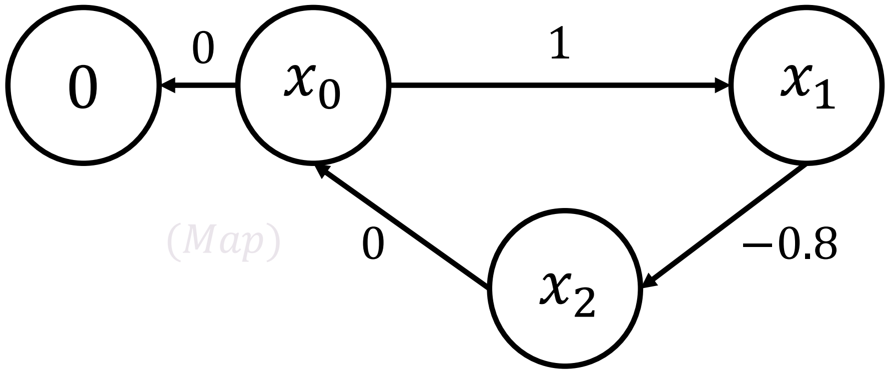
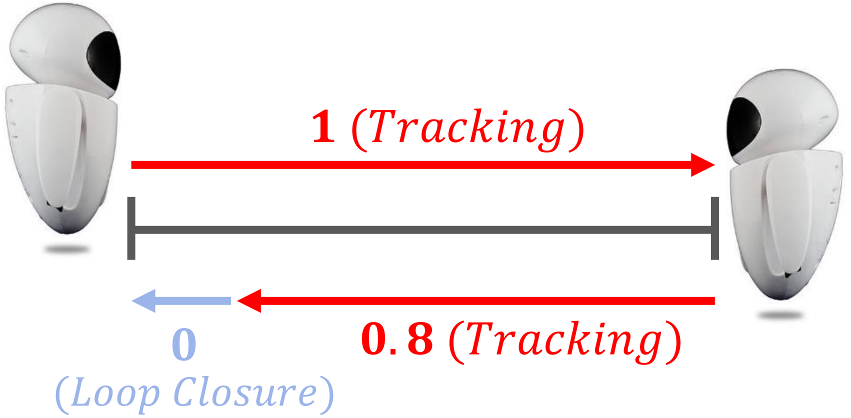
Robotic Navigation and Exploration

Week 6: Graph-based SLAM

Min-Chun Hu anitahu@cs.nthu.edu.tw
CS, NTHU

Graph Optimization: 1D Example





Error function

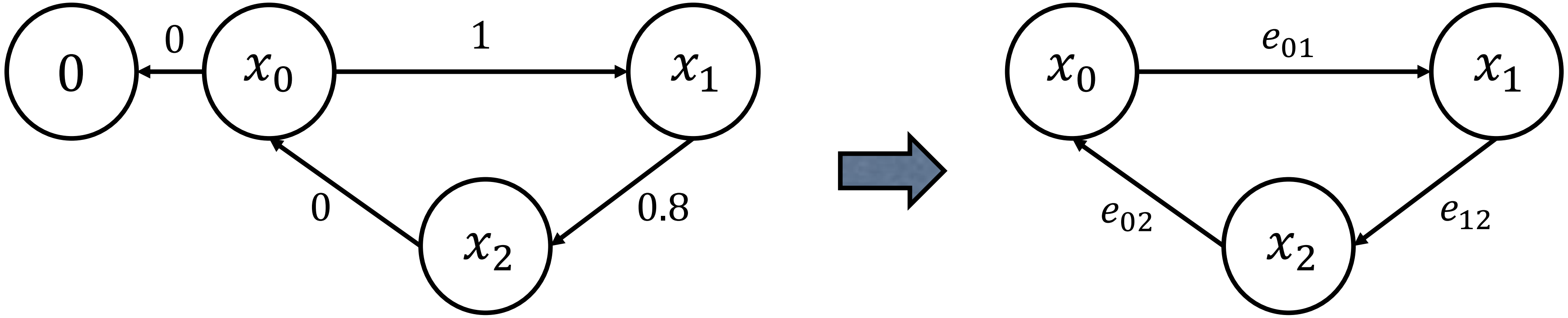
$$
\begin{aligned}
x_0 &= 0 \\
x_1 &= x_0 + 1 \\
x_2 &= x_1 - 0.8 \\
x_0 &= x_2 + 0
\end{aligned}
$$

$$
\begin{aligned}
f_1 &= x_0 \\
f_2 &= x_1 - x_0 - 1 \\
f_3 &= x_2 - x_1 + 0.8 \\
f_4 &= x_0 - x_2
\end{aligned}
$$

$$\min_x \sum_i w_i f_i^2 = w_1 x_0^2 + w_2(x_1 - x_0 - 1)^2 + w_3(x_2 - x_1 + 0.8)^2 + w_4(x_0 - x_2)^2$$

(Optimization)

Graph Optimization: 1D Example



Error Function

$$e_{01} = x_1 - x_0 - 1$$
$$e_{12} = x_2 - x_1 - 0.8$$
$$e_{02} = x_0 - x_2$$

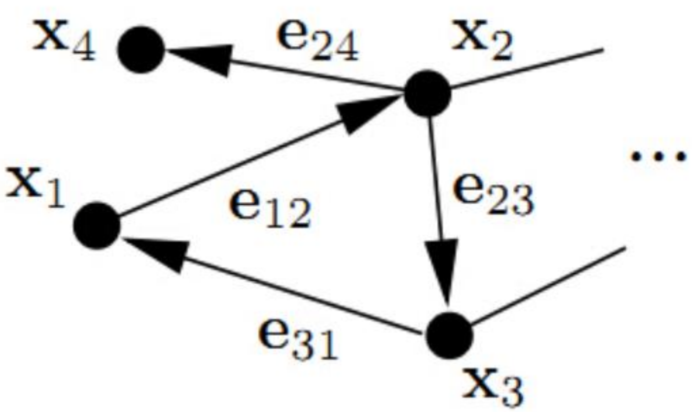
$$\min_x \sum_{i,j} w_{ij} e_{ij}^2 = w_{01}(x_1 - x_0 - 1)^2 + w_{12}(x_2 - x_1 + 0.8)^2 + w_{02}(x_0 - x_2)^2$$

Graph Optimization: General Form

$$\min_x \sum_{i,j} w_{ij} e_{ij}^2 = w_{01}(x_1 - x_0 - 1)^2 + w_{12}(x_2 - x_1 + 0.8)^2 + w_{02}(x_0 - x_2)^2$$

$$\mathbf{F}(\mathbf{x}) = \sum_{\langle i,j \rangle \in \mathcal{C}} \underbrace{\mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij})^\top \boldsymbol{\Omega}_{ij} \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij})}_{\mathbf{F}_{ij}} \quad (1)$$

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}} \mathbf{F}(\mathbf{x}). \quad (2)$$



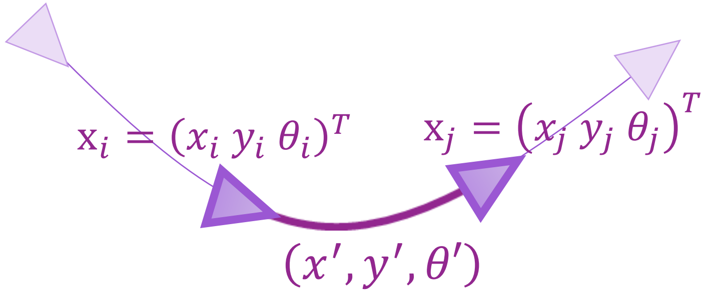
$$\begin{aligned} \mathbf{F}(\mathbf{x}) = \mathbf{e}_{12}^\top \, \boldsymbol{\Omega}_{12} \, \mathbf{e}_{12} \\ + \mathbf{e}_{23}^\top \, \boldsymbol{\Omega}_{23} \, \mathbf{e}_{23} \\ + \mathbf{e}_{31}^\top \, \boldsymbol{\Omega}_{31} \, \mathbf{e}_{31} \\ + \mathbf{e}_{24}^\top \, \boldsymbol{\Omega}_{24}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

Graph Optimization for 2D Pose

- Consider the relation between two poses:

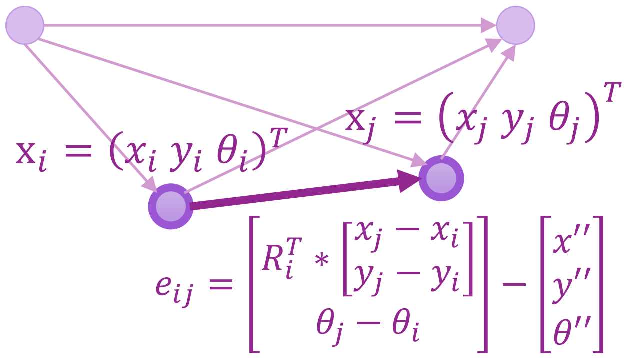
$$\begin{bmatrix} x_j \\ y_j \\ \theta_j \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} + \begin{bmatrix} R_i * \begin{bmatrix} x' \\ y' \end{bmatrix} \\ \theta' \end{bmatrix}$$, in which $R_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}$

And get $$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} R_i^T * \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \\ \theta_j - \theta_i \end{bmatrix}$$



- After measuring the transform (x'', y'', θ'') between two nodes, we can write down the error term:

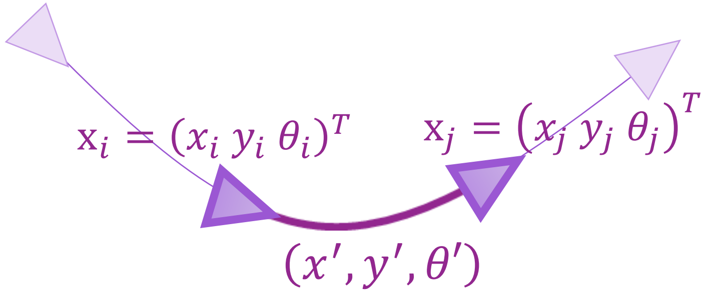
$$e_{ij} = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} - \begin{bmatrix} x'' \\ y'' \\ \theta'' \end{bmatrix} = \begin{bmatrix} R_i^T * \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \\ \theta_j - \theta_i \end{bmatrix} - \begin{bmatrix} x'' \\ y'' \\ \theta'' \end{bmatrix}$$



Graph Optimization for 2D Pose

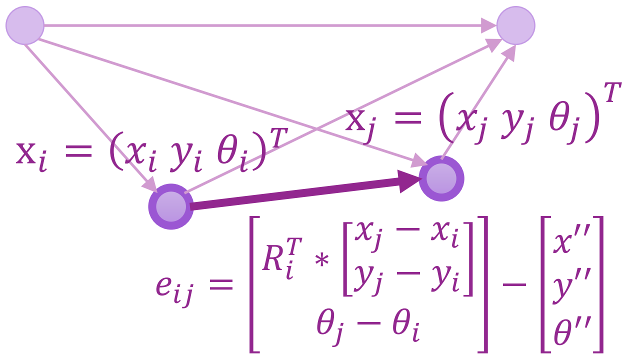
- The goal is to find the optimal poses

$$F = \sum_{i,j} e_{ij}^{\mathrm{T}} \Omega e_{ij} \qquad \mathrm{x} = (x, y, \theta)^{\mathrm{T}}$$
$$\mathrm{x}^* = \underset{\mathrm{x}}{\operatorname{argmax}} F(\mathrm{x})$$



- Approximate the object function by 1st order Taylor:

$$F \approx \sum_{i,j} e_{ij}(\mathrm{x}_i + \Delta\mathrm{x}_i, \mathrm{x}_j + \Delta\mathrm{x}_j)^T \Omega e_{ij}(\mathrm{x}_i + \Delta\mathrm{x}_i, \mathrm{x}_j + \Delta\mathrm{x}_j)$$
$$= \sum_{i,j} \left(e_{ij}(x_i, x_j) + A_{ij}\Delta\mathrm{x}_i + B_{ij}\Delta\mathrm{x}_j\right)^T \Omega \left(e_{ij}(x_i, x_j) + A_{ij}\Delta\mathrm{x}_i + B_{ij}\Delta\mathrm{x}_j\right) = \bar{\mathrm{F}}$$



, in which

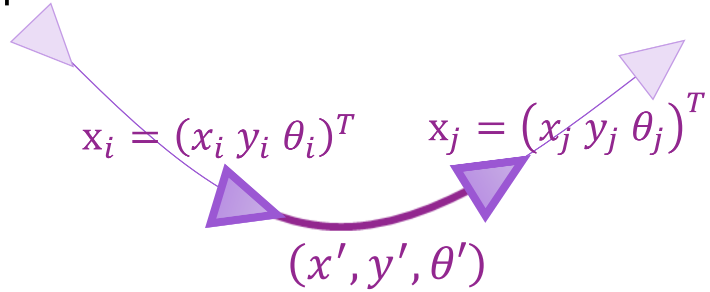
$$A_{ij} = \frac{\partial e_{ij}}{\partial \mathrm{x}_i} = \begin{bmatrix} -R_i^T & \frac{\partial R_i^T}{\partial \theta_i} \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \\ 0 & -1 \end{bmatrix}_{3\times 3}, B_{ij} = \frac{\partial e_{ij}}{\partial \mathrm{x}_j} = \begin{bmatrix} R_i^T & 0 \\ 0 & -1 \end{bmatrix}_{3\times 3}$$

Graph Optimization for 2D Pose

- Apply Gauss-Newton method, we solve the 1st order approximation of object function:

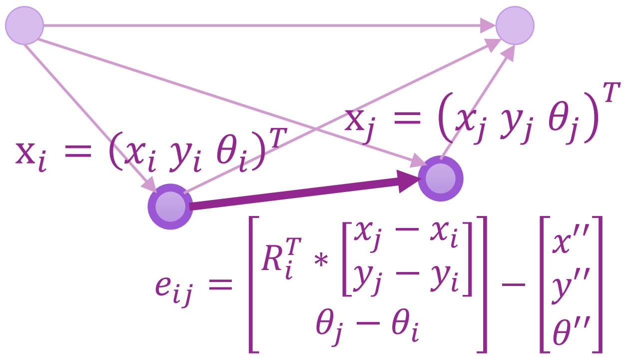
$$\frac{\partial \bar{\mathrm{F}}}{\partial \Delta \mathrm{x_i}} = A_{ij}^T \Omega A_{ij} \Delta x_i + A_{ij}^T \Omega B_{ij} \Delta x_j + A_{ij}^T \Omega e_{ij} = 0,$$

$$\frac{\partial \bar{\mathrm{F}}}{\partial \Delta \mathrm{x_j}} = B_{ij}^T \Omega A_{ij} \Delta x_i + B_{ij}^T \Omega B_{ij} \Delta x_j + B_{ij}^T \Omega e_{ij} = 0$$



- Transform the equation into matrix form:

$$\begin{bmatrix} A_{ij}^T \Omega A_{ij} & A_{ij}^T \Omega B_{ij} \\ B_{ij}^T \Omega A_{ij} & B_{ij}^T \Omega B_{ij} \end{bmatrix} * \begin{bmatrix} \Delta \mathrm{x_i} \\ \Delta \mathrm{x_j} \end{bmatrix} = \begin{bmatrix} -A_{ij}^T \Omega e_{ij} \\ -B_{ij}^T \Omega e_{ij} \end{bmatrix}$$



Solve the linear system by Cholesky Factorization

$$H \Delta \mathrm{x} = -b \qquad\qquad (H + \lambda I)\Delta \mathrm{x} = -b$$

$\mathbf{H} \approx \mathbf{J}^{\mathbf{T}}\mathbf{J}$ (Gauss-Newton) (Levenberg-Marquardt)

Complete Algorithm

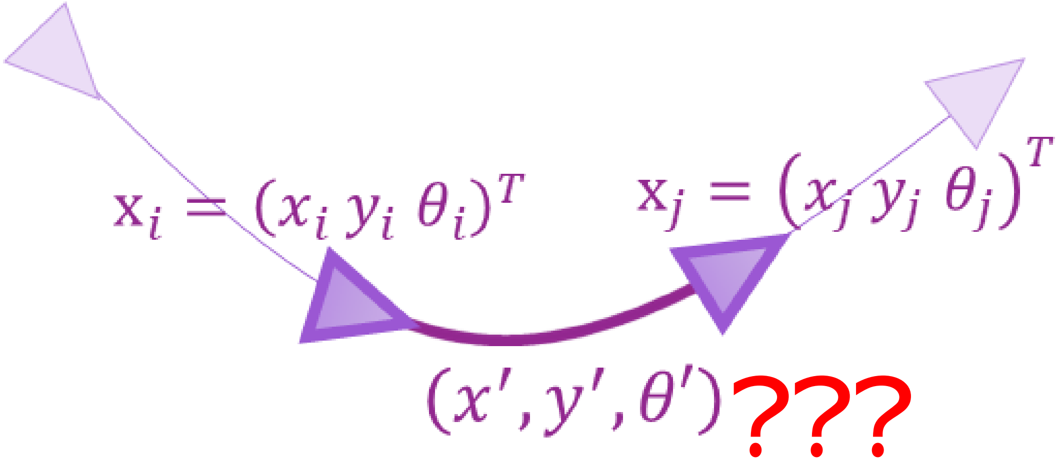
$$\mathbf{J}_{ij} = \left(\mathbf{0} \cdots \mathbf{0} \;\; \underbrace{\mathbf{A}_{ij}}_{\text{node } i} \;\; \mathbf{0} \cdots \mathbf{0} \;\; \underbrace{\mathbf{B}_{ij}}_{\text{node } j} \;\; \mathbf{0} \cdots \mathbf{0} \right).$$

$$\mathbf{H}_{ij} = \begin{pmatrix} \ddots & & & & \\ & \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{A}_{ij} & \cdots & \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{B}_{ij} & \\ & \vdots & \ddots & \vdots & \\ & \mathbf{B}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{A}_{ij} & \cdots & \mathbf{B}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{B}_{ij} & \\ & & & & \ddots \end{pmatrix}$$

$$\mathbf{b}_{ij} = \begin{pmatrix} \vdots \\ \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij} \\ \vdots \\ \mathbf{B}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij} \\ \vdots \end{pmatrix}$$

Require: $\breve{\mathbf{x}} = \breve{\mathbf{x}}_{1:T}$: initial guess. $\mathcal{C} = \{\langle \mathbf{e}_{ij}(\cdot), \boldsymbol{\Omega}_{ij} \rangle\}$: constraints
Ensure: $\mathbf{x}^*$: new solution, $\mathbf{H}^*$ new information matrix
// find the maximum likelihood solution
while $\neg$converged **do**
  $\mathbf{b} \leftarrow \mathbf{0}$ $\mathbf{H} \leftarrow \mathbf{0}$
  for all $\langle \mathbf{e}_{ij}, \boldsymbol{\Omega}_{ij} \rangle \in \mathcal{C}$ **do**
    // Compute the Jacobians $\mathbf{A}_{ij}$ and $\mathbf{B}_{ij}$ of the error function
    $\mathbf{A}_{ij} \leftarrow \left.\frac{\partial \mathbf{e}_{ij}(\mathbf{x})}{\partial \mathbf{x}_i}\right|_{\mathbf{x}=\breve{\mathbf{x}}}$ $\mathbf{B}_{ij} \leftarrow \left.\frac{\partial \mathbf{e}_{ij}(\mathbf{x})}{\partial \mathbf{x}_j}\right|_{\mathbf{x}=\breve{\mathbf{x}}}$
    // compute the contribution of this constraint to the linear system
    $\mathbf{H}_{[ii]} \mathrel{+}= \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{A}_{ij}$ $\mathbf{H}_{[ij]} \mathrel{+}= \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{B}_{ij}$
    $\mathbf{H}_{[ji]} \mathrel{+}= \mathbf{B}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{A}_{ij}$ $\mathbf{H}_{[jj]} \mathrel{+}= \mathbf{B}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{B}_{ij}$
    // compute the coefficient vector
    $\mathbf{b}_{[i]} \mathrel{+}= \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}$ $\mathbf{b}_{[j]} \mathrel{+}= \mathbf{B}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}$
  end for
  // keep the first node fixed
  $\mathbf{H}_{[11]} \mathrel{+}= \mathbf{I}$
  // solve the linear system using sparse Cholesky factorization
  $\boldsymbol{\Delta}\mathbf{x} \leftarrow \text{solve}(\mathbf{H}\,\boldsymbol{\Delta}\mathbf{x} = -\mathbf{b})$
  // update the parameters
  $\breve{\mathbf{x}} \mathrel{+}= \boldsymbol{\Delta}\mathbf{x}$
end while
$\mathbf{x}^* \leftarrow \breve{\mathbf{x}}$
$\mathbf{H}^* \leftarrow \mathbf{H}$
// release the first node
$\mathbf{H}^*_{[11]} \mathrel{-}= \mathbf{I}$
return $\langle \mathbf{x}^*, \mathbf{H}^* \rangle$

How to get the transformation ?



Scan-to-Scan Registration

- Given two matching points sets p_i and q_i, we aims to minimize the least square of registration error:

$$J = \frac{1}{2}\sum_{i=1}^{n}\|q_i - Rp_i - t\|^2$$

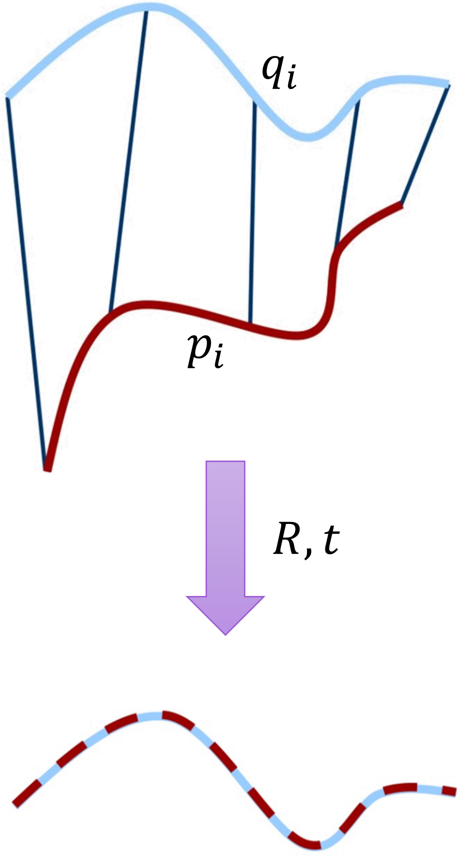
- Define the mean of points sets μ_p and μ_q, we can get

$$\frac{1}{2}\sum_{i=1}^{n}\|q_i - Rp_i - t\|^2 = \frac{1}{2}\sum_{i=1}^{n}\left\|q_i - Rp_i - t - (\mu_q - R\mu_p) + (\mu_q - R\mu_p)\right\|^2$$

$$= \frac{1}{2}\sum_{i=1}^{n}\left\|\left(q_i - \mu_q - R(p_i - \mu_p)\right) + (\mu_q - R\mu_p - t)\right\|^2$$

$$= \frac{1}{2}\sum_{i=1}^{n}\left\|\left(q_i - \mu_q - R(p_i - \mu_p)\right)\right\|^2 + \|\mu_q - R\mu_p - t\|^2 \cancel{+ 2\left(q_i - \mu_q - R(p_i - \mu_p)\right)^T(\mu_q - R\mu_p - t)}$$

$$\boxed{\begin{aligned}&\sum_{i=1}^{n}\left(q_i - \mu_q - R(p_i - \mu_p)\right)^T(\mu_q - R\mu_p - t) = (\mu_q - R\mu_p - t)^T\sum_{i=1}^{n}\left(q_i - \mu_q - R(p_i - \mu_p)\right)\\ &= (\mu_q - R\mu_p - t)^T\left(n\mu_q - n\mu_q - R(n\mu_p - n\mu_p)\right) = 0\end{aligned}}$$

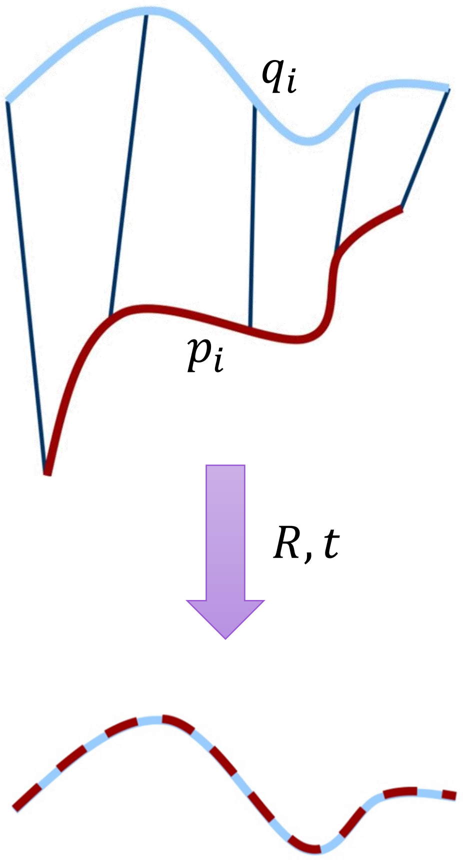


Scan-to-Scan Registration

- Define the relative location p_i' and q_i', the objective function becomes:

$$\frac{1}{2}\sum_{i=1}^{n}\left\|\left(q_i-\mu_q-R\left(p_i-\mu_p\right)\right)\right\|^2+\left\|\mu_q-R\mu_p-t\right\|^2$$
$$=\frac{1}{2}\sum_{i=1}^{n}\|(q_i{}'-Rp_i{}')\|^2+\left\|\mu_q-R\mu_p-t\right\|^2$$

$$\boxed{\begin{aligned}p_i' &= p_i-\mu_p,\\ q_i' &= q_i-\mu_q\end{aligned}}$$

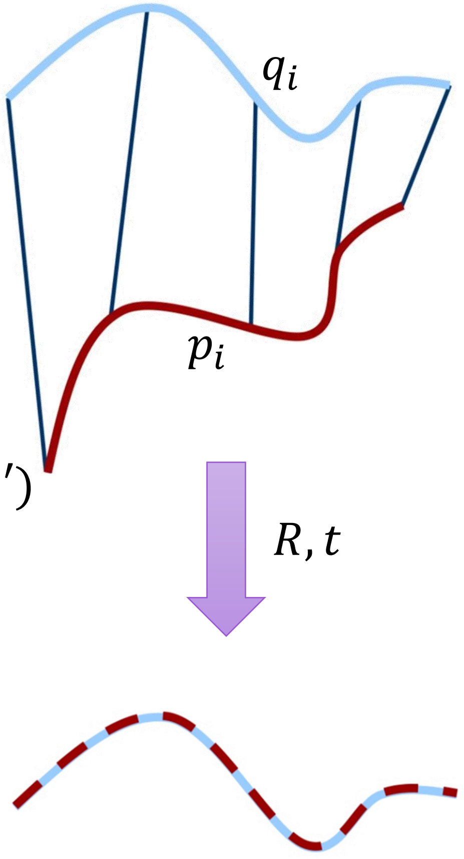


- Divide the optimization process into two steps:

1. Rotation $R^* = \underset{\mathrm{R}}{\operatorname{argmin}}\frac{1}{2}\sum_{i=1}^{n}\|(q_i{}'-Rp_i{}')\|^2$

2. Translation $t^* = \mu_q - R^*\mu_p$

Scan-to-Scan Registration



- Solve the rotation term:

$$R^* = \underset{\mathrm{R}}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} \|(q_i{}' - Rp_i{}')\|^2 = \underset{\mathrm{R}}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} (q_i'^T q_i' + p_i'^T R^T R p_i' - 2 q_i'^T R p_i{}')$$

$$= \underset{\mathrm{R}}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} (q_i'^T q_i' + p_i'^T p_i' - 2 q_i'^T R p_i{}') = \underset{\mathrm{R}}{\operatorname{argmin}} \sum_{i=1}^{n} -q_i'^T R p_i{}'$$

- Minimizing the function is equivalent to maximizing

$$F = \sum_{i=1}^{n} q_i'^T R p_i{}' = Trace\left(\sum_{i=1}^{n} R q_i'^T p_i{}'\right) = Trace(RH)$$

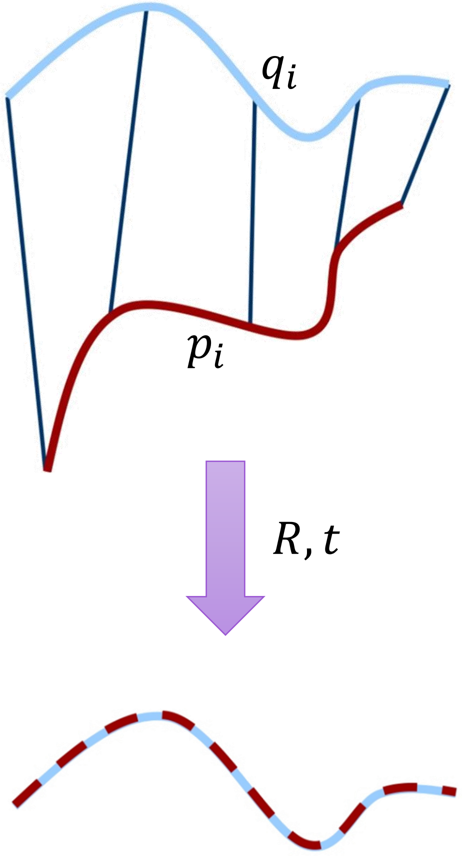
, where $\quad H = \sum_{i=1}^{n} q_i'^T p_i{}'$

Scan-to-Scan Registration

- we can solve the rotation by the SVD decomposition of H :

$$\underset{R}{\operatorname{argmax}}\, Trace(RH) \Rightarrow H = U\Lambda V^T \Rightarrow R^* = VU^T$$

- Proof:



Lemma:
For any positive definite matrix AA^T
, and any orthonormal matrix B,

$$Trace(AA^T) \geq Trace(BAA^T)$$

Proof of Lemma:
Let a_i be the $\boldsymbol{ith}$ column of A. Then

$$Trace(BAA^T) = Trace(A^TBA) = \sum_i a_i^T(Ba_i)$$

The Cauchy-Schwarz Inequality:

$$a_i^T(Ba_i) \leq \sqrt{(a_i^Ta_i)(a_i^TB^TBa_i)} = a_i^Ta_i$$

Hence, $Trace(BAA^T) \leq \sum_i a_i^Ta_i = Trace(AA^T)$

SVD decomposition of H :

$$H = U\Lambda V^T$$

Set $X = VU^T$, and we have

$$XH = VU^TU\Lambda V^T = V\Lambda V^T$$ (positive definite)

From the Lemma, for ant orthonormal matrix B

$$Trace(XH) \geq Trace(\boldsymbol{B}XH)$$

Any other rotation

Theorem C.1 **(Cauchy–Schwarz)** *Let V be a linear space with inner product $\langle ., . \rangle$, then for each $\mathbf{a}, \mathbf{b} \in V$ we have:*

$$|\langle \mathbf{a}, \mathbf{b} \rangle|^2 \leq ||\mathbf{a}|| \cdot ||\mathbf{b}||.$$

Proof If $\langle \mathbf{a}, \mathbf{b} \rangle = 0$ then the result is self evident. We therefore assume that $\langle \mathbf{a}, \mathbf{b} \rangle = \alpha \neq 0$, α may of course be complex. We start with the inequality

$$||\mathbf{a} - \lambda\alpha\mathbf{b}||^2 \geq 0$$

where λ is a real number. Now,

$$||\mathbf{a} - \lambda\alpha\mathbf{b}||^2 = \langle \mathbf{a} - \lambda\alpha\mathbf{b}, \mathbf{a} - \lambda\alpha\mathbf{b} \rangle.$$

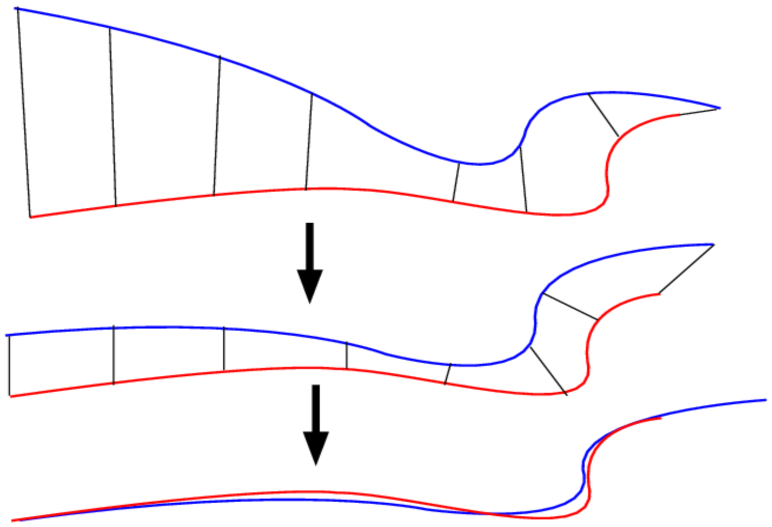
We use the properties of the inner product to expand the right hand side as follows:-

$$\begin{aligned}
\langle \mathbf{a} - \lambda\alpha\mathbf{b}, \mathbf{a} - \lambda\alpha\mathbf{b} \rangle = \langle \mathbf{a}, \mathbf{a} \rangle - \lambda\langle \alpha\mathbf{b}, \mathbf{a} \rangle - \lambda\langle \mathbf{a}, \alpha\mathbf{b} \rangle + \lambda^2|\alpha|^2\langle \mathbf{b}, \mathbf{b} \rangle \geq 0 \\
\text{so } ||\mathbf{a}||^2 - \lambda\alpha\langle \mathbf{b}, \mathbf{a} \rangle - \lambda\bar{\alpha}\langle \mathbf{a}, \mathbf{b} \rangle + \lambda^2|\alpha|^2||\mathbf{b}||^2 \geq 0 \\
\text{i.e. } ||\mathbf{a}||^2 - \lambda\alpha\bar{\alpha} - \lambda\bar{\alpha}\alpha + \lambda^2|\alpha|^2||\mathbf{b}||^2 \geq 0 \\
\text{so } ||\mathbf{a}||^2 - 2\lambda|\alpha|^2 + \lambda^2|\alpha|^2||\mathbf{b}||^2 \geq 0.
\end{aligned}$$

Scan-to-Scan Registration

- Iterative Closest Points (ICP) Algorithm

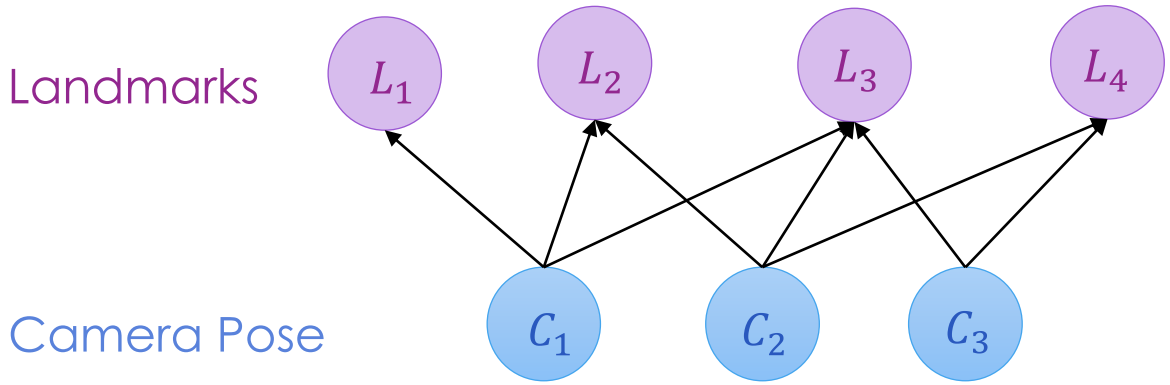
Given two points sets P and Q



Initialize $R_0 = I, t_0 = 0$
Build the kd-tree of Q
Repeat
  Transform the points set $\hat{p}_i = R_k p_i + t_k$
  Search the nearest points pairs $[q_i, \hat{p}_i]$
  Compute mean of points sets and the relative location $\hat{p}_i{}' = \hat{p}_i{}' - \mu_{\hat{p}}$ =and $q_i' = q_i - \mu_q$
  SVD Decomposition: $H = U\Lambda V^T$, where $H = \sum_{i=1}^{n} q_i'^T \hat{p}_i{}'$
  Get the optimize transformation $R^* = VU^T$ and $t^* = \mu_q - R^* \mu_p$
  Update the transformation $R_k = R^* R_{k-1}$ and $t_k = R^* t_{k-1} + t^*$
Until Convergence

Graph Optimization for Map and Pose

- Bundle Adjustment
- The bipartite optimization graph



- Given observation model $z_{ij} = h(C_i, L_j)$, the objective is to minimize the observation error:

$$F = \sum_{ij} \left\| z_{ij}^{obs} - h(C_i, L_j) \right\|^2$$

Sparse Hessian and Marginalization

- The Jacobian matrix of observation error and the approximated Hessian:

$$J_{ij} = \frac{\partial e_{ij}}{\partial \mathrm{x}} = [\underbrace{0, \dots, 0, \frac{\partial e_{ij}}{\partial C_i}, 0, \dots,}_{\text{Camera Pose}} \underbrace{0, 0, \dots, 0, \frac{\partial e_{ij}}{\partial L_j}, 0, \dots, 0}_{\text{Landmarks}}] \qquad H \cong J^T J = \begin{bmatrix} H_{ii} & H_{ij} \\ H_{ji} & H_{jj} \end{bmatrix}$$

(Arrow-Like Matrix)

- Schur Elimination and Marginalization

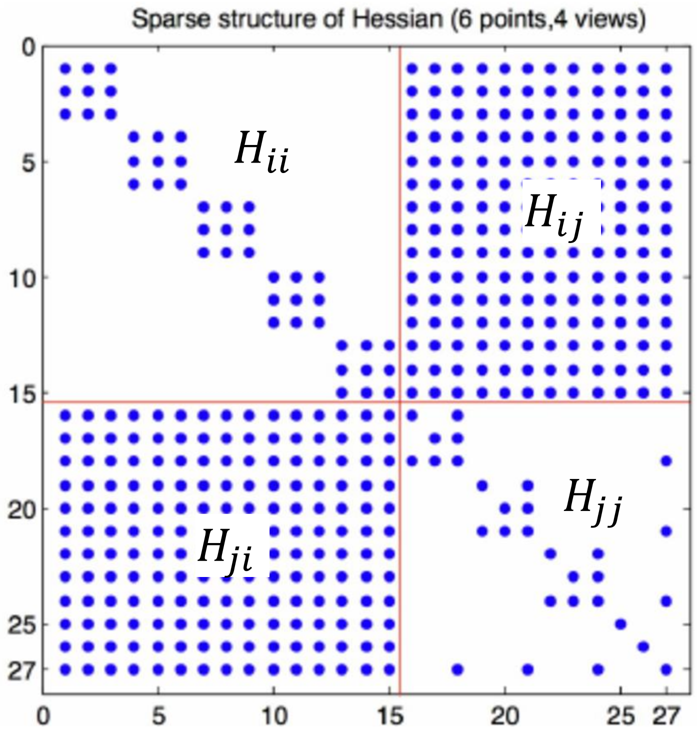
$$H\Delta\mathrm{x} = -b \rightarrow \begin{bmatrix} H_{ii} & H_{ij} \\ H_{ij}^T & H_{jj} \end{bmatrix} \begin{bmatrix} \Delta\mathrm{x}_C \\ \Delta\mathrm{x}_L \end{bmatrix} = \begin{bmatrix} v \\ w \end{bmatrix}$$

$$\begin{bmatrix} I & -H_{ij}H_{jj}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} H_{ii} & H_{ij} \\ H_{ij}^T & H_{jj} \end{bmatrix} \begin{bmatrix} \Delta\mathrm{x}_C \\ \Delta\mathrm{x}_L \end{bmatrix} = \begin{bmatrix} I & -H_{ij}H_{jj}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}$$

$$\begin{bmatrix} H_{ii} - H_{ij}H_{jj}^{-1}H_{ij}^T & 0 \\ H_{ij}^T & H_{jj} \end{bmatrix} \begin{bmatrix} \Delta\mathrm{x}_C \\ \Delta\mathrm{x}_L \end{bmatrix} = \begin{bmatrix} v - H_{ij}H_{jj}^{-1}w \\ w \end{bmatrix}$$

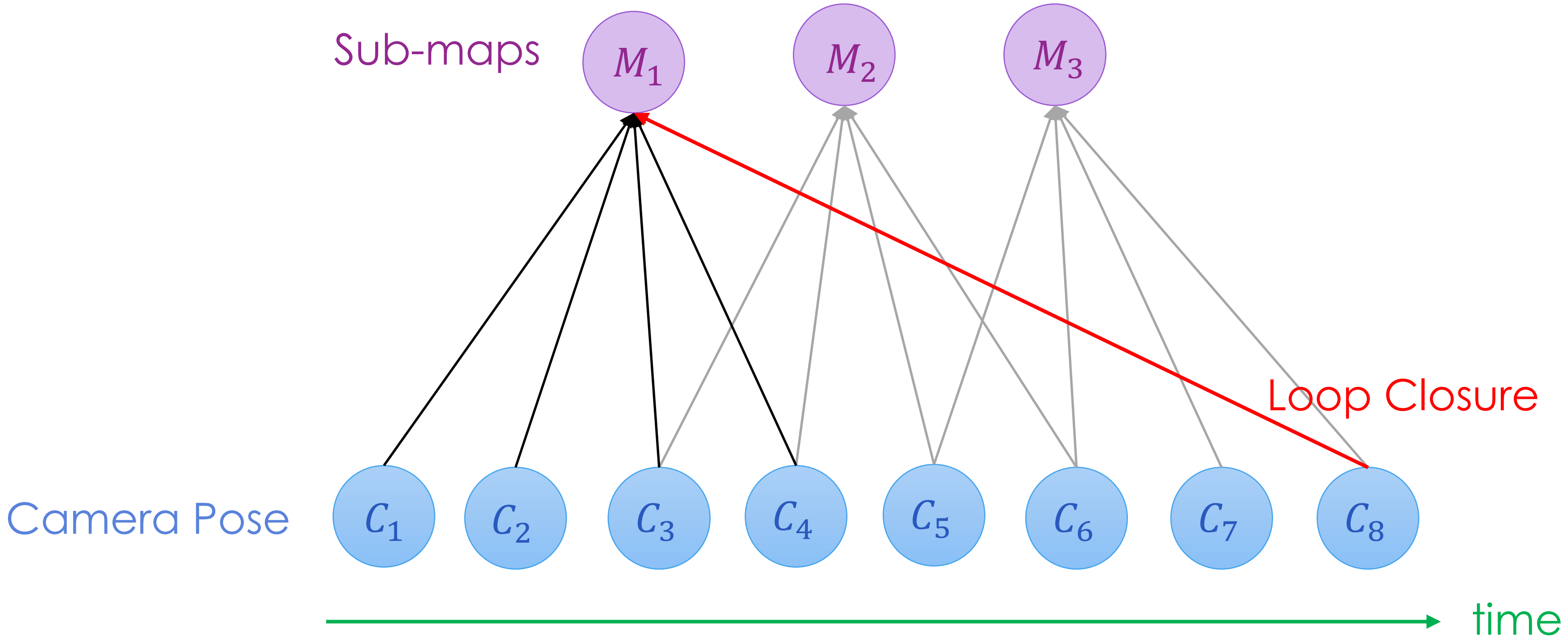
$$[H_{ii} - H_{ij}H_{jj}^{-1}H_{ij}^T]\Delta\mathrm{x}_C = v - H_{ij}H_{jj}^{-1}w$$

Easy to compute !!



Graph Optimization for Grid-based SLAM

- Karto-SLAM (Open-Source) / Cartographer (Google)



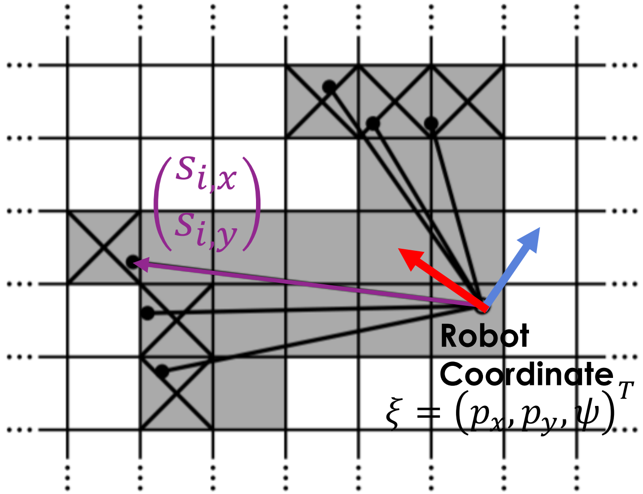
Scan-to-Map Matching

- Define the Robot Pose State $\xi = (p_x, p_y, \psi)^T$ and the Optimization Objective:

$$\xi^* = \operatorname{argmin}_\xi \sum_{i=1}^{n} [1 - M(S_i(\xi))]^2 \text{ , where } S_i(\xi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{pmatrix} \begin{pmatrix} s_{i,x} \\ s_{i,y} \end{pmatrix} + \begin{pmatrix} p_x \\ p_y \end{pmatrix}$$

- Apply the 1st order Taylor approximation

$$\sum_{i=1}^{n} [1 - M(S_i(\xi))]^2 \approx \sum_{i=1}^{n} \left[1 - M(S_i(\xi)) - \nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \Delta\xi\right]^2$$



- Partial Derivative to $\Delta\xi$

$$2\sum_{i=1}^{n} \left[\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi}\right]^T \left[1 - M(S_i(\xi)) - \nabla M(s_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \Delta\xi\right] = 0$$

Scan-to-Map Matching

- Solving the problem by GN methods:

$$2\sum_{i=1}^{n}\left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial \xi}\right]^T\left[1-M\big(S_i(\xi)\big)-\nabla M(s_i(\xi))\frac{\partial S_i(\xi)}{\partial \xi}\Delta\xi\right]=0$$

$$\underbrace{\left[\nabla M\big(S_i(\xi)\big)\frac{\partial S_i(\xi)}{\partial \xi}\right]^T\left[\nabla M\big(S_i(\xi)\big)\frac{\partial S_i(\xi)}{\partial \xi}\right]}_{H}\underbrace{\Delta\xi}_{\Delta \mathrm{x}}=\underbrace{\sum_{i=1}^{n}\left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial \xi}\right]^T[1-M(S_i(\xi))]}_{-b}$$

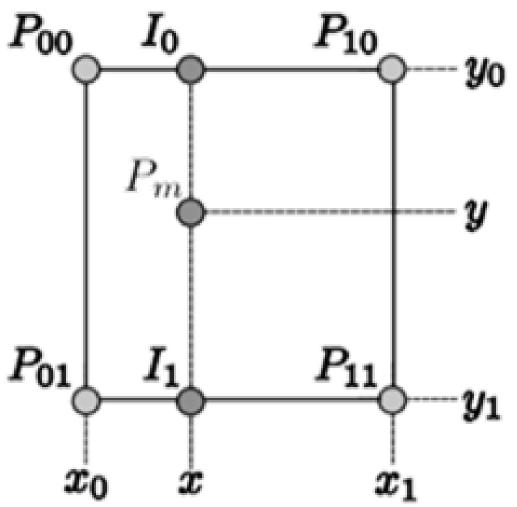
$$\Delta\xi = H^{-1}\sum_{i=1}^{n}\left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial \xi}\right]^T[1-M(S_i(\xi))]$$

$$\boxed{\frac{\partial S_i(\xi)}{\partial \xi}=\begin{pmatrix}1 & 0 & -\sin(\psi)\,s_{i,x}-cos(\psi)\,s_{i,y}\\ 0 & 1 & \cos(\psi)\,s_{i,x}-\sin(\psi)\,s_{i,y}\end{pmatrix}}$$

, where $H=\left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial \xi}\right]^T\left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial \xi}\right]$

Scan-to-Map Matching

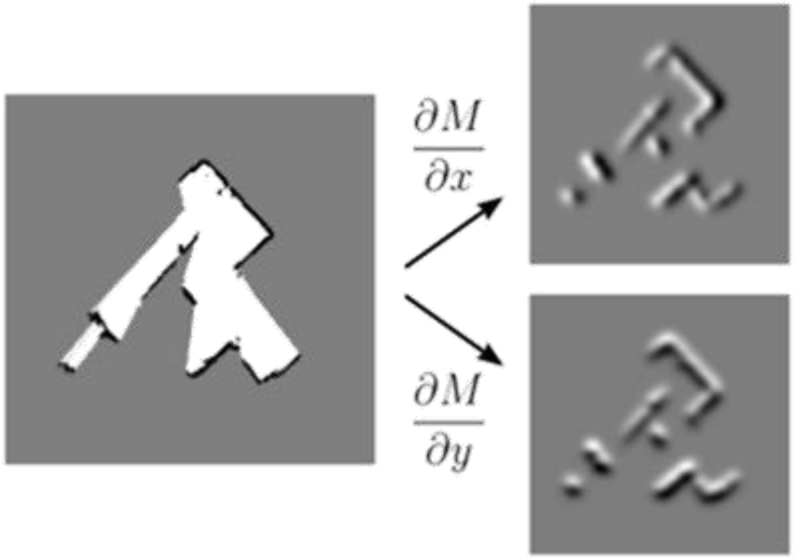
- The derivative of map with respect to location.



$$M(P_m) \approx \frac{y - y_0}{y_1 - y_0}\left(\frac{x - x_0}{x_1 - x_0}M(P_{11}) + \frac{x_1 - x}{x_1 - x_0}M(P_{01})\right) + \frac{y_1 - y}{y_1 - y_0}\left(\frac{x - x_0}{x_1 - x_0}M(P_{10}) + \frac{x_1 - x}{x_1 - x_0}M(P_{00})\right)$$

$$\frac{\partial M}{\partial x}(P_m) \approx \frac{y - y_0}{y_1 - y_0}(M(P_{11}) - M(P_{01})) + \frac{y_1 - y}{y_1 - y_0}(M(P_{10}) - M(P_{00}))$$

$$\frac{\partial M}{\partial y}(P_m) \approx \frac{x - x_0}{x_1 - x_0}(M(P_{11}) - M(P_{10})) + \frac{x_1 - x}{x_1 - x_0}(M(P_{01}) - M(P_{00}))$$



Cartographer Demo