CSE-571
Probabilistic Robotics

Fast-SLAM Mapping

Rao-Blackwellized Mapping

Compute a posterior over the map and possible trajectories of the robot :

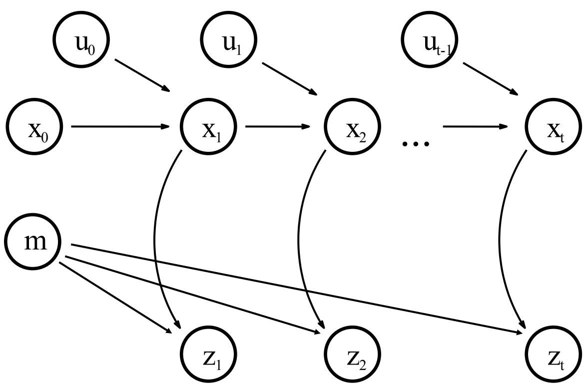
map and trajectory

measurements

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

map robot motion trajectory

A Graphical Model of Rao-Blackwellized Mapping



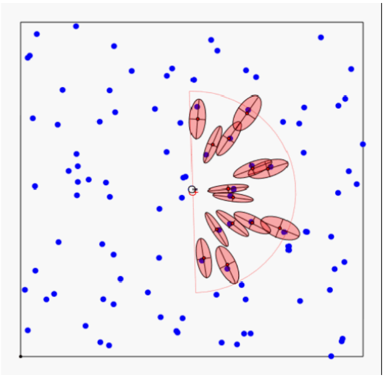
FastSLAM

	Robot Pose	2 x 2 Kalman Filters			
Particle #1	x, y, θ	Landmark 1	Landmark 2	...	Landmark N
Particle #2	x, y, θ	Landmark 1	Landmark 2	...	Landmark N
Particle #3	x, y, θ	Landmark 1	Landmark 2	...	Landmark N
⋮					
Particle M	x, y, θ	Landmark 1	Landmark 2	...	Landmark N

[Begin courtesy of Mike Montemerlo]

FastSLAM – Simulation

- Up to 100,000 landmarks
- 100 particles
- 10^3 times fewer parameters than EKF SLAM



Blue line = true robot path
Red line = estimated robot path
Black dashed line = odometry

Victoria Park Results

- 4 km traverse
- 100 particles
- Uses negative evidence to remove spurious landmarks

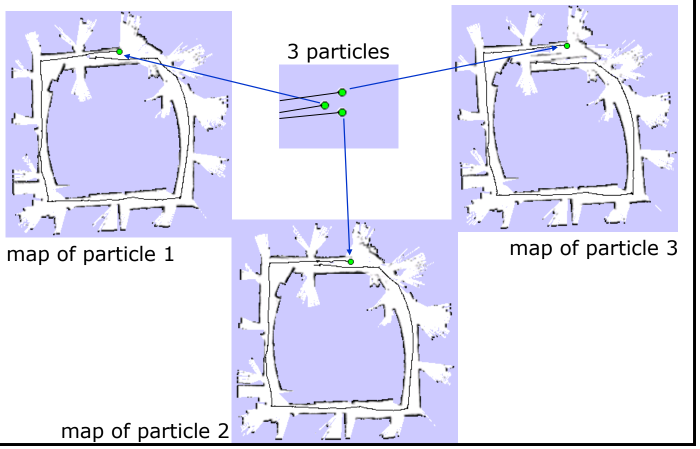
Blue path = odometry
Red path = estimated path

[End courtesy of Mike Montemerlo]

Tasks to be Solved

- Mapping (occupancy grids)
  - Each particle carries its own map m.
  - The history of each particle represents a potential trajectory of the robot.
- Localization
  - Propagate the particles according to the motion model (draw from $p(x|u,x')$).
  - Compute importance weight according to the likelihood of the observation z given the pose x and the map m of the particle.

Example



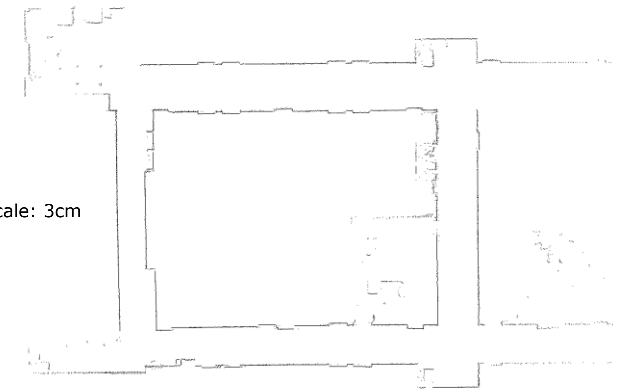
Map Maintenance Challenges

- High resolution maps are big
- Typically 100's or 1000's of particles are needed
- One full map per particle requires
  - O($|m| \cdot n$) work (re-sampling)
  - Gigabytes of memory movement

Begin courtesy of Eliazar & Parr

DP-SLAM Results

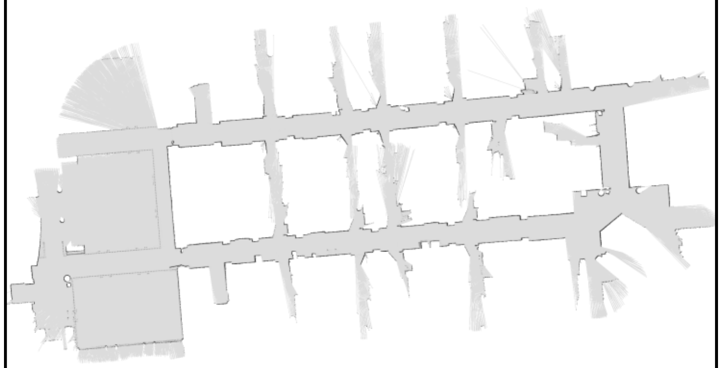
scale: 3cm



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

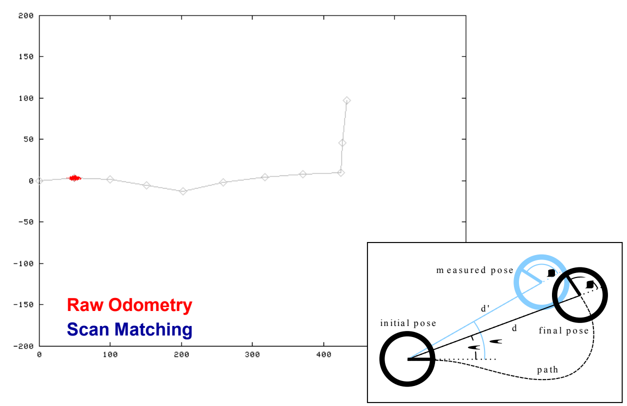
Techniques to Reduce the Number of Particles Needed

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

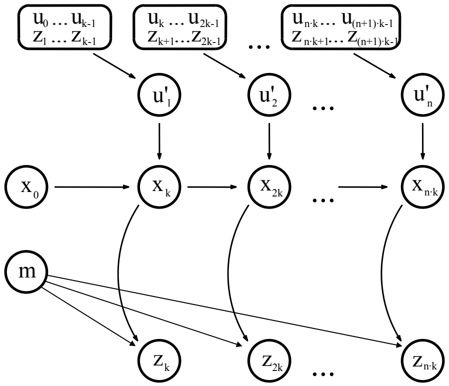
Generating better Proposals

- 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.

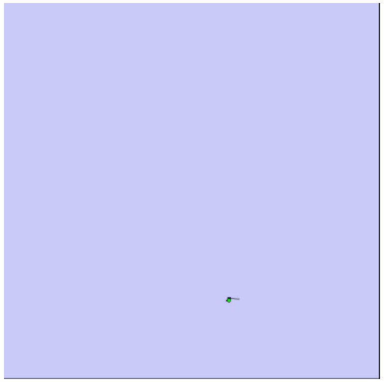
Motion Model for Scan Matching



Graphical Model for Mapping with Improved Odometry

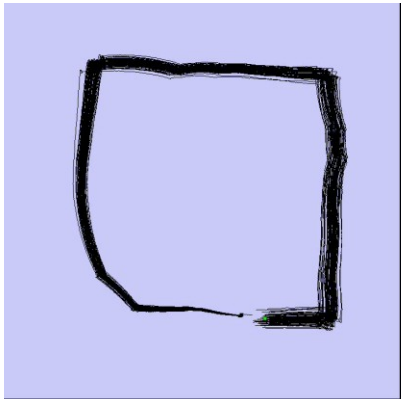


Rao-Blackwellized Mapping with Scan-Matching



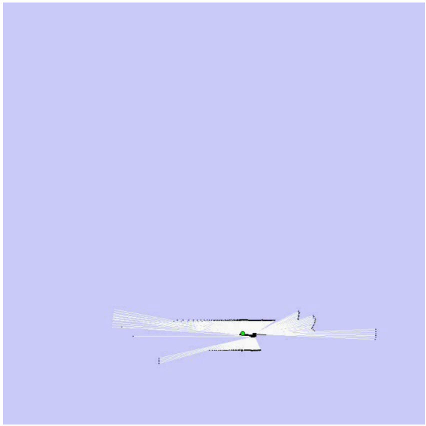
Map: Intel Research Lab Seattle

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**
- $250x250m^2$
- 1.088 miles (odometry)
- 20cm resolution during scan matching
- 30cm resolution in final map

Work by Grisetti et al.

Fast-SLAM Summary

- Full and online version of SLAM
- Factorizes posterior into robot trajectories (particles) and map (EKFs).
- Landmark locations are independent!
- More efficient proposal distribution through Kalman filter prediction
- Data association per particle