Robot Mapping

FastSLAM – Feature-Based SLAM with Particle Filters

Cyrill Stachniss

Particle Filter

- Non-parametric recursive Bayes filter
- Posterior is represented by a set of weighted samples
- Can model arbitrary distributions
- Works well in low-dimensional spaces
- 3-Step procedure
  - Sampling from proposal
  - Importance Weighting
  - Resampling

Particle Filter Algorithm

1. Sample the particles from the proposal distribution

$$x_t^{[j]} \sim \pi(x_t \mid \ldots)$$

2. Compute the importance weights

$$w_t^{[j]} = \frac{target(x_t^{[j]})}{proposal(x_t^{[j]})}$$

3. Resampling: Draw sample i with probability $w_t^{[i]}$ and repeat J times

Particle Representation

- A set of weighted samples

$$\mathcal{X} = \left\{ \left\langle x^{[i]}, w^{[i]} \right\rangle \right\}_{i=1,\ldots,N}$$

- Think of a sample as one hypothesis about the state
- For feature-based SLAM:

$$x = (\underbrace{x_{1:t}}_{\textbf{poses}}, \underbrace{m_{1,x}, m_{1,y}, \ldots, m_{M,x}, m_{M,y}}_{\textbf{landmarks}})^T$$

Dimensionality Problem

Particle filters are effective in low dimensional spaces as the likely regions of the state space need to be covered with samples.

$$x = (x_{1:t}, m_{1,x}, m_{1,y}, \ldots, m_{M,x}, m_{M,y})^T$$

high-dimensional

Can We Exploit Dependencies Between the Different Dimensions of the State Space?

$$x_{1:t}, m_1, \ \ldots, m_M$$

If We Know the Poses of the Robot, Mapping is Easy!

$$\underline{x_{1:t}}, \underline{m_1, \ \ldots, m_M}$$

Key Idea

$$\underline{x_{1:t},}\ \underline{m_1, \ \ldots, m_M}$$

If we use the particle set only to model the robot's path, each sample is a path hypothesis. For each sample, we can compute an individual map of landmarks.

Rao-Blackwellization

- Factorization to exploit dependencies between variables:

$$p(a, b) \;=\; p(b \mid a)\, p(a)$$

- If $p(b \mid a)$ can be computed efficiently, represent only $p(a)$ with samples and compute $p(b \mid a)$ for every sample

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

poses map observations & movements

$$p(x_{0:t}, m_{1:M} \mid z_{1:t}, u_{1:t}) =$$

First introduced for SLAM by Murphy in 1999

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

poses map observations & movements

$$p(x_{0:t}, m_{1:M} \mid z_{1:t}, u_{1:t}) =$$

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

path posterior map posterior

First introduced for SLAM by Murphy in 1999

Rao-Blackwellization for SLAM

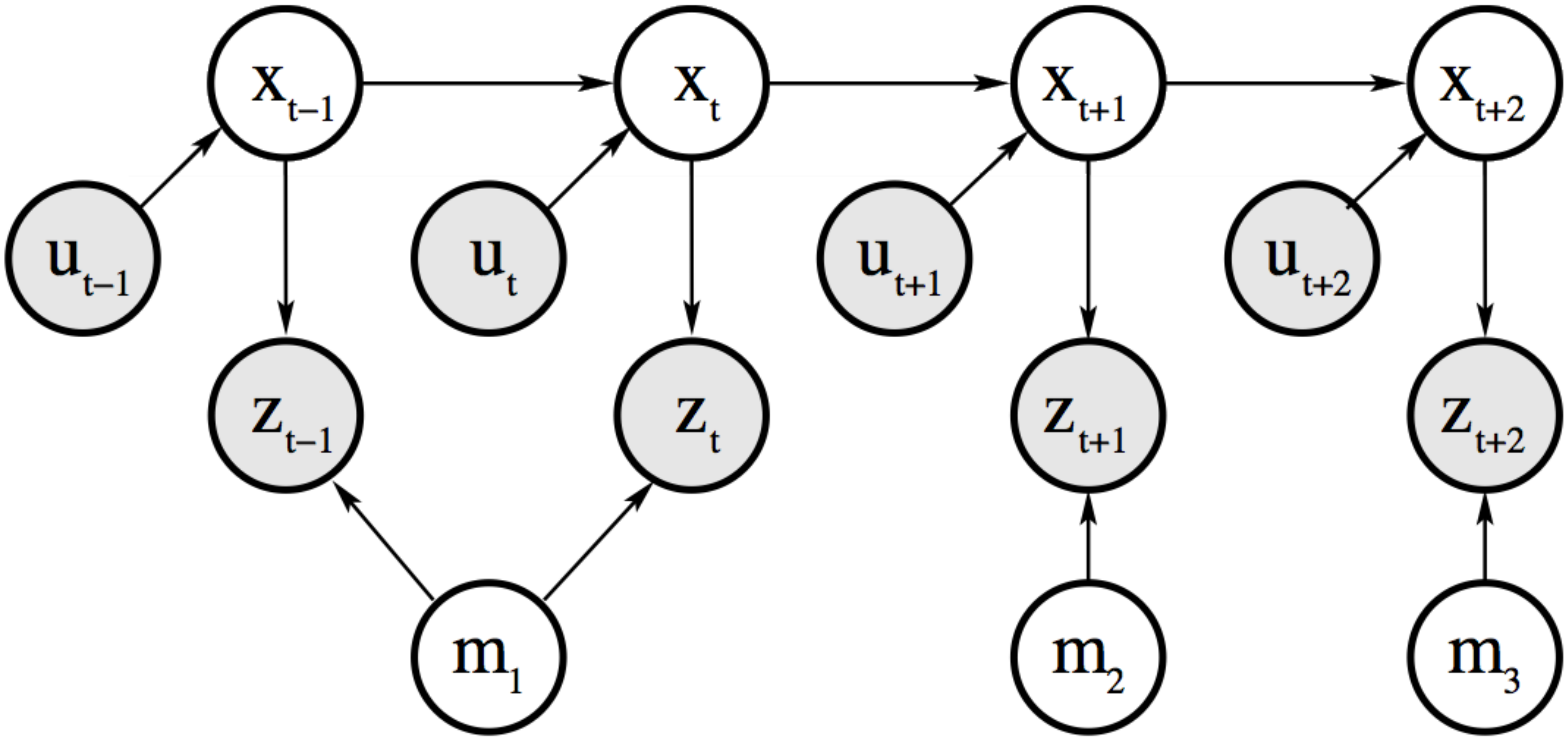
- Factorization of the SLAM posterior

$$p(x_{0:t}, m_{1:M} \mid z_{1:t}, u_{1:t}) =$$
$$p(x_{0:t} \mid z_{1:t}, u_{1:t}) \, \underline{p(m_{1:M} \mid x_{0:t}, z_{1:t})}$$

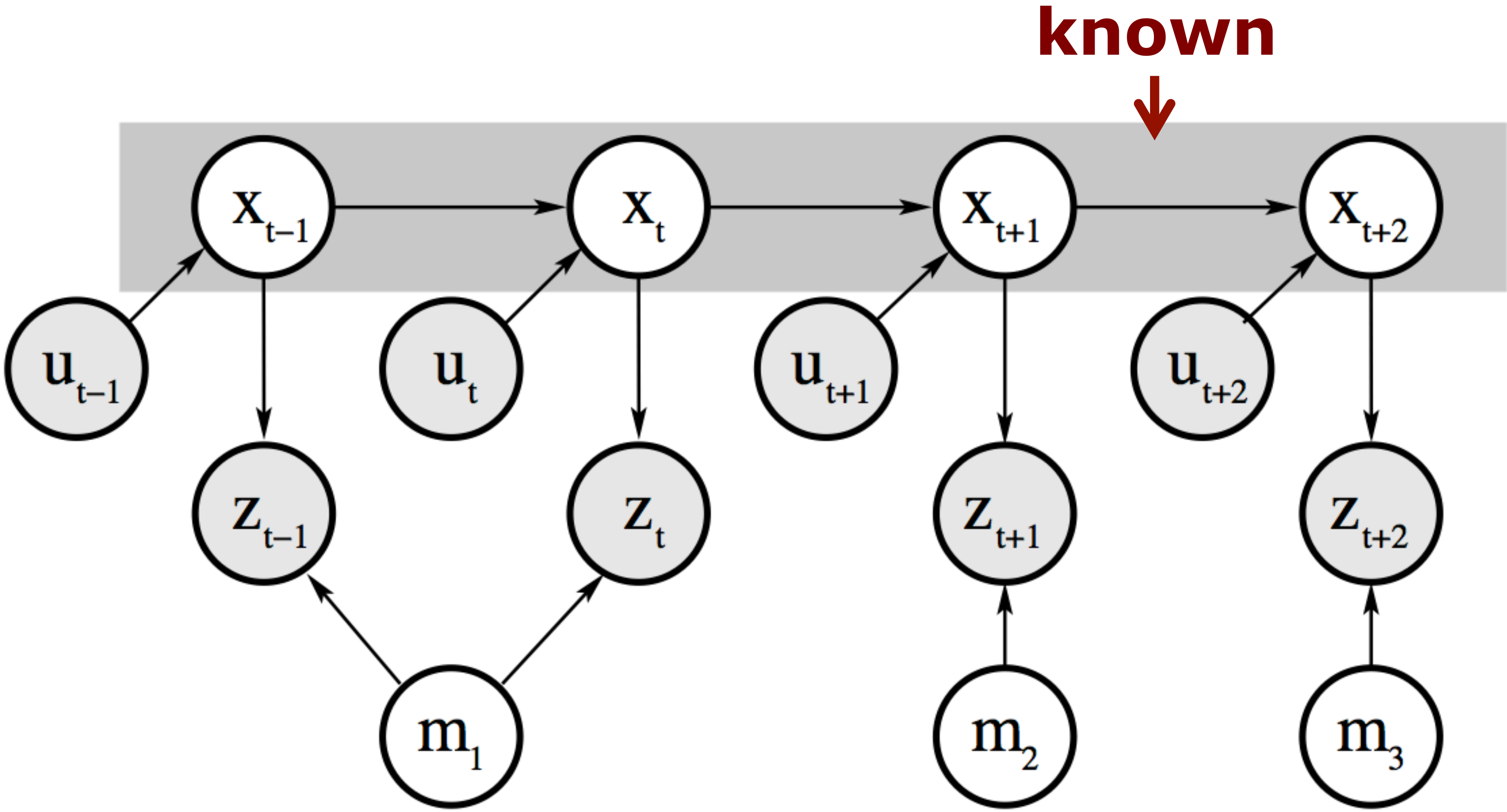
How to compute this term efficiently?

First introduced for SLAM by Murphy in 1999

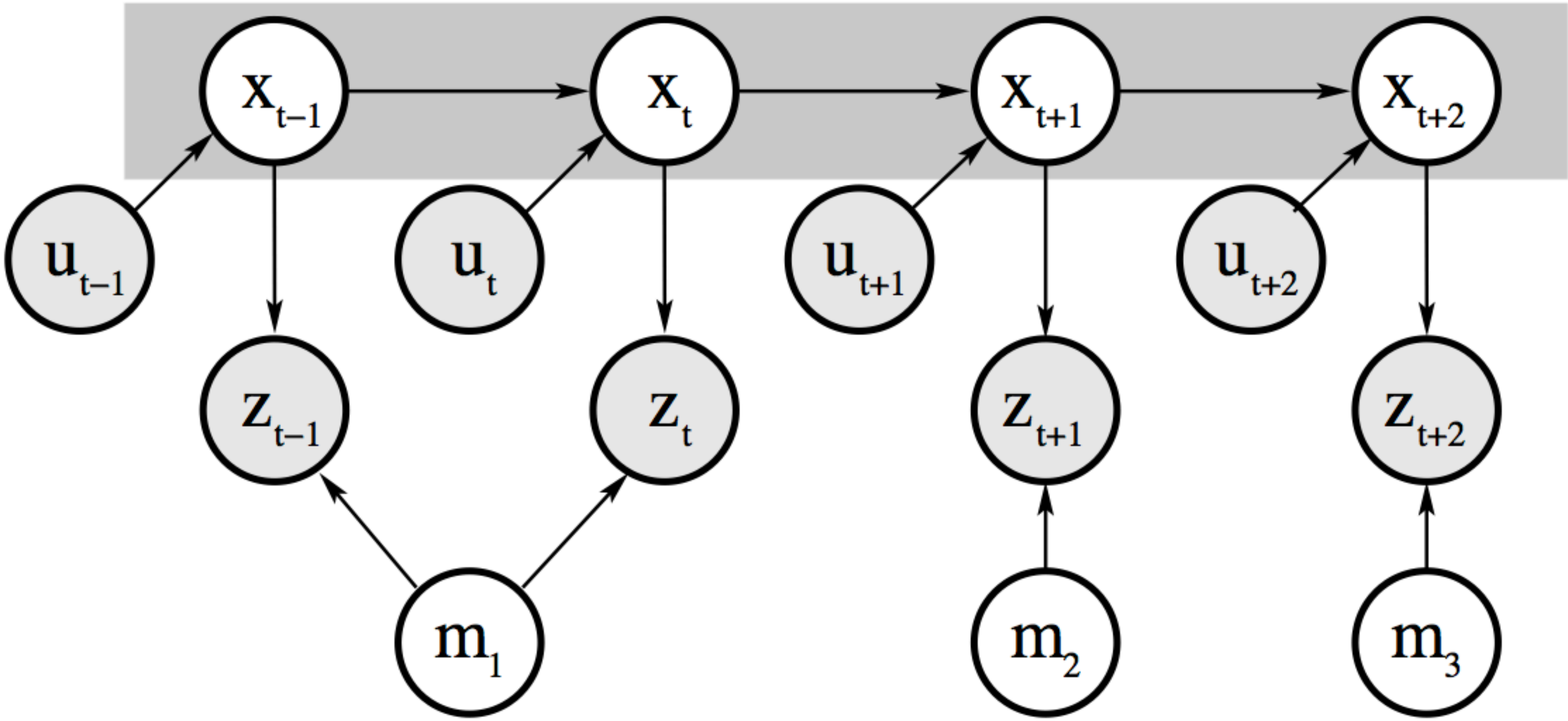
Revisit the Graphical Model



Revisit the Graphical Model



Landmarks are Conditionally Independent Given the Poses



Landmark variables are all disconnected (i.e. independent) given the robot's path

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

$$p(x_{0:t}, m_{1:M} \mid z_{1:t}, u_{1:t}) =$$
$$p(x_{0:t} \mid z_{1:t}, u_{1:t}) \, \underline{p(m_{1:M} \mid x_{0:t}, z_{1:t})}$$

Landmarks are conditionally independent given the poses

First exploited in FastSLAM by Montemerlo et al., 2002

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

$$p(x_{0:t}, m_{1:M} \mid z_{1:t}, u_{1:t}) =$$
$$p(x_{0:t} \mid z_{1:t}, u_{1:t})\, p(m_{1:M} \mid x_{0:t}, z_{1:t})$$
$$p(x_{0:t} \mid z_{1:t}, u_{1:t}) \prod_{i=1}^{M} p(m_i \mid x_{0:t}, z_{1:t})$$

First exploited in FastSLAM by Montemerlo et al., 2002

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

$$p(x_{0:t}, m_{1:M} \mid z_{1:t}, u_{1:t}) =$$

$$p(x_{0:t} \mid z_{1:t}, u_{1:t}) \; p(m_{1:M} \mid x_{0:t}, z_{1:t})$$

$$p(x_{0:t} \mid z_{1:t}, u_{1:t}) \; \prod_{i=1}^{M} \underline{p(m_i \mid x_{0:t}, z_{1:t})}$$

2-dimensional EKFs!

First exploited in FastSLAM by Montemerlo et al., 2002

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

$$p(x_{0:t}, m_{1:M} \mid z_{1:t}, u_{1:t}) =$$

$$p(x_{0:t} \mid z_{1:t}, u_{1:t})\; p(m_{1:M} \mid x_{0:t}, z_{1:t})$$

$$\underline{p(x_{0:t} \mid z_{1:t}, u_{1:t})} \prod_{i=1}^{M} \underline{p(m_i \mid x_{0:t}, z_{1:t})}$$

particle filter similar to MCL

2-dimensional EKFs!

First exploited in FastSLAM by Montemerlo et al., 2002

Modeling the Robot's Path

- Sample-based representation for $p(x_{0:t} \mid z_{1:t}, u_{1:t})$
- Each sample is a path hypothesis

$$x_0 \qquad x_1 \qquad x_2 \qquad \ldots$$

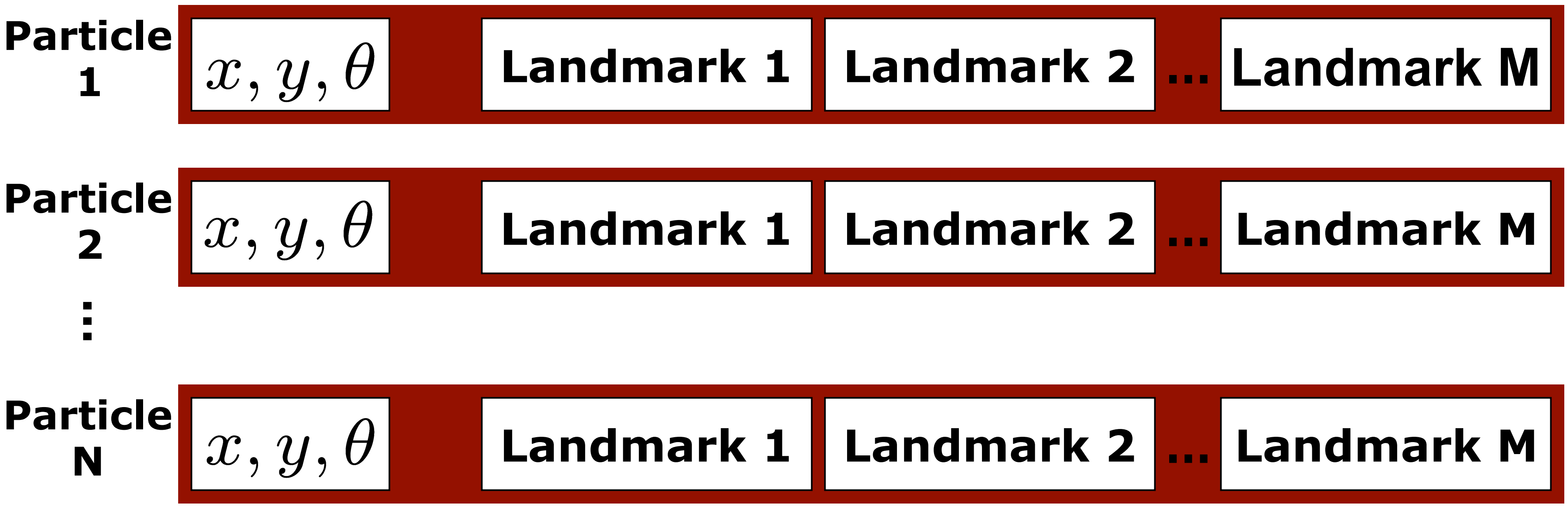
starting location, typically (0,0,0)

pose hypothesis at time t=1

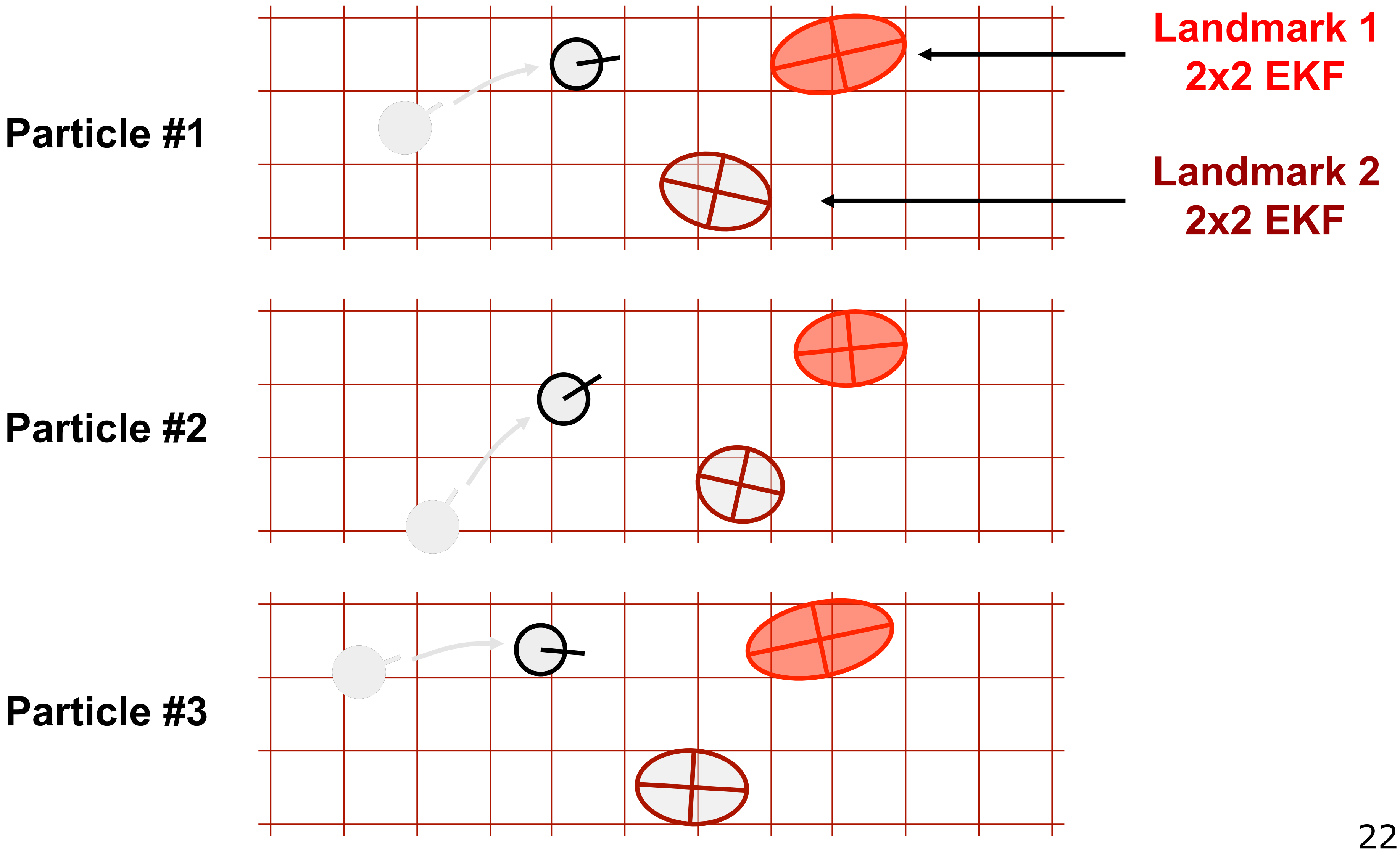
- Past poses of a sample are not revised
- No need to maintain past poses in the sample set

FastSLAM

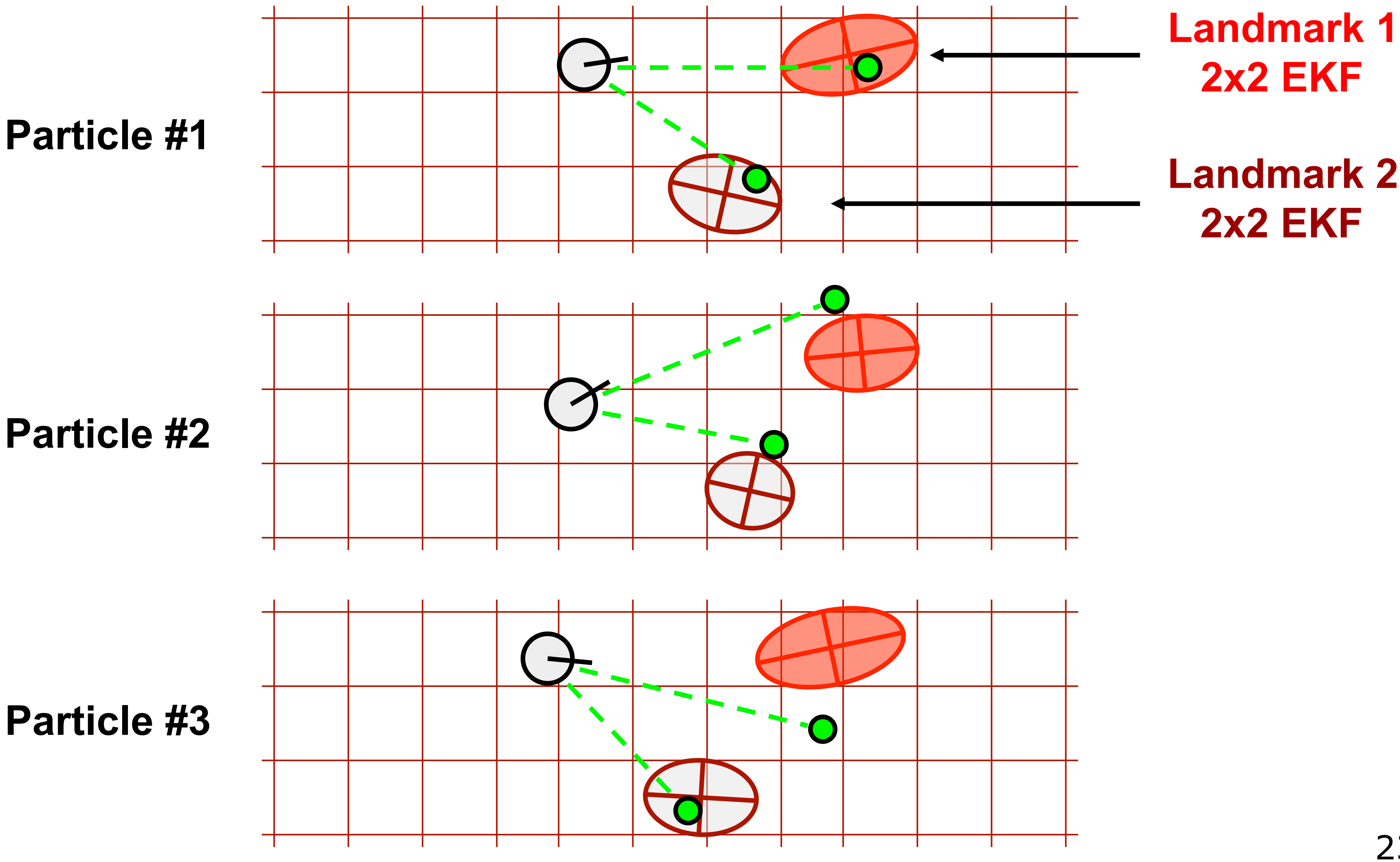
- Proposed by Montemerlo et al. in 2002
- Each landmark is represented by a 2x2 EKF
- Each particle therefore has to maintain M individual EKFs



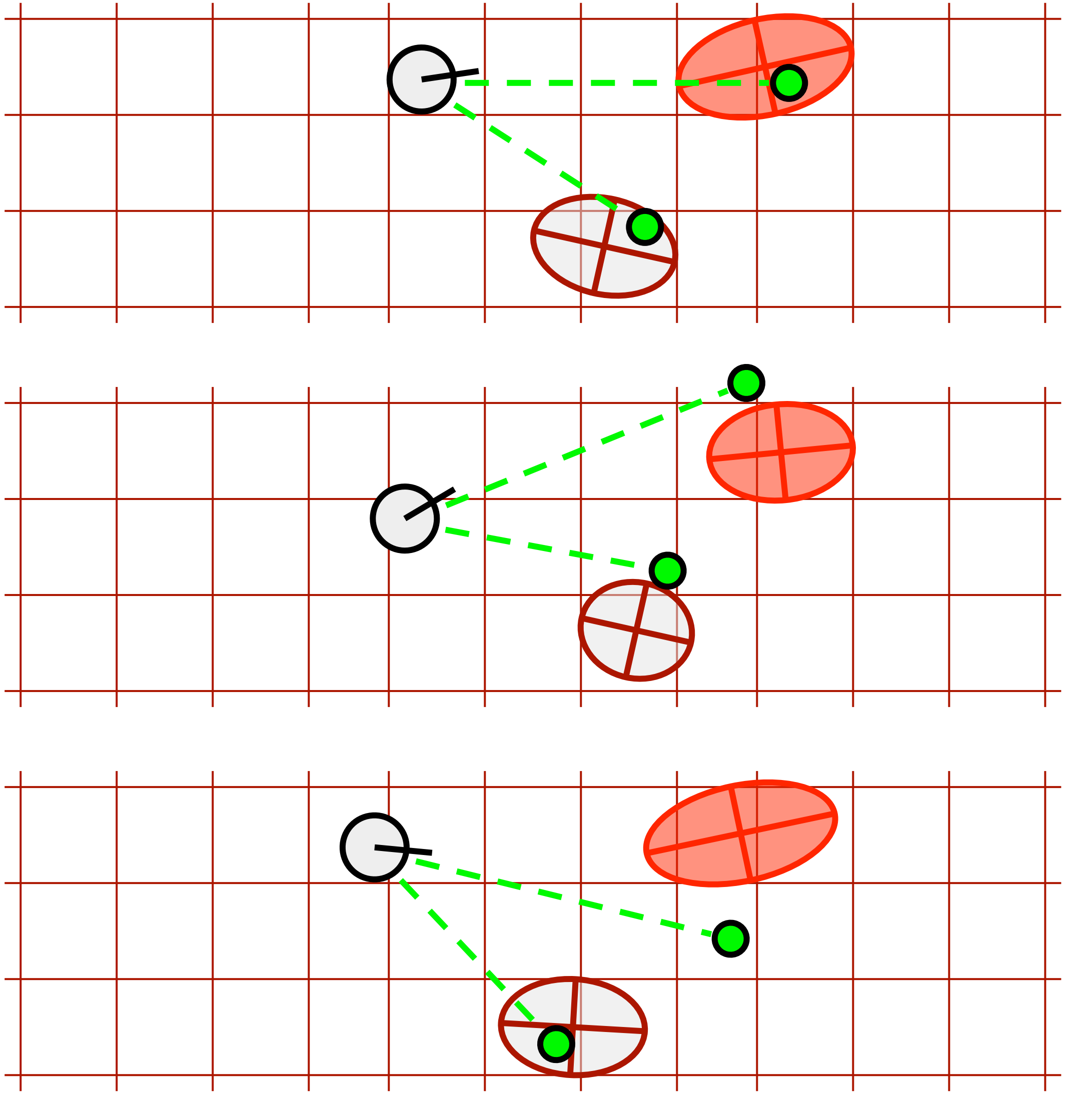
FastSLAM – Action Update



FastSLAM – Sensor Update



FastSLAM – Sensor Update



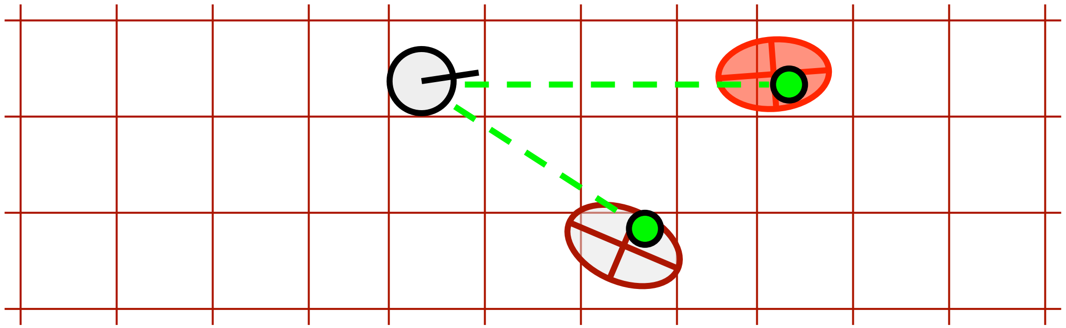
Particle #1 — **Weight = 0.8**

Particle #2 — **Weight = 0.4**

Particle #3 — **Weight = 0.1**

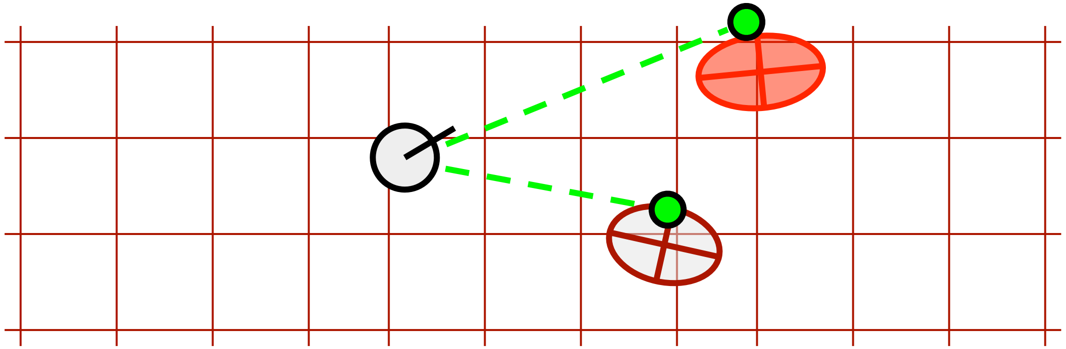
FastSLAM – Sensor Update

Particle #1



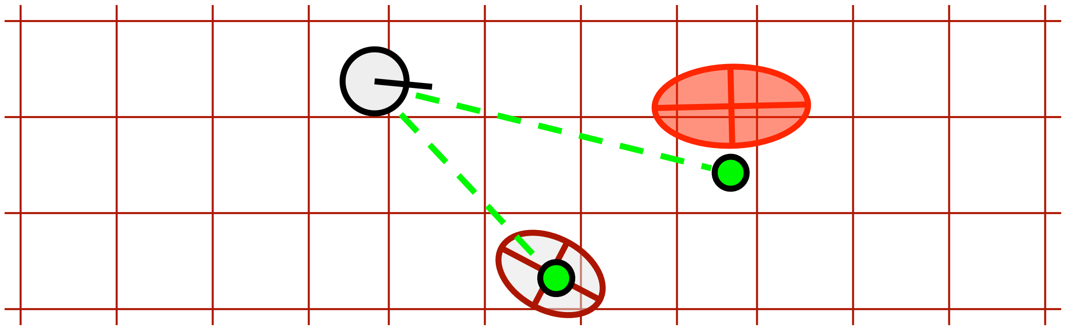
Update map of particle 1

Particle #2



Update map of particle 2

Particle #3



Update map of particle 3

Key Steps of FastSLAM 1.0

- Extend the path posterior by sampling a new pose for each sample

$$x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$$

- Compute particle weight

exp. observation

$$w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \ \exp\left\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} \ (z_t - \hat{z}^{[k]})\right\}$$

measurement covariance

- Update belief of observed landmarks (EKF update rule)
- Resample

FastSLAM 1.0 – Part 1

FastSLAM1.0_known_correspondence$(z_t, c_t, u_t, \mathcal{X}_{t-1})$**:**

for $k = 1$ *to* N *do* // *loop over all particles*

Let $\left\langle x_{t-1}^{[k]}, \left\langle \mu_{1,t-1}^{[k]}, \Sigma_{1,t-1}^{[k]} \right\rangle, \ldots \right\rangle$ *be particle* k *in* $\mathcal{X}_{t-1}$

$x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$ // *sample pose*

FastSLAM 1.0 – Part 1

FastSLAM1.0_known_correspondence$(z_t, c_t, u_t, \mathcal{X}_{t-1})$**:**

for $k = 1$ to N do — // loop over all particles

  Let $\left\langle x_{t-1}^{[k]}, \left\langle \mu_{1,t-1}^{[k]}, \Sigma_{1,t-1}^{[k]} \right\rangle, \ldots \right\rangle$ be particle k in $\mathcal{X}_{t-1}$

  $x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$ — // sample pose

  $j = c_t$ — // observed feature

  if feature j never seen before

    $\mu_{j,t}^{[k]} = h^{-1}(z_t, x_t^{[k]})$ — // initialize mean

    $H = h'(\mu_{j,t}^{[k]}, x_t^{[k]})$ — // calculate Jacobian

    $\Sigma_{j,t}^{[k]} = H^{-1}\, Q_t\, (H^{-1})^T$ — // initialize covariance

    $w^{[k]} = p_0$ — // default importance weight

  else

FastSLAM 1.0 – Part 2

else

$\langle \mu_{j,t}^{[k]}, \Sigma_{j,t}^{[k]} \rangle = \textit{EKF-Update}(\ldots)$ // *update landmark*

$$w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} (z_t - \hat{z}^{[k]}) \right\}$$

measurement cov. $Q = H\,\Sigma_{j,t-1}^{[k]}\,H^T + Q_t$ **exp. observation**

endif

for all unobserved features j' *do*

$\langle \mu_{j',t}^{[k]}, \Sigma_{j',t}^{[k]} \rangle = \langle \mu_{j',t-1}^{[k]}, \Sigma_{j',t-1}^{[k]} \rangle$ // *leave unchanged*

endfor

endfor

$$\mathcal{X}_t = \text{resample}\left(\left\langle x_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]} \right\rangle, \ldots, w^{[k]} \right\rangle_{k=1,\ldots,N} \right)$$

return $\mathcal{X}_t$

FastSLAM 1.0 – Part 2 (long)

EKF update

else

$\hat{z}^{[k]} = h(\mu^{[k]}_{j,t-1}, x^{[k]}_t)$ // *measurement prediction*

$H = h'(\mu^{[k]}_{j,t-1}, x^{[k]}_t)$ // *calculate Jacobian*

$Q = H\ \Sigma^{[k]}_{j,t-1}\ H^T + Q_t$ // *measurement covariance*

$K = \Sigma^{[k]}_{j,t-1}\ H^T\ Q^{-1}$ // *calculate Kalman gain*

$\mu^{[k]}_{j,t} = \mu^{[k]}_{j,t-1} + K(z_t - \hat{z}^{[k]})$ // *update mean*

$\Sigma^{[k]}_{j,t} = (I - K\ H)\Sigma^{[k]}_{j,t-1}$ // *update covariance*

$w^{[k]} = |2\pi Q|^{-\frac{1}{2}}\ \exp\left\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T\ Q^{-1}\ (z_t - \hat{z}^{[k]})\right\}$ // *importance factor*

endif

for all unobserved features j' do

$\langle \mu^{[k]}_{j',t}, \Sigma^{[k]}_{j',t} \rangle = \langle \mu^{[k]}_{j',t-1}, \Sigma^{[k]}_{j',t-1} \rangle$ // *leave unchanged*

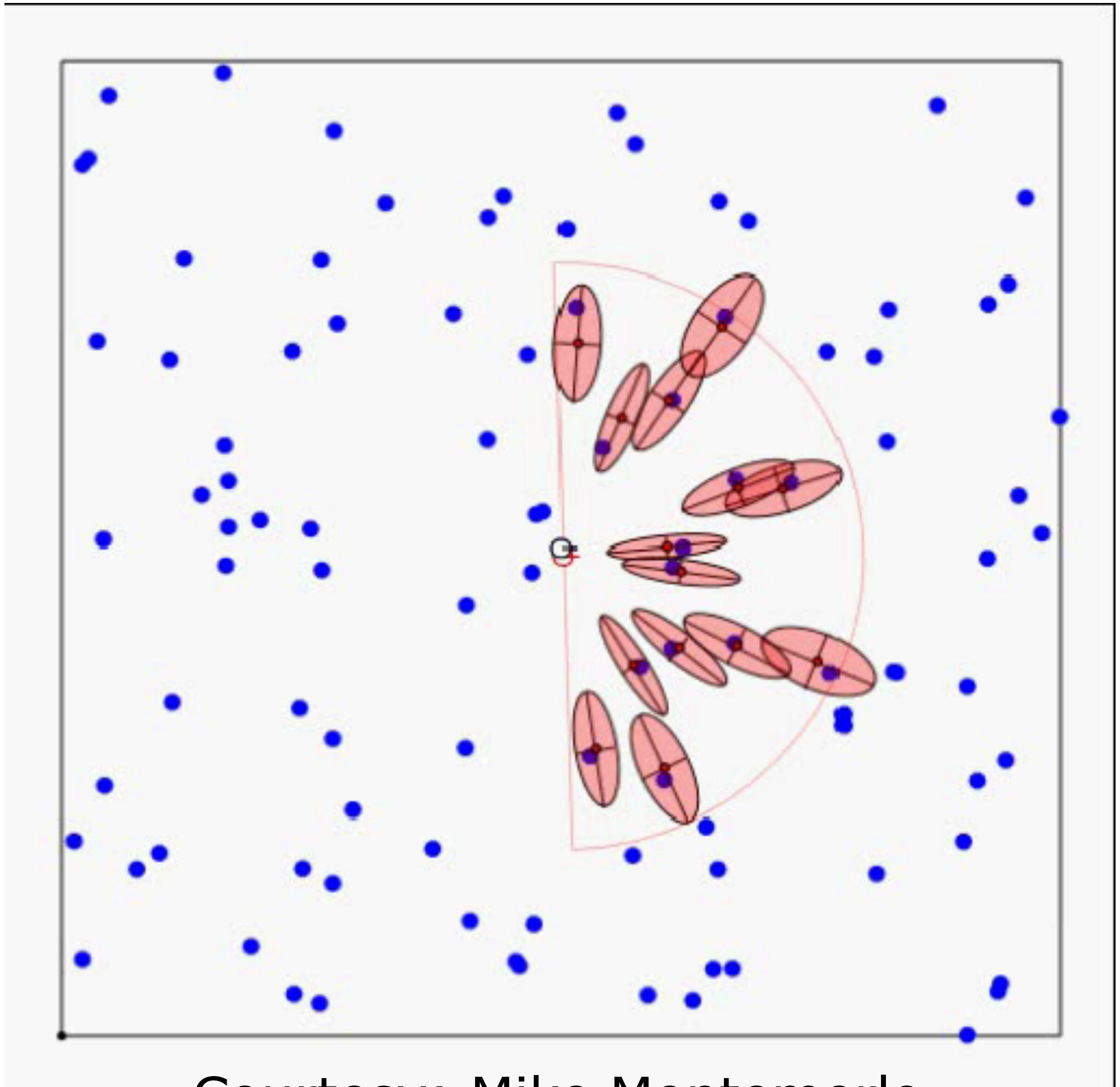
endfor

endfor

$\mathcal{X}_t = \text{resample}\left(\left\langle x^{[k]}_t, \left\langle \mu^{[k]}_{1,t}, \Sigma^{[k]}_{1,t} \right\rangle, \ldots, w^{[k]} \right\rangle_{k=1,\ldots,N}\right)$

return $\mathcal{X}_t$

FastSLAM in Action



Courtesy: Mike Montemerlo

The Weight is a Result From the Importance Sampling Principle

- Importance weight is given by the ratio of target and proposal in $x^{[k]}$
- See: importance sampling principle

$$w^{[k]} = \frac{\text{target}(x^{[k]})}{\text{proposal}(x^{[k]})}$$

The Importance Weight

- The target distribution is

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

- The proposal distribution is

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

- Proposal is used step-by-step

$$\begin{aligned} & p(x_{1:t} \mid z_{1:t-1}, u_{1:t}) \\ = \quad & \underbrace{p(x_t \mid x_{t-1}, u_t)}_{\text{from } \mathcal{X}_{t-1} \text{ to } \bar{\mathcal{X}}_t} \underbrace{p(x_{1:t-1} \mid z_{1:t-1}, u_{1:t-1})}_{\mathcal{X}_{t-1}} \end{aligned}$$

The Importance Weight

$$
\begin{aligned}
w^{[k]} &= \frac{\mathrm{target}(x^{[k]})}{\mathrm{proposal}(x^{[k]})} \\
&= \frac{p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t})}{p(x_t^{[k]} \mid x_{t-1}, u_t)\, p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}
\end{aligned}
$$

The Importance Weight

$$
\begin{aligned}
w^{[k]} &= \frac{\text{target}(x^{[k]})}{\text{proposal}(x^{[k]})} \\
&= \frac{p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t})}{p(x_t^{[k]} \mid x_{t-1}, u_t)\; p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}
\end{aligned}
$$

Bayes rule + factorization

The Importance Weight

$$
\begin{aligned}
w^{[k]} &= \frac{\text{target}(x^{[k]})}{\text{proposal}(x^{[k]})} \\
&= \frac{p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t})}{p(x_t^{[k]} \mid x_{t-1}, u_t)\, p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})} \\
&= \frac{\eta\, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1})\, p(x_t \mid x_{t-1}^{[k]}, u_t)}{p(x_t^{[k]} \mid x_{t-1}^{[k]}, u_t)} \\
&\quad \frac{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}
\end{aligned}
$$

The Importance Weight

$$
\begin{aligned}
w^{[k]} &= \frac{\text{target}(x^{[k]})}{\text{proposal}(x^{[k]})} \\
&= \frac{p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t})}{p(x_t^{[k]} \mid x_{t-1}, u_t)\; p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})} \\
&= \frac{\eta\; p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1})\; \cancel{p(x_t \mid x_{t-1}^{[k]}, u_t)}}{\cancel{p(x_t^{[k]} \mid x_{t-1}^{[k]}, u_t)}} \\
&\quad \frac{\cancel{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}}{\cancel{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}}
\end{aligned}
$$

The Importance Weight

$$
\begin{aligned}
w^{[k]} &= \frac{\mathrm{target}(x^{[k]})}{\mathrm{proposal}(x^{[k]})} \\
&= \frac{p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t})}{p(x_t^{[k]} \mid x_{t-1}, u_t)\; p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})} \\
&= \frac{\eta\; p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1})\; \cancel{p(x_t \mid x_{t-1}^{[k]}, u_t)}}{\cancel{p(x_t^{[k]} \mid x_{t-1}^{[k]}, u_t)}} \\
&\quad \frac{\cancel{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}}{\cancel{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}} \\
&= \eta\; p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1})
\end{aligned}
$$

The Importance Weight

- Integrating over the pose of the observed landmark leads to

$$
\begin{aligned}
w^{[k]} & \\
&= \eta \, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}) \\
&= \eta \int p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}, m_j) \, p(m_j \mid x_{1:t}^{[k]}, z_{1:t-1}) \, dm_j
\end{aligned}
$$

The Importance Weight

- Integrating over the pose of the observed landmark leads to

$$
\begin{aligned}
w^{[k]} & \\
&= \eta\, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}) \\
&= \eta \int p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}, m_j)\, p(m_j \mid x_{1:t}^{[k]}, z_{1:t-1})\, dm_j \\
&= \eta \int p(z_t \mid x_t^{[k]}, m_j)\;\, p(m_j \mid x_{1:t-1}^{[k]}, z_{1:t-1})\;\, dm_j
\end{aligned}
$$

The Importance Weight

- Integrating over the pose of the observed landmark leads to

$$
\begin{aligned}
w^{[k]} & \\
= \quad & \eta \, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}) \\
= \quad & \eta \int p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}, m_j) \, p(m_j \mid x_{1:t}^{[k]}, z_{1:t-1}) \, dm_j \\
= \quad & \eta \int \underbrace{p(z_t \mid x_t^{[k]}, m_j)}_{\mathcal{N}(z_t; \hat{z}^{[k]}, Q_t)} \, \underbrace{p(m_j \mid x_{1:t-1}^{[k]}, z_{1:t-1})}_{\mathcal{N}(m_j; \mu_{j,t-1}^{[k]}, \Sigma_{j,t-1}^{[k]})} \, dm_j
\end{aligned}
$$

The Importance Weight

- This leads to

$$w^{[k]} = \eta \int \underbrace{p(m_j \mid x^{[k]}_{1:t-1}, z_{1:t-1})}_{\mathcal{N}(m_j;\mu^{[k]}_{j,t-1},\Sigma^{[k]}_{j,t-1})} \underbrace{p(z_t \mid x^{[k]}_t, m_j)}_{\mathcal{N}(z_t;\hat{z}^{[k]},Q_t)} \, dm_j$$

$$Q = H \, \Sigma^{[k]}_{j,t-1} \, H^T + Q_t$$

measurement covariance (pose uncertainty of the landmark estimate plus measurement noise)

The Importance Weight

- This leads to

$$w^{[k]} = \eta \int \underbrace{p(m_j \mid x_{1:t-1}^{[k]}, z_{1:t-1})}_{\mathcal{N}(m_j; \mu_{j,t-1}^{[k]}, \Sigma_{j,t-1}^{[k]})} \underbrace{p(z_t \mid x_t^{[k]}, m_j)}_{\mathcal{N}(z_t; \hat{z}^{[k]}, Q_t)} \, dm_j$$

$$Q = H \, \Sigma_{j,t-1}^{[k]} \, H^T + Q_t$$

$$w^{[k]} \simeq |2\pi Q|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} (z_t - \hat{z}^{[k]})\right\}$$

FastSLAM 1.0 – Part 2

else

$\langle \mu_{j,t}^{[k]}, \Sigma_{j,t}^{[k]} \rangle = \textit{EKF-Update}(\dots)$ // *update landmark*

$$\boxed{w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} (z_t - \hat{z}^{[k]})\right\}}$$

endif

for all unobserved features j' do

$\langle \mu_{j',t}^{[k]}, \Sigma_{j',t}^{[k]} \rangle = \langle \mu_{j',t-1}^{[k]}, \Sigma_{j',t-1}^{[k]} \rangle$ // *leave unchanged*

endfor

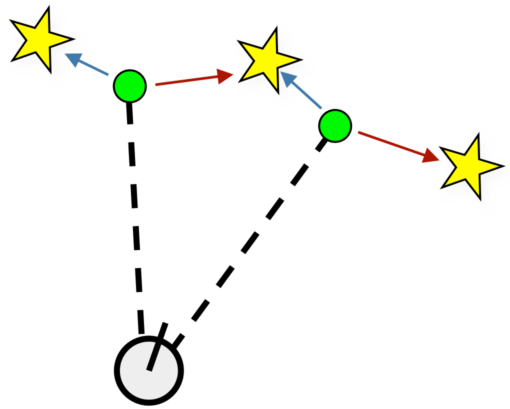
endfor

$$\mathcal{X}_t = \text{resample}\left(\left\langle x_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]} \right\rangle, \dots, w^{[k]} \right\rangle_{k=1,\dots,N}\right)$$

return $\mathcal{X}_t$

Data Association Problem

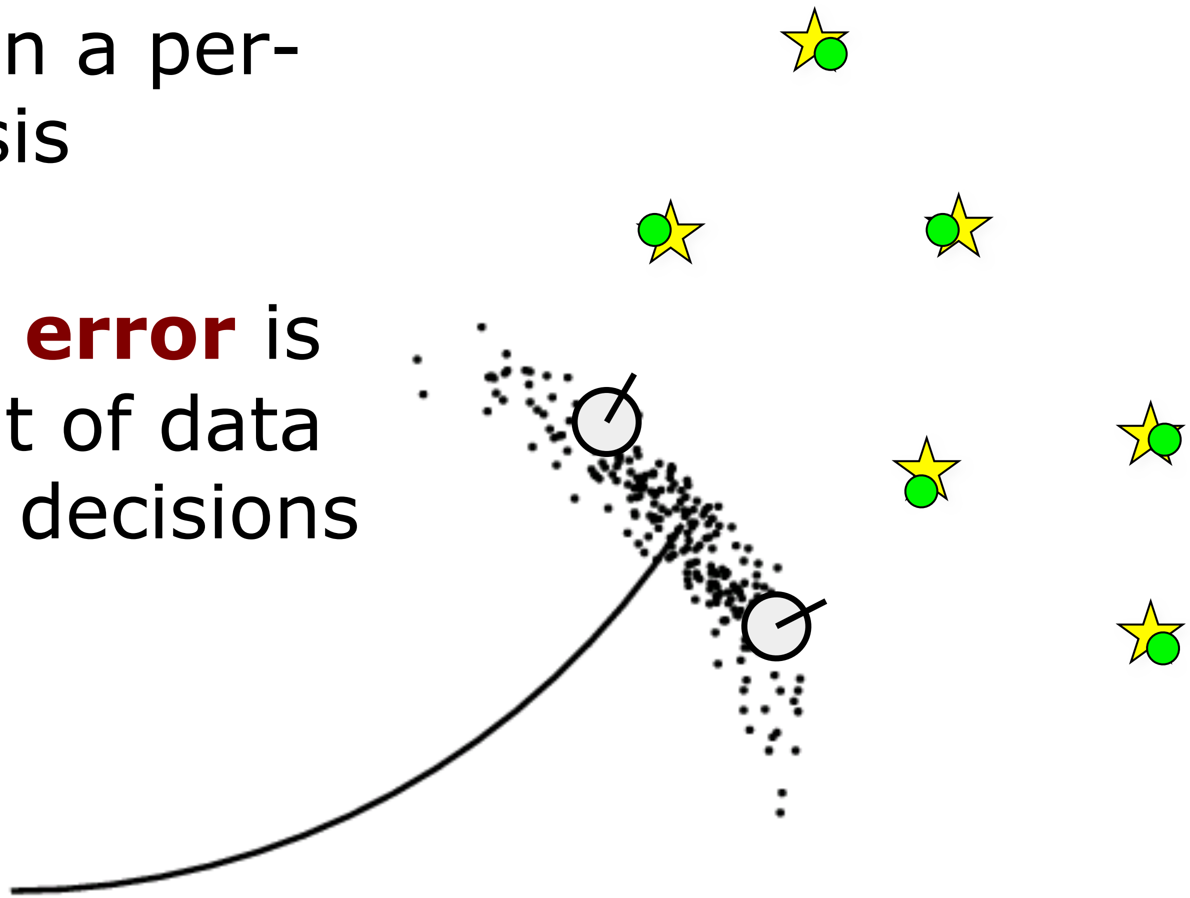
- Which observation belongs to which landmark?



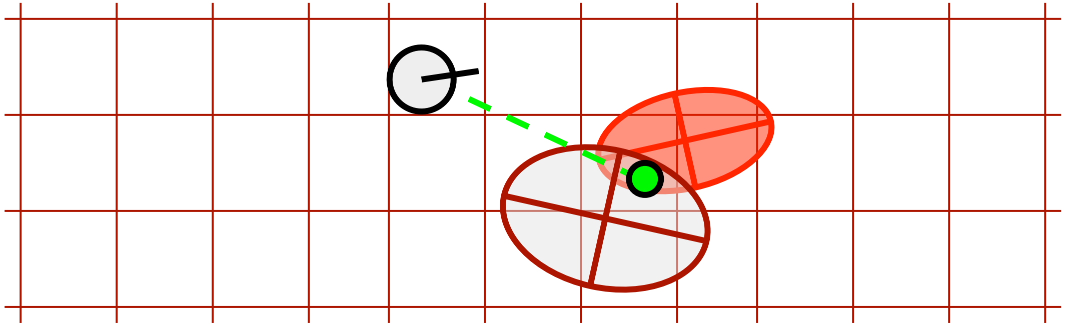
- More than one possible association
- **Potential data associations depend on the pose of the robot**

Particles Support for Multi-Hypotheses Data Association

- Decisions on a per-particle basis
- Robot pose **error** is factored out of data association decisions



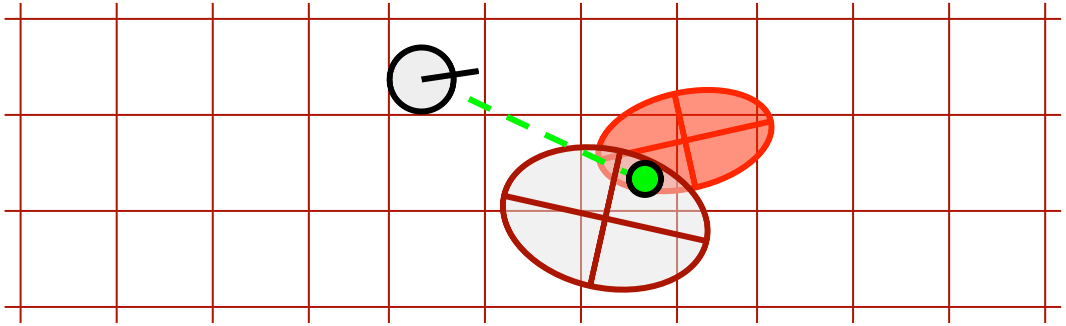
Per-Particle Data Association



Was the observation generated by the **red** or by the **brown** landmark?

P(observation|red) = 0.3   P(observation|brown) = 0.7

Per-Particle Data Association

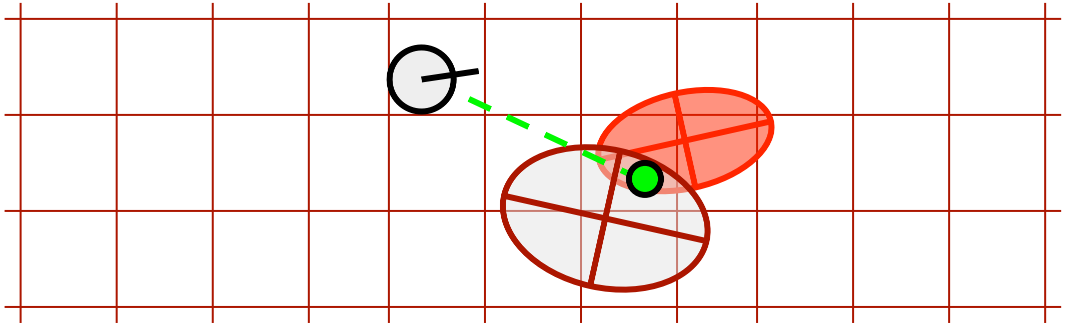


Was the observation generated by the **red** or by the **brown** landmark?

P(observation|red) = 0.3 P(observation|brown) = 0.7

- Two options for per-particle data association
  - Pick the most probable match
  - Pick an random association weighted by the observation likelihoods
- If the probability for an assignment is too low, generate a new landmark

Per-Particle Data Association



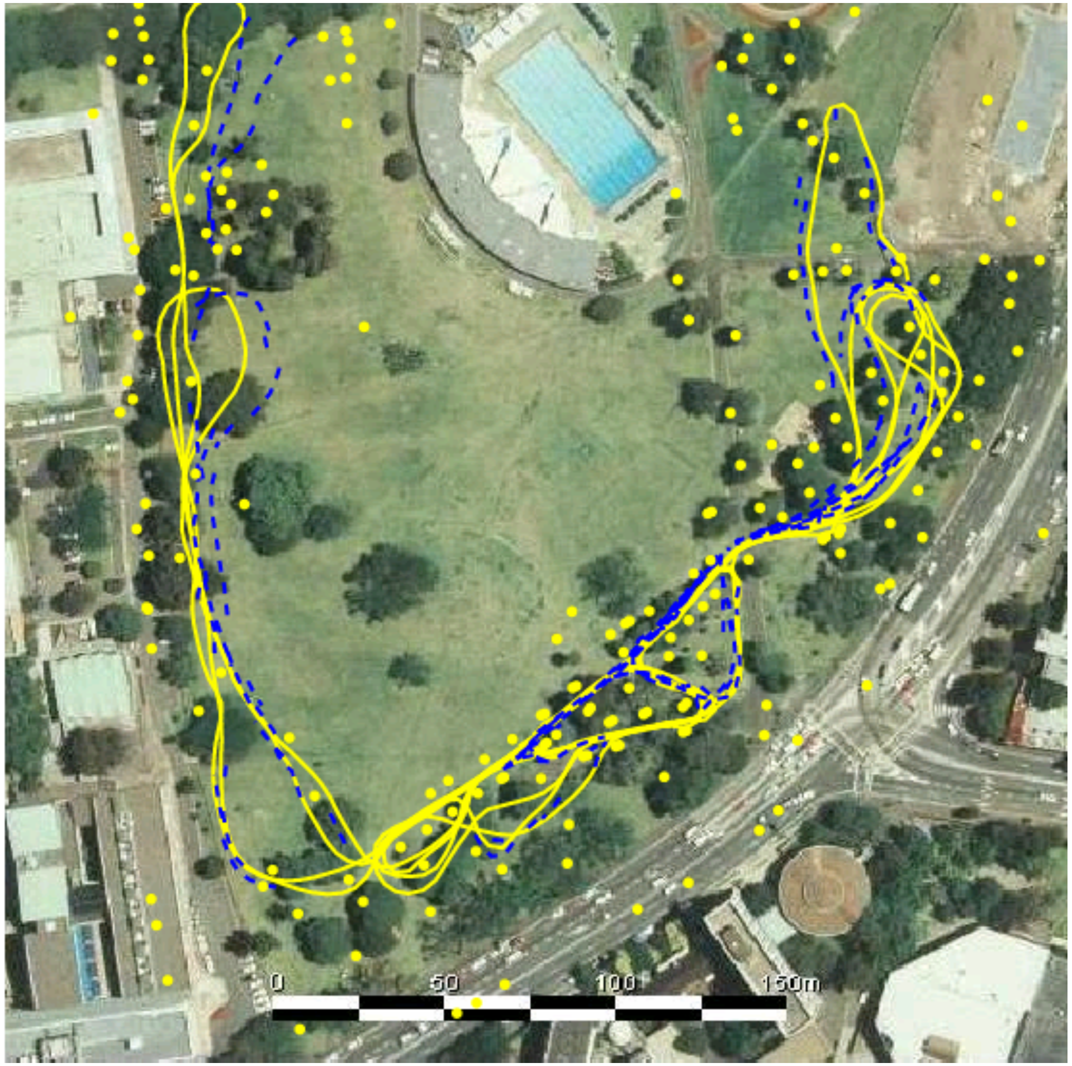
Was the observation generated by the **red** or by the **brown** landmark?

- Multi-modal belief
- Pose error is factored out of data association decisions
- **Simple but effective** data association
- Big **advantage of FastSLAM** over EKF

Results – Victoria Park

- 4 km traverse
- < 2.5 m RMS position error
- 100 particles

Blue = GPS
Yellow = FastSLAM

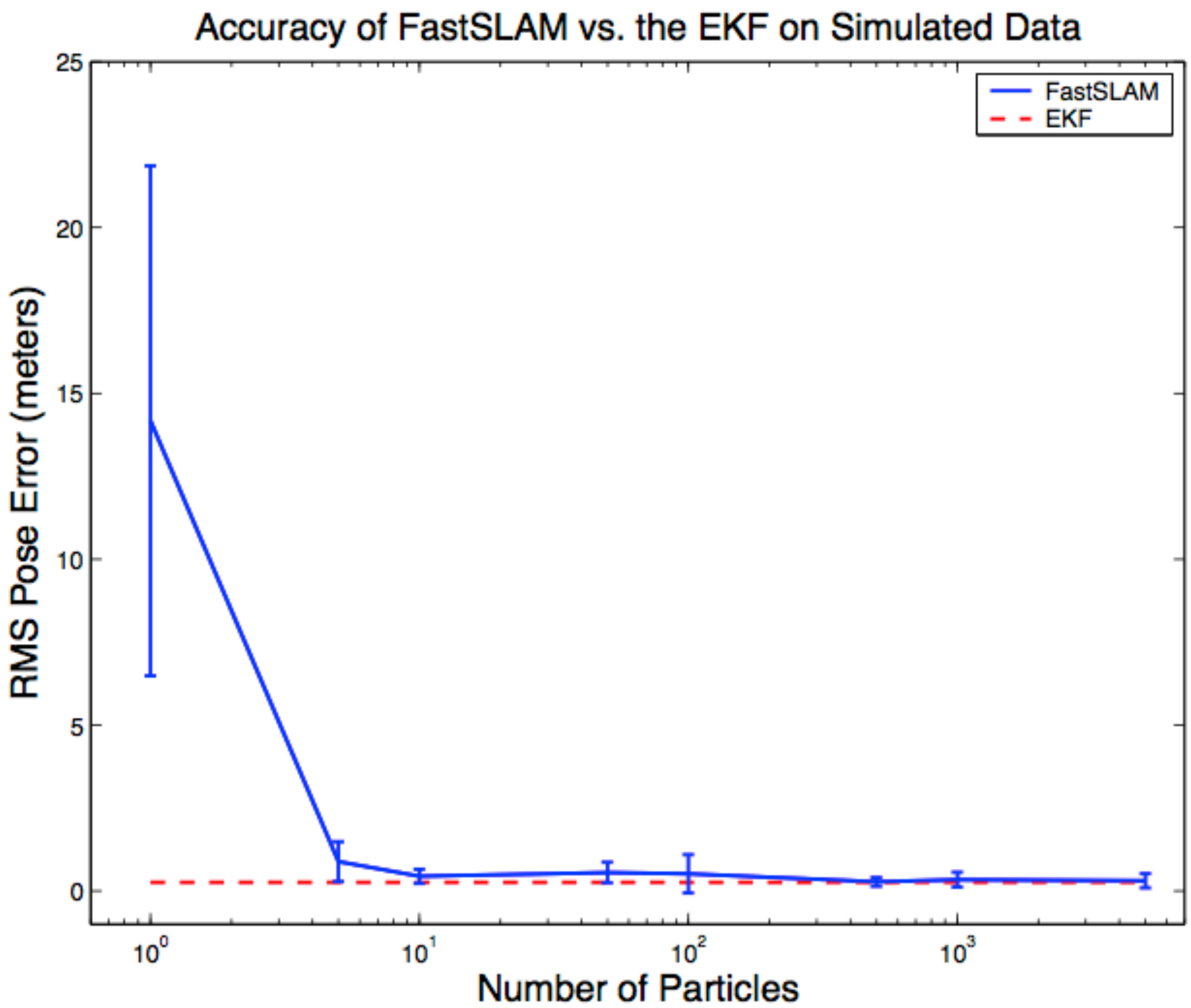


Courtesy: Mike Montemerlo

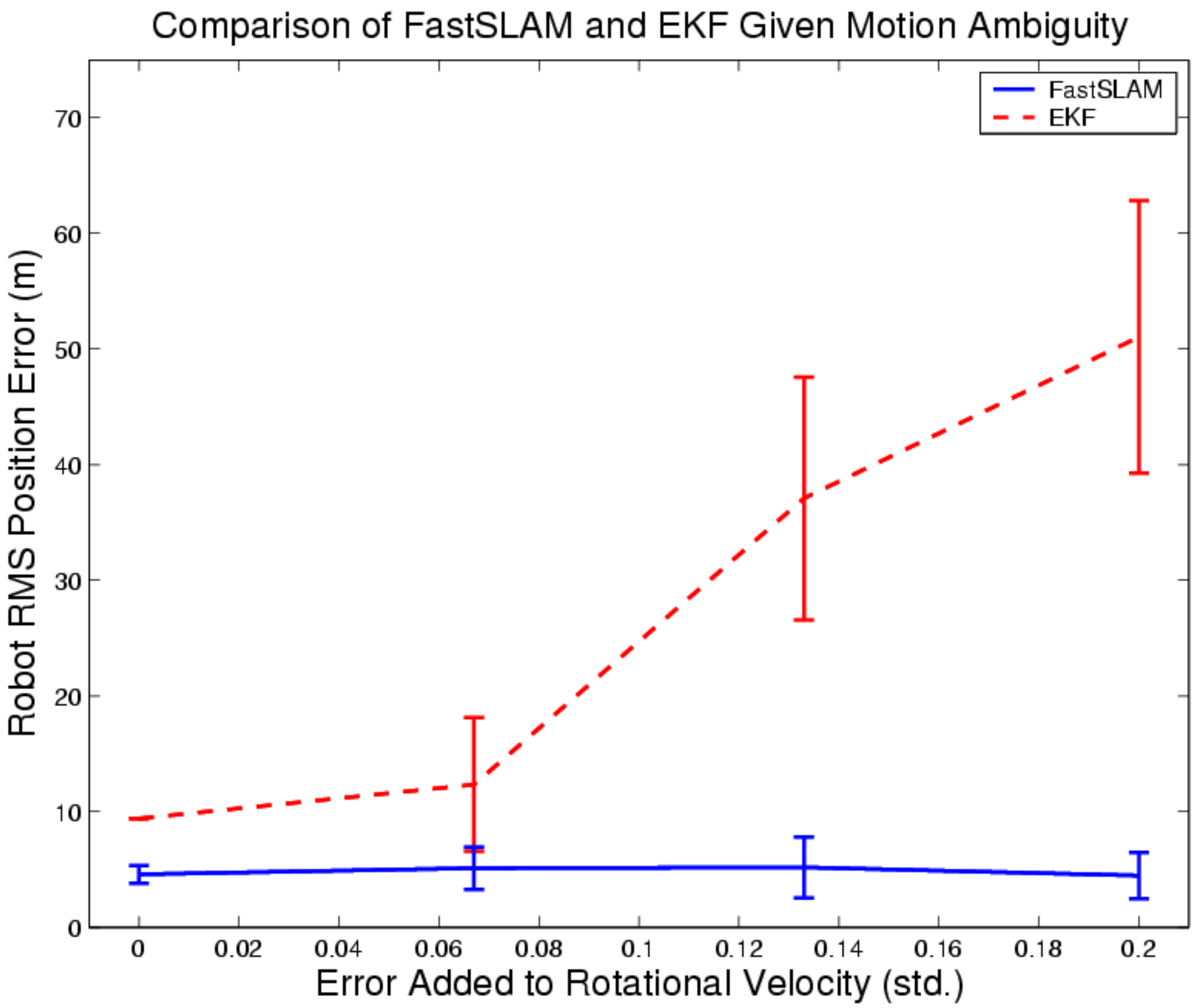
Results – Victoria Park (Video)

Courtesy: Mike Montemerlo

Results (Sample Size)



Results (Motion Uncertainty)



FastSLAM 1.0 Summary

- Use a particle filter to model the belief
- Factors the SLAM posterior into low-dimensional estimation problems
- Model only the robot’s path by sampling
- Compute the landmarks given the path
- Per-particle data association
- No robot pose uncertainty in the per-particle data association

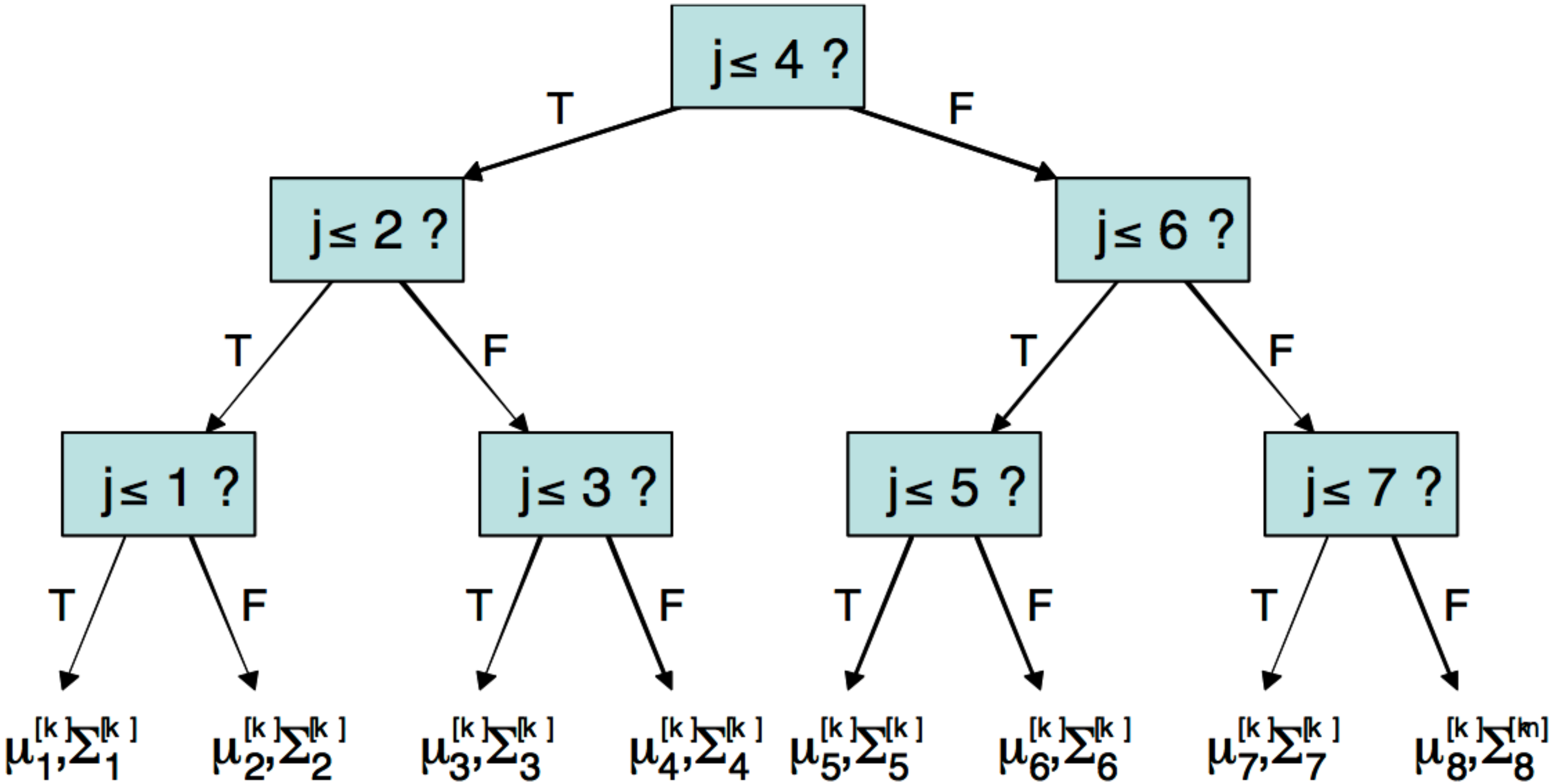
FastSLAM Complexity – Simple Implementation

- Update robot particles based on the control $\mathcal{O}(N)$
- Incorporate an observation into the Kalman filters $\mathcal{O}(N)$
- Resample particle set $\mathcal{O}(NM)$

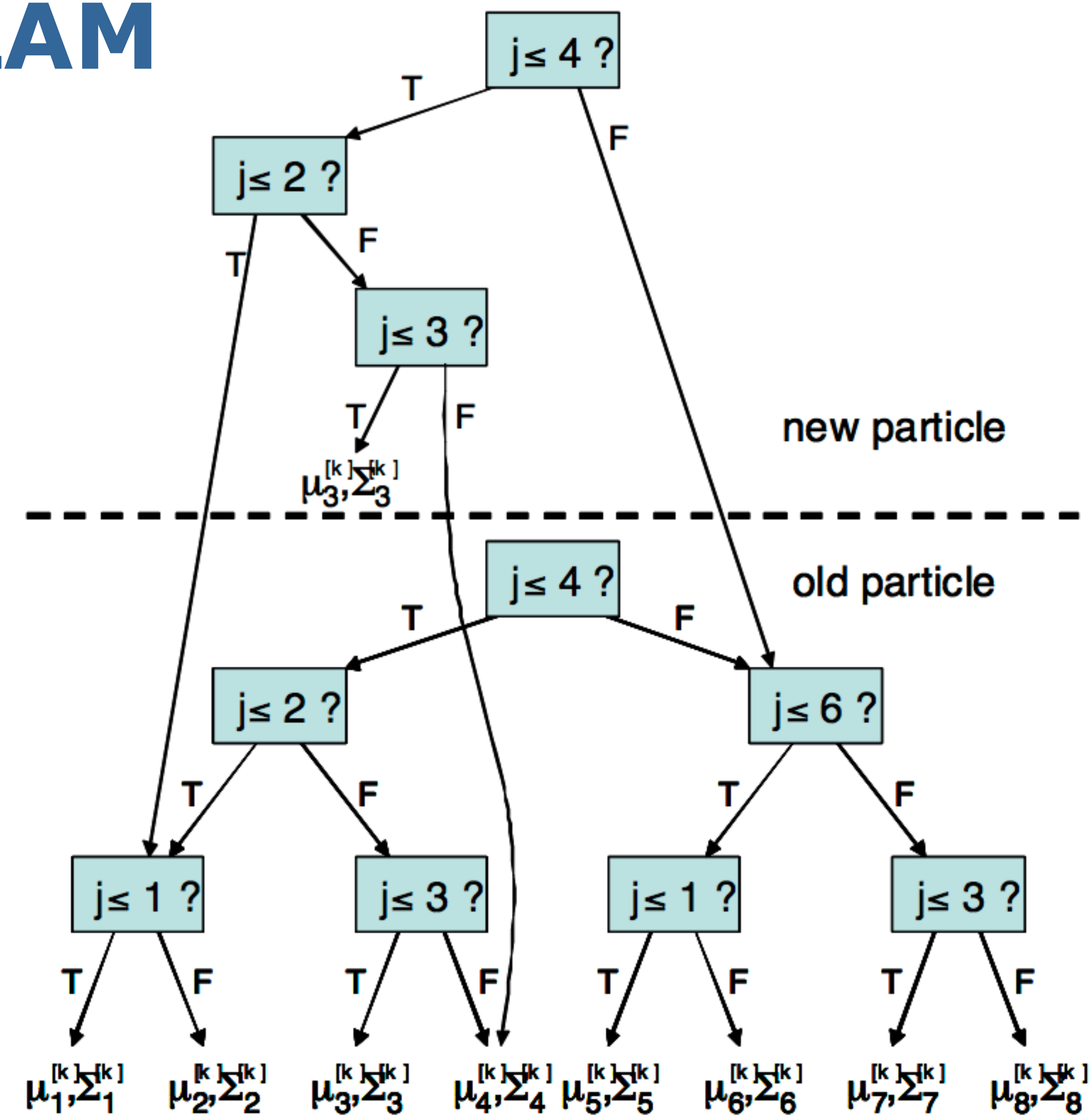
$$\mathcal{O}(NM)$$

N = Number of particles
M = Number of map features

A Better Data Structure for FastSLAM



A Better Data Structure for FastSLAM



FastSLAM Complexity

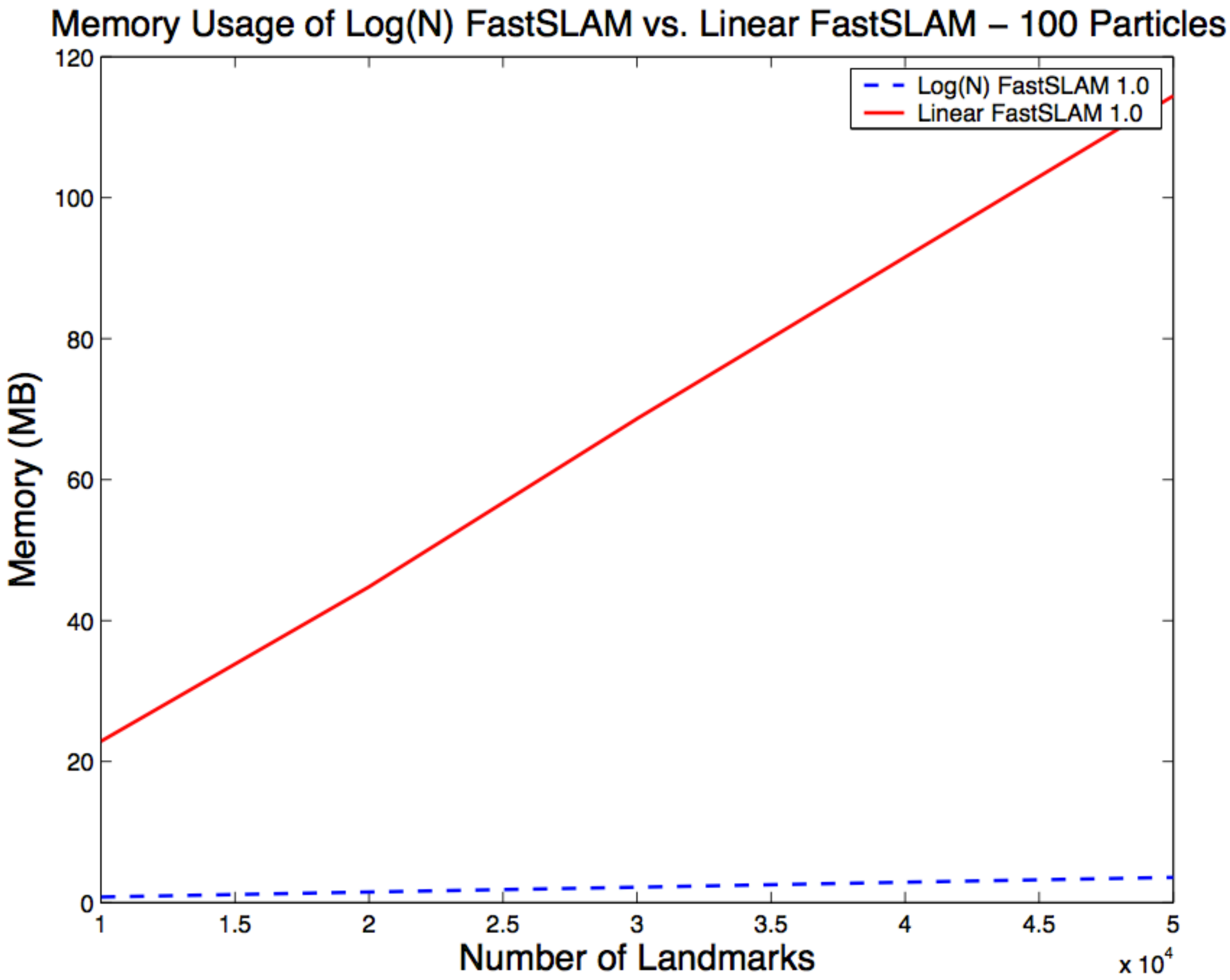
- Update robot particles based on the control $\mathcal{O}(N)$
- Incorporate an observation into the Kalman filters $\mathcal{O}(N \log M)$
- Resample particle set $\mathcal{O}(N \log M)$

$$\overline{\phantom{\mathcal{O}(N \log M)}}$$

$$\mathcal{O}(N \log M)$$

N = Number of particles
M = Number of map features

Memory Complexity



FastSLAM 1.0

- FastSLAM 1.0 uses the motion model as the proposal distribution

$$x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$$

- **Is there a better distribution to sample from?**

[Montemerlo et al., 2002]

FastSLAM 1.0 to FastSLAM 2.0

- FastSLAM 1.0 uses the motion model as the proposal distribution

$$x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$$

- FastSLAM 2.0 **considers also the measurements during sampling**
- Especially useful if an accurate sensor is used (compared to the motion noise)

[Montemerlo et al., 2003]

FastSLAM 2.0 (Informally)

- FastSLAM 2.0 samples from

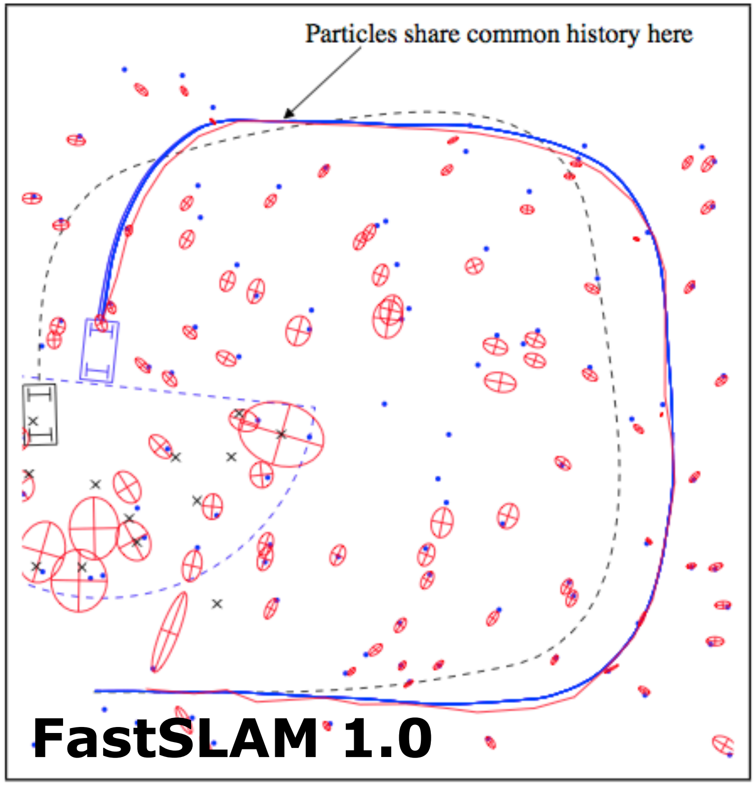
$$x_t^{[k]} \quad \sim \quad p(x_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t})$$

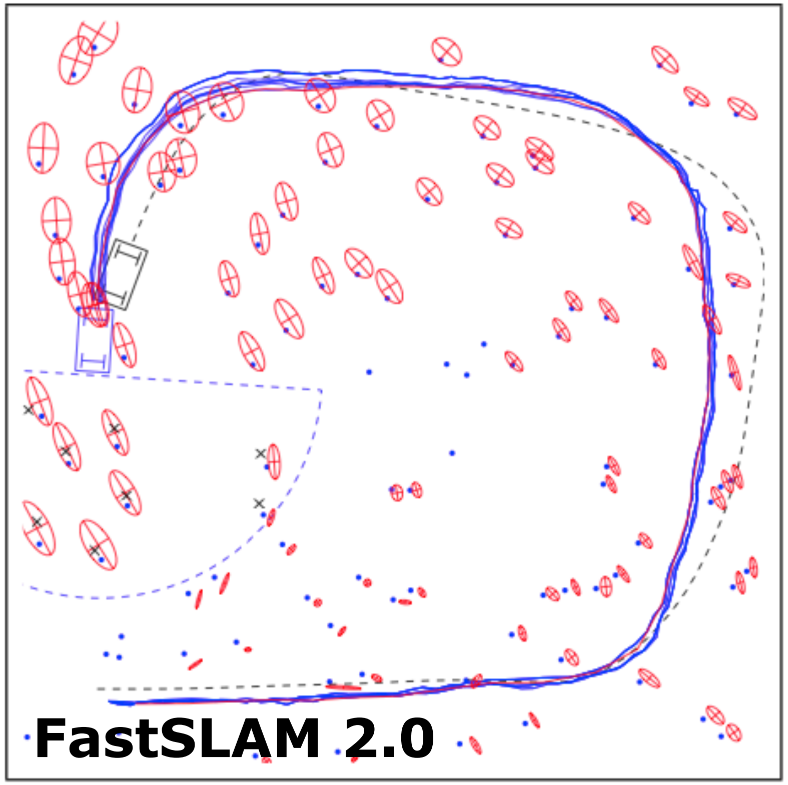
- Results in a more peaked proposal distribution
- Less particles are required
- More robust and accurate
- But more complex…

[Montemerlo et al., 2003]

FastSLAM Problems

- How to determine the sample size?
- Particle deprivation, especially when closing (multiple) loops





FastSLAM Summary

- Particle filter-based SLAM
- Rao-Blackwellization: model the robot's path by sampling and compute the landmarks given the poses
- Allow for per-particle data association
- FastSLAM 1.0 and 2.0 differ in the proposal distribution
- Complexity $\mathcal{O}(N \log M)$

FastSLAM Results

- Scales well (1 million+ features)
- Robust to ambiguities in the data association
- Advantages compared to the classical EKF approach (especially with non-linearities)

Literature

FastSLAM

- Thrun et al.: “Probabilistic Robotics”, Chapter 13.1-13.3 + 13.8 (see errata!)
- Montemerlo, Thrun, Kollar, Wegbreit: FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem, 2002
- Montemerlo and Thrun: Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM, 2003