FastSLAM – Feature based SLAM using Particle Filters

Nived Chebrolu

Slides adapted from Maren Bennewitz and Cyrill Stachniss at Uni. Bonn.

Recap: SLAM So Far

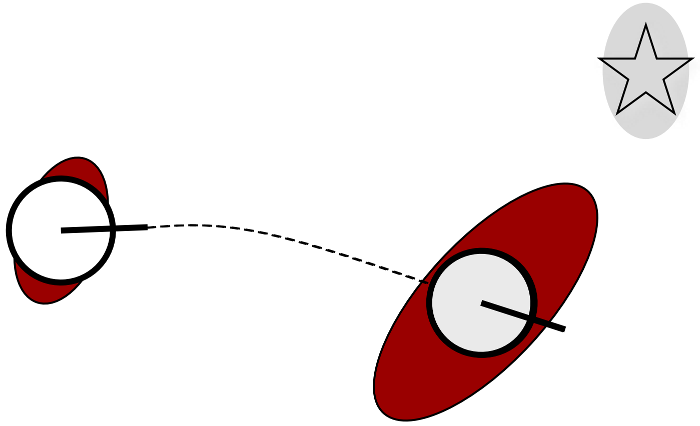
- Application of the **EKF** to **feature-based SLAM**
- Estimate **robot pose** and **landmark locations**
- Known data association: correspondences between observed landmarks and landmarks in the map are given
- State space:

$$x_t = (\underbrace{x, y, \theta}_{\text{robot's pose}}, \underbrace{m_{1,x}, m_{1,y}}_{\text{landmark 1}}, \ldots, \underbrace{m_{n,x}, m_{n,y}}_{\text{landmark n}})^T$$

Recap:
EKF SLAM: State Representation

$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \dots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \dots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \dots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

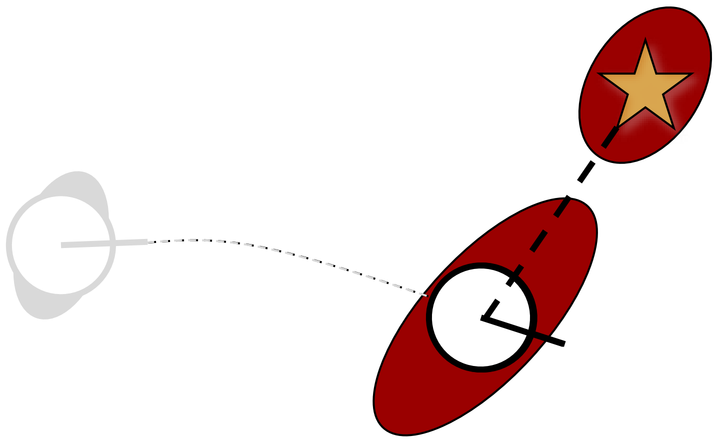
Recap:
EKF SLAM: State Prediction



updated entries based
on motion update

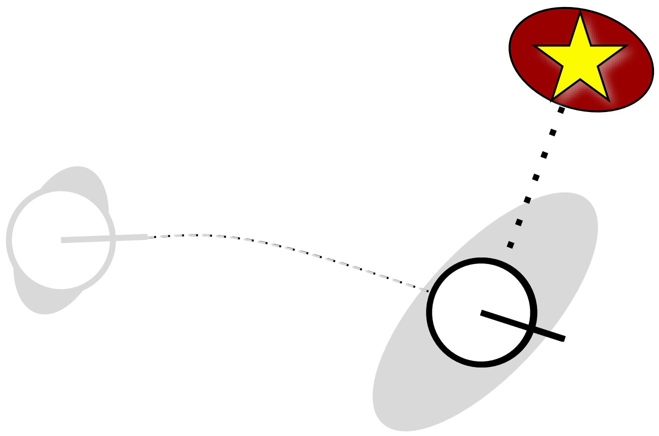
$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

Recap: EKF SLAM: Measurement Prediction



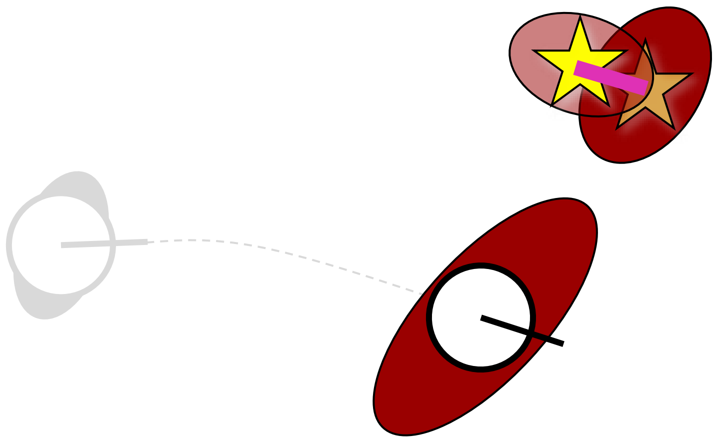
$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

Recap:
EKF SLAM: Obtained Measurement



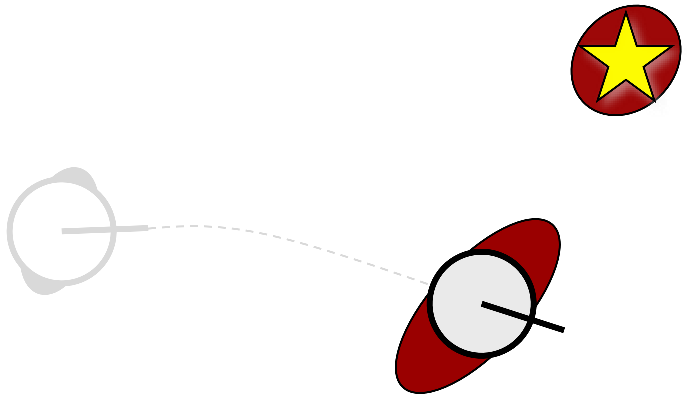
$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

Recap: EKF SLAM: Data Association and Difference to Predicted Observation



$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

Recap: EKF SLAM: Update Step



all entries are updated based
on re-observed landmark

$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

Recap: Drawbacks of EKF SLAM

- Can only deal with a single mode
- Successful only in medium-scale scenes
- Computationally intractable for large maps

Recap: Drawbacks of EKF SLAM

- Can only deal with a single mode
- Successful only in medium-scale scenes
- Computationally intractable for large maps

- **Today**: Application of the **particle filter** principle to solve the SLAM problem

Particle Representation for SLAM

- A set of weighted samples

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

- Each sample is a hypothesis about the state, i.e., robot trajectory and landmarks

- For feature-based SLAM:

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

Particle Filter Algorithm

1. Sample the particles using the proposal distribution

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

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

Dimensionality Problem

- PF are effective in low-dimensional spaces
- Number of particles needed to represent the posterior grows **exponentially** with the **dimension of the state space**

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

high-dimensional!

Dependencies

- Exploit dependencies between the dimensions of the state space
- **Map depends on the robot poses** from which measurements were obtained
- Use this dependency to solve the state estimation problem efficiently

Dependencies Between the Different Dimensions of the State Space

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

If We Know the Robot Poses, Mapping is Easy!

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

Key Idea

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

- Use the particle set to model the robot's path
- For **each sample**, compute an **individual** map of landmarks
- In the following, derivation of the posterior

Rao-Blackwellization

- Use factorization to exploit dependencies between variables:

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

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

Rao-Blackwellization for SLAM

Factorization of the SLAM posterior

poses map observations & movements

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

[Murphy, 1999]

Rao-Blackwellization for SLAM

Factorization of the SLAM posterior

poses map observations & movements

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

Application of the product rule given background information

[Murphy, 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: localization problem, use particle filter

map posterior

[Murphy, 1999]

Rao-Blackwellization for SLAM

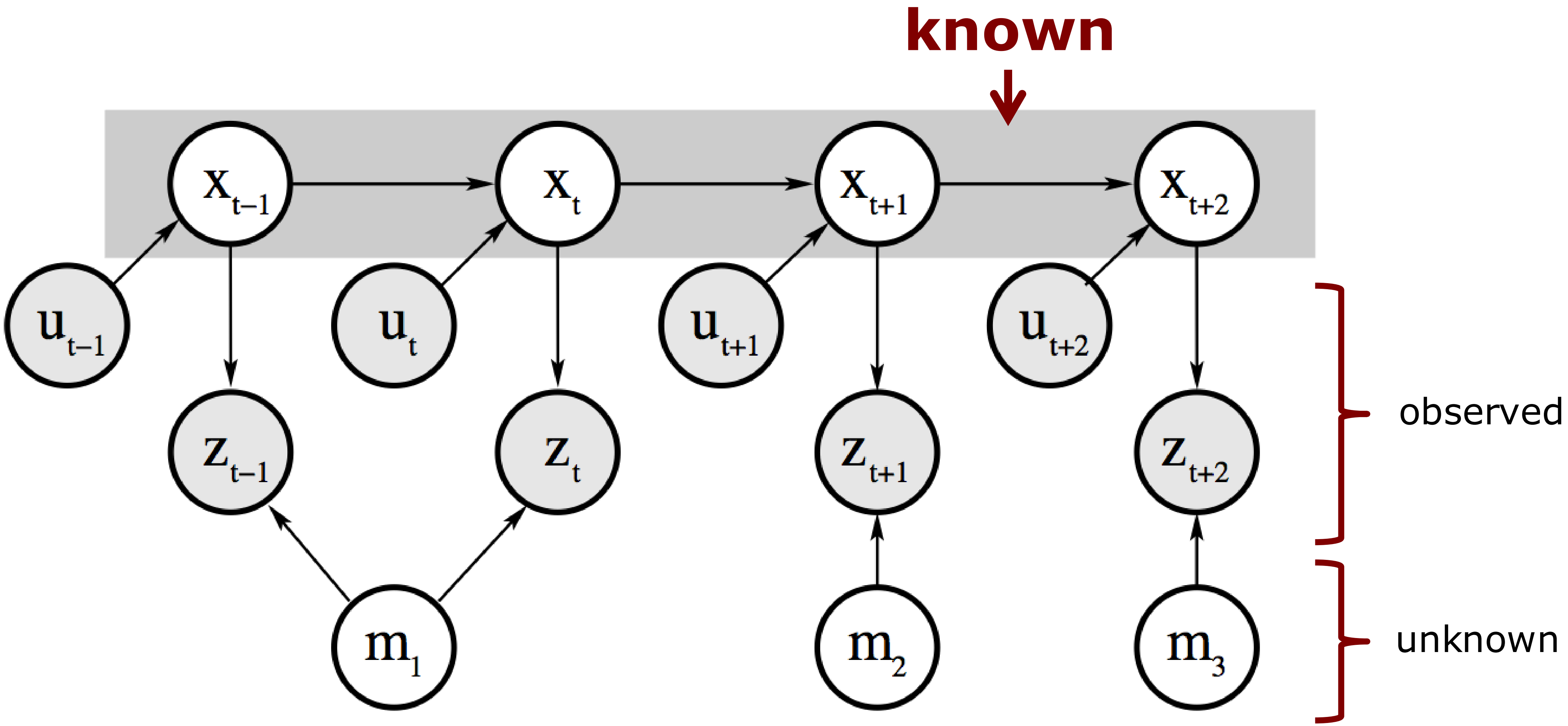
Factorization of the SLAM posterior

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

How to compute this term efficiently?

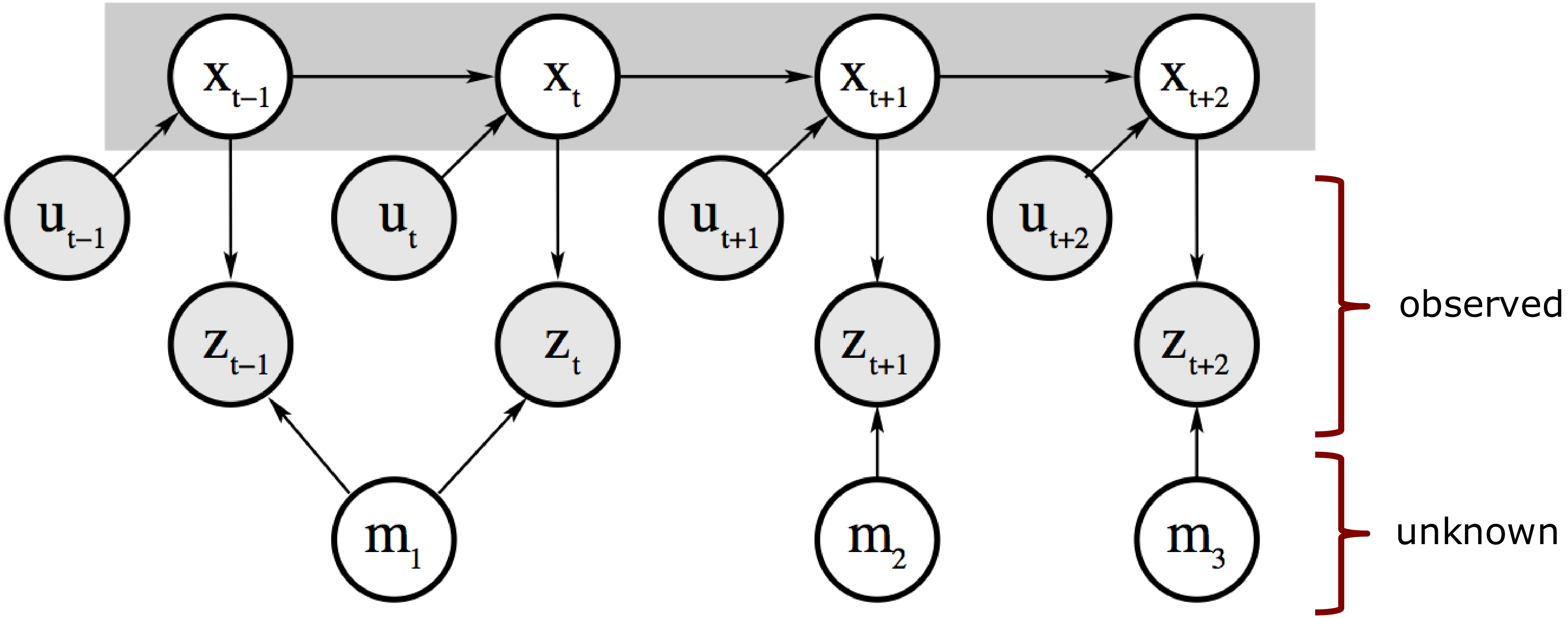
[Murphy, 1999]

Revisiting the Graphical Model



Courtesy: Thrun, Burgard, Fox

Revisiting the Graphical Model



Landmarks are conditionally independent given the robot poses

Observation

- If **robot poses** are **known:**
  Knowledge about the exact location of one landmark will give **no additional** information about the other landmarks
- In contrast to EKF SLAM, where landmarks are correlated

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 robot poses

Rao-Blackwellization for SLAM

Factorization of the SLAM posterior

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

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

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

[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!

[Montemerlo et al., 2002]

Rao-Blackwellization for SLAM

Factorization of the SLAM posterior

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

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

$$\underbrace{p(x_{0:t} \mid z_{1:t}, u_{1:t})}_{\text{particle filter similar to MCL}} \prod_{i=1}^{M} \underbrace{p(m_i \mid x_{0:t}, z_{1:t})}_{\text{2-dimensional EKFs!}}$$

[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 represents a path hypothesis

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

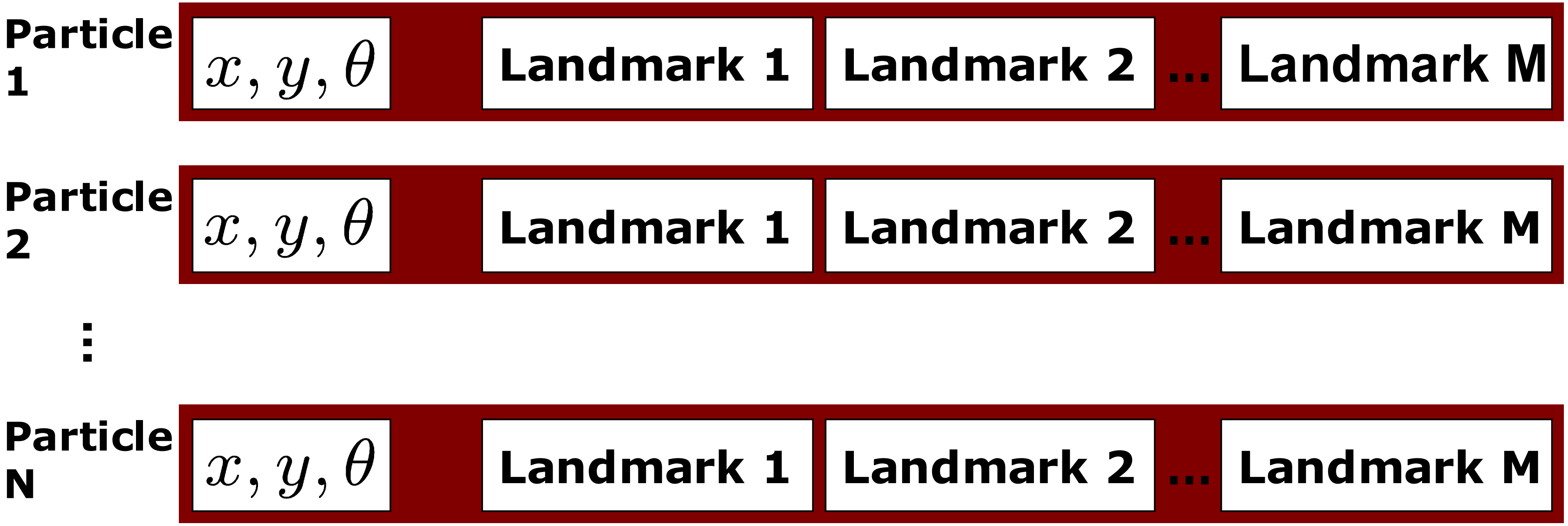
starting location

pose hypothesis
at time t=1

- But: Past poses of a sample are not revised
- No need to maintain past poses in the sample set, just keep track of current pose

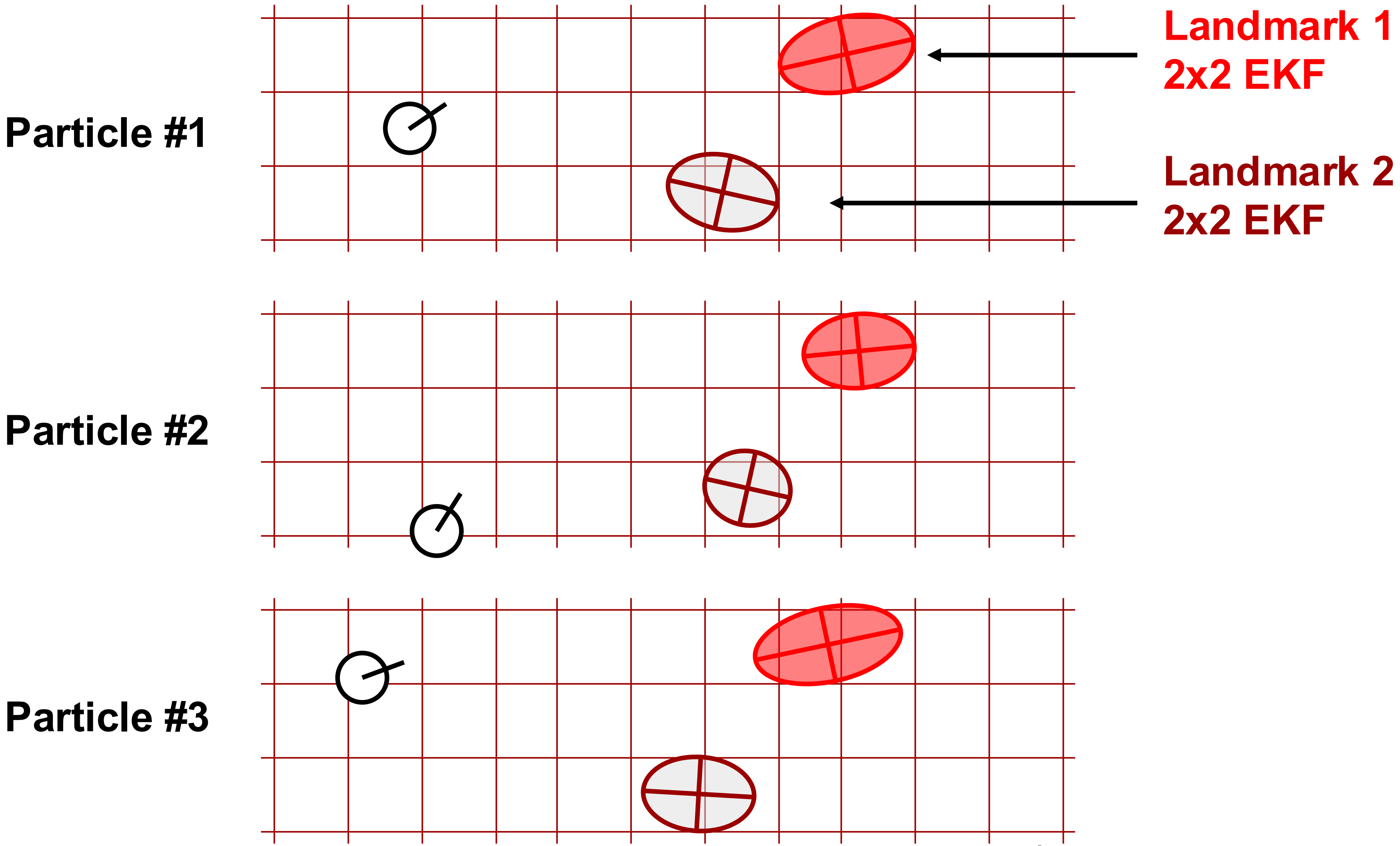
FastSLAM

- Each landmark is represented by a 2x2 EKF
- Thus, each particle has to maintain M individual EKFs



[Montemerlo et al., 2002]

FastSLAM – Example Illustration



Courtesy: M. Montemerlo

Key Steps of FastSLAM

- Sample a new pose for each particle

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

Key Steps of FastSLAM

- Sample a new pose for each particle

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

- Update belief of observed landmarks for each particle: **individual** data association, standard EKF update rule

Key Steps of FastSLAM

- Sample a new pose for each particle

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

- Update belief of observed landmarks for each particle: **individual** data association, standard EKF update rule

- Compute the particle weights

measured

expected

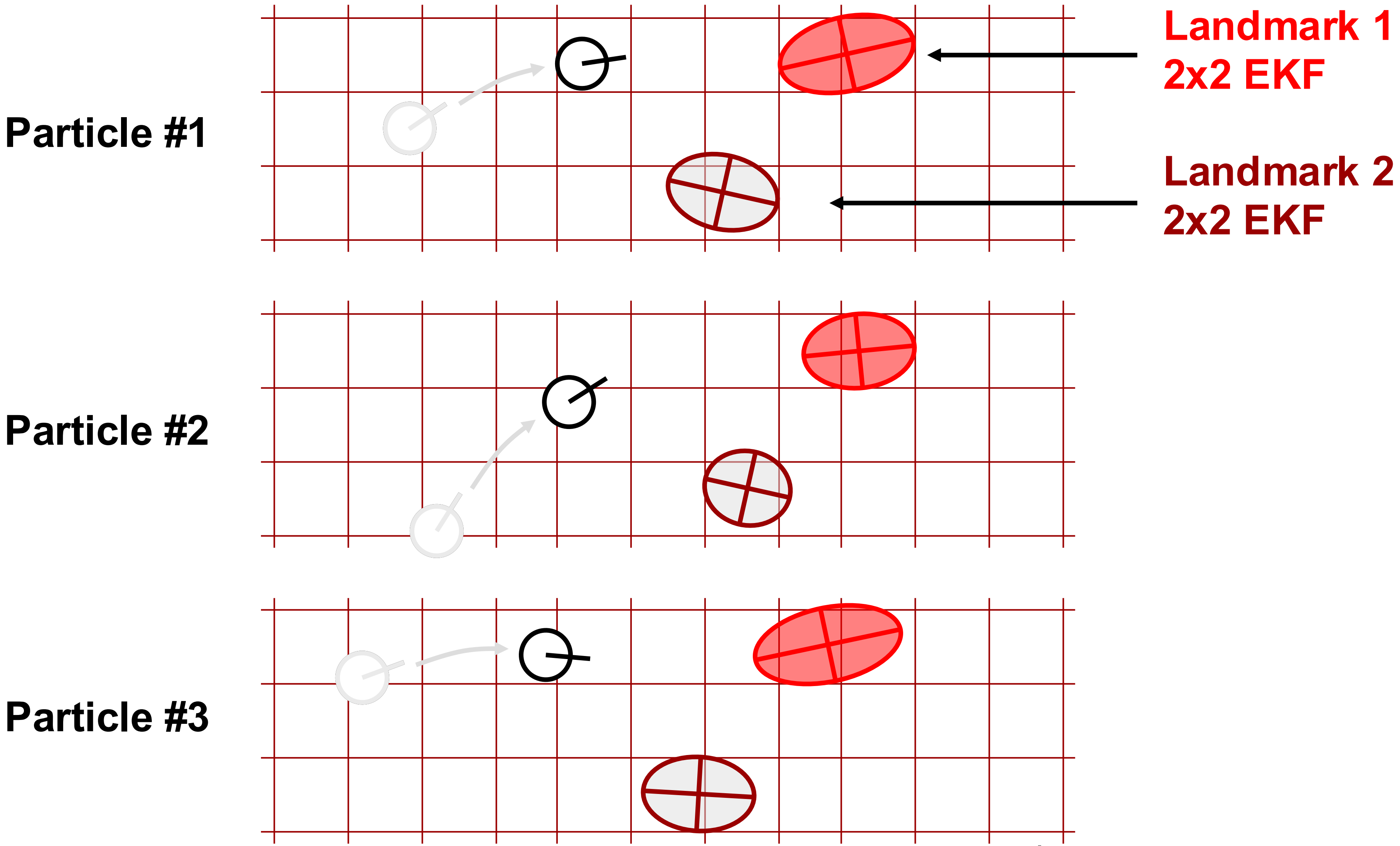
$$w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \ \exp\left\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} \ (z_t - \hat{z}^{[k]})\right\}$$

measurement covariance

(includes landmark position uncertainty as well as measurement noise)

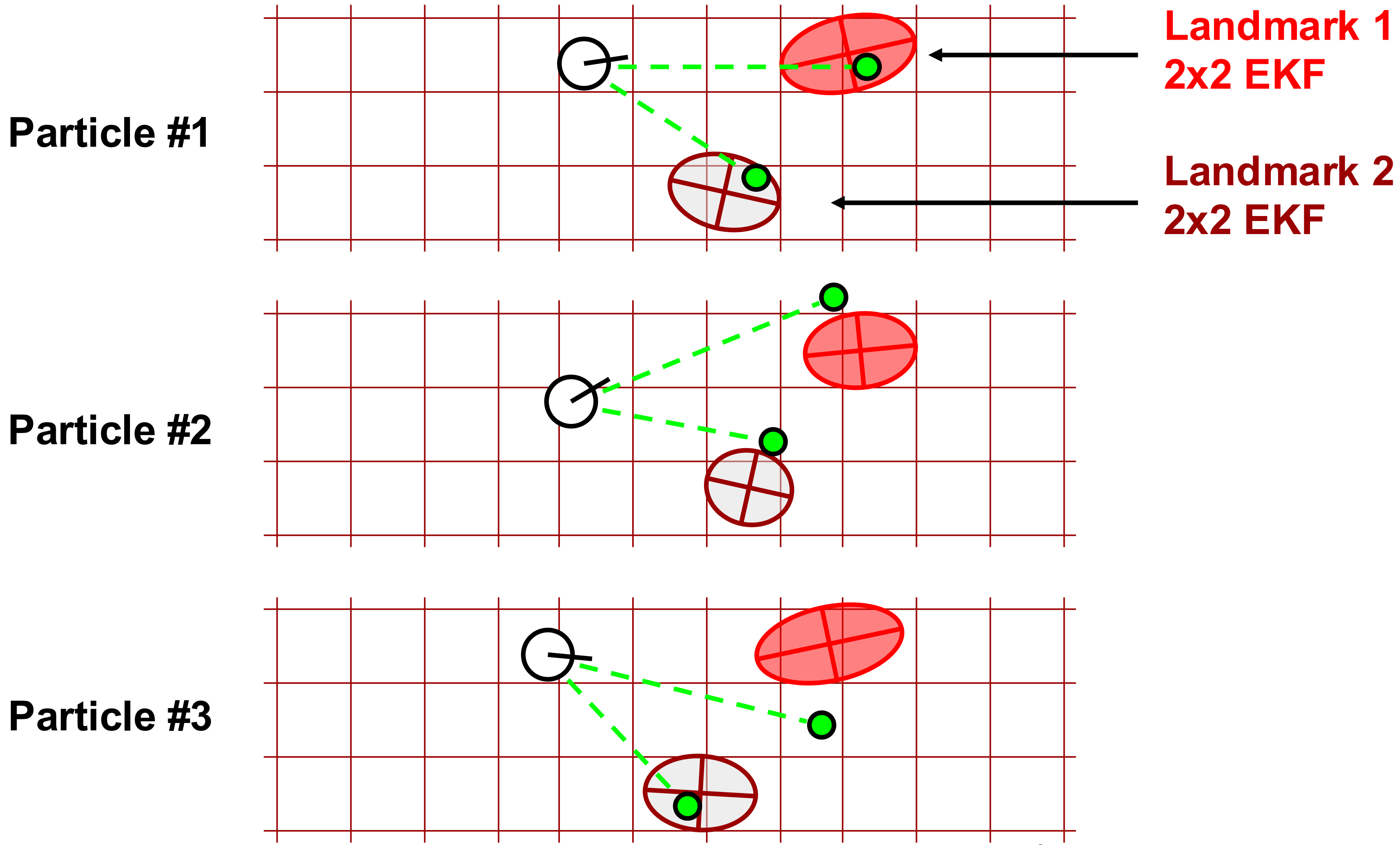
- Resample

FastSLAM – Motion Update



Courtesy: M. Montemerlo

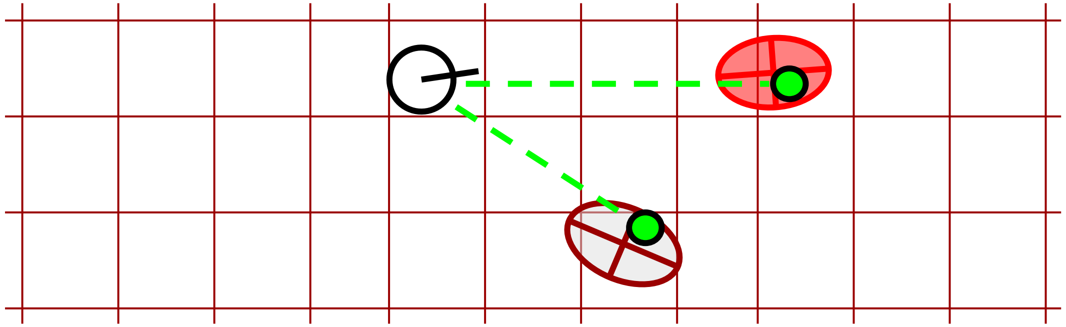
FastSLAM – Sensor Update



Courtesy: M. Montemerlo

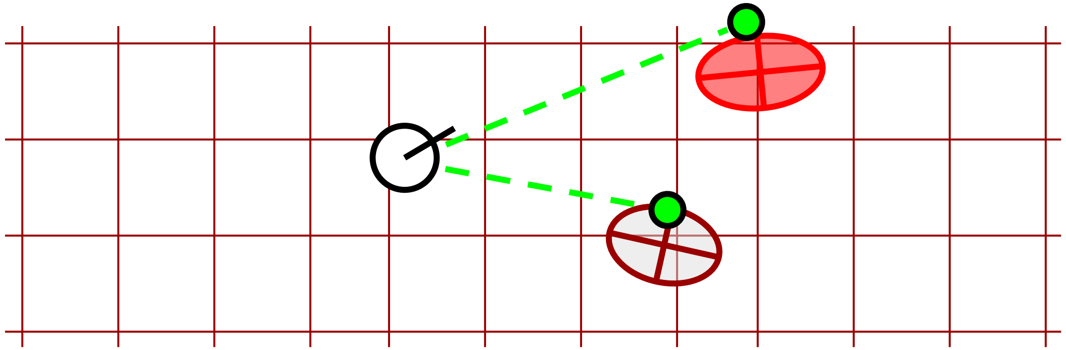
FastSLAM – Sensor Update

Particle #1



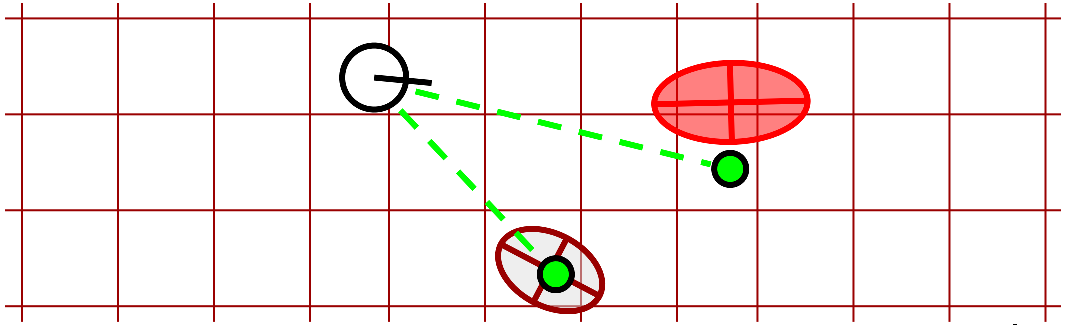
Update map of particle 1

Particle #2



Update map of particle 2

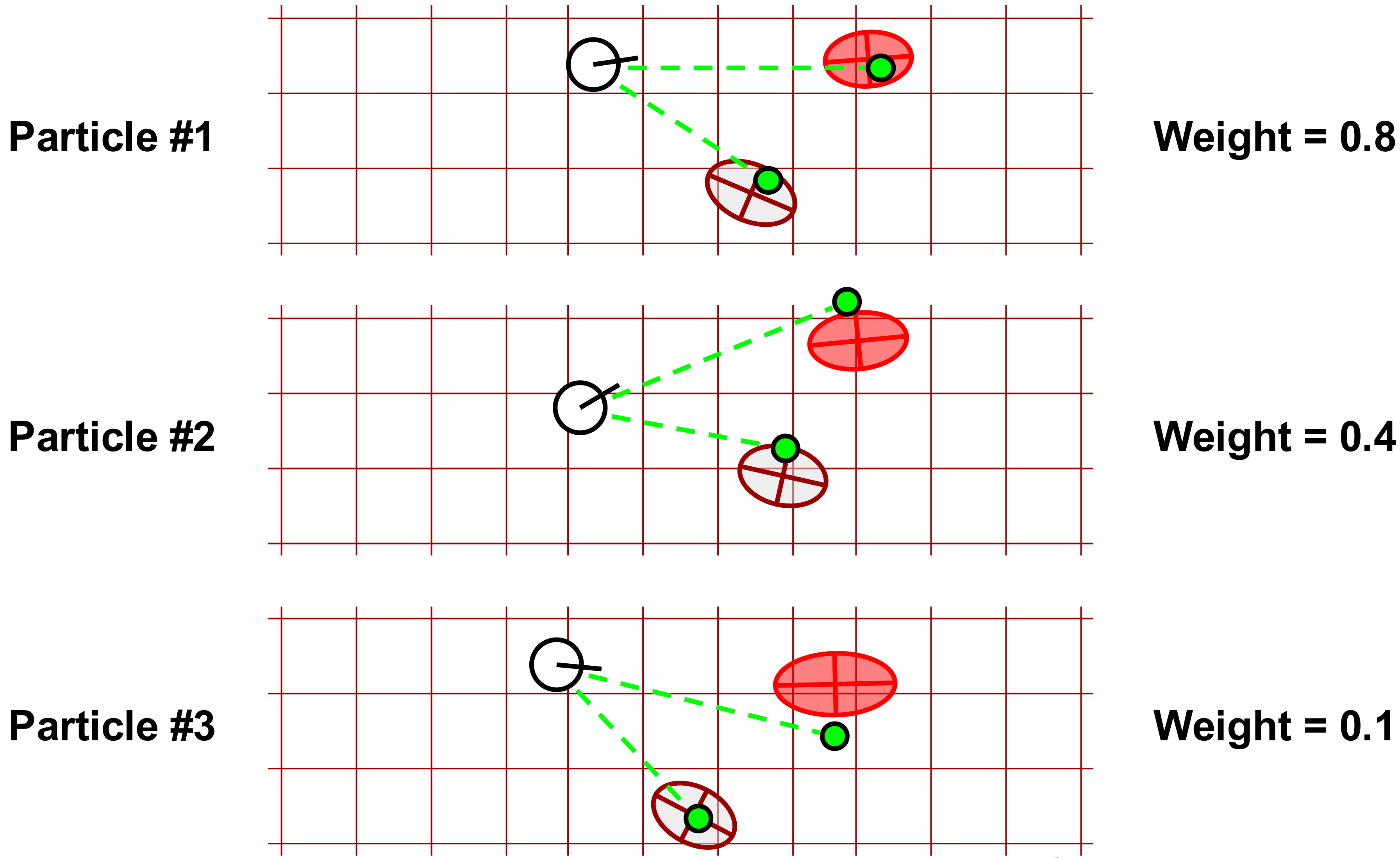
Particle #3



Update map of particle 3

Courtesy: M. Montemerlo

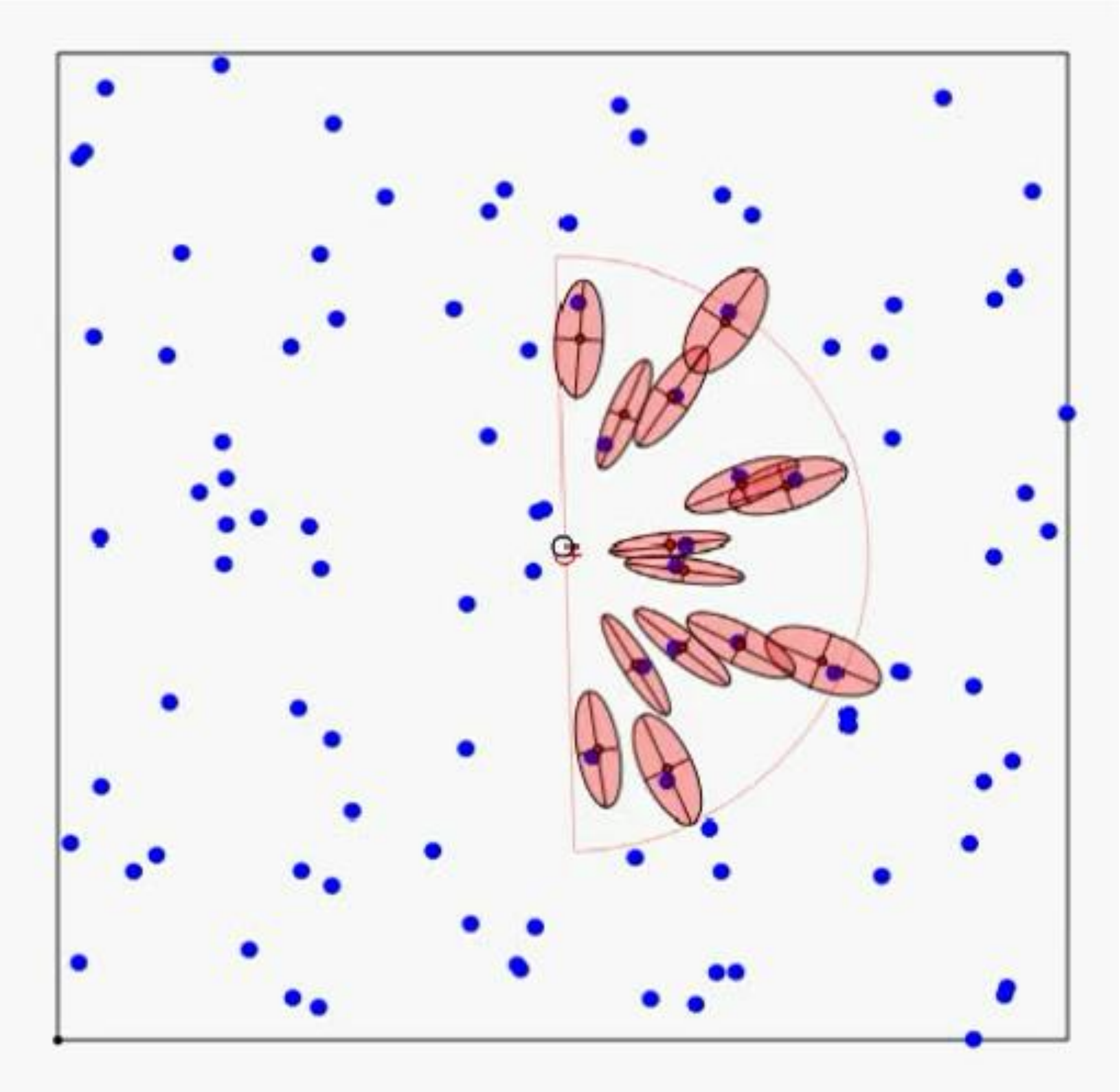
FastSLAM – Sensor Update



Courtesy: M. Montemerlo

FastSLAM in Action

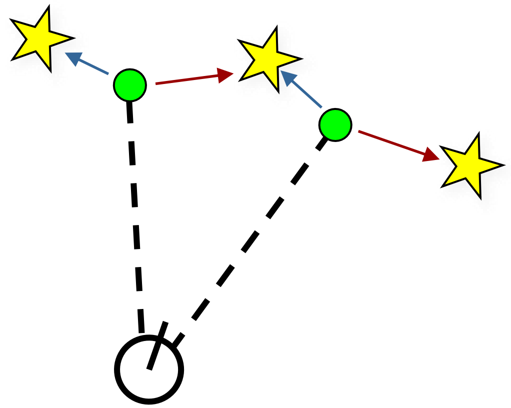
shown is the result for the most likely particle



Courtesy: M. Montemerlo

Data Association Problem

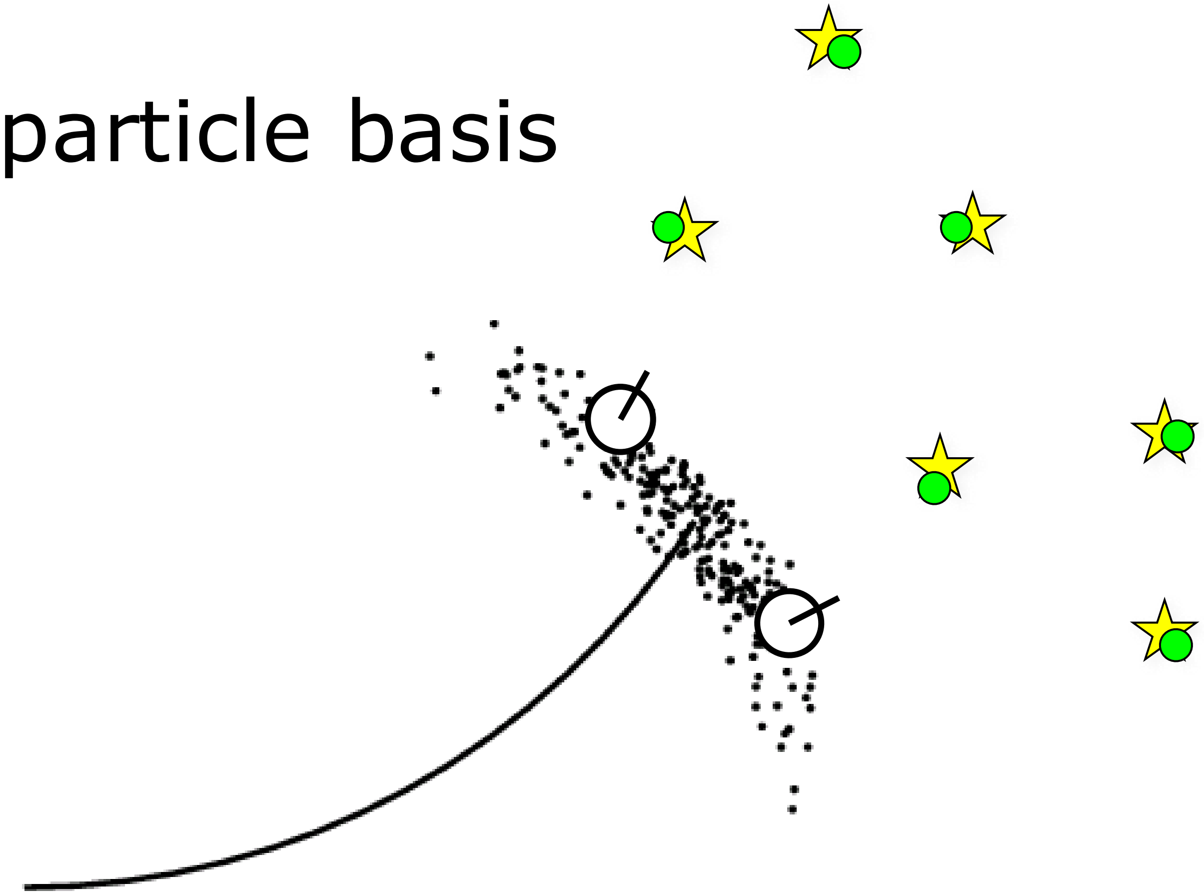
- Which observation belongs to which landmark?



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

Particles Support Multi-Hypotheses Data Association

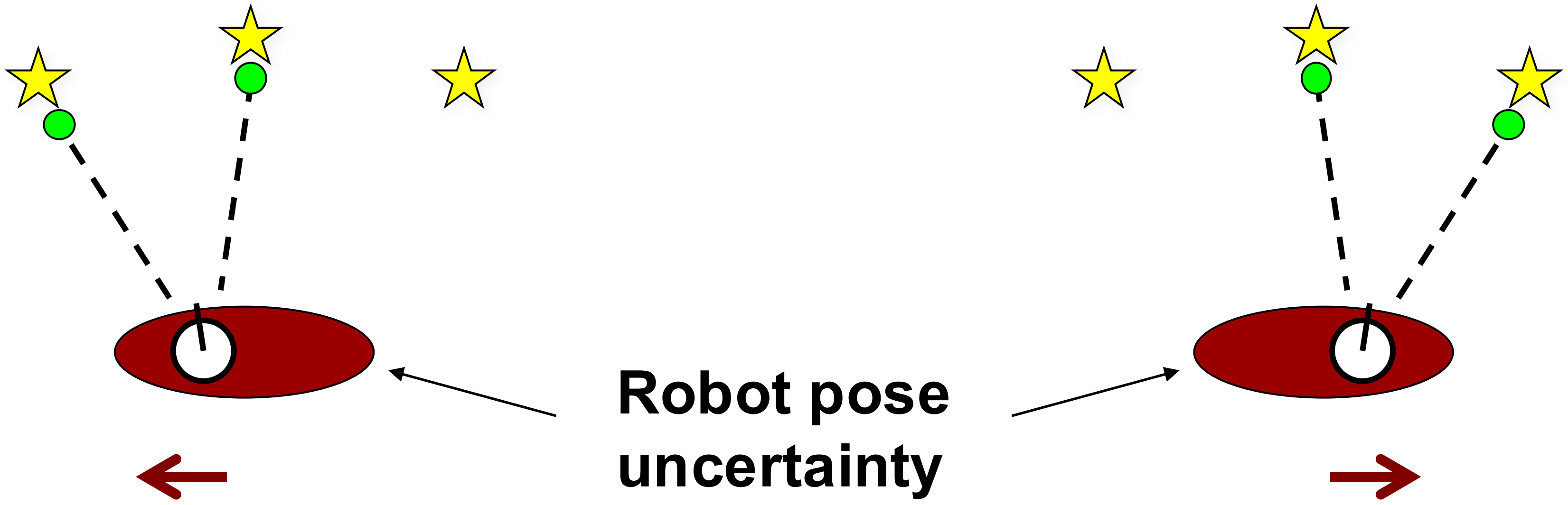
Decisions on per-particle basis



Courtesy: M. Montemerlo

Recap: Data Association in EKF SLAM

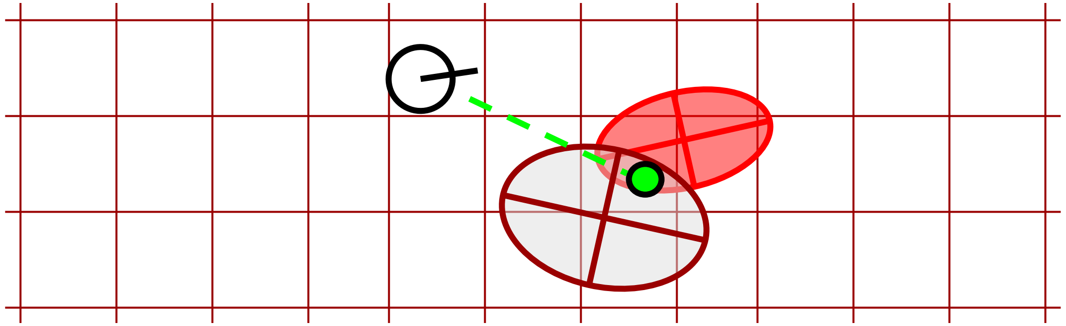
The best data association is not obvious even within the error ellipse of the pose estimate



Data Association (DA)

- **EKF SLAM**: We have to decide on one DA, since we keep track of only one mode
- **FastSLAM**: We just need some particles doing the correct data association
- As long as a subset of the particles does the correct DA, DA errors are not as fatal as in EKF approaches

Per-Particle Data Association



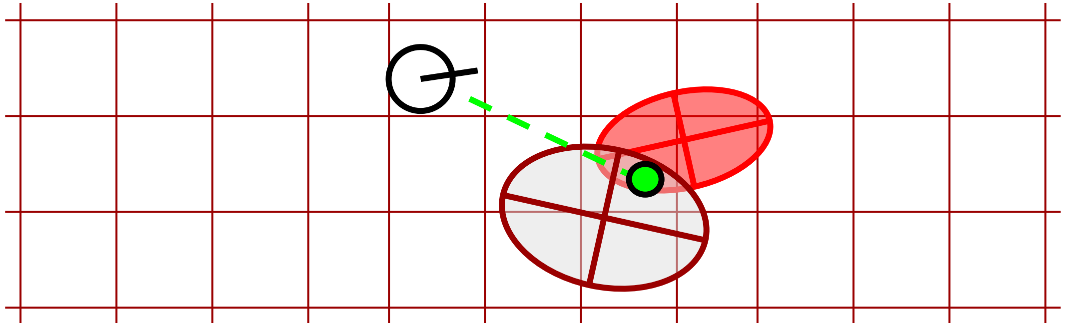
Is the observation generated by the **red** or by the **brown** landmark?

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

Calculation of the observation likelihood, see Slide 34

Courtesy: M. Montemerlo

Per-Particle Data Association



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

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

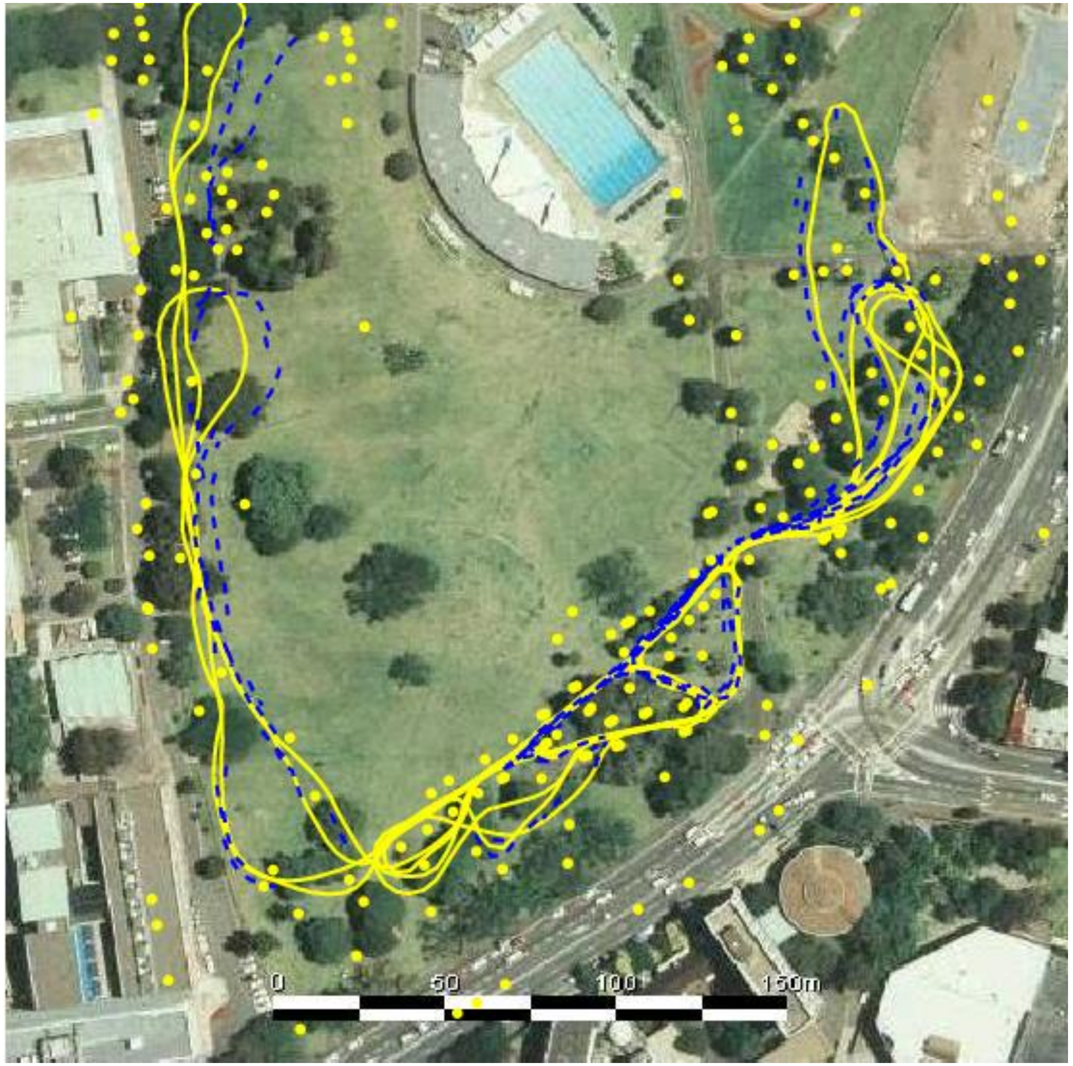
- Take most likely assignment for each particle individually (in case of several observed landmarks: maximum joint likelihood)
- If likelihood of an assignment is too low, initialize new landmark

Per-Particle Data Association

- Multi-modal belief supports **multi-hypotheses data association**
- Simple but effective data association is possible and sufficient
- **Big advantage of FastSLAM over EKF**

Results – Victoria Park

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



Blue = GPS
Yellow = FastSLAM

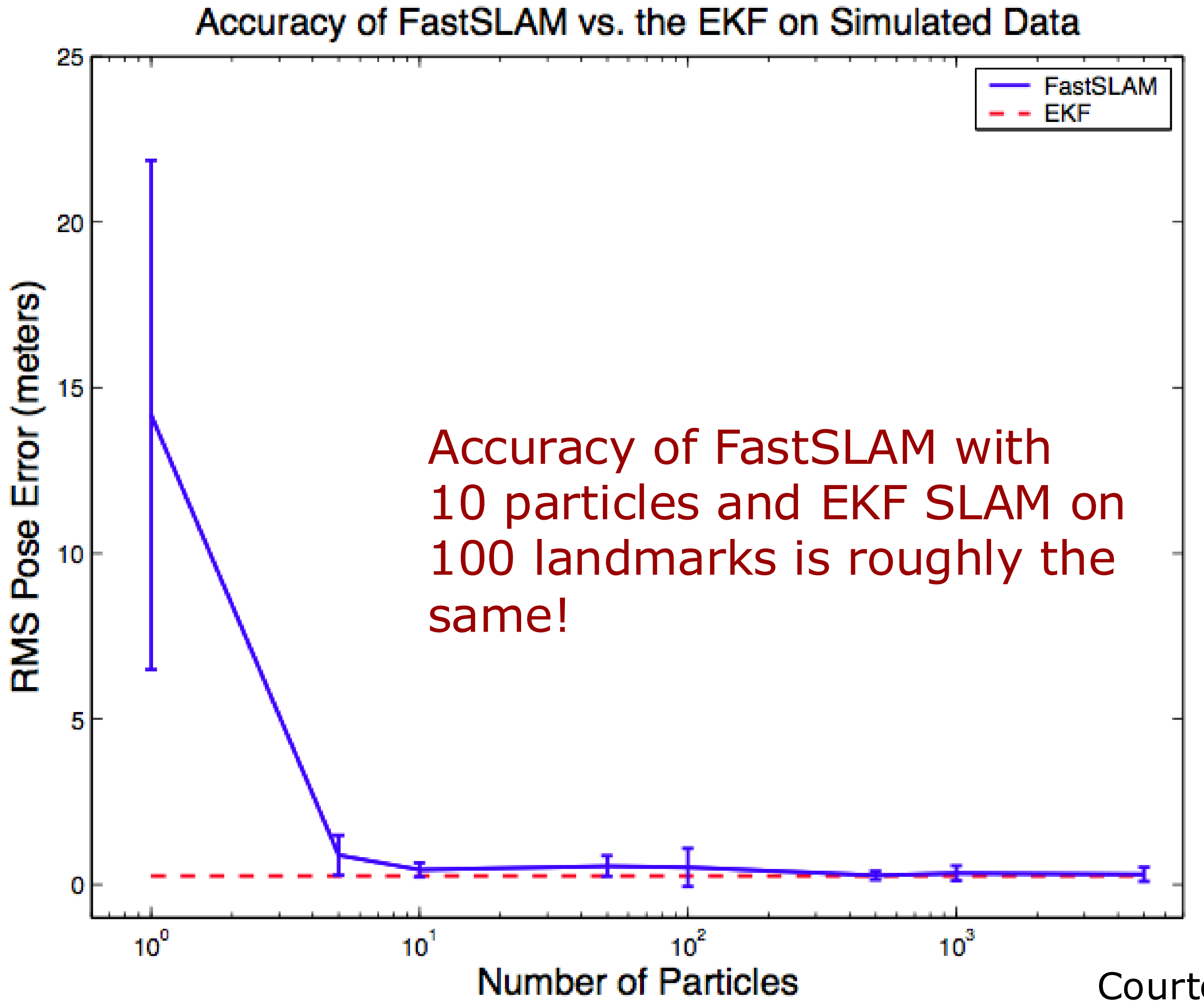
Courtesy: M. Montemerlo

Results – Victoria Park (Video)

shown is the result for the most likely particle

Courtesy: M. Montemerlo

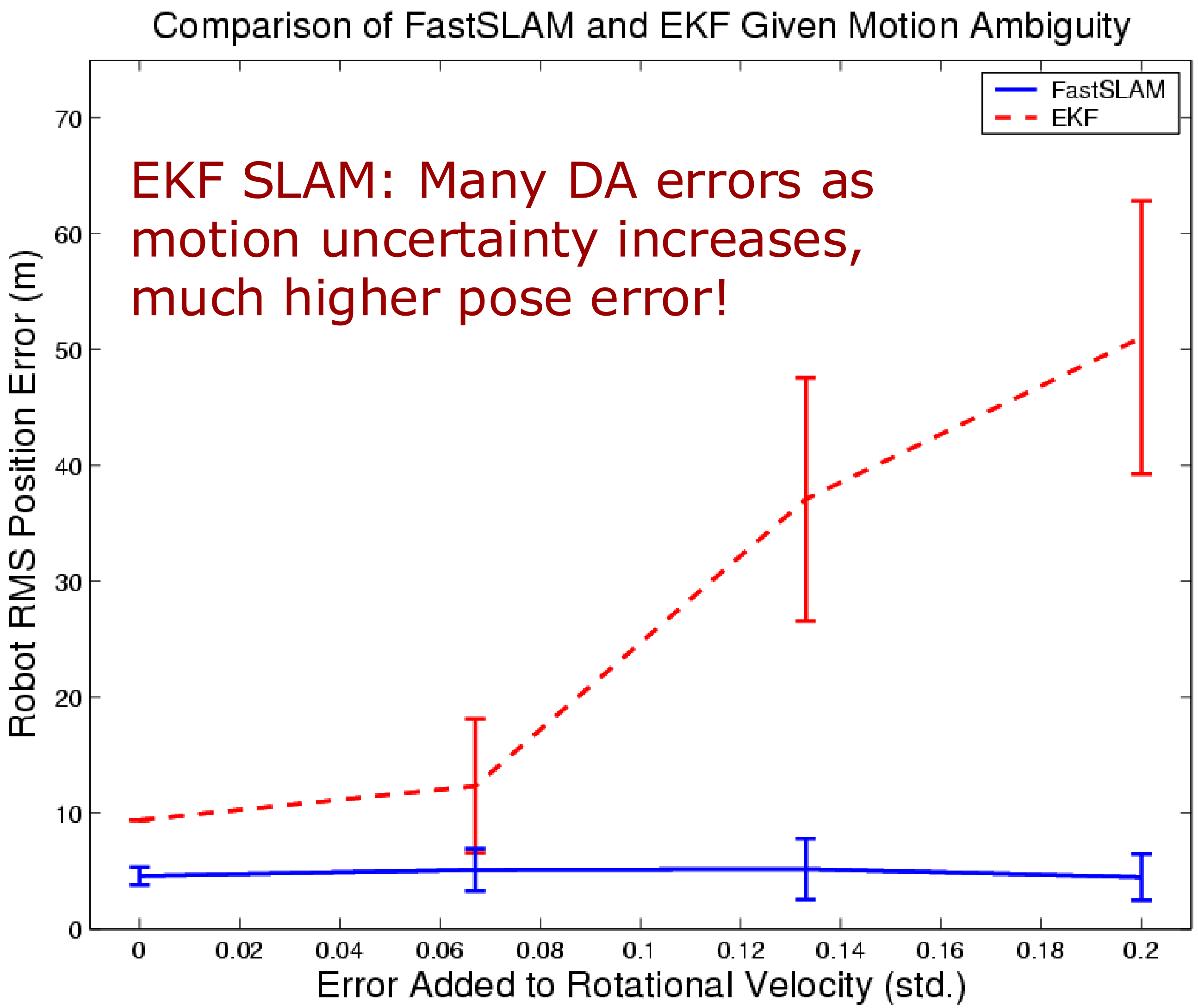
Pose Error wrt. #Particles



Accuracy of FastSLAM with 10 particles and EKF SLAM on 100 landmarks is roughly the same!

Courtesy: M. Montemerlo

Pose Error wrt. Motion Uncertainty



Courtesy: M. Montemerlo

FastSLAM Complexity – Simple Implementation

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

N = Number of particles
M = Number of landmarks

$\mathcal{O}(NM)$

Summary: Feature-Based FastSLAM

- Feature-based SLAM with particle filter
- Estimates the **landmark positions for each particle individually** given particle pose
- **Per-particle data association** and no robot pose uncertainty in the data association
- **Robust to ambiguities** in the data association
- Scales well (1 million+ features)
- Several advantages compared to EKF SLAM

Literature

FastSLAM

- Thrun et al.: “Probabilistic Robotics”, Chapter 13