Cognitive Robotics

10. Grid-Based FastSLAM

AbdElMoniem Bayoumi, PhD

Spring 2022

FastSLAM

- Rao-Blackwellization: Model the robot’s path by sampling and compute the map given the robot poses
- No uncertainty about the robot pose
- Each particle has its own map
- Last lecture: feature-based FastSLAM
- Today: Use the ideas of FastSLAM to build grid maps

Recap: Rao-Blackwellization for SLAM

Factorization of the SLAM posterior

poses map observations movements

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

[Murphy, 1999]

Recap: Rao-Blackwellization for SLAM

Factorization of the SLAM posterior

poses map observations movements

$$p(x_{0:t}, m \mid z_{1:t}, u_{1:t})$$
$$= \underbrace{p(x_{0:t} \mid z_{1:t}, u_{1:t})}_{\text{path posterior (particle filter)}} \; \underbrace{p(m \mid x_{1:t}, z_{1:t})}_{\text{map posterior (given the path)}}$$

path posterior
(particle filter)

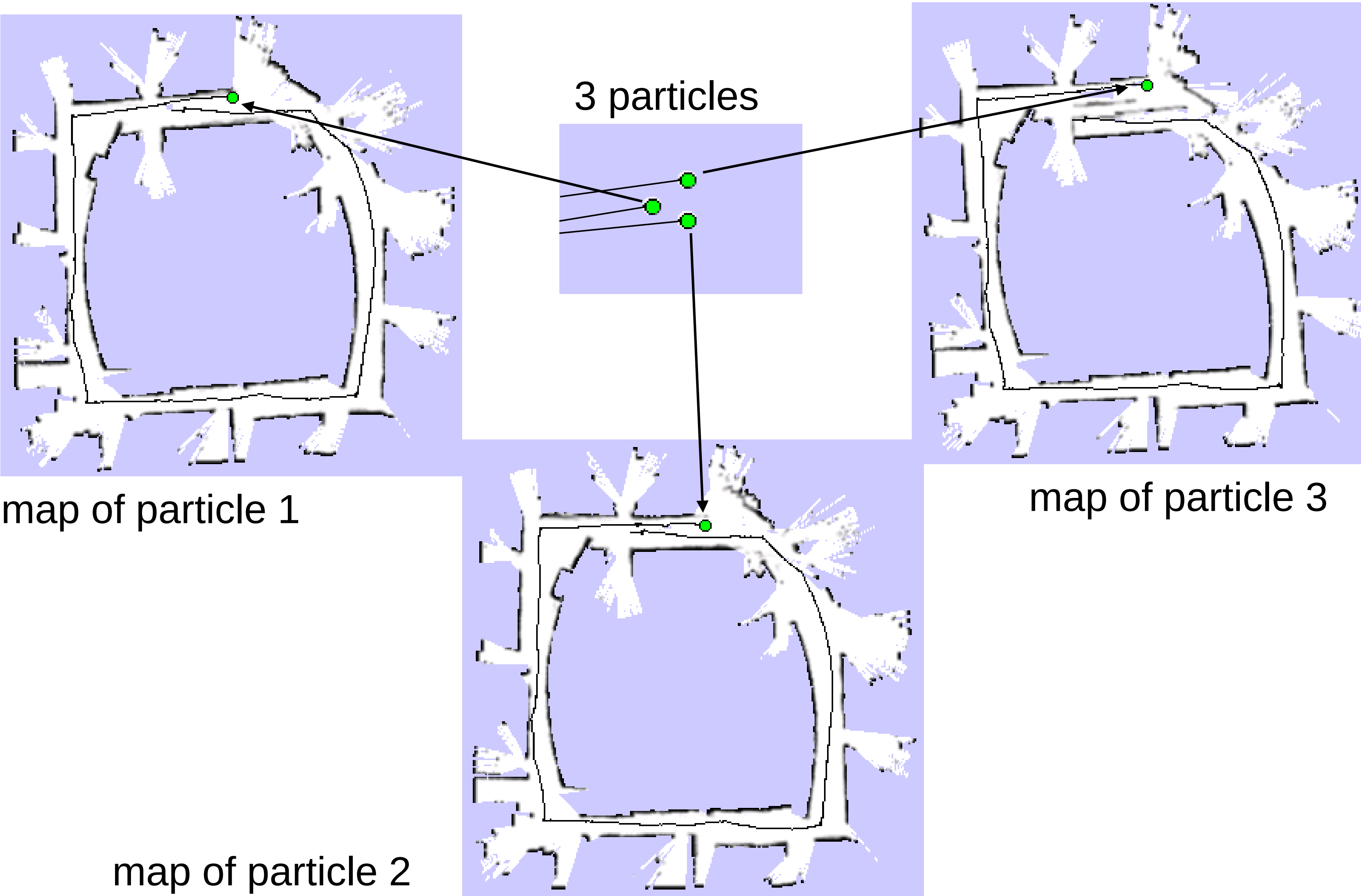
map posterior
(given the path)

[Murphy, 1999]

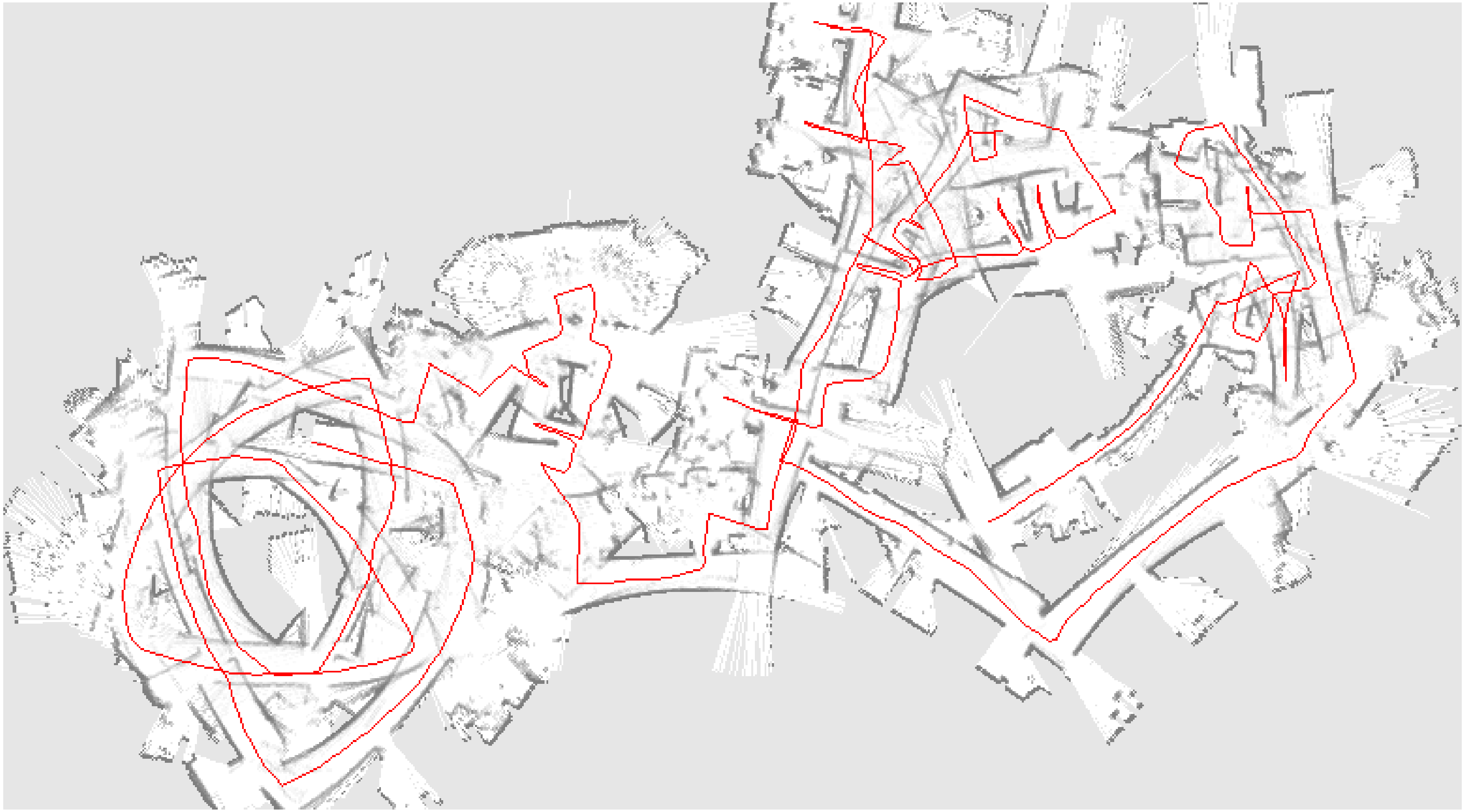
Grid-Based Mapping with Rao-Blackwellized Particle Filters

- Each particle represents a possible trajectory of the robot
- Each particle maintains its own map
- Each particle updates its map based on "mapping with known poses"
- Grid-based FastSLAM uses parts of the MCL and mapping algorithms

Particle Filter Example



Performance of Grid-Based FastSLAM 1.0



Problem

- Too many samples are needed to sufficiently model the motion noise
- Increasing the number of samples is difficult as each map is quite large

- **Idea:** Improve the pose estimate **before** applying the particle filter

Recap: Pose Correction Using Scan Matching

Maximize the likelihood of the **current** pose relative to the **previous** pose and map

$$x_t^* = \underset{x_t}{\operatorname{argmax}} \left\{ p(z_t \mid x_t, m_{t-1}) \, p(x_t \mid u_t, x_{t-1}^*) \right\}$$

current measurement

robot motion

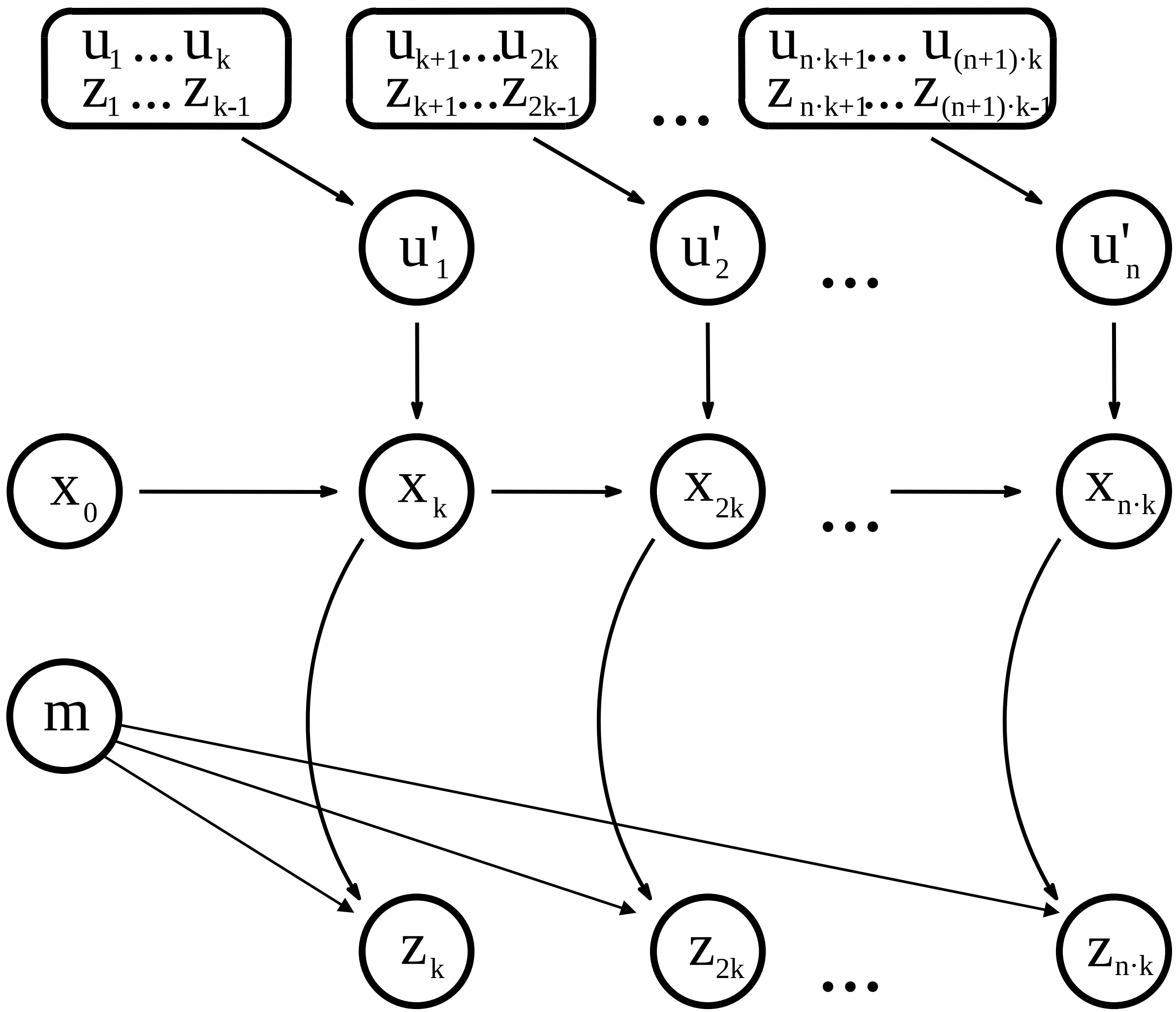
map constructed so far

Grid-Based FastSLAM with Improved Odometry

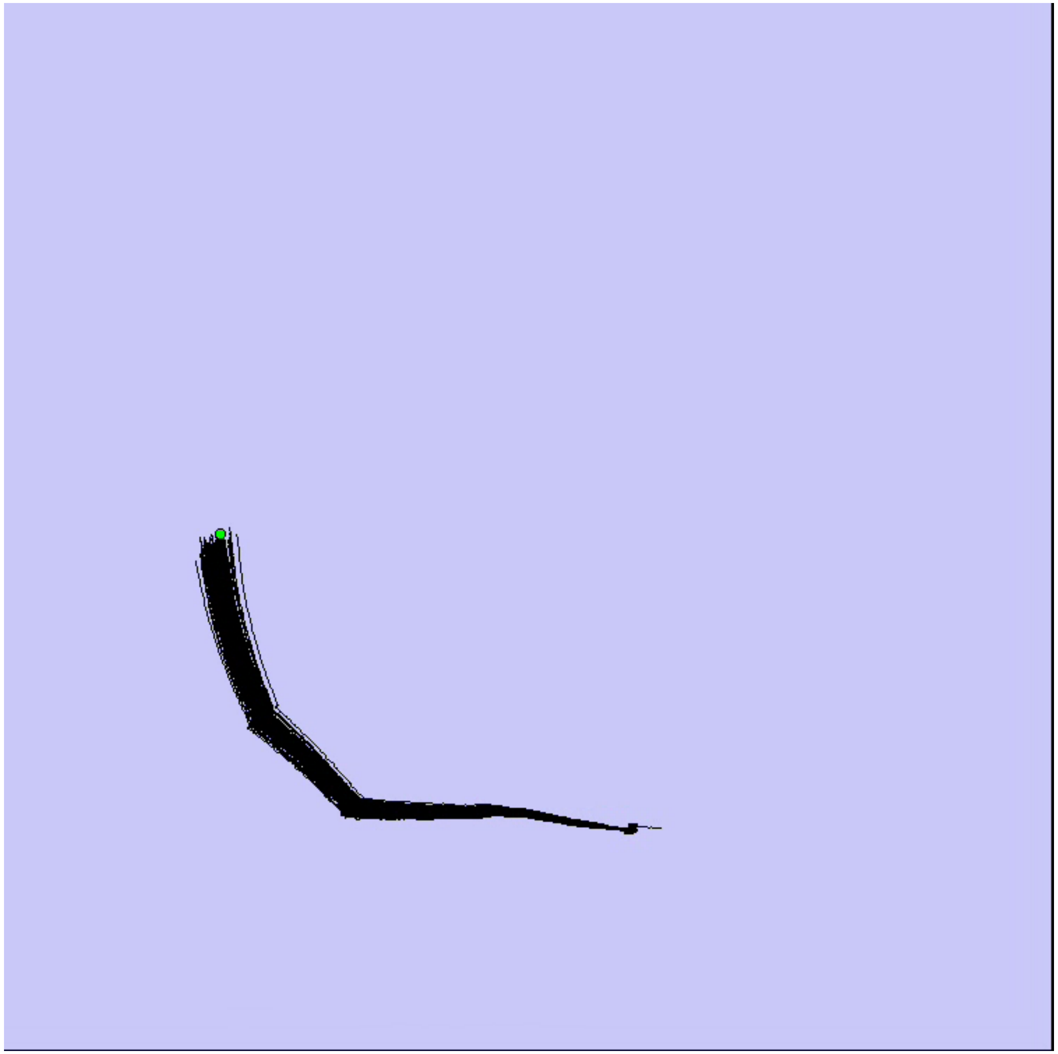
- Scan matching provides a **locally consistent** pose correction
- **Idea:** Pre-correct short odometry sequences using scan matching and use those as input to FastSLAM
- Fewer particles are needed, since the error in the input in smaller

[Hähnel et al., 2003]

Graphical Model for Mapping with Improved Odometry

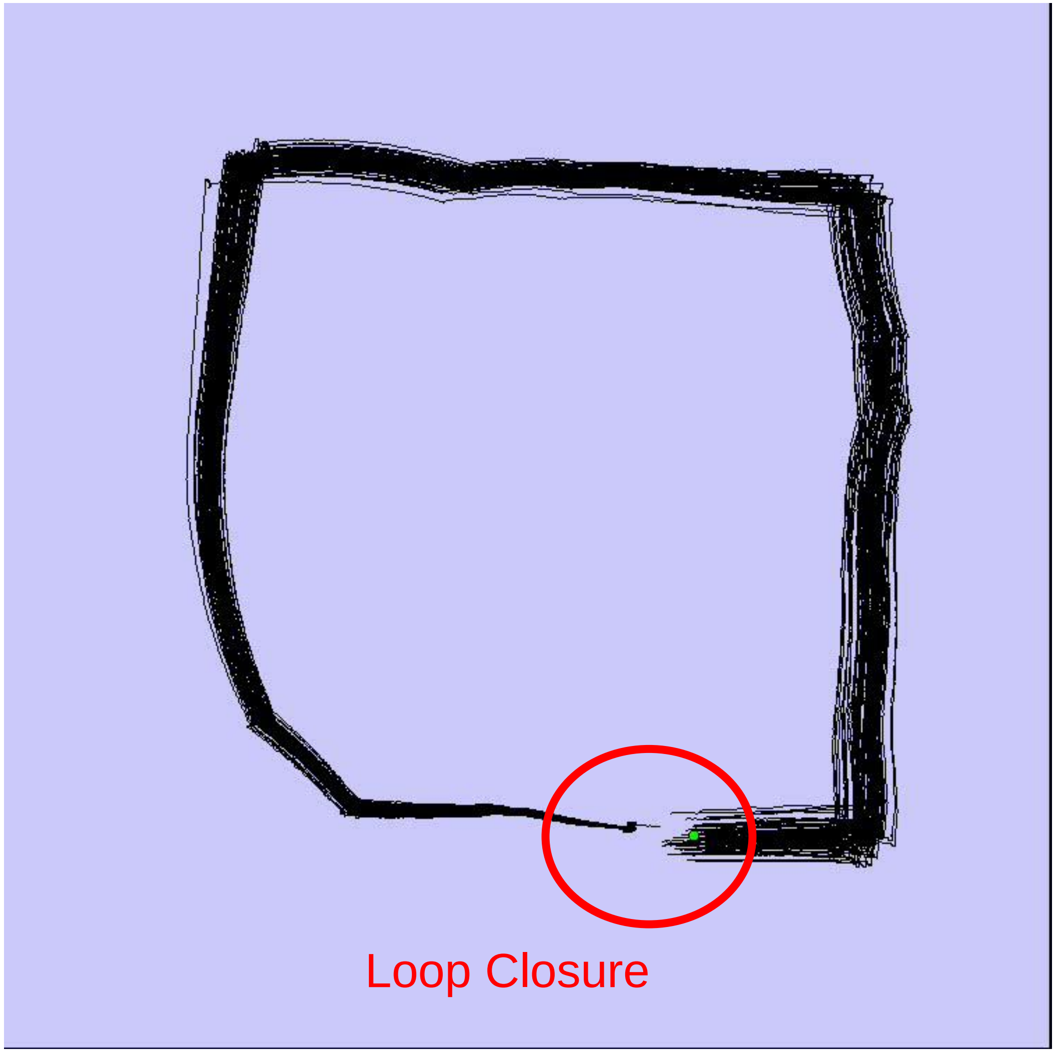


Grid-Based FastSLAM with Scan-Matching



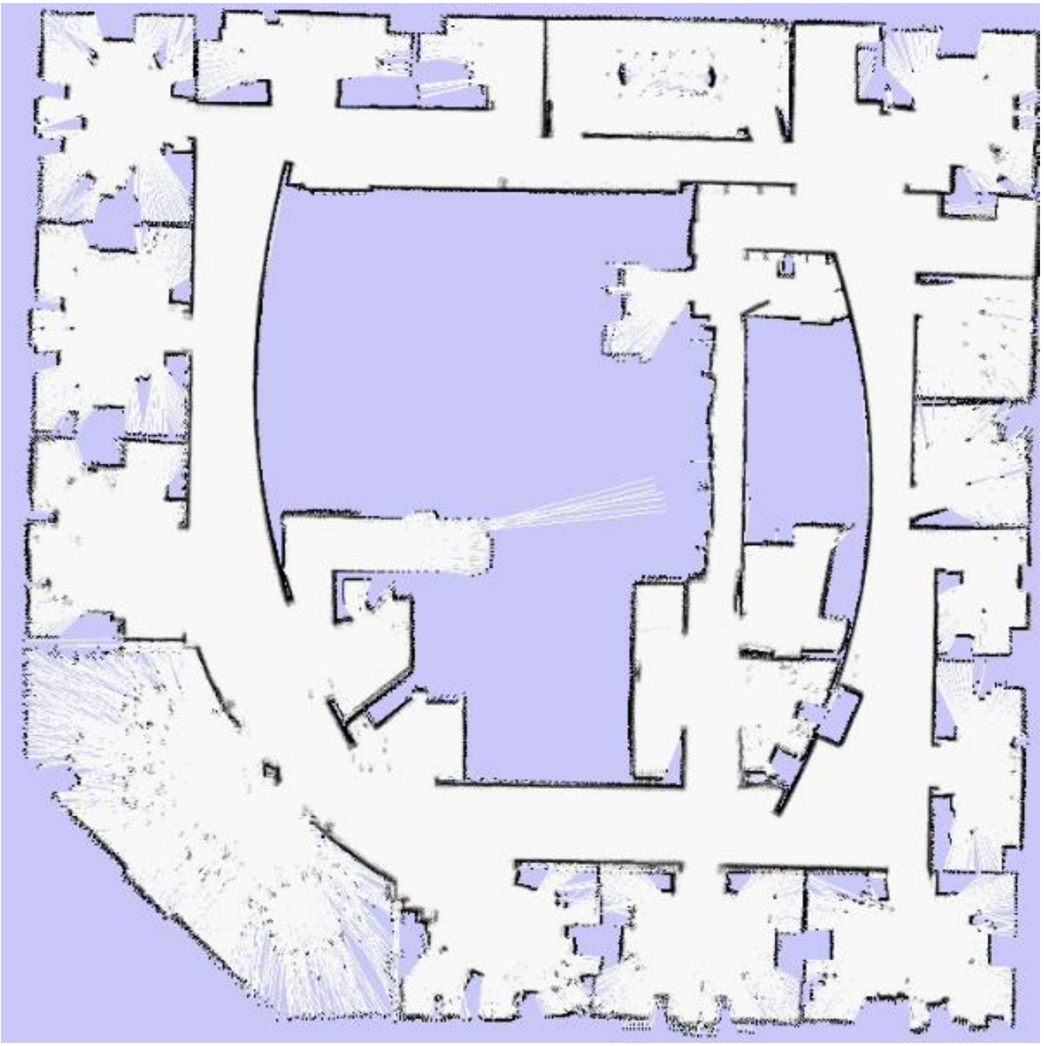
Courtesy:
Dirk Hähnel

Grid-Based FastSLAM with Scan-Matching



Courtesy:
Dirk Hähnel

Grid-Based FastSLAM with Scan-Matching



Courtesy:
Dirk Hähnel

Summary so far

- An approach to grid-based SLAM that combines scan matching and FastSLAM
- Scan matching to generate improved odometry estimates
- This version of grid-based FastSLAM can handle larger environments than before

FastSLAM 2.0

- Compute an **improved** proposal that considers the most recent observation
- Draw from the posterior

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

As a result:

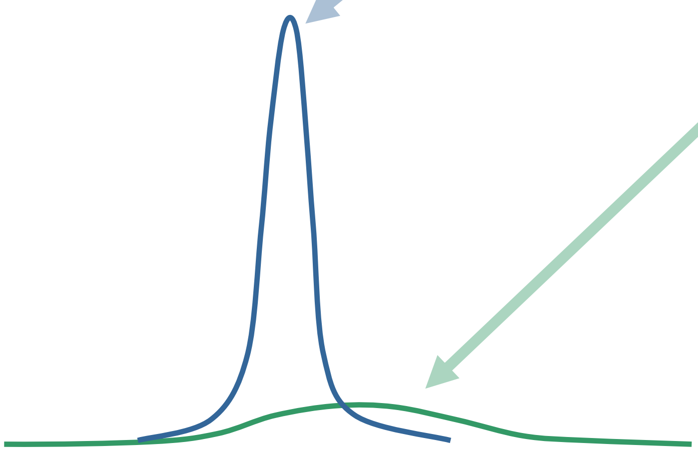
- More precise sampling
- Less particles needed
- More accurate maps

Summarized Key Idea

- Perform scan matching for each particle using its own map
- Fit a Gaussian by sampling points around the maximum of scan matcher
- Calculate importance weights using measurement likelihood relative to sampled points
- Selective Resampling

The Optimal Proposal Distribution [Arulampalam et al., 2001]

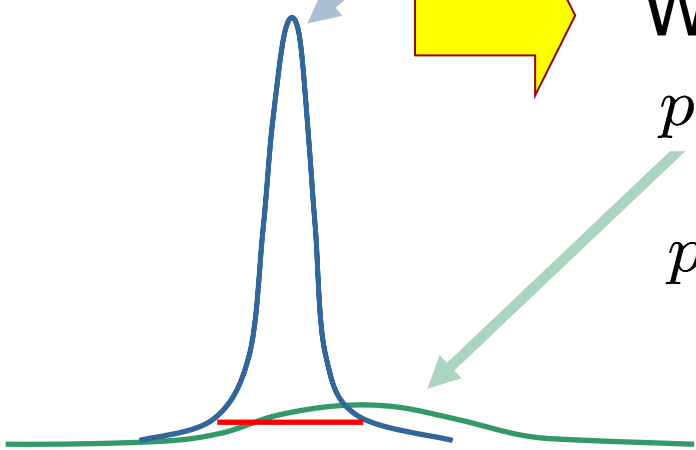
$$p(x_t \mid x_{t-1}^{[i]}, m^{[i]}, z_t, u_t) = \frac{p(z_t \mid x_t, m^{[i]})\; p(x_t \mid x_{t-1}^{[i]}, u_t)}{\int p(z_t | x_t, m^{[i]}) p(x_t | x_{t-1}^{[i]}, u_t) d\, x_t}$$



For lasers $p(z_t \mid x_t, m^{[i]})$ is typically peaked and dominates the product

The Optimal Proposal Distribution [Arulampalam et al., 2001]

$$p(x_t \mid x_{t-1}^{[i]}, m^{[i]}, z_t, u_t) = \frac{p(z_t \mid x_t, m^{[i]})\; p(x_t \mid x_{t-1}^{[i]}, u_t)}{\int p(z_t|x_t, m^{[i]}) p(x_t|x_{t-1}^{[i]}, u_t) d\, x_t}$$



We can safely approximate $p(x_t|x_{t-1}^{[i]}, u_t)$ by a constant:

$$p(x_t|x_{t-1}^{[i]}, u_t) \mid_{x_t : p(z_t|x_t, m^{[i]}) > \epsilon} = c$$

Resulting Proposal Distribution

$$p(x_t|x_{t-1}^{[i]}, m^{[i]}, z_t, u_t) \simeq \frac{p(z_t|x_t, m^{[i]})}{\int_{x_t \in \{x|p(z_t|x,m^{[i]})>\epsilon\}} p(z_t|x_t, m^{[i]}) dx_t}$$

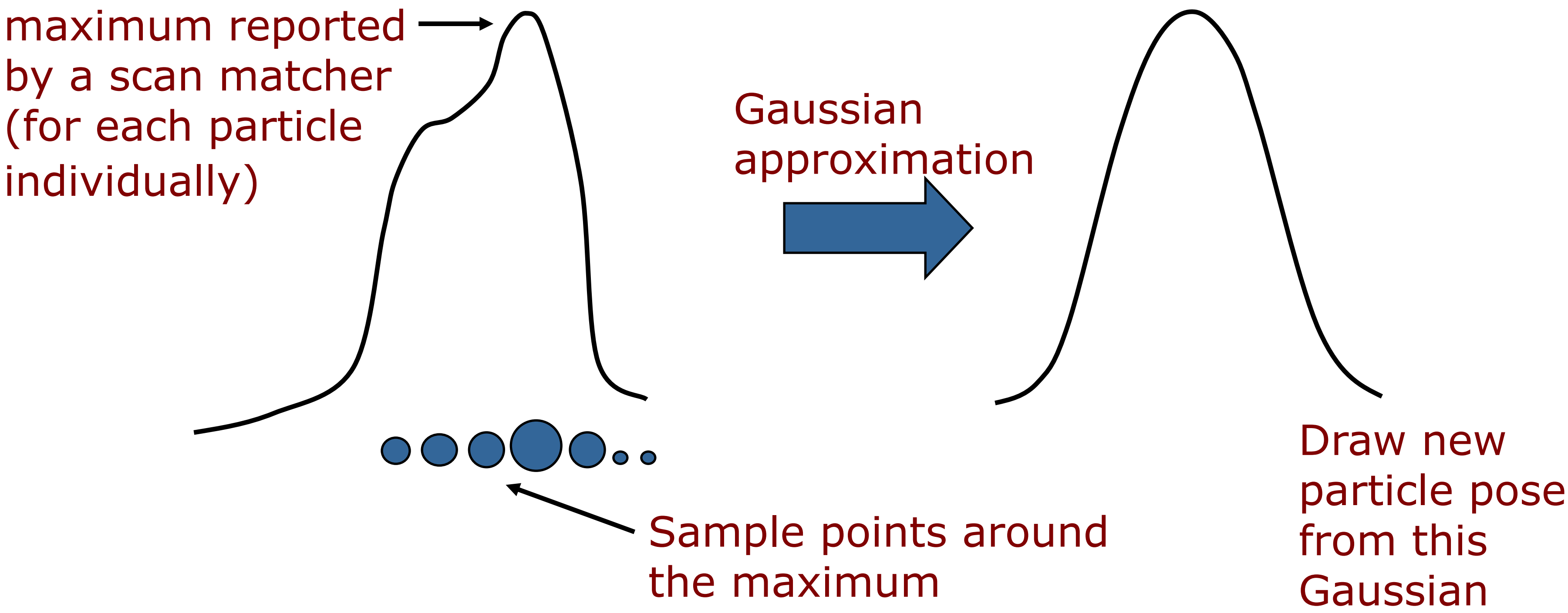
Gaussian approximation:

$$p(x_t|x_{t-1}^{[i]}, m^{[i]}, z_t, u_t) \simeq \mathcal{N}(\mu^{[i]}, \Sigma^{[i]})$$

Resulting Proposal Distribution

$$p(x_t | x_{t-1}^{[i]}, m^{[i]}, z_t, u_t) \simeq \frac{p(z_t | x_t, m^{[i]})}{\int_{x_t \in \{x | p(z_t | x, m^{[i]}) > \epsilon\}} p(z_t | x_t, m^{[i]}) dx_t}$$

Approximate this equation by a Gaussian:



Estimating the Parameters of the Gaussian for Each Particle

$$\mu^{[i]} = \frac{1}{\eta} \sum_{j=1}^{K} x_j^{[i]} p(z_t | x_j^{[i]}, m^{[i]})$$

$$\Sigma^{[i]} = \frac{1}{\eta} \sum_{j=1}^{K} (x_j^{[i]} - \mu^{[i]})(x_j^{[j]} - \mu^{[i]})^T p(z_t | x_j^{[i]}, m^{[i]})$$

$x_j^{[i]}$ are the points sampled around the result of the scan matcher for particle i

[Grisetti et al., 2007]

Computing the Importance Weights

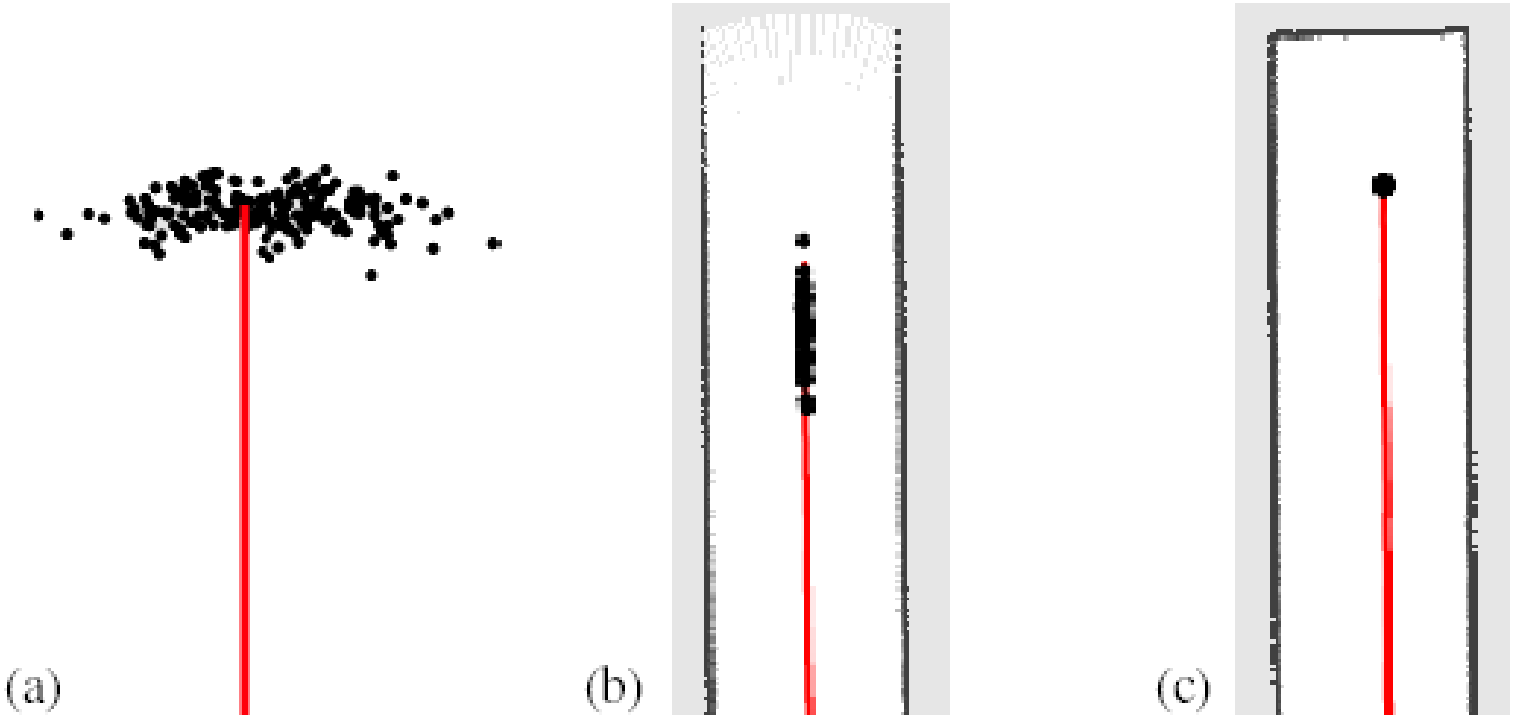
$$
\begin{aligned}
w_t^{[i]} &\simeq w_{t-1}^{[i]} \int p(z_t | x_t, m^{[i]}) p(x_t | x_{t-1}^{[i]}, u_t) d\, x_t \\
&\simeq w_{t-1}^{[i]} c \int_{x_t \in \{x | p(z_t | x, m^{[i]}) > \epsilon\}} p(z_t | x_t, m^{[i]}) dx_t \\
&\simeq w_{t-1}^{[i]} c \sum_{j=1}^{K} p(z_t | x_j^{[i]}, m^{[i]})
\end{aligned}
$$

Sampled points around the maximum of the likelihood function found by scan-matching

[Grisetti et al., 2007]

Improved Proposal

The proposal adapts to the structure of the environment

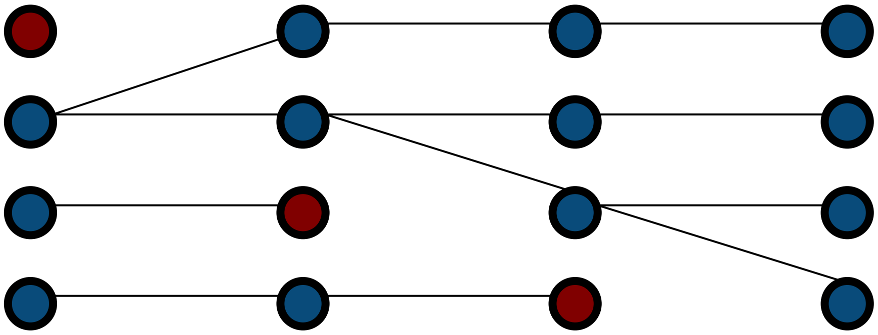


Summarized Key Idea

- Perform scan matching for each particle using its own map
- Fit a Gaussian by sampling points around the maximum of scan matcher
- Calculate importance weights using measurement likelihood relative to sampled points
- Selective Resampling

Resampling

- Resampling at each step limits the "memory"
- Suppose we loose each time 25% of the particles, this may lead to:



Goal: Reduce the resampling actions

Selective Resampling

- Resampling is necessary to achieve convergence
- Resampling is dangerous, since important samples might get lost (“particle depletion”)
- Resampling makes only sense if particle weights differ significantly

Key question: When to resample?

Number of Effective Particles

- Empirical measure of how well the target distribution is approximated by samples drawn from the proposal

$$n_{eff} = \frac{1}{\sum_i \left(w_t^{[i]}\right)^2}$$

- n_{eff} describes "the inverse variance of the normalized particle weights"
- For equal weights, the sample approximation is close to the target

Resampling with n_{eff}

- If the approximation is close to the target, no resampling is needed
- Only resample when n_{eff} drops below a given threshold

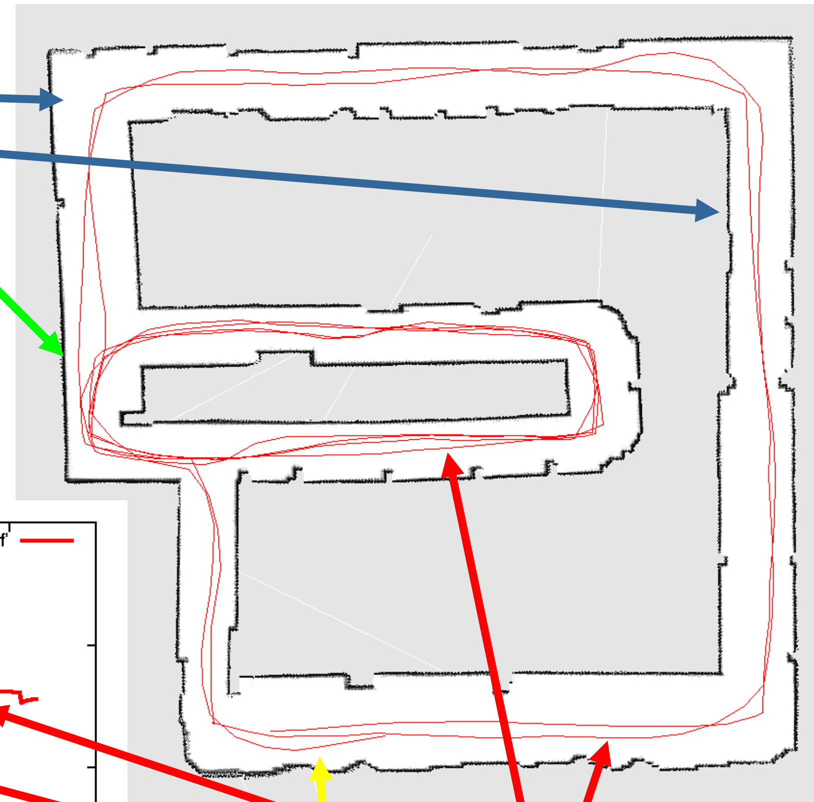
$$\frac{1}{\sum_i \left(w_t^{[i]}\right)^2} \overset{?}{<} N/2$$

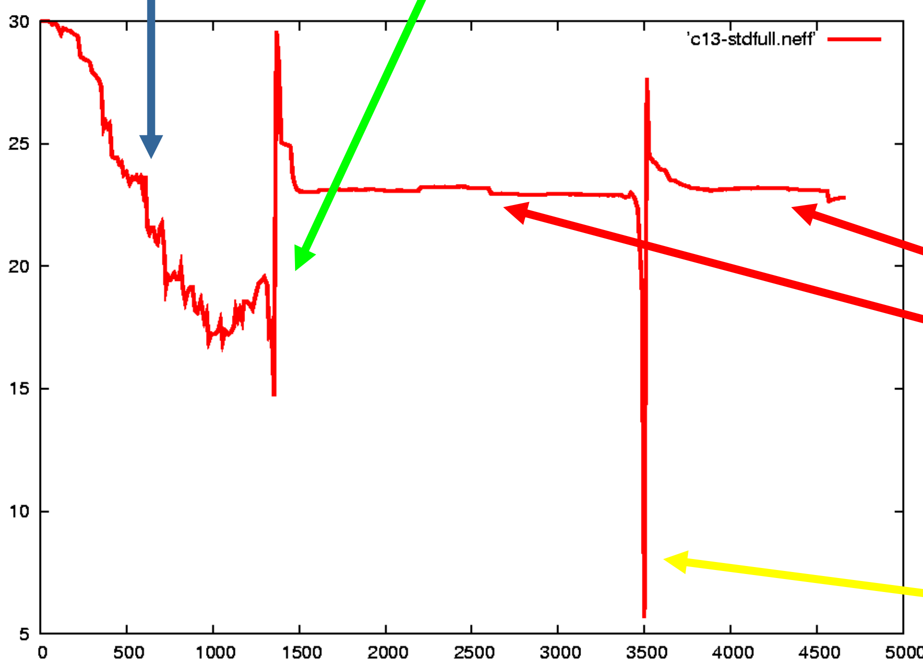
[Doucet, '98; Arulampalam, '01]

Typical Evolution of n_{eff}

visiting new areas

closing the first loop





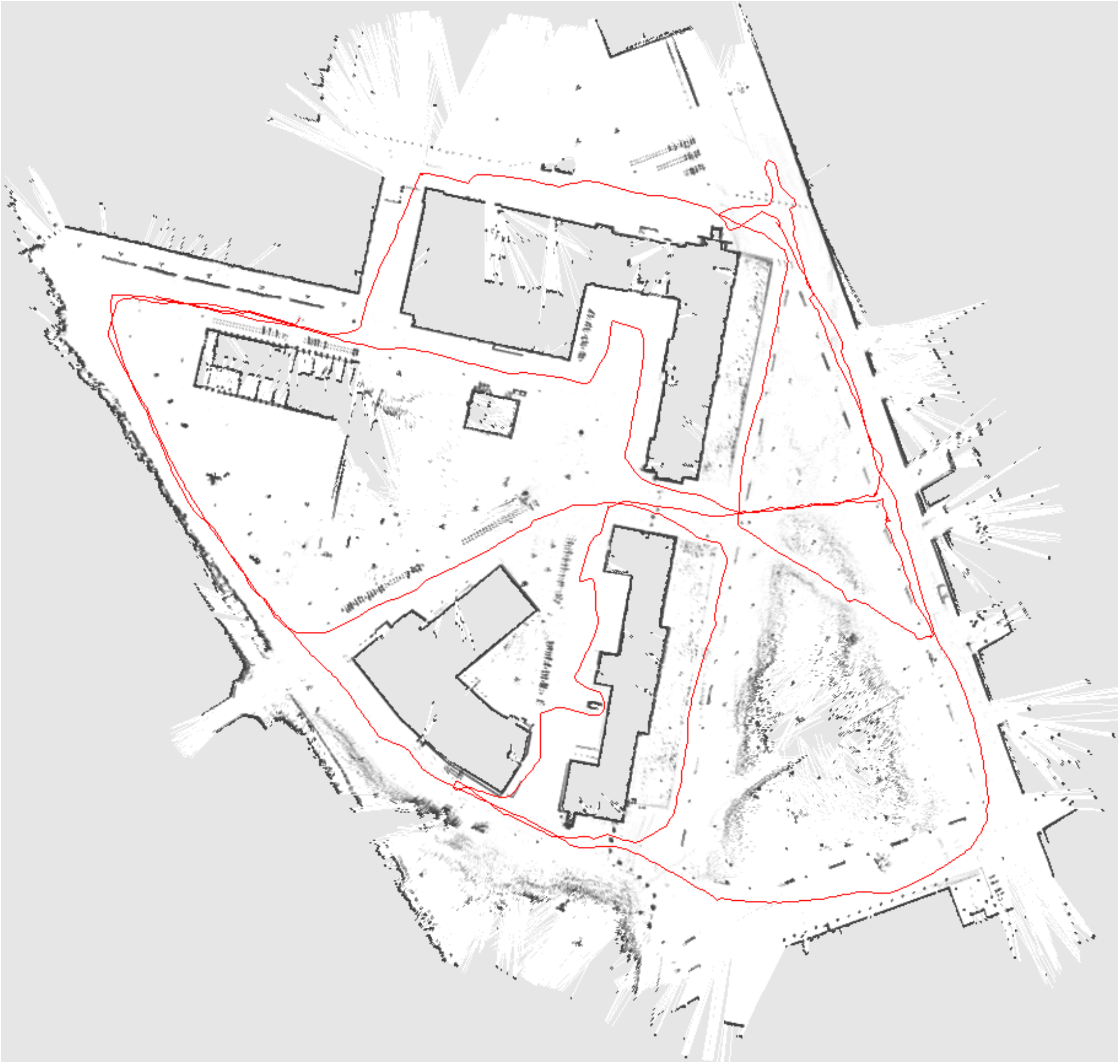
visiting known areas

second loop closure

Intel Lab

- **15 particles**
- four times faster than real-time P4, 2.8GHz
- 5cm resolution during scan matching
- 1cm resolution in final map

Outdoor Campus Map



- **30 particles**
- $250 \times 250 m^2$
- 1.75 km (odometry)
- 30cm resolution in final map

Summary: Grid-Based FastSLAM

- The ideas of FastSLAM can also be applied in the context of grid maps
- Improved proposals are essential
- Similar to scan-matching on a per-particle base
- Selective resampling reduces the risk of particle depletion
- Substantial reduction of the required number of particles

Acknowledgment

- These slides have been created by Wolfram Burgard, Cyrill Stachniss and Maren Bennewitz