Robot Mapping

Grid-Based FastSLAM

Cyrill Stachniss
Motivation

- So far, we addressed landmark-based SLAM (KF-based SLAM, 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

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

First introduced for SLAM by Murphy in 1999
Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

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

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

map of particle 1

map of particle 2

map of particle 3

3 particles
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

\[ x_t^* = \arg\max_{x_t} \left\{ p(z_t | x_t, m_{t-1}) \ p(x_t | u_t, x_{t-1}^*) \right\} \]
Motion Model for Scan Matching

Raw Odometry
Scan Matching

Courtesy: Dirk Hähnel
Mapping using Scan Matching

Courtesy: Dirk Hähnel
Grid-Based 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

[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

![Diagram of Loop Closure](image)

Courtesy: Dirk Hähnel
Grid-Based FastSLAM with Scan-Matching

Courtesy: Dirk Hähnel
Summary so far ...

- Approach to SLAM that combines 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]

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

For lasers \( p(z_t | x_t, m^i) \) is typically peaked and dominates the product
Proposal Distribution

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

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

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

\[
p(z_t | x_{t-1}^{[i]}, m^{[i]}, u_t) = \int p(z_t | x_t, m^{[i]}) p(x_t | x_{t-1}^{[i]}, u_t) \, dx_t
\]
Proposal Distribution

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

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

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

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

**locally** limits the area over which to integrate (measurement)

**globally** limits the area over which to integrate (odometry)
Proposal Distribution

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

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

\[ p(x_t \mid x_{t-1}^{[i]}, m^{[i]}, z_t, u_t) \approx \frac{\tau(x_t)}{\int\{x_t \mid \tau(x_t) > \epsilon\} \tau(x_t) \, dx_t} \]

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

How to sample from this term?

Gaussian approximation:

\[ \tau(x_t) \sim \mathcal{N}(\mu^{[i]}, \Sigma^{[i]}) \]
Gaussian Proposal Distribution

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

Approximate by a Gaussian:

maximum reported by a scan matcher

\[ \tau(x_t) \]

Sampled points around the maximum

Gaussian approximation

Draw next generation of samples
Estimating the Parameters of the Gaussian for Each Particle

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

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

\(x_j\) are the points sampled around the result of the scan matcher
Gaussian Proposal Distribution

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

\[ \tau(x_t) \sim \mathcal{N}(\mu^{[i]}, \Sigma^{[i]}) \]

\[
\begin{align*}
\mu^{[i]} &= \frac{1}{\eta} \sum_{j=1}^{K} x_j \tau(x_j) \\
\Sigma^{[i]} &= \frac{1}{\eta} \sum_{j=1}^{K} (x_j - \mu^{[i]}) (x_j - \mu^{[i]})^T \tau(x_j)
\end{align*}
\]

\[
\int_{\{x_t \mid \tau(x_t) > \epsilon\}} \tau(x_t) \, dx_t \approx \sum_{j=1}^{K} \tau(x_j)
\]
The Importance Weight

\[ w_t^{[i]} = \frac{\text{target}(x_t^{[i]})}{\text{proposal}(x_t^{[i]})} \]
The Importance Weight

\[ w_t^{[i]} = \frac{\text{target}(x_t^{[i]})}{\text{proposal}(x_t^{[i]})} \]

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

\[ \frac{p(x_{1:t-1}^{[i]} \mid z_{1:t-1}, u_{1:t-1})}{\pi(x_{1:t-1}^{[i]} \mid z_{1:t-1}, u_{1:t-1})} \]
The Importance Weight

\[ w_t^{[i]} = \frac{\text{target}(x_t^{[i]})}{\text{proposal}(x_t^{[i]})} \]

\[ \propto \frac{p(z_t | m^{[i]}, x_t^{[i]}) p(x_t^{[i]} | x_{t-1}^{[i]}, u_t)}{\pi(x_t^{[i]} | m^{[i]}, x_{t-1}^{[i]}, z_t, u_t)} \]

\[ \frac{p(x_{1:t-1}^{[i]} | z_{1:t-1}, u_{1:t-1})}{\pi(x_{1:t-1}^{[i]} | z_{1:t-1}, u_{1:t-1})} \]

\[ = \left( \frac{p(z_t | m^{[i]}, x_t^{[i]}) p(x_t^{[i]} | x_{t-1}^{[i]}, u_t)}{\int p(z_t | m^{[i]}, x_t) p(x_t | x_{t-1}^{[i]}, u_t) dx_t} \right) w_{t-1}^{[i]} \]
The Importance Weight

\[
w_t^{[i]} = \frac{\text{target}(x_t^{[i]})}{\text{proposal}(x_t^{[i]})} \propto \frac{p(z_t \mid m^{[i]}, x_t^{[i]}) p(x_t^{[i]} \mid x_{t-1}^{[i]}, u_t)}{\pi(x_t^{[i]} \mid m^{[i]}, x_{t-1}^{[i]}, z_t, u_t)} \\
\quad \times \frac{p(x_{1:t-1}^{[i]} \mid z_{1:t-1}, u_{1:t-1})}{\pi(x_{1:t-1}^{[i]} \mid z_{1:t-1}, u_{1:t-1})} \\
= \frac{p(z_t \mid m^{[i]}, x_t^{[i]}) p(x_t^{[i]} \mid x_{t-1}^{[i]}, u_t)}{p(z_t \mid m^{[i]}, x_t^{[i]}) p(x_t^{[i]} \mid x_{t-1}^{[i]}, u_t) \int p(z_t \mid m^{[i]}, x_t) p(x_t \mid x_{t-1}^{[i]}, u_t) \, dx_t} w_{t-1}^{[i]} \\
= w_{t-1}^{[i]} \int p(z_t \mid m^{[i]}, x_t) p(x_t \mid x_{t-1}^{[i]}, u_t) \, dx_t
\]
The Importance Weight

\[ w_t^{[i]} = w_{t-1}^{[i]} \int p(z_t | x_t, m^{[i]}) p(x_t | x_{t-1}^{[i]}, u_t) \, dx_t \]
The Importance Weight

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

\[ \approx w_{t-1}^{[i]} \int_{\{x_t \mid \tau(x_t) > \epsilon\}} \tau(x_t) \, dx_t \]
The Importance Weight

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

\[ \approx w_{t-1}^{[i]} \int_{\{x_t \mid \tau(x_t) > \epsilon\}} \tau(x_t) \, dx_t \]

\[ \approx w_{t-1}^{[i]} \sum_{j=1}^{K} \tau(x_j) \]

Already computed for the proposal! 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:

  ![Diagram showing particle loss](image)

- 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

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

- \( n_{\text{eff}} \) describes “the inverse variance of the normalized particle weights”

- For equal weights, the sample approximation is close to the target
Resampling with $n_{\text{eff}}$

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

\[
\frac{1}{\sum_i \left( w_t^{[i]} \right)^2} \leq \frac{N}{2}
\]

- Note: weights need to be normalized

[Doucet, ’98; Arulampalam, ’01]
Typical Evolution of $n_{\text{eff}}$

- 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
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?

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Gauss</th>
<th>Non-Gauss; 1 mode</th>
<th>Multi-modal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Research Lab</td>
<td>89.2%</td>
<td>7.2%</td>
<td>3.6%</td>
</tr>
<tr>
<td>FHW Museum</td>
<td>84.5%</td>
<td>10.4%</td>
<td>5.1%</td>
</tr>
<tr>
<td>Belgioioso</td>
<td>84.0%</td>
<td>10.4%</td>
<td>5.6%</td>
</tr>
<tr>
<td>MIT CSAIL</td>
<td>78.1%</td>
<td>15.9%</td>
<td>6.0%</td>
</tr>
<tr>
<td>MIT Killian Court</td>
<td>75.1%</td>
<td>19.1%</td>
<td>5.8%</td>
</tr>
<tr>
<td>Freiburg Bldg. 79</td>
<td>74.0%</td>
<td>19.4%</td>
<td>6.6%</td>
</tr>
</tbody>
</table>
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 use this as the start point for scan matching

![Diagram showing two modes of odometry with and without uncertainty]
The Two-Step Sampling Works!

...with nearly zero overhead
Proposal Error Evaluation

Gaussian Proposal

MIT Killian Court

Two-Step Sampling

FHW Museum

Intel Research Lab
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-Based 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


Grid-FastSLAM & Scan-Matching

GMapping

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