VIO from scratch 8-2, 8-3

Sunday, April 12, 2020 9:27 AM

IMU pre-integration:

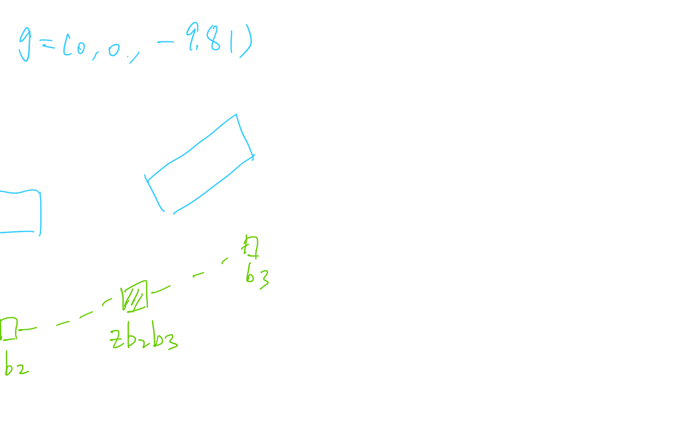
① $\tilde{\omega}^b = \omega^b + b^g + n^g$

$\tilde{a}^b = q_{bw}(a^w + g^w) + b^a + n^a$

② $\alpha_{b_ib_j} = \iint_{t\in[i,j]} (q_{b_ib_t} a^{b_t}) \delta t^2$

$\beta_{b_ib_j} = \int_{t\in[i,j]} (q_{b_ib_t} a^{b_t}) \delta t$

$q_{b_ib_j} = \int_{t\in[i,j]} q_{b_ib_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2}\omega^{b_t} \end{bmatrix} \delta t$



$q_{b_ib_j}$ means the relative rotation between b_i, b_j.

$\alpha_{b_ib_j}$ means the relative position between b_i, b_j.

$\beta_{b_ib_j}$ means the relative velocity between b_i, b_j.

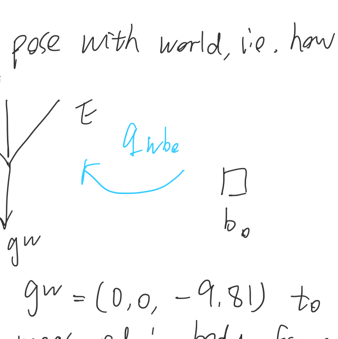
Pre-integration can constrain the relative motion between images. At every time instance, IMU has a noise. We have used preintegration to change the 1000Hz data into one value, we need to compute the covariance. ③

Three important topics:
1. IMU model
2. Pre-integration
3. Covariance propagation.

Multiview geometry:
1. Keypoints extraction
2. Use E, H matrix to recover pose. (up to scale)
3. Triangulation
4. With known 3D features, obtain new camera pose using PnP.

Issues:

① How to align IMU pose with world, i.e. how to compute q_{wb}?



q_{wb} can transform $g^w = (0, 0, -9.81)$ to body frame, i.e. we can transform the acc measured in body frame a_b to world frame a^w.

② How to align IMU and camera poses? How to obtain camera-IMU extrinsics? How to obtain initial velocity V_k, bias?

Link between visual and IMU:

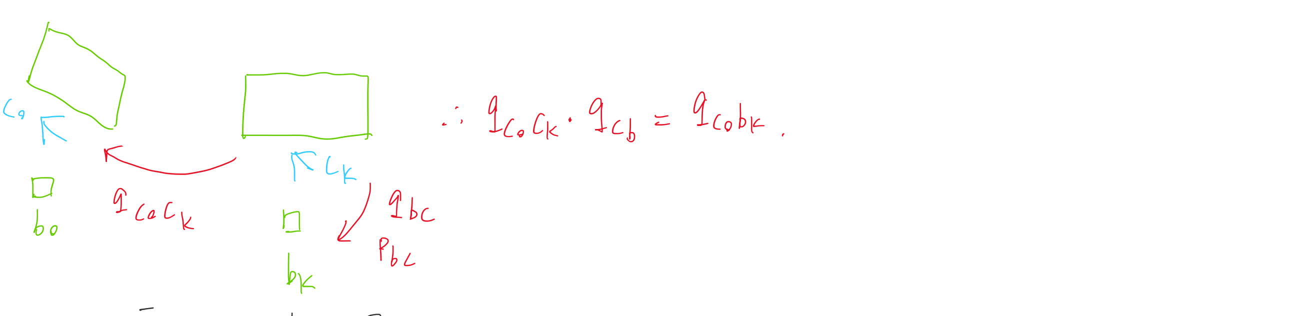
Consider camera frame c_0 as world frame, extrinsics are q_{bc}, t_{bc}.

$q_{c_0b_k} = q_{c_0c_k} \otimes q_{bc}^{-1}$

$s\bar{p}_{c_0b_k} = s\bar{p}_{c_0c_k} - R_{c_0b_k} p_{bc}$

s: scale factor, $\bar{p}$: up-to-scale translation, we get:

$\bar{p}_{c_0c_k} = \frac{1}{s} R_{c_0b_k} p_{bc} + \bar{p}_{c_0b_k}$



$\therefore q_{c_0b_k} \cdot q_{bc} = q_{c_0c_k}$

$\begin{bmatrix} R_{c_0b_k} & \bar{p}_{c_0b_k} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{s}p_{bc} \\ 1 \end{bmatrix} = \bar{p}_{c_0c_k}$

Steps:

q_{bc} unknown, we estimate q_{bc} first.

Then use the rotation constraint to estimate bias

$q_{c_0b_k} = q_{c_0c_k} \otimes q_{bc}^{-1}$

Use translation constraint to estimate gravity direction, velocity and scale.

$s\bar{p}_{c_0b_k} = s\bar{p}_{c_0c_k} - R_{c_0b_k} p_{bc}$

optimize gravity vector g_{c_0}.

Solve for q_{wc_0}.

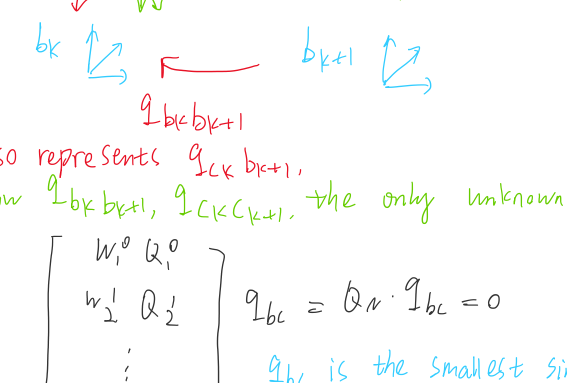
How to estimate q_{bc}:

between time k, k+1, we have IMU preintegration $q_{b_kb_{k+1}}$, visual measurements $q_{c_kc_{k+1}}$.

$q_{b_kb_{k+1}} \otimes q_{bc} = q_{bc} \otimes q_{c_kc_{k+1}}$

$\left([q_{b_kb_{k+1}}]_L - [q_{c_kc_{k+1}}]_R\right) q_{bc} = Q^k_{k+1} \cdot q_{bc} = 0$

$[\cdot]_L, [\cdot]_R$ are left, right quaternion multiplication.



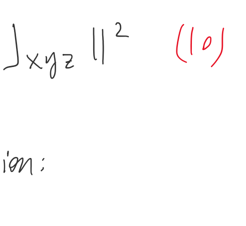
This also represents $q_{c_kb_{k+1}}$.

We know $q_{b_kb_{k+1}}, q_{c_kc_{k+1}}$, the only unknown is q_{bc}.

$$\begin{bmatrix} w_1^0 Q_1^0 \\ w_2^1 Q_2^1 \\ \vdots \\ w_N^{N-1} Q_N^{N-1} \end{bmatrix} q_{bc} = Q_N \cdot q_{bc} = 0$$

q_{bc} is the smallest singular value's vector.

where $w^k_{k+1} = \begin{cases} 1 & r^k_{k+1} < \text{threshold} \\ \frac{\text{threshold}}{r^k_{k+1}} & \text{otherwise} \end{cases}$ Huber cost



$\because tr(R) = 1 + 2\cos\theta, \therefore \theta = \arccos(tr(R) - 1)/2$

$r^k_{k+1} = a\cos\left(\left(tr\left(\hat{R}_{bc}^{-1} R_{b_kb_{k+1}}^{-1} \hat{R}_{bc} R_{c_kc_{k+1}}\right) - 1\right)/2\right)$

In VINS, this part is in initial_ex_rotation.cpp in CalibrationExRotation().

We also need to check whether the second last singular value is too small. If it's too small, we need to re-collect the data.

Geoscope bias

with known q_{bc}, to estimate bias: $q^{-1}_{c_0b_{k+1}} q_{c_0b_k} = q_{b_{k+1}b_k}$

$$\underset{\delta b^g}{\arg\min} \sum_{k\in B} \left\| 2 \left[q^{-1}_{c_0b_{k+1}} \otimes q_{c_0b_k} \otimes q_{b_kb_{k+1}} \right]_{xyz} \right\|^2 \quad (10)$$

($q^{-1}_{c_0b_{k+1}}$: known, $q_{c_0b_k}$: known)

where B represents all keyframes. Using taylor expansion:

$$q_{b_kb_{k+1}} \approx \hat{q}_{b_kb_{k+1}} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} J^q_{b^g} \delta b^g \end{bmatrix} \quad (11)$$

Code in: intial_aligment.cpp, solveGyroscopeBias().

$\begin{bmatrix} 1 \\ \frac{1}{2}\theta_{err} \end{bmatrix} = q^{-1}_{err} = \left(\hat{q}^c_{b_kb_{k+1}}\right)^{-1} \otimes q_{b_kb_{k+1}}$

$\therefore 2\left[q^{-1}_{c_0b_{k+1}} \otimes q_{c_0b_k} \otimes q_{b_kb_{k+1}}\right]_{xyz} = \theta_{err}$

$\therefore \theta_{err} = \frac{1}{2} J^q_{b^g} \delta b^g$

$\therefore$ we can construct $Hx = b$

Initial state:

$X_I = [V_0^{b_0}, V_1^{b_1}, \cdots, V_n^{b_n}, g^{c_0}, s]^T$

where $V_k^{b_k}$ is the body frame velocity in body frame at time k.

g^{c_0} is gravity at c_0 frame.

s is scale.

In world frame w: not observable, set it to (0, 0, 0).

$$\alpha_{b_ib_j} = q_{b_iw}\left(p_{wb_j} - p_{wb_i} - V_i^w \Delta t + \frac{1}{2} g^w \Delta t^2\right)$$
$$\beta_{b_ib_j} = q_{b_iw}\left(V_j^w - V_i^w + g^w \Delta t\right) \quad (13)$$

Change world frame w to c_0, (13) becomes

$$\alpha_{b_kb_{k+1}} = R_{b_kc_0}\left(s(\bar{p}_{c_0b_{k+1}} - \bar{p}_{c_0b_k}) + \frac{1}{2} g^{c_0} \Delta t_k^2 - R_{c_0b_k} V_k^{b_k} \Delta t_k\right)$$
$$\beta_{b_kb_{k+1}} = R_{b_kc_0}\left(R_{c_0b_{k+1}} V_{k+1}^{b_{k+1}} + g^{c_0} \Delta t_k - R_{c_0b_k} V_k^{b_k}\right) \quad (14)$$

$q_{c_0b_k} = q_{c_0c_k} \otimes q_{bc}^{-1}$

$s\bar{p}_{c_0b_k} = s\bar{p}_{c_0c_k} - R_{c_0b_k} p_{bc}$

$$\therefore \alpha_{b_kb_{k+1}} = sR_{b_kc_0}(\bar{p}_{c_0c_{k+1}} - \bar{p}_{c_0c_k}) - R_{b_kc_0}R_{c_0b_{k+1}} p_{bc} + p_{bc} + \frac{1}{2} R_{b_kc_0} g^{c_0} \Delta t_k^2 - V_k^{b_k} \Delta t_k \quad (15)$$

$$\therefore \hat{z}^{b_k}_{b_{k+1}} = \begin{bmatrix} \hat{\alpha}_{b_kb_{k+1}} - p_{bc} + R_{b_kc_0} R_{c_0b_{k+1}} p_{bc} \\ \hat{\beta}_{b_kb_{k+1}} \end{bmatrix} = H^{b_k}_{b_{k+1}} X^k_I + n^{b_k}_{b_{k+1}} \quad (16)$$

($R_{b_kc_0}R_{c_0b_{k+1}} = R_{b_kb_{k+1}}$; known. g^{c_0}, velocity $V_k^{b_k}$, s are unknown)

From Eq. 13 to Eq. 14:

$p_{wb_j} \to \bar{p}_{c_0b_{k+1}}$, $p_{wb_i} \to \bar{p}_{c_0b_k}$

$V_i^w \Delta t \to R_{c_0b_k} V_k^{b_k} \Delta t_k$ velocity is in body frame

$g^w \to g^{c_0}$. g^w is (0, 0, -9.81). g^{c_0} is unknown.

$\therefore X^k_I = [V_k^{b_k}, V_{k+1}^{b_{k+1}}, g^{c_0}, s]^T$

$$H^{b_k}_{b_{k+1}} = \begin{bmatrix} -I\Delta t_k & 0 & \frac{1}{2} R_{b_kc_0} \Delta t_k^2 & R_{b_kc_0}(\bar{p}^{c_0}_{c_{k+1}} - \bar{p}^{c_0}_{c_k}) \\ -I & R_{b_kc_0} R_{c_0b_{k+1}} & R_{b_kc_0} \Delta t_k & 0 \end{bmatrix}$$

$$\therefore \min_{X_I} \sum_{k\in B} \left\| \hat{z}^{b_k}_{b_{k+1}} - H^{b_k}_{b_{k+1}} X^k_I \right\|^2_\Sigma \Rightarrow \Lambda X = b$$

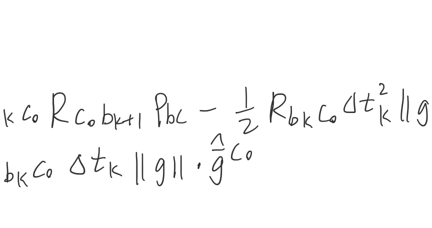
code: initial_alignment.cpp, LinearAlignment() in VINS Mono.

Estimate g^{c_0}:

In the optimization above, we don't use $\|g^{c_0}\| = 9.81$. In fact, g^{c_0} only has 2 Dof, not 3 Dof.

If a normal vector's norm = 1, then it's moving on a sphere

$\hat{g}^{c_0} = \|g\| \cdot \hat{\bar{g}}^{c_0} + w_1 \vec{b}_1 + w_2 \vec{b}_2$



where w_1, w_2 are parameters.

$\vec{b}_1 = \begin{cases} (\hat{\bar{g}}^{c_0} \times [1, 0, 0]), & \hat{\bar{g}}^{c_0} \neq [1, 0, 0]^T \\ (\hat{\bar{g}}^{c_0} \times [0, 0, 1]), & \text{otherwise} \end{cases}$

$\vec{b}_2 = \hat{\bar{g}}^{c_0} \times \vec{b}_1$

$\|g\| = 9.81$, $\hat{\bar{g}}^{c_0}$ is a normalized vector.

$\therefore$ We have reparameterized g^{c_0} to have 2 Dof.

Substitute new parameterization:

$$X^k_I = \begin{bmatrix} V_k^{b_k} \\ V_{k+1}^{b_{k+1}} \\ g^{c_0} \\ s \end{bmatrix} \to \begin{bmatrix} V_k^{b_k} \\ V_{k+1}^{b_{k+1}} \\ w^{c_0} \\ s \end{bmatrix}$$

$$\therefore \hat{z}^{b_k}_{b_{k+1}} = \begin{bmatrix} \alpha_{b_kb_{k+1}} - p_{bc} + R_{b_kc_0} R_{c_0b_{k+1}} p_{bc} - \frac{1}{2} R_{b_kc_0} \Delta t_k^2 \|g\| \cdot \hat{\bar{g}}^{c_0} \\ \beta_{b_kb_{k+1}} - R_{b_kc_0} \Delta t_k \|g\| \cdot \hat{\bar{g}}^{c_0} \end{bmatrix}$$

We can then optimize X^k_I.

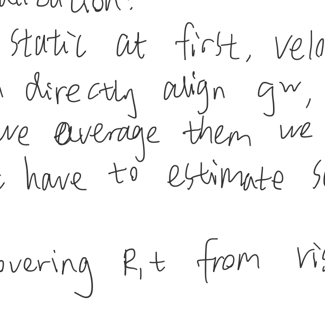
Align camera frame to world frame:

1. find c_0 to w transformation:

$R_{wc_0} = \exp([\theta u])$ — $g^{c_0} g^w \sin\theta$, $g^{c_0} g^w \cos\theta$

$u = \frac{\hat{g}^{c_0} \times \hat{g}^w}{\|\hat{g}^{c_0} \times \hat{g}^w\|}$, $\theta = atan2(\|\hat{g}^{c_0} \times \hat{g}^w\|, \hat{g}^{c_0} \cdot \hat{g}^w)$

2. transform all variables in c_0 frame to w frame.
3. find the scale.



we don't use g^{c_0}, but use (0, 0, -9.81) in our system. Afterwards, we'll use (0, 0, 9.81) in every keyframe's optimization.

The bias for acc. is very small, and can be ignored compared with g^w.

$\therefore$ In VINS, we don't estimate bias of acc.

Also, we don't estimate p_{bc}, since p_{bc} is linear, and p_{bc} is very small.

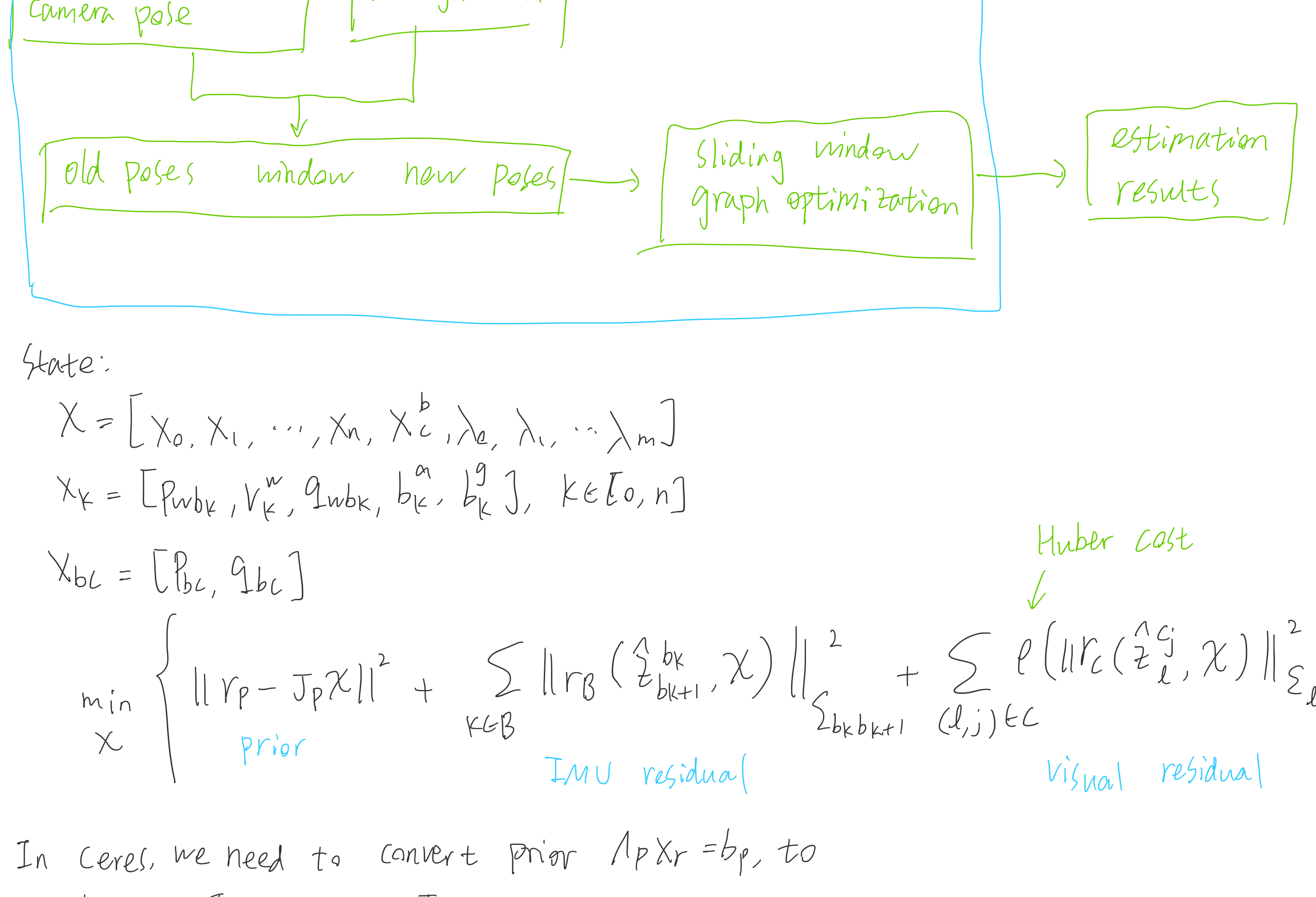
Alternative initialization:

1. When robot is static at first, velocity $V_0 = 0$. IMU acc equals to g^{b_0}, so we can directly align g^w, g^{b_0}. The gyro measurement is $w = 0 + b + n$, if we average them we get gyro bias. If we are using stereo, we don't have to estimate scale.

2. Instead of recovering R, t from visual measurements, use the visual features directly.

3. Instead of aligning camera trajectory and IMU trajectory, since IMU measurements are noisy, we shouldn't integrate them. The camera poses are less noisy, so we can use b spline to interpolate IMU measurements, to compare with the real measurements.

VINS review:
1. Front end: keypoint extraction, preintegration.
2. Initialization: initialize state, eg. gravity direction, velocity, scale.
3. Back end: sliding window optimization.



State:

$X = [x_0, x_1, \cdots, x_n, x^b_c, \lambda_0, \lambda_1, \cdots \lambda_m]$

$x_k = [p_{wb_k}, v^w_k, q_{wb_k}, b^a_k, b^g_k]$, $k \in [0, n]$

$x_{bc} = [p_{bc}, q_{bc}]$

$$\min_X \left\{ \|r_p - J_p X\|^2 + \sum_{k\in B} \left\| r_B(\hat{z}^{b_k}_{b_{k+1}}, X) \right\|^2_{\Sigma_{b_kb_{k+1}}} + \sum_{(l,j)\in C} \rho\left(\left\| r_C(\hat{z}^{c_j}_l, X) \right\|^2_{\Sigma^{c_j}_l} \right) \right\}$$

(prior; IMU residual; visual residual; ρ: Huber cost)

In Ceres, we need to convert prior $\Lambda_p X_r = b_p$, to

$\Lambda_p, b_p \to J_p^T J_p X_r = -J_p^T r_p$

Since Ceres cannot directly use info. matrix.