Probabilistic Robotics
Chapter 13: The FastSLAM Algorithm

MSc course Artificial Intelligence 2018
http://staff.fnwi.uva.nl/a.visser/education/ProbabilisticRobotics/

Arnoud Visser & Emiel Hoogeboom
Informatics Institute
Universiteit van Amsterdam
A.Visser@uva.nl

Probabilistic Robotics: FastSLAM

Sebastian Thrun & Alex Teichman

Stanford Artificial Intelligence Lab

Slide credits: Wolfram Burgard, Dieter Fox, Cyrill Stachniss, Giorgio Grisetti, Maren Bennewitz, Christian Plagemann, Dirk Haehnel, Mike Montemerlo, Nick Roy, Kai Arras, Patrick Pfaff and others

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 and
    a pose estimate is needed to build a map

The SLAM Problem

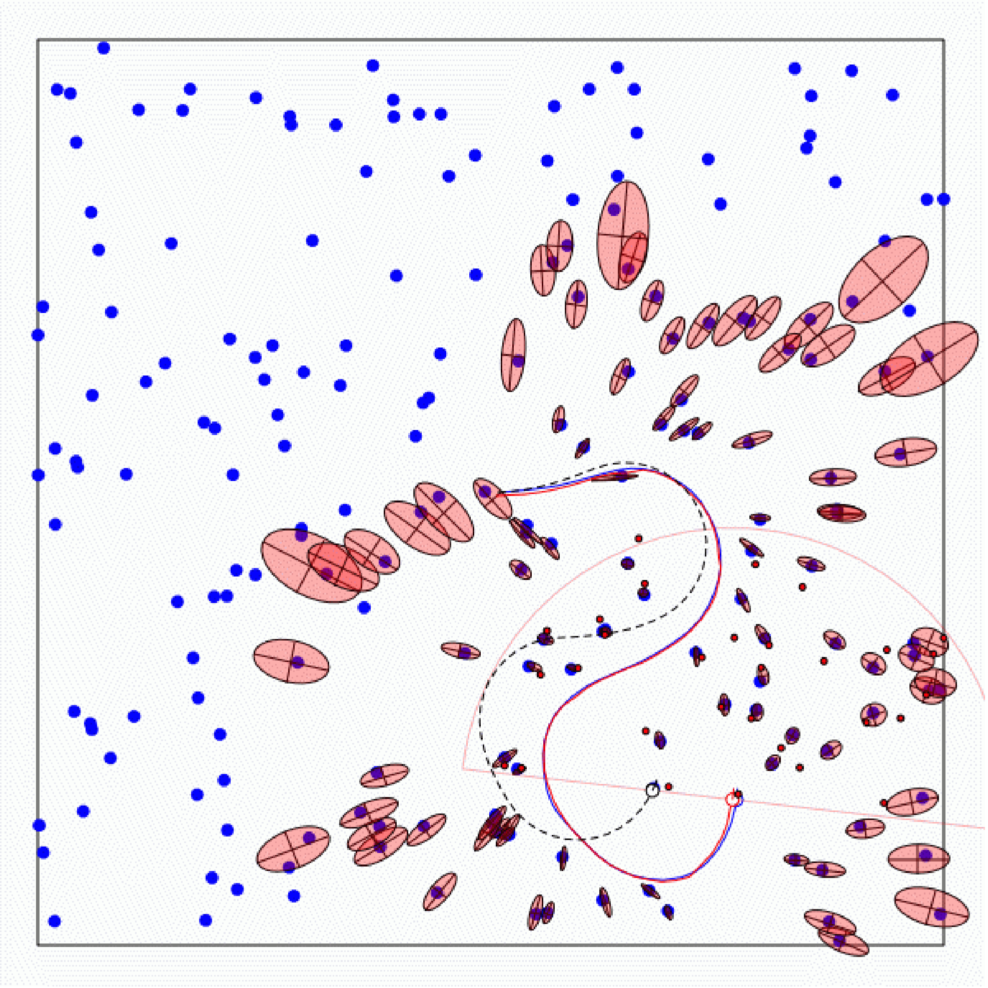
A robot moving though an unknown, static environment

Given:

- The robot's controls
- Observations of nearby features

Estimate:

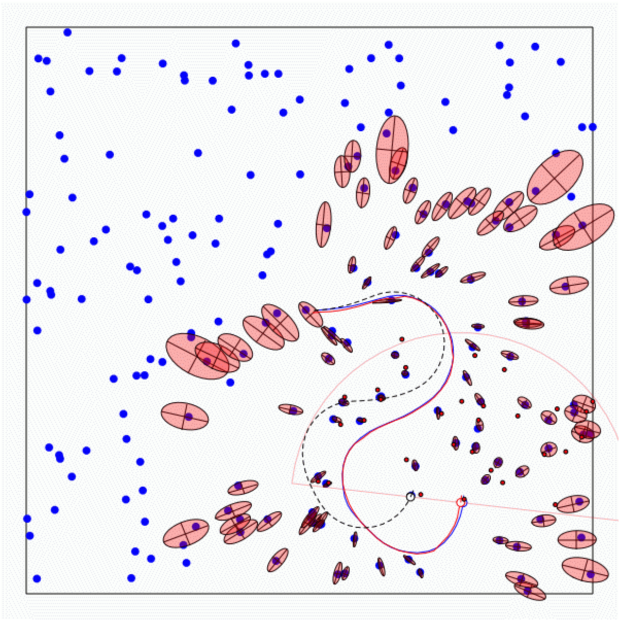
- Map of features
- Path of the robot



Map Representations

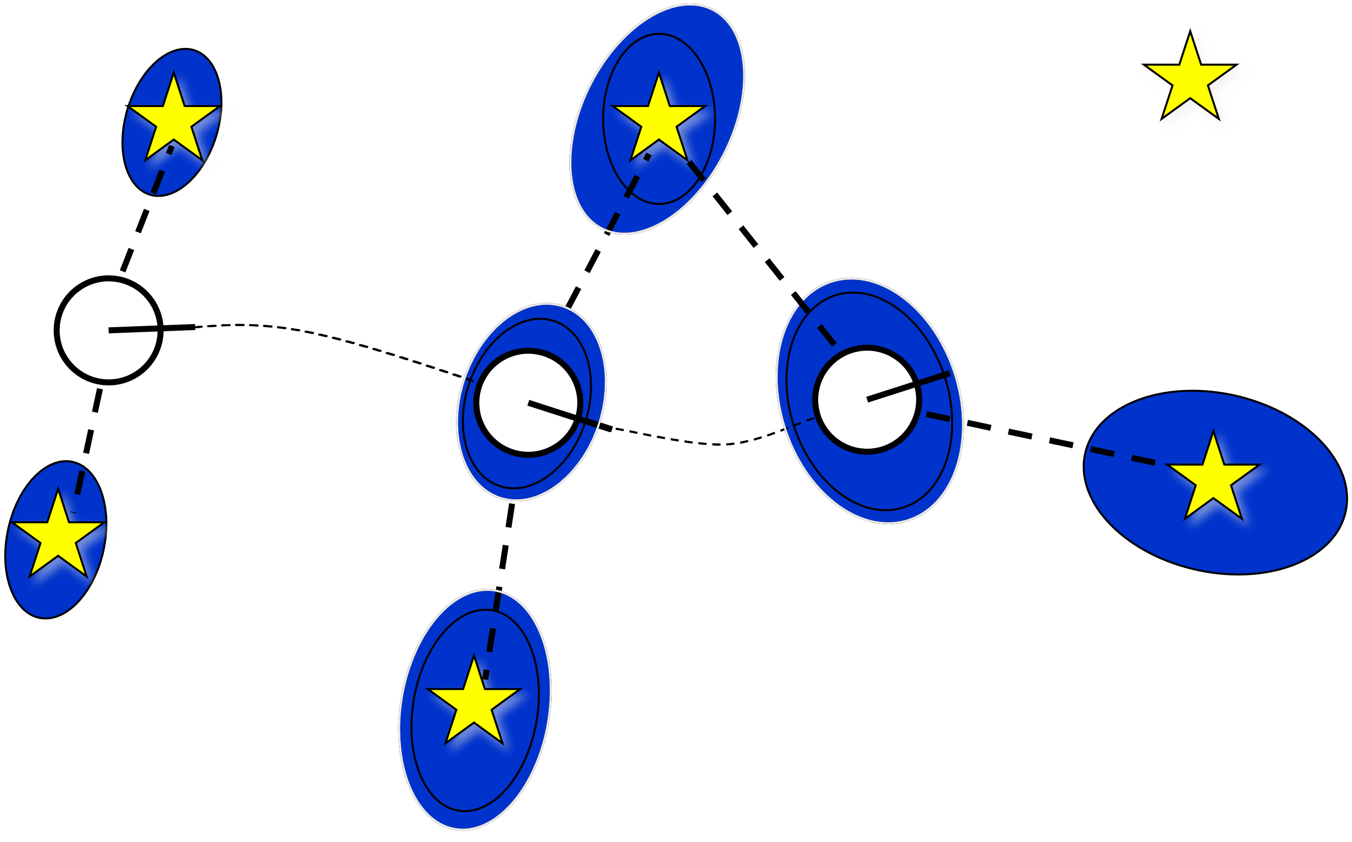
Typical models are:

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



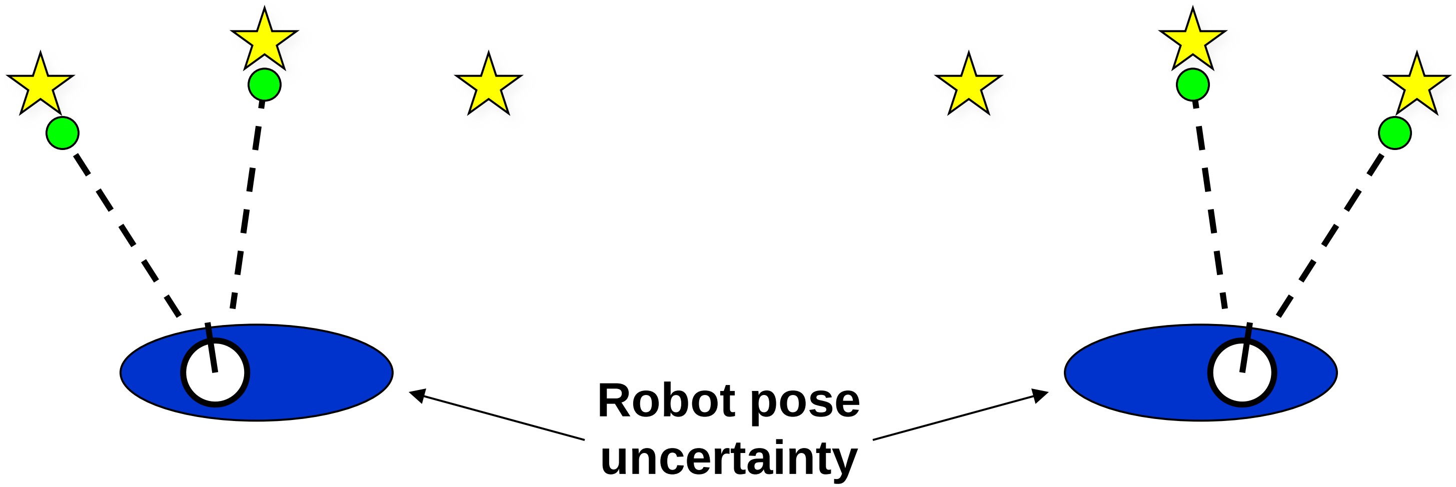
Why is SLAM a hard problem?

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



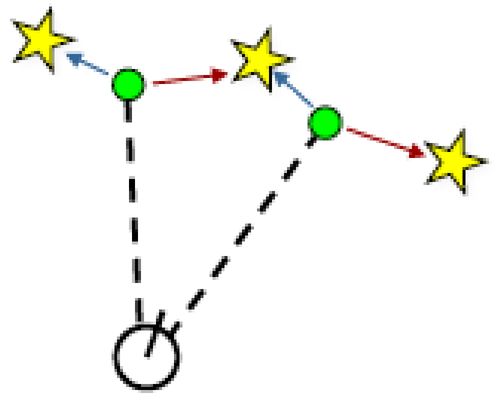
Robot path error correlates errors in the map

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"

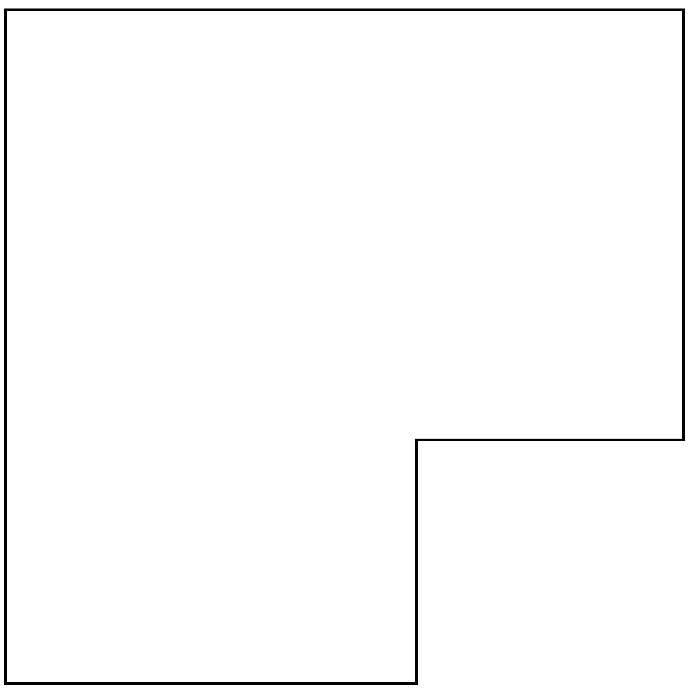
Particle Filters

- 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
  - Resampling

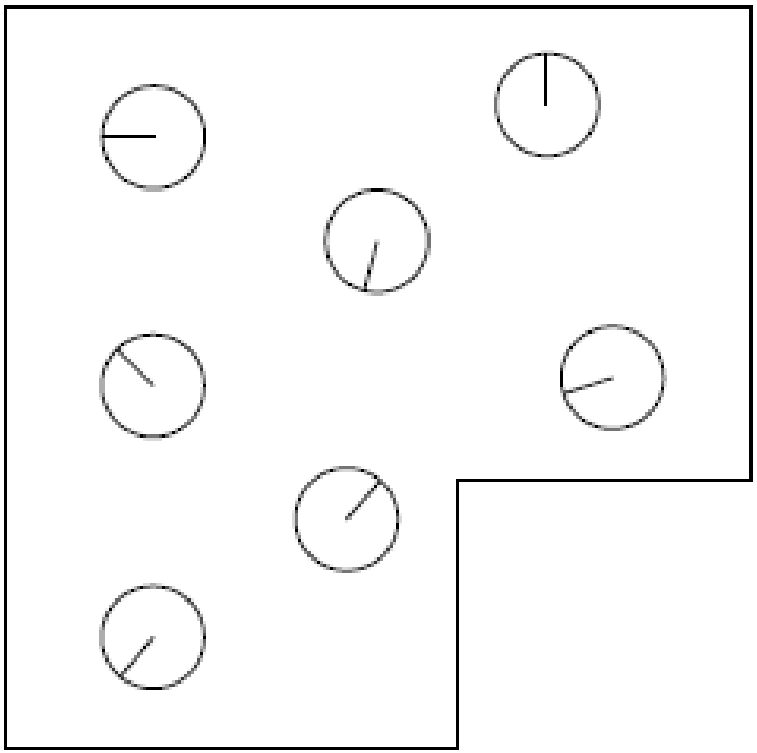
- Typical application scenarios are tracking, localization, ...

Recap: Particle Filter Localization



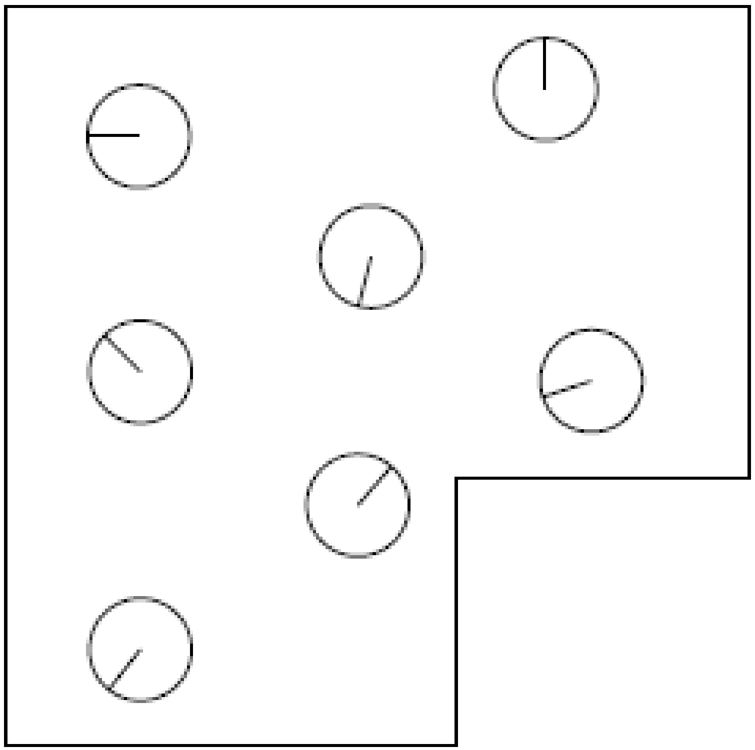
1. initialize particles
2. apply motion model
3. weight particles (sensor model)
4. resample according to weight

Recap: Particle Filter Localization



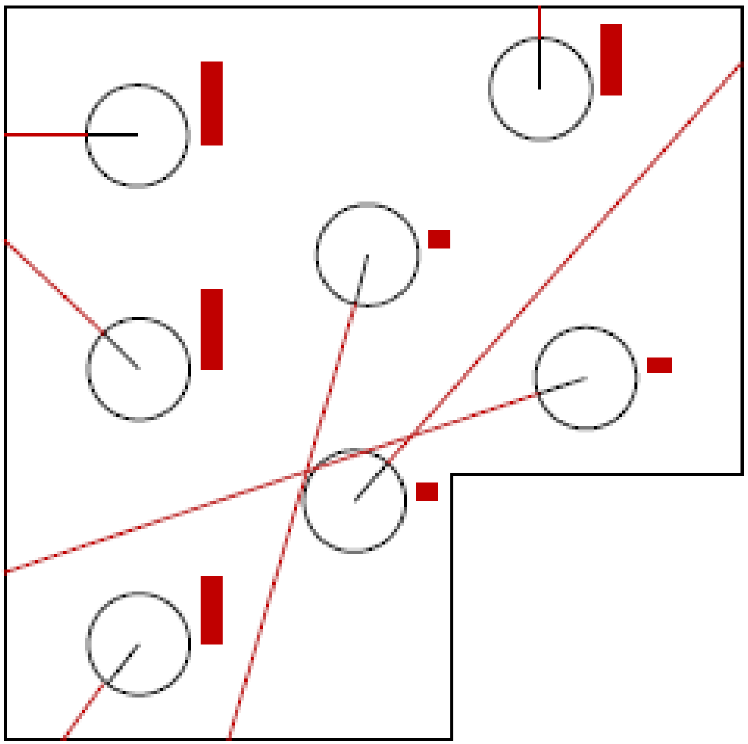
1. initialize particles
2. apply motion model
3. weight particles (sensor model)
4. resample according to weight

Recap: Particle Filter Localization



1. initialize particles
2. apply motion model
3. weight particles (sensor model)
4. resample according to weight

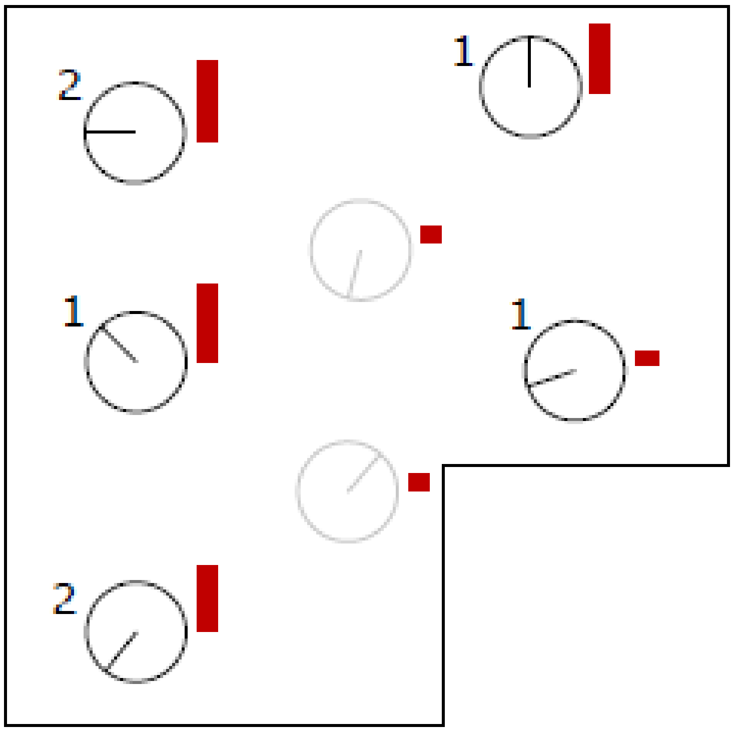
Recap: Particle Filter Localization



Actual measurement:

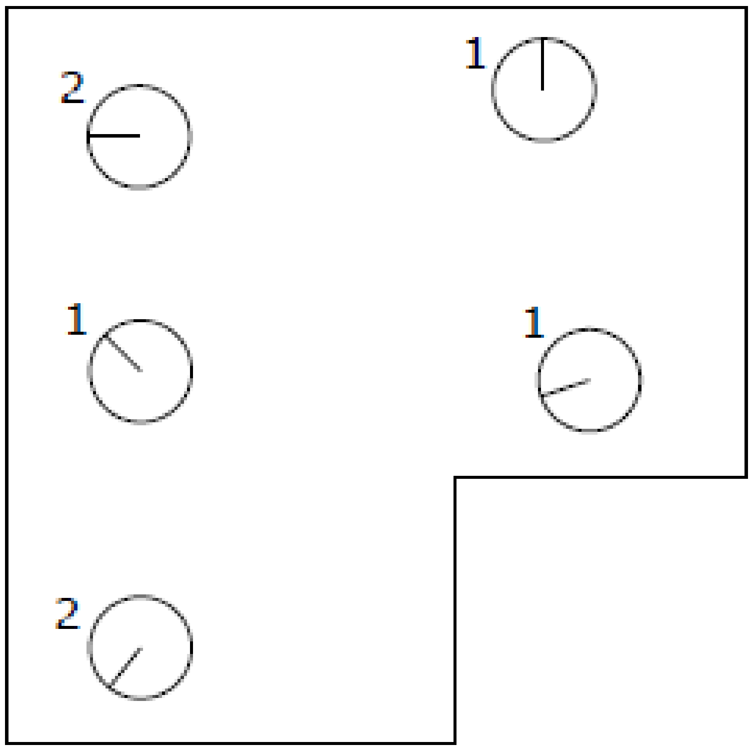
1. initialize particles
2. apply motion model
3. weight particles (sensor model)
4. resample according to weight

Recap: Particle Filter Localization



1. initialize particles
2. apply motion model
3. weight particles (sensor model)
4. resample according to weight

Recap: Particle Filter Localization



1. initialize particles
2. apply motion model
3. weight particles (sensor model)
4. resample according to weight

Localization vs. SLAM

- A particle filter can be used to solve both problems
- Localization: state space $< x, y, \theta >$
- SLAM: state space $< x, y, \theta, map >$
  - for landmark maps = $< l_1, l_2, \ldots, l_m >$
  - for grid maps = $< c_{11}, c_{12}, \ldots, c_{1n}, c_{21}, \ldots, c_{nm} >$

- **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.

Rao-Blackwellization

- Factorization to exploit dependencies between variables:

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

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

- It comes from the Rao-Blackwell theorem

Factored Posterior (Landmarks)

poses map observations & movements

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

Factorization first introduced by Kevin Murphy at NIPS'99

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)

poses map observations & movements

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

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

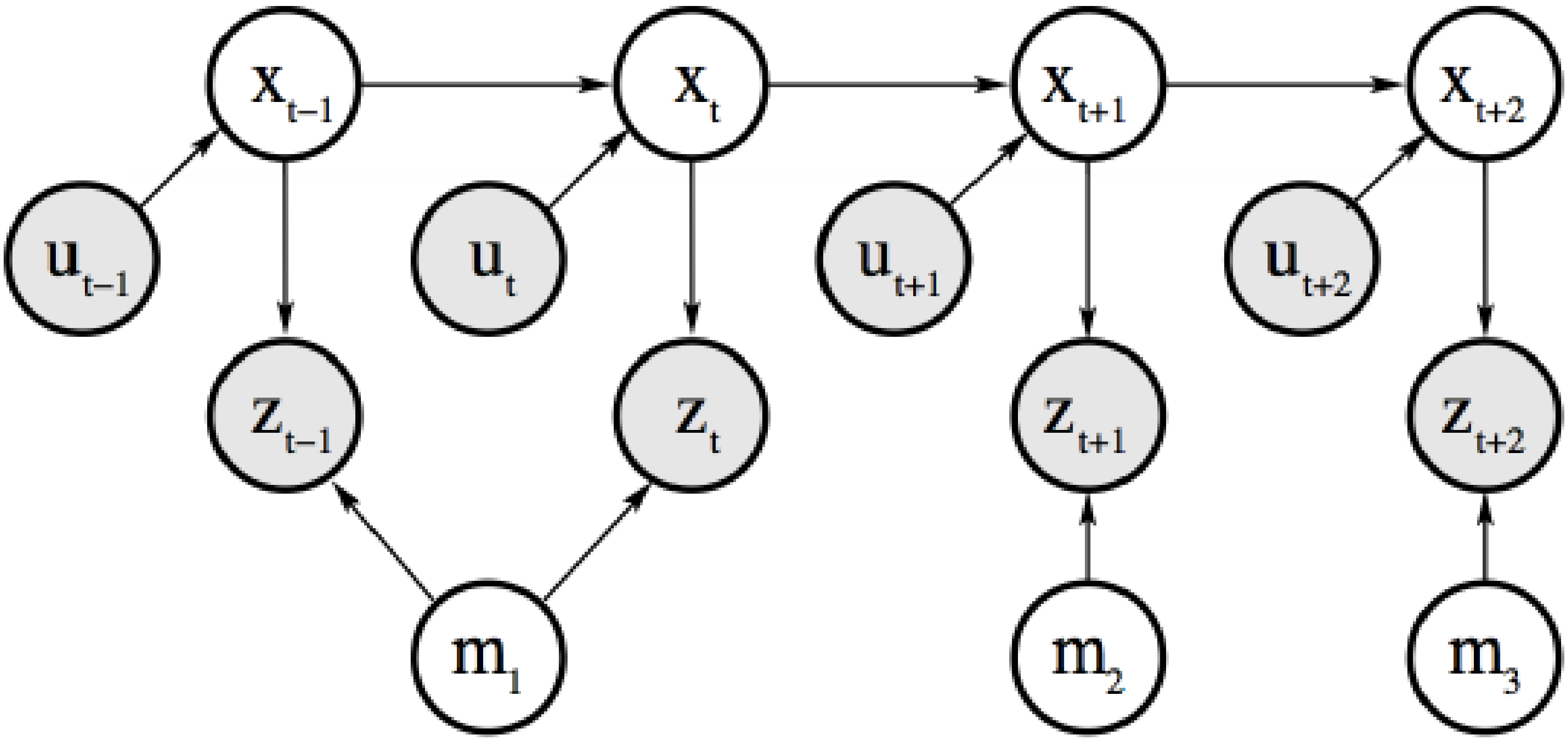
SLAM posterior

Robot path posterior

landmark positions

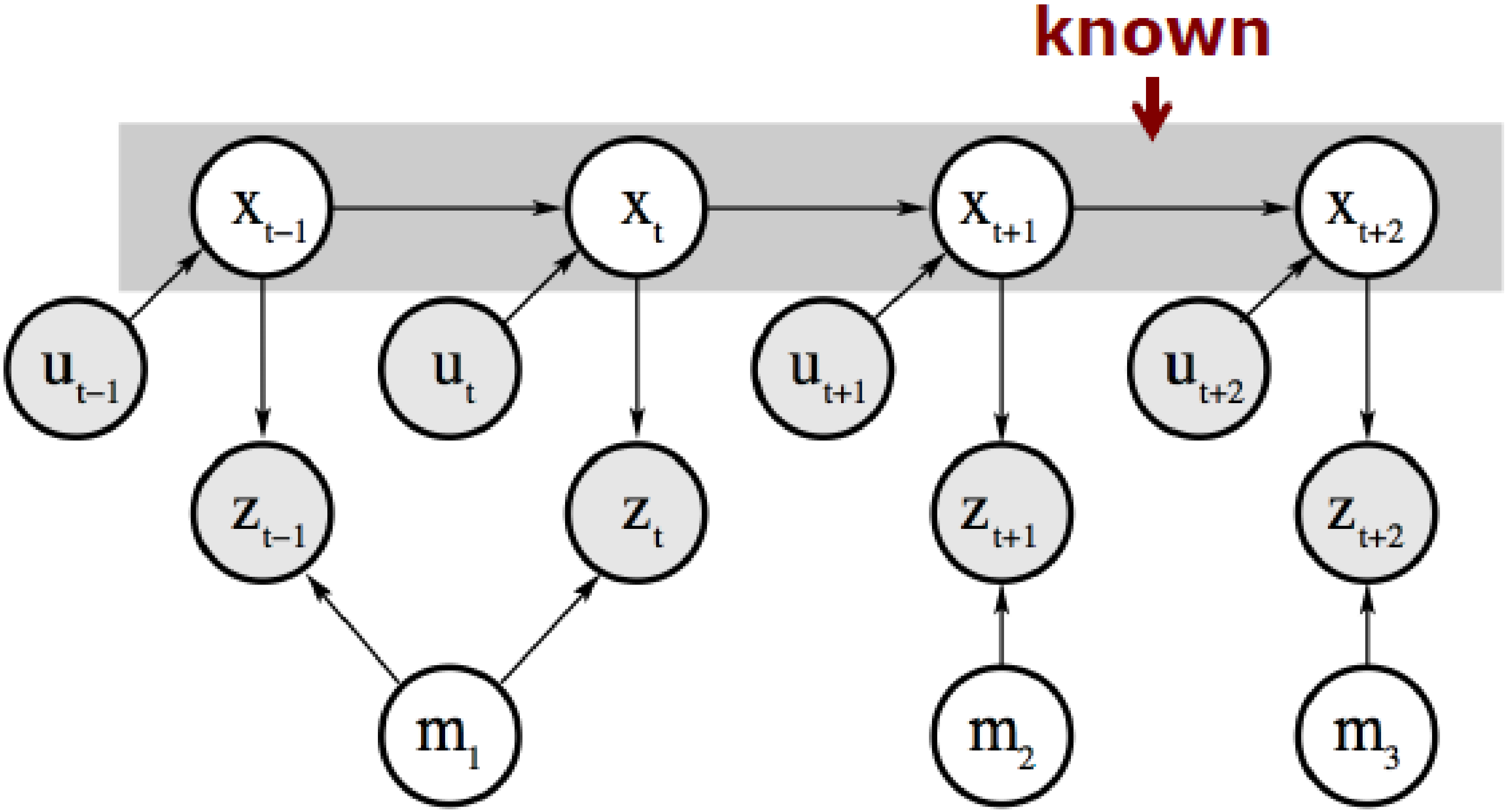
Does this help to solve the problem?

Revisit the Graphical Model

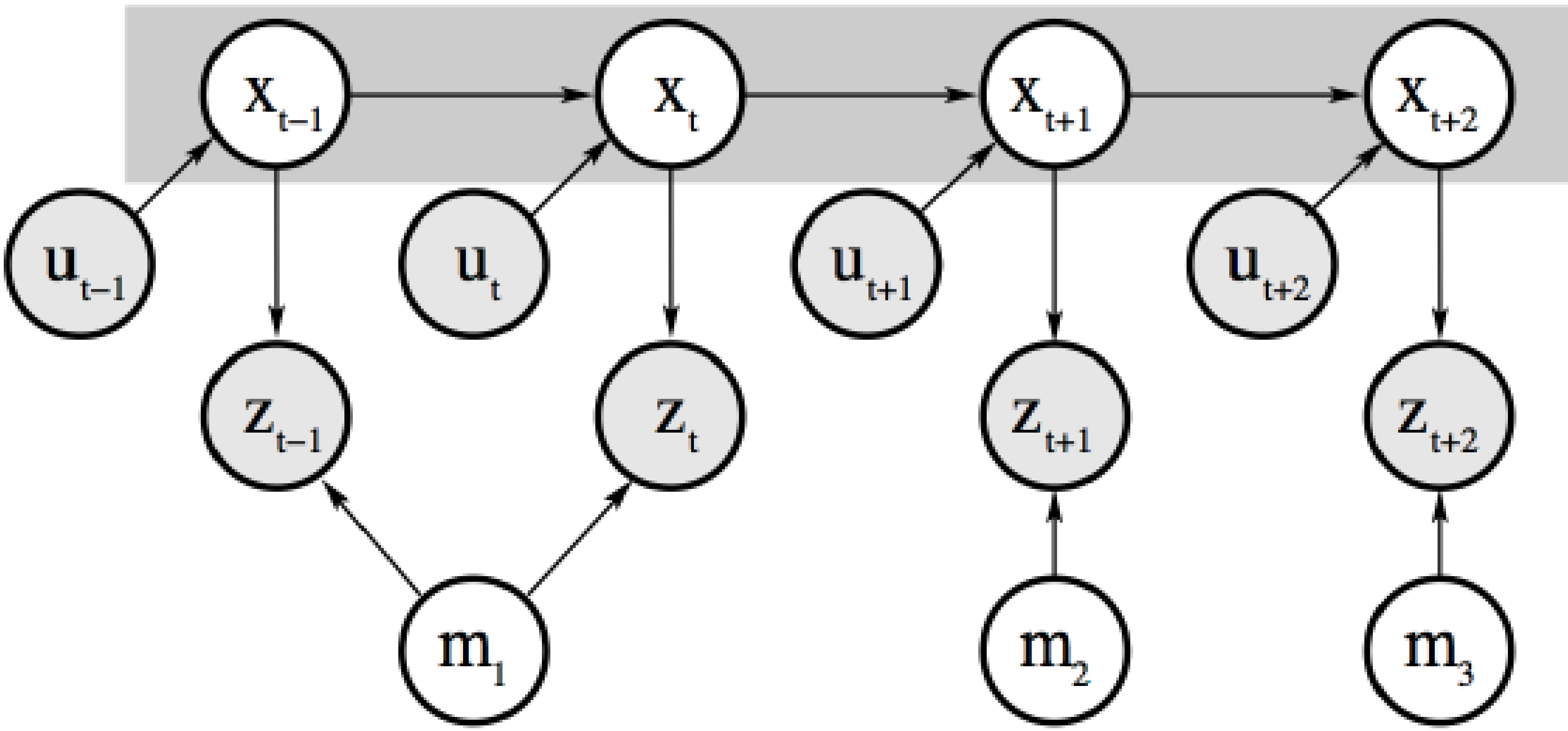


Courtesy: Thrun, Burgard, Fox

Revisit the Graphical Model

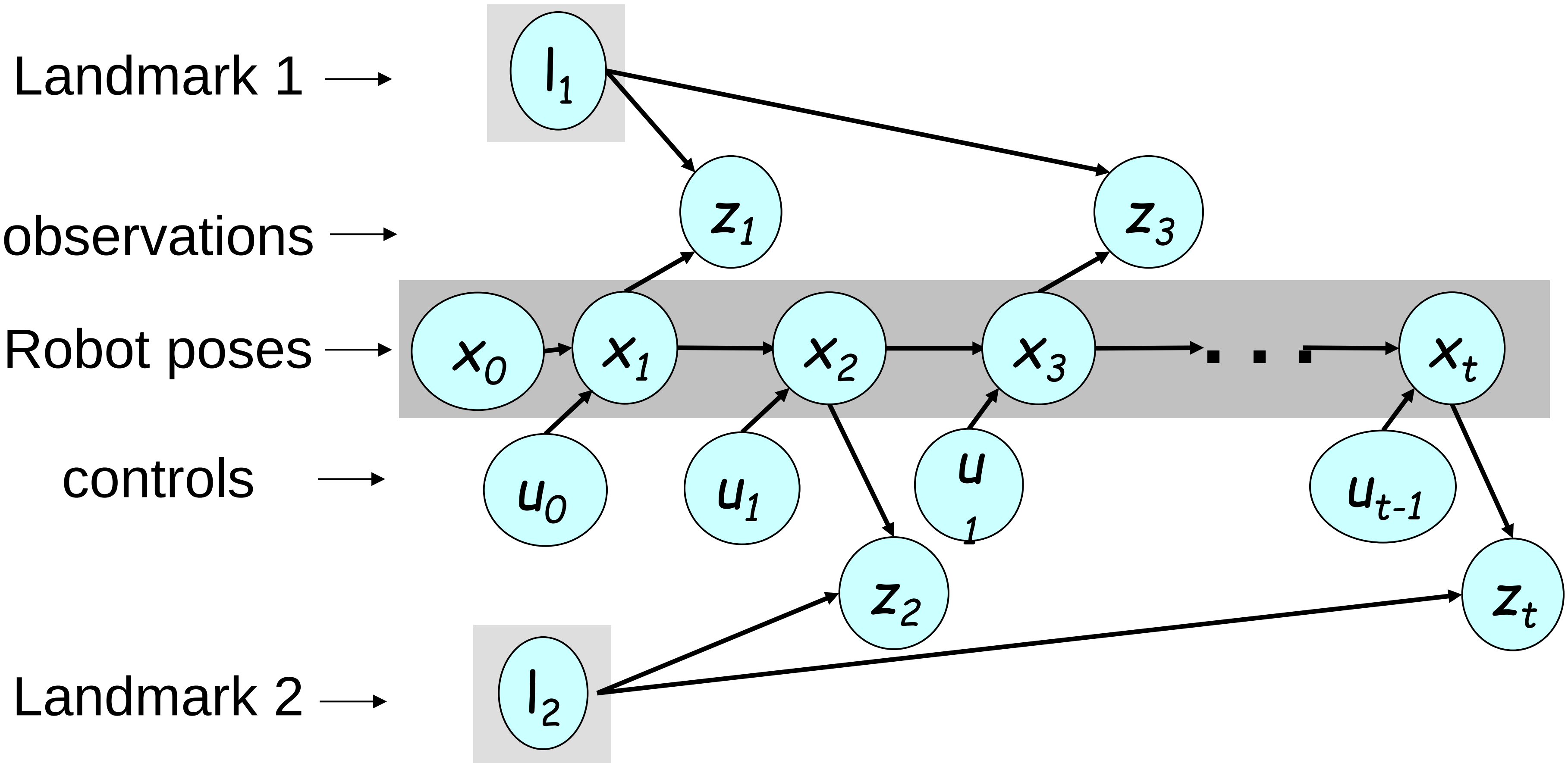


Landmarks are Conditionally Independent Given the Poses



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

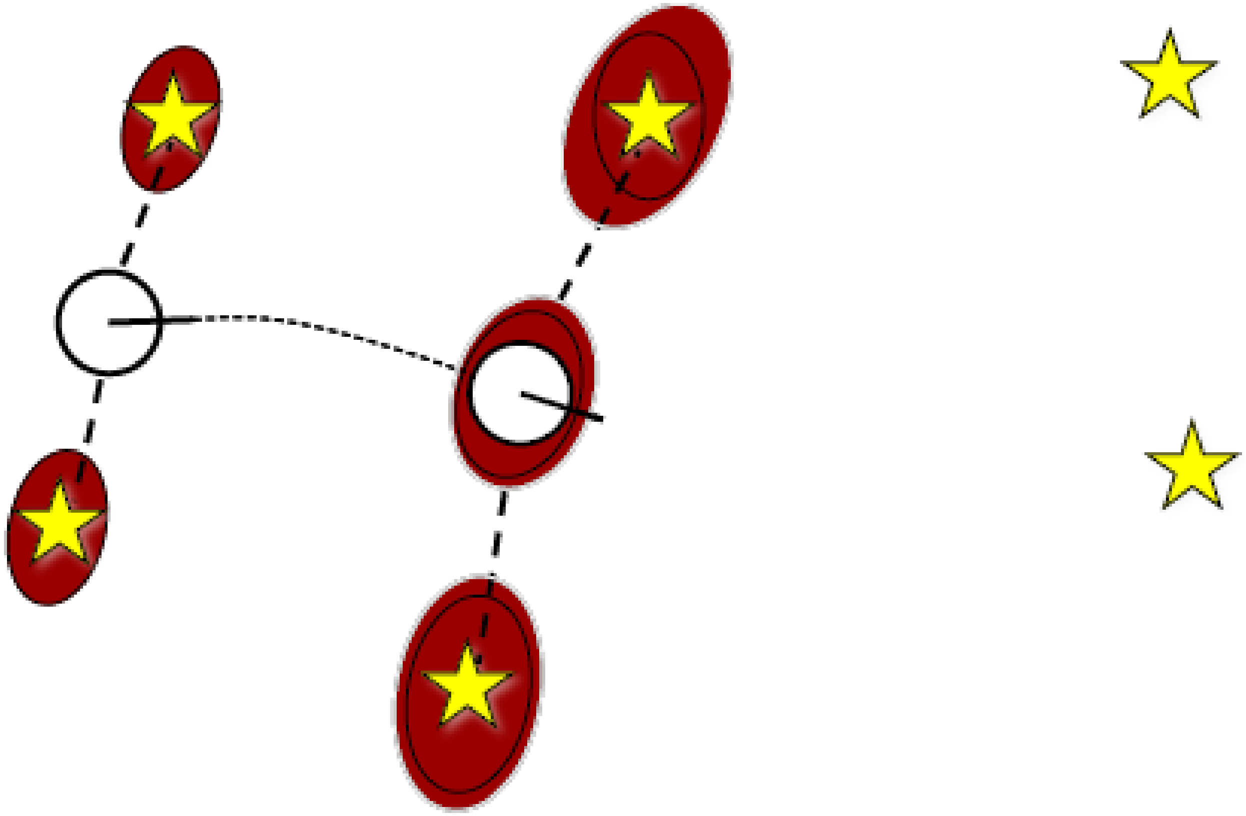
Mapping using Landmarks



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

Remember: Landmarks Correlated

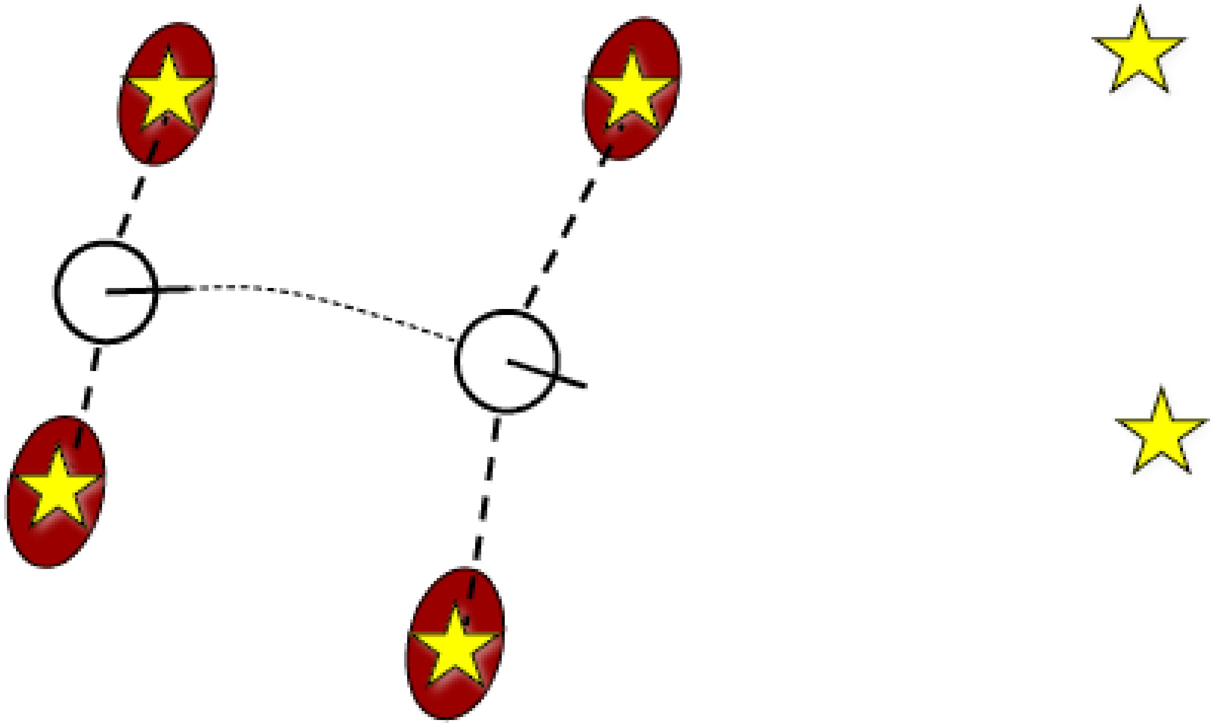
SLAM: robot path and map are both **unknown!**



Robot path error correlates errors in the map

After Factorization

For estimating landmarks: robot path **known!**



Landmarks are not correlated

Factored Posterior

$$
\begin{aligned}
& p(x_{1:t}, l_{1:m} \mid z_{1:t}, u_{0:t-1}) \\
&= p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot p(l_{1:m} \mid x_{1:t}, z_{1:t}) \\
&= \underbrace{p(x_{1:t} \mid z_{1:t}, u_{0:t-1})}_{\text{Robot path posterior (localization problem)}} \cdot \prod_{i=1}^{M} \underbrace{p(l_i \mid x_{1:t}, z_{1:t})}_{\text{Conditionally independent landmark positions}}
\end{aligned}
$$

Robot path posterior (localization problem)

Conditionally independent landmark positions

Rao-Blackwellization for SLAM

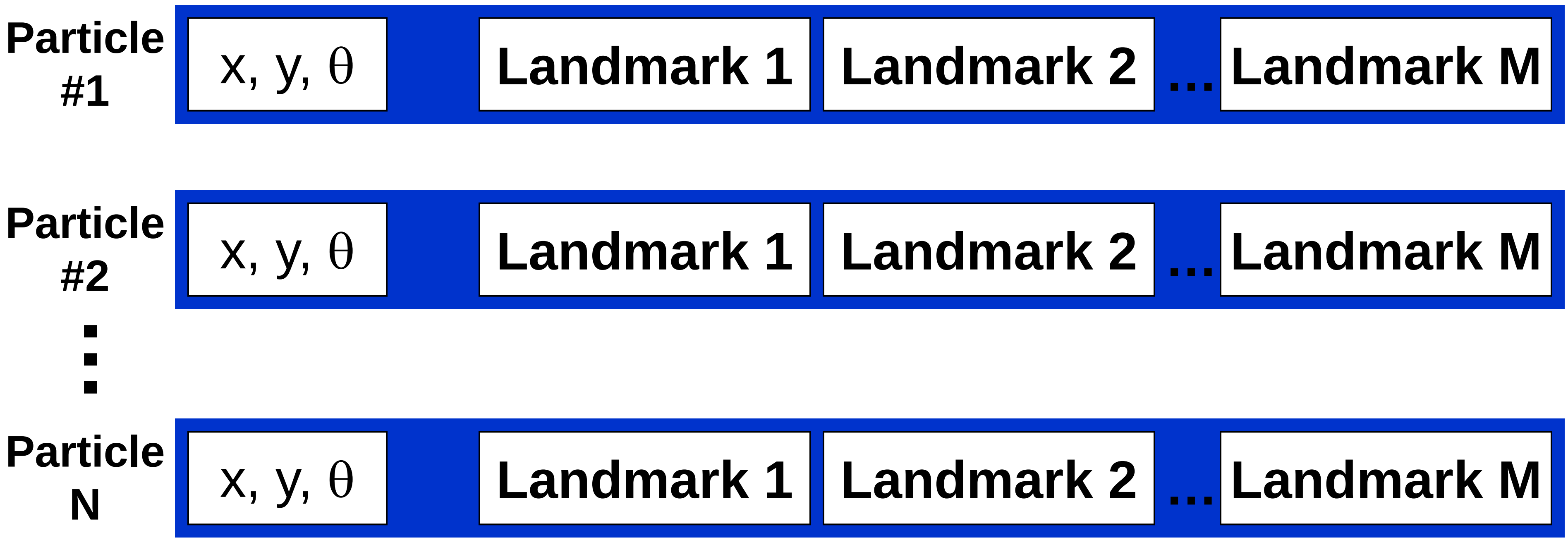
$$p(x_{1:t}, l_{1:m} \mid z_{1:t}, u_{0:t-1}) =$$

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

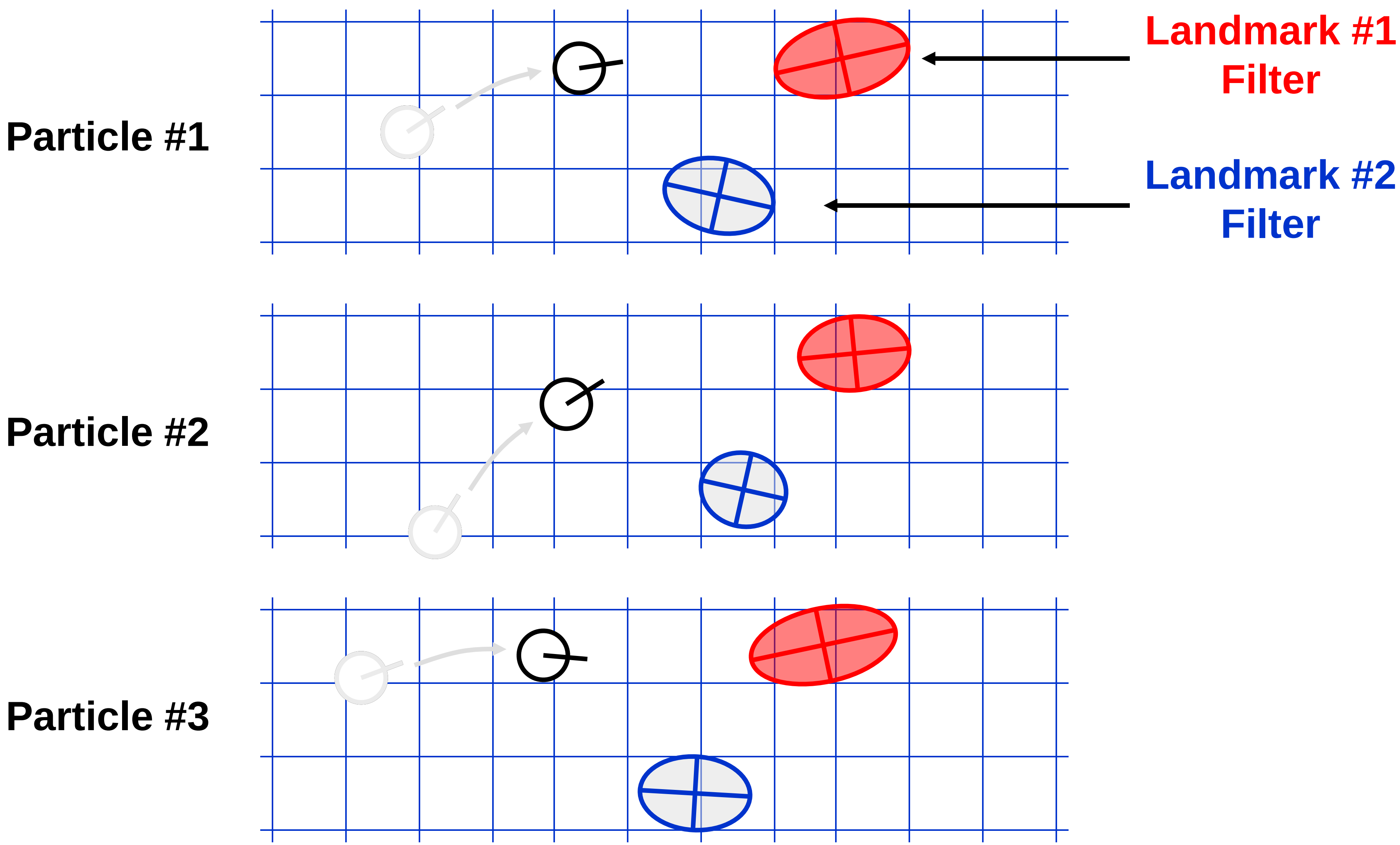
- Given that the second term can be computed efficiently, particle filtering becomes possible!

FastSLAM

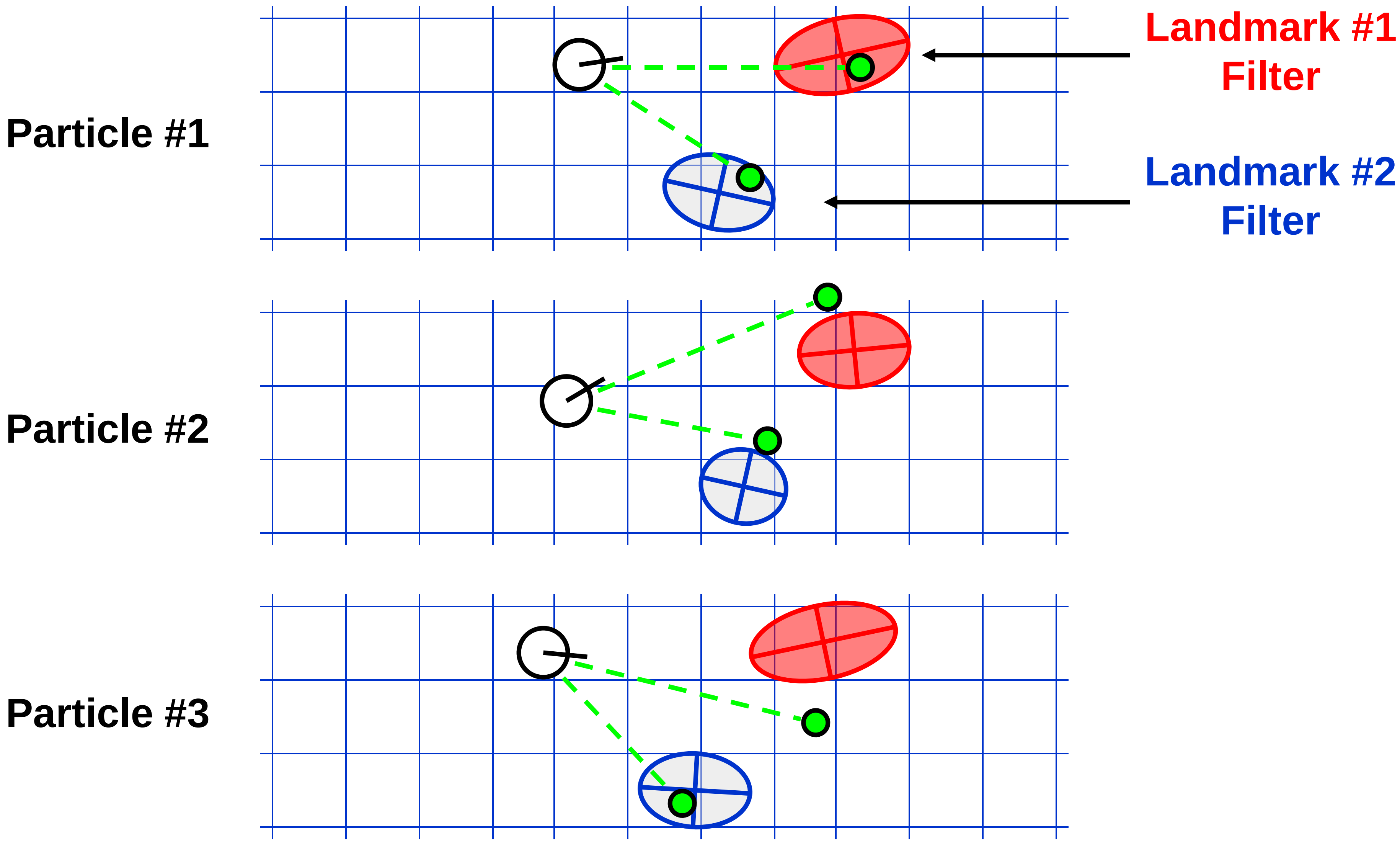
- Rao-Blackwellized particle filtering based on landmarks [Montemerlo et al., 2002]
- Each landmark is represented by a 2x2 Extended Kalman Filter (EKF)
- Each particle therefore has to maintain *M* EKFs



FastSLAM – Action Update

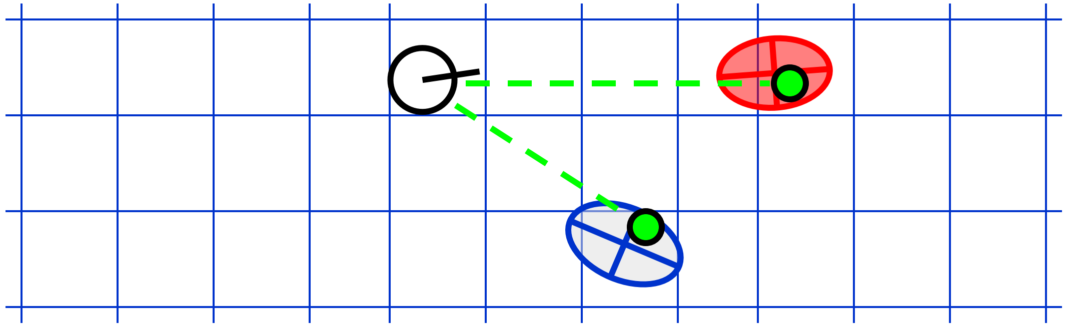


FastSLAM – Sensor Update



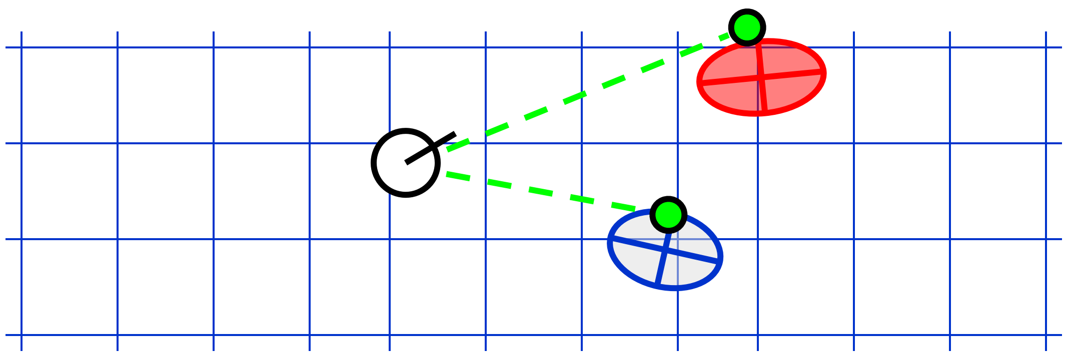
FastSLAM – Sensor Update

Particle #1



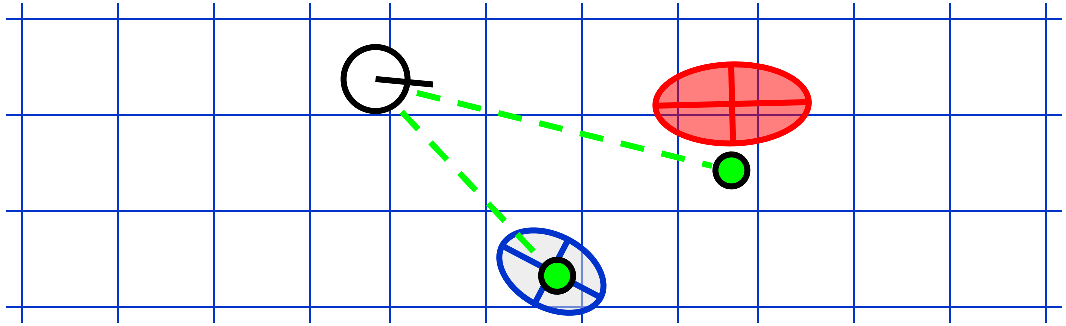
Weight = 0.8

Particle #2



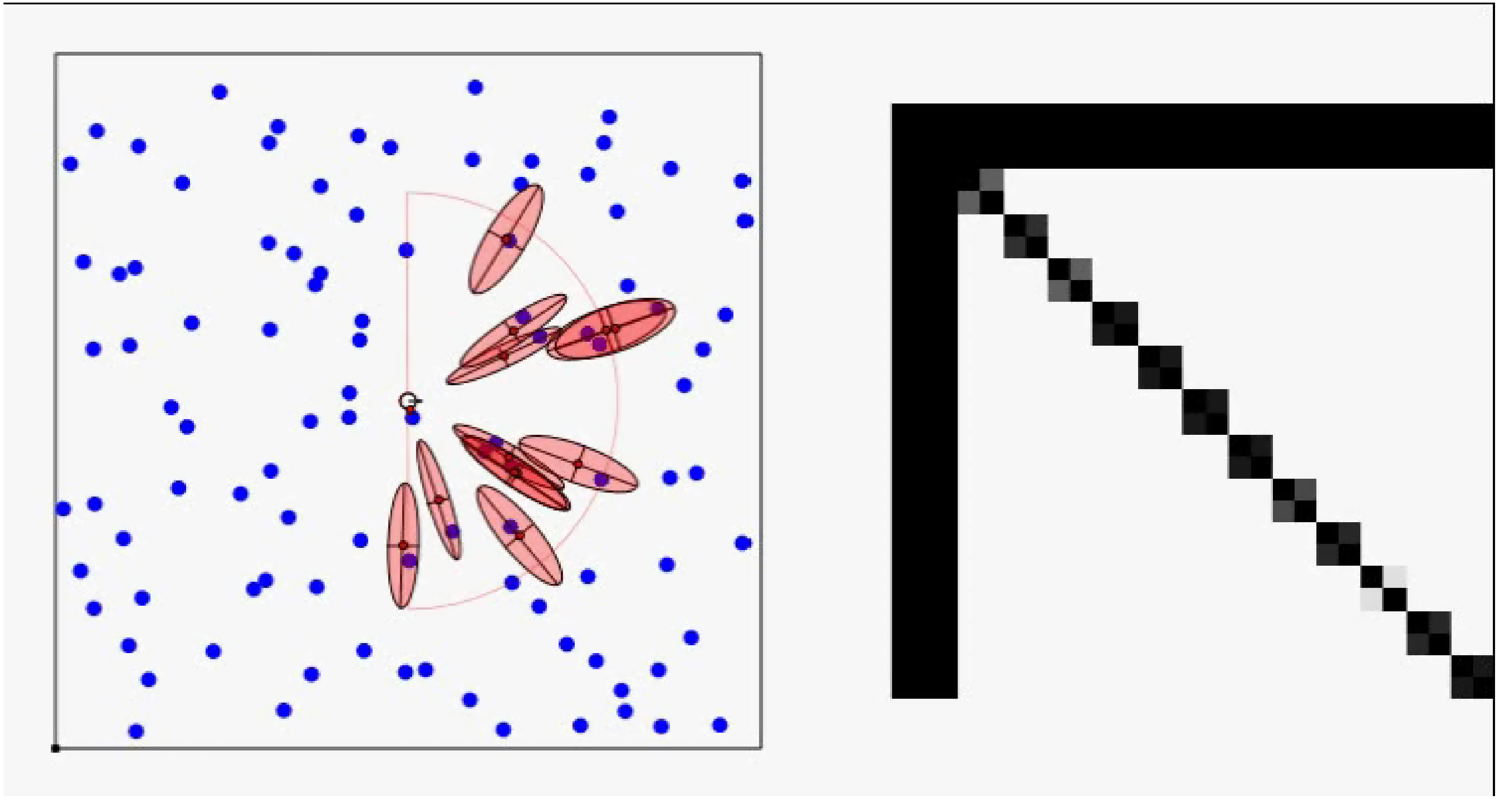
Weight = 0.4

Particle #3



Weight = 0.1

FastSLAM - Video



Michael Montemerlo *et al*. "Fastslam: A factored solution to the simultaneous localization and mapping problem." In Proceedings of the AAAI National Conference on Artificial Intelligence. 2002.

FastSLAM Complexity

- Update robot particles based on control u_{t-1}
- Incorporate observation z_t into Kalman filters
- Resample particle set

N = Number of particles
M = Number of map features

FastSLAM Complexity - Naive

- Update robot particles based on control u_{t-1} — $O(N)$
  Constant time per particle

- Incorporate observation z_t into Kalman filters — $O(N)$

- Resample particle set — $O(N \bullet M)$

N = Number of particles
M = Number of map features

$O(N \bullet M)$

FastSLAM Complexity – binary tree

- 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

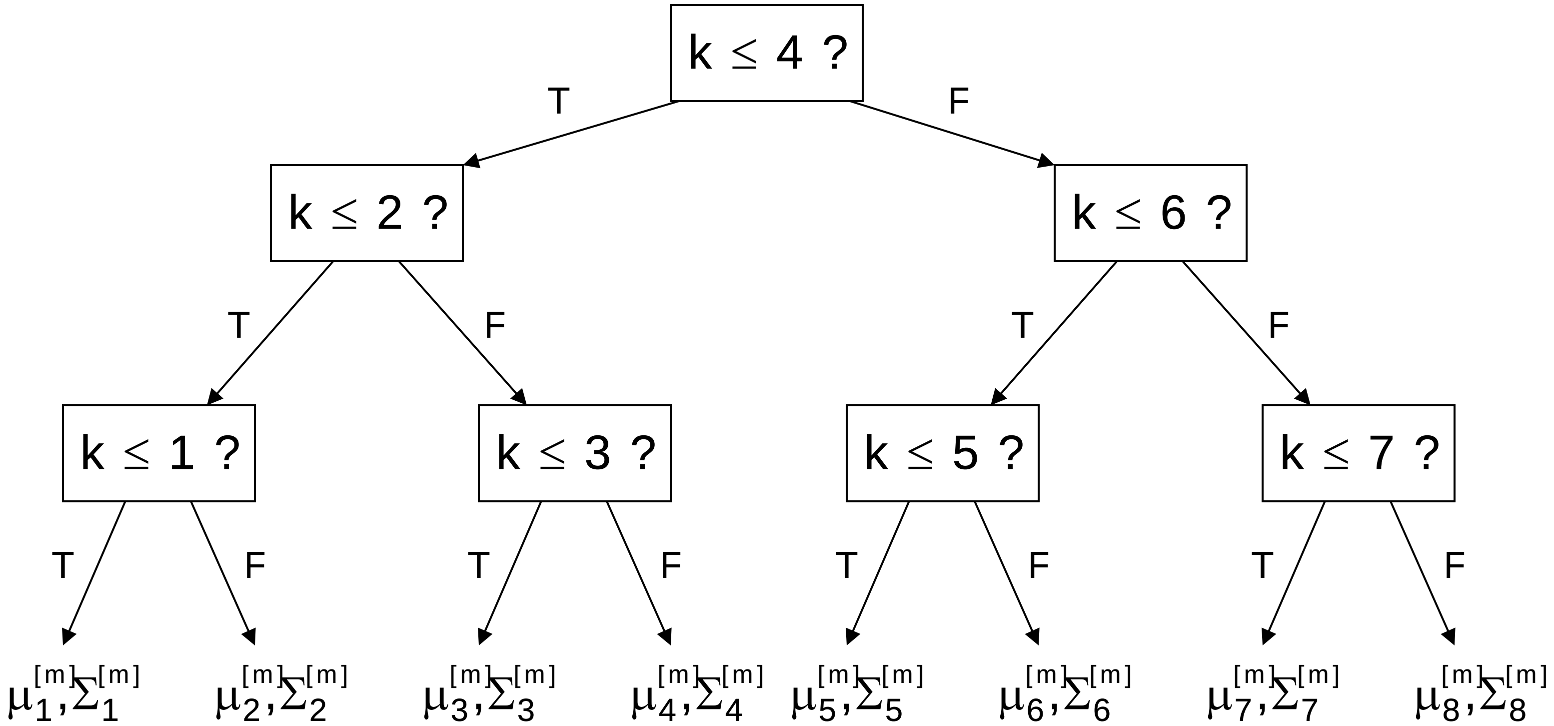
  $O(N)$
  Constant time per particle

N = Number of particles
M = Number of map features

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

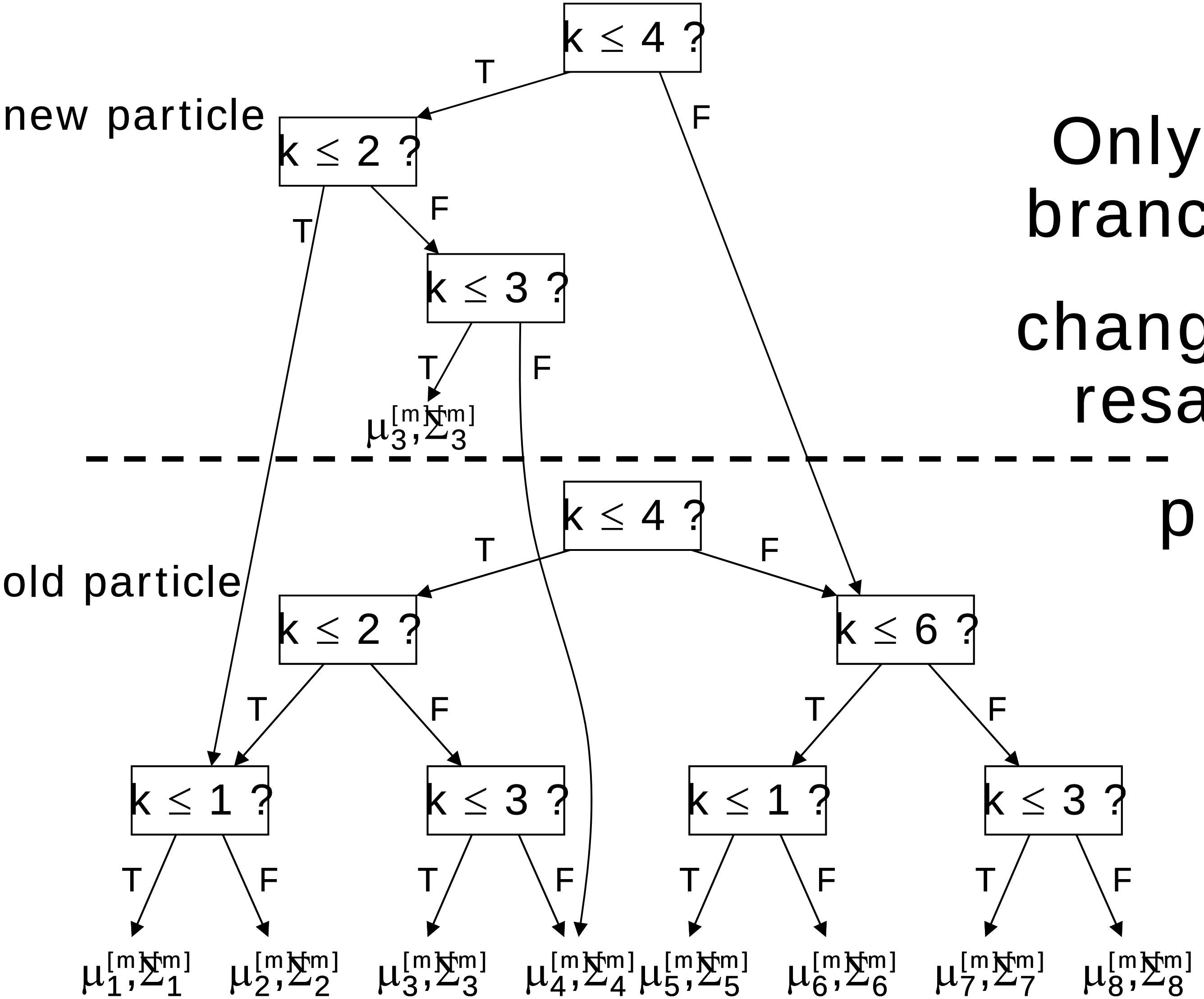
Log(M) Algorithm

Represent particle as tree of Kalman Filters



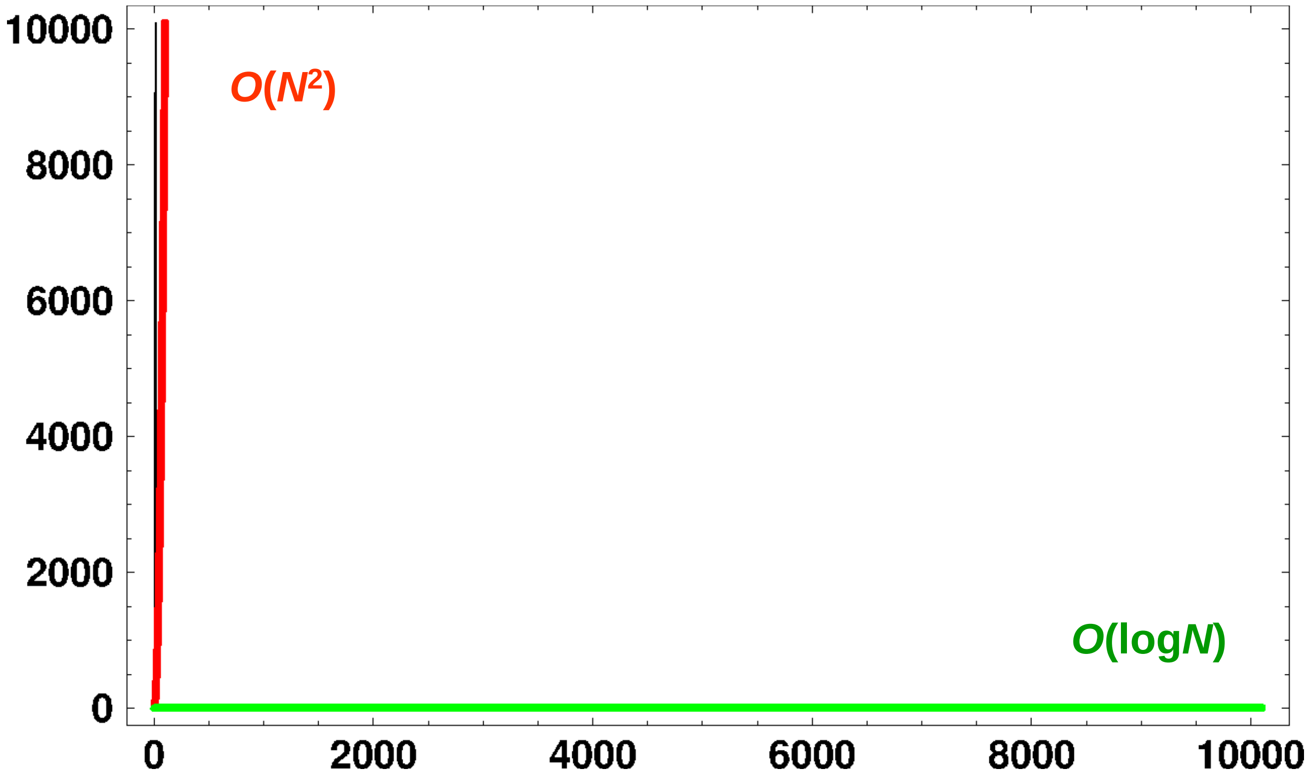
Courtesy Michael Montemerlo.

Log(M) Algorithm



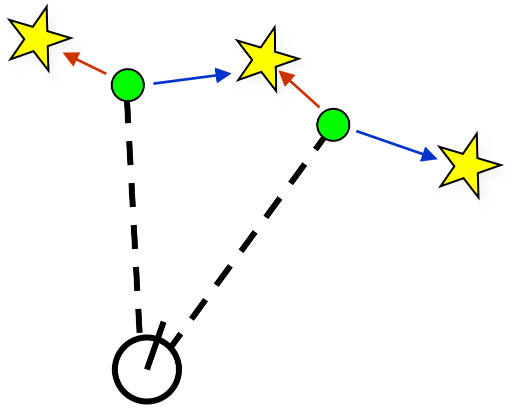
Only update branches that change during resampling phase

The importance of scaling



Data Association Problem

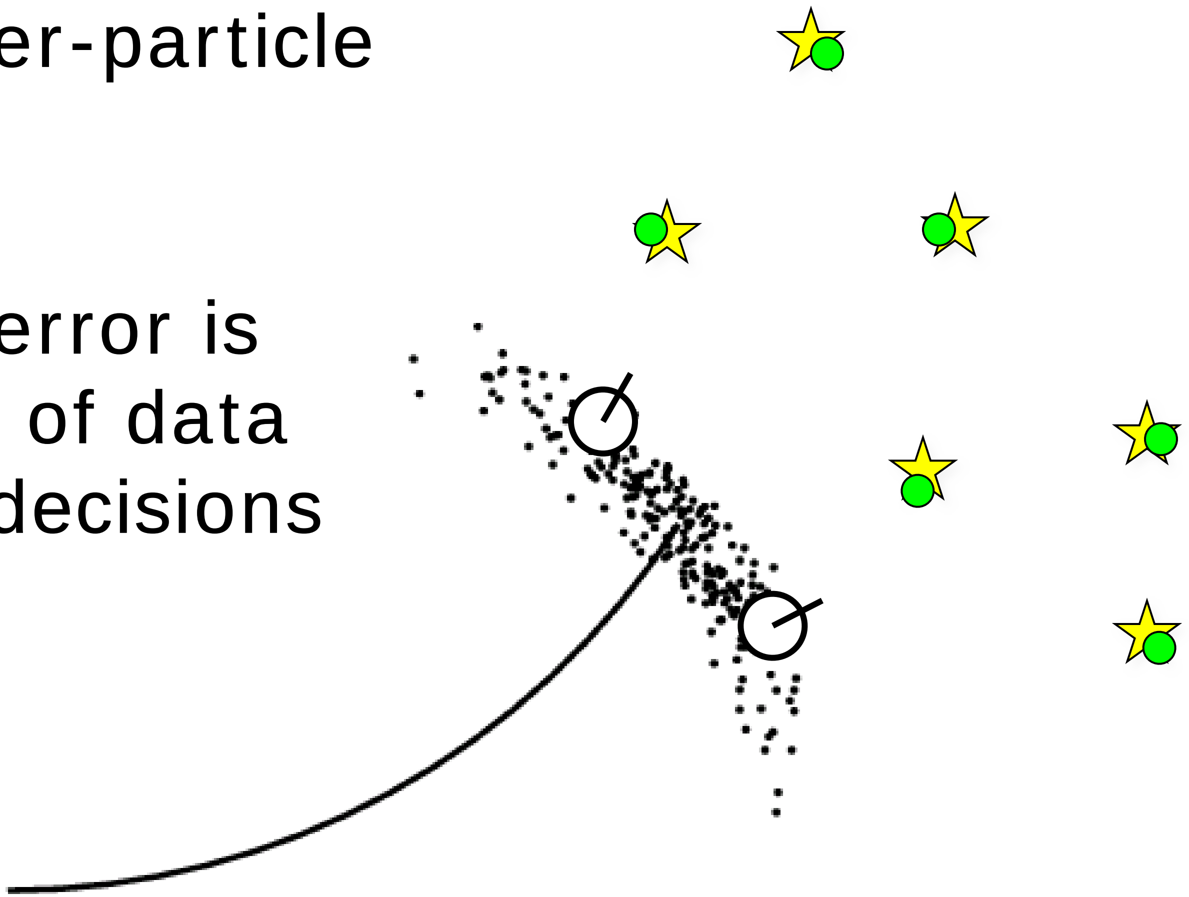
- Which observation belongs to which landmark?



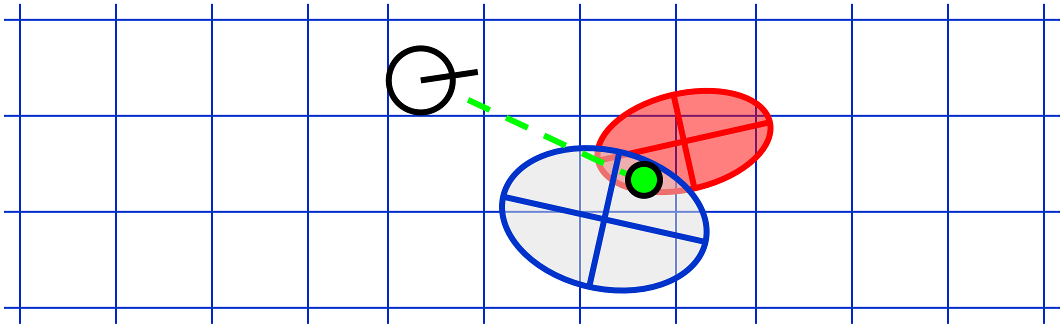
- A robust SLAM must consider possible data associations
- Potential data associations depend also on the pose of the robot

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



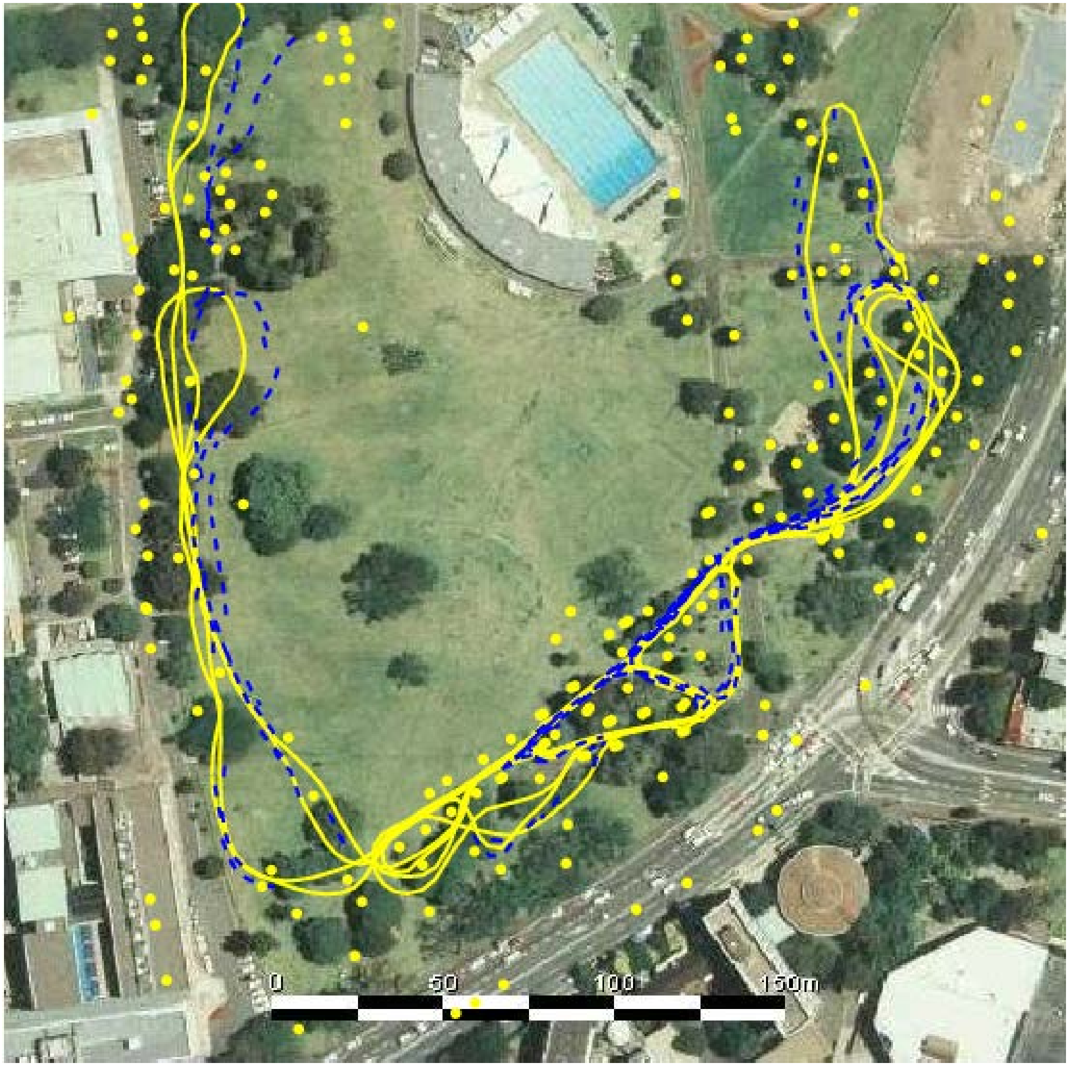
Was the observation generated by the red or the blue landmark?

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

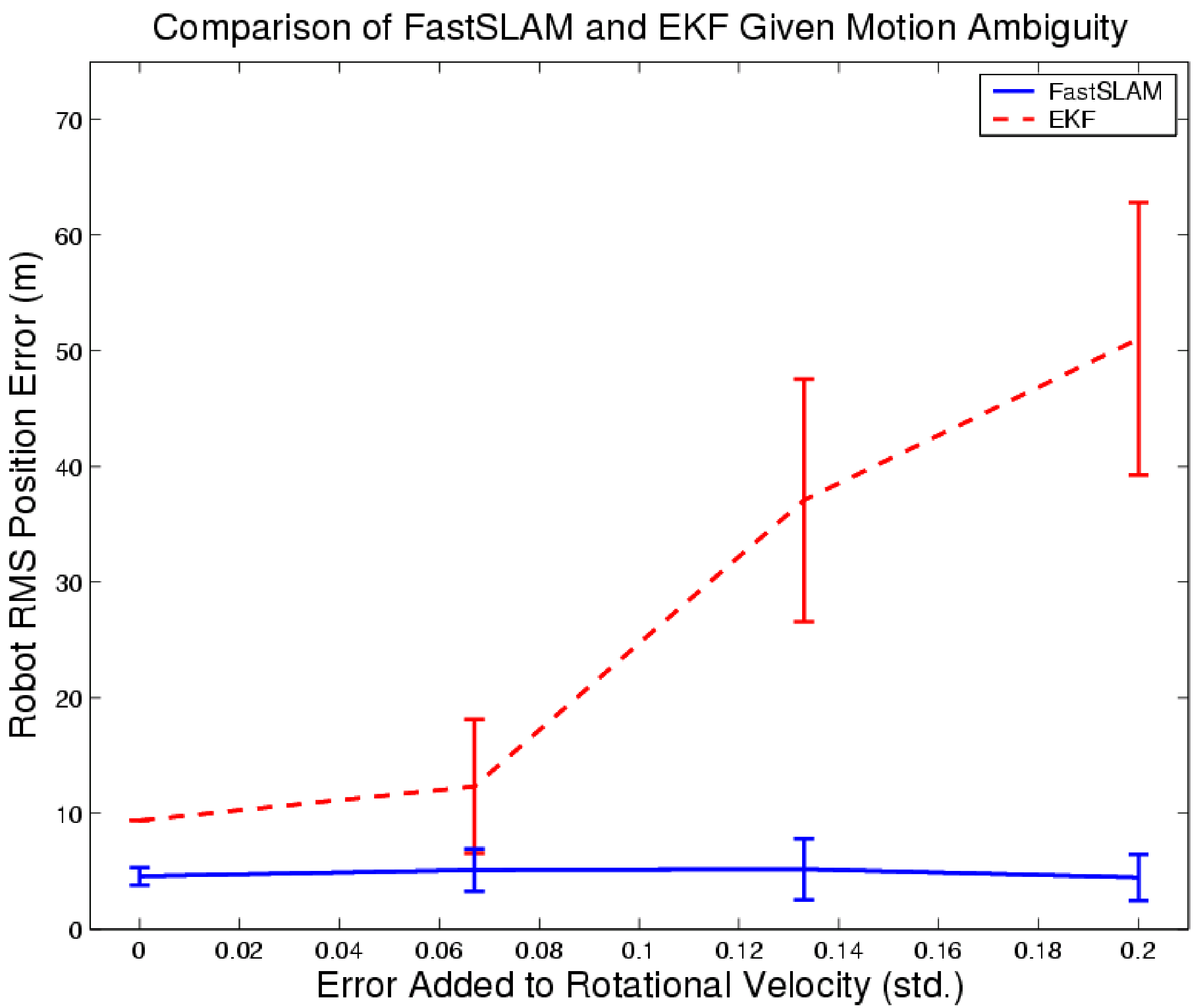
- 4 km traverse
- < 5 m RMS position error
- 100 particles



Dataset courtesy of University of Sydney

Blue = GPS
Yellow = FastSLAM

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 $O(N \log M)$

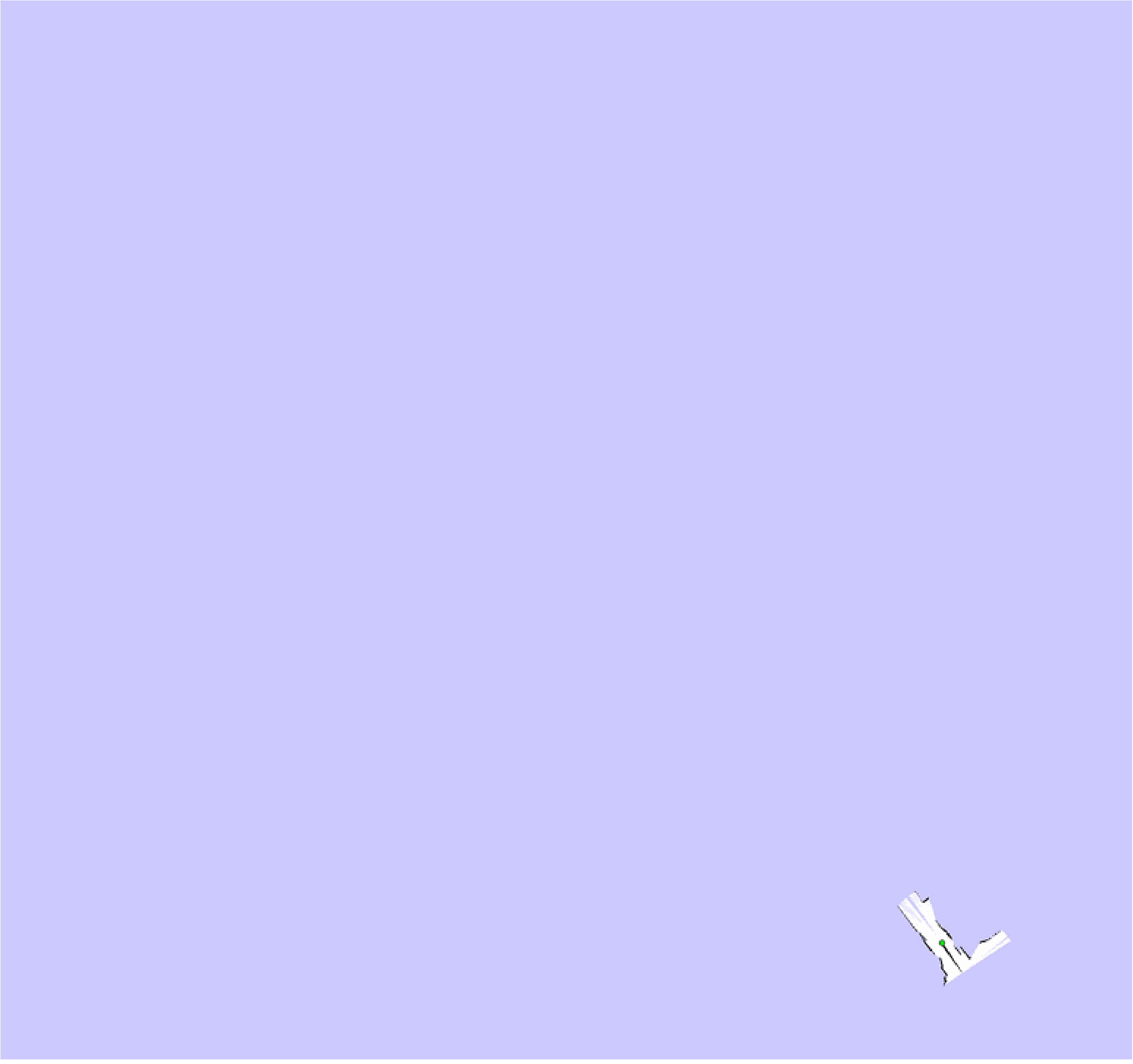
FastSLAM with Grid Maps

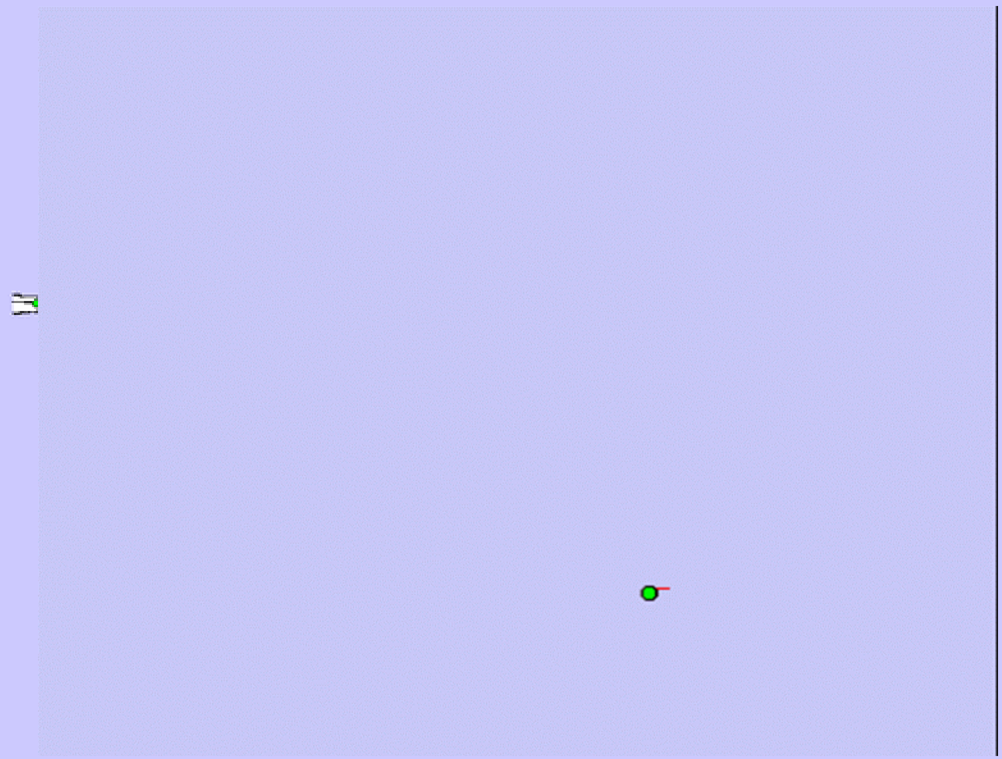
- Idea: Replace EKF Landmark map with occupancy grid map
- Q: Is this valid?

Mapping Abandoned Coal Mines

Mapping Abandoned Coal Mines

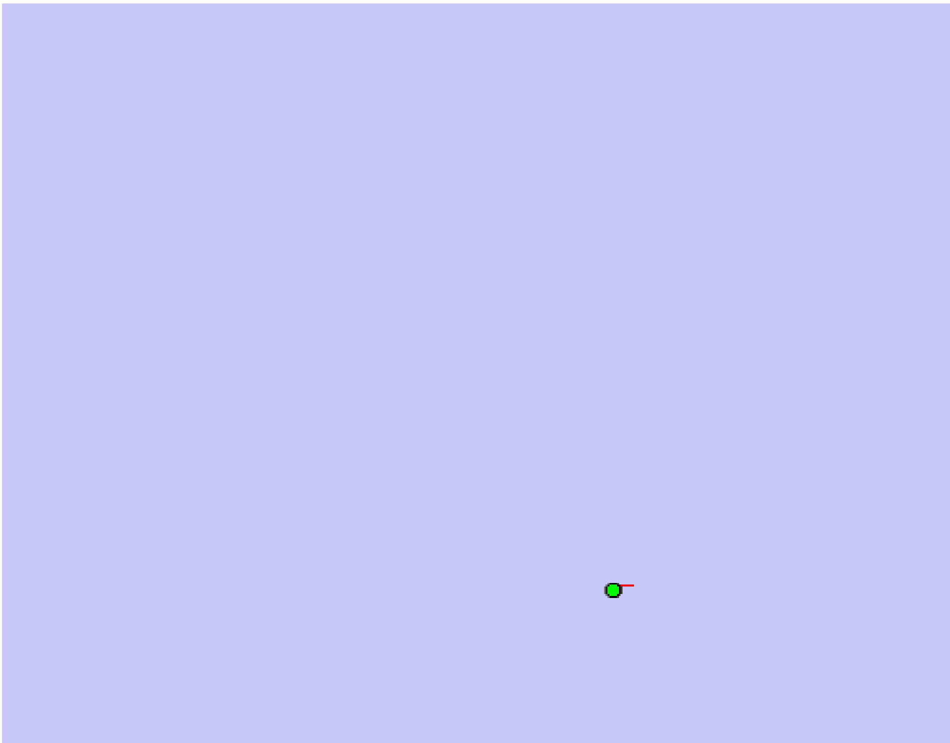
Particles in Mine Mapping



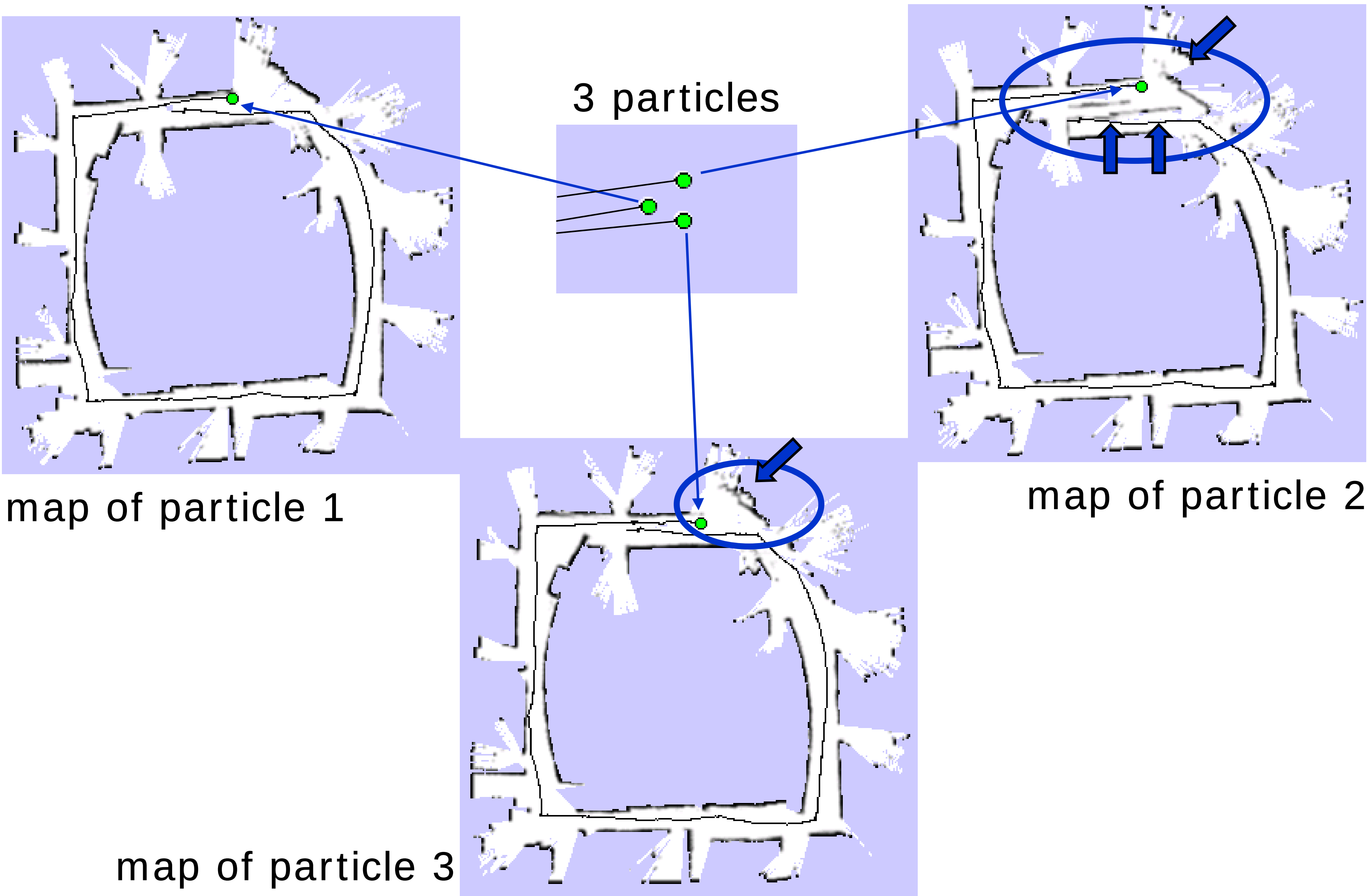


The Importance of Particle

without particles
with particles



FastSLAM with Grid Maps



Quality of 2D Maps

112 m

Outdoor Campus Map

- **30 particles**
- 250x250m^2
- 1.75 km (odometry)
- 20cm resolution during scan matching
- 30cm resolution in final map

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
- Update Complexity of $O(N \log M)$