

Introduction to Mobile Robotics

SLAM – Landmark-based FastSLAM

Kshitij Sirohi



Partial slide courtesy of Mike Montemerlo

The SLAM Problem

- SLAM stands for simultaneous localization and mapping
- The task of building a map while estimating the pose of the robot relative to this map
- Why is SLAM hard?
Chicken-or-egg problem:
 - A map is needed to localize the robot
 - A pose estimate is needed to build a map

The SLAM Problem

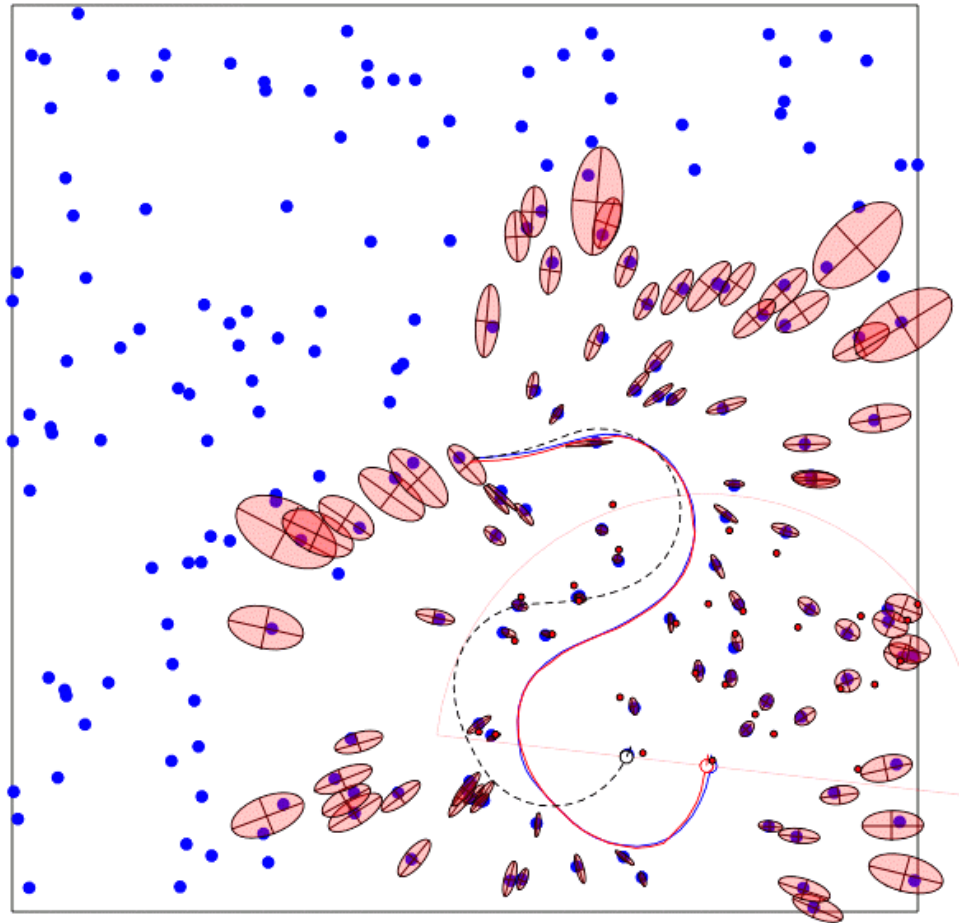
A robot moving through an unknown, static environment

Given:

- The robot's controls
- Observations of nearby features

Estimate:

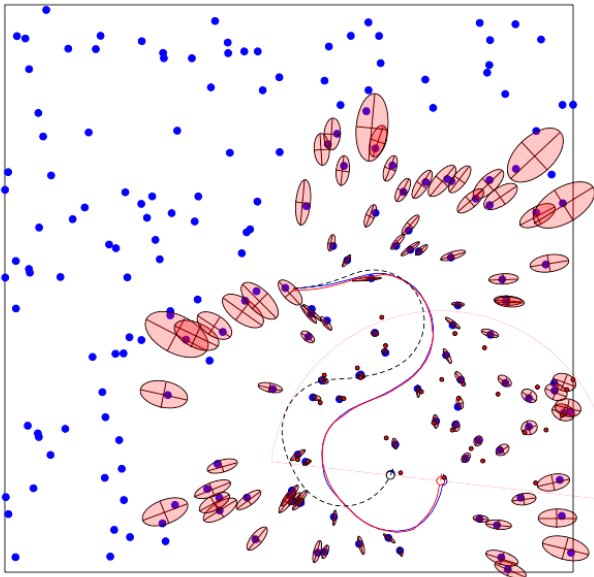
- Map of features
- Path of the robot



Map Representations

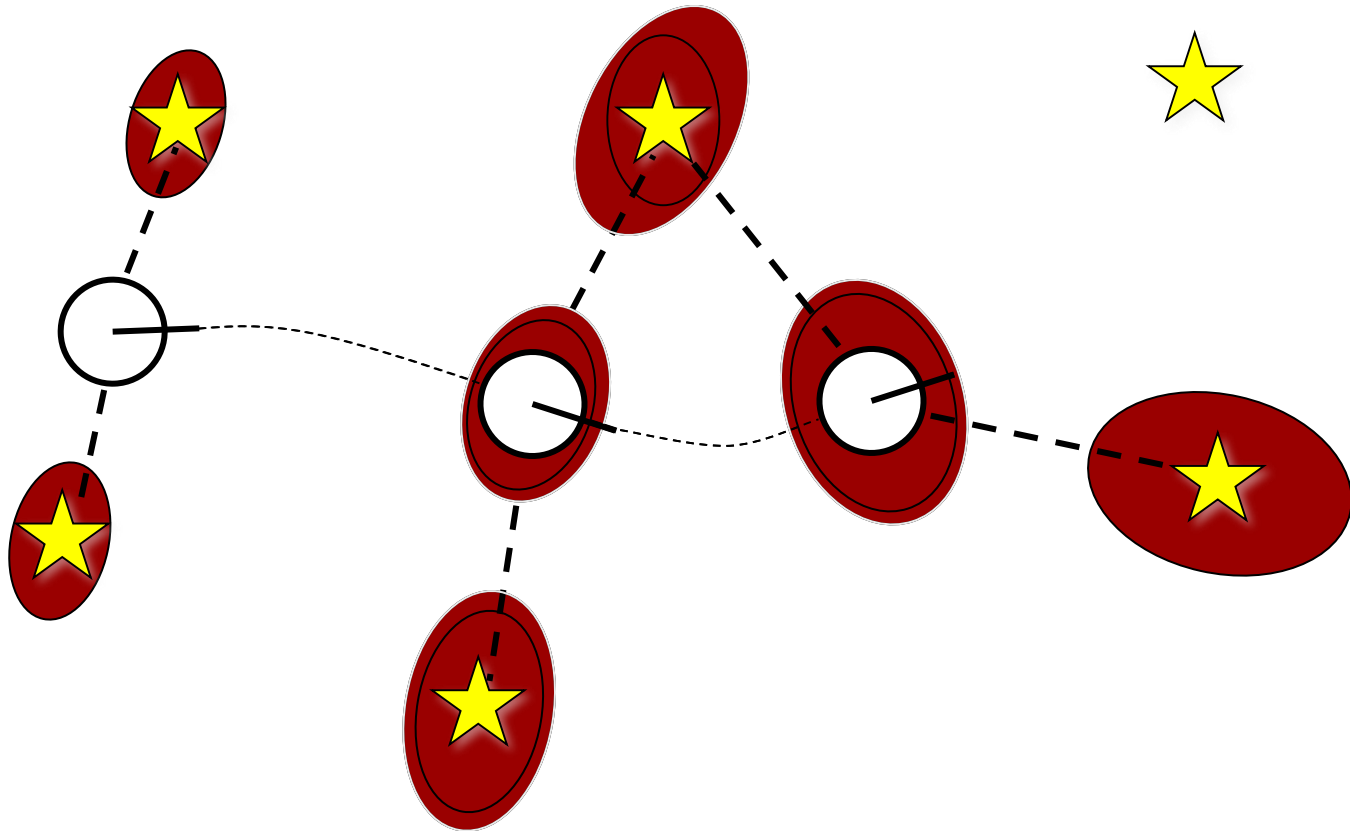
Typical models are:

- Feature maps
- Grid maps (occupancy or reflection probability maps)



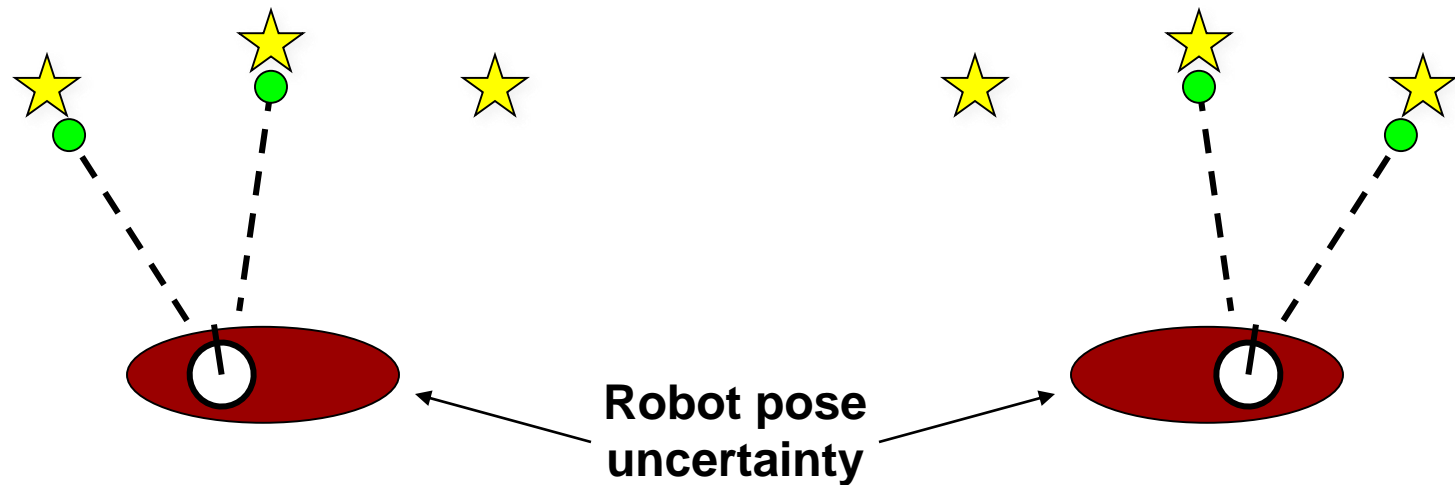
Why is SLAM a Hard Problem?

SLAM: robot path and map are both **unknown!**



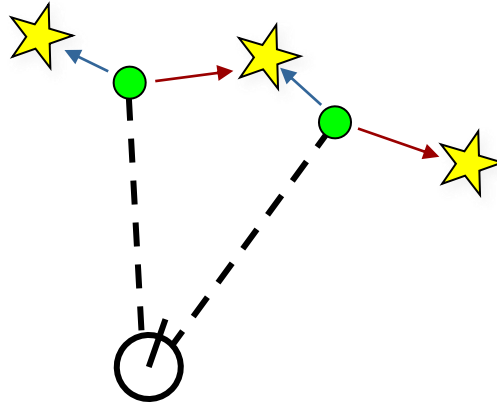
Robot path error correlates errors in the map

Why is SLAM a Hard Problem?



- In the real world, the mapping between observations and landmarks is unknown
- Picking wrong data associations can have catastrophic consequences
- Pose error correlates data associations

Data Association Problem



- A data association is an assignment of observations to landmarks
- In general there are more than $\binom{n}{m}$ (n observations, m landmarks) possible associations
- Also called “assignment problem”

Particle Filters

- Represent belief by random samples
- Estimation of non-Gaussian, nonlinear processes
- Sampling Importance Resampling (SIR) principle
 - Draw the new generation of particles
 - Assign an importance weight to each particle
 - Resample
- Typical application scenarios are tracking, localization, ...

Localization vs. SLAM

- A particle filter can be used to solve both problems
- Localization: state space $\langle x, y, \theta \rangle$
- SLAM: state space $\langle x, y, \theta, map \rangle$
 - for landmark maps = $\langle l_1, l_2, \dots, l_m \rangle$
 - for grid maps = $\langle c_{11}, c_{12}, \dots, c_{1n}, c_{21}, \dots, c_{nm} \rangle$
- **Problem:** The number of particles needed to represent a posterior grows exponentially with the dimension of the state space!

Dependencies


- Is there a dependency between certain dimensions of the state space?
- If so, can we use the dependency to solve the problem more efficiently?

Dependencies

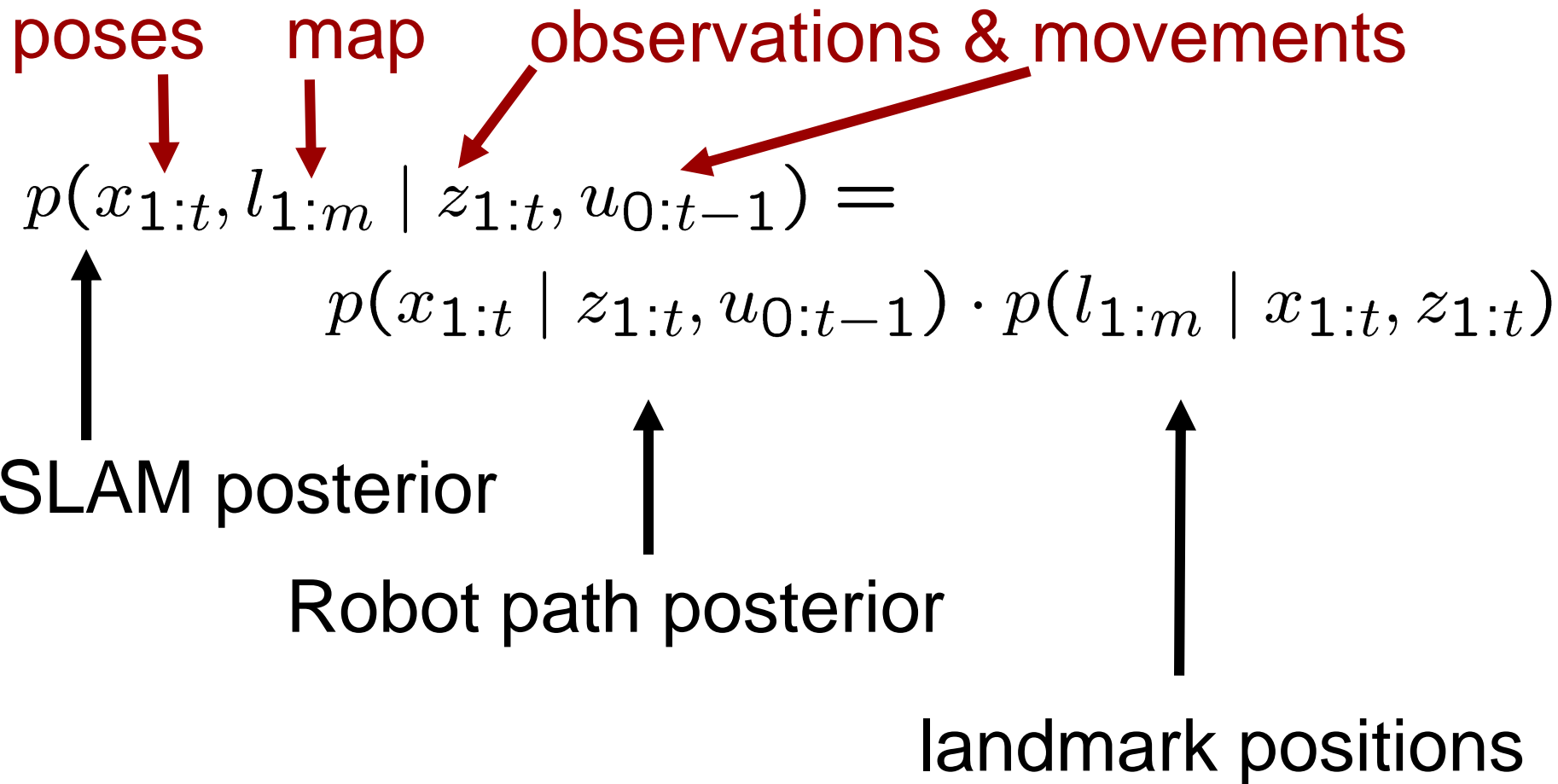
- Is there a dependency between certain dimensions of the state space?
- If so, can we use the dependency to solve the problem more efficiently?
- In the SLAM context
 - The map depends on the poses of the robot.
 - We know how to build a map given the position of the sensor is known.

Factored Posterior (Landmarks)

poses map observations & movements


$$p(x_{1:t}, l_{1:m} \mid z_{1:t}, u_{0:t-1}) =$$
$$p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot p(l_{1:m} \mid x_{1:t}, z_{1:t})$$

Factored Posterior (Landmarks)



Does this help to solve the problem?

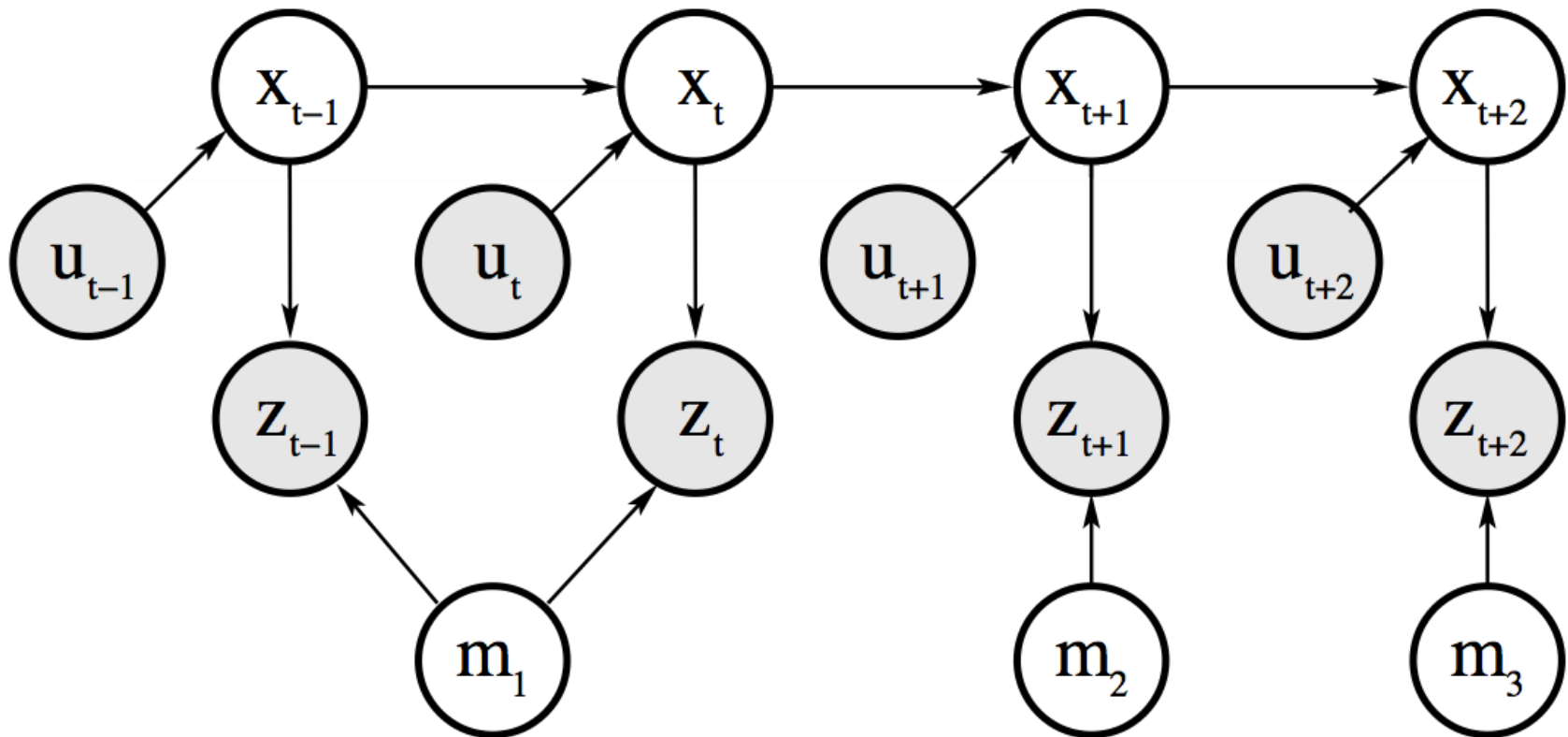
Rao-Blackwellization

- Factorization to exploit dependencies between variables:

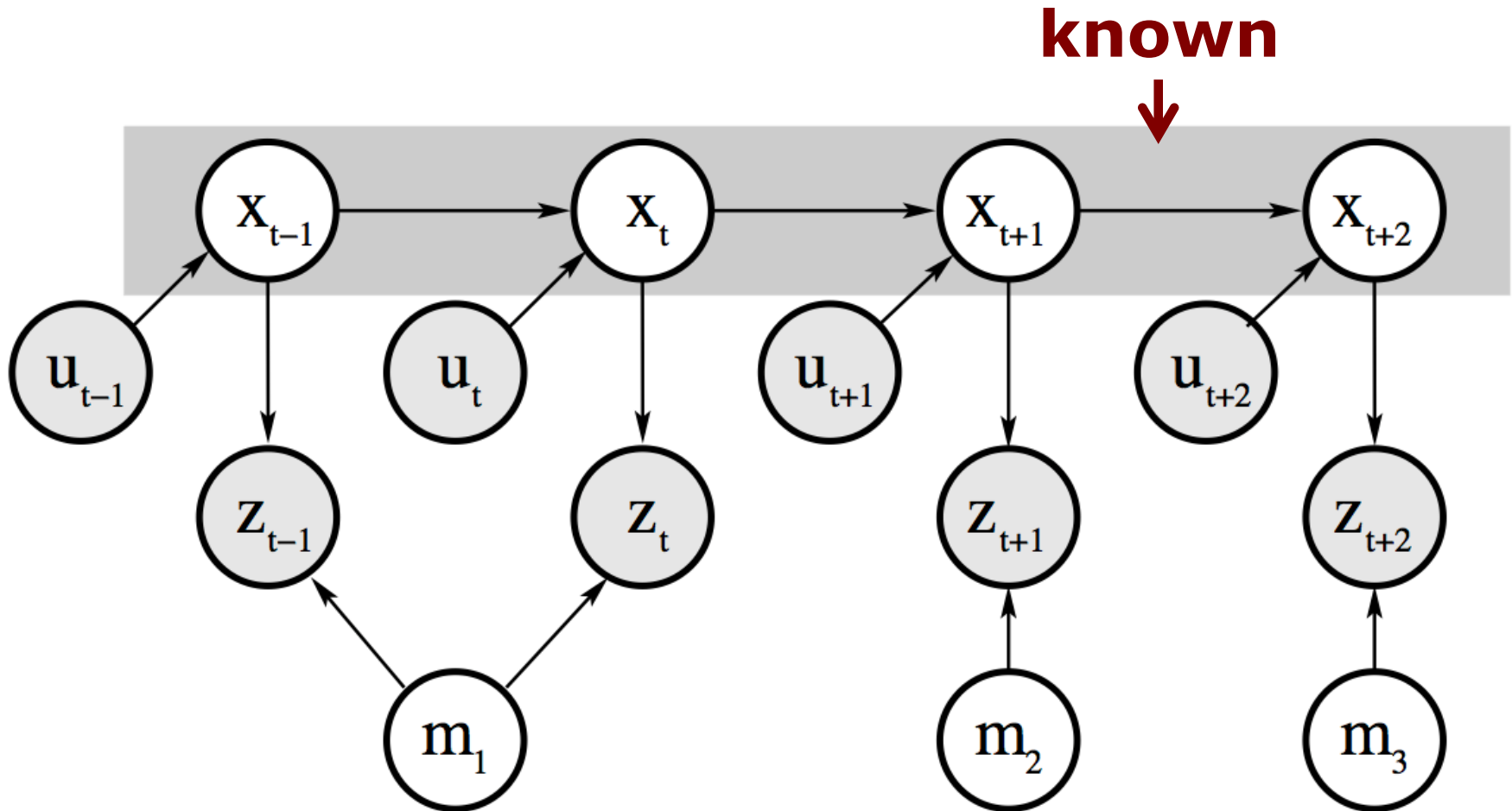
$$p(a, b) = p(a) \cdot p(b | a)$$

- If $p(b | a)$ can be computed in closed form, represent only $p(a)$ with samples and compute $p(b | a)$ for every sample
- It comes from the Rao-Blackwell theorem

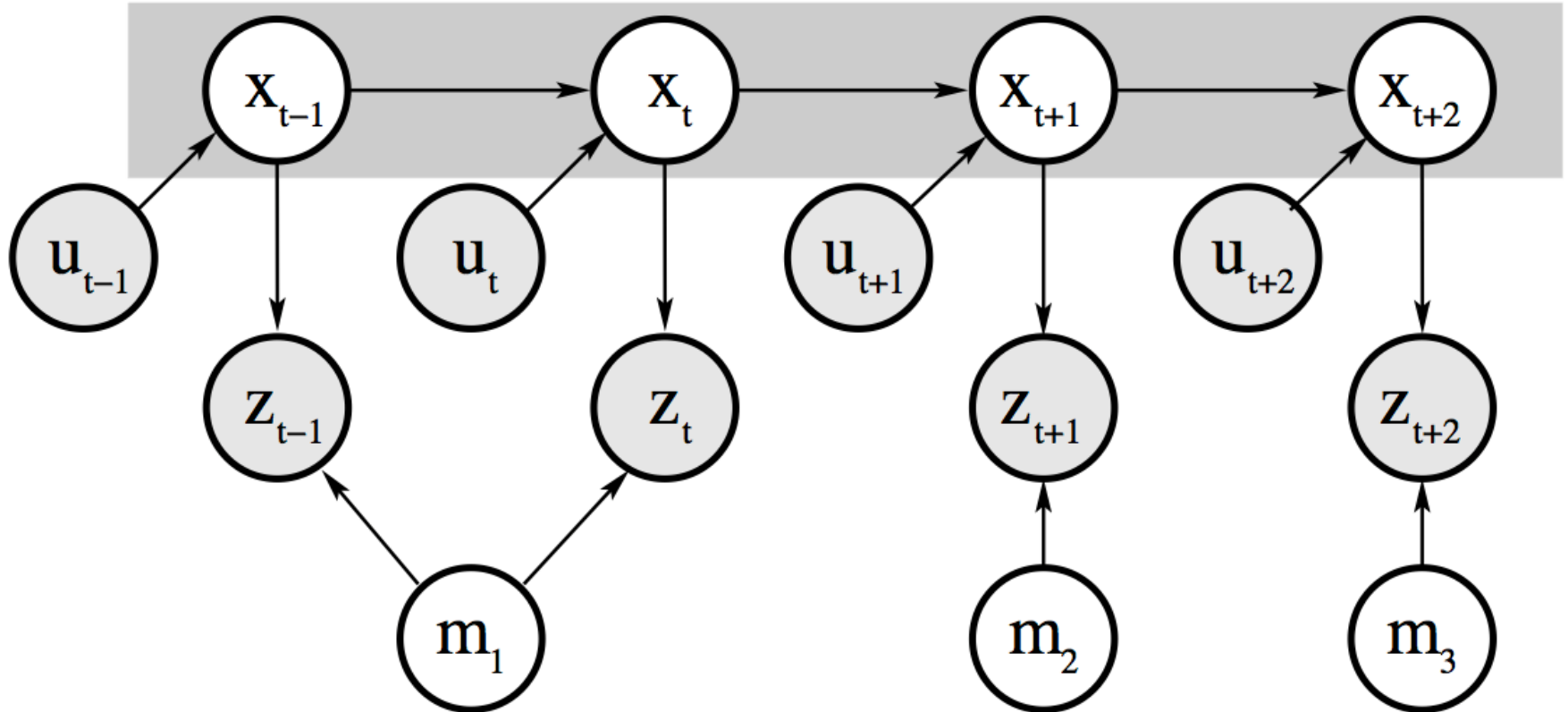
Revisit the Graphical Model



Revisit the Graphical Model



Landmarks are Conditionally Independent Given the Poses




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


Factored Posterior

$$\begin{aligned} & p(x_{1:t}, l_{1:m} \mid z_{1:t}, u_{0:t-1}) \\ &= p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot p(l_{1:m} \mid x_{1:t}, z_{1:t}) \\ &= p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot \prod_{i=1}^M p(l_i \mid x_{1:t}, z_{1:t}) \end{aligned}$$

Robot path posterior
(localization problem)



Conditionally
independent
landmark positions



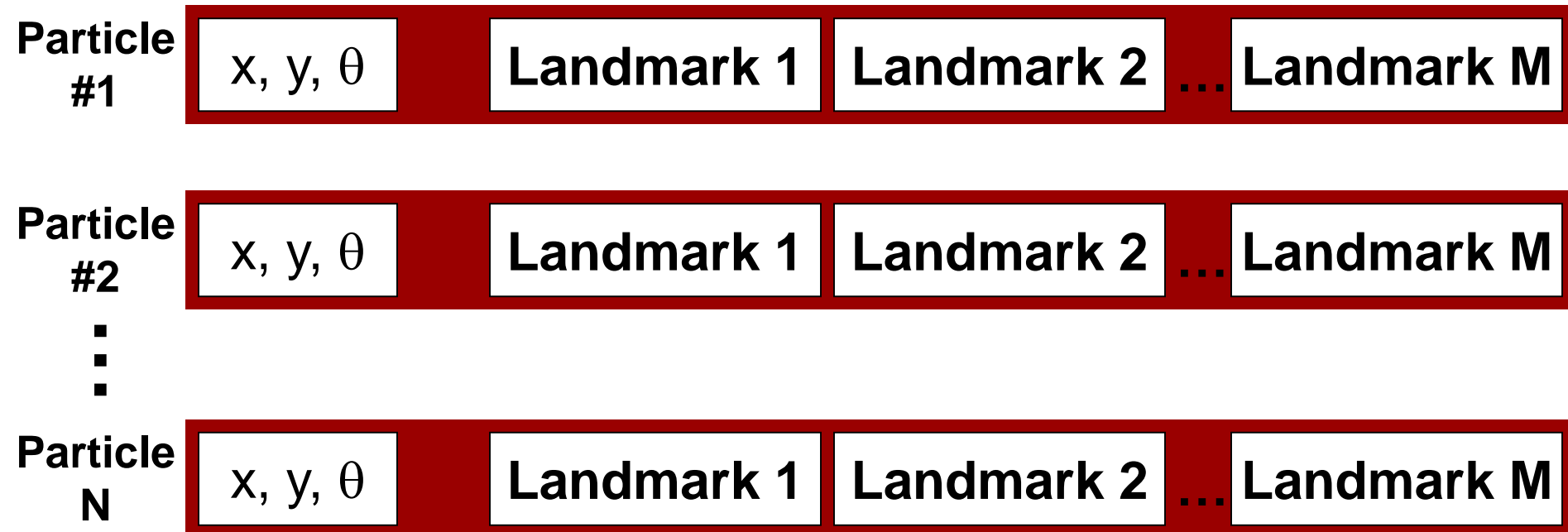
Rao-Blackwellization for SLAM

$$p(x_{1:t}, l_{1:m} \mid z_{1:t}, u_{0:t-1}) = p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot \prod_{i=1}^M p(l_i \mid x_{1:t}, z_{1:t})$$

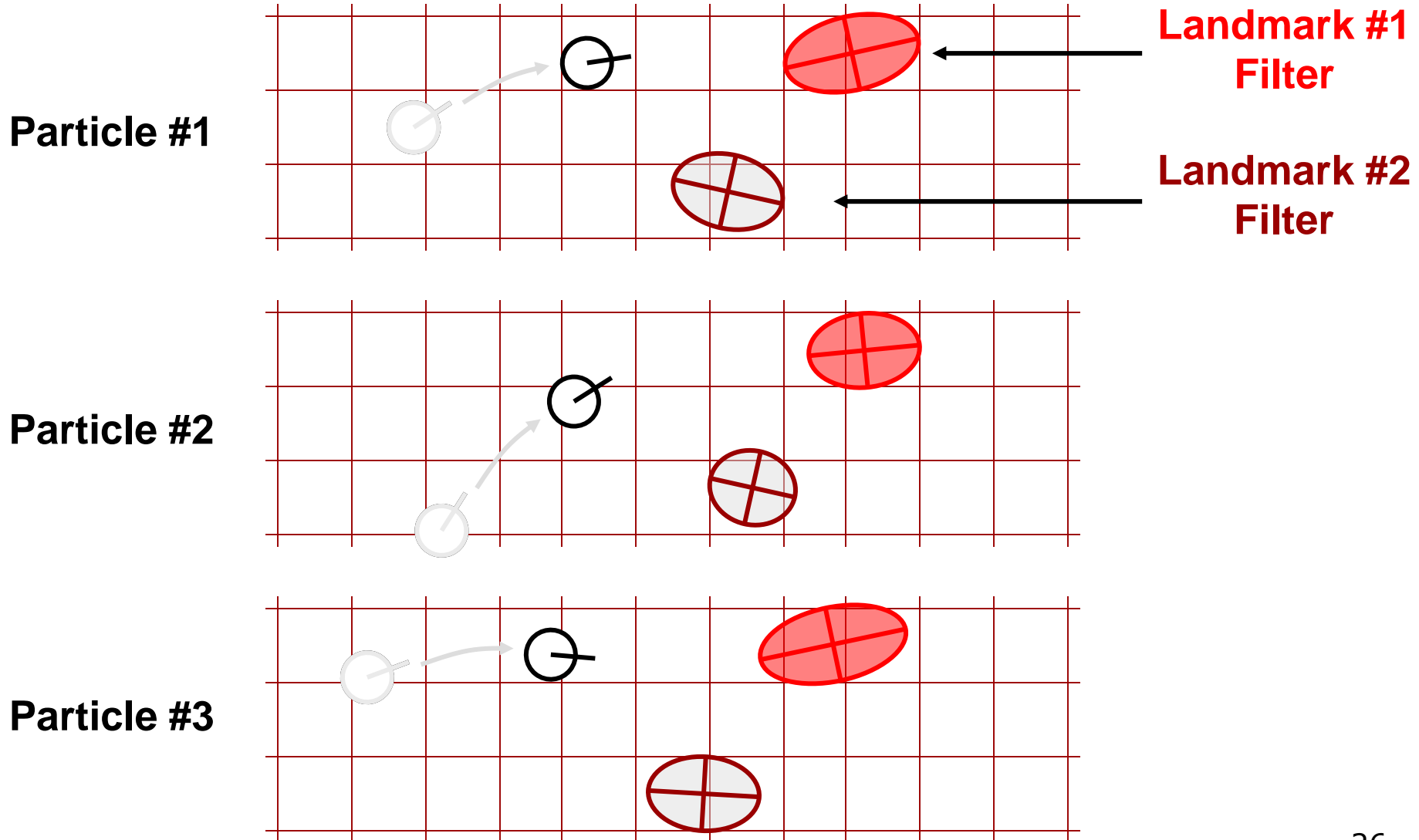
- Given that the second term can be computed efficiently, particle filtering becomes possible!

FastSLAM

- Rao-Blackwellized particle filtering based on landmarks [Montemerlo et al., 2002]
- Each landmark is represented by a 2x2 Extended Kalman Filter (EKF)
- Each particle therefore has to maintain M EKFs

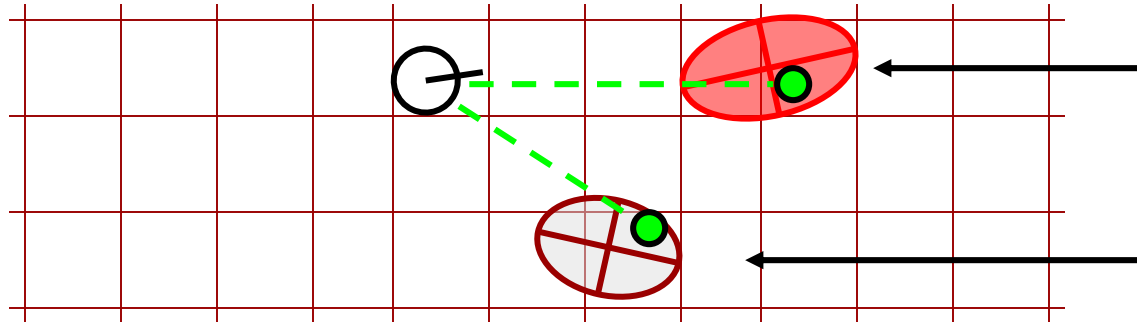


FastSLAM – Action Update



FastSLAM – Sensor Update

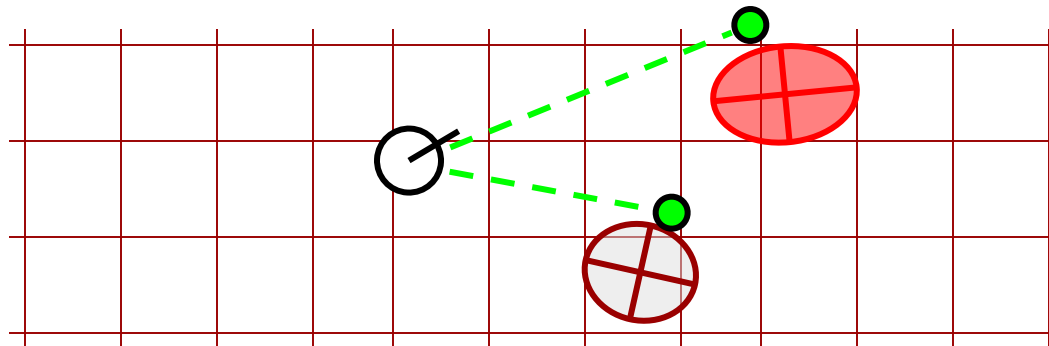
Particle #1



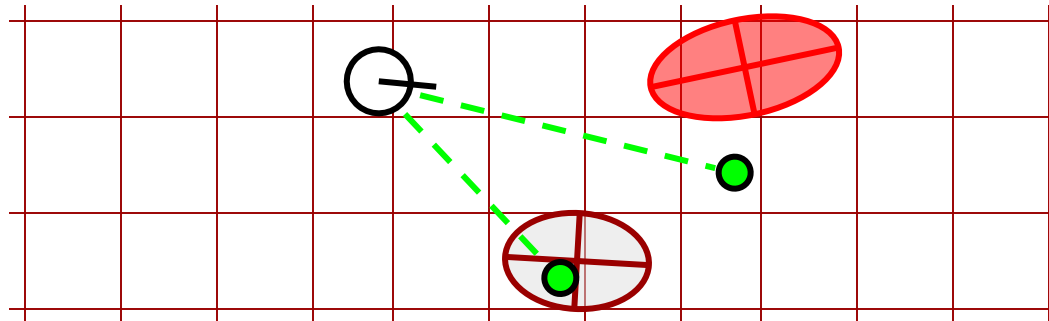
Landmark #1
Filter

Landmark #2
Filter

Particle #2

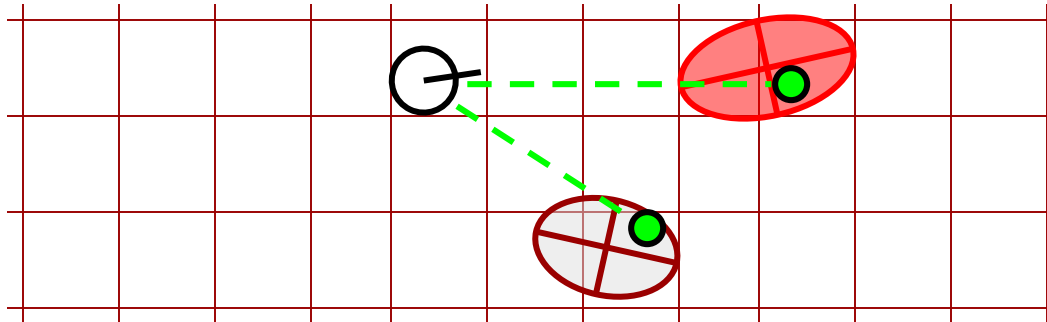


Particle #3



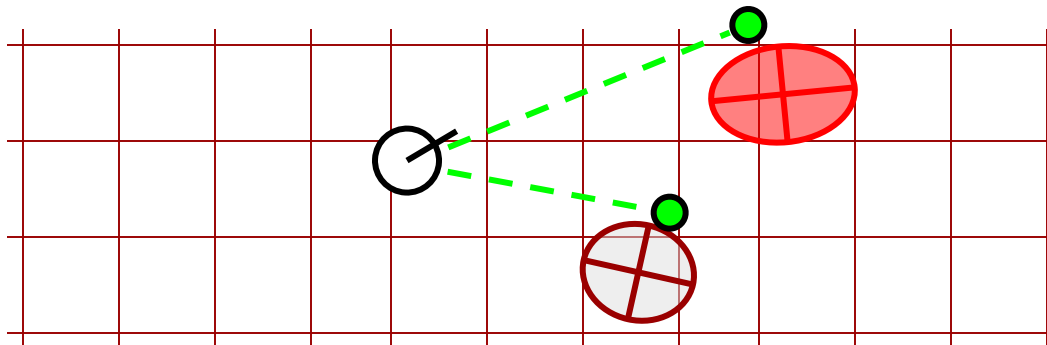
FastSLAM – Sensor Update

Particle #1



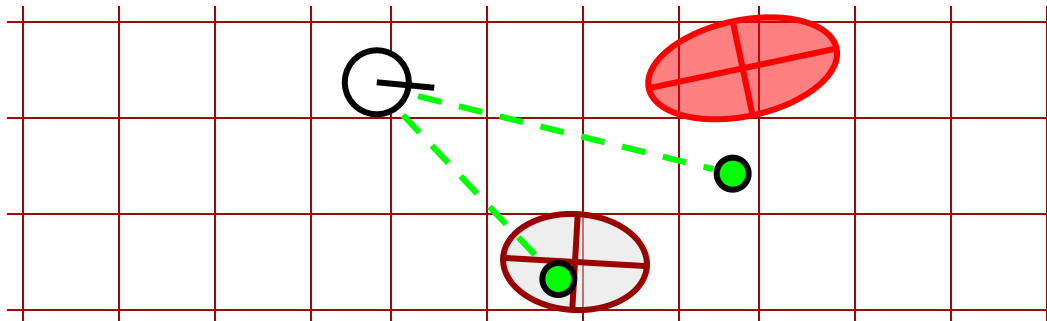
Weight = 0.8

Particle #2



Weight = 0.4

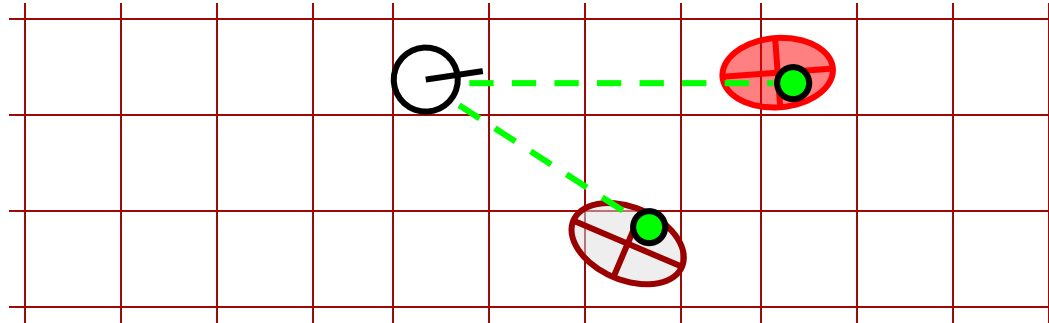
Particle #3



Weight = 0.1

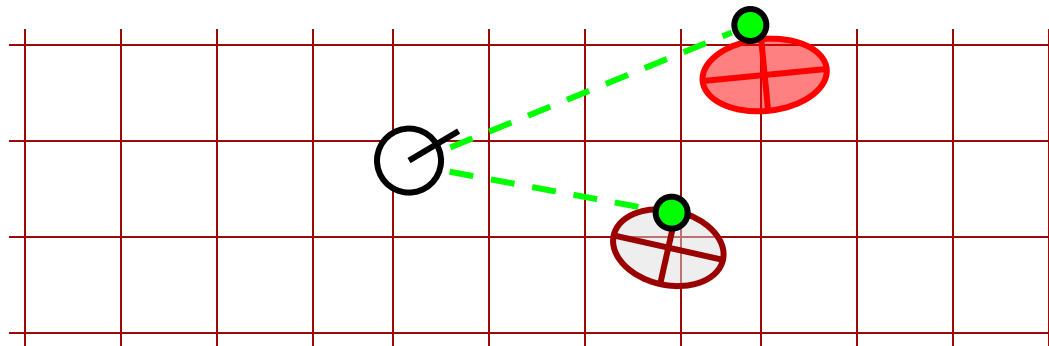
FastSLAM – Sensor Update

Particle #1



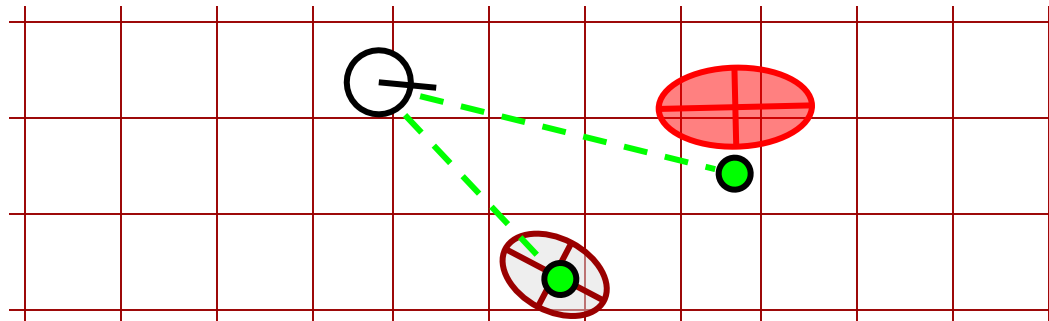
Update map
of particle #1

Particle #2



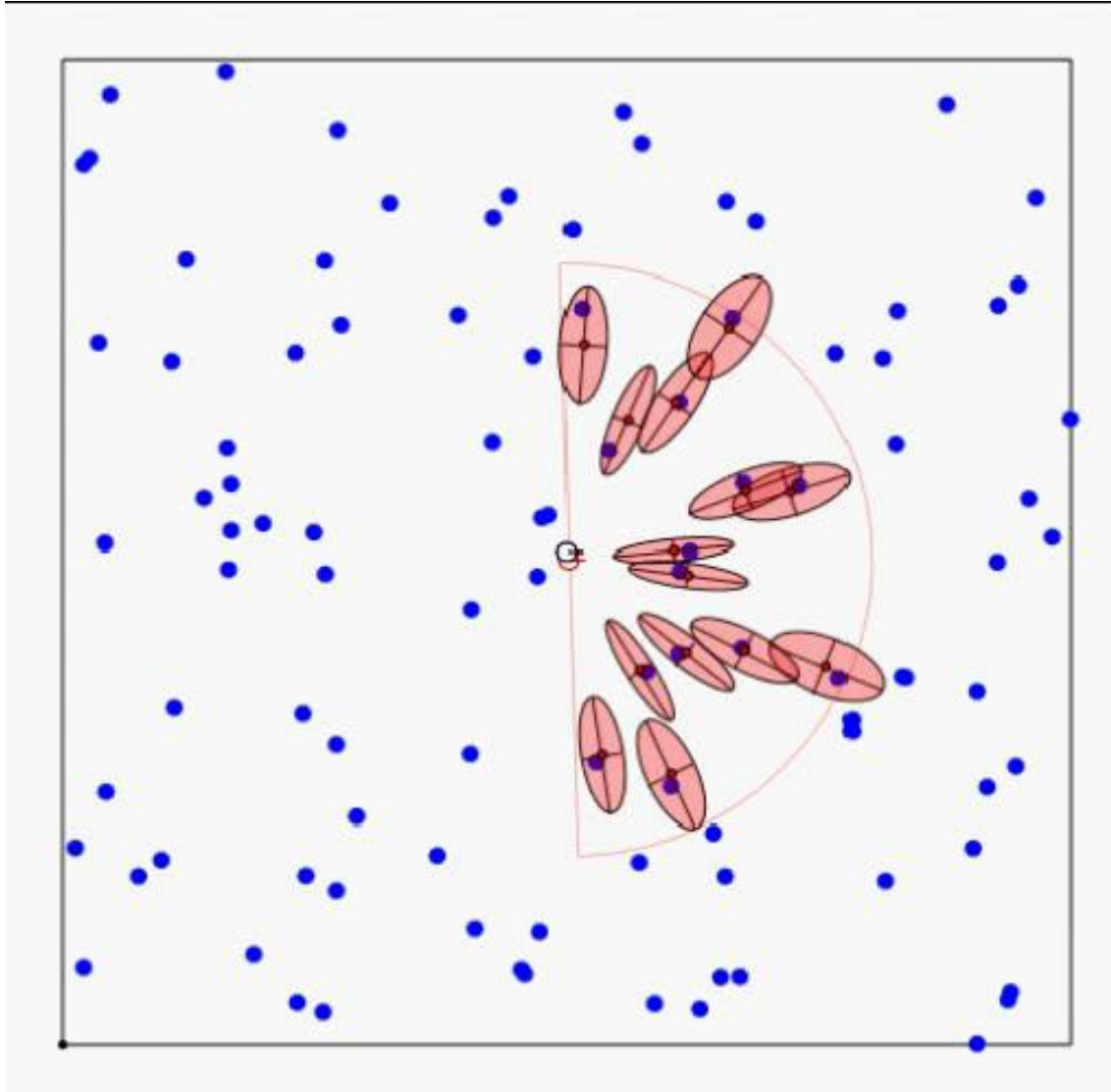
Update map
of particle #2

Particle #3



Update map
of particle #3

FastSLAM - Video



FastSLAM Complexity – Naive

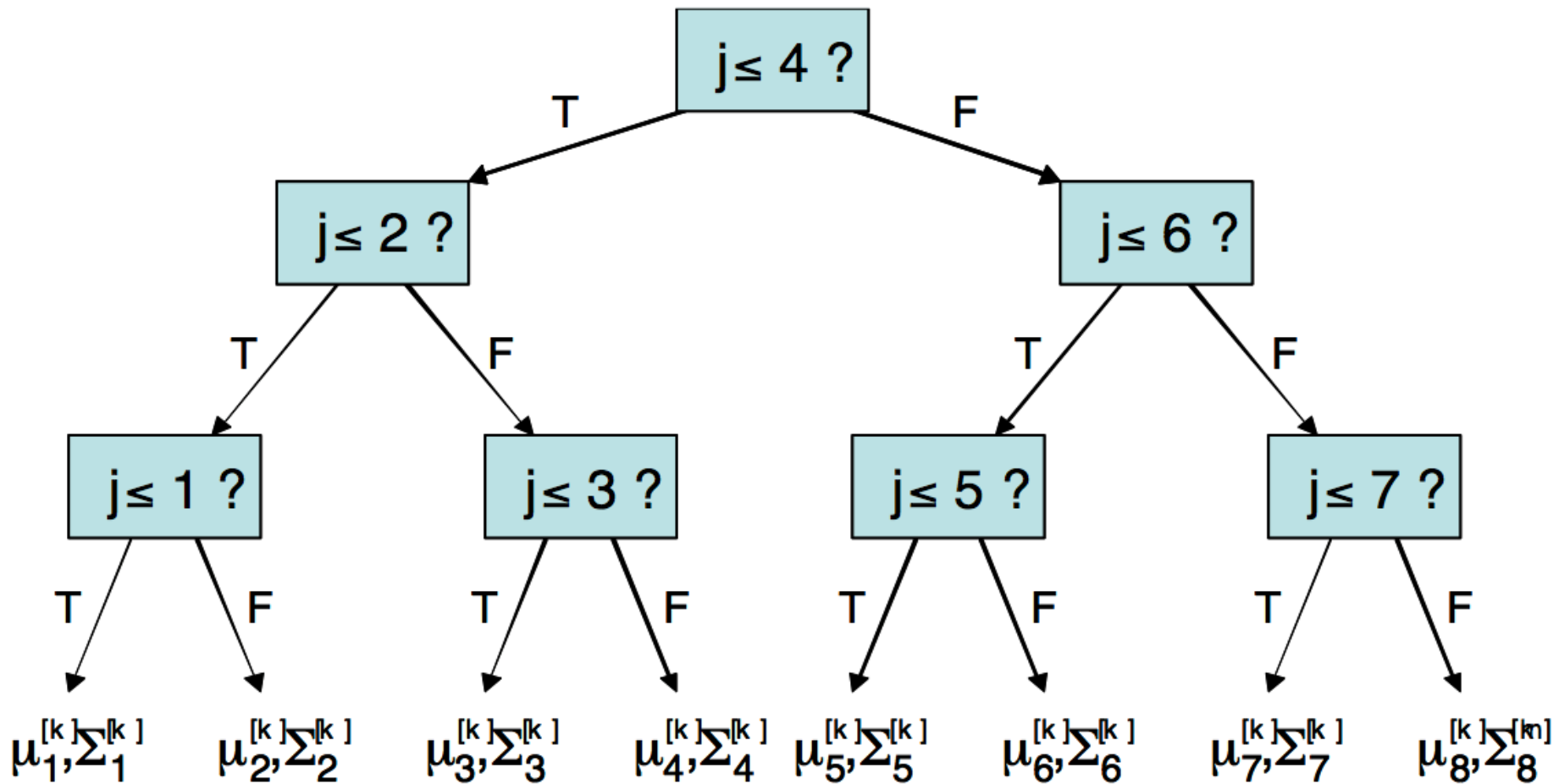
- Update robot particles based on the control $\mathcal{O}(N)$
- Incorporate an observation into the Kalman filters (given the data association) $\mathcal{O}(N)$
- Resample particle set $\mathcal{O}(NM)$

N = Number of particles

M = Number of map features

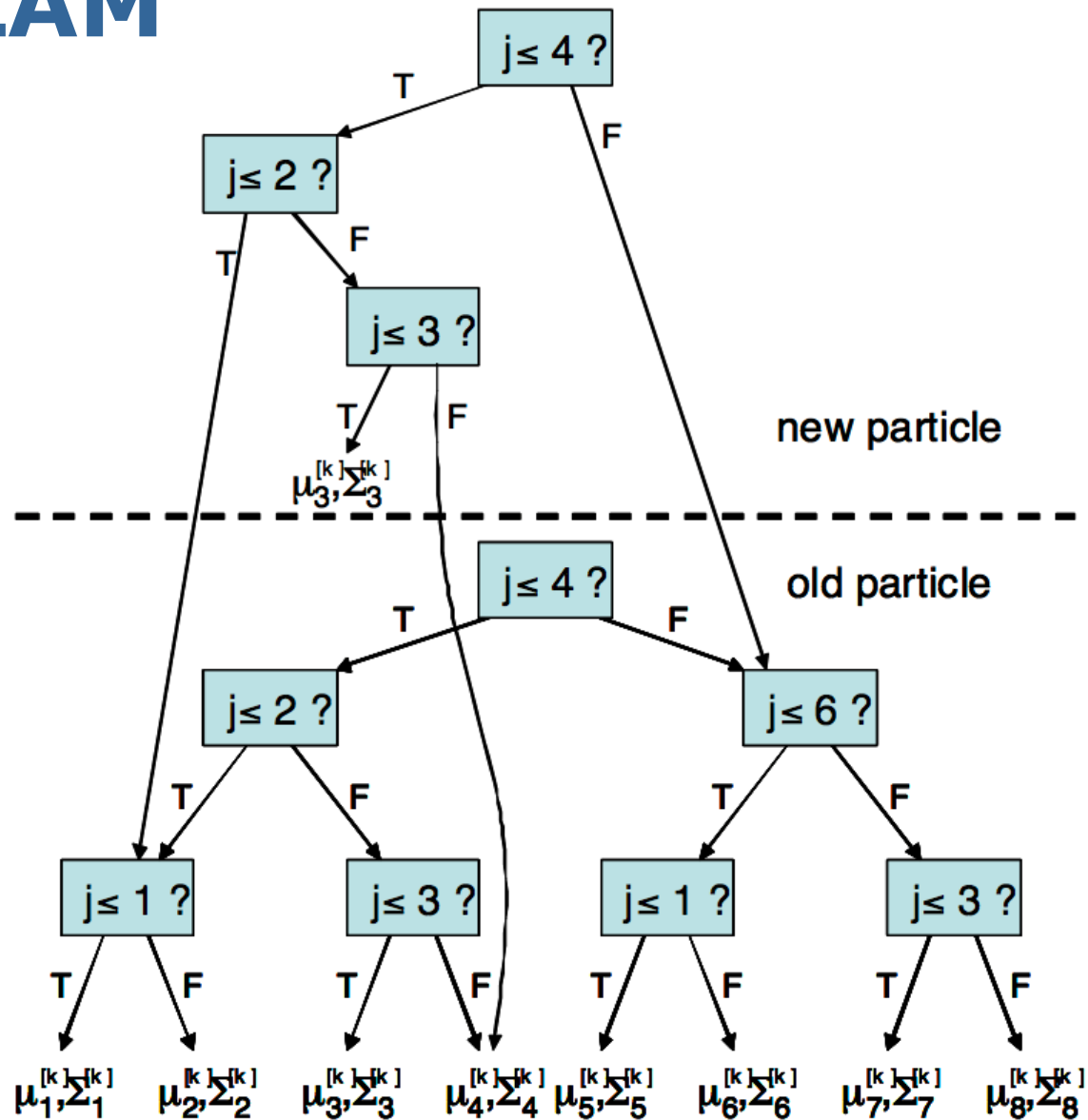
$$\mathcal{O}(NM)$$

A Better Data Structure for FastSLAM



Courtesy: M. Montemerlo

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 (given the data association) $\mathcal{O}(N \log M)$
- Resample particle set $\mathcal{O}(N \log M)$

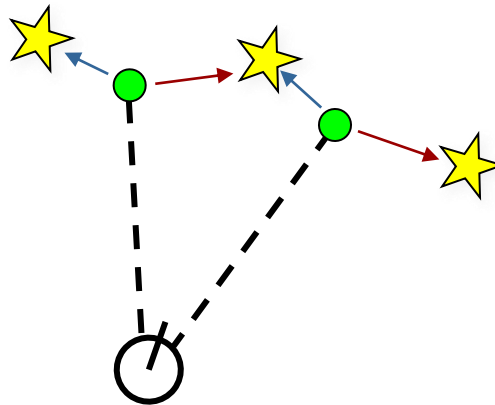
N = Number of particles

M = Number of map features

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

Data Association Problem

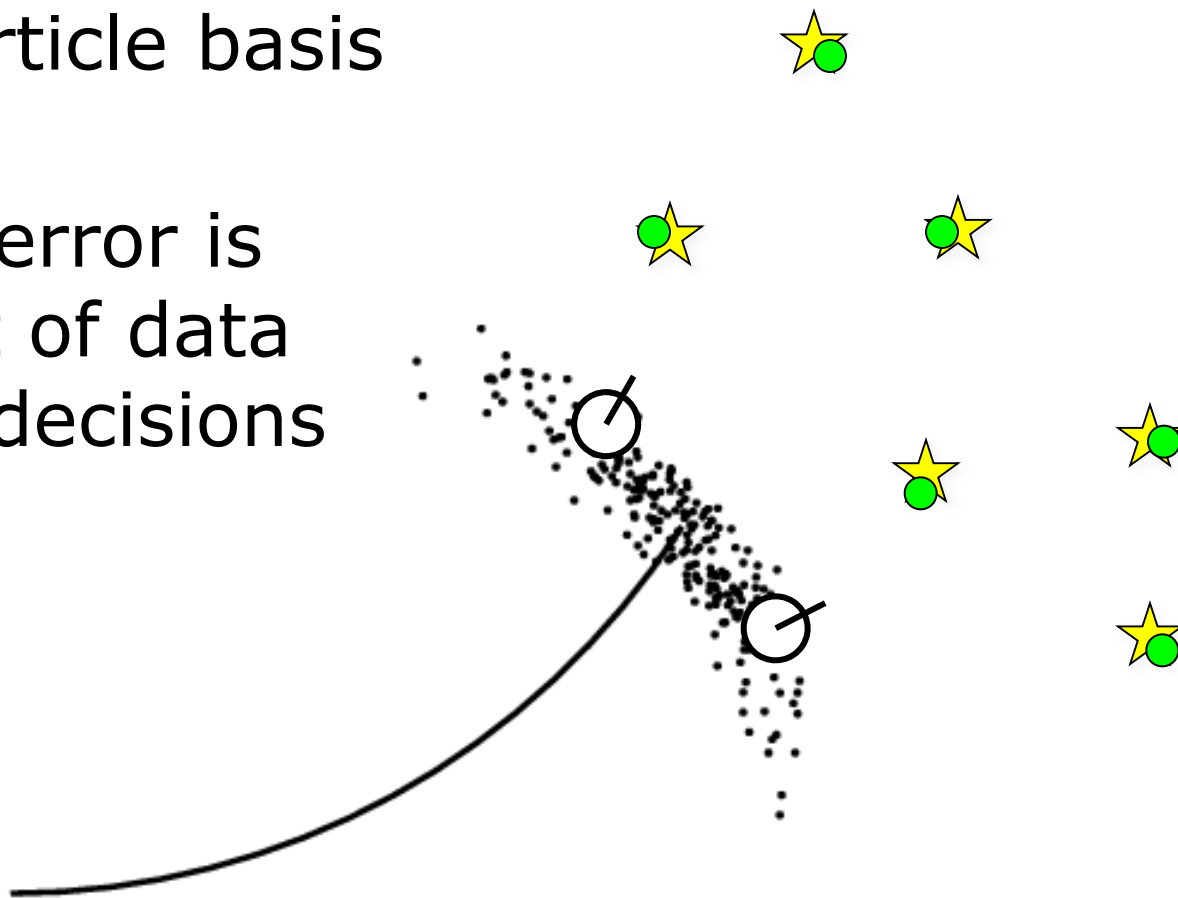
- Which observation belongs to which landmark?



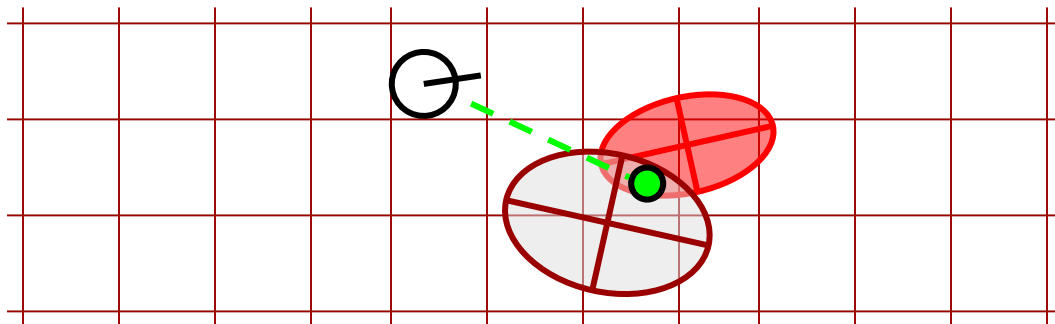
- A robust SLAM solution must consider possible data associations
- Potential data associations depend also on the pose of the robot

Multi-Hypothesis Data Association

- Data association is done 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 the brown landmark?

$$P(\text{observation}|\text{red}) = 0.3$$

$$P(\text{observation}|\text{brown}) = 0.7$$

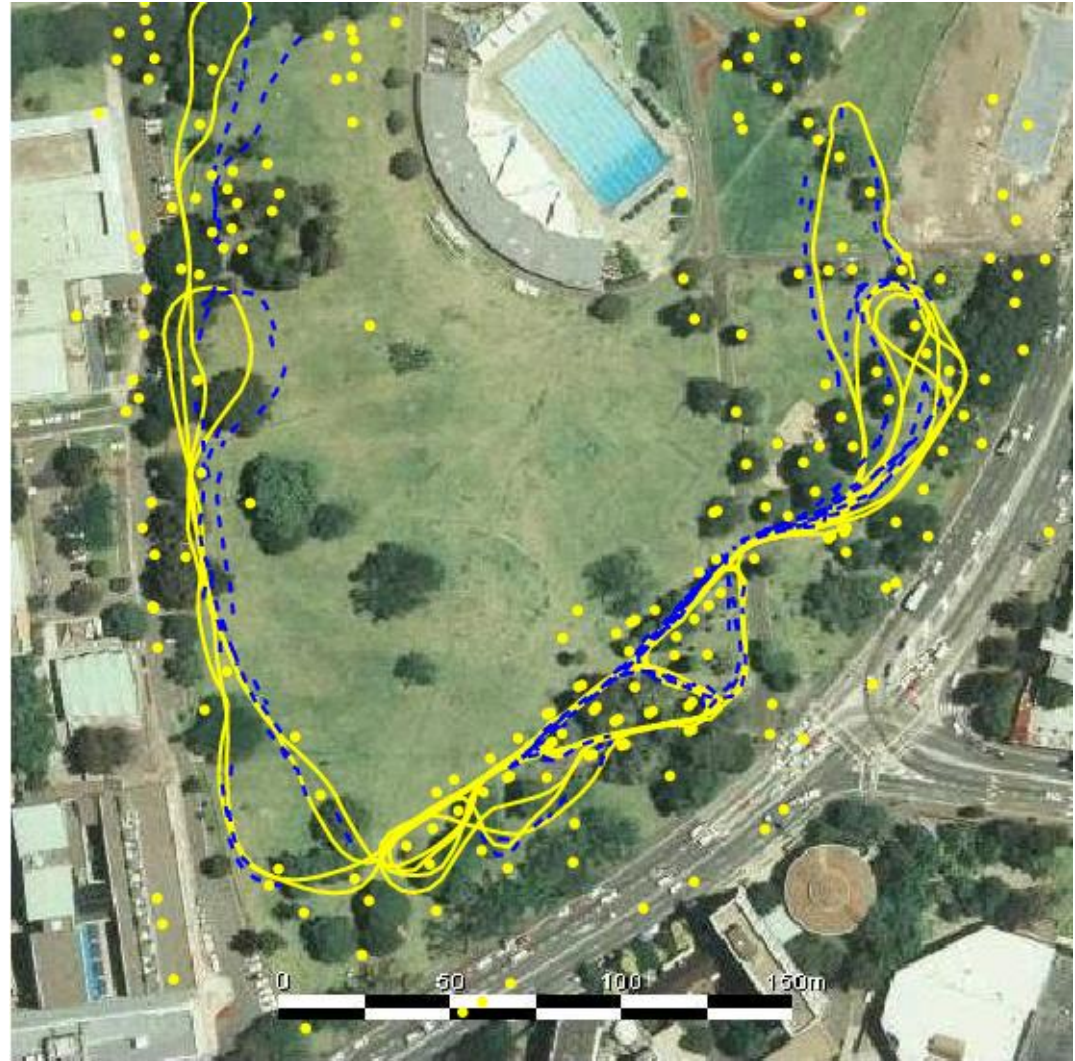
- Two options for per-particle data association
 - Pick the most probable match
 - Pick a random association weighted by the observation likelihoods
- If the probability is too low, generate a new landmark

Results – Victoria Park

- 4 km traverse
- < 5 m RMS position error
- 100 particles

Blue = GPS

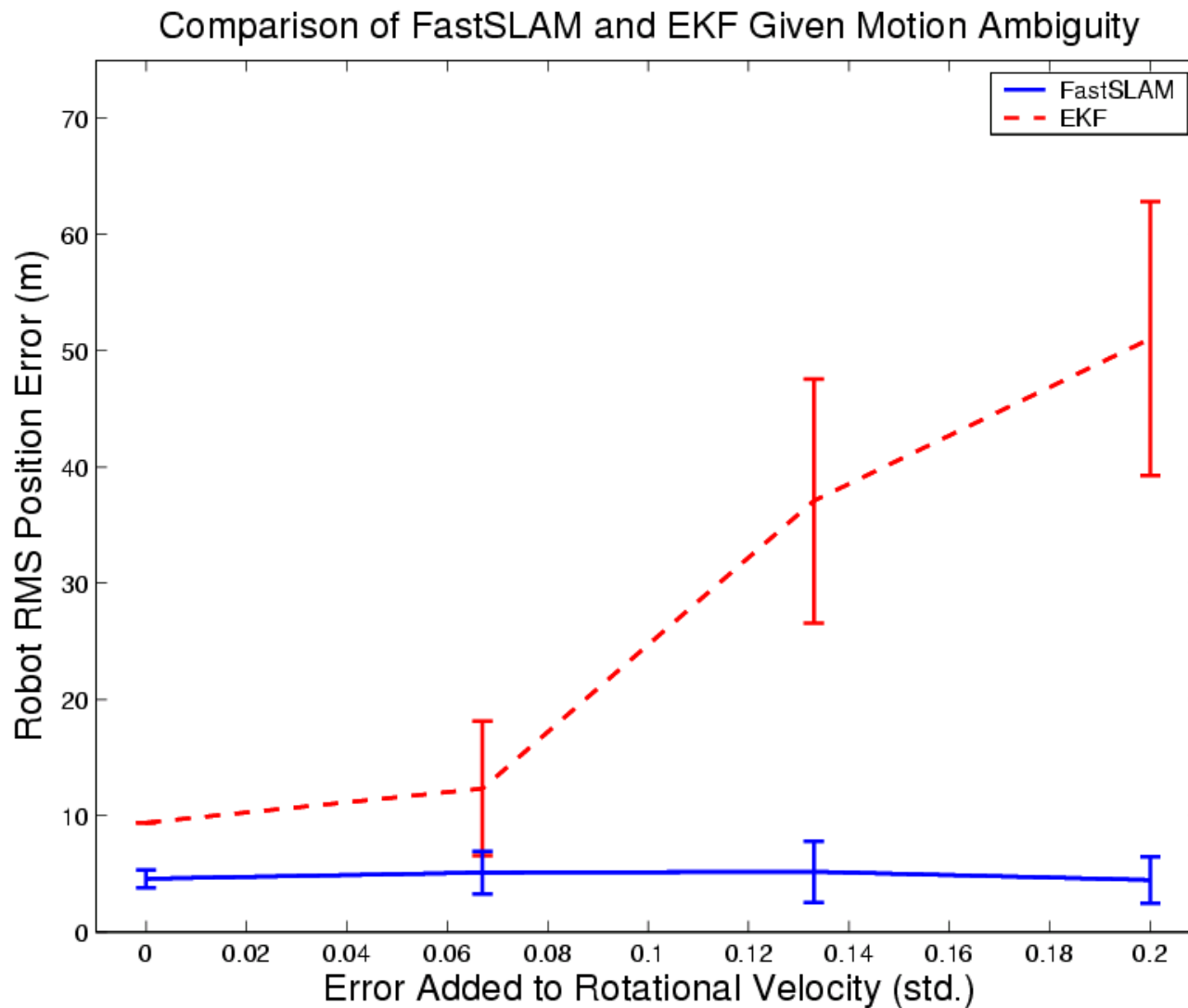
Yellow = FastSLAM



Results – Victoria Park (Video)



Results – Data Association



FastSLAM Summary

- FastSLAM factors the SLAM posterior into low-dimensional estimation problems
 - Scales to problems with over 1 million features
- FastSLAM factors robot pose uncertainty out of the data association problem
 - Robust to significant ambiguity in data association
 - Allows data association decisions to be delayed until unambiguous evidence is collected
- Advantages compared to the classical EKF approach (especially with non-linearities)
- Complexity of $O(N \log M)$