CSE-571
Robotics

Fast-SLAM Mapping
Particle Representation

- A set of weighted samples
  \[ x = \{ \langle x[i], w[i] \rangle \}_{i=1,...,N} \]

- Think of a sample as one hypothesis about the state

- For feature-based SLAM:
  \[ x = \begin{pmatrix} x_{1:t}, m_1, x, m_1, y, \ldots, m_M, x, m_M, y \end{pmatrix}^T \]
  poses landmarks

Courtesy: C. Stachniss
Dimensionality Problem

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

\[ x = (x_1:t, m_1, x, m_1, y, \ldots, m_M, x, m_M, y)^T \]

Courtesy: C. Stachniss
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!

$x_1: t, m_1, \ldots, m_M$
Key Idea

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

Courtesy: C. Stachniss
Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

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

First introduced for SLAM by Murphy in 1999


Courtesy: C. Stachniss
Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

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

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>$x, y, \theta$</th>
<th>Landmark 1</th>
<th>Landmark 2</th>
<th>…</th>
<th>Landmark $M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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>$N$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FastSLAM – Motion Update

Particle #1

Particle #2

Particle #3

Landmark 1
2x2 EKF

Landmark 2
2x2 EKF

Courtesy: M. Montemerlo
FastSLAM – Sensor Update

Particle #1

Particle #2

Particle #3

Landmark 1
2x2 EKF

Landmark 2
2x2 EKF

Courtesy: M. Montemerlo
FastSLAM – Sensor Update

Particle #1
Weight = 0.8

Particle #2
Weight = 0.4

Particle #3
Weight = 0.1

Courtesy: M. Montemerlo
FastSLAM – Sensor Update

Particle #1

Update map of particle 1

Particle #2

Update map of particle 2

Particle #3

Update map of particle 3

Courtesy: M. Montemerlo
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
  \[ w^{[k]} = \frac{1}{2\pi Q} \exp \left\{ -\frac{1}{2} (z_t - \hat{z}^{[k]})^T Q^{-1} (z_t - \hat{z}^{[k]}) \right\} \]

- Update belief of observed landmarks (EKF update rule)

- Resample

Courtesy: C. Stachniss
FastSLAM in Action
FastSLAM – Video – All Maps
FastSLAM – Video – “Best” particle in terms of Cum Log Prob
Data Association Problem

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

Courtesy: M. Montemerlo
Particles Support for Multi-Hypotheses Data Association

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

Courtesy: M. Montemerlo
Per-Particle Data Association

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

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

Courtesy: M. Montemerlo
Results – Victoria Park

- 4 km traverse
- < 2.5 m RMS position error
- 100 particles

Blue = GPS
Yellow = FastSLAM

Courtesy: M. Montemerlo
Results – Victoria Park (Video)

Courtesy: M. Montemerlo
Results (Sample Size)

Accuracy of FastSLAM vs. the EKF on Simulated Data

RMS Pose Error (meters)

Number of Particles

Courtesy: M. Montemerlo
Results (Motion Uncertainty)

Comparison of FastSLAM and EKF Given Motion Ambiguity

Robot RMS Position Error (m)

Error Added to Rotational Velocity (std.)

Courtesy: M. Montemerlo
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
Rao-Blackwellized Mapping with Scan-Matching
Loop Closure Example

map of particle 1

3 particles

map of particle 2

map of particle 3
Rao-Blackwellized Mapping with Scan-Matching
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

Work by Grisetti et al.
Outdoor Campus Map

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

Work by Grisetti et al.
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

- Thrun et al.: “Probabilistic Robotics”, Chapter 13.1-13.3 + 13.8 (see errata!)
6D OBJECT POSE ESTIMATION

6D Object Pose

3D Translation

3D Orientation

3D Translation 3D Orientation

x y z

x

y

z
POSE-CNN

Handles symmetric, texture-less objects under partial occlusions

- Provides object mask and 3D position and orientation of object relative to camera
- Operates at 10Hz, sufficient to initialize a tracker
- With ICP, state of the art results on LineMod and YCB-Video

Centers and masks
6D poses

[Xiang-Schmidt-Narayanan-Fox: RSS-18]
RELATED WORK

Single image

- Object symmetries ignored or special cases
  - Tremblay et al. CoRL 2018 (DOPE)
  - Tekin et al. CVPR 2018
  - Xiang et al. RSS 2018 (PoseCNN)
  - Li et al. ECCV 2018 (DeepIM)
  - Manhardt et al. ECCV 2018

Techniques aim at a unique pose estimate

6D object pose tracking

- Unimodal tracking
  - Kehl et al. CVPR 2018
  - Tjaden et al. ICCV 2017
  - Prisacariu et al. IJCV 2017 (PWP3D)
  - Srivatsan et al. RSS 2017

- 6D particle filter
  - Choi et al. IROS 2013

Not designed to estimate multi-modal distributions
ORIENTATION UNCERTAINTY

Depends on context, shape, sensor

- **Shape symmetry**
- **Texture breaks symmetry**
- **View-based uncertainty**
TRAINING VIEW-BASED SIMILARITY
Randomly Sample Views onto the Textured Object Model

Bottleneck encodes viewpoint information

[Sundermeyer-Marton-Durner-Brucker-Trieben: ECCV-18]
TRAINING AN ENCODER IN SIMULATION

Randomly Sample Views onto the Textured Object Model

Encoder

Cosine Similarity

128 dim

Codebook (5 deg discretization; 191,808 views)
PoseRBPF: 6D PARTICLE FILTER

\[ X_i = \{T_i, P(R_i|T_i, Z_{1:k})\} \]

- 3D Translation \( T_i \)
- Orientation Distribution \( P(R_i|T_i, Z_{1:k}) \)

YCB-Video RGB-(D)
- PoseRBPF:
  ADD: 62.1, ADD-S: 78.4
- PoseCNN:
  ADD: 53.7, ADD-S: 75.9

191,808 bins
5 deg resolution
PoseRBPF: Observation Update

Compute posterior

\[ P(R_k, T_k^i, Z_{1:k}) \propto P(R_k | T_k, Z_k) P(R_k | R_{k-1}) \]

Weights:

Encoder

Observation Likelihood

Normalizer
EXAMPLE RESULTS

PoseRBPF(RGB)  PoseRBPF(RGBD)  PoseCNN[1]
GLOBAL LOCALIZATION EXAMPLE

Sample Uniformly in Translation Space

1st frame: 5,000 particles, then 500 particles until strong match, then 50 particles
500 particles: 2.6 fps; 50 particles: 20 fps