Fast-SLAM Mapping

Particle Representation

- A set of weighted samples
  \[ \mathcal{X} = \{ (x[i], w[i]) \}_{i=1,...,N} \]

- Think of a sample as one hypothesis about the state
- For feature-based SLAM:
  \[ x = (x_{1:t}, m_{1,x}, m_{1,y}, \ldots, m_{M,x}, m_{M,y})^T \]

Dimensionality Problem

Particle filters are effective in low dimensional spaces as the likely regions of the state space need to be covered with samples.

\[ x = (x_{1:t}, m_{1,x}, m_{1,y}, \ldots, m_{M,x}, m_{M,y})^T \]

Can We Exploit Dependencies Between the Different Dimensions of the State Space?

\[ x_{1:t}, m_1, \ldots, m_M \]
If We Know the Poses of the Robot, Mapping is Easy!

$\mathbf{x}_{1:t}, \mathbf{m}_1, \ldots, \mathbf{m}_M$

Key Idea

$\mathbf{x}_{1:t}, \mathbf{m}_1, \ldots, \mathbf{m}_M$

If we use the particle set only to model the robot’s path, each sample is a path hypothesis. For each sample, we can compute an individual map of landmarks.

Rao-Blackwellization

- Factorization to exploit dependencies between variables:
  
  $p(a, b) = p(b \mid a) \, p(a)$

- If $p(b \mid a)$ can be computed efficiently, represent only $p(a)$ with samples and compute $p(b \mid a)$ for every sample

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

$$p(\mathbf{x}_{0:t}, \mathbf{m}_{1:M} \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) =$$

First introduced for SLAM by Murphy in 1999


Courtesy: C. Stachniss
Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

\[ p(x_0:t, m_1:M | z_{1:t}, u_{1:t}) = \]
\[ p(x_0:t | z_{1:t}, u_{1:t}) p(m_1:M | x_0:t, z_{1:t}) \]

First introduced for SLAM by Murphy in 1999


FastSLAM

- Proposed by Montemerlo et al. in 2002
- Each landmark is represented by a 2x2 EKF
- Each particle therefore has to maintain M individual EKFs

<table>
<thead>
<tr>
<th>Particle</th>
<th>x, y, θ</th>
<th>Landmark 1</th>
<th>Landmark 2</th>
<th>…</th>
<th>Landmark M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Particle 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Particle 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Particle N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FastSLAM – Motion Update

<table>
<thead>
<tr>
<th>Particle #1</th>
<th>Landmark 1 2x2 EKF</th>
<th>Landmark 2 2x2 EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Particle #2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Particle #3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FastSLAM – Sensor Update

<table>
<thead>
<tr>
<th>Particle #1</th>
<th>Landmark 1 2x2 EKF</th>
<th>Landmark 2 2x2 EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Particle #2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Particle #3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Courtesy: C. Stachniss

First introduced for SLAM by Murphy in 1999


First introduced for SLAM by Murphy in 1999

Key Steps of FastSLAM 1.0

- Extend the path posterior by sampling a new pose for each sample
  \[ x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t) \]

- Compute particle weight
  \[ w_t^{[k]} = \frac{1}{(2\pi)^{n/2} |Q|^{-1/2}} \exp \left\{ -\frac{1}{2} (z_t - \hat{z}_t^{[k]})^T Q^{-1} (z_t - \hat{z}_t^{[k]}) \right\} \]

- Update belief of observed landmarks
  (EKF update rule)

- Resample
Data Association Problem

- Which observation belongs to which landmark?
- More than one possible association
- Potential data associations depend on the pose of the robot

Particles Support for Multi-Hypotheses Data Association

- Decisions on a per-particle basis
- Robot pose error is factored out of data association decisions
Per-Particle Data Association

Was the observation generated by the red or by the blue landmark?

\[ P(\text{observation} \mid \text{red}) = 0.3 \quad P(\text{observation} \mid \text{blue}) = 0.7 \]

Two options for per-particle data association
- Pick the most probable match
- Pick a random association weighted by the observation likelihoods
- If the probability for an assignment is too low, generate a new landmark

Results – Victoria Park
- 4 km traverse
- < 2.5 m RMS position error
- 100 particles

Blue = GPS
Yellow = FastSLAM

Results – Victoria Park (Video)
Techniques to Reduce the Number of Particles Needed

- Better proposals (put the particles in the right place in the prediction step).
- Avoid particle depletion (re-sample only when needed).

Generating better Proposals

- Use scan-matching to compute highly accurate odometry measurements from consecutive range scans.
- Use the improved odometry in the prediction step to get highly accurate proposal distributions.
Motion Model for Scan Matching

Raw Odometry
Scan Matching

Initial pose

Final pose

Measured pose

Path

Loop Closure Example

Map: Intel Research Lab Seattle

Rao-Blackwellized Mapping with Scan-Matching

Map: Intel Research Lab Seattle
Rao-Blackwellized Mapping with Scan-Matching

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

FastSLAM Summary
- Particle filter-based SLAM
- Rao-Blackwellization: model the robot's path by sampling and compute the landmarks given the poses
- Allow for per-particle data association
- Complexity $O(N \log M)$
Literature

FastSLAM