# ECMR 2007 Tutorial

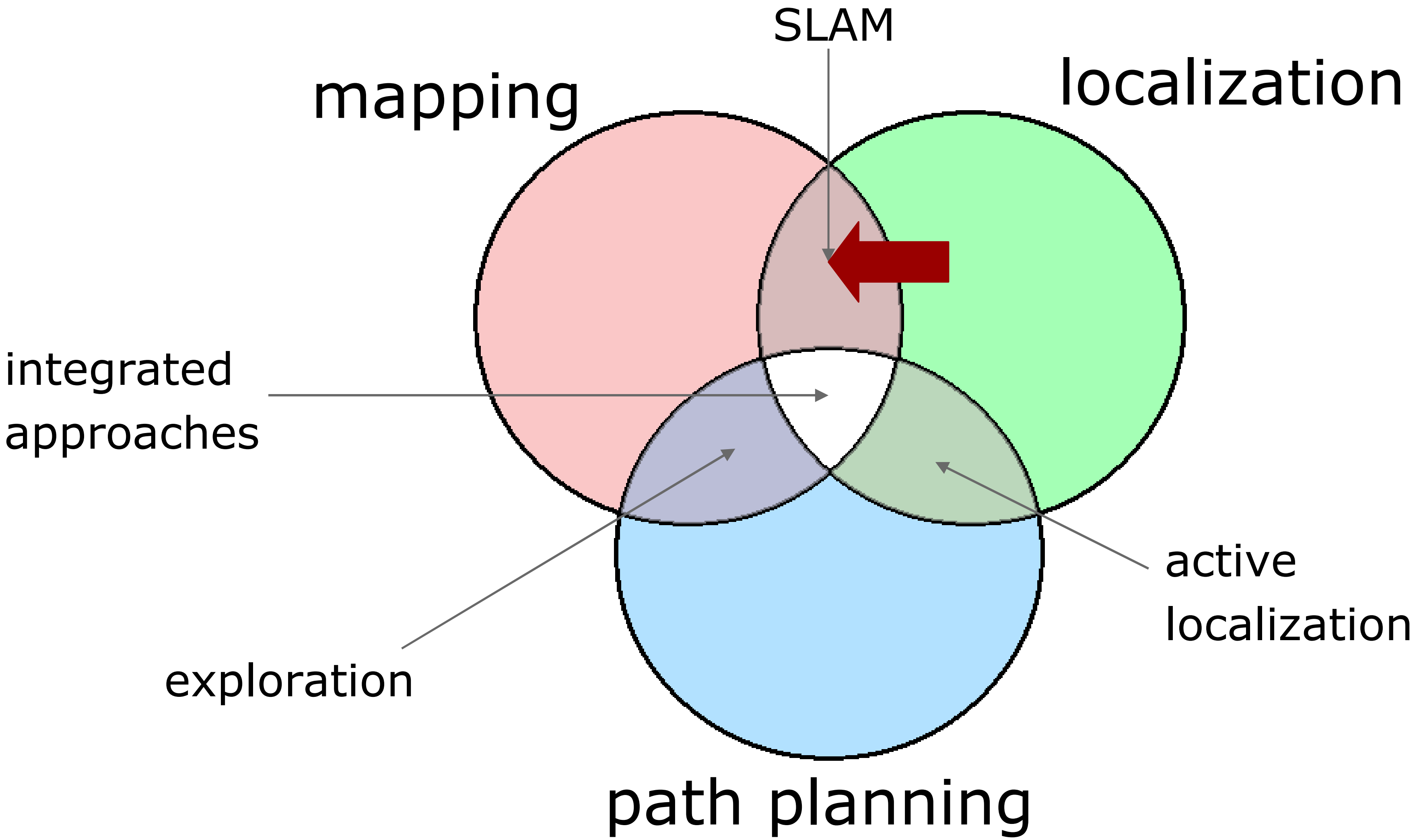
# Learning Grid Maps with Rao-Blackwellized Particle Filters

**Giorgio Grisetti and Cyrill Stachniss**

**University of Freiburg, Germany**

**Special thanks to Dirk Haehnel**

# What is this Talk About?



# What is “SLAM” ?

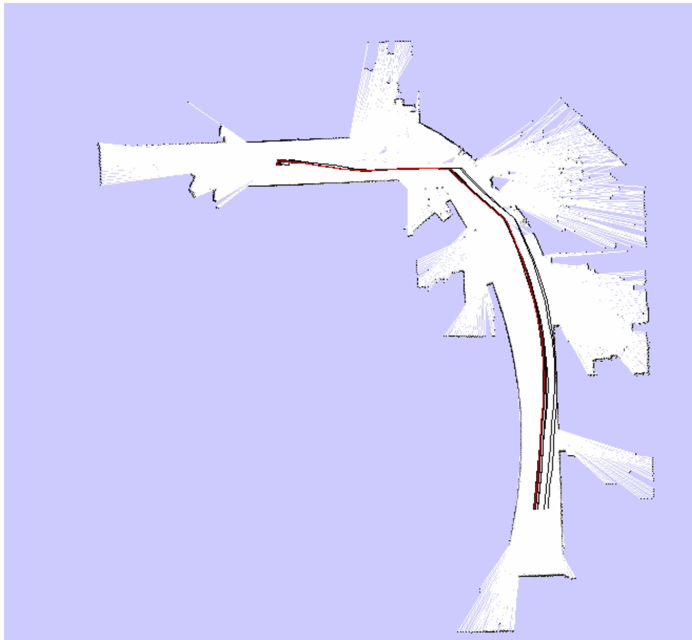
- Estimate the pose and the map of a mobile robot at the same time

$$p(x, m \mid z, u)$$

poses map observations & movements



Courtesy of Dirk Haehnel



[video]

# Particle Filters

Who knows how a particle filter works

?

Explain Particle Filters

Skip Explanation

# Introduction to Particle Filters

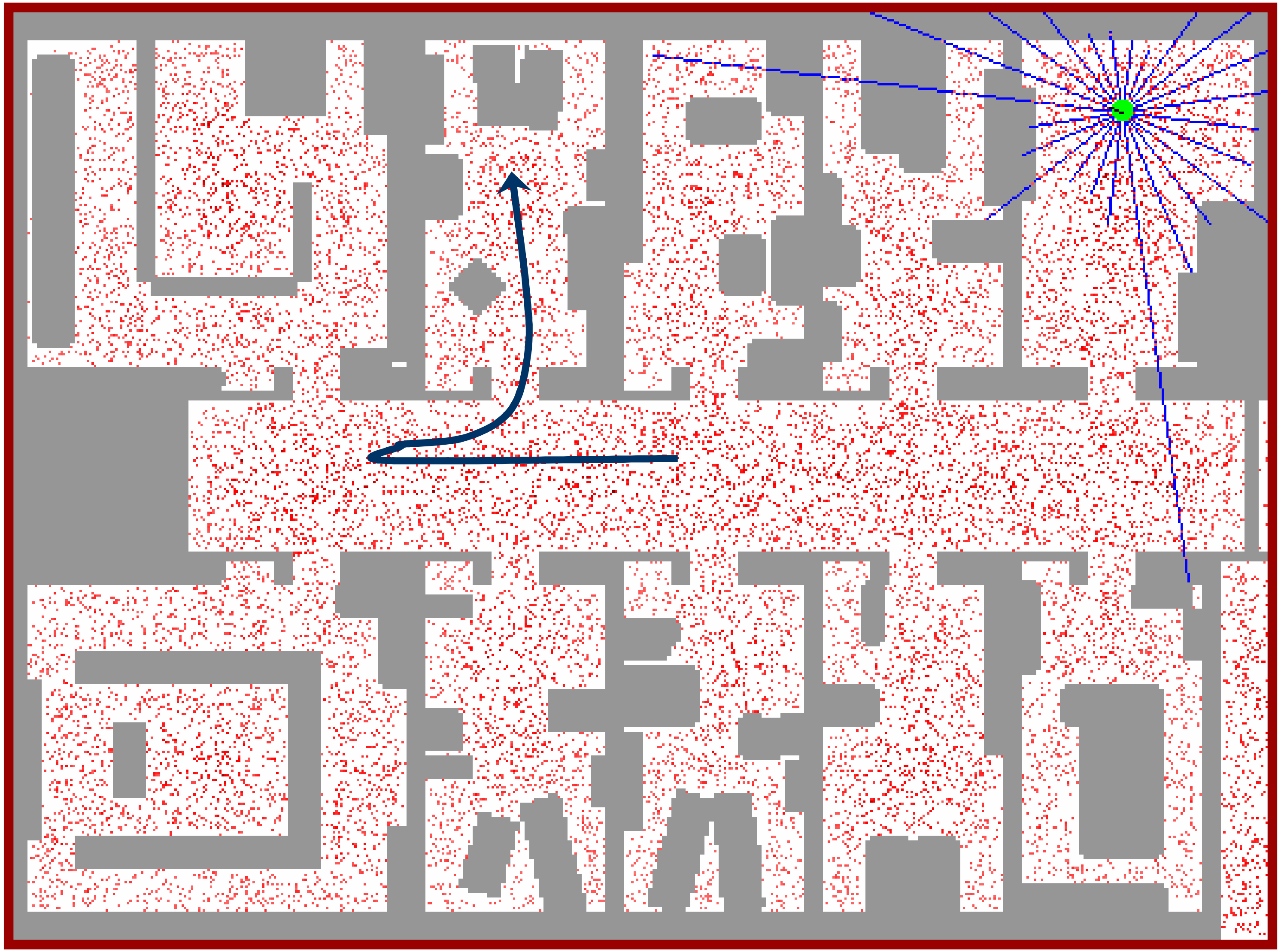
What is a particle filter?

- It is a Bayes filter
- Particle filters are a way to efficiently represent non-Gaussian distribution

Basic principle

- Set of state hypotheses (“particles”)
- Survival-of-the-fittest

# Sample-based Localization (sonar)



[video]

Courtesy of Dieter Fox

# Sample-based Posteriors

- Set of weighted samples

$$S = \{ \langle s^{(i)}, w^{(i)} \rangle \mid i = 1, \dots, N \}$$

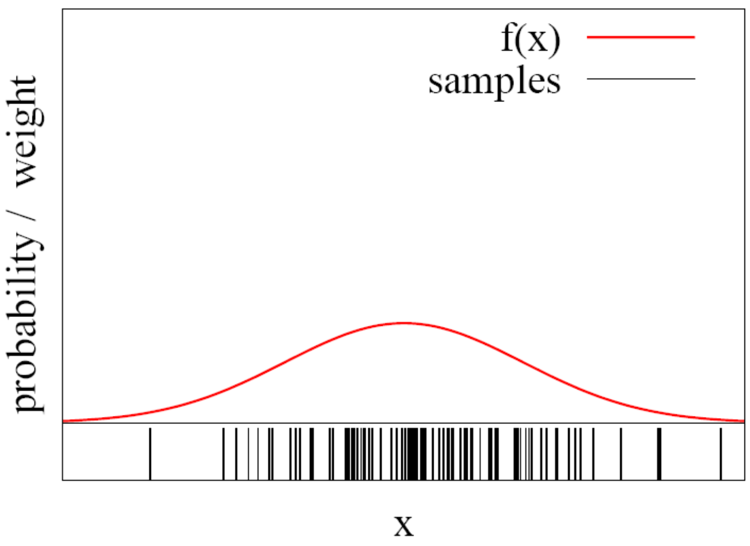
State hypothesis Importance weight

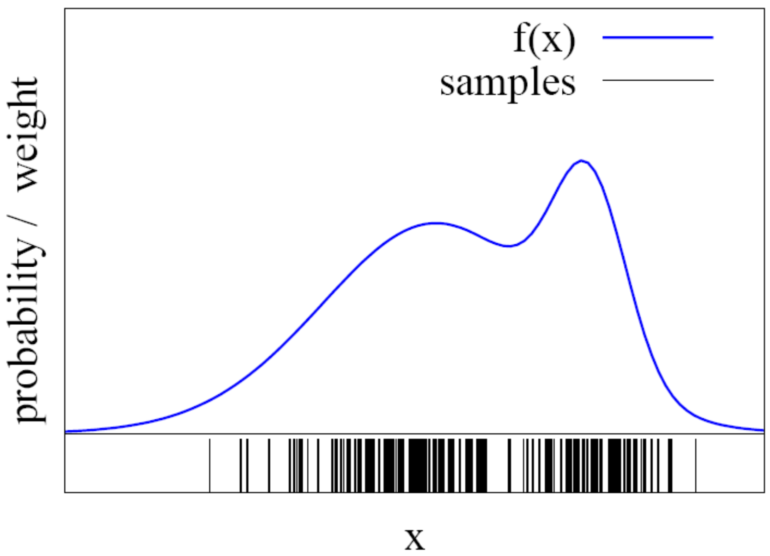
- The samples represent the posterior

$$p(x) = \sum_{i=1}^{N} w_i \cdot \delta_{s^{(i)}}(x)$$

# Posterior Approximation

- Particle sets can be used to approximate functions



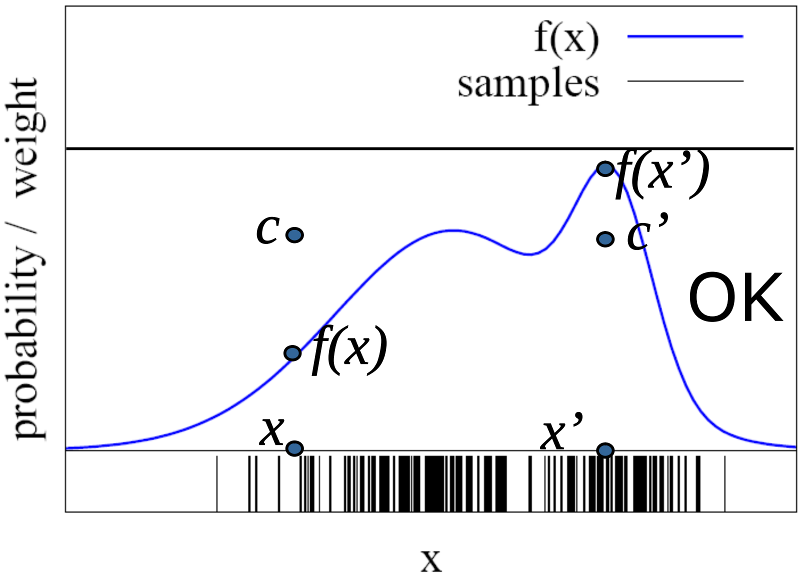




- The more particles fall into an interval, the higher the probability of that interval

- How to draw samples form a function/distribution?

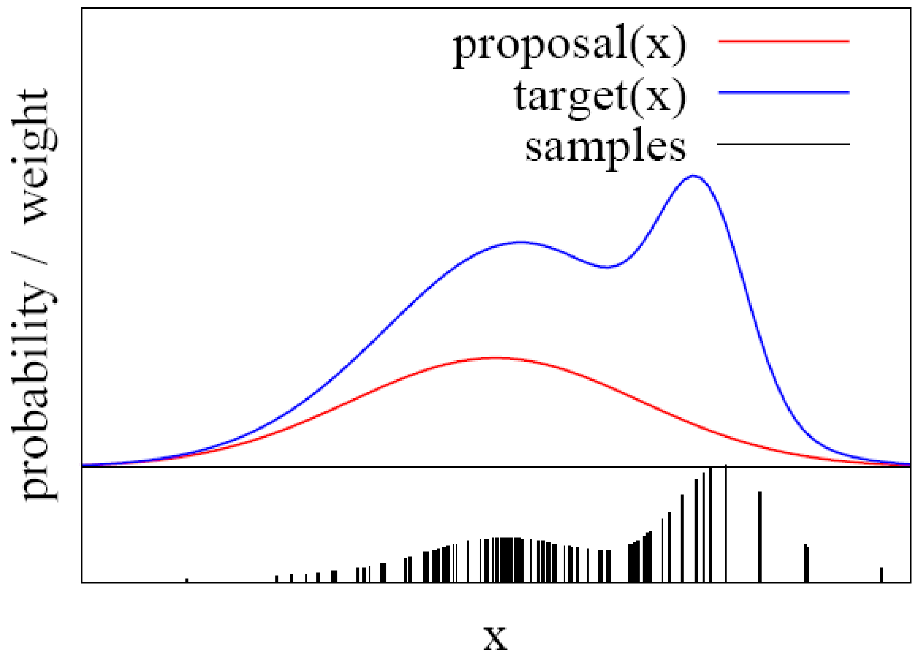
# Rejection Sampling

- Let us assume that $f(x)<1$ for all $x$
- Sample $x$ from a uniform distribution
- Sample $c$ from [0,1]
- if $f(x) > c$ keep the sample
  otherwise reject the sample



# Importance Sampling Principle

- We can even use a different distribution $g$ to generate samples from $f$
- By introducing an importance weight $w$, we can account for the "differences between $g$ and $f$"
- $w = f / g$
- $f$ is called target
- $g$ is called proposal



# From Sampling to a Particle Filter

- Set of samples describes the posterior
- Updates are based on actions and observations

Three sequential steps:

1. Sampling from the proposal distribution
   (Bayes filter: prediction step)

2. Compute the particle weight (importance sampling)
   (Bayes filter: correction step)

3. Resampling

# Monte-Carlo Localization

- For each **motion** $\Delta$ do:
  - **Sampling**: Generate from each sample in a new sample according to the motion model

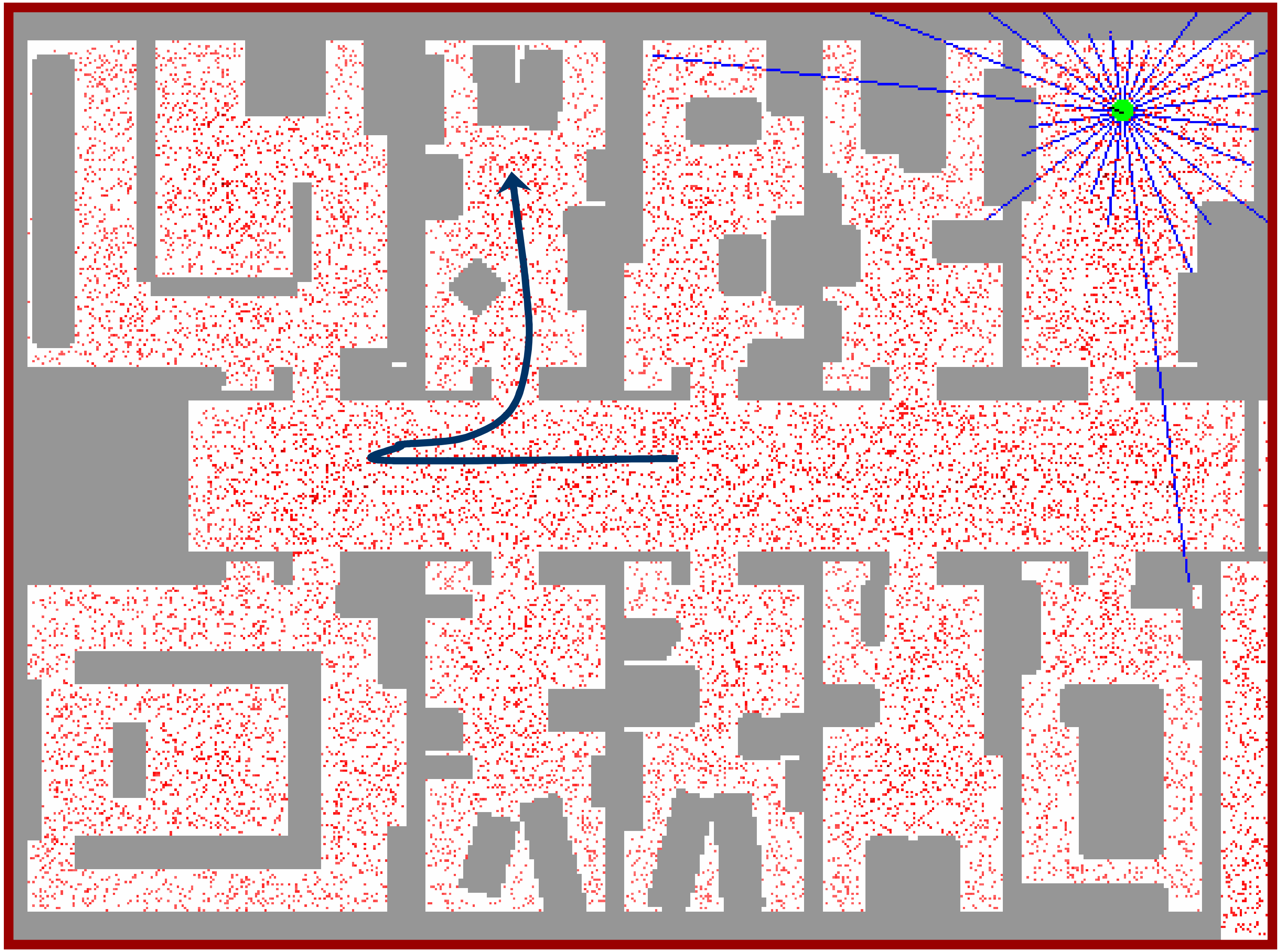
$$x^{(i)} \leftarrow x^{(i)} + \Delta'$$

- For each **observation** do:
  - **Weight** the samples with the observation likelihood

$$w^{(i)} \leftarrow p(z \mid m, x^{(i)})$$

  - **Resampling**

# Sample-based Localization (sonar)

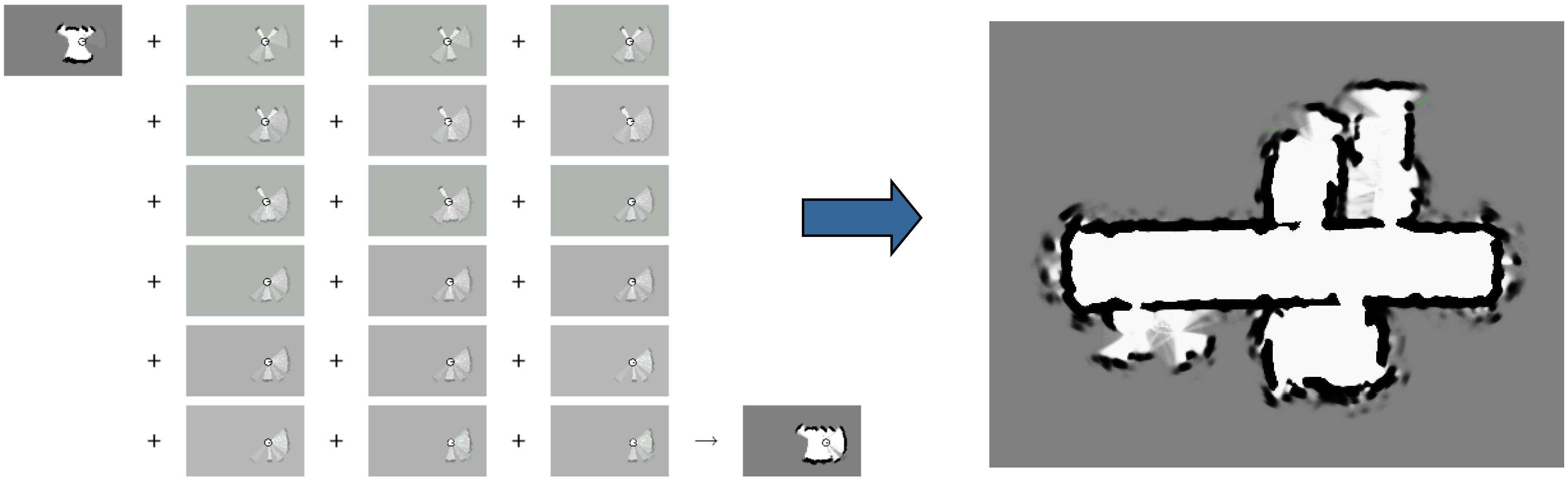


[video]

Courtesy of Dieter Fox

# Grids Maps

- Grid maps are a discretization of the environment into free and occupied cells
- Mapping with known robot poses is easy.



# Mapping using Raw Odometry

- 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

[video] Courtesy of Dirk Haehnel

# SLAM with Particle Filters

- Particle filters have successfully been applied to localization, can we use them to solve the SLAM problem?
- Posterior over poses *x* and maps *m*

$$p(x \mid m, z, u) \Rightarrow p(x, m \mid z, u)$$

(localization) (SLAM)

**Observations:**

- The map depends on the poses of the robot during data acquisition
- If the poses are known, mapping is easy

## Rao-Blackwellization

poses map observations & movements

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

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

Factorization first introduced by Murphy in 1999

# 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

Factorization first introduced by Murphy in 1999

# 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

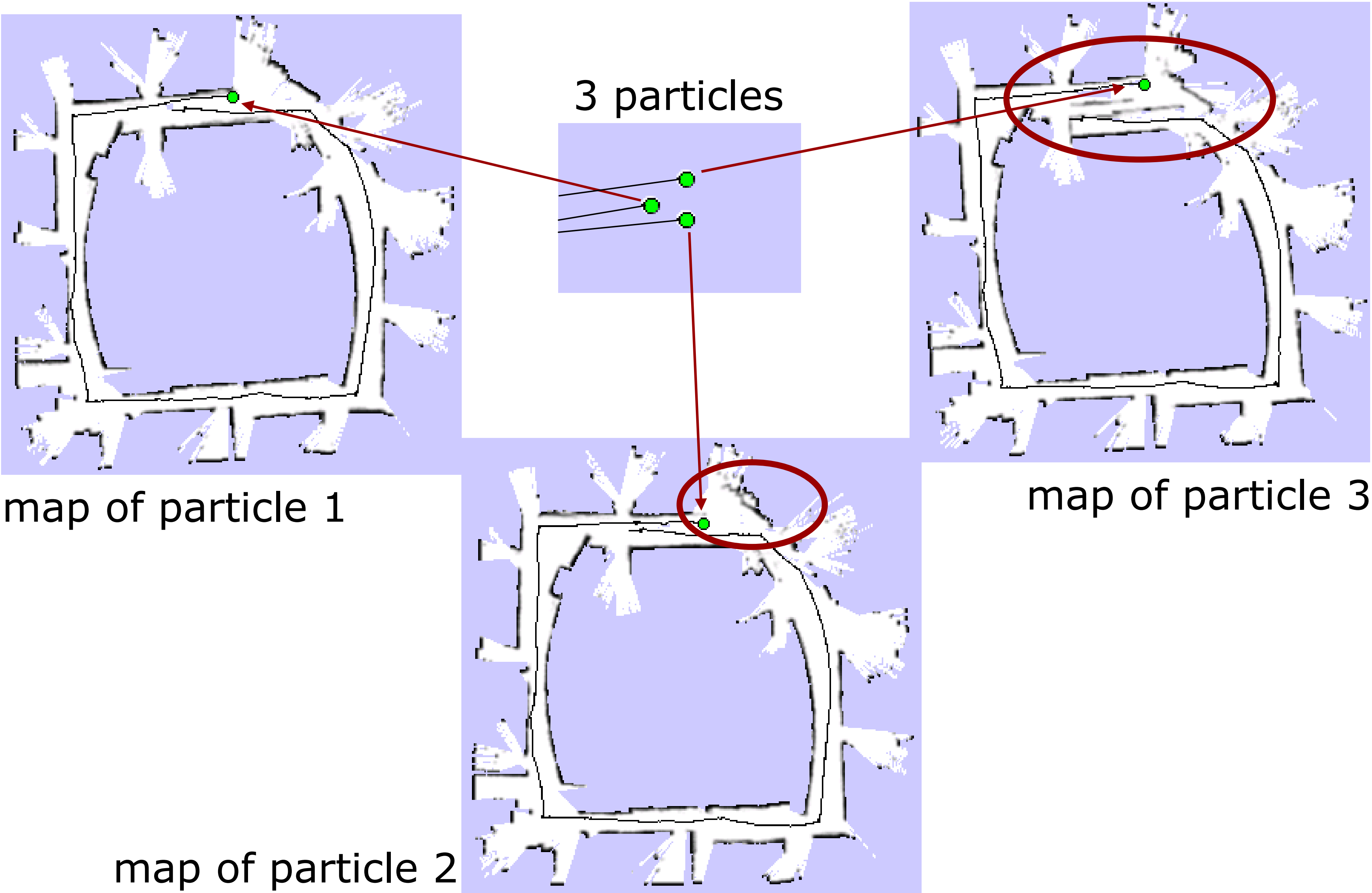
Use the pose estimate from the MCL and apply mapping with known poses

# A Solution to the SLAM Problem

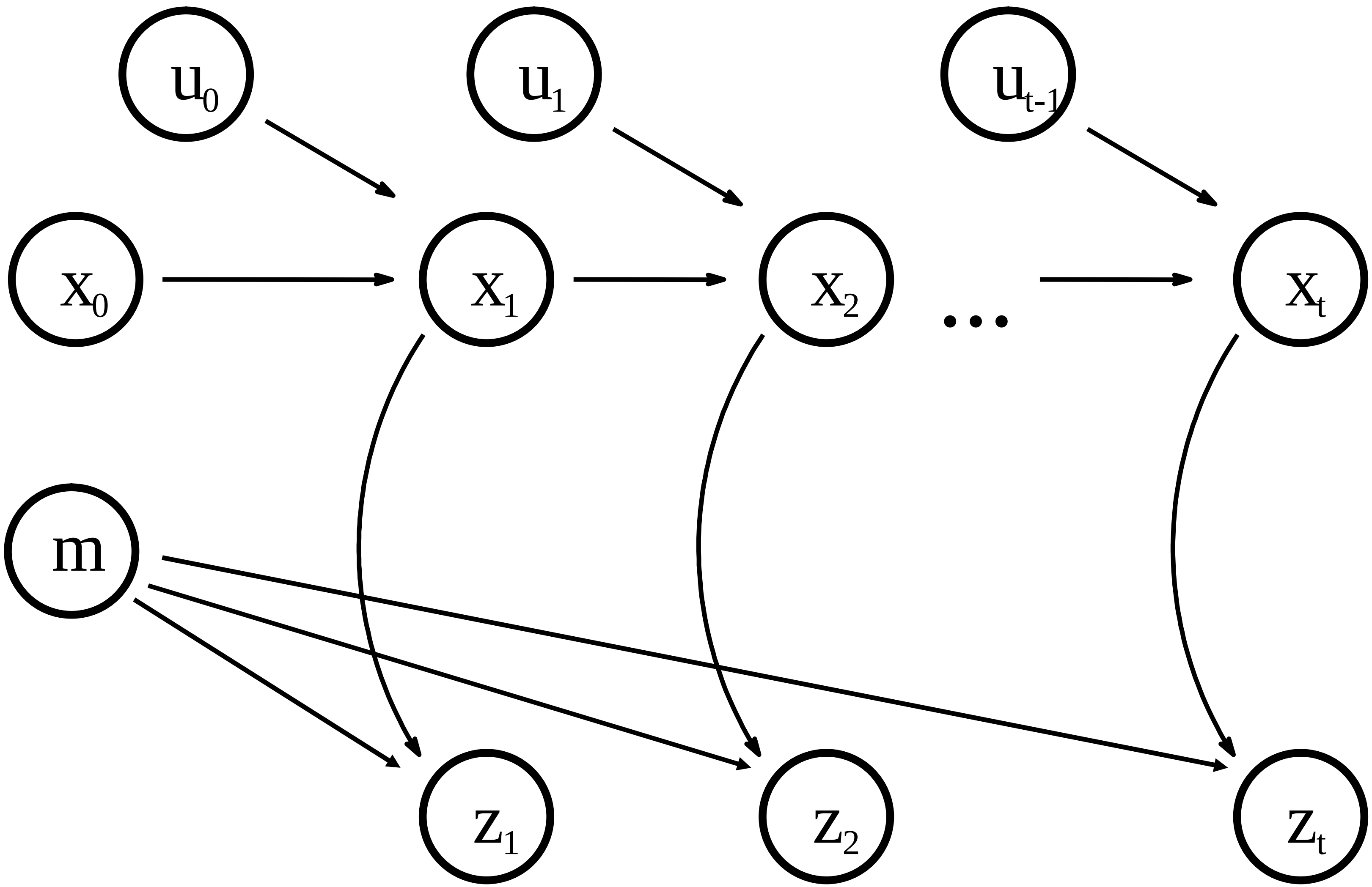
- Use a particle filter to **represent potential trajectories of the robot**
- **Each particle** carries its **own map**
- Each particle survives with a probability proportional to the likelihood of the observations relative to its own map
- We have a **joint posterior** about the poses of the robot and the map

[Murphy, 99; Montemerlo et al., 03; Haehnel et al., 03; Eliazar and Parr, 03; Grisetti et al., 05]

# Example


3 particles
map of particle 1
map of particle 3
map of particle 2

# A Graphical Model of Rao-Blackwellized Mapping



# Problems in Practice

- 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 relative to the (i-1)-th pose

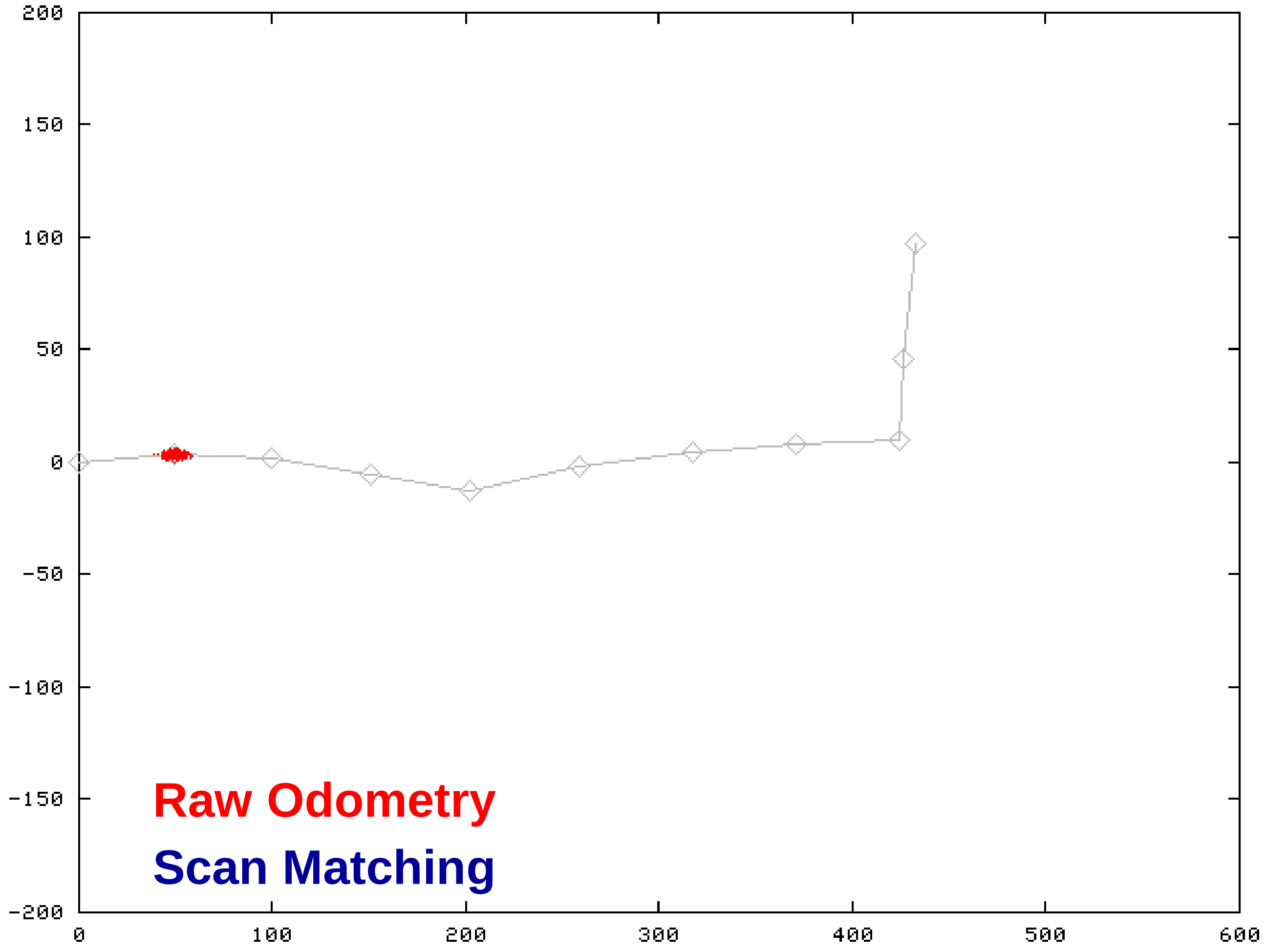
$$x_t^* = \operatorname*{argmax}_{x_t} p(z_t \mid x_t, m_{t-1}) \cdot p(x_t \mid x_{t-1}^*, u_{t-1})$$

current measurement

robot motion

map constructed so far

# Motion Model for Scan Matching




Courtesy of Dirk Haehnel

# Mapping using Scan Matching

[video]

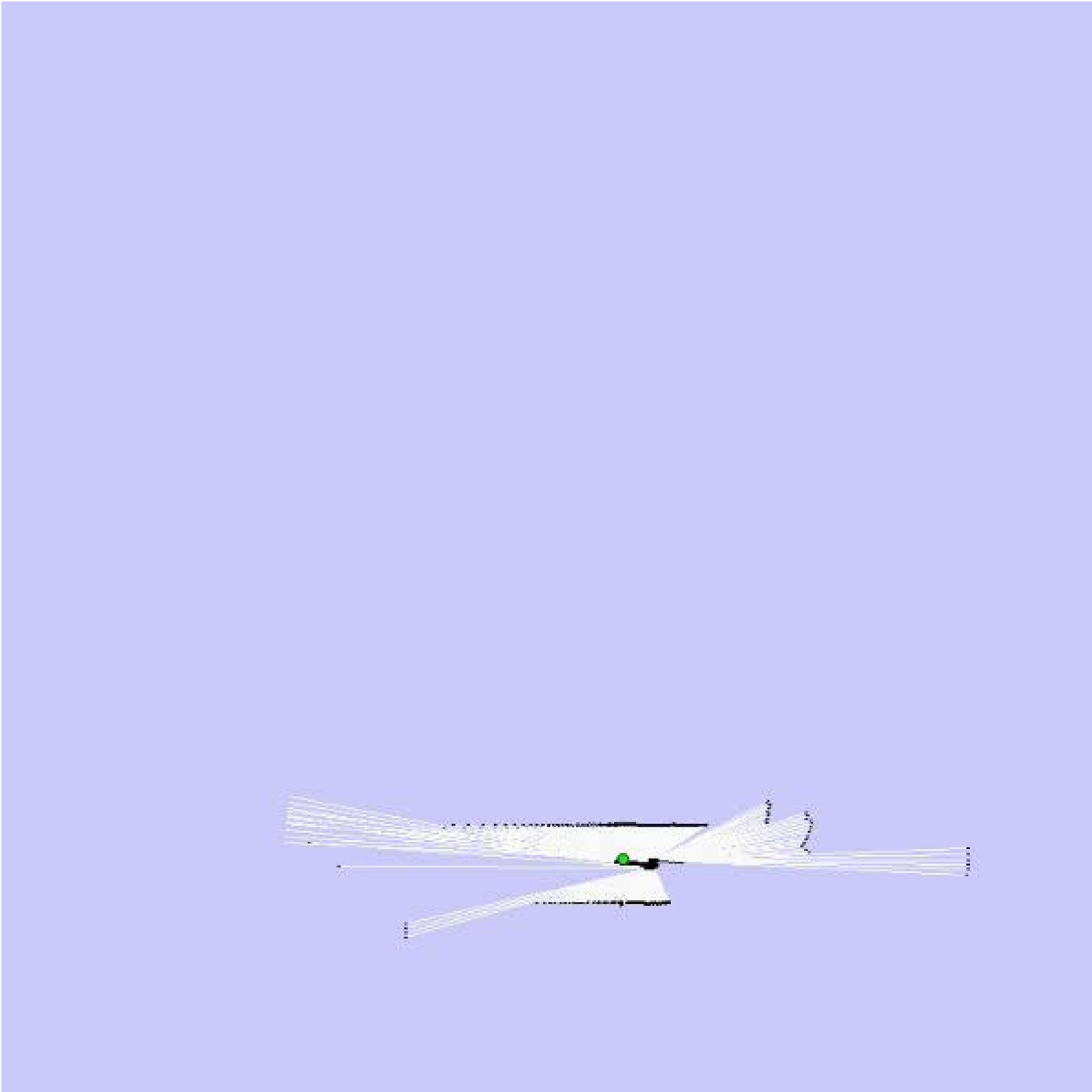
Courtesy of Dirk Haehnel

# RBPF-SLAM 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 the Rao-Blackwellized PF
- Fewer particles are needed, since the error in the input in smaller

[Haehnel et al., 2003]

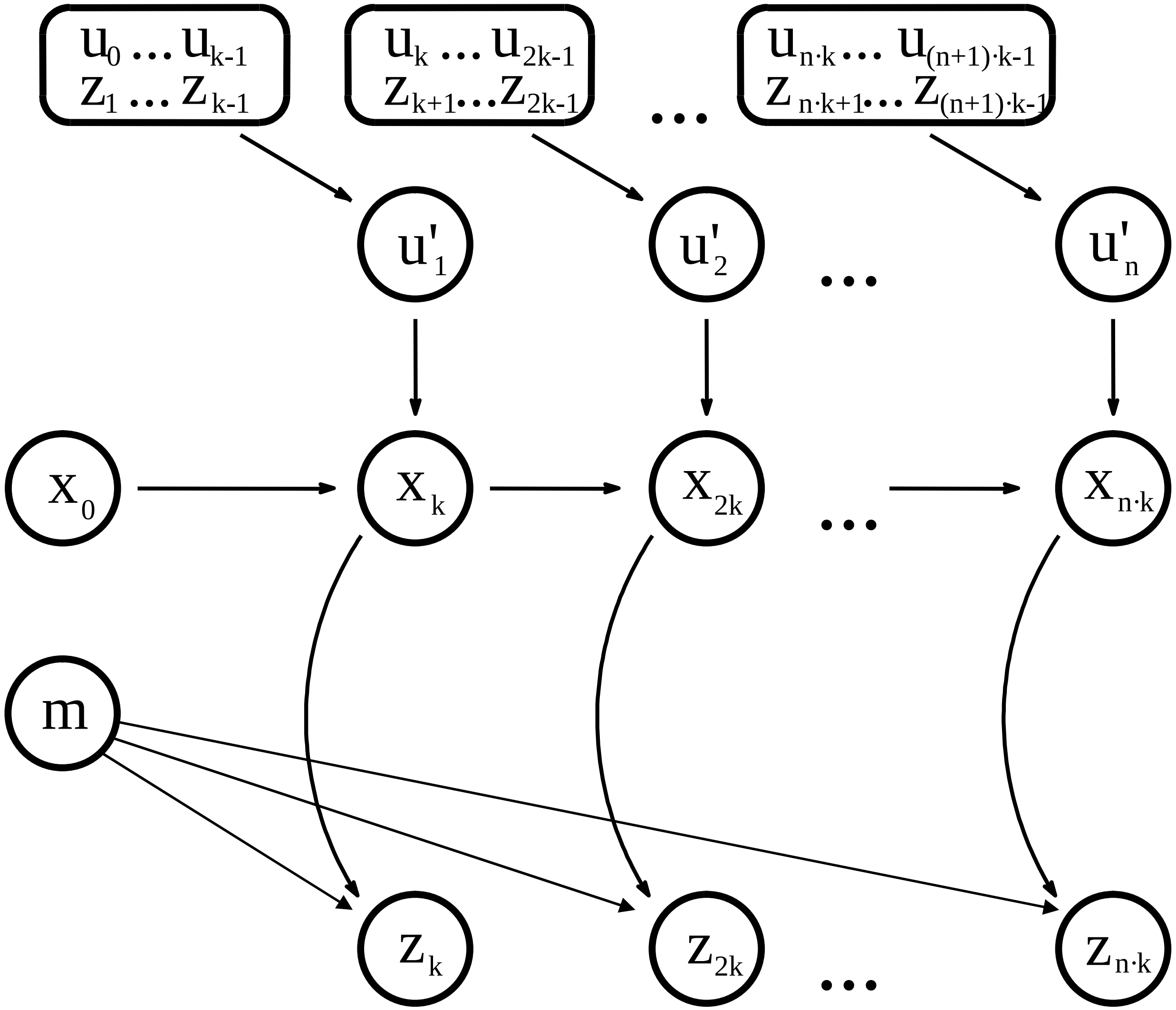
# RBPF-SLAM with Scan-Matching



Map: Intel Research Lab Seattle

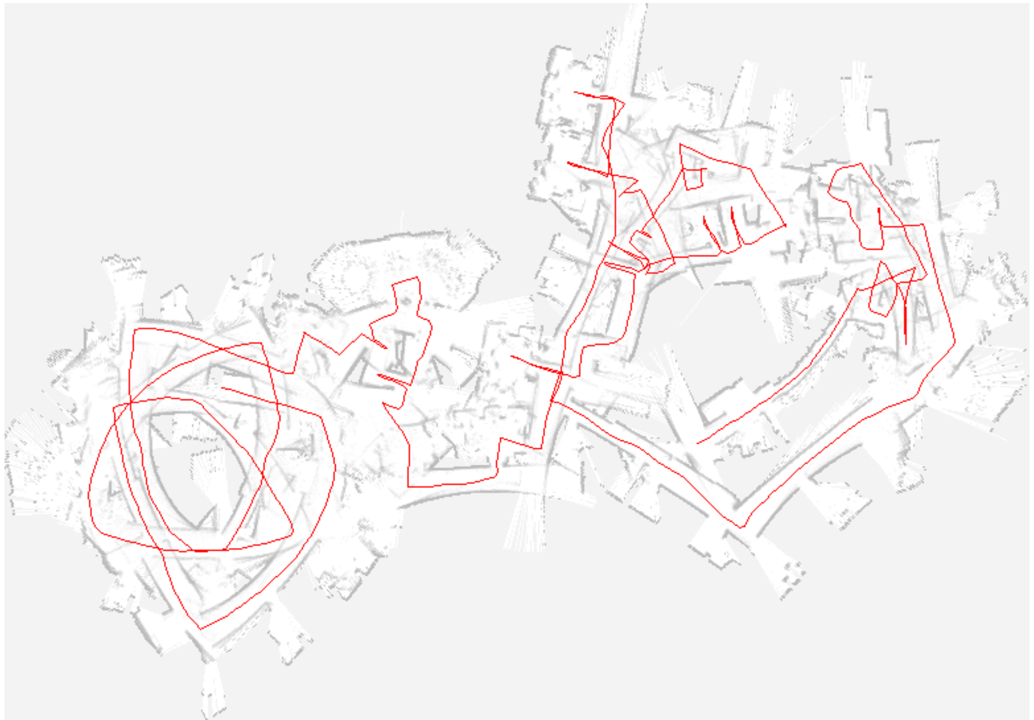
[video] Courtesy of Dirk Haehnel

# Graphical Model for Mapping with Improved Odometry



# Comparison to Standard RBPF-SLAM

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



Courtesy of Dirk Haehnel

# Conclusion (so far…)

- The presented approach is efficient
- It is easy to implement
- Scan matching is used to transform sequences of laser measurements into odometry measurements
- Provides good results for most datasets

# What's Next?

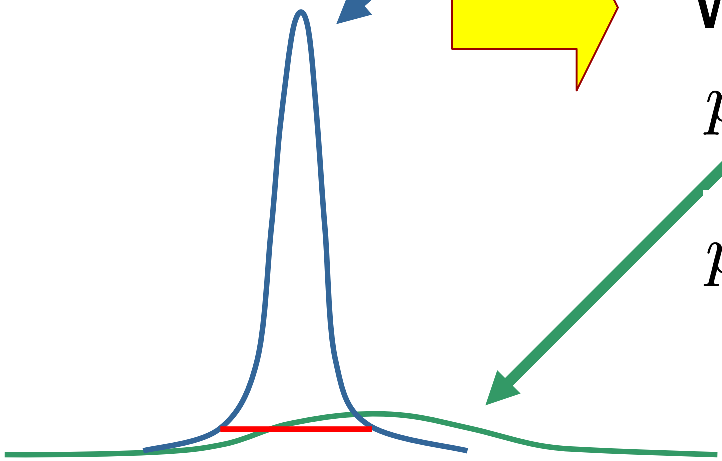
- Further reduce the number of particles
- Improved proposals will lead to more accurate maps
- Use the properties of our sensor when drawing the next generation of particles

# The Optimal Proposal Distribution

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

[Doucet, 98]

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



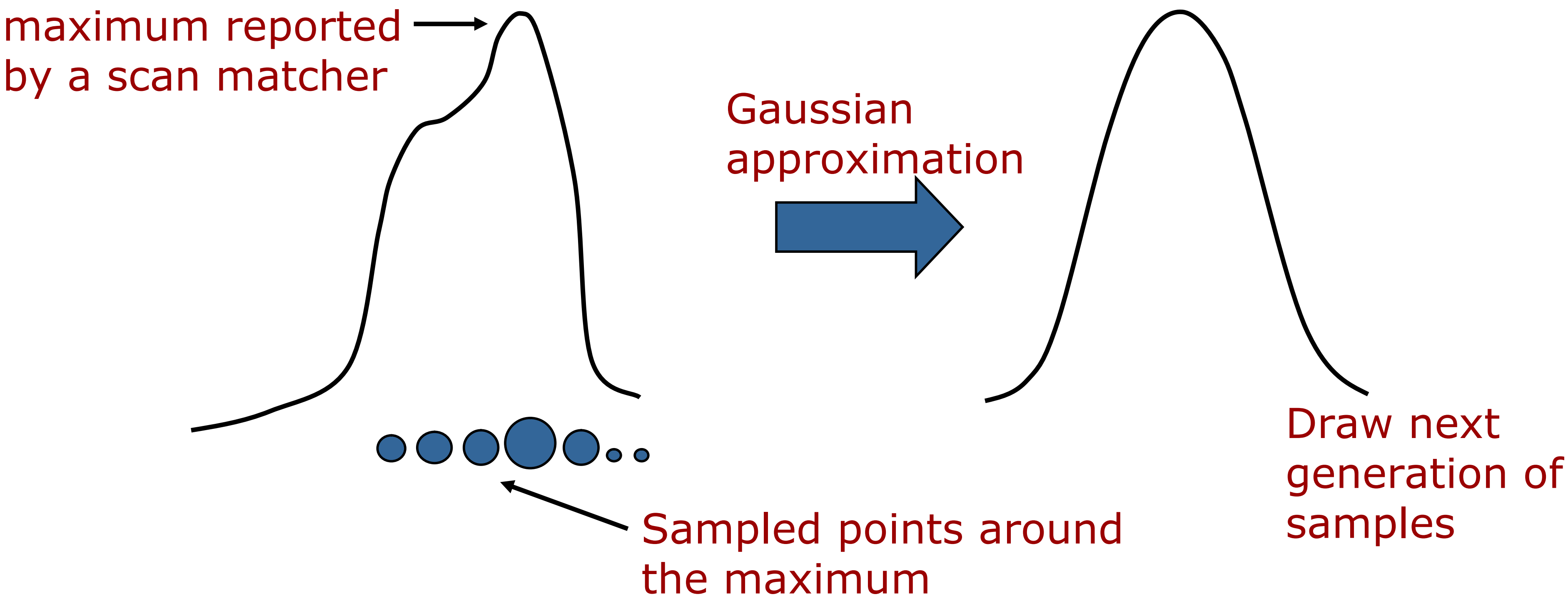
We can safely approximate $p(x_t|x_{t-1}^{(i)}, u_t)$ by a constant:

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

# Resulting Proposal Distribution

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

Approximate this equation by a Gaussian:



# 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:

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

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

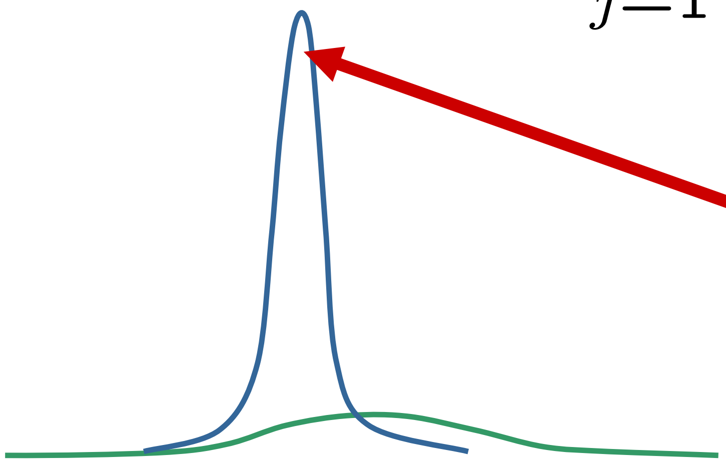
$$\Sigma^{(i)} = \frac{1}{\eta} \sum_{j=1}^{K} (x_j - \mu^{(i)})(x_j - \mu^{(i)})^T p(z_t|x_j, m^{(i)})$$

$h$ is a normalizer

Sampled around the scan-match maxima

# Computing the Importance Weight

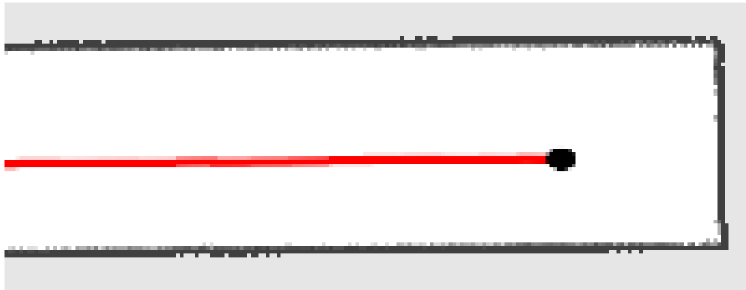
$$
\begin{aligned}
w_t^{(i)} &= w_{t-1}^{(i)} p(z_t | x_{t-1}^{(i)}, m^{(i)}) \\
&= w_{t-1}^{(i)} \int p(z_t | x_t, m^{(i)}) p(x_t | x_{t-1}^{(i)}, u_t) d\, x_t \\
&\simeq w_{t-1}^{(i)} c \int_{x_t \in \{x | p(z_t | x, m^{(i)}) > \epsilon\}} p(z_t | x_t, m^{(i)}) dx_t \\
&\propto w_{t-1}^{(i)} \sum_{j=1}^{K}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



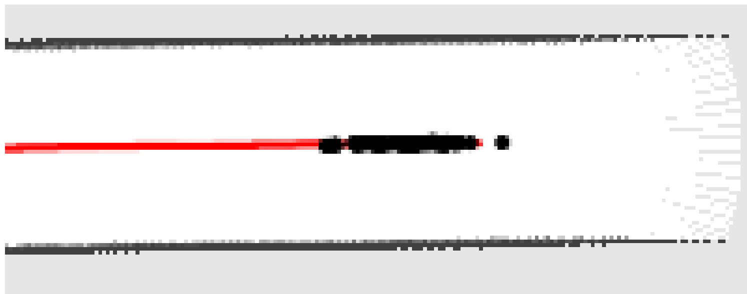
Sampled points around the maximum of the observation likelihood

# Improved Proposal

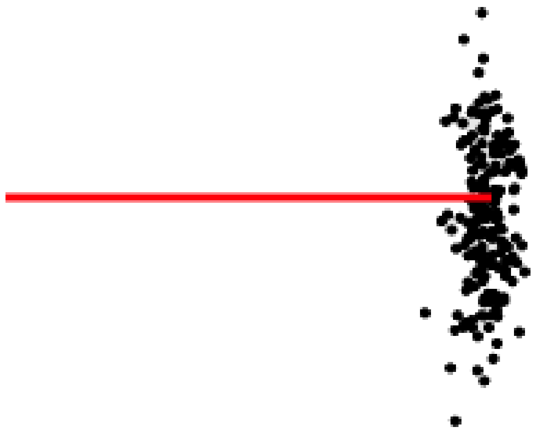
**End of a corridor:**



**Corridor:**



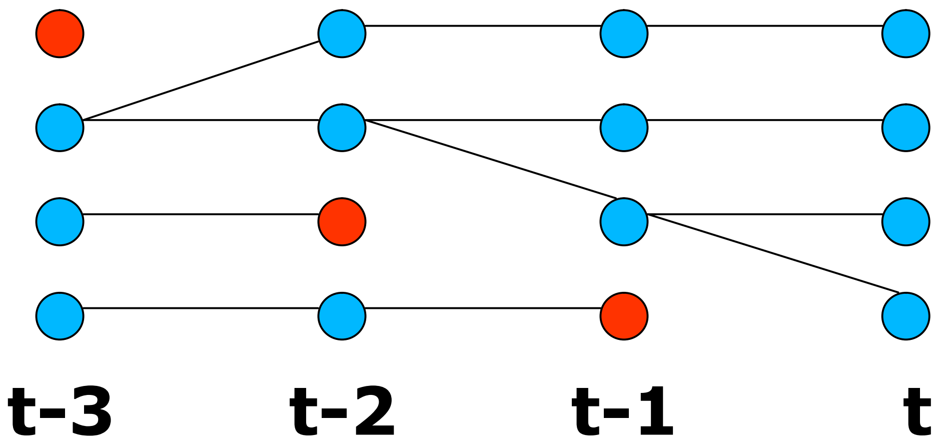
**Free space:**



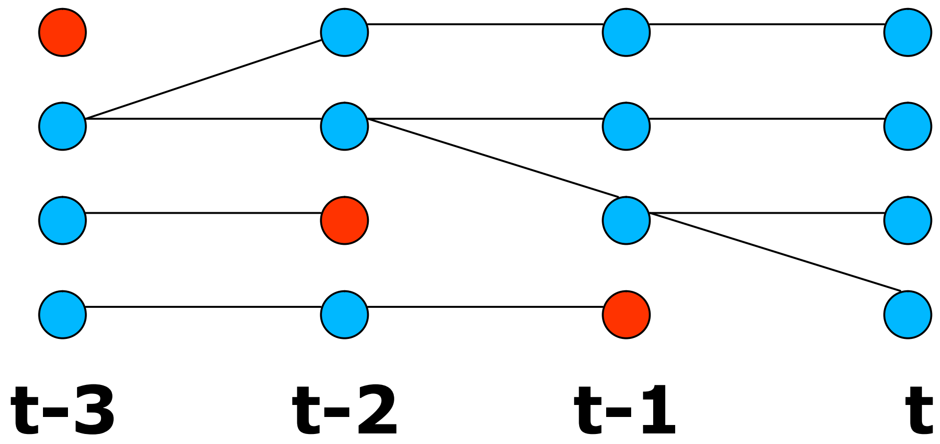
# Resampling

- In case of suboptimal/bad proposal distributions resampling is necessary to achieve convergence

- Resampling is dangerous, since important samples might get lost
(particle depletion problem)



# When to Resample?




- Key question: When should we resample?

- Resampling makes only sense if the samples have significantly different weights

# Effective Number of 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

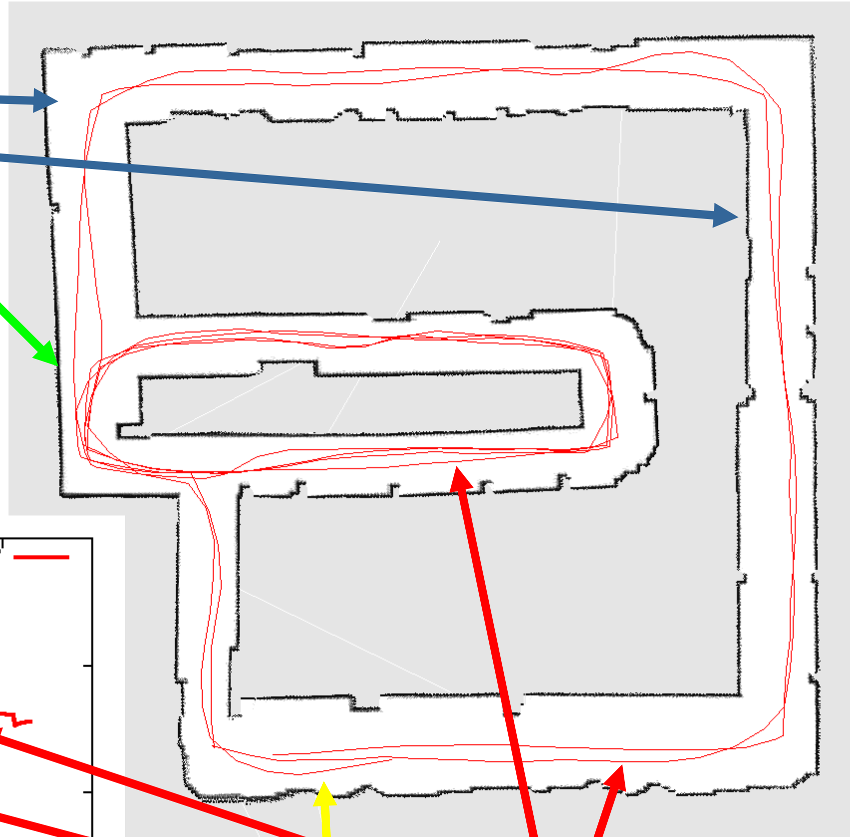
# Resampling with $N_{eff}$

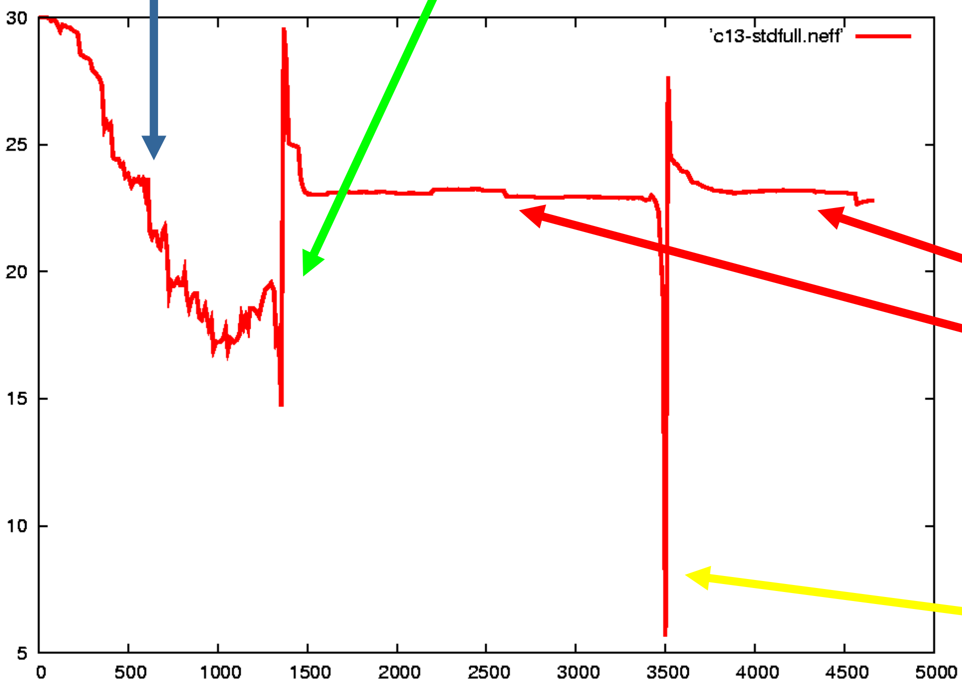
- If our approximation is close to the proposal, no resampling is needed
- We only resample when $N_{eff}$ drops below a given threshold (N/2)
- See [Doucet, ’98; Arulampalam, ’01]

# Typical Evolution of $N_{eff}$

visiting new areas

closing the first loop






visiting known areas

second loop closure

# Intel Research Lab

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

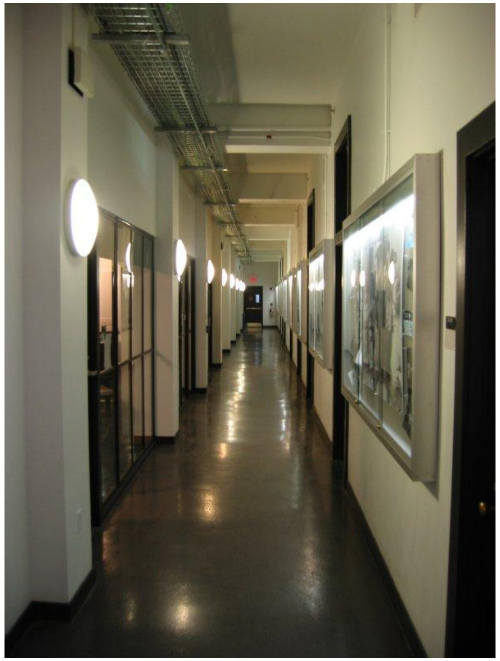
[video]

# Outdoor Campus Map

[video]

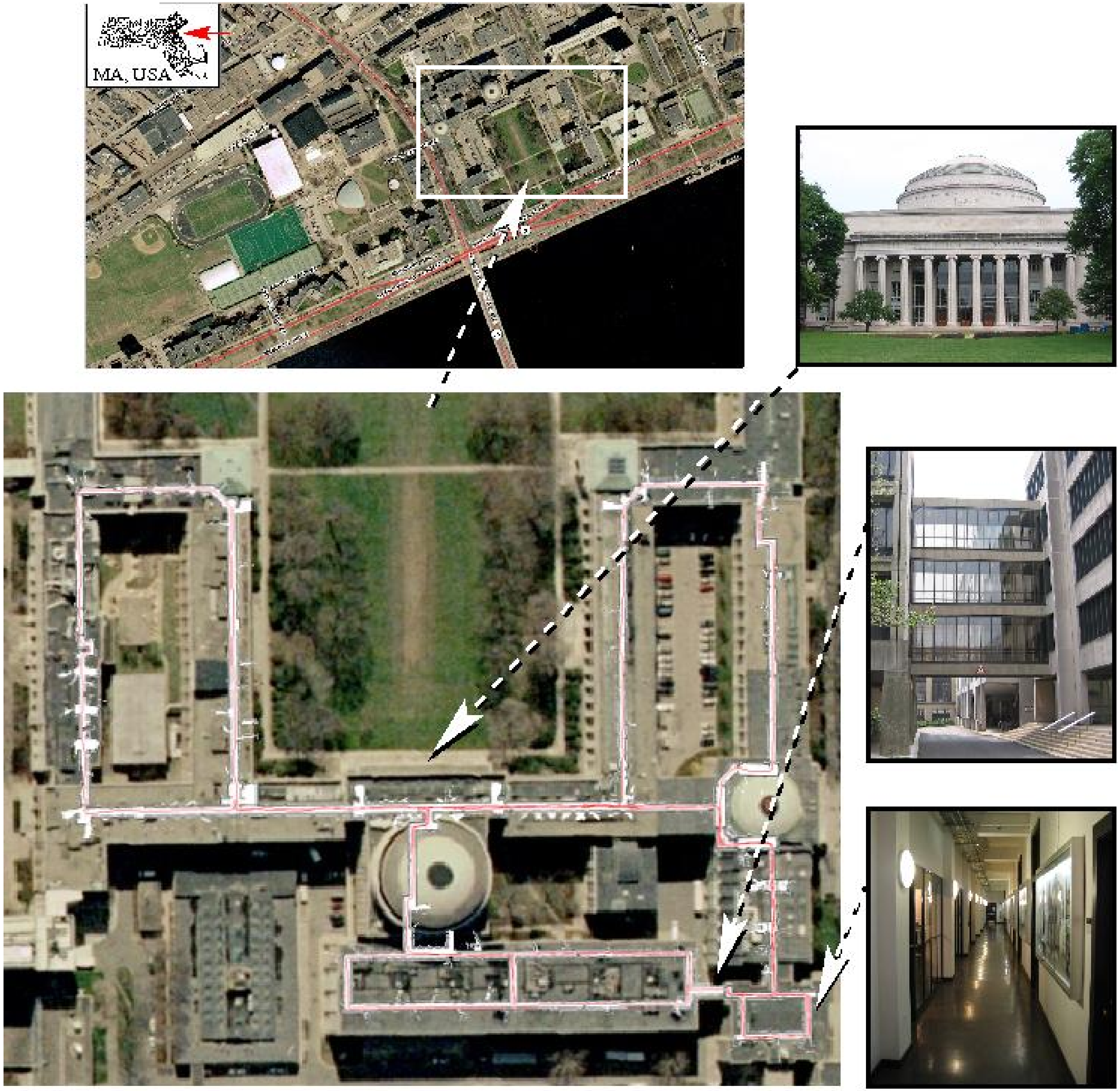
- **30 particles**
- 250x250m$^2$
- 1.088 miles (odometry)
- 20cm resolution during scan matching
- 30cm resolution in final map

# MIT Killian Court

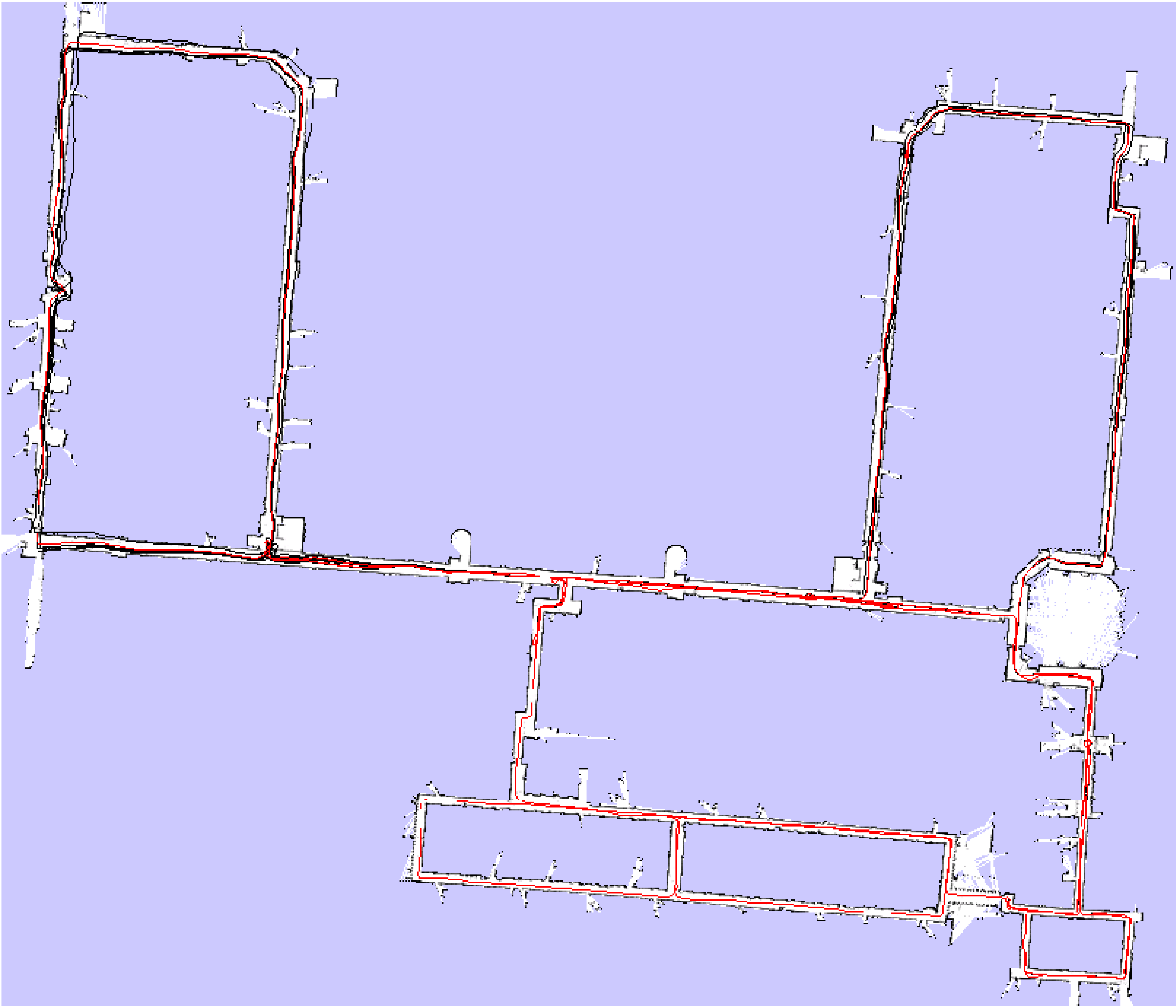



- The **"infinite-corridor-dataset"** at MIT

# MIT Killian Court



# MIT Killian Court

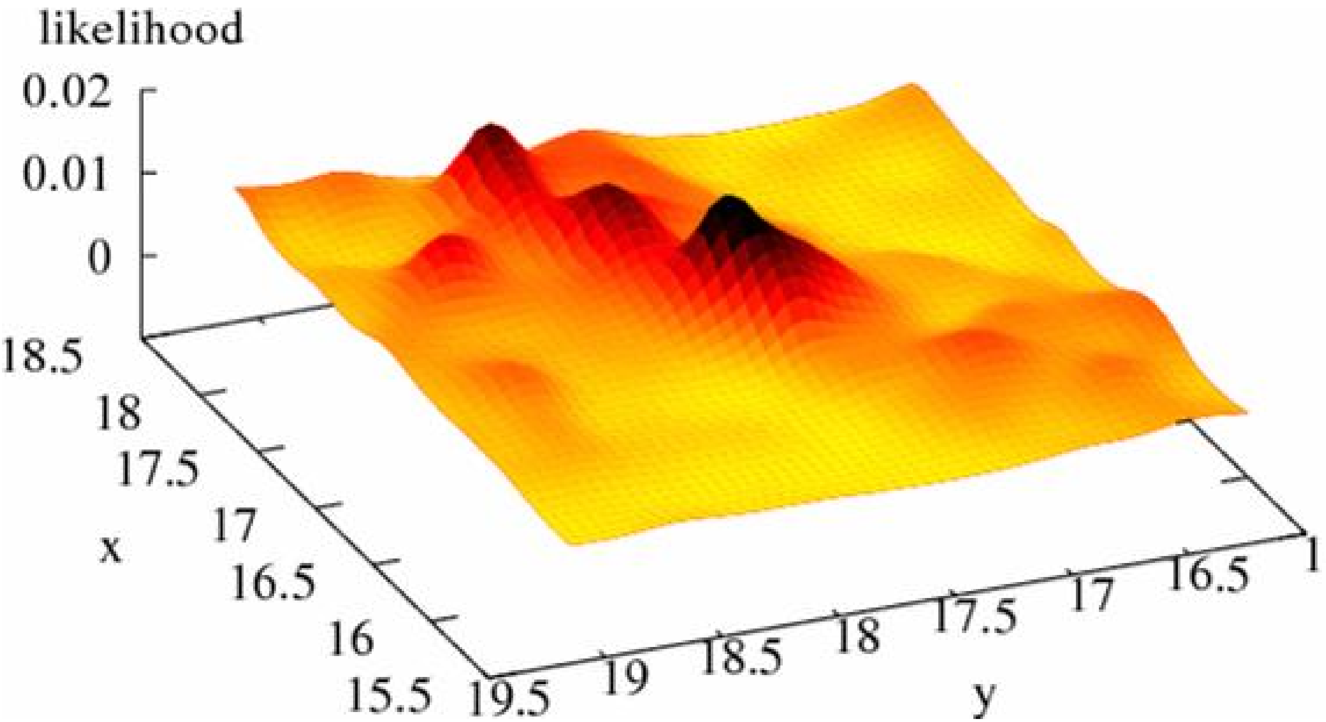


Dataset courtesy of Mike Bosse and John Leonard

[video]

# Problems of the Gaussian Proposal

- Gaussians are uni-model distributions
- In case of loop-closures, the likelihood function might be multi-modal

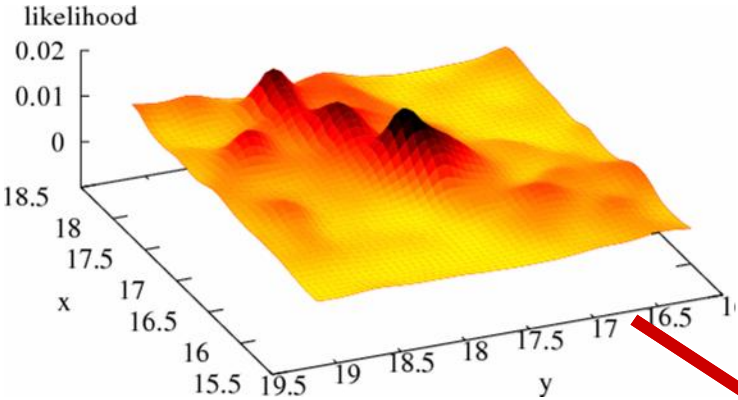


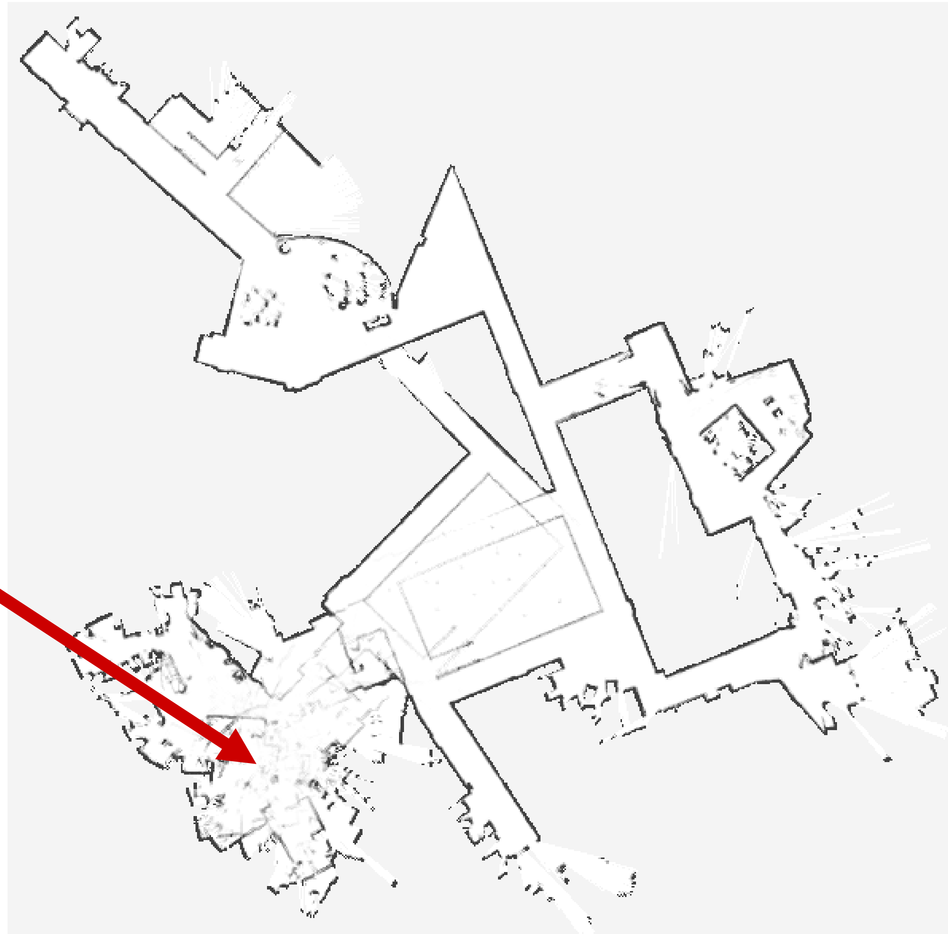
# Is a Gaussian an Accurate Representation for the Proposal?

| Dataset | Gauss | Non-Gauss 1 mode | Multi-modal |
|---|---|---|---|
| Intel Research Lab | 89.2% | 7.2% | 3.6% |
| FHW Museum | 84.5% | 10.4% | 5.1% |
| Belgioioso | 84.0% | 10.4% | 5.6% |
| MIT CSAIL | 78.1% | 15.9% | 6.0% |
| MIT Killian Court | 75.1% | 19.1% | 5.8% |
| Freiburg Bldg. 79 | 74.0% | 19.4% | 6.6% |

# Problems of the Gaussian Proposal

- Multi-modal likelihood function can cause filter divergence



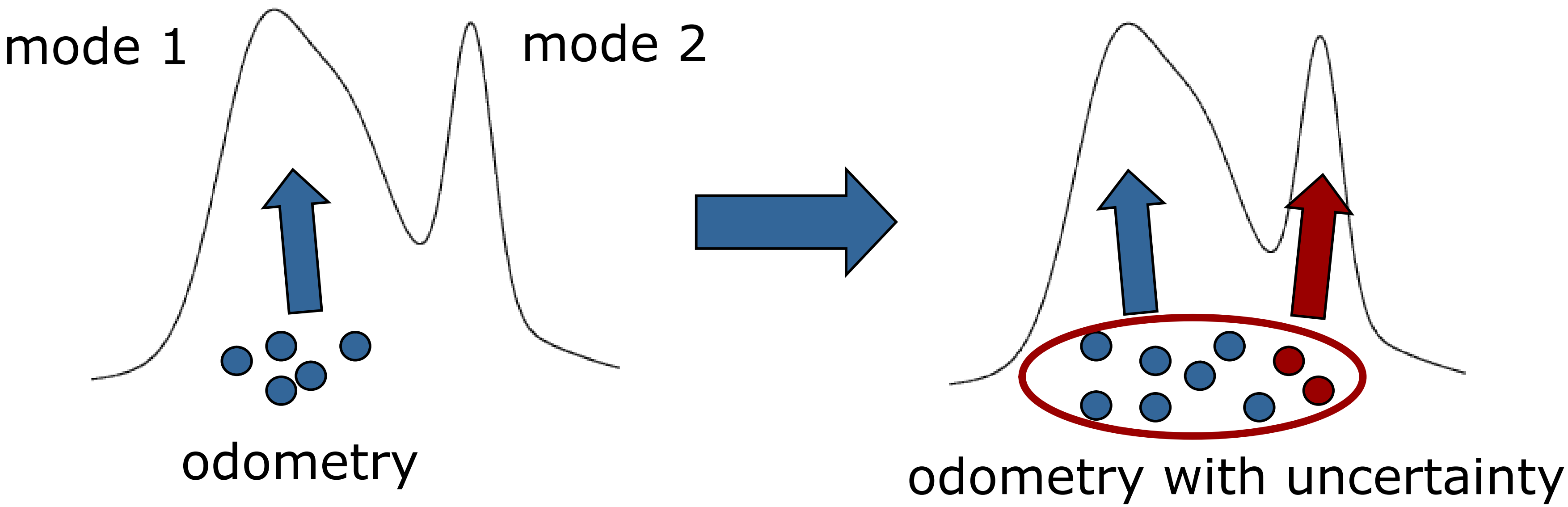
# How to Overcome this Limitation?

- Sampling from the optimal proposal:
  - Compute the full 3d histogram
  - Sample from the histogram

| Dataset | N | Execution time | |
|---|---|---|---|
| | | optimal | Gausian proposal |
| MIT Killian Court | 80 | 155 h | 112 min |
| Freiburg Bldg. 79 | 30 | 84 h | 62 min |
| Intel Research Lab | 30 | 40 h | 29 min |
| FHW Museum | 30 | 38 h | 27 min |
| Belgioioso | 30 | 18 h | 13 min |
| MIT CSAIL | 30 | 10 h | 7 min |

# How to Overcome this Limitation?

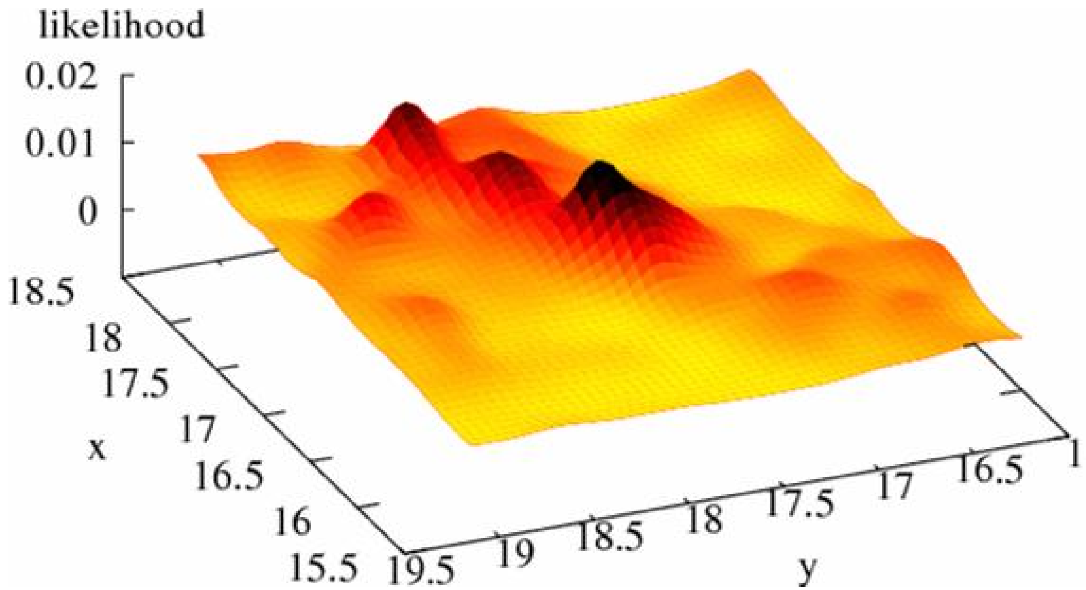
- Approximate the likelihood in a better way!

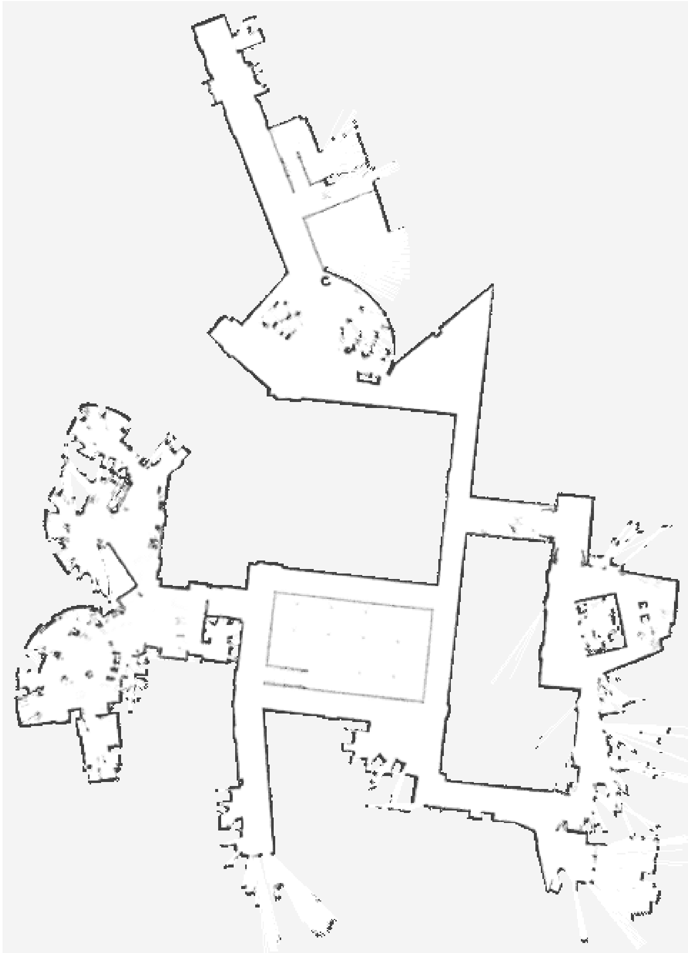



- Sample from odometry first and the use this as the start point for scan matching

# Final Approach

- It work’s with nearly zero overhead



# Conclusion

- Rao-Blackwellized Particle Filters are means to represent a joint posterior about the poses of the robot and the map
- Utilizing accurate sensor observation leads to good proposals and highly efficient filters
- It is similar to scan-matching on a per-particle base with some extra noise
- The number of necessary particles and re-sampling steps can seriously be reduced
- How to deal with non-Gaussian observation likelihood functions
- Highly accurate and large scale map

# More Details

- 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)
- M. Montemerlo, S. Thrun D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, IJCAI03. (FastSLAM 2.0 – improved proposal for FastSLAM)
- D. Haehnel, W. Burgard, D. Fox, and S. Thrun. An efcient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements, IROS03 (FastSLAM on grid-maps using scan-matched input)
- A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultainous localization and mapping without predetermined landmarks, IJCAI03 (A representation to handle big particle sets)

# More Details (Own Work)

- Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters, Transactions on Robotics, Volume 23, pages 34-46, 2007
(Informed proposal using laser observation, adaptive resampling)

- G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling, ICRA'05
(Informed proposal using laser observation, adaptive resampling)

- Cyrill Stachniss, Grisetti Giorgio, Wolfram Burgard, and Nicholas Roy. Analyzing Gaussian Proposal Distributions for Mapping with Rao-Blackwellized Particle Filters, IROS07
(Gaussian assumption for computing the improved proposal)

# From Theory to Practice

- Implementation available a open source project "GMapping" on
  www.OpenSLAM.org

- Written in C++
- Can be used as a black box library

**Now: 1h Practical Course on GMapping**