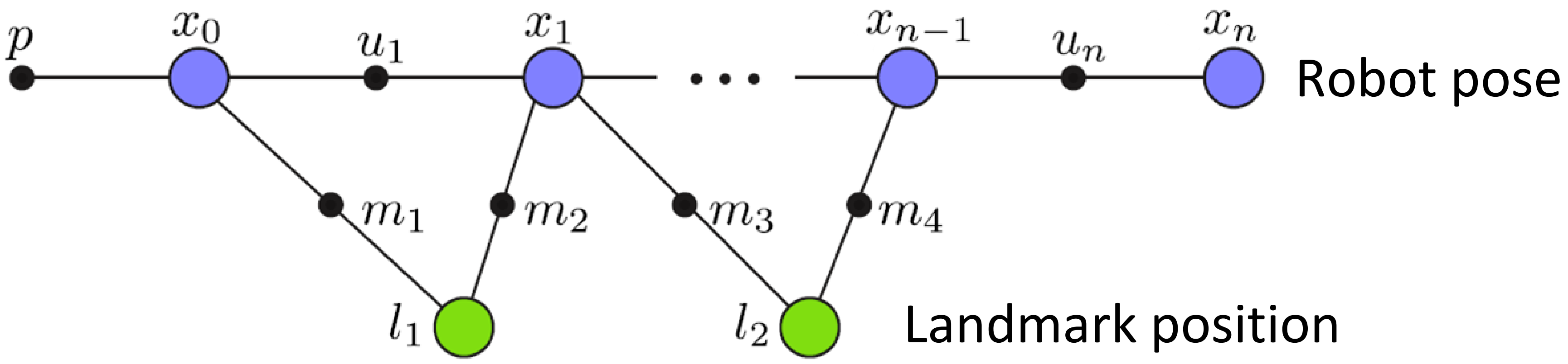
SLAM: Exploiting Sparsity

Robot Localization and Mapping
16-833

Michael Kaess

October 4+7, 2024

Factor Graph Representation of SLAM

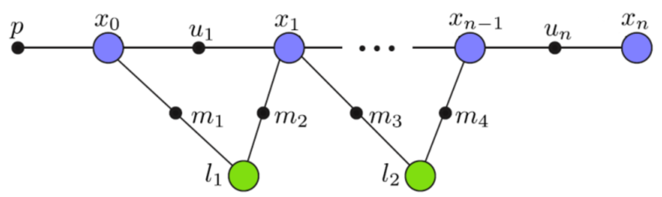


Variables: $\Theta = \{x_0, x_1 \cdots x_n, l_1, l_2\}$

Measurements: $Z = \{p, u_1 \cdots u_n, m_1 \cdots m_4\}$

Factorization: $\mathrm{p}(\mathrm{Z}|\Theta) = \mathrm{argmax}_\Theta \prod_{z \in Z} p(z|\Theta)$

SLAM as a Least-Squares Problem



$$\operatorname{argmax}_\Theta \prod_{z \in Z} p\left(z|\Theta\right)$$

Gaussian noise

$$\operatorname{argmin}_\Theta \sum_i \|h_i(\Theta) - z_i\|_\Sigma^2$$

$h()$ linear

$$\operatorname{argmin}_x \|Ax - b\|^2$$

Normal equations:

$$A^T A x = A^T b$$

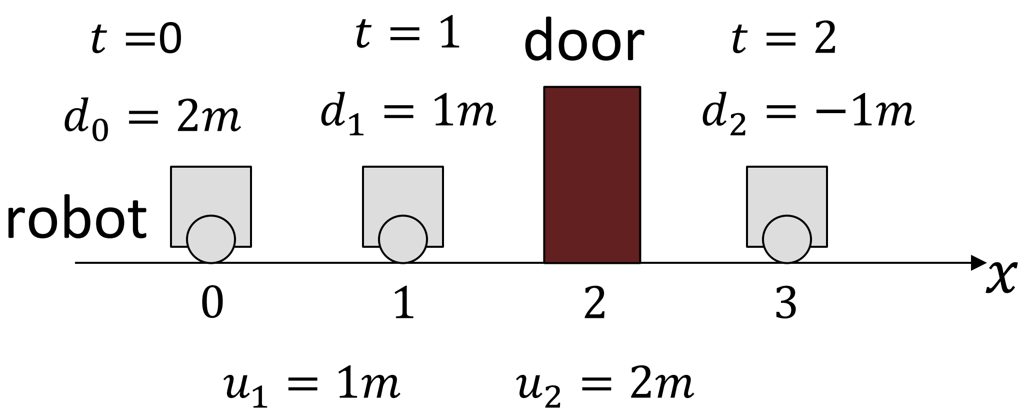
Solving for θ by matrix inversion is too expensive!

SLAM as a Least-Squares Problem: Example

- On the board:
  - Linear 1D example

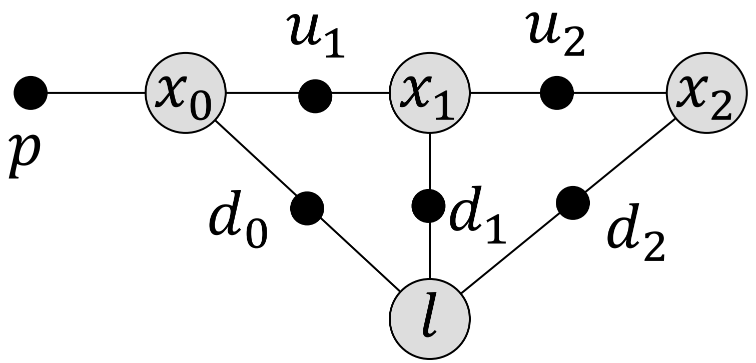
SLAM Least-Squares Example

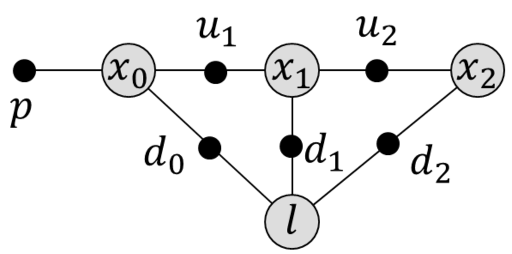
Localize robot and door based on 1D range measurements



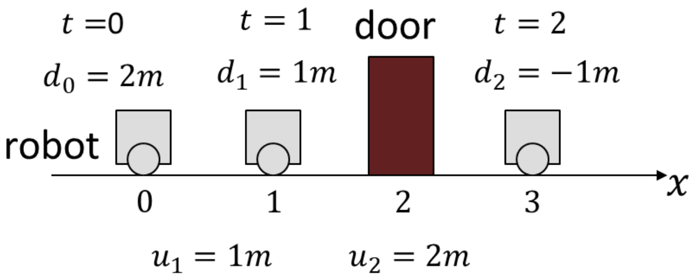
Measurements: distance to the door, signed

Factor graph:





u_1
u_2
p
x_0
x_1
x_2
d_0
d_1
d_2
l

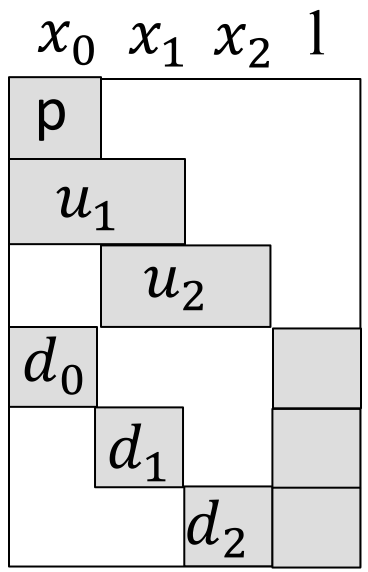


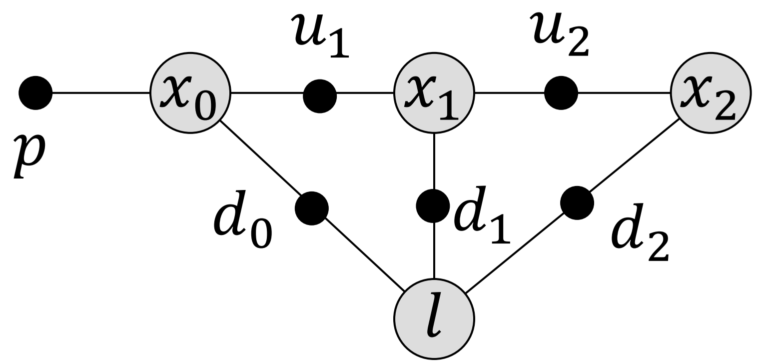
$t = 0$
$t = 1$
door
$t = 2$
$d_0 = 2m$
$d_1 = 1m$
$d_2 = -1m$
robot
0
1
2
3
x
$u_1 = 1m$
$u_2 = 2m$

SLAM Least-Squares Example

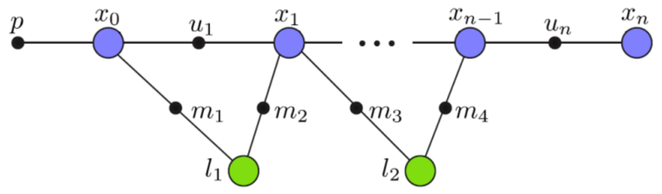
Localize robot and door based on 1D range measurements

Matrix A:
Each row corresponds to a factor
Each column to a variable
A is sparse!





SLAM as a Sparse Least-Squares Problem



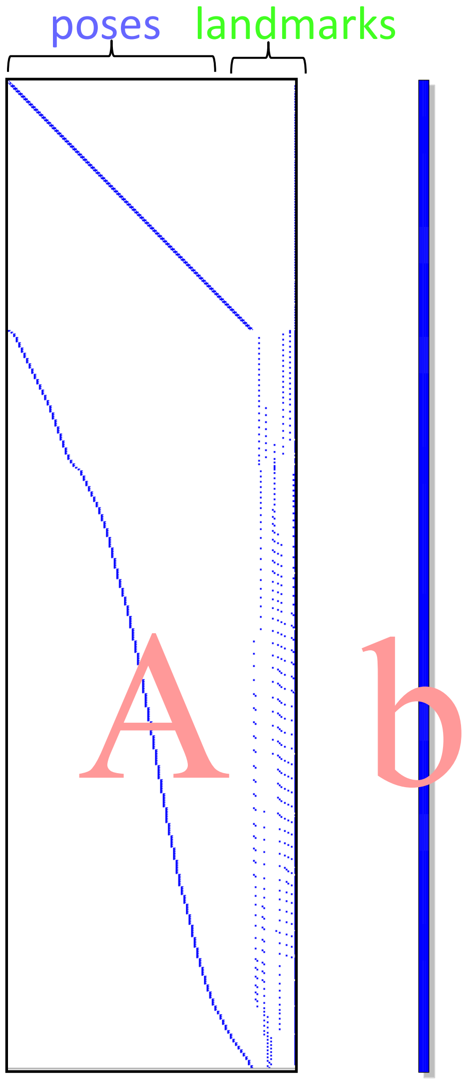
$$\operatorname{argmax}_\Theta \prod_{z \in Z} p\left(z|\Theta\right)$$

Gaussian noise

$$\operatorname{argmin}_\Theta \sum_i \|h_i(\Theta)\|^2$$

$$\operatorname{argmin}_x \|Ax - b\|^2$$

Sparse measurement Jacobian



Efficient Solution

- On the board:
  - Sparse matrix factorization
  - Solving by backsubstitution

Efficient Solution: Cholesky Factorization

Cholesky factor R is an upper triangular matrix so that

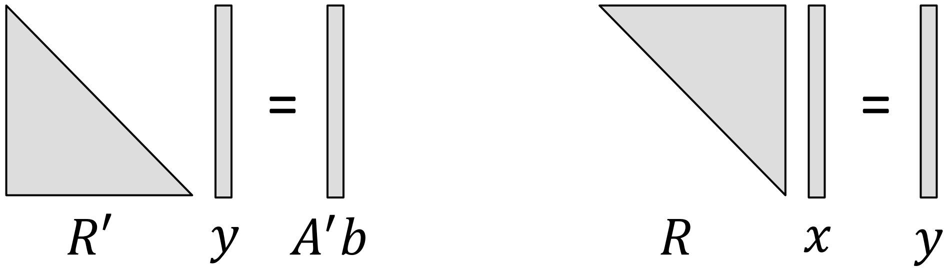
$$R'R = A'A$$

Yielding

$$R'Rx = A'b$$

Solve by forward-/backsubstitution

$$R'y = A'b$$
$$Rx = y$$



Similar: LDL' factorization, faster than Cholesky, avoids square roots

Efficient Solution: QR Factorization

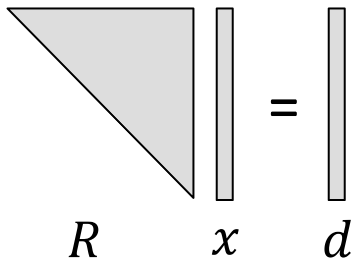
$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$$

Yielding

$$\|Ax - b\|^2 = \left\| Q \begin{bmatrix} R \\ 0 \end{bmatrix} x - b \right\|^2 = \left\| Q'Q \begin{bmatrix} R \\ 0 \end{bmatrix} x - \begin{bmatrix} d \\ e \end{bmatrix} \right\|^2 = \|Rx - d\|^2 + \|e\|^2$$

Solve by backsubstitution

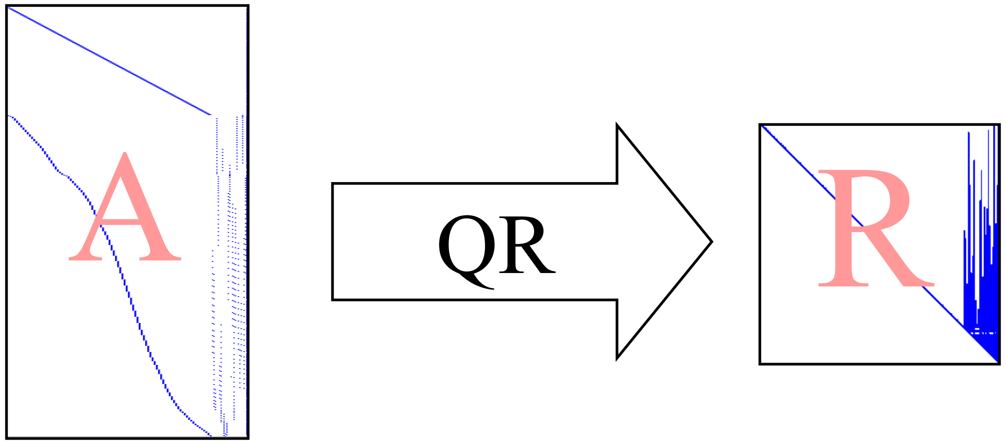
$$Rx = d$$



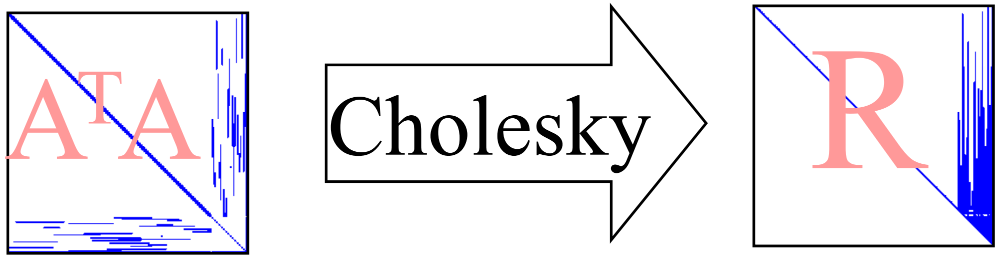
Note that in practice Q is never explicitly formed.

Matrix Factorization

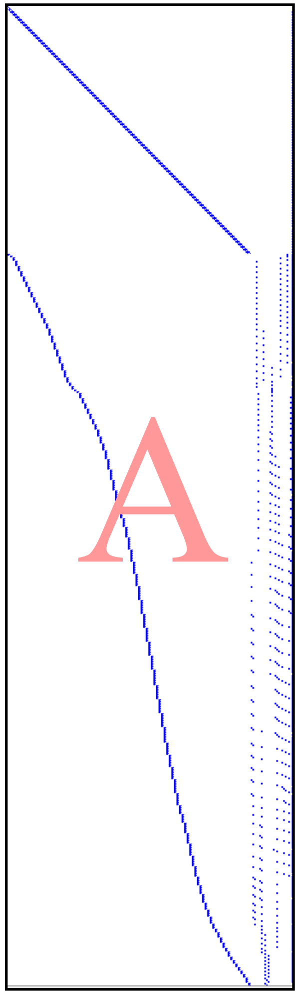
- QR on A: Numerically more stable



- Cholesky on A^TA: Faster



Solving the Sparse Linear Least-Squares System

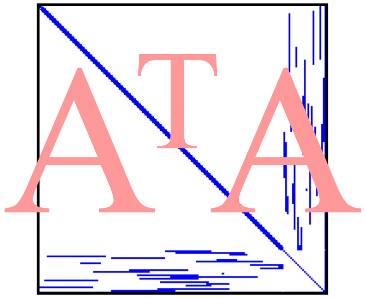


Measurement Jacobian

Solve: $\operatorname{argmin}_{\theta} \|A\theta - b\|^2$

Normal equations

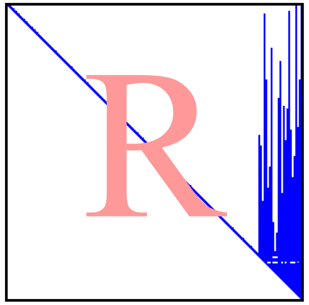
$$A^T A \theta = A^T b$$



Information matrix

Matrix factorization

$$A^T A = R^T R$$



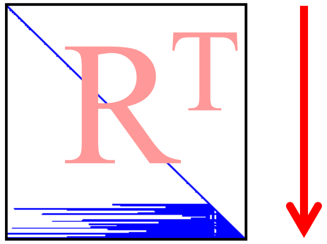
Square root information matrix

Solving by Backsubstitution

After factorization: $R^T R\, x = A^T b$

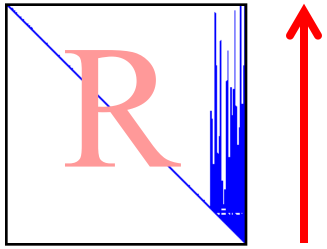
- Forward substitution

  $R^T y = A^T b$, solve for y



- Backsubstition

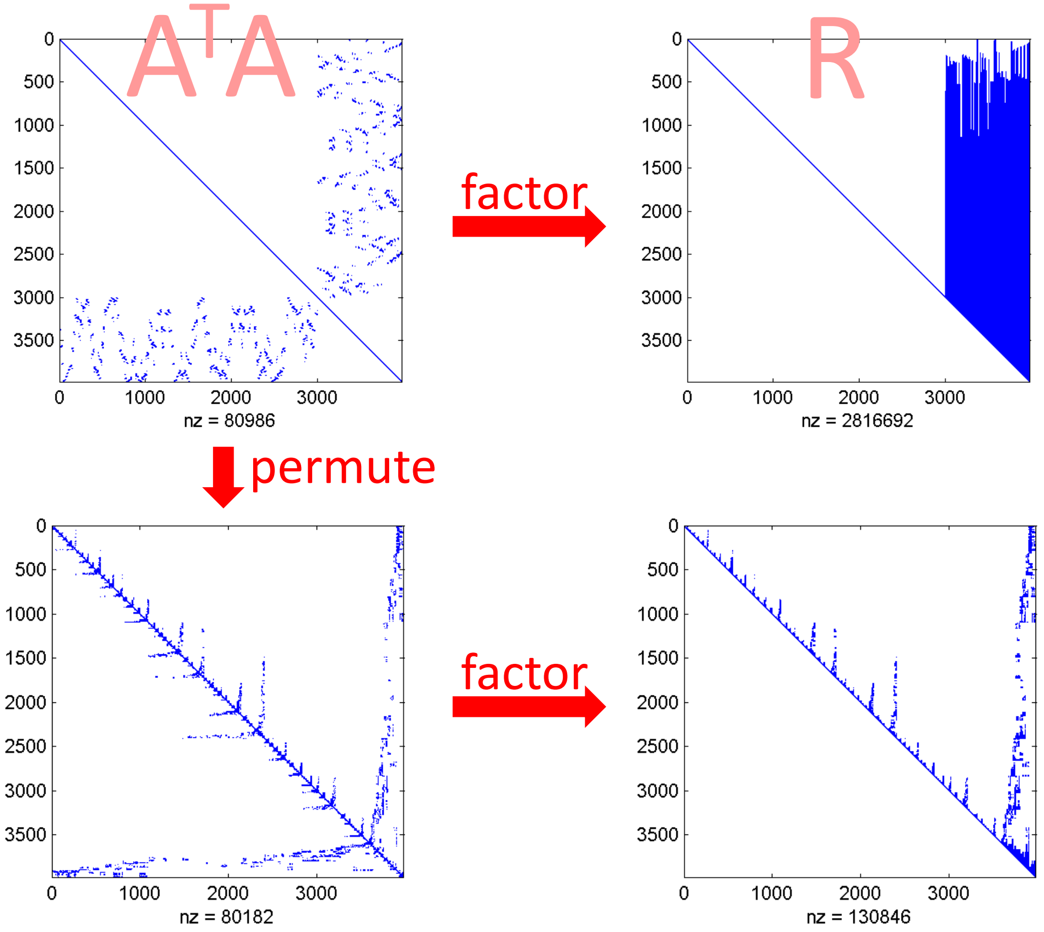
  R x = y, solve for x



[Dellaert and Kaess, IJRR 06]

Retaining Sparsity: Variable Ordering

Fill-in depends on elimination order:



Default ordering

(poses, landmarks)

Ordering based on

COLAMD heuristic [Davis04]

(best order: NP hard)

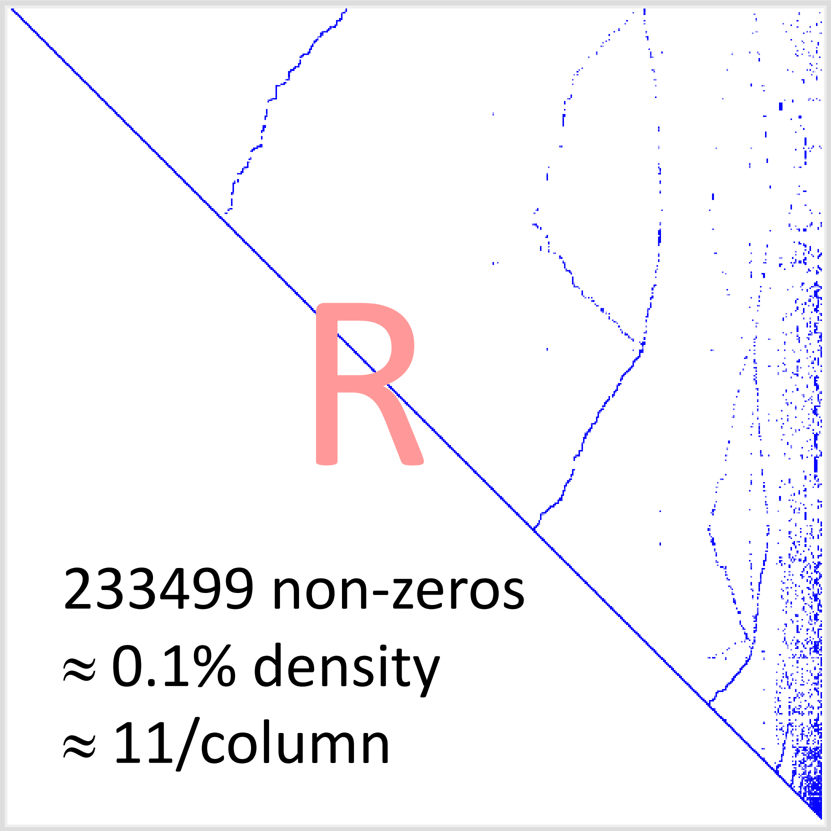
Sparse Factorization Example

Example from real sequence:

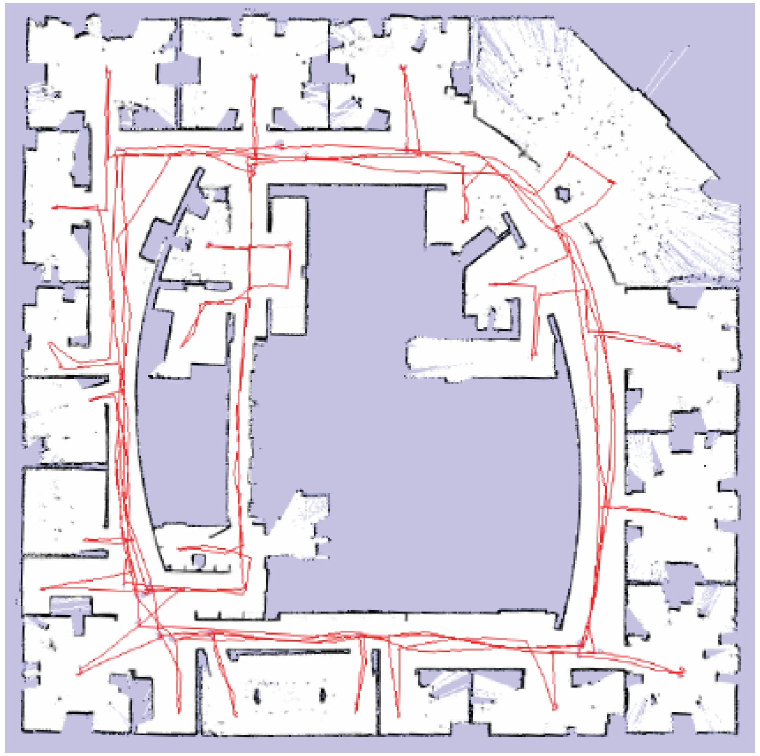
Square root inf. matrix

Side length: 21000 variables

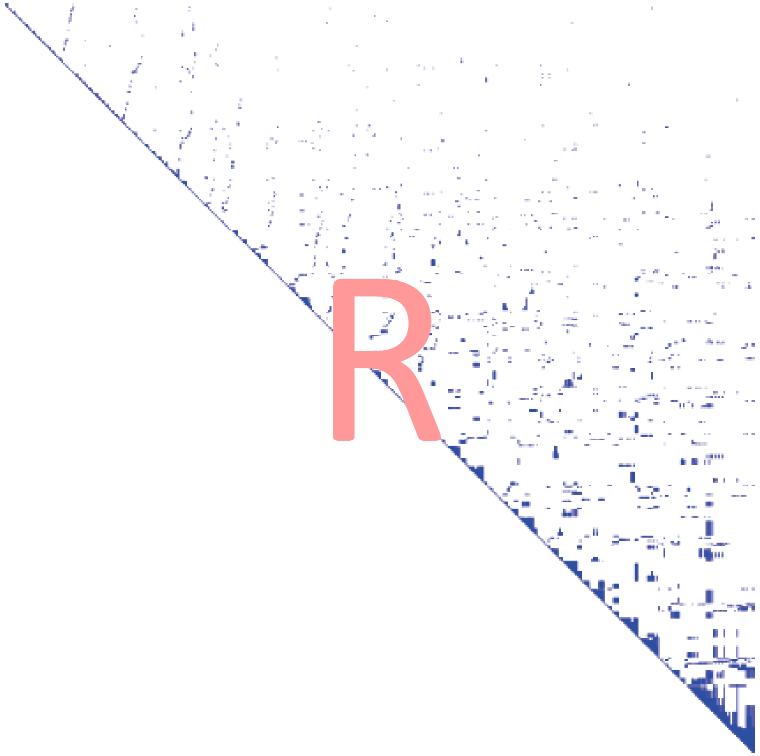
Dense: 1.7GB, sparse: 1MB



Example 2 - Standard Intel Dataset



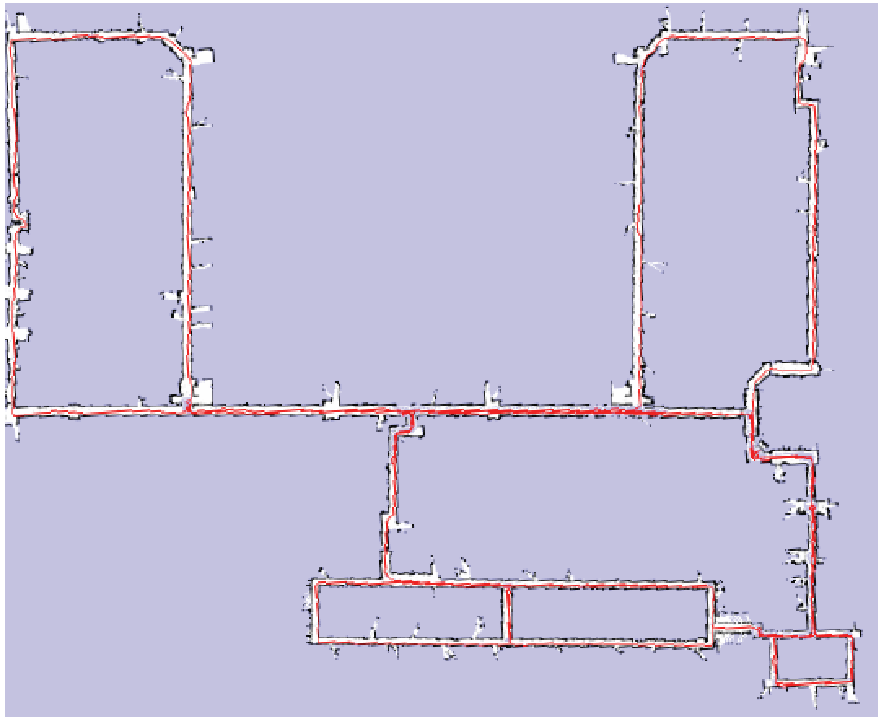
(b) Final trajectory and evidence grid map.



(c) Final R factor with side length 2730.

910 poses, 4453 constraints

Example 3 - MIT Killian Court Dataset



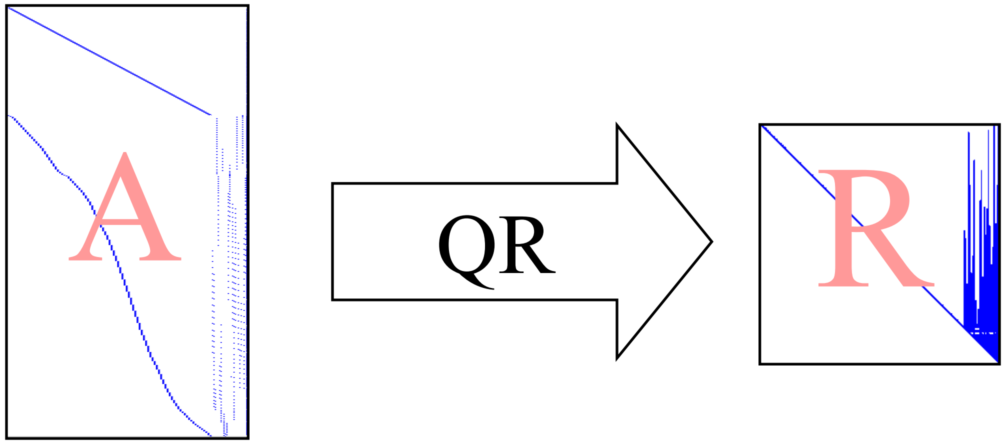
(b) Final trajectory and evidence grid map.

(c) Final R factor with side length 5823.

1941 poses, 2190 constraints

Questions

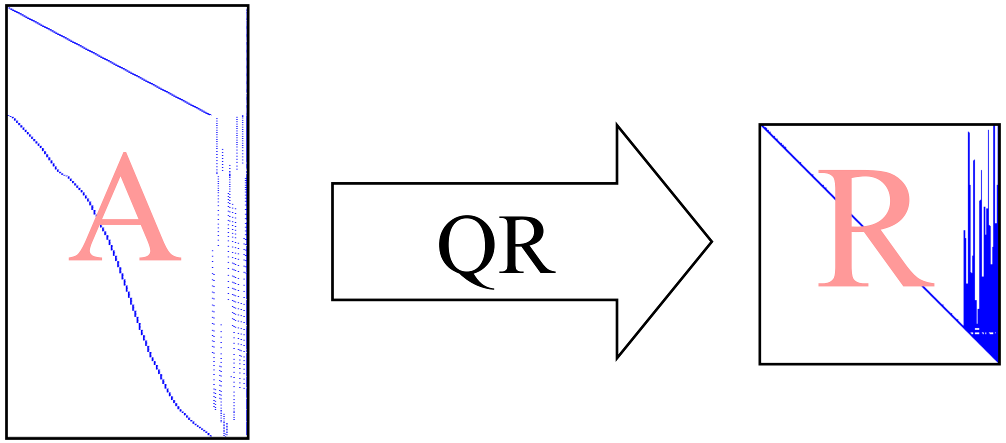
QR on A:



- Does the order of the rows of A impact fill-in?

Questions

QR on A:



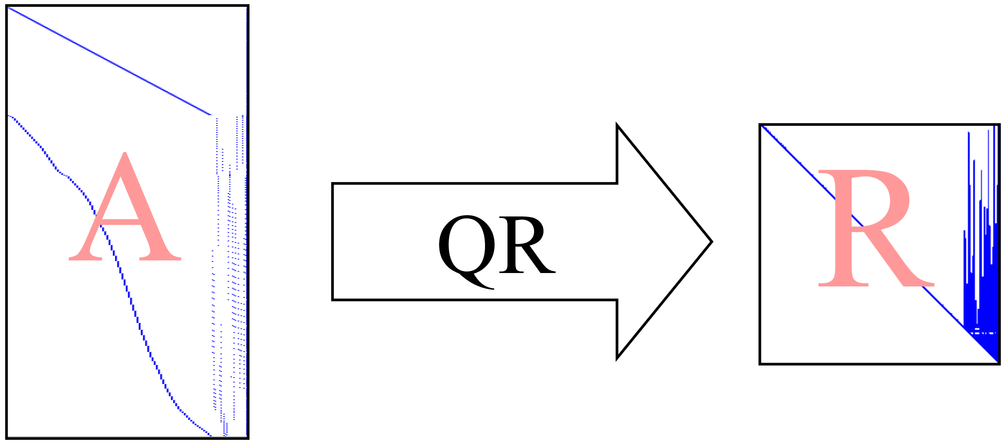
- Does the order of the rows of A impact fill-in?

  No!

- Does the order of the columns of A impact fill-in?

Questions

QR on A:



- Does the order of the rows of A impact fill-in?

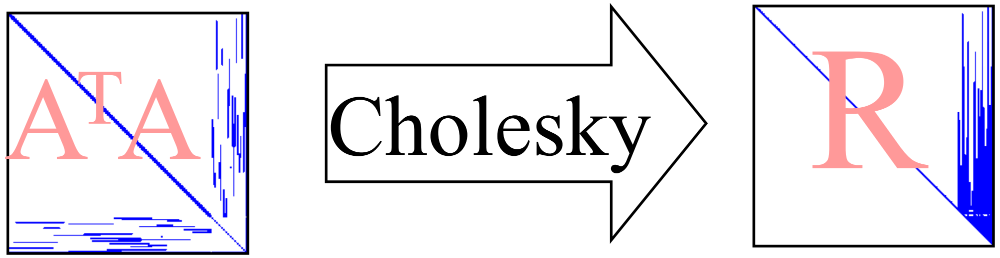
  No!

- Does the order of the columns of A impact fill-in?

  Yes, the order will influence fill-in in R and therefore efficiency!

Questions

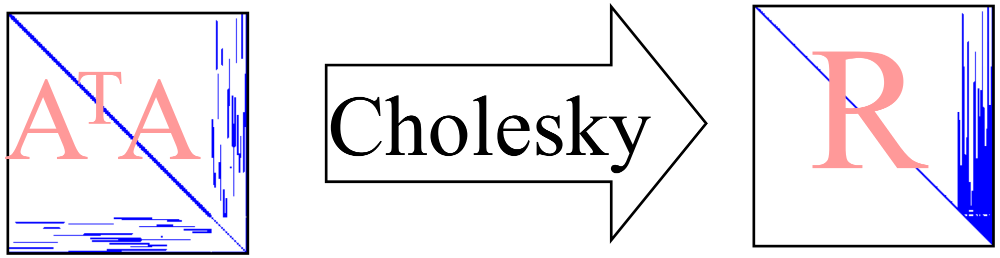
Cholesky on A^TA



- Does the order of the rows of A^TA impact fill-in?

Questions

Cholesky on A^TA



- Does the order of the rows of A^TA impact fill-in?

  The information matrix is symmetric, have to permute both rows and columns at the same time!

  The order of rows and columns does impact fill-in in R.

Summary

- Exploiting sparsity is key to efficiency
- Sparse matrix factorization avoids explicitly calculating the (dense) inverse matrix

- Next: Nonlinear Least-Squares
  Factor Graphs for Robot Perception: 2.5