

Probabilistic Robotics Course

Multi-Pose Registration Graph-SLAM

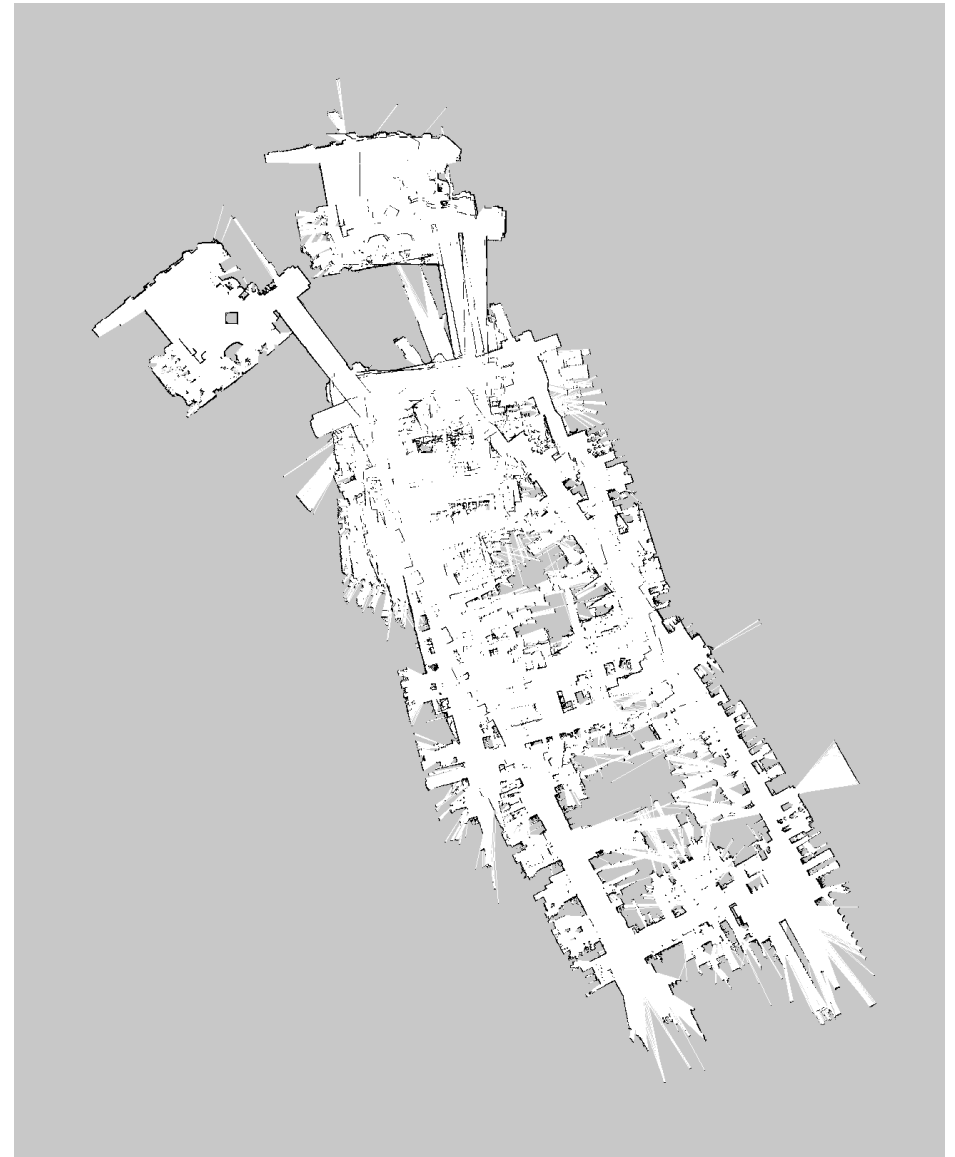
Giorgio Grisetti

`grisetti@dis.uniroma1.it`

Dept of Computer Control and Management Engineering
Sapienza University of Rome

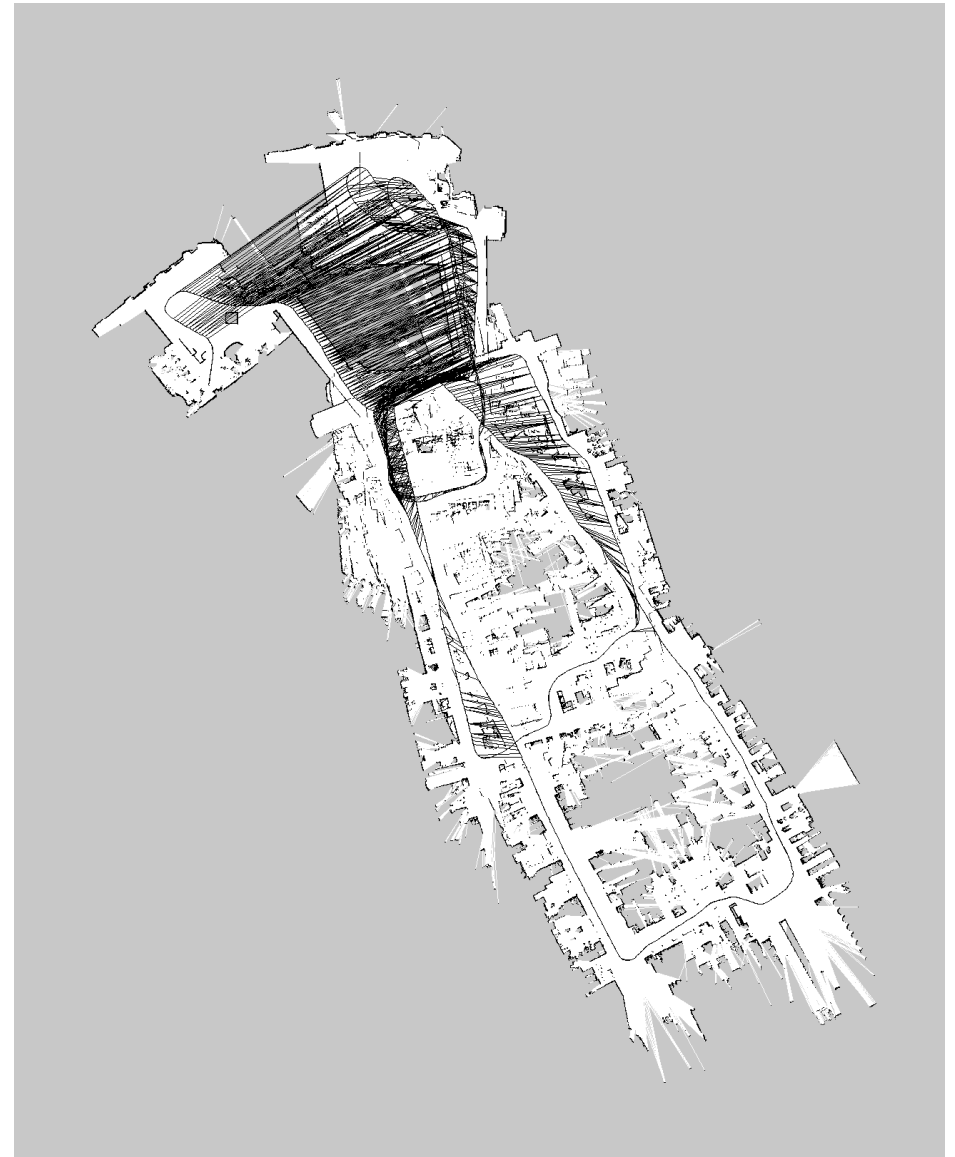
Graph-Based SLAM in a Nutshell

- 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



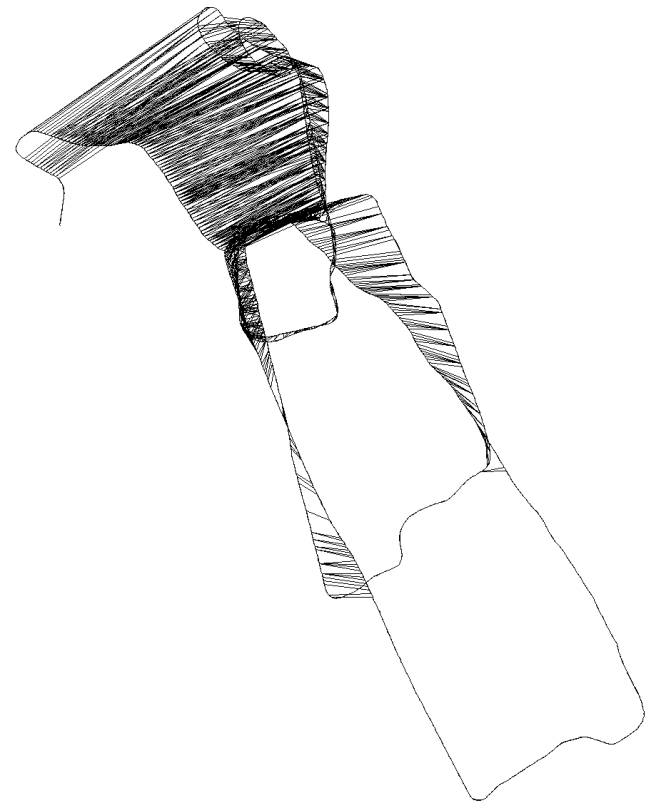
Graph-Based SLAM in a Nutshell

- 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



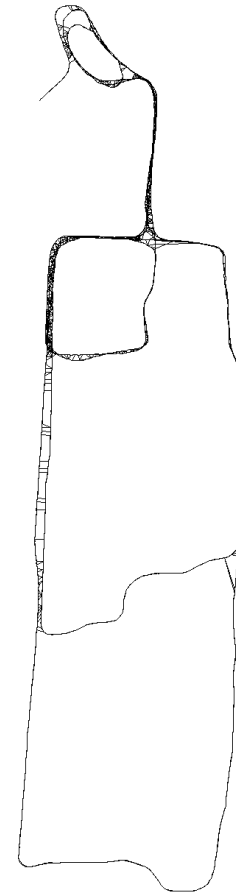
Graph-Based SLAM in a Nutshell

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



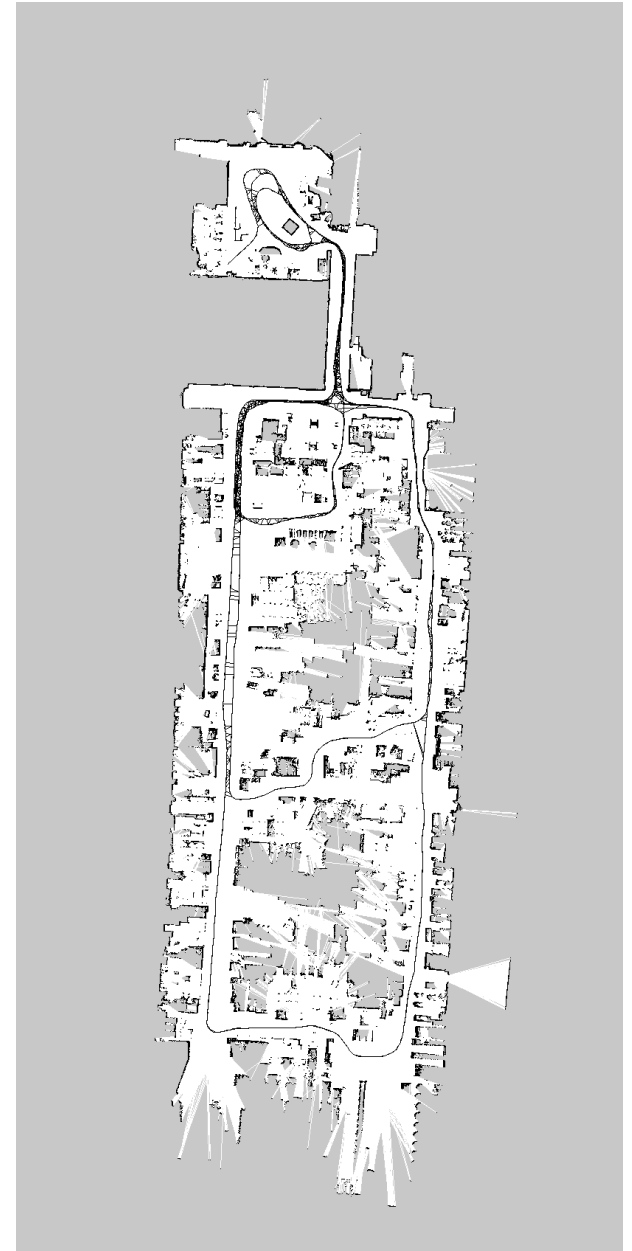
Graph-Based SLAM in a Nutshell

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



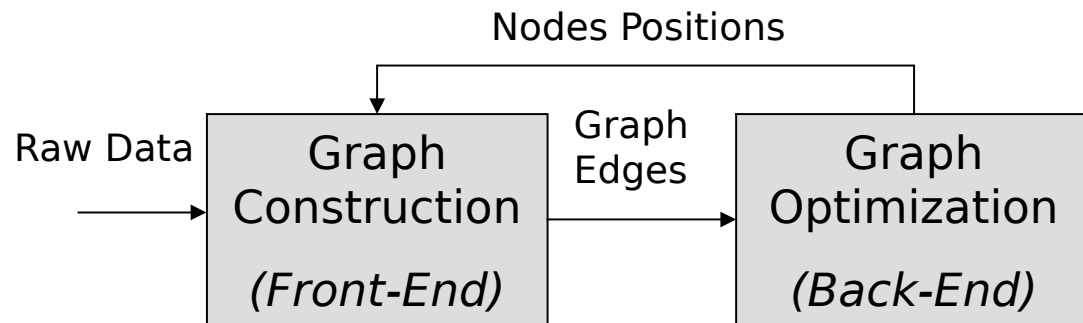
Graph-Based SLAM in a Nutshell

- 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.

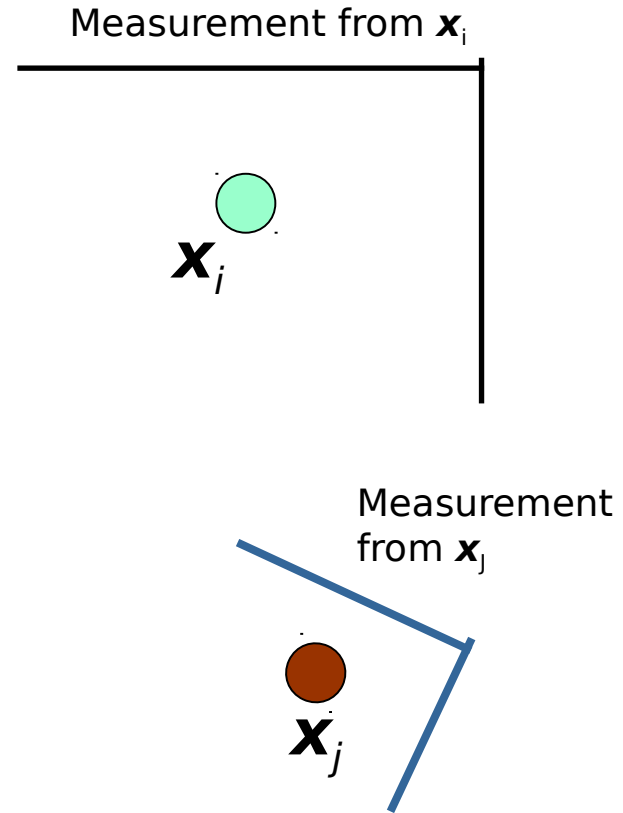


What Does the Graph Look Like?

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

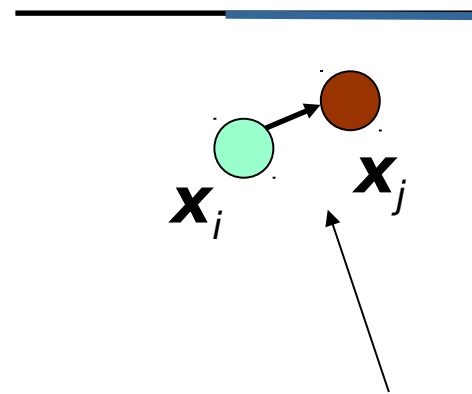
What Does the Graph Look Like?

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

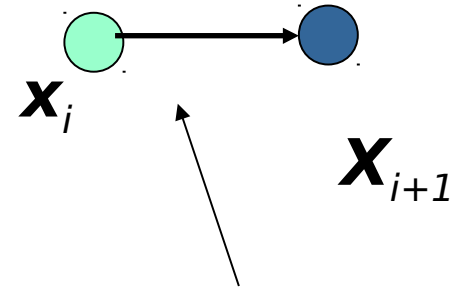
- It has n nodes $\mathbf{x}=\mathbf{x}_{1:n}$
 - Each node \mathbf{x}_i is a 2D or 3D transformation representing the pose of the robot at time \mathbf{t}_i .
- There is a constraint e_{ij} between the node \mathbf{x}_i and the node \mathbf{x}_j if
 - **either**
 - the robot observed the same part of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

What Does the Graph Look Like?

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

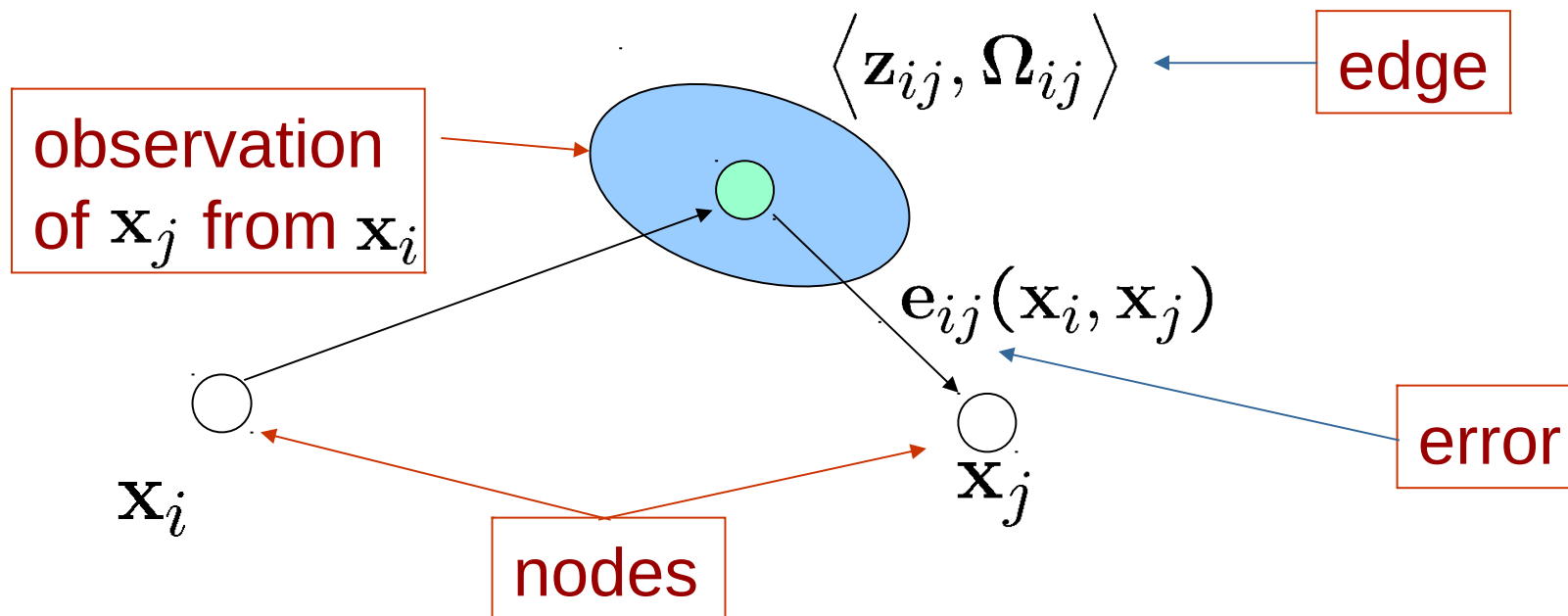
- To account for the different nature of the observations we add to the edge an information matrix $\mathbf{\Omega}_{ij}$ to encode the uncertainty of the edge.
- The “bigger” (in matrix sense) $\mathbf{\Omega}_{ij}$ is, the more the edge “matters” in the optimization procedure.

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:

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

State

The state is a collection of robot poses

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

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

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

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

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

Boxplus

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

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

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

Measurements and Predictions

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

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

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

$$\mathbf{h}^{[i,j]}(\mathbf{X}) = \mathbf{X}_r^{[i]-1} \mathbf{X}_r^{[j]}$$

$$\mathbf{e}^{[i,j]}(\mathbf{X}) = \mathbf{X}_r^{[i]-1} \mathbf{X}_r^{[j]} \boxminus \mathbf{Z}^{[i,j]}$$

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

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

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

$\mathbf{J}_i^{[i,j]}$
 $\mathbf{J}_j^{[i,j]}$

$\mathbf{J}_i^{[i,j]}$
 $\mathbf{J}_j^{[i,j]}$

$$= \begin{pmatrix}
 \cdots \mathbf{0}_{6 \times 6} \cdots \mathbf{J}_i^{[i,j]} \cdots \mathbf{0}_{6 \times 6} \cdots \mathbf{J}_j^{[i,j]} \cdots \mathbf{0}_{6 \times 6} \cdots
 \end{pmatrix}$$

Information Matrix

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

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

prediction

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

derivative of error w.r.t measurement

$$\tilde{\boldsymbol{\Omega}}_r^{[i,j]} \leftarrow (\mathbf{J}_e^{[i,j]} \boldsymbol{\Omega}^{[i,j]-1} \mathbf{J}_e^{[i,j]T})^{-1}$$

Adapted Information matrix for one iteration

H Matrix and B vector

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

$$\begin{aligned}\mathbf{H}^{[i,j]} &= \mathbf{J}^{[i,j]T} \tilde{\mathbf{\Omega}}_r^{[i,j]} \mathbf{J}^{[i,j]} \\ &= \begin{pmatrix} \mathbf{J}_r^{[i,j]T} \tilde{\mathbf{\Omega}}_r^{[i,j]} \mathbf{J}_r^{[i,j]} & \mathbf{J}_r^{[i,j]T} \tilde{\mathbf{\Omega}}_r^{[i,j]} \mathbf{J}_l^{[i,j]} \\ \mathbf{J}_l^{[i,j]T} \tilde{\mathbf{\Omega}}_r^{[i,j]} \mathbf{J}_l^{[i,j]} & \mathbf{J}_l^{[i,j]T} \tilde{\mathbf{\Omega}}_r^{[i,j]} \mathbf{J}_l^{[i,j]} \end{pmatrix}\end{aligned}$$

$$\begin{aligned}\mathbf{b}^{[i,j]} &= \mathbf{J}^{[i,j]T} \tilde{\mathbf{\Omega}}_r^{[i,j]} \mathbf{e}^{[i,j]} \\ &= \begin{pmatrix} \mathbf{J}_r^{[i,j]T} \tilde{\mathbf{\Omega}}_r^{[i,j]} \mathbf{e}^{[i,j]} \\ \mathbf{J}_l^{[i,j]T} \tilde{\mathbf{\Omega}}_r^{[i,j]} \mathbf{e}^{[i,j]} \end{pmatrix}\end{aligned}$$

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

$$\mathbf{X} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ & 1 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{pmatrix}$$

$$\text{flatten}(\mathbf{X}) = \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \\ \mathbf{t} \end{pmatrix}$$

Chordal Prediction and Error

With flattening we can rewrite prediction error as follows

$$\mathbf{h}^{[n,m]}(\mathbf{X}) = \text{flatten}(\mathbf{X}_r^{[i]-1} \mathbf{X}_l^{[j]})$$

$$\mathbf{e}^{[n,m]}(\mathbf{X}) = \text{flatten}(\mathbf{X}_r^{[i]-1} \mathbf{X}_l^{[j]}) - \text{flatten}(\mathbf{Z}^{[i,j]})$$

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

The error becomes
12 dimensions!

easier to compute
with pen and paper

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