iSAM: Incremental Smoothing and Mapping

Michael Kaess, Ananth Ranganathan, Frank Dellaert

Antonio Terán Espinoza
MRG Group Meeting
MIT CSAIL
Dec 3, 2018

What is iSAM?

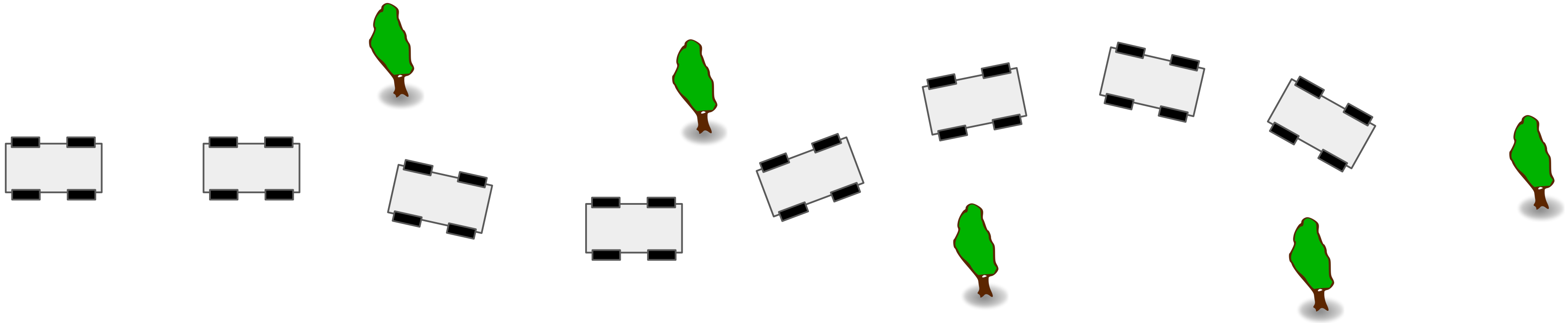
An approach to tackle the SLAM problem based on fast incremental matrix factorization.

It allows us to compute an efficient and exact solution by exploiting the sparsity behind the underlying structure of SLAM problem.

Works by incrementally updating a QR factorization of the smoothing information matrix every time new measurements are obtained.

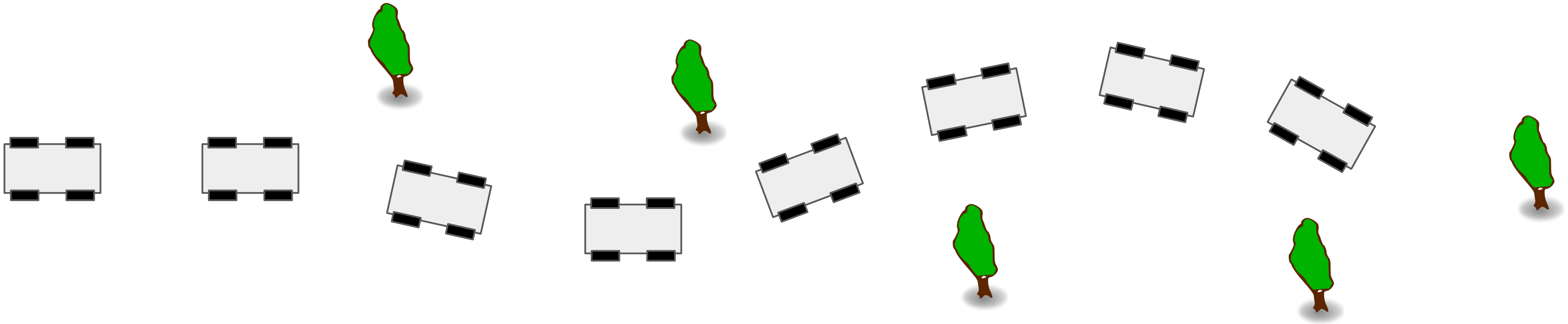
What is SLAM (simultaneous localization and mapping)?

Its goal is to provide an estimate for both the robot trajectory and the map of its environment, given all available sensor information.



What is SLAM (simultaneous localization and mapping)?

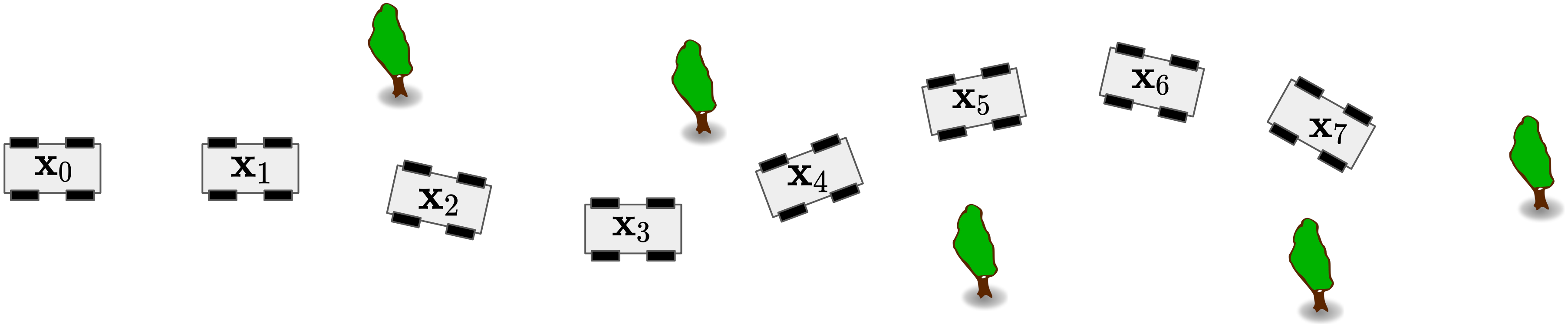
Its goal is to provide an estimate for both the robot trajectory and the map of its environment, given all available sensor information.



SLAM involves four main ingredients:

What is SLAM (simultaneous localization and mapping)?

Its goal is to provide an estimate for both the robot trajectory and the map of its environment, given all available sensor information.

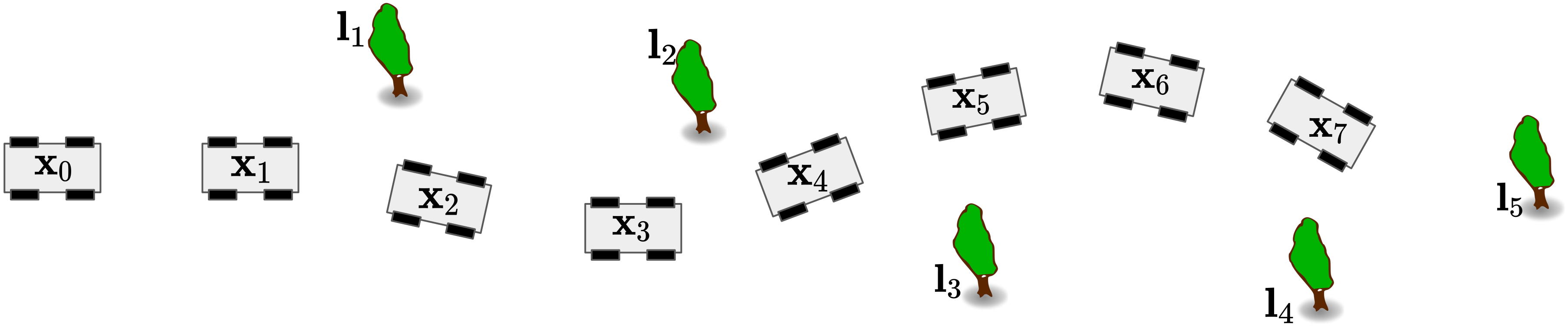


SLAM involves four main ingredients:

1. robot poses: $\mathbf{X} = \{\mathbf{x}_0, \ldots, \mathbf{x}_M\}$

What is SLAM (simultaneous localization and mapping)?

Its goal is to provide an estimate for both the robot trajectory and the map of its environment, given all available sensor information.



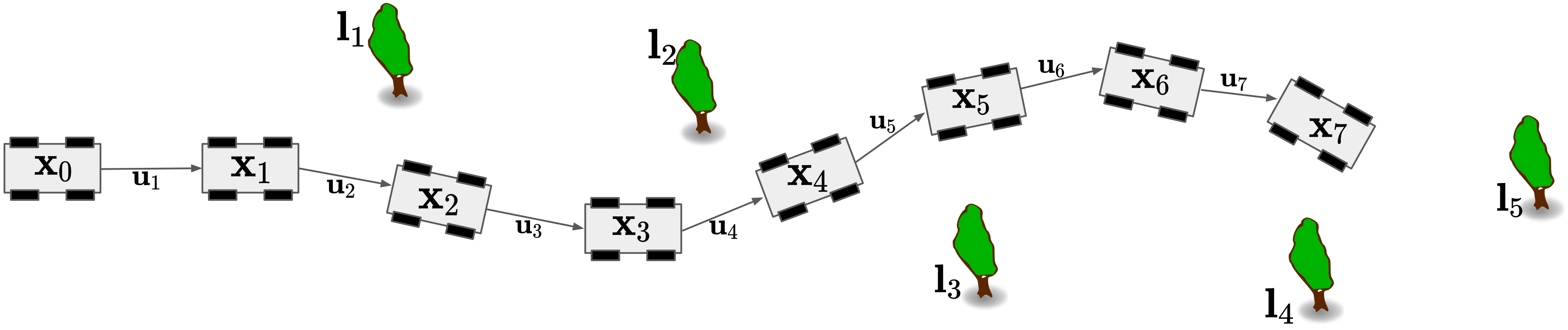
SLAM involves four main ingredients:

1. robot poses: $\mathbf{X} = \{\mathbf{x}_0, \ldots, \mathbf{x}_M\}$

2. landmarks/map: $\mathbf{L} = \{\mathbf{l}_0, \ldots, \mathbf{l}_N\}$

What is SLAM (simultaneous localization and mapping)?

Its goal is to provide an estimate for both the robot trajectory and the map of its environment, given all available sensor information.



SLAM involves four main ingredients:

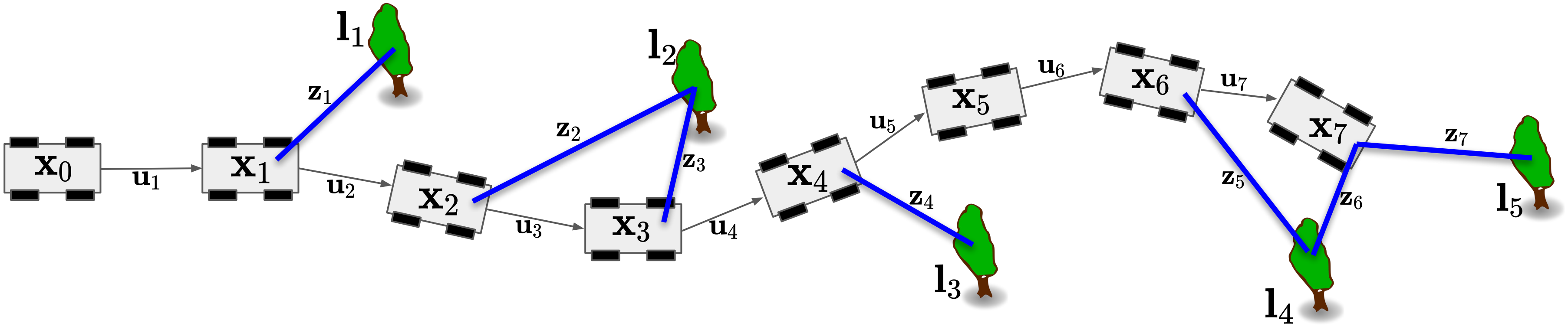
1. robot poses: $\mathbf{X} = \{\mathbf{x}_0, \ldots, \mathbf{x}_M\}$

2. landmarks/map: $\mathbf{L} = \{\mathbf{l}_0, \ldots, \mathbf{l}_N\}$

3. control inputs: $\mathbf{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_M\}$

What is SLAM (simultaneous localization and mapping)?

Its goal is to provide an estimate for both the robot trajectory and the map of its environment, given all available sensor information.



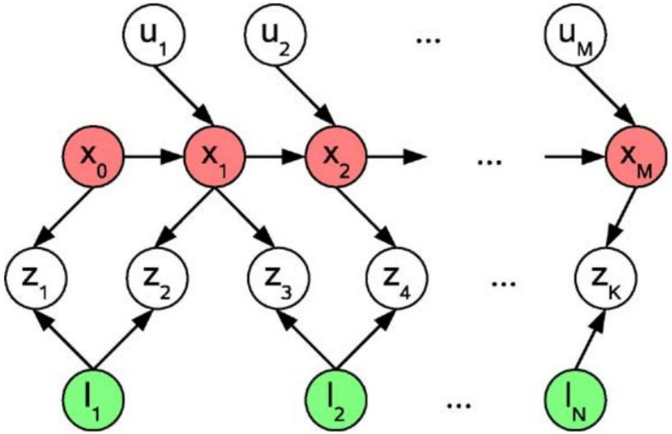
SLAM involves four main ingredients:

1. robot poses: $\mathbf{X} = \{\mathbf{x}_0, \ldots, \mathbf{x}_M\}$
2. landmarks/map: $\mathbf{L} = \{\mathbf{l}_0, \ldots, \mathbf{l}_N\}$
3. control inputs: $\mathbf{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_M\}$
4. measurements: $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_K\}$

Probabilistic model for SLAM

Express the joint probability of all variables and measurements as Bayesian net:

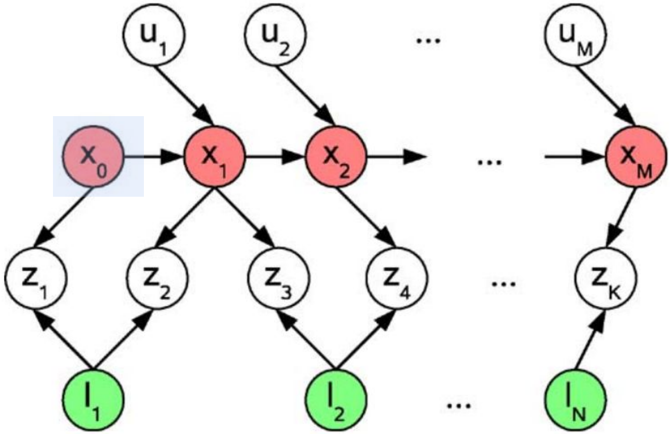
$$P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) =$$



Probabilistic model for SLAM

Express the joint probability of all variables and measurements as Bayesian net:

$$P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) =$$

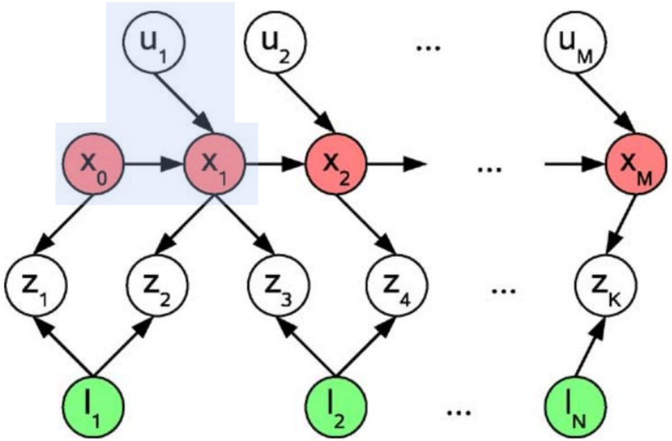


$$\propto P(\mathbf{x}_0)$$

Probabilistic model for SLAM

Express the joint probability of all variables and measurements as Bayesian net:

$$P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) =$$

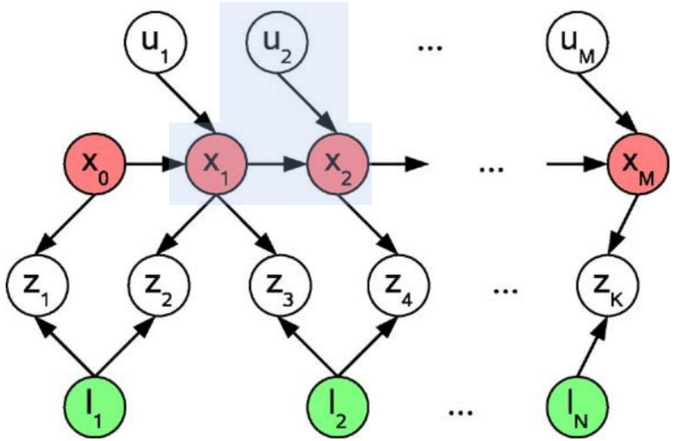


$$\propto P(\mathbf{x}_0) \cdot P(\mathbf{x}_1 | \mathbf{x}_0, \mathbf{u}_1)$$

Probabilistic model for SLAM

Express the joint probability of all variables and measurements as Bayesian net:

$$P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) =$$

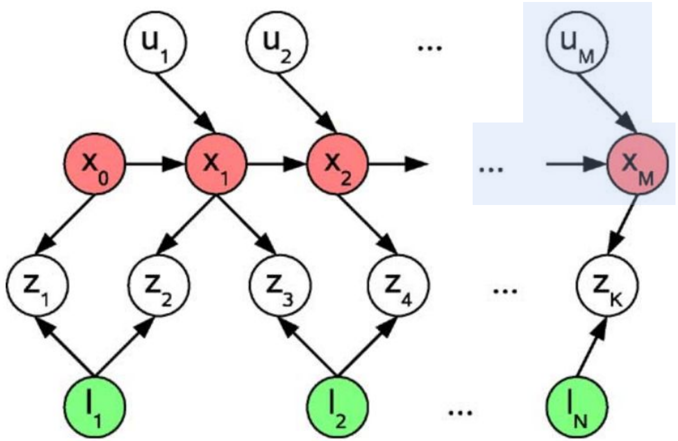


$$\propto P(\mathbf{x}_0) \cdot P(\mathbf{x}_1 | \mathbf{x}_0, \mathbf{u}_1) \cdot P(\mathbf{x}_2 | \mathbf{x}_1, \mathbf{u}_2)$$

Probabilistic model for SLAM

Express the joint probability of all variables and measurements as Bayesian net:

$$P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) =$$

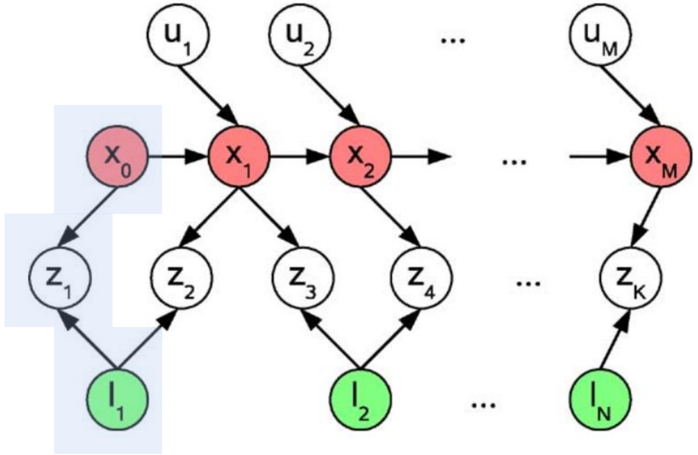


$$\propto P(\mathbf{x}_0) \cdot P(\mathbf{x}_1 | \mathbf{x}_0, \mathbf{u}_1) \cdot P(\mathbf{x}_2 | \mathbf{x}_1, \mathbf{u}_2) \cdots P(\mathbf{x}_M | \mathbf{x}_{M-1}, \mathbf{u}_M)$$

Probabilistic model for SLAM

Express the joint probability of all variables and measurements as Bayesian net:

$$P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) =$$

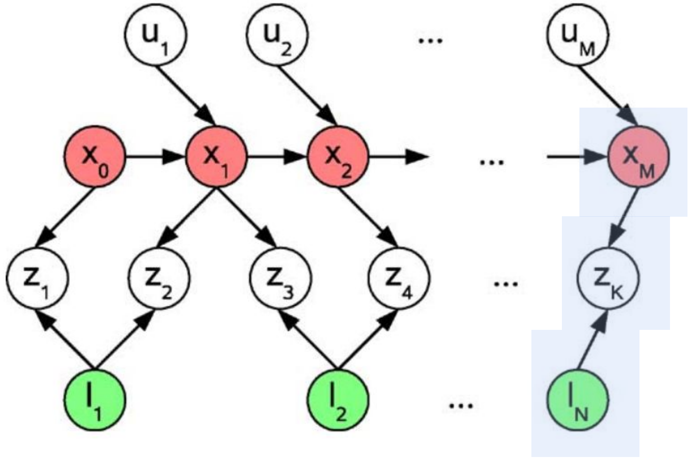


$$\propto P(\mathbf{x}_0) \cdot P(\mathbf{x}_1 | \mathbf{x}_0, \mathbf{u}_1) \cdot P(\mathbf{x}_2 | \mathbf{x}_1, \mathbf{u}_2) \cdots P(\mathbf{x}_M | \mathbf{x}_{M-1}, \mathbf{u}_M) \cdot P(\mathbf{z}_1 | \mathbf{x}_0, \mathbf{l}_1)$$

Probabilistic model for SLAM

Express the joint probability of all variables and measurements as Bayesian net:

$$P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) =$$

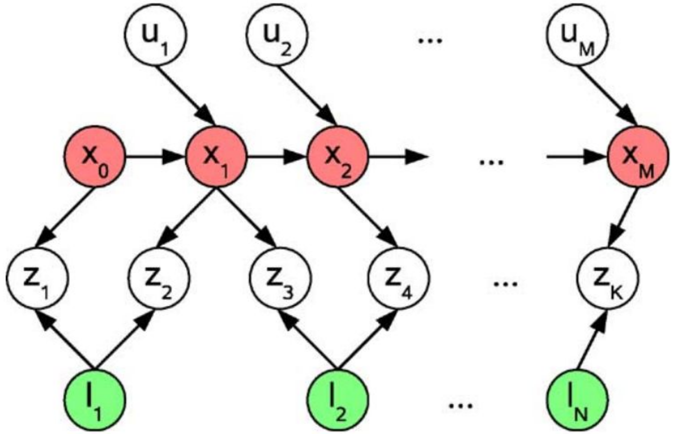


$$\propto P(\mathbf{x}_0) \cdot P(\mathbf{x}_1 | \mathbf{x}_0, \mathbf{u}_1) \cdot P(\mathbf{x}_2 | \mathbf{x}_1, \mathbf{u}_2) \cdots P(\mathbf{x}_M | \mathbf{x}_{M-1}, \mathbf{u}_M) \cdot P(\mathbf{z}_1 | \mathbf{x}_0, \mathbf{l}_1) \cdots P(\mathbf{z}_k | \mathbf{x}_M, \mathbf{l}_N)$$

Probabilistic model for SLAM

Express the joint probability of all variables and measurements as Bayesian net:

$$P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) =$$



$$\propto P(\mathbf{x}_0) \cdot P(\mathbf{x}_1 | \mathbf{x}_0, \mathbf{u}_1) \cdot P(\mathbf{x}_2 | \mathbf{x}_1, \mathbf{u}_2) \cdots P(\mathbf{x}_M | \mathbf{x}_{M-1}, \mathbf{u}_M) \cdot P(\mathbf{z}_1 | \mathbf{x}_0, \mathbf{l}_1) \cdots P(\mathbf{z}_k | \mathbf{x}_M, \mathbf{l}_N)$$

$$= P(\mathbf{x}_0) \prod_{i=1}^{M} P(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathbf{u}_i) \prod_{k=1}^{K} P(\mathbf{z}_i | \mathbf{x}_{i_k}, \mathbf{l}_{j_k})$$

Gaussian process and measurement models

$$P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) \propto P(\mathbf{x}_0) \prod_{i=1}^{M} P(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathbf{u}_i) \prod_{k=1}^{K} P(\mathbf{z}_i | \mathbf{x}_{i_k}, \mathbf{l}_{j_k})$$

Gaussian process and measurement models

$$P(\mathbf{X},\mathbf{L},\mathbf{U},\mathbf{Z}) \propto P(\mathbf{x}_0) \prod_{i=1}^{M} \underbrace{P(\mathbf{x}_i|\mathbf{x}_{i-1},\mathbf{u}_i)} \prod_{k=1}^{K} P(\mathbf{z}_i|\mathbf{x}_{i_k},\mathbf{l}_{j_k})$$

Motion model: $\mathbf{x}_i = f_i(\mathbf{x}_{i-1},\mathbf{u}_i) + \mathbf{w}_i$, with $\mathbf{w}_i \sim \mathcal{N}(0,\mathbf{\Lambda}_i)$

$$\Rightarrow \frac{1}{K_i}\exp\left\{-\frac{1}{2}(f_i(\mathbf{x}_{i-1},\mathbf{u}_i)-\mathbf{x}_i)^\top \mathbf{\Lambda}_i^{-1}(f_i(\mathbf{x}_{i-1},\mathbf{u}_i)-\mathbf{x}_i)\right\}$$

Gaussian process and measurement models

$$P(\mathbf{X},\mathbf{L},\mathbf{U},\mathbf{Z}) \propto P(\mathbf{x}_0)\ \prod_{i=1}^{M} \underbrace{P(\mathbf{x}_i|\mathbf{x}_{i-1},\mathbf{u}_i)}\ \prod_{k=1}^{K} \underbrace{P(\mathbf{z}_i|\mathbf{x}_{i_k},\mathbf{l}_{j_k})}$$

Motion model: $\mathbf{x}_i = f_i(\mathbf{x}_{i-1},\mathbf{u}_i) + \mathbf{w}_i$, with $\mathbf{w}_i \sim \mathcal{N}(0,\mathbf{\Lambda}_i)$

$$\Rightarrow \frac{1}{K_i}\exp\left\{-\frac{1}{2}(f_i(\mathbf{x}_{i-1},\mathbf{u}_i)-\mathbf{x}_i)^\top \mathbf{\Lambda}_i^{-1}\left(f_i(\mathbf{x}_{i-1},\mathbf{u}_i)-\mathbf{x}_i\right)\right\}$$

Measurement model: $\mathbf{z}_k = h_k(\mathbf{x}_{i_k},\mathbf{l}_{j_k}) + \mathbf{v}_i$, with $\mathbf{v}_i \sim \mathcal{N}(0,\mathbf{\Gamma}_k)$

$$\Rightarrow \frac{1}{K_i}\exp\left\{-\frac{1}{2}\left(h_k(\mathbf{x}_{i_k},\mathbf{l}_{j_k})-\mathbf{z}_k\right)^\top \mathbf{\Gamma}_k^{-1}\left(h_k(\mathbf{x}_{i_k},\mathbf{l}_{j_k})-\mathbf{z}_k\right)\right\}$$

SLAM as a nonlinear least squares problem

$$\mathbf{X}^*, \mathbf{L}^* = \arg\max_{\mathbf{X},\mathbf{L}} P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z})$$

SLAM as a nonlinear least squares problem

$$
\begin{aligned}
\mathbf{X}^*, \mathbf{L}^* &= \arg\max_{\mathbf{X},\mathbf{L}} P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) \\
&= \arg\min_{\mathbf{X},\mathbf{L}} -\log P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z})
\end{aligned}
$$

SLAM as a nonlinear least squares problem

$$
\begin{aligned}
\mathbf{X}^*, \mathbf{L}^* &= \arg\max_{\mathbf{X},\mathbf{L}} P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) \\
&= \arg\min_{\mathbf{X},\mathbf{L}} -\log P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) \\
&= \arg\min_{\mathbf{X},\mathbf{L}} \left\{ \sum_{i=1}^{M} ||f_i(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i||^2_{\mathbf{\Lambda}_i} + \sum_{k=1}^{K} ||h_i(\mathbf{x}_{i_k}, \mathbf{l}_{j_k}) - \mathbf{z}_k||^2_{\mathbf{\Gamma}_k} \right\}
\end{aligned}
$$

SLAM as a nonlinear least squares problem

$$
\begin{aligned}
\mathbf{X}^*, \mathbf{L}^* &= \arg\max_{\mathbf{X},\mathbf{L}} P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) \\
&= \arg\min_{\mathbf{X},\mathbf{L}} - \log P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z}) \\
&= \arg\min_{\mathbf{X},\mathbf{L}} \left\{ \sum_{i=1}^{M} ||f_i(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i||^2_{\mathbf{\Lambda}_i} + \sum_{k=1}^{K} ||h_i(\mathbf{x}_{i_k}, \mathbf{l}_{j_k}) - \mathbf{z}_k||^2_{\mathbf{\Gamma}_k} \right\}
\end{aligned}
$$

with $||\mathbf{e}||^2_{\mathbf{\Sigma}} = \mathbf{e}^\top \mathbf{\Sigma}^{-1} \mathbf{e}$ denoting the squared Mahalanobis distance.

First order linearization via Taylor expansion

$$\mathbf{X}^*, \mathbf{L}^* = \arg\min_{\mathbf{X},\mathbf{L}} \left\{ \sum_{i=1}^{M} ||f_i(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i||^2_{\mathbf{\Lambda}_i} + \sum_{k=1}^{K} ||h_i(\mathbf{x}_{i_k}, \mathbf{l}_{j_k}) - \mathbf{z}_k||^2_{\mathbf{\Gamma}_k} \right\}$$

First order linearization via Taylor expansion

$$\mathbf{X}^*, \mathbf{L}^* = \arg\min_{\mathbf{X},\mathbf{L}} \left\{ \sum_{i=1}^{M} ||\underline{f_i(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i}||^2_{\mathbf{\Lambda}_i} + \sum_{k=1}^{K} ||h_i(\mathbf{x}_{i_k}, \mathbf{l}_{j_k}) - \mathbf{z}_k||^2_{\mathbf{\Gamma}_k} \right\}$$

$$\begin{aligned} &\underline{f_i(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i} \\ \approx &\{f_i(\mathbf{x}^0_{i-1}, \mathbf{u}_i) + F_i^{i-1}\delta\mathbf{x}_{i-1}\} - \{\mathbf{x}^0_i + \delta\mathbf{x}_i\} \\ = &\{F_i^{i-1}\delta\mathbf{x}_{i-1} - \delta\mathbf{x}_i\} - \{\mathbf{a}_i\} \end{aligned}$$

$$F_i^{i-1} := \left.\frac{\partial f_i(\mathbf{x}_{x-1}, \mathbf{u}_i)}{\partial \mathbf{x}_{i-1}}\right|_{\mathbf{x}^0_{i-1}}$$

Jacobian of process model

$$\mathbf{a}_i := \mathbf{x}^0_i - f_i(\mathbf{x}^0_{i-1}, \mathbf{u}_i)$$

odometry prediction error

First order linearization via Taylor expansion

$$\mathbf{X}^*, \mathbf{L}^* = \arg\min_{\mathbf{X},\mathbf{L}} \left\{ \sum_{i=1}^{M} \|\underline{f_i(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i}\|^2_{\mathbf{\Lambda}_i} + \sum_{k=1}^{K} \|\underline{h_i(\mathbf{x}_{i_k}, \mathbf{l}_{j_k}) - \mathbf{z}_k}\|^2_{\mathbf{\Gamma}_k} \right\}$$

$$\begin{aligned} &\underline{f_i(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i} \\ &\approx \{f_i(\mathbf{x}^0_{i-1}, \mathbf{u}_i) + F_i^{i-1}\delta\mathbf{x}_{i-1}\} - \{\mathbf{x}^0_i + \delta\mathbf{x}_i\} \\ &= \{F_i^{i-1}\delta\mathbf{x}_{i-1} - \delta\mathbf{x}_i\} - \{\mathbf{a}_i\} \end{aligned}$$

$$F_i^{i-1} := \left.\frac{\partial f_i(\mathbf{x}_{x-1}, \mathbf{u}_i)}{\partial \mathbf{x}_{i-1}}\right|_{\mathbf{x}^0_{i-1}}$$

Jacobian of process model

$$\mathbf{a}_i := \mathbf{x}^0_i - f_i(\mathbf{x}^0_{i-1}, \mathbf{u}_i)$$

odometry prediction error

$$\begin{aligned} &\underline{h_k(\mathbf{x}_{i_k}, \mathbf{l}_{j_k}) - \mathbf{z}_k} \\ &\approx \{h_k(\mathbf{x}^0_{i_k}, \mathbf{l}^0_{j_k}) + H_k^{i_k}\delta\mathbf{x}_{i_k} + J_k^{j_k}\delta\mathbf{l}_{j_k}\} - \mathbf{z}_k \\ &= \{H_k^{i_k}\delta\mathbf{x}_{i_k} + J_k^{j_k}\delta\mathbf{l}_{j_k}\} - \{\mathbf{c}_k\} \end{aligned}$$

$$H_k^{i_k} := \left.\frac{\partial h_k(\mathbf{x}_{i_k}, \mathbf{l}_{j_k})}{\partial \mathbf{x}_{i_k}}\right|_{(\mathbf{x}^0_{i_k}, \mathbf{l}^0_{j_k})} \qquad J_k^{j_k} := \left.\frac{\partial h_k(\mathbf{x}_{i_k}, \mathbf{l}_{j_k})}{\partial \mathbf{l}_{j_k}}\right|_{(\mathbf{x}^0_{i_k}, \mathbf{l}^0_{j_k})}$$

Jacobians of measurement model

$$\mathbf{c}_k := \mathbf{z}_k - h_k(\mathbf{x}^0_{i_k}, \mathbf{k}^0_{j_k})$$

measurement prediction error

Eliminating the covariances (whitening)

$$||\mathbf{e}||_{\boldsymbol{\Sigma}}^2 := \mathbf{e}^\top \boldsymbol{\Sigma}^{-1} \mathbf{e} = \left(\boldsymbol{\Sigma}^{-1/2}\mathbf{e}\right)^\top \left(\boldsymbol{\Sigma}^{-1/2}\mathbf{e}\right) = ||\boldsymbol{\Sigma}^{-1/2}\mathbf{e}||_2^2$$

Eliminating the covariances (whitening)

$$||\mathbf{e}||^2_{\mathbf{\Sigma}} := \mathbf{e}^\top \mathbf{\Sigma}^{-1} \mathbf{e} = \left(\mathbf{\Sigma}^{-1/2}\mathbf{e}\right)^\top \left(\mathbf{\Sigma}^{-1/2}\mathbf{e}\right) = ||\mathbf{\Sigma}^{-1/2}\mathbf{e}||^2_2$$

rewriting terms

$$||h_k(\mathbf{x}_{i_k}, \mathbf{l}_{j_k}) - \mathbf{z}_k||^2_{\mathbf{\Gamma}_k} \longrightarrow ||H^{i_k}_k \delta\mathbf{x}_{i_k} - J^{j_k}_k \delta\mathbf{l}_{j_k} - \mathbf{c}_k||^2_{\mathbf{\Gamma}_k}$$

Eliminating the covariances (whitening)

$$||\mathbf{e}||^2_{\boldsymbol{\Sigma}} := \mathbf{e}^\top \boldsymbol{\Sigma}^{-1}\mathbf{e} = \left(\boldsymbol{\Sigma}^{-1/2}\mathbf{e}\right)^\top \left(\boldsymbol{\Sigma}^{-1/2}\mathbf{e}\right) = ||\boldsymbol{\Sigma}^{-1/2}\mathbf{e}||^2_2$$

rewriting terms

$$||h_k(\mathbf{x}_{i_k}, \mathbf{l}_{j_k}) - \mathbf{z}_k||^2_{\boldsymbol{\Gamma}_k} \longrightarrow ||H_k^{i_k}\,\delta\mathbf{x}_{i_k} - J_k^{j_k}\,\delta\mathbf{l}_{j_k} - \mathbf{c}_k||^2_{\boldsymbol{\Gamma}_k}$$

premultiplying and stacking

$$\mathbf{A}_k = \begin{bmatrix} \boldsymbol{\Gamma}_k^{-1/2} H_k^{i_k} & \boldsymbol{\Gamma}_k^{-1/2} J_k^{j_k} \end{bmatrix} \qquad \mathbf{b}_k = \boldsymbol{\Gamma}_k^{-1/2}\mathbf{c}_k \qquad \delta\boldsymbol{\theta}_k = \begin{bmatrix} \delta\mathbf{x}_{i_k} \\ \delta\mathbf{l}_{j_k} \end{bmatrix}$$

Eliminating the covariances (whitening)

$$||\mathbf{e}||^2_{\boldsymbol{\Sigma}} := \mathbf{e}^\top \boldsymbol{\Sigma}^{-1}\mathbf{e} = \left(\boldsymbol{\Sigma}^{-1/2}\mathbf{e}\right)^\top \left(\boldsymbol{\Sigma}^{-1/2}\mathbf{e}\right) = ||\boldsymbol{\Sigma}^{-1/2}\mathbf{e}||^2_2$$

rewriting terms

$$||h_k(\mathbf{x}_{i_k}, \mathbf{l}_{j_k}) - \mathbf{z}_k||^2_{\boldsymbol{\Gamma}_k} \longrightarrow ||H_k^{i_k}\delta\mathbf{x}_{i_k} - J_k^{j_k}\delta\mathbf{l}_{j_k} - \mathbf{c}_k||^2_{\boldsymbol{\Gamma}_k}$$

premultiplying and stacking

$$\mathbf{A}_k = \begin{bmatrix} \boldsymbol{\Gamma}_k^{-1/2} H_k^{i_k} & \boldsymbol{\Gamma}_k^{-1/2} J_k^{j_k} \end{bmatrix} \qquad \mathbf{b}_k = \boldsymbol{\Gamma}_k^{-1/2}\mathbf{c}_k \qquad \delta\boldsymbol{\theta}_k = \begin{bmatrix} \delta\mathbf{x}_{i_k} \\ \delta\mathbf{l}_{j_k} \end{bmatrix}$$

$$\Rightarrow ||\mathbf{A}_k\delta\boldsymbol{\theta}_k - \mathbf{b}_k||^2_2$$ obtaining standard least squares!

Standard least square with measurement Jacobian

Stacking matrices for each term in the summation, we obtain

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} ||\mathbf{A}\boldsymbol{\theta} - \mathbf{b}||^2$$

$\boldsymbol{\theta} \in \mathbb{R}^n$ vector containing all pose and landmark variables

$\mathbf{A} \in \mathbb{R}^{m \times n}$ sparse measurement Jacobian w/ *m* measurement rows

$\mathbf{b} \in \mathbb{R}^m$ right-hand side (RHS) vector

Solution: solving the so called normal equations $\mathbf{A}^\top \mathbf{A}\boldsymbol{\theta} = \mathbf{A}^\top \mathbf{b}$

Solving by QR factorization

To avoid having to calculate the information matrix $\mathbf{A}^\top \mathbf{A}$, standard QR factorization is applied to the measurements Jacobian $\mathbf{A}$:

$$\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$ with

$\mathbf{R} \in \mathbb{R}^{n \times n}$: upper triangular square root info matrix

$\mathbf{Q} \in \mathbb{R}^{m \times m}$: orthogonal matrix,

transforming
$$\begin{aligned} ||\mathbf{A}\boldsymbol{\theta} - \mathbf{b}||^2 &= \left|\left|\mathbf{Q}\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}\boldsymbol{\theta} - \mathbf{b}\right|\right|^2 \\ &= \left|\left|\mathbf{Q}^\top\mathbf{Q}\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}\boldsymbol{\theta} - \mathbf{Q}^\top\mathbf{b}\right|\right|^2 \\ &= \left|\left|\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}\boldsymbol{\theta} - \begin{bmatrix} \mathbf{d} \\ \mathbf{e} \end{bmatrix}\right|\right|^2 \\ &= ||\mathbf{R}\boldsymbol{\theta} - \mathbf{d}||^2 + ||\mathbf{e}||^2 \end{aligned}$$

with $[\mathbf{d},\ \mathbf{e}]^\top := \mathbf{Q}^\top \mathbf{b}$

$\mathbf{d} \in \mathbb{R}^n$

$||\mathbf{e}||^2$:least squares residual

Linear system with unique solution

$$||\mathbf{A}\boldsymbol{\theta} - \mathbf{b}||^2 = ||\mathbf{R}\boldsymbol{\theta} - \mathbf{d}||^2 + ||\mathbf{e}||^2$$

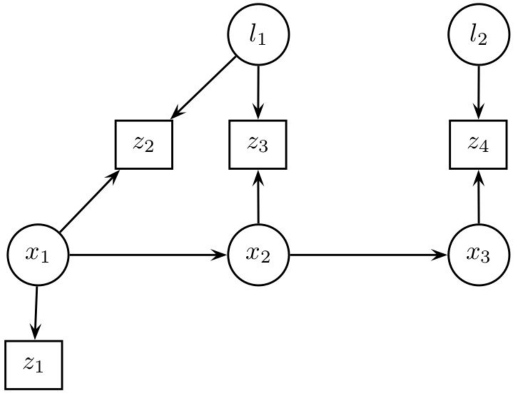
$$\Rightarrow \mathbf{R}\boldsymbol{\theta}^* = \mathbf{d}$$ obtain solution by simple back-substitution!

$\boldsymbol{\theta}^*$ is the least squares estimate for the complete robot trajectory and the map, conditioned on all measurements

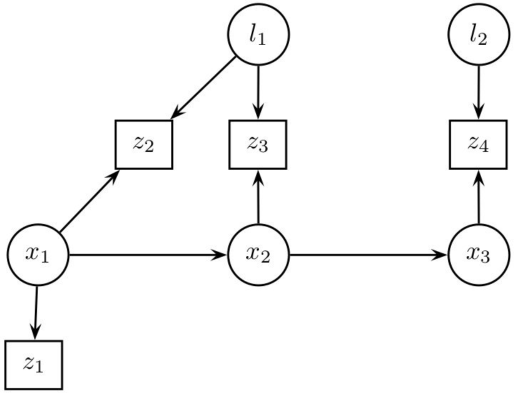
So far so good

Simple toy example:

- Factorize the joint distribution $P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z})$
- Cast as nonlinear squares problem



So far so good

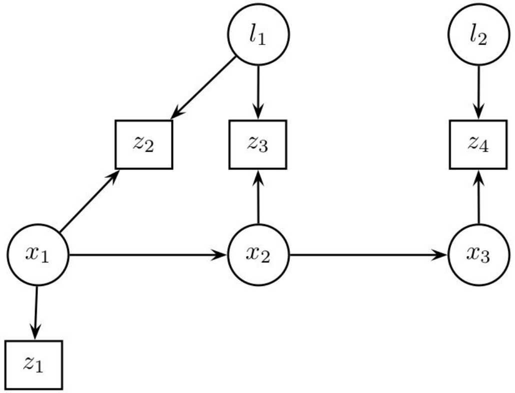


Simple toy example:

- Factorize the joint distribution $P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z})$
- Cast as nonlinear squares problem
- Linearize and create measurement Jacobian $\mathbf{A}$

$$[\mathbf{A}|\mathbf{b}] = \begin{array}{c} \begin{array}{cccccc} \delta l_1 & \delta l_2 & \delta x_1 & \delta x_2 & \delta x_3 & b \end{array} \\ \left[\begin{array}{ccccc|c} & & A_{13} & & & b_1 \\ & & A_{23} & A_{24} & & b_2 \\ & & & A_{34} & A_{35} & b_3 \\ A_{41} & & & & & b_4 \\ & A_{52} & & & & b_5 \\ & & A_{63} & & & b_6 \\ A_{71} & & A_{73} & & & b_7 \\ A_{81} & & & A_{84} & & b_8 \\ & A_{92} & & & A_{95} & b_9 \end{array}\right] \end{array}$$

So far so good



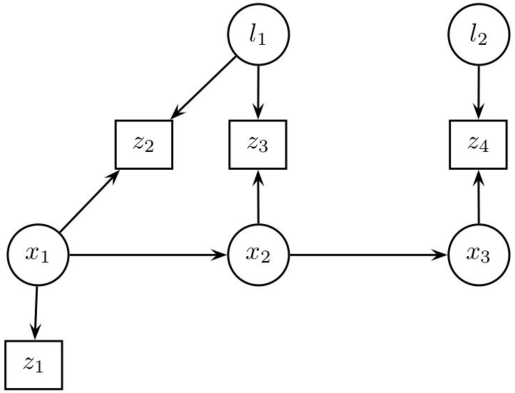
Simple toy example:

- Factorize the joint distribution $P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z})$
- Cast as nonlinear squares problem
- Linearize and create measurement Jacobian $\mathbf{A}$

$$
[\mathbf{A}|\mathbf{b}] = \begin{array}{c} \begin{array}{cccccc} \delta l_1 & \delta l_2 & \delta x_1 & \delta x_2 & \delta x_3 & b \end{array} \\ \left[\begin{array}{ccccc|c}
 & & A_{13} & & & b_1 \\
 & & A_{23} & A_{24} & & b_2 \\
 & & & A_{34} & A_{35} & b_3 \\
A_{41} & & & & & b_4 \\
 & A_{52} & & & & b_5 \\
 & & A_{63} & & & b_6 \\
A_{71} & & A_{73} & & & b_7 \\
A_{81} & & & A_{84} & & b_8 \\
 & A_{92} & & & A_{95} & b_9
\end{array}\right] \end{array}
\longrightarrow
\mathbf{R} = \begin{bmatrix}
R_1 & & T_{13} & T_{14} & \\
 & R_2 & & & T_{25} \\
 & & R_3 & T_{34} & \\
 & & & R_4 & T_{45} \\
 & & & & \mathbf{R}_5
\end{bmatrix}
$$

- Perform QR factorization for obtaining square root information matrix $\mathbf{R}$

So far so good



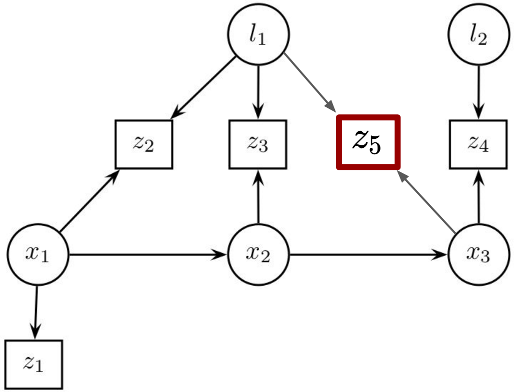
Simple toy example:

- Factorize the joint distribution $P(\mathbf{X}, \mathbf{L}, \mathbf{U}, \mathbf{Z})$
- Cast as nonlinear squares problem
- Linearize and create measurement Jacobian $\mathbf{A}$

$$
[\mathbf{A}|\mathbf{b}] = \begin{array}{c} \begin{array}{cccccc} \delta l_1 & \delta l_2 & \delta x_1 & \delta x_2 & \delta x_3 & b \end{array} \\ \left[\begin{array}{ccccc|c}
 & & A_{13} & & & b_1 \\
 & & A_{23} & A_{24} & & b_2 \\
 & & & A_{34} & A_{35} & b_3 \\
A_{41} & & & & & b_4 \\
 & A_{52} & & & & b_5 \\
 & & A_{63} & & & b_6 \\
A_{71} & & A_{73} & & & b_7 \\
A_{81} & & & A_{84} & & b_8 \\
 & A_{92} & & & A_{95} & b_9
\end{array}\right] \end{array}
\longrightarrow
\mathbf{R} = \begin{bmatrix}
R_1 & & T_{13} & T_{14} & \\
 & R_2 & & & T_{25} \\
 & & R_3 & T_{34} & \\
 & & & R_4 & T_{45} \\
 & & & & \mathbf{R}_5
\end{bmatrix}
\longrightarrow
\boldsymbol{\theta}^* = \begin{bmatrix} \mathbf{X}^* \\ \mathbf{L}^* \end{bmatrix}
$$

- Perform QR factorization for obtaining square root information matrix $\mathbf{R}$
- Obtain solution by solving linear system via back-substitution

What if I keep exploring?



When a new measurement arrives, instead of updating and refactoring the matrix $\mathbf{A}$, ***incremental QR-updating*** is performed:

new information $\mathbf{w}^\top \in \mathbb{R}^n$

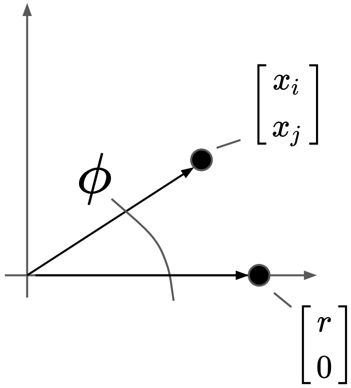
QR-updating: $\begin{bmatrix} Q^T & \\ & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \times & \times & \times & \times & \times \end{bmatrix}$

Use **Givens rotations** to clean out non-zero entries!

Givens rotations

Rotation matrix $\Phi := \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$ such that when applied to $\begin{bmatrix} x_i \\ x_j \end{bmatrix}$ we obtain

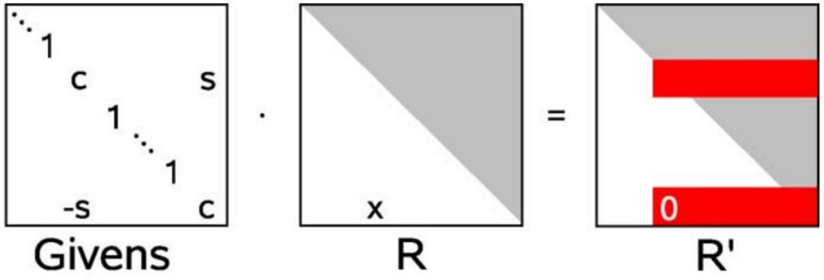
$$\begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}, \text{ with } r = \sqrt{x_i^2 + x_j^2}$$



$$\Rightarrow \sin\phi = \frac{x_j}{\sqrt{x_i^2 + x_j^2}}$$

$$\Rightarrow \cos\phi = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}$$

example



Back to our example:

$$\begin{bmatrix} Q^T & \\ & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \times & \times & \times & \times & \times \end{bmatrix}$$

defining $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix}$

Back to our example:

$$\begin{bmatrix} Q^T & \\ & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \times & \times & \times & \times & \times \end{bmatrix}$$

defining $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix}$

clear 1st element

$$\mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \times & \times & \times & \times \end{bmatrix}$$

Back to our example:

$$\begin{bmatrix} Q^T & \\ & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \times & \times & \times & \times & \times \end{bmatrix}$$

defining $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix}$

clear 1st element

clear 2nd element

$$\mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \times & \times & \times & \times \end{bmatrix} \longrightarrow \mathbf{G}_{j_2} \mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \times & \times & \times \end{bmatrix}$$

Back to our example:

$$\begin{bmatrix} Q^T & \\ & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \times & \times & \times & \times & \times \end{bmatrix}$$

defining $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix}$

clear 1st element

$$\mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \times & \times & \times & \times \end{bmatrix}$$

clear 2nd element

$$\longrightarrow \mathbf{G}_{j_2} \mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \times & \times & \times \end{bmatrix}$$

clear 3rd element

$$\longrightarrow \mathbf{G}_{j_3} \mathbf{G}_{j_2} \mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \times & \times \end{bmatrix}$$

Back to our example:

$$\begin{bmatrix} Q^T & \\ & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \times & \times & \times & \times & \times \end{bmatrix}$$

defining $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix}$

clear 1st element

clear 2nd element

clear 3rd element

...

$$\mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \times & \times & \times & \times \end{bmatrix} \longrightarrow \mathbf{G}_{j_2}\mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \times & \times & \times \end{bmatrix} \longrightarrow \mathbf{G}_{j_3}\mathbf{G}_{j_2}\mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \times & \times \end{bmatrix} \longrightarrow \mathbf{G}_{j_4}\mathbf{G}_{j_3}\mathbf{G}_{j_2}\mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \times \end{bmatrix}$$

Back to our example:

$$\begin{bmatrix} Q^T & \\ & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \times & \times & \times & \times & \times \end{bmatrix}$$

defining $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix}$

clear 1st element

$$\mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \times & \times & \times & \times \end{bmatrix}$$

clear 2nd element

$$\longrightarrow \mathbf{G}_{j_2} \mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \times & \times & \times \end{bmatrix}$$

clear 3rd element

$$\longrightarrow \mathbf{G}_{j_3} \mathbf{G}_{j_2} \mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \times & \times \end{bmatrix}$$

...

$$\longrightarrow \mathbf{G}_{j_4} \mathbf{G}_{j_3} \mathbf{G}_{j_2} \mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \times \end{bmatrix}$$

$$\mathbf{G}_{j_5} \mathbf{G}_{j_4} \mathbf{G}_{j_3} \mathbf{G}_{j_2} \mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}} \\ 0 \end{bmatrix}$$

Back to our example:

$$\begin{bmatrix} Q^T & \\ & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \times & \times & \times & \times & \times \end{bmatrix}$$

defining $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix}$

clear 1st element | clear 2nd element | clear 3rd element | ...

$$\mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \times & \times & \times & \times \end{bmatrix} \longrightarrow \mathbf{G}_{j_2} \mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \times & \times & \times \end{bmatrix} \longrightarrow \mathbf{G}_{j_3} \mathbf{G}_{j_2} \mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \times & \times \end{bmatrix} \longrightarrow \mathbf{G}_{j_4} \mathbf{G}_{j_3} \mathbf{G}_{j_2} \mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \times \end{bmatrix}$$

$$\mathbf{G}_{j_5} \mathbf{G}_{j_4} \mathbf{G}_{j_3} \mathbf{G}_{j_2} \mathbf{G}_{j_1} \begin{bmatrix} \mathbf{R} \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & & + \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}} \\ 0 \end{bmatrix}$$

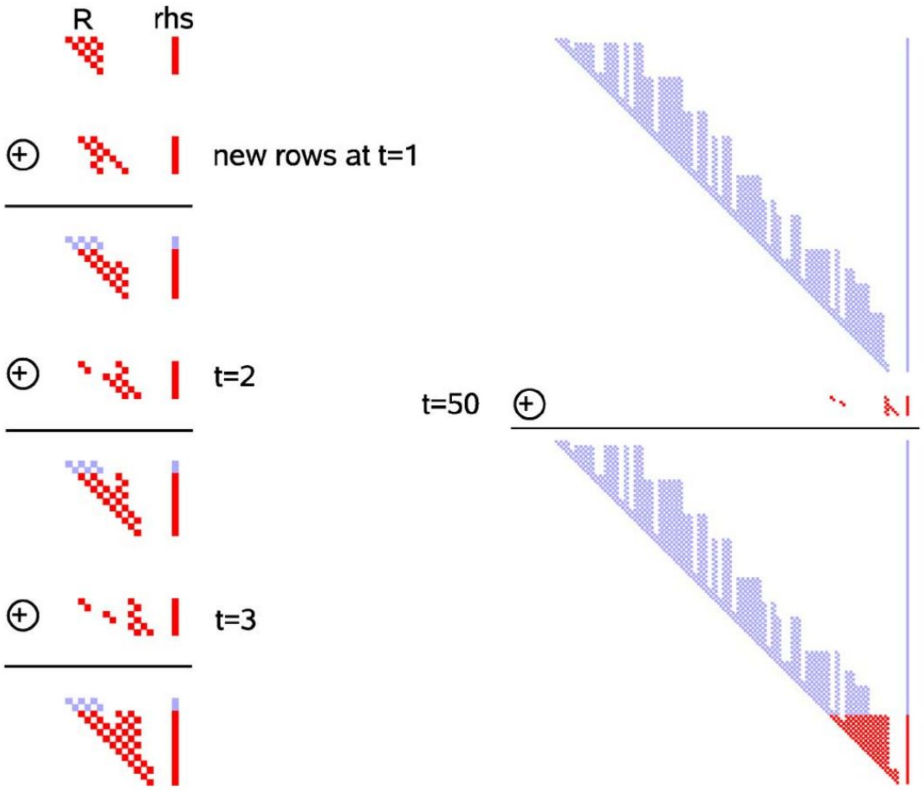
$$\Rightarrow \begin{bmatrix} \mathbf{A} \\ \mathbf{w}^T \end{bmatrix} = \tilde{\mathbf{A}} = \underbrace{\begin{bmatrix} Q & \\ & 1 \end{bmatrix} \mathbf{G}_{j_1}^T \mathbf{G}_{j_2}^T \mathbf{G}_{j_3}^T \mathbf{G}_{j_4}^T \mathbf{G}_{j_5}^T}_{\tilde{\mathbf{Q}}} \begin{bmatrix} \tilde{\mathbf{R}} \\ \mathbf{0} \end{bmatrix}$$

Get solution from updated square root info matrix

At each time step, SLAM solution can be obtained via back-substitution:

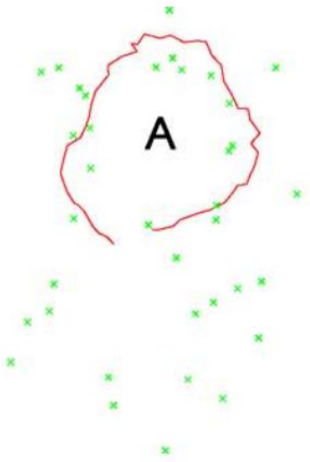
$$\tilde{\mathbf{R}}_t \tilde{\boldsymbol{\theta}}_t^* = \tilde{\mathbf{d}}_t$$

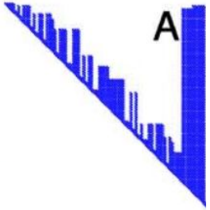
Pictorial QR-updating example over a couple of time steps:



Name of the game: keep it sparse!

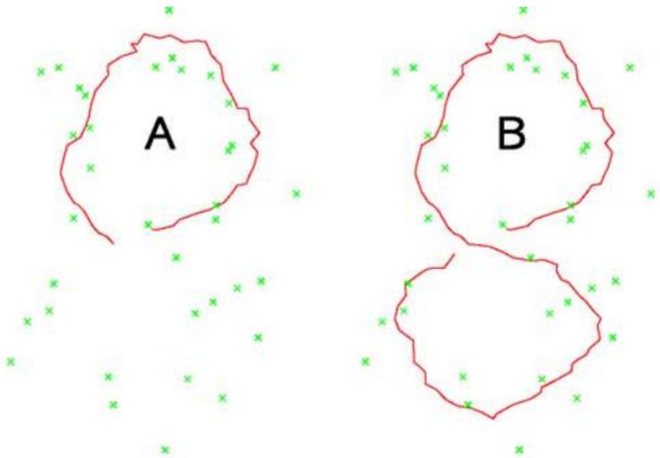
Significant fill-in can occur during incremental factorization

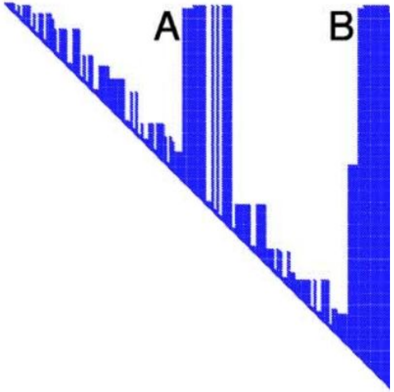




Name of the game: keep it sparse!

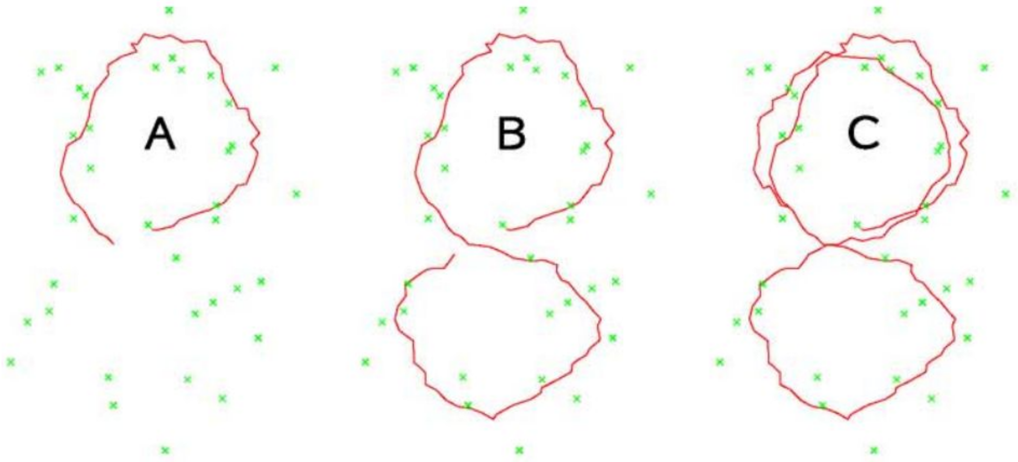
Significant fill-in can occur during incremental factorization

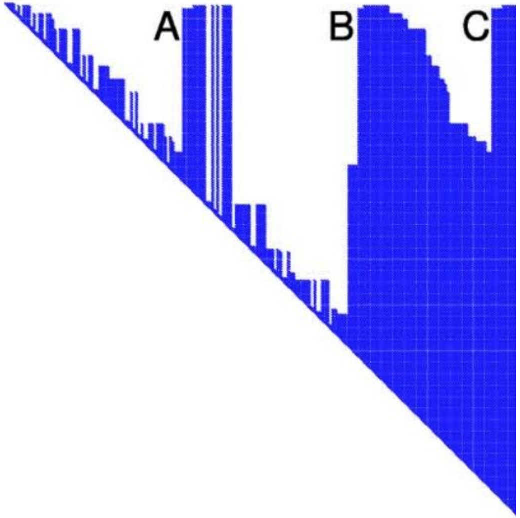




Name of the game: keep it sparse!

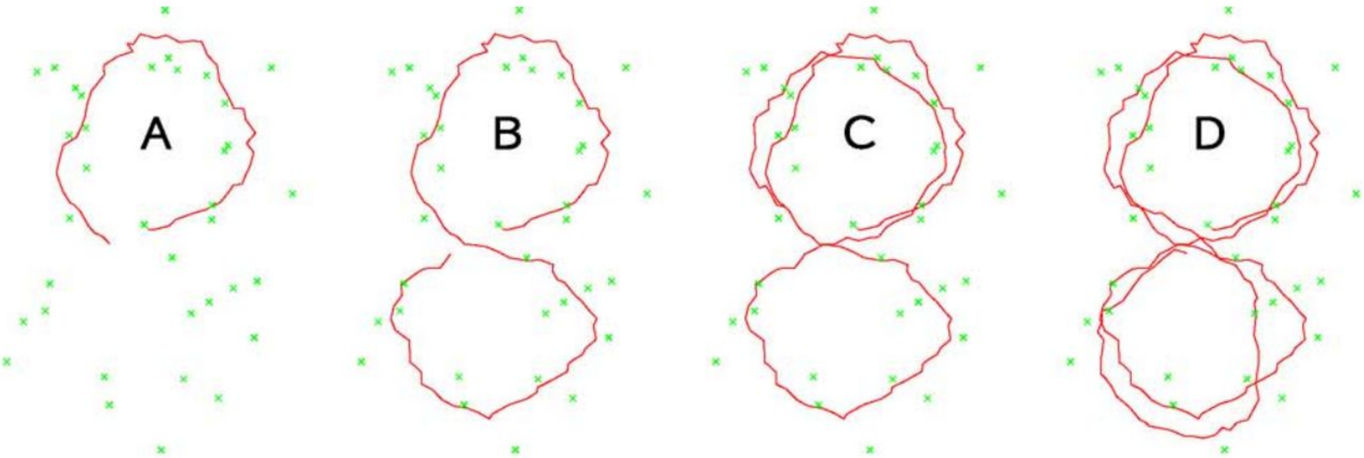
Significant fill-in can occur during incremental factorization

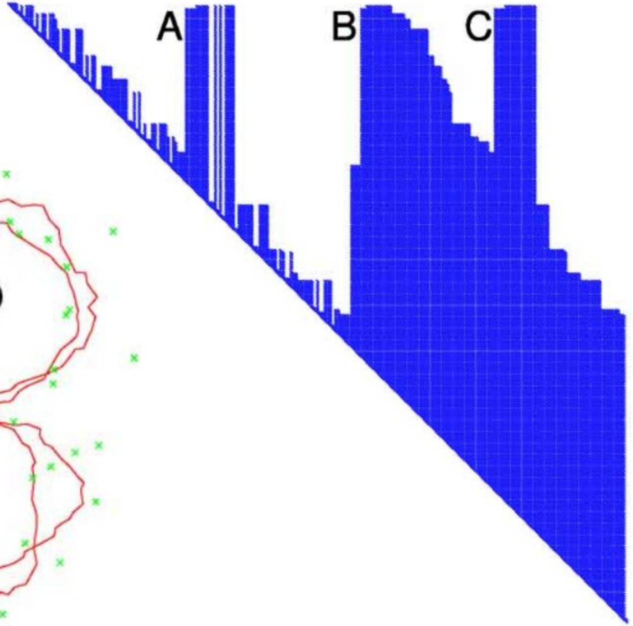




Name of the game: keep it sparse!

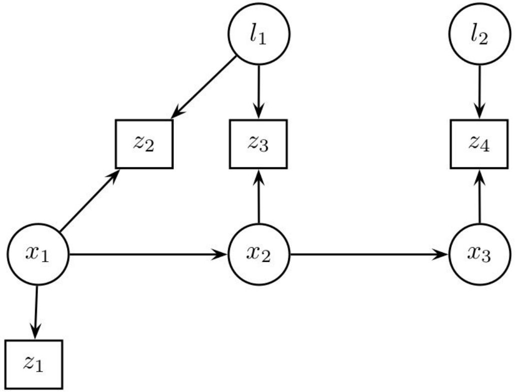
Significant fill-in can occur during incremental factorization





Variable ordering is important

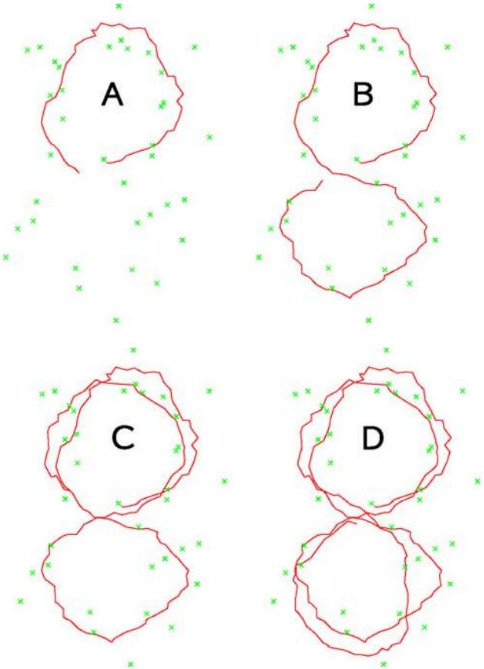
Revisiting the toy example:

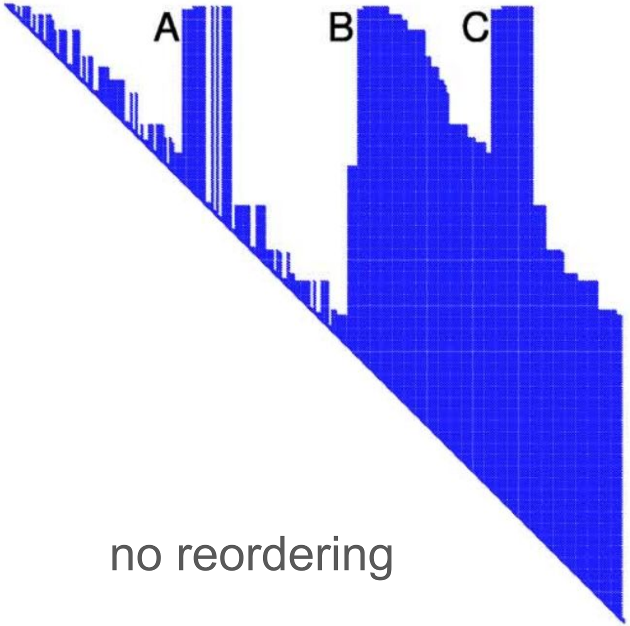


$$\begin{array}{c} \begin{array}{ccccc} l_1 & l_2 & x_1 & x_2 & x_3 \end{array} \\ \begin{bmatrix} R_1 & & T_{13} & T_{14} & \\ & R_2 & & & T_{25} \\ & & R_3 & T_{34} & \\ & & & R_4 & T_{45} \\ & & & & \mathbf{R}_5 \end{bmatrix} \end{array} \quad \text{vs} \quad \begin{array}{c} \begin{array}{ccccc} x_3 & x_2 & x_1 & l_2 & l_1 \end{array} \\ \begin{bmatrix} R_1 & T_{12} & & T_{14} & \\ & R_2 & T_{23} & T_{24} & T_{25} \\ & & R_3 & T_{34} & T_{35} \\ & & & R_4 & T_{45} \\ & & & & R_5 \end{bmatrix} \end{array}$$

How to obtain best column variable ordering?

NP hard problem, but good heuristics exist, such as COLAMD (*col*umn *a*pproximate *m*inimum *d*egree), which iSAM uses:

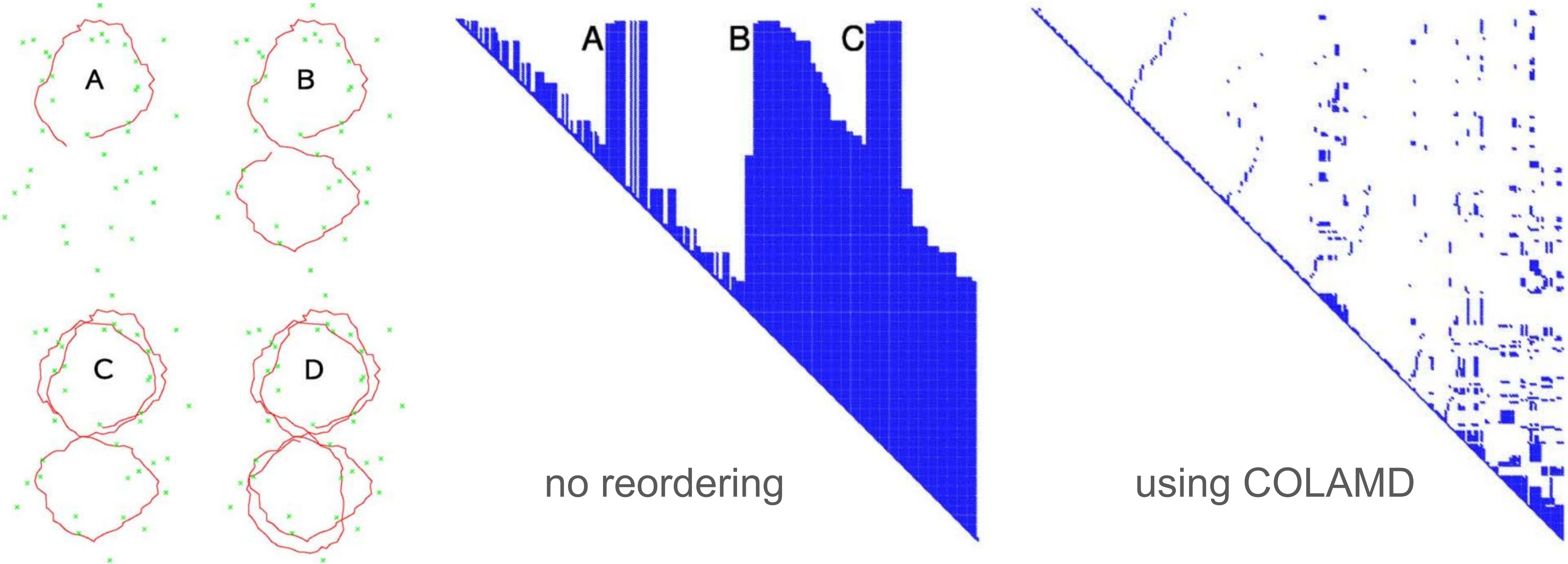




no reordering

How to obtain best column variable ordering?

NP hard problem, but good heuristics exist, such as COLAMD (*col*umn *a*pproximate *m*inimum *d*egree), which iSAM uses:



no reordering

using COLAMD

Data association approach

Objective: assign each measurement to the correct landmark.

Maximum likelihood (ML) formulation: minimum cost assignment problem using a Mahalanobis distance.

Mahalanobis distance based on projection of combined pose and landmark uncertainties:

$$\Xi = J\Sigma J^T + \Gamma$$

Σ: combined pose and landmark uncertainties

Marginal covariances

$$\Sigma = \begin{bmatrix} \Sigma_{jj} & \Sigma_{ij}^T \\ \Sigma_{ij} & \Sigma_{ii} \end{bmatrix}$$

for pose $\mathbf{x}_i$ and landmark $\mathbf{l}_j$

Marginal covariances

$$\Sigma = \begin{bmatrix} \Sigma_{jj} & \Sigma_{ij}^T \\ \Sigma_{ij} & \Sigma_{ii} \end{bmatrix}$$ for pose $\mathbf{x}_i$ and landmark $\mathbf{l}_j$

$\Longrightarrow$ Reorder variables and place latest pose $\mathbf{x}_i$ in the last column of matrix $\mathbf{R}$

Solve $R^T RX = B$, with $B = \begin{bmatrix} 0_{(n-d_\mathrm{x})\times d_\mathrm{x}} \\ I_{d_\mathrm{x}} \times d_\mathrm{x} \end{bmatrix}$ by forward and back substitution:

$$R^T Y = B, \quad RX = Y$$

Structure uncertainties

$$\Sigma = \begin{bmatrix} \boxed{\Sigma_{jj}} & \Sigma_{ij}^T \\ \Sigma_{ij} & \Sigma_{ii} \end{bmatrix}$$

for pose $\mathbf{x}_i$ and landmark $\mathbf{l}_j$

$\Rightarrow$ Conservative estimates using current pose uncertainty:

$$\tilde{\Sigma}_{jj} = \bar{J} \begin{bmatrix} \Sigma_{ii} & \\ & \Gamma \end{bmatrix} \bar{J}^T$$

Compute exact covariance using efficient method for all nonzero entries σ_{ij}:

$$\sigma_{ll} = \frac{1}{r_{ll}} \left(\frac{1}{r_{ll}} - \sum_{j=l+1, r_{lj} \neq 0}^{n} r_{lj} \sigma_{jl} \right), \quad \sigma_{il} = \frac{1}{r_{ii}} \left(- \sum_{j=i+1, r_{ij} \neq 0}^{l} r_{ij} \sigma_{jl} - \sum_{j=l+1, r_{ij} \neq 0}^{n} r_{ij} \sigma_{lj} \right)$$

Examples

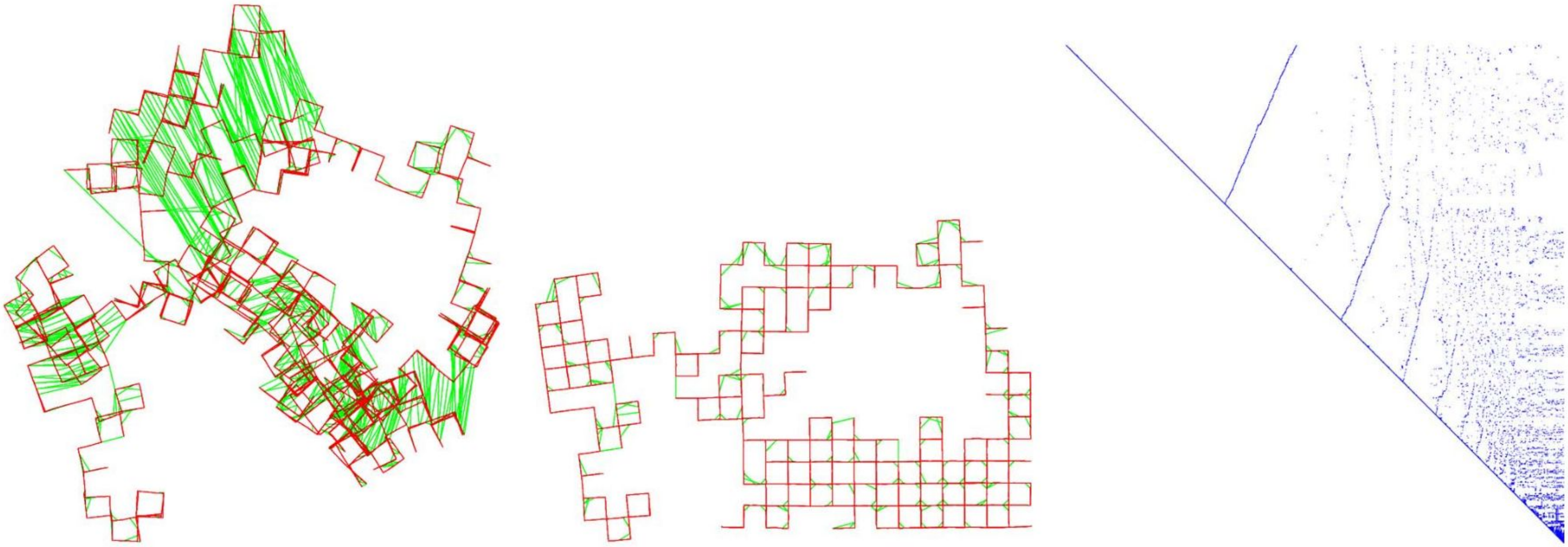
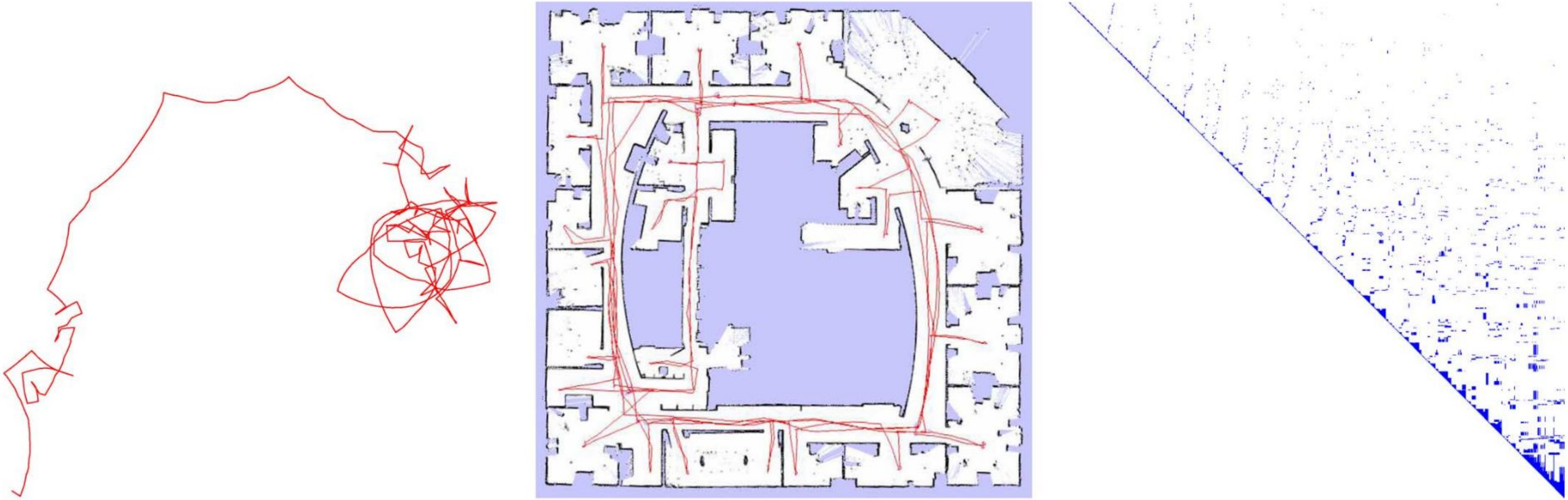

(a) (b) (c)

Fig. 9. iSAM results for the simulated Manhattan world from [21] takes about 40 ms/step. The resulting R factor has 187 423 entries, which corresponds to 0.34% or an average of 17.8 entries per column. (a) Original noisy dataset. (b) Trajectory after incremental optimization. (c) Final R factor with side length 10 500.

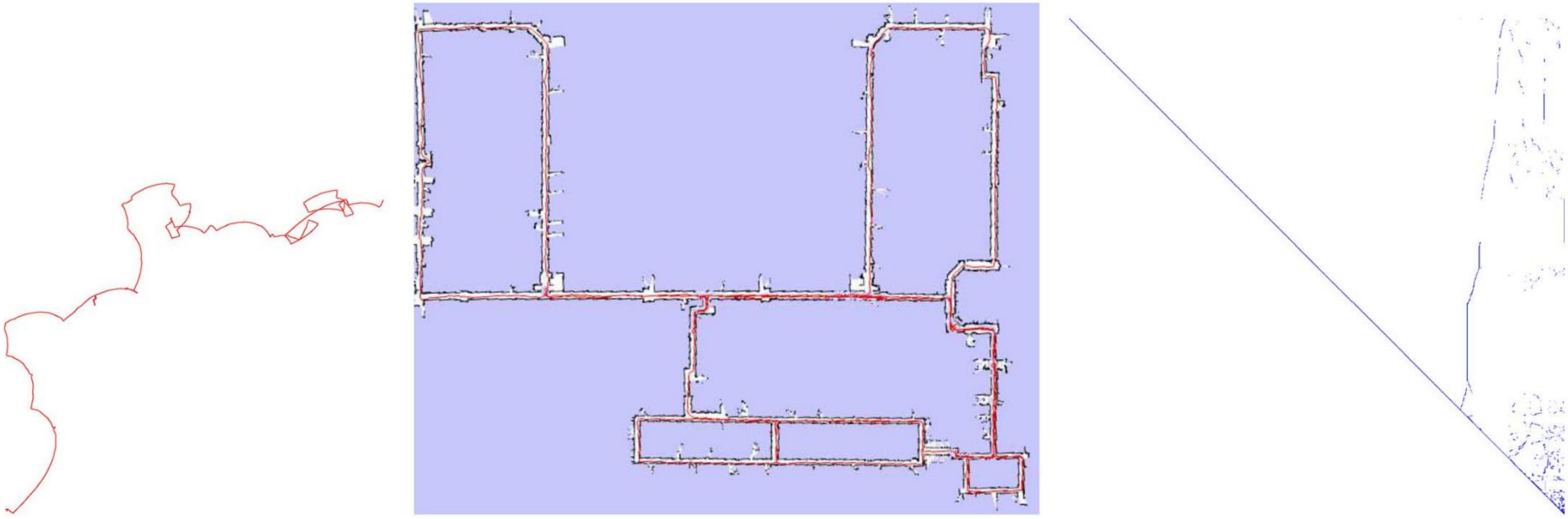
Examples



(a) (b) (c)

Fig. 10. Results from iSAM applied to the Intel dataset. iSAM calculates the full solution for 910 poses and 4453 constraints with an average of 85 ms/step while reordering the variables every 20 steps. The problem has $910 \times 3 = 2730$ variables and $4453 \times 3 = 13\,359$ measurement equations. The R factor contains 90 363 entries, which corresponds to 2.42% or 33.1 entries per column. (a) Trajectory based on odometry only. (b) Final trajectory and evidence grid map. (c) Final R factor with side length 2730.

Examples



(a) (b) (c)

Fig. 11. iSAM results for the MIT Killian Court dataset. iSAM calculates the full solution for the 1941 poses and 2190 pose constraints with an average of 12.2 ms per step. The R factor contains 52 414 entries for 5823 variables, which corresponds to 0.31% or 9.0 per column. (a) Trajectory based on odometry only. (b) Final trajectory and evidence grid map. (c) Final R factor with side length 5823.

Thanks