# Graph-based SLAM

Pranav Srinivas Kumar

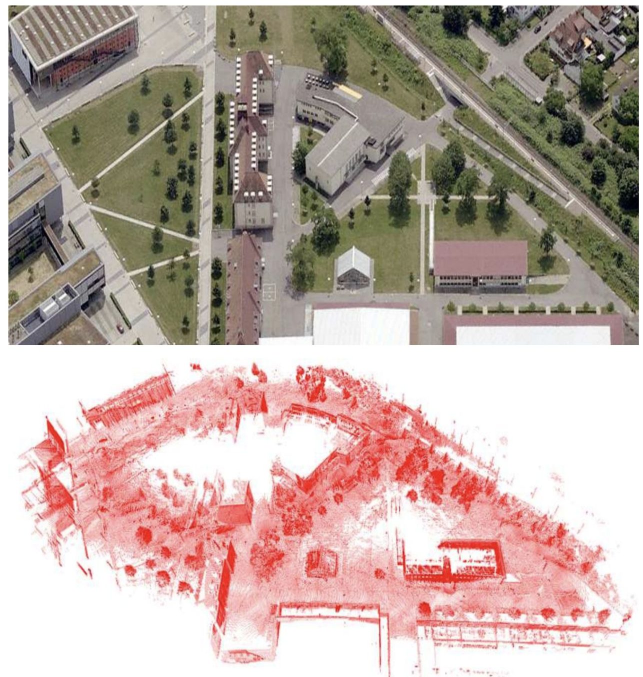
# Introduction

- Learning maps under pose uncertainty is SLAM
- Filtering approaches (Kalman filters, information filters)
    - Online state estimation (state = current robot position and the map) - usually called **online SLAM**
    - Estimation is refined by incorporating the new measurements
- Smoothing approaches
    - Estimate the full trajectory of the robot from the full set of measurements - usually called **full SLAM**
    - Least-square error minimization techniques

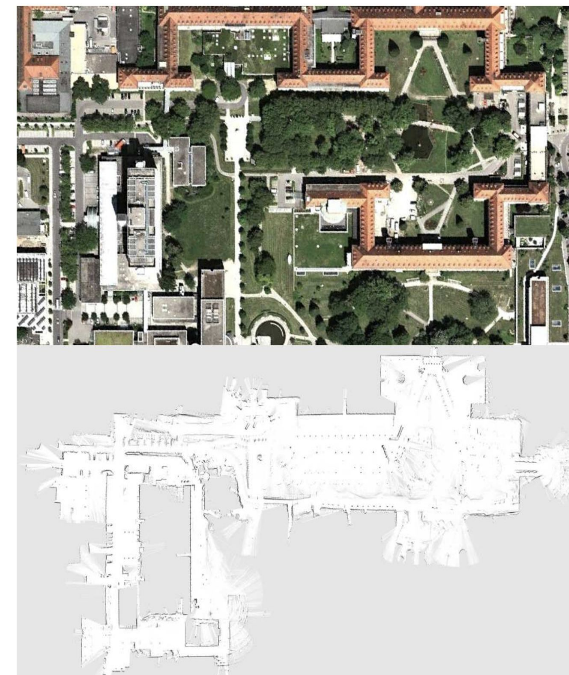

# Dense Map Representations for 3D and 2D

3D map acquired with instrumented car

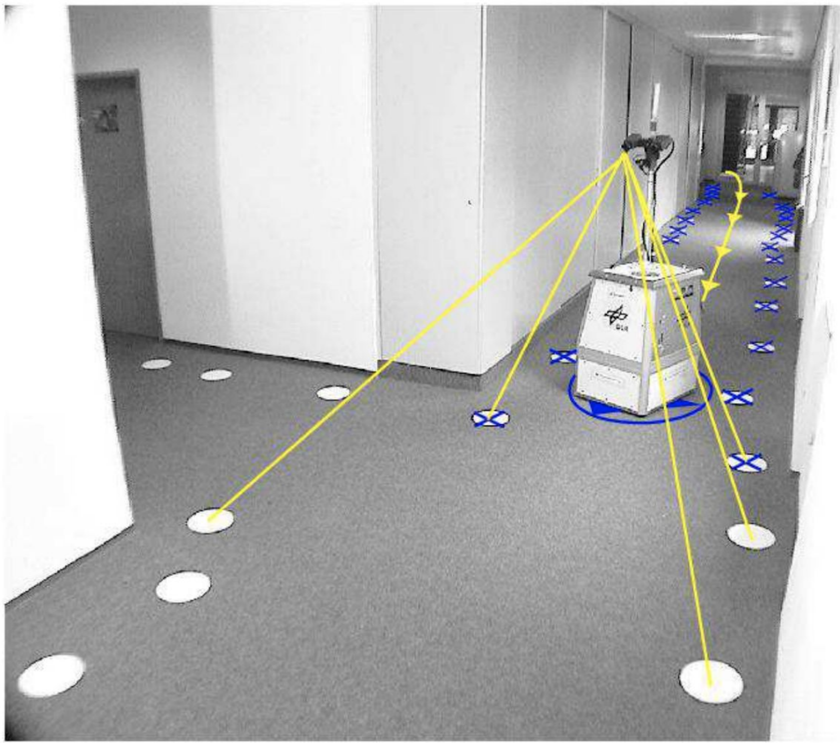
Point-cloud map and relative satellite image



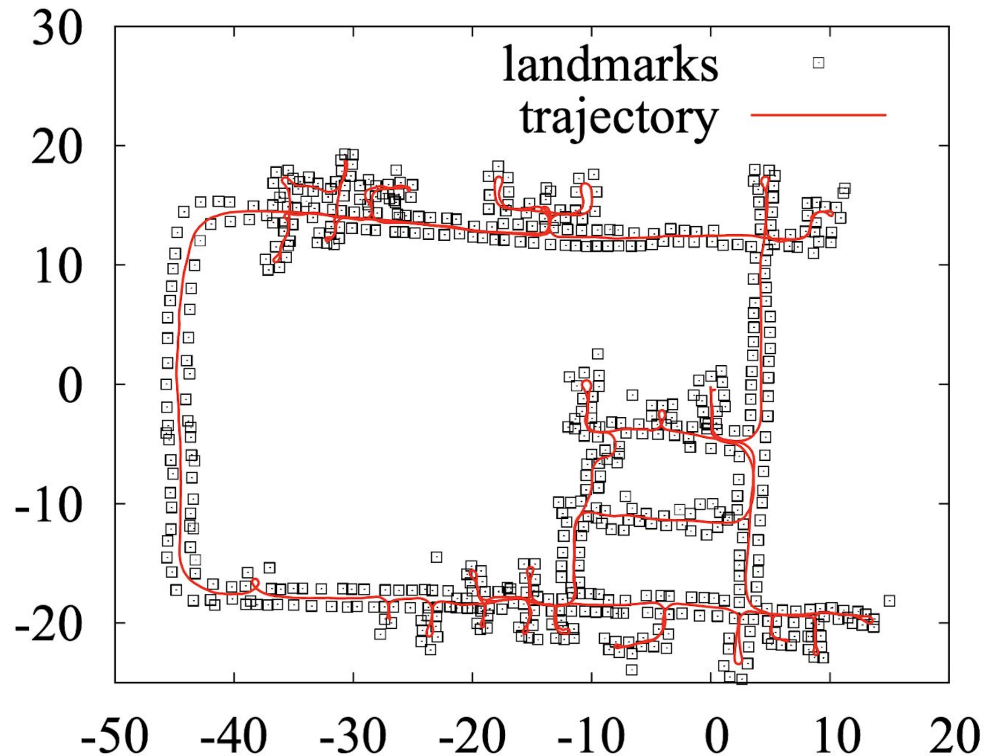
Occupancy Grid



# Landmark-based Map

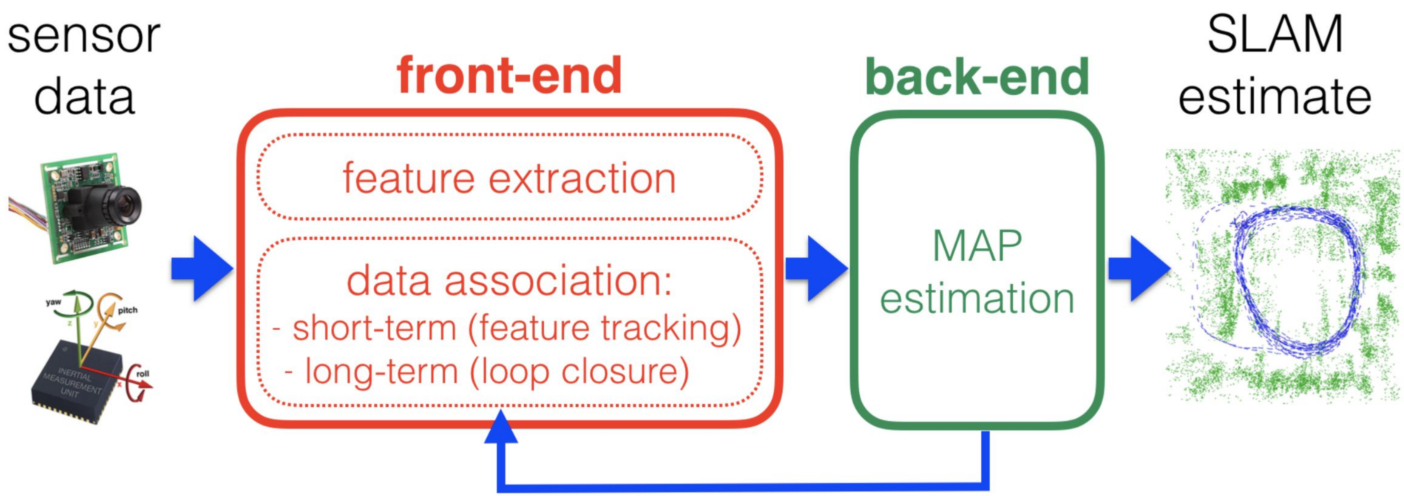


German Aerospace Center



# Graph-based 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 a graph and find a node configuration that minimize the error introduced by the constraints



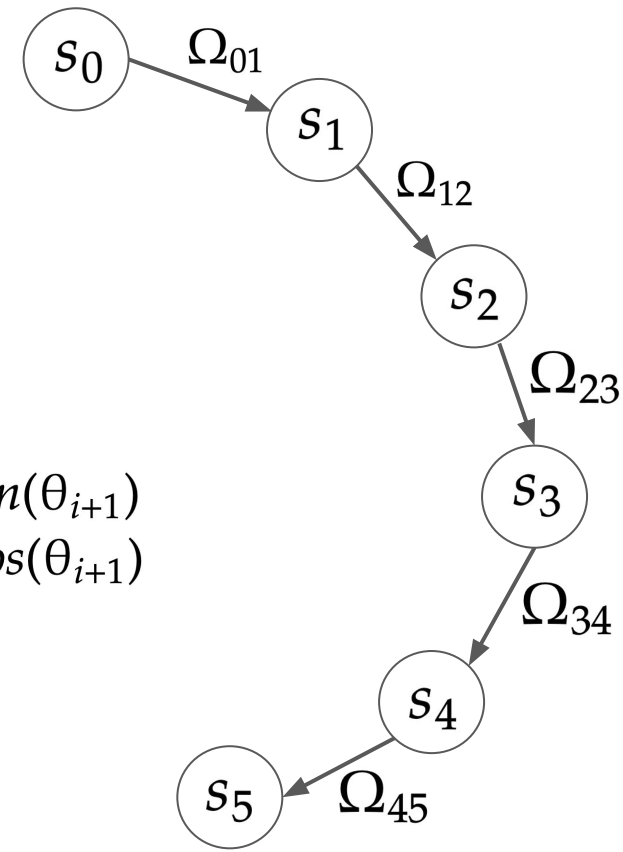
# Graph-based SLAM

- Robot pose (robot position + orientation)

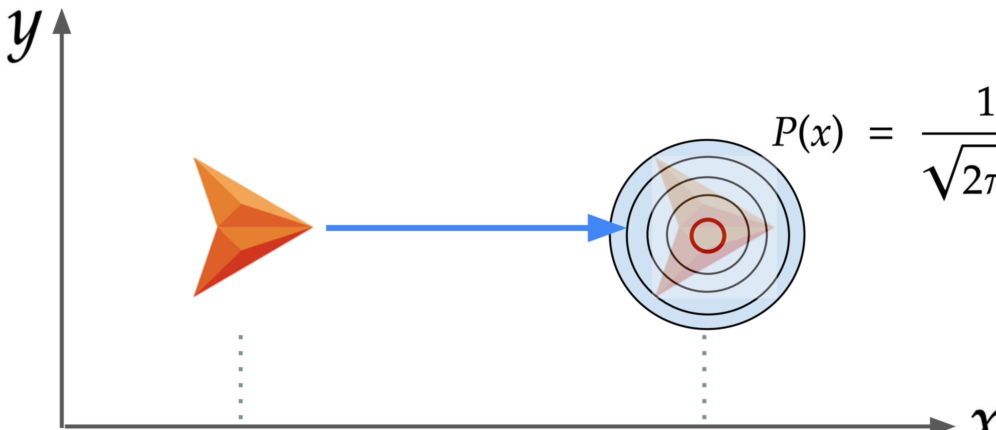
$$s_{1:N} = \{s_1, \ldots, s_N\}$$

- Constraints $\Omega_{i,j}$ on the poses relative to another
- Example in 2D
  - State $(x, y, \theta)$
  - For some translation $\delta$

$$x_{i+1} = x_i + \delta_{i,x} * cos(\theta_{i+1}) + \delta_{i,y} * sin(\theta_{i+1})$$
$$y_{i+1} = y_i + \delta_{i,x} * sin(\theta_{i+}1) + \delta_{i,y} * cos(\theta_{i+1})$$



# Edge Constraints




$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$x_0 = 0$
$y_0 = 0$

$x_1 = x_0 + 10$
$y_1 = y_0$

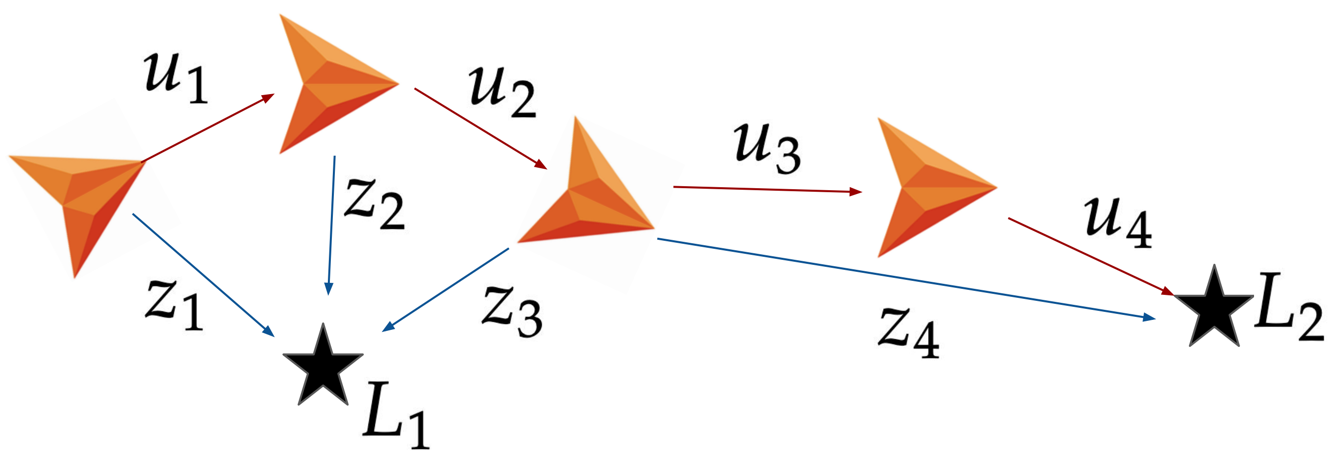
**Edge Constraint**
**Product of Gaussians**

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_1-(x_0+10))^2}{2\sigma^2}}$$
$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_1-y_0)^2}{2\sigma^2}}$$

- Maximize the likelihood of position $x_1$ given the initial position constraint $(0,\ 0)$

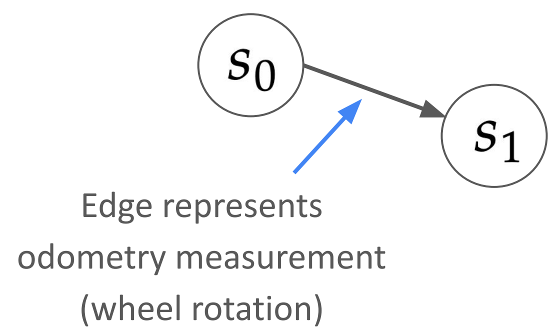
# Graph-based SLAM

- Define probabilities as a sequence of constraints
- Initial Location Constraints
- Relative Motion Constraints
- Relative Measurement Constraints
- Find the most likely robot trajectory given the constraints



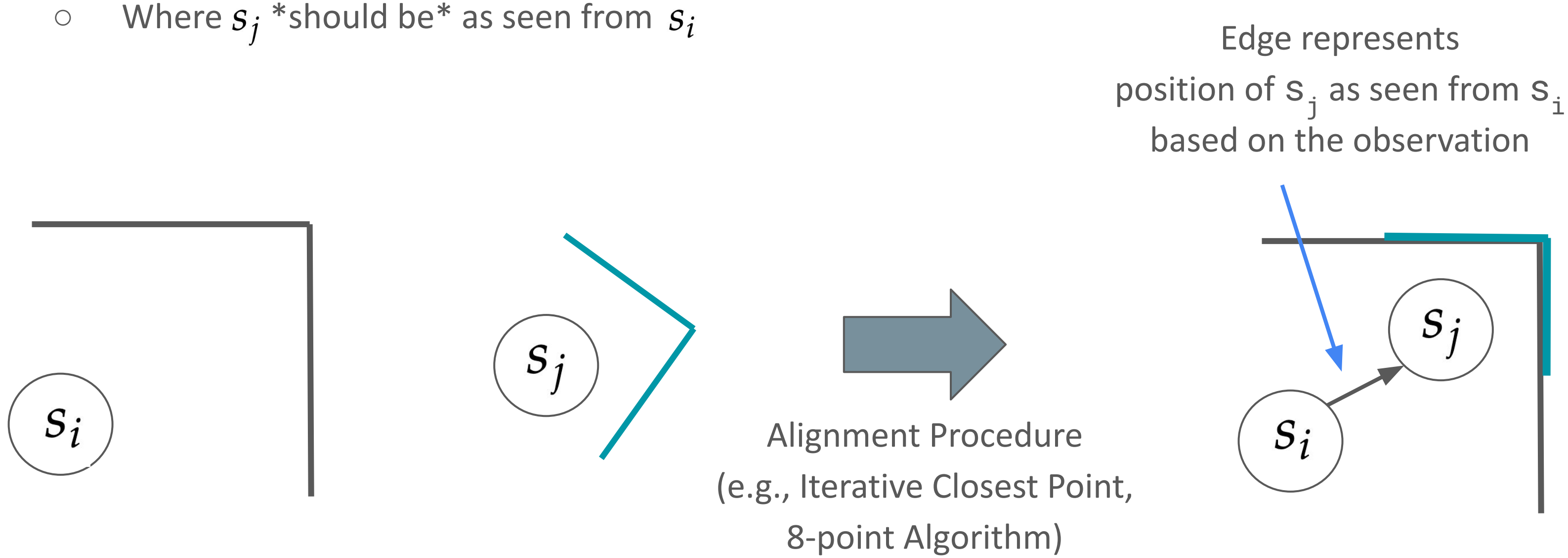
# Creating Edges - Odometry Measurement

- Robot moves from $s_i \rightarrow (s_{i+1})$
- Based on Odometry
  - Robot can estimate where it *should* be
  - Motion follows the kinematic model that the robot has about itself
  - Counting translational steering, rotational steering etc.
  - Mean estimate of where the robot has been moving from `i` to `i+1`
  - Encodes relative pose information
    - How $s_i$ can be seen from $s_{i+1}$
    - How $s_{i+1}$ can be seen from $s_i$



# Creating Edges - Arbitrary Poses - Loop Closure

- Robot observes the same environment from $s_i$ and $s_j$
- Relate this information using sensor data
- Virtual measurement
  - Where $s_j$ *should be* as seen from $s_i$



# Spatial Transformations

- Homogeneous coordinates
  - System in Projective Geometry
  - Single Matrix
  - Affine and Projective Transformations
- Odometry-based Edge

$$\left(S_i^{-1}\ S_{i+1}\right)$$

- Observation-based Edge

$$\left(S_i^{-1}\ S_j\right)$$

General Transformation Matrix

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{11} & m_{11} & m_{11} \\ m_{11} & m_{11} & m_{11} & m_{11} \\ m_{11} & m_{11} & m_{11} & m_{11} \\ m_{11} & m_{11} & m_{11} & m_{11} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ z1 \\ 1 \end{bmatrix}$$

Translation Matrix

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{aligned} x_2 &= x_1 + t_x \\ y_2 &= y_1 + t_y \\ z_2 &= z_1 + t_z \end{aligned}$$

Rotation Matrix

$$R = \begin{bmatrix} R^{3D} & 0 \\ 0 & 1 \end{bmatrix}$$

# Error Function

- Given a state, we can compute what we **expect** to perceive
- We have **real observations** relating the nodes with each other

**Find a configuration of the nodes so that the real and predicted observations are as similar as possible**

- Relative transformation (according to odometry) as measurement $z_{ij}$
- If two poses are exactly the same, $z_{ij} = 0$
- 2 sequences → 2 unconnected pose graphs
- Adding loop closure edges connects poses across sequences
- Error in the loop

$$e_{ij} = z_{ij} - (s_j - s_i)$$

# Total Likelihood

- Set of independent and identically distributed points

$$x = \{x_1, x_2, x_3, \ldots, x_N\}$$

- Total likelihood is the product of likelihood of each point

$$p(X \mid \Theta) = \prod_{i=1}^{N} p(x_i \mid \Theta)$$

where $\Theta$ are the model parameters: vector of means $\mu$ and covariance matrix $\Sigma$

- In the multivariate Gaussian case,

$$p(x|\mu, \Sigma) = \frac{1}{\sqrt{2\pi\Sigma}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

# Log Likelihood

- If we use the log of likelihood, we end up with a <u>sum</u> instead of a product

$$ln\, p(X \mid \Theta) = \sum_{i=1}^{N} p(x_i \mid \Theta)$$

- In the Gaussian case

$$ln\, p(x|\mu,\ \Sigma) = -\frac{N}{2}\, ln\, |\Sigma| - \boxed{\frac{1}{2} \sum_{i=1}^{N} (x-\mu)^T \Sigma^{-1} (x-\mu)} + constant$$

# Weighted Error Function

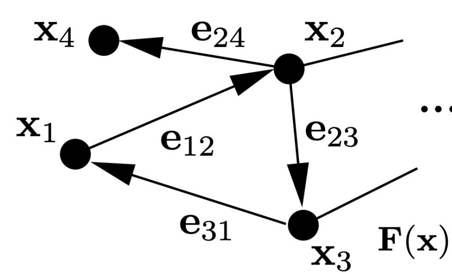
- Uncertainties in spatial loop closure and odometry constraints
- Weighted error function - a sum of squared non-linear terms

$$F(s) = \underbrace{\sum_{i,j} e_{ij}^{T} \Omega_{ij} e_{ij}}_{F_{ij}}$$

- Minimization problem

$$s^* = \underset{s}{argmin}\, F(s)$$

- Find values of $s$ for which the error is small
  - The whole state vector needs to be changed
  - Changing the pose of one node is not sufficient




$$\begin{aligned}\mathbf{F}(\mathbf{x}) = &\ \mathbf{e}_{12}^{\top}\,\mathbf{\Omega}_{12}\,\mathbf{e}_{12}\\ &+ \mathbf{e}_{23}^{\top}\,\mathbf{\Omega}_{23}\,\mathbf{e}_{23}\\ &+ \mathbf{e}_{31}^{\top}\,\mathbf{\Omega}_{31}\,\mathbf{e}_{31}\\ &+ \mathbf{e}_{24}^{\top}\,\mathbf{\Omega}_{24}\,\mathbf{e}_{24}\\ &+ \dots\end{aligned}$$

# Gauss-Newton - 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 the procedure until convergence

# Linearizing the Error Function

- Approximate the error around an initial guess $\hat{s}$ via Taylor expansion

$$e_{ij}(\hat{s} + \Delta s) \approxeq e_{ij}(\hat{s}) + J_{ij}\Delta s$$

$$with\ J_{ij} = \frac{\partial e_{ij}(s)}{\partial s}$$

- Does one error term $e_{ij}(s)$ depend on all state variables?
    - In the SLAM problem, the answer is NO
    - This error term only relates the 2 poses/variables to which this edge is connected
- What is the structure of the Jacobian?

# Jacobian - Partial Derivatives of Error Function

- Non-zero only in rows corresponding to $s_i$ and $s_j$

$$J_{ij} = \left( \frac{\delta e_{ij}}{\delta s_0}, \frac{\delta e_{ij}}{\delta s_1}, \ldots \boxed{\frac{\delta e_{ij}}{\delta s_i}}, \ldots \boxed{\frac{\delta e_{ij}}{\delta s_j}}, \ldots \frac{\delta e_{ij}}{\delta s_{N-1}}, \frac{\delta e_{ij}}{\delta s_N} \right)$$

$$= (0,\ 0,\ 0, \ldots,\ A_{ij},\ B_{ij},\ 0,\ 0,\ 0)$$

Sparse Jacobian

# Local Approximation of Error Function

$$
\begin{aligned}
F_{ij}(\hat{s} + \Delta s) &= e_{ij}(\hat{s} + \Delta s)^T\, \Omega_{ij}\, e_{ij}(\hat{s} + \Delta s) \\
&\simeq (e_{ij} + J_{ij}\Delta s)^T\, \Omega_{ij}\, (e_{ij} + J_{ij}\Delta s) \\
&= \underbrace{e_{ij}^T \Omega_{ij} e_{ij}}_{c_{ij}} + 2\underbrace{e_{ij}^T \Omega_{ij} J_{ij}}_{b_{ij}} \Delta s + \underbrace{J_{ij}^T \Omega_{ij} J_{ij}}_{H_{ij}} \Delta s \\
&= c_{ij} + 2b_{ij}\Delta s + \Delta s^T\, H_{ij}\Delta s
\end{aligned}
$$

# Linearized System

$$F(\hat{s} + \Delta s) = \sum_{ij} Fij(\hat{s} + \Delta s)$$

$$H = \sum_{ij} H_{ij}$$

$$= c + 2b^T \Delta s + \Delta s^T H \Delta s$$

- Quadratic form can be minimized in $\Delta s$ by solving

$$H \Delta s = -b$$

- Linearized Solution: Add initial guess with computed increments

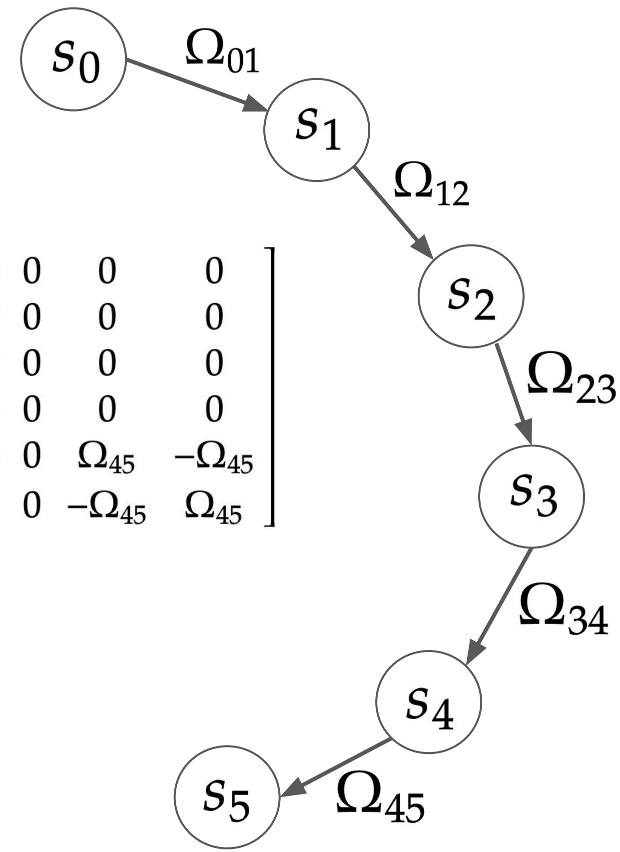
$$s^* = \hat{s} + \Delta s^*$$

# Example: H Matrix

$$\mathbf{H} = \mathbf{H}_{01} \;+ \mathbf{H}_{12} \;+ \mathbf{H}_{23} \;+ \mathbf{H}_{34} \;+ \mathbf{H}_{45} \;=$$

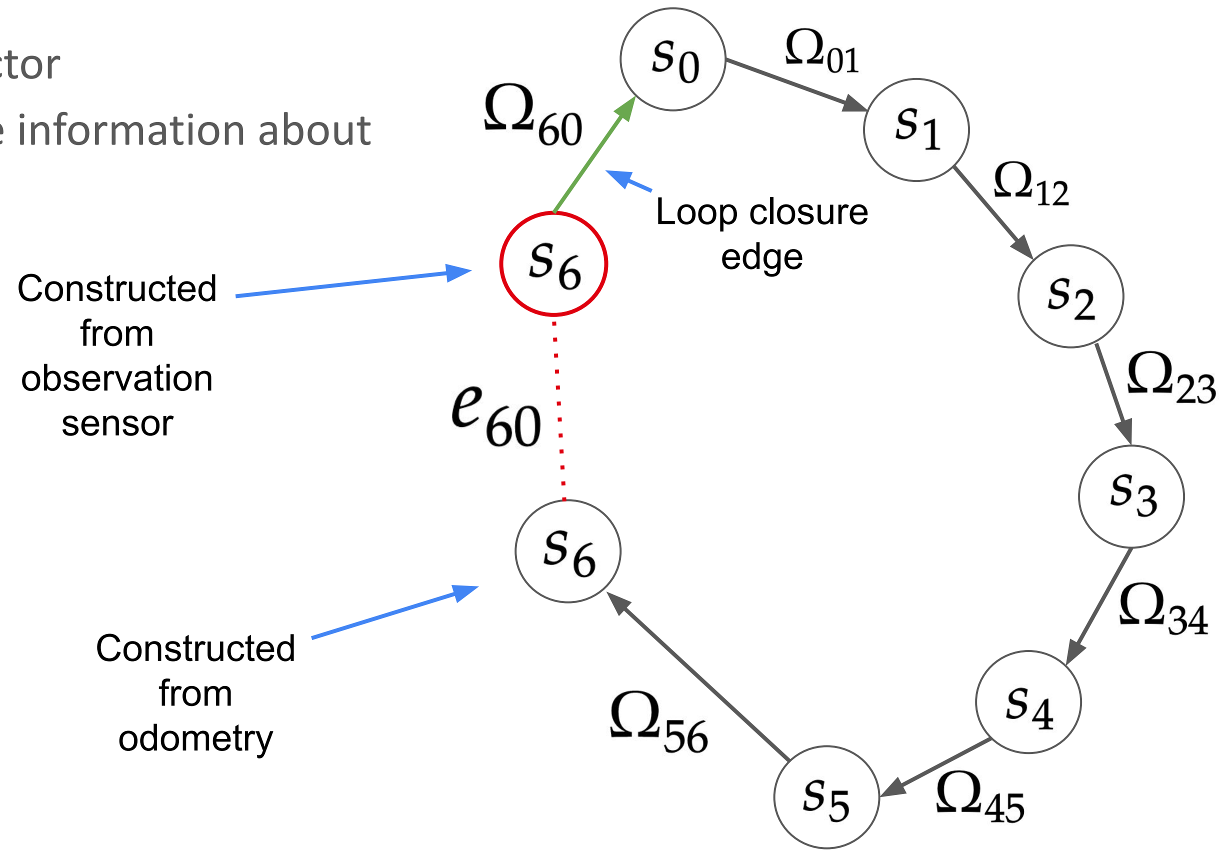
$$\begin{bmatrix} \Omega_{01} & -\Omega_{01} & 0 & 0 & 0 & 0 \\ -\Omega_{01} & \Omega_{01} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \Omega_{12} & -\Omega_{12} & 0 & 0 & 0 \\ 0 & -\Omega_{12} & \Omega_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \ldots + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Omega_{45} & -\Omega_{45} \\ 0 & 0 & 0 & 0 & -\Omega_{45} & \Omega_{45} \end{bmatrix}$$

$$= \begin{bmatrix} \Omega_{01} & -\Omega_{01} & 0 & 0 & 0 & 0 \\ -\Omega_{01} & \Omega_{01} + \Omega_{12} & -\Omega_{12} & 0 & 0 & 0 \\ 0 & -\Omega_{12} & \Omega_{12} + \Omega_{23} & -\Omega_{23} & 0 & 0 \\ 0 & 0 & -\Omega_{23} & \Omega_{23} + \Omega_{34} & \Omega_{34} & 0 \\ 0 & 0 & 0 & -\Omega_{34} & \Omega_{34} + \Omega_{45} & -\Omega_{45} \\ 0 & 0 & 0 & 0 & -\Omega_{45} & \Omega_{45} + \Omega_{56} \end{bmatrix}$$



# Example: Loop Closure

- Red $s_6$ not in state vector
- Shows mismatch in the information about that node



# Example: Linear System Construction after Loop Closure

$$H \, \Delta s \; = \; -b$$

$$\begin{bmatrix} \Omega_{01} + \Omega_{60} & -\Omega_{01} & 0 & 0 & 0 & 0 & -\Omega 60 \\ -\Omega_{01} & \Omega_{01} + \Omega_{12} & -\Omega_{12} & 0 & 0 & 0 & 0 \\ 0 & -\Omega_{12} & \Omega_{12} + \Omega_{23} & -\Omega_{23} & 0 & 0 & 0 \\ 0 & 0 & -\Omega_{23} & \Omega_{23} + \Omega_{34} & \Omega_{34} & 0 & 0 \\ 0 & 0 & 0 & -\Omega_{34} & \Omega_{34} + \Omega_{45} & -\Omega_{45} & 0 \\ 0 & 0 & 0 & 0 & -\Omega_{45} & \Omega_{45} + \Omega_{56} & -\Omega_{56} \\ -\Omega_{60} & 0 & 0 & 0 & 0 & -\Omega_{56} & \Omega_{56} + \Omega_{60} \end{bmatrix} \begin{bmatrix} \Delta s_0 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \\ \Delta s_4 \\ \Delta s_5 \\ \Delta s_6 \end{bmatrix} = \begin{bmatrix} e_{ij}\Omega_{61} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -e_{ij}\Omega_{61} \end{bmatrix}$$

- Diagonal Elements - Accumulated value of different edges
  - Total probability of correctness of that node in the graph
- Off-Diagonal Elements
  - Probability of correctness of two poses relative to each other

# Online SLAM - Graph Pruning

- Size of the H matrix grows linearly with the number of poses
  - Add a new odometry pose? Expand the matrix by 1 row + 1 column
- Running “full” graph-based SLAM online - Prune the graph
  - Fix the memory - The number of poses remembered by the graph

$$
\begin{bmatrix}
\Omega_{01} + \Omega_{60} & -\Omega_{01} & 0 & 0 & 0 & 0 & -\Omega 60 \\
-\Omega_{01} & \Omega_{01} + \Omega_{12} & -\Omega_{12} & 0 & 0 & 0 & 0 \\
0 & -\Omega_{12} & \Omega_{12} + \Omega_{23} & -\Omega_{23} & 0 & 0 & 0 \\
0 & 0 & -\Omega_{23} & \Omega_{23} + \Omega_{34} & \Omega_{34} & 0 & 0 \\
0 & 0 & 0 & -\Omega_{34} & \Omega_{34} + \Omega_{45} & -\Omega_{45} & 0 \\
0 & 0 & 0 & 0 & -\Omega_{45} & \Omega_{45} + \Omega_{56} & -\Omega_{56} \\
-\Omega_{60} & 0 & 0 & 0 & 0 & -\Omega_{56} & \Omega_{56} + \Omega_{60}
\end{bmatrix}
$$

# Integrate away the variable $s_1$ - Second-oldest pose

$$\begin{bmatrix} \Omega_{01}+\Omega_{60} & -\Omega_{01} & 0 & 0 & 0 & 0 & -\Omega 60 \\ -\Omega_{01} & \Omega_{01}+\Omega_{12} & -\Omega_{12} & 0 & 0 & 0 & 0 \\ 0 & -\Omega_{12} & \Omega_{12}+\Omega_{23} & -\Omega_{23} & 0 & 0 & 0 \\ 0 & 0 & -\Omega_{23} & \Omega_{23}+\Omega_{34} & \Omega_{34} & 0 & 0 \\ 0 & 0 & 0 & -\Omega_{34} & \Omega_{34}+\Omega_{45} & -\Omega_{45} & 0 \\ 0 & 0 & 0 & 0 & -\Omega_{45} & \Omega_{45}+\Omega_{56} & -\Omega_{56} \\ -\Omega_{60} & 0 & 0 & 0 & 0 & -\Omega_{56} & \Omega_{56}+\Omega_{60} \end{bmatrix} \begin{bmatrix} \Delta s_0 \\ \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \\ \Delta s_4 \\ \Delta s_5 \\ \Delta s_6 \end{bmatrix}$$

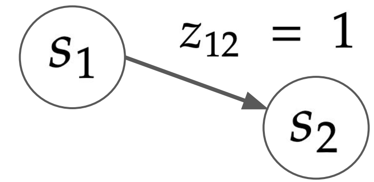
$$\begin{array}{ccc} \begin{array}{cc} B & A \end{array} & & C \\ \left[\begin{array}{c|cc} & \Omega_{01}+\Omega_{12} & \begin{matrix} -\Omega_{12} & 0 & 0 & 0 & 0 \end{matrix} \\ \hline A^{-1} & \begin{matrix} -\Omega_{12} \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} & \Omega' \end{array}\right] & & \left[\begin{array}{c} \Delta s_1 \\ \hline \xi' \end{array}\right] \end{array}$$

$$\Omega = \Omega' - A^T B^{-1} A \qquad \xi = \xi' - A^T B^{-1} C$$

# Algorithm

```
optimize(s)
    while(!converged):
        (H, b) = BuildLinearSystem(s)
        Δs = SolveSparse(H Δs = −b)
        s = s + Δs
    end
return s
```

# Trivial 1D Example




$$
\begin{aligned}
s &= (s_1,\ s_2)^T = (0,\ 0) \\
z_{12} &= 1 \\
\Omega &= 2 \\
e_{12} &= z_{12} - (x_2 - x_1) = 1 - (0 - 0) = 1 \\
J_{12} &= (1 \quad -1) \\
b_{12}^T &= e_{12}^T \Omega_{12} J_{12} = (2 \quad -2) \\
H_{12} &= J_{12}^T \Omega_{12} J_{12} = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} \\
\Delta s &= -H_{12}^{-1} b_{12} = ???
\end{aligned}
$$

$$\boxed{|H_{12}| = 0}$$

# Role of a Prior

- Constraints are <u>relative</u>
- We don't know where $s_1$ and $s_2$ actually are in a global reference frame
- **One node needs to be fixed - Create a 'prior' node**
  - Constraint of a gaussian distribution
  - Fixes the first node as the reference frame with a certain uncertainty
  - Additional constraint to our H matrix, setting $\Delta s_1 = 0$

$$H_{12} = J_{12}^T \Omega_{12} J_{12} = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} + \boxed{\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}} = \begin{bmatrix} 3 & -2 \\ -2 & 2 \end{bmatrix}$$

$$\Delta s = -H_{12}^{-1} b_{12} = -1 \begin{bmatrix} 3 & -2 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ -2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

- Updated State Vector

$$s = s + \Delta s = \begin{bmatrix} 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

# References


- G. Grisetti, R. Kümmerle, C. Stachniss and W. Burgard, "**A Tutorial on Graph-Based SLAM**," in *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31-43, winter 2010, doi: 10.1109/MITS.2010.939925.
- Appel, R.N. and Folmer, H.H (2016) "**Analysis, optimization, and design of a SLAM solution for an implementation on reconfigurable hardware (FPGA) using CλaSH.**"
- C. Cadena *et al*., "**Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age**," in *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309-1332, Dec. 2016, doi: 10.1109/TRO.2016.2624754
- Thrun S, Montemerlo M. "**The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures**". *The International Journal of Robotics Research*. 2006;25(5-6):403-429. doi:10.1177/0278364906065387
- Cyrill Stachniss, "**Graph-based SLAM using Pose Graphs (Cyrill Stachniss, 2020)**" https://www.youtube.com/watch?v=uHbRKvD8TWg