CSE-571
Robotics

Fast-SLAM Mapping

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, …

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.

Particle Filter Algorithm

1. Sample the particles from the proposal distribution

$$x_t^{[j]} \sim \pi(x_t \mid \ldots)$$

2. Compute the importance weights

$$w_t^{[j]} = \frac{target(x_t^{[j]})}{proposal(x_t^{[j]})}$$

1. Resampling: Draw sample i with probability $w_t^{[i]}$ and repeat J times

Courtesy: C. Stachniss

Particle Representation

- A set of weighted samples

$$\mathcal{X} = \left\{ \left\langle x^{[i]}, w^{[i]} \right\rangle \right\}_{i=1,\ldots,N}$$

- Think of a sample as one hypothesis about the state
- For feature-based SLAM:

$$x = (\underbrace{x_{1:t}}_{\text{poses}}, \underbrace{m_{1,x}, m_{1,y}, \ldots, m_{M,x}, m_{M,y}}_{\text{landmarks}})^T$$

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.

$$x = \underbrace{(x_{1:t}, m_{1,x}, m_{1,y}, \ldots, m_{M,x}, m_{M,y})^T}_{\text{high-dimensional}}$$

Courtesy: C. Stachniss

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

$$x_{1:t}, m_1, \ \ldots, m_M$$

Courtesy: C. Stachniss

If We Know the Poses of the Robot, Mapping is Easy!

$$\underline{x_{1:t}}, \underline{m_1, \ \ldots, m_M}$$

Courtesy: C. Stachniss

Key Idea

$$x_{1:t}, m_1, \ldots, m_M$$

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.

Courtesy: C. Stachniss

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

poses map observations & movements

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

First introduced for SLAM by Murphy in 1999

K. Murphy, Bayesian map learning in dynamic environments, In Proc. Advances in Neural Information Processing Systems, 1999

Courtesy: C. Stachniss

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

poses map observations & movements

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

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

path posterior map posterior

First introduced for SLAM by Murphy in 1999

K. Murphy, Bayesian map learning in dynamic environments, In Proc. Advances in Neural Information Processing Systems, 1999

Courtesy: C. Stachniss

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

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

Landmarks are conditionally independent given the poses

First exploited in FastSLAM by Montemerlo et al., 2002

Courtesy: C. Stachniss

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

$$p(x_{0:t}, m_{1:M} \mid z_{1:t}, u_{1:t}) =$$
$$p(x_{0:t} \mid z_{1:t}, u_{1:t})\ p(m_{1:M} \mid x_{0:t}, z_{1:t})$$
$$p(x_{0:t} \mid z_{1:t}, u_{1:t})\ \prod_{i=1}^{M} p(m_i \mid x_{0:t}, z_{1:t})$$

First exploited in FastSLAM by Montemerlo et al., 2002

Courtesy: C. Stachniss

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

$$p(x_{0:t}, m_{1:M} \mid z_{1:t}, u_{1:t}) =$$
$$p(x_{0:t} \mid z_{1:t}, u_{1:t})\ p(m_{1:M} \mid x_{0:t}, z_{1:t})$$
$$p(x_{0:t} \mid z_{1:t}, u_{1:t})\ \prod_{i=1}^{M} \underline{p(m_i \mid x_{0:t}, z_{1:t})}$$

2-dimensional EKFs!

First exploited in FastSLAM by Montemerlo et al., 2002

Courtesy: C. Stachniss

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

$$p(x_{0:t}, m_{1:M} \mid z_{1:t}, u_{1:t}) =$$
$$p(x_{0:t} \mid z_{1:t}, u_{1:t})\ p(m_{1:M} \mid x_{0:t}, z_{1:t})$$
$$\underline{p(x_{0:t} \mid z_{1:t}, u_{1:t})}\ \prod_{i=1}^{M} \underline{p(m_i \mid x_{0:t}, z_{1:t})}$$

particle filter similar to MCL

2-dimensional EKFs!

First exploited in FastSLAM by Montemerlo et al., 2002

Courtesy: C. Stachniss

Modeling the Robot's Path

- Sample-based representation for $p(x_{0:t} \mid z_{1:t}, u_{1:t})$
- Each sample is a path hypothesis

x_0 x_1 x_2 $\ldots$

starting location, typically (0,0,0)

pose hypothesis at time $t=1$

- Past poses of a sample are not revised
- No need to maintain past poses in the sample set

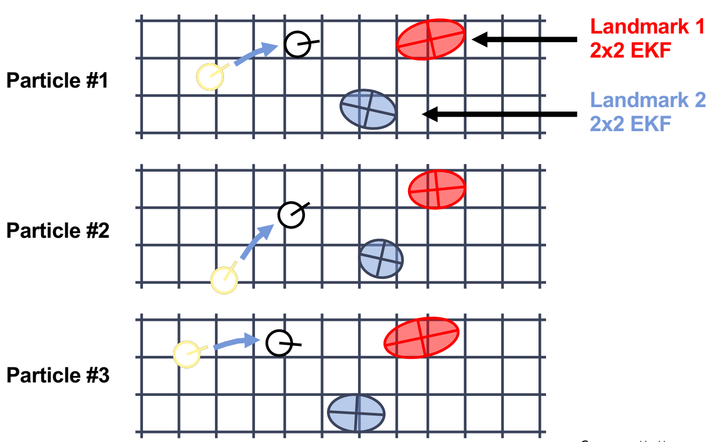
Courtesy: C. Stachniss

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

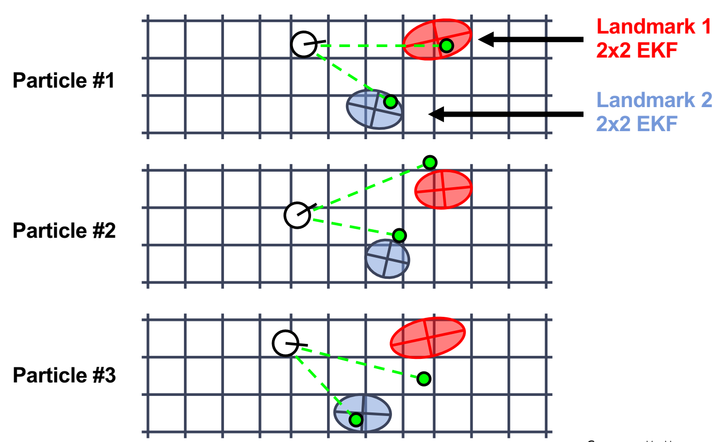
Particle 1	x, y, θ	Landmark 1	Landmark 2	…	Landmark *M*
Particle 2	x, y, θ	Landmark 1	Landmark 2	…	Landmark *M*
⋮					
Particle *N*	x, y, θ	Landmark 1	Landmark 2	…	Landmark *M*

FastSLAM – Motion Update



Courtesy: M. Montemerlo

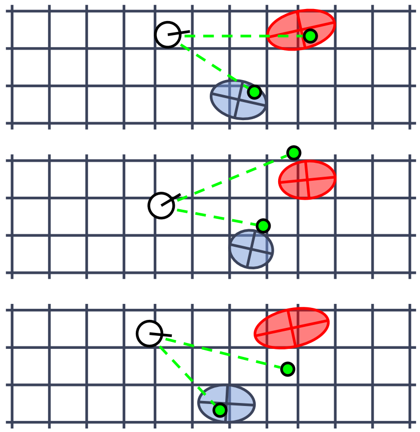
FastSLAM – Sensor Update



Courtesy: M. Montemerlo

FastSLAM – Sensor Update

Particle #1



Weight = 0.8

Particle #2

Weight = 0.4

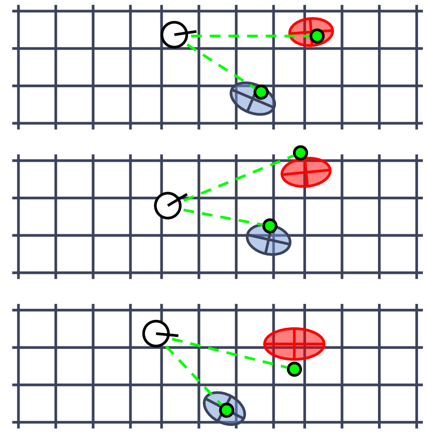
Particle #3

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

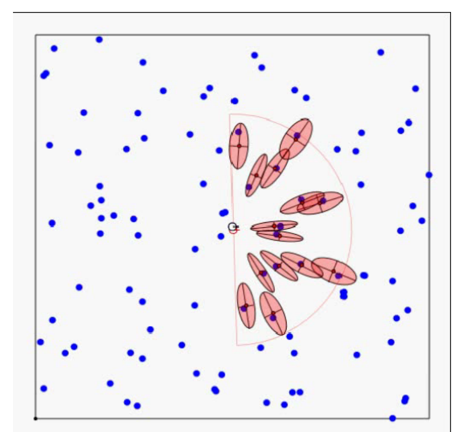
Courtesy: M. Montemerlo

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
  exp. observation
  $$w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \ \exp\left\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} \ (z_t - \hat{z}^{[k]})\right\}$$
  innovation covariance
- Update belief of observed landmarks (EKF update rule)
- Resample

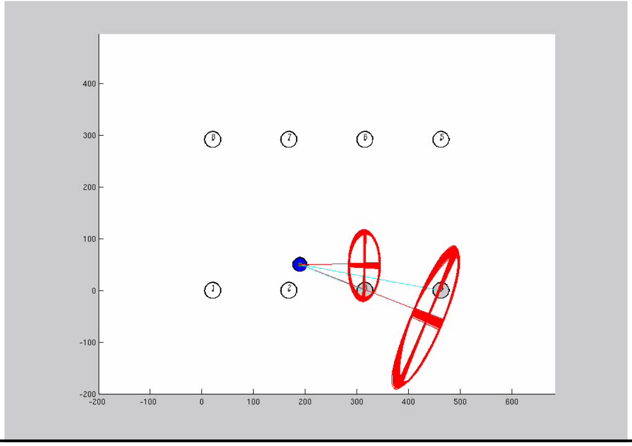
Courtesy: C. Stachniss

FastSLAM in Action

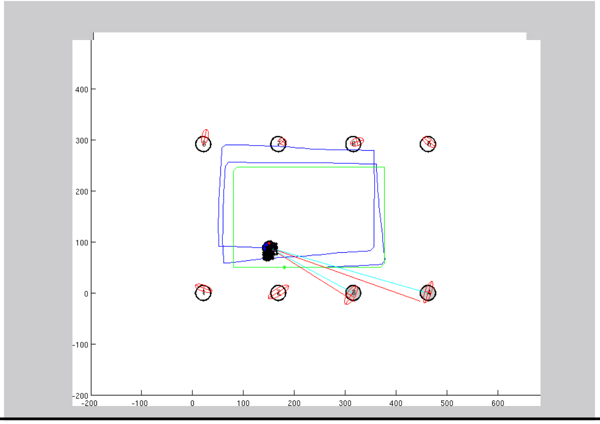


Courtesy: M. Montemerlo

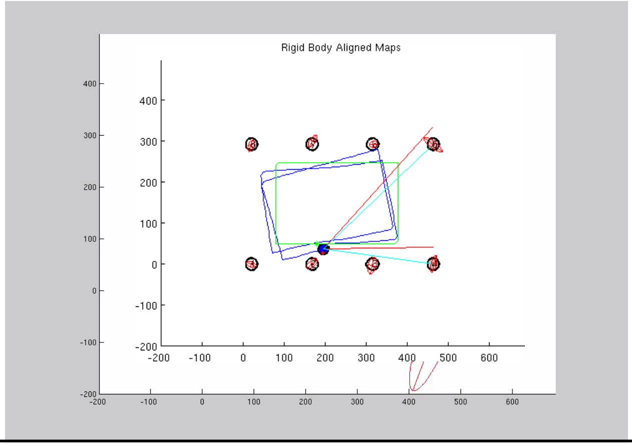
FastSLAM – Video – All Maps



FastSLAM – Video – "Best" particle in terms of Mode of the Posterior

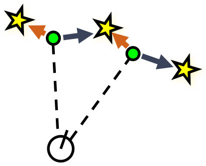


FastSLAM – Video – "Best" particle in terms of Cum Log Prob



Data Association Problem

- Which observation belongs to which landmark?

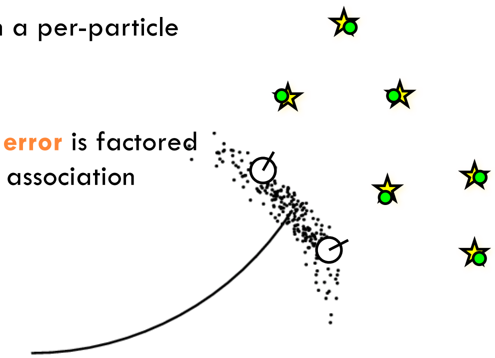


- More than one possible association
- **Potential data associations depend on the pose of the robot**

Courtesy: M. Montemerlo

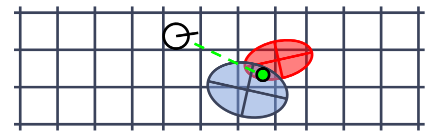
Particles Support for Multi-Hypotheses Data Association

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



Courtesy: M. Montemerlo

Per-Particle Data Association

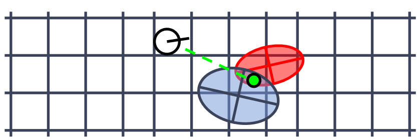


Was the observation generated by the **red** or by the **blue** landmark?

P(observation | red) = 0.3   P(observation | blue) = 0.7

Courtesy: M. Montemerlo

Per-Particle Data Association



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

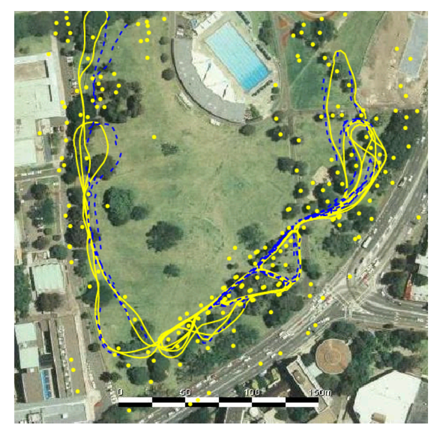
P(observation | red) = 0.3   P(observation | blue) = 0.7

- 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



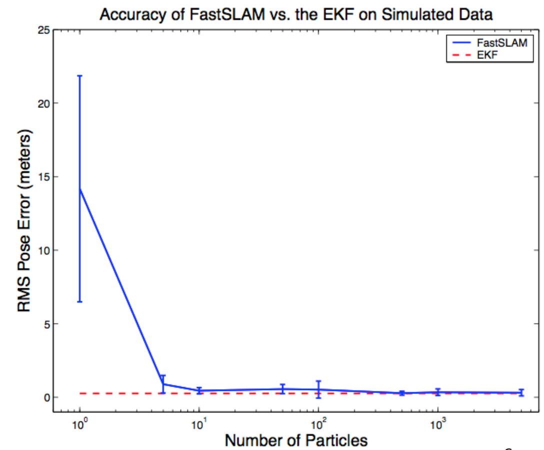
Blue = GPS
Yellow = FastSLAM

Courtesy: M. Montemerlo

Results – Victoria Park (Video)

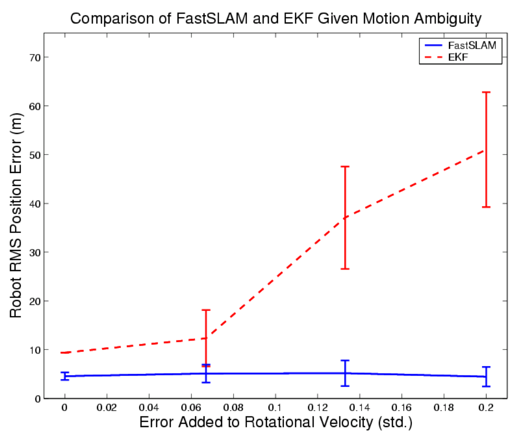
Courtesy: M. Montemerlo

Results (Sample Size)



Courtesy: M. Montemerlo

Results (Motion Uncertainty)



Courtesy: M. Montemerlo

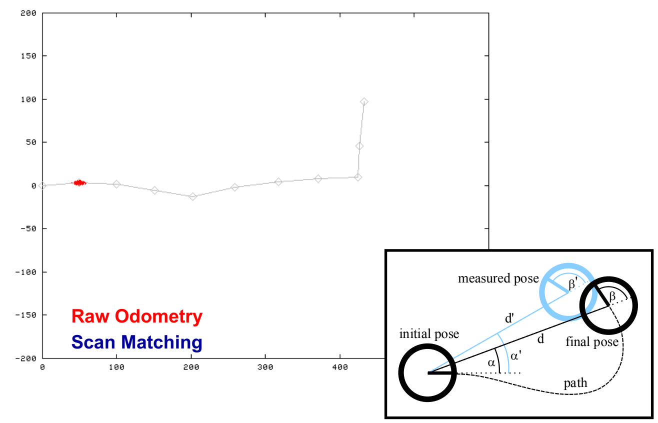
Techniques to Reduce the Number of Particles Needed

- 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.

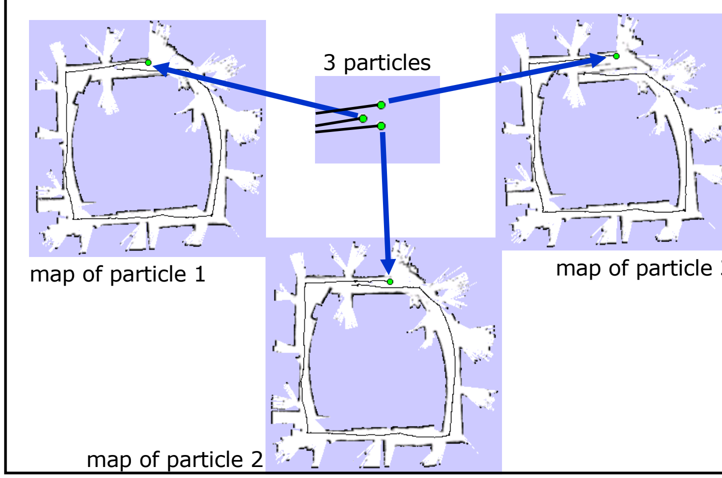
Motion Model for Scan Matching



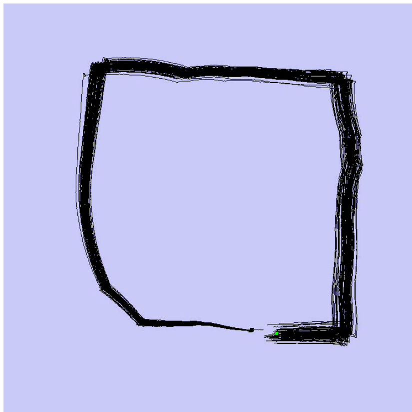
Rao-Blackwellized Mapping with Scan-Matching

Map: Intel Research Lab Seattle

Loop Closure Example

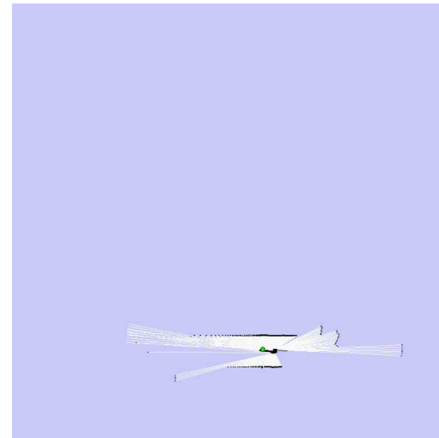


Rao-Blackwellized Mapping with Scan-Matching



Map: Intel Research Lab Seattle

Rao-Blackwellized Mapping with Scan-Matching



Map: Intel Research Lab Seattle

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**
- $250 \times 250 m^2$
- 1.088 miles (odometry)
- 20cm resolution during scan matching
- 30cm resolution in final map

Work by Grisetti et al.

FastSLAM 1.0

- FastSLAM 1.0 uses the motion model as the proposal distribution

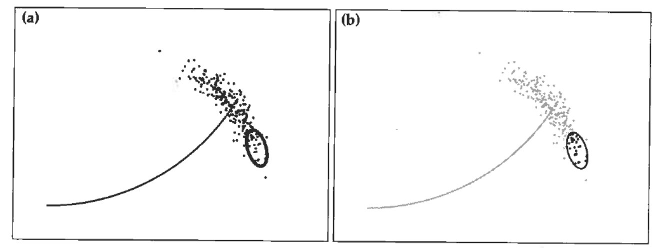
$$x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$$

- **Is there a better distribution to sample from?**

[Montemerlo et al., 2002] Courtesy: C. Stachniss

Weakness of FastSLAM 1.0

- Proposal Distribution
- Importance weighting



FastSLAM 1.0 to FastSLAM 2.0

- FastSLAM 1.0 uses the motion model as the proposal distribution

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

- FastSLAM 2.0 **considers also the measurements during sampling**
- Especially useful if an accurate sensor is used (compared to the motion noise)

[Montemerlo et al., 2003] Courtesy: C. Stachniss

FastSLAM 2.0 (Informally)

- FastSLAM 2.0 samples from

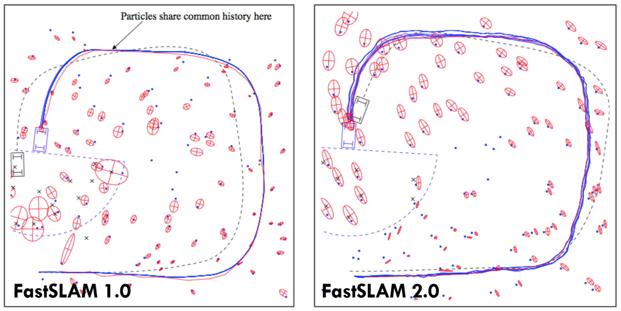
$$x_t^{[k]} \sim p(x_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t})$$

- Results in a more peaked proposal distribution
- Less particles are required
- More robust and accurate
- But more complex…

[Montemerlo et al., 2003] Courtesy: C. Stachniss

FastSLAM Problems

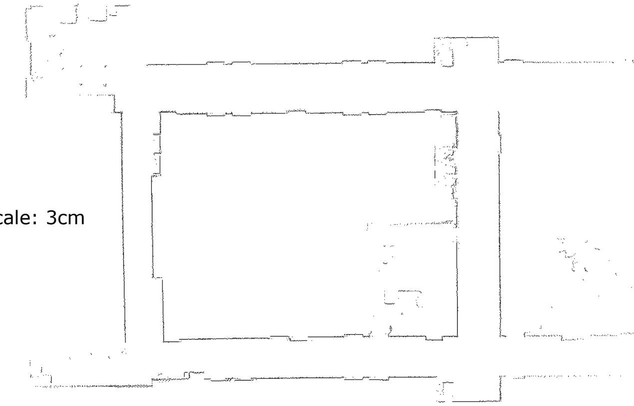
- How to determine the sample size?
- Particle deprivation, especially when closing (multiple) loops



Courtesy: M. Montemerlo

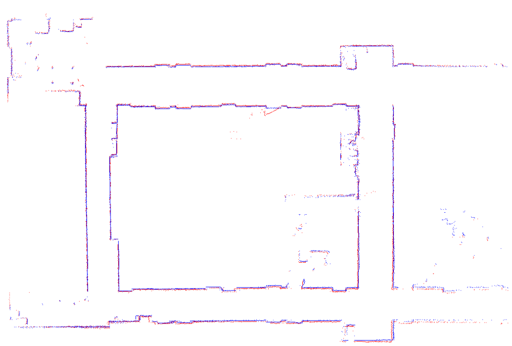
DP-SLAM: High-Res Fast-SLAM via History Sharing

scale: 3cm

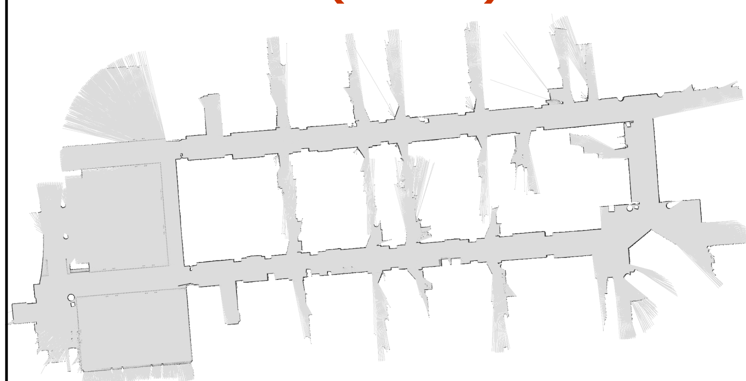


Run at real-time speed on 2.4GHz Pentium 4 at 10cm/s

Consistency



Results obtained with DP-SLAM 2.0 (offline)



Eliazar & Parr, 04

Close up

End courtesy of Eliazar & Parr

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
- FastSLAM 1.0 and 2.0 differ in the proposal distribution
- Complexity $\mathcal{O}(N \log M)$

Courtesy: C. Stachniss

Literature

FastSLAM

- Thrun et al.: "Probabilistic Robotics", Chapter 13.1-13.3 + 13.8 (see errata!)
- Montemerlo, Thrun, Kollar, Wegbreit: FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem, 2002
- Montemerlo and Thrun: Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM, 2003

Courtesy: C. Stachniss