Robot Mapping

Sparse Extended Information Filter for SLAM

Cyrill Stachniss
Two Parameterizations for a 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)
- $\Omega_{ij}$ tells us the strength of a link
- Larger values for nearby features
- Most off-diagonal elements in the information are close to 0 (but $\neq 0$)
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

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

```
<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>
```

```
<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>
```
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

- One of the key aspects of SEIF SLAM to obtain efficiency

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

Passive Landmarks
- All others
Active vs. Passive Landmarks

was active, now passive

active

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. Update of the state estimate
3. Measurement update
4. Sparsification

**EIF updates:** The mean is needed to apply the motion update and for computing an expected measurement
Four Steps of SEIF SLAM

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: \(\mu_t = \text{SEIF\_update\_state\_estimate}(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t)\)
3: \(\xi_t, \Omega_t = \text{SEIF\_measurement\_update}(\bar{\xi}_t, \bar{\Omega}_t, \mu_t, z_t)\)
4: \(\tilde{\xi}_t, \tilde{\Omega}_t = \text{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t)\)
5: return \(\tilde{\xi}_t, \tilde{\Omega}_t, \mu_t\)

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: \[\mu_t = \text{SEIF\_update\_state\_estimate}(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t)\]

3: \[\bar{\xi}_t, \bar{\Omega}_t = \text{SEIF\_measurement\_update}(\bar{\xi}_t, \bar{\Omega}_t, \mu_t, z_t)\]

4: \[\tilde{\xi}_t, \tilde{\Omega}_t = \text{SEIF\_sparsification}(\bar{\xi}_t, \bar{\Omega}_t, \mu_t)\]

5: \[\text{return } \tilde{\xi}_t, \tilde{\Omega}_t, \mu_t\]
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:

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

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

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

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

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

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

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

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

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

\[
5: \quad \tilde{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + F_x^T R_t F_x
\]

\[
\text{use that as a building block for the IF update…}
\]
Algorithm `SEIF_motion_update(ξ_{t-1}, Ω_{t-1}, μ_{t-1}, u_t)`:

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

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

- Computing the information matrix
  \[ \bar{\Omega}_t = \Sigma_t^{-1} = \left[ G_t \Omega_{t-1}^{-1} G_t^T + R_t \right]^{-1} \]

- Define
  \[ \Phi_t = \left[ G_t \Omega_{t-1}^{-1} G_t^T \right]^{-1} = \left[ G_t^{-1} \right] \Omega_{t-1} G_t^{-1} \]

- Which leads to
  \[ \bar{\Omega}_t = \left[ \Phi_t^{-1} + R_t \right]^{-1} \]
Information Matrix

- We can expand the noise matrix R

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

- Apply the matrix inversion lemma

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

**3x3 matrix**
Information Matrix

- Apply the matrix inversion lemma

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

3x3 matrix

Zero except 3x3 block

Zero except 3x3 block
Information Matrix

- Apply the matrix inversion lemma

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

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

- This can be written as

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- \( \Psi_t \) 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 Computing of $\Phi_t$

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

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

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

- Compute $\Psi_t$
- Compute $\lambda_t$ based on $\Psi_t$
- Compute $\Phi_t$ based on $\lambda_t$
- Compute $\kappa_t$ based on $\Phi_t$
- Compute $\Omega_t$ based on $\kappa_t$
SEIF – Prediction Step (2/3)

SEIF\_motion\_update(ξ_{t-1}, Ω_{t-1}, μ_{t-1}, u_t):

2: \quad F_x = \cdots
3: \quad δ = \cdots
4: \quad Δ = \cdots
5: \quad Ψ_t = F^T_x [(I + Δ)^{-1} - I] F_x
6: \quad λ_t = Ψ^T_t Ω_{t-1} + Ω_{t-1} Ψ_t + Ψ^T_t Ω_{t-1} Ψ_t
7: \quad Φ_t = Ω_{t-1} + λ_t
8: \quad κ_t = Φ_t F^T_x (R_t^{-1} + F_x Φ_t F^T_x)^{-1} F_x Φ_t
9: \quad ̄Ω_t = Φ_t - κ_t

Information matrix is computed, now do the same for the information vector and the mean
Compute 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: \[ F_x = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \end{pmatrix}_{2N} \]

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

- We obtain the information vector by

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

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

- We obtain the information vector by

$$\bar{\xi}_t$$

$$= \Omega_t (\mu_{t-1} + F_x^T \delta_t)$$

$$= \Omega_t (\Omega_{t-1}^{-1} \xi_{t-1} + F_x^T \delta_t)$$

$$= \Omega_t \Omega_{t-1}^{-1} \xi_{t-1} + \Omega_t F_x^T \delta_t$$

$$= (\Omega_t - \Phi_t + \Phi_t^T - \Omega_{t-1} + \Omega_{t-1}) \Omega_{t-1}^{-1} \xi_{t-1} + \Omega_t F_x^T \delta_t$$

$$= (-\kappa_t \Omega_{t-1}^{-1} \xi_{t-1} + \Omega_{t-1} \Omega^{-1}_{t-1} \xi_{t-1} + \Omega_t F_x^T \delta_t$$

$$= -\kappa_t \lambda_t \mu_{t-1} = I$$
Compute the Information Vector

- We obtain the information vector by

\[ \bar{\xi}_t = \bar{\Omega}_t \left( \mu_{t-1} + F_x^T \delta_t \right) \]
\[ = \bar{\Omega}_t \left( \Omega_{t-1}^{-1} \xi_{t-1} + F_x^T \delta_t \right) \]
\[ = \bar{\Omega}_t \Omega_{t-1}^{-1} \xi_{t-1} + \bar{\Omega}_t F_x^T \delta_t \]
\[ = (\bar{\Omega}_t - \Phi_t + \Phi_t - \Omega_{t-1} + \Omega_{t-1}) \Omega_{t-1}^{-1} \xi_{t-1} + \bar{\Omega}_t F_x^T \delta_t \]
\[ = (\Omega_{t-1}^{-1} \xi_{t-1} + \Omega_{t-1} \Omega_{t-1}^{-1} \xi_{t-1} + \bar{\Omega}_t F_x^T \delta_t \]
\[ = \xi_{t-1} + (\lambda_t - \kappa_t) \mu_{t-1} + \bar{\Omega}_t F_x^T \delta_t \]
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: \(\Psi_t = F_x^T [(I + \Delta)^{-1} - I] F_x\)
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: \(\kappa_t = \Phi_t F_x^T (R_t^{-1} + F_x \Phi_t F_x^T)^{-1} F_x \Phi_t\)
9: \(\bar{\Omega}_t = \Phi_t - \kappa_t\)
10: \(\bar{\xi}_t = \xi_{t-1} + (\lambda_t - \kappa_t) \mu_{t-1} + \bar{\Omega}_t F_x^T \delta_t\)
11: \(\bar{\mu}_t = \mu_{t-1} + F_x^T \delta\)
12: return \(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t\)
Four Steps of SEIF SLAM

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

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

2: \( \mu_t = SEIF\_update\_state\_estimate(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t) \)

3: \( \xi_t, \Omega_t = SEIF\_measurement\_update(\bar{\xi}_t, \bar{\Omega}_t, \mu_t, z_t) \)

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

5: return \( \tilde{\xi}_t, \tilde{\Omega}_t, \mu_t \)
SEIF – Measurement (1/2)

SEIF_measurement_update(\(\bar{\xi}_t, \bar{\Omega}_t, \mu_t, \bar{z}_t\))

1: \(Q_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix}\)

2: for all observed features \(z_t^i = (r_t^i, \phi_t^i)^T\) do

3: \(j = c_t^i\) (data association)

4: if landmark \(j\) never seen before

5: \(\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \end{pmatrix} + \begin{pmatrix} r_t^i \cos(\phi_t^i + \bar{\mu}_{t,\theta}) \\ r_t^i \sin(\phi_t^i + \bar{\mu}_{t,\theta}) \end{pmatrix}\)

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: \(\hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \text{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \end{pmatrix}\)

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

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

11: endfor

12: \[ \xi_t = \tilde{\xi}_t + \sum_i H_{t}^{iT} Q_t^{-1} [z_t^i - \hat{z}_t^i + H_t^i \mu_t] \]

13: \[ \Omega_t = \tilde{\Omega}_t + \sum_i H_{t}^{iT} Q_t^{-1} H_t^i \]

14: return \( \xi_t, \Omega_t \)

**Difference to EKF (but as in EIF):**

\[
\xi_t = \tilde{\xi}_t + \sum_i H_{t}^{iT} Q_t^{-1} [z_t^i - \hat{z}_t^i + H_t^i \mu_t]
\]

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

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

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

2: \quad \mu_t = \text{SEIF\_update\_state\_estimate}(\tilde{\xi}_t, \tilde{\Omega}_t, \tilde{\mu}_t) \quad \text{DONE}

3: \quad \xi_t, \Omega_t = \text{SEIF\_measurement\_update}(\tilde{\xi}_t, \tilde{\Omega}_t, \mu_t, z_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
Sparsification

- Question: what does sparsification of the information matrix means?
Sparsification

- Question: what does sparsification of the information matrix mean?
- It means ignoring direct links between random variables (assuming a 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

\[
p(a, b, c) = p(a \mid b, c) p(b \mid c) p(c) \\
\approx p(a \mid c) p(b \mid c) p(c) \\
= p(a \mid c) \frac{p(c)}{p(c)} p(b \mid c) p(c) \\
= \frac{p(a, c) p(b, c)}{p(c)}
\]
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 \quad active \quad passive 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!**

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

\[\approx \ldots\]
Sparsification

- Dependencies from $z, u$ not shown:

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

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:

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

\textbf{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:

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

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

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

- The sparsification replaces \( \Omega, \xi \) by approximated values

- Express \( \tilde{\Omega} \) as a sum of three matrices

\[
\tilde{\Omega}_t = \Omega^1_t - \Omega^2_t + \Omega^3_t
\]
Information Vector Update

- 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 Step**

\[ \text{SEIF}\_\text{sparsification}(\xi_t, \Omega_t, \mu_t): \]

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

2: 
\[
\tilde{\Omega}_t = \Omega_t - \Omega_t^0 F_{m_0} (F_{m_0}^T \Omega_t^0 F_{m_0})^{-1} F_{m_0}^T \Omega_t^0 \\
+ \Omega_t^0 F_{x,m_0} (F_{x,m_0}^T \Omega_t^0 F_{x,m_0})^{-1} F_{x,m_0}^T \Omega_t^0 \\
- \Omega_t F_x (F_x^T \Omega_t F_x)^{-1} F_x^T \Omega_t
\]

3: 
\[ \tilde{\xi}_t = \xi_t + (\tilde{\Omega}_t - \Omega_t) \mu_t \]

4: return \( \tilde{\xi}_t, \tilde{\Omega}_t \)

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

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

1: \[\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t = \text{SEIF\_motion\_update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t)\]
\[\mu_t = \text{SEIF\_update\_state\_estimate}(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t)\] \textbf{DONE}

2: \[\bar{\xi}_t, \bar{\Omega}_t = \text{SEIF\_measurement\_update}(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t, z_t)\] \textbf{DONE}

3: \[\tilde{\xi}_t, \tilde{\Omega}_t = \text{SEIF\_sparsification}(\bar{\xi}_t, \bar{\Omega}_t, \mu_t)\] \textbf{DONE}

4: return \(\tilde{\xi}_t, \tilde{\Omega}_t, \mu_t\)
Recovering the Mean

- Computing the exact mean requires $\mu = \Omega^{-1}\xi$, which is costly!

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)
Approximation of the Mean

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

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

- Finding the mean 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

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

2: \[\mu_t = \text{SEIF\_update\_state\_estimate}(\tilde{\xi}_t, \tilde{\Omega}_t, \tilde{\mu}_t)\]

3: \[\xi_t, \Omega_t = \text{SEIF\_measurement\_update}(\tilde{\xi}_t, \tilde{\Omega}_t, \mu_t, z_t)\]

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

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

The graph shows the CPU time per iteration (in seconds) as a function of the number of landmarks. The SEIF method consistently takes less CPU time compared to the EKF method. As the number of landmarks increases, the CPU time for both methods increases, but the SEIF method maintains a lower CPU time throughout the range of landmarks considered.
SEIF & EKF: Memory Usage
SEIF & EKF: Error Comparison

![Graph showing error comparison between SEIF and EKF]
Influence of the Active Features
Influence of the Active Features

reasonable values for the number of active features
Summary in 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

- Thrun et al.: "Probabilistic Robotics", Chapter 12.1-12.7