Robot Mapping

Sparse Extended Information Filter for SLAM

Cyrill Stachniss

Two Parameterizations for a Gaussian Distribution

\[
\begin{align*}
\Sigma &= \Omega^{-1} \\
\mu &= \Omega^{-1}\xi \\
\Omega &= \Sigma^{-1} \\
\xi &= \Sigma^{-1}\mu
\end{align*}
\]

covariance matrix  
mean vector  
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
- \textbf{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

Effect of Motion Update on the Information Matrix

before 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

Effect of Motion Update on the Information Matrix

effect of the robot’s movement

Sparsification

before 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

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.

---

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 $\hat{\xi}_t$, $\hat{\Omega}_t$, $\hat{\mu}_t$ 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...

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

\[
3: \quad \bar{\mu}_t = \mu_{t-1} + F_z^T \begin{pmatrix} -\frac{\Delta t}{2\Omega_0} \sin \mu_{t-1,\theta} + \frac{\Delta t}{2\Omega_0} \sin(\mu_{t-1,\theta} + \omega_1 \Delta t) \\ \frac{\Delta t}{2\Omega_0} \cos \mu_{t-1,\theta} - \frac{\Delta t}{2\Omega_0} \cos(\mu_{t-1,\theta} + \omega_1 \Delta t) \end{pmatrix}
\]

\[
4: \quad G_t = I + F_z^T \begin{pmatrix} 0 & 0 & -\frac{\Delta t}{\omega_1} \cos \mu_{t-1,\theta} + \frac{\Delta t}{\omega_1} \cos(\mu_{t-1,\theta} + \omega_1 \Delta t) \\ 0 & 0 & \frac{\Delta t}{\omega_1} \sin \mu_{t-1,\theta} + \frac{\Delta t}{\omega_1} \sin(\mu_{t-1,\theta} + \omega_1 \Delta t) \end{pmatrix} F_z
\]

\[
5: \quad \Sigma_t = G_t \Sigma_{t-1} G_t^T + \frac{F_z^T R_z^T F_z}{R_t}
\]
SEIF – Prediction Step (1/3)

Algorithm SEIF.motion.update(\(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t\)):

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

3: \( \delta = \begin{pmatrix} -\frac{\nu_t}{\omega_t} \sin \mu_{t-1,0} + \frac{\nu_t}{\omega_t} \sin(\mu_{t-1,0} + \omega_t \Delta t) \\ \frac{\nu_t}{\omega_t} \cos \mu_{t-1,0} - \frac{\nu_t}{\omega_t} \cos(\mu_{t-1,0} + \omega_t \Delta t) \\ 0 \\ 0 \end{pmatrix} \)

4: \( \Delta = \begin{pmatrix} 0 & 0 & \frac{\nu_t}{\omega_t} \cos \mu_{t-1,0} - \frac{\nu_t}{\omega_t} \cos(\mu_{t-1,0} + \omega_t \Delta t) \\ 0 & 0 & \frac{\nu_t}{\omega_t} \sin \mu_{t-1,0} - \frac{\nu_t}{\omega_t} \sin(\mu_{t-1,0} + \omega_t \Delta t) \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{pmatrix} \)

Information Matrix

- We can expand the noise matrix \(R\)

\[
\tilde{\Omega}_t = [\Phi_t^{-1} + R_t]^{-1} = [\Phi_t^{-1} + F_x^T R_x^T F_x]^{-1}
\]

Information Matrix

- Computing the information matrix

\[
\tilde{\Omega}_t = \sum_{t=1}^{\infty} \Omega_t^{-1} = [G_t \Omega_t^{-1} G_t^T + R_t]^{-1}
\]

- Define

\[
\Phi_t = [G_t \Omega_t^{-1} G_t^T]^{-1} = [G_t^T]^{-1} \Omega_t^{-1} G_t^{-1}
\]

- Which leads to

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

Information Matrix

- Apply the matrix inversion lemma

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

3x3 matrix
**Information Matrix**

- Apply the matrix inversion lemma

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

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

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

\[
\begin{array}{c}
\text{Zero except} \\
\text{3x3 block}
\end{array}
\]

\[
\begin{array}{c}
\text{Zero except} \\
\text{3x3 block}
\end{array}
\]

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

**Information Matrix**

- This can be written as

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

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

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

**Computing** \( \Phi_t = [G_t^T]^{-1} \Omega_{t-1} G_t^{-1} \)

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

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

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

\[
\begin{array}{c}
\text{3x3 identity} \\
\text{2Nx2N identity}
\end{array}
\]
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

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

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

We have

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

with

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

\[ \Psi_t \] 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

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

  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($\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t$):

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: $\kappa_t = \Phi_t F_x^T (R_x^{-1} + F_x \Phi_t F_x^T)^{-1} F_x \Phi_t$
9: $\Omega_t = \Phi_t - \kappa_t$

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
  \[ \tilde{\mu}_t = \mu_{t-1} + F_x^T \delta \]
- Reminder (from SEIF motion update)

**Compute the Information Vector**

- We obtain the information vector by
  \[ \tilde{\xi}_t = \bar{\Omega}_t (\mu_{t-1} + F_x^T \delta_t) \]
  \[ = \bar{\Omega}_t (\Omega_{t-1}^{-1} \xi_{t-1} + F_x^T \delta_t) \]
  \[ = \bar{\Omega}_t \Omega_{t-1}^{-1} \xi_{t-1} + \bar{\Omega}_t F_x^T \delta_t \]

\[ F_x = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \frac{\Delta t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\ \frac{\Delta t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \end{pmatrix} \]
Compute the Information Vector

- We obtain the information vector by

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

SEIF – Prediction Step (3/3)

SEIF\_motion\_update(\(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t\)):
1. \( F_x = \cdot \cdot \cdot \)
2. \( \phi_t = \cdot \cdot \cdot \)
3. \( \Psi_t = F_x^T \left( (I + \Delta)^{-1} - I \right) F_x \)
4. \( \lambda_t = \Psi_t^T \Omega_{t-1} + \Omega_{t-1} \Psi_t + \Psi_t^T \Omega_{t-1} \Psi_t \)
5. \( \Phi_t = \Omega_{t-1} + \lambda_t \)
6. \( \kappa_t = \Phi_t F_x^T (\Phi_t + F_x^T \Phi_t)^{-1} F_x \phi_t \)
7. \( \tilde{\Omega}_t = \Phi_t - \kappa_t \)
8. \( \tilde{\xi}_t = \xi_{t-1} + \lambda_t - \kappa_t ) \mu_{t-1} + \tilde{\Omega}_t F_x^T \delta_t \)
9. \( \tilde{\mu}_t = \mu_{t-1} + F_x^T \delta \)
10. \( \text{return } \tilde{\xi}_t, \tilde{\Omega}_t, \tilde{\mu}_t \)

Four Steps of SEIF SLAM

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

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

identical to the EKF SLAM

Four Steps of SEIF SLAM

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

SEIF – Measurement (2/2)

\(H_t^i = \frac{1}{q} \begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & 0 & \ldots & 0 & +\sqrt{q}\delta_y & +\sqrt{q}\delta_x & 0 & 0 & \ldots \end{pmatrix}
\)
11: endfor
12: \(\xi_t = \xi_t + \sum_i H_t^iT_t^{-1} [z^i_t - \tilde{z}_t^i + H_t^i \mu_t]\)
13: \(\Omega_t = \Omega_t + \sum_i H_t^iT_t^{-1} H_t^i\)
14: return \(\xi_t, \Omega_t\)

Difference to EKF (but as in EIF):

\[
\begin{align*}
\xi_t &= \bar{\xi}_t + \sum_i H_t^iT_t^{-1} [z^i_t - \tilde{z}_t^i + H_t^i \mu_t] \\
\Omega_t &= \bar{\Omega}_t + \sum_i H_t^iT_t^{-1} H_t^i
\end{align*}
\]

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)

\[
x_{t+1} m_1 m_2 m_3 \\
x_{t+1} m_1 m_2 m_3
\]

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

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

**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!**

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

- Dependencies from $z, u$ not shown:

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

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 $m^+, m^- = 0$)

Information Matrix Update

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

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

The sparsification replaces $\Omega, \xi$ by approximated values

- Express $\hat{\Omega}$ as a sum of three matrices

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

- The information vector can be recovered directly by:

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

**Sparsification Step**

\[
\text{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t):
1: \text{define } F_{m_0}, F_{x,m_0}, F_x \text{ as projection matrices to } m_0, \{x, m_0\}, \text{ and } x, \text{ respectively}
2: \tilde{\Omega}_t = \Omega_t - \Omega^0_t F_{m_0} (F_{m_0}^T \Omega^0_t F_{m_0})^{-1} F_{m_0}^T \Omega^0_t + \Omega^0_t F_{x,m_0} (F_{x,m_0}^T \Omega^0_t F_{x,m_0})^{-1} F_{x,m_0}^T \Omega^0_t - \tilde{\Omega}_t F_x (F_x^T \tilde{\Omega}_t F_x)^{-1} F_x^T \tilde{\Omega}_t \\
3: \tilde{\xi}_t = \xi_t + (\tilde{\Omega}_t - \Omega_t) \mu_t \\
4: \text{return } \tilde{\xi}_t, \tilde{\Omega}_t
\]

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

**Four Steps of SEIF SLAM**

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

**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
  \[ \hat{\mu} = \text{argmax} \ p(\mu) \]
- Finding the mean that maximize the probability density function?

Four Steps of SEIF SLAM

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

1: \(\dot{\xi}_t, \Omega_t, \dot{\mu}_t = \text{SEIF motion update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t, z_t)\) **DONE**
2: \(\mu_t = \text{SEIF update state estimate}(\xi_t, \Omega_t, \mu_t)\) **DONE**
3: \(\xi_t, \Omega_t = \text{SEIF measurement update}(\xi_t, \Omega_t, \mu_t, z_t)\) **DONE**
4: \(\dot{\xi}_t, \Omega_t = \text{SEIF sparsification}(\xi_t, \Omega_t, \mu_t)\) **DONE**
5: return \(\dot{\xi}_t, \Omega_t, \mu_t\)

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)
Influence of the 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**