

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
 - Resampling
- 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 the dimensions of the state space?
- If so, can we use the dependency to solve the problem more efficiently?

Dependencies

- Is there a dependency between the 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

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

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

↑
SLAM posterior


↑
Robot path posterior

↑
landmark positions

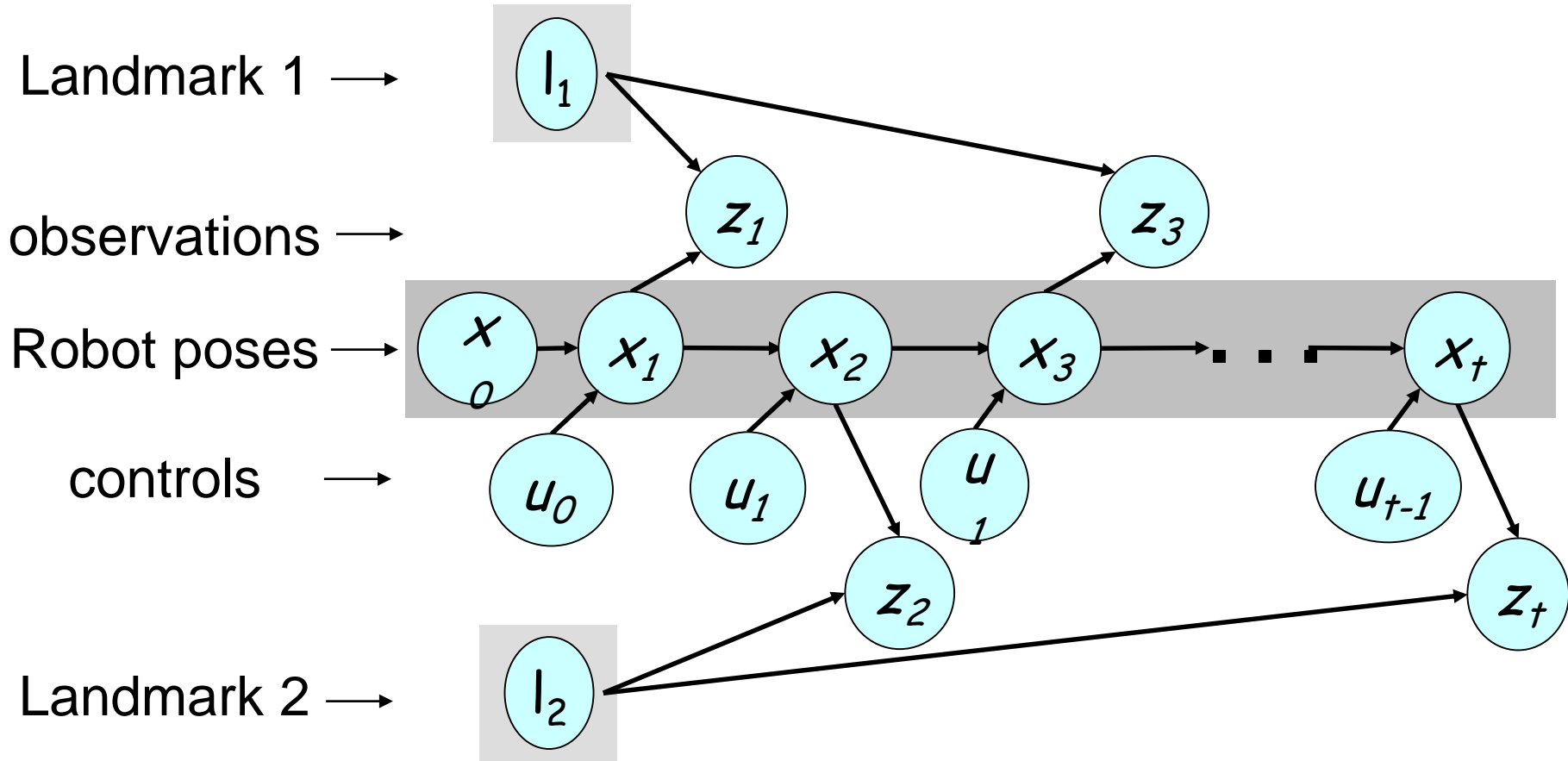
Does this help to solve the problem?

Factored Posterior (Landmarks)

poses map observations & movements


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

Mapping using Landmarks




Knowledge of the robot's true path renders landmark positions conditionally independent


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



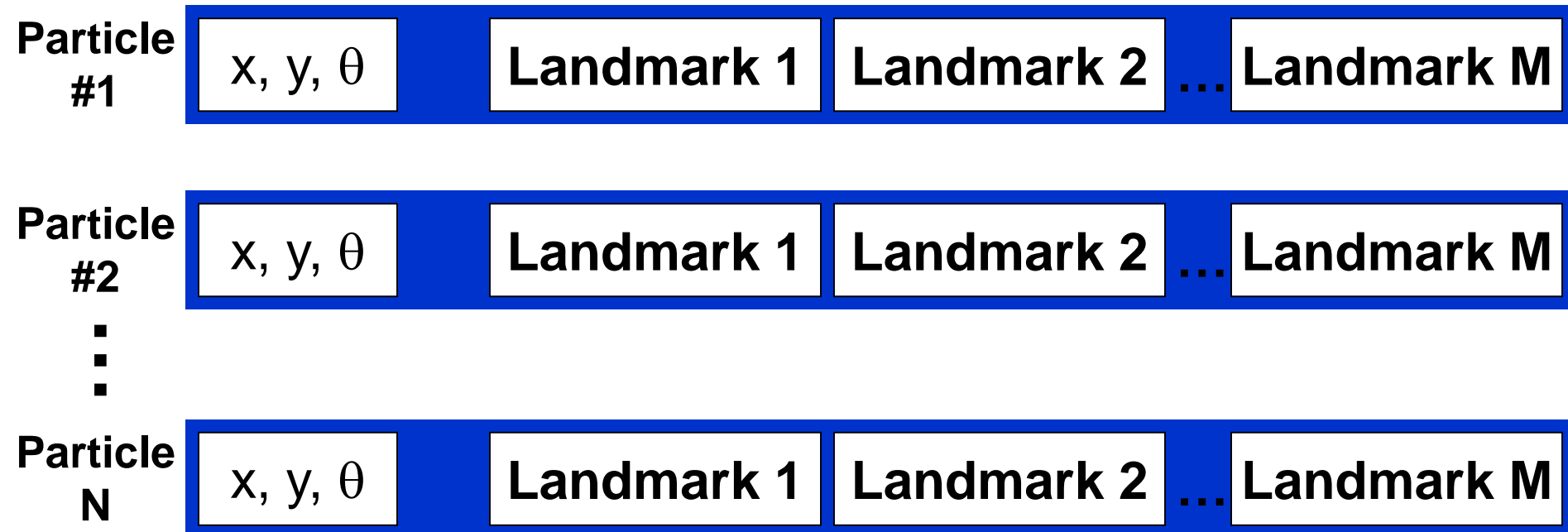
Rao-Blackwellization

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

- This factorization is also called Rao-Blackwellization
- Given that the second term can be computed efficiently, particle filtering becomes possible!

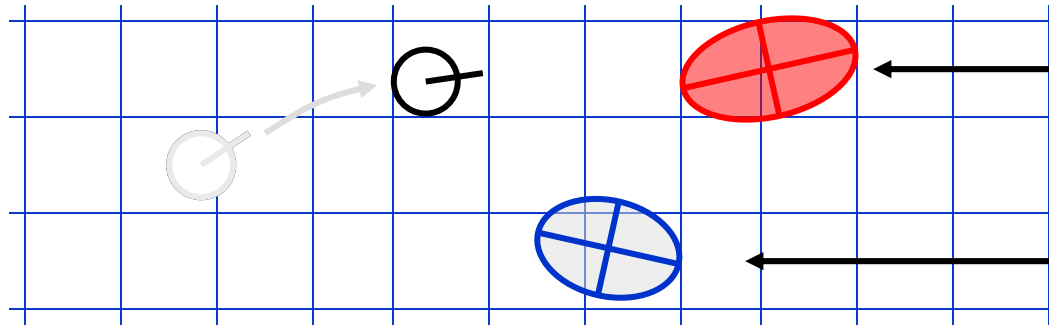
FastSLAM

- 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

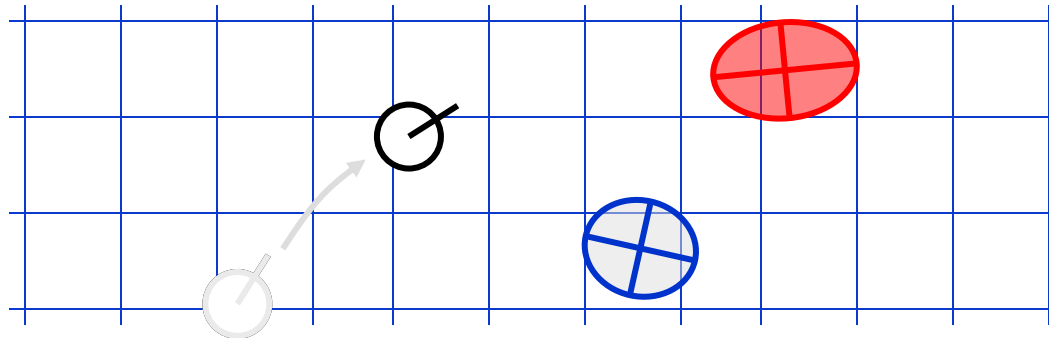
Particle #1



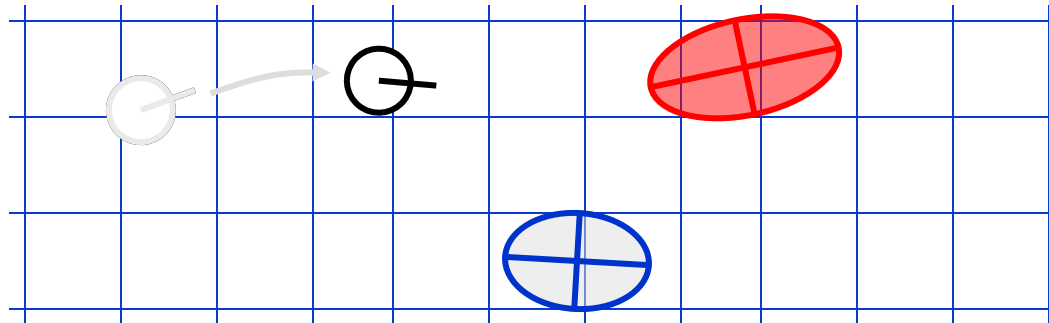
Landmark #1
Filter

Landmark #2
Filter

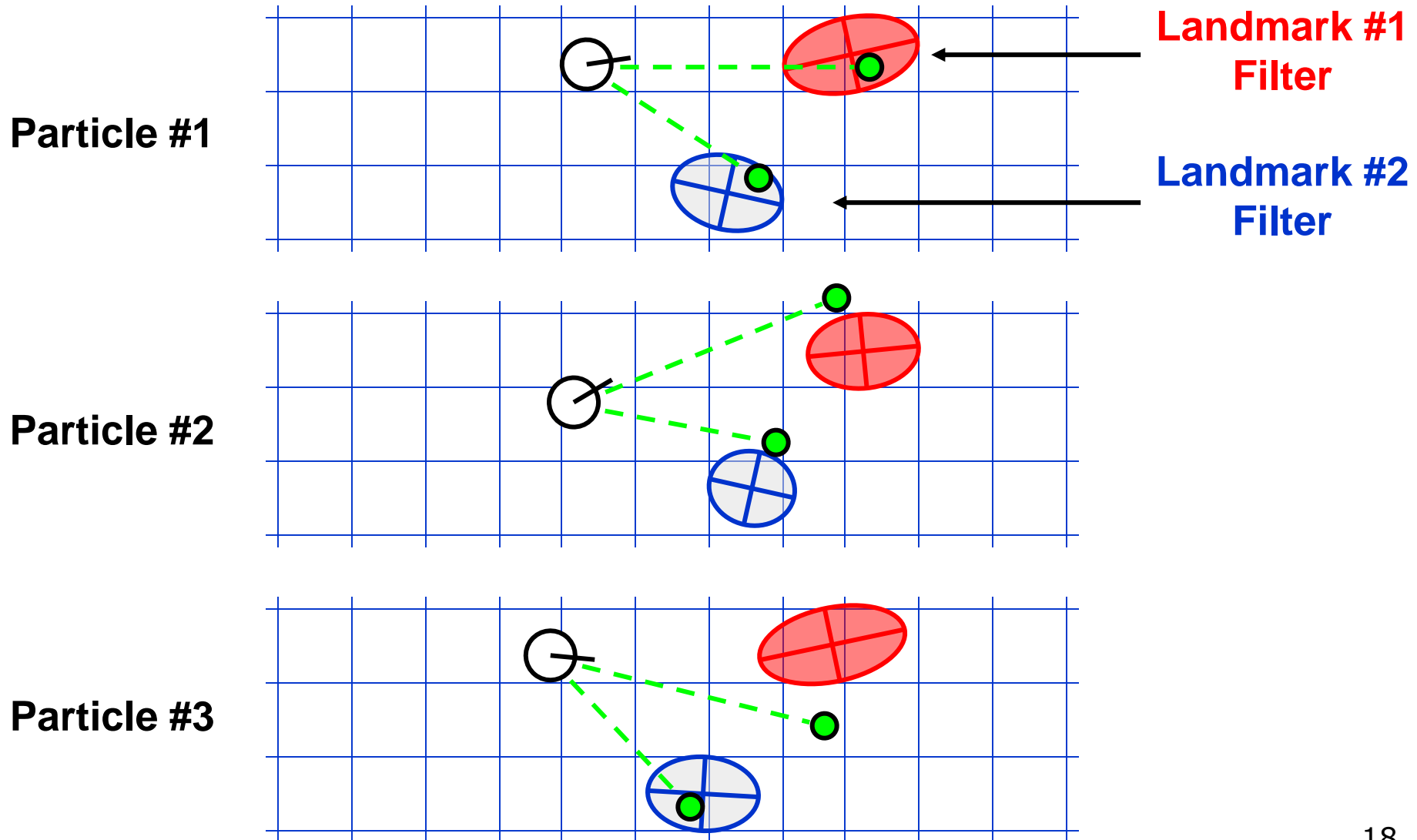
Particle #2



Particle #3

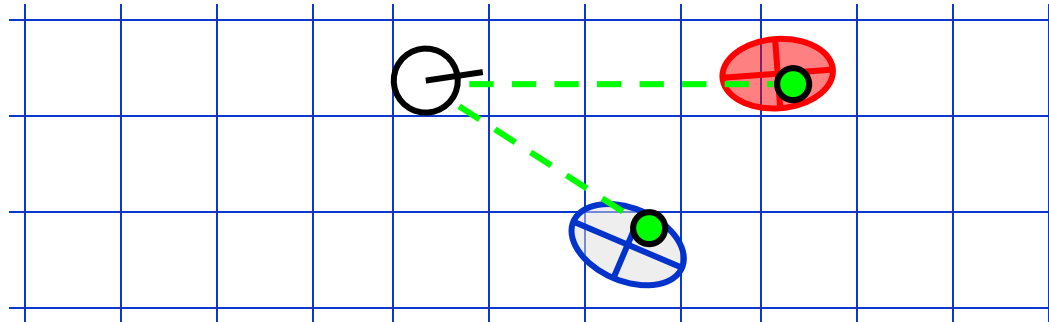


FastSLAM – Sensor Update



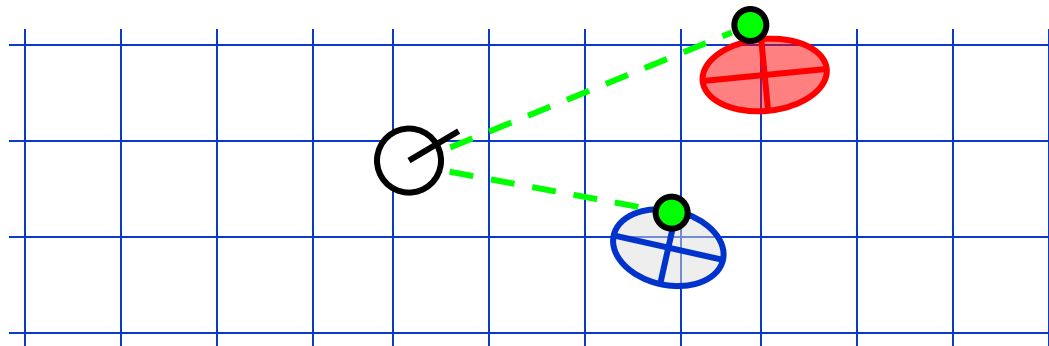
FastSLAM – Sensor Update

Particle #1



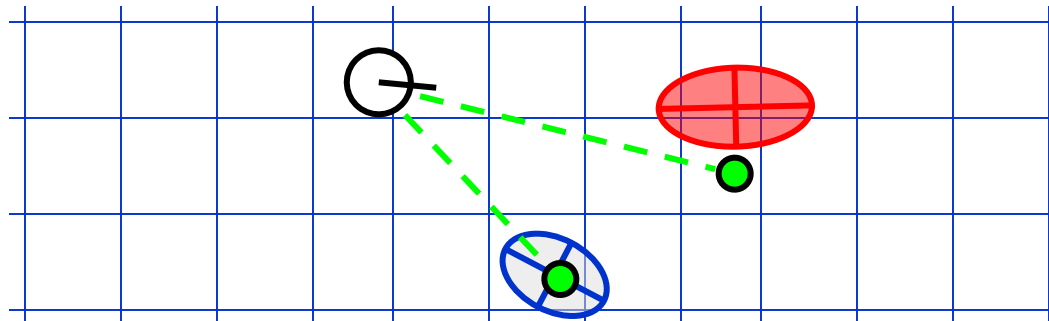
Weight = 0.8

Particle #2



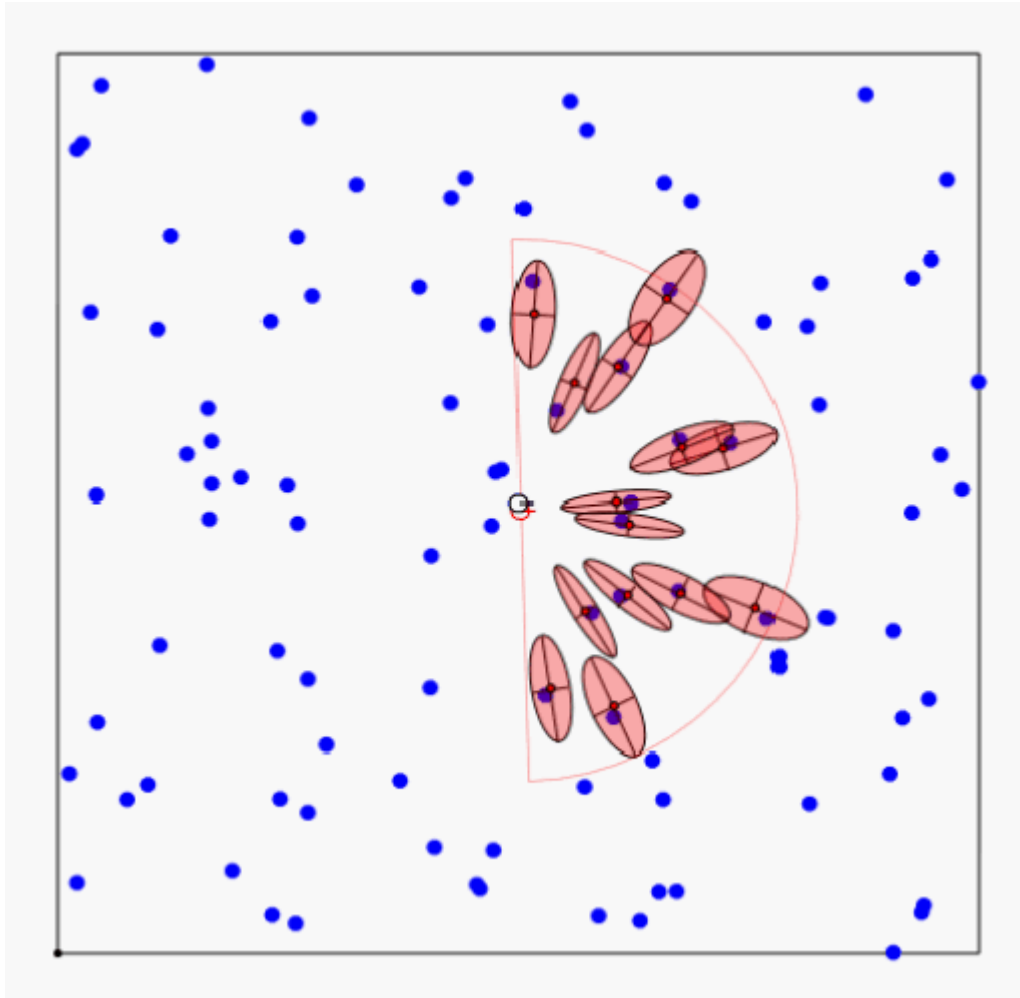
Weight = 0.4

Particle #3



Weight = 0.1

FastSLAM - Video



FastSLAM Complexity

- Update robot particles based on control u_{t-1}

$O(N)$
Constant time per particle

- Incorporate observation z_t into Kalman filters

$O(N \cdot \log(M))$
Log time per particle

- Resample particle set

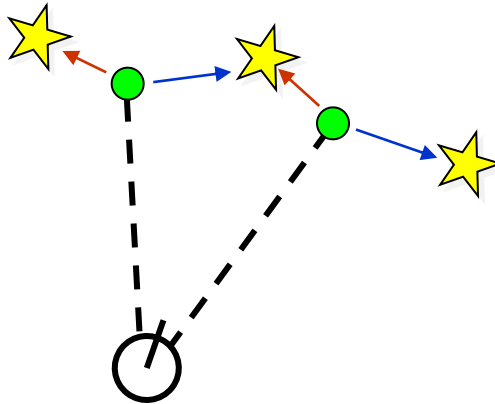
$O(N \cdot \log(M))$
Log time per particle

N = Number of particles
M = Number of map features

$O(N \cdot \log(M))$
Log time per particle

Data Association Problem

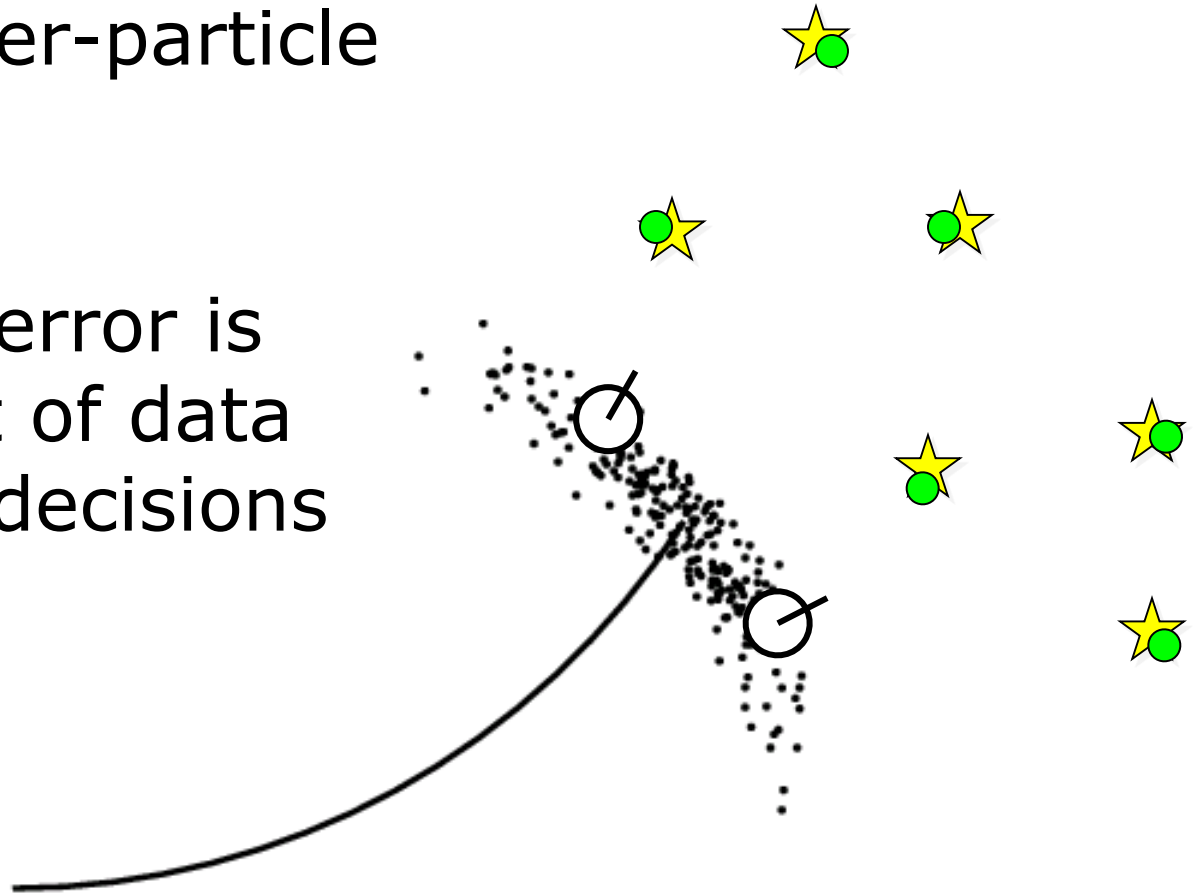
- Which observation belongs to which landmark?



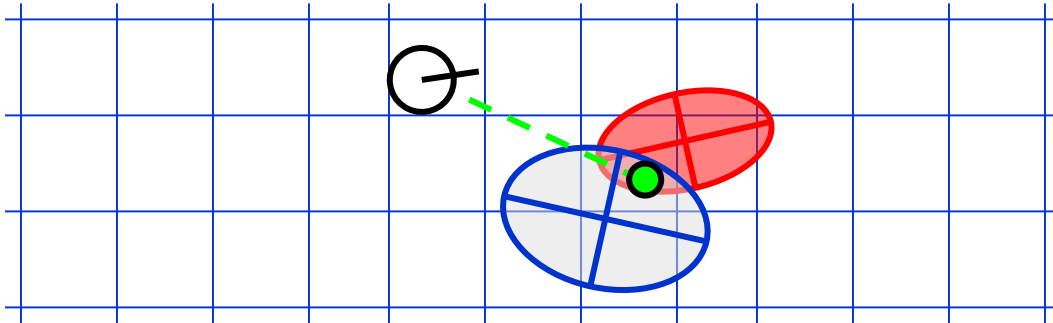
- A robust SLAM must consider possible data associations
- Potential data associations depend also on the pose of the robot

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 blue landmark?

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

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

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

Results – Victoria Park

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

Blue = GPS

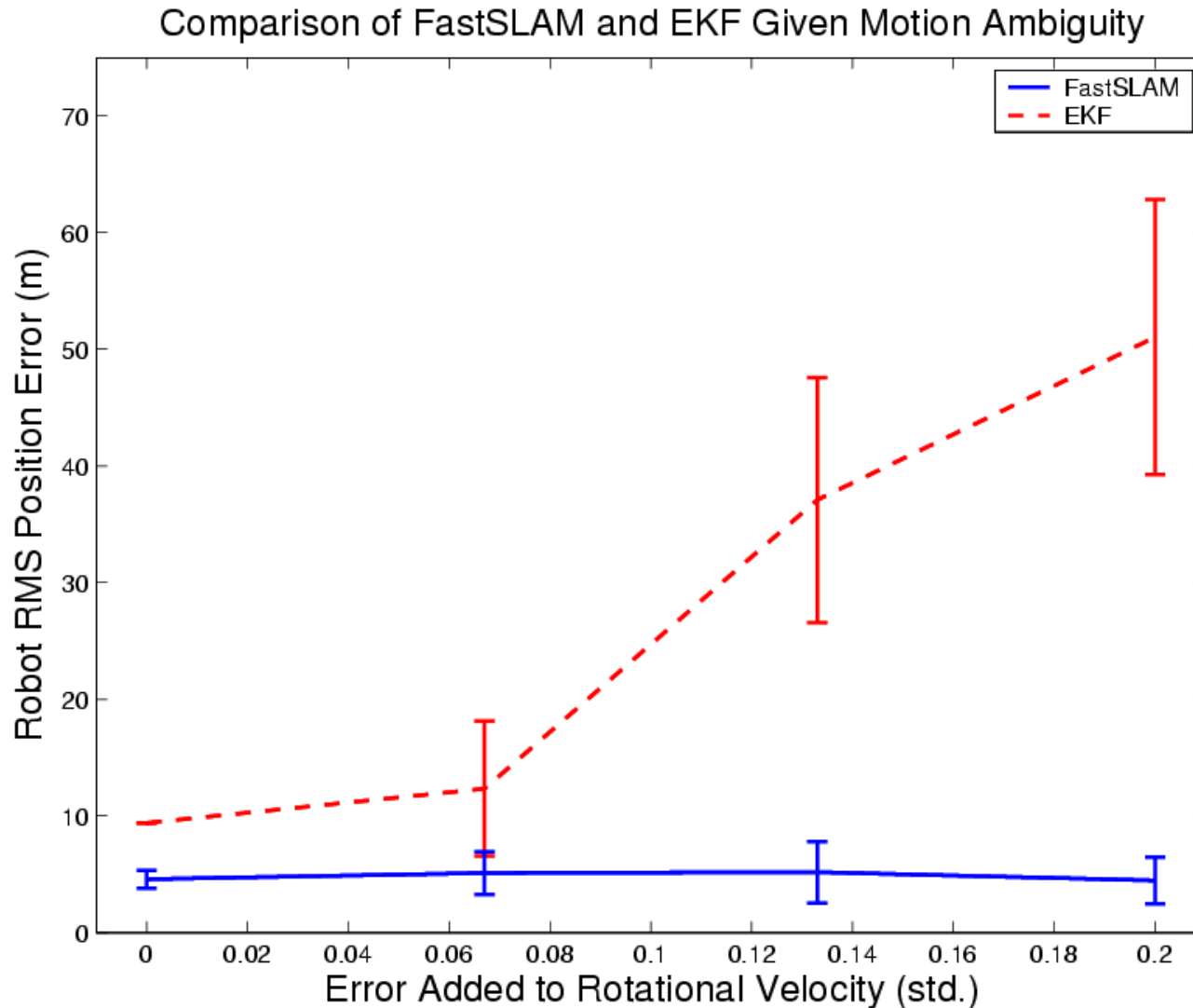
Yellow = FastSLAM



Results – Victoria Park



Results – Data Association



Results – Accuracy

