Probabilistic Robotics Course

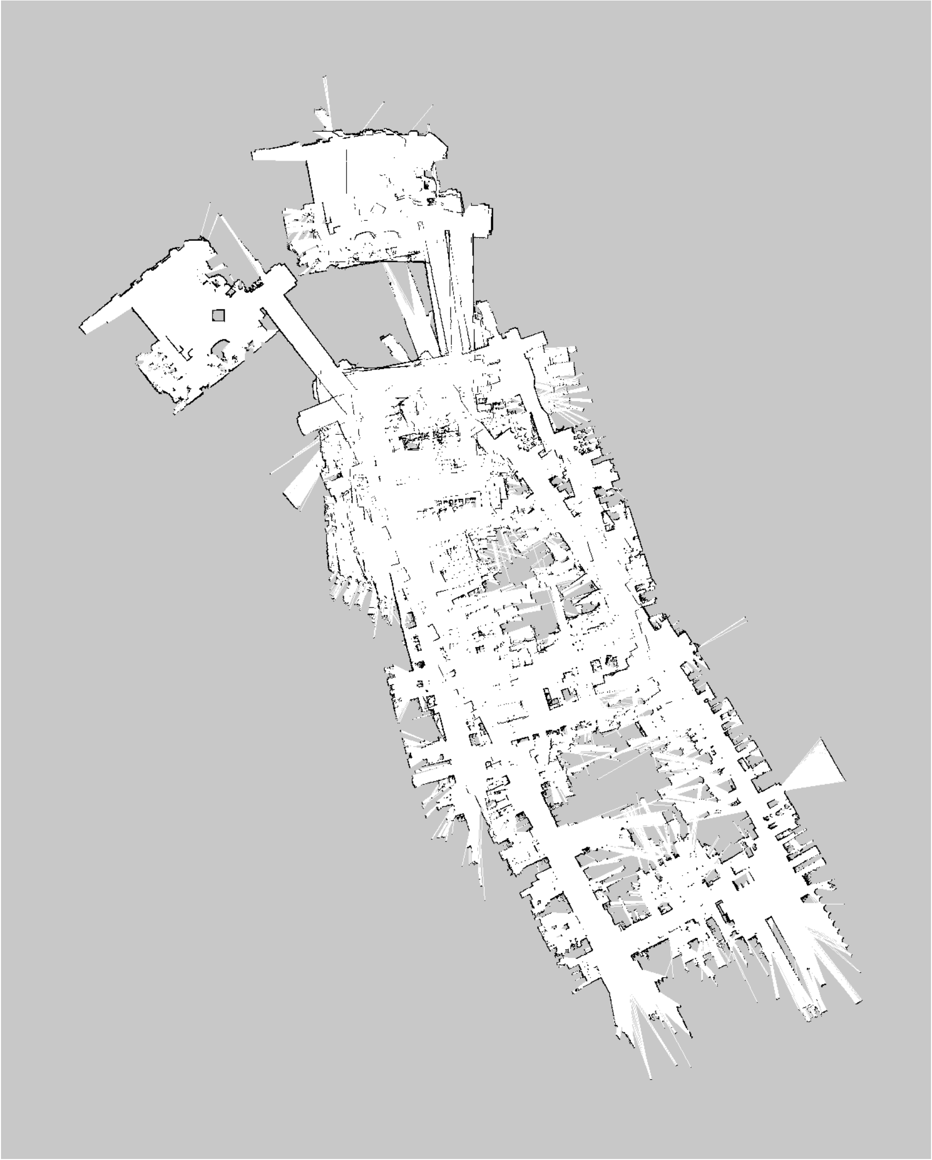
Multi-Pose Registration and Graph-SLAM

Giorgio Grisetti
grisetti@diag.uniroma1.it

Department of Computer, Control, and Management Engineering
Sapienza University of Rome

Graph-Based SLAM in a Nutshell

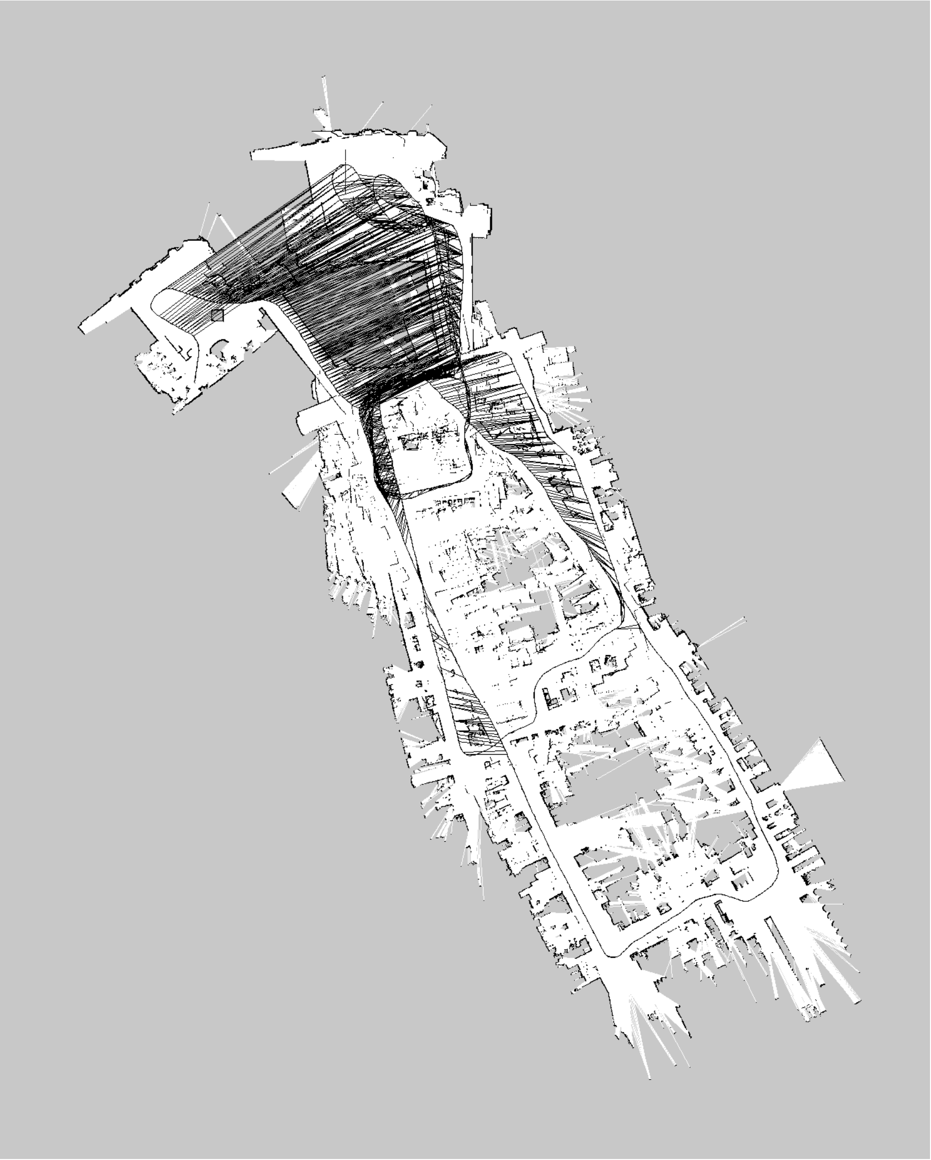
- Problem described as a graph
  - Every node corresponds to a robot position and to a laser measurement
  - An edge between two nodes represents a data-dependent spatial constraint between the nodes



KUKA Halle 22, courtesy of the P. Pfaff

Graph-Based SLAM in a Nutshell

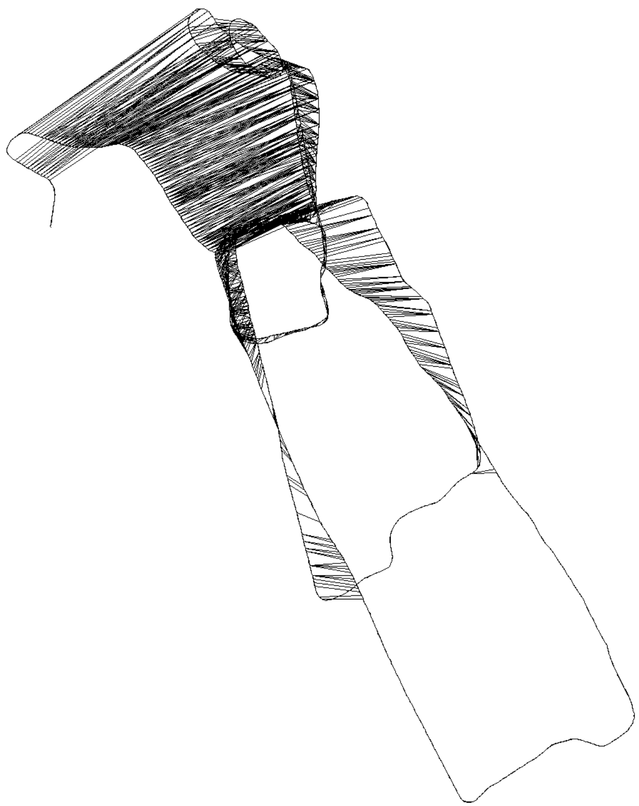
- Problem described as a graph
  - Every node corresponds to a robot position and to a laser measurement
  - An edge between two nodes represents a data-dependent spatial constraint between the nodes



KUKA Halle 22, courtesy of the P. Pfaff

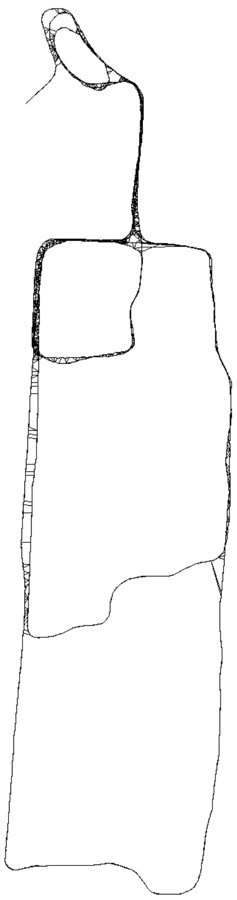
Graph-Based SLAM in a Nutshell

- Once we have the graph we determine the most likely map by “moving” the nodes



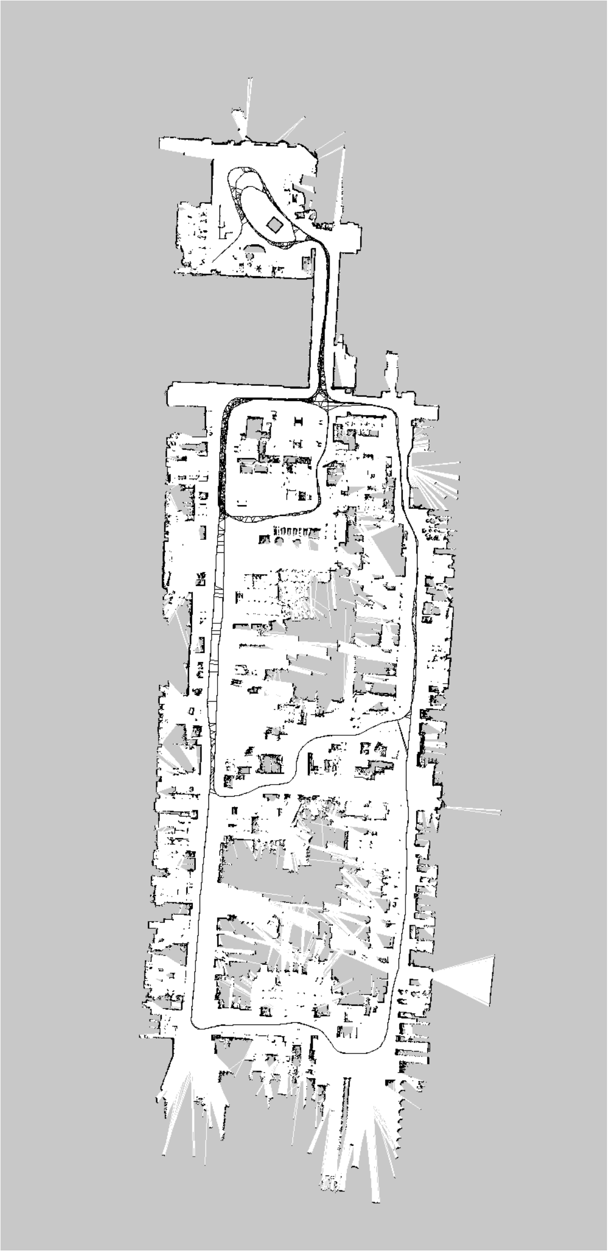
Graph-Based SLAM in a Nutshell

- Once we have the graph we determine the most likely map by “moving” the nodes
- … like this



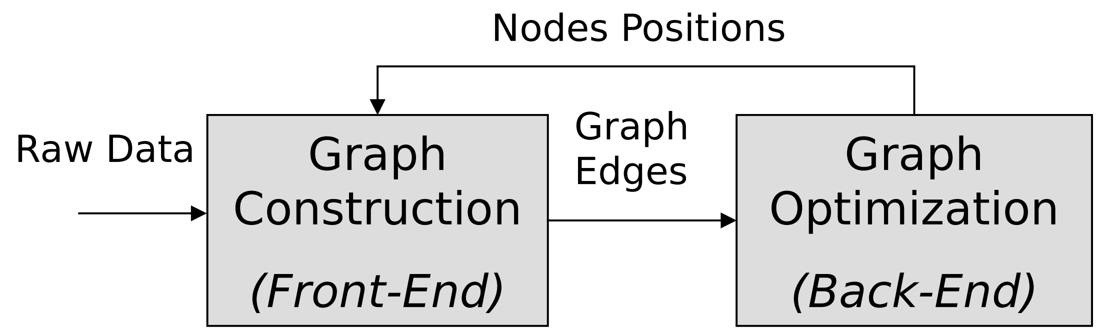
Graph-Based SLAM in a Nutshell

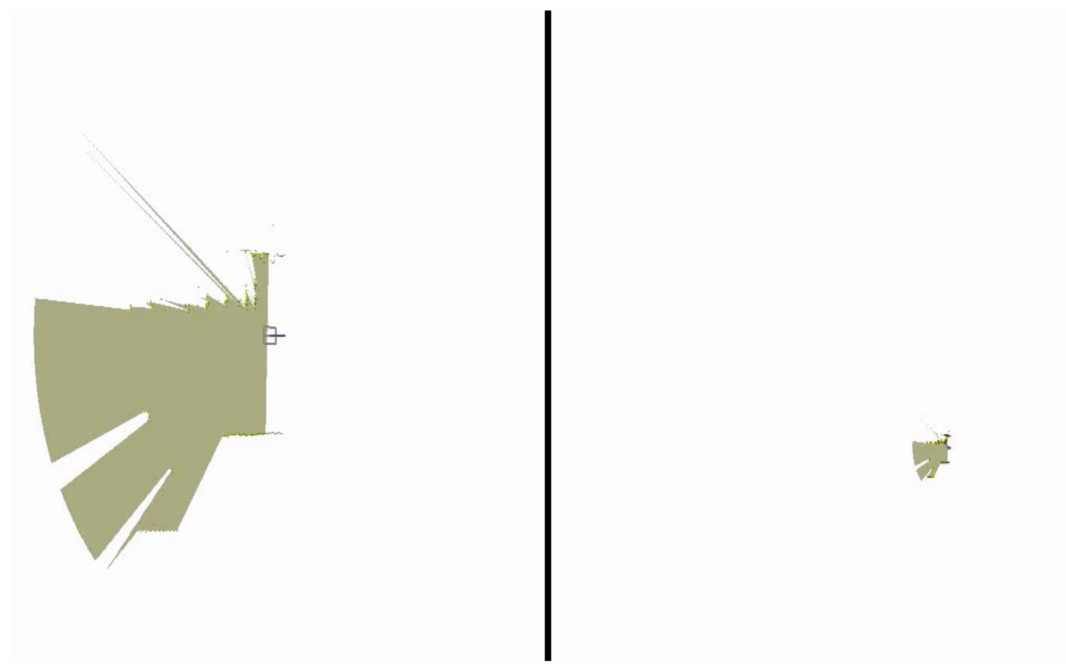
- Once we have the graph we determine the most likely map by "moving" the nodes
- ... like this
- Then, we can render a map based on the known poses



Graph Optimization

- In this lecture, we will ***not*** address the how to construct the graph but how to retrieve the position of its nodes which is maximally consistent the observations in the edges.
- A general Graph-based SLAM algorithm interleaves the two steps
  - Graph construction
  - Graph optimization
- A consistent map helps in determining the new constraints by reducing the search space.



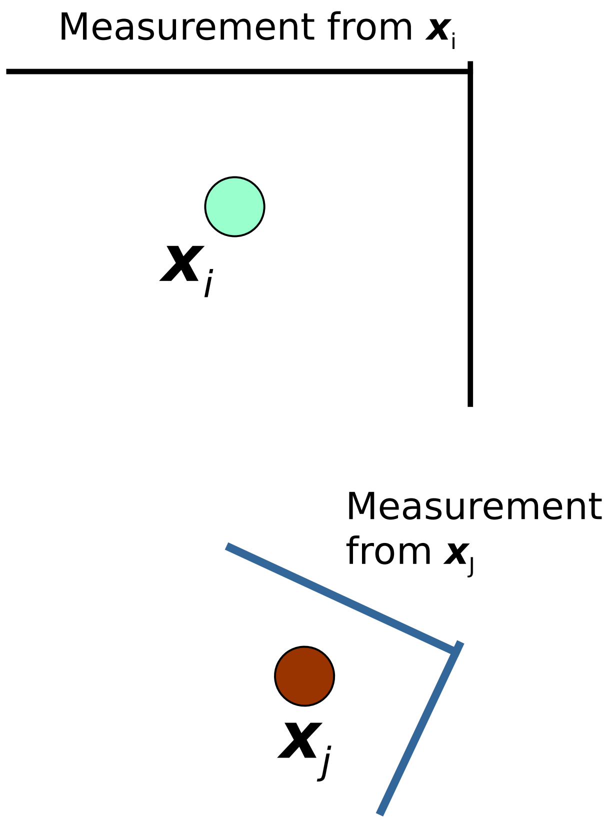


What Does the Graph Look Like?

- It has $\boldsymbol{n}$ nodes $\boldsymbol{x}=\boldsymbol{x}_{1:n}$
  - Each node $\boldsymbol{x}_i$ is a 2D or 3D transformation representing the pose of the robot at time $\boldsymbol{t}_i$.
- There is a constraint e_{ij} between the node $\boldsymbol{x}_i$ and the node $\boldsymbol{x}_j$ if
  - either
    - the robot observed the same part of the environment from both $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and,
    - via this common observation it constructs a “virtual measurement” about the position of $\boldsymbol{x}_j$ seen from.
  - Or
    - the positions are subsequent in time and there is an odometry measurement between the two.

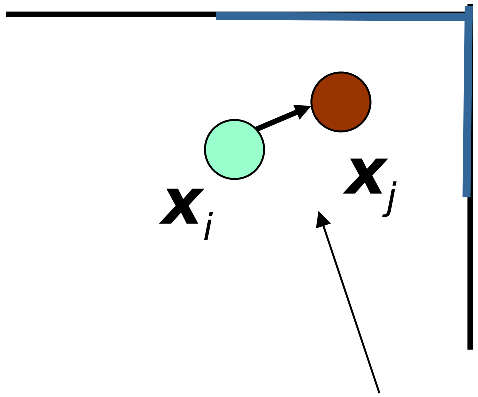
What Does the Graph Look Like?

- It has ***n*** nodes $\boldsymbol{x}=\boldsymbol{x}_{1:n}$
  - Each node $\boldsymbol{x}_i$ is a 2D or 3D transformation representing the pose of the robot at time $\boldsymbol{t}_i$.
- There is a constraint e_{ij} between the node $\boldsymbol{x}_i$ and the node $\boldsymbol{x}_j$ if
  - either
    - the robot observed the same part of the environment from both $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and,
    - via this common observation it constructs a "virtual measurement" about the position of $\boldsymbol{x}_j$ seen from.
  - Or
    - the positions are subsequent in time and there is an odometry measurement between the two.



What Does the Graph Look Like?

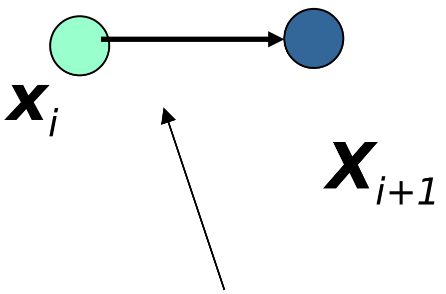
- It has ***n*** nodes $\boldsymbol{x}=\boldsymbol{x}_{1:n}$
  - Each node $\boldsymbol{x}_i$ is a 2D or 3D transformation representing the pose of the robot at time $\boldsymbol{t}_i$.
- There is a constraint e_{ij} between the node $\boldsymbol{x}_i$ and the node $\boldsymbol{x}_j$ if
  - either
    - the robot observed the same part of the environment from both $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and,
    - via this common observation it constructs a “virtual measurement” about the position of $\boldsymbol{x}_j$ seen from.
  - Or
    - the positions are subsequent in time and there is an odometry measurement between the two.



The edge represents the position of $\boldsymbol{x}_j$ seen from $\boldsymbol{x}_i$, based on the ***observations***

What Does the Graph Look Like?

- It has $\boldsymbol{n}$ nodes $\boldsymbol{x}=\boldsymbol{x}_{1:n}$
  - Each node $\boldsymbol{x}_i$ is a 2D or 3D transformation representing the pose of the robot at time $\boldsymbol{t}_i$.
- There is a constraint e_{ij} between the node $\boldsymbol{x}_i$ and the node $\boldsymbol{x}_j$ if
  - either
    - the robot observed the same part of the environment from both $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and,
    - via this common observation it constructs a "virtual measurement" about the position of $\boldsymbol{x}_j$ seen from.
  - Or
    - the positions are subsequent in time and there is an odometry measurement between the two.



The edge represents the **odometry** measurement

The Edge Information Matrices

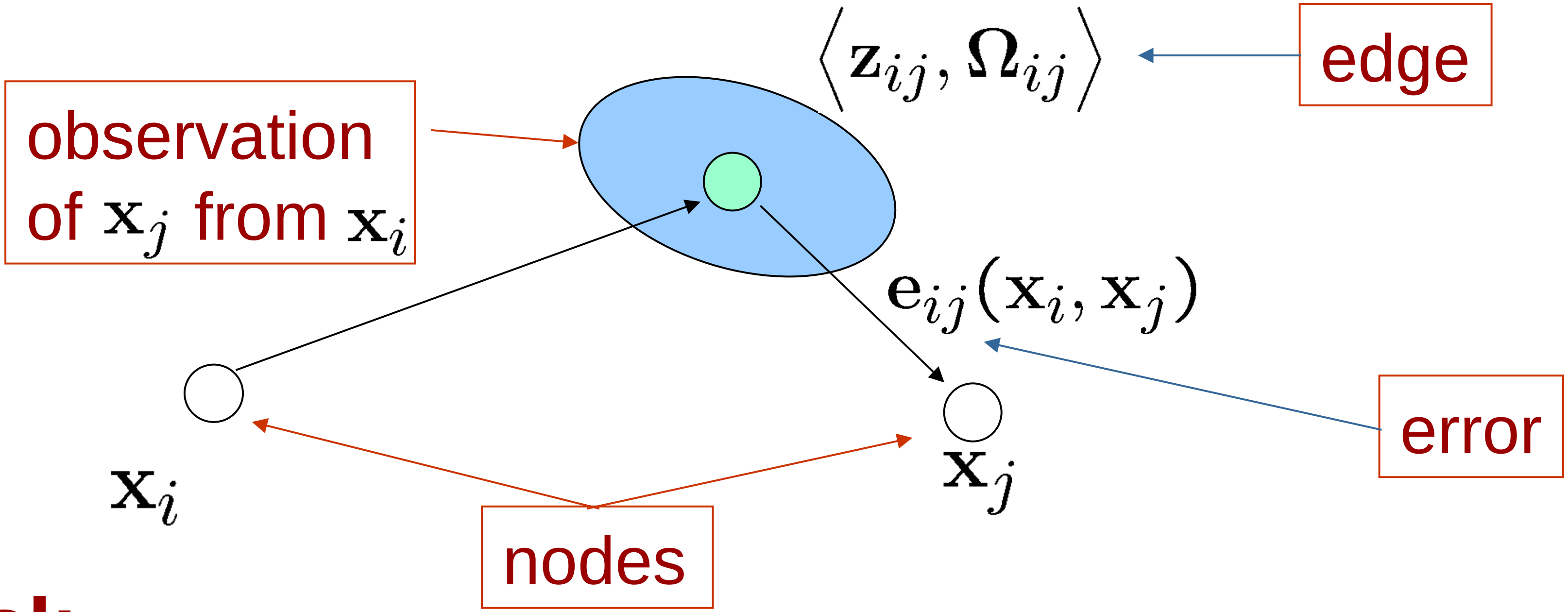
- To account for the different nature of the observations we add to the edge an information matrix $\boldsymbol{\Omega}_{ij}$ to encode the uncertainty of the edge.
- The "bigger" (in matrix sense) $\boldsymbol{\Omega}_{ij}$ is, the more the edge "matters" in the optimization procedure.

Questions:

- Any idea about the information matrices of the system in case we use scan-matching and odometry?
- What should these matrices look like in an endless corridor in both cases?

Pose Graph

- The input for the optimization procedure is a graph annotated as follows:



- **Goal:**
  - Find the assignment of poses to the nodes of the graph which minimizes the negative log likelihood of the observations:

$$\hat{\mathbf{x}} = \operatorname{argmin} \sum_{ij} \mathbf{e}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij}$$

State

The state is a collection of robot poses

$$\mathbf{X} \quad : \quad \mathbf{X} = \{\mathbf{X}_r^{[1]}, \ldots, \mathbf{X}_r^{[N]}\}$$

$$\mathbf{X}_r^{[n]} \in SE(3) \quad : \quad \mathbf{X}^{[n]} = \left(\mathbf{R}^{[n]} \middle| \mathbf{t}^{[n]}\right)$$

The increments are represented by a large vector containing the minimal perturbation for each state variable

$$\mathbf{\Delta x} \in \Re^{6N} \quad : \quad \mathbf{\Delta x} = \left(\mathbf{\Delta x}_r^{[1]T}, \ldots, \mathbf{\Delta x}_r^{[N]T}\right)^T$$

$$\mathbf{\Delta x}_r^{[n]T} \in \Re^6 \quad : \quad \mathbf{\Delta x}_r^{[n]T} = (\underbrace{\Delta x^{[n]} \ \Delta y^{[n]} \ \Delta z^{[n]}}_{\mathbf{\Delta t}^{[n]}} \ \underbrace{\Delta \alpha_x^{[n]} \ \Delta \alpha_y^{[n]} \ \Delta \alpha_z^{[n]}}_{\Delta \alpha^{[n]}})^T$$

Boxplus

The boxplus has to be adapted to apply the individual perturbations for each variable block

$$\mathbf{X}' = \mathbf{X} \boxplus \mathbf{\Delta x}$$

$$\begin{aligned} \mathbf{X}_r^{[n]'} &= \mathbf{X}_r^{[n]} \boxplus \mathbf{\Delta x}_r^{[n]} \\ &= v2t(\mathbf{\Delta x}_r^{[n]})\mathbf{X}_r^{[n]} \end{aligned}$$

Measurements and Predictions

A measurement of the robot pose *j*, performed from robot pose *i* is as follows

$$\mathbf{Z}^{[i,j]} \in SE(3) \quad : \quad \mathbf{Z}^{[i,j]} = \left(\mathbf{R}^{[i,j]} \middle| \mathbf{t}^{[i,j]}\right)$$

The prediction and the error of is the boxminus between prediction and measurement

$$\begin{aligned}
\mathbf{h}^{[i,j]}(\mathbf{X}) &= \mathbf{X}_r^{[i]-1}\mathbf{X}_r^{[j]} \\
\mathbf{e}^{[i,j]}(\mathbf{X}) &= \mathbf{X}_r^{[i]-1}\mathbf{X}_r^{[j]} \boxminus \mathbf{Z}^{[i,j]} \\
\mathbf{e}^{[i,j]}(\mathbf{X} \boxplus \mathbf{\Delta x}) &= t2v\left(\mathbf{Z}^{[i,j]-1}\left(v2t(\mathbf{\Delta x}_r^{[i]})\mathbf{X}_r^{[i]}\right)^{-1}\left(v2t(\mathbf{\Delta x}_r^{[j]})\mathbf{X}_r^{[j]}\right)\right) \\
&= t2v\left(\mathbf{Z}^{[i,j]-1}\mathbf{X}_r^{[i]-1}v2t(\mathbf{\Delta x}_r^{[i]})^{-1}v2t(\mathbf{\Delta x}_r^{[j]})\mathbf{X}_r^{[j]}\right)
\end{aligned}$$

Jacobians

The prediction depends only on the observing and the observed robot poses so it will be mostly 0

not easy to compute with pen and paper

$$
\begin{aligned}
\frac{\partial \mathbf{e}^{[i,j]}(\mathbf{X} \boxplus \mathbf{\Delta x})}{\partial \mathbf{\Delta x}} &= \left(\cdots \mathbf{0}_{6\times 6} \cdots \frac{\partial \mathbf{e}^{[i,j]}(\mathbf{X} \boxplus \mathbf{\Delta x})}{\partial \mathbf{\Delta x}_r^i} \cdots \mathbf{0}_{6\times 6} \cdots \frac{\partial \mathbf{e}^{[i,j]}(\mathbf{X} \boxplus \mathbf{\Delta x})}{\partial \mathbf{\Delta x}_r^j} \cdots \mathbf{0}_{6\times 6} \cdots \right) \\
\left.\frac{\partial \mathbf{e}^{[i,j]}(\mathbf{X} \boxplus \mathbf{\Delta x})}{\partial \mathbf{\Delta x}_r^i}\right|_{\mathbf{\Delta x}_r^{[i]}=\mathbf{0}} &= \mathbf{J}_i^{[i,j]} \\
\left.\frac{\partial \mathbf{e}^{[i,j]}(\mathbf{X} \boxplus \mathbf{\Delta x})}{\partial \mathbf{\Delta x}_r^j}\right|_{\mathbf{\Delta x}_r^{[j]}=\mathbf{0}} &= \mathbf{J}_j^{[i,j]} \\
\mathbf{J}^{[i,j]} &= \left(\cdots \mathbf{0}_{6\times 6} \cdots \mathbf{J}_i^{[i,j]} \cdots \mathbf{0}_{6\times 6} \cdots \mathbf{J}_j^{[i,j]} \cdots \mathbf{0}_{6\times 6} \cdots \right)
\end{aligned}
$$

Information Matrix

The measurements live on a non-Euclidean space, we need to handle the Information Matrices

$$\hat{\mathbf{Z}}^{[i,j]} = \mathbf{X}_r^{[i]-1}\mathbf{X}_r^{[j]}$$

prediction

$$\mathbf{J}_{\mathbf{e}}^{[i,j]} = \left.\frac{\hat{\mathbf{Z}}^{[i,j]} \boxminus \mathbf{Z}}{\partial \mathbf{Z}}\right|_{\mathbf{Z}=\mathbf{Z}^{[i,j]}}$$

derivative of error w.r.t measurement

$$\tilde{\mathbf{\Omega}}_r^{[i,j]} \leftarrow \left(\mathbf{J}_{\mathbf{e}}^{[i,j]} \mathbf{\Omega}^{[i,j]-1} \mathbf{J}_{\mathbf{e}}^{[i,j]^T}\right)^{-1}$$

Adapted Information matrix for one iteration

H Matrix and B vector

H and b for a measurement have only few non zero blocks

$$
\begin{aligned}
\mathbf{H}^{[i,j]} &= \mathbf{J}^{[i,j]T}\tilde{\mathbf{\Omega}}^{[i,j]}\mathbf{J}^{[i,j]} \\
&= \begin{pmatrix}
 & \vdots & & \vdots & \\
\cdots & \mathbf{J}_i^{[i,j]T}\tilde{\mathbf{\Omega}}^{[i,j]}\mathbf{J}_i^{[i,j]} & \cdots & \mathbf{J}_i^{[i,j]T}\tilde{\mathbf{\Omega}}^{[i,j]}\mathbf{J}_j^{[i,j]} & \cdots \\
 & \vdots & & \vdots & \\
\cdots & \mathbf{J}_j^{[i,j]T}\tilde{\mathbf{\Omega}}^{[i,j]}\mathbf{J}_i^{[i,j]} & \cdots & \mathbf{J}_j^{[i,j]T}\tilde{\mathbf{\Omega}}^{[i,j]}\mathbf{J}_j^{[i,j]} & \cdots \\
 & \vdots & & \vdots &
\end{pmatrix} \\
\mathbf{b}^{[i,j]} &= \mathbf{J}^{[i,j]T}\tilde{\mathbf{\Omega}}^{[i,j]}\mathbf{e}^{[i,j]} \\
&= \begin{pmatrix}
\vdots \\
\mathbf{J}_i^{[i,j]T}\tilde{\mathbf{\Omega}}^{[i,j]}\mathbf{e}^{[i,j]} \\
 \\
\mathbf{J}_j^{[i,j]T}\tilde{\mathbf{\Omega}}^{[i,j]}\mathbf{e}^{[i,j]} \\
\vdots
\end{pmatrix}
\end{aligned}
$$

Chordal Distance

The t2v function in the error is highly non-linear. We can simplify the problem and the derivatives by using the chordal distance.

Given two transformation matrices, the chordal distance is the difference between

- each vector in the rotation matrix
- the translation vectors

This is a 12x1 vector!

We can still use in this case the regular minus to express differences between transforms

Chordal Distance

We introduce the “flatten” function, that turns a transformation matrix in a vector containing its components

$$
\begin{aligned}
\mathbf{X} &= \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ & 1 \end{pmatrix} \\
\mathbf{R} &= \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{pmatrix} \\
\mathrm{flatten}(\mathbf{X}) &= \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \\ \mathbf{t} \end{pmatrix}
\end{aligned}
$$

Chordal Prediction and Error

With flattening we can rewrite prediction error as follows

$$
\begin{aligned}
\mathbf{g}^{[i,j]}(\mathbf{X}) &= \mathbf{X}_r^{[i]-1}\mathbf{X}_r^{[j]} \\
\mathbf{h}^{[i,j]}(\mathbf{X}) &= \mathrm{flatten}(\mathbf{g}^{[i,j]}) \\
\mathbf{e}^{[i,j]}(\mathbf{X}) &= \mathrm{flatten}(\mathbf{X}_r^{[i]-1}\mathbf{X}_r^{[j]}) - \mathrm{flatten}(\mathbf{Z}^{[i,j]})
\end{aligned}
$$

$$
\frac{\partial \mathbf{e}^{[i,j]}(\mathbf{X}\boxplus\mathbf{\Delta x})}{\partial \mathbf{\Delta x}} = \left(\cdots \mathbf{0}_{12\times 6} \cdots \frac{\partial \mathbf{e}^{[i,j]}(\mathbf{X}\boxplus\mathbf{\Delta x})}{\partial \mathbf{\Delta x}_r^i} \cdots \mathbf{0}_{12\times 6} \cdots \frac{\partial \mathbf{e}^{[i,j]}(\mathbf{X}\boxplus\mathbf{\Delta x})}{\partial \mathbf{\Delta x}_r^j} \cdots \mathbf{0}_{12\times 6} \cdots \right)
$$

The error becomes
12 dimensions!

easier to compute
with pen and paper

Chordal Jacobian

- We expand the prediction at the perturbations, bearing in mind that the derivative will be evaluated in 0

$$
\begin{aligned}
\mathbf{g}(\mathbf{X}_i \boxplus \boldsymbol{\Delta}\mathbf{x}_i, \mathbf{X}_j \boxplus \boldsymbol{\Delta}\mathbf{x}_j) &= \mathbf{X}_i^{-1} v2t(\boldsymbol{\Delta}\mathbf{x}_i)^{-1} v2t(\boldsymbol{\Delta}\mathbf{x}_j)\mathbf{X}_j \\
v2t(\boldsymbol{\Delta}\mathbf{x})^{-1} &\simeq v2t(-\boldsymbol{\Delta}\mathbf{x}) \text{ for small } \boldsymbol{\Delta}\mathbf{x} \\
\mathbf{g}(\mathbf{X}_i \boxplus \boldsymbol{\Delta}\mathbf{x}_i, \mathbf{X}_j \boxplus \boldsymbol{\Delta}\mathbf{x}_j) &= \underbrace{\begin{pmatrix} \mathbf{R}_i^T & -\mathbf{R}_i^T\mathbf{t}_i \\ & 1 \end{pmatrix}}_{\mathbf{X}_i^{-1}} \underbrace{\begin{pmatrix} \mathbf{R}(-\boldsymbol{\Delta}\alpha_i) & -\boldsymbol{\Delta}\mathbf{t}_i \\ & 1 \end{pmatrix}}_{v2t(\boldsymbol{\Delta}\mathbf{x}_i)^{-1}} \underbrace{\begin{pmatrix} \mathbf{R}(\boldsymbol{\Delta}\alpha_j) & \boldsymbol{\Delta}\mathbf{t}_j \\ & 1 \end{pmatrix}}_{v2t(\boldsymbol{\Delta}\mathbf{x}_j)} \underbrace{\begin{pmatrix} \mathbf{R}_j & \mathbf{t}_j \\ & 1 \end{pmatrix}}_{\mathbf{X}_j}
\end{aligned}
$$

- Looking at the upper equation, we can say that

$$
\left.\frac{\partial \mathbf{g}(\mathbf{X}_i \boxplus \boldsymbol{\Delta}\mathbf{x}_i, \mathbf{X}_j \boxplus \boldsymbol{\Delta}\mathbf{x}_j)}{\partial \boldsymbol{\Delta}\mathbf{x}_i}\right|_{\boldsymbol{\Delta}\mathbf{x}=0} = -\left.\frac{\partial \mathbf{g}(\mathbf{X}_i \boxplus \boldsymbol{\Delta}\mathbf{x}_i, \mathbf{X}_j \boxplus \boldsymbol{\Delta}\mathbf{x}_j)}{\partial \boldsymbol{\Delta}\mathbf{x}_j}\right|_{\boldsymbol{\Delta}\mathbf{x}=0}
$$

It’s a minus!

Chordal Jacobian

- We then focus our effort to compute the derivative w.r.t x_j, being the derivative w.r.t x_i its opposite

$$\begin{aligned}
\mathbf{g}(\mathbf{X}_i, \mathbf{X}_j \boxplus \mathbf{\Delta x}_j) &= \begin{bmatrix} \mathbf{R}_i^T & -\mathbf{R}_i^T \mathbf{t}_i \end{bmatrix} \begin{bmatrix} \mathbf{R}(\mathbf{\Delta}\alpha_j) & \mathbf{\Delta t}_j \end{bmatrix} \begin{bmatrix} \mathbf{R}_j & \mathbf{t}_j \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{R}_i^T \mathbf{R}(\mathbf{\Delta}\alpha_j) \mathbf{R}_j & \mathbf{R}_i^T (\mathbf{R}(\mathbf{\Delta}\alpha_j)\mathbf{t}_j + \mathbf{\Delta t}_j - \mathbf{t}_i) \end{bmatrix}
\end{aligned}$$

- The derivation w.r.t each component of x_j gives us a 4x4 matrix, of which only the first three rows are relevant

Chordal Jacobian

- Recalling that

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{pmatrix} \quad \mathbf{R}_y = \begin{pmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s & 0 & c \end{pmatrix} \quad \mathbf{R}_z = \begin{pmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{R}'_x = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -s & -c \\ 0 & c & -s \end{pmatrix} \quad \mathbf{R}'_y = \begin{pmatrix} -s & 0 & c \\ 0 & 0 & 0 \\ -c & 0 & -s \end{pmatrix} \quad \mathbf{R}'_z = \begin{pmatrix} -s & -c & 0 \\ c & -s & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{R}'_{x0} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \quad \mathbf{R}'_{y0} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \quad \mathbf{R}'_{z0} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\frac{\partial \mathbf{g}}{\partial \Delta\alpha_x} = \begin{pmatrix} \mathbf{R}_i^T \mathbf{R}'_{x0} \mathbf{R}_j & \mathbf{R}_i^T \mathbf{R}'_{x0} \mathbf{t}_j \end{pmatrix} \qquad \frac{\partial \mathbf{g}}{\partial \mathbf{\Delta t}_x} = \begin{pmatrix} \mathbf{0}_{3\times 3} & \mathbf{R}_i^T (1\ 0\ 0)^T \end{pmatrix}$$

$$\frac{\partial \mathbf{g}}{\partial \Delta\alpha_y} = \begin{pmatrix} \mathbf{R}_i^T \mathbf{R}'_{y0} \mathbf{R}_j & \mathbf{R}_i^T \mathbf{R}'_{y0} \mathbf{t}_j \end{pmatrix} \qquad \frac{\partial \mathbf{g}}{\partial \mathbf{\Delta t}_y} = \begin{pmatrix} \mathbf{0}_{3\times 3} & \mathbf{R}_i^T (0\ 1\ 0)^T \end{pmatrix}$$

$$\frac{\partial \mathbf{g}}{\partial \Delta\alpha_z} = \begin{pmatrix} \mathbf{R}_i^T \mathbf{R}'_{z0} \mathbf{R}_j & \mathbf{R}_i^T \mathbf{R}'_{z0} \mathbf{t}_j \end{pmatrix} \qquad \frac{\partial \mathbf{g}}{\partial \mathbf{\Delta t}_z} = \begin{pmatrix} \mathbf{0}_{3\times 3} & \mathbf{R}_i^T (0\ 0\ 1)^T \end{pmatrix}$$

Chordal Jacobian

- The final jacobian is assembled in a 12x6 matrix, by flattening the contribution of the components

$$\left.\frac{\partial \mathbf{h}(\mathbf{X}_i, \mathbf{X}_j \boxplus \boldsymbol{\Delta}\mathbf{x}_j)}{\partial \boldsymbol{\Delta}\mathbf{x}_j}\right|_{\boldsymbol{\Delta}\mathbf{x}=0} = \begin{pmatrix} d\mathbf{h}_x & d\mathbf{h}_y & d\mathbf{h}_z & d\mathbf{h}_{\alpha_x} & d\mathbf{h}_{\alpha_y} & d\mathbf{h}_{\alpha_z} \end{pmatrix}$$

$$\begin{aligned}
d\mathbf{h}_x &= \text{flatten}\left(\frac{\partial \mathbf{g}}{\partial \boldsymbol{\Delta}\mathbf{t}_x}\right) \\
d\mathbf{h}_y &= \text{flatten}\left(\frac{\partial \mathbf{g}}{\partial \boldsymbol{\Delta}\mathbf{t}_y}\right) \\
d\mathbf{h}_z &= \text{flatten}\left(\frac{\partial \mathbf{g}}{\partial \boldsymbol{\Delta}\mathbf{t}_z}\right) \\
d\mathbf{h}_{\alpha_x} &= \text{flatten}\left(\frac{\partial \mathbf{g}}{\partial \boldsymbol{\Delta}\alpha_x}\right) \\
d\mathbf{h}_{\alpha_y} &= \text{flatten}\left(\frac{\partial \mathbf{g}}{\partial \boldsymbol{\Delta}\alpha_y}\right) \\
d\mathbf{h}_{\alpha_z} &= \text{flatten}\left(\frac{\partial \mathbf{g}}{\partial \boldsymbol{\Delta}\alpha_z}\right)
\end{aligned}$$

- The jacobian w.r.t x_i is the opposite

$$\left.\frac{\partial \mathbf{h}(\mathbf{X}_i \boxplus \boldsymbol{\Delta}\mathbf{x}_i, \mathbf{X}_j)}{\partial \boldsymbol{\Delta}\mathbf{x}_i}\right|_{\boldsymbol{\Delta}\mathbf{x}=0} = -\left.\frac{\partial \mathbf{h}(\mathbf{X}_i, \mathbf{X}_j \boxplus \boldsymbol{\Delta}\mathbf{x}_j)}{\partial \boldsymbol{\Delta}\mathbf{x}_j}\right|_{\boldsymbol{\Delta}\mathbf{x}=0}$$

Chordal Jacobian

- Final Jacobian with respect to x_j :

$$
\begin{aligned}
\mathbf{r}'_x &= \text{flatten}\left(\mathbf{R}_i^T \mathbf{R}'_{x0} \mathbf{R}_j\right) \\
\mathbf{r}'_y &= \text{flatten}\left(\mathbf{R}_i^T \mathbf{R}'_{y0} \mathbf{R}_j\right) \\
\mathbf{r}'_z &= \text{flatten}\left(\mathbf{R}_i^T \mathbf{R}'_{z0} \mathbf{R}_j\right)
\end{aligned}
$$

$$
\mathbf{J}_j^{[i,j]}(\mathbf{X}) = \begin{bmatrix} \mathbf{0}_{9\times 3} & \left[\mathbf{r}'_x \,|\, \mathbf{r}'_y \,|\, \mathbf{r}'_z\right] \\ \\ \mathbf{R}_i^T & -\mathbf{R}_i^T [\mathbf{t}_j]_\times \end{bmatrix}
$$

Conclusions

You can find an integrated octave example to approach a problem with

- pose-landmark
- pose-pose constraints

Using the chordal distance for pose-pose measurements.

All considerations on sparsity and low rank made for the pose-landmark problem still hold