Robot Mapping

Grid-based FastSLAM

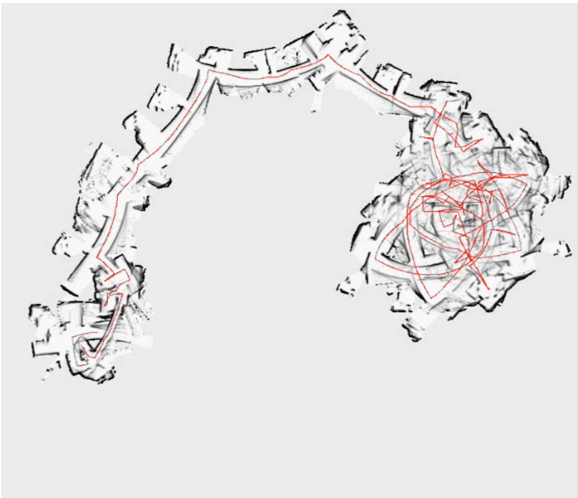
Cyrill Stachniss

Motivation

- So far, we addressed landmark-based SLAM (EKF, SEIF, FastSLAM)
- We learned how to build grid maps assuming "known poses"

Today: SLAM for building grid maps

Mapping With Raw Odometry



Courtesy: Dirk Hähnel

Observation

- **Assuming known poses fails!**

Questions

- Can we solve the SLAM problem if no pre-defined landmarks are available?
- Can we use the ideas of FastSLAM to build grid maps?

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

poses — map — observations & movements

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

First introduced for SLAM by Murphy in 1999

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

poses — map — observations & movements

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

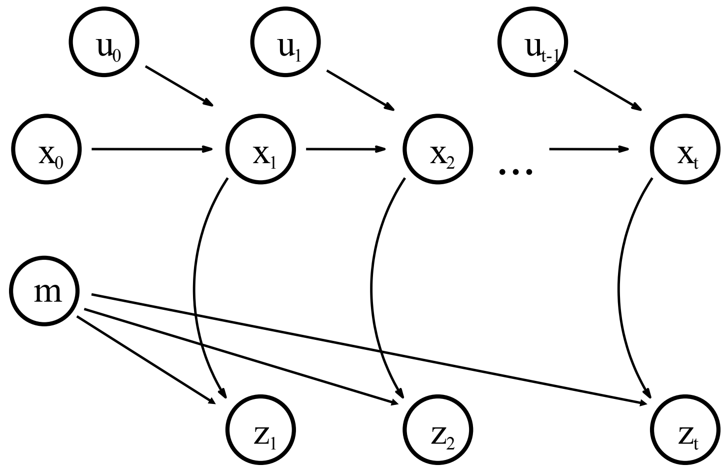
path posterior (particle filter) — map posterior (given the path)

First introduced for SLAM by Murphy in 1999

Grid-based SLAM

- 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")

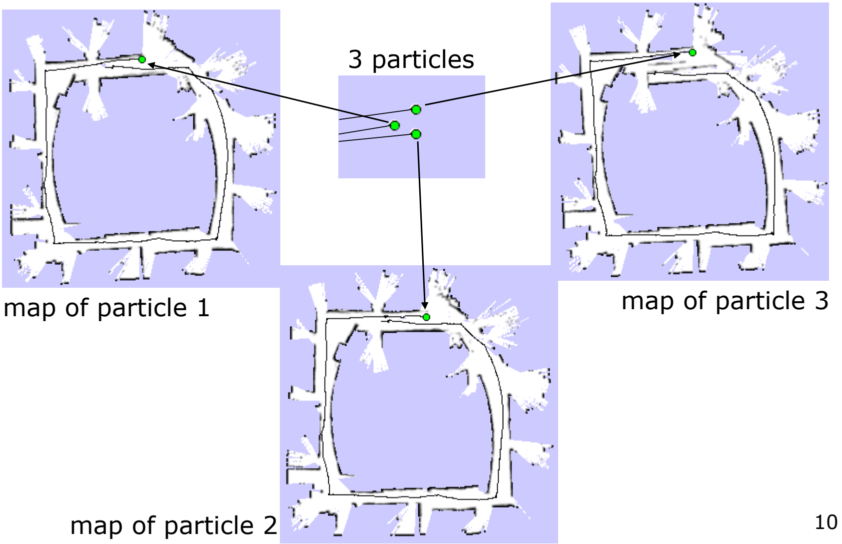
A Graphical Model for Grid-Based SLAM



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 it upon "mapping with known poses"

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

Pose Correction Using Scan-Matching

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

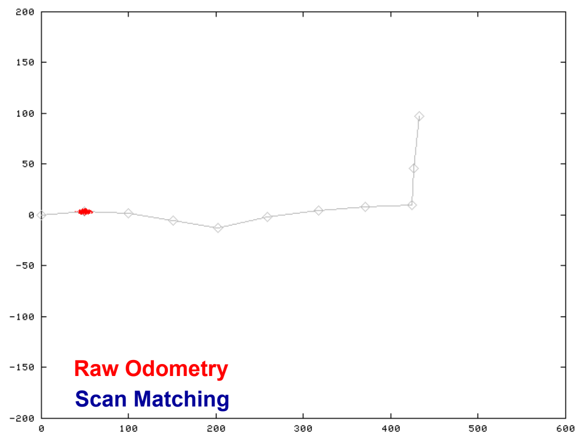
$$x_t^* = \underset{x_t}{\operatorname{argmax}} \left\{ p(z_t \mid x_t, m_{t-1}) \; p(x_t \mid u_{t-1}, x_{t-1}^*) \right\}$$

current measurement

robot motion

map constructed so far

Motion Model for Scan Matching



Courtesy: Dirk Hähnel

Mapping using Scan Matching

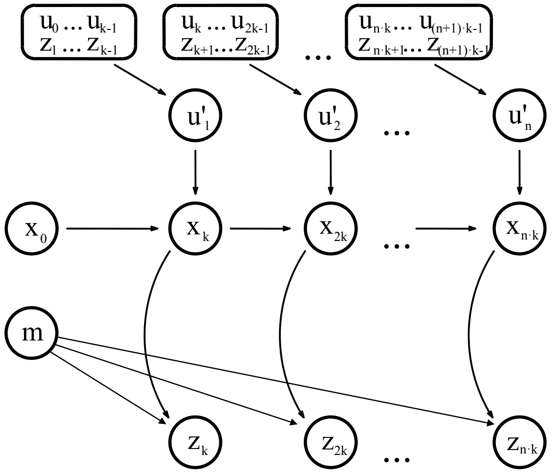
Courtesy: Dirk Hähnel

FastSLAM with Improved Odometry

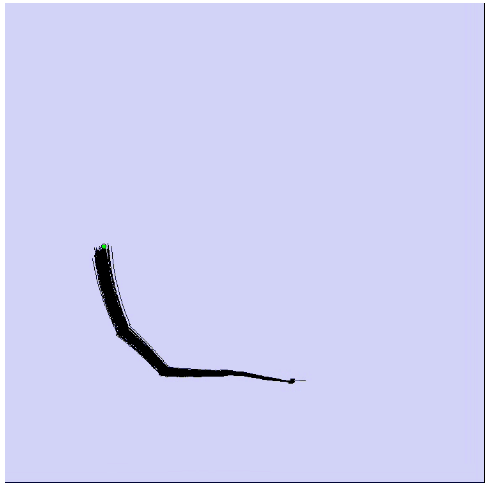
- Scan-matching provides a **locally consistent** pose correction
- Pre-correct short odometry sequences using scan-matching and use them as input to FastSLAM
- Fewer particles are needed, since the error in the input in smaller

[Hähnel et al., 2003]

Graphical Model for Mapping with Improved Odometry

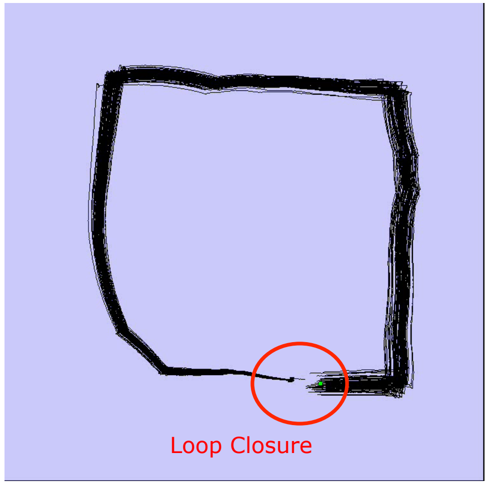


Grid-Based FastSLAM with Scan-Matching



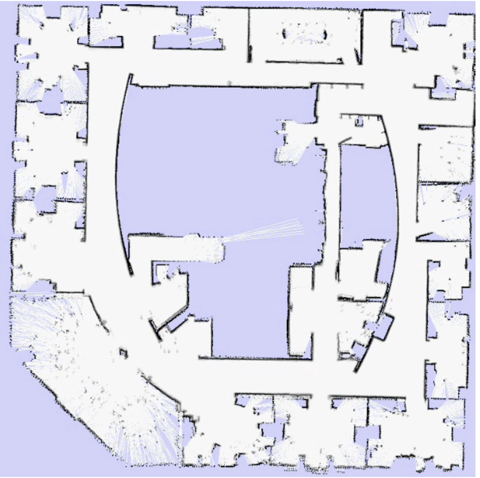
Courtesy:
Dirk Hähnel

Grid-Based FastSLAM with Scan-Matching



Courtesy:
Dirk Hähnel

Grid-Based FastSLAM with Scan-Matching



Courtesy:
Dirk Hähnel

Summary so far …

- Efficient algorithm for SLAM combining ideas of scan matching and FastSLAM
- Scan matching to generate virtual 'high quality' motion commands
- Can be seen as an ad-hoc solution to an improved proposal distribution

What's Next?

- Compute an improved proposal that considers the most recent observation

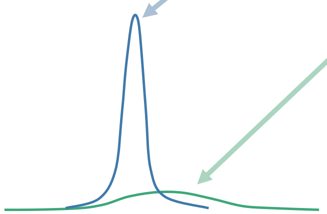
$$x_t^{[k]} \sim p(x_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t})$$

Goals:

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

The Optimal Proposal Distribution [Arulampalam et al., 01]

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



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

Proposal Distribution

$$p(x_t \mid x_{t-1}^{[i]}, m^{[i]}, z_t, u_t) = \frac{\overbrace{p(z_t \mid x_t, m^{[i]})\, p(x_t \mid x_{t-1}^{[i]}, u_t)}^{\tau(x_t)}}{p(z_t \mid x_{t-1}^{[i]}, m^{[i]}, u_t)}$$

Proposal Distribution

$$p(x_t \mid x_{t-1}^{[i]}, m^{[i]}, z_t, u_t) = \frac{\overbrace{p(z_t \mid x_t, m^{[i]})\, p(x_t \mid x_{t-1}^{[i]}, u_t)}^{\tau(x_t)}}{p(z_t \mid x_{t-1}^{[i]}, m^{[i]}, u_t)}$$

$$p(z_t \mid x_{t-1}^{[i]}, m^{[i]}, u_t) = \int_{x_t} p(z_t \mid x_t, m^{[i]})\, p(x_t \mid x_{t-1}^{[i]}, u_t)\, dx_t$$

Proposal Distribution

$$p(x_t \mid x_{t-1}^{[i]}, m^{[i]}, z_t, u_t) = \frac{\overbrace{p(z_t \mid x_t, m^{[i]})\, p(x_t \mid x_{t-1}^{[i]}, u_t)}^{\tau(x_t)}}{p(z_t \mid x_{t-1}^{[i]}, m^{[i]}, u_t)}$$

$$p(z_t \mid x_{t-1}^{[i]}, m^{[i]}, u_t) = \int_{x_t} p(z_t \mid x_t, m^{[i]})\, p(x_t \mid x_{t-1}^{[i]}, u_t)\, dx_t$$

$$p(x_t \mid x_{t-1}^{[i]}, m^{[i]}, z_t, u_t) = \frac{\tau(x_t)}{\int_{x_t} \tau(x_t)\, dx_t}$$

Proposal Distribution

$$p(x_t \mid x_{t-1}^{[i]}, m^{[i]}, z_t, u_t)$$

$$= \frac{\overbrace{p(z_t \mid x_t, m^{[i]})\, p(x_t \mid x_{t-1}^{[i]}, u_t)}^{\tau(x_t)}}{\int_{x_t} \underline{p(z_t \mid x_t, m^{[i]})}\, \underline{p(x_t \mid x_{t-1}^{[i]}, u_t)}\, dx_t}$$

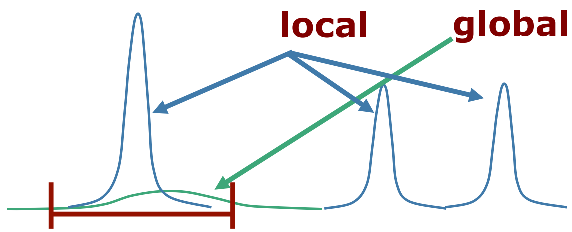
locally limits the area over which to integrate (measurement)

globally limits the area over which to integrate (odometry)

Proposal Distribution

$$p(x_t \mid x_{t-1}^{[i]}, m^{[i]}, z_t, u_t)$$

$$= \frac{\overbrace{p(z_t \mid x_t, m^{[i]})\, p(x_t \mid x_{t-1}^{[i]}, u_t)}^{\tau(x_t)}}{\int_{x_t} \underline{p(z_t \mid x_t, m^{[i]})}\, \underline{p(x_t \mid x_{t-1}^{[i]}, u_t)}\, dx_t}$$



Proposal Distribution

$$p(x_t \mid x_{t-1}^{[i]}, m^{[i]}, z_t, u_t) \quad \simeq \quad \frac{\tau(x_t)}{\int_{\{x_t \mid \tau(x_t) > \epsilon\}} \tau(x_t) \; dx_t}$$

with $\tau(x_t) \quad := \quad p(z_t \mid x_t, m^{[i]}) \; p(x_t \mid x_{t-1}^{[i]}, u_t)$

How to sample from this term?

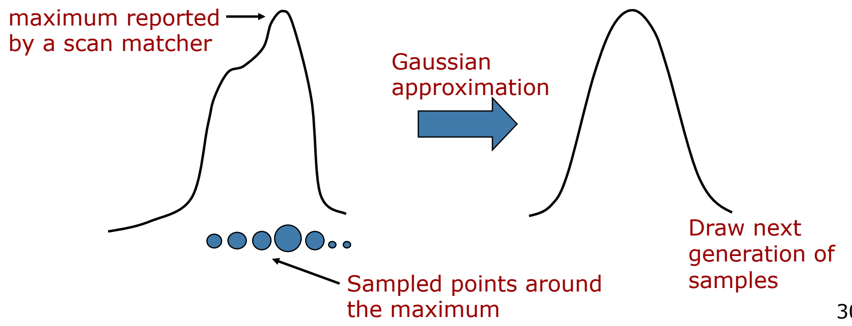
Gaussian approximation:

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

Gaussian Proposal Distribution

$$p(x_t \mid x_{t-1}^{[i]}, m^{[i]}, z_t, u_t) \quad \simeq \quad \frac{\tau(x_t)}{\int_{\{x_t \mid \tau(x_t) > \epsilon\}} \tau(x_t) \; dx_t}$$

Approximate this equation by a Gaussian:



Estimating the Parameters of the Gaussian for Each Particle

$$\mu^{[i]} \quad = \quad \frac{1}{\eta} \sum_{j=1}^{K} x_j \; \tau(x_j)$$

$$\Sigma^{[i]} \quad = \quad \frac{1}{\eta} \sum_{j=1}^{K} (x_j - \mu^{[i]})(x_j - \mu^{[i]})^T \; \tau(x_j)$$

x_j are a set of points sampled around the point x^* the scan matching has converged to

Computing the Importance Weight

$$w_t^{[i]} \quad = \quad w_{t-1}^{[i]} \; p(z_t \mid x_{t-1}^{[i]}, m^{[i]}, u_t)$$

[Arulampalam et al., 01]

Computing the Importance Weight

$$
\begin{aligned}
w_t^{[i]} &= w_{t-1}^{[i]}\, p(z_t \mid x_{t-1}^{[i]}, m^{[i]}, u_t) \\
&= w_{t-1}^{[i]} \int_{x_t} p(z_t \mid x_t, m^{[i]})\, p(x_t \mid x_{t-1}^{[i]}, u_t)\ dx_t
\end{aligned}
$$

Computing the Importance Weight

$$
\begin{aligned}
w_t^{[i]} &= w_{t-1}^{[i]}\, p(z_t \mid x_{t-1}^{[i]}, m^{[i]}, u_t) \\
&= w_{t-1}^{[i]} \int_{x_t} \underbrace{p(z_t \mid x_t, m^{[i]})\, p(x_t \mid x_{t-1}^{[i]}, u_t)}_{\tau(x_t)}\ dx_t
\end{aligned}
$$

Computing the Importance Weight

$$
\begin{aligned}
w_t^{[i]} &= w_{t-1}^{[i]}\, p(z_t \mid x_{t-1}^{[i]}, m^{[i]}, u_t) \\
&= w_{t-1}^{[i]} \int_{x_t} p(z_t \mid x_t, m^{[i]})\, p(x_t \mid x_{t-1}^{[i]}, u_t)\ dx_t \\
&\simeq w_{t-1}^{[i]} \int_{\{x_t \mid \tau(x_t) > \epsilon\}} \tau(x_t)\ dx_t
\end{aligned}
$$

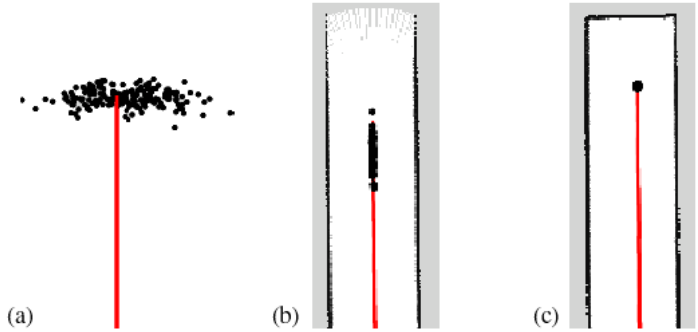
Computing the Importance Weight

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

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

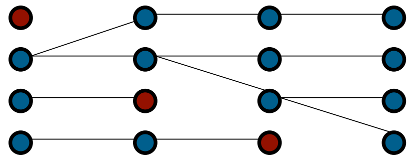
Improved Proposal

- The proposal adapts to the structure of the environment



Resampling

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



- Goal: Reduce the resampling actions

Selective Re-sampling

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

- **Key question: When to re-sample?**

Number of Effective Particles

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

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

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

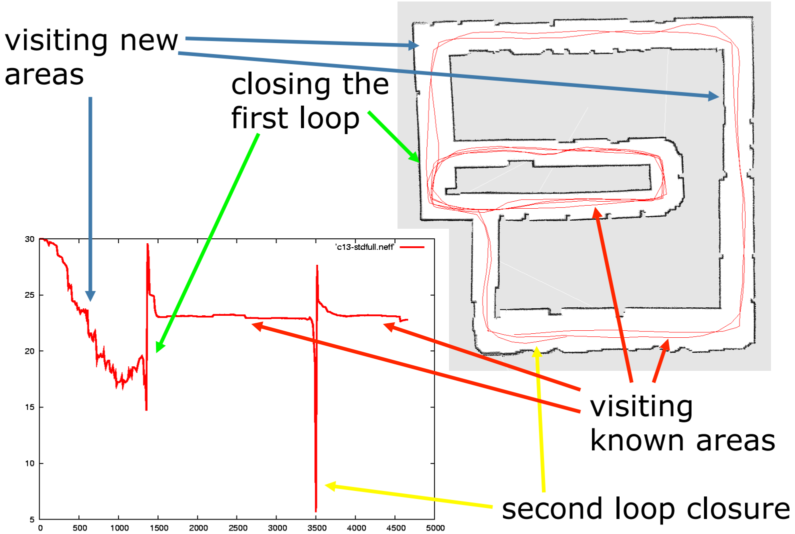
Resampling with n_{eff}

- If our approximation is close to the target, no resampling is needed
- We only re-sample when n_{eff} drops below a given threshold ($N/2$)

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

[Doucet, '98; Arulampalam, '01]

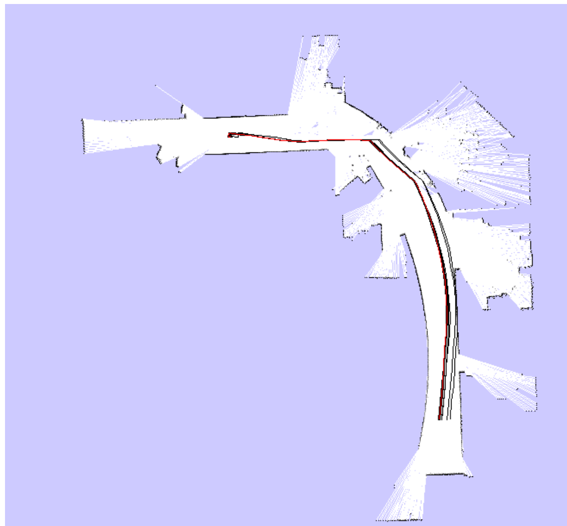
Typical Evolution of n_{eff}



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



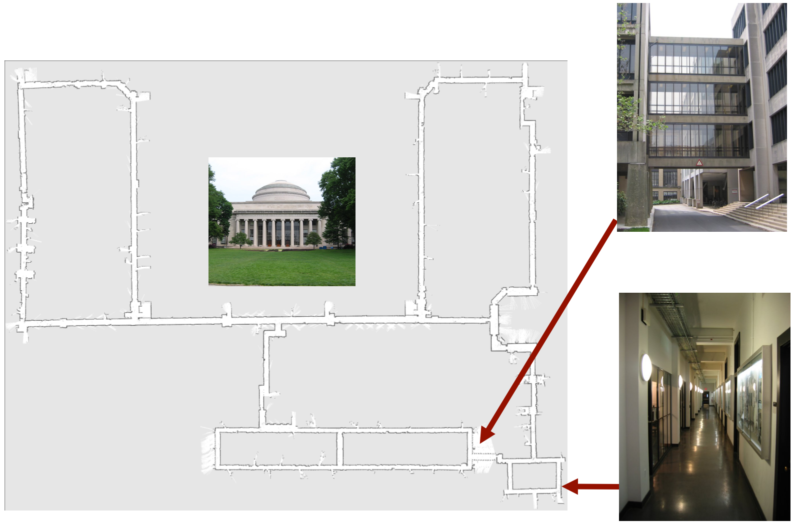
Outdoor Campus Map

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

MIT Killian Court

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

MIT Killian Court



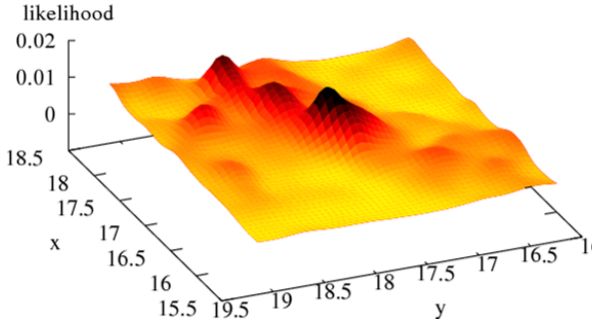
MIT Killian Court – Video

Real World Application

- This guy uses a similar technique…

Problems of Gaussian Proposals

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



Gaussian or Non-Gaussian?

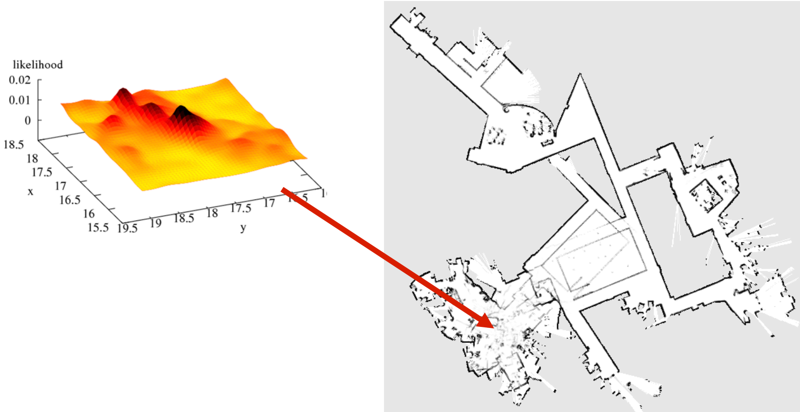
- Statistical test to check whether or not sample a generated from a Gaussian
- Anderson-Darling test (based on the cumulative density function)
- Difference between the Gaussian and the optimal proposal via KLD

Is a Gaussian an Accurate Choice 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%

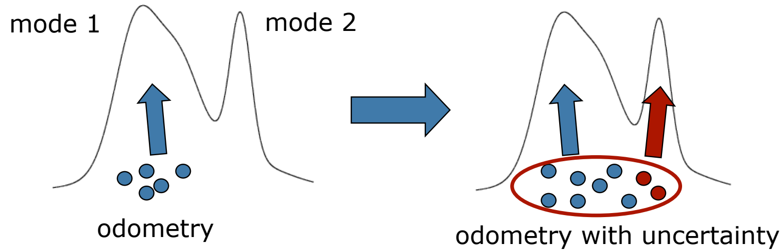
Problems of Gaussian Proposals

- Multi-modal likelihood function can cause filter divergence



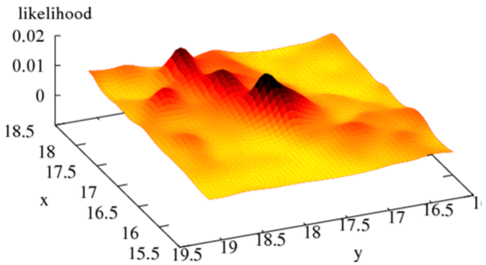
Efficient Multi-Modal Sampling

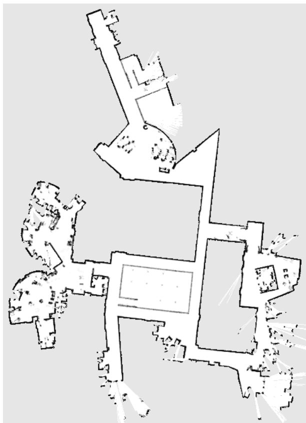
- Approximate the likelihood in a better way!



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

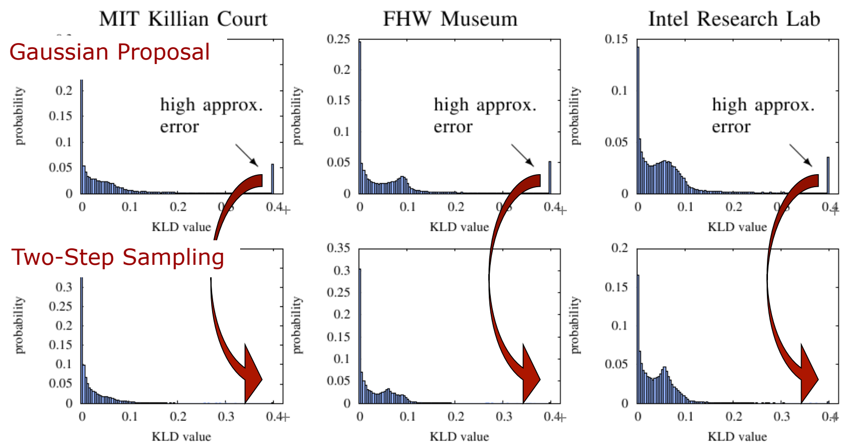
The Two-Step Sampling Works!





...with nearly zero overhead

Proposal Error Evaluation



Effect of Two-Step Sampling

- Allows for better modeling multi-modal likelihood functions (high KLD values do not occur)
- For uni-modal cases, identical results
- Minimal computational overhead

Gaussian Proposal: Yes or No?

- Gaussian allow for efficient sampling
- Problematic in multi-model cases
- Laser-baser SLAM: 3-6% multi-modal distribution (for the datasets here)
- Gaussian proposals can lead to divergence
- Two-step sampling process overcomes this problem effectively and efficiently

Conclusion

- 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 resamples reduces the risk of particle depletion
- Substantial reduction of the required number of particles

Literature

Grid-FastSLAM with Improved Proposals

- Grisetti, Stachniss, Burgard: Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters, 2007
- Stachniss, Giorgio, Burgard, Roy. Analyzing Gaussian Proposal Distributions for Mapping with Rao-Blackwellized Particle Filters, 2007

Grid-FastSLAM & Scan-Matching

- Hähnel, Burgard, Fox, Thrun. An efficient FastSLAM Algorithm for Generating Maps of Large-Scale Cyclic Environments from Raw Laser Range Measurements, 2003

GMapping

- Efficient open source implementation of the presented method (2005-2008)
- C++ Code available via
  svn co https://svn.openslam.org/data/svn/gmapping