Introduction to Mobile Robotics

SLAM: Simultaneous Localization and Mapping

Wolfram Burgard, Cyrill Stachniss, Maren Bennewitz, Diego Tipaldi, Luciano Spinello
What is SLAM?

- Estimate the pose of a robot and the map of the environment at the same time
- SLAM is hard, because
  - a map is needed for localization and
  - a good pose estimate is needed for mapping
- **Localization**: inferring location given a map
- **Mapping**: inferring a map given locations
- **SLAM**: learning a map and locating the robot simultaneously
The SLAM Problem

- SLAM is a **chicken-or-egg** problem:
  - a map is needed for localization and
  - a pose estimate is needed for mapping
SLAM Applications

- SLAM is central to a range of indoor, outdoor, in-air and underwater applications for both manned and autonomous vehicles.

Examples:
- At home: vacuum cleaner, lawn mower
- Air: surveillance with unmanned air vehicles
- Underwater: reef monitoring
- Underground: exploration of mines
- Space: terrain mapping for localization
SLAM Applications

Indoors

Undersea

Space

Underground
Map Representations

Examples: Subway map, city map, landmark-based map

Maps are **topological** and/or **metric models** of the environment
Map Representations in Robotics

- Grid maps or scans, 2d, 3d

  [Lu & Milios, 97; Gutmann, 98; Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01; Grisetti et al., 05; ...]

- Landmark-based

  [Leonard et al., 98; Castelanos et al., 99; Dissanayake et al., 2001; Montemerlo et al., 2002;...]
The SLAM Problem

- SLAM is considered a fundamental problems for robots to become truly autonomous
- Large variety of different SLAM approaches have been developed
- The majority uses probabilistic concepts
- History of SLAM dates back to the mid-eighties
Feature-Based SLAM

**Given:**
- The robot’s controls
  \[ U_{1:k} = \{u_1, u_2, \ldots, u_k\} \]
- Relative observations
  \[ Z_{1:k} = \{z_1, z_2, \ldots, z_k\} \]

**Wanted:**
- Map of features
  \[ m = \{m_1, m_2, \ldots, m_n\} \]
- Path of the robot
  \[ X_{1:k} = \{x_1, x_2, \ldots, x_k\} \]
Feature-Based SLAM

- **Absolute** robot poses
- **Absolute** landmark positions
- But only **relative** measurements of landmarks
Why is SLAM a hard problem?

1. Robot path and map are both unknown

2. Errors in map and pose estimates correlated
Why is SLAM a hard problem?

- The **mapping between observations and landmarks is unknown**
- Picking **wrong** data associations can have **catastrophic** consequences (divergence)
SLAM: Simultaneous Localization And Mapping

- **Full SLAM:**
  \[ p(x_{0:t}, m \mid z_{1:t}, u_{1:t}) \]
  Estimates entire path and map!

- **Online SLAM:**
  \[ p(x_t, m \mid z_{1:t}, u_{1:t}) = \int \int \cdots \int p(x_{1:t}, m \mid z_{1:t}, u_{1:t}) \, dx_1 dx_2 \cdots dx_{t-1} \]
  Estimates most recent pose and map!

- Integrations (marginalization) typically done recursively, one at a time
Graphical Model of Full SLAM

\[
p(x_{1:t}, m \mid z_{1:t}, u_{1:t})
\]
Graphical Model of Online SLAM

\[ p(x_t, m \mid z_{1:t}, u_{1:t}) = \int \int \ldots \int p(x_{1:t}, m \mid z_{1:t}, u_{1:t}) \, dx_1 \, dx_2 \ldots dx_{t-1} \]
Motion and Observation Model

\[
x_k = f(x_{k-1}, u_k)
\]
"Motion model"

\[
z_k = h(x_k, m)
\]
"Observation model"
Remember the KF Algorithm

1. Algorithm **Kalman_filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

2. Prediction:
   
   $$
   \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t
   $$
   
   $$
   \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t
   $$

3. Correction:
   
   $$
   K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}
   $$
   
   $$
   \mu_t = \mu_t + K_t (z_t - C_t \mu_t)
   $$
   
   $$
   \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t
   $$

4. Return $\mu_t, \Sigma_t$
EKF SLAM: State representation

- **Localization**
  - 3x1 pose vector
  - 3x3 cov. matrix

\[
x_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \quad C_k = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{y\theta} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta}^2 \end{bmatrix}
\]

- **SLAM**

Landmarks **simply extend** the state.
**Growing** state vector and covariance matrix!

\[
x_k = \begin{bmatrix} x_R \\ m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}_k \quad C_k = \begin{bmatrix} C_R & C_{RM_1} & C_{RM_2} & \cdots & C_{RM_n} \\ C_{M_1 R} & C_{M_1} & C_{M_1 M_2} & \cdots & C_{M_1 M_n} \\ C_{M_2 R} & C_{M_2 M_1} & C_{M_2} & \cdots & C_{M_2 M_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{M_n R} & C_{M_n M_1} & C_{M_n M_2} & \cdots & C_{M_n} \end{bmatrix}_k
\]
EKF SLAM: State representation

- Map with $n$ landmarks: $(3+2n)$-dimensional Gaussian

\[
\text{Bel}(x_t, m_t) = \begin{pmatrix}
    x \\
    y \\
    \theta \\
    l_1 \\
    l_2 \\
    \vdots \\
    l_N
\end{pmatrix}
\begin{pmatrix}
    \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} \\
    \sigma_{xy} & \sigma_y^2 & \sigma_{y\theta} \\
    \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 \\
    \sigma_{xl_1} & \sigma_{yl_1} & \sigma_{\theta l_1} \\
    \sigma_{xl_2} & \sigma_{yl_2} & \sigma_{\theta l_2} \\
    \vdots & \vdots & \vdots \\
    \sigma_{xl_N} & \sigma_{yl_N} & \sigma_{\theta l_N}
\end{pmatrix}
\begin{pmatrix}
    \sigma_{xl_1} & \sigma_{xl_2} & \cdots & \sigma_{xl_N} \\
    \sigma_{yl_1} & \sigma_{yl_2} & \cdots & \sigma_{yl_N} \\
    \sigma_{\theta l_1} & \sigma_{\theta l_2} & \cdots & \sigma_{\theta l_N} \\
    \sigma_{l_1l_1} & \sigma_{l_1l_2} & \cdots & \sigma_{l_1l_N} \\
    \sigma_{l_1l_2} & \sigma_{l_2l_2} & \cdots & \sigma_{l_2l_N} \\
    \vdots & \vdots & \vdots & \vdots \\
    \sigma_{l_1l_N} & \sigma_{l_2l_N} & \cdots & \sigma_{l_Nl_N}
\end{pmatrix}
\]

- Can handle hundreds of dimensions
EKF SLAM: Building the Map

Filter Cycle, Overview:

1. State prediction (odometry)
2. Measurement prediction
3. Observation
4. Data Association
5. Update
6. Integration of new landmarks
State Prediction

Odometry:
\[
\hat{x}_R = f(x_R, u)
\]
\[
\hat{C}_R = F_x C_R F_x^T + F_u U F_u^T
\]

Robot-landmark cross-covariance prediction:
\[
\hat{C}_{RM_i} = F_x C_{RM_i}
\]
(skipping time index \(k\))
EKF SLAM: Building the Map

- Measurement Prediction

Global-to-local frame transform $h$

$$\hat{z}_k = h(\hat{x}_k)$$

$$x_k = \begin{bmatrix} x_R \\ m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}_k$$

$$C_k = \begin{bmatrix}
C_R & C_{RM_1} & C_{RM_2} & \cdots & C_{RM_n} \\
C_{M_1R} & C_{M_1} & C_{M_1M_2} & \cdots & C_{M_1M_n} \\
C_{M_2R} & C_{M_2M_1} & C_{M_2} & \cdots & C_{M_2M_n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{M_nR} & C_{M_nM_1} & C_{M_nM_2} & \cdots & C_{M_n} \\
\end{bmatrix}_k$$
EKF SLAM: Building the Map

- Observation

\[(x,y)\text{-point landmarks}\]

\[z_k = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}\]

\[R_k = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}\]

\[x_k = \begin{bmatrix} x_R \\ m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}_{k}\]

\[C_k = \begin{bmatrix} C_R & C_{RM_1} & C_{RM_2} & \cdots & C_{RM_n} \\ C_{M_1 R} & C_{M_1} & C_{M_1 M_2} & \cdots & C_{M_1 M_n} \\ C_{M_2 R} & C_{M_2 M_1} & C_{M_2} & \cdots & C_{M_2 M_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{M_n R} & C_{M_n M_1} & C_{M_n M_2} & \cdots & C_{M_n} \end{bmatrix}_{k}\]
EKF SLAM: Building the Map

- Data Association

Associates predicted measurements $\hat{z}^i_k$ with observation $z^j_k$

$$
\nu^i_j = z^j_k - \hat{z}^i_k \\
S^i_j = R^j_k + H^i \hat{C}_k H^i \text{T}
$$

$$
\begin{align*}
\mathbf{x}_k &= \begin{bmatrix} x_R \\ m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}_k \\
\mathbf{C}_k &= \begin{bmatrix} C_R & C_{RM_1} & C_{RM_2} & \cdots & C_{RM_n} \\ C_{M_1 R} & C_{M_1} & C_{M_1 M_2} & \cdots & \cdots \\ C_{M_2 R} & C_{M_2 M_1} & C_{M_2} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{M_n R} & C_{M_n M_1} & C_{M_n M_2} & \cdots & C_{M_n} \end{bmatrix}_k
\end{align*}
$$
EKF SLAM: Building the Map

- Filter Update

The usual Kalman filter expressions

\[ K_k = \hat{C}_k H^T S_k^{-1} \]

\[ x_k = \hat{x}_k + K_k v_k \]

\[ C_k = (I - K_k H) \hat{C}_k \]
EKF SLAM: Building the Map

- Integrating New Landmarks

State augmented by
\[ m_{n+1} = g(x_R, z_j) \]
\[ C_{M_{n+1}} = G_R C_R G_R^T + G_z R_j G_z^T \]

Cross-covariances:
\[ C_{M_{n+1}M_i} = G_R C_{RM_i} \]
\[ C_{M_{n+1}R} = G_R C_R \]
EKF SLAM

Map

Correlation matrix
EKF SLAM

Map       Correlation matrix
EKF SLAM

Map

Correlation matrix
EKF SLAM: Correlations Matter

- What if we neglected cross-correlations?

\[
C_k = \begin{bmatrix}
C_R & 0 & \cdots & 0 \\
0 & C_{M_1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & C_{M_n}
\end{bmatrix}
\]

\[
C_{RM_i} = 0_{3\times2}
\]

\[
C_{M_iM_{i+1}} = 0_{2\times2}
\]
EKF SLAM: Correlations Matter

- What if we neglected cross-correlations?

\[ C_k = \begin{bmatrix} C_R & 0 & \cdots & 0 \\ 0 & C_{M_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_{M_n} \end{bmatrix}_k \]

\[ C_{RM_i} = 0_{3 \times 2} \]

\[ C_{M_iM_{i+1}} = 0_{2 \times 2} \]

- Landmark and robot uncertainties would become overly optimistic
- Data association would fail
- Multiple map entries of the same landmark
- Inconsistent map
Error Propagation (cont.)

- Want to derive:

\[ C_{YZ} = A \ C_{XZ} \]

- In words: how is the cross-correlation \( C_{XZ} \) between two normally distributed RVs \( X \) and \( Z \) with moments \( x \), \( C_{X} \) and \( z \), \( C_{Z} \) affected by a linear transform of \( X \) of the form

\[ y = A \ x + B \]

- We recall that the following holds:

\[ C_{Y} = A \ C_{X} \ A^{T} \]
Error Propagation (cont.)

- We augment the linear mapping by the variable of interest

\[
\begin{bmatrix}
  y \\
  z
\end{bmatrix}
= \begin{bmatrix}
  A & 0 \\
  0 & I
\end{bmatrix}
\begin{bmatrix}
  x \\
  z
\end{bmatrix}
+ \begin{bmatrix}
  B \\
  0
\end{bmatrix}
\]

- Note that this implements

\[
y = A \, x + B
\]

\[
z = z
\]
Renaming the variables of the augmented system

\[ \mathbf{x}' = [\mathbf{x} \quad \mathbf{z}]^T \quad \mathbf{y}' = [\mathbf{y} \quad \mathbf{z}]^T \]

gives \[ \mathbf{y}' = \mathbf{A}' \mathbf{x}' + \mathbf{B}' \] with the augmented covariance matrices

\[
C_{Y'} = \begin{bmatrix}
C_Y & C_{YZ} \\
C_{ZY} & C_Z
\end{bmatrix} \quad C_{X'} = \begin{bmatrix}
C_X & C_{XZ} \\
C_{ZX} & C_Z
\end{bmatrix}
\]

The augmented covariance matrix is again given by

\[ C_{Y'} = \mathbf{A}' \ C_{X'} \ \mathbf{A}'^T \]
Resubstitution yields

\[
C_{Y'} = \begin{bmatrix}
A & 0 \\
0 & I
\end{bmatrix}
\begin{bmatrix}
C_X & C_{XZ} \\
C_{ZX} & C_Z
\end{bmatrix}
\begin{bmatrix}
A^T & 0 \\
0 & I
\end{bmatrix}
\]

\[
= \begin{bmatrix}
AC_X & AC_{XZ} \\
C_{ZX} & C_Z
\end{bmatrix}
\begin{bmatrix}
A^T & 0 \\
0 & I
\end{bmatrix}
\]

\[
= \begin{bmatrix}
AC_X A^T & AC_{XZ} \\
C_{ZX} A^T & C_Z
\end{bmatrix}
\]

Thus:

\[
C_{YZ} = AC_{XZ}
\]
SLAM: Loop Closure

- **Recognizing an already mapped area**, typically after a long exploration path (the robot "closes a loop")

- Structurally identical to data association, but
  - high levels of ambiguity
  - possibly useless validation gates
  - environment symmetries

- Uncertainties **collapse** after a loop closure (whether the closure was correct or not)
SLAM: Loop Closure

- Before loop closure
SLAM: Loop Closure

- After loop closure
SLAM: Loop Closure

- By revisiting already mapped areas, uncertainties in robot and landmark estimates can be reduced.

- This can be exploited when exploring an environment for the sake of better (e.g. more accurate) maps.

- Exploration: the problem of where to acquire new information

→ See separate chapter on exploration.
KF-SLAM Properties (Linear Case)

- The determinant of any sub-matrix of the map covariance matrix decreases monotonically as successive observations are made.
- When a new landmark is initialized, its uncertainty is maximal.
- Landmark uncertainty decreases monotonically with each new observation.

[Dissanayake et al., 2001]
KF-SLAM Properties (Linear Case)

- In the limit, the landmark estimates become **fully correlated**

[Dissanayake et al., 2001]
KF-SLAM Properties (Linear Case)

- In the limit, the **covariance** associated with any single landmark location estimate is determined only by the **initial covariance in the vehicle location estimate**.

[Dissanayake et al., 2001]
EKF SLAM Example: Victoria Park Dataset
Victoria Park: Data Acquisition

[courtesy by E. Nebot]
Victoria Park: Estimated Trajectory

courtesy by E. Nebot
Victoria Park: Landmarks

[courtesy by E. Nebot]
EKF SLAM Example: Tennis Court

[courtesy by J. Leonard]
EKF SLAM Example: Tennis Court

Odometry Profile of the Robot Locations

odometry

estimated trajectory

[courtesy by John Leonard]
EKF SLAM Example: Line Features

- KTH Bakery Data Set

[Wulf et al., ICRA 04]
EKF-SLAM: Complexity

- Cost per step: quadratic in $n$, the number of landmarks: $O(n^2)$
- Total cost to build a map with $n$ landmarks: $O(n^3)$
- Memory consumption: $O(n^2)$
- Problem: becomes computationally intractable for large maps!
- There exists variants to circumvent these problems
SLAM Techniques

- EKF SLAM
- FastSLAM
- Graph-based SLAM
- Topological SLAM (mainly place recognition)
- Scan Matching / Visual Odometry (only locally consistent maps)
- Approximations for SLAM: Local submaps, Sparse extended information filters, Sparse links, Thin junction tree filters, etc.
- ...
EKF-SLAM: Summary

- The first SLAM solution
- Convergence proof for linear Gaussian case
- Can diverge if nonlinearities are large (and the reality is nonlinear...)
- Can deal only with a single mode
- Successful in medium-scale scenes
- Approximations exist to reduce the computational complexity