Introduction to SLAM

Ron Parr

CPS 1/296

Mapping as Filtering

- Goals of Simultaneous Localization and Mapping
  - Constant time computation per sensor sweep
  - No accumulating error
- Insight: Track map+robot state together
  - SLAM problem is a big HMM/Kalman filter
  - Filtering equations give correct probability distribution over map and robot position, integrating all evidence up to current time step
- Proposed by Smith, Self and Cheeseman in 1990, but not immediately pursued

SLAM Pseudocode

Localization

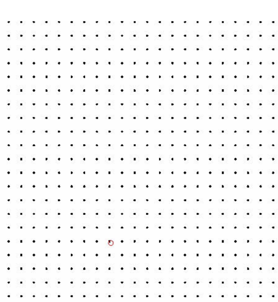
- Project robot state distribution forward (robot motion model)
- Observe environment (laser scans)
- Update robot state by P(O|S)
- Update map (add new objects)
- Repeat

Laser cast tracing
Laser error model

Kalman Filter SLAM Properties

- Assumes:
  - Linear motion model
  - Gaussian noise
- Produces
  - Robot position estimates
  - Landmark position estimates
  - Means and full covariance matrix
- (In most cases, must use EKF)

KF SLAM Example



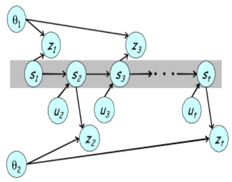
Video courtesy of Mark Paskin

Problems with KF SLAM

- Reality is not linear Gaussian (partially addressed by EKF/UKF)
- Produces only a map of landmarks
- n landmarks: $O(n^2)$ cost
- Data association problem

FastSLAM (Montemerlo et al. 2002)

- View problem as a Bayes net (insight from Murphy)



[Image from Montemerlo et al.]

- Rao Blackwellization for SLAM
  - Samples robot positions
  - KF for landmark positions
  - Benefits of sampling:
    - Fixes unrealistic linear-Gaussian assumption
    - Landmark positions become independent
    - Linear cost in no. of landmarks seen

Map Storage for FastSLAM

- Each map requires linear space in number of landmarks
- Expensive with larger numbers of particles and maps
- Solution: Use copy-on-write

Limitations of FastSLAM

- Doesn't address data association problem
- Doesn't address landmark sparseness issue
- Tends to require a lot of particles over long trajectories
  - See videos from Mark Paskin
  - Why? (discussion)