Sliding Window for Lidar Localization.

1. There is a cost term for each residual term

$$e = \frac{1}{2} \| r + J\Delta x \|_{\Sigma}$$

$$= \frac{1}{2} (r + J\Delta x)^T \Sigma^{-1} (r + J\Delta x)$$

$$= \frac{1}{2} \Delta x^T J^T \Sigma^{-1} J \Delta x + r^T \Sigma^{-1} J \Delta x + \frac{1}{2} r^T \Sigma^{-1} r$$

$$\therefore \frac{\partial e}{\partial \Delta x} = 0 \Rightarrow J^T \Sigma^{-1} J \Delta x = -J^T \Sigma^{-1} r$$

$\therefore$ Define $\quad H_i = J_i^T \Sigma^{-1} J_i$

$$b_i = -J_i^T \Sigma^{-1} r_i$$

The optimization problem can be formatted as

$$\sum H_i \cdot \Delta x = \sum b_i$$

$\therefore$ for each type of constraint we must define

$$\begin{cases} \text{residual}, \ r \\ \text{Jacobian of residual w.r.t params}, \ J \end{cases}$$

for real-time localization, the analytic expression for r, J must be given to Ceres.

2. There are $\underline{\underline{3}}$ types of constraints

a. pose from map matching (scan-context)
or
GNSS position

b. relative pose from lidar frontend
(ICP / NDT / LOAM)

c. IMU pre-integration

∵ For constraint type $\underline{\underline{c}}$, the results from VIO can be readily used.

∴ Here the derivation will focus on type $\underline{\underline{a}}$ & $\underline{\underline{b}}$

3. a. pose from map matching / GNSS position

Residual:

$$r_p = t - t_{obs}$$

$$r_q = \ln(R \cdot R_{obs}^T)^\vee$$

Jacobian:

$$\frac{\partial r_p}{\partial t} = I_3$$

$$\frac{\partial r_q}{\partial R} = \lim_{\phi \to 0} \frac{\ln(R \exp \phi^\wedge R_{obs}^T)^\vee - \ln(R R_{obs}^T)^\vee}{\phi}$$

$$= \lim_{\phi \to 0} \frac{\ln(R R_{obs}^T R_{obs} \exp \phi^\wedge R_{obs}^T)^\vee - \ln(R R_{obs}^T)^\vee}{\phi}$$

$$= \lim_{\phi \to 0} \frac{\ln[R R_{obs}^T \cdot \exp(R_{obs}\phi)^\wedge]^\vee - \ln(R R_{obs}^T)^\vee}{\phi}$$

$$= \lim_{\phi \to 0} \frac{\ln(R R_{obs}^T)^\vee + J_r^{-1} R_{obs}\phi - \ln(R R_{obs}^T)^\vee}{\phi}$$

$$= J_r^{-1}[\ln(R R_{obs}^T)^\vee] R_{obs}$$

J_r should be implemented.

3.b. relative pose from lidar frontend.

Residual:

$$\because T_i^{-1} T_j = \begin{bmatrix} R_i^T, & -R_i^T t_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_j, & t_j \\ 0, & 1 \end{bmatrix}$$

$$= \begin{bmatrix} R_i^T R_j, & R_i^T (t_j - t_i) \\ 0, & 1 \end{bmatrix}$$

$$\therefore r_p = R_i^T (t_j - t_i) - t_{obs}$$

$$r_q = \ln (R_i^T R_j R_{obs}^T)^\vee$$

Jacobian, pos:

$$\frac{\partial r_p}{\partial t_i} = -R_i^T$$

$$\frac{\partial r_p}{\partial t_j} = R_i^T$$

Jacobian, ori:

$$\frac{\partial r_q}{\partial R_i} = \lim_{\phi \to 0} \frac{\ln[\exp(-\phi^\wedge) R_i^T R_j R_{obs}^T]^\vee - \ln[R_i^T R_j R_{obs}^T]^\vee}{\phi}$$

$$= \lim_{\phi \to 0} \frac{\ln[R_i^T R_j R_{obs}^T \cdot \exp(-R_{obs} R_j^T R_i \phi)^\wedge]^\vee - \ln(R_i^T R_j R_{obs}^T)^\vee}{\phi}$$

$$= -J_r^{-1}[\ln(R_i^T R_j R_{obs}^T)^\vee] R_{obs} R_j^T R_i$$

$$\frac{\partial r_q}{\partial R_j} = \lim_{\phi \to 0} \frac{\ln(R_i^T R_j \exp \phi^\wedge R_{obs}^T)^\vee - \ln(R_i^T R_j R_{obs}^T)^\vee}{\phi}$$

$$= \lim_{\phi \to 0} \frac{\ln[R_i^T R_j R_{obs}^T \cdot \exp(R_{obs}\phi)^\wedge]^\vee - \ln(R_i^T R_j R_{obs}^T)^\vee}{\phi}$$

$$= J_r^{-1}[\ln(R_i^T R_j R_{obs}^T)^\vee] R_{obs}$$

J_r should be implemented.

4. In order to adapt Marginalization to Ceres solver

a. When the sliding window has been filled, start to create Marginalization factor for next optimization

b. In order to fit into Ceres solver marginalization has to be implemented as follows:

$$\because \begin{cases} H_{rr} - H_{rm} H_{mm}^{-1} H_{mr} \doteq H = J^T J \\ b_r - H_{rm} H_{mm}^{-1} b_m \doteq b = -J^T r \end{cases}$$

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

$$\therefore J = \sqrt{\Lambda}\, V^T \quad \rightarrow \text{Marg. Res. Block. Jacobians}$$

$$\therefore r = J^{-T} b = \sqrt{\Lambda}^{-1} V^T b \rightarrow \text{Marg. Res. Block. Residuals.}$$

c. Add the Marg. Factor directly to next Ceres problem.