Robot Mapping

Sparse Extended Information Filter for SLAM

Cyrill Stachniss
Reminder: Parameterizations for the Gaussian Distribution

**moments**

\[
\Sigma = \Omega^{-1} \\
\mu = \Omega^{-1} \xi
\]

covariance matrix
mean vector

**canonical**

\[
\Omega = \Sigma^{-1} \\
\xi = \Sigma^{-1} \mu
\]

information matrix
information vector
Motivation

Gaussian estimate (map & pose)  normalized covariance matrix  normalized information matrix
Motivation

small but non-zero

normalized information matrix
Most Features Have Only a Small Number of Strong Links
Information Matrix

- Information matrix can be interpreted as a graph of constraints/links between nodes (variables)
- Can be interpreted as a MRF
- Missing links indicate conditional independence of the random variables
- $\Omega_{ij}$ tells us the strength of a link
- Larger values for nearby features
- Most off-diagonal elements in the information are close to 0 (but $\neq 0$)
Create Sparsity

- “Set” most links to zero/avoid fill-in
- Exploit sparseness of $\Omega$ in the computations

- **sparse** = finite number of non-zero off-diagonals, independent of the matrix size
Effect of **Measurement Update** on the Information Matrix

<table>
<thead>
<tr>
<th></th>
<th>$x_t$</th>
<th>$m_1$</th>
<th>$m_2$</th>
<th>$m_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_t$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m_3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

before any observations
Effect of Measurement Update on the Information Matrix

robot observes landmark 1
Effect of **Measurement Update** on the Information Matrix

robot observes landmark 2
Effect of **Measurement Update** on the **Information Matrix**

- Adds information between the robot’s pose and the observed feature
Effect of **Motion Update** on the Information Matrix

before the robot’s movement
Effect of **Motion Update** on the **Information Matrix**

after the robot’s movement
Effect of Motion Update on the Information Matrix

effect of the robot’s movement
Effect of Motion Update on the Information Matrix

- Weakens the links between the robot’s pose and the landmarks
- Add links between landmarks
Sparsification

before sparsification
Sparsification

before sparsification
Sparsification

removal of the link between $m_1$ and $x_{t+1}$
Sparsification

effect of the sparsification
Sparsification

- Sparsification means “ignoring” links (assuming conditional independence)
- Here: links between the robot’s pose and some of the features
Active and Passive Landmarks

Key element of SEIF SLAM to obtain an efficient algorithm

Active Landmarks
- A subset of all landmarks
- Includes the currently observed ones

Passive Landmarks
- All others
Active vs. Passive Landmarks

$xt+1 m_1 m_2 m_3$

was active, now passive
Sparsification in Every Step

- SEIF SLAM conducts a **sparsification** steps *in each iteration*

**Effect:**

- The robot’s pose is linked to the active landmarks only
- Landmarks have only links to nearby landmarks (landmarks that have been active at the same time)
Key Steps of SEIF SLAM

1. Motion update
2. Measurement update
3. Sparsification
Four Steps of SEIF SLAM

1. Motion update
2. Measurement update
3. Update of the state estimate
4. Sparsification

The mean is needed to apply the motion update, for computing an expected measurement and for sparsification.
Four Steps of SEIF SLAM

\[
\text{SEIF\_SLAM}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t, z_t):
\]

1: \( \tilde{\xi}_t, \tilde{\Omega}_t, \tilde{\mu}_t = \text{SEIF\_motion\_update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t) \)

2: \( \xi_t, \Omega_t = \text{SEIF\_measurement\_update}(\tilde{\xi}_t, \tilde{\Omega}_t, \tilde{\mu}_t, z_t) \)

3: \( \mu_t = \text{SEIF\_update\_state\_estimate}(\xi_t, \Omega_t, \tilde{\mu}_t) \)

4: \( \tilde{\xi}_t, \tilde{\Omega}_t = \text{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t) \)

5: \( \text{return } \tilde{\xi}_t, \tilde{\Omega}_t, \mu_t \)

\[\text{Note: we maintain } \xi_t, \Omega_t, \mu_t \]
Four Steps of SEIF SLAM

\[
\text{SEIF\_SLAM}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t, z_t):
\]

1: \( \bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t = \text{SEIF\_motion\_update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t) \)

2: \( \xi_t, \Omega_t = \text{SEIF\_measurement\_update}(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t, z_t) \)

3: \( \mu_t = \text{SEIF\_update\_state\_estimate}(\xi_t, \Omega_t, \bar{\mu}_t) \)

4: \( \tilde{\xi}_t, \tilde{\Omega}_t = \text{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t) \)

5: \( \text{return } \tilde{\xi}_t, \tilde{\Omega}_t, \mu_t \)

The corrected mean \( \mu_t \) is estimated after the measurement update of the canonical parameters \( \xi_t, \Omega_t \)
Four Steps of SEIF SLAM

$$\text{SEIF\_SLAM}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t, z_t):$$

1: $$\xi_t, \Omega_t, \mu_t = \text{SEIF\_motion\_update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t)$$
2: $$\xi_t, \Omega_t = \text{SEIF\_measurement\_update}(\xi_t, \Omega_t, \mu_t, z_t)$$
3: $$\mu_t = \text{SEIF\_update\_state\_estimate}(\xi_t, \Omega_t, \mu_t)$$
4: $$\tilde{\xi}_t, \tilde{\Omega}_t = \text{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t)$$
5: return $$\tilde{\xi}_t, \tilde{\Omega}_t, \mu_t$$
Matrix Inversion Lemma

- Before we start, let us re-visit the matrix inversion lemma

- For any invertible quadratic matrices $R$ and $Q$ and any matrix $P$, the following holds:

\[
(R + PQ^TP^T)^{-1} = R^{-1} - R^{-1}P(Q^{-1} + P^TR^{-1}P)^{-1}P^TR^{-1}
\]
SEIF SLAM – Prediction Step

- Goal: Compute $\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t$ from motion and the previous estimate $\bar{\xi}_{t-1}, \bar{\Omega}_{t-1}, \bar{\mu}_{t-1}$
- Efficiency by exploiting sparseness of the information matrix
Let us start from EKF SLAM...

\[
\text{EKF\_SLAM\_Prediction}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, R_t):
\]

2: \( F_x = \begin{pmatrix}
1 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 1 & 0 & \cdots & 0
\end{pmatrix} \)

3: \( \bar{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix}
-\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\
\frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\
\omega_t \Delta t
\end{pmatrix} \)

4: \( G_t = I + F_x^T \begin{pmatrix}
0 & 0 & -\frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\
0 & 0 & -\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\
0 & 0 & 0
\end{pmatrix} F_x \)

5: \( \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + F_x^T R_t F_x \)
Let us start from EKF SLAM...

\[
\text{EKF\_SLAM\_Prediction}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, R_t): \\
2: \quad F_x = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}
\]

\[
3: \quad \bar{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}
\]

\[
4: \quad G_t = I + F_x^T \begin{pmatrix} 0 & 0 & -\frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\ 0 & 0 & -\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\ 0 & 0 & 0 \end{pmatrix} F_x
\]

\[
5: \quad \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + F_x^T R_t^x F_x
\]
Let us start from EKF SLAM…

**EKF\_SLAM\_Prediction**\((\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, R_t)\):

2: \[ F_x = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix} \]

3: \[ \tilde{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix} \]

4: \[ G_t = I + F_x^T \begin{pmatrix} 0 & 0 & -\frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\ 0 & 0 & -\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\ 0 & 0 & 0 \end{pmatrix} F_x \]

5: \[ \tilde{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + \begin{pmatrix} F_x^T R_x F_x \end{pmatrix} \]

let’s begin with computing the information matrix...
Algorithm **SEIF\_motion\_update**($\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t$):

2: $F_x = \begin{pmatrix}
1 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & \cdots & 0 \\
\end{pmatrix}

3: $\delta = \begin{pmatrix}
-\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\
\frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\
\omega_t \Delta t \\
\end{pmatrix}$

4: $\Delta = \begin{pmatrix}
0 & 0 & \frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\
0 & 0 & \frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} - \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\
0 & 0 & 0 \\
\end{pmatrix}$
Compute the Information Matrix

- Computing the information matrix

\[
\bar{\Omega}_t = \bar{\Sigma}_t^{-1} \\
= \left[ G_t \Omega_{t-1}^{-1} G_t^T + R_t \right]^{-1} \\
= \left[ \Phi_t^{-1} + R_t \right]^{-1}
\]

- with the term \( \Phi_t \) defined as

\[
\Phi_t = \left[ G_t \Omega_{t-1}^{-1} G_t^T \right]^{-1} \\
= \left[ G_t^T \right]^{-1} \Omega_{t-1} G_t^{-1}
\]
Compute the Information Matrix

- We can expand the noise matrix $R$

\[
\bar{\Omega}_t = \left[ \Phi_t^{-1} + R_t \right]^{-1} \\
= \left[ \Phi_t^{-1} + F_x^T R_t^x F_x \right]^{-1}
\]
Compute the Information Matrix

- Apply the matrix inversion lemma

\[
\tilde{\Omega}_t = \left[ \Phi_t^{-1} + R_t \right]^{-1}
\]
\[
= \left[ \Phi_t^{-1} + F_x^T R_t^x F_x \right]^{-1}
\]
\[
= \Phi_t - \Phi_t F_x^T (R_t^{x^{-1}} + F_x \Phi_t F_x^T)^{-1} F_x \Phi_t
\]

3x3 matrix
Compute the Information Matrix

- Apply the matrix inversion lemma

\[ \tilde{\Omega}_t = \left[ \Phi_t^{-1} + R_t \right]^{-1} \]
\[ = \left[ \Phi_t^{-1} + F_x^T R_t^x F_x \right]^{-1} \]
\[ = \Phi_t - \Phi_t F_x^T \left( R_t^x^{-1} + F_x \Phi_t F_x^T \right)^{-1} F_x \Phi_t \]

3x3 matrix

Zero except 3x3 block

Zero except 3x3 block
Compute the Information Matrix

- Apply the matrix inversion lemma

\[
\tilde{\Omega}_t = \left[ \Phi_t^{-1} + R_t \right]^{-1}
\]

\[
= \left[ \Phi_t^{-1} + F_x^T R_t^x F_x \right]^{-1}
\]

\[
= \Phi_t - \Phi_t \ F_x^T \left( R_t^{x-1} + F_x \Phi_t F_x^T \right)^{-1} \ F_x \Phi_t
\]

- Constant complexity if \( \Phi_t \) is sparse!
Compute the Information Matrix

- This can be written as

\[
\bar{\Omega}_t = \left[ \Phi_t^{-1} + R_t \right]^{-1} \\
= \left[ \Phi_t^{-1} + F_x^T R_t^x F_x \right]^{-1} \\
= \Phi_t - \underbrace{\Phi_t F_x^T (R_t^{x^{-1}} + F_x \Phi_t F_x^T)^{-1} F_x \Phi_t}_{\kappa_t} \\
= \bar{\Phi}_t - \kappa_t
\]

- Question: Can we compute \( \Phi_t \) efficiently (\( \Phi_t = [G_t^T]^{-1} \Omega_{t-1} G_t^{-1} \))?
Computing \( \Phi_t = [G_t^T]^{-1} \Omega_{t-1} G_t^{-1} \)

- Goal: constant time if \( \Omega_{t-1} \) is sparse
Computing $\Phi_t = \left[ G_t^T \right]^{-1} \Omega_{t-1} G_t^{-1}$

- **Goal:** constant time if $\Omega_{t-1}$ is sparse

$$G_t^{-1} = (I + F_x^T \Delta F_x)^{-1}$$

$$= \begin{pmatrix}
\Delta + I_3 & 0 \\
0 & I_{2N}
\end{pmatrix}^{-1}$$

3x3 identity 2Nx2N identity
Computing $\Phi_t = \left[ G_t^T \right]^{-1} \Omega_{t-1} G_t^{-1}$

- Goal: constant time if $\Omega_{t-1}$ is sparse

$$G_t^{-1} = (I + F_x^T \Delta F_x)^{-1}$$
$$= \left( \begin{array}{cc} \Delta + I_3 & 0 \\ 0 & I_{2N} \end{array} \right)^{-1}$$
$$= \left( \begin{array}{cc} (\Delta + I_3)^{-1} & 0 \\ 0 & I_{2N} \end{array} \right)$$

holds for all block matrices where the off-diagonal blocks are zero
Computing $\Phi_t = [G_t^T]^{-1} \Omega_{t-1} G_t^{-1}$

- Goal: constant time if $\Omega_{t-1}$ is sparse

\[
G_t^{-1} = (I + F_x^T \Delta F_x)^{-1} \\
= \begin{pmatrix} \Delta + I_3 & 0 \\ 0 & I_{2N} \end{pmatrix}^{-1} \\
= \begin{pmatrix} (\Delta + I_3)^{-1} & 0 \\ 0 & I_{2N} \end{pmatrix} \\
= I_{3+2N} + \begin{pmatrix} (\Delta + I_3)^{-1} - I_3 & 0 \\ 0 & 0 \end{pmatrix}
\]

Note: 3x3 matrix
Computing $\Phi_t = [G_t^T]^{-1} \Omega_{t-1} G_t^{-1}$

- Goal: constant time if $\Omega_{t-1}$ is sparse

\[
G_t^{-1} = (I + F_x^T \Delta F_x)^{-1} \\
= \begin{pmatrix} \Delta + I_3 & 0 \\ 0 & I_{2N} \end{pmatrix}^{-1} \\
= \begin{pmatrix} (\Delta + I_3)^{-1} & 0 \\ 0 & I_{2N} \end{pmatrix} \\
= I_{3+2N} + \begin{pmatrix} (\Delta + I_3)^{-1} - I_3 & 0 \\ 0 & 0 \end{pmatrix} \\
= I + F_x^T [(I + \Delta)^{-1} - I] F_x \\
= I + \psi_t
\]
Computing \( \Phi_t = [G_t^T]^{-1} \Omega_{t-1} G_t^{-1} \)

- We have
  \[
  G_t^{-1} = I + \Psi_t \quad \quad \quad [G_t^T]^{-1} = I + \Psi_t^T
  \]

- with
  \[
  \Psi_t = F_x^T [(I + \Delta)^{-1} - I] F_x \\
  \text{3x3 matrix}
  \]

- \( \Psi_t \) is zero except of a 3x3 block
- \( G_t^{-1} \) is an identity except of a 3x3 block
Computing $\Phi_t = [G^T_t]^{-1} \Omega_{t-1} G_t^{-1}$

Given that:
- $G_t^{-1}$ and $[G^T_t]^{-1}$ are identity matrices except of a 3x3 block
- The information matrix is sparse
- This implies that

$$\Phi_t = [G^T_t]^{-1} \Omega_{t-1} G_t^{-1}$$

- can be computed in constant time
Constant Time Computation of $\Phi_t$

- Given $\Omega_{t-1}$ is sparse, the constant time update can be seen by

\[
\Phi_t = [G_t^T]^{-1} \Omega_{t-1} G_t^{-1} \\
= (I + \Psi_t^T) \Omega_{t-1} (I + \Psi_t) \\
= \Omega_{t-1} + \lambda_t \\
= \Omega_{t-1} + \Psi_t \Omega_{t-1} + \Omega_{t-1} \Psi_t + \Psi_t \Omega_{t-1} \Psi_t \\
= \Omega_{t-1} + \lambda_t
\]

all elements zero except a constant number of entries
Prediction Step in Brief

- Compute $\Psi_t$
- Compute $\lambda_t$ using $\Psi_t$
- Compute $\Phi_t$ using $\lambda_t$
- Compute $\kappa_t$ using $\Phi_t$
- Compute $\bar{\Omega}_t$ using $\Phi_t$ and $\kappa_t$
SEIF – Prediction Step (2/3)

SEIF\_motion\_update(\(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t\)):

2: \(F_x = \cdots\)
3: \(\delta = \cdots\)
4: \(\Delta = \cdots\)
5: \(\Psi_t = F_x^T \left((I + \Delta)^{-1} - I\right) F_x\)
6: \(\lambda_t = \Psi_t^T \Omega_{t-1} + \Omega_{t-1} \Psi_t + \Psi_t^T \Omega_{t-1} \Psi_t\)
7: \(\Phi_t = \Omega_{t-1} + \lambda_t\)
8: \(\kappa_t = \Phi_t F_x^T \left(R_t^{-1} + F_x \Phi_t F_x^T\right)^{-1} F_x \Phi_t\)
9: \(\bar{\Omega}_t = \Phi_t - \kappa_t\)

Information matrix is computed, now do the same for the information vector and the mean
Compute the Mean

- The mean is computed as in the EKF

\[
\bar{\mu}_t = \mu_{t-1} + F_x^T \delta
\]

- Reminder (from SEIF motion update)

\[
2: \quad F_x = \begin{pmatrix}
1 & 0 & 0 & 0 & \ldots & 0 \\
0 & 1 & 0 & 0 & \ldots & 0 \\
0 & 0 & 1 & 0 & \ldots & 0 \\
\end{pmatrix}_{2N}
\]

\[
3: \quad \delta = \begin{pmatrix}
-\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\
\frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\
\omega_t \Delta t
\end{pmatrix}
\]
Compute the Information Vector

- We obtain the information vector by

\[ \bar{\xi}_t = \bar{\Omega}_t \left( \mu_{t-1} + F_x^T \delta_t \right) \]

\[ = \bar{\Omega}_t \left( \Omega_{t-1}^{-1} \xi_{t-1} + F_x^T \delta_t \right) \]
Compute the Information Vector

- We obtain the information vector by

\[ \bar{\xi}_t = \bar{\Omega}_t \left( \mu_{t-1} + F_x^T \delta_t \right) \]
\[ = \bar{\Omega}_t \left( \Omega_{t-1}^{-1} \xi_{t-1} + F_x^T \delta_t \right) \]
\[ = \bar{\Omega}_t \Omega_{t-1}^{-1} \xi_{t-1} + \bar{\Omega}_t F_x^T \delta_t \]
Compute the Information Vector

- We obtain the information vector by

\[
\tilde{\xi}_t = \tilde{\Omega}_t \left( \mu_{t-1} + F_x^T \delta_t \right) \\
= \tilde{\Omega}_t \left( \Omega_{t-1}^{-1} \xi_{t-1} + F_x^T \delta_t \right) \\
= \tilde{\Omega}_t \left( \Omega_{t-1}^{-1} \xi_{t-1} + \tilde{\Omega}_t F_x^T \delta_t \right) \\
= \left( \tilde{\Omega}_t - \Phi_t + \Phi_t - \Omega_{t-1} + \Omega_{t-1} \right) \Omega_{t-1}^{-1} \xi_{t-1} + \tilde{\Omega}_t F_x^T \delta_t \\
= 0 + 0
\]
Compute the Information Vector

We obtain the information vector by

\[ \tilde{\xi}_t = \bar{\Omega}_t \left( \mu_{t-1} + F_x^T \delta_t \right) \]

\[ = \bar{\Omega}_t \left( \Omega_{t-1}^{-1} \xi_{t-1} + F_x^T \delta_t \right) \]

\[ = \bar{\Omega}_t \Omega_{t-1}^{-1} \xi_{t-1} + \bar{\Omega}_t F_x^T \delta_t \]

\[ = \left( \bar{\Omega}_t - \Phi_t + \Phi_t - \Omega_{t-1} + \Omega_{t-1} \right) \Omega_{t-1}^{-1} \xi_{t-1} + \bar{\Omega}_t F_x^T \delta_t \]

\[ = \left( \bar{\Omega}_t - \Phi_t + \Phi_t - \Omega_{t-1} \right) \Omega_{t-1}^{-1} \xi_{t-1} + \Omega_{t-1} \Omega_{t-1}^{-1} \xi_{t-1} + \bar{\Omega}_t F_x^T \delta_t \]

\[ = -\kappa_t \xi_{t-1} + \lambda_t \Omega_{t-1}^{-1} \xi_{t-1} + \mu_{t-1} \xi_{t-1} + I \Omega_{t-1}^{-1} \xi_{t-1} + \bar{\Omega}_t F_x^T \delta_t \]

\[ = \left( \Omega_{t-1}^{-1} \xi_{t-1} + \mu_{t-1} \right) + \bar{\Omega}_t F_x^T \delta_t \]
Compute the Information Vector

- We obtain the information vector by

\[
\tilde{\xi}_t
= \tilde{\Omega}_t \left( \mu_{t-1} + F_x^T \delta_t \right)
= \tilde{\Omega}_t \left( \Omega_{t-1}^{-1} \xi_{t-1} + F_x^T \delta_t \right)
= \tilde{\Omega}_t \Omega_{t-1}^{-1} \xi_{t-1} + \tilde{\Omega}_t F_x^T \delta_t
= \left( \tilde{\Omega}_t - \Phi_t + \Phi_t - \Omega_{t-1} + \Omega_{t-1} \right) \Omega_{t-1}^{-1} \xi_{t-1} + \tilde{\Omega}_t F_x^T \delta_t
= \left( \tilde{\Omega}_t - \Phi_t + \Phi_t - \Omega_{t-1} \right) \Omega_{t-1}^{-1} \xi_{t-1} + \Omega_{t-1} \Omega_{t-1}^{-1} \xi_{t-1} + \tilde{\Omega}_t F_x^T \delta_t
= \xi_{t-1} + \left( \lambda_t - \kappa_t \right) \mu_{t-1} + \tilde{\Omega}_t F_x^T \delta_t
\]
SEIF – Prediction Step (3/3)

SEIF\_motion\_update(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t):\n\begin{align*}
2: & \quad F_x = \cdots \\
3: & \quad \delta = \cdots \\
4: & \quad \Delta = \cdots \\
5: & \quad \Psi_t = F_x^T [(I + \Delta)^{-1} - I] F_x \\
6: & \quad \lambda_t = \Psi_t^T \Omega_{t-1} + \Omega_{t-1} \Psi_t + \Psi_t^T \Omega_{t-1} \Psi_t \\
7: & \quad \Phi_t = \Omega_{t-1} + \lambda_t \\
8: & \quad \kappa_t = \Phi_t F_x^T (R_t^{-1} + F_x \Phi_t F_x^T)^{-1} F_x \Phi_t \\
9: & \quad \bar{\Omega}_t = \Phi_t - \kappa_t \\
10: & \quad \bar{\xi}_t = \xi_{t-1} + (\lambda_t - \kappa_t) \mu_{t-1} + \bar{\Omega}_t F_x^T \delta_t \\
11: & \quad \bar{\mu}_t = \mu_{t-1} + F_x^T \delta \\
12: & \quad \text{return } \bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t
\end{align*}
Four Steps of SEIF SLAM

\[ \text{SEIF\_SLAM}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t, z_t): \]

1: \( \bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t = \text{SEIF\_motion\_update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t) \)  
2: \( \xi_t, \Omega_t = \text{SEIF\_measurement\_update}(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t, z_t) \)  
3: \( \mu_t = \text{SEIF\_update\_state\_estimate}(\xi_t, \Omega_t, \bar{\mu}_t) \)  
4: \( \tilde{\xi}_t, \tilde{\Omega}_t = \text{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t) \)  
5: return \( \tilde{\xi}_t, \tilde{\Omega}_t, \mu_t \)
SEIF – Measurement (1/2)

SEIF Measurement update ($\bar{\xi}_t, \bar{\Omega}_t, \mu_t, \tilde{z}_t$)

1: $Q_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix}$
2: for all observed features $z^i_t = (r^i_t, \phi^i_t)^T$ do
3: $j = c^i_t$ (data association)
4: if landmark $j$ never seen before
5: \[
\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \end{pmatrix} + \begin{pmatrix} r^i_t \cos(\phi^i_t + \bar{\mu}_{t,\theta}) \\ r^i_t \sin(\phi^i_t + \bar{\mu}_{t,\theta}) \end{pmatrix}
\]
6: endif
7: $\delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix}$
8: $q = \delta^T \delta$
9: $\tilde{z}^i_t = \begin{pmatrix} \sqrt{q} \\ \text{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \end{pmatrix}$

identical to the EKF SLAM
SEIF – Measurement (2/2)

10: \[ H_t^i = \frac{1}{q} \begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & 0 & \ldots & 0 & +\sqrt{q}\delta_x & \sqrt{q}\delta_y & 0 & \ldots & 0 \\ \delta_y & -\delta_x & -q & \underbrace{0 & \ldots & 0}_{2j-2} & -\delta_y & +\delta_x & \underbrace{0 & \ldots & 0}_{2N-2j} \end{pmatrix} \]

11: \text{endfor}
12: \[ \xi_t = \bar{\xi}_t + \sum_i H_t^{iT} Q_t^{-1} \left[ z_t^i - \hat{z}_t^i + H_t^i \mu_t \right] \]
13: \[ \Omega_t = \bar{\Omega}_t + \sum_i H_t^{iT} Q_t^{-1} H_t^i \]
14: \text{return } \xi_t, \Omega_t

Difference to EKF (but as in EIF):

\[ \xi_t = \bar{\xi}_t + \sum_i H_t^{iT} Q_t^{-1} \left[ z_t^i - \hat{z}_t^i + H_t^i \mu_t \right] \]

\[ \Omega_t = \bar{\Omega}_t + \sum_i H_t^{iT} Q_t^{-1} H_t^i \]
Four Steps of SEIF SLAM

\textbf{SEIF\_SLAM}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t, z_t):

1: \quad \bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t = \text{SEIF\_motion\_update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t, z_t)

2: \quad \xi_t, \Omega_t = \text{SEIF\_measurement\_update}(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t, z_t)

3: \quad \mu_t = \text{SEIF\_update\_state\_estimate}(\xi_t, \Omega_t, \bar{\mu}_t)

4: \quad \tilde{\xi}_t, \tilde{\Omega}_t = \text{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t)

5: \quad \text{return } \tilde{\xi}_t, \tilde{\Omega}_t, \mu_t
Recovering the Mean

The mean is needed for the

- linearized motion model (pose)
- linearized measurement model (pose and visible landmarks)
- sparsification step (pose and subset of the landmarks)
**Recovering the Mean**

- In the motion update step, we can compute the predicted mean easily

<table>
<thead>
<tr>
<th>SEIF_motion_update((\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t)):</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-10:...</td>
</tr>
<tr>
<td>11: (\bar{\mu}<em>t = \mu</em>{t-1} + F_x^T \delta)</td>
</tr>
<tr>
<td>12: return (\xi_t, \Omega_t, \bar{\mu}_t)</td>
</tr>
</tbody>
</table>
Recovering the Mean

- Computing the corrected mean, however, cannot be done as easy
- Computing the mean from the information vector is costly:

\[ \mu = \Omega^{-1} \xi \]

- Thus, SEIF SLAM approximates the computation for the corrected mean
Approximation of the Mean

- Compute a **few dimensions** of the mean in an **approximated** way
- Idea: Treat that as an optimization problem and seek to find

\[
\hat{\mu} = \arg\max_{\mu} p(\mu)
\]

\[
= \arg\max_{\mu} \exp \left(-\frac{1}{2} \mu^T \Omega \mu + \xi^T \mu \right)
\]

- Seeks to find the value that maximize the probability density function
Approximation of the Mean

- Derive function
- Set first derivative to zero
- Solve equation(s)
- Iterate

- Can be done effectively given that only a few dimensions of $\mu$ are needed (robot’s pose and active landmarks)

no further details here...
Four Steps of SEIF SLAM

\[
\text{SEIF\_SLAM}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t, z_t):
\]

1: \quad \bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t = \text{SEIF\_motion\_update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t) \quad \text{DONE}

2: \quad \xi_t, \Omega_t = \text{SEIF\_measurement\_update}(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t, z_t) \quad \text{DONE}

3: \quad \mu_t = \text{SEIF\_update\_state\_estimate}(\xi_t, \Omega_t, \bar{\mu}_t) \quad \text{DONE}

4: \quad \tilde{\xi}_t, \tilde{\Omega}_t = \text{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t)

5: \quad \text{return } \tilde{\xi}_t, \tilde{\Omega}_t, \mu_t
Sparsification

- In order to perform all previous computations efficiently, we assumed a **sparse information matrix**
- Sparsification step ensures that
- **Question:** what does sparsifying the information matrix mean?
Sparsification

- Question: what does sparsifying the information matrix mean?
- It means “ignoring” some direct links
- Assuming conditional independence
Sparsification in General

- Replace the distribution
  \[ p(a, b, c) \]

- by an approximation \( \tilde{p} \) so that \( a \) and \( b \) are independent given \( c \)

\[
\tilde{p}(a \mid b, c) = p(a \mid c) \\
\tilde{p}(b \mid a, c) = p(b \mid c)
\]
Approximation by Assuming Conditional Independence

This leads to

\[
p(a, b, c) = p(a \mid b, c) \cdot p(b \mid c) \cdot p(c)
\]

\[
\approx p(a \mid c) \cdot p(b \mid c) \cdot p(c)
\]

\[
= p(a \mid c) \cdot \frac{p(c)}{p(c)} \cdot p(b \mid c) \cdot p(c)
\]

\[
= \frac{p(a, c) \cdot p(b, c)}{p(c)}
\]
Sparsification in SEIFs

- Goal: approximate $\Omega$ so that it is and stays sparse
- Realized by maintaining only links between the robot and a few landmarks
- This also limits the number of links between landmarks
Limit Robot-Landmark Links

- Consider a set of **active landmarks** during the updates
Active and Passive Landmarks

Active Landmarks
- A subset of all landmarks
- Includes the currently observed ones

Passive Landmarks
- All others
Sparsification Considers Three Sets of Landmarks

- Active ones that stay active
- Active ones that become passive
- Passive ones

\[ m = m^+ + m^0 + m^- \]

active \hspace{1cm} active \hspace{1cm} passive

\hspace{2cm} to passive
Sparsification

- Remove links between robot’s pose and active landmarks that become passive
- Equal to conditional independence given the other landmarks
- No change in the links of passive ones
- **Sparsification is an approximation!**

\[
p(x_t, m \mid z_{1:t}, u_{1:t}) = p(x_t, m^+, m^0, m^- \mid z_{1:t}, u_{1:t}) \]
\[\Rightarrow \ldots\]
Sparsification

- Dependencies from $z, u$ not shown:

\[
p(x_t, m) = p(x_t, m^+, m^0, m^-) \\
= p(x_t \mid m^+, m^0, m^-) \ p(m^+, m^0, m^-) \\
= p(x_t \mid m^+, m^0, m^- = 0) \ p(m^+, m^0, m^-) \\
\sim \ldots
\]

Given the active landmarks, the passive landmarks do not matter for computing the robot’s pose (so set to zero)
Sparsification

- Dependencies from $z, u$ not shown:

\[
p(x_t, m) = p(x_t, m^+, m^0, m^-) \\
= p(x_t | m^+, m^0, m^-) p(m^+, m^0, m^-) \\
= p(x_t | m^+, m^0, m^- = 0) p(m^+, m^0, m^-) \\
\approx p(x_t | m^+, m^- = 0) p(m^+, m^0, m^-)
\]

Sparsification: assume conditional independence of the robot’s pose from the landmarks that become passive (given $m^+, m^- = 0$)
Sparsification

- Dependencies from $z, u$ not shown:

\[
p(x_t, m) = p(x_t, m^+, m^0, m^-) \\
= p(x_t \mid m^+, m^0, m^-) \ p(m^+, m^0, m^-) \\
= p(x_t \mid m^+, m^0, m^- = 0) \ p(m^+, m^0, m^-) \\
\approx p(x_t \mid m^+, m^- = 0) \ p(m^+, m^0, m^-) \\
= \frac{p(x_t, m^+ \mid m^- = 0)}{p(m^+ \mid m^- = 0)} \ p(m^+, m^0, m^-) \\
= \hat{p}(x_t, m)
\]
Information Matrix Update

- Sparsifying the direct links between the robot’s pose and $m^0$ results in

$$\tilde{p}(x_t, m | z_{1:t}, u_{1:t}) \approx \frac{p(x_t, m^{+} | m^{-} = 0, z_{1:t}, u_{1:t})}{p(m^{+} | m^{-} = 0, z_{1:t}, u_{1:t})} p(m^0, m^{+}, m^{-} | z_{1:t}, u_{1:t})$$

- The sparsification replaces $\Omega, \xi$ by approximated values
- Express $\tilde{\Omega}$ as a sum of three matrices

$$\tilde{\Omega}_t = \Omega^1_t - \Omega^2_t + \Omega^3_t$$
Sparsified Information Matrix

\[ \tilde{p}(x_t, m \mid z_{1:t}, u_{1:t}) \approx \frac{p(x_t, m^+ \mid m^- = 0, z_{1:t}, u_{1:t})}{p(m^+ \mid m^- = 0, z_{1:t}, u_{1:t})} p(m^0, m^+, m^- \mid z_{1:t}, u_{1:t}) \]

- Conditioning \( \Omega_t \) on \( m^- = 0 \) yields \( \Omega_t^0 \)
- Marginalizing \( m^0 \) from \( \Omega_t^0 \) yields \( \Omega_t^1 \)
- Marginalizing \( x, m^0 \) from \( \Omega_t^0 \) yields \( \Omega_t^2 \)
- Marginalizing \( x \) from \( \Omega_t \) yields \( \Omega_t^3 \)
- Compute sparsified information matrix

\[ \tilde{\Omega}_t = \Omega_t^1 - \Omega_t^2 + \Omega_t^3 \]
The information vector can be recovered directly by:

\[
\tilde{\xi}_t = \tilde{\Omega}_t \mu_t
\]

\[
= (\Omega_t - \Omega_t + \tilde{\Omega}_t) \mu_t
\]

\[
= \Omega_t \mu_t + (\tilde{\Omega}_t - \Omega_t) \mu_t
\]

\[
= \xi_t + (\tilde{\Omega}_t - \Omega_t) \mu_t
\]
Sparsification

\textbf{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t):

1: \quad \text{define } F_{m_0}, F_{x,m_0}, F_x \text{ as projection matrices to } m_0, \{x,m_0\}, \text{ and } x, \text{ respectively}

2: \quad \Omega_t^0 = F_{x,m^+,m_0} F_{x,m^+,m_0}^T \Omega_t F_{x,m^+,m_0} F_{x,m^+,m_0}^T

3: \quad \tilde{\Omega}_t = \Omega_t - \Omega_t^0 F_{m_0} \left(F_{m_0}^T \Omega_t^0 F_{m_0}\right)^{-1} F_{m_0}^T \Omega_t^0
\quad \quad + \Omega_t^0 F_{x,m_0} \left(F_{x,m_0}^T \Omega_t^0 F_{x,m_0}\right)^{-1} F_{x,m_0}^T \Omega_t^0
\quad \quad - \Omega_t F_x \left(F_x^T \Omega_t F_x\right)^{-1} F_x^T \Omega_t

4: \quad \tilde{\xi}_t = \xi_t + (\tilde{\Omega}_t - \Omega_t) \mu_t

5: \quad \text{return } \tilde{\xi}_t, \tilde{\Omega}_t

\tilde{\Omega}_t = \Omega_t^{1} - \Omega_t^{2} + \Omega_t^{3}
Four Steps of SEIF SLAM

\[
SEIF_{\text{SLAM}}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t, z_t):
\]

1: \( \tilde{\xi}_t, \tilde{\Omega}_t, \tilde{\mu}_t = \text{SEIF\_motion\_update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t) \) \hspace{1cm} \text{DONE}

2: \( \xi_t, \Omega_t = \text{SEIF\_measurement\_update}(\tilde{\xi}_t, \tilde{\Omega}_t, \tilde{\mu}_t, z_t) \) \hspace{1cm} \text{DONE}

3: \( \mu_t = \text{SEIF\_update\_state\_estimate}(\xi_t, \Omega_t, \tilde{\mu}_t) \) \hspace{1cm} \text{DONE}

4: \( \tilde{\xi}_t, \tilde{\Omega}_t = \text{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t) \) \hspace{1cm} \text{DONE}

5: \( \text{return } \tilde{\xi}_t, \tilde{\Omega}_t, \mu_t \)
Effect of the Sparsification
SEIF SLAM vs. EKF SLAM

- Roughly **constant time** complexity vs. quadratic complexity of the EKF
- **Linear memory** complexity vs. quadratic complexity of the EKF
- SEIF SLAM is **less accurate** than EKF SLAM (sparsification, mean recovery)
SEIF & EKF: CPU Time
SEIF & EKF: Memory Usage

![Graph showing the memory usage of SEIF and EKF with respect to the number of landmarks.](image)
SEIF & EKF: Error Comparison

![Graph comparing SEIF and EKF error with number of landmarks]
Influence of the Active Features

![Graph showing the influence of active features on update time](image-url)
Influence of the Active Features

reasonable values for the number of active features
Summary on SEIF SLAM

- SEIFs are an efficient approximation of the EIF for the SLAM problem
- Neglects direct links by sparsification
- Mean computation is an approximation
- Constant time updates of the filter (for known correspondences)
- Linear memory complexity
- Inferior quality compared to EKF SLAM
Literature

Sparse Extended Information Filter