Probabilistic Robotics

FastSLAM

Sebastian Thrun

(abridged and adapted by Rodrigo Ventura in Oct-2008)
The SLAM Problem

- SLAM stands for simultaneous localization and mapping
- The task of building a map while estimating the pose of the robot relative to this map

- 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
The SLAM Problem

A robot moving through an unknown, static environment

**Given:**

- The robot’s controls
- Observations of nearby features

**Estimate:**

- Map of features
- Path of the robot
Why is SLAM a hard problem?

**SLAM**: robot path and map are both **unknown**!

Robot path error correlates errors in the map
Why is SLAM a hard problem?

- In the real world, the mapping between observations and landmarks is unknown
- Picking wrong data associations can have catastrophic consequences
- Pose error correlates data associations
Data Association Problem

- A data association is an assignment of observations to landmarks
- In general there are more than \( \binom{n}{m} \) \((n \text{ observations, } m \text{ landmarks})\) possible associations
- Also called “assignment problem”
Particle Filters

- Represent belief by random samples
- Estimation of non-Gaussian, nonlinear processes

- Sampling Importance Resampling (SIR) principle
  - Draw the new generation of particles
  - Assign an importance weight to each particle
  - Resampling

- Typical application scenarios are tracking, localization, ...
Localization vs. SLAM

- A particle filter can be used to solve both problems

- Localization: state space \( \langle x, y, \theta \rangle \)

- SLAM: state space \( \langle x, y, \theta, map \rangle \)
  - for landmark maps = \( \langle l_1, l_2, \ldots, l_m \rangle \)
  - for grid maps = \( \langle c_{11}, c_{12}, \ldots, c_{1n}, c_{21}, \ldots, c_{nm} \rangle \)

- **Problem**: The number of particles needed to represent a posterior grows exponentially with the dimension of the state space!
Dependencies

- Is there a dependency between the dimensions of the state space?
- If so, can we use the dependency to solve the problem more efficiently?
Dependencies

- Is there a dependency between the dimensions of the state space?
- If so, can we use the dependency to solve the problem more efficiently?

In the SLAM context
  - The map depends on the poses of the robot.
  - We know how to build a map given the position of the sensor is known.
Factored Posterior (Landmarks)

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

Factorization first introduced by Murphy in 1999
Factored Posterior (Landmarks)

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

does this help to solve the problem?

Factorization first introduced by Murphy in 1999
Knowledge of the robot’s true path renders landmark positions conditionally independent
Factored Posterior

\[ p(x_{1:t}, l_{1:m} \mid z_{1:t}, u_{0:t-1}) = p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot p(l_{1:m} \mid x_{1:t}, z_{1:t}) = p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot \prod_{i=1}^{M} p(l_i \mid x_{1:t}, z_{1:t}) \]
Rao-Blackwellization

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

\[ p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot \prod_{i=1}^{M} p(l_i \mid x_{1:t}, z_{1:t}) \]

- This factorization is also called Rao-Blackwellization
- Given that the second term can be computed efficiently, particle filtering becomes possible!
FastSLAM

- Rao-Blackwellized particle filtering based on landmarks [Montemerlo et al., 2002]
- Each landmark is represented by a 2x2 Extended Kalman Filter (EKF)
- Each particle therefore has to maintain $M$ EKFs
FastSLAM – Action Update

Particle #1

Particle #2

Particle #3

Landmark #1

Filter

Landmark #2

Filter
FastSLAM – Sensor Update

Particle #1

Particle #2

Particle #3

Landmark #1 Filter

Landmark #2 Filter
FastSLAM – Sensor Update

Particle #1

Weight = 0.8

Particle #2

Weight = 0.4

Particle #3

Weight = 0.1
FastSLAM Complexity

- Update robot particles based on control \( u_{t-1} \)
  - \( O(N) \) Constant time per particle

- Incorporate observation \( z_t \) into Kalman filters
  - \( O(N \cdot \log(M)) \) Log time per particle

- Resample particle set
  - \( O(N \cdot \log(M)) \) Log time per particle

\[ N = \text{Number of particles} \]
\[ M = \text{Number of map features} \]
Data Association Problem

• Which observation belongs to which landmark?

• A robust SLAM must consider possible data associations
• Potential data associations depend also on the pose of the robot
Multi-Hypothesis Data Association

- Data association is done 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 the blue landmark?

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

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

- 4 km traverse
- < 5 m RMS position error
- 100 particles

**Blue** = GPS
**Yellow** = FastSLAM

Dataset courtesy of University of Sydney
Results – Data Association

Comparison of FastSLAM and EKF Given Motion Ambiguity

Robot RMS Position Error (m)

Error Added to Rotational Velocity (std.)
Results – Accuracy

Accuracy of FastSLAM vs. the EKF on Simulated Data

RMS Pose Error (meters)

Number of Particles
Grid-based SLAM

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

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

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

poses → map → observations & movements

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 part and apply mapping with known poses
A Graphical Model of Rao-Blackwellized Mapping
Rao-Blackwellized Mapping

- Each particle represents a possible trajectory of the robot

- Each particle
  - maintains its own map and
  - updates it upon “mapping with known poses”

- Each particle survives with a probability proportional to the likelihood of the observations relative to its own map
Particle Filter Example

map of particle 1

map of particle 2

map of particle 3

3 particles
Problem

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

\[ \hat{x}_t = \arg \max_{x_t} \left\{ p(z_t \mid x_t, \hat{m}_{t-1}) \cdot p(x_t \mid u_{t-1}, \hat{x}_{t-1}) \right\} \]

- \( p(z_t \mid x_t, \hat{m}_{t-1}) \): current measurement
- \( p(x_t \mid u_{t-1}, \hat{x}_{t-1}) \): robot motion
- \( \hat{m}_{t-1} \): map constructed so far
Motion Model for Scan Matching

![Graph showing comparison between Raw Odometry and Scan Matching](image)

- **Raw Odometry**
- **Scan Matching**
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

[Haehnel et al., 2003]
Further Improvements

- Improved proposals will lead to more accurate maps
- They can be achieved by adapting the proposal distribution according to the most recent observations
- Flexible re-sampling steps can further improve the accuracy.
Improved Proposal

- The proposal adapts to the structure of the environment
Selective Re-sampling

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

- In case of suboptimal proposal distributions re-sampling is necessary to achieve convergence.

- Key question: When should we re-sample?
Number of Effective Particles

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

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

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

- \( n_{\text{eff}} \) is maximal for equal weights. In this case, the distribution is close to the proposal.
Resampling with Neff

- Only re-sample when $n_{eff}$ drops below a given threshold (n/2)

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

visiting new areas

closing the first loop

visiting known areas

second loop closure
Conclusion

- The ideas of FastSLAM can also be applied in the context of grid maps
- Utilizing accurate sensor observation leads to good proposals and highly efficient filters
- It is similar to scan-matching on a per-particle base
- The number of necessary particles and re-sampling steps can seriously be reduced
- Improved versions of grid-based FastSLAM can handle larger environments than naïve implementations in “real time” since they need one order of magnitude fewer samples
More Details on FastSLAM

- M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to simultaneous localization and mapping, *AAAI02*

- D. Haehnel, W. Burgard, D. Fox, and S. Thrun. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements, *IROS03*


- G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling, *ICRA05*

- A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks, *IJCAI03*