ELEC 3210
Introduction to Mobile Robotics
Lecture 12

(Machine Learning and Infomation Processing for Robotics)

Huan YIN

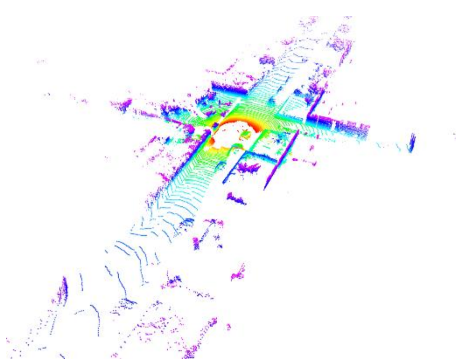
Research Assistant Professor, Dept. of ECE

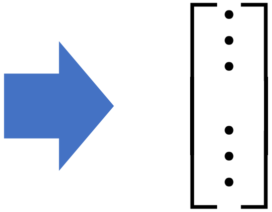
eehyin@ust.hk

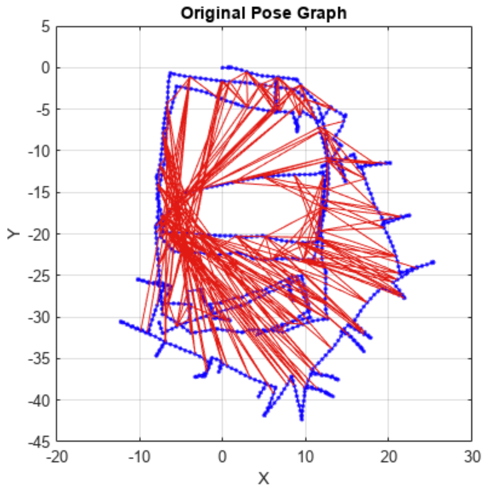



Recap L11

- Place recognition
  - Have I been the place before?
  - Data retrieval problem
  - LiDAR PR- Scan Context
- Close the loop for SLAM





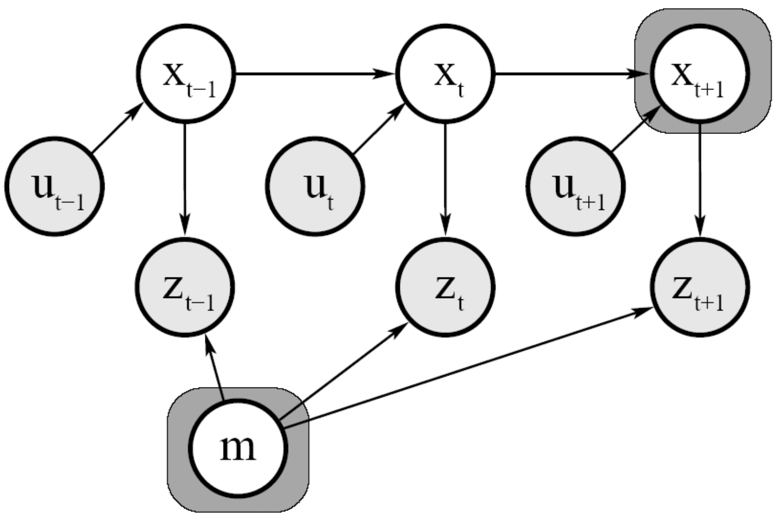


LiDAR Point Cloud **Descriptor** **Connected Poses**

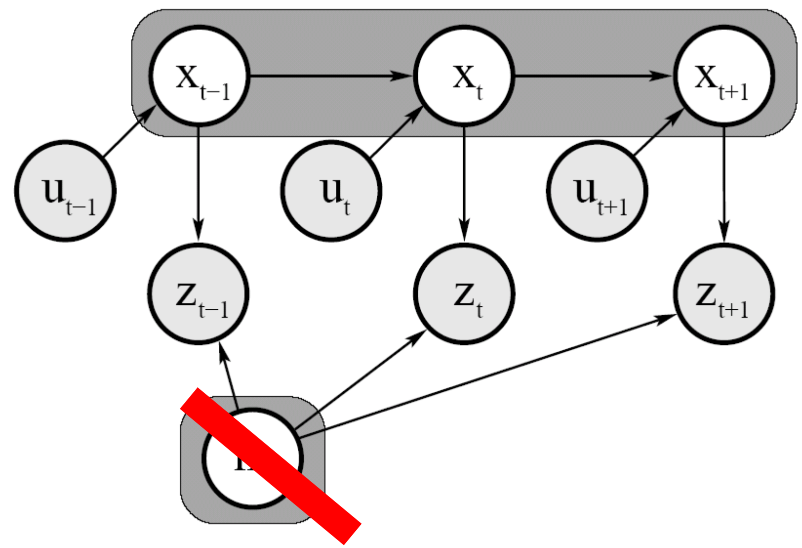
Pose Graph Today

- Achieve global consistent mapping with loops
- From recursive filter to batch processing

$$p(x_t, m \mid z_{1:t}, u_{1:t})$$



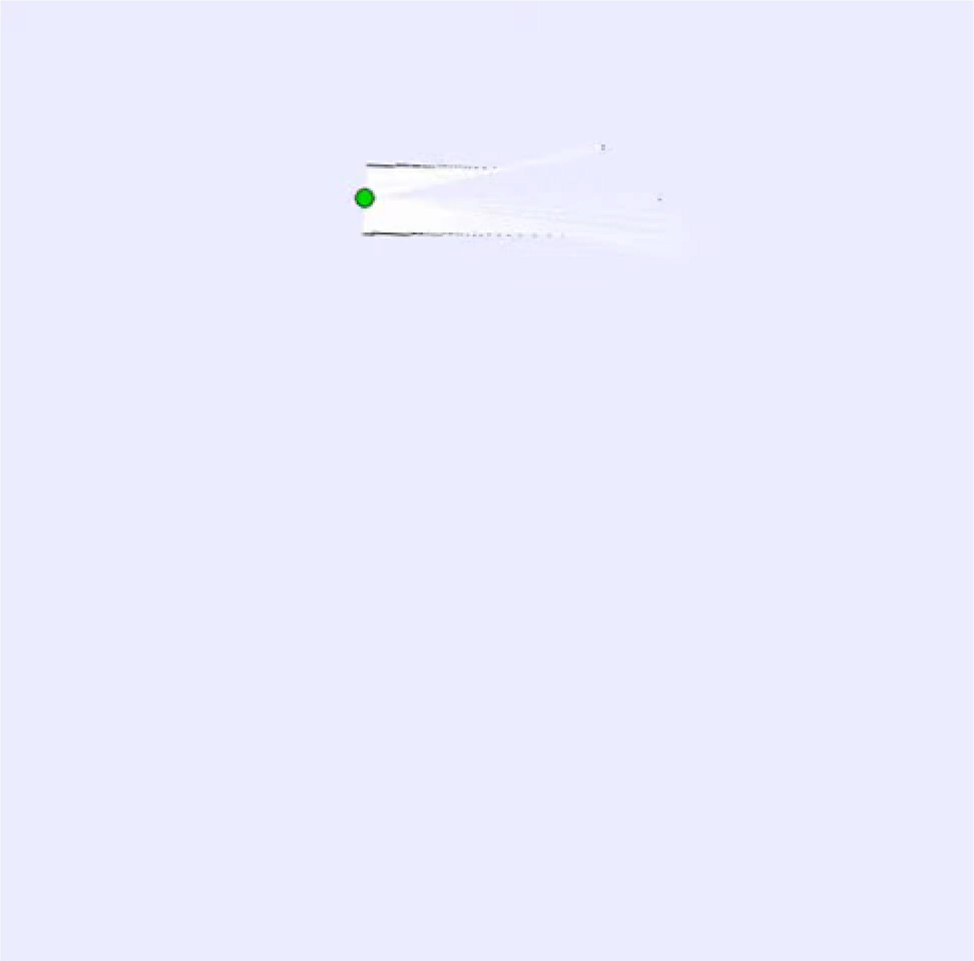
$$p(x_{1:t}, \cancel{m} \mid z_{1:t}, u_{1:t})$$



Courtesy: Thrun, Burgard, and Fox

Close the Loop

- 2D laser mapping



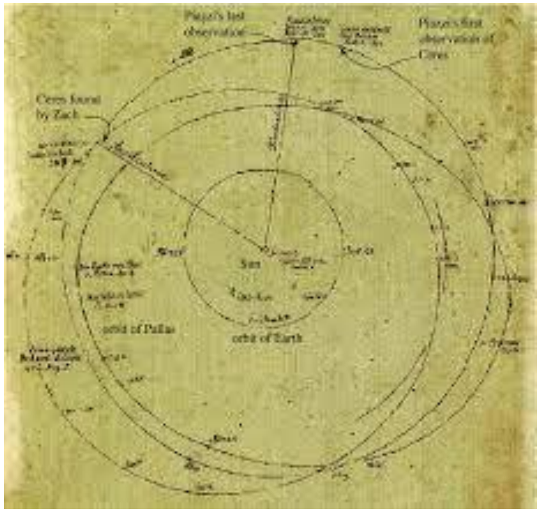
Courtesy: Cyrill Stachniss

Prior - Least Squares

Least Squares History

- Method developed by Carl Friedrich Gauss in 1795
- (he was 18 years old)
- First showcase: predicting the future location of the asteroid Ceres in 1801

Gauss, 1840



Courtesy: Cyrill Stachniss and Wikipedia

Problem Definition

- Given a system described by a set of n observation functions

$$\{f_i(\mathbf{x})\}_{i=1:n}$$

- Let
  - $\mathbf{x}$ be the state vector
  - $\mathbf{z}_i$ be a measurement of the state $\mathbf{x}$
  - $\hat{\mathbf{z}}_i = f_i(\boldsymbol{x})$ be a function which maps $\mathbf{x}$ to a predicted measurement $\hat{\mathbf{z}}_i$
- Given n noisy measurements $\mathbf{z}_{1:n}$ about the state $\mathbf{x}$
- **Goal:** Estimate the state $\mathbf{x}$ which bests "explains" the measurements $\mathbf{z}_{1:n}$

Courtesy: Cyrill Stachniss

Graphical Explanation

- Multiple Measurements, Multiple Constraints
- Different from EKF SLAM in Lecture 10

$$
\mathbf{x} \quad
\begin{array}{lcl}
f_1(\mathbf{x}) = \hat{\mathbf{z}}_1 & \qquad & \mathbf{z}_1 \\
f_2(\mathbf{x}) = \hat{\mathbf{z}}_2 & \qquad & \mathbf{z}_2 \\
\qquad\quad \cdots & & \\
f_n(\mathbf{x}) = \hat{\mathbf{z}}_n & \qquad & \mathbf{z}_n
\end{array}
$$

Unknown State	Predicted Measurements	Real Measurements

Courtesy: Cyrill Stachniss

Error Function

- Error $\mathbf{e}_i$ is typically the difference between the predicted and actual measurement

$$\mathbf{e}_i(\mathbf{x}) = \mathbf{z}_i - f_i(\mathbf{x})$$

- We assume that the error has zero mean and is normally distributed
- Gaussian error with information matrix $\mathbf{\Omega}_i$
  - $\mathbf{\Omega}_i = \mathbf{\Sigma}_i^{-1}$, encodes the "weights" of errors
- The squared error of a measurement depends only on the state and is a scalar

Courtesy: Cyrill Stachniss

Goal: Find the Minimum

- Find the state $\mathbf{x}^*$ which minimizes the error given all measurements

$$
\begin{aligned}
\mathbf{x}^* &= \underset{\mathbf{x}}{\operatorname{argmin}}\, F(\mathbf{x}) && \text{global error (scalar)} \\
&= \underset{\mathbf{x}}{\operatorname{argmin}} \sum_i e_i(\mathbf{x}) && \text{squared error terms (scalar)} \\
&= \underset{\mathbf{x}}{\operatorname{argmin}} \sum_i \mathbf{e}_i^T(\mathbf{x})\boldsymbol{\Omega}_i \mathbf{e}_i(\mathbf{x})
\end{aligned}
$$

error terms (vector)

Courtesy: Cyrill Stachniss

How?

Assume we have a "good" initial guess

Solve Via Iterative Local Linearizations

- Linearize the error terms around the current solution/initial guess
- Compute the first derivative (Jacobian)of the squared error function
- Set it to zero and solve linear system
- Obtain the new state (that is hopefully closer to the minimum)
- Iterate

Courtesy: Cyrill Stachniss

Linearizing the Error Function

- Approximate the error functions around an initial guess $\mathbf{x}$ via Taylor expansion (Used in EKF, Lecture 9)

$$\mathbf{e}_i(\mathbf{x}+\Delta\mathbf{x}) \simeq \underbrace{\mathbf{e}_i(x)}_{\mathbf{e}_i} + \mathbf{J}_i(\mathbf{x})\Delta\mathbf{x}$$

- Reminder: Jacobian

$$\mathbf{J}_f(x) = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \cdots & \frac{\partial f_2(x)}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_m(x)}{\partial x_1} & \frac{\partial f_m(x)}{\partial x_2} & \cdots & \frac{\partial f_m(x)}{\partial x_n} \end{pmatrix}$$

Courtesy: Cyrill Stachniss

One Term - Squared Error (1)

- With the previous linearization, we can fix $\mathbf{x}$ and carry out the minimization in the increments $\mathbf{\Delta x}$
- We replace the Taylor expansion in the squared error terms:

$$
\begin{aligned}
e_i(\mathbf{x}+\Delta\mathbf{x}) =& \mathbf{e}_i^T(\mathbf{x}+\Delta\mathbf{x})\Omega_i\mathbf{e}_i(\mathbf{x}+\Delta\mathbf{x}) \\
\simeq& \left(\mathbf{e}_i+\mathbf{J}_i\Delta\mathbf{x}\right)^T \Omega_i \left(\mathbf{e}_i+\mathbf{J}_i\Delta\mathbf{x}\right) \\
=& \mathbf{e}_i^T\Omega_i\mathbf{e}_i + \\
& \mathbf{e}_i^T\Omega_i\mathbf{J}_i\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{J}_i^T\Omega_i\mathbf{e}_i + \\
& \Delta\mathbf{x}^T\mathbf{J}_i^T\Omega_i\mathbf{J}_i\Delta\mathbf{x}
\end{aligned}
$$

Courtesy: Cyrill Stachniss

One Term - Squared Error (2)

- All summands are scalar so the transposition has no effect
- By grouping similar terms, we obtain:

$$
\begin{aligned}
e_i(\mathbf{x} + \Delta\mathbf{x}) & \\
&\simeq \mathbf{e}_i^T \Omega_i \mathbf{e}_i + \\
&\quad \mathbf{e}_i^T \Omega_i \mathbf{J}_i \Delta\mathbf{x} + \Delta\mathbf{x}^T \mathbf{J}_i^T \Omega_i \mathbf{e}_i + \\
&\quad \Delta\mathbf{x}^T \mathbf{J}_i^T \Omega_i \mathbf{J}_i \Delta\mathbf{x} \\
&= \underbrace{\mathbf{e}_i^T \Omega_i \mathbf{e}_i}_{c_i} + 2 \underbrace{\mathbf{e}_i^T \Omega_i \mathbf{J}_i}_{\mathbf{b}_i^T} \Delta\mathbf{x} + \Delta\mathbf{x}^T \underbrace{\mathbf{J}_i^T \Omega_i \mathbf{J}_i}_{\mathbf{H}_i} \Delta\mathbf{x} \\
&= c_i + 2\mathbf{b}_i^T \Delta\mathbf{x} + \Delta\mathbf{x}^T \mathbf{H}_i \Delta\mathbf{x}
\end{aligned}
$$

Courtesy: Cyrill Stachniss

Global Squared Error

- The global error is the sum of the squared errors terms corresponding to the individual measurements

$$
\begin{aligned}
F(\mathbf{x}+\Delta\mathbf{x}) &\simeq \sum_i \left(c_i + \mathbf{b}_i^T \Delta\mathbf{x} + \Delta\mathbf{x}^T \mathbf{H}_i \Delta\mathbf{x}\right) \\
&= \underbrace{\sum_i c_i}_{c} + 2(\underbrace{\left(\sum_i \mathbf{b}_i^T\right)}_{\mathbf{b}^T} \Delta\mathbf{x} + \Delta\mathbf{x}^T \underbrace{\left(\sum_i \mathbf{H}_i\right)}_{\mathbf{H}} \Delta\mathbf{x} \\
&= c + 2\mathbf{b}^T \Delta\mathbf{x} + \Delta\mathbf{x}^T \mathbf{H} \Delta\mathbf{x}
\end{aligned}
$$

with

$$
\mathbf{b}^T = \sum_i \mathbf{e}_i^T \boldsymbol{\Omega}_i \mathbf{J}_i
$$

$$
\mathbf{H} = \sum_i \mathbf{J}_i^T \boldsymbol{\Omega} \mathbf{J}_i
$$

Courtesy: Cyrill Stachniss

Deriving the Quadratic Form

- Deriving the global linearized error w.r.t. $\Delta\mathbf{x}$

$$\frac{\partial F(\mathbf{x}+\Delta\mathbf{x})}{\partial \Delta\mathbf{x}} \simeq 2\mathbf{b} + 2\mathbf{H}\Delta\mathbf{x}$$

- Deriving a Quadratic Form

$$0 = 2\mathbf{b} + 2\mathbf{H}\Delta\mathbf{x}$$

- Which leads to the linear system

$$\mathbf{H}\Delta\mathbf{x} = -\mathbf{b}$$

- The solution for the increment $\Delta\mathbf{x}^*$ is

$$\Delta\mathbf{x}^* = -\mathbf{H}^{-1}\ \mathbf{b}$$

- (The Matrix Cookbook, Section 2.2.4)

Courtesy: Cyrill Stachniss

Gauss-Newton Solution

Iterate the following steps:

- Linearize around $\mathbf{x}$ and compute for each measurement

$$\mathbf{e}_i(\mathbf{x} + \Delta\mathbf{x}) \simeq \mathbf{e}_i(\mathbf{x}) + \mathbf{J}_i \Delta\mathbf{x}$$

- Compute the terms for the linear system

$$\mathbf{b}^T = \sum_i \mathbf{e}_i^T \mathbf{\Omega}_i \mathbf{J}_i$$

$$\mathbf{H} = \sum_i \mathbf{J}_i^T \mathbf{\Omega}_i \mathbf{J}_i$$

- Solve the linear system

$$\Delta\mathbf{x}^* = -\mathbf{H}^{-1}\mathbf{b}$$

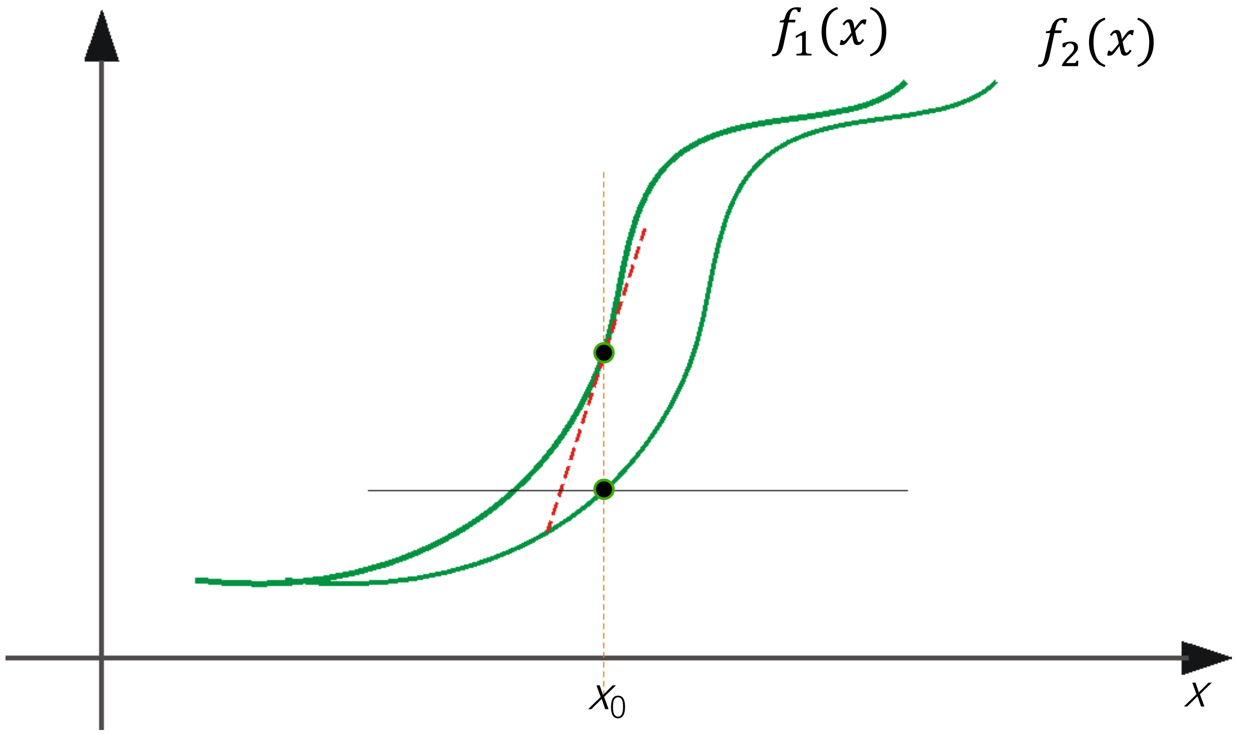
- Updating state

$$\mathbf{x} \leftarrow \mathbf{x} + \Delta\mathbf{x}^*$$

Courtesy: Cyrill Stachniss

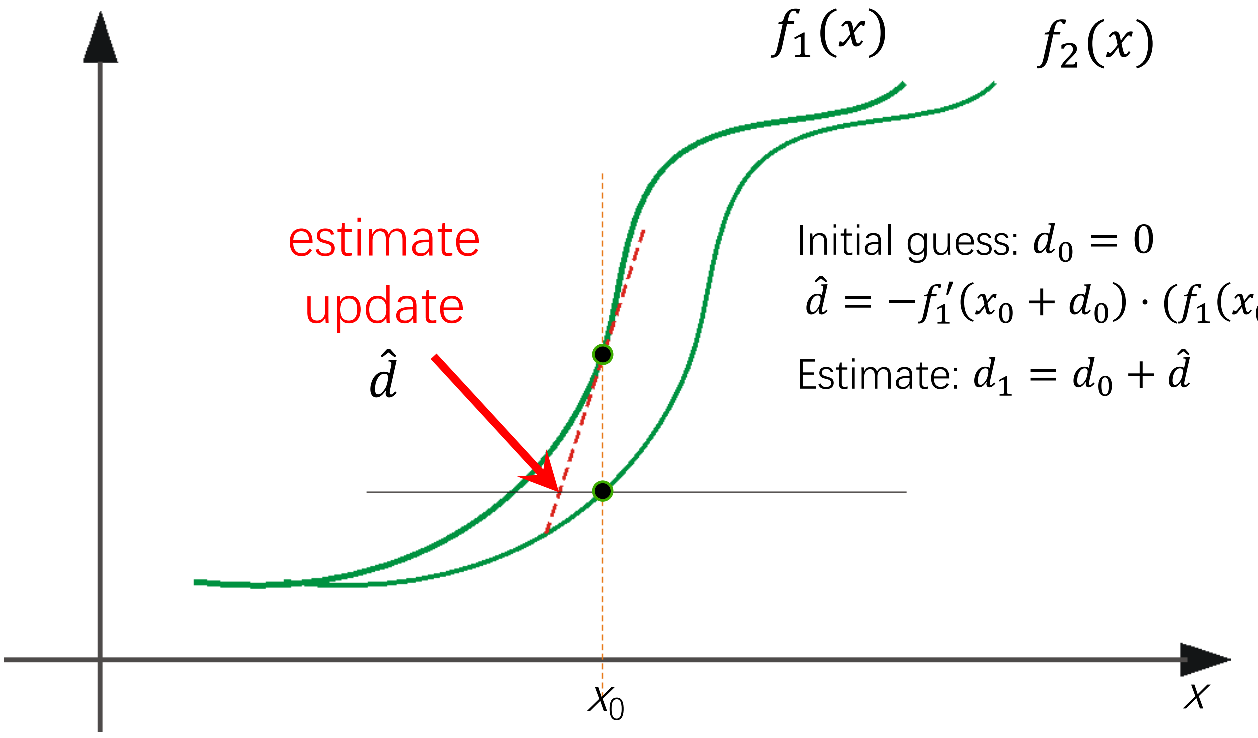
Graphical Understanding

- Assume in 1-Dimensional problem
- Compute d to minimize $\| f_1(x+d) - f_2(x) \|^2$



Gauss Newton Method

- Compute d to minimize $\| f_1(x+d) - f_2(x) \|^2$



Initial guess: $d_0 = 0$

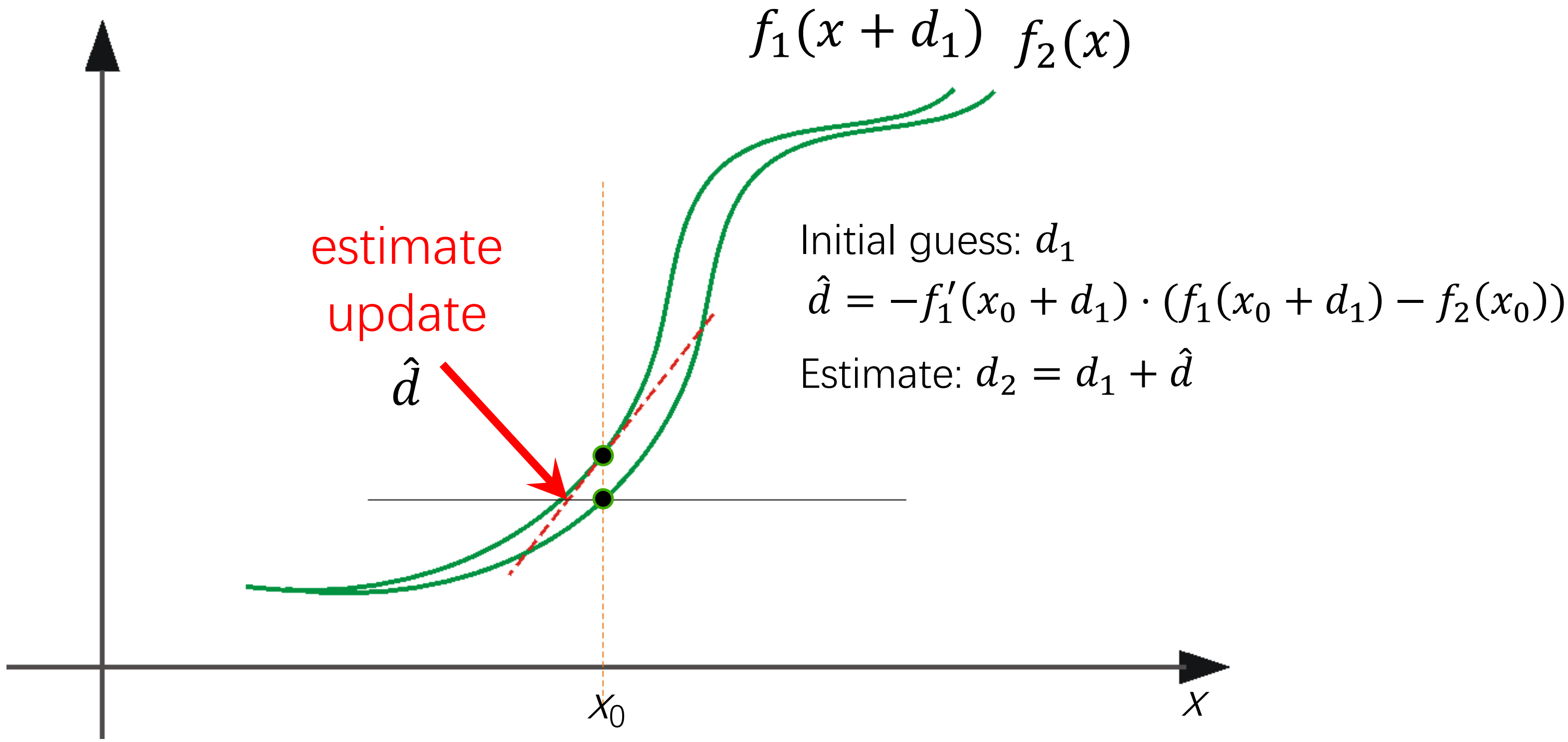
$\hat{d} = -f_1'(x_0 + d_0) \cdot (f_1(x_0 + d_0) - f_2(x_0))$

Estimate: $d_1 = d_0 + \hat{d}$

Courtesy: Shaojie Shen

Gauss Newton Method

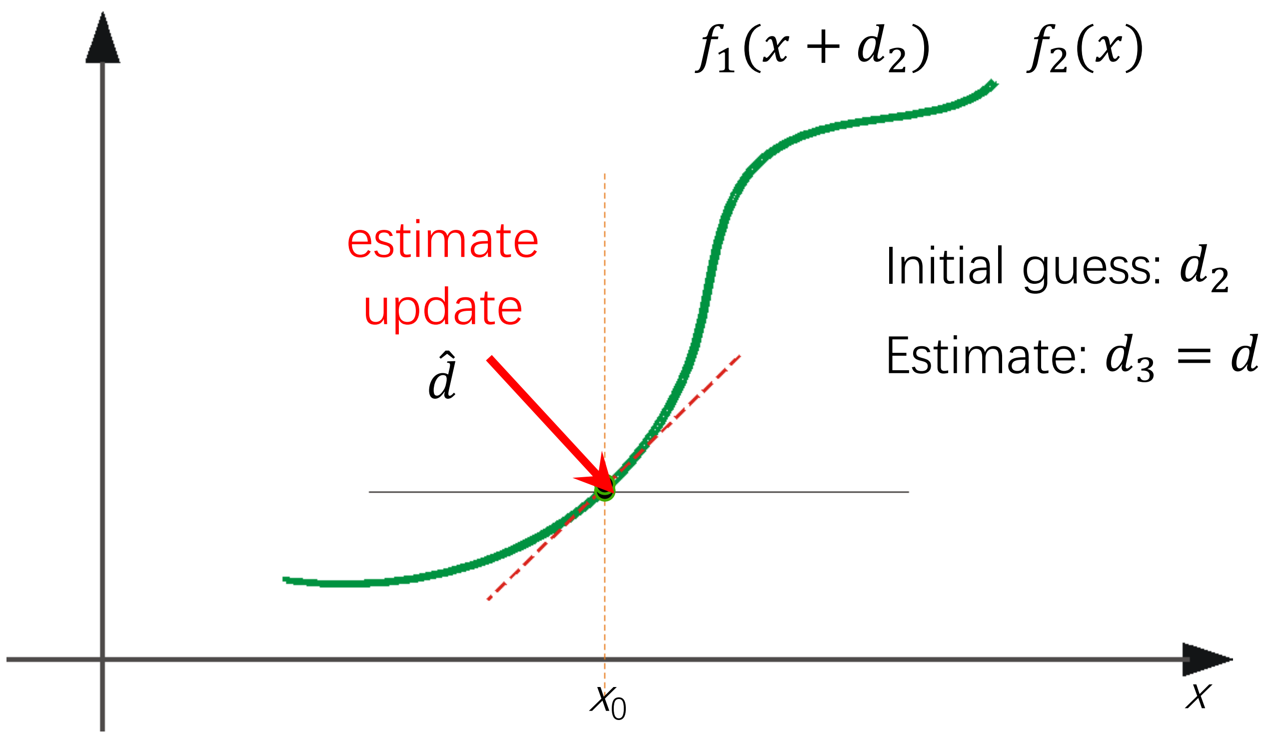
- Compute d to minimize $\|f_1(x+d) - f_2(x)\|^2$



Courtesy: Shaojie Shen

Gauss Newton Method

- Compute d to minimize $\| f_1(x+d) - f_2(x) \|^2$



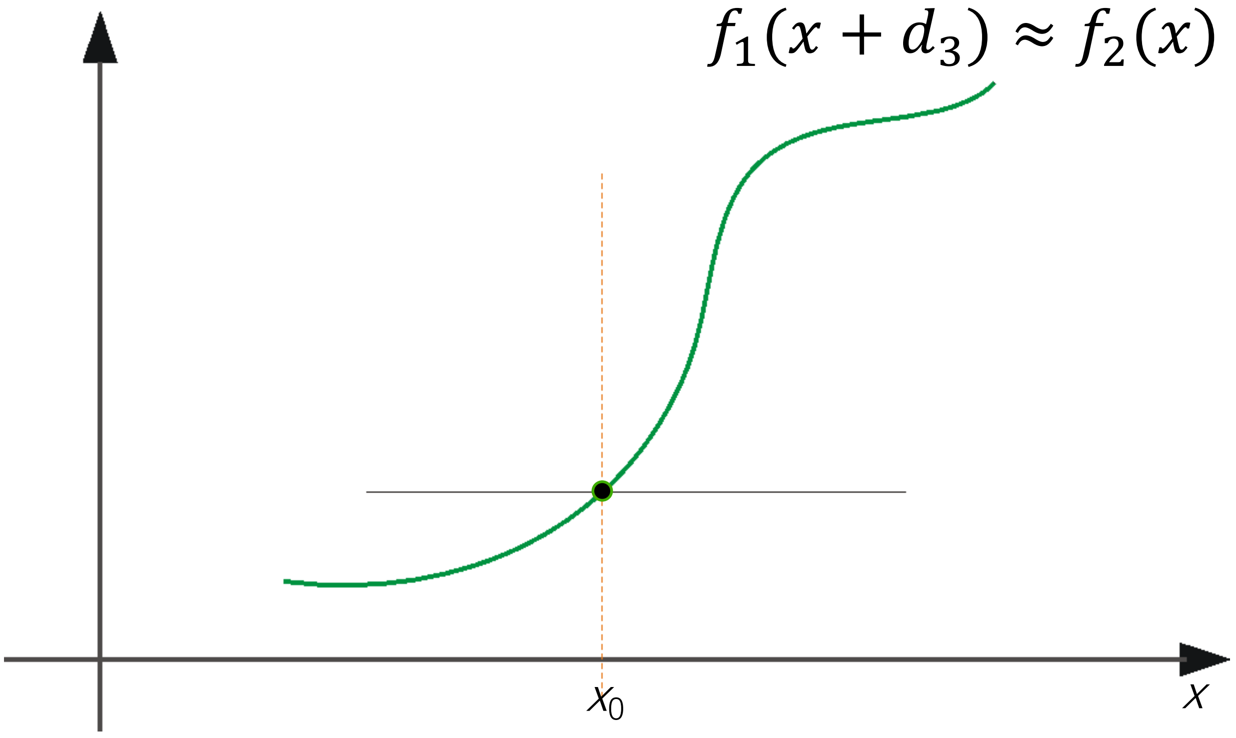
Initial guess: d_2

Estimate: $d_3 = d_2 + \hat{d}$

Courtesy: Shaojie Shen

Gauss Newton Method

- Compute d to minimize $\| f_1(x+d) - f_2(x) \|^2$



Courtesy: Shaojie Shen

Least Squares in General

- Approach for computing a solution for an overdetermined system
- "More equations than unknowns"
- Minimizes the sum of the squarederrors in the equations
- Standard approach to a large set of problems
- Equivalent to maximizing the log likelihood of independent Gaussians! (Relation to Probabilistic State Estimation)
- Today: Application to Pose Graph SLAM

Courtesy: Cyrill Stachniss

Relation to Probabilistic Estimation

- Bayes rule, independence and Markov assumptions allow us to write

$$p(x_{0:t} \mid z_{1:t}, u_{1:t}) = \eta p(x_0) \prod_t \left[p(x_t \mid x_{t-1}, u_t) \, p(z_t \mid x_t) \right]$$

- maximize the $p(x_{1:t} \mid z_{1:t}, u_{1:t})$
- Written as the log likelihood, leads to

$$\log p(x_{0:t} \mid z_{1:t}, u_{1:t}) = \text{const.} + \log p(x_0) + \sum_t \left[\log p(x_t \mid x_{t-1}, u_t) + \log p(z_t \mid x_t) \right]$$

Courtesy: Cyrill Stachniss

Gaussian Distributions

- Assume Gaussian sistributions,

$$\log p\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right) = \text{const.} + \log \underbrace{p\left(x_0\right)}_{\mathcal{N}} + \sum_t \left[\log \underbrace{p\left(x_t \mid x_{t-1}, u_t\right)}_{\mathcal{N}} + \log \underbrace{p\left(z_t \mid x_t\right)}_{\mathcal{N}}\right]$$

- Log likelihood of a Gaussian, constant equivalent to the error functions used before

$$\log \mathcal{N}(x, \mu, \Sigma) = \text{const.} - \frac{1}{2} \underbrace{\underbrace{(x-\mu)^T}_{\mathbf{e}^T(x)} \underbrace{\Sigma^{-1}}_{\Omega} \underbrace{(x-\mu)}_{\mathbf{e}(x)}}_{e(x)}$$

Courtesy: Cyrill Stachniss

Gaussian Distributions

- Assuming Gaussian distributions

$$\log p\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right)$$
$$= \text{ const. } -\frac{1}{2}e_p(x) - \frac{1}{2}\sum_t \left[e_{u_t}(x) + e_{z_t}(x)\right]$$

- Maximizing the log likelihood leads to

$$\operatorname{argmax} \log p\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right)$$
$$= \operatorname{argmin} e_p(x) + \sum_t \left[e_{u_t}(x) + e_{z_t}(x)\right]$$

Courtesy: Cyrill Stachniss

Conclusion

- **minimizing the squared error is equivalent to maximimzing the log likelihood of independent Gaussian distributions!**

- with individual error terms for the motions, measurements, and prior

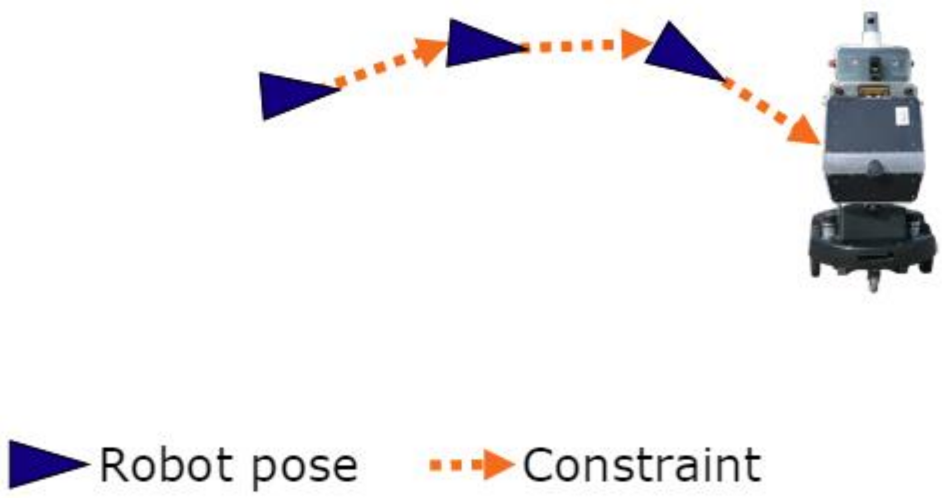
$$\operatorname{argmax} \log p\left(x_{0:t} \mid z_{1:t}, u_{1:t}\right)$$
$$= \operatorname{argmin} e_p(x) + \sum_t \left[e_{u_t}(x) + e_{z_t}(x)\right]$$

Courtesy: Cyrill Stachniss

Pose Graph SLAM

Graph-Based SLAM

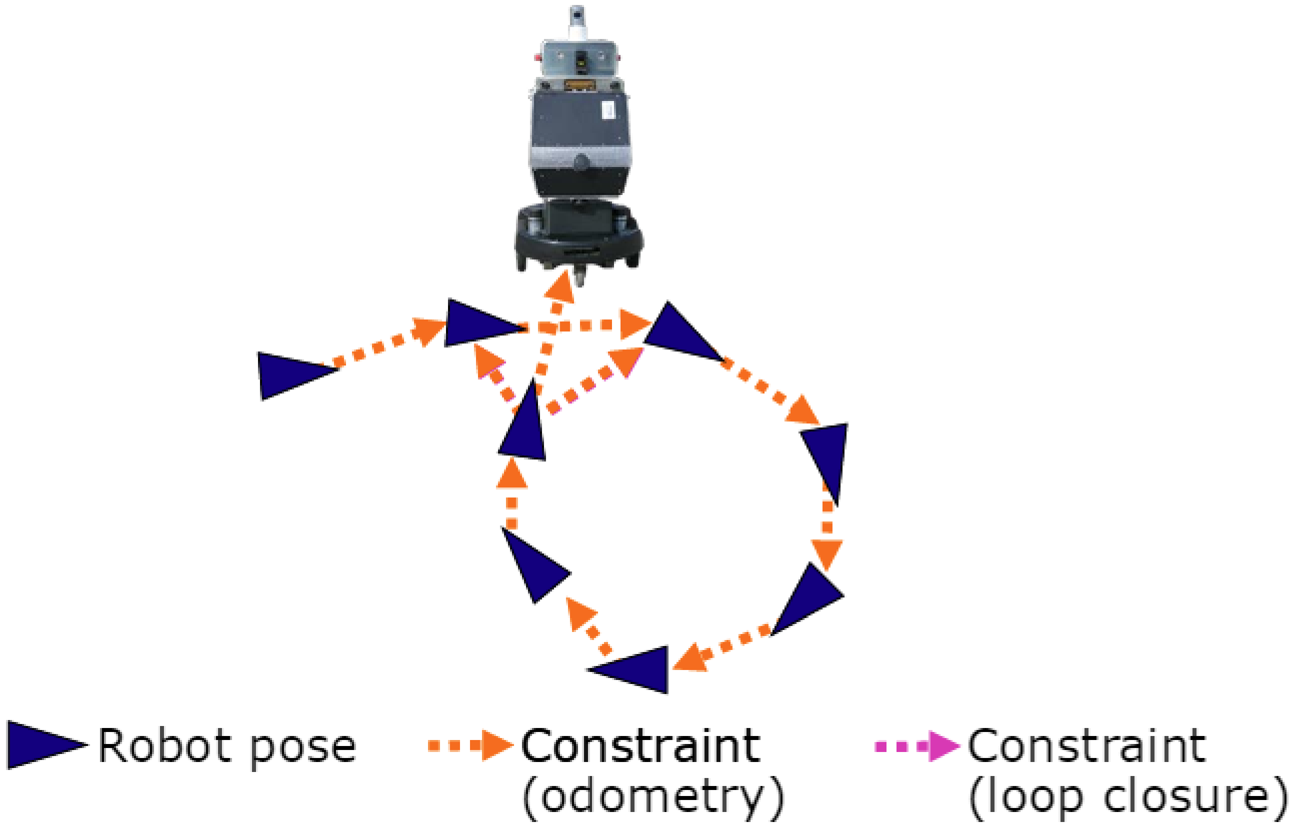
- Constraints connect the poses of therobot while it is moving
- Constraints are inherently uncertain



Courtesy: Wolfram Burgard

Graph-Based SLAM

- Observing previously seen areas generates constraints between non-successive poses (Close the Loop)



Courtesy: Wolfram Burgard

Idea of Graph SLAM

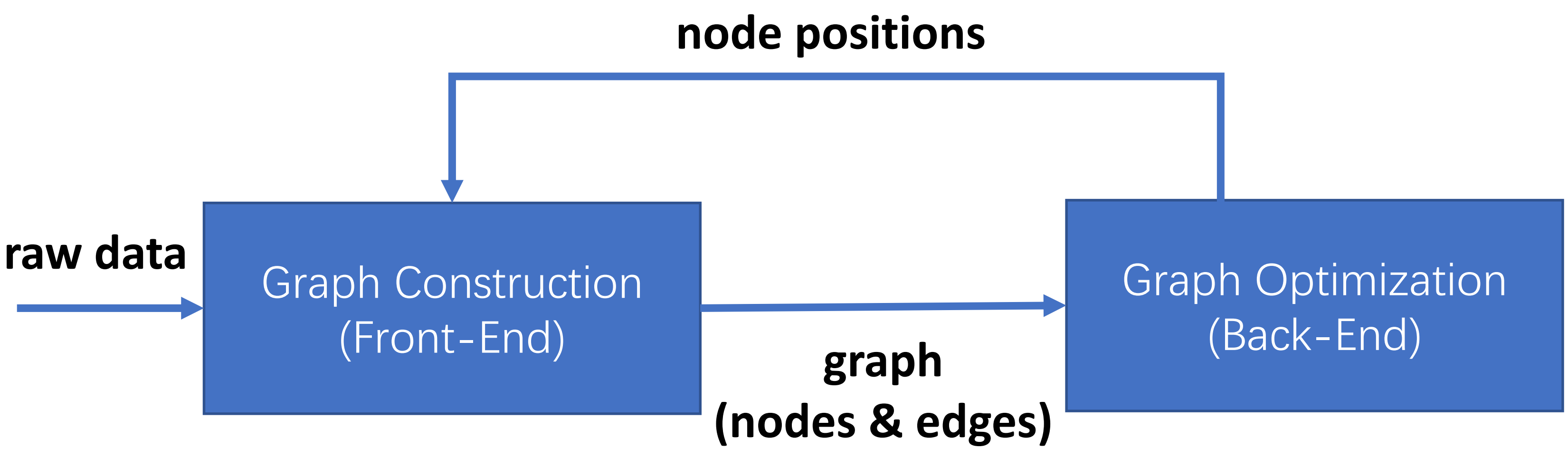
- Use a **graph** to represent the problem
- Every **node** in the graph correspondsto a pose of the robot during mapping
- Every **edge** between two nodes corresponds to a spatial constraint between them

- **Graph-Based SLAM:**
- Build the graph and find a node configuration thatminimize the error introduced by theconstraints

Courtesy: Cyrill Stachniss

The Overall SLAM System

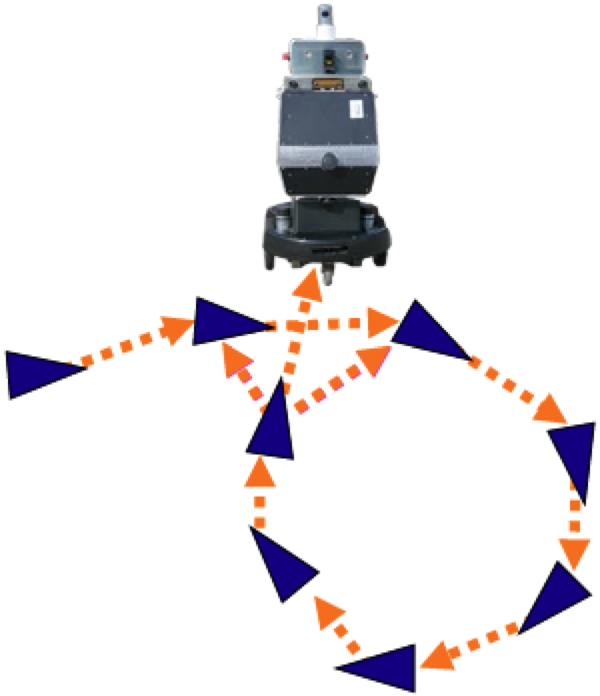
- Interplay of front-end and back-end
- A consistent map helps to determine newconstraints by reducing the search space
- This lecture focuses only on the optimization



Courtesy: Cyrill Stachniss

The Graph

- It consists of n nodes $\mathbf{x} = \mathbf{x}_{1:n}$
- Each $\mathbf{x}_i$ is a 2D or 3D transformation(the pose of the robot at time t_i)
- A constraint/edge exists between thenodes $\mathbf{x}_i$ and $\mathbf{x}_j$, if …



Courtesy: Cyrill Stachniss

Create an Edge if … (1)

- … the robot moves from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$
- Edge corresponds to odometry
  - Lecture 3, Forward Kinematics
  - Lecture 5, Iterative Closest Points
  - Lecture 10, Velocity-based Odometry
  - etc.

Courtesy: Cyrill Stachniss

Create an Edge if … (2)

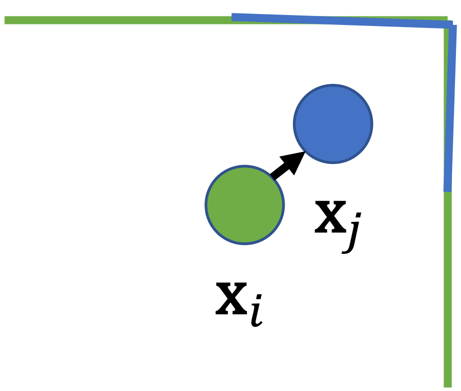
- … the robot observes the same part of the environment from $\mathbf{x}_i$ and from $\mathbf{x}_j$
- Construct a virtual measurement about the position of $\mathbf{x}_j$ seen from $\mathbf{x}_i$
  - Lecture 11, Place Recognition
  - Lecture 5, Iterative Closest Points
  - etc.

Measurement from $\mathbf{x}_i$ **Measurement from $\mathbf{x}_j$**

Courtesy: Cyrill Stachniss

Create an Edge if … (2)

- … the robot observes the same part of the environment from $\mathbf{x}_i$ and from $\mathbf{x}_j$
- Construct a virtual measurement about the position of $\mathbf{x}_j$ seen from $\mathbf{x}_i$
  - Lecture 11, Place Recognition
  - Lecture 5, Iterative Closest Points
  - etc.



Edge represents the position of $\mathbf{x}_j$ seen from $\mathbf{x}_i$ based on the observation

Courtesy: Cyrill Stachniss

Transformations

- Transformations can be expressed using homogenous coordinates
  - Lecutre 2, Pose and Rotations
- Odometry-based edge

$$\left(\mathbf{X}_i^{-1}\mathbf{X}_{i+1}\right)$$

- Observation-based edge

$$\left(\mathbf{X}_i^{-1}\mathbf{X}_j\right)$$

Courtesy: Cyrill Stachniss

Homogenous Coordinates

- H.C. are a system of coordinates usedin projective geometry
- Projective geometry is an alternative representation of geometric objectsand transformations
- N-dim space expressed in N+1 dim
- 4 dim. for modeling the 3D space
- Translation:

$$T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

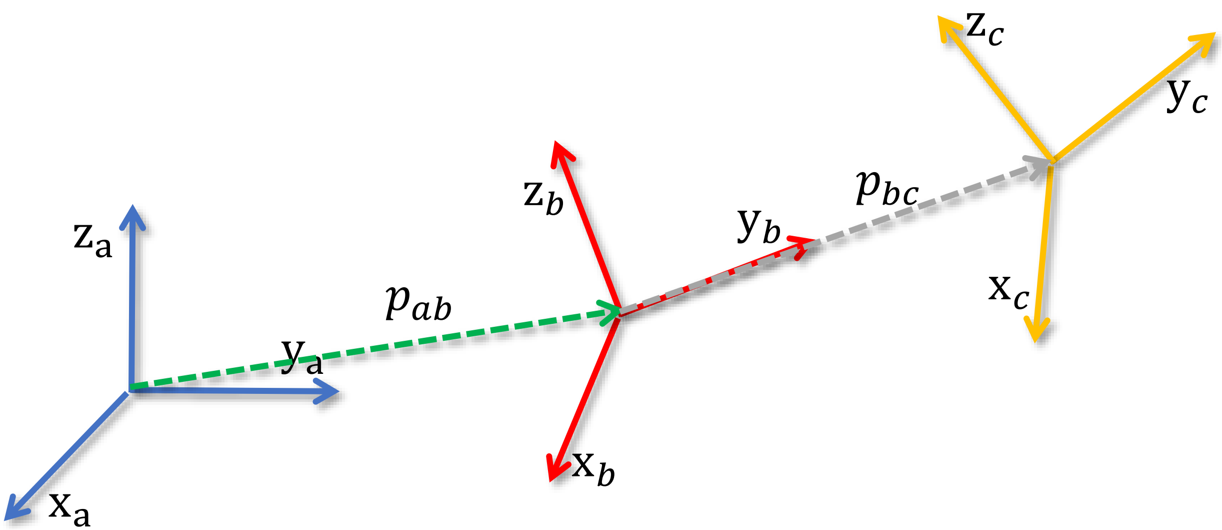
- Rotation:

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

Courtesy: Cyrill Stachniss

Recap L2 - Rigid Body Motion

- Homogeneous representation of rigid body motion:
  - $\bar{g}_{ab} = \begin{bmatrix} R_{ab} & p_{ab} \\ 0 & 1 \end{bmatrix}$
- Composition rule for rigid body motions:
  - $\bar{g}_{ac} = \bar{g}_{ab} \cdot \bar{g}_{bc} = \begin{bmatrix} R_{ab}R_{bc} & R_{ab}p_{bc} + p_{ab} \\ 0 & 1 \end{bmatrix}$
  - Compare with composition of rotational motion: $R_{ac} = R_{ab} \cdot R_{bc}$



Courtesy: Shaojie Shen

The Edge Information

- Observations are affected by noise
- Information matrix $\mathbf{\Omega}_{ij}$ for each edgeto encode its uncertainty
- The "bigger" $\mathbf{\Omega}_{ij}$, the more the edge "matters"in the optimization

Questions

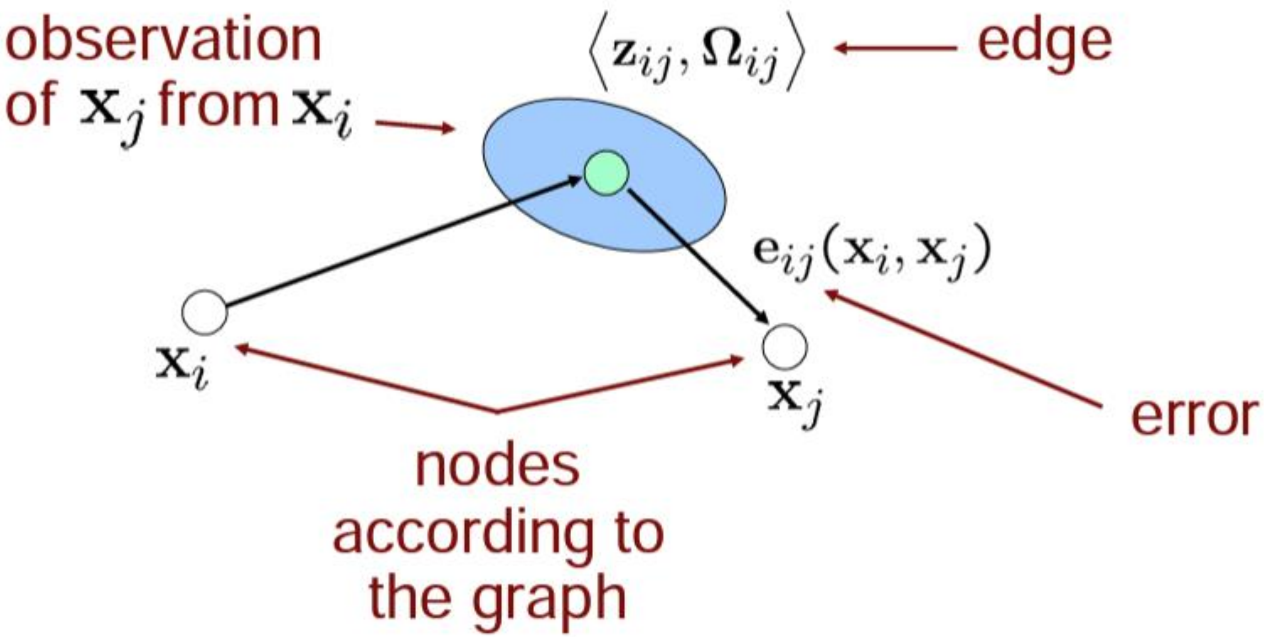
- How do the information matrices look like incase of scan-matching vs. odometry?
- How will these matrices look like when moving in a long, featureless corridor?

Courtesy: Cyrill Stachniss

Pose Graph

- A Least Squares Problem

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



Courtesy: Cyrill Stachniss

Least Squares SLAM

- This error function looks suitable for least squares error minimization

$$\begin{aligned}\mathbf{x}^* &= \underset{\mathbf{x}}{\operatorname{argmin}} \sum_k \mathbf{e}_k^T(\mathbf{x}) \Omega_k \mathbf{e}_k(\mathbf{x}) \\ &= \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{ij} \mathbf{e}_{ij}^T(\mathbf{x}_i, \mathbf{x}_j)\, \Omega_{ij} \mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j)\end{aligned}$$

Questions

- What is the state vector?

$$\mathbf{x}^T = \begin{pmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T & \cdots & \mathbf{x}_n^T \end{pmatrix}$$

- How to specify the error function?

Courtesy: Cyrill Stachniss

The Error Function

- Error function for single constraint

$$\mathbf{e}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathrm{t2v}\left(\underbrace{\mathbf{Z}_{ij}^{-1}}_{\text{Measurement}}\underbrace{\left(\mathbf{X}_i^{-1}\mathbf{X}_j\right)}_{\text{Relative Transformation}}\right)$$

- t2v means transformation (3x3) to vector (3x1)
- Error as a function of the whole state vector

$$\mathbf{e}_{ij}(\mathbf{x}) = \mathrm{t2v}\left(\mathbf{Z}_{ij}^{-1}\left(\mathbf{X}_i^{-1}\mathbf{X}_j\right)\right)$$

- Error takes a value of zero if

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

Courtesy: Cyrill Stachniss

Recap Gauss-Newton

- Define the error function
- Linearize the error function
- Compute its derivative
- Set the derivative to zero
- Solve the linear system
- Iterate this procedure until convergence

Courtesy: Cyrill Stachniss

Linearizing the Function

- We can approximate the errorfunctions around an initial guess x via Taylor expansion

$$\mathbf{e}_{ij}(\mathbf{x} + \Delta\mathbf{x}) \simeq \mathbf{e}_{ij}(\mathbf{x}) + \mathbf{J}_{ij}\Delta\mathbf{x}$$
$$\text{with } \mathbf{J}_{ij} = \frac{\partial \mathbf{e}_{ij}(\mathbf{x})}{\partial \mathbf{x}}$$

- The one term $\mathbf{e}_{ij}(\mathbf{x})$ depends only on $\mathbf{x}_i$ and $\mathbf{x}_j$, not all state variables
- Is there any consequence on thestructure of the Jacobian?
- Yes,

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

Courtesy: Cyrill Stachniss

Jacobians and Sparity

- The one term $\mathbf{e}_{ij}(\mathbf{x})$ depends only on $\mathbf{x}_i$ and $\mathbf{x}_j$,

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

- The Jacobian will be 0 everywhere **but** in the columns of $\mathbf{x}_i$ and $\mathbf{x}_j$

$$\mathbf{J}_{ij} = \begin{pmatrix} \mathbf{0} \cdots \mathbf{0} & \underbrace{\frac{\partial \mathbf{e}(\mathbf{x}_i)}{\partial \mathbf{x}_i}}_{\mathbf{A}_{ij}} & \mathbf{0} \cdots \mathbf{0} & \underbrace{\frac{\partial \mathbf{e}(\mathbf{x}_j)}{\partial \mathbf{x}_j}}_{\mathbf{B}_{ij}} & \mathbf{0} \cdots \mathbf{0} \end{pmatrix}$$

Courtesy: Cyrill Stachniss

Consequences of the Sparsity

- We need to compute the coefficient vectors and the coefficient matrices

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

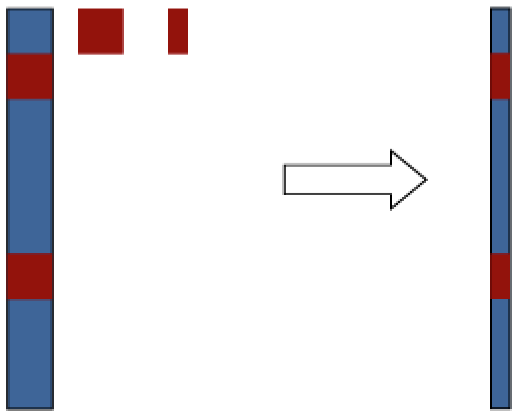
$$\mathbf{H} = \sum_{ij} \mathbf{H}_{ij} = \sum_{ij} \mathbf{J}_{ij}^T \boldsymbol{\Omega} \mathbf{J}_{ij}$$

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

Courtesy: Cyrill Stachniss

Illustration of the Structure

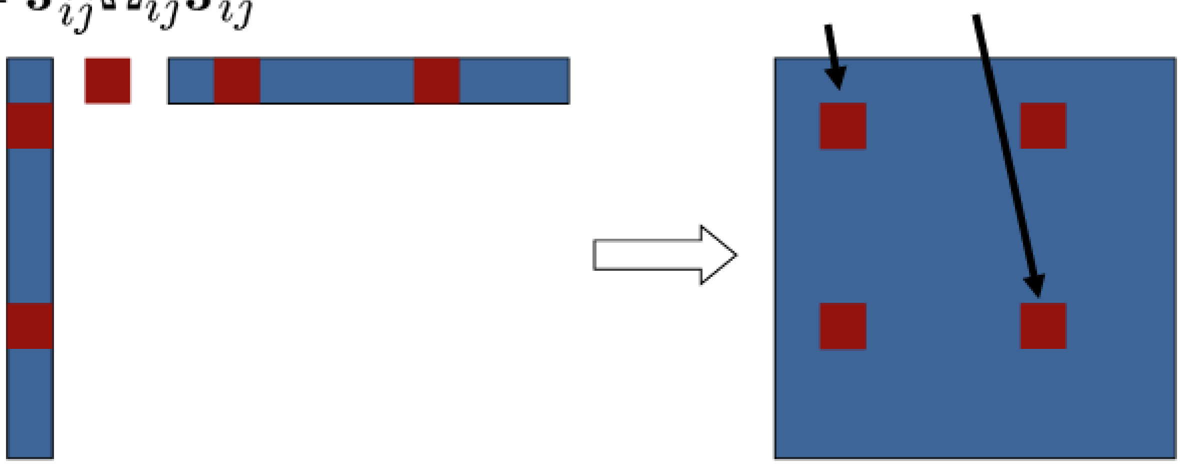
$$\mathbf{b}_{ij} = \mathbf{J}_{ij}^T \Omega_{ij} \mathbf{e}_{ij}$$



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

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

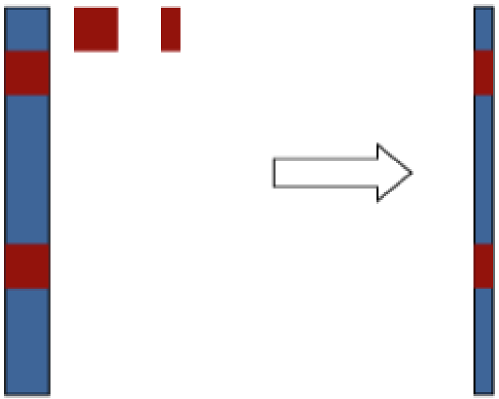
$$\mathbf{H}_{ij} = \mathbf{J}_{ij}^T \Omega_{ij} \mathbf{J}_{ij}$$



Courtesy: Cyrill Stachniss

Illustration of the Structure

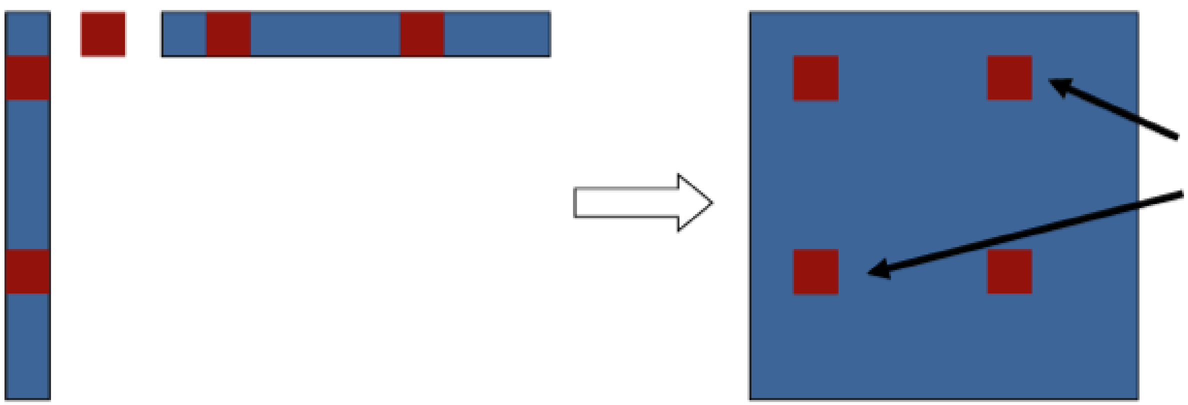
$$\mathbf{b}_{ij} = \mathbf{J}_{ij}^T \Omega_{ij} \mathbf{e}_{ij}$$



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

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

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

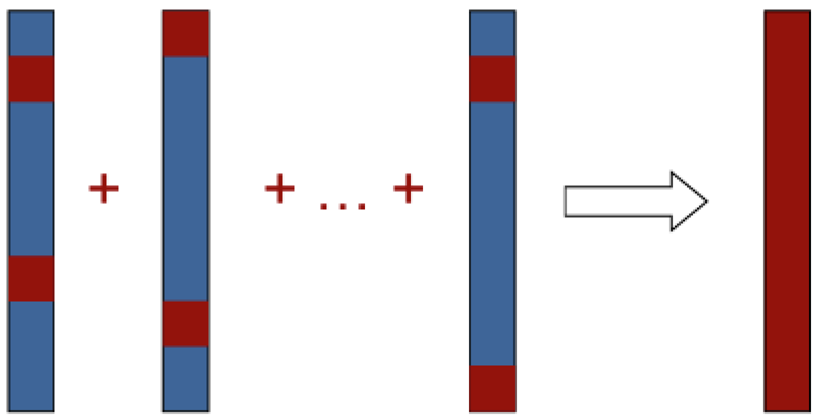


... and at the blocks ij,ji

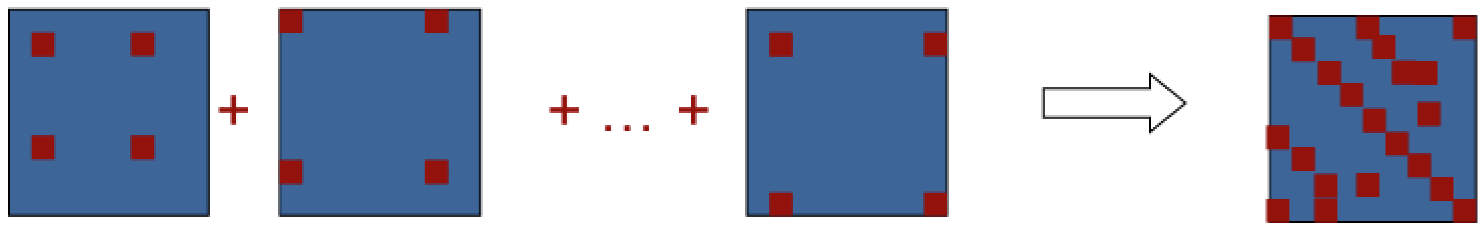
Courtesy: Cyrill Stachniss

Illustration of the Structure

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



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



Courtesy: Cyrill Stachniss

On the Linearized System

For **each** constraint:

- Compute error

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{t2v}\left(\mathbf{Z}_{ij}^{-1}\left(\mathbf{X}_i^{-1}\mathbf{X}_j\right)\right)$$

- Compute the blocks of the Jacobian: (the edge contributes to)

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

- Update the coefficient vector

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

- Update the system matrix

$$\begin{aligned} \bar{\mathbf{H}}^{ii} &\mathrel{+}= \mathbf{A}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{A}_{ij} & \bar{\mathbf{H}}^{ij} &\mathrel{+}= \mathbf{A}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{B}_{ij} \\ \bar{\mathbf{H}}^{ji} &\mathrel{+}= \mathbf{B}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{A}_{ij} & \bar{\mathbf{H}}^{jj} &\mathrel{+}= \mathbf{B}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{B}_{ij} \end{aligned}$$

Courtesy: Cyrill Stachniss

Pose Graph SLAM

optimize($\mathbf{x}$):

while ($!\mathit{converged}$)

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

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

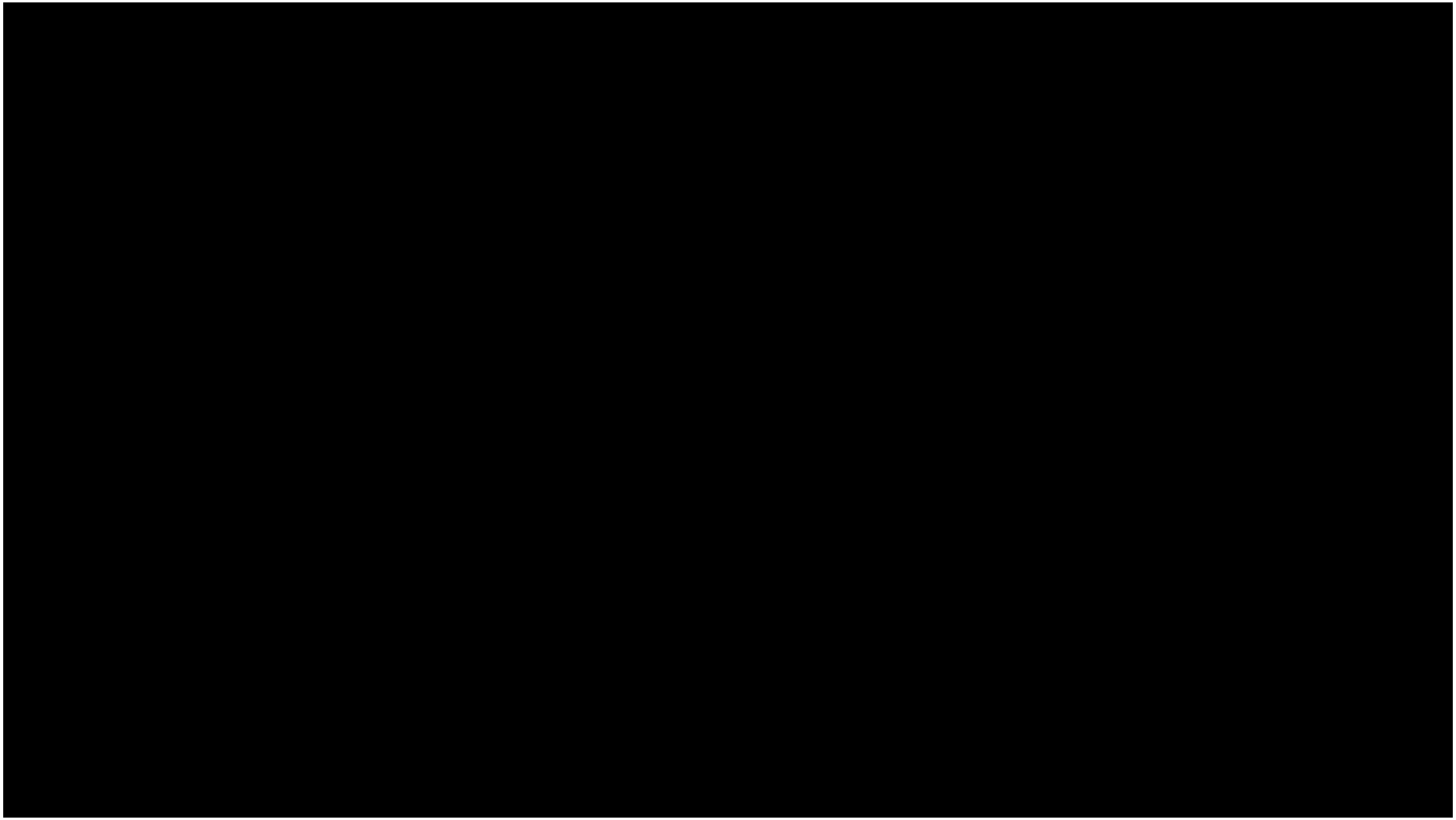
$\mathbf{x} = \mathbf{x} + \mathbf{\Delta x}$

end

return $\mathbf{x}$

Courtesy: Cyrill Stachniss

Application (1)



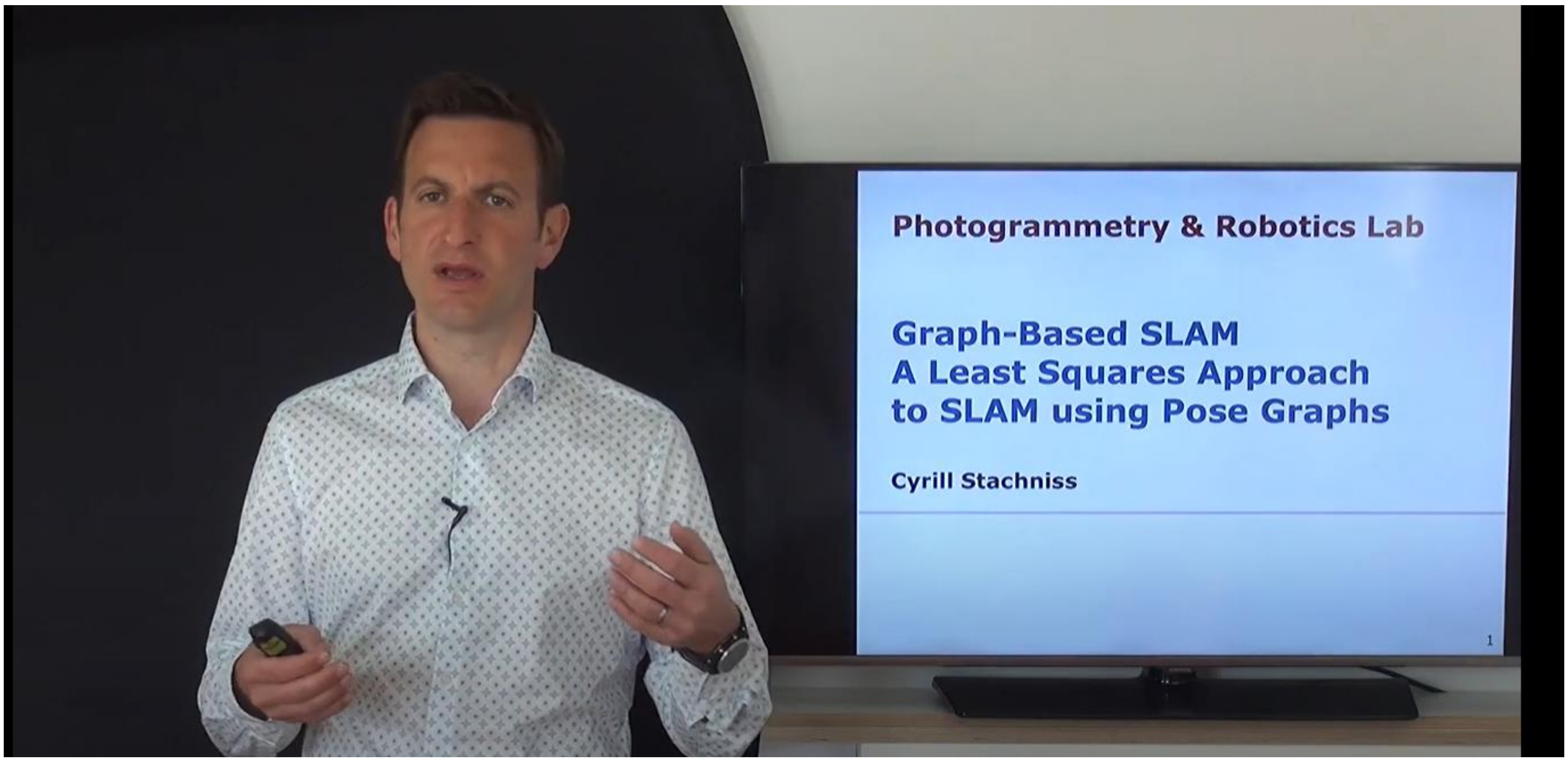
Courtesy: Smart Mobility Research Team@AIST

Application (2)

Courtesy: HKUST UAV Group

Resources

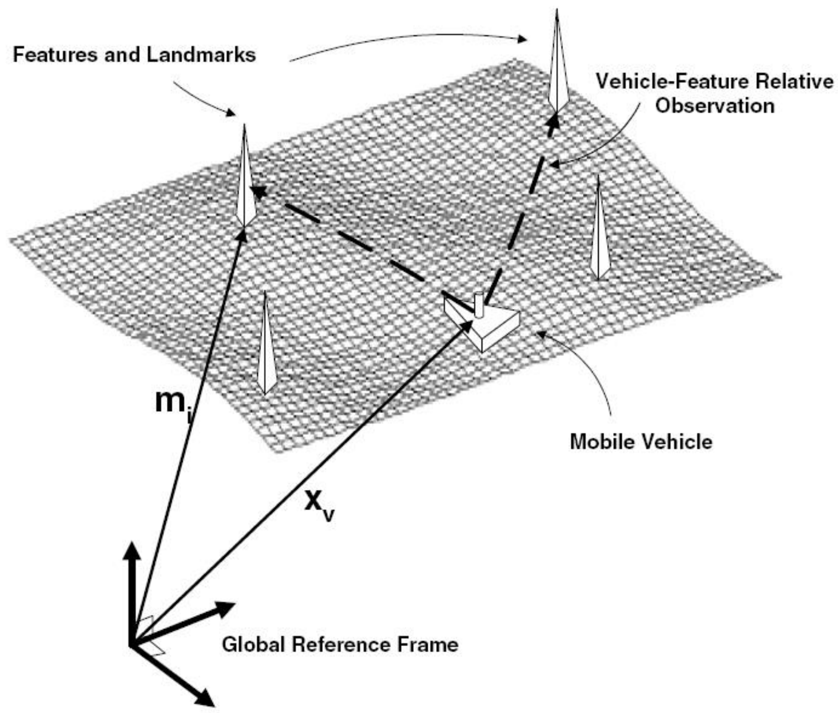
- Grisetti, G., Kümmerle, R., Stachniss, C. and Burgard, W., 2010. **A tutorial on graph-based SLAM**. IEEE Intelligent Transportation Systems Magazine, 2(4), pp.31-43.
- Prof. Cyrill Stachniss



Courtesy:

How about Landmark-based?

- In Lecture 10 EKF SLAM, we use landmarks as map representations.
- How to achieve pose graph slam with landmarks?



Next Lecture

- Pose Graph SLAM with Landmarks
- Quizz