VAN course
Lesson 10

Dr. Refael Vivanti
refael.vivanti@mail.huji.ac.il

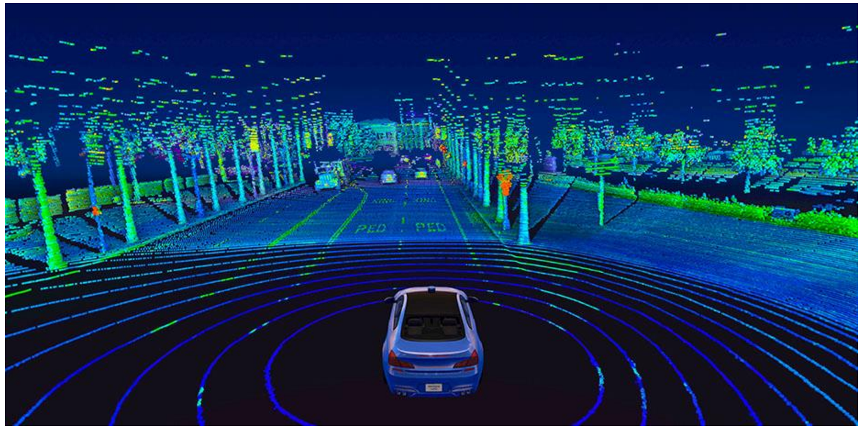
Today’s topics:

- Pose Graph
  - Depth cameras:
    - Lidar, Depth Cameras, TOF, RADAR, stereo
  - From point clouds to constraints
  - Our Pose Graph flavour
- How sparsity helps?
- Back to some statistics:
  - Information matrix and vector
  - Marginalization vs conditioning
- Compromises in our Pose Graph
- Our Pose Graph – how to

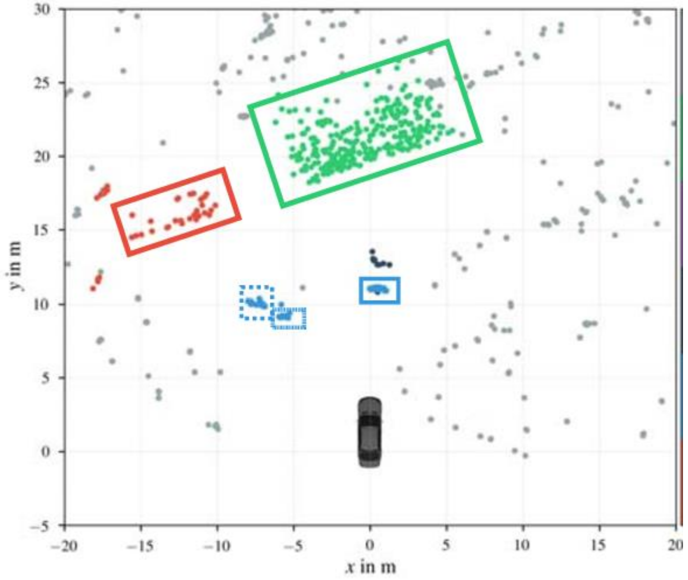
Pose Graph

Pose Graph

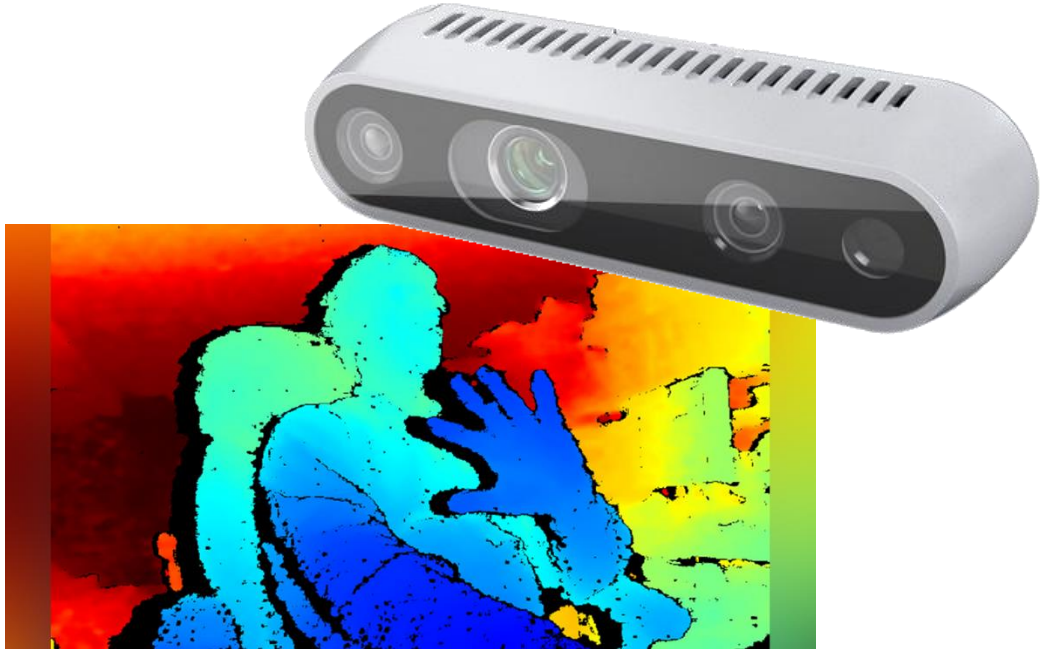
3D sensors:



LiDAR

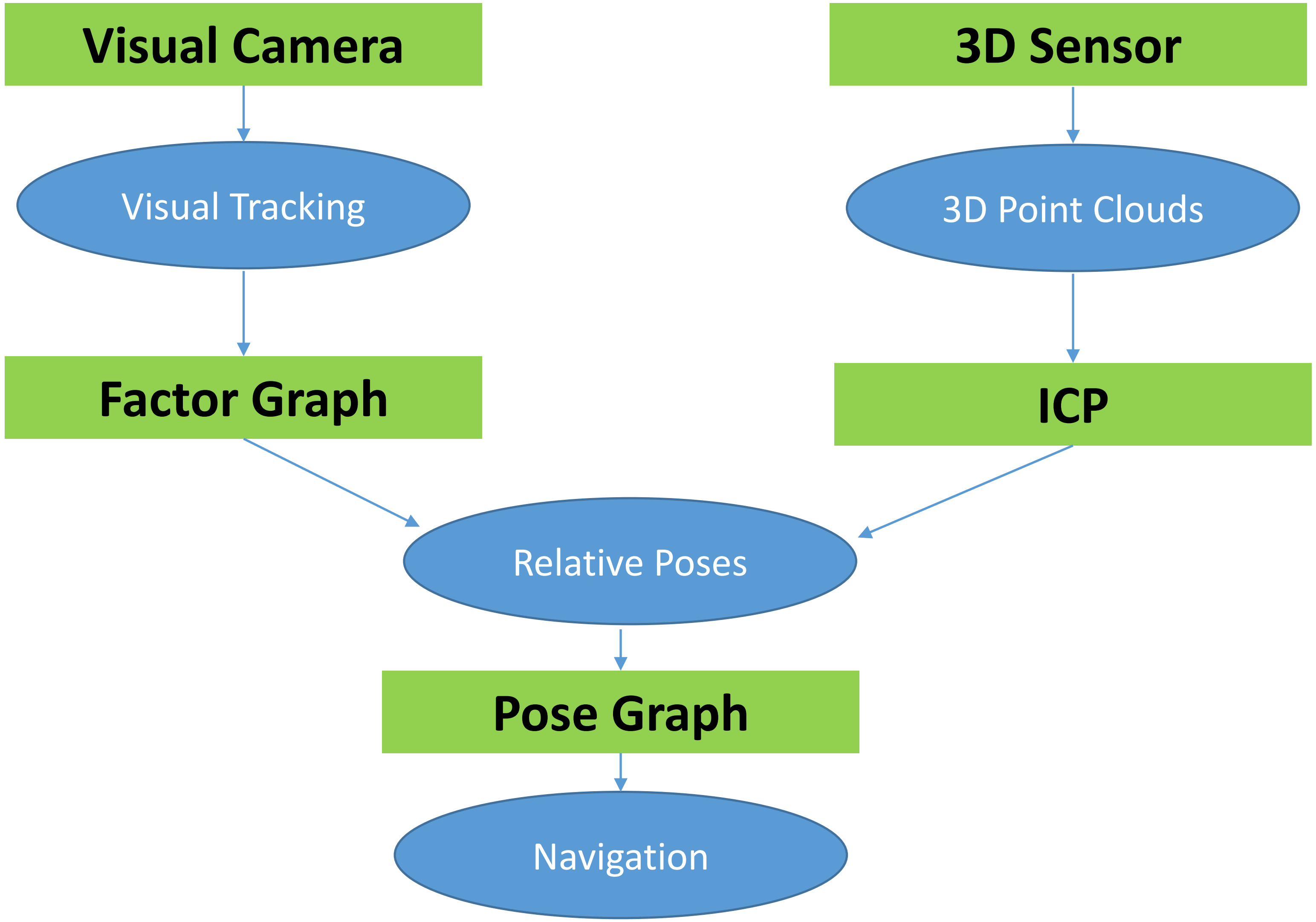


RADAR



Depth Camera

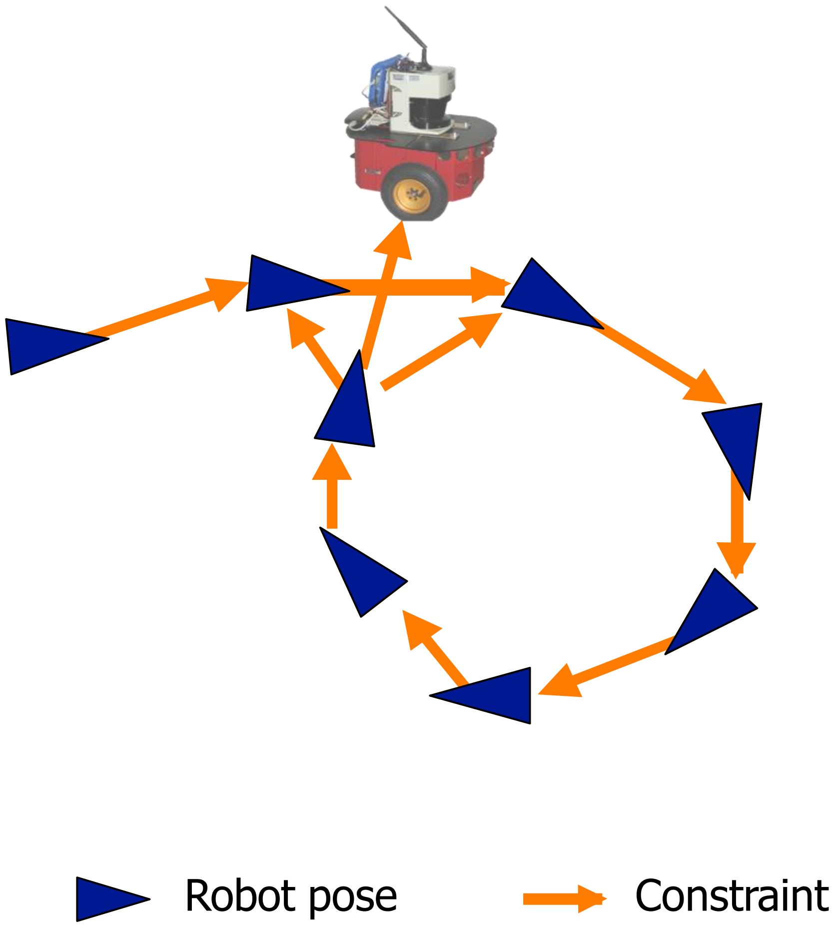
Pose Graph



Visual Camera
3D Sensor
Visual Tracking
3D Point Clouds
Factor Graph
ICP
Relative Poses
Pose Graph
Navigation

Pose Graph

Graph-Based SLAM



Pose Graph

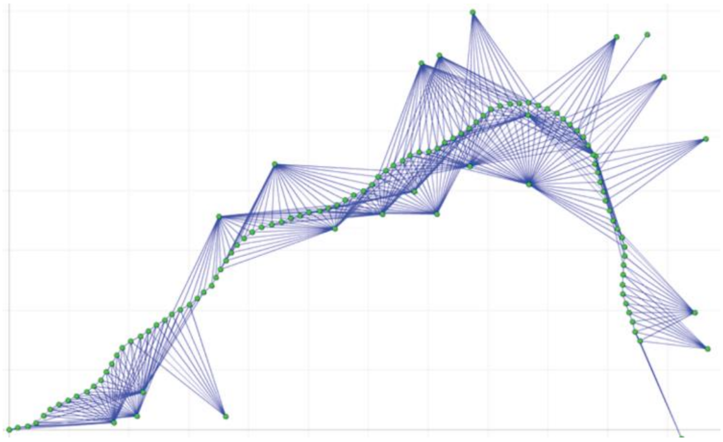
Idea of Pose Graph SLAM

- Use a **graph** to represent the problem
- Every **node** in the graph corresponds to a pose of the robot during mapping
- Every **edge** between two nodes corresponds to a spatial constraint between them
- **Graph-Based SLAM:** Build the graph and find a node configuration that minimize the error introduced by the constraints

Pose Graph

How many computations did we save?

- Case:
  - 1000 cameras, each sees 100 points.
- Full Factor graph:
  - Constraints: $2*10^5$
  - Parameters: $6*10^3$ (cameras) + $3*10^4$ (3d points)
  - Jacobian: ~10^{10}, Information matrix: ~10^9
- Pose graph
  - Key Frame every 10 frames – 100 KFs
  - 600 constraints
    - Parameters: $6*10^2$ (cameras)
  - Jacobian: ~$4*10^5$, Information matrix: ~$4*10^5$, very sparse

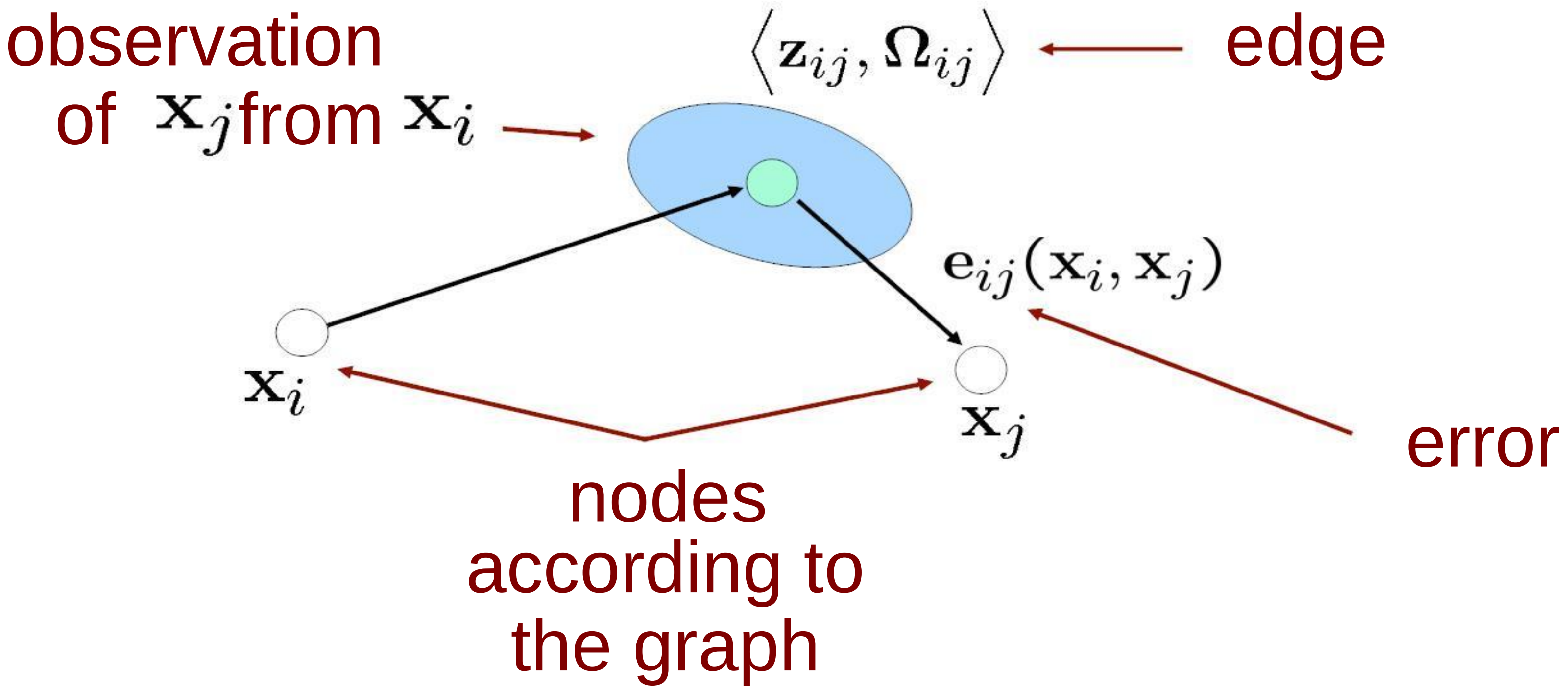


Courtesy of Dellaert06ijrr: "Square Root SAM"

How Sparsity Helps?

How sparsity helps?

Reminder: Factor Graph



Goal: $\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{ij} \mathbf{e}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij}$

$\Omega = \Sigma^{-1}$

How sparsity helps?

The Error Function

Error function for a single constraint

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{t2v}(\underbrace{\mathbf{Z}_{ij}^{-1}}_{\text{measurement}}\underbrace{(\mathbf{X}_i^{-1}\mathbf{X}_j)}_{x_j \text{ referenced w.r.t. } x_i})$$

$$X = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

Error takes a value of zero if

$$\mathbf{Z}_{ij} = (\mathbf{X}_i^{-1}\mathbf{X}_j)$$

t2v: X -> (x, y, z, α, β, γ)

How sparsity helps?

Gauss-Newton: The Overall Error Minimization Procedure

1. Define the error function
2. Linearize the error function
3. Compute its derivative
4. Set the derivative to zero
5. Solve the linear system
6. Iterate this procedure until convergence

How sparsity helps?

Jacobians and Sparsity

- Error $\mathbf{e}_{ij}(\mathbf{x})$ depends only on the two parameter blocks $\mathbf{x}_i$ and $\mathbf{x}_j$

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

- The Jacobian will be zero everywhere except in the columns of $\mathbf{x}_i$ and $\mathbf{x}_j$

$$\mathbf{J}_{ij} = \left(\mathbf{0} \cdots \mathbf{0} \quad \underbrace{\frac{\partial \mathbf{e}(\mathbf{x}_i)}{\partial \mathbf{x}_i}}_{\mathbf{A}_{ij}} \quad \mathbf{0} \cdots \mathbf{0} \quad \underbrace{\frac{\partial \mathbf{e}(\mathbf{x}_j)}{\partial \mathbf{x}_j}}_{\mathbf{B}_{ij}} \quad \mathbf{0} \cdots \mathbf{0} \right)$$

How sparsity helps?

Consequences of the Sparsity

- We need to compute the coefficient vector $\mathbf{b}$ and matrix $\mathbf{H}$:

$$\begin{aligned} \mathbf{b}^T &= \sum_{ij} \mathbf{b}_{ij}^T &= \sum_{ij} \mathbf{e}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{J}_{ij} \\ \mathbf{H} &= \sum_{ij} \mathbf{H}_{ij} &= \sum_{ij} \mathbf{J}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{J}_{ij} \end{aligned}$$

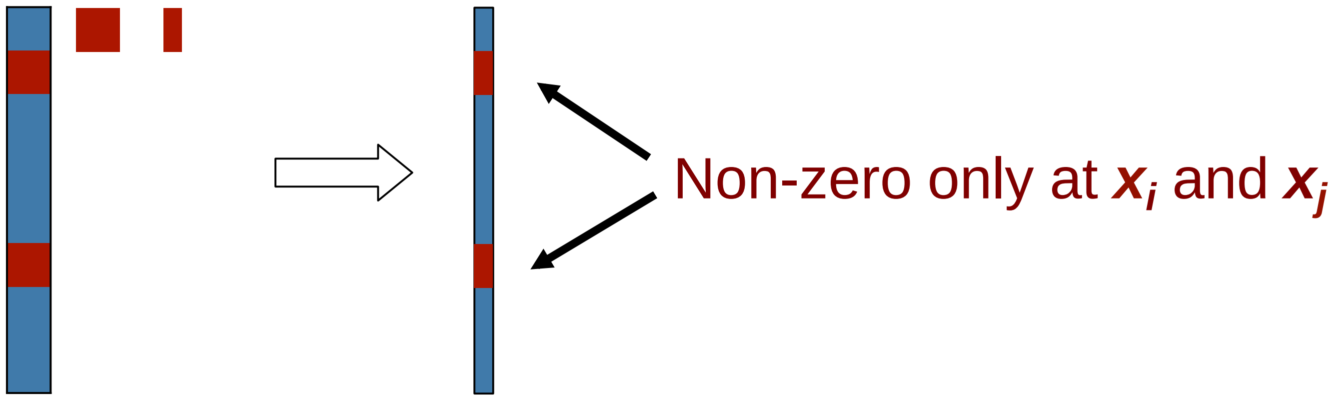
$$\Omega = \Sigma^{-1}$$

- The sparse structure of $\mathbf{J}_{ij}$ will result in a sparse structure of $\mathbf{H}$
- This structure reflects the adjacency matrix of the graph

How sparsity helps?

Illustration of the Structure

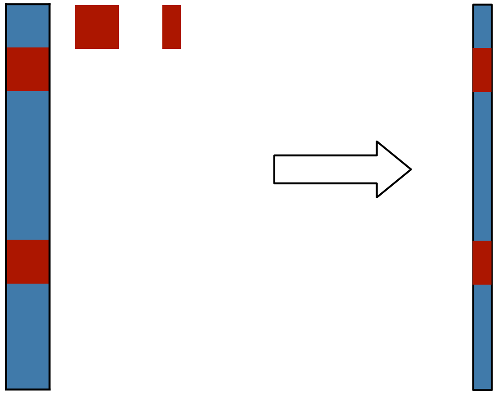
$$\mathbf{b}_{ij} = \mathbf{J}_{ij}^T \Omega_{ij} \mathbf{e}_{ij}$$



How sparsity helps?

Illustration of the Structure

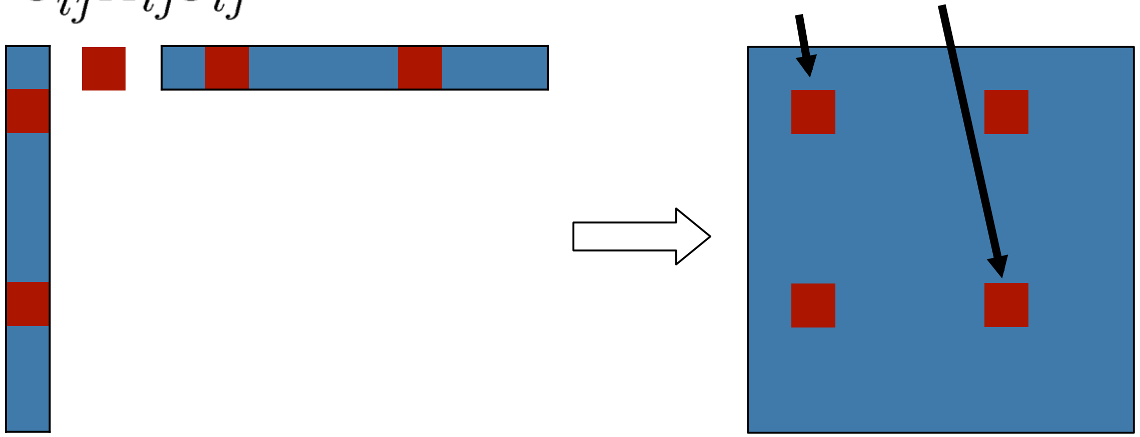
$$\mathbf{b}_{ij} = \mathbf{J}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}$$



Non-zero only at $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$

$$\mathbf{H}_{ij} = \mathbf{J}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{J}_{ij}$$

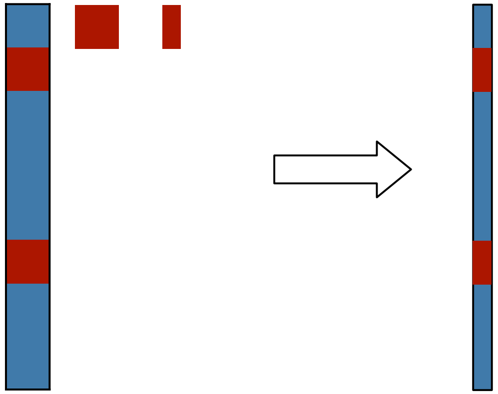
Non-zero on the main diagonal at $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$



How sparsity helps?

Illustration of the Structure

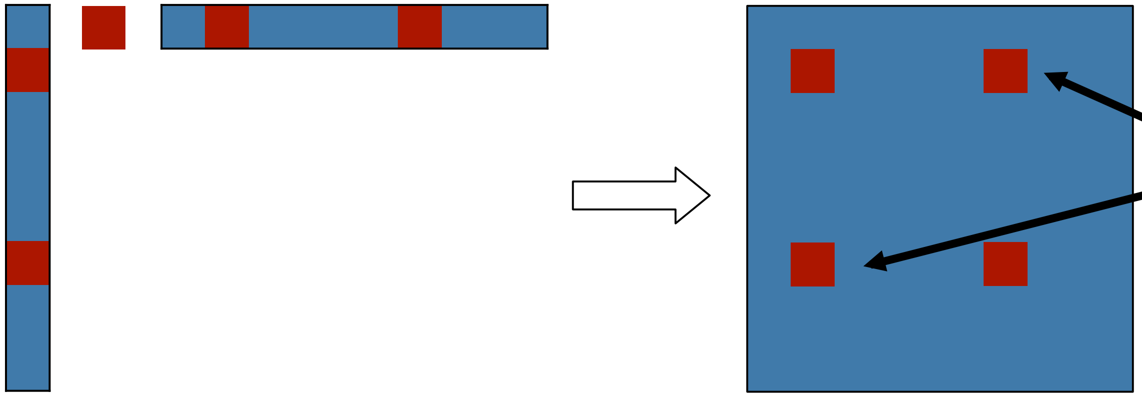
$$\mathbf{b}_{ij} = \mathbf{J}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}$$



Non-zero only at $\mathbf{x}_i$ and $\mathbf{x}_j$

$$\mathbf{H}_{ij} = \mathbf{J}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{J}_{ij}$$

Non-zero on the main diagonal at $\mathbf{x}_i$ and $\mathbf{x}_j$

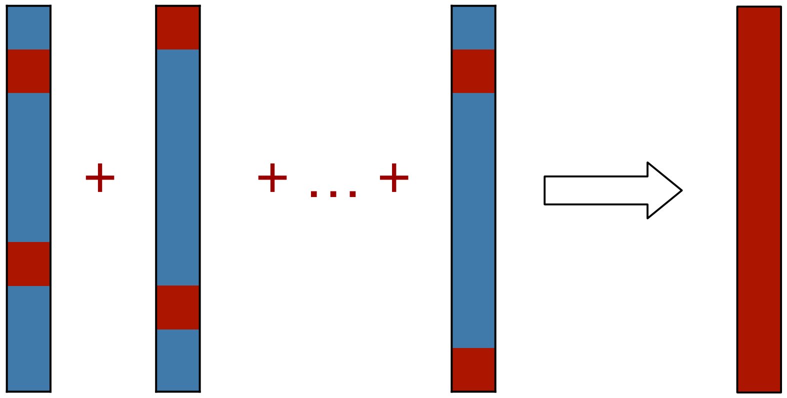


... and at the blocks *ij*,*ji*

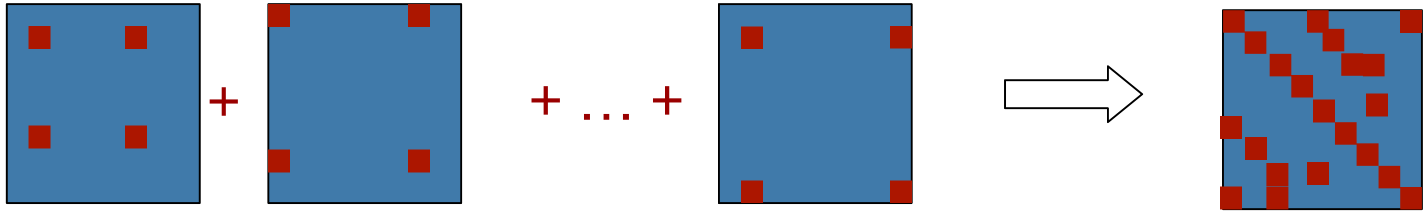
How sparsity helps?

Illustration of the Structure

$$\mathrm{b} = \sum_{ij} \mathrm{b}_{ij}$$



$$\mathbf{H} = \sum_{ij} \mathbf{H}_{ij}$$



How sparsity helps?

Building the Linear System

For each constraint:

- Compute error $\mathbf{e}_{ij} = \text{t2v}(\mathbf{Z}_{ij}^{-1}(\mathbf{X}_i^{-1}\mathbf{X}_j))$
- Compute the building-blocks:

$$\mathbf{A}_{ij} = \frac{\partial \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} \qquad \mathbf{B}_{ij} = \frac{\partial \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_j}$$

- Update the coefficient vector:

$$\bar{\mathbf{b}}_i^T += \mathbf{e}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{A}_{ij} \qquad \bar{\mathbf{b}}_j^T += \mathbf{e}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{B}_{ij}$$

- Update the system matrix:

$$\bar{\mathbf{H}}^{ii} += \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{A}_{ij} \qquad \bar{\mathbf{H}}^{ij} += \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{B}_{ij}$$

$$\bar{\mathbf{H}}^{ji} += \mathbf{B}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{A}_{ij} \qquad \bar{\mathbf{H}}^{jj} += \mathbf{B}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{B}_{ij}$$

How sparsity helps?
Algorithm

optimize($\mathbf{x}$):

while (!*converged*)

$(\mathbf{H}, \mathbf{b}) = \text{buildLinearSystem}(\mathbf{x})$

$\mathbf{\Delta x} = \text{solveSparse}(\mathbf{H}\mathbf{\Delta x} = -\mathbf{b})$

$\mathbf{x} = \mathbf{x} + \mathbf{\Delta x}$

end

return $\mathbf{x}$

less calculations

Pose Graph

So we saved a lot of computation time:

- Dropping most of our information
  - Leaving only the Key frames and their relative poses
- Using the problem sparsity

- But at what cost?