Sensor Fusion - 07 Sliding Window - YanGe FAD

1. Jacobians for tightly coupled LOAM

$\because$ for lio-mapping

$$\bar{X}^L_{(k+1,i)} = T_{wk}^{-1} T_{wk+1} X^L_{(k+1,i)}$$
$$= R_k^T (R_{k+1} X^L_{(k+1,i)} + t_{k+1} - t_k)$$

$$\therefore \frac{\partial \bar{X}^L_{(k+1,i)}}{\partial R_k} = \bar{X}^{L\wedge}_{(k+1,i)}$$

$$\therefore \frac{\partial \bar{X}^L_{(k+1,i)}}{\partial t_k} = -R_k^T$$

$$\therefore \frac{\partial \bar{X}^L_{(k+1,i)}}{\partial R_{k+1}} = R_k^T \frac{\partial R_{k+1} X^L_{(k+1,i)}}{\partial R_{k+1}}$$
$$= -R_k^T R_{k+1} X^{L\wedge}_{(k+1,i)}$$

$$\therefore \frac{\partial \bar{X}^L_{(k+1,i)}}{\partial t_{k+1}} = R_k^T$$

$\because$ for point-line association:

$$\frac{\partial d_{pl}}{\partial \bar{X}^{L}_{(k+1,i)}} = \alpha^{T}$$

$$\alpha = \frac{v_{il} \times v_{im} \times v_{lm}}{\| v_{il} \times v_{im} \times v_{lm} \|_2}$$

in which:
$$\begin{cases} v_{il} = \bar{X}^{L}_{(k+1,i)} - \bar{X}^{L}_{(k,l)} \\ v_{im} = \bar{X}^{L}_{(k+1,i)} - \bar{X}^{L}_{(k,m)} \\ v_{lm} = \bar{X}^{L}_{(k,m)} - \bar{X}^{L}_{(k,l)} \end{cases}$$

$\because$ for point-plane association:

$$\frac{\partial d_{pp}}{\partial \bar{X}^{L}_{(k+1,i)}} = \beta^{T}$$

$$\beta = \mathrm{sgn}(c) \cdot n$$

in which:
$$n = \frac{(\bar{X}^{L}_{(k,j)} - \bar{X}^{L}_{(k,l)}) \times (\bar{X}^{L}_{(k,j)} - \bar{X}^{L}_{(k,m)})}{\| (\bar{X}^{L}_{(k,j)} - \bar{X}^{L}_{(k,l)}) \times (\bar{X}^{L}_{(k,j)} - \bar{X}^{L}_{(k,m)}) \|_2}$$

$$c = (\bar{X}^{L}_{(k+1,i)} - \bar{X}^{L}_{(k,j)})^{T} n$$

$$\mathrm{sgn}(\sigma) = \begin{cases} 1 & \sigma \geq 0 \\ -1 & \sigma < 0 \end{cases}$$

$\because$ Jacobian for tightly-coupled LOAM are

a. point-line:

$$\text{pose } k\,(i)\begin{cases}\dfrac{\partial d_{pL}}{\partial R_k} = \alpha^T \bar{X}^{L}_{(k+1,i)}{}^{\wedge} \\[2ex] \dfrac{\partial d_{pl}}{\partial t_k} = -\alpha^T R_k^T\end{cases}$$

$$\text{pose } k{+}1\,(j)\begin{cases}\dfrac{\partial d_{pl}}{\partial R_{k+1}} = -\alpha^T R_k^T R_{k+1} X^{L}_{(k+1,i)}{}^{\wedge} \\[2ex] \dfrac{\partial d_{pl}}{\partial t_{k+1}} = \alpha^T R_k^T\end{cases}$$

b. point-plane

$$\text{pose } k\,(i)\begin{cases}\dfrac{\partial d_{pp}}{\partial R_k} = \beta^T \bar{X}^{L}_{(k+1,i)}{}^{\wedge} \\[2ex] \dfrac{\partial d_{pp}}{\partial t_k} = -\beta^T R_k^T\end{cases}$$

$$\text{pose } k{+}1\,(j)\begin{cases}\dfrac{\partial d_{pp}}{\partial R_{k+1}} = -\beta^T R_k^T R_{k+1} X^{L}_{(k+1,i)}{}^{\wedge} \\[2ex] \dfrac{\partial d_{pp}}{\partial t_{k+1}} = \beta^T R_k^T\end{cases}$$

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

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

$\therefore$ 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.

But for this project the extended pose will be parameterized using so3 rather than quaternion

3. a. pose from map matching / GNSS position

Residual:

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

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

Jacobian:

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

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

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

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

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

$$= J_r^{-1}(r_q) R_{obs}$$

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:

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

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

$$= -J_r^{-1}(r_q) \exp(r_q^\wedge).\text{inverse}()$$

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

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

$$= J_r^{-1}(r_q) R_{obs}$$

J_r should be implemented.

3.c IMU pre-integration:

The residual, when parameterized using $\underline{\underline{so3}}$, is as follows:

①
$$r_p = R_i^T(p_j - p_i - v_i T + \frac{1}{2} g T^2) - \alpha_{ij}$$

$\propto p_i, r_i, v_i, b_{ai}, b_{gi}$ / p_j, b_{aj}, b_{gj}

$\therefore \underline{\underline{i}}$ $\frac{\partial r_p}{\partial p_i} = -R_i^T$ $\therefore \underline{\underline{j}}$ $\frac{\partial r_p}{\partial p_j} = R_i^T$

$\frac{\partial r_p}{\partial r_i} = [R_i^T(p_j - p_i - v_i T + \frac{1}{2} g T^2)]^\wedge$ $\frac{\partial r_p}{\partial r_j} = 0$

$\frac{\partial r_p}{\partial v_i} = -T \cdot R_i^T$ $\frac{\partial r_p}{\partial v_j} = 0$

$\frac{\partial r_p}{\partial b_{ai}} = -J.\text{block}<3,3>(P, A)$ $\frac{\partial r_p}{\partial b_{aj}} = 0$

$\frac{\partial r_p}{\partial b_{gi}} = -J.\text{block}<3,3>(P, G)$ $\frac{\partial r_p}{\partial b_{gj}} = 0$

②

$$r_r = \ln(R_{ij}^T R_i^T R_j)^\vee$$

$\propto \quad r_i,\ b_{gi} \qquad\qquad r_j,\ b_{gj}$

$\therefore\ \underline{\underline{i}}$

$$\frac{\partial r_r}{\partial r_i} = \lim_{\varphi \to 0} \frac{\ln[R_{ij}^T R_i^T R_j \exp(-R_j^T R_i \varphi)^\wedge]^\vee - r_r}{\varphi} = -J_r^{-1}(r_r) R_j^T R_i$$

$$\frac{\partial r_r}{\partial b_{gi}} = \lim_{\delta b_{gi} \to 0} \frac{\ln[\exp(-J_{b_{gi}}^q \delta b_{gi})^\wedge R_{ij}^T R_i^T R_j)]^\vee - r_r}{\delta b_{gi}}$$

$$= -J_r^{-1}(r_r)\ \exp(r_r^\wedge).\mathrm{inverse}()\ J.\mathrm{block}<3,3>(R,G)$$

$\therefore\ \underline{\underline{j}}$

$$\frac{\partial r_r}{\partial r_j} = \lim_{\varphi \to 0} \frac{\ln[R_{ij}^T R_i^T R_j \exp\varphi^\wedge]^\vee - r_r}{\varphi} = J_r^{-1}(r_r)$$

$$\frac{\partial r_r}{\partial b_{gj}} = 0$$

③.

$$r_v = R_i^T(v_j - v_i + gT) - \beta_{ij}$$

$$\propto r_i, v_i, b_{ai}, b_{gi} \qquad v_j, b_{aj}, b_{gj}$$

$\therefore \underline{\underline{i}}$: $\dfrac{\partial r_v}{\partial r_i} = [R_i^T(v_j - v_i + gT)]^\wedge$

$$\frac{\partial r_v}{\partial v_i} = -R_i^T \qquad \frac{\partial r_v}{\partial v_j} = R_i^T$$

$$\frac{\partial r_v}{\partial b_{ai}} = -J.block<3,3>(V,A) \qquad \frac{\partial r_v}{\partial b_{aj}} = 0$$

$$\frac{\partial r_v}{\partial b_{gi}} = -J.block<3,3>(V,G) \qquad \frac{\partial r_v}{\partial b_{gj}} = 0$$

④: $r_a = b_{aj} - b_{ai} \qquad r_g = b_{gj} - b_{gi}$

$$\frac{\partial r_a}{\partial b_{ai}} = -I \quad \frac{\partial r_a}{\partial b_{aj}} = I \qquad \frac{\partial r_g}{\partial b_{gi}} = -I \quad \frac{\partial r_g}{\partial b_{gj}} = I$$

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:

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

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

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

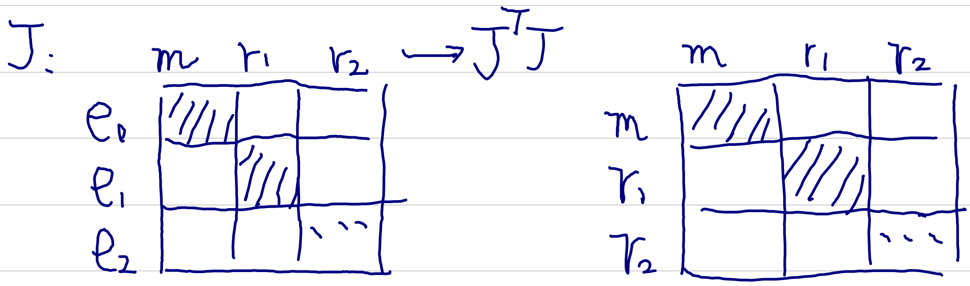
$$\therefore r = J^{-T} b = \sqrt{\Lambda}^{-1} V^T b \rightarrow \text{Marg. Res. Block. Residuals.}$$

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

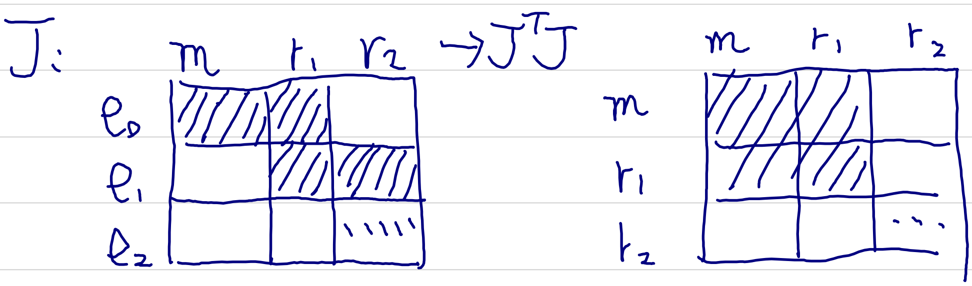
Marginalization Residual Block Building, PRVAG in $\underline{\underline{SO3}}$

for the to-be-marginalized param block m
and its next param block r_1

a. Map Matching:



b. Relative Pose & IMU Pre Integration



∴ The marginalization res. block is only relevant to param block $\underline{\underline{r_1}}$

and H_{rr}, H_{rm}, H_{mm}, H_{mr}, b_r, b_m

should be computed using res. block $\begin{cases} \text{MapMatching}(m) \\ \text{RelativePose}(m, r_1) \\ \text{IMU}(m, r_1) \end{cases}$

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

$$J^T J \Delta x + J^T r = 0$$

$$J^T J \Delta x = -J^T r$$