

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

Ziqiang Wang

March, 2019

**News: We present evaluation results and**  
**[video](#) on EuRoC dataset.**

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.....	8
Chapter2 IMU Error Factors.....	10
2.1 Time-closest measurements selection strategy.....	10
2.2 Errors and covariance calculation pseudo code .....	10
2.3 Jacobian derivation .....	13
Chapter3 Photo Error Factors.....	14
3.1 Construction residual errors .....	14
3.2 Jacobian derivation .....	15
3.2.1 Dynamic Parameter.....	15
3.2.2 Static Parameter .....	18
References.....	19

## 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 helicopter 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

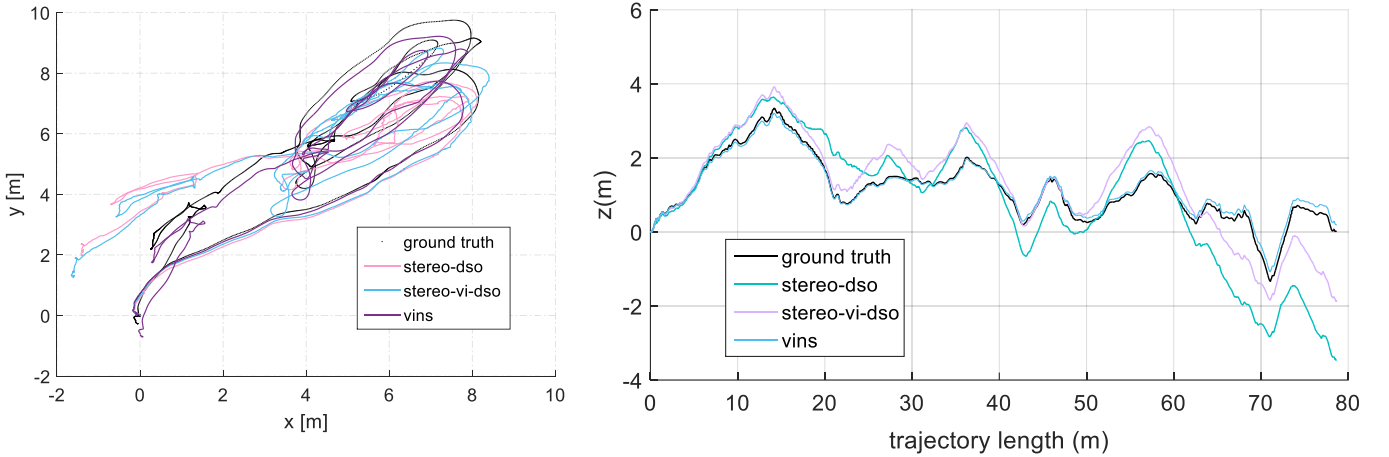


Fig. 1: Trajectory and height estimates in MH1.

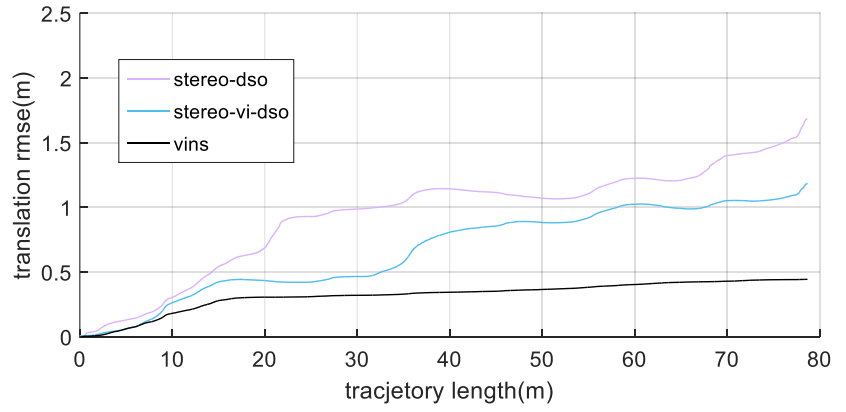
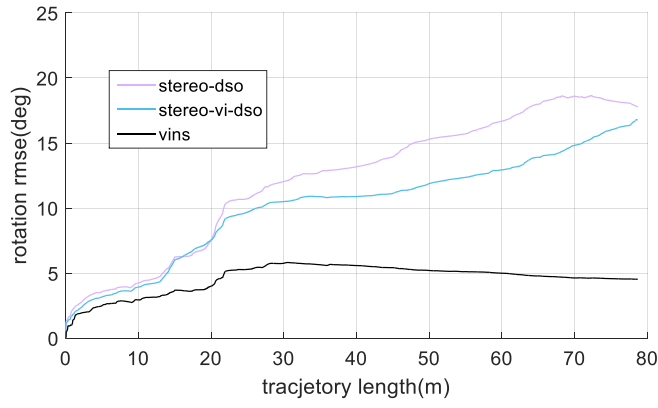


Fig. 2: Rotation and translation rmse in MH1. Rmse increases as distance travelled increases.

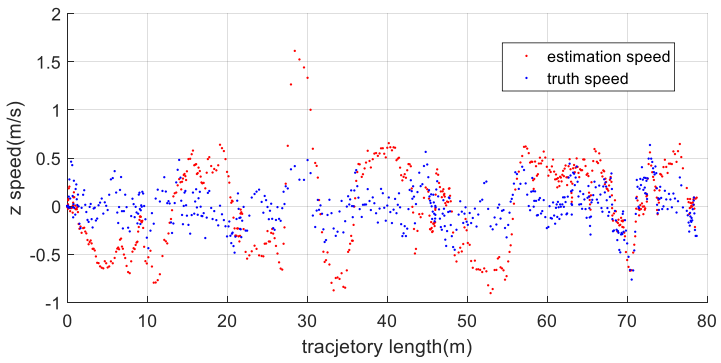
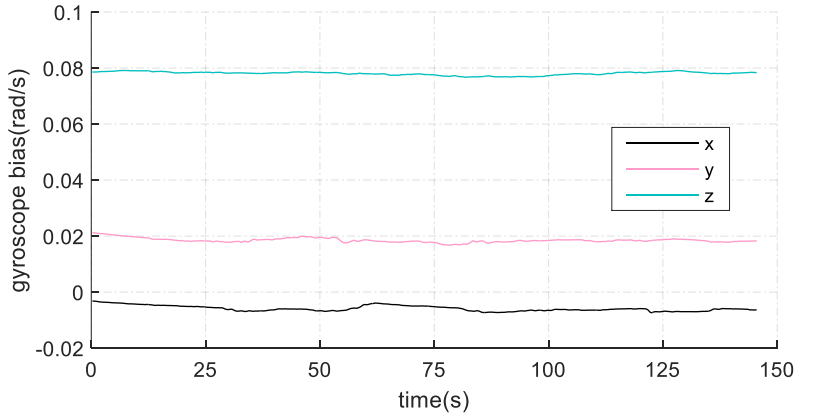
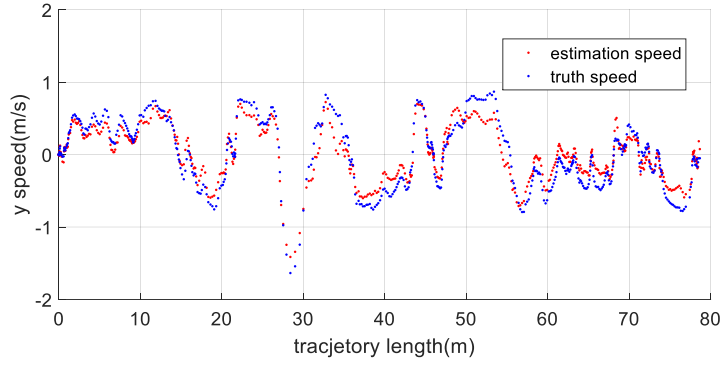
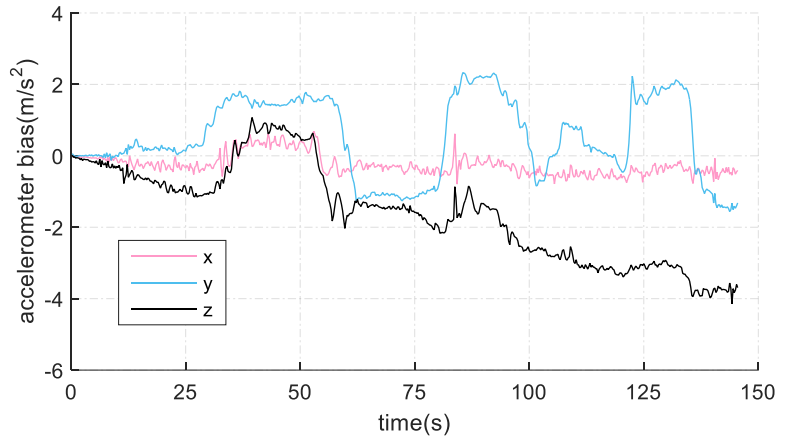
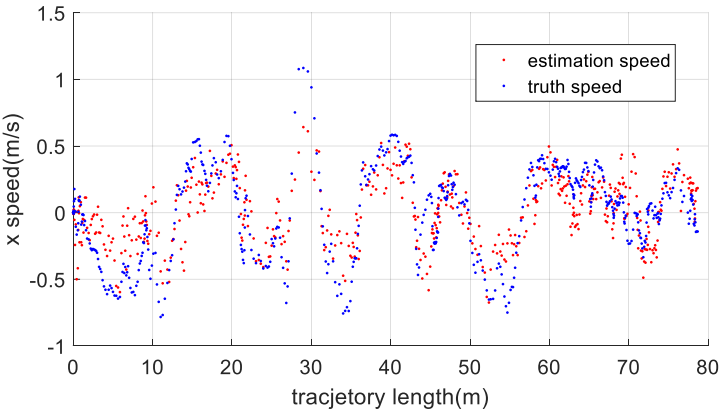


Fig. 3: Velocity and IMU bias estimates in MH1.

It can be seen that the estimated velocity change is similar to the truth values and gyro bias converged to stable values.

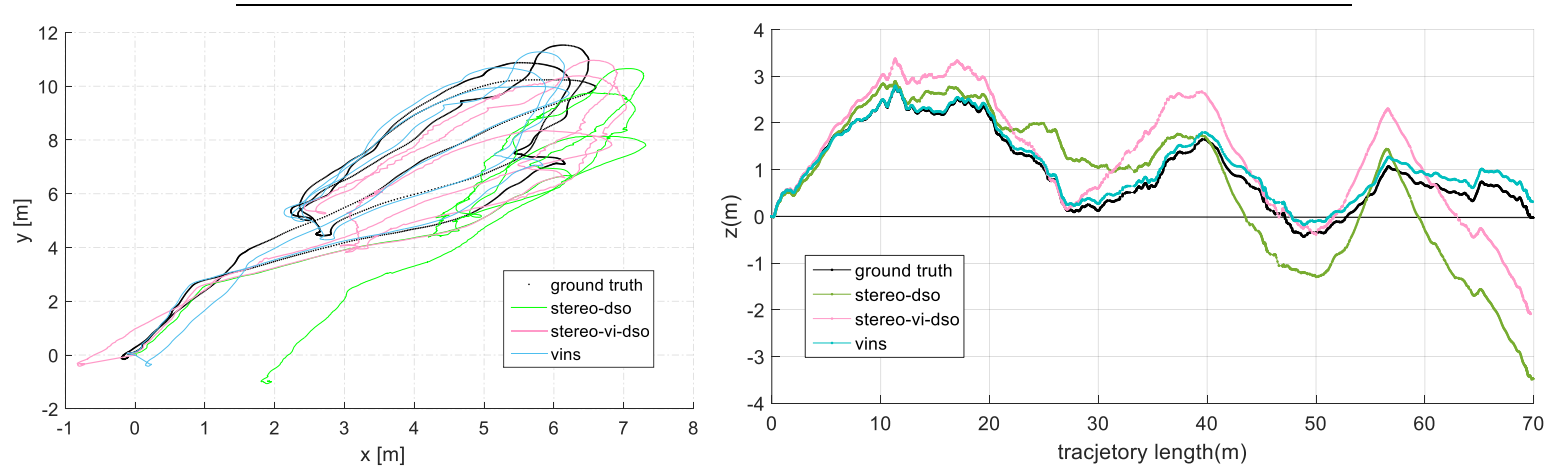


Fig. 4: Trajectory and height estimates in MH2.

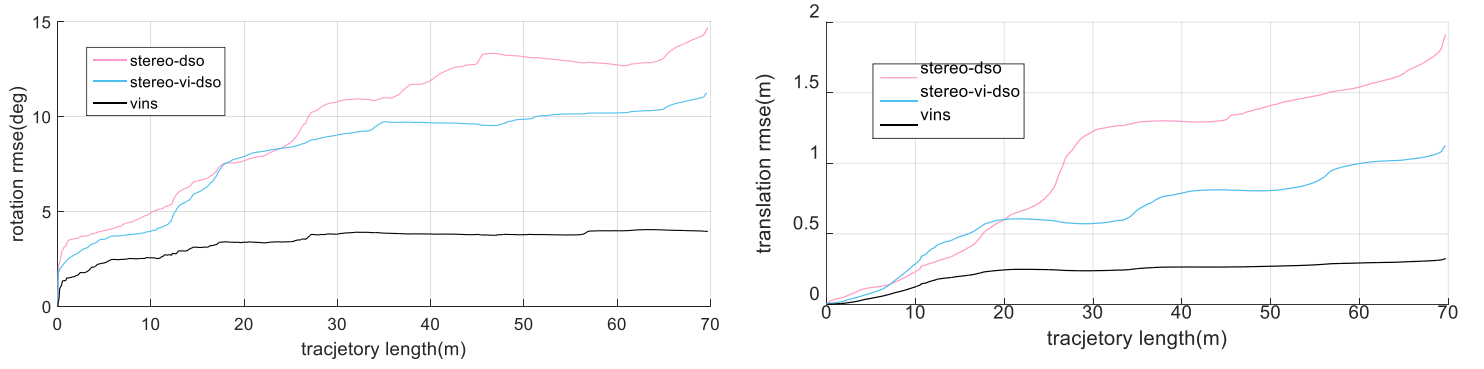


Fig. 5: Rotation and translation rmse in MH2.

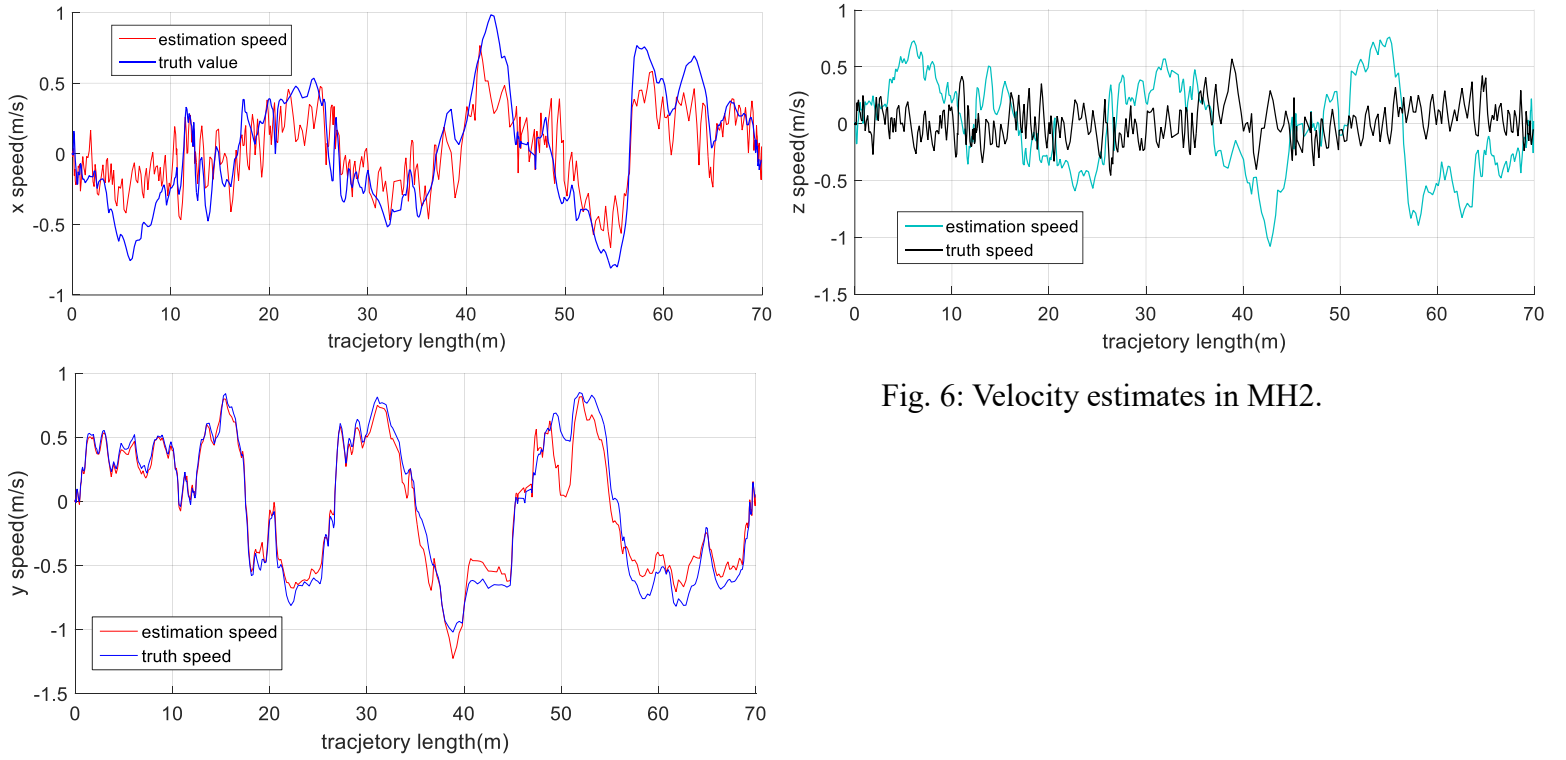


Fig. 6: Velocity estimates in MH2.

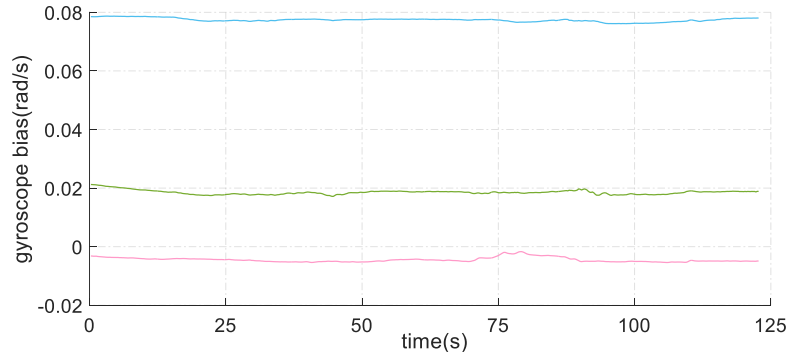
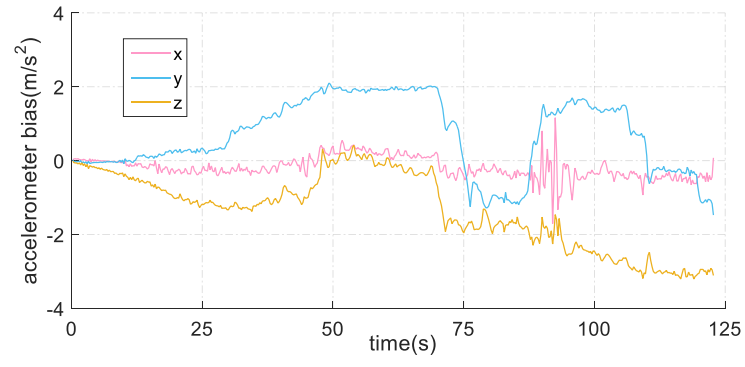


Fig. 7: IMU bias estimates in MH2.

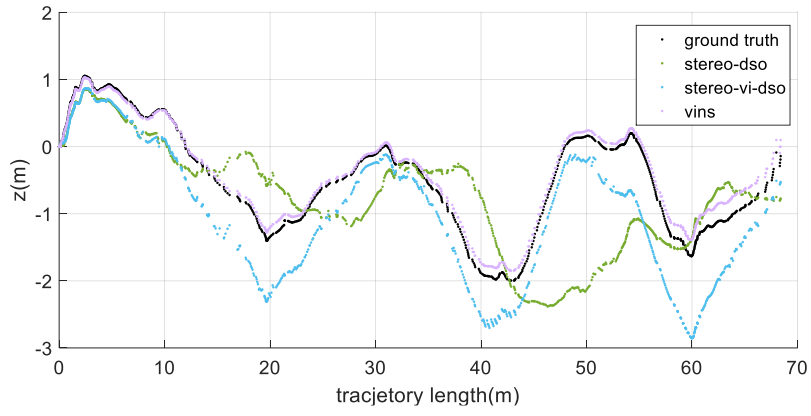
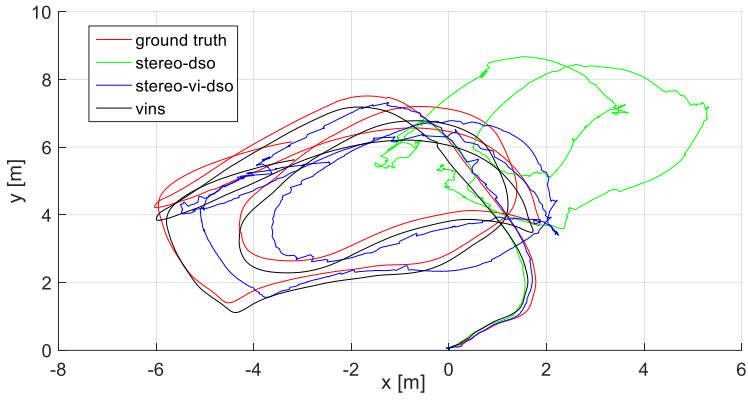


Fig. 8: Trajectory and height estimates in MH3

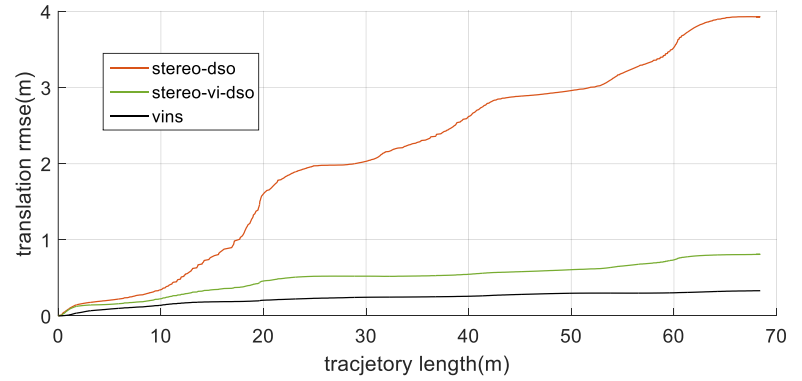
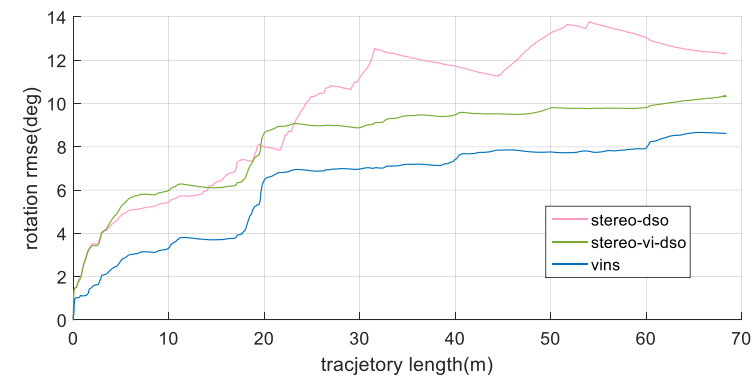


Fig. 9: Rotation and translation rmse in MH3.

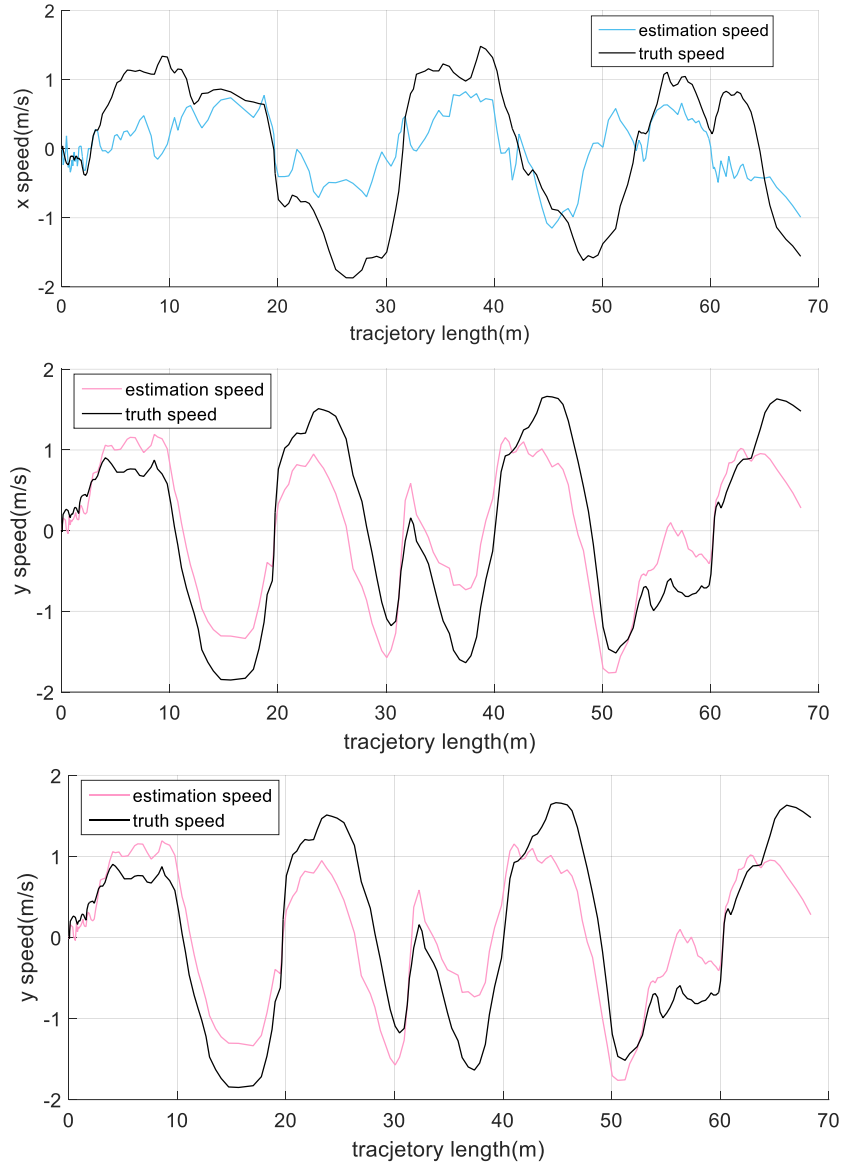


Fig. 10: Velocity estimates in MH3.

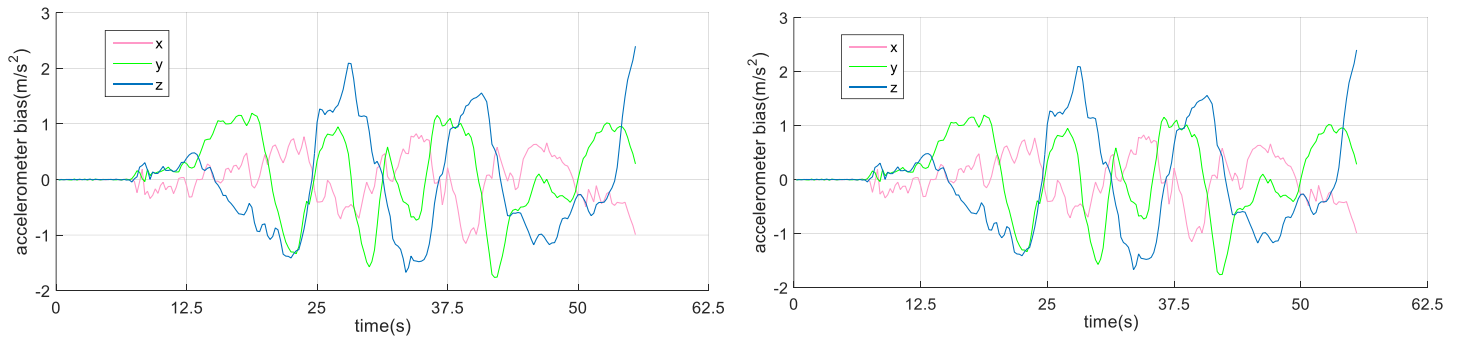


Fig. 11: IMU bias estimates in MH3.

---

## Chapter1 Visual-inertial Preliminaries

In our main paper [IV], The term  $J_r(\xi)$  is the right Jacobian of  $SE(3)$  can be calculated by (1.1) and (1.2) .

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

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

Homogeneous camera calibration matrices are denoted by  $\mathbf{K}$  as (1.3). and homogeneous 2D image coordinate point  $\mathbf{p}$  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  $\mathbf{p}_c$  is denoted as (1.3).  $\Pi_{\mathbf{K}}$  are used to denote camera projection functions. The jacobian of  $\mathbf{I}_i^L$ ,  $\Pi_{\mathbf{K}}$  is denoted as (1.3)



---


$$\begin{aligned}
\mathbf{K} &= \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{K}^{-1} = \begin{pmatrix} f_x^{-1} & 0 & -f_x^{-1}c_x & 0 \\ 0 & f_y^{-1} & -f_y^{-1}c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \\
\mathbf{p} &= \begin{pmatrix} u^i \\ v^i \\ 1 \\ d_{\mathbf{p}} \end{pmatrix}, \mathbf{p}_{\mathbf{c}} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, d_{\mathbf{p}} = z^{-1}, \mathbf{p} = d_{\mathbf{p}} \mathbf{K} \mathbf{p}_{\mathbf{c}} = \Pi_{\mathbf{K}}(\mathbf{p}_{\mathbf{c}}) \\
\frac{\partial(\mathbb{I}_i^L(\mathbf{p}))}{\partial \mathbf{p}} &= (g_x, g_y, 0, 0), \frac{\partial \mathbf{p}}{\partial \mathbf{p}_{\mathbf{c}}} = \frac{\partial \Pi_{\mathbf{K}}(\mathbf{p}_{\mathbf{c}})}{\partial \mathbf{p}_{\mathbf{c}}} = \begin{pmatrix} f_x z^{-1} & 0 & -x f_x z^{-2} & 0 \\ 0 & f_y z^{-1} & -y f_y z^{-2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & z^{-2} & 0 \end{pmatrix}
\end{aligned} \tag{1.3}$$

---

## 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 according 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:    $t \leftarrow acc\_list[s].timestamp$ ,  $i \leftarrow s$ 
3:   while true do
4:     if  $i \geq gyro\_list.size$  then
5:       return gyro_list.back
6:     else
7:        $t_{now} \leftarrow gyro\_list[i].timestamp$ 
8:        $t_{next} \leftarrow gyro\_list[i + 1].timestamp$ 
9:       if  $t_{now} < t$  then
10:        if  $t_{next} > t$  then
11:           $t_{front} \leftarrow abs(t_{now} - t)$ ,  $t_{back} \leftarrow abs(t_{next} - t)$ 
12:          return  $t_{front} > t_{back} ? gyro\_list[i + 1] : gyro\_list[i]$ 
13:        else
14:           $i = i + 1$ 
15:        end if
16:      else if  $t_{now} > t$  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  $m, n$ . We have 8 error items to define:

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

$\Delta\bar{\mathbf{v}}_{ij}, \frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^g}, \Delta\bar{\mathbf{p}}_{ij}, \frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^g}$  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  $\Delta\bar{\mathbf{R}}_{ij}, \frac{\partial\Delta\bar{\mathbf{R}}_{ij}}{\partial\mathbf{b}^g}$  are presented here in (2.1), (2.2).

$$\Delta\bar{\mathbf{R}}_{ik} = \begin{cases} \mathbf{I}_{3 \times 3}, & k = i \\ \prod_{m=i}^{k-1} \mathbf{Exp}((\tilde{\omega}_m - \bar{\mathbf{b}}_i^g)\Delta t), & k > i \end{cases}$$

*e.g.*  $k : 0 \rightarrow 44, i = 0$

$$\begin{aligned} \Delta\bar{\mathbf{R}}_{00} &= \mathbf{I}_{3 \times 3} \\ \Delta\bar{\mathbf{R}}_{01} &= \mathbf{Exp}((\tilde{\omega}_0 - \bar{\mathbf{b}}_0^g)\Delta t) \\ \Delta\bar{\mathbf{R}}_{02} &= \mathbf{Exp}((\tilde{\omega}_0 - \bar{\mathbf{b}}_0^g)\Delta t)\mathbf{Exp}((\tilde{\omega}_1 - \bar{\mathbf{b}}_0^g)\Delta t) \\ &\vdots \\ \Delta\bar{\mathbf{R}}_{0(44)} &= \mathbf{Exp}((\tilde{\omega}_0 - \bar{\mathbf{b}}_0^g)\Delta t)\mathbf{Exp}((\tilde{\omega}_1 - \bar{\mathbf{b}}_0^g)\Delta t) \cdots \mathbf{Exp}((\tilde{\omega}_{43} - \bar{\mathbf{b}}_0^g)\Delta t) \end{aligned} \quad (2.1)$$

$$\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^g} = \begin{cases} \mathbf{0}_{3 \times 3}, & k = i \\ \sum_{m=i}^{k-1} -\Delta\bar{\mathbf{R}}_{m+1k}^T \mathbf{J}_r^m \Delta t, & k > i \end{cases}$$

$$= \begin{cases} \mathbf{0}_{3 \times 3}, & k = i \\ \mathbf{J}_r^0 \Delta t, & k = i + 1 \\ \Delta\bar{\mathbf{R}}_{(k-1)k}^T \frac{\partial\Delta\bar{\mathbf{R}}_{i(k-1)}}{\partial\mathbf{b}^g} + \mathbf{J}_r^{k-1} \Delta t, & k > i + 1 \end{cases}$$

*e.g.*  $i = 0, k : 0 \rightarrow 45$

$$\begin{aligned} \frac{\partial\Delta\bar{\mathbf{R}}_{00}}{\partial\mathbf{b}^g} &= \mathbf{0}_{3 \times 3} \\ \frac{\partial\Delta\bar{\mathbf{R}}_{01}}{\partial\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 \\ \frac{\partial\Delta\bar{\mathbf{R}}_{02}}{\partial\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\mathbf{b}^g} + \mathbf{J}_r^1 \Delta t \\ \frac{\partial\Delta\bar{\mathbf{R}}_{03}}{\partial\mathbf{b}^g} &= \sum_{m=0}^2 \Delta\bar{\mathbf{R}}_{(m+1)3}^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 \\ &= (\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 \\ &= \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 \\ &= \Delta\bar{\mathbf{R}}_{23}^T \frac{\partial\Delta\bar{\mathbf{R}}_{02}}{\partial\mathbf{b}^g} + \mathbf{J}_r^2 \Delta t \\ &\vdots \\ \frac{\partial\Delta\bar{\mathbf{R}}_{0(44)}}{\partial\mathbf{b}^g} &= \sum_{m=0}^{43} \Delta\bar{\mathbf{R}}_{(m+1)44}^T \mathbf{J}_r^m \Delta t \\ &= \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 \\ &= \Delta\bar{\mathbf{R}}_{43(44)}^T \frac{\partial\Delta\bar{\mathbf{R}}_{0(43)}}{\partial\mathbf{b}^g} + \mathbf{J}_r^{43} \Delta t \\ \frac{\partial\Delta\bar{\mathbf{R}}_{0(45)}}{\partial\mathbf{b}^g} &= \sum_{m=0}^{44} \Delta\bar{\mathbf{R}}_{(m+1)45}^T \mathbf{J}_r^m \Delta t \\ &= \Delta\bar{\mathbf{R}}_{44(45)}^T \frac{\partial\Delta\bar{\mathbf{R}}_{0(44)}}{\partial\mathbf{b}^g} + \mathbf{J}_r^{44} \Delta t \end{aligned} \quad (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 Preintegration for IMU

---

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

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

```

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 ← rotate_list[i − 1]
4:     rot.timestamp ← gyro_list[i].timestamp
5:     rot.ω ← gyro_list[i].ω − big
6:     rot.ΔRik ← last_r.ΔRik * Exp(rot.ω * Δt)
7:     rot.ΔR(k-1)k ← Exp(rot.ω * Δt)
8:     rot.∂ΔRik∂bg ← ΔR(k-1)kT * last_r.∂ΔRik∂bg − Jr(rot.ω * Δt) * Δt
9:     rot.∂Δvik∂ba ← last_r.∂Δvik∂ba − last_r.ΔRik * Δt
10:    rot.∂Δpik∂ba ← last_r.∂Δpik∂ba + last_r.∂Δvik∂ba * Δt − ½ last_r.ΔRik * Δt2
11:    rotate_list.push(rot)
12:  end for
13:  ΔRij = rotate_list.end.ΔRik
14:  ∂ΔRij∂bg = rotate_list.end.∂ΔRik∂bg
15:  ∂Δvij∂ba = rotate_list.end.∂Δvik∂ba
16:  ∂Δpij∂ba = rotate_list.end.∂Δpik∂ba
17:  for all acc_list[i], i : 0 → n do
18:    cls_r ← time_closest_select(rotate_list, acc_list[i])
19:    acc ← acc_list[i] − bia
20:    Δvij + = cls_r.ΔRik * acc * Δt
21:    ∂Δvij∂bg + = cls_r.ΔRik * acc^ * cls_r.∂ΔRik∂bg * Δt
22:    Δpij + = Δvij * Δt + ½ cls_r.ΔRik * acc * Δt2
23:    ∂Δpij∂bg + = cls_r.∂Δvik∂bg * Δt − ½ cls_r.ΔRik * acc^ * cls_r.∂ΔRik∂bg * Δt2
24:     $A = \begin{pmatrix} cls\_r.\Delta \bar{\mathbf{R}}_{(k-1)k}^T & \mathbf{0} & \mathbf{0} \\ -cls\_r.\Delta \bar{\mathbf{R}}_{ik} * acc^\wedge * \Delta t & \mathbf{I} & \mathbf{0} \\ -\frac{1}{2}cls\_r.\Delta \bar{\mathbf{R}}_{ik} * acc^\wedge * \Delta t^2 & \Delta t \mathbf{I} & \mathbf{I} \end{pmatrix}$ 
25:     $B = \begin{pmatrix} J_r(rot.\omega * \Delta t) * \Delta t & \mathbf{0} \\ \mathbf{0} & cls\_r.\Delta \bar{\mathbf{R}}_{ik} * \Delta t \\ \mathbf{0} & \frac{1}{2}cls\_r.\Delta \bar{\mathbf{R}}_{ik} * \Delta t^2 \end{pmatrix}$ 
26:    Σij = A * Σij * AT + B * Ση * BT
27:  end for
28: 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).

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

---


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

---


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

## Chapter3 Photo Error Factors

### 3.1 Construction residual errors

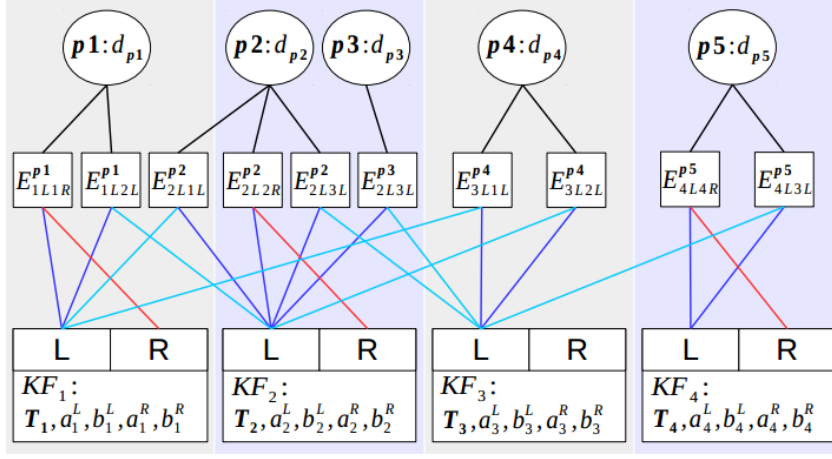


Fig.1

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

$$\chi = \begin{pmatrix} (\phi_1, \dots, \phi_4)^T \\ (\mathbf{p}_1^T, \dots, \mathbf{p}_4^T)^T \\ (\mathbf{v}_1^T, \dots, \mathbf{v}_4^T)^T \\ (\mathbf{b}_1^T, \dots, \mathbf{b}_4^T)^T \\ (d_{\mathbf{p}_1}, \dots, d_{\mathbf{p}_5})^T \\ (a_1^L, a_1^R, b_1^L, b_1^R)^T \\ \vdots \\ (a_4^L, a_4^R, b_4^L, b_4^R)^T \end{pmatrix} \in \mathbb{R}^{81}, \quad \begin{matrix} \phi_i = \text{Log}(\mathbf{R}_i), \\ \xi_i = (\phi_i^T, \mathbf{p}_i^T)^T \end{matrix} \quad (3.1)$$

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

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

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

$$\begin{aligned} \mathbf{J}_s &= \begin{pmatrix} \frac{\partial r_{\mathbf{p}_1}^s}{\partial \delta \xi_1} \dots & \frac{\partial r_{\mathbf{p}_1}^s}{\partial \delta \xi_4} & \frac{\partial r_{\mathbf{p}_1}^s}{\partial d_{\mathbf{p}_1}} \dots & \frac{\partial r_{\mathbf{p}_1}^s}{\partial d_{\mathbf{p}_5}} & \frac{\partial r_{\mathbf{p}_1}^s}{\partial a_1^L} \dots & \frac{\partial r_{\mathbf{p}_1}^s}{\partial b_4^R} \\ \frac{\partial r_{\mathbf{p}_2}^s}{\partial \delta \xi_1} \dots & \frac{\partial r_{\mathbf{p}_2}^s}{\partial \delta \xi_4} & \frac{\partial r_{\mathbf{p}_2}^s}{\partial d_{\mathbf{p}_1}} \dots & \frac{\partial r_{\mathbf{p}_2}^s}{\partial d_{\mathbf{p}_5}} & \frac{\partial r_{\mathbf{p}_2}^s}{\partial a_1^L} \dots & \frac{\partial r_{\mathbf{p}_2}^s}{\partial b_4^R} \\ \frac{\partial r_{\mathbf{p}_5}^s}{\partial \delta \xi_1} \dots & \frac{\partial r_{\mathbf{p}_5}^s}{\partial \delta \xi_4} & \frac{\partial r_{\mathbf{p}_5}^s}{\partial d_{\mathbf{p}_1}} \dots & \frac{\partial r_{\mathbf{p}_5}^s}{\partial d_{\mathbf{p}_5}} & \frac{\partial r_{\mathbf{p}_5}^s}{\partial a_1^L} \dots & \frac{\partial r_{\mathbf{p}_5}^s}{\partial b_4^R} \end{pmatrix}_{3 \times 49} \\ \mathbf{J}_d &= \begin{pmatrix} \frac{\partial (r_{\mathbf{p}_1}^d)_{12}}{\partial \delta \xi_1} \dots & \frac{\partial (r_{\mathbf{p}_1}^d)_{12}}{\partial \delta \xi_4} & \frac{\partial (r_{\mathbf{p}_1}^d)_{12}}{\partial d_{\mathbf{p}_1}} \dots & \frac{\partial (r_{\mathbf{p}_1}^d)_{12}}{\partial d_{\mathbf{p}_5}} & \frac{\partial (r_{\mathbf{p}_1}^d)_{12}}{\partial a_1^L} \dots & \frac{\partial (r_{\mathbf{p}_1}^d)_{12}}{\partial b_4^R} \\ \frac{\partial (r_{\mathbf{p}_1}^d)_{21}}{\partial \delta \xi_1} \dots & \frac{\partial (r_{\mathbf{p}_1}^d)_{21}}{\partial \delta \xi_4} & \frac{\partial (r_{\mathbf{p}_1}^d)_{21}}{\partial d_{\mathbf{p}_1}} \dots & \frac{\partial (r_{\mathbf{p}_1}^d)_{21}}{\partial d_{\mathbf{p}_5}} & \frac{\partial (r_{\mathbf{p}_1}^d)_{21}}{\partial a_1^L} \dots & \frac{\partial (r_{\mathbf{p}_1}^d)_{21}}{\partial b_4^R} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial (r_{\mathbf{p}_5}^d)_{43}}{\partial \delta \xi_1} \dots & \frac{\partial (r_{\mathbf{p}_5}^d)_{43}}{\partial \delta \xi_4} & \frac{\partial (r_{\mathbf{p}_5}^d)_{43}}{\partial d_{\mathbf{p}_1}} \dots & \frac{\partial (r_{\mathbf{p}_5}^d)_{43}}{\partial d_{\mathbf{p}_5}} & \frac{\partial (r_{\mathbf{p}_5}^d)_{43}}{\partial a_1^L} \dots & \frac{\partial (r_{\mathbf{p}_5}^d)_{43}}{\partial b_4^R} \end{pmatrix}_{7 \times 49} \end{aligned} \quad (3.3)$$

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

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

## 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):

$$\frac{\partial (r_{\mathbf{p}}^d)_{ij}}{\partial \delta \xi_m} = \frac{\partial (r_{\mathbf{p}}^d)_{ij}}{\partial \delta \xi_n} = \mathbf{0}^T, \text{ so } \frac{\partial (r_{\mathbf{p}_1}^d)_{12}}{\partial \delta \xi_3} = \frac{\partial (r_{\mathbf{p}_1}^d)_{12}}{\partial \delta \xi_4} = \dots = \mathbf{0}^T, \quad (3.6)$$

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

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

### 3.2.1.1 Jacobian of Affine Brightness Parameters

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

$$\begin{aligned}
(r_{\mathbf{p}}^d)_{ij} &= I_j^L(\mathbf{p}') - b_j^L - \frac{e^{a_j^L}}{e^{a_i^L}}(I_i^L(\mathbf{p}) - b_i^L) \\
\frac{\partial(r_{\mathbf{p}}^d)_{ij}}{\partial a_i^L} &= \frac{e^{a_j^L}}{e^{a_i^L}}(I_i^L(\mathbf{p}) - b_i^L), \quad \frac{\partial(r_{\mathbf{p}}^d)_{ij}}{\partial a_j^L} = -\frac{e^{a_j^L}}{e^{a_i^L}}(I_i^L(\mathbf{p}) - b_i^L) \\
\frac{\partial(r_{\mathbf{p}}^d)_{ij}}{\partial b_i^L} &= \frac{e^{a_j^L}}{e^{a_i^L}}, \quad \frac{\partial(r_{\mathbf{p}}^d)_{ij}}{\partial b_j^L} = -1
\end{aligned} \tag{3.8}$$

### 3.2.1.2 Right Jacobian of Pose

We can use the chain rule to get jacobian of  $\xi_i$  in (3.9):

$$\begin{aligned}
(r_{\mathbf{p}}^d)_{ij} &= I_j^L(\mathbf{p}') - b_j^L - \frac{e^{a_j^L}}{e^{a_i^L}}(I_i^L(\mathbf{p}) - b_i^L) \\
\frac{\partial(r_{\mathbf{p}}^d)_{ij}}{\partial \delta \xi_i} &= \frac{\partial(I_j^L(\mathbf{p}'))}{\partial \mathbf{p}'} \frac{\partial \mathbf{p}'}{\partial \mathbf{p}_c'} \frac{\partial \mathbf{p}_c'}{\partial \delta \xi_i} \\
\frac{\partial(I_j^L(\mathbf{p}'))}{\partial \mathbf{p}'} &= (g'_x, g'_y, 0, 0)^T \\
\frac{\partial \mathbf{p}'}{\partial \mathbf{p}_c'} &= \begin{pmatrix} f_x(z')^{-1} & 0 & -x' f_x(z')^{-2} & 0 \\ 0 & f_y(z')^{-1} & -y' f_y(z')^{-2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & (z')^{-2} & 0 \end{pmatrix} \\
\frac{\partial \mathbf{p}_c'}{\partial \delta \xi_i} &= \frac{\partial(\mathbf{T}_{BC}^{-1} \mathbf{T}_j^{-1} \mathbf{T}_i \mathbf{p}_B)}{\partial \delta \xi_i} = \mathbf{T}_{BC}^{-1} \mathbf{T}_j^{-1} \begin{pmatrix} -\mathbf{R}_i & \mathbf{R}_i \mathbf{p}_B^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{pmatrix}
\end{aligned} \tag{3.9}$$

The jacobian of  $\xi_j$  is enclosed in (3.10):

$$\begin{aligned}
\mathbf{T}_i \mathbf{p}_B &\doteq \mathbf{i} \mathbf{p}_B \\
\frac{\partial \mathbf{p}_c'}{\partial \delta \xi_j} &= \frac{\partial(\mathbf{T}_{BC}^{-1} \mathbf{T}_j^{-1} \mathbf{T}_i \mathbf{p}_B)}{\partial \delta \xi_j} = \mathbf{T}_{BC}^{-1} \frac{\partial(\mathbf{T}_j^{-1} \mathbf{i} \mathbf{p}_B)}{\partial \delta \xi_j} \\
&= \mathbf{T}_{BC}^{-1} \begin{pmatrix} -\mathbf{R}_j^{-1} & \mathbf{R}_j^{-1} \mathbf{i} \mathbf{p}_B^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{pmatrix}
\end{aligned} \tag{3.10}$$



### 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}_{\text{BC}}^{-1} \mathbf{T}_j^{-1} \mathbf{T}_i ((d_{\mathbf{p}}^{i^L})^{-1} \mathbf{T}_{\text{BC}} \mathbf{K}^{-1} \mathbf{p})) \\
&= d_{\mathbf{p}}^{j^L} \mathbf{K}(\mathbf{T}_{\text{BC}}^{-1} \mathbf{T}_j^{-1} \mathbf{T}_i \mathbf{T}_{\text{BC}}) \mathbf{p}_{\text{C}} \\
\mathbf{T}_{\text{BC}}^{-1} \mathbf{T}_j^{-1} \mathbf{T}_i \mathbf{T}_{\text{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}'_{\text{C}} &= \mathbf{T}^{\spadesuit} \mathbf{p}_{\text{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}}^{j^L} \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}'_{\text{C}}} \frac{\partial \mathbf{p}'_{\text{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,  $\xi_i, \xi_j$  do not appear in the expression of  $r_p^s$  as (3.12), the corresponding jacobians are zero.

$$r_p^s := I_i^R(\mathbf{p}') - b_i^R - \frac{e^{a_i^R}}{e^{a_i^L}}(I_i^L(\mathbf{p}) - b_i^L) \quad (3.12)$$

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  $d_p^{iL} \approx d_p^{iR}$ , and  $\mathbf{T}_{RL}$  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) :

$$\begin{aligned} \mathbf{p} &= \begin{pmatrix} u^i \\ v^i \\ 1 \\ d_p^{iL} \end{pmatrix}, \mathbf{p}_c = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, d_p^{iL} = z^{-1}, \mathbf{p}_c = (d_p^{iL})^{-1} \mathbf{K}^{-1} \mathbf{p} \\ &= \begin{pmatrix} f_x^{-1}(d_p^{iL})^{-1}(u^i - c_x) \\ f_y^{-1}(d_p^{iL})^{-1}(v^i - c_y) \\ (d_p^{iL})^{-1} \\ 1 \end{pmatrix}, \mathbf{T}_{RL} = \begin{pmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ \mathbf{p}' &= d_p^{iR} \mathbf{K}(\mathbf{T}_{RL} \mathbf{p}_c) \\ &= d_p^{iL} \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_x^{-1}(d_p^{iL})^{-1}(u^i - c_x) + t_1 \\ f_y^{-1}(d_p^{iL})^{-1}(v^i - c_y) \\ (d_p^{iL})^{-1} \\ 1 \end{pmatrix} = \begin{pmatrix} u^i + t_1 f_x d_p^{iL} \\ v^i \\ 1 \\ d_p^{iL} \end{pmatrix} \\ \frac{\partial r_p^s}{\partial d_p^{iL}} &= \frac{\partial(I_i^R(\mathbf{p}')) - \frac{e^{a_i^R}}{e^{a_i^L}}(I_i^L(\mathbf{p}))}{\partial d_p^{iL}} = \left( \frac{\partial(I_i^R(\mathbf{p}'))}{\partial \mathbf{p}'} - \frac{e^{a_i^R}}{e^{a_i^L}} \frac{\partial(I_i^L(\mathbf{p}))}{\partial \mathbf{p}'} \right) \frac{\partial \mathbf{p}'}{\partial d_p^{iL}} \\ &= [(g_x^{iR}, g_y^{iR}, 0, 0) - \mathbf{0}^T] \begin{pmatrix} t_1 f_x \\ 0 \\ 0 \\ 1 \end{pmatrix} = g_x^{iR} t_1 f_x \end{aligned} \quad (3.13)$$

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

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

---

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