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

Ziqiang Wang, Chengcheng Guo

Starts on September 2018

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 University, Shanghai, China.(e-mail: ziqiang.wang@ uisee.com).

Content

Chapter0 Euroc evaluation results.....	3
Chapter1 Visual-inertial Preliminaries.....	10
Chapter2 IMU Error Factors.....	12
2.1 Time-closest measurements selection strategy.....	12
2.2 Errors and covariance calculation pseudo code	12
2.3 Jacobian derivation	15
Chapter3 Photo Error Factors.....	16
3.1 Construction residual errors	16
3.2 Jacobian derivation	17
3.2.1 Dynamic Parameter.....	17
3.2.2 Static Parameter	20
References.....	21

Chapter0 Euroc evaluation results

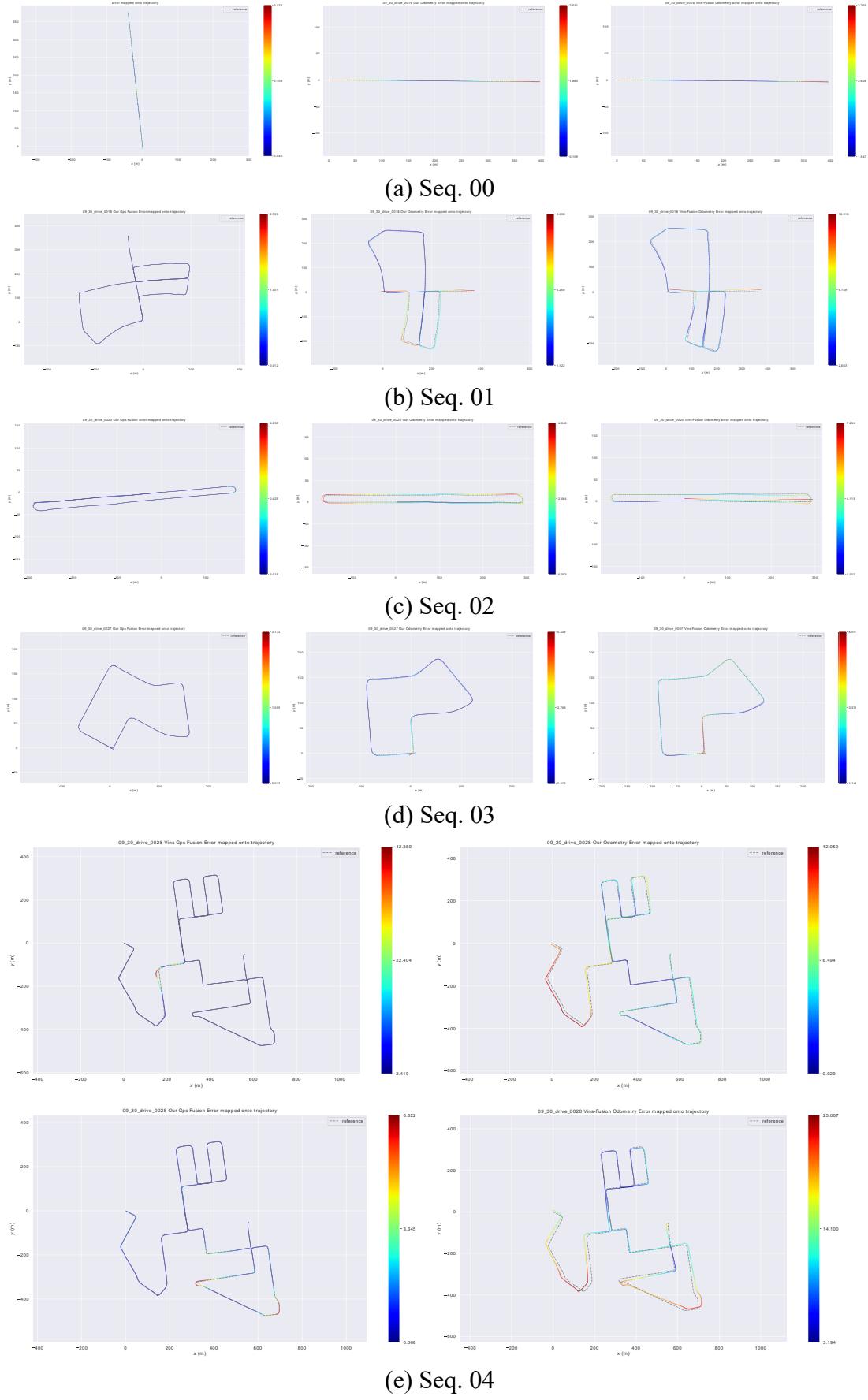
We tested the proposed Stereo-VI-DSO on part of sequences in EuRoC dataset [1], in which a FireFly hex-rotor helicoptere quipped with VI-sensor (an IMU @ 200Hz and dual cameras 752×480 pixels @ 20Hz) was used for data collection.

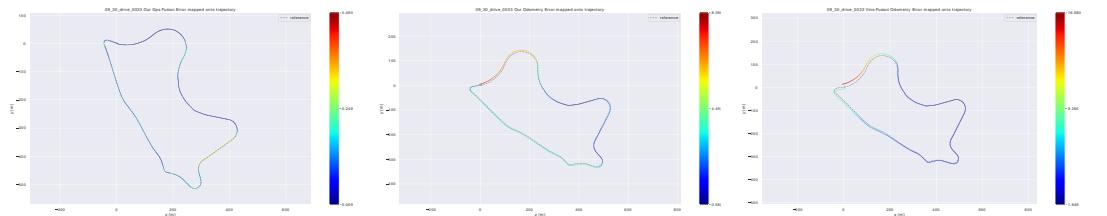
In Table I, we present results of Stereo-DSO and Stereo-VI-DSO. The results of Stereo-DSO come from our approach removing the IMU constraint. VINS[5] and OKVIS[3] are open-source and the state of the art works. For comparison, we also provide accuracy RMSE results of VINS, OKVIS.

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.

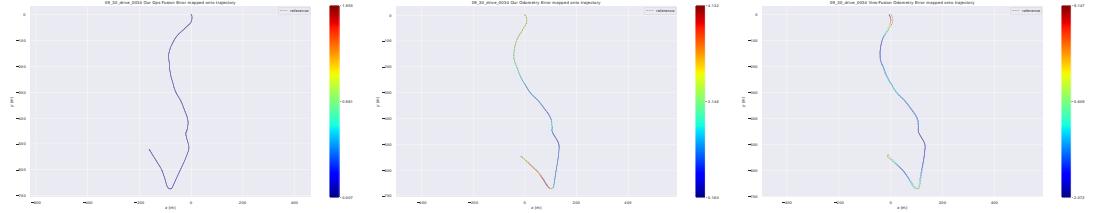
TABLE I: Accuracy of the estimated trajectory on the EuRoC dataset for several methods. We run and calculate RMSE of VINS and OKVIS in our own laptop.

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
MH2	70.1	14.652	1.933	11.461	1.136	3.939	0.327
MH3	72.4	12.224	3.913	10.352	0.801	8.612	0.335

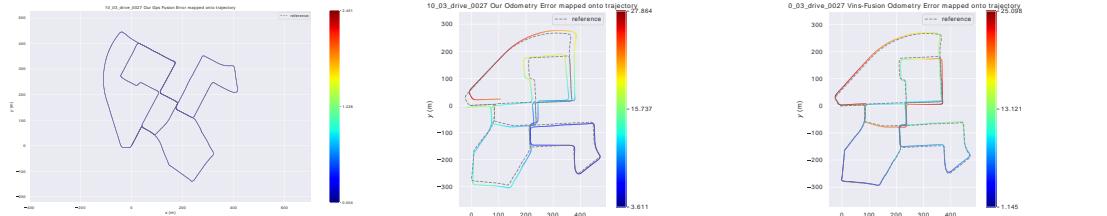




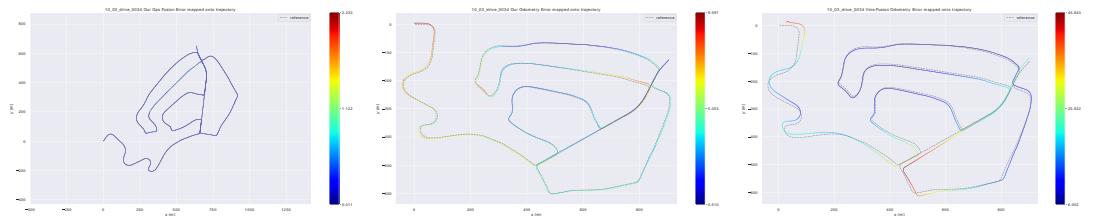
(f) Seq. 05



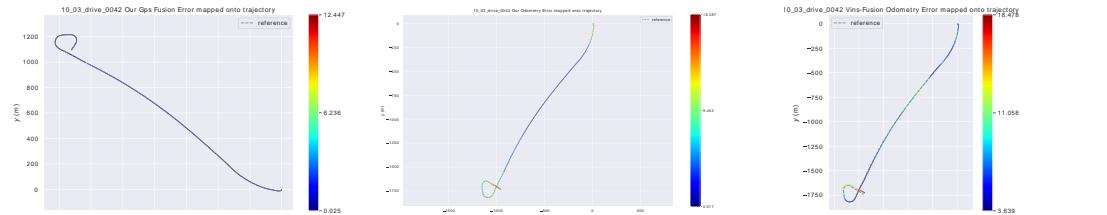
(g) Seq. 06



(h) Seq. 07



(i) Seq. 08



(j) Seq. 09

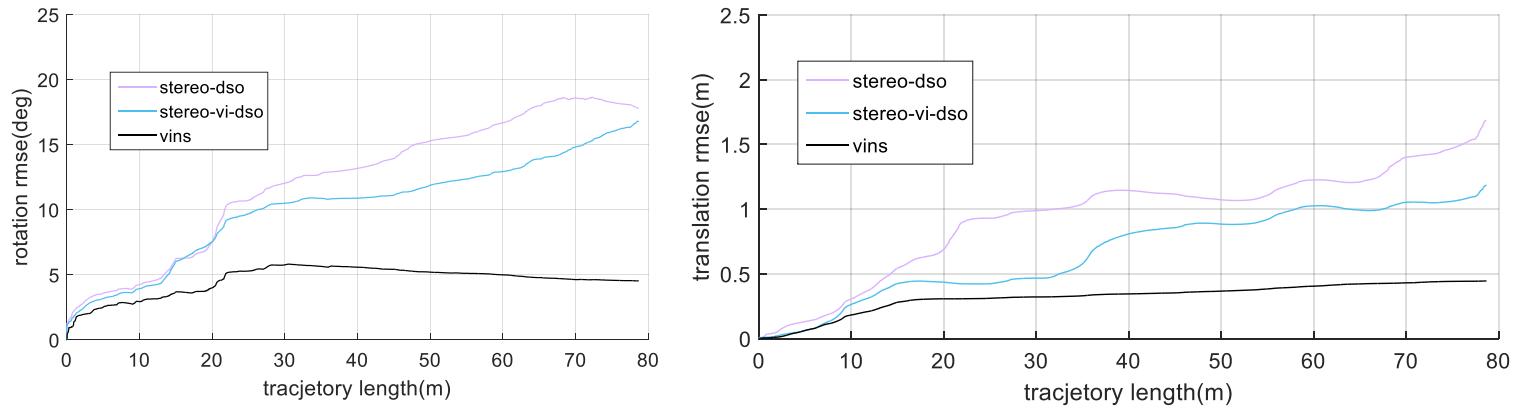


Fig. 2: Rotation and translation rmse in MH1. Rmse increases as distance travelled increases.

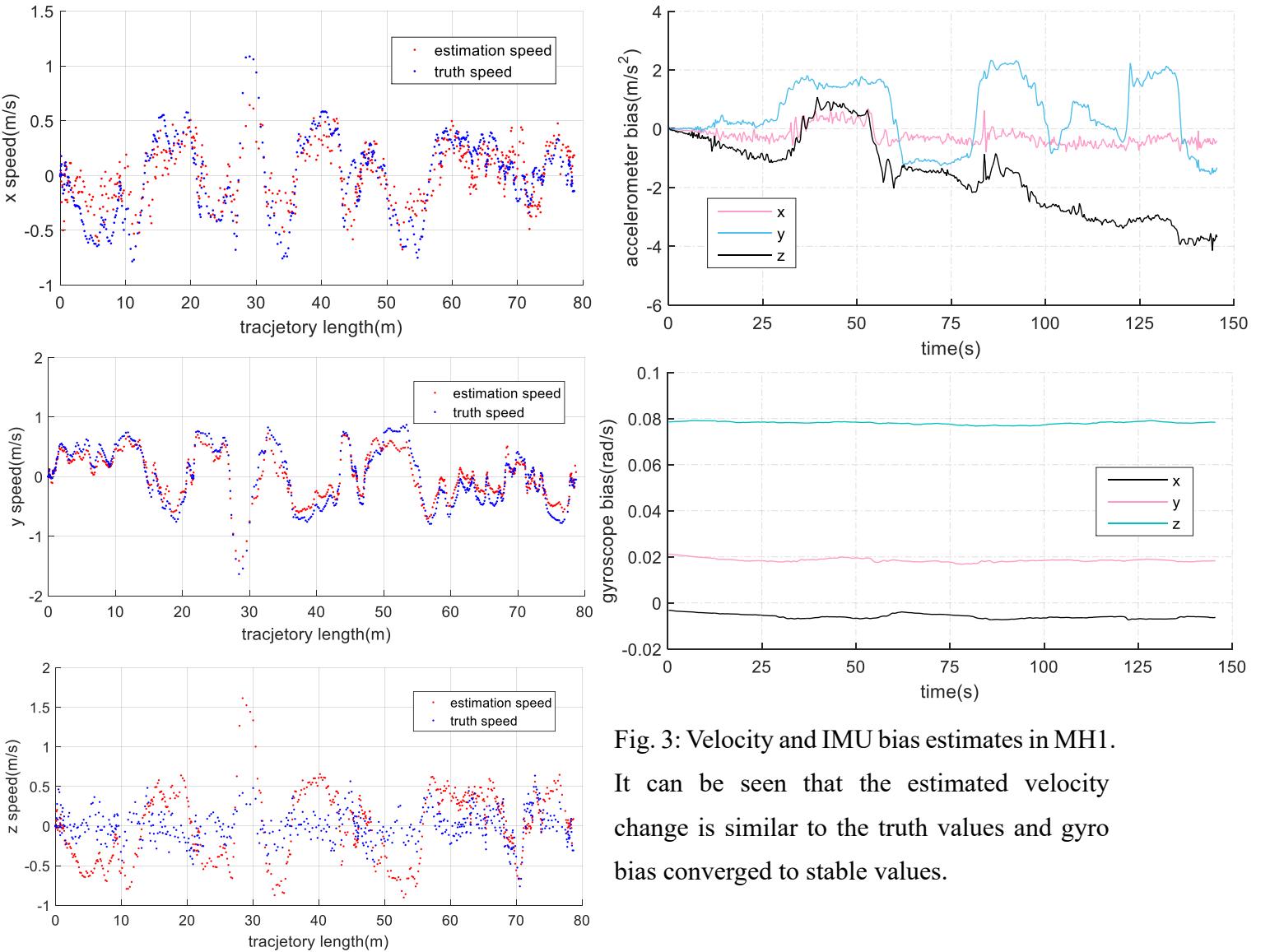


Fig. 3: Velocity and IMU bias estimates in MH1.
It can be seen that the estimated velocity change is similar to the truth values and gyro bias converged to stable values.

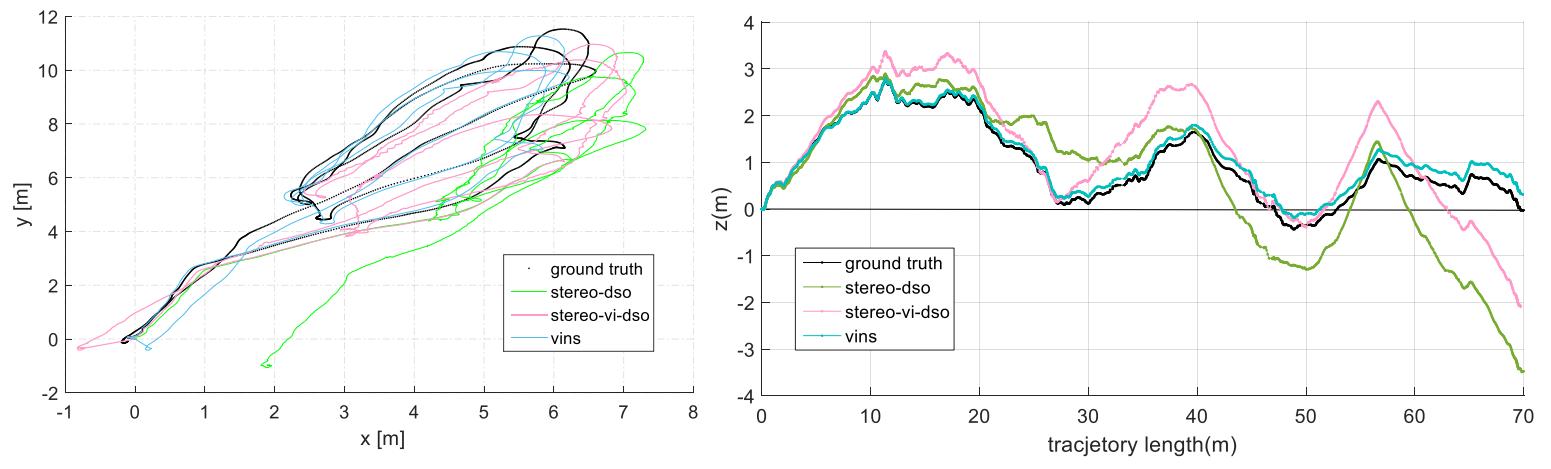


Fig. 4: Trajectory and height estimates in MH2.

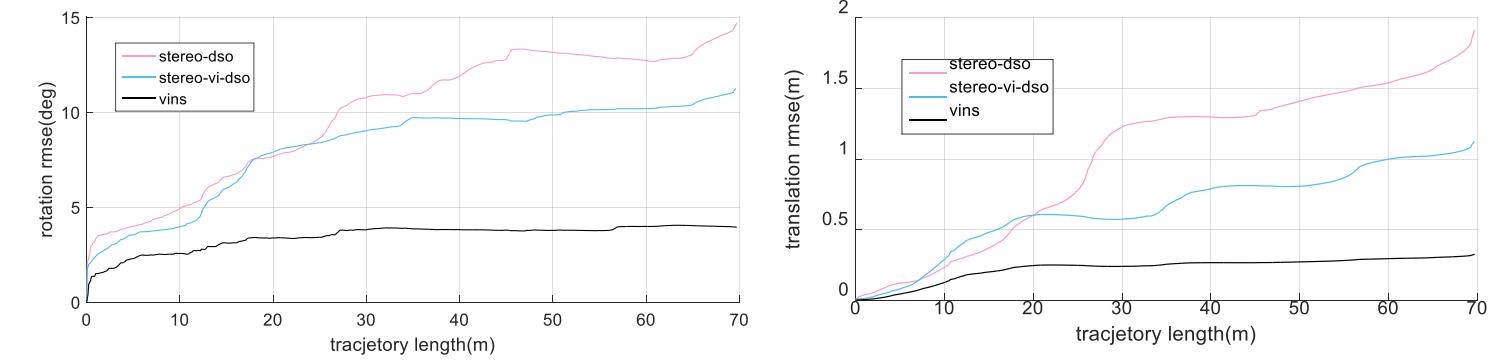


Fig. 5: Rotation and translation rmse in MH2.

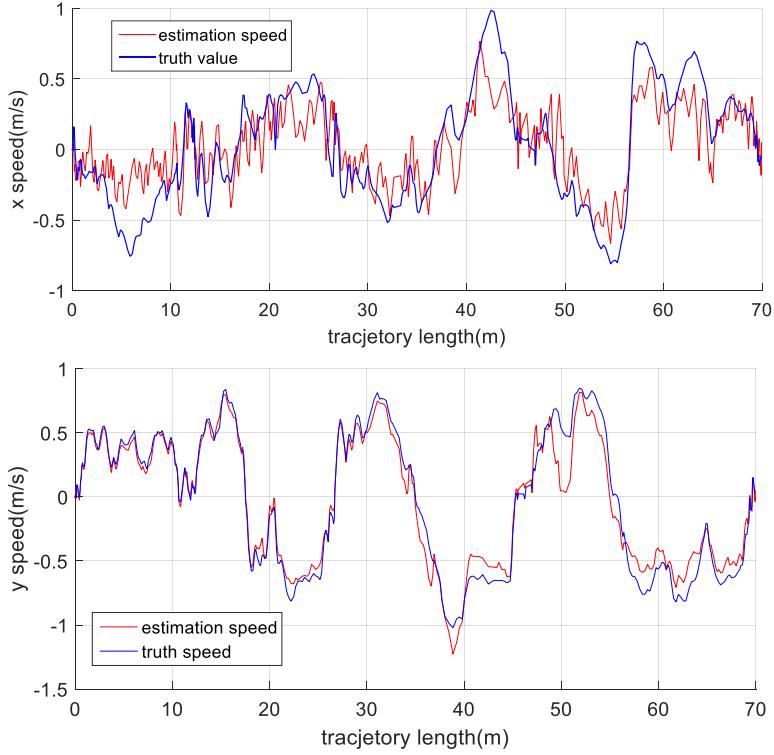


Fig. 6: Velocity estimates in MH2.

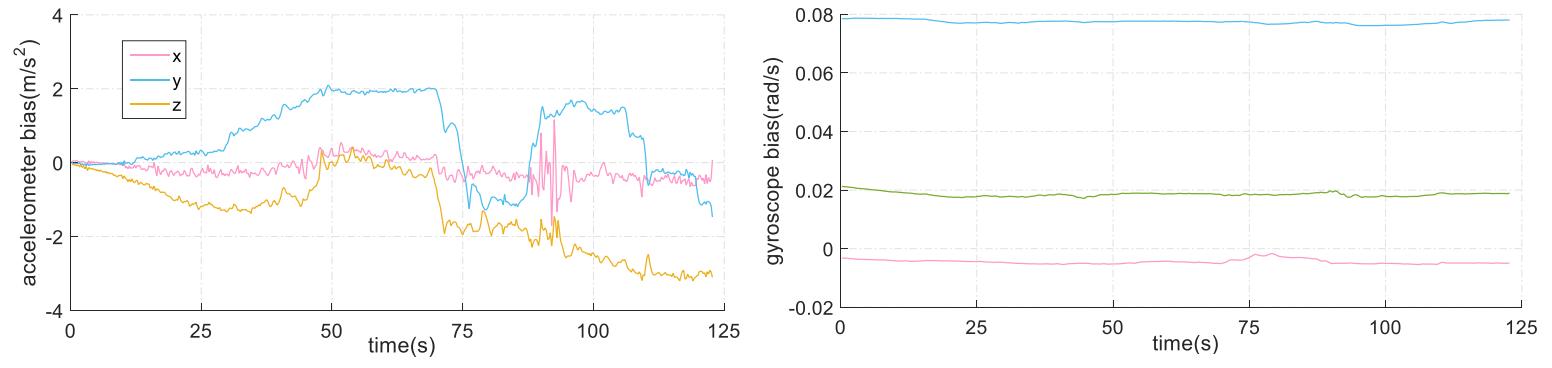


Fig. 7: IMU bias estimates in MH2.

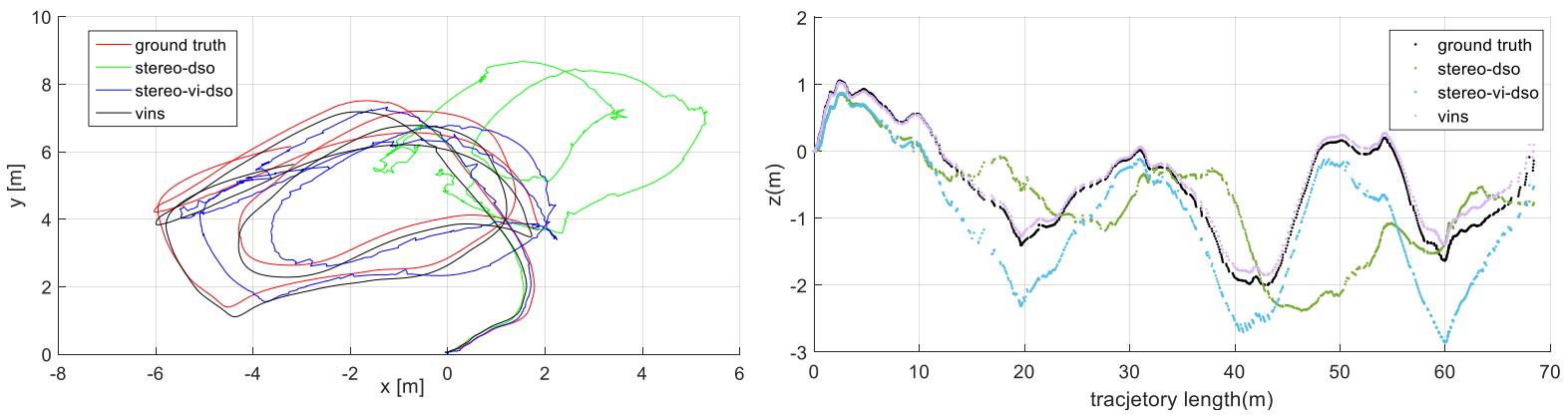


Fig. 8: Trajectory and height estimates in MH3

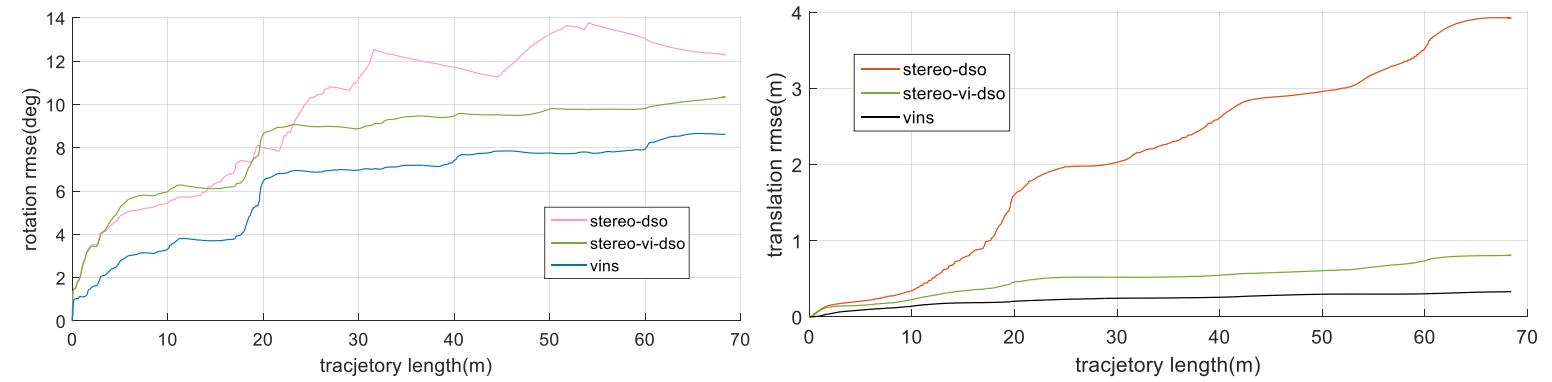


Fig. 9: Rotation and translation rmse in MH3.

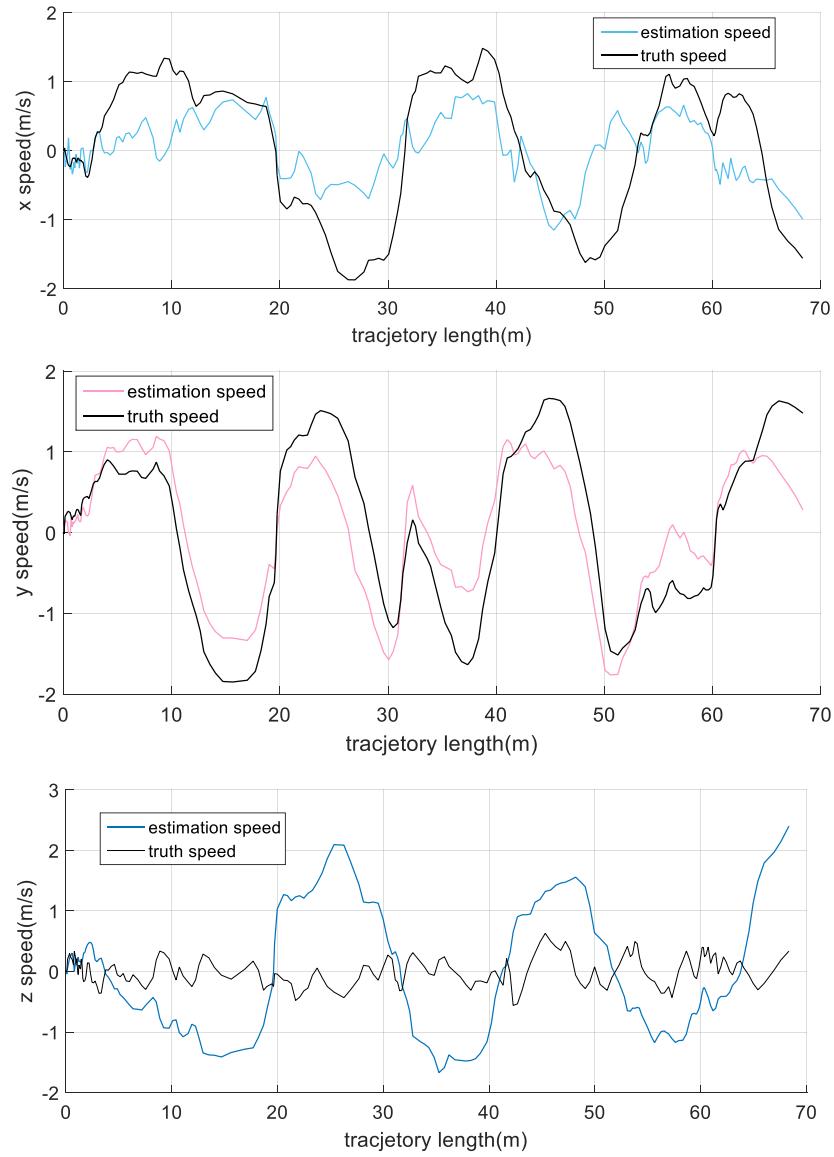


Fig. 10: Velocity estimates in MH3.

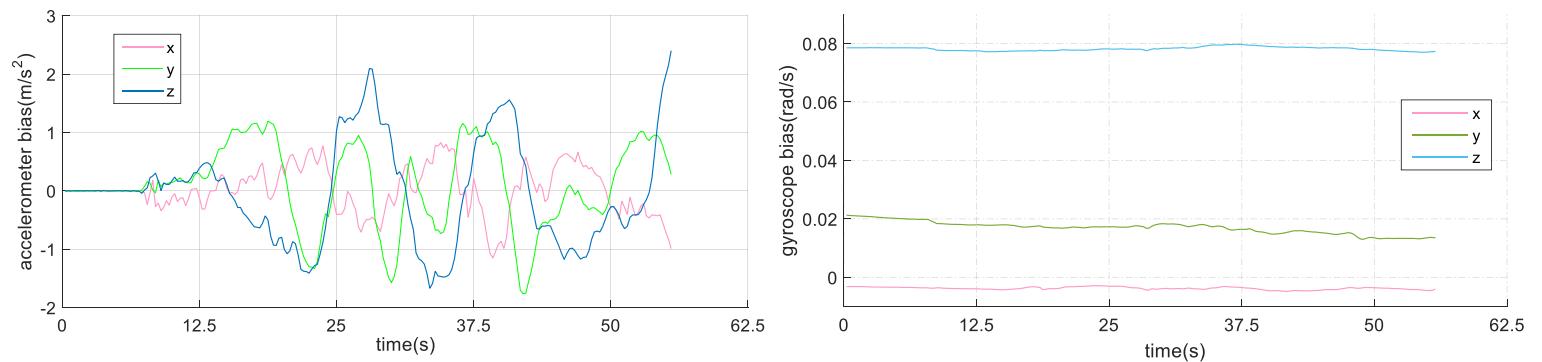


Fig. 11: IMU bias estimates in MH3.

Chapter1 Visual-inertial Preliminaries

In our main paper [IV], The term $J_r(\mathbf{q})$ 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).

$$\begin{aligned}
 \text{Exp}(\mathbf{q}^\wedge) &= T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}_{4 \times 4}; \text{Exp}(\dot{\mathbf{q}}^\wedge) = \begin{bmatrix} \dot{R} & \dot{t} \\ 0^T & 1 \end{bmatrix}_{4 \times 4}; p \in \mathbb{R}^3 \\
 \frac{\partial(Tp)}{\partial \dot{\mathbf{q}}} &= \lim_{\mu \rightarrow 0} \frac{\text{Exp}(\mathbf{q}^\wedge) \text{Exp}(\dot{\mathbf{q}}^\wedge) p + \text{Exp}(\mathbf{q}^\wedge) p}{\dot{\mathbf{q}}} \\
 \frac{1}{\mu} \lim_{\mu \rightarrow 0} \frac{\text{Exp}(\mathbf{q}^\wedge)(I + \dot{\mathbf{q}}^\wedge)p + \text{Exp}(\mathbf{q}^\wedge)p}{\mu R \dot{R}^\wedge p + \mu \dot{t} + t} &= \lim_{\mu \rightarrow 0} \frac{\text{Exp}(\mathbf{q}^\wedge) \dot{\mathbf{q}}^\wedge p}{\mu R \dot{R}^\wedge p + R \dot{t} + t} \\
 &= \lim_{\mu \rightarrow 0} \frac{1}{\mu \frac{\dot{\mathbf{q}}^\wedge}{R p + R \dot{t} + t}} = \lim_{\mu \rightarrow 0} \frac{1}{\dot{\mathbf{q}}} \\
 &= \lim_{\mu \rightarrow 0} \frac{1}{\mu \frac{1}{\dot{\mathbf{q}}}} = \frac{R}{0^T} \frac{i}{0^T} \frac{R p}{\dot{\mathbf{q}}} \quad 4 \times 6
 \end{aligned} \tag{1.1}$$

$$\begin{aligned}
 \frac{\partial(T^{-1}p)}{\partial \dot{\mathbf{q}}} &= \lim_{\mu \rightarrow 0} \frac{(\text{Exp}(\mathbf{q}^\wedge) \text{Exp}(\dot{\mathbf{q}}^\wedge))^{-1} p + \text{Exp}(\mathbf{q}^\wedge) p}{\dot{\mathbf{q}}} \\
 &= \lim_{\mu \rightarrow 0} \frac{(\text{Exp}(\dot{\mathbf{q}}^\wedge))^{-1} (\text{Exp}(\mathbf{q}^\wedge))^{-1} p + \text{Exp}(\mathbf{q}^\wedge) p}{\dot{\mathbf{q}}} \\
 &= \lim_{\mu \rightarrow 0} \frac{\text{Exp}(\dot{\mathbf{q}}^\wedge) \text{Exp}(\mathbf{q}^\wedge) p + \text{Exp}(\mathbf{q}^\wedge) p}{\dot{\mathbf{q}}} \\
 \frac{1}{\mu} \lim_{\mu \rightarrow 0} \frac{(I + \dot{\mathbf{q}}^\wedge) \text{Exp}(\mathbf{q}^\wedge) p + \text{Exp}(\mathbf{q}^\wedge) p}{\mu \dot{R}^\wedge R^{-1} p + \mu \dot{t} + t} &= \lim_{\mu \rightarrow 0} i \frac{\dot{\mathbf{q}}^\wedge \text{Exp}(\mathbf{q}^\wedge) p}{\dot{\mathbf{q}}} \\
 &= \lim_{\mu \rightarrow 0} i \frac{1}{\mu \frac{\dot{R}^\wedge R^{-1} p + \dot{t} + t}{\dot{\mathbf{q}}}} \\
 &= \lim_{\mu \rightarrow 0} i \frac{1}{\mu \frac{1}{\dot{\mathbf{q}}^\wedge (R^{-1} p + R^{-1} t) + \dot{t}}} \\
 &= \lim_{\mu \rightarrow 0} i \frac{1}{\mu \frac{1}{i (R^{-1} p + R^{-1} t)^\top \dot{\mathbf{q}} + \dot{t}}} = \frac{\mu}{0^T} I_3 \frac{(R^{-1} p + R^{-1} t)^\top}{0^T} \quad 4 \times 6
 \end{aligned} \tag{1.2}$$

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 p_C is denoted as (1.3). $\cdot|_K$ are used to denote camera projection functions. The jacobian of I_i^L , $\cdot|_K$ is denoted as (1.3)

$$K = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; K^{-1} = \begin{bmatrix} f_x^{-1} & 0 & -f_x^{-1}c_x & 0 \\ 0 & f_y^{-1} & -f_y^{-1}c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix};$$

$$p = \begin{bmatrix} u^i \\ v^i \\ 1 \end{bmatrix}; p_C = \begin{bmatrix} x \\ y \\ z \end{bmatrix}; d_p = z^{i-1}; p = d_p K p_C = \cdot|_K(p_C) \quad (1.3)$$

$$\frac{\partial(I_i^L(p))}{\partial p} = (g_x; g_y; 0; 0); \frac{\partial p}{\partial p_C} = \frac{\partial \cdot|_K(p_C)}{\partial p_C} = \begin{bmatrix} 0 & f_x z^{i-1} & 0 & i x f_x z^{i-2} & 1 \\ 0 & f_y z^{i-1} & 0 & i y f_y z^{i-2} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & i z^{i-2} & 0 & 0 \end{bmatrix}$$

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)
1: function Time_closest_select(gyro_list; i)
2:   t ~ acc_list[s]: timestamp; i ~ s
3:   while true do
4:     if i >= gyro_list:size then
5:       return gyro_list.back
6:     else
7:       tnow ~ gyro_list[i]:timestamp
8:       tnext ~ gyro_list[i+1]:timestamp
9:       if tnow < t then
10:         if tnext > t then
11:           tfront ~ abs(tnow - t); tback ~ abs(tnext - t)
12:           return tfront > tback?gyro_list[i+1]:gyro_list[i]
13:         else
14:           i = i + 1
15:         end if
16:       else if tnow > t then
17:         i = i - 1
18:       else
19:         return gyro_list[i]
20:       end if
21:     end if
22:   end while
23: 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 $m \times n$. We have 8 error items to define:

$\dot{\phi} \dot{R}_{ij}; \frac{\partial \dot{\phi} \dot{R}_{ij}}{\partial b^g}; \frac{\partial \dot{\phi} \dot{v}_{ij}}{\partial b^a}; \frac{\partial \dot{\phi} \dot{p}_{ij}}{\partial b^a}$ are pure rotation values and aren't related to accelerometer data.

$\dot{\phi} \dot{v}_{ij}; \frac{\partial \dot{\phi} \dot{v}_{ij}}{\partial b^g}; \dot{\phi} \dot{p}_{ij}; \frac{\partial \dot{\phi} \dot{p}_{ij}}{\partial b^g}$ 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 $\dot{\phi} \dot{R}_{ik}; \frac{\partial \dot{\phi} \dot{R}_{ik}}{\partial b^g}$ are presented here in (2.1), (2.2).

$$\dot{\phi} \dot{R}_{ik} = \begin{cases} I_{3x3}; & k = i \\ \prod_{m=i}^{k-1} \exp((\tilde{\omega}_m \cdot b_i^g) \dot{\phi} t); & k > i \end{cases}$$

e.g: $k : 0 ! 44; i = 0$

$$\begin{aligned} \dot{\phi} \dot{R}_{00} &= I_{3x3} & (2.1) \\ \dot{\phi} \dot{R}_{01} &= \exp((\tilde{\omega}_0 \cdot b_0^g) \dot{\phi} t) \\ \dot{\phi} \dot{R}_{02} &= \exp((\tilde{\omega}_0 \cdot b_0^g) \dot{\phi} t) \exp((\tilde{\omega}_1 \cdot b_0^g) \dot{\phi} t) \\ &\vdots \\ \dot{\phi} \dot{R}_{0(44)} &= \exp((\tilde{\omega}_0 \cdot b_0^g) \dot{\phi} t) \exp((\tilde{\omega}_1 \cdot b_0^g) \dot{\phi} t) \dots \exp((\tilde{\omega}_{43} \cdot b_0^g) \dot{\phi} t) \end{aligned}$$

$$\begin{aligned} \frac{\partial \dot{\phi} \dot{R}_{ik}}{\partial b^g} &= \begin{cases} 0_{3x3}; & k = i \\ \sum_{m=i}^{k-1} i \dot{\phi} \dot{R}_{m+1k}^T J_r^m \dot{\phi} t; & k > i \end{cases} \\ &= \begin{cases} 0_{3x3}; & k = i \\ J_r^0 \dot{\phi} t; & k = i+1 \\ \dot{\phi} \dot{R}_{(k-1)k}^T \frac{\partial \dot{\phi} \dot{R}_{i(k-1)}}{\partial b^g} + J_r^{k-1} \dot{\phi} t; & k > i+1 \end{cases} \\ \text{e.g: } i &= 0; \quad k : 0 ! 45 \\ \frac{\partial \dot{\phi} \dot{R}_{00}}{\partial b^g} &= 0_{3x3} \\ \frac{\partial \dot{\phi} \dot{R}_{01}}{\partial b^g} &= \sum_{m=0}^0 \dot{\phi} \dot{R}_{(m+1)1}^T J_r^m \dot{\phi} t = \dot{\phi} \dot{R}_{11}^T J_r^0 \dot{\phi} t = J_r^0 \dot{\phi} t \\ \frac{\partial \dot{\phi} \dot{R}_{02}}{\partial b^g} &= \sum_{m=0}^1 \dot{\phi} \dot{R}_{(m+1)2}^T J_r^m \dot{\phi} t = \dot{\phi} \dot{R}_{12}^T J_r^0 \dot{\phi} t + \dot{\phi} \dot{R}_{22}^T J_r^1 \dot{\phi} t = \dot{\phi} \dot{R}_{12}^T \frac{\partial \dot{\phi} \dot{R}_{01}}{\partial b^g} + J_r^1 \dot{\phi} t \\ \frac{\partial \dot{\phi} \dot{R}_{03}}{\partial b^g} &= \sum_{m=0}^2 \dot{\phi} \dot{R}_{(m+1)2}^T J_r^m \dot{\phi} t = \dot{\phi} \dot{R}_{13}^T J_r^0 \dot{\phi} t + \dot{\phi} \dot{R}_{23}^T J_r^1 \dot{\phi} t + \dot{\phi} \dot{R}_{33}^T J_r^2 \dot{\phi} t \\ &= (\dot{\phi} \dot{R}_{12} \dot{\phi} \dot{R}_{23})^T J_r^0 \dot{\phi} t + \dot{\phi} \dot{R}_{23}^T J_r^1 \dot{\phi} t + J_r^2 \dot{\phi} t \\ &= \dot{\phi} \dot{R}_{23}^T \dot{\phi} \dot{R}_{12}^T J_r^0 \dot{\phi} t + \dot{\phi} \dot{R}_{23}^T J_r^1 \dot{\phi} t + J_r^2 \dot{\phi} t \\ &= \dot{\phi} \dot{R}_{23}^T \frac{\partial \dot{\phi} \dot{R}_{02}}{\partial b^g} + J_r^2 \dot{\phi} t \\ &\vdots \\ \frac{\partial \dot{\phi} \dot{R}_{0(44)}}{\partial b^g} &= \sum_{m=0}^{43} \dot{\phi} \dot{R}_{(m+1)44}^T J_r^m \dot{\phi} t \\ &= \dot{\phi} \dot{R}_{1(44)}^T J_r^0 \dot{\phi} t + \dot{\phi} \dot{R}_{2(44)}^T J_r^1 \dot{\phi} t + \dots + \dot{\phi} \dot{R}_{43(44)}^T J_r^{42} \dot{\phi} t + \dot{\phi} \dot{R}_{44(44)}^T J_r^{43} \dot{\phi} t \\ &= \dot{\phi} \dot{R}_{43(44)}^T \frac{\partial \dot{\phi} \dot{R}_{0(43)}}{\partial b^g} + J_r^{43} \dot{\phi} t \end{aligned} \quad (2.2)$$

$$\begin{aligned} \frac{\partial \dot{\phi} \dot{R}_{0(45)}}{\partial b^g} &= \sum_{m=0}^{44} \dot{\phi} \dot{R}_{(m+1)45}^T J_r^m \dot{\phi} t \\ &= \dot{\phi} \dot{R}_{44(45)}^T \frac{\partial \dot{\phi} \dot{R}_{0(44)}}{\partial b^g} + J_r^{44} \dot{\phi} t \end{aligned}$$

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 Preintegration for IMU

Input: gyro_list; acc_list; m; n; rotate_list

Output: $(\dot{\phi} R_{ij}; \frac{\partial \dot{\phi} R_{ij}}{\partial b^g}, \frac{\partial \dot{\phi} v_{ij}}{\partial b^a}, \frac{\partial \dot{\phi} p_{ij}}{\partial b^a}); (\dot{\phi} v_{ij}; \frac{\partial \dot{\phi} v_{ij}}{\partial b^g}; \dot{\phi} p_{ij}; \frac{\partial \dot{\phi} p_{ij}}{\partial b^g}); \$_{ij}$

1: function IMU_Preintegration(gyro_list; acc_list; m; n; rotate_list)

2: for all gyro_list[i]; i : 0 ! m do

3: last_r \leftarrow rotate_list[i; 1]

4: rot:tim estam p \leftarrow gyro_list[i]:tim estam p

5: rot!: \tilde{A} gyro_list[i]:! i b_i^{1a}

6: rot: $\dot{\phi} R_{ik} \leftarrow$ last_r: $\dot{\phi} R_{ik}$ Exp(rot!: $\otimes t$)

7: rot: $\dot{\phi} R_{(k_i+1)k} \leftarrow$ Exp(rot!: $\otimes t$)

8: rot: $\frac{\partial \dot{\phi} R_{ik}}{\partial b^g} \leftarrow \dot{\phi} R_{(k_i+1)k}^T \otimes$ last_r: $\frac{\partial \dot{\phi} R_{ik}}{\partial b^g} i J_r(\text{rot}:! \otimes t) \otimes t$

9: rot: $\frac{\partial \dot{\phi} v_{ik}}{\partial b^a} \leftarrow \text{last}_r: \frac{\partial \dot{\phi} v_{ik}}{\partial b^a} i \text{last}_r: \dot{\phi} R_{ik} \otimes t$

10: rot: $\frac{\partial \dot{\phi} p_{ik}}{\partial b^a} \leftarrow \text{last}_r: \frac{\partial \dot{\phi} p_{ik}}{\partial b^a} + \text{last}_r: \frac{\partial \dot{\phi} v_{ik}}{\partial b^a} \otimes t + \frac{1}{2} \text{last}_r: \dot{\phi} R_{ik} \otimes t^2$

11: rotate_list.push(rot)

12: end for

13: $\dot{\phi} R_{ij} = \text{rotate_list: end} : \dot{\phi} R_{ik}$

14: $\frac{\partial \dot{\phi} R_{ij}}{\partial b^g} = \text{rotate_list: end} : \frac{\partial \dot{\phi} R_{ik}}{\partial b^g}$

15: $\frac{\partial \dot{\phi} v_{ij}}{\partial b^a} = \text{rotate_list: end} : \frac{\partial \dot{\phi} v_{ik}}{\partial b^a}$

16: $\frac{\partial \dot{\phi} p_{ij}}{\partial b^a} = \text{rotate_list: end} : \frac{\partial \dot{\phi} p_{ik}}{\partial b^a}$

17: for all acc_list[i]; i : 0 ! n do

18: cls_r \leftarrow tim_e_closest_select(rotate_list; acc_list[i])

19: acc \leftarrow acc_list[i] $i b_i^{1a}$

20: $\dot{\phi} v_{ij}^+ = \text{cls}_r: \dot{\phi} R_{ik} \otimes \text{acc} \otimes t$

21: $\frac{\partial \dot{\phi} v_{ij}}{\partial b^g} i = \text{cls}_r: \dot{\phi} R_{ik} \otimes \text{acc} \otimes \text{cls}_r: \frac{\partial \dot{\phi} R_{ik}}{\partial b^g} \otimes t$

22: $\dot{\phi} p_{ij}^+ = \dot{\phi} v_{ij} \otimes t + \frac{1}{2} \text{cls}_r: \dot{\phi} R_{ik} \otimes \text{acc} \otimes t^2$

23: $\frac{\partial \dot{\phi} p_{ij}}{\partial b^g} _0^+ = \text{cls}_r: \frac{\partial \dot{\phi} v_{ik}}{\partial b^g} \dot{\phi} t i + \frac{1}{2} \text{cls}_r: \dot{\phi} R_{ik} \otimes \text{acc} \otimes \text{cls}_r: \frac{\partial \dot{\phi} R_{ik}}{\partial b^g} \otimes t^2$

24: $A = @ i \text{cls}_r: \dot{\phi} R_{ik} \otimes \text{acc} \otimes t I 0 A$
 $0 i \frac{1}{2} \text{cls}_r: \dot{\phi} R_{ik} \otimes \text{acc} \otimes t^2 \dot{\phi} t I I 1$
 $J_r(\text{rot}:! \otimes t) \otimes t 0$

25: B = @ 0 $\text{cls}_r: \dot{\phi} R_{ik} \otimes t A$
 $0 \frac{1}{2} \text{cls}_r: \dot{\phi} R_{ik} \otimes t^2$

26: $\$_{ij} = A \otimes \$_{ij} \otimes A^T + B \otimes \$ \otimes B^T$

27: end for

28: end function

2.3 Jacobian derivation

The derivation of the Jacobians of $r_{\epsilon R_{ij}}, r_{\epsilon v_{ij}}, r_{\epsilon p_{ij}}$ likes (2.3), (2.4), (2.5).

$$\begin{aligned}
 \frac{\partial r_{\epsilon R_{ij}}}{\partial \dot{p}_i} &= 0 \\
 \frac{\partial r_{\epsilon R_{ij}}}{\partial \dot{p}_i} &= i J_r^{i-1}(r_{\epsilon R_{ij}}) R_j^T R_i \\
 \frac{\partial r_{\epsilon R_{ij}}}{\partial \dot{x}_i} &= 0 \\
 \frac{\partial r_{\epsilon R_{ij}}}{\partial \dot{p}_j} &= 0 \\
 \frac{\partial r_{\epsilon R_{ij}}}{\partial \dot{p}_j} &= J_r^{i-1}(r_{\epsilon R_{ij}}) \\
 \frac{\partial r_{\epsilon R_{ij}}}{\partial \dot{x}_j} &= 0 \\
 \frac{\partial r_{\epsilon R_{ij}}}{\partial b_i^a} &= 0 \\
 \frac{\partial r_{\epsilon R_{ij}}}{\partial b_i^g} &= i J_r^{i-1}(r_{\epsilon R_{ij}}) \text{Exp}(r_{\epsilon R_{ij}})^T J_r \left(\frac{\partial \dot{p}_{R_{ij}}}{\partial b^g} \dot{p}_i^g \right) \frac{\partial \dot{p}_{R_{ij}}}{\partial b^g} \tag{2.3}
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial r_{\epsilon v_{ij}}}{\partial \dot{p}_i} &= 0 \\
 \frac{\partial r_{\epsilon v_{ij}}}{\partial \dot{p}_i} &= (R_i^T (v_j i - v_i j - g \dot{p} t_{ij})) \wedge \\
 \frac{\partial r_{\epsilon v_{ij}}}{\partial \dot{x}_i} &= i R_i^T \\
 \frac{\partial r_{\epsilon v_{ij}}}{\partial \dot{p}_j} &= 0 \\
 \frac{\partial r_{\epsilon v_{ij}}}{\partial \dot{p}_j} &= 0 \\
 \frac{\partial r_{\epsilon v_{ij}}}{\partial \dot{x}_j} &= R_i^T \\
 \frac{\partial r_{\epsilon v_{ij}}}{\partial b_i^a} &= i \frac{\partial \dot{p}_{V_{ij}}}{\partial b^a} \\
 \frac{\partial r_{\epsilon v_{ij}}}{\partial b_i^g} &= i \frac{\partial \dot{p}_{V_{ij}}}{\partial b^g} \tag{2.4}
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial r_{\epsilon p_{ij}}}{\partial \dot{p}_i} &= i I \\
 \frac{\partial r_{\epsilon p_{ij}}}{\partial \dot{p}_i} &= (R_i^T (p_j i - p_i j - v_i \dot{p} t_{ij} i - \frac{1}{2} g \dot{p} t_{ij}^2)) \wedge \\
 \frac{\partial r_{\epsilon p_{ij}}}{\partial \dot{x}_i} &= i R_i^T \dot{p} t_{ij} \\
 \frac{\partial r_{\epsilon p_{ij}}}{\partial \dot{p}_j} &= R_i^T R_j \\
 \frac{\partial r_{\epsilon p_{ij}}}{\partial \dot{x}_j} &= 0 \\
 \frac{\partial r_{\epsilon p_{ij}}}{\partial b_i^a} &= 0 \\
 \frac{\partial r_{\epsilon p_{ij}}}{\partial b_i^g} &= i \frac{\partial \dot{p}_{P_{ij}}}{\partial b^a} \\
 \frac{\partial r_{\epsilon p_{ij}}}{\partial b_i^g} &= i \frac{\partial \dot{p}_{P_{ij}}}{\partial b^g} \tag{2.5}
 \end{aligned}$$

Chapter3 Photo Error Factors

3.1 Construction residual errors

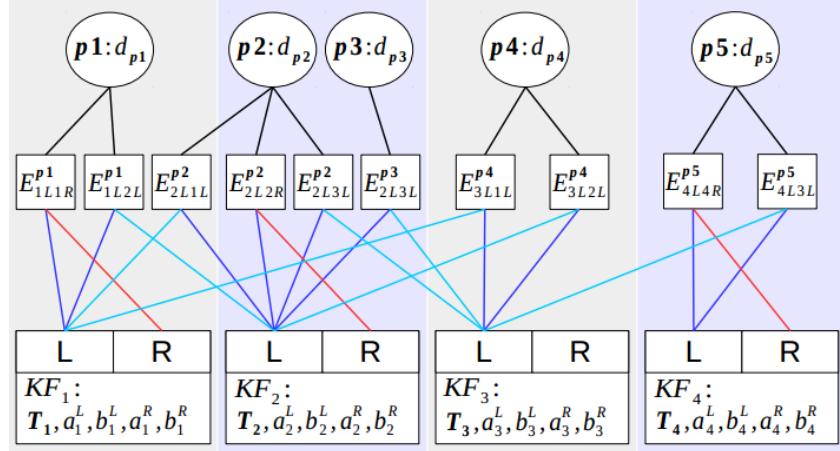


Fig.1

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).

$$\hat{A} = \begin{matrix} (\hat{A}_1; \dots; \hat{A}_4)^T \\ (p_1^T; \dots; p_4^T)^T \\ (v_1^T; \dots; v_4^T)^T \\ (b_1^T; \dots; b_4^T)^T \\ (d_{p_1}; \dots; d_{p_5})^T \\ (a_1^L; a_1^R; b_1^L; b_1^R)^T \\ \vdots \\ (a_4^L; a_4^R; b_4^L; b_4^R)^T \end{matrix} \quad \begin{matrix} 1 \\ C \\ C \\ C \\ C \\ C \\ A \\ A \end{matrix} \quad \begin{matrix} 2 \\ R^{81} \\ ; \\ ; \\ ; \\ ; \\ \end{matrix} \quad \begin{matrix} \hat{A}_i = \text{Log}(R_i); \\ \Rightarrow_i = (\hat{A}_i^T; p_i^T)^T \end{matrix} \quad (3.1)$$

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

$$\begin{aligned} E(\hat{A}) &= E_{1L2L}^{p1} + E_{2L1L}^{p2} + E_{2L3L}^{p2} + E_{2L3L}^{p3} + E_{3L1L}^{p4} + E_{3L2L}^{p4} + E_{4L3L}^{p5} \\ &\quad + E_{1L1R}^{p1} + E_{2L2R}^{p2} + E_{4L4R}^{p5} \\ &= E_d(\hat{A}) + E_s(\hat{A}) \\ E_d(\hat{A}) &= \begin{matrix} 0 & 1 & 0 & 0 & \dots & 0 & 1 & 0 & 1 \\ (r_{p_1}^d)_{12} & 0 & w_{p_1} & 0 & \dots & 0 & (r_{p_1}^d)_{12} & 1 \\ (r_{p_1}^d)_{21} & 0 & w_{p_1} & 0 & \dots & 0 & (r_{p_1}^d)_{21} & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & w_{p_5} & 0 & (r_{p_5}^d)_{43} & 1 \end{matrix} = (r^d)^T W^d r^d \quad (3.2) \\ E_s(\hat{A}) &= \begin{matrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ r_{p_1}^s & 0 & w_{p_1} & 0 & 0 & 0 & r_{p_1}^s & 1 \\ r_{p_2}^s & 0 & w_{p_2} & 0 & 0 & 0 & r_{p_2}^s & 1 \\ \vdots & \vdots \\ 0 & 0 & 0 & , & w_{p_5} & 0 & r_{p_5}^s & 1 \end{matrix} = (r^s)^T W^s r^s \end{aligned}$$

We first note that $(v_1^T ; \dots ; v_4^T)^T ; (b_1^T ; \dots ; b_4^T)^T$ do not appear in the expression of $E_d(\hat{A}) ; E_s(\hat{A})$, hence the corresponding Jacobians are zero, we omit them for writing simply. The remaining Jacobians can be computed as follows (3.3):

$$J_s = \begin{array}{c} \begin{array}{cccccc} \frac{\partial r_p^s}{\partial p_1} & \dots & \frac{\partial r_p^s}{\partial p_4} & \dots & \frac{\partial r_p^s}{\partial p_5} & \dots & \frac{\partial r_p^s}{\partial a_1^L} \\ \frac{\partial r_p^s}{\partial d_p_1} & \dots & \frac{\partial r_p^s}{\partial d_p_4} & \dots & \frac{\partial r_p^s}{\partial d_p_5} & \dots & \frac{\partial r_p^s}{\partial b_4^R} \\ \frac{\partial r_p^s}{\partial p_2} & \dots & \frac{\partial r_p^s}{\partial p_4} & \dots & \frac{\partial r_p^s}{\partial p_5} & \dots & \frac{\partial r_p^s}{\partial a_1^L} \\ \frac{\partial r_p^s}{\partial d_p_1} & \dots & \frac{\partial r_p^s}{\partial d_p_4} & \dots & \frac{\partial r_p^s}{\partial d_p_5} & \dots & \frac{\partial r_p^s}{\partial b_4^R} \end{array} \end{array} \quad (3.3)$$

$$J_d = \begin{array}{c} \begin{array}{cccccc} \frac{\partial (r_p^d)_{12}}{\partial p_1} & \dots & \frac{\partial (r_p^d)_{12}}{\partial p_4} & \dots & \frac{\partial (r_p^d)_{12}}{\partial p_5} & \dots & \frac{\partial (r_p^d)_{12}}{\partial a_1^L} \\ \frac{\partial (r_p^d)_{21}}{\partial p_1} & \dots & \frac{\partial (r_p^d)_{21}}{\partial p_4} & \dots & \frac{\partial (r_p^d)_{21}}{\partial p_5} & \dots & \frac{\partial (r_p^d)_{21}}{\partial b_4^R} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \frac{\partial (r_p^d)_{43}}{\partial p_1} & \dots & \frac{\partial (r_p^d)_{43}}{\partial p_4} & \dots & \frac{\partial (r_p^d)_{43}}{\partial p_5} & \dots & \frac{\partial (r_p^d)_{43}}{\partial b_4^R} \end{array} \end{array} \quad (3.3)$$

3E 49 1
7E 49

Iteration \hat{A} can be calculated by (3.4):

$$(J_s^T, W^s J_s + J_d^T W^d J_d) \hat{A} = i (J_s^T, W^s r^s + J_d^T W^d r^d) \quad (3.4)$$

$$J_s \in \mathbb{R}^{3E 49}; W^s \in \mathbb{R}^{3E 3}; J_d \in \mathbb{R}^{7E 49}; W^d \in \mathbb{R}^{7E 7}$$

3.2 Jacobian derivation

3.2.1 Dynamic Parameter

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

$$\frac{\partial (r_p^d)_{ij}}{\partial \underline{p}_m} = \frac{\partial (r_p^d)_{ij}}{\partial \underline{p}_n} = 0^T; \text{ so } \frac{\partial (r_p^d)_{12}}{\partial \underline{p}_3} = \frac{\partial (r_p^d)_{12}}{\partial \underline{p}_4} = \dots = 0^T; \quad (3.6)$$

Otherwise, assuming the hostframe of 2D image coordinate point p is i^L , and corresponding homogeneous 3D camera coordinate point is p_C in (3.7), body coordinate is $p_B = T_{BC}p_C$. We transform p_C from frame i^L to j^L by $p_B^0 = T_j^{-1}T_i p_B$, then transform p_B^0 to camera coordinate point $p_C^0 = T_{BC}^{-1}p_B^0$. At last, p_C^0 is projected to 2D image coordinate point with p^0 .

$$p_C = \begin{array}{c} \begin{array}{c} f_x^{i^{-1}}(d_p^{i^L})^{i^{-1}}(u^i; c_x)^1 \\ f_y^{i^{-1}}(d_p^{i^L})^{i^{-1}}(v^i; c_y)^1 \\ (d_p^{i^L})^{i^{-1}} \end{array} \end{array} ; p_C^0 = \begin{array}{c} \begin{array}{c} x^0 \\ y^0 \\ z^0 \end{array} \end{array} \quad (3.7)$$

$$p^0 = d_p^{j^L} K (T_{BC}^{-1} T_j^{-1} T_i ((d_p^{i^L})^{i^{-1}} T_{BC} K^{-1} p)) = d_p^{j^L} K p_C^0$$

3.2.1.1 Jacobian of Affine Brightness Parameters

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

$$\begin{aligned}
 (r_p^d)_{ij} &= I_j^L(p^0) i \ b_j^L \ i \ \frac{e^{a_j^L}}{e^{a_i^L}} (I_i^L(p) i \ b_i^L) \\
 \frac{\partial (r_p^d)_{ij}}{\partial a_i^L} &= \frac{e^{a_j^L}}{e^{a_i^L}} (I_i^L(p) i \ b_i^L); \frac{\partial (r_p^d)_{ij}}{\partial a_j^L} = i \ \frac{e^{a_j^L}}{e^{a_i^L}} (I_i^L(p) i \ b_i^L) \\
 \frac{\partial (r_p^d)_{ij}}{\partial b_i^L} &= \frac{e^{a_j^L}}{e^{a_i^L}}; \quad \frac{\partial (r_p^d)_{ij}}{\partial b_j^L} = i \ 1
 \end{aligned} \tag{3.8}$$

3.2.1.2 Right Jacobian of Pose

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

$$\begin{aligned}
 (r_p^d)_{ij} &= I_j^L(p^0) i \ b_j^L \ i \ \frac{e^{a_j^L}}{e^{a_i^L}} (I_i^L(p) i \ b_i^L) \\
 \frac{\partial (r_p^d)_{ij}}{\partial \mathbf{v}_i} &= \frac{\partial (I_j^L(p^0))}{\partial p^0} \frac{\partial p^0}{\partial p_C^0} \frac{\partial p_C^0}{\partial \mathbf{v}_i} \\
 \frac{\partial (I_j^L(p^0))}{\partial p^0} &= (g_x^0; g_y^0; 0; 0)^T \\
 \frac{\partial p^0}{\partial p_C^0} &= \begin{pmatrix} 0 & f_x(z^0)^{i-1} & 0 & i x^0 f_x(z^0)^{i-2} & 0 \\ 0 & 0 & f_y(z^0)^{i-1} & i y^0 f_y(z^0)^{i-2} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & (z^0)^{i-2} & 0 & 0 \end{pmatrix} \\
 \frac{\partial p_C^0}{\partial \mathbf{v}_i} &= \frac{\partial (T_{BC}^{-1} T_j^{-1} T_i p_B)}{\mu \partial \mathbf{v}_i} = T_{BC}^{-1} T_j^{-1} \frac{\partial (T_i p_B)}{\partial \mathbf{v}_i} \\
 &= T_{BC}^{-1} T_j^{-1} R_i \ i \ R_i^{-1} p_B
 \end{aligned} \tag{3.9}$$

According to (1.2), the jacobian of \mathbf{v}_j is enclosed in (3.10):

$$\begin{aligned}
 T_i p_B &\stackrel{.}{=} v_i p_B \\
 \frac{\partial p_C^0}{\partial \mathbf{v}_j} &= \frac{\partial (T_{BC}^{-1} T_j^{-1} T_i p_B)}{\mu \partial \mathbf{v}_j} = T_{BC}^{-1} \frac{\partial (T_j^{-1} v_i p_B)}{\partial \mathbf{v}_j} \\
 &= T_{BC}^{-1} \ i \ I_3 \ (R_j^{-1} v_i p_B \ i \ R_j^{-1} t_j)
 \end{aligned} \tag{3.10}$$

3.2.1.3 Jacobian of inverse Depth

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

$$\begin{aligned}
 p^0 &= d_p^{j^L} K (T_{BC}^{-1} T_j^{-1} T_i ((d_p^{i^L})^{i^{-1}} T_{BC} K^{i^{-1}} p)) \\
 &= d_p^{j^L} K (T_{BC}^{-1} T_j^{-1} T_i T_{BC}) p_C \\
 T_{BC}^{-1} T_j^{-1} T_i T_{BC} &\stackrel{:=}{=} T^{\tilde{A}} ; p_C^0 \stackrel{:=}{=} T^{\tilde{A}} p_C
 \end{aligned}$$

$$\left(\frac{\partial (r_p^d)_{ij}}{\partial d_p^{i^L}} \right) = \frac{\partial (I_j^L(p^0))}{\partial d_p^{i^L}} = \frac{\partial (I_j^L(p^0))}{\partial p^0} \frac{\partial p^0}{\partial p_C^0} \frac{\partial p_C^0}{\partial d_p^{i^L}} \quad (3.11)$$

$$\frac{\partial p_C^0}{\partial d_p^{i^L}} = T^{\tilde{A}} \frac{\partial p_C}{\partial d_p^{i^L}} = T^{\tilde{A}} K^{i^{-1}} \frac{r ((d_p^{i^L})^{i^{-1}} p)}{r d_p^{i^L}} = T^{\tilde{A}} K^{i^{-1}} \begin{bmatrix} 0 & u^{i=(d_p^{i^L})^2} \\ v^{i=(d_p^{i^L})^2} & 0 \\ l^{i=(d_p^{i^L})^2} & 0 \end{bmatrix}$$

3.2.2 Static Parameter

Firstly, $\partial r_p^s / \partial j$ do not appear in the expression of r_p^s as (3.12), the corresponding jacobians are zero.

$$r_p^s = I_i^R(p^0) \mid b_i^R \mid \frac{e^{a_i^R}}{e^{a_i^L}} (I_i^L(p) \mid b_i^L) \quad (3.12)$$

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 $d_p^{i^L} \neq d_p^{i^R}$, and T_{RL} is only related to baseline of stereo cameras. Left frame i^L pixel p is projected to right frame i^R with p^0 as (3.13):

$$\begin{aligned} p &= \begin{bmatrix} u^i \\ v^i \\ 1 \end{bmatrix}; p_C = \begin{bmatrix} x \\ y \\ z \end{bmatrix}; d_p^{i^L} = z^{i^L}; p_C = (d_p^{i^L})^{i^L} K^{-1} p \\ &= \begin{bmatrix} 0 & d_p^{i^L} & 1 \\ f_x^{i^L} (d_p^{i^L})^{i^L} (u^i - c_x) & 1 & 0 \\ f_y^{i^L} (d_p^{i^L})^{i^L} (v^i - c_y) & 0 & 1 \\ (d_p^{i^L})^{i^L} & 0 & 0 \end{bmatrix}; T_{RL} = \begin{bmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ p^0 &= d_p^{i^R} K (T_{RL} p_C) \\ &= \begin{bmatrix} 0 & f_x & 0 & c_x & 0 & 1 & 0 \\ 0 & f_y & 0 & c_y & 0 & f_y^{i^L} (d_p^{i^L})^{i^L} (v^i - c_y) & 1 \\ 0 & 0 & 1 & 0 & 0 & (d_p^{i^L})^{i^L} & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & d_p^{i^L} \end{bmatrix} = \begin{bmatrix} 0 & u^i + t_1 f_x d_p^{i^L} & 1 \\ v^i & 1 & 1 \\ d_p^{i^L} & 1 & 1 \end{bmatrix} \\ \frac{\partial r_p^s}{\partial d_p^{i^L}} &= \frac{\partial (I_i^R(p^0)) \mid e^{a_i^R}}{\partial d_p^{i^L}} (I_i^L(p)) = \left(\frac{\partial (I_i^R(p^0))}{\partial p^0} \mid \frac{e^{a_i^R}}{e^{a_i^L}} \frac{\partial (I_i^L(p))}{\partial p^0} \right) \frac{\partial p^0}{\partial d_p^{i^L}} \\ &= [(g_x^{i^R}; g_y^{i^R}; 0; 0) \mid 0^T] \begin{bmatrix} 0 & 1 \\ t_1 f_x & 1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} = g_x^{i^R} t_1 f_x \end{aligned} \quad (3.13)$$

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

$$\begin{aligned} \frac{\partial (r_p^s)_{ij}}{\partial a_i^L} &= \frac{e^{a_i^R}}{e^{a_i^L}} (I_i^L(p) \mid b_i^L); \frac{\partial (r_p^s)_{ij}}{\partial a_i^R} = \mid \frac{e^{a_i^R}}{e^{a_i^L}} (I_i^L(p) \mid b_i^L) \\ \frac{\partial (r_p^s)_{ij}}{\partial b_i^L} &= \frac{e^{a_i^R}}{e^{a_i^L}}; \quad \frac{\partial (r_p^s)_{ij}}{\partial b_i^R} = \mid 1 \end{aligned} \quad (3.14)$$

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.