SEIF, EnKF, EKF SLAM

Pieter Abbeel
UC Berkeley EECS

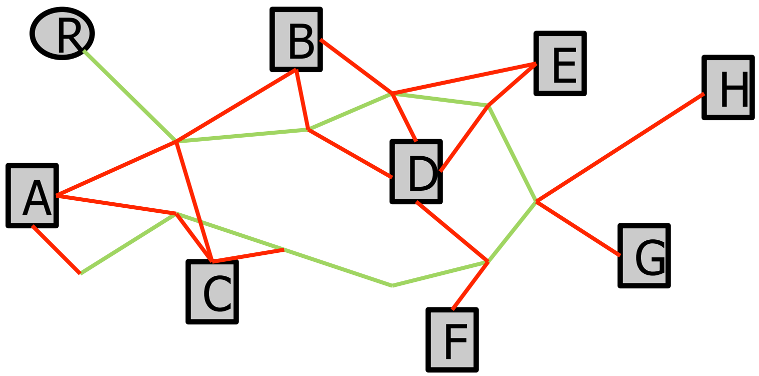
Information Filter

- From an analytical point of view == Kalman filter
- Difference: keep track of the inverse covariance rather than the covariance matrix [matter of some linear algebra manipulations to get into this form]
- Why interesting?
  - Inverse covariance matrix = 0 is easier to work with than covariance matrix = infinity (case of complete uncertainty)
  - Inverse covariance matrix is often sparser than the covariance matrix --- for the "insiders": inverse covariance matrix entry (i,j) = 0 if x_i is conditionally independent of x_j given some set $\{x_k, x_l, \ldots\}$
  - Downside: when extended to non-linear setting, need to solve a linear system to find the mean (around which one can then linearize)
  - See Probabilistic Robotics pp. 78-79 for more in-depth pros/cons and Probabilistic Robotics Chapter 12 for its relevance to SLAM (then often referred to as the "sparse extended information filter (SEIF)")

KF Summary

- Kalman filter exact under linear Gaussian assumptions
- Extension to non-linear setting:
  - Extended Kalman filter
  - Unscented Kalman filter
- Extension to extremely large scale settings:
  - Ensemble Kalman filter
  - Sparse Information filter
- Main limitation: restricted to unimodal / Gaussian looking distributions
- Can alleviate by running multiple XKFs + keeping track of the likelihood; but this is still limited in terms of representational power unless we allow a very large number of them

EKF/UKF SLAM



- State: (n_R, e_R, θ_R, n_A, e_A, n_B, e_B, n_C, e_C, n_D, e_D, n_E, e_E, n_F, e_F, n_G, e_G, n_H, e_H)
  - Now map = location of landmarks (vs. gridmaps)
- Transition model:
  - Robot motion model; Landmarks stay in place

Simultaneous Localization and Mapping (SLAM)

- In practice: robot is not aware of all landmarks from the beginning
- Moreover: no use in keeping track of landmarks the robot has not received any measurements about

→ Incrementally grow the state when new landmarks get encountered.

Simultaneous Localization and Mapping (SLAM)

- Landmark measurement model: robot measures [x_k; y_k], the position of landmark k expressed in coordinate frame attached to the robot:
  - $h(n_R, e_R, \theta_R, n_k, e_k) = [x_k; y_k] = R(\theta) ([n_k; e_k] - [n_R; e_R])$

- Often also some odometry measurements
  - E.g., wheel encoders

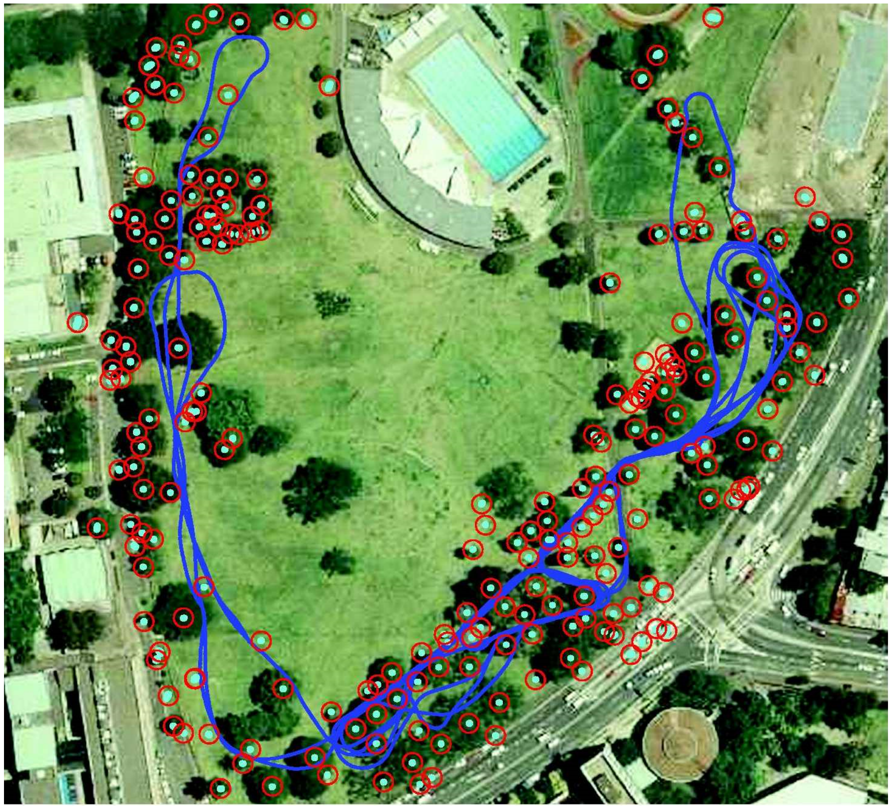
Victoria Park Data Set Vehicle

[courtesy by E. Nebot]

Data Acquisition

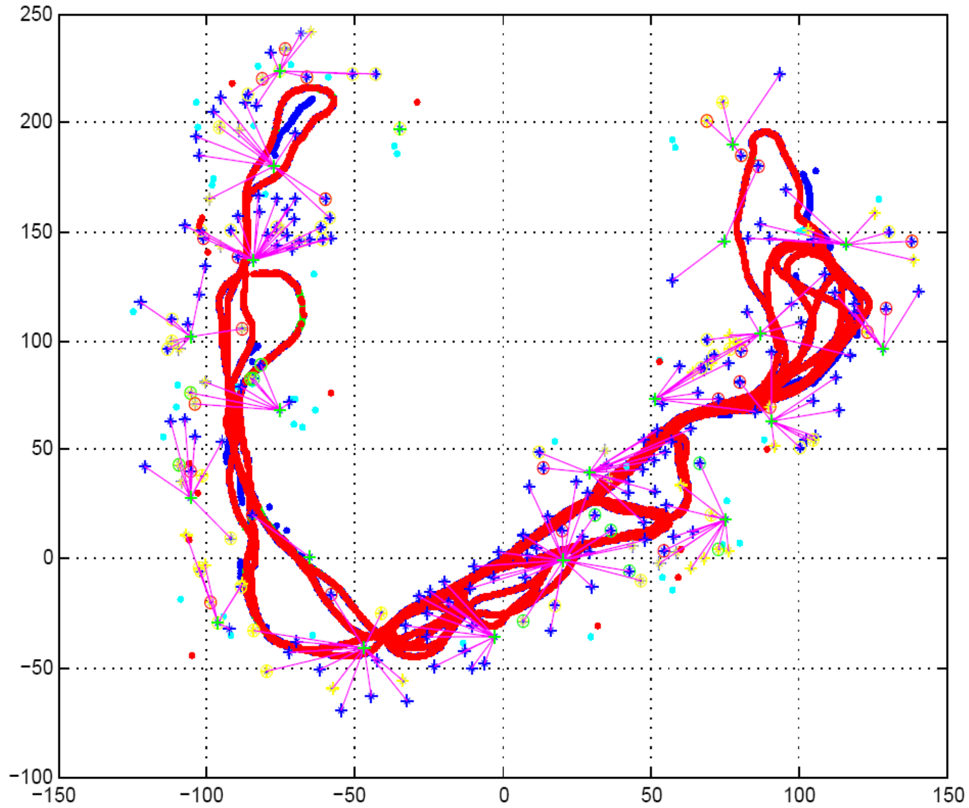
[courtesy by E. Nebot]

Victoria Park Data Set



[courtesy by E. Nebot]

Estimated Trajectory

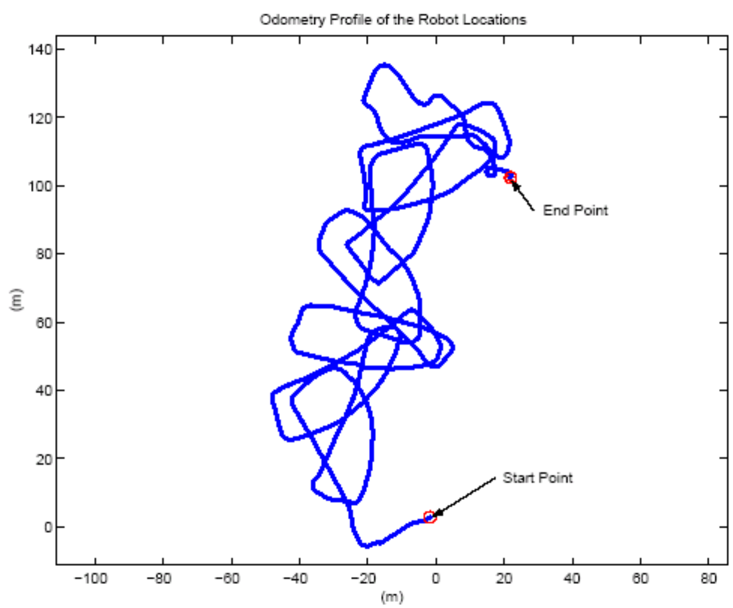


[courtesy by E. Nebot]

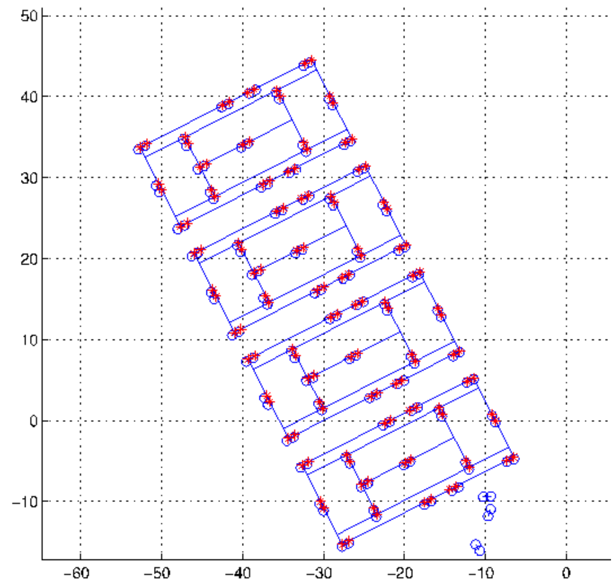
EKF SLAM Application

[courtesy by J. Leonard]

EKF SLAM Application



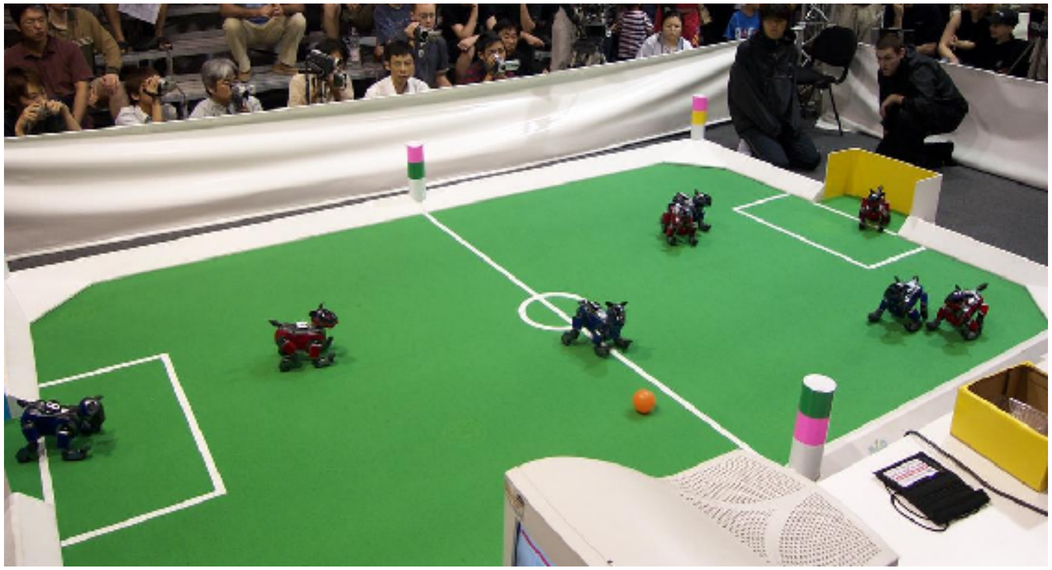
odometry



estimated trajectory

[courtesy by John Leonard]

Landmark-based Localization



EKF-SLAM: practical challenges

- Defining landmarks
  - Laser range finder: Distinct geometric features (e.g. use RANSAC to find lines, then use corners as features)
  - Camera: “interest point detectors”, textures, color, …
- Often need to track multiple hypotheses
  - Data association/Correspondence problem: when seeing features that constitute a landmark --- Which landmark is it?
  - Closing the loop problem: how to know you are closing a loop?
  - → Can split off multiple EKFs whenever there is ambiguity;
  - → Keep track of the likelihood score of each EKF and discard the ones with low likelihood score
- Computational complexity with large numbers of landmarks.