Supplementary Material to: Direct Sparse Visual-Inertial Odometry with Stereo Cameras

Ziqiang Wang, Chengcheng Guo

April, 2019

News: We present evaluation results on EuRoC dataset and a video at：

https://youtu.be/sam8bpsO0V0

Ziqiang Wang is an engineer of UISEE Technologies Shanghai Co., Ltd and was a postgraduate student with the college of Electronics and Information Engineering of Tongji Universty, Shanghai, China.(e-mail: ziqiang.wang@ uisee.com).

Content

Chapter0 Euroc evaluation results

We tested the proposed Stereo-VI-DSO on part of sequences in EuRoC dataset [1], in which a FireFly hex-rotor helicoptere quipped with VI-sensor (an IMU @ 200Hz and dual cameras 752×480 pixels @ 20Hz) was used for data collection.

In Table I, we present results of Stereo-DSO and Stereo-VI-DSO. The results of Stereo-DSO come from our approach removing the IMU constraint. VINS[5] and OKVIS[3] are open-source and the state of the art works. For comparison, we also provide accuracy RMSE results of VINS, OKVIS.

We also present all robot states estimation results in Fig1-27. We can draw a conclusion that Stereo-VI-DSO have a significant improvement over Stereo-DSO in accuracy.

TABLE I: Accuracy of the estimated trajectory on the EuRoC dataset for several methods. We run and calculate RMSE of VINS and OKVIS in our own laptop.

	Length(m)	Stereo-DSO		Stereo-VI-DSO		VINS		OKVIS
		Orien.(deg)	Pos.(m)	Orien.(deg)	Pos.(m)	Orien.(deg)	Pos.(m)	Pos. (m)
MH1	78.7	17.808	1.682	16.798	1.126	4.536	0.444	0.597
MH2	70.1	14.652	1.933	11.461	1.136	3.939	0.327	0.698
MH3	72.4	12.224	3.913	10.352	0.801	8.612	0.335	0.551

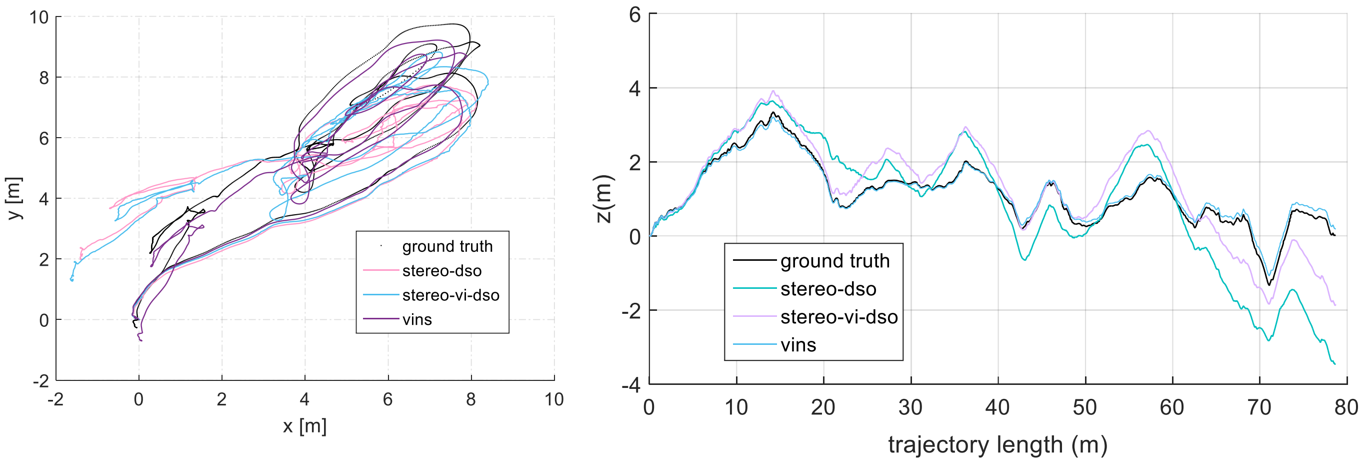


Fig. 1: Trajectory and height estimates in MH1.

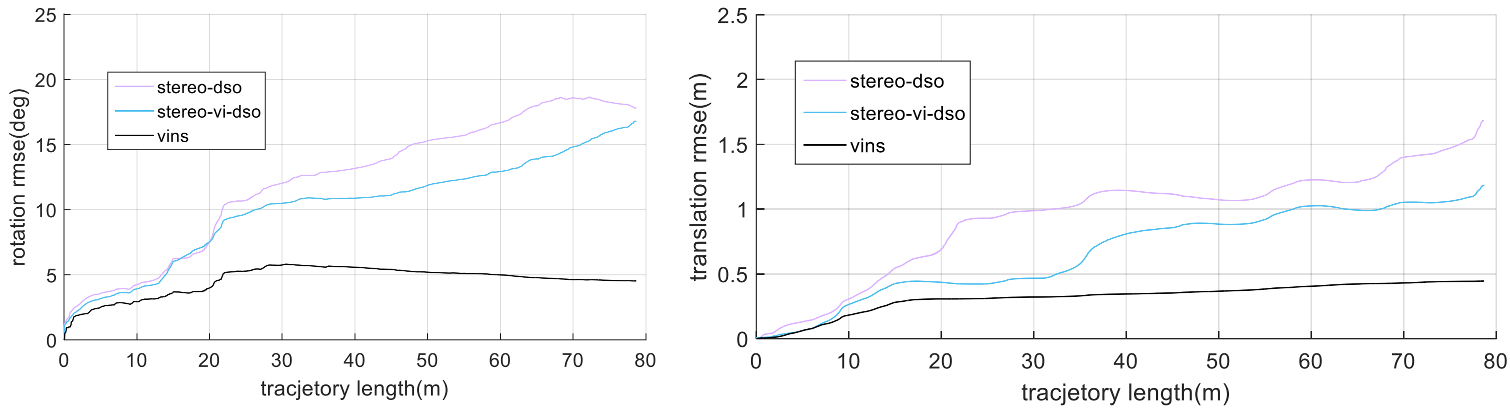
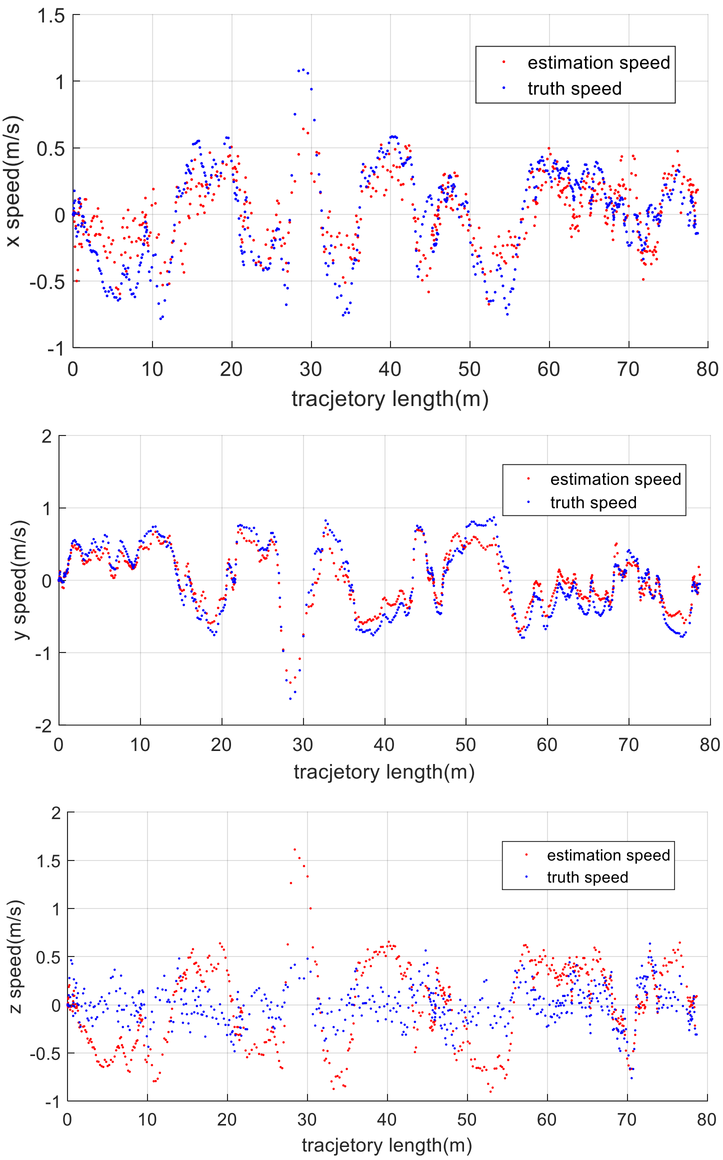


Fig. 2: Rotation and translation rmse in MH1. Rmse increases as distance travelled increases.



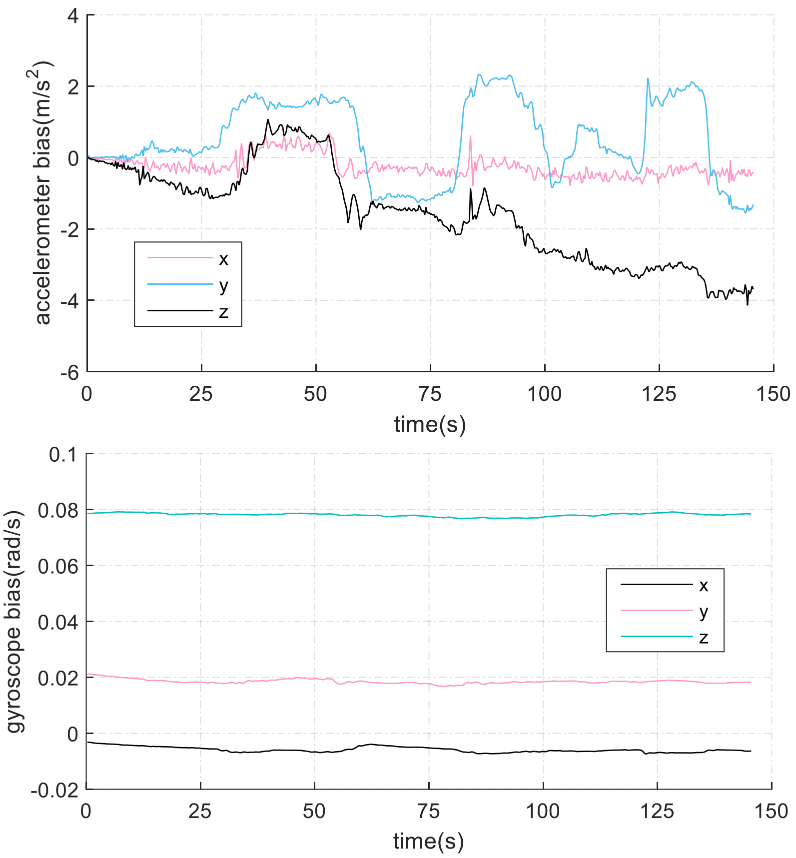


Fig. 3: Velocity and IMU bias estimates in MH1. It can be seen that the estimated velocity change is similar to the truth values and gyro bias converged to stable values.

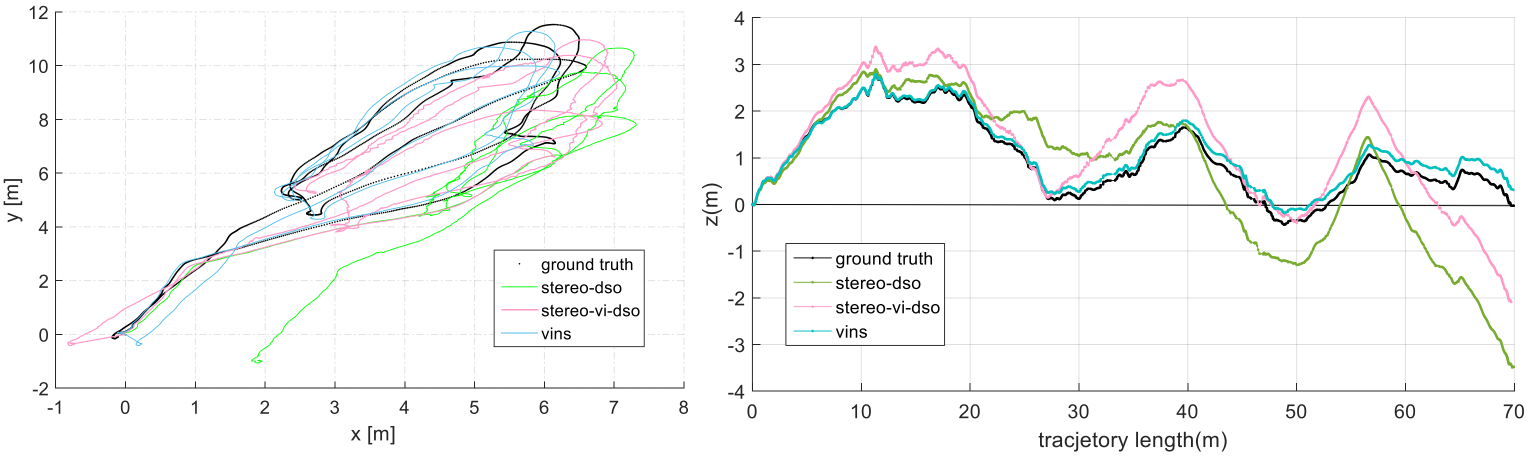


Fig. 4: Trajectory and height estimates in MH2.

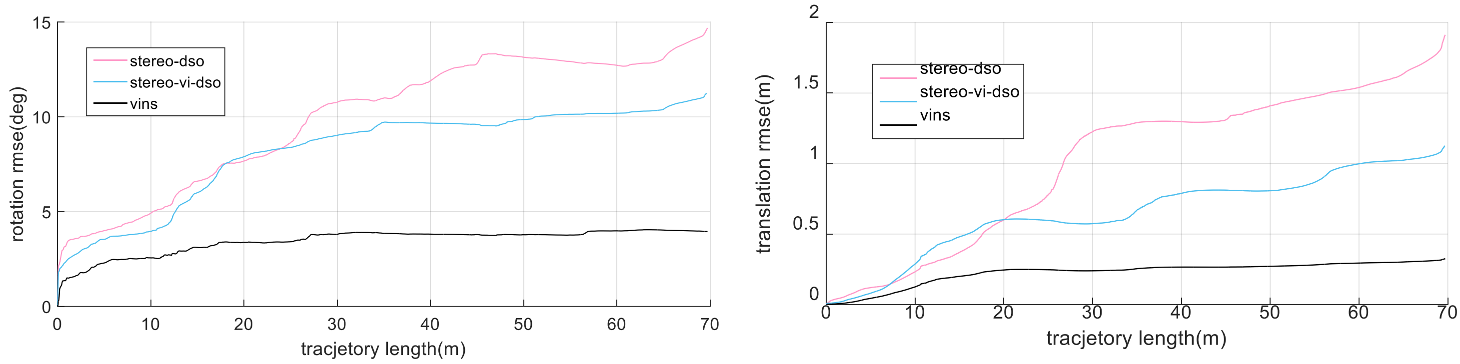
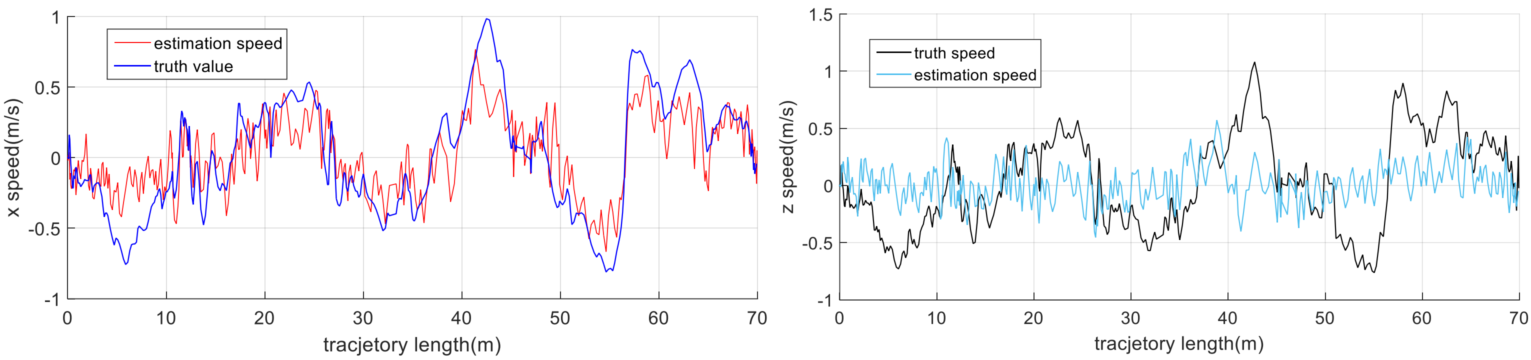


Fig. 5: Rotation and translation rmse in MH2.



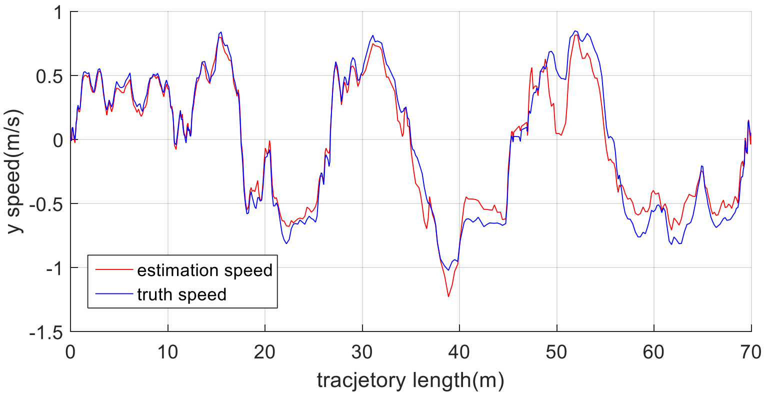


Fig. 6: Velocity estimates in MH2.

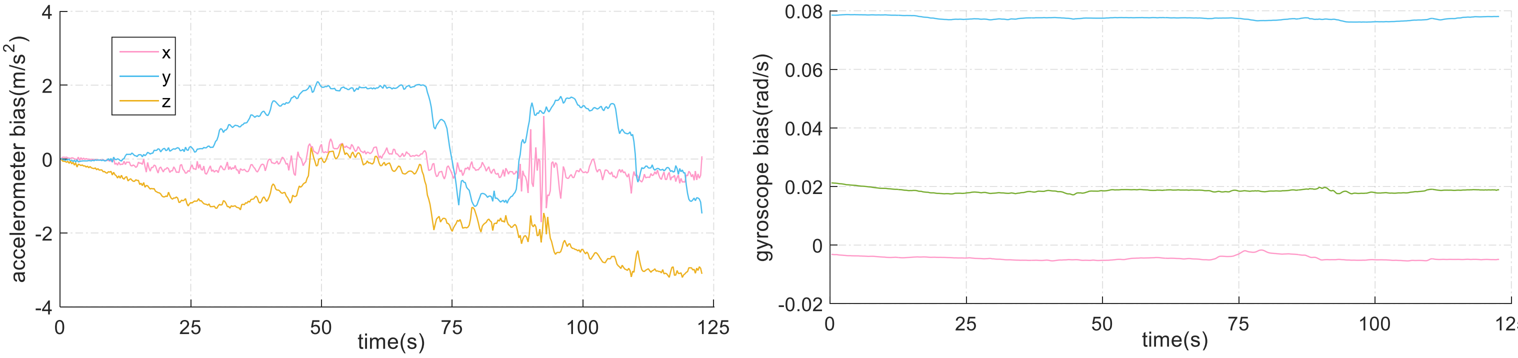


Fig. 7: IMU bias estimates in MH2.

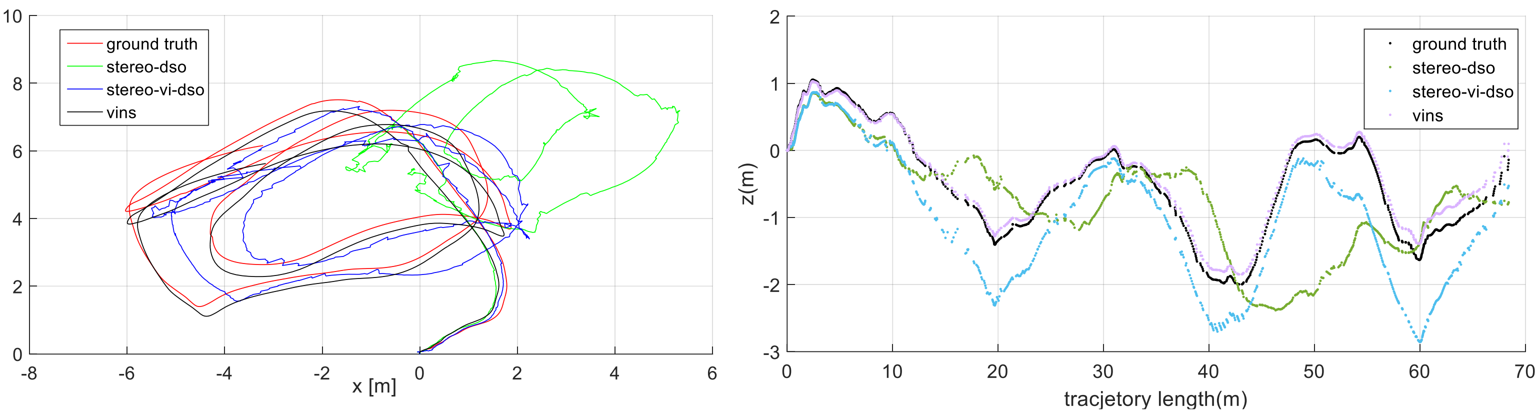


Fig. 8: Trajectory and height estimates in MH3

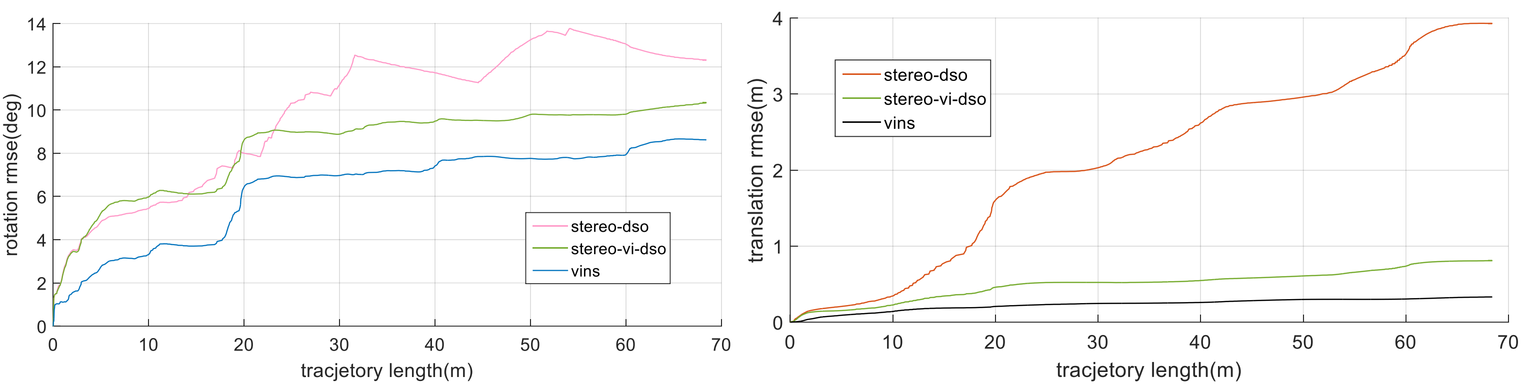


Fig. 9: Rotation and translation rmse in MH3.

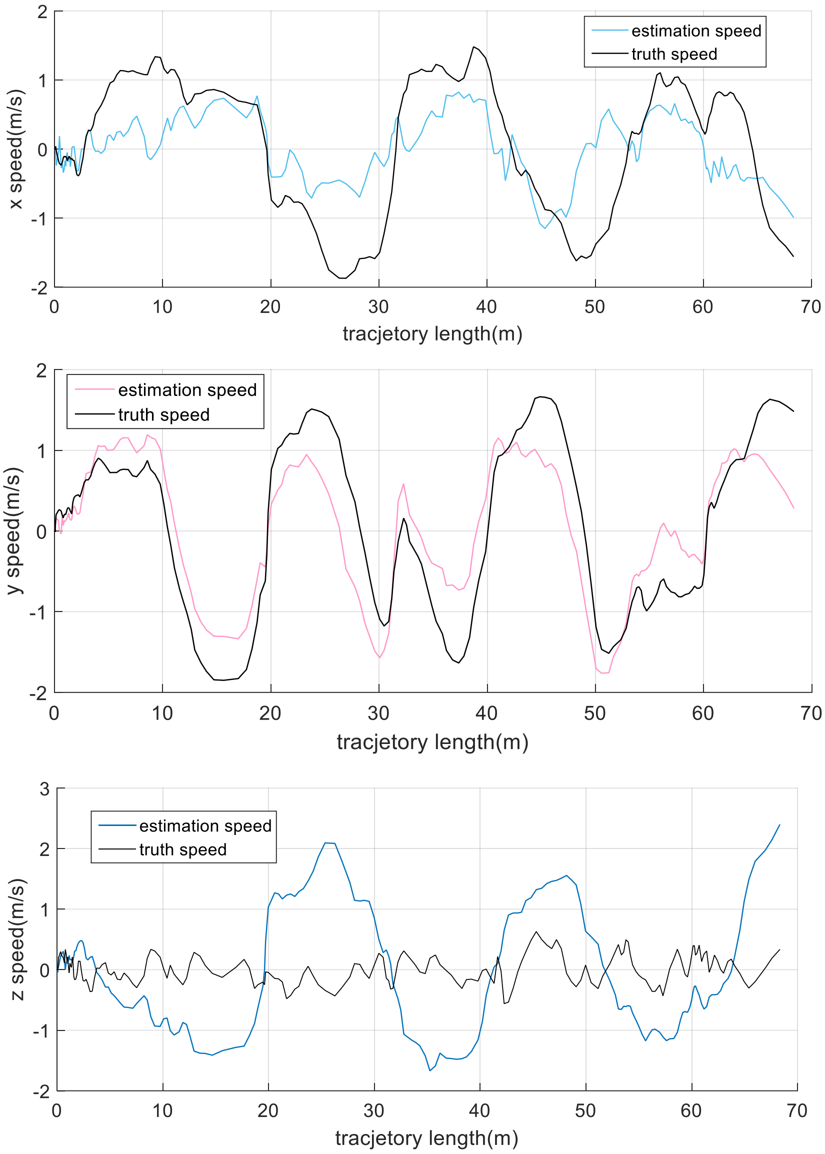


Fig. 10: Velocity estimates in MH3.

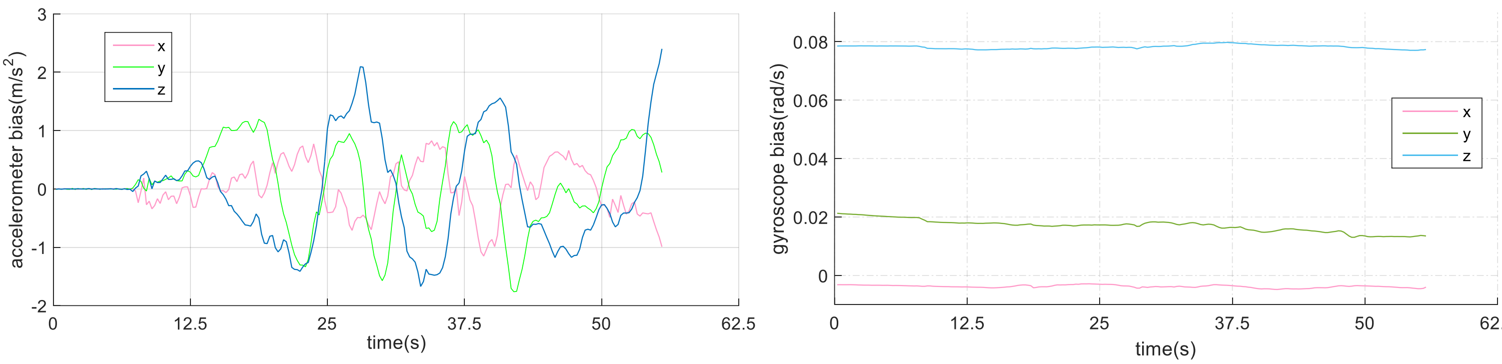


Fig. 11: IMU bias estimates in MH3.

Chapter1 Visual-inertial Preliminaries

In our main paper [Ⅳ], The term $\mathrm{J}_r(\boldsymbol{\xi})$ and its inverse are the right jacobian of $SE(3)$. In [6], authors have given the formula derivation of left Jacobian. We follow them and give derivation of right jacobian in (1.1) and (1.2) .

$$\begin{aligned}
&\mathrm{Exp}(\boldsymbol{\xi}^\wedge) = \mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}_{4\times 4}, \mathrm{Exp}(\delta\boldsymbol{\xi}^\wedge) = \begin{pmatrix} \delta\boldsymbol{\phi}^\wedge & \delta\boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{pmatrix}_{4\times 4}, \mathbf{p} \in \mathbb{R}^3 \\
&\frac{\partial(\mathbf{T}\mathbf{p})}{\partial\delta\boldsymbol{\xi}} = \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\mathrm{Exp}(\boldsymbol{\xi}^\wedge)\mathrm{Exp}(\delta\boldsymbol{\xi}^\wedge)\mathbf{p} - \mathrm{Exp}(\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&\approx \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\mathrm{Exp}(\boldsymbol{\xi}^\wedge)(\mathbf{I}+\delta\boldsymbol{\xi}^\wedge)\mathbf{p} - \mathrm{Exp}(\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta\boldsymbol{\xi}} = \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\mathrm{Exp}(\boldsymbol{\xi}^\wedge)\delta\boldsymbol{\xi}^\wedge\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}\begin{pmatrix} \delta\boldsymbol{\phi}^\wedge\mathbf{p} + \delta\boldsymbol{\rho} \\ 1 \end{pmatrix}}{\delta\boldsymbol{\xi}} = \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\begin{pmatrix} \mathbf{R}\delta\boldsymbol{\phi}^\wedge\mathbf{p} + \mathbf{R}\delta\boldsymbol{\rho} + \mathbf{t} \\ 1 \end{pmatrix}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\begin{pmatrix} -\mathbf{R}\mathbf{p}^\wedge\delta\boldsymbol{\phi} + \mathbf{R}\delta\boldsymbol{\rho} + \mathbf{t} \\ 1 \end{pmatrix}}{\begin{pmatrix} \delta\boldsymbol{\rho} \\ \delta\boldsymbol{\phi} \end{pmatrix}} = \begin{pmatrix} \mathbf{R} & -\mathbf{R}\mathbf{p}^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{pmatrix}_{4\times 6}
\end{aligned} \tag{1.1}$$

$$\begin{aligned}
&\frac{\partial(\mathbf{T}^{-1}\mathbf{p})}{\partial\delta\boldsymbol{\xi}} = \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{(\mathrm{Exp}(\boldsymbol{\xi}^\wedge)\mathrm{Exp}(\delta\boldsymbol{\xi}^\wedge))^{-1}\mathbf{p} - \mathrm{Exp}(-\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{(\mathrm{Exp}(\delta\boldsymbol{\xi}^\wedge))^{-1}(\mathrm{Exp}(\boldsymbol{\xi}^\wedge))^{-1}\mathbf{p} - \mathrm{Exp}(-\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\mathrm{Exp}(-\delta\boldsymbol{\xi}^\wedge)\mathrm{Exp}(-\boldsymbol{\xi}^\wedge)\mathbf{p} - \mathrm{Exp}(-\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&\approx \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{(\mathbf{I}-\delta\boldsymbol{\xi}^\wedge)\mathrm{Exp}(-\boldsymbol{\xi}^\wedge)\mathbf{p} - \mathrm{Exp}(-\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta\boldsymbol{\xi}} = \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} -\frac{\delta\boldsymbol{\xi}^\wedge\mathrm{Exp}(-\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} -\frac{\begin{pmatrix} \delta\boldsymbol{\phi}^\wedge & \delta\boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{pmatrix}\begin{pmatrix} \mathbf{R}^{-1} & -\mathbf{R}^{-1}\mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}\begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} -\frac{\begin{pmatrix} \delta\boldsymbol{\phi}^\wedge & \delta\boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{pmatrix}\begin{pmatrix} \mathbf{R}^{-1}\mathbf{p} - \mathbf{R}^{-1}\mathbf{t} \\ 1 \end{pmatrix}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} -\frac{\begin{pmatrix} \delta\boldsymbol{\phi}^\wedge(\mathbf{R}^{-1}\mathbf{p} - \mathbf{R}^{-1}\mathbf{t}) + \delta\boldsymbol{\rho} \\ 1 \end{pmatrix}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} -\frac{\begin{pmatrix} -(\mathbf{R}^{-1}\mathbf{p} - \mathbf{R}^{-1}\mathbf{t})^\wedge\delta\boldsymbol{\phi} + \delta\boldsymbol{\rho} \\ 1 \end{pmatrix}}{\begin{pmatrix} \delta\boldsymbol{\rho} \\ \delta\boldsymbol{\phi} \end{pmatrix}} = \begin{pmatrix} -\mathbf{I}_3 & (\mathbf{R}^{-1}\mathbf{p} - \mathbf{R}^{-1}\mathbf{t})^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{pmatrix}_{4\times 6}
\end{aligned} \tag{1.2}$$

Homogeneous camera calibration matrices are denoted by K as (1.3). and homogeneous 2D image coordinate point p is represented by its image coordinate and inverse depth as (1.3) relative to its host keyframe i^L. Corresponding homogeneous 3D camera coordinate point $\mathbf{p}_{\mathtt{C}}$ is denoted as (1.3). $\Pi_{\mathbf{K}}$ are used to denote camera projection functions. The jacobian of $\mathtt{I}_i^L$, $\Pi_{\mathbf{K}}$ is denoted as (1.3)

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{K}^{-1} = \begin{pmatrix} f_x^{-1} & 0 & -f_x^{-1}c_x & 0 \\ 0 & f_y^{-1} & -f_y^{-1}c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{p} = \begin{pmatrix} u^i \\ v^i \\ 1 \\ d_{\mathbf{p}} \end{pmatrix}, \mathbf{p}_{\mathtt{C}} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, d_{\mathbf{p}} = z^{-1}, \mathbf{p} = d_{\mathbf{p}}\mathbf{K}\mathbf{p}_{\mathtt{C}} = \Pi_{\mathbf{K}}(\mathbf{p}_{\mathtt{C}}) \tag{1.3}$$

$$\frac{\partial(\mathtt{I}_i^L(\mathbf{p}))}{\partial \mathbf{p}} = (g_x, g_y, 0, 0), \frac{\partial \mathbf{p}}{\partial \mathbf{p}_{\mathtt{C}}} = \frac{\partial \Pi_{\mathbf{K}}(\mathbf{p}_{\mathtt{C}})}{\partial \mathbf{p}_{\mathtt{C}}} = \begin{pmatrix} f_x z^{-1} & 0 & -x f_x z^{-2} & 0 \\ 0 & f_y z^{-1} & -y f_y z^{-2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -z^{-2} & 0 \end{pmatrix}$$

Chapter2 IMU Error Factors

2.1 Time-closest measurements selection strategy

Because of asynchronous but same frequency for accelerometer and gyroscope data, there will be different quantity samples of these two sensors between two consecutive keyframes. We select time-closest gyroscope measurement for one accelerometer measurement accordind to (Alg.1)

Algorithm 1 Time-closest measurements selection

Input: $gyro\_list$,$acc\_list[s]$(an element in acc_list)
Output: $gyro\_measure$(time closest element in gyro_list)

```
function TIME_CLOSEST_SELECT(gyro_list, i)
    t ← acc_list[s].timestamp, i ← s
    while true do
        if i >= gyro_list.size then
            return gyro_list.back
        else
            t_now ← gyro_list[i].timestamp
            t_next ← gyro_list[i + 1].timestamp
            if t_now < t then
                if t_next > t then
                    t_front ← abs(t_now − t), t_back ← abs(t_next − t)
                    return t_front > t_back ? gyro_list[i + 1] : gyro_list[i]
                else
                    i = i + 1
                end if
            else if t_now > t then
                i = i − 1
            else
                return gyro_list[i]
            end if
        end if
    end while
end function
```

2.2 Errors and covariance calculation pseudo code

In our main paper [Fig. 2], we can get gyroscope, accelerometer data lists whose size is m, n. We have 8 error items to define:

$\Delta\bar{\mathrm{R}}_{ij}, \frac{\partial \Delta\bar{\mathbf{R}}_{ij}}{\partial \mathbf{b}^g}, \frac{\partial \Delta\bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}^a}, \frac{\partial \Delta\bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}^a}$ are pure rotation values and aren't related to accelerometer data.

$\Delta\bar{\mathbf{v}}_{ij}, \frac{\partial \Delta\bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}^g}, \Delta\bar{\mathrm{p}}_{ij}, \frac{\partial \Delta\bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}^g}$ are rotation "plus" translation values and are related to both gyroscope and accelerometer data.

We calculate these error items by recursion. As an example, the recurrence of $\Delta\bar{\mathrm{R}}_{ij}, \frac{\partial \Delta\bar{\mathbf{R}}_{ij}}{\partial \mathbf{b}^g}$ are presented here in (2.1), (2.2).

$$
\begin{aligned}
&\Delta\bar{\mathbf{R}}_{ik} = \begin{cases} \mathbf{I}_{3\times3}, & k=i \\ \prod_{m=i}^{k-1} \mathbf{Exp}((\tilde{\boldsymbol{\omega}}_m - \bar{\mathbf{b}}_i^g)\Delta t), & k>i \end{cases} \\
&e.g. \quad k: 0 \to 44, i=0 \\
&\quad \Delta\bar{\mathbf{R}}_{00} = \mathbf{I}_{3\times3} \\
&\quad \Delta\bar{\mathbf{R}}_{01} = \mathbf{Exp}((\tilde{\boldsymbol{\omega}}_0 - \bar{\mathbf{b}}_0^g)\Delta t) \\
&\quad \Delta\bar{\mathbf{R}}_{02} = \mathbf{Exp}((\tilde{\boldsymbol{\omega}}_0 - \bar{\mathbf{b}}_0^g)\Delta t)\mathbf{Exp}((\tilde{\boldsymbol{\omega}}_1 - \bar{\mathbf{b}}_0^g)\Delta t) \\
&\qquad \vdots \\
&\quad \Delta\bar{\mathbf{R}}_{0(44)} = \mathbf{Exp}((\tilde{\boldsymbol{\omega}}_0 - \bar{\mathbf{b}}_0^g)\Delta t)\mathbf{Exp}((\tilde{\boldsymbol{\omega}}_1 - \bar{\mathbf{b}}_0^g)\Delta t)\cdots\mathbf{Exp}((\tilde{\boldsymbol{\omega}}_{43} - \bar{\mathbf{b}}_0^g)\Delta t)
\end{aligned}
\tag{2.1}
$$

$$
\begin{aligned}
&\frac{\partial \Delta\bar{\mathbf{R}}_{ik}}{\partial \bar{\mathbf{b}}^g} = \begin{cases} \mathbf{0}_{3\times3}, & k=i \\ \sum_{m=i}^{k-1} -\Delta\bar{\mathbf{R}}_{m+1k}^T \mathbf{J}_r^m \Delta t, & k>i \end{cases} \\
&= \begin{cases} \mathbf{0}_{3\times3}, & k=i \\ \mathbf{J}_r^0 \Delta t, & k=i+1 \\ \Delta\bar{\mathbf{R}}_{(k-1)k}^T \frac{\partial \Delta\bar{\mathbf{R}}_{i(k-1)}}{\partial \bar{\mathbf{b}}^g} + \mathbf{J}_r^{k-1}\Delta t, & k>i+1 \end{cases} \\
&e.g. \quad i=0, \quad k: 0 \to 45 \\
&\quad \frac{\partial \Delta\bar{\mathbf{R}}_{00}}{\partial \bar{\mathbf{b}}^g} = \mathbf{0}_{3\times3} \\
&\quad \frac{\partial \Delta\bar{\mathbf{R}}_{01}}{\partial \bar{\mathbf{b}}^g} = \sum_{m=0}^{0} \Delta\bar{\mathbf{R}}_{(m+1)1}^T \mathbf{J}_r^m \Delta t = \Delta\bar{\mathbf{R}}_{11}^T \mathbf{J}_r^0 \Delta t = \mathbf{J}_r^0 \Delta t \\
&\quad \frac{\partial \Delta\bar{\mathbf{R}}_{02}}{\partial \bar{\mathbf{b}}^g} = \sum_{m=0}^{1} \Delta\bar{\mathbf{R}}_{(m+1)2}^T \mathbf{J}_r^m \Delta t = \Delta\bar{\mathbf{R}}_{12}^T \mathbf{J}_r^0 \Delta t + \Delta\bar{\mathbf{R}}_{22}^T \mathbf{J}_r^1 \Delta t = \Delta\bar{\mathbf{R}}_{12}^T \frac{\partial \Delta\bar{\mathbf{R}}_{01}}{\partial \bar{\mathbf{b}}^g} + \mathbf{J}_r^1 \Delta t \\
&\quad \frac{\partial \Delta\bar{\mathbf{R}}_{03}}{\partial \bar{\mathbf{b}}^g} = \sum_{m=0}^{2} \Delta\bar{\mathbf{R}}_{(m+1)2}^T \mathbf{J}_r^m \Delta t = \Delta\bar{\mathbf{R}}_{13}^T \mathbf{J}_r^0 \Delta t + \Delta\bar{\mathbf{R}}_{23}^T \mathbf{J}_r^1 \Delta t + \Delta\bar{\mathbf{R}}_{33}^T \mathbf{J}_r^2 \Delta t \\
&\qquad\qquad = (\Delta\bar{\mathbf{R}}_{12}\Delta\bar{\mathbf{R}}_{23})^T \mathbf{J}_r^0 \Delta t + \Delta\bar{\mathbf{R}}_{23}^T \mathbf{J}_r^1 \Delta t + \mathbf{J}_r^2 \Delta t \\
&\qquad\qquad = \Delta\bar{\mathbf{R}}_{23}^T \Delta\bar{\mathbf{R}}_{12}^T \mathbf{J}_r^0 \Delta t + \Delta\bar{\mathbf{R}}_{23}^T \mathbf{J}_r^1 \Delta t + \mathbf{J}_r^2 \Delta t \\
&\qquad\qquad = \Delta\bar{\mathbf{R}}_{23}^T \frac{\partial \Delta\bar{\mathbf{R}}_{02}}{\partial \bar{\mathbf{b}}^g} + \mathbf{J}_r^2 \Delta t \\
&\qquad\quad \vdots \\
&\quad \frac{\partial \Delta\bar{\mathbf{R}}_{0(44)}}{\partial \bar{\mathbf{b}}^g} = \sum_{m=0}^{43} \Delta\bar{\mathbf{R}}_{(m+1)44}^T \mathbf{J}_r^m \Delta t \\
&\qquad\qquad = \Delta\bar{\mathbf{R}}_{1(44)}^T \mathbf{J}_r^0 \Delta t + \Delta\bar{\mathbf{R}}_{2(44)}^T \mathbf{J}_r^1 \Delta t + \cdots + \Delta\bar{\mathbf{R}}_{43(44)}^T \mathbf{J}_r^{42} \Delta t + \Delta\bar{\mathbf{R}}_{44(44)}^T \mathbf{J}_r^{43} \Delta t \\
&\qquad\qquad = \Delta\bar{\mathbf{R}}_{43(44)}^T \frac{\partial \Delta\bar{\mathbf{R}}_{0(43)}}{\partial \bar{\mathbf{b}}^g} + \mathbf{J}_r^{43} \Delta t \\
&\quad \frac{\partial \Delta\bar{\mathbf{R}}_{0(45)}}{\partial \bar{\mathbf{b}}^g} = \sum_{m=0}^{44} \Delta\bar{\mathbf{R}}_{(m+1)45}^T \mathbf{J}_r^m \Delta t \\
&\qquad\qquad = \Delta\bar{\mathbf{R}}_{44(45)}^T \frac{\partial \Delta\bar{\mathbf{R}}_{0(44)}}{\partial \bar{\mathbf{b}}^g} + \mathbf{J}_r^{44} \Delta t
\end{aligned}
\tag{2.2}
$$

Furthermore, in order to calculate conveniently, we introduce a $rotate\_list$ to store all pure rotation values. All error items can be seen in (Alg.2).

Algorithm 1 On-Manifold Preintegeration for IMU

Input: $gyro\_list, acc\_list, m, n, rotate\_list$

Output: $(\Delta\bar{\mathbf{R}}_{ij}, \frac{\partial\Delta\bar{\mathbf{R}}_{ij}}{\partial\mathbf{b}^g}, \frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^a}, \frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^a}), (\Delta\bar{\mathbf{v}}_{ij}, \frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^g}, \Delta\bar{\mathbf{p}}_{ij}, \frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^g}), \Sigma_{ij}$

function IMU_PREINTEGERATION$(gyro_list, acc_list, m, n, rotate_list)$

  for all $gyro\_list[i], i: 0 \to m$ **do**

    $last\_r \leftarrow rotate\_list[i-1]$

    $rot.timestamp \leftarrow gyro\_list[i].timestamp$

    $rot.\omega \leftarrow gyro\_list[i].\omega - \bar{\mathbf{b}}_i^g$

    $rot.\Delta\bar{\mathbf{R}}_{ik} \leftarrow last\_r.\Delta\bar{\mathbf{R}}_{ik} * \mathtt{Exp}(rot.\omega * \Delta t)$

    $rot.\Delta\bar{\mathbf{R}}_{(k-1)k} \leftarrow \mathtt{Exp}(rot.\omega * \Delta t)$

    $rot.\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^g} \leftarrow \Delta\bar{\mathbf{R}}_{(k-1)k}^T * last\_r.\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^g} - \mathtt{J}_r(rot.\omega * \Delta t) * \Delta t$

    $rot.\frac{\partial\Delta\bar{\mathbf{v}}_{ik}}{\partial\mathbf{b}^a} \leftarrow last\_r.\frac{\partial\Delta\bar{\mathbf{v}}_{ik}}{\partial\mathbf{b}^a} - last\_r.\Delta\bar{\mathbf{R}}_{ik} * \Delta t$

    $rot.\frac{\partial\Delta\bar{\mathbf{p}}_{ik}}{\partial\mathbf{b}^a} \leftarrow last\_r.\frac{\partial\Delta\bar{\mathbf{p}}_{ik}}{\partial\mathbf{b}^a} + last\_r.\frac{\partial\Delta\bar{\mathbf{v}}_{ik}}{\partial\mathbf{b}^a} * \Delta t - \frac{1}{2}last\_r.\Delta\bar{\mathbf{R}}_{ik} * \Delta t^2$

    $rotate\_list.push(rot)$

  end for

  $\Delta\bar{\mathbf{R}}_{ij} = rotate\_list.end.\Delta\bar{\mathbf{R}}_{ik}$

  $\frac{\partial\Delta\bar{\mathbf{R}}_{ij}}{\partial\mathbf{b}^g} = rotate\_list.end.\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^g}$

  $\frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^a} = rotate\_list.end.\frac{\partial\Delta\bar{\mathbf{v}}_{ik}}{\partial\mathbf{b}^a}$

  $\frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^a} = rotate\_list.end.\frac{\partial\Delta\bar{\mathbf{p}}_{ik}}{\partial\mathbf{b}^a}$

  for all $acc\_list[i], i: 0 \to n$ **do**

    $cls\_r \leftarrow time\_closest\_select(rotate\_list, acc\_list[i])$

    $acc \leftarrow acc\_list[i] - \bar{\mathbf{b}}_i^a$

    $\Delta\bar{\mathbf{v}}_{ij} += cls\_r.\Delta\bar{\mathbf{R}}_{ik} * acc * \Delta t$

    $\frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^g} -= cls\_r.\Delta\bar{\mathbf{R}}_{ik} * acc^\wedge * cls\_r.\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^g} * \Delta t$

    $\Delta\bar{\mathbf{p}}_{ij} += \Delta\bar{\mathbf{v}}_{ij} * \Delta t + \frac{1}{2}cls\_r.\Delta\bar{\mathbf{R}}_{ik} * acc * \Delta t^2$

    $\frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^g} += cls\_r.\frac{\partial\Delta\bar{\mathbf{v}}_{ik}}{\partial\mathbf{b}^g}\Delta t - \frac{1}{2}cls\_r.\Delta\bar{\mathbf{R}}_{ik} * acc^\wedge * cls\_r.\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^g} * \Delta t^2$

    $$A = \begin{pmatrix} cls\_r.\Delta\bar{\mathbf{R}}_{(k-1)k}^T & \mathbf{0} & \mathbf{0} \\ -cls\_r.\Delta\bar{\mathbf{R}}_{ik} * acc^\wedge * \Delta t & \mathbf{I} & \mathbf{0} \\ -\frac{1}{2}cls\_r.\Delta\bar{\mathbf{R}}_{ik} * acc^\wedge * \Delta t^2 & \Delta t\mathbf{I} & \mathbf{I} \end{pmatrix}$$

    $$B = \begin{pmatrix} \mathtt{J}_r(rot.\omega * \Delta t) * \Delta t & \mathbf{0} \\ \mathbf{0} & cls\_r.\Delta\bar{\mathbf{R}}_{ik} * \Delta t \\ \mathbf{0} & \frac{1}{2}cls\_r.\Delta\bar{\mathbf{R}}_{ik} * \Delta t^2 \end{pmatrix}$$

    $\Sigma_{ij} = A * \Sigma_{ij} * A^T + B * \Sigma_\eta * B^T$

  end for

end function

2.3 Jacobian derivation

The derivation of the Jacobians of $\mathbf{r}_{\Delta\mathbf{R}_{ij}}, \mathbf{r}_{\Delta\mathbf{v}_{ij}}, \mathbf{r}_{\Delta\mathbf{p}_{ij}}$ likes (2.3), (2.4), (2.5).

$$
\begin{aligned}
\frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\mathbf{p}_i} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\boldsymbol{\phi}_i} &= -\mathrm{J}_r^{-1}(\mathbf{r}_{\Delta\mathbf{R}_{ij}})\mathbf{R}_j^T\mathbf{R}_i \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\mathbf{v}_i} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\mathbf{p}_j} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\boldsymbol{\phi}_j} &= \mathrm{J}_r^{-1}(\mathbf{r}_{\Delta\mathbf{R}_{ij}}) \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\mathbf{v}_j} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\mathbf{b}_i^a} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\mathbf{b}_i^g} &= -\mathrm{J}_r^{-1}(\mathbf{r}_{\Delta\mathbf{R}_{ij}})\mathrm{Exp}(\mathbf{r}_{\Delta\mathbf{R}_{ij}})^T\mathrm{J}_r(\frac{\partial \Delta\bar{\mathbf{R}}_{ij}}{\partial \mathbf{b}^g}\delta\mathbf{b}_i^g)\frac{\partial \Delta\bar{\mathbf{R}}_{ij}}{\partial \mathbf{b}^g}
\end{aligned}
\tag{2.3}
$$

$$
\begin{aligned}
\frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\mathbf{p}_i} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\boldsymbol{\phi}_i} &= (\mathbf{R}_i^T(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}))^{\wedge} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\mathbf{v}_i} &= -\mathbf{R}_i^T \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\mathbf{p}_j} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\boldsymbol{\phi}_j} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\mathbf{v}_j} &= \mathbf{R}_i^T \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\mathbf{b}_i^a} &= -\frac{\partial \Delta\bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}^a} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\mathbf{b}_i^g} &= -\frac{\partial \Delta\bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}^g}
\end{aligned}
\tag{2.4}
$$

$$
\begin{aligned}
\frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\mathbf{p}_i} &= -\mathbf{I} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\boldsymbol{\phi}_i} &= (\mathbf{R}_i^T(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2))^{\wedge} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\mathbf{v}_i} &= -\mathbf{R}_i^T\Delta t_{ij} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\mathbf{p}_j} &= \mathbf{R}_i^T\mathbf{R}_j \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\boldsymbol{\phi}_j} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\mathbf{v}_j} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\mathbf{b}_i^a} &= -\frac{\partial \Delta\bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}^a} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\mathbf{b}_i^g} &= -\frac{\partial \Delta\bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}^g}
\end{aligned}
\tag{2.5}
$$

Chapter3 Photo Error Factors

3.1 Construction residual errors

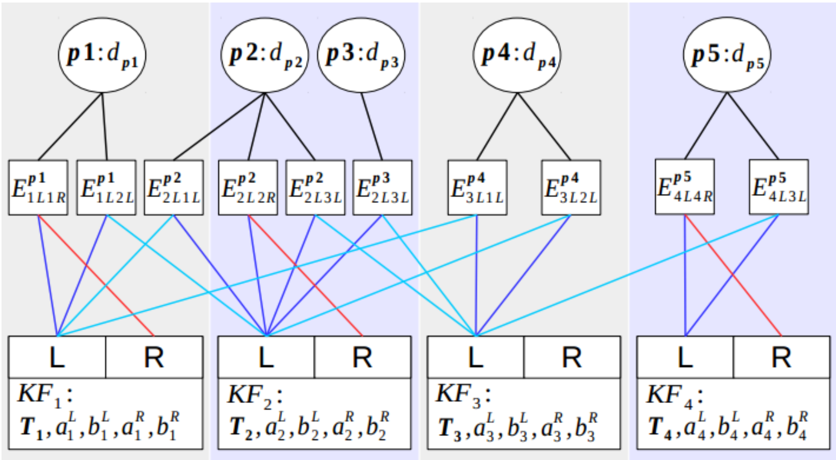


Fig.1

Here, we take [Fig.1] as factor graph to illustrate photometric error optimaztion. According to our main paper [V.B], The parameters we want to optimize are enclosed in (3.1).

$$\boldsymbol{\chi} = \begin{pmatrix} (\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_4)^T \\ (\mathbf{p}_1^T, \ldots, \mathbf{p}_4^T)^T \\ (\mathbf{v}_1^T, \ldots, \mathbf{v}_4^T)^T \\ (\mathbf{b}_1^T, \ldots, \mathbf{b}_4^T)^T \\ (d_{\mathbf{p}_1}, \ldots, d_{\mathbf{p}_5})^T \\ (a_1^L, a_1^R, b_1^L, b_1^R)^T \\ \vdots \\ (a_4^L, a_4^R, b_4^L, b_4^R)^T \end{pmatrix} \begin{matrix} \in \mathbb{R}^{81}, \\ \boldsymbol{\phi}_i = \mathrm{Log}(\mathbf{R}_i), \\ \boldsymbol{\xi}_i = (\boldsymbol{\phi}_i^T, \mathbf{p}_i^T)^T \end{matrix} \tag{3.1}$$

In this example, there are 7 dynamic residuals and 3 static residuals, Factor graph of the residuals function is in (3.2)

$$\begin{aligned} E(\boldsymbol{\chi}) &= E^{\mathbf{p}_1}_{1L2L} + E^{\mathbf{p}_2}_{2L1L} + E^{\mathbf{p}_2}_{2L3L} + E^{\mathbf{p}_3}_{2L3L} + E^{\mathbf{p}_4}_{3L1L} + E^{\mathbf{p}_4}_{3L2L} + E^{\mathbf{p}_5}_{4L3L} \\ &\quad + E^{\mathbf{p}_1}_{1L1R} + E^{\mathbf{p}_2}_{2L2R} + E^{\mathbf{p}_5}_{4L4R} \\ &= E_d(\boldsymbol{\chi}) + E_s(\boldsymbol{\chi}) \\ E_d(\boldsymbol{\chi}) &= \begin{pmatrix} (r^d_{\mathbf{p}_1})_{12} \\ (r^d_{\mathbf{p}_1})_{21} \\ \vdots \\ (r^d_{\mathbf{p}_5})_{43} \end{pmatrix}^T \begin{pmatrix} w_{\mathbf{p}_1} & 0 & \cdots & 0 \\ 0 & w_{\mathbf{p}_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{\mathbf{p}_5} \end{pmatrix} \begin{pmatrix} (r^d_{\mathbf{p}_1})_{12} \\ (r^d_{\mathbf{p}_1})_{21} \\ \vdots \\ (r^d_{\mathbf{p}_5})_{43} \end{pmatrix} = (\mathbf{r}^d)^T \mathbf{W}^d \mathbf{r}^d \\ E_s(\boldsymbol{\chi}) &= \begin{pmatrix} r^s_{\mathbf{p}_1} \\ r^s_{\mathbf{p}_2} \\ r^s_{\mathbf{p}_5} \end{pmatrix}^T \begin{pmatrix} \lambda w_{\mathbf{p}_1} & 0 & 0 \\ 0 & \lambda w_{\mathbf{p}_2} & 0 \\ 0 & 0 & \lambda w_{\mathbf{p}_5} \end{pmatrix} \begin{pmatrix} r^s_{\mathbf{p}_1} \\ r^s_{\mathbf{p}_2} \\ r^s_{\mathbf{p}_5} \end{pmatrix} = (\mathbf{r}^s)^T \mathbf{W}^s \mathbf{r}^s \end{aligned} \tag{3.2}$$

We first note that $(\mathbf{v}_1^T, \ldots, \mathbf{v}_4^T)^T, (\mathbf{b}_1^T, \ldots, \mathbf{b}_4^T)^T$ do not appear in the expression of $E_d(\boldsymbol{\chi}), E_s(\boldsymbol{\chi})$, hence the corresponding Jacobians are zero, we omit them for writing simplely. The remaining Jacobians can be computed as follows (3.3):

$$
\begin{aligned}
\mathbf{J}_s &= \begin{pmatrix}
\frac{\partial r^s_{\mathbf{p}_1}}{\partial \delta \boldsymbol{\xi}_1} \cdots & \frac{\partial r^s_{\mathbf{p}_1}}{\partial \delta \boldsymbol{\xi}_4} & \frac{\partial r^s_{\mathbf{p}_1}}{\partial d_{\mathbf{p}_1}} \cdots & \frac{\partial r^s_{\mathbf{p}_1}}{\partial d_{\mathbf{p}_5}} & \frac{\partial r^s_{\mathbf{p}_1}}{\partial a^L_1} \cdots & \frac{\partial r^s_{\mathbf{p}_1}}{\partial b^R_4} \\
\frac{\partial r^s_{\mathbf{p}_2}}{\partial \delta \boldsymbol{\xi}_1} \cdots & \frac{\partial r^s_{\mathbf{p}_2}}{\partial \delta \boldsymbol{\xi}_4} & \frac{\partial r^s_{\mathbf{p}_2}}{\partial d_{\mathbf{p}_1}} \cdots & \frac{\partial r^s_{\mathbf{p}_2}}{\partial d_{\mathbf{p}_5}} & \frac{\partial r^s_{\mathbf{p}_2}}{\partial a^L_1} \cdots & \frac{\partial r^s_{\mathbf{p}_2}}{\partial b^R_4} \\
\frac{\partial r^s_{\mathbf{p}_5}}{\partial \delta \boldsymbol{\xi}_1} \cdots & \frac{\partial r^s_{\mathbf{p}_5}}{\partial \delta \boldsymbol{\xi}_4} & \frac{\partial r^s_{\mathbf{p}_5}}{\partial d_{\mathbf{p}_1}} \cdots & \frac{\partial r^s_{\mathbf{p}_5}}{\partial d_{\mathbf{p}_5}} & \frac{\partial r^s_{\mathbf{p}_5}}{\partial a^L_1} \cdots & \frac{\partial r^s_{\mathbf{p}_5}}{\partial b^R_4}
\end{pmatrix}_{3\times 49} \\
\mathbf{J}_d &= \begin{pmatrix}
\frac{\partial (r^d_{\mathbf{p}_1})_{12}}{\partial \delta \boldsymbol{\xi}_1} \cdots & \frac{\partial (r^d_{\mathbf{p}_1})_{12}}{\partial \delta \boldsymbol{\xi}_4} & \frac{\partial (r^d_{\mathbf{p}_1})_{12}}{\partial d_{\mathbf{p}_1}} \cdots & \frac{\partial (r^d_{\mathbf{p}_1})_{12}}{\partial d_{\mathbf{p}_5}} & \frac{\partial (r^d_{\mathbf{p}_1})_{12}}{\partial a^L_1} \cdots & \frac{\partial (r^d_{\mathbf{p}_1})_{12}}{\partial b^R_4} & & \\
\frac{\partial (r^d_{\mathbf{p}_1})_{21}}{\partial \delta \boldsymbol{\xi}_1} \cdots & \frac{\partial (r^d_{\mathbf{p}_1})_{21}}{\partial \delta \boldsymbol{\xi}_4} & \frac{\partial (r^d_{\mathbf{p}_1})_{21}}{\partial d_{\mathbf{p}_1}} \cdots & \frac{\partial (r^d_{\mathbf{p}_1})_{21}}{\partial d_{\mathbf{p}_5}} & \frac{\partial (r^d_{\mathbf{p}_1})_{21}}{\partial a^L_1} \cdots & \frac{\partial (r^d_{\mathbf{p}_1})_{21}}{\partial b^R_4} & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial (r^d_{\mathbf{p}_5})_{43}}{\partial \delta \boldsymbol{\xi}_1} \cdots & \frac{\partial (r^d_{\mathbf{p}_5})_{43}}{\partial \delta \boldsymbol{\xi}_4} & \frac{\partial (r^d_{\mathbf{p}_5})_{43}}{\partial d_{\mathbf{p}_1}} \cdots & \frac{\partial (r^d_{\mathbf{p}_5})_{43}}{\partial d_{\mathbf{p}_5}} & \frac{\partial (r^d_{\mathbf{p}_5})_{43}}{\partial a^L_1} \cdots & \frac{\partial (r^d_{\mathbf{p}_5})_{43}}{\partial b^R_4} & &
\end{pmatrix}_{7\times 49}
\end{aligned}
\tag{3.3}
$$

Iteration $\delta\boldsymbol{\chi}$ can be calculated by (3.4):

$$
\begin{aligned}
&(\mathbf{J}_s^T \lambda \mathbf{W}^s \mathbf{J}_s + \mathbf{J}_d^T \mathbf{W}^d \mathbf{J}_d)\delta\boldsymbol{\chi} = -(\mathbf{J}_s^T \lambda \mathbf{W}^s \mathbf{r}^s + \mathbf{J}_d^T \mathbf{W}^d \mathbf{r}^d) \\
&\mathbf{J}_s \in \mathbb{R}^{3\times 49}, \mathbf{W}^s \in \mathbb{R}^{3\times 3}, \mathbf{J}_s \in \mathbb{R}^{7\times 49}, \mathbf{W}^s \in \mathbb{R}^{7\times 7}
\end{aligned}
\tag{3.4}
$$

3.2 Jacobian derivation

3.2.1 Dynamic Parameter

Firstly, if $\mathbf{p}$ is neither observed by frame m^L, m^R nor hosted by n^L, n^R, corresponding jacobians are zero as (3.6):

$$
\frac{\partial (r^d_{\mathbf{p}})_{ij}}{\partial \delta \boldsymbol{\xi}_m} = \frac{\partial (r^d_{\mathbf{p}})_{ij}}{\partial \delta \boldsymbol{\xi}_n} = \mathbf{0}^T, so \quad \frac{\partial (r^d_{\mathbf{p}_1})_{12}}{\partial \delta \boldsymbol{\xi}_3} = \frac{\partial (r^d_{\mathbf{p}_1})_{12}}{\partial \delta \boldsymbol{\xi}_4} = \ldots = \mathbf{0}^T,
\tag{3.6}
$$

Otherwise, assuming the hostframe of 2D image coordinate point $\mathbf{p}$ is i^L, and corresponding homogeneous 3D camera coordinate point is $\mathbf{p}_{\mathsf{C}}$ in (3.7), body coordinate is $\mathbf{p}_{\mathsf{B}} = \mathbf{T}_{\mathsf{BC}}\mathbf{p}_{\mathsf{C}}$. We transform $\mathbf{p}_{\mathsf{C}}$ from frame i^L to j^L by $\mathbf{p}_{\mathsf{B}}^{'} = \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{p}_{\mathsf{B}}$, then transform $\mathbf{p}_{\mathsf{B}}^{'}$ to camera coordinate point $\mathbf{p}_{\mathsf{C}}^{'} = \mathbf{T}_{\mathsf{BC}}^{-1}\mathbf{p}_{\mathsf{B}}^{'}$. At last, $\mathbf{p}_{\mathsf{C}}^{'}$ is projected to 2D image coordinate point with $\mathbf{p}^{'}$.

$$
\begin{aligned}
&\mathbf{p}_{\mathsf{C}} = \begin{pmatrix} f_x^{-1}(d_{\mathbf{p}}^{i^L})^{-1}(u^i - c_x) \\ f_y^{-1}(d_{\mathbf{p}}^{i^L})^{-1}(v^i - c_y) \\ (d_{\mathbf{p}}^{i^L})^{-1} \\ 1 \end{pmatrix}, \mathbf{p}_{\mathsf{C}}^{'} \doteq \begin{pmatrix} x^{'} \\ y^{'} \\ z^{'} \\ 1 \end{pmatrix} \\
&\mathbf{p}^{'} = d_{\mathbf{p}}^{j^L}\mathbf{K}(\mathbf{T}_{\mathsf{BC}}^{-1}\mathbf{T}_j^{-1}\mathbf{T}_i((d_{\mathbf{p}}^{i^L})^{-1}\mathbf{T}_{\mathsf{BC}}\mathbf{K}^{-1}\mathbf{p})) = d_{\mathbf{p}}^{j^L}\mathbf{K}\mathbf{p}_{\mathsf{C}}^{'}
\end{aligned}
\tag{3.7}
$$

3.2.1.1 Jacobian of Affine Brightness Parameters

It is convenient to give jacobian of affine brightness parameters in (3.8).

$$
\begin{aligned}
&(r^d_{\mathbf{p}})_{ij} = I^L_j(\mathbf{p}^{'}) - b^L_j - \frac{e^{a^L_j}}{e^{a^L_i}}(I^L_i(\mathbf{p}) - b^L_i) \\
&\frac{\partial (r^d_{\mathbf{p}})_{ij}}{\partial a^L_i} = \frac{e^{a^L_j}}{e^{a^L_i}}(I^L_i(\mathbf{p}) - b^L_i), \frac{\partial (r^d_{\mathbf{p}})_{ij}}{\partial a^L_j} = -\frac{e^{a^L_j}}{e^{a^L_i}}(I^L_i(\mathbf{p}) - b^L_i) \\
&\frac{\partial (r^d_{\mathbf{p}})_{ij}}{\partial b^L_i} = \frac{e^{a^L_j}}{e^{a^L_i}}, \qquad \frac{\partial (r^d_{\mathbf{p}})_{ij}}{\partial b^L_j} = -1
\end{aligned}
\tag{3.8}
$$

3.2.1.2 Right Jacobian of Pose

According to (1.1), we can use the chain rule to get jacobian of $\boldsymbol{\xi}_i$ in (3.9):

$$
\begin{aligned}
&(r^d_{\mathbf{p}})_{ij} = I^L_j(\mathbf{p}^{'}) - b^L_j - \frac{e^{a^L_j}}{e^{a^L_i}}(I^L_i(\mathbf{p}) - b^L_i) \\
&\frac{\partial (r^d_{\mathbf{p}})_{ij}}{\partial \delta \boldsymbol{\xi}_i} = \frac{\partial (I^L_j(\mathbf{p}^{'}))}{\partial \mathbf{p}^{'}} \frac{\partial \mathbf{p}^{'}}{\partial \mathbf{p}^{'}_{\mathtt{C}}} \frac{\partial \mathbf{p}^{'}_{\mathtt{C}}}{\partial \delta \boldsymbol{\xi}_i} \\
&\frac{\partial (I^L_j(\mathbf{p}^{'}))}{\partial \mathbf{p}^{'}} = (g^{'}_x, g^{'}_y, 0, 0)^T \\
&\frac{\partial \mathbf{p}^{'}}{\partial \mathbf{p}^{'}_{\mathtt{C}}} = \begin{pmatrix} f_x(z^{'})^{-1} & 0 & -x^{'} f_x(z^{'})^{-2} & 0 \\ 0 & f_y(z^{'})^{-1} & -y^{'} f_y(z^{'})^{-2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & (z^{'})^{-2} & 0 \end{pmatrix} \\
&\frac{\partial \mathbf{p}^{'}_{\mathtt{C}}}{\partial \delta \boldsymbol{\xi}_i} = \frac{\partial (\mathbf{T}^{-1}_{\mathtt{BC}} \mathbf{T}^{-1}_j \mathbf{T}_i \mathbf{p}_{\mathtt{B}})}{\partial \delta \boldsymbol{\xi}_i} = \mathbf{T}^{-1}_{\mathtt{BC}} \mathbf{T}^{-1}_j \frac{\partial (\mathbf{T}_i \mathbf{p}_{\mathtt{B}})}{\partial \delta \boldsymbol{\xi}_i} \\
&= \mathbf{T}^{-1}_{\mathtt{BC}} \mathbf{T}^{-1}_j \begin{pmatrix} \mathbf{R}_i & -\mathbf{R}_i \mathbf{p}^{\wedge}_{\mathtt{B}} \\ \mathbf{0}^T & \mathbf{0}^T \end{pmatrix}
\end{aligned}
\tag{3.9}
$$

According to (1.2), the jacobian of $\boldsymbol{\xi}_j$ is enclosed in (3.10):

$$
\begin{aligned}
&\mathbf{T}_i \mathbf{p}_{\mathtt{B}} \doteq {}_{\mathtt{i}}\mathbf{p}_{\mathtt{B}} \\
&\frac{\partial \mathbf{p}^{'}_{\mathtt{C}}}{\partial \delta \boldsymbol{\xi}_j} = \frac{\partial (\mathbf{T}^{-1}_{\mathtt{BC}} \mathbf{T}^{-1}_j \mathbf{T}_i \mathbf{p}_{\mathtt{B}})}{\partial \delta \boldsymbol{\xi}_j} = \mathbf{T}^{-1}_{\mathtt{BC}} \frac{\partial (\mathbf{T}^{-1}_j \, {}_{\mathtt{i}}\mathbf{p}_{\mathtt{B}})}{\partial \delta \boldsymbol{\xi}_j} \\
&= \mathbf{T}^{-1}_{\mathtt{BC}} \begin{pmatrix} -\mathbf{I}_3 & (\mathbf{R}^{-1}_j \, {}_{\mathtt{i}}\mathbf{p}_{\mathtt{B}} - \mathbf{R}^{-1}_j \mathbf{t}_j)^{\wedge} \\ \mathbf{0}^T & \mathbf{0}^T \end{pmatrix}
\end{aligned}
\tag{3.10}
$$

3.2.1.3 Jacobian of inverse Depth

The inverse depth of $\mathbf{p}$ is $d_{\mathbf{p}}^{i^L}$ in 3D camera coordinate of i^L. The jacobian of $d_{\mathbf{p}}^{i^L}$ is enclosed in (3.11):

$$
\begin{aligned}
\mathbf{p}' &= d_{\mathbf{p}}^{j^L}\mathbf{K}(\mathbf{T}_{\mathtt{BC}}^{-1}\mathbf{T}_j^{-1}\mathbf{T}_i((d_{\mathbf{p}}^{i^L})^{-1}\mathbf{T}_{\mathtt{BC}}\mathbf{K}^{-1}\mathbf{p})) \\
&= d_{\mathbf{p}}^{j^L}\mathbf{K}(\mathbf{T}_{\mathtt{BC}}^{-1}\mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{T}_{\mathtt{BC}})\mathbf{p}_{\mathtt{C}} \\
\mathbf{T}_{\mathtt{BC}}^{-1}\mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{T}_{\mathtt{BC}} &\doteq \mathbf{T}^{\spadesuit} \doteq \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
\mathbf{p}_{\mathtt{C}}' &= \mathbf{T}^{\spadesuit}\mathbf{p}_{\mathtt{C}} \\
&= \begin{pmatrix} r_{11}f_x^{-1}(d_{\mathbf{p}}^{i^L})^{-1}(u^i-c_x)+r_{12}f_y^{-1}(d_{\mathbf{p}}^{i^L})^{-1}(v^i-c_y)+r_{13}(d_{\mathbf{p}}^{i^L})^{-1}+t_1 \\ r_{21}f_x^{-1}(d_{\mathbf{p}}^{i^L})^{-1}(u^i-c_x)+r_{22}f_y^{-1}(d_{\mathbf{p}}^{i^L})^{-1}(v^i-c_y)+r_{23}(d_{\mathbf{p}}^{i^L})^{-1}+t_2 \\ r_{31}f_x^{-1}(d_{\mathbf{p}}^{i^L})^{-1}(u^i-c_x)+r_{32}f_y^{-1}(d_{\mathbf{p}}^{i^L})^{-1}(v^i-c_y)+r_{33}(d_{\mathbf{p}}^{i^L})^{-1}+t_3 \\ 1 \end{pmatrix} \\
&\doteq \begin{pmatrix} \frac{a}{d_{\mathbf{p}}^{i^L}}+t_1 \\ \frac{b}{d_{\mathbf{p}}^{i^L}}+t_2 \\ \frac{c}{d_{\mathbf{p}}^{i^L}}+t_3 \\ 1 \end{pmatrix} \doteq \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \Rightarrow d_{\mathbf{p}}^{j^L} = (z')^{-1}, \mathbf{p}' = \begin{pmatrix} f_x x' d_{\mathbf{p}}^{j^L}+c_x \\ f_y y' d_{\mathbf{p}}^{j^L}+c_y \\ 1 \\ d_{\mathbf{p}}^{jL} \end{pmatrix} \\
&\Rightarrow \frac{\partial(r_{\mathbf{p}}^d)_{ij}}{\partial d_{\mathbf{p}}^{i^L}} = \frac{\partial(I_j^L(\mathbf{p}'))}{\partial d_{\mathbf{p}}^{i^L}} = \frac{\partial(I_j^L(\mathbf{p}'))}{\partial \mathbf{p}'}\frac{\partial \mathbf{p}'}{\partial \mathbf{p}_{\mathtt{C}}'}\frac{\partial \mathbf{p}_{\mathtt{C}}'}{\partial d_{\mathbf{p}}^{i^L}} \\
&= -\frac{g_x' f_x a}{z'(d_{\mathbf{p}}^{i^L})^2} - \frac{g_y' f_y b}{z'(d_{\mathbf{p}}^{i^L})^2} + \frac{c(g_x' x' f_x + g_y' y' f_y)}{(z' d_{\mathbf{p}}^{i^L})^2} \\
&= \frac{c(g_x' x' f_x + g_y' y' f_y) - g_x' f_x a z' - g_y' f_y b z'}{(z' d_{\mathbf{p}}^{i^L})^2}
\end{aligned}
\tag{3.11}
$$

3.2.2 Static Parameter

Firstly, $\boldsymbol{\xi}_i, \boldsymbol{\xi}_j$ do not appear in the expression of $r^s_{\mathbf{p}}$ as (3.12), the corresponding jacobians are zero.

$$r^s_{\mathbf{p}} := I^R_i(\mathbf{p}') - b^R_i - \frac{e^{a^R_i}}{e^{a^L_i}}(I^L_i(\mathbf{p}) - b^L_i) \tag{3.12}$$

Secondly, we can follow chapter 3.2.1.3 to calculate jacobians of inverse depth. But some strategies can be used to reduce computation. For a pair of stereo frame i^L, i^R: inverse depth $d^{i^L}_{\mathbf{p}} \approx d^{i^R}_{\mathbf{p}}$, and $\mathbf{T}_{\text{RL}}$ is only related to baseline of stereo cameras. Left frame i^L pixel $\mathbf{p}$ is projected to right frame i^R with $\mathbf{p}'$ as (3.13) :

$$\begin{aligned}
&\mathbf{p} = \begin{pmatrix} u^i \\ v^i \\ 1 \\ d^{i^L}_{\mathbf{p}} \end{pmatrix}, \mathbf{p}_{\text{C}} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, d^{i^L}_{\mathbf{p}} = z^{-1}, \mathbf{p}_{\text{C}} = (d^{i^L}_{\mathbf{p}})^{-1}\mathbf{K}^{-1}\mathbf{p} \\
&= \begin{pmatrix} f_x^{-1}(d^{i^L}_{\mathbf{p}})^{-1}(u^i - c_x) \\ f_y^{-1}(d^{i^L}_{\mathbf{p}})^{-1}(v^i - c_y) \\ (d^{i^L}_{\mathbf{p}})^{-1} \\ 1 \end{pmatrix}, \mathbf{T}_{\text{RL}} = \begin{pmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
&\mathbf{p}' = d^{i^R}_{\mathbf{p}}\mathbf{K}(\mathbf{T}_{\text{RL}}\mathbf{p}_{\text{C}}) \\
&= d^{i^L}_{\mathbf{p}} \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_x^{-1}(d^{i^L}_{\mathbf{p}})^{-1}(u^i - c_x) + t_1 \\ f_y^{-1}(d^{i^L}_{\mathbf{p}})^{-1}(v^i - c_y) \\ (d^{i^L}_{\mathbf{p}})^{-1} \\ 1 \end{pmatrix} = \begin{pmatrix} u^i + t_1 f_x d^{i^L}_{\mathbf{p}} \\ v^i \\ 1 \\ d^{i^L}_{\mathbf{p}} \end{pmatrix} \\
&\frac{\partial r^s_{\mathbf{p}}}{\partial d^{i^L}_{\mathbf{p}}} = \frac{\partial(I^R_i(\mathbf{p}')) - \frac{e^{a^R_i}}{e^{a^L_i}}(I^L_i(\mathbf{p}))}{\partial d^{i^L}_{\mathbf{p}}} = (\frac{\partial(I^R_i(\mathbf{p}'))}{\partial \mathbf{p}'} - \frac{e^{a^R_i}}{e^{a^L_i}}\frac{\partial(I^L_i(\mathbf{p}))}{\partial \mathbf{p}'})\frac{\partial \mathbf{p}'}{\partial d^{i^L}_{\mathbf{p}}} \\
&= [(g^{i^R}_x, g^{i^R}_y, 0, 0) - \mathbf{0}^T] \begin{pmatrix} t_1 f_x \\ 0 \\ 0 \\ 1 \end{pmatrix} = g^{i^R}_x t_1 f_x
\end{aligned} \tag{3.13}$$

At last, we give jacobian of affine brightness parameters in (3.14).

$$\begin{aligned}
&\frac{\partial(r^s_{\mathbf{p}})_{ij}}{\partial a^L_i} = \frac{e^{a^R_i}}{e^{a^L_i}}(I^L_i(\mathbf{p}) - b^L_i), \frac{\partial(r^s_{\mathbf{p}})_{ij}}{\partial a^R_i} = -\frac{e^{a^R_i}}{e^{a^L_i}}(I^L_i(\mathbf{p}) - b^L_i) \\
&\frac{\partial(r^s_{\mathbf{p}})_{ij}}{\partial b^L_i} = \frac{e^{a^R_i}}{e^{a^L_i}}, \qquad \frac{\partial(r^s_{\mathbf{p}})_{ij}}{\partial b^R_i} = -1
\end{aligned} \tag{3.14}$$

REFERENCES

[1] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W.Achtelik, and R. Siegwart, “The euroc micro aerial vehicle datasets,” The International Journal of Robotics Research, vol. 35, no. 10, pp.1157–1163, 2016.

[2] R. Mur-Artal and J. D. Tardos, “Visual-inertial monocular slam with map reuse,” IEEE Robot. and Autom. Lett., vol. 2, no. 2, 2017.

[3] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, ″Keyframe-based visual–inertial odometry using nonlinear optimization,” The International Journal of Robotics Research, vol. 34, no. 3, pp. 314–334, 2015.

[4] L. von Stumberg, V. Usenko and D. Cremers, “Direct Sparse Visual-Inertial Odometry using Dynamic Marginalization”, In International Conference on Robotics and Automation (ICRA), 2018.

[5] T. Qin , L. Peiliang , and S. Shaojie . "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator." IEEE Transactions on Robotics (2018):1-17.

[6] T.barfoot, “State estimation for robotics”, Cambridge University Press, pp. 245-254, 2017.