Robot Mapping

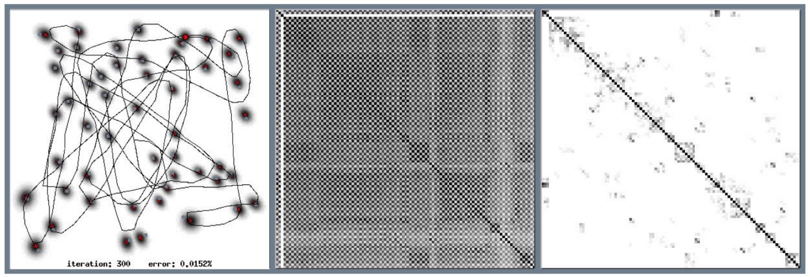
Sparse Extended Information Filter for SLAM

Cyrill Stachniss

Reminder: Parameterizations for the Gaussian Distribution

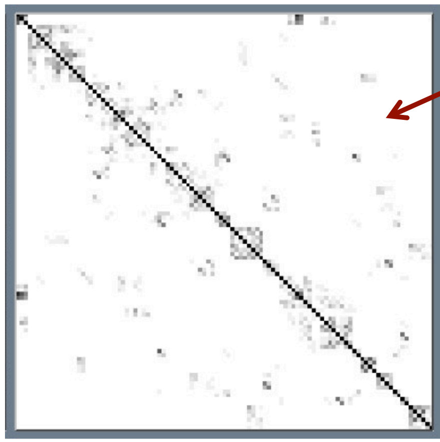
moments	canonical
$\Sigma = \Omega^{-1}$	$\Omega = \Sigma^{-1}$
$\mu = \Omega^{-1}\xi$	$\xi = \Sigma^{-1}\mu$
covariance matrix mean vector	information matrix information vector

Motivation



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

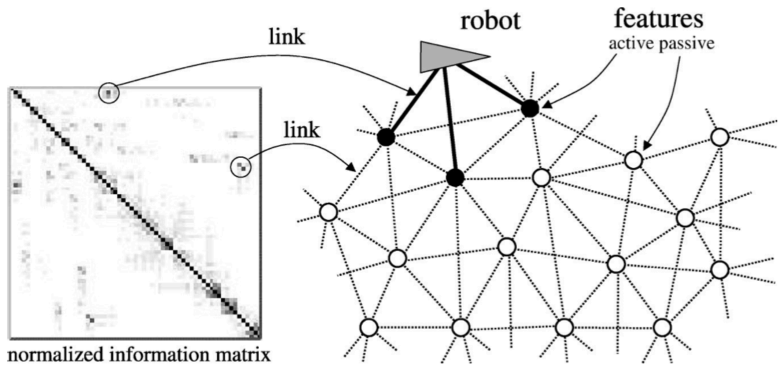
Motivation



small but non-zero

normalized information matrix

Most Features Have Only a Small Number of **Strong** Links



Information Matrix

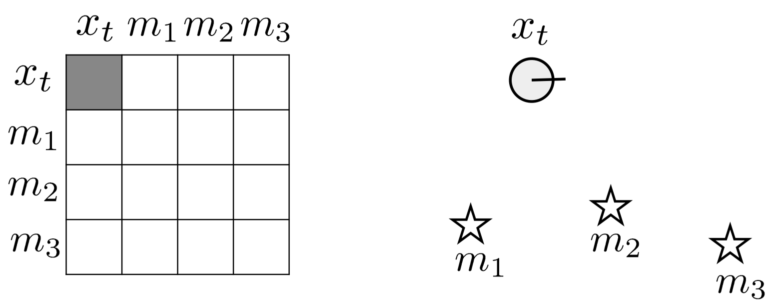
- Information matrix can be interpreted as a graph of constraints/links between nodes (variables)
- Can be interpreted as a MRF
- Missing links indicate conditional independence of the random variables
- Ω_{ij} tells us the strength of a link
- Larger values for nearby features
- Most off-diagonal elements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

Create Sparsity

- "Set" most links to zero/avoid fill-in
- Exploit sparseness of Ω in the computations

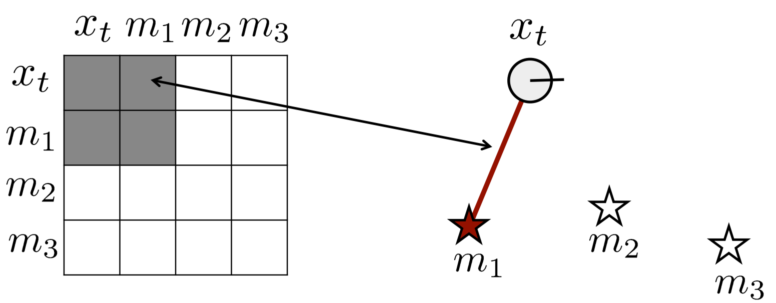
- **sparse** = finite number of non-zero off-diagonals, independent of the matrix size

Effect of **Measurement Update** on the Information Matrix



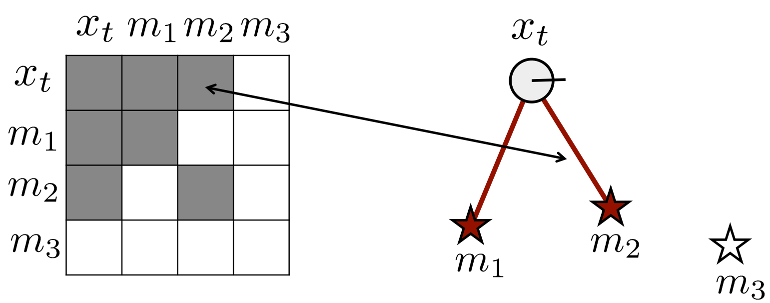
before any observations

Effect of Measurement Update on the Information Matrix



robot observes landmark 1

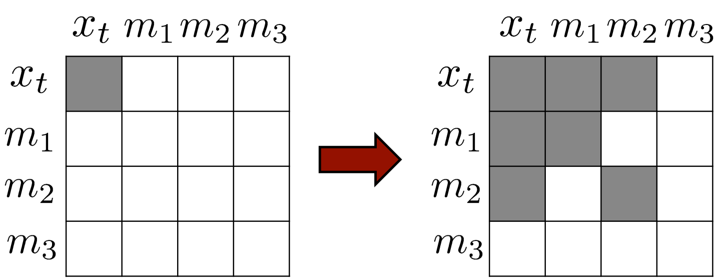
Effect of Measurement Update on the Information Matrix



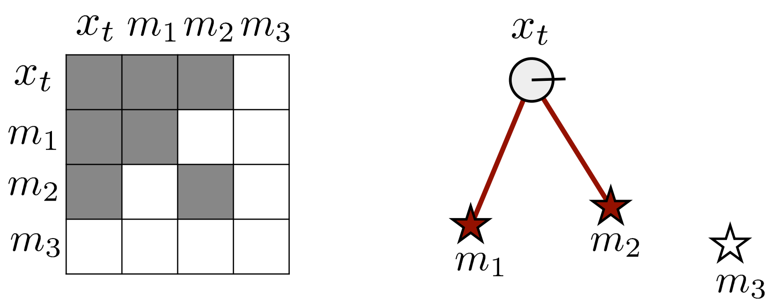
robot observes landmark 2

Effect of Measurement Update on the Information Matrix

- Adds information between the robot’s pose and the observed feature

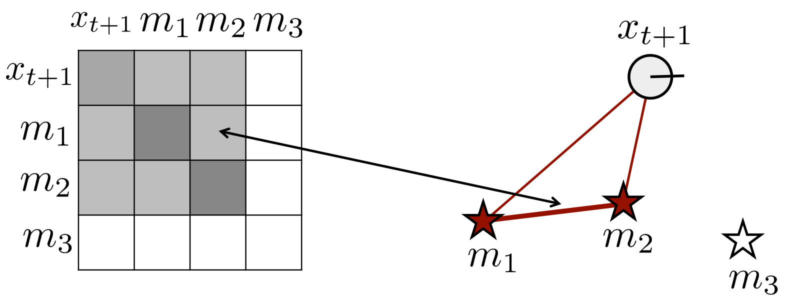


Effect of Motion Update on the Information Matrix



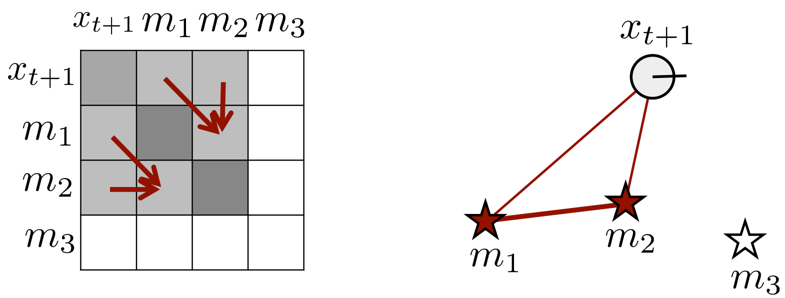
before the robot’s movement

Effect of Motion Update on the Information Matrix



after the robot's movement

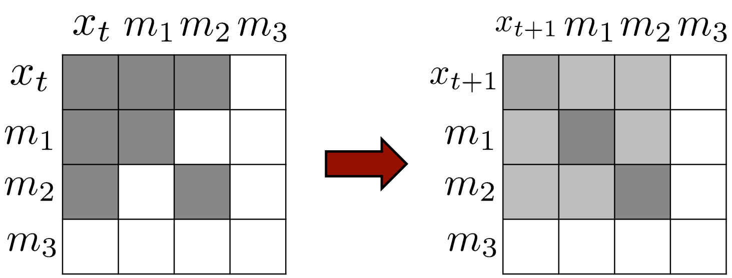
Effect of Motion Update on the Information Matrix



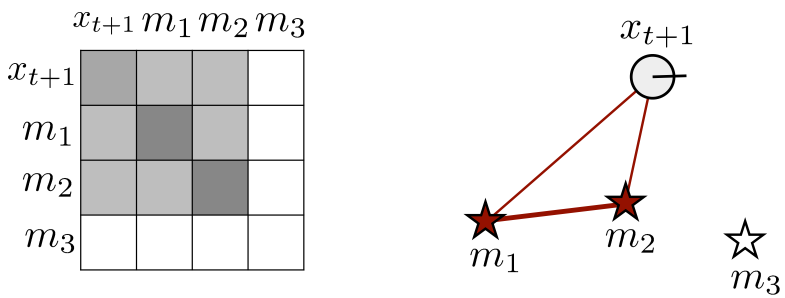
effect of the robot's movement

Effect of Motion Update on the Information Matrix

- Weakens the links between the robot's pose and the landmarks
- Add links between landmarks

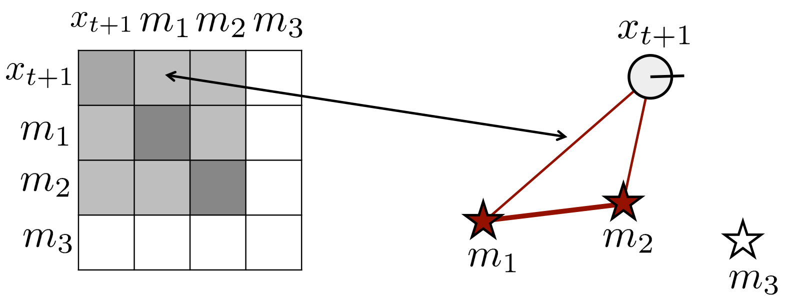


Sparsification



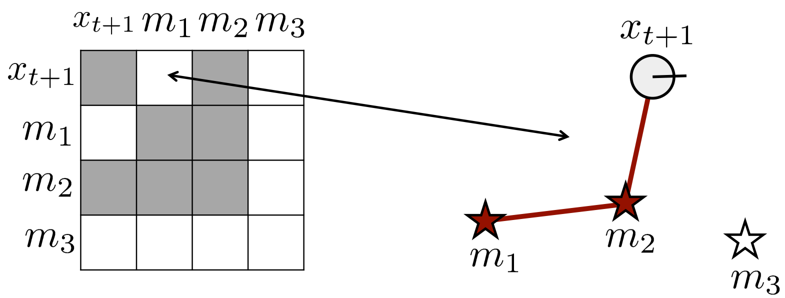
before sparsification

Sparsification



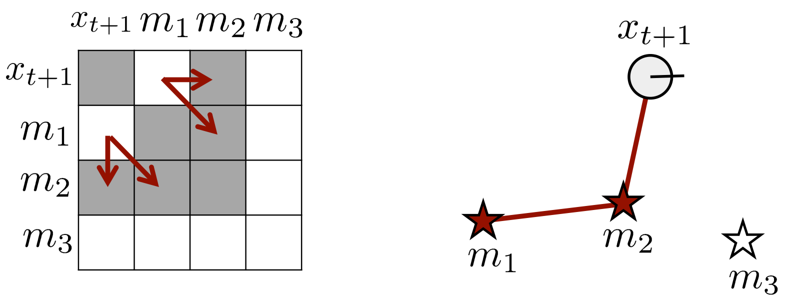
before sparsification

Sparsification



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

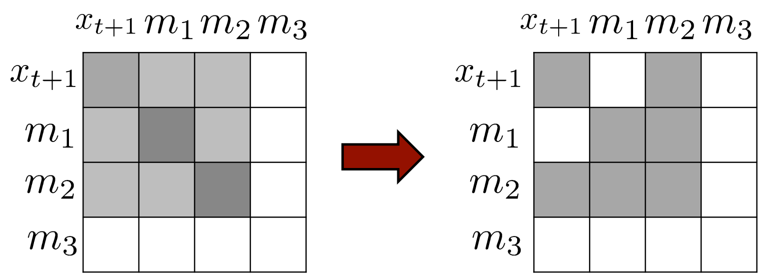
Sparsification



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

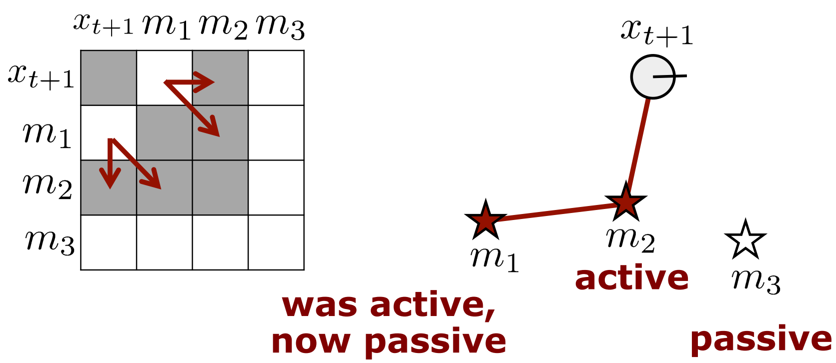
Active Landmarks

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

Passive Landmarks

- All others

Active vs. Passive Landmarks



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

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

$\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t = \textbf{SEIF\_motion\_update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t)$
$\xi_t, \Omega_t = \textbf{SEIF\_measurement\_update}(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t, z_t)$
$\mu_t = \textbf{SEIF\_update\_state\_estimate}(\xi_t, \Omega_t, \bar{\mu}_t)$
$\tilde{\xi}_t, \tilde{\Omega}_t = \textbf{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t)$
return $\tilde{\xi}_t, \tilde{\Omega}_t, \mu_t$

Note: we maintain ξ_t, Ω_t, μ_t

Four Steps of SEIF SLAM

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

$\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t = \textbf{SEIF\_motion\_update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t)$
$\xi_t, \Omega_t = \textbf{SEIF\_measurement\_update}(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t, z_t)$
$\mu_t = \textbf{SEIF\_update\_state\_estimate}(\xi_t, \Omega_t, \bar{\mu}_t)$
$\tilde{\xi}_t, \tilde{\Omega}_t = \textbf{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t)$
return $\tilde{\xi}_t, \tilde{\Omega}_t, \mu_t$

The corrected mean μ_t is estimated after the measurement update of the canonical parameters ξ_t, Ω_t

Four Steps of SEIF SLAM

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

$\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t = \textbf{SEIF\_motion\_update}(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t)$
$\xi_t, \Omega_t = \textbf{SEIF\_measurement\_update}(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t, z_t)$
$\mu_t = \textbf{SEIF\_update\_state\_estimate}(\xi_t, \Omega_t, \bar{\mu}_t)$
$\tilde{\xi}_t, \tilde{\Omega}_t = \textbf{SEIF\_sparsification}(\xi_t, \Omega_t, \mu_t)$
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 $\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t$ 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…

EKF_SLAM_Prediction($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, R_t$):

2: $F_x = \begin{pmatrix} 1 & 0 & 0 & 0\cdots0 \\ 0 & 1 & 0 & 0\cdots0 \\ 0 & 0 & 1 & 0\cdots0 \end{pmatrix}$

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: $\bar{\Sigma}_t = G_t\,\Sigma_{t-1}\,G_t^T + \underbrace{F_x^T\,R_t^x\,F_x}_{R_t}$

Let us start from EKF SLAM…

EKF_SLAM_Prediction($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, R_t$):

2: $F_x = \begin{pmatrix} 1 & 0 & 0 & 0\cdots0 \\ 0 & 1 & 0 & 0\cdots0 \\ 0 & 0 & 1 & 0\cdots0 \end{pmatrix}$ copy & paste

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

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

5: $\bar{\Sigma}_t = G_t\,\Sigma_{t-1}\,G_t^T + \underbrace{F_x^T\,R_t^x\,F_x}_{R_t}$

Let us start from EKF SLAM...

EKF_SLAM_Prediction($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, R_t$):

2: $F_x = \begin{pmatrix} 1 & 0 & 0 & 0\cdots 0 \\ 0 & 1 & 0 & 0\cdots 0 \\ 0 & 0 & 1 & 0\cdots 0 \end{pmatrix}$ copy & paste

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

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

5: $\bar{\Sigma}_t = G_t\,\Sigma_{t-1}\,G_t^T + \underbrace{F_x^T\,R_t^x\,F_x}_{R_t}$

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

SEIF – Prediction Step (1/3)

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

2: $F_x = \begin{pmatrix} 1 & 0 & 0 & 0\cdots 0 \\ 0 & 1 & 0 & 0\cdots 0 \\ 0 & 0 & 1 & \underbrace{0\cdots 0}_{2N} \end{pmatrix}$

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

Compute the Information Matrix

- Computing the information matrix

$$\begin{aligned} \bar{\Omega}_t &= \bar{\Sigma}_t^{-1} \\ &= \left[G_t\,\Omega_{t-1}^{-1}\,G_t^T + R_t\right]^{-1} \\ &= \left[\Phi_t^{-1} + R_t\right]^{-1} \end{aligned}$$

- with the term Φ_t defined as

$$\begin{aligned} \Phi_t &= \left[G_t\,\Omega_{t-1}^{-1}\,G_t^T\right]^{-1} \\ &= [G_t^T]^{-1}\,\Omega_{t-1}\,G_t^{-1} \end{aligned}$$

Compute the Information Matrix

- We can expand the noise matrix R

$$\begin{aligned} \bar{\Omega}_t &= \left[\Phi_t^{-1} + R_t\right]^{-1} \\ &= \left[\Phi_t^{-1} + F_x^T\,R_t^x\,F_x\right]^{-1} \end{aligned}$$

Compute the Information Matrix

- Apply the matrix inversion lemma

$$
\begin{aligned}
\bar{\Omega}_t &= \left[\Phi_t^{-1} + R_t\right]^{-1} \\
&= \left[\Phi_t^{-1} + F_x^T\ R_t^x\ F_x\right]^{-1} \\
&= \Phi_t - \Phi_t\ F_x^T \underbrace{(R_t^{x-1} + F_x\ \Phi_t\ F_x^T)^{-1}}_{\textbf{3x3 matrix}} F_x\ \Phi_t
\end{aligned}
$$

Compute the Information Matrix

- Apply the matrix inversion lemma

$$
\begin{aligned}
\bar{\Omega}_t &= \left[\Phi_t^{-1} + R_t\right]^{-1} \\
&= \left[\Phi_t^{-1} + F_x^T\ R_t^x\ F_x\right]^{-1} \\
&= \Phi_t - \Phi_t\ \underset{\text{Zero except 3x3 block}}{F_x^T} \underbrace{(R_t^{x-1} + F_x\ \Phi_t\ F_x^T)^{-1}}_{\textbf{3x3 matrix}} \underset{\text{Zero except 3x3 block}}{F_x}\ \Phi_t
\end{aligned}
$$

Compute the Information Matrix

- Apply the matrix inversion lemma

$$
\begin{aligned}
\bar{\Omega}_t &= \left[\Phi_t^{-1} + R_t\right]^{-1} \\
&= \left[\Phi_t^{-1} + F_x^T\ R_t^x\ F_x\right]^{-1} \\
&= \Phi_t - \Phi_t\ \underset{\text{Zero except 3x3 block}}{F_x^T} \underbrace{(R_t^{x-1} + F_x\ \Phi_t\ F_x^T)^{-1}}_{\text{3x3 matrix}} \underset{\text{Zero except 3x3 block}}{F_x}\ \Phi_t
\end{aligned}
$$

- **Constant complexity if Φ_t is sparse!**

Compute the Information Matrix

- This can be written as

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

- Question: Can we compute Φ_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 Ω_{t-1} is sparse

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

- Goal: constant time if Ω_{t-1} is sparse

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

3x3 identity

2Nx2N identity

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

- Goal: constant time if Ω_{t-1} is sparse

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

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 Ω_{t-1} is sparse

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

Note: 3x3 matrix

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

- Goal: constant time if Ω_{t-1} is sparse

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

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

- We have

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

- with

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

3x3 matrix

- Ψ_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 Φ_t

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

$$\begin{aligned} \Phi_t &= [G_t^T]^{-1}\, \Omega_{t-1}\, G_t^{-1} \\ &= (I + \Psi_t^T)\, \Omega_{t-1}\, (I + \Psi_t) \\ &= \Omega_{t-1} + \underbrace{\Psi_t^T\, \Omega_{t-1} + \Omega_{t-1}\, \Psi_t + \Psi_t^T\, \Omega_{t-1}\, \Psi_t}_{\lambda_t} \\ &= \Omega_{t-1} + \underline{\lambda_t} \end{aligned}$$

all elements zero except a constant number of entries

Prediction Step in Brief

- Compute Ψ_t
- Compute λ_t using Ψ_t
- Compute Φ_t using λ_t
- Compute κ_t using Φ_t
- Compute $\bar{\Omega}_t$ using Φ_t and κ_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: $\bar{\Omega}_t = \Phi_t - \kappa_t$

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

$$\bar{\mu}_t \quad = \quad \mu_{t-1} + F_x^T \, \delta$$

- Reminder (from SEIF motion update)

$$2: \quad F_x = \begin{pmatrix} 1 & 0 & 0 & 0 \cdots 0 \\ 0 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 1 & \underbrace{0 \cdots 0}_{2N} \end{pmatrix}$$

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

Compute the Information Vector

- We obtain the information vector by

$$\begin{aligned} \bar{\xi}_t & \\ &= \bar{\Omega}_t \, (\mu_{t-1} + F_x^T \, \delta_t) \\ &= \bar{\Omega}_t \, (\Omega_{t-1}^{-1} \, \xi_{t-1} + F_x^T \, \delta_t) \end{aligned}$$

Compute the Information Vector

- We obtain the information vector by

$$
\begin{aligned}
&\bar{\xi}_t \\
&= \bar{\Omega}_t \, (\mu_{t-1} + F_x^T \, \delta_t) \\
&= \bar{\Omega}_t \, (\Omega_{t-1}^{-1} \, \xi_{t-1} + F_x^T \, \delta_t) \\
&= \bar{\Omega}_t \, \Omega_{t-1}^{-1} \, \xi_{t-1} + \bar{\Omega}_t \, F_x^T \, \delta_t
\end{aligned}
$$

Compute the Information Vector

- We obtain the information vector by

$$
\begin{aligned}
&\bar{\xi}_t \\
&= \bar{\Omega}_t \, (\mu_{t-1} + F_x^T \, \delta_t) \\
&= \bar{\Omega}_t \, (\Omega_{t-1}^{-1} \, \xi_{t-1} + F_x^T \, \delta_t) \\
&= \bar{\Omega}_t \, \Omega_{t-1}^{-1} \, \xi_{t-1} + \bar{\Omega}_t \, F_x^T \, \delta_t \\
&= (\bar{\Omega}_t \underbrace{-\Phi_t + \Phi_t}_{=0} \underbrace{-\Omega_{t-1} + \Omega_{t-1}}_{=0}) \, \Omega_{t-1}^{-1} \, \xi_{t-1} + \bar{\Omega}_t \, F_x^T \, \delta_t
\end{aligned}
$$

Compute the Information Vector

- We obtain the information vector by

$$
\begin{aligned}
&\bar{\xi}_t \\
&= \bar{\Omega}_t \, (\mu_{t-1} + F_x^T \, \delta_t) \\
&= \bar{\Omega}_t \, (\Omega_{t-1}^{-1} \, \xi_{t-1} + F_x^T \, \delta_t) \\
&= \bar{\Omega}_t \, \Omega_{t-1}^{-1} \, \xi_{t-1} + \bar{\Omega}_t \, F_x^T \, \delta_t \\
&= (\bar{\Omega}_t \underbrace{-\Phi_t + \Phi_t}_{=0} \underbrace{-\Omega_{t-1} + \Omega_{t-1}}_{=0}) \, \Omega_{t-1}^{-1} \, \xi_{t-1} + \bar{\Omega}_t \, F_x^T \, \delta_t \\
&= (\underbrace{\bar{\Omega}_t - \Phi_t}_{= -\kappa_t} + \underbrace{\Phi_t - \Omega_{t-1}}_{= \lambda_t}) \, \underbrace{\Omega_{t-1}^{-1} \, \xi_{t-1}}_{= \mu_{t-1}} + \underbrace{\Omega_{t-1} \, \Omega_{t-1}^{-1}}_{= I} \, \xi_{t-1} + \bar{\Omega}_t \, F_x^T \, \delta_t
\end{aligned}
$$

Compute the Information Vector

- We obtain the information vector by

$$
\begin{aligned}
&\bar{\xi}_t \\
&= \bar{\Omega}_t \, (\mu_{t-1} + F_x^T \, \delta_t) \\
&= \bar{\Omega}_t \, (\Omega_{t-1}^{-1} \, \xi_{t-1} + F_x^T \, \delta_t) \\
&= \bar{\Omega}_t \, \Omega_{t-1}^{-1} \, \xi_{t-1} + \bar{\Omega}_t \, F_x^T \, \delta_t \\
&= (\bar{\Omega}_t \underbrace{-\Phi_t + \Phi_t}_{=0} \underbrace{-\Omega_{t-1} + \Omega_{t-1}}_{=0}) \, \Omega_{t-1}^{-1} \, \xi_{t-1} + \bar{\Omega}_t \, F_x^T \, \delta_t \\
&= (\underbrace{\bar{\Omega}_t - \Phi_t}_{= -\kappa_t} + \underbrace{\Phi_t - \Omega_{t-1}}_{= \lambda_t}) \, \underbrace{\Omega_{t-1}^{-1} \, \xi_{t-1}}_{= \mu_{t-1}} + \underbrace{\Omega_{t-1} \, \Omega_{t-1}^{-1}}_{= I} \, \xi_{t-1} + \bar{\Omega}_t \, F_x^T \, \delta_t \\
&= \xi_{t-1} + (\lambda_t - \kappa_t) \, \mu_{t-1} + \bar{\Omega}_t \, F_x^T \, \delta_t
\end{aligned}
$$

SEIF – Prediction Step (3/3)

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

$F_x = \cdots$

$\delta = \cdots$

$\Delta = \cdots$

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

$\lambda_t = \Psi_t^T \, \Omega_{t-1} + \Omega_{t-1} \, \Psi_t + \Psi_t^T \, \Omega_{t-1} \, \Psi_t$

$\Phi_t = \Omega_{t-1} + \lambda_t$

$\kappa_t = \Phi_t \, F_x^T (R_t^{-1} + F_x \, \Phi_t \, F_x^T)^{-1} \, F_x \, \Phi_t$

$\bar{\Omega}_t = \Phi_t - \kappa_t$

$\bar{\xi}_t = \xi_{t-1} + (\lambda_t - \kappa_t) \, \mu_{t-1} + \bar{\Omega}_t \, F_x^T \, \delta_t$

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

return $\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t$

Four Steps of SEIF SLAM

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

$\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t =$ **SEIF_motion_update**$(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t)$ DONE

$\xi_t, \Omega_t =$ **SEIF_measurement_update**$(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t, z_t)$

$\mu_t =$ **SEIF_update_state_estimate**$(\xi_t, \Omega_t, \bar{\mu}_t)$

$\tilde{\xi}_t, \tilde{\Omega}_t =$ **SEIF_sparsification**(ξ_t, Ω_t, μ_t)

return $\tilde{\xi}_t, \tilde{\Omega}_t, \mu_t$

SEIF – Measurement (1/2)

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

$Q_t = \begin{pmatrix} \sigma_r{}^2 & 0 \\ 0 & \sigma_\phi{}^2 \end{pmatrix}$

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

$\quad j = c_t^i$ (data association)

$\quad$ *if landmark j never seen before*

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

$\quad$ *endif*

$\quad \delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix}$

$\quad q = \delta^T \delta$

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

identical to the EKF SLAM

SEIF – Measurement (2/2)

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

endfor

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

$\Omega_t = \bar{\Omega}_t + \sum_i H_t^{iT} \, Q_t^{-1} \, H_t^i$

return ξ_t, Ω_t

Difference to EKF (but as in EIF):

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

$$\Omega_t = \bar{\Omega}_t + \sum_i H_t^{iT} \, Q_t^{-1} \, H_t^i$$

Four Steps of SEIF SLAM

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

$\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t =$ **SEIF_motion_update**$(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t)$ **DONE**

$\xi_t, \Omega_t =$ **SEIF_measurement_update**$(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t, z_t)$ **DONE**

$\mu_t =$ **SEIF_update_state_estimate**$(\xi_t, \Omega_t, \bar{\mu}_t)$

$\tilde{\xi}_t, \tilde{\Omega}_t =$ **SEIF_sparsification**(ξ_t, Ω_t, μ_t)

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

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

....

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

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

$$\begin{aligned}\hat{\mu} &= \operatorname{argmax} p(\mu) \\ &= \underset{\mu}{\operatorname{argmax}} \exp\left(-\frac{1}{2}\mu^T\Omega\mu + \xi^T\mu\right)\end{aligned}$$

- 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 μ are needed (robot's pose and active landmarks)

no further details here…

Four Steps of SEIF SLAM

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

$\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t =$ ~~**SEIF_motion_update**$(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t)$~~ DONE

$\xi_t, \Omega_t =$ ~~**SEIF_measurement_update**$(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t, z_t)$~~ DONE

$\mu_t =$ ~~**SEIF_update_state_estimate**$(\xi_t, \Omega_t, \bar{\mu}_t)$~~ DONE

$\tilde{\xi}_t, \tilde{\Omega}_t =$ **SEIF_sparsification**(ξ_t, Ω_t, μ_t)

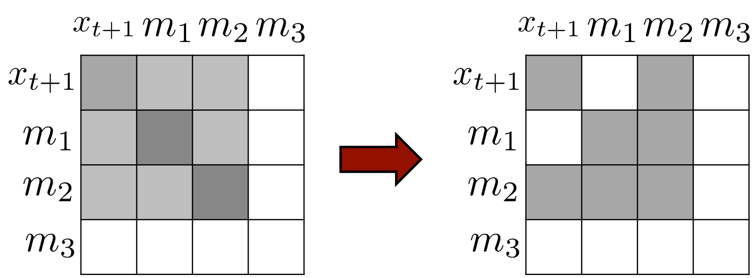
return $\tilde{\xi}_t, \tilde{\Omega}_t, \mu_t$

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

$$\begin{aligned} p(a, b, c) &= p(a \mid b, c)\, p(b \mid c)\, p(c) \\ &\simeq p(a \mid c)\, p(b \mid c)\, p(c) \\ &= p(a \mid c)\, \frac{p(c)}{p(c)}\, p(b \mid c)\, p(c) \\ &= \frac{p(a, c)\, p(b, c)}{p(c)} \end{aligned}$$

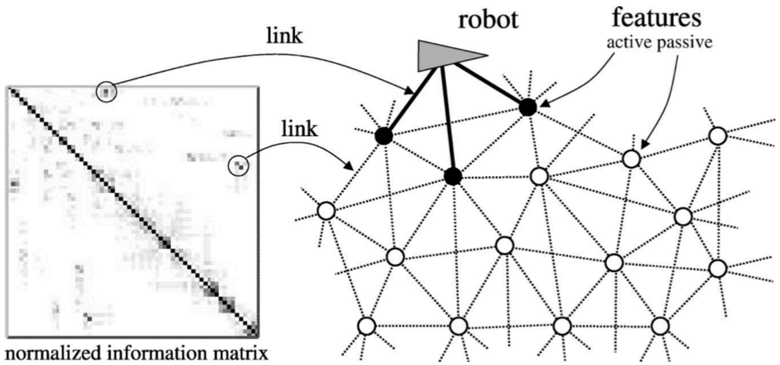
approximation

Sparsification in SEIFs

- Goal: approximate Ω 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

$$m = \underbrace{m^+}_{\text{active}} + \underbrace{m^0}_{\text{active to passive}} + \underbrace{m^-}_{\text{passive}}$$

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!**

$$\begin{aligned} p(x_t, m \mid z_{1:t}, u_{1:t}) &= p(x_t, m^+, m^0, m^- \mid z_{1:t}, u_{1:t}) \\ &\simeq \ldots \end{aligned}$$

Sparsification

- Dependencies from z, u not shown:

$$
\begin{aligned}
p(x_t, m) &= p(x_t, m^+, m^0, m^-) \\
&= p(x_t \mid m^+, m^0, m^-)\ p(m^+, m^0, m^-) \\
&= p(x_t \mid m^+, m^0, m^- = 0)\ p(m^+, m^0, m^-) \\
&\simeq \ldots
\end{aligned}
$$

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:

$$
\begin{aligned}
p(x_t, m) &= p(x_t, m^+, m^0, m^-) \\
&= p(x_t \mid m^+, m^0, m^-)\ p(m^+, m^0, m^-) \\
&= p(x_t \mid m^+, m^0, m^- = 0)\ p(m^+, m^0, m^-) \\
&\simeq p(x_t \mid m^+, m^- = 0)\ p(m^+, m^0, m^-)
\end{aligned}
$$

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:

$$
\begin{aligned}
p(x_t, m) &= p(x_t, m^+, m^0, m^-) \\
&= p(x_t \mid m^+, m^0, m^-)\ p(m^+, m^0, m^-) \\
&= p(x_t \mid m^+, m^0, m^- = 0)\ p(m^+, m^0, m^-) \\
&\simeq p(x_t \mid m^+, m^- = 0)\ p(m^+, m^0, m^-) \\
&= \frac{p(x_t, m^+ \mid m^- = 0)}{p(m^+ \mid m^- = 0)}\ p(m^+, m^0, m^-) \\
&= \tilde{p}(x_t, m)
\end{aligned}
$$

Information Matrix Update

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

$$
\begin{aligned}
&\tilde{p}(x_t, m \mid z_{1:t}, u_{1:t}) \\
&\quad \simeq \frac{p(x_t, m^+ \mid m^- = 0, z_{1:t}, u_{1:t})}{p(m^+ \mid m^- = 0, z_{1:t}, u_{1:t})}\ p(m^0, m^+, m^- \mid z_{1:t}, u_{1:t})
\end{aligned}
$$

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

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

Sparsified Information Matrix

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

- Conditioning Ω_t on $m^- = 0$ yields Ω_t^0
- Marginalizing m^0 from Ω_t^0 yields Ω_t^1
- Marginalizing x, m^0 from Ω_t^0 yields Ω_t^2
- Marginalizing x from Ω_t yields Ω_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:

$$\begin{aligned}
\tilde{\xi}_t &= \tilde{\Omega}_t \, \mu_t \\
&= (\Omega_t - \Omega_t + \tilde{\Omega}_t) \, \mu_t \\
&= \Omega_t \, \mu_t + (\tilde{\Omega}_t - \Omega_t) \, \mu_t \\
&= \xi_t + (\tilde{\Omega}_t - \Omega_t) \, \mu_t
\end{aligned}$$

Sparsification

SEIF_sparsification(ξ_t, Ω_t, μ_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: $\Omega_t^0 = F_{x,m^+,m^0} \, F_{x,m^+,m^0}^T \, \Omega_t \, F_{x,m^+,m^0} \, F_{x,m^+,m^0}^T$

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

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

5: *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_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**$(\xi_{t-1}, \Omega_{t-1}, \mu_{t-1}, u_t)$~~ **DONE**

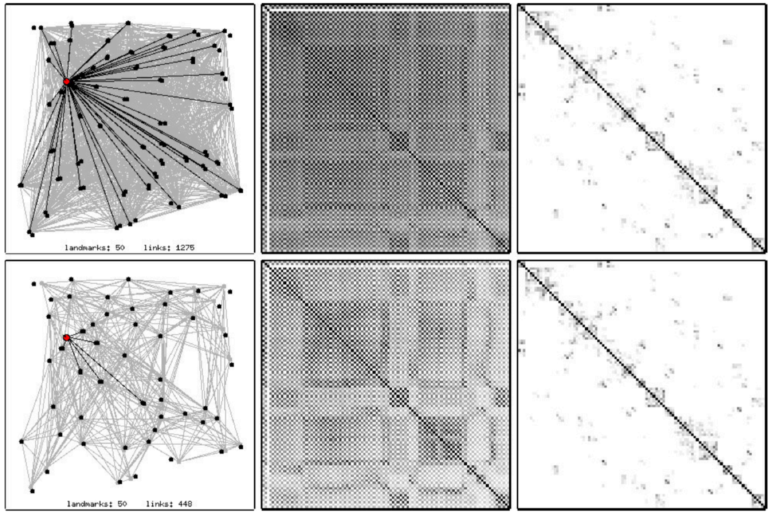
2: $\xi_t, \Omega_t =$ ~~**SEIF_measurement_update**$(\bar{\xi}_t, \bar{\Omega}_t, \bar{\mu}_t, z_t)$~~ **DONE**

3: $\mu_t =$ ~~**SEIF_update_state_estimate**$(\xi_t, \Omega_t, \bar{\mu}_t)$~~ **DONE**

4: $\tilde{\xi}_t, \tilde{\Omega}_t =$ ~~**SEIF_sparsification**(ξ_t, Ω_t, μ_t)~~ **DONE**

5: *return* $\tilde{\xi}_t, \tilde{\Omega}_t, \mu_t$

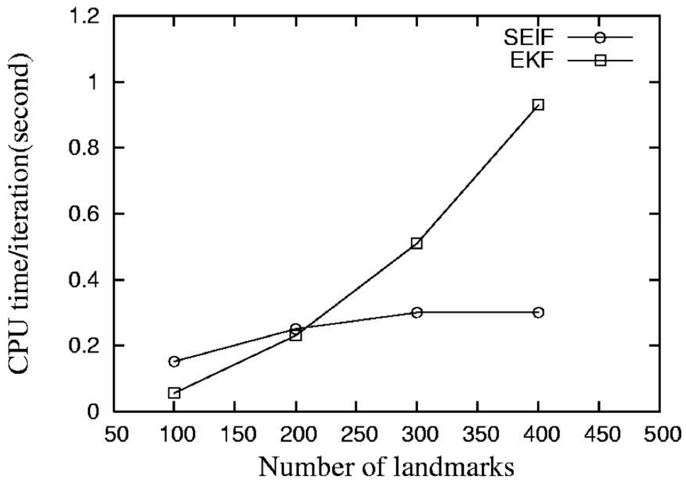
Effect of the Sparsification



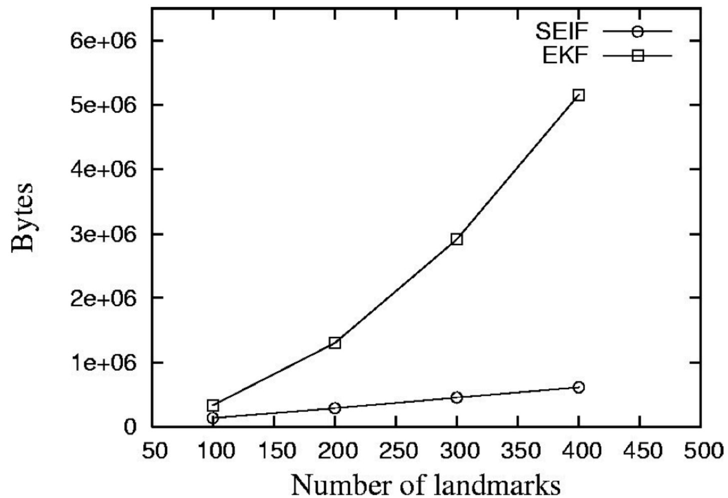
SEIF SLAM vs. EKF SLAM

- 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)

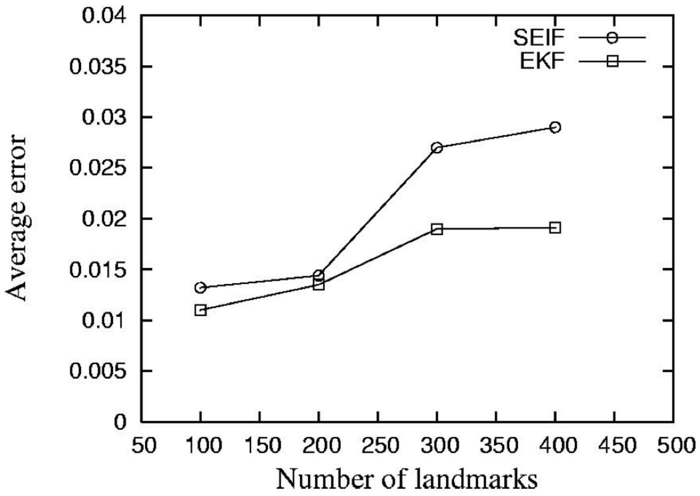
SEIF & EKF: CPU Time



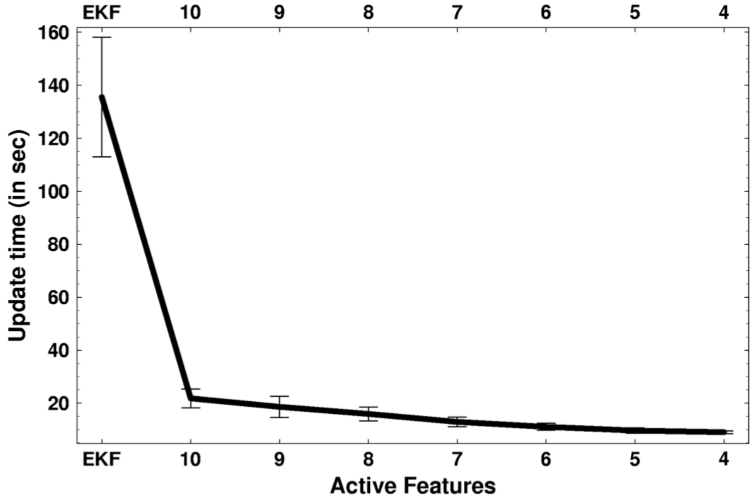
SEIF & EKF: Memory Usage



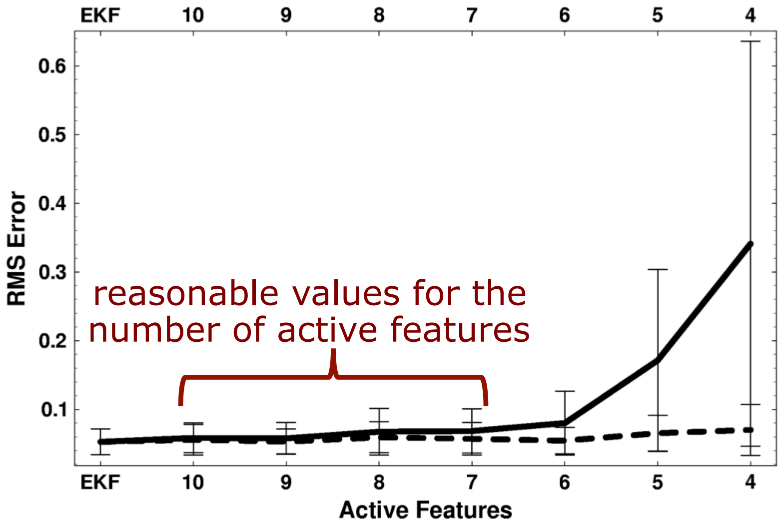
SEIF & EKF: Error Comparison



Influence of the Active Features



Influence of the 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 approxmation
- **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