

Introduction to Mobile Robotics

SLAM – Grid-based FastSLAM

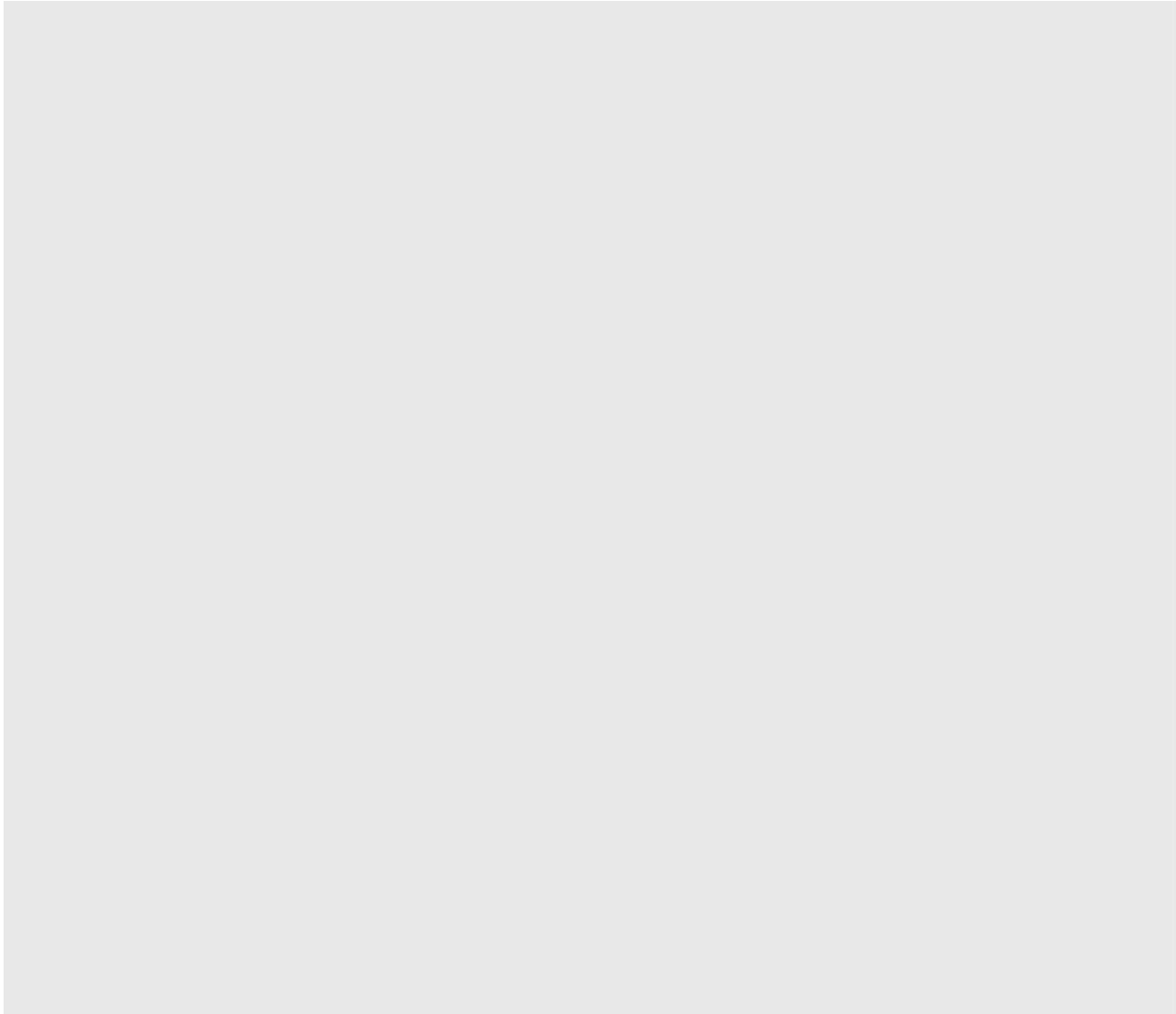
Wolfram Burgard, Cyrill Stachniss, Maren
Bennewitz, Giorgio Grisetti, Kai Arras



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 and egg problem:
a map is needed to localize the robot and
a pose estimate is needed to build a map

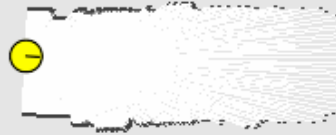
Mapping using Raw Odometry



Grid-based SLAM

- Can we solve the SLAM problem if no pre-defined landmarks are available?
- Can we use the ideas of FastSLAM to build grid maps?
- As with landmarks, the map depends on the poses of the robot during data acquisition
- If the poses are known, grid-based mapping is easy (“mapping with known poses”)


Mapping with Known Poses



- Mapping with known poses using laser range data

Rao-Blackwellization

poses map observations & movements


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

Rao-Blackwellization

poses map observations & movements

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

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

↑
SLAM posterior

↑
Robot path posterior

↑
Mapping with known poses

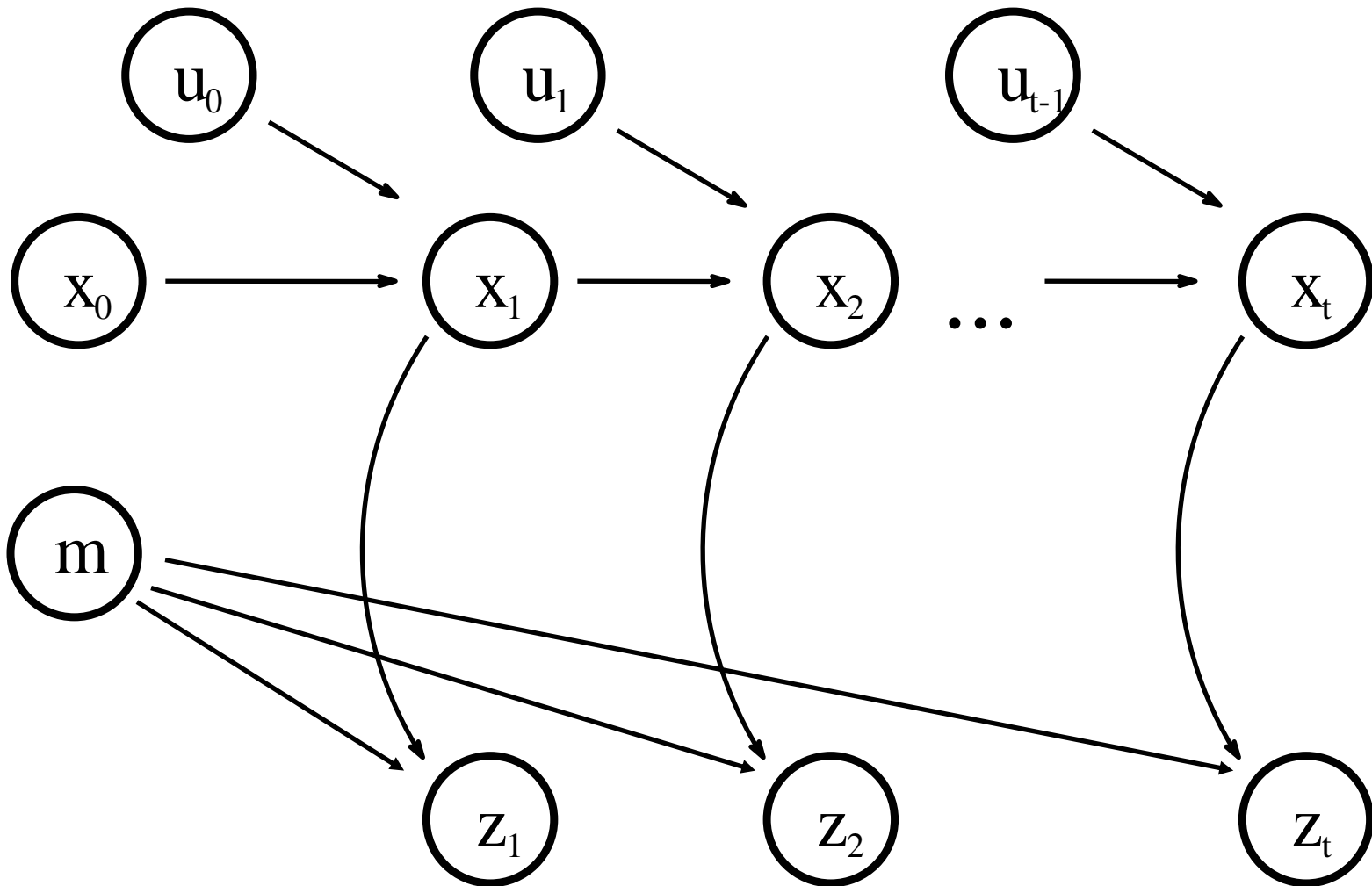
Rao-Blackwellization

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

This is localization, use MCL

Use the pose estimate from the MCL and apply mapping with known poses

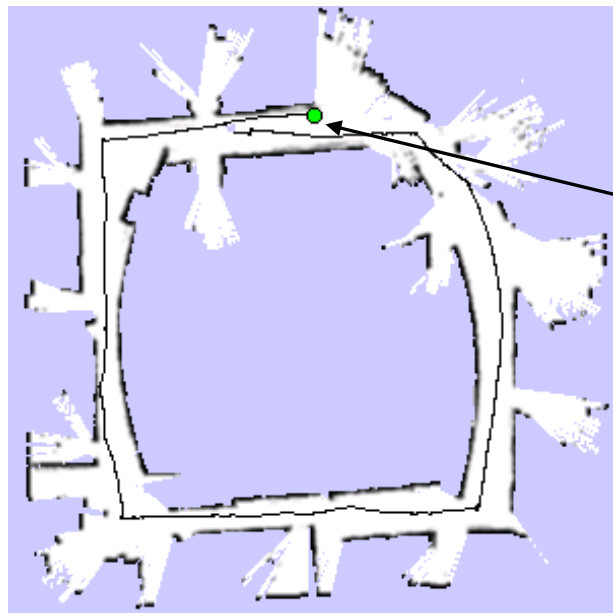
A Graphical Model of Rao-Blackwellized Mapping



Rao-Blackwellized Mapping

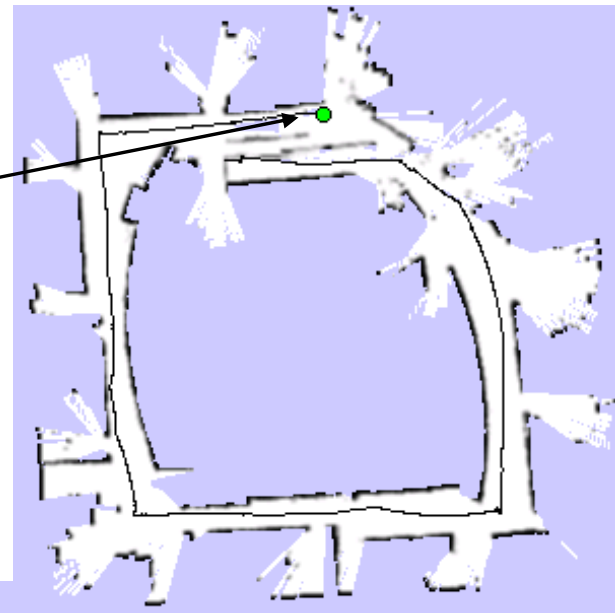
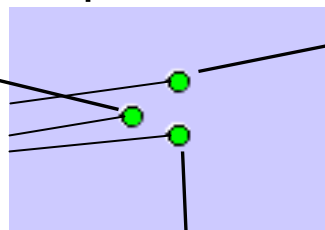
- Each particle represents a possible trajectory of the robot
- Each particle
 - maintains its own map and
 - updates it upon “mapping with known poses”
- Each particle survives with a probability proportional to the likelihood of the observations relative to its own map

Particle Filter Example

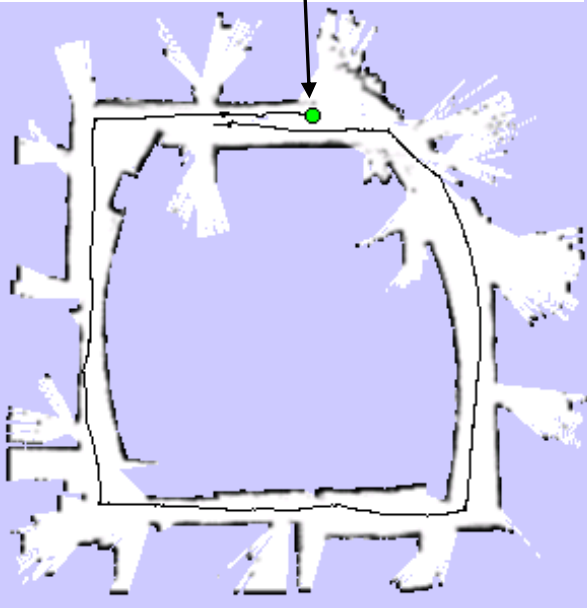


map of particle 1

3 particles



map of particle 3



map of particle 2

Problem

- Each map is quite big in case of grid maps
- Since each particle maintains its own map
- Therefore, one needs to keep the number of particles small

- **Solution:**
Compute better proposal distributions!
- **Idea:**
Improve the pose estimate **before** applying the particle filter

Pose Correction Using Scan Matching

Maximize the likelihood of the i -th pose and map relative to the $(i-1)$ -th pose and map

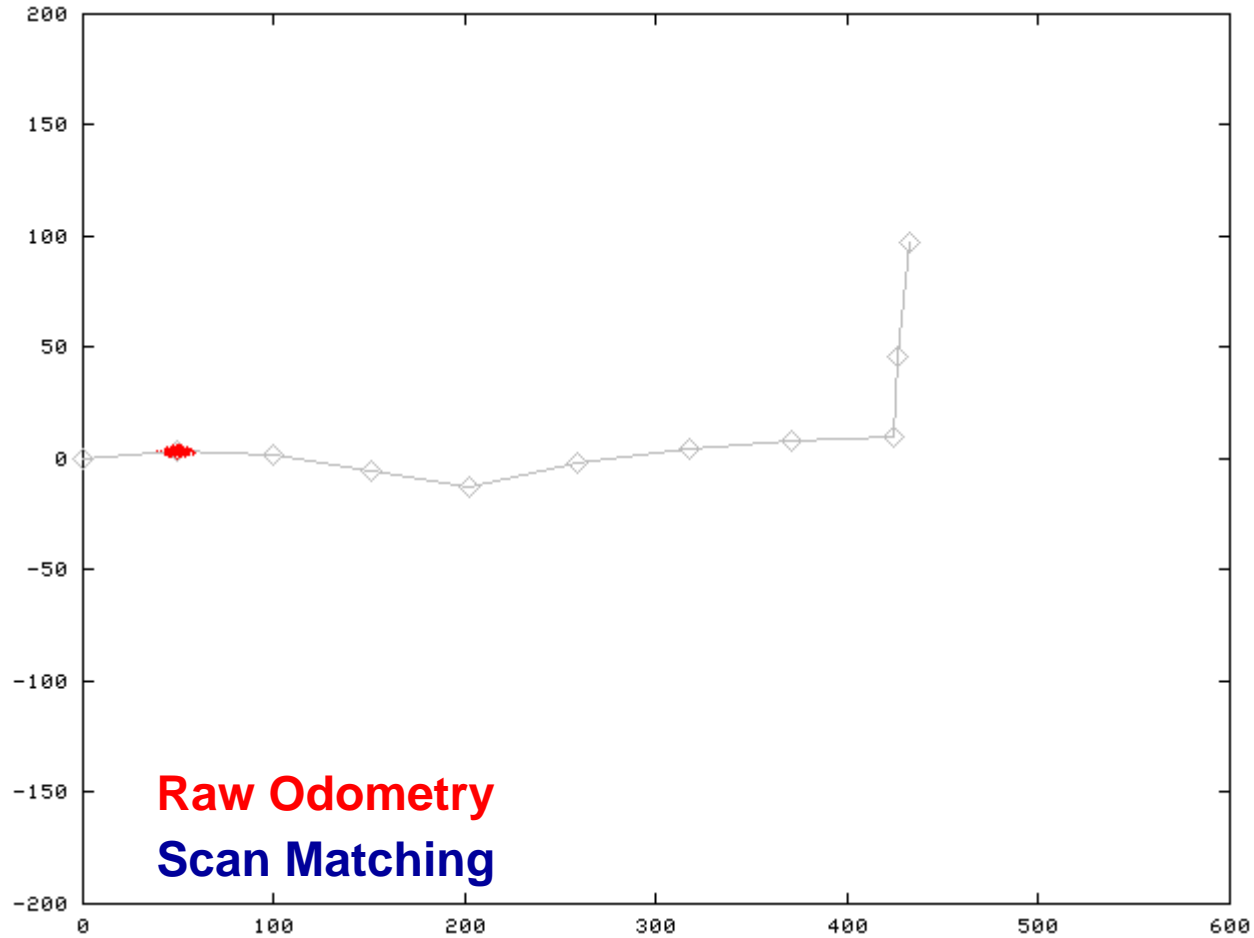
$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}} \left\{ p(z_t | x_t, \hat{m}_{t-1}) \cdot p(x_t | u_{t-1}, \hat{x}_{t-1}) \right\}$$

current measurement

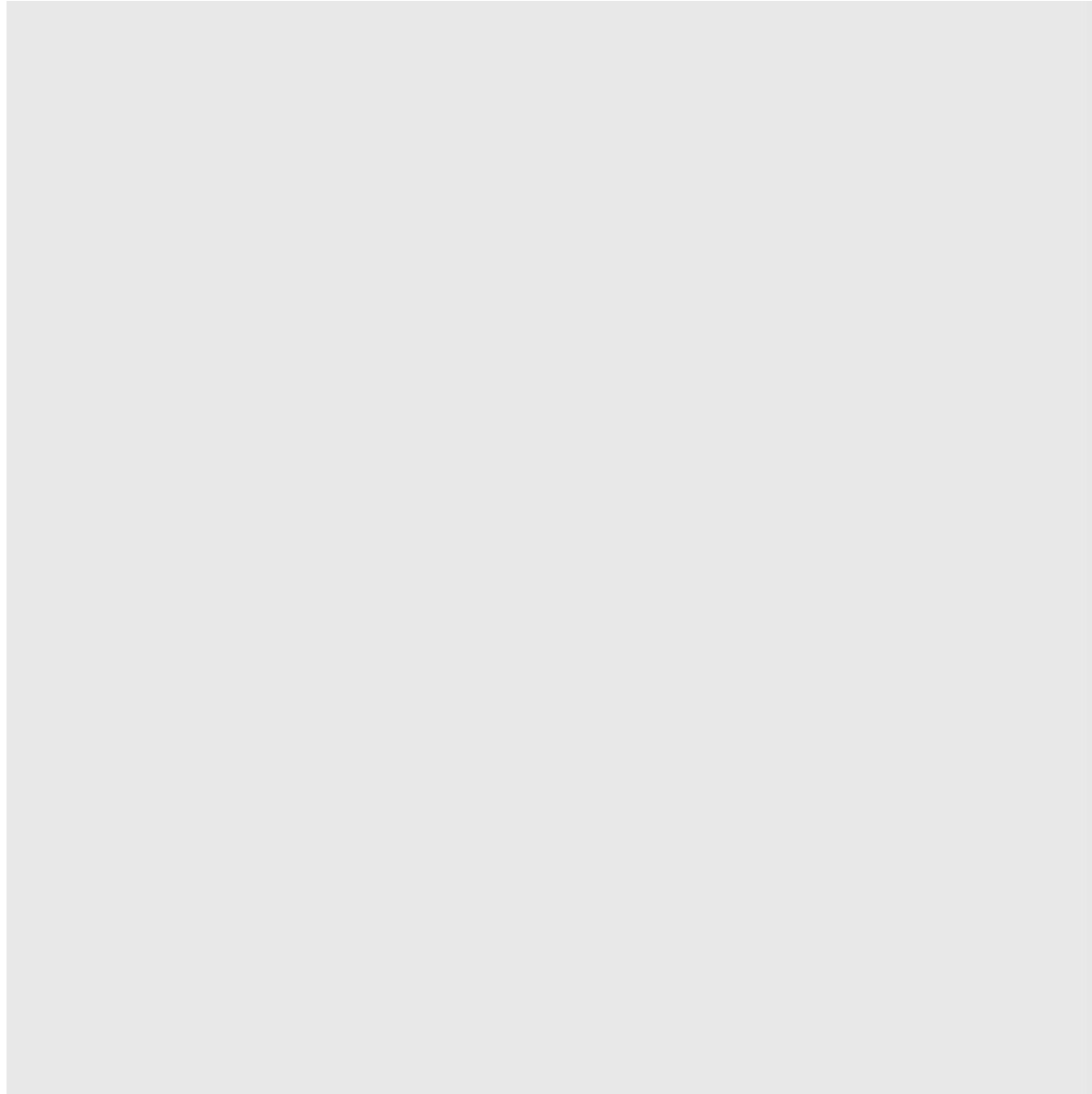
robot motion

map constructed so far

Motion Model for Scan Matching



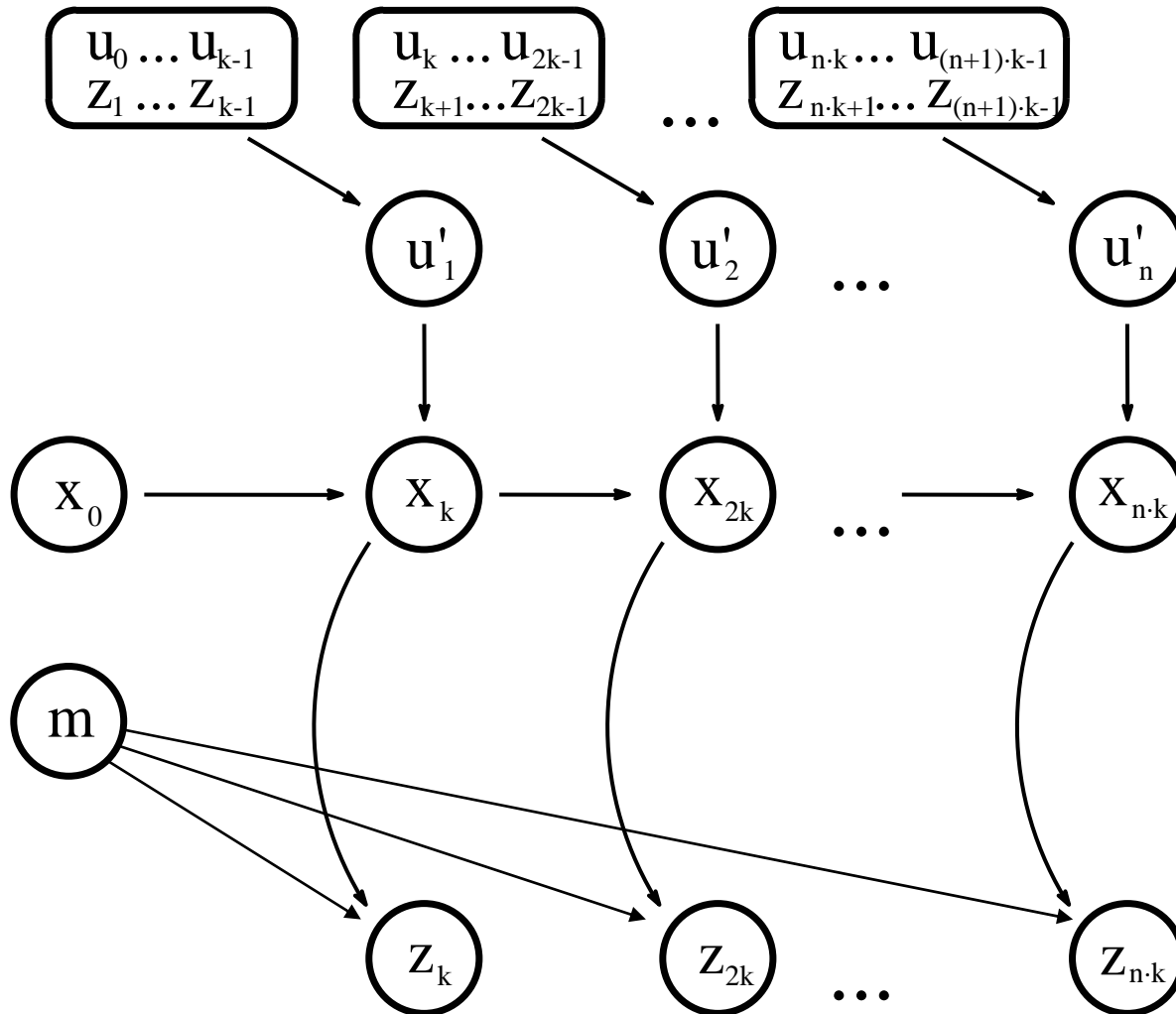
Mapping using Scan Matching



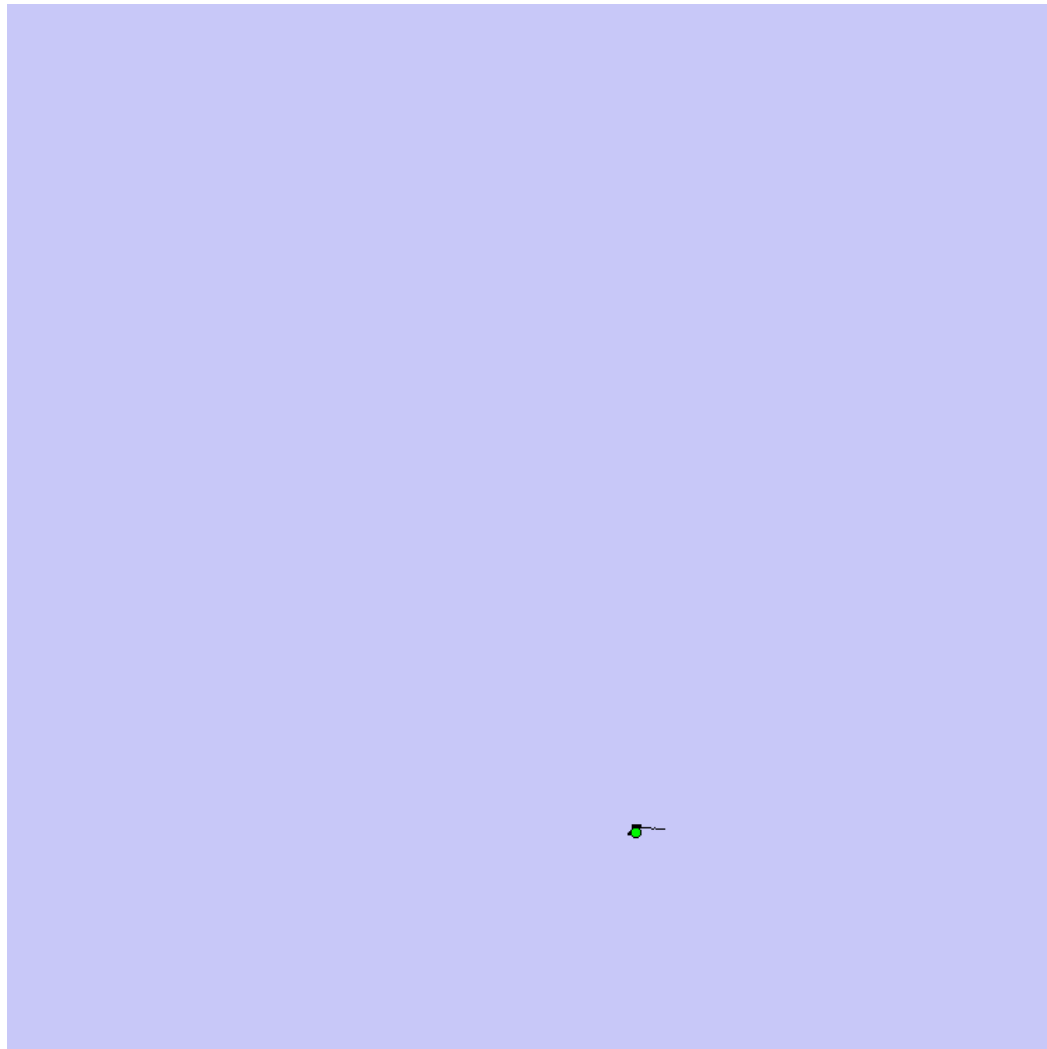
FastSLAM with Improved Odometry

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

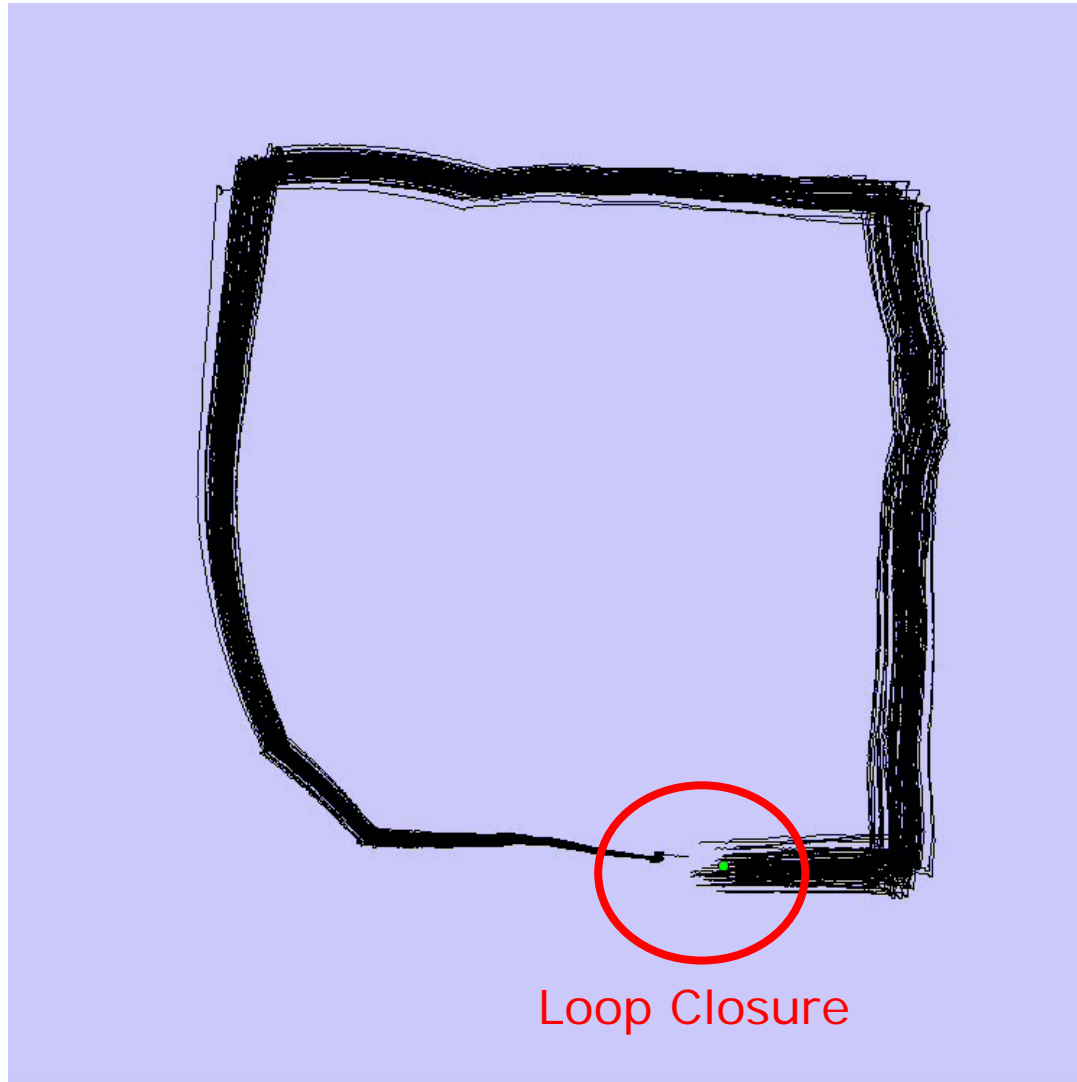
Graphical Model for Mapping with Improved Odometry



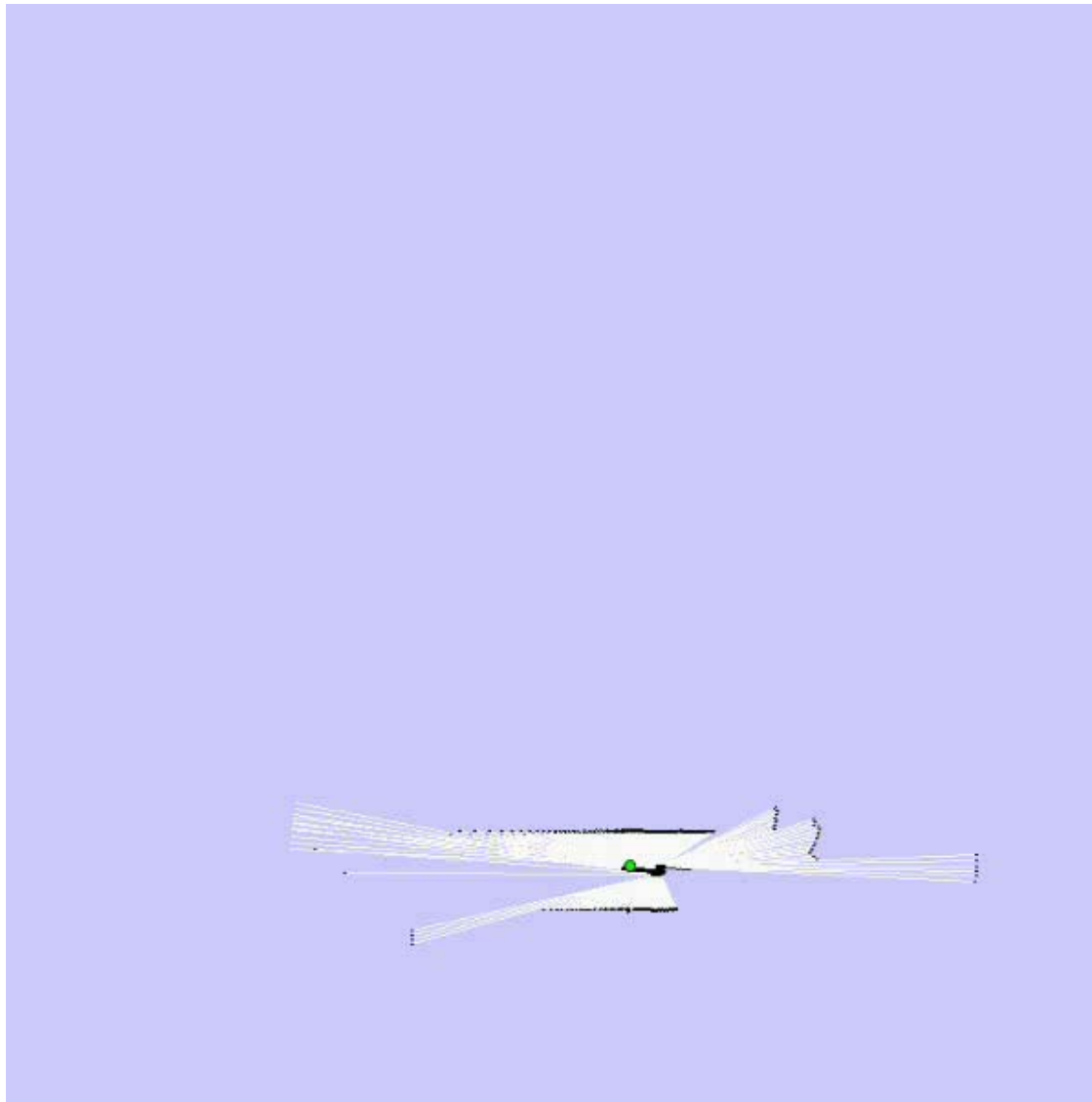
FastSLAM with Scan-Matching



FastSLAM with Scan-Matching

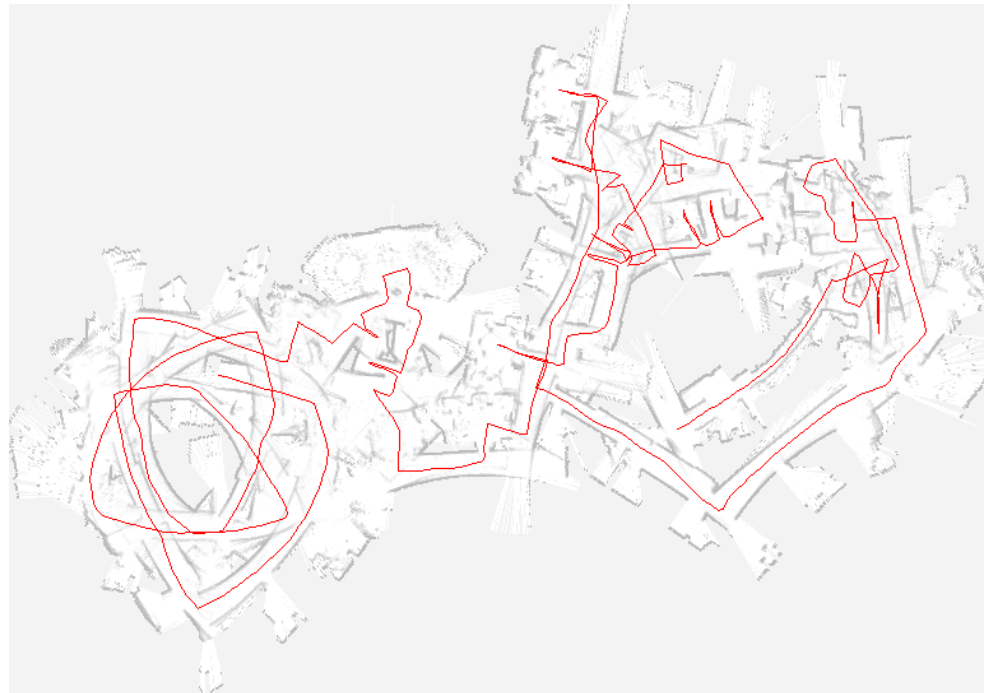


FastSLAM with Scan-Matching



Comparison to Standard FastSLAM

- Same model for observations
- Odometry instead of scan matching as input
- Number of particles varying from 500 to 2.000
- Typical result:



Conclusion (so far...)

- The presented approach is a highly efficient algorithm for SLAM combining ideas of scan matching and FastSLAM
- Scan matching is used to transform sequences of laser measurements into odometry measurements
- This version of grid-based FastSLAM can handle larger environments than before in “real time”

What's Next?

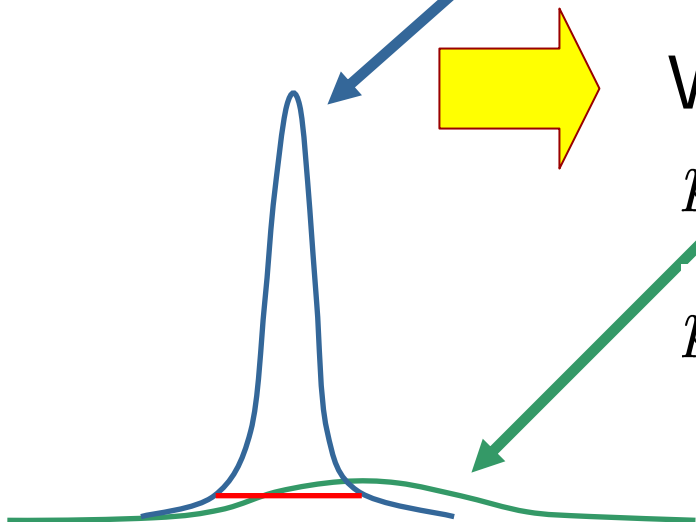
- Further reduce the number of particles
- Improved proposals will lead to more accurate maps
- Use the properties of our sensor when drawing the next generation of particles

The Optimal Proposal Distribution

$$p(x_t | x_{t-1}^{(i)}, m^{(i)}, z_t, u_t) = \frac{p(z_t | x_t, m^{(i)}) p(x_t | x_{t-1}^{(i)}, u_t)}{\int p(z_t | x_t, m^{(i)}) p(x_t | x_{t-1}^{(i)}, u_t) dx_t}$$

[Arulampalam et al., 01]

For lasers $p(z_t | x_t, m^{(i)})$ is extremely peaked and dominates the product.



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

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

Resulting Proposal Distribution

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

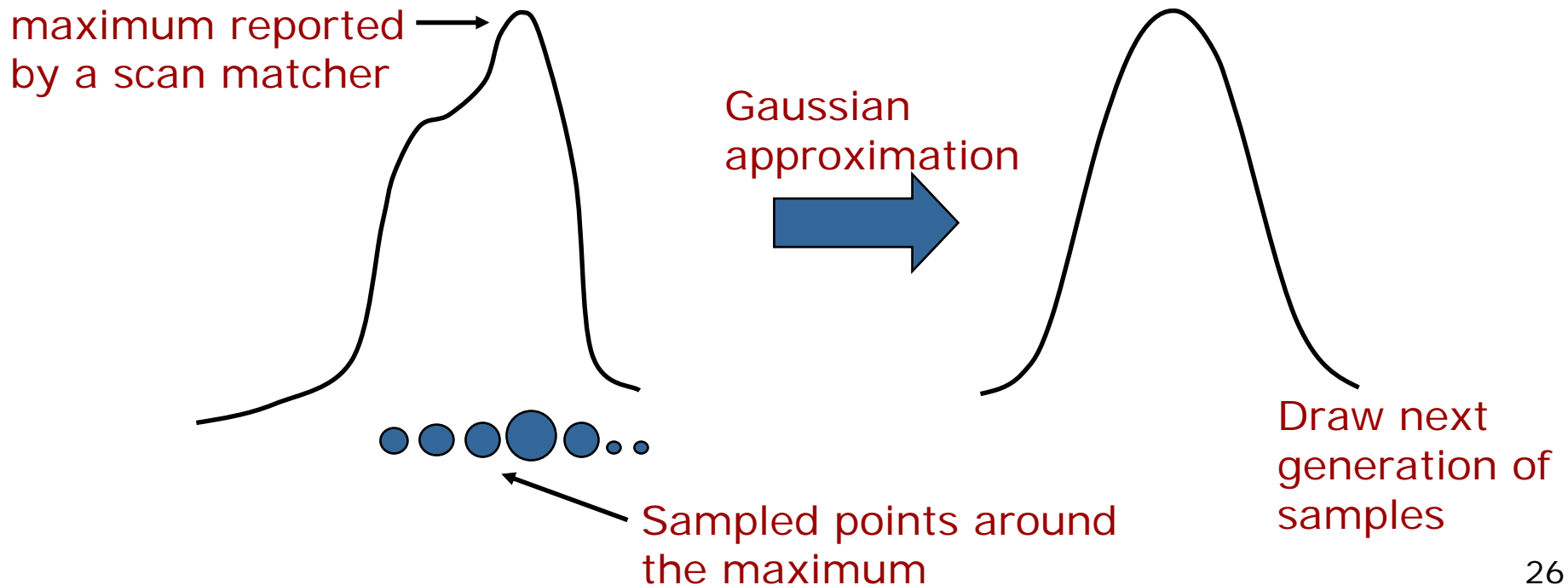
Gaussian approximation:

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

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

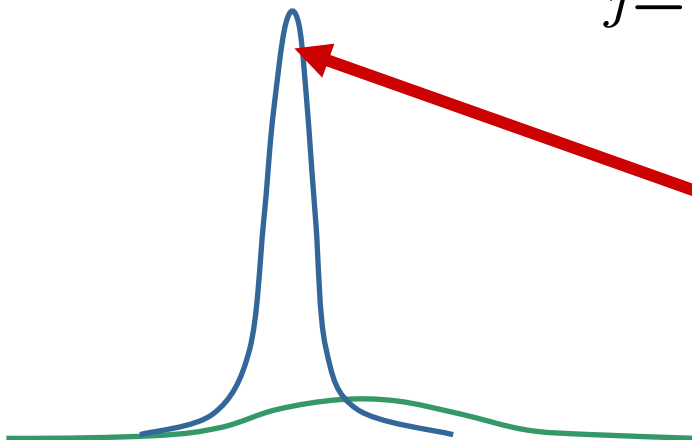
$$\mu^{(i)} = \frac{1}{\eta} \sum_{j=1}^K x_j p(z_t | x_j, m^{(i)})$$

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

- x_j are a set of sample points around the point x^* the scan matching has converged to.
- η is a normalizing constant

Computing the Importance Weight

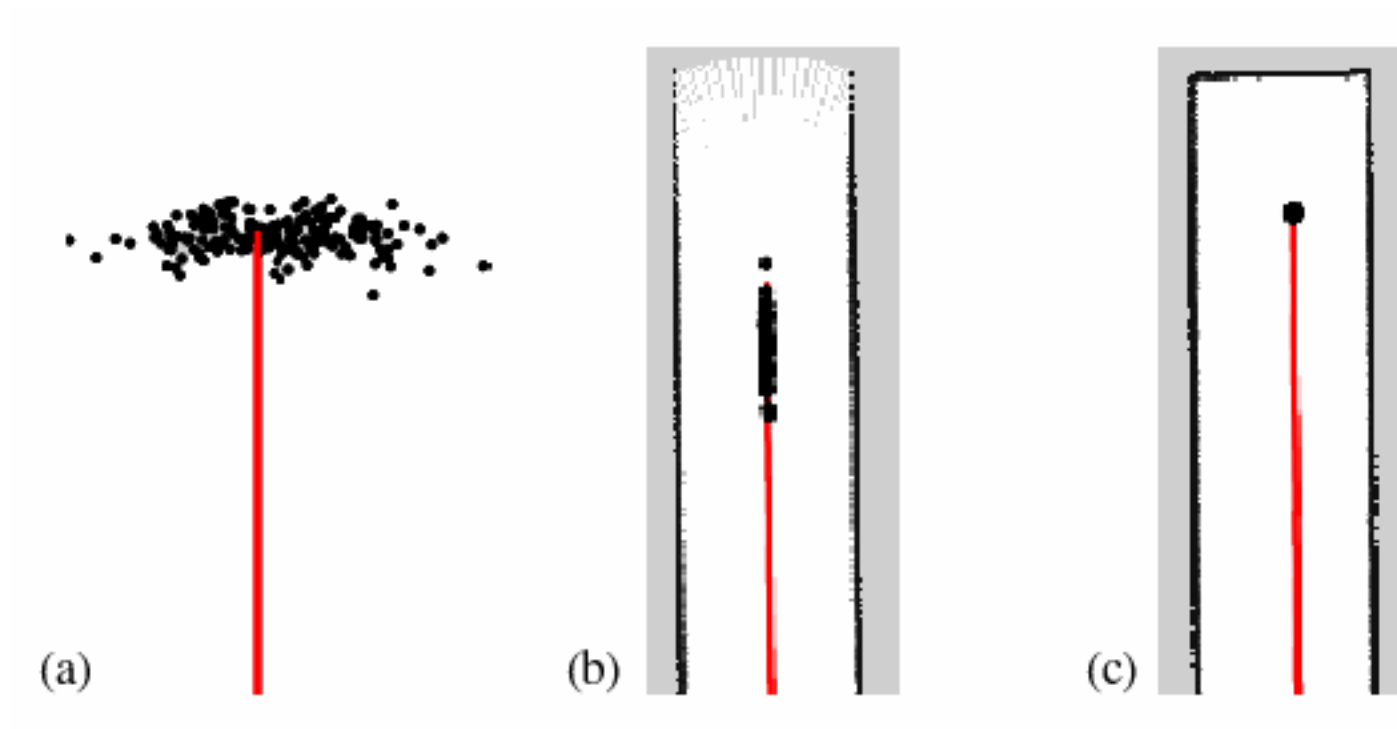
$$\begin{aligned}w_t^{(i)} &= w_{t-1}^{(i)} p(z_t | x_{t-1}^{(i)}, m^{(i)}) \\ &\approx w_{t-1}^{(i)} \int p(z_t | x_t, m^{(i)}) p(x_t | x_{t-1}^{(i)}, u_t) dx_t \\ &\approx w_{t-1}^{(i)} c \int_{x_t \in \{x | p(z_t | x, m^{(i)}) > \epsilon\}} p(z_t | x_t, m^{(i)}) dx_t \\ &\approx w_{t-1}^{(i)} c \sum_{j=1}^K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



Sampled points around the maximum of the observation likelihood

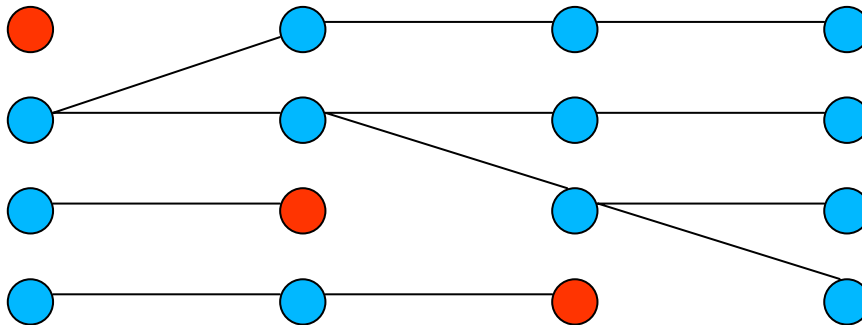
Improved Proposal

- The proposal adapts to the structure of the environment



Resampling

- Sampling from an improved proposal reduces the effects of resampling
- However, resampling at each step limits the “memory” of our filter
- Supposing of loosing at each frame 25% of the particles, in the worst case we have a memory of only 4 steps.



Goal: reduce the number of resampling actions

Selective Re-sampling

- Re-sampling is dangerous, since important samples might get lost (particle depletion problem)
- In case of suboptimal proposal distributions re-sampling is necessary to achieve convergence.
- Key question: When should we re-sample?

Number of Effective Particles

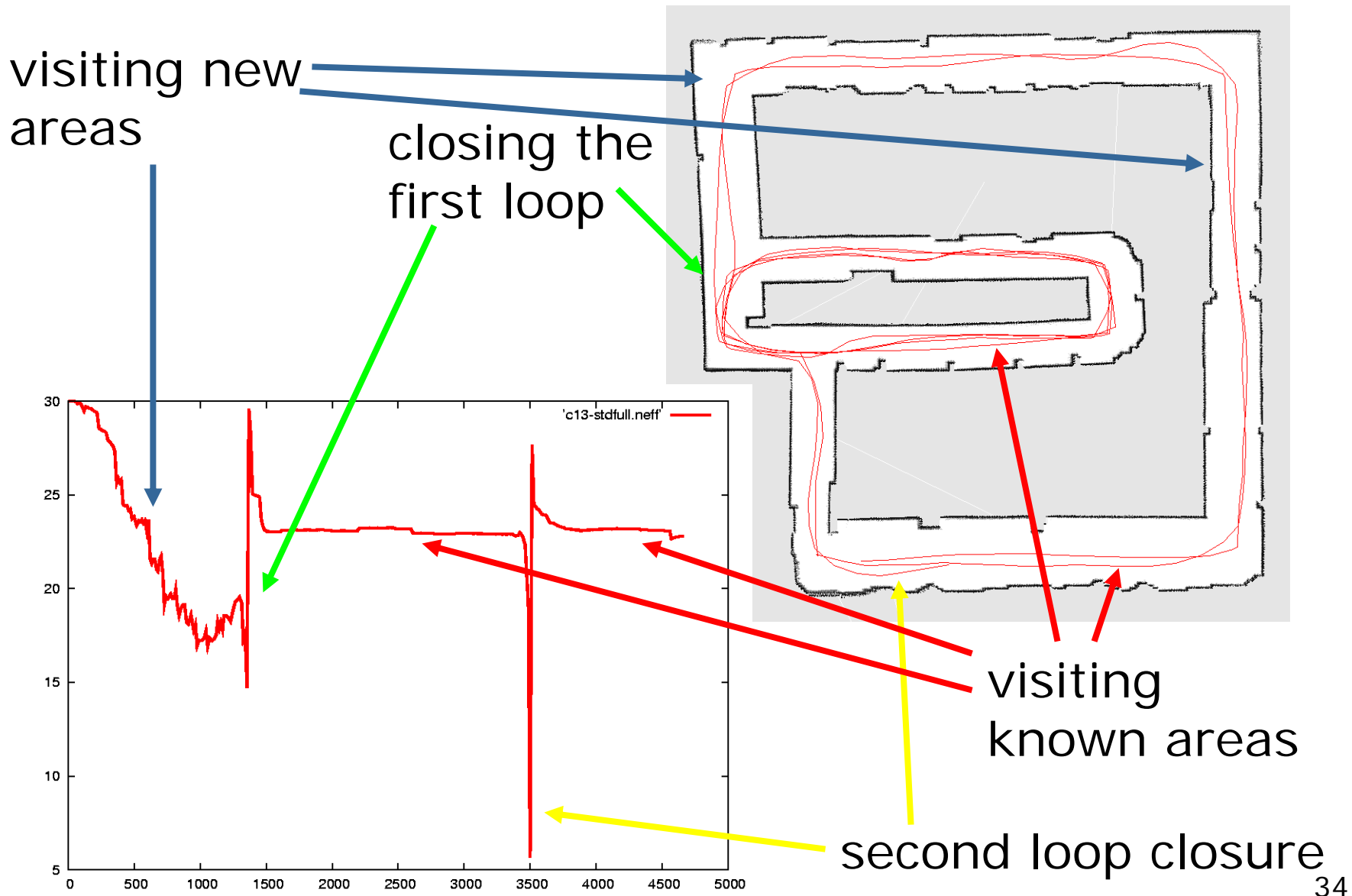
$$n_{eff} = \frac{1}{\sum_i \left(w_t^{(i)}\right)^2}$$

- Empirical measure of how well the goal distribution is approximated by samples drawn from the proposal
- n_{eff} describes “the variance of the particle weights”
- n_{eff} is maximal for equal weights. In this case, the distribution is close to the proposal

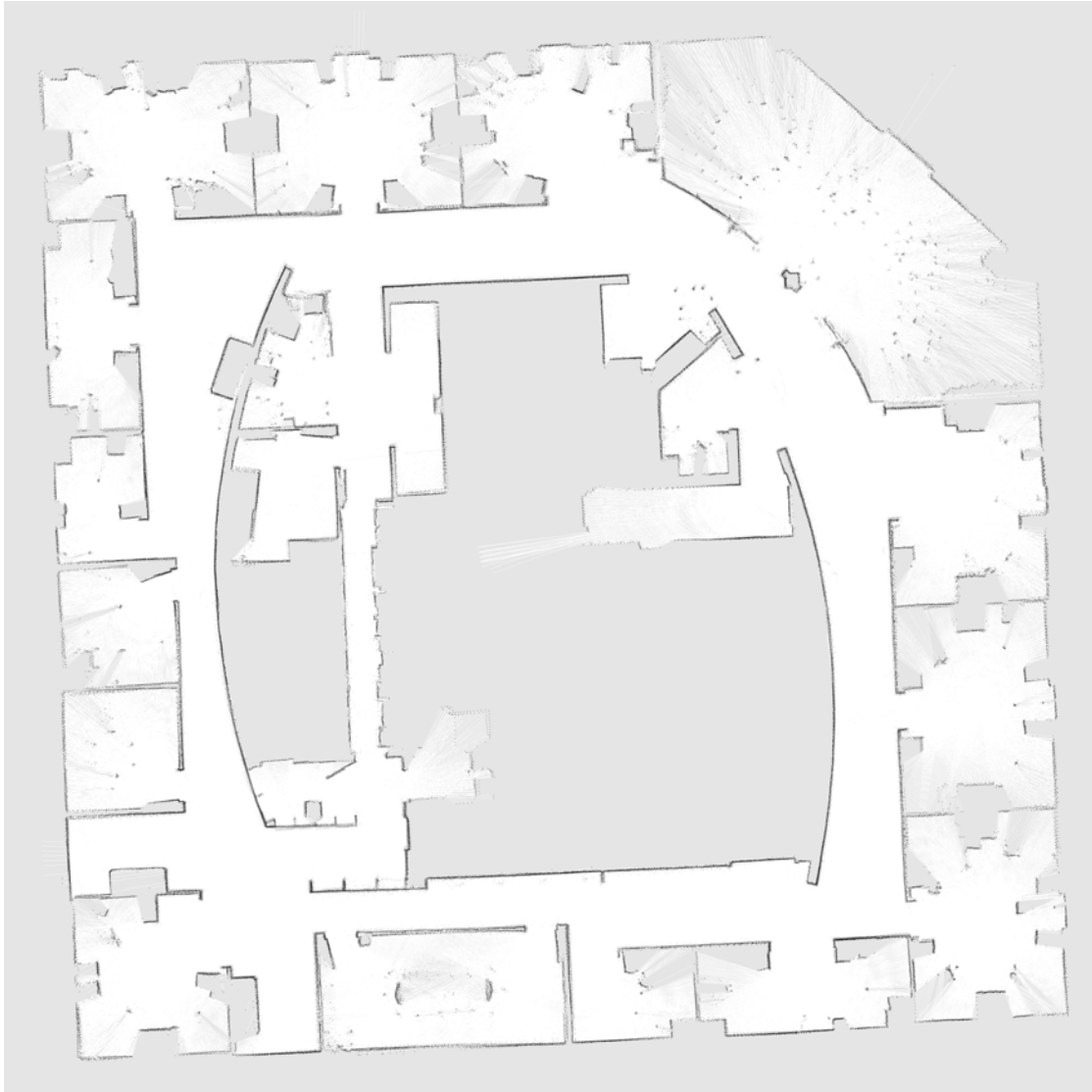
Resampling with Neff

- If our approximation is close to the proposal, no resampling is needed
- We only re-sample when n_{eff} drops below a given threshold ($n/2$)
- See [Doucet, '98; Arulampalam, '01]

Typical Evolution of n_{eff}

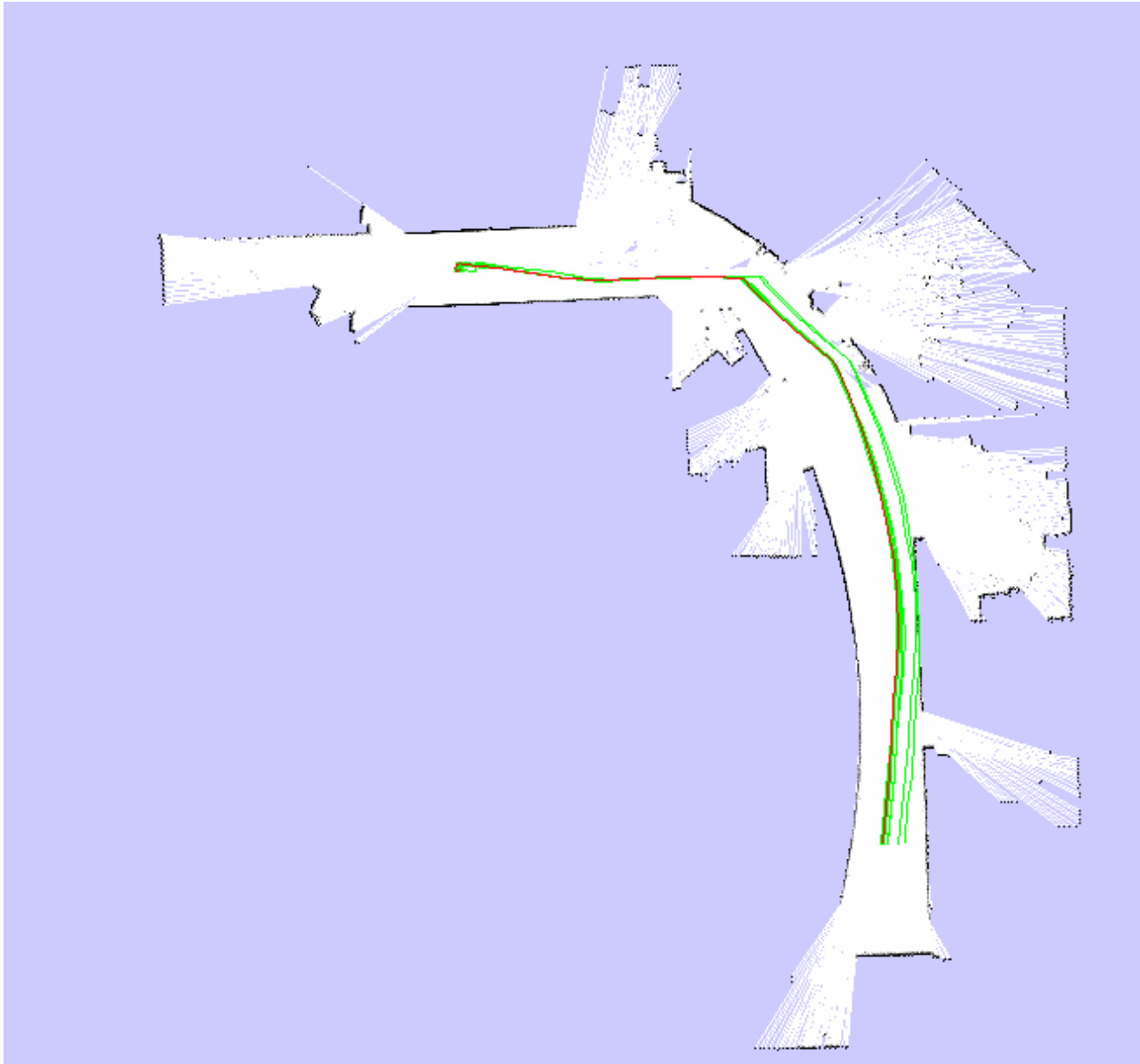


Intel Lab



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

Intel Lab



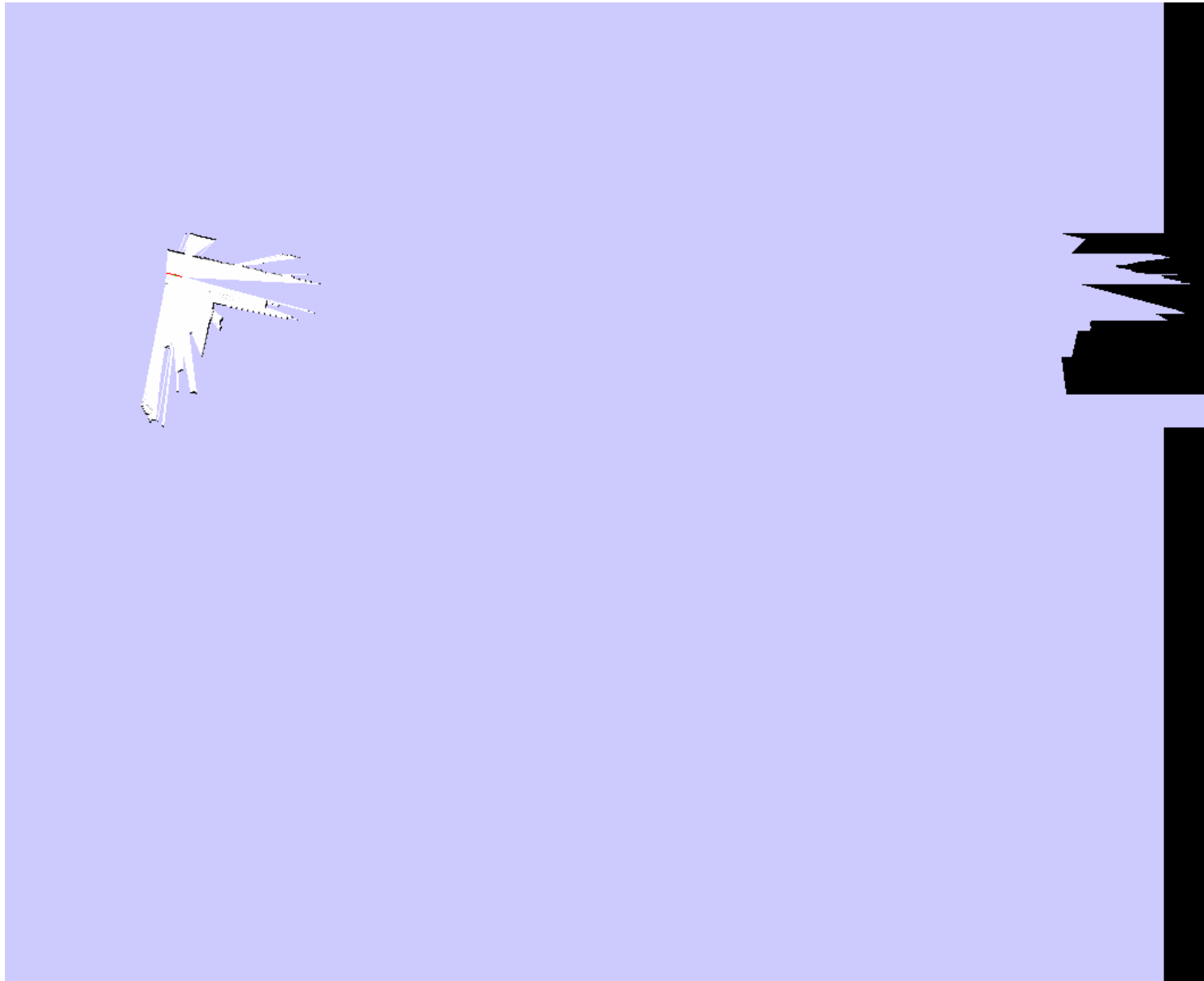
- **15 particles**
- Compared to FastSLAM with Scan-Matching, the particles are propagated closer to the true distribution

Outdoor Campus Map



- **30 particles**
- 250x250m²
- 1.088 miles (odometry)
- 20cm resolution during scan matching
- 30cm resolution in final map

Outdoor Campus Map - Video

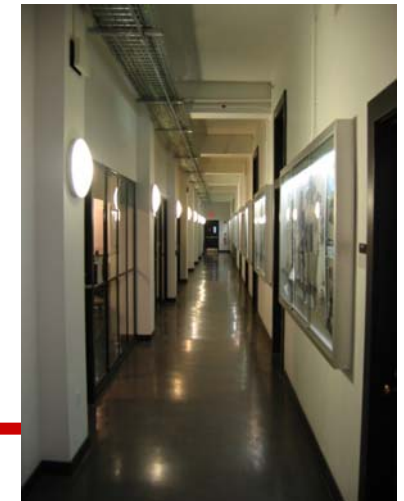
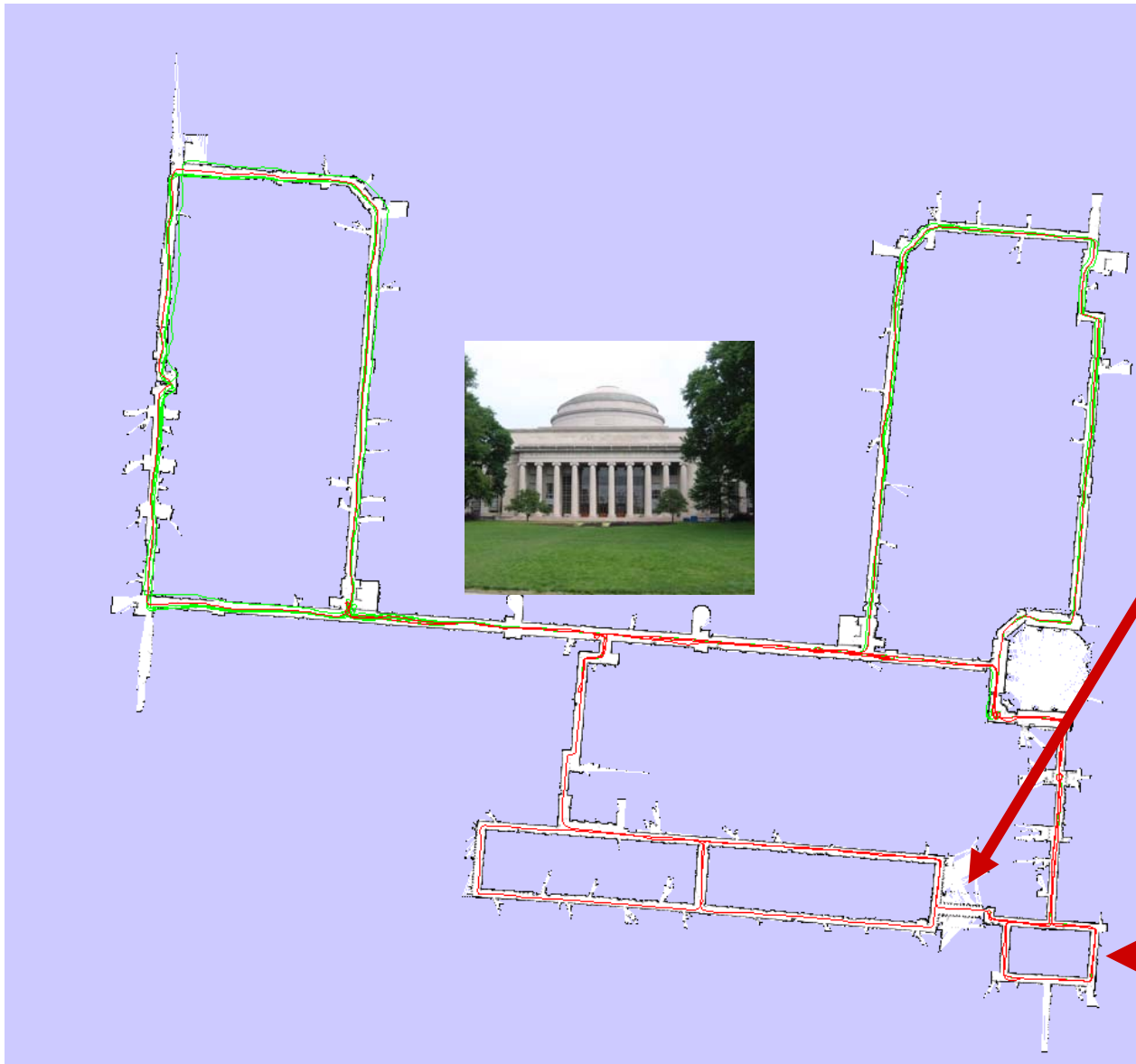


MIT Killian Court

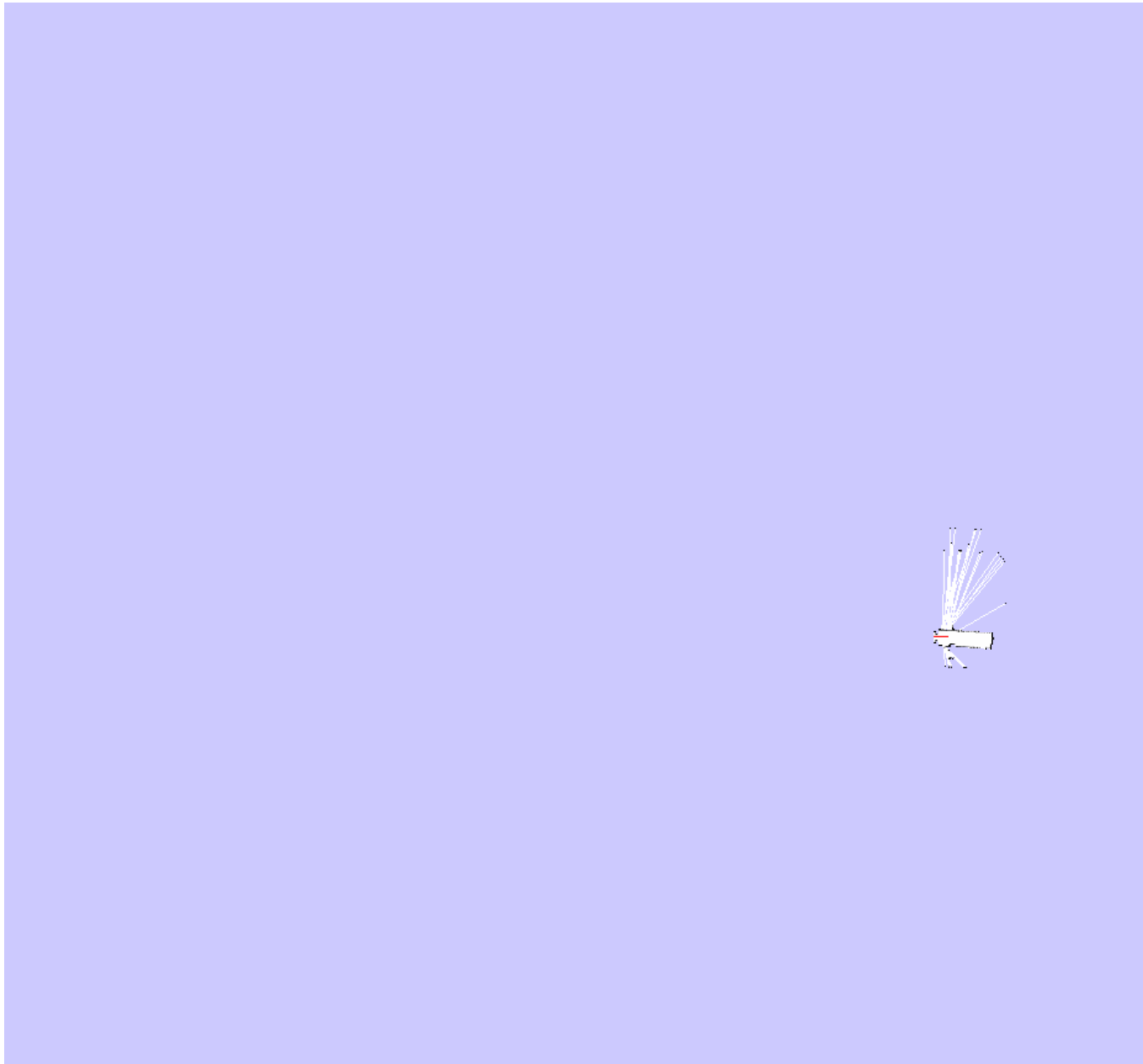


- The **“infinite-corridor-dataset”** at MIT

MIT Killian Court



MIT Killian Court - Video



Conclusion

- The ideas of FastSLAM can also be applied in the context of grid maps
- Utilizing accurate sensor observation leads to good proposals and highly efficient filters
- It is similar to scan-matching on a per-particle base
- The number of necessary particles and re-sampling steps can seriously be reduced
- Improved versions of grid-based FastSLAM can handle larger environments than naïve implementations in “real time” since they need one order of magnitude fewer samples

More Details on FastSLAM

- M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to simultaneous localization and mapping, *AAAI02*
(*The classic FastSLAM paper with landmarks*)
- D. Haehnel, W. Burgard, D. Fox, and S. Thrun. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements, *IROS03*
(*FastSLAM on grid-maps using scan-matched input*)
- G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling, *ICRA05*
(*Proposal using laser observation, adaptive resampling*)
- A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks, *IJCAI03*
(*A representation to handle big particle sets*)