text{-}1}} \mathcal{N}(x_t \,|\, A x_{t\text{-}1} + a, Q)\, \mathcal{N}(x_{t\text{-}1} \,|\, s_{t\text{-}1}, S_{t\text{-}1})\, dx_{t\text{-}1} \\
&= \mathcal{N}(y_t \,|\, C x_t + c, W)\, \mathcal{N}(x_t \,|\, \underbrace{A s_{t\text{-}1} + a}_{=:\hat{s}_t},\ \underbrace{Q + A S_{t\text{-}1} A^\top}_{=:\hat{S}_t}) \\
&= \mathcal{N}(C x_t + c \,|\, y_t, W)\, \mathcal{N}(x_t \,|\, \hat{s}_t, \hat{S}_t) \\
&= \mathcal{N}(x_t \,|\, s_t, S_t) \cdot \langle \text{terms indep. of } x_t \rangle
\end{aligned}$$

Kalman filter derivation

$$\begin{aligned}
\text{state:} \quad & p_t(x_t) = \mathcal{N}(x_t \,|\, s_t, S_t) \\
\text{transition model:} \quad & P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1}) = \mathcal{N}(x_t \,|\, A x_{t\text{-}1} + a, Q) \\
\text{observation model:} \quad & P(y_t \,|\, x_t) = \mathcal{N}(y_t \,|\, C x_t + c, W)
\end{aligned}$$

$$\begin{aligned}
p_t(x_t) &\propto P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})\, p_{t\text{-}1}(x_{t\text{-}1})\, dx_{t\text{-}1} \\
&= \mathcal{N}(y_t \,|\, C x_t + c, W) \int_{x_{t\text{-}1}} \mathcal{N}(x_t \,|\, A x_{t\text{-}1} + a, Q)\, \mathcal{N}(x_{t\text{-}1} \,|\, s_{t\text{-}1}, S_{t\text{-}1})\, dx_{t\text{-}1} \\
&= \mathcal{N}(y_t \,|\, C x_t + c, W)\, \mathcal{N}(x_t \,|\, \underbrace{A s_{t\text{-}1} + a}_{=:\hat{s}_t},\ \underbrace{Q + A S_{t\text{-}1} A^\top}_{=:\hat{S}_t}) \\
&= \mathcal{N}(C x_t + c \,|\, y_t, W)\, \mathcal{N}(x_t \,|\, \hat{s}_t, \hat{S}_t) \\
&= \mathcal{N}(x_t \,|\, s_t, S_t) \cdot \langle \text{terms indep. of } x_t \rangle \\
S_t &= (C^\top W^{\text{-}1} C + \hat{S}_t^{\text{-}1})^{\text{-}1} = \hat{S}_t - \underbrace{\hat{S}_t C^\top (W + C \hat{S}_t C^\top)^{\text{-}1}}_{\text{“Kalman gain” } K} C \hat{S}_t
\end{aligned}$$

new covariance S_t; uses the general Woodbury identity

Kalman filter derivation

$$\begin{aligned}
\text{state:} \quad & p_t(x_t) = \mathcal{N}(x_t \,|\, s_t, S_t) \\
\text{transition model:} \quad & P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1}) = \mathcal{N}(x_t \,|\, Ax_{t\text{-}1} + a, Q) \\
\text{observation model:} \quad & P(y_t \,|\, x_t) = \mathcal{N}(y_t \,|\, Cx_t + c, W)
\end{aligned}$$

$$\begin{aligned}
p_t(x_t) &\propto P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})\, p_{t\text{-}1}(x_{t\text{-}1})\, dx_{t\text{-}1} \\
&= \mathcal{N}(y_t \,|\, Cx_t + c, W) \int_{x_{t\text{-}1}} \mathcal{N}(x_t \,|\, Ax_{t\text{-}1} + a, Q)\, \mathcal{N}(x_{t\text{-}1} \,|\, s_{t\text{-}1}, S_{t\text{-}1})\, dx_{t\text{-}1} \\
&= \mathcal{N}(y_t \,|\, Cx_t + c, W)\, \mathcal{N}(x_t \,|\, \underbrace{As_{t\text{-}1} + a}_{=:\hat{s}_t},\ \underbrace{Q + AS_{t\text{-}1}A^\top}_{=:\hat{S}_t}) \\
&= \mathcal{N}(Cx_t + c \,|\, y_t, W)\, \mathcal{N}(x_t \,|\, \hat{s}_t, \hat{S}_t) \\
&= \mathcal{N}(x_t \,|\, s_t, S_t) \cdot \langle \text{terms indep. of } x_t \rangle \\
S_t &= (C^\top W^{\text{-}1} C + \hat{S}_t^{\text{-}1})^{\text{-}1} = \hat{S}_t - \underbrace{\hat{S}_t C^\top (W + C\hat{S}_t C^\top)^{\text{-}1}}_{\text{“Kalman gain” } K} C\hat{S}_t \\
s_t &= S_t[C^\top W^{\text{-}1}(y_t - c) + \hat{S}_t^{\text{-}1}\hat{s}_t] = \hat{s}_t + K(y_t - C\hat{s}_t - c)
\end{aligned}$$

new mean s_t; uses $S_t C^\top W^{\text{-}1} = K$ and $S_t \hat{S}_t^{\text{-}1} = \mathbf{I} - KC$

Extended Kalman filter (EKF) and Unscented Transform

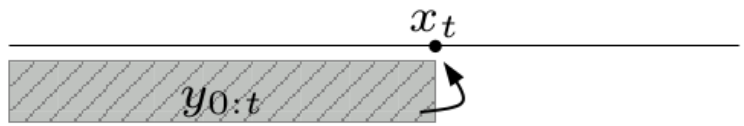
Bayes Filter: $p_t(x_t) \;\propto\; P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})\; p_{t\text{-}1}(x_{t\text{-}1})\; dx_{t\text{-}1}$

- Can be computed analytically for linear-Gaussian observations and transitions:
  $P(y_t \,|\, x_t) = \mathcal{N}(y_t \,|\, Cx_t + c, W)$
  $P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1}) = \mathcal{N}(x_t \,|\, A(u_{t\text{-}1})x_{t\text{-}1} + a(u_{t\text{-}1}), Q)$

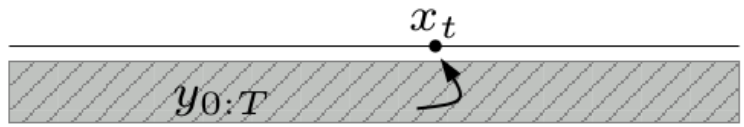
- If $P(y_t \,|\, x_t)$ or $P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})$ are not linear:
  $P(y_t \,|\, x_t) = \mathcal{N}(y_t \,|\, g(x_t), W)$
  $P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1}) = \mathcal{N}(x_t \,|\, f(x_{t\text{-}1}, u_{t\text{-}1}), Q)$
  – approximate f and g as locally linear (*Extended Kalman Filter*)
  – or sample locally from them and reapproximate as Gaussian (*Unscented Transform*)

Bayes smoothing

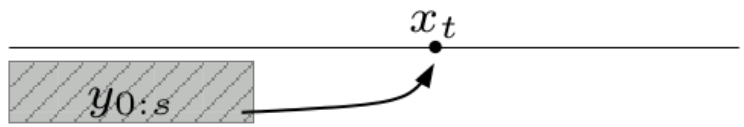
Filtering: $P(x_t|y_{0:t})$



Smoothing: $P(x_t|y_{0:T})$



Prediction: $P(x_t|y_{0:s})$



Bayes smoothing

- Let $\mathcal{P} = y_{0:t}$ past observations, $\mathcal{F} = y_{t+1:T}$ future observations

$$
\begin{aligned}
P(x_t \,|\, \mathcal{P}, \mathcal{F}, u_{0:T}) &\propto P(\mathcal{F} \,|\, x_t, \mathcal{P}, u_{0:T})\; P(x_t \,|\, \mathcal{P}, u_{0:T}) \\
&= \underbrace{P(\mathcal{F} \,|\, x_t, u_{t:T})}_{=:\beta_t(x_t)}\; \underbrace{P(x_t \,|\, \mathcal{P}, u_{0:t\text{-}1})}_{=:p(x_t)}
\end{aligned}
$$

Bayesian smoothing fuses a forward filter $p_t(x_t)$ with a backward "filter" $\beta_t(x_t)$

Bayes smoothing

- Let $\mathcal{P} = y_{0:t}$ past observations, $\mathcal{F} = y_{t+1:T}$ future observations

$$\begin{aligned} P(x_t \,|\, \mathcal{P}, \mathcal{F}, u_{0:T}) &\propto P(\mathcal{F} \,|\, x_t, \mathcal{P}, u_{0:T})\; P(x_t \,|\, \mathcal{P}, u_{0:T}) \\ &= \underbrace{P(\mathcal{F} \,|\, x_t, u_{t:T})}_{=:\beta_t(x_t)}\; \underbrace{P(x_t \,|\, \mathcal{P}, u_{0:t\text{-}1})}_{=:p(x_t)} \end{aligned}$$

Bayesian smoothing fuses a forward filter $p_t(x_t)$ with a backward "filter" $\beta_t(x_t)$

- Backward recursion (derivation analogous to the Bayesian filter)

$$\begin{aligned} \beta_t(x_t) &:= P(y_{t+1:T} \,|\, x_t, u_{t:T}) \\ &= \int_{x_{t+1}} \beta_{t+1}(x_{t+1})\; P(y_{t+1} \,|\, x_{t+1})\; P(x_{t+1} \,|\, x_t, u_t)\; dx_{t+1} \end{aligned}$$

Simultaneous Localization and Mapping (SLAM)

Localization and Mapping

- The Bayesian filter requires an observation model $P(y_t \mid x_t)$

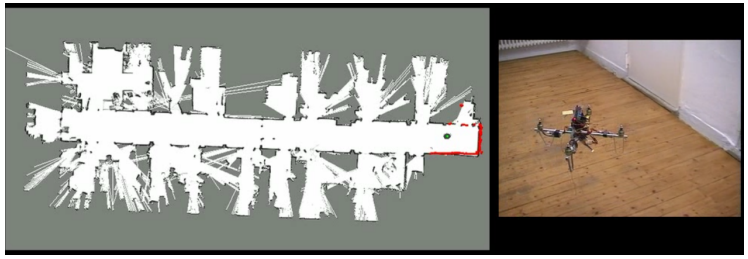
- A **map** is something that provides the observation model:
  A map tells us for each x_t what the sensor readings y_t might look like

Types of maps

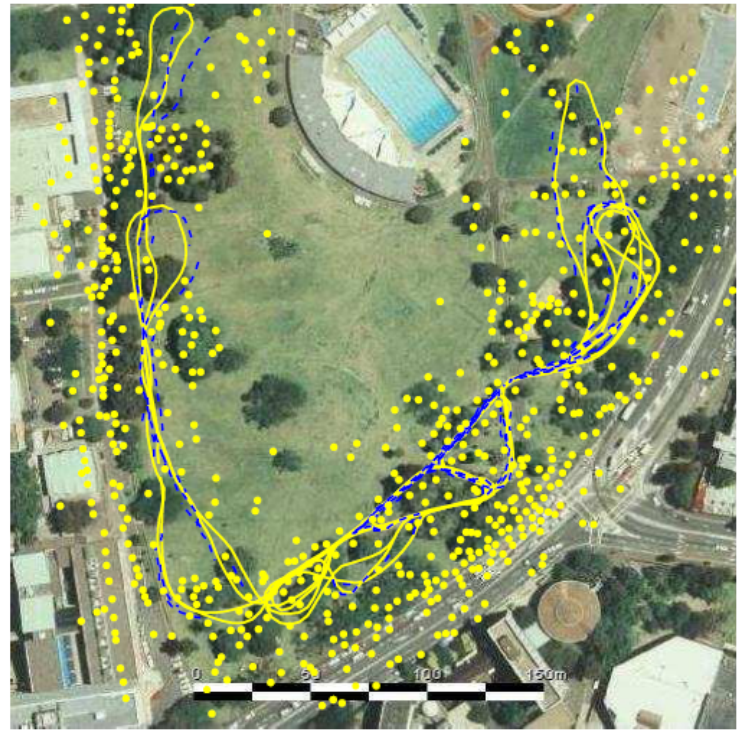
Grid map

K. Murphy (1999): *Bayesian map learning in dynamic environments.*
Grisetti, Tipaldi, Stachniss, Burgard, Nardi: *Fast and Accurate SLAM with Rao-Blackwellized Particle Filters*

Laser scan map



Landmark map



Victoria Park data set

M. Montemerlo, S. Thrun, D. Koller, & B. Wegbreit (2003): *FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges.* IJCAI, 1151–1156.

Simultaneous Localization and Mapping Problem

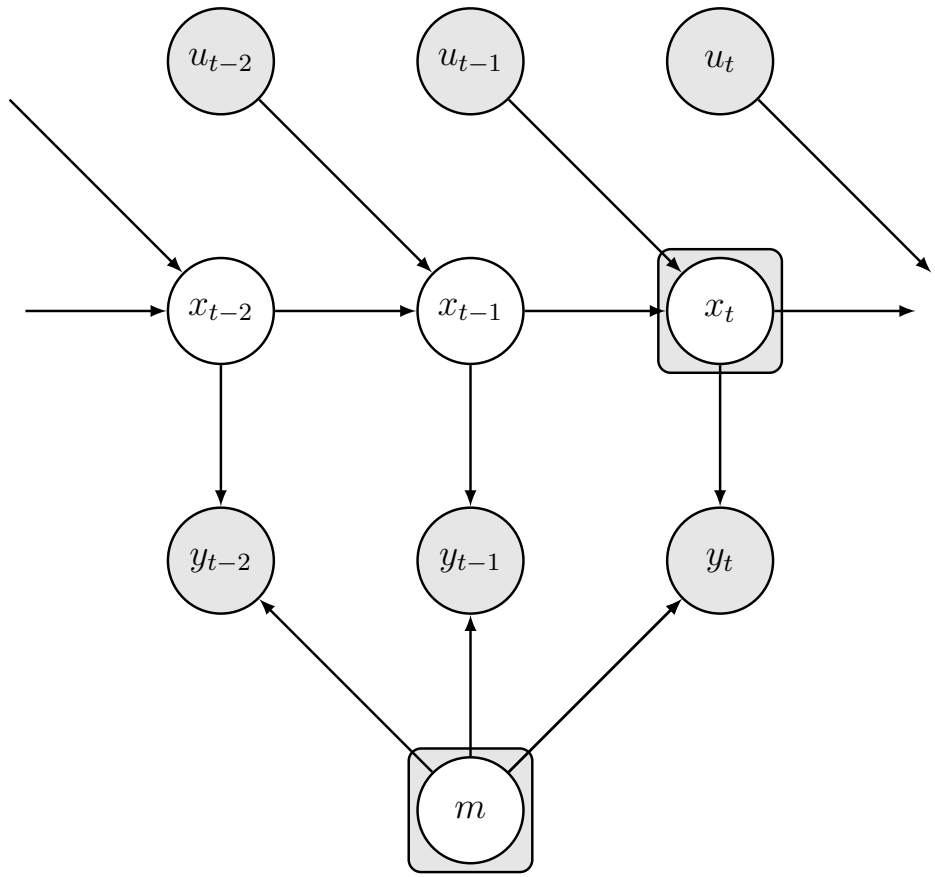
- Notation:

  x_t = state (location) at time t

  y_t = sensor readings at time t

  $u_{t\text{-}1}$ = control command (action, steering, velocity) at time t-1

  m = the map; *formally: a map is the parameters that define* $P(y_t \mid x_t, m)$



Simultaneous Localization and Mapping Problem

- Notation:
  x_t = state (location) at time t
  y_t = sensor readings at time t
  $u_{t\text{-}1}$ = control command (action, steering, velocity) at time t-1
  m = the map; *formally: a map is the parameters that define* $P(y_t \mid x_t, m)$

- Given the history $y_{0:t}$ and $u_{0:t\text{-}1}$, we want to compute the belief over the pose and the map m at time t

$$p_t(x_t, m) := P(x_t, m \mid y_{0:t}, u_{0:t\text{-}1})$$

- We assume to know the:
  – transition model $P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1})$
  – observation model $P(y_t \mid x_t, m)$ *(defined by the map)*

SLAM: classical “chicken or egg problem”

- If we knew the map we could use a Bayes filter to compute the belief over the state

$$P(x_t \mid m, y_{0:t}, u_{0:t\text{-}1})$$

- If we knew the state trajectory $x_{0:t}$ we could efficiently compute the belief over the map

$$P(m \mid x_{0:t}, y_{0:t}, u_{0:t\text{-}1})$$

SLAM: classical “chicken or egg problem”

- If we knew the map we could use a Bayes filter to compute the belief over the state

$$P(x_t \mid m, y_{0:t}, u_{0:t\text{-}1})$$

- If we knew the state trajectory $x_{0:t}$ we could efficiently compute the belief over the map

$$P(m \mid x_{0:t}, y_{0:t}, u_{0:t\text{-}1})$$

- SLAM requires to tie state estimation and map building together:
  1) Joint inference on x_t and m
     $\rightarrow$ Kalman-SLAM
  2) Tie a state hypothesis (=particle) to a map hypothesis
     $\rightarrow$ particle SLAM
  3) Frame everything as a graph optimization problem
     $\rightarrow$ graph SLAM

Joint Bayesian Filter over x and m

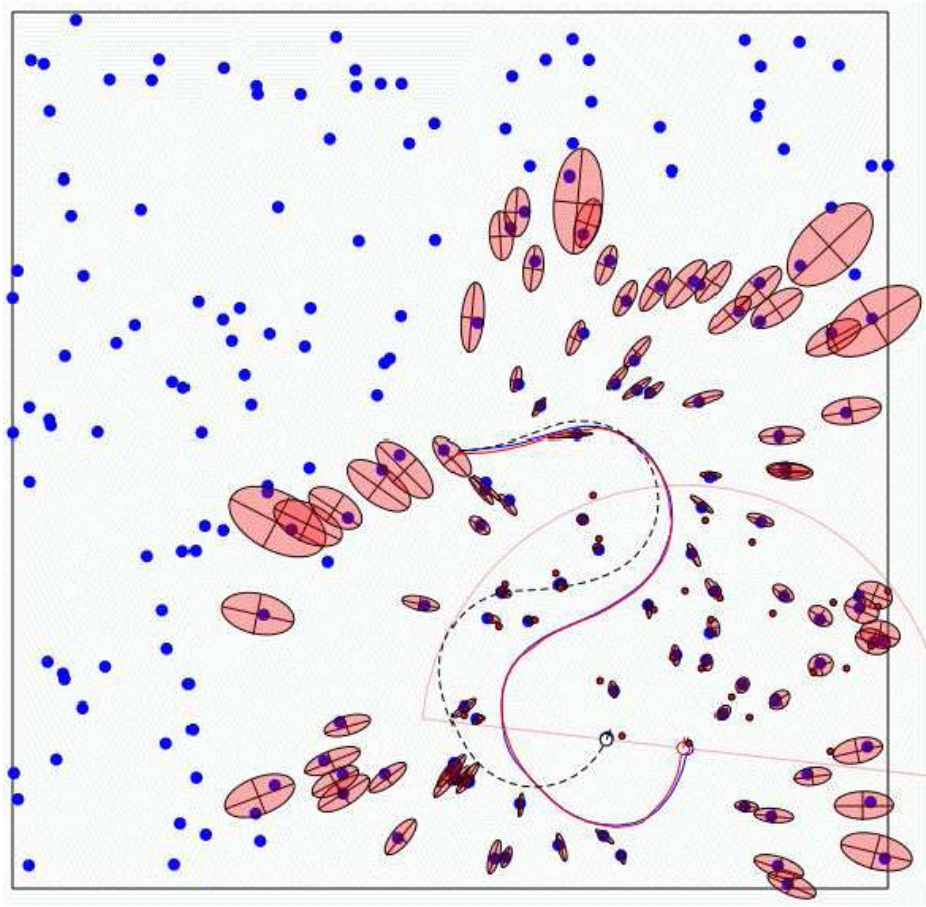
- A (formally) straight-forward approach is the joint Bayesian filter

$$p_t(x_t, m) \;\propto\; P(y_t \mid x_t, m) \int_{x_{t\text{-}1}} P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1})\; p_{t\text{-}1}(x_{t\text{-}1}, m)\; dx_{t\text{-}1}$$

But: How to represent a belief over high-dimensional x_t, m?

Map uncertainty

- In this case the map $m = (\theta_1, .., \theta_N)$ is a set of N landmarks, $\theta_j \in \mathbb{R}^2$



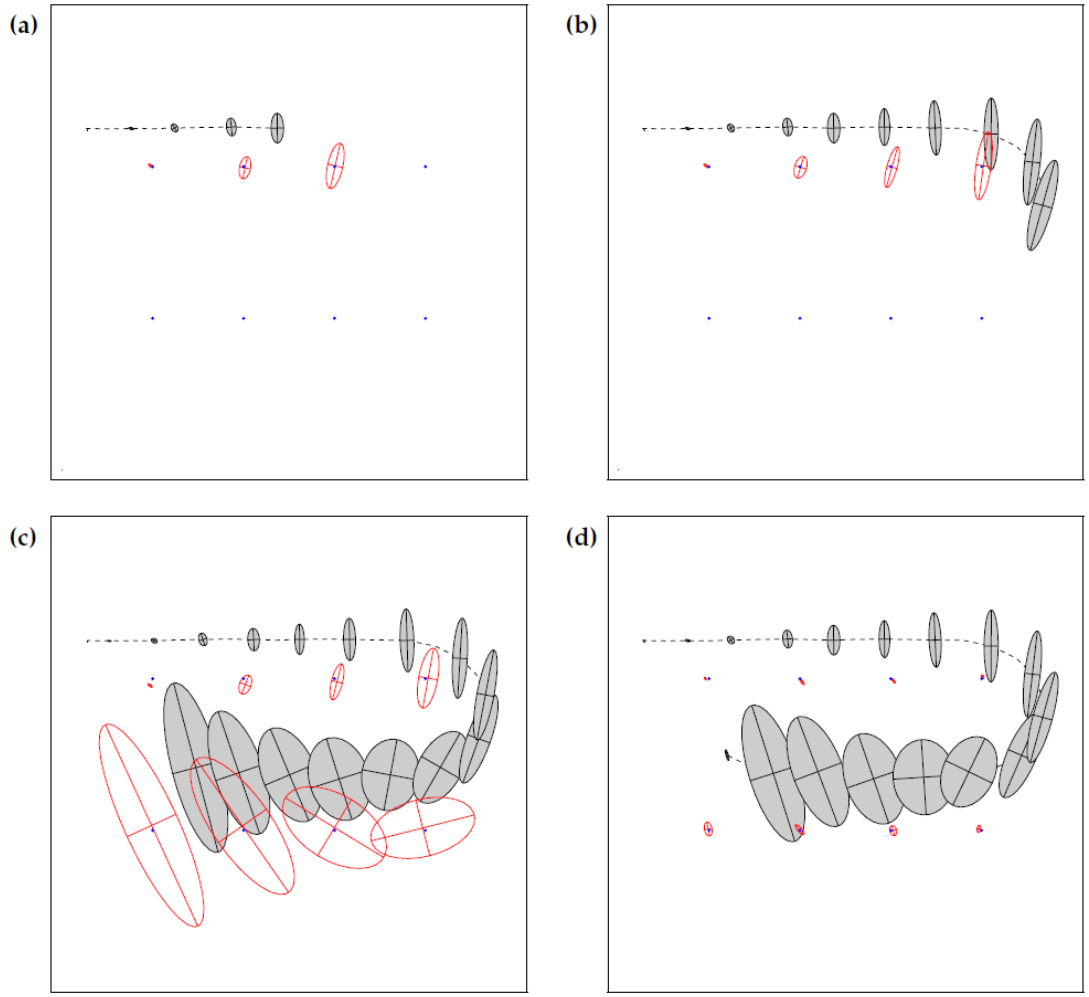
- Use Gaussians to represent the uncertainty of landmark positions

(Extended) Kalman Filter SLAM

- Analogous to Localization with Gaussian for the pose belief $p_t(x_t)$
  - But now: joint belief $p_t(x_t, \theta_{1:N})$ is $3 + 2N$-dimensional Gaussian
  - Assumes the map $m = (\theta_1, .., \theta_N)$ is a set of N landmarks, $\theta_j \in \mathbb{R}^2$
  - Exact update equations (under the Gaussian assumption)
  - Conceptually very simple

- Drawbacks:
  - Scaling (full covariance matrix is $O(N^2)$)
  - Sometimes non-robust (uni-modal, "data association problem")
  - Lacks advantages of Particle Filter
    (multiple hypothesis, more robust to non-linearities)

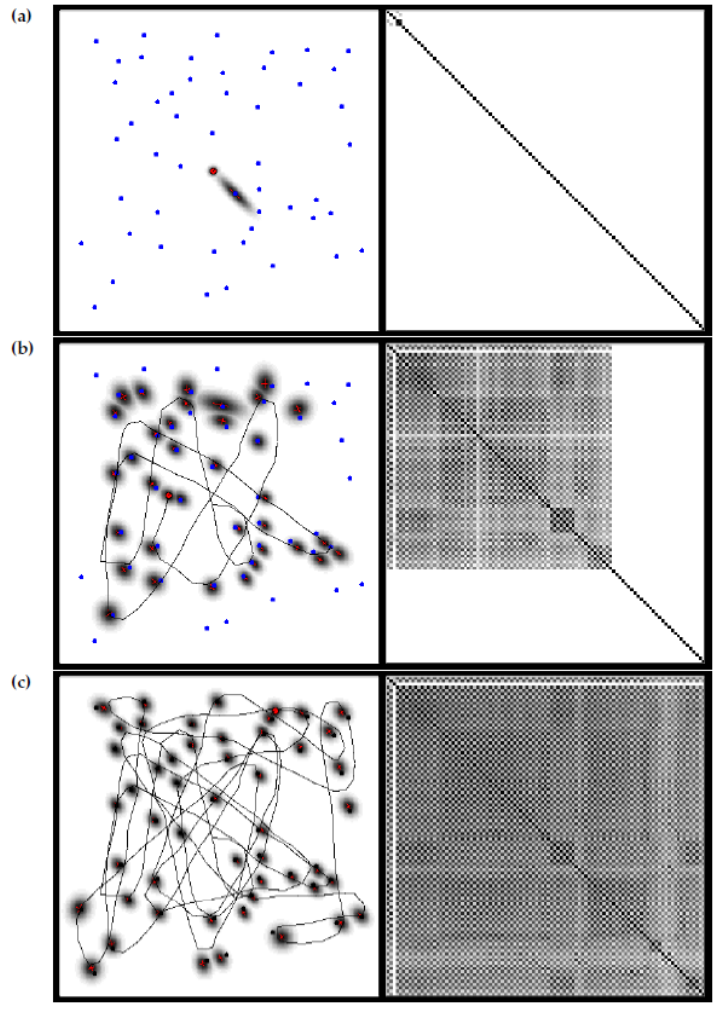
EKF Slam Example 1



(from *Probabilistic Robotics* Sebastian Thrun, Wolfram Burgard, Dieter Fox)

- The landmark uncertainty decreases when the robot sees the first landmark again.

EKF Slam Example 2



(from *Probabilistic Robotics* Sebastian Thrun, Wolfram Burgard, Dieter Fox)

- The right images show the correlation matrix. After some time all landmarks become fully correlated.

SLAM with particles

Core idea: Each particle carries its own map belief

SLAM with particles

Core idea: Each particle carries its own map belief

- As for the localization problem use particles to represent the *pose belief* $p_t(x_t)$

  (Note: Each particle actually "has a history $x^i_{0:t}$" – a whole trajectory!)

- Factorization of the SLAM posterior:

  $$P(x_{0:t}, m | y_{0:t}, u_{0:t-1}) = P(x_{0:t} | y_{0:t}, u_{0:t-1}) \underbrace{P(m | x_{0:t}, y_{0:t})}_{p_t(m)}$$

SLAM with particles

Core idea: Each particle carries its own map belief

- As for the localization problem use particles to represent the *pose belief* $p_t(x_t)$

  (Note: Each particle actually "has a history $x^i_{0:t}$" – a whole trajectory!)

- Factorization of the SLAM posterior:

  $$P(x_{0:t}, m | y_{0:t}, u_{0:t-1}) = P(x_{0:t} | y_{0:t}, u_{0:t-1}) \underbrace{P(m | x_{0:t}, y_{0:t})}_{p_t(m)}$$

- Each particle i stores its path estimate $x^i_{0:t}$ and an estimate of the map belief $p^i_t(m)$ conditioned on the particle history $x^i_{0:t}$. The conditional beliefs $p^i_t(m)$ may be factorized over grid points or landmarks of the map.

K. Murphy (1999): *Bayesian map learning in dynamic environments.*
`http://www.cs.ubc.ca/~murphyk/Papers/map_nips99.pdf`

Map estimation for a *given* particle history

- Given $x_{0:t}$ (e.g. a trajectory of a particle), what is the posterior over the map m?

  $\rightarrow$ simplified Bayes Filter:

$$p_t(m) := P(m \mid x_{0:t}, y_{0:t}) \propto P(y_t \mid m, x_t)\, p_{t\text{-}1}(m)$$

  (no transtion model: assumption that map is constant)

Map estimation for a *given* particle history

- Given $x_{0:t}$ (e.g. a trajectory of a particle), what is the posterior over the map m?

  $\rightarrow$ simplified Bayes Filter:

  $$p_t(m) := P(m \mid x_{0:t}, y_{0:t}) \propto P(y_t \mid m, x_t)\, p_{t\text{-}1}(m)$$

  (no transtion model: assumption that map is constant)

- In the case of landmarks (FastSLAM):
  $m = (\theta_1, .., \theta_N)$
  θ_j = position of the j-th landmark, $j \in \{1, .., N\}$
  n_t = which landmark we observe at time t, $\quad n_t \in \{1, .., N\}$

  We can use a separate (Gaussian) Bayes Filter for each θ_j
  (conditioned on $x_{0:t}$, each θ_j is independent from each θ_k)

  $$P(\theta_{1:N} \mid x_{0:t}, y_{0:n}, n_{0:t}) = \prod_j P(\theta_j \mid x_{0:t}, y_{0:n}, n_{0:t})$$

Particle likelihood in SLAM

- Particle likelihood for Localization Problem:
$$w_t^i = P(y_t \mid x_t^i)$$
(determines the new importance weight w_t^i)

- In SLAM the map is uncertain $\rightarrow$ each particle is weighted with the *expected* likelihood:
$$w_t^i = \int P(y_t \mid x_t^i, m)\ p_{t-1}(m)\ dm$$

- In case of landmarks (FastSLAM):
$$w_t^i = \int P(y_t \mid x_t^i, \theta_{n_t}, n_t)\ p_{t-1}(\theta_{n_t})\ d\theta_{n_t}$$

- Data association problem (actually we don't know n_t):
For each particle separately choose $n_t^i = \mathrm{argmax}_{n_t}\ w_t^i(n_t)$

Particle-based SLAM summary

- We have a set of N particles $\{(x^i, w^i)\}_{i=1}^N$ to represent the pose belief $p_t(x_t)$

- For each particle we have a separate map belief $p_t^i(m)$; in the case of landmarks, this factorizes in N separate 2D-Gaussians

- Iterate
  1. Resample particles to get N weight-1-particles: $\{\hat{x}_{t\text{-}1}^i\}_{i=1}^N$
  2. Use motion model to get new "predictive" particles $\{x_t^i\}_{i=1}^N$
  3. Update the map belief $p_m^i(m) \propto P(y_t \mid m, x_t)\, p_{t\text{-}1}^i(m)$ for each particle
  4. Compute new importance weights $w_t^i \propto \int P(y_t \mid x_t^i, m)\, p_{t-1}(m)\, dm$ using the observation model and the map belief

Demo: Visual SLAM

- Map building from a freely moving camera

Demo: Visual SLAM

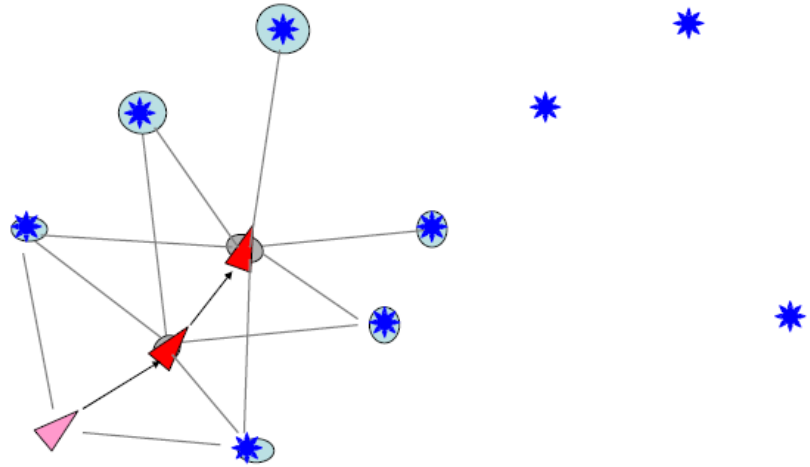
- Map building from a freely moving camera

  – SLAM has become a big topic in the vision community..
  – features are typically landmarks $\theta_{1:N}$ with SURF/SIFT features
  – PTAM (Parallel Tracking and Mapping) parallelizes computations...

  PTAM DTAM LSD-SLAM

  Klein & Murray: *Parallel Tracking and Mapping for Small AR Workspaces* `http://www.robots.ox.ac.uk/~gk/PTAM/`
  Newcombe, Lovegrove & Davison: *DTAM: Dense Tracking and Mapping in Real-Time* ICCV 2011.
  Engel, Schöps & Cremers: LSD-SLAM: Large-Scale Direct Monocular SLAM, ECCV 2014.
  `http://vision.in.tum.de/_media/spezial/bib/engel14eccv.pdf`

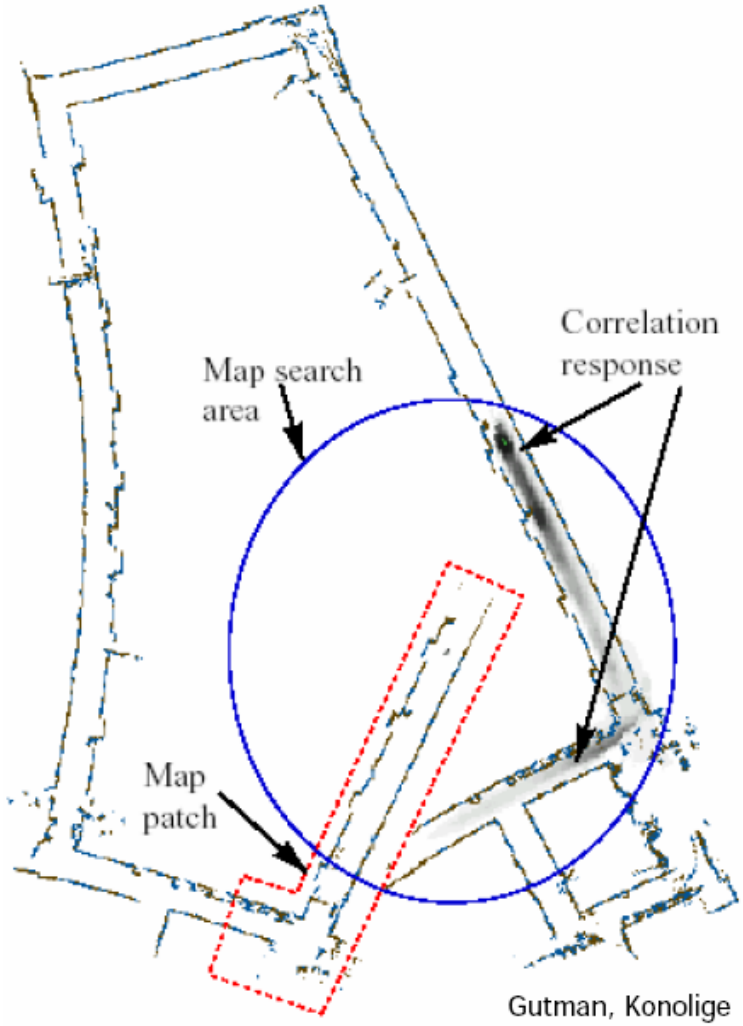
Alternative SLAM approach: Graph-based



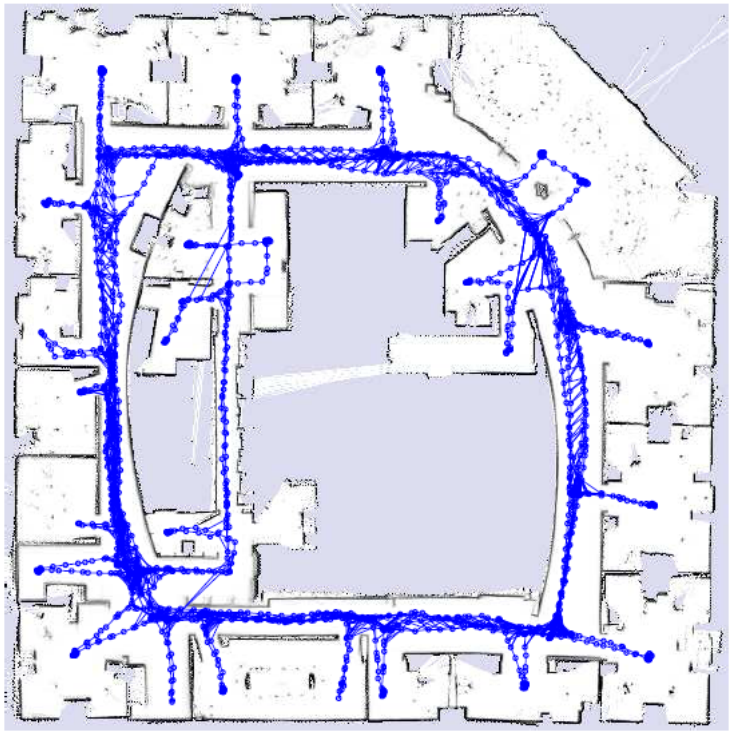
- Represent the previous trajectory as a graph
  – nodes = estimated positions & observations
  – edges = transition & step estimation based on scan matching
- Loop Closing: check if some nodes might coincide $\rightarrow$ new edges
- Classical Optimization:
  The whole graph defines an optimization problem: Find poses that minimize sum of edge & node errors

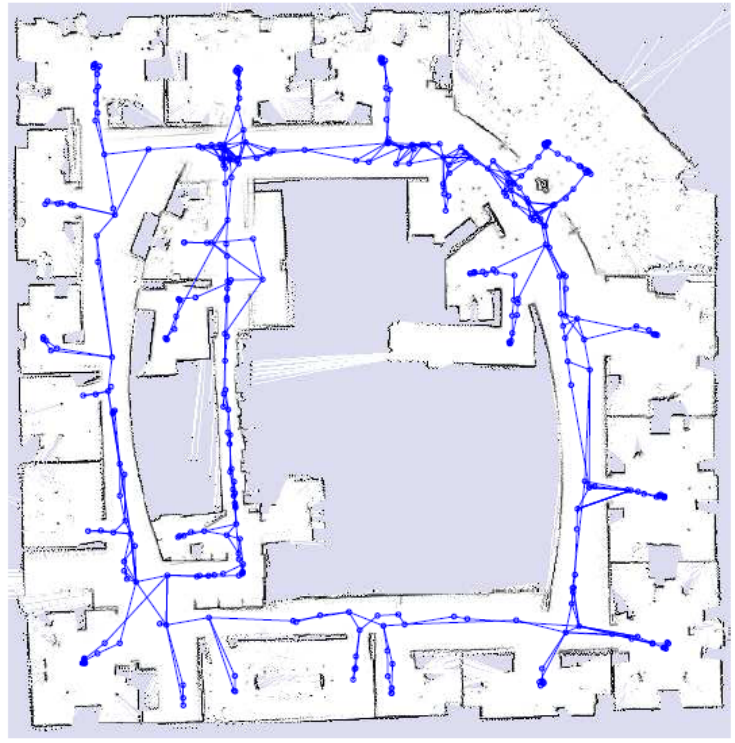
Loop Closing Problem

(Doesn't explicitly exist in Particle Filter methods: If particles cover the belief, then "data association" solves the "loop closing problem")



Graph-based SLAM





Life-long Map Learning for Graph-based SLAM Approaches in Static Environments

Kretzschmar, Grisetti, Stachniss

Additional material

- *Robot Mapping Course*, Cyrill Stachniss, Freiburg, (slides + videos):
  `http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/`
- *Sebastian Thrun, Wolfram Burghard, Dieter Fox:* Probabilistic Robotics
  `probabilistic-robotics.org`