Robot Mapping

Sparse Extended Information Filter for SLAM

Cyrill Stachniss

Reminder: Parameterizations for the Gaussian Distribution

<table>
<thead>
<tr>
<th>moments</th>
<th>canonical</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Sigma = \Omega^{-1}$</td>
<td>$\Omega = \Sigma^{-1}$</td>
</tr>
<tr>
<td>$\mu = \Omega^{-1}\xi$</td>
<td>$\xi = \Sigma^{-1}\mu$</td>
</tr>
</tbody>
</table>

covariance matrix\nmean vector\ninformation matrix\ninformation vector

Motivation

Gaussian estimate \n(normalized covariance matrix \n(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

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

before sparsification

Effect of Motion Update on the Information Matrix

effect of the robot’s movement
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

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)

Active vs. Passive Landmarks

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

Note: we maintain $\xi_t, \Omega_t, \mu_t$

The corrected mean $\mu_t$ is estimated after the measurement update of the canonical parameters $\xi_t, \Omega_t$
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:

$$(R + P \ Q \ P^T)^{-1} = R^{-1} - R^{-1} \ P \ (Q^{-1} + P^T \ R^{-1} \ P)^{-1} \ P^T \ R^{-1}$$

Let us start from EKF SLAM...

SEIF SLAM – Prediction Step

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

\[
\text{EKF}_{\text{SLAM}} \text{ Prediction}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, R_t):
\]

1. \[ F_x = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \end{pmatrix} \]
2. \[ \tilde{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} -\frac{\Delta t}{\omega} \sin \mu_{t-1,\theta} + \frac{\Delta t}{\omega} \sin \left( \mu_{t-1,\theta} + \omega_1 \Delta t \right) \\ \frac{\Delta t}{\omega} \cos \mu_{t-1,\theta} - \frac{\Delta t}{\omega} \cos \left( \mu_{t-1,\theta} + \omega_1 \Delta t \right) \end{pmatrix} \]
3. \[ \Delta = \begin{pmatrix} 0 & 0 & \Delta t \\ 0 & 0 & \frac{\Delta t}{\omega} \sin \mu_{t-1,\theta} - \frac{\Delta t}{\omega} \cos \left( \mu_{t-1,\theta} + \omega_1 \Delta t \right) \end{pmatrix} \]

let's begin with computing the information matrix...

SEIF – Prediction Step (1/3)

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

1. \[ F_x = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \end{pmatrix} \]
2. \[ \tilde{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} -\frac{\Delta t}{\omega} \sin \mu_{t-1,\theta} + \frac{\Delta t}{\omega} \sin \left( \mu_{t-1,\theta} + \omega_1 \Delta t \right) \\ \frac{\Delta t}{\omega} \cos \mu_{t-1,\theta} - \frac{\Delta t}{\omega} \cos \left( \mu_{t-1,\theta} + \omega_1 \Delta t \right) \end{pmatrix} \]
3. \[ \delta = \begin{pmatrix} -\frac{\Delta t}{\omega} \sin \mu_{t-1,\theta} + \frac{\Delta t}{\omega} \sin \left( \mu_{t-1,\theta} + \omega_1 \Delta t \right) \\ \frac{\Delta t}{\omega} \cos \mu_{t-1,\theta} - \frac{\Delta t}{\omega} \cos \left( \mu_{t-1,\theta} + \omega_1 \Delta t \right) \end{pmatrix} \]
4. \[ \Delta = \begin{pmatrix} 0 & 0 & \Delta t \\ 0 & 0 & \frac{\Delta t}{\omega} \sin \mu_{t-1,\theta} - \frac{\Delta t}{\omega} \cos \left( \mu_{t-1,\theta} + \omega_1 \Delta t \right) \end{pmatrix} \]

Compute the Information Matrix

- Computing the information matrix

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

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

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

Compute the Information Matrix

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

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

- Apply the matrix inversion lemma

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

3x3 matrix

- Zero except 3x3 block
- Zero except 3x3 block

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

Compute the Information Matrix

- Apply the matrix inversion lemma

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

3x3 matrix

- Zero except 3x3 block
- Zero except 3x3 block

This can be written as

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

\[= \Phi_t - \kappa_t \]

Question: Can we compute \( \Phi_t \) efficiently \( \left( \Phi_t = [G_t^T]^{-1} \Omega_{t-1} G_t^{-1} \right) \)?
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: $3\times 3$ matrix
Computing $\Phi_t = [G_t^T]^{-1} \Omega_{t-1} G_t^{-1}$

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

\[
G_t^{-1} = (I + F_x^T \Delta F_x)^{-1} \\
\quad = \left( \begin{array}{cc}
\Delta + I_3 & 0 \\
0 & I_{2N}
\end{array} \right)^{-1} \\
\quad = \left( \begin{array}{cc}
(\Delta + I_3)^{-1} & 0 \\
0 & I_{2N}
\end{array} \right) \\
\quad = I_{3+2N} + \left( \begin{array}{cc}
(\Delta + I_3)^{-1} - I_3 & 0 \\
0 & 0
\end{array} \right) \\
\quad = I + F_x^T [(I + \Delta)^{-1} - I] F_x \\
\quad = I + \Psi_t
\]

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

Computing $\Phi_t = [G_t^T]^{-1} \Omega_{t-1} G_t^{-1}$

- We have

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

- with

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

\underline{3x3 matrix}

- $\Psi_t$ is zero except of a 3x3 block
- $G_t^{-1}$ is an identity except of a 3x3 block

Constant Time Computation of $\Phi_t$

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

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

\underline{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 $\Omega_t$ 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: $\Psi_t = F_x^T [(I + \Delta)^{-1} - I] F_x$
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: $\Omega_t = \Phi_t - \kappa_t$

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: $F_x = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \cdots & \frac{1}{2N} \end{pmatrix}_{2N}$
3: $\delta = \begin{pmatrix} -\frac{\mu_{t-1, x}}{\omega_t} \sin(\mu_{t-1, \theta} + \omega_t \Delta t) \\ \frac{\mu_{t-1, x}}{\omega_t} \cos(\mu_{t-1, \theta} + \omega_t \Delta t) - \frac{\mu_{t-1, x}}{\omega_t} \cos(\mu_{t-1, \theta} + \omega_t \Delta t) \\ \omega_t \Delta t \\ \omega_t \Delta t \\ \cdots \\ \omega_t \Delta t \\ \cdots \\ \omega_t \Delta t \\ \cdots \end{pmatrix}$

**Compute the Information Vector**

- We obtain the information vector by

$$\tilde{\xi}_t = \begin{pmatrix} \mu_{t-1} + F_x^T \delta_t \\ \Omega_t (\Omega_{t-1}^{-1} \xi_{t-1} + F_x^T \delta_t) \end{pmatrix}$$
Compute the Information Vector

- We obtain the information vector by

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

53

Compute the Information Vector

- We obtain the information vector by

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

54

Compute the Information Vector

- We obtain the information vector by

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

55

Compute the Information Vector

- We obtain the information vector by

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

56
SEIF – Prediction Step (3/3)

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

1: \( F_x = \ldots \)
2: \( \delta = \ldots \)
3: \( \Delta = \ldots \)
4: \( \Psi_t = F_x^T [(I + \Delta)^{-1} - I] F_x \)
5: \( \delta_t = \Psi_t^T \Omega_{t-1} + \Omega_t \Psi_t + \Psi_t^T \Omega_{t-1} \Psi_t \)
6: \( \Phi_t = \Omega_{t-1} + \delta_t \)
7: \( \kappa_t = \Phi_t F_x^T (R_t^{-1} + F_x \Phi_t F_x^T)^{-1} F_x \Phi_t \)
8: \( \Omega_t = \Phi_t - \kappa_t \)
9: \( \xi_t = \xi_{t-1} + (\lambda_t - \kappa_t) \mu_{t-1} + \bar{\xi}_t \)
10: \( \mu_t = \mu_{t-1} + \bar{\mu}_t \)
11: return \( \xi_t, \Omega_t, \mu_t \)

SEIF – Measurement (1/2)

\[
\text{SEIF}_\text{measurement_update}(\xi_t, \Omega_t, \mu_t, z_t)
\]

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

identical to the EKF SLAM

SEIF – Measurement (2/2)

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

1: \( \xi_t, \Omega_t, \mu_t = \text{SEIF}_\text{motion_update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t, z_t) \)
2: \( \xi_t, \Omega_t = \text{SEIF}_\text{measurement_update}(\xi_t, \Omega_t, \mu_t, z_t) \)
3: \( \mu_t = \text{SEIF}_\text{update_state_estimate}(\xi_t, \Omega_t, \mu_t) \)
4: \( \xi_t, \Omega_t = \text{SEIF}_\text{sparsification}(\xi_t, \Omega_t, \mu_t) \)
5: return \( \xi_t, \Omega_t, \mu_t \)

Four Steps of SEIF SLAM

1: \( \xi_t, \Omega_t, \mu_t = \text{SEIF}_\text{motion_update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t, z_t) \)
2: \( \xi_t, \Omega_t = \text{SEIF}_\text{measurement_update}(\xi_t, \Omega_t, \mu_t, z_t) \)
3: \( \mu_t = \text{SEIF}_\text{update_state_estimate}(\xi_t, \Omega_t, \mu_t) \)
4: \( \xi_t, \Omega_t = \text{SEIF}_\text{sparsification}(\xi_t, \Omega_t, \mu_t) \)
5: return \( \xi_t, \Omega_t, \mu_t \)

Difference to EKF (but as in EIF):

\[
\begin{align*}
\xi_t &= \bar{\xi}_t + \sum_i H_t^T Q_t^{-1} [z^i_t - \tilde{z}^i_t + H^i_t \mu_t] \\
\Omega_t &= \bar{\Omega}_t + \sum_i H_t^T Q_t^{-1} H_t^i
\end{align*}
\]
Four Steps of SEIF SLAM

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

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

2: \(\xi_t, \Omega_t = \text{SEIF\_measurement\_update}(\xi_t, \Omega_t, \mu_t, z_t)\) \text{ DONE}

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

4: \(\xi_t, \Omega_t = \text{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t)\)

5: \text{return } \xi_t, \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

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

2-10:.....

11: \(\bar{\mu}_t = \mu_{t-1} + F^T \delta\)

12: \text{return } \xi_t, \Omega_t, \bar{\mu}_t

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

**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?

**Four Steps of SEIF SLAM**

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

1: \(\xi_t, \Omega_t, \hat{\mu}_t = \text{SEIF motion update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, z_t)\) **DONE**
2: \(\xi_t, \Omega_t = \text{SEIF measurement update}(\xi_t, \Omega_t, \mu_t, z_t)\) **DONE**
3: \(\mu_t = \text{SEIF update state estimate}(\xi_t, \Omega_t, \hat{\mu}_t)\) **DONE**
4: \(\xi_t, \Omega_t = \text{SEIF sparsification}(\xi_t, \Omega_t, \mu_t)\)
5: return \(\xi_t, \Omega_t, \mu_t\)
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
  \[ p(a, b, c) = p(a \mid b, c) p(b \mid c) p(c) \]
  \[ \simeq p(a \mid c) p(b \mid c) p(c) \]
  \[ = p(a \mid c) \frac{p(c)}{p(c)} p(b \mid c) p(c) \]
  \[ = \frac{p(a, c) p(b, c)}{p(c)} \]

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 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:

$$p(x_t, m) = p(x_t, m^+, m^0, m^-)$$
$$= p(x_t \mid m^+, m^0, m^-) p(m^+, m^0, m^-)$$
$$= p(x_t \mid m^+, m^0, m^- = 0) p(m^+, m^0, m^-)$$
$$\approx \ldots$$

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:

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

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

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

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

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

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

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

- 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

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

**Information Vector Update**

- The information vector can be recovered directly by:

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

**Four Steps of SEIF SLAM**

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

\[ \tilde{\Omega}_t = \Omega_t^1 - \Omega_t^2 + \Omega_t^3 \]
**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**
Influence of the Active Features

- 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