Probabilistic Robotics Course

Multi-Pose Registration and Graph-SLAM

Giorgio Grisetti

grisetti@diag.uniroma1.it

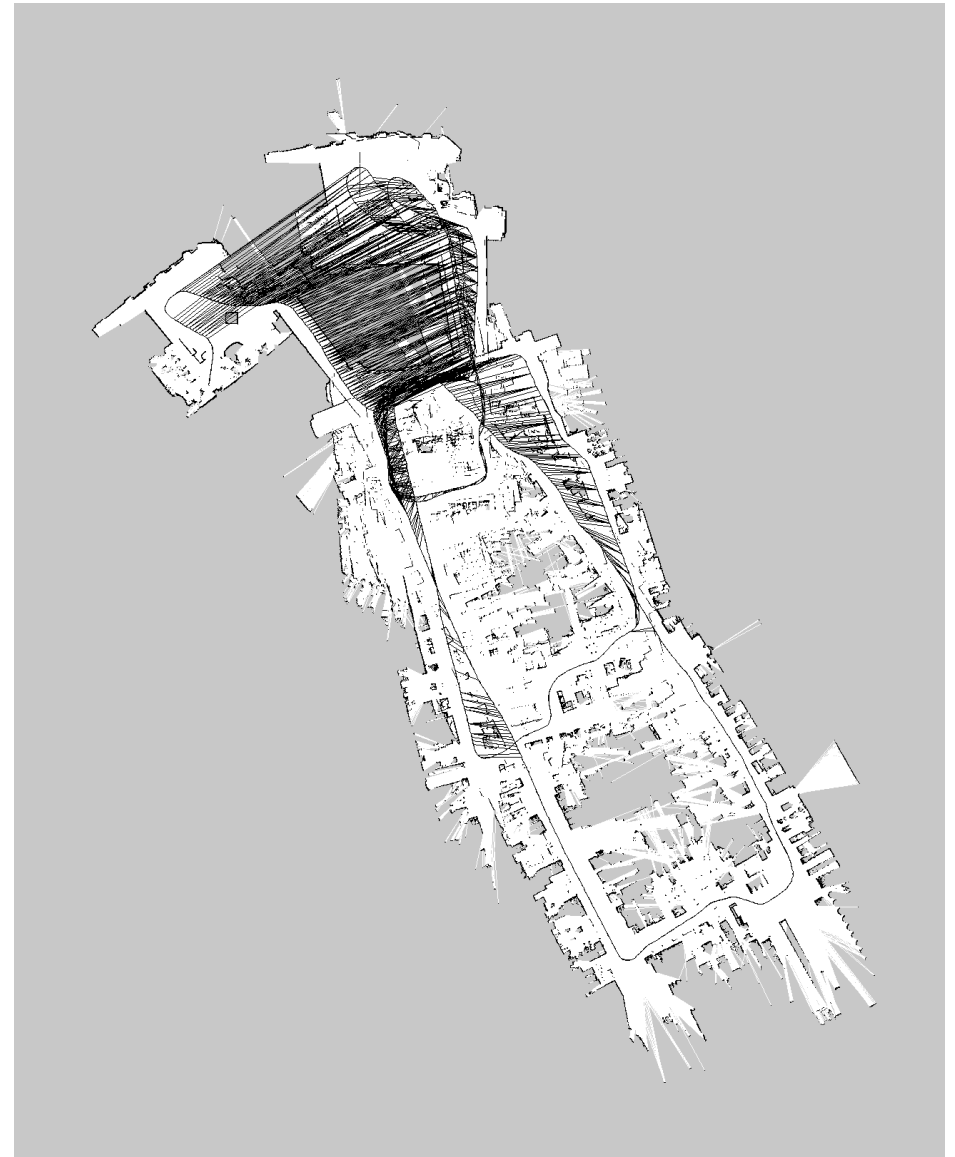Department of Computer, Control, and Management Engineering
Sapienza University of Rome

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
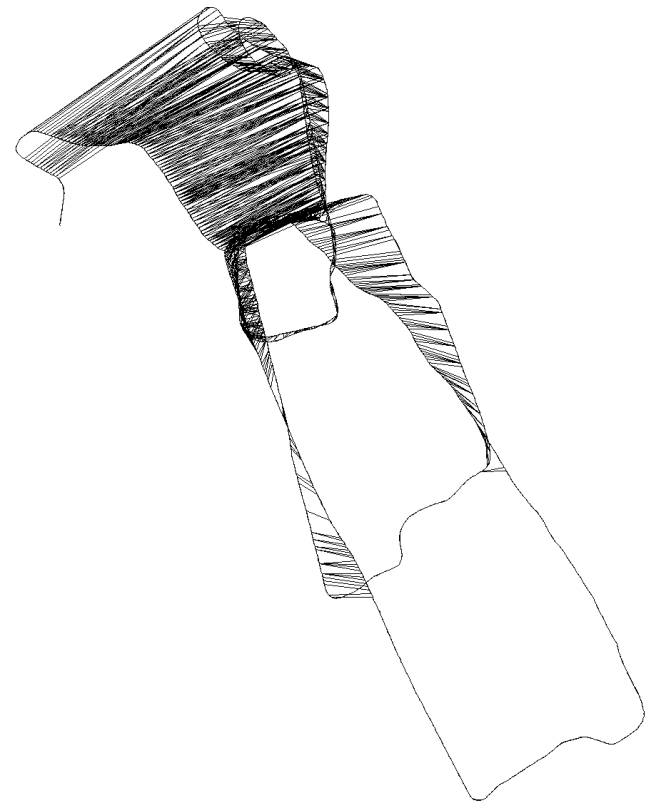

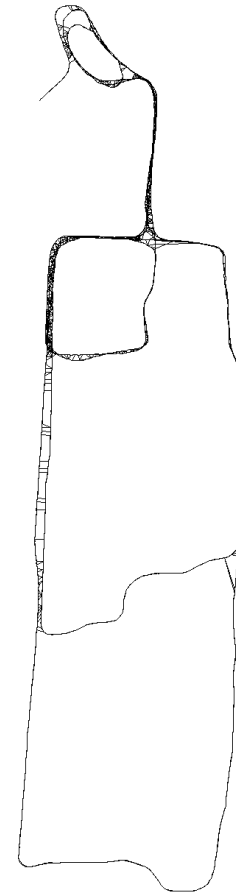KUKA Halle 22, courtesy of the P. Pfaff
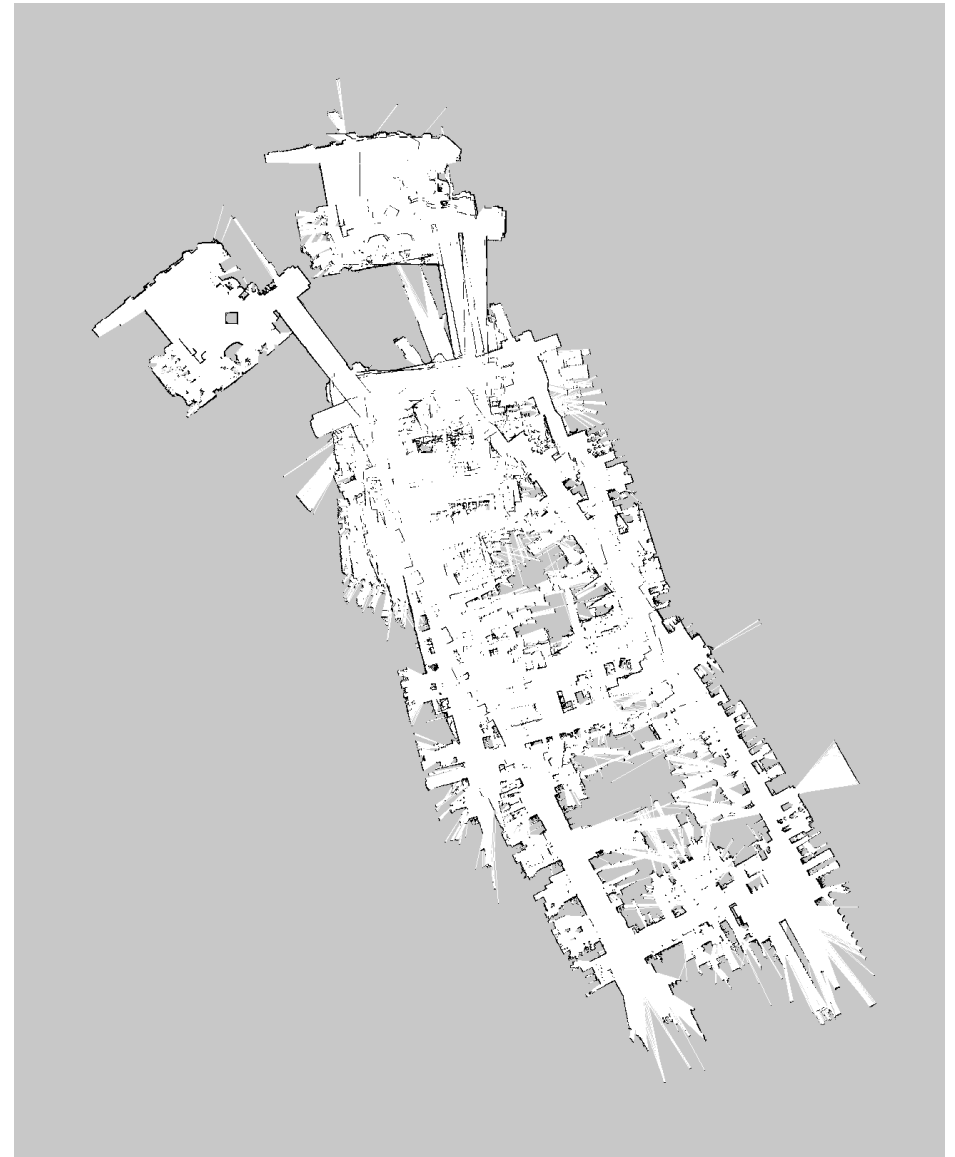
Graph-Based SLAM in a Nutshell

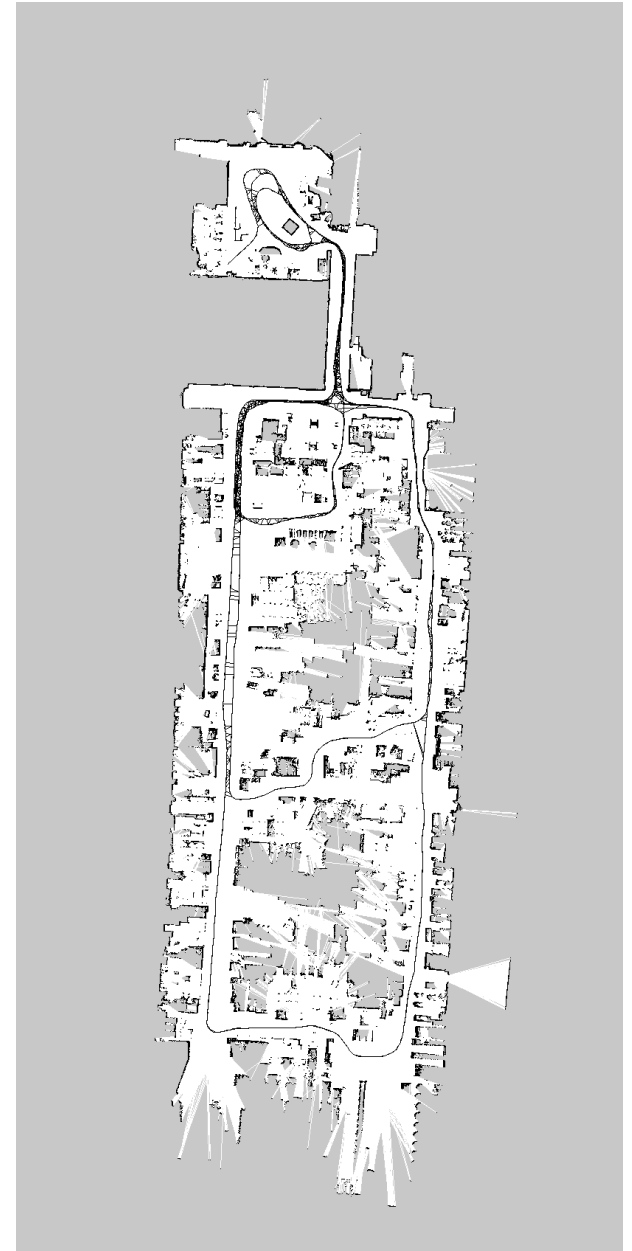- Once we have the graph we determine the most likely map by "moving" the nodes

Graph-Based SLAM in a Nutshell

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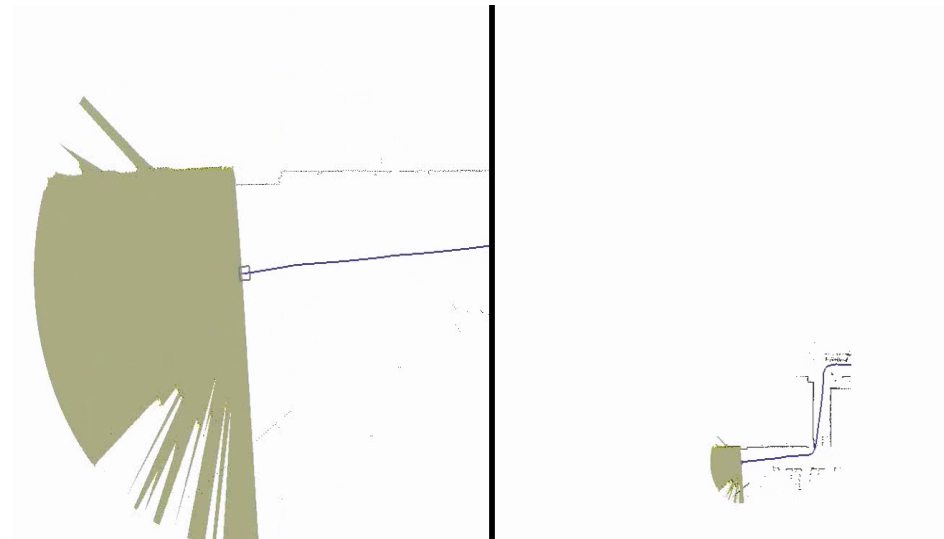- ... like this

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.
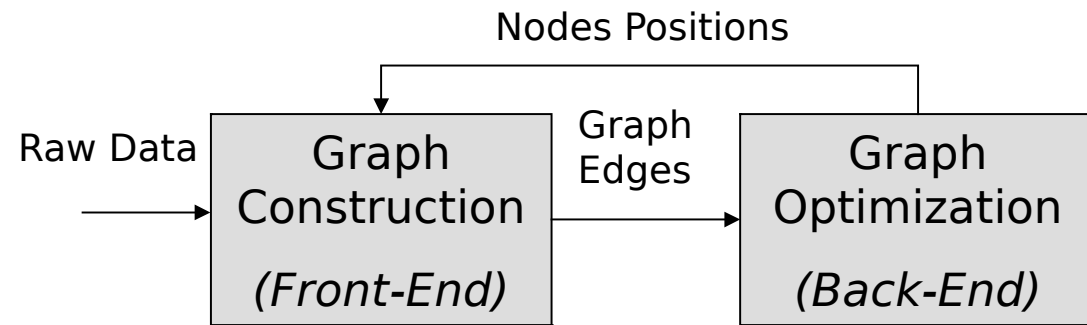
Nodes Positions

Raw Data → | Graph Construction *(Front-End)* | → Graph Edges → | Graph Optimization *(Back-End)* |

What Does the Graph Look Like?

- It has n nodes $x=x_{1:n}$

    - Each node 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 x_i and the node x_j if

    - either

        - the robot observed the same part of the environment from both x_i and x_j and,

        - via this common observation it constructs a "virtual measurement" about the position of x_j seen from.

    - Or

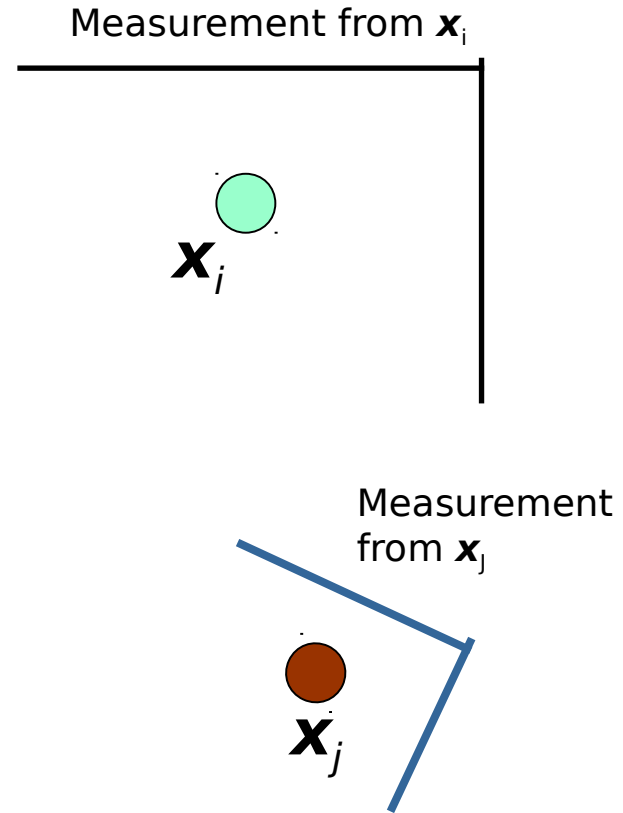   - the positions are subsequent in time and there is an odometry measurement between the two.

What Does the Graph Look Like?

- It has n nodes $x=x_{1:n}$
  - Each node 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 x_i and the node x_j if
  - either
    - the robot observed the same part of the environment from both x_i and x_j and,
    - via this common observation it constructs a "virtual measurement" about the position of x_j seen from.
  - Or
    - the positions are subsequent in time and there is an odometry measurement between the two.

Measurement from x_i

x_i

Measurement from x_j

x_j

What Does the Graph Look Like?

- It has n nodes $x = x_{1:n}$
  - Each node 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 x_i and the node x_j if
  - either
    - the robot observed the same part of the environment from both x_i and x_j and,
    - via this common observation it constructs a "virtual measurement" about the position of x_j seen from.
  - Or
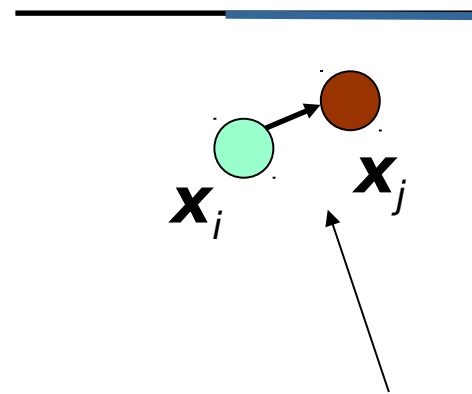   - the positions are subsequent in time and there is an odometry measurement between the two.

The edge represents the position of x_j seen from x_i, based on the **observations**

What Does the Graph Look Like?

- It has n nodes $x=x_{1:n}$
  - Each node 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 x_i and the node x_j if
  - either
    - the robot observed the same part of the environment from both x_i and x_j and,
    - via this common observation it constructs a "virtual measurement" about the position of x_j seen from.
  - Or
    - the positions are subsequent in time and there is an odometry measurement between the two.

x_i

x_{i+1}
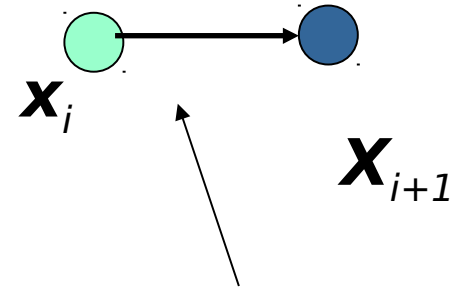
The edge represents the **odometry** measurement

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:

$$\langle \mathbf{z}_{ij}, \boldsymbol{\Omega}_{ij} \rangle \quad \longleftarrow \quad \text{edge}$$

observation of $\mathbf{x}_j$ from $\mathbf{x}_i$

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$$

error

$$\mathbf{x}_i \qquad \mathbf{x}_j$$
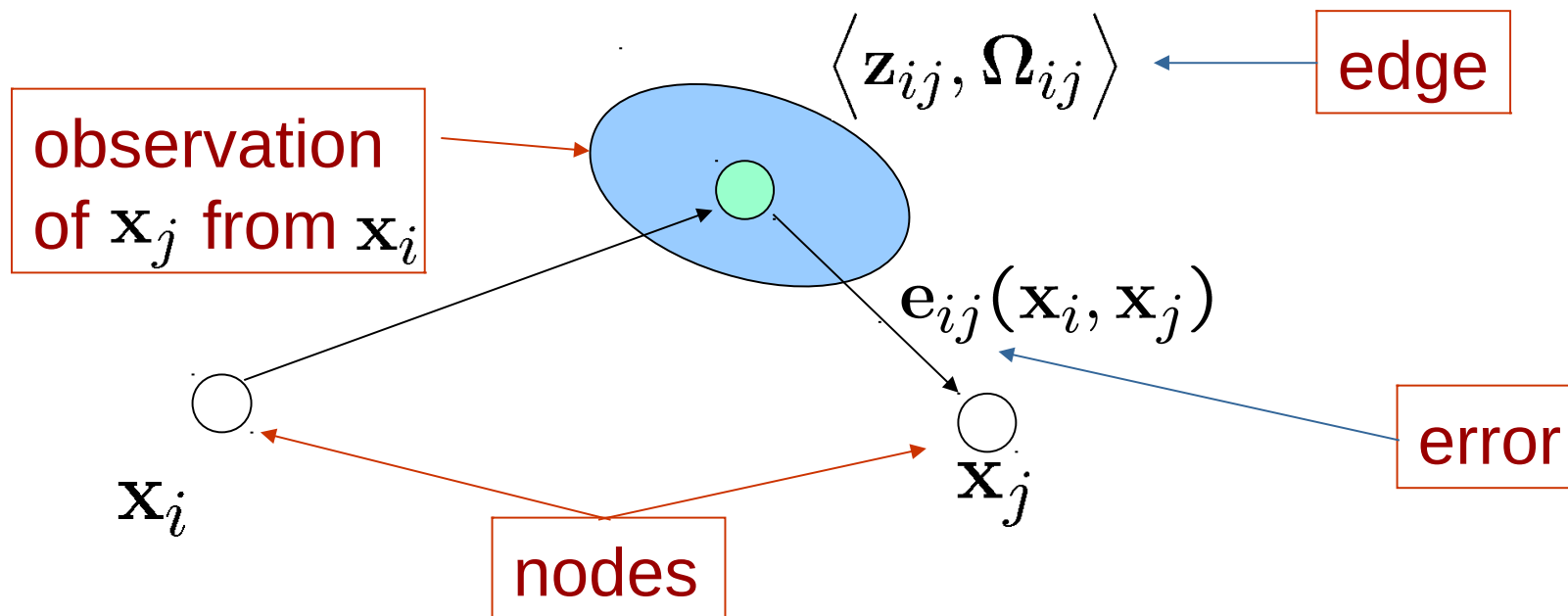
nodes

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

$$\widehat{\mathbf{x}} = \arg\min \sum_{ij} \mathbf{e}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}$$

State

The state is a collection of robot poses

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

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

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

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

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

Boxplus

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

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

$$\mathbf{X}_r^{[n]'} = \mathbf{\Delta x}_r^{[n]} \boxplus \mathbf{X}_r^{[n]}$$

$$= v2t(\mathbf{\Delta x}_r^{[n]})\mathbf{X}_r^{[n]}$$

Measurements and Predictions

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

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

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

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

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

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

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

Jacobians

The prediction depends only on the $\boxed{\text{observing}}$ and the $\boxed{\text{observed}}$ robot poses so it will be mostly 0

not easy to compute with pen and paper

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

$$\left. \frac{\partial \mathbf{e}^{[i,j]}(\mathbf{X} \boxplus \boldsymbol{\Delta}\mathbf{x})}{\partial \boldsymbol{\Delta}\mathbf{x}_r^i} \right|_{\boldsymbol{\Delta}\mathbf{x}_r^{[i]}=\mathbf{0}} = \mathbf{J}_i^{[i,j]}$$

$$\left. \frac{\partial \mathbf{e}^{[i,j]}(\mathbf{X} \boxplus \boldsymbol{\Delta}\mathbf{x})}{\partial \boldsymbol{\Delta}\mathbf{x}_r^j} \right|_{\boldsymbol{\Delta}\mathbf{x}_r^{[j]}=\mathbf{0}} = \mathbf{J}_j^{[i,j]}$$

$$\mathbf{J}^{[i,j]} = \left(\cdots \mathbf{0}_{6\times 6} \cdots \boxed{\mathbf{J}_i^{[i,j]}} \cdots \mathbf{0}_{6\times 6} \cdots \boxed{\mathbf{J}_j^{[i,j]}} \cdots \mathbf{0}_{6\times 6} \cdots \right)$$

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

$$\mathbf{J_e}^{[i,j]} = \left.\frac{\hat{\mathbf{z}}^{[i,j]}\boxminus\mathbf{z}}{\partial\mathbf{Z}}\right|_{\mathbf{Z}=\mathbf{Z}^{[i,j]}}$$ derivative of error w.r.t measurement

$$\tilde{\mathbf{\Omega}}_r^{[i,j]} \leftarrow \left(\mathbf{J_e}^{[i,j]}\mathbf{\Omega}^{[i,j]-1}\mathbf{J_e}^{[i,j]^T}\right)^{-1}$$ Adapted Information matrix for one iteration

H Matrix and B vector

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

$$\mathbf{H}^{[i,j]} = \mathbf{J}^{[i,j]T}\tilde{\boldsymbol{\Omega}}^{[i,j]}\mathbf{J}^{[i,j]}$$

$$= \begin{pmatrix} \mathbf{J}_i^{[i,j]T}\tilde{\boldsymbol{\Omega}}^{[i,j]}\mathbf{J}_i^{[i,j]} & \mathbf{J}_i^{[i,j]T}\tilde{\boldsymbol{\Omega}}^{[i,j]}\mathbf{J}_j^{[i,j]} \\ \mathbf{J}_j^{[i,j]T}\tilde{\boldsymbol{\Omega}}^{[i,j]}\mathbf{J}_l^{[i,j]} & \mathbf{J}_j^{[i,j]T}\tilde{\boldsymbol{\Omega}}^{[i,j]}\mathbf{J}_j^{[i,j]} \end{pmatrix}$$

$$\mathbf{b}^{[i,j]} = \mathbf{J}^{[i,j]T}\tilde{\boldsymbol{\Omega}}^{[i,j]}\mathbf{e}^{[i,j]}$$

$$= \begin{pmatrix} \mathbf{J}_i^{[i,j]T}\tilde{\boldsymbol{\Omega}}^{[i,j]}\mathbf{e}^{[i,j]} \\ \mathbf{J}_j^{[i,j]T}\tilde{\boldsymbol{\Omega}}^{[i,j]}\mathbf{e}^{[i,j]} \end{pmatrix}$$

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

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

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

$$\frac{\partial \mathbf{e}^{[i,j]}(\mathbf{X} \boxplus \mathbf{\Delta x})}{\partial \mathbf{\Delta x}} = \left(\cdots \mathbf{0}_{12\times6} \cdots \frac{\partial \mathbf{e}^{[i,j]}(\mathbf{X} \boxplus \mathbf{\Delta x})}{\partial \mathbf{\Delta x}_r^i} \cdots \mathbf{0}_{12\times6} \cdots \frac{\partial \mathbf{e}^{[i,j]}(\mathbf{X} \boxplus \mathbf{\Delta x})}{\partial \mathbf{\Delta x}_r^j} \cdots \mathbf{0}_{12\times6} \cdots \right)$$

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

$$\mathbf{h}(\mathbf{X}_i \boxplus \Delta\mathbf{x}_i, \mathbf{X}_j \boxplus \Delta\mathbf{x}_j) = \mathbf{X}_i^{-1} v2t(\Delta\mathbf{x}_i)^{-1} v2t(\Delta\mathbf{x}_j)\mathbf{X}_j$$

$$v2t(\Delta\mathbf{x})^{-1} \simeq v2t(-\Delta\mathbf{x}) \text{ for small } \Delta\mathbf{x}$$

$$\mathbf{h}(\mathbf{X}_i \boxplus \Delta\mathbf{x}_i, \mathbf{X}_j \boxplus \Delta\mathbf{x}_j) = \underbrace{\begin{pmatrix} \mathbf{R}_i^T & -\mathbf{R}_i^T \mathbf{t}_i \\ & 1 \end{pmatrix}}_{\mathbf{X}_i} \underbrace{\begin{pmatrix} \mathbf{R}(-\Delta\alpha_i) & -\Delta\mathbf{t}_i \\ & 1 \end{pmatrix}}_{t2v(\Delta\mathbf{x}_i)^{-1}} \underbrace{\begin{pmatrix} \mathbf{R}(\Delta\alpha_j) & \Delta\mathbf{t}_j \\ & 1 \end{pmatrix}}_{t2v(\Delta\mathbf{x}_j)} \underbrace{\begin{pmatrix} \mathbf{R}_j & \mathbf{t}_j \\ & 1 \end{pmatrix}}_{\mathbf{X}_j}$$

- Looking at the upper equation, we can say that

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

It's a minus!

Chordal Jacobian

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

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

$$= \begin{pmatrix} \mathbf{R}_i^T \mathbf{R}(\mathbf{\Delta}\alpha_j) \mathbf{R}_j & \mathbf{R}_i^T (\mathbf{R}(\mathbf{\Delta}\alpha_j) \mathbf{t}_j + \mathbf{\Delta t}_j - \mathbf{t}_i) \end{pmatrix}$$

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

$$\frac{\partial \mathbf{h}}{\partial \mathbf{\Delta}\alpha_{x,y,z}}\bigg|_{\mathbf{\Delta x}=0} = \begin{pmatrix} \mathbf{R}_i^T \frac{\partial \mathbf{R}(\mathbf{\Delta}\alpha)}{\partial \mathbf{\Delta}\alpha_{x,y,z}}\bigg|_{\mathbf{\Delta}\alpha=0} \mathbf{R}_j & \mathbf{R}_i^T \frac{\partial \mathbf{R}(\mathbf{\Delta}\alpha)}{\partial \mathbf{\Delta}\alpha_{x,y,z}}\bigg|_{\mathbf{\Delta}\alpha=0} \mathbf{t}_j) \end{pmatrix}$$

$$\frac{\partial \mathbf{h}}{\partial \mathbf{\Delta t}_{x,y,z}}\bigg|_{\mathbf{\Delta x}=0} = \begin{pmatrix} \mathbf{R}_i^T \mathbf{R}_j & \mathbf{R}_i^T \frac{\partial \mathbf{\Delta t}}{\partial \mathbf{\Delta t}_{x,y,z}}\bigg|_{\mathbf{\Delta t}=0} \end{pmatrix}$$

Chordal Jacobian

- Recalling that

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

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

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

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

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

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

Chordal Jacobian

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

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

$$d\mathbf{h}_x = \mathrm{flatten}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{\Delta t}_x}\right)$$

$$d\mathbf{h}_y = \mathrm{flatten}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{\Delta t}_y}\right)$$

$$d\mathbf{h}_z = \mathrm{flatten}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{\Delta t}_z}\right)$$

$$d\mathbf{h}_{\alpha_x} = \mathrm{flatten}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{\Delta \alpha}_x}\right)$$

$$d\mathbf{h}_{\alpha_y} = \mathrm{flatten}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{\Delta \alpha}_y}\right)$$

$$d\mathbf{h}_{\alpha_z} = \mathrm{flatten}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{\Delta \alpha}_z}\right)$$

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

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

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