

# Visual Inertial SLAM using Inertial Preintegration

Liyang Liu

June 3, 2016

## Abstract

This document is a report on Visual Inertial SLAM using an efficient method of IMU pre-integration. The pre-integration method combines many IMU data as a single observation before fusion with camera images, resulting in a much reduced state and observation graph structure. An accurate map and robot path is hence obtainable in real-time. We present the pre-integration theory, including formulation of motion state, observation, uncertainty, observation model and the Jacobians. We provide an implementation, compare the performance difference for VIN systems with and without pre-integration. We also include a discussion of various initialization setups along with their impact on linearization of the original problem. Based on our experimental results, we propose future research plans.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>The original VIN problem</b>	<b>3</b>
<b>3</b>	<b>The Preintegration algorithm</b>	<b>4</b>
3.1	Formulation of Preintegration . . . . .	5
3.2	IMU bias . . . . .	6
3.2.1	Modified process model for Inertial-delta + Bias . . . . .	6
3.3	Inertial delta $\Sigma$ and Bias Jacobian . . . . .	6
3.4	Pre-Integration observation model . . . . .	8
3.4.1	Bias treatment . . . . .	8
3.4.2	Feature observation model . . . . .	9
3.4.3	Complete observation model . . . . .	10
<b>4</b>	<b>Experimental results</b>	<b>10</b>
4.1	Comparison of Naive VIN vs Pre-Integration . . . . .	10
4.2	Incremental initialization of states . . . . .	11
4.3	Future improvements . . . . .	11
<b>A</b>	<b>Rotation representation – Euler angles</b>	<b>12</b>
A.1	Rotation matrix . . . . .	12
A.2	Rotation Rate Matrix . . . . .	12
A.3	Camera frame to Global transform . . . . .	12
<b>B</b>	<b>Preintegration VIN Jacobian</b>	<b>12</b>
B.1	Jacobian of Inertial Delta to X . . . . .	12
<b>C</b>	<b>Naive VIN</b>	<b>14</b>
C.1	Jacobian Matrix . . . . .	15
<b>5</b>	<b>Bibliography</b>	<b>17</b>

# 1 Introduction

In this work, we firstly simulate a navigation system equipped with an IMU and an RGB camera.

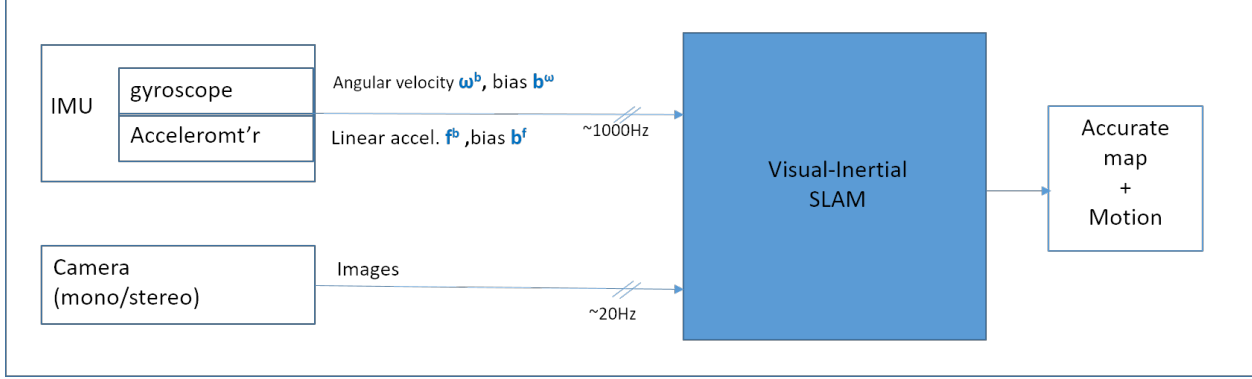


Figure 1: VIN SLAM

**States** The estimated state vector contains the 3-D vehicle position  $\mathbf{p}^n$ , velocity  $\mathbf{v}^n$  and Euler angles  $\mathbf{A}^n = [\alpha, \beta, \gamma]$ ; as well as the  $M$  feature locations  $(\mathbf{f}_i^n)$  in the environment where  $i = 1, \dots, N$ .

**Sensor data** The IMU readings include 3-D linear acceleration  $\mathbf{f}^b$  and angular rate  $\boldsymbol{\omega}^b$ , both given in body frame and come with non-zero bias :  $\mathbf{b}_f$  and  $\mathbf{b}_\omega$ .

Further, due to its design principal, the IMU can only measure acceleration with the gravity taken out, therefore the true vehicle's acceleration in the world frame should be

$$\mathbf{f}^n = R_b^n(\mathbf{f}^b - \mathbf{b}_f) + \mathbf{g}^n$$

$R_b^n$  and  $E_b^n$  are the rotation and rotation rate matrices. Superscript refers to the reference frame,  $^n$  is the navigation frame.

**The Original VIN model** The motion model based on IMU reading can be stated as

$$\begin{aligned} \Delta t &= t_{t+1} - t_t \\ \mathbf{f}_t^n &= R_{bt}^n(\mathbf{f}_t^b - \mathbf{b}_f) \\ \mathbf{v}_{t+1} &= \mathbf{v}_t + \mathbf{f}_t^n \Delta t + \mathbf{g}^n \Delta t \\ \mathbf{p}_{t+1} &= \mathbf{p}_t + \mathbf{v}_t \Delta t \\ \mathbf{A}_{t+1} &= \mathbf{A}_t + E_{bt}^n(\boldsymbol{\omega}_t^b - \mathbf{b}_\omega) \Delta t \end{aligned}$$

## 2 The original VIN problem

For a system composed of an IMU and a RGB camera navigating with  $N$  camera poses,  $K$  IMU samples per image and  $M$  features. The naive VIN problem includes robot poses at all IMU

sample times.

**State Vector** the state vector  $\mathbf{X}$  is defined as:

$$\mathbf{x} = \left( \overbrace{\mathbf{A}_{10}, \mathbf{p}_{10}, \mathbf{A}_{20}, \mathbf{p}_{20}, \dots, \mathbf{p}_{K-1,0}, \mathbf{p}_{K-1,0}, \mathbf{A}_{0,1}, \mathbf{p}_{0,1}, \dots, \mathbf{A}_{K-1,1}, \mathbf{p}_{K-1,1}, \dots, \mathbf{A}_{K-1,N-1}, \mathbf{p}_{K-1,N-1}, \mathbf{A}_{0N}, \mathbf{p}_{0N}}^{K \times (N-1) \text{ poses}}, \right. \\ \underbrace{\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_M}_{M \text{ features}}, \\ \underbrace{\mathbf{v}_{00}, \mathbf{v}_{10}, \dots, \mathbf{v}_{K-1,0}, \dots, \mathbf{v}_{0,N-1}, \dots, \mathbf{v}_{K-1,N-1}, \mathbf{v}_{0N}}_{(K \times (N-1) + 1) \text{ velocities}}, \\ \left. \mathbf{g}^n, \mathbf{A}_{u2c}, \mathbf{T}_{u2c}, \mathbf{b}_f, \mathbf{b}_w \right)'$$

**Observation Vector** The Observation vector  $\mathbf{Z}$  is defined as:

$$\mathbf{z}_{raw} = (\mathbf{z}_{camera}, \mathbf{z}_{IMUraw}, \mathbf{z}_{Tv})' \\ = (\overbrace{\mathbf{uv}_{11}, \mathbf{uv}_{21}, \dots, \mathbf{uv}_{M1}, \dots, \mathbf{uv}_{1N}, \mathbf{uv}_{2N}, \dots, \mathbf{uv}_{MN}}^{M \times N \text{ pixels}}, \\ \overbrace{\omega \mathbf{f}_{01}, \omega \mathbf{f}_{11}, \dots, \omega \mathbf{f}_{(K-1)1}, \dots, \omega \mathbf{f}_{0(N-1)}, \omega \mathbf{f}_{1(N-1)}, \dots, \omega \mathbf{f}_{(K-1)(N-1)}}^{K \times (N-1) \text{ IMU readings}}, \\ \underbrace{\mathbf{0}, \mathbf{0}, \dots, \mathbf{0}}_{K \times (N-1) \text{ zero constraints}})' \quad (1)$$

$$\quad (2)$$

See Appendix C for details on Naive VIN motion model and optimization details. Clearly, such a large state space quickly becomes difficult to manage in practice. Further, each step of re-linearization, the integration from acceleration to velocity then to position has to be recomputed.

### 3 The Preintegration algorithm

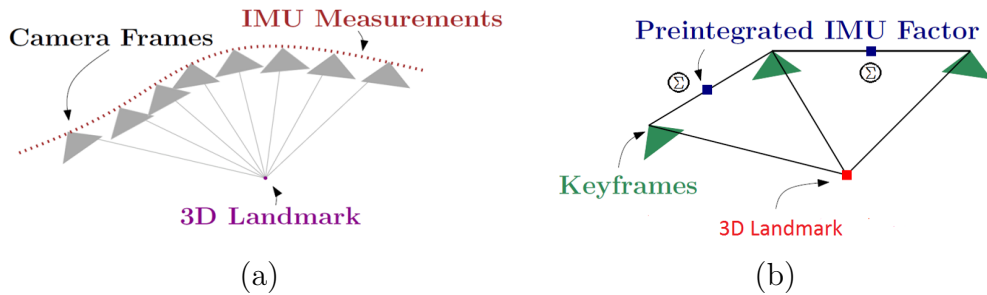


Figure 2: a) Samples: Camera+IMU b) Inertial-delta: preintegrated information [Forster and et al.(2015)]

Todd Lupton proposed the Preintegration method in 2012 [Lupton and Sukkarieh(2012)]: integrate a large number of high rate IMU observations into a single observation, making it

faster and easier to deal with in a SLAM or navigation filter. IMU data are integrated in a body fixed frame that moves with the vehicle, transformation to navigation frame only happens at end of integration, hence the algorithm is referred to as Pre-Integration.

The navigation frame in Preintegration VIN is defined as the body frame at initial robot pose, instead of the traditional globally referenced frame.

With a stereocamera, Todd Lupton also provided a way to recover the absolute global frame after 3 images of observation. However this is not possible with monocular camera setup.

**Inertial Delta observation** The integrated term, including position, velocity and attitude is referred to as inertial delta observation  $\Delta \mathbf{I}$ .

**Motion from Inertial Delta** Relationship between robot motion state to inertial delta is shown in Equations (3 - 5)

$$\mathbf{p}_{t_2}^n = \mathbf{p}_{t_1}^n + (t_2 - t_1)\mathbf{v}_{t_1}^n + \frac{1}{2}(t_2 - t_1)^2 \mathbf{g}^n + R_{bt1}^n \Delta \mathbf{p}_{t_2}^{t_1+} \quad (3)$$

$$\mathbf{v}_{t_2}^n = \mathbf{v}_{t_1}^n + (t_2 - t_1)\mathbf{g}^n + R_{bt1}^n \Delta \mathbf{v}_{t_2}^{t_1} \quad (4)$$

$$\mathbf{A}_{t_2}^n = \text{EulerFromDCM}(R_{bt1}^n \Delta R_{bt2}^{bt1}) \quad (5)$$

---

**Algorithm 1** The Pre-integration Method Based on Inertial Raw Data

---

Inertial-delta observation  $\Delta \mathbf{I} = \begin{bmatrix} \Delta \mathbf{p}_t^+ \\ \Delta \mathbf{v}_t \\ \Delta \mathbf{A}_t \end{bmatrix}$ , initially set to  $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

**for**  $t_1 < t < t_2$  **do**

$\Delta t = t_{t+1} - t_t$

$\mathbf{f}_t^{bt1} = R_{bt}^{bt1}(\mathbf{f}_t^b - \mathbf{b}_f)$

$\Delta \mathbf{v}_{t+1} = \Delta \mathbf{v}_t + \mathbf{f}_t^{bt1} \Delta t$

$\Delta \mathbf{p}_{t+1}^+ = \Delta \mathbf{p}_t^+ + \Delta \mathbf{v}_t \Delta t$

$\Delta \mathbf{A}_{t+1} = \Delta \mathbf{A}_t + E_{bt}^{bt1}(\omega_t^b - \mathbf{b}_\omega) \Delta t$

**end for**

---

### 3.1 Formulation of Preintegration

**State Vector**  $\mathbf{X}_{prn}$  is defined as:

$$\mathbf{X}_{prn} = (\overbrace{\mathbf{A}_2^u, \mathbf{p}_2^u, \dots, \mathbf{A}_N^u, \mathbf{p}_N^u}^{N-1 \text{ poses}}, \overbrace{\mathbf{F}_1, \dots, \mathbf{F}_M}^{M \text{ features}}, \overbrace{\mathbf{v}_1, \dots, \mathbf{v}_N}^{N \text{ velocities}}, \mathbf{g}^n, \mathbf{A}_{u2c}, \mathbf{T}_{u2c}, \mathbf{b}_f, \mathbf{b}_w)'$$

where  $\mathbf{A}_i^u = (\alpha_i^u, \beta_i^u, \gamma_i^u)$ ,  $\mathbf{p}_i^u = (x_i^u, y_i^u, z_i^u)$ ,  $\mathbf{F}_i = (x_i, y_i, z_i)$ .

**Observation vector** becomes:

$$\mathbf{Z}_{prn} = (\overbrace{\mathbf{uv}_1, \dots, \mathbf{uv}_N, \dots, \mathbf{uv}_{1N}, \dots, \mathbf{uv}_{MN}}^{M \times N \text{ pixels}}, \overbrace{\Delta \mathbf{p}_2^+, \Delta \mathbf{v}_2, \Delta \mathbf{A}_2, \dots, \Delta \mathbf{p}_N^+, \Delta \mathbf{v}_N, \Delta \mathbf{A}_N}^{N-1 \text{ inertialDeltas}})'$$

where  $\mathbf{uv}_{ij} = (u_{ij}, v_{ij})$  represents the image of the  $i$ th feature point at the  $j$ th camera pose.

### 3.2 IMU bias

Algorithm 1 gives the pre-integration process model, it is therefore possible to compute is uncertainty from that of IMU measurements. Also, the IMU readings contain non-zero biase terms, so do not reflect the true motion's angular rate and acceleration. We tackle this problem in two steps, we assume  $\mathbf{b}$  is known in computing inertial delta  $\Delta \mathbf{I}$ ; then correct for the error in the optimization stage.

#### 3.2.1 Modified process model for Inertial-delta + Bias

To analyze the effect of the unknown bias term on Inertial Delta observation, we modified the process model in Algorithm 1 to include evolution of Inertial Delta plus bias. The expanded Inertial delta observation is denoted  $\Delta \mathbf{I}^+$ , the new model is illustrated in algorithm 2

---

**Algorithm 2** The Pre-integration Method Based on Inertial Raw Data

---

Extended Inertial-delta observation  $\Delta \mathbf{I}^+ = \begin{bmatrix} \Delta \mathbf{p}_t^+ \\ \Delta \mathbf{v}_t \\ \Delta \mathbf{A}_t \\ \mathbf{b}_f \\ \mathbf{b}_\omega \end{bmatrix}$ , initially set to  $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

**for**  $t_1 < t < t_2$  **do**  
 $\Delta t = t_{t+1} - t_t$   
 $\mathbf{f}_t^{\text{bt1}} = R_{\text{bt}}^{\text{bt1}}(\mathbf{f}_t^{\text{b}} - \mathbf{b}_f)$   
 $\Delta \mathbf{v}_{t+1} = \Delta \mathbf{v}_t + \mathbf{f}_t^{\text{bt1}} \Delta t$   
 $\Delta \mathbf{p}_{t+1}^+ = \Delta \mathbf{p}_t^+ + \Delta \mathbf{v}_t \Delta t$   
 $\Delta \mathbf{A}_{t+1} = \Delta \mathbf{A}_t + E_{\text{bt}}^{\text{bt1}}(\omega_t^{\text{b}} - \mathbf{b}_\omega) \Delta t$   
 $\mathbf{b}_f = \mathbf{b}_f$   
 $\mathbf{b}_\omega = \mathbf{b}_\omega$   
**end for**

---

### 3.3 Inertial delta $\Sigma$ and Bias Jacobian

In order to apply the SLAM optimization framework, we need to provide the inertial delta's uncertainty and variability due to bias (i.e. Jacobian due to biase). To achieve this, we interpret Algorithm 2 as an EKF in its own. This EKF has a state vector defined as  $\Delta \mathbf{I}_t^+$ , it is recursively computed as a function of previous state vector  $\Delta \mathbf{I}_{t-1}^+$  and IMU readings, summarized in equation below:

$$\Delta \mathbf{I}_t^+ = \mathbf{f}(\Delta \mathbf{I}_{t-1}^+, \mathbf{f}_t^{\text{b}}, \omega_t^{\text{b}}) \quad (6)$$

The integration process is recursive, therefore the inertial delta's uncertainty  $\Sigma$  and Jacobian  $J$  (now against both inertial delta and bias terms ) are computed recursively, starting with a zero uncertainty and an identity matrix in the beginning, as illustrated in Figure 3

In each discrete summation step, state covariance  $\Sigma_t$  is related to the uncertainty  $\Sigma_t$  at time  $t - 1$  (Fig 4) and current measurement noise by  $F_t$  and  $G_t$ , where  $F_t$  and  $G_t$  are the Jacobians of the state transition function  $\mathbf{f}()$  w.r.t the state vector  $\Delta \mathbf{I}^+$  and the noise input  $\boldsymbol{\eta}_t$  respectively.

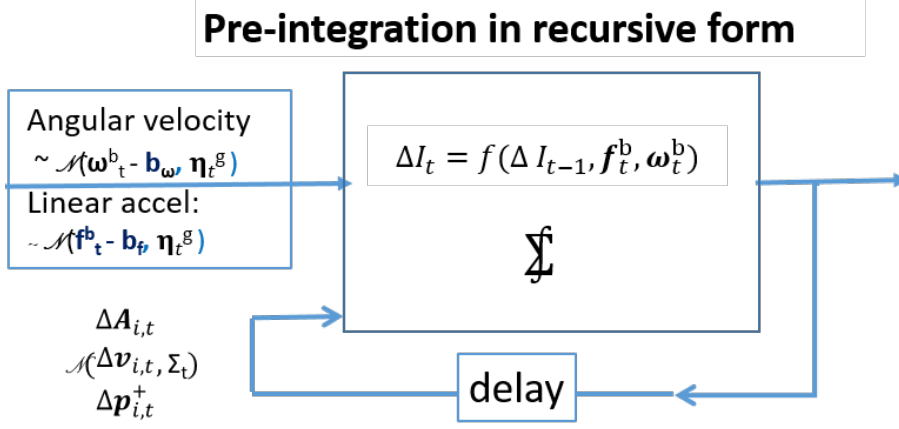


Figure 3: Pre-Integration

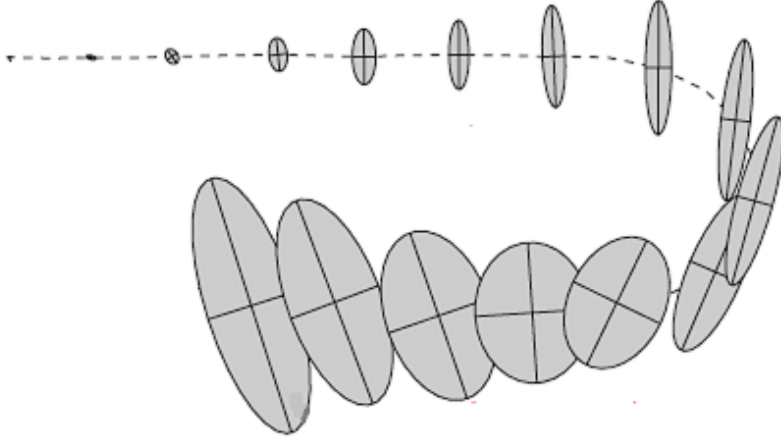


Figure 4: Evolution of uncertainty during Pre-Integration

The modified process's Jacobian  $J_t$  is a chain product of previous process's Jacobian  $J_{t-1}$  and state transition Jacobian  $F_{t-1}$ .

$$\begin{aligned}
 J_{t1}^{t2} &= \frac{\partial \Delta I_{t2}^+}{\partial \Delta I_{t1}^+} \\
 &= \frac{\partial \Delta I_{t1+1}^+}{\partial \Delta I_{t1}^+} \cdots \frac{\partial \Delta I_{t2-1}^+}{\partial \Delta I_{t2-2}^+} \cdot \frac{\partial \Delta I_{t2}^+}{\partial \Delta I_{t2-1}^+}
 \end{aligned} \tag{7}$$

The final structure of Jacobian  $J_{t1}^{t2}$  is shown in Equation 8. As the initial vehicle states are set to zero at the beginning of each delta, only the last two columns of this matrix, relating to the bias terms, are important for the result but the whole matrix is required for the internal calculations. The last 2 columns of the matrix is the Jacobian's bias component, it reflects how the inertial delta  $\Delta \mathbf{I}$  changes versus the biasing terms. This will be useful in the subsequent VIN optimization process. The complete covariance and Jacobian calculation algorithm are listed in

algorithm 3.

---

**Algorithm 3** The Covariance Matrix for the Pre-integration Method
 

---

```

 $J_t = \mathbf{I}_{15}$ 
 $\Sigma_t = \mathbf{I}_{15}$ 
for  $t_1 < t < t_2$  do
   $\Delta t = t_{t+1} - t_t$ 
   $\alpha = \frac{dR_{bt}^{bt1}(\mathbf{f}_t - \mathbf{b}_f)}{d\mathbf{A}_t}$ 
   $\beta = \frac{dE_{bt}^{bt1}(\omega_t - \mathbf{b}_\omega)}{d\mathbf{A}_t}$ 
   $F_t = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3 \Delta t & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \alpha \Delta t & -R_{bt}^{bt1} \Delta t & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 + \beta \Delta t & \mathbf{0}_3 & -E_{bt}^{bt1} \Delta t \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}$ 
   $G_t = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ R_{bt}^{bt1} \Delta t & \mathbf{0}_3 \\ \mathbf{0}_3 & E_{bt}^{bt1} \Delta t \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$ 
   $J_{t+1} = F_t J_t$ 
   $\Sigma_{t+1} = F_t \Sigma_t F_t' + G_t Q_t G_t'$ 
end for
 $J_{t1}^{t2} = J_t$ 
 $\Sigma_{t1}^{t2} = \Sigma_t$ 

```

---

$$J = \begin{bmatrix} \frac{\partial \Delta \mathbf{p}_{t2}^+}{\partial \mathbf{p}_{t1}^{t1}} & \frac{\partial \Delta \mathbf{p}_{t2}^+}{\partial \mathbf{v}_{t1}^{t1}} & \frac{\partial \Delta \mathbf{p}_{t2}^+}{\partial \mathbf{A}_{t1}^{t1}} & \frac{\partial \Delta \mathbf{p}_{t2}^+}{\partial \mathbf{b}_f} & \frac{\partial \Delta \mathbf{p}_{t2}^+}{\partial \mathbf{b}_\omega} \\ \mathbf{0}_3 & \frac{\partial \Delta \mathbf{v}_{t2}}{\partial \mathbf{v}_{t1}^{t1}} & \frac{\partial \Delta \mathbf{v}_{t2}}{\partial \mathbf{A}_{t1}^{t1}} & \frac{\partial \Delta \mathbf{v}_{t2}}{\partial \mathbf{b}_f} & \frac{\partial \Delta \mathbf{v}_{t2}}{\partial \mathbf{b}_\omega} \\ \mathbf{0}_3 & \mathbf{0}_3 & \frac{\partial \Delta \mathbf{A}_{t2}}{\partial \mathbf{A}_{t1}^{t1}} & \mathbf{0}_3 & \frac{\partial \Delta \mathbf{A}_{t2}}{\partial \mathbf{b}_\omega} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \frac{\partial \mathbf{b}_f^{obs}}{\partial \mathbf{b}_f} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \frac{\partial \mathbf{b}_\omega^{obs}}{\partial \mathbf{b}_\omega} \end{bmatrix} \quad (8)$$

### 3.4 Pre-Integration observation model

#### 3.4.1 Bias treatment

In previous section, the bias  $\mathbf{b}$  is assumed to be known. We tackle this in two steps. We first assume  $\mathbf{b}$  is known, referred to as  $\mathbf{b}_0$ . Now let the difference between true bias and observed bias



be  $\delta \mathbf{b} = \mathbf{b}^{est} - \mathbf{b}_0$ , use first order expansion to get inertial delta's modified observation function.

$$\Delta \mathbf{p}^+(\mathbf{b}^0) = \Delta \mathbf{p}^+(\mathbf{b}^{est}) - \frac{\partial \Delta \mathbf{p}^+(\mathbf{b}^0)}{\partial \mathbf{b}} * \delta \mathbf{b} \quad (9)$$

$$\Delta \mathbf{v}(\mathbf{b}^0) = \Delta \mathbf{v}(\mathbf{b}^{est}) - \frac{\partial \Delta \mathbf{v}(\mathbf{b}^0)}{\partial \mathbf{b}} * \delta \mathbf{b} \quad (10)$$

$$\Delta \mathbf{A}(\mathbf{b}^0) = \Delta \mathbf{A}(\mathbf{b}^{est}) - \frac{\partial \Delta \mathbf{A}(\mathbf{b}^0)}{\partial \mathbf{b}} * \delta \mathbf{b} \quad (11)$$

$$\mathbf{b}^0 = \mathbf{b}^{est} - \mathbf{I}_3 * \delta \mathbf{b} \quad (12)$$

Note this  $\delta \mathbf{b}$  is exactly the delta increment we want to compute in each Gauss Newton iteration. Therefore an analytic formula of inertial delta to Jacobian to bias is needed.

Now the expanded Pre-Integration observation model (from Eq 9 - 11) is:

$$\Delta \mathbf{p}_i^+ = R_i(\mathbf{p}_{i+1} - \mathbf{p}_i - \mathbf{v}_i \Delta t - \frac{1}{2} \mathbf{g}^n (\Delta t)^2) - \frac{\partial \Delta \mathbf{p}_t^+}{\partial \mathbf{b}_f} (\mathbf{b}_f - \mathbf{b}_{f0}) - \frac{\partial \Delta \mathbf{p}_t^+}{\partial \mathbf{b}_\omega} (\mathbf{b}_\omega - \mathbf{b}_{\omega0}) \quad (13)$$

$$\Delta \mathbf{v}_i = R_i(\mathbf{v}_{i+1} - \mathbf{v}_i - \mathbf{g} \Delta t) - \frac{\partial \Delta \mathbf{v}_t}{\partial \mathbf{b}_f} (\mathbf{b}_f - \mathbf{b}_{f0}) - \frac{\partial \Delta \mathbf{v}_t}{\partial \mathbf{b}_\omega} (\mathbf{b}_\omega - \mathbf{b}_{\omega0}) \quad (14)$$

$$\Delta \mathbf{A}_i = fn\_ABGFromR(R_{i+1} * R'_i) - \frac{\partial \Delta \mathbf{A}_t}{\partial \mathbf{b}_\omega} (\mathbf{b}_\omega - \mathbf{b}_{\omega0}) \quad (15)$$

where  $[\mathbf{b}_{\omega0}, \mathbf{b}_{f0}]'$  is aforementioned  $\mathbf{b}^0$ , and  $[\mathbf{b}_\omega, \mathbf{b}_f]'$  is aforementioned  $\mathbf{b}^{est}$ .  $i = 2, \dots, N$ ,  $fn\_ABGFromR$  is a function that can obtain Euler angles from a corresponding rotation matrix, and  $R_i, E_i$  correspond to the rotation matrix and rotation rate matrix for the IMU at the time step  $i$  respectively.

The bias Jacobians, found in last 2 column blocks of equation 8 can be borrowed to substitute in the pre-integration observation model.

### 3.4.2 Feature observation model

From feature to its observation by the camera, [Hartley and Zisserman(2004)] tell us the following relationship holds:

$$\mathbf{F}_{ij} = (x_{ij}, y_{ij}, z_{ij}) = R_{cj}(\mathbf{F}_{fi} - \mathbf{p}_{c0j}) \quad (16)$$

$$u_{ij} = f * x_{ij} / z_{ij} + cx_0 \quad (17)$$

$$v_{ij} = f * y_{ij} / z_{ij} + cy_0 \quad (18)$$

$$d_{ij} = z_{ij} \quad (19)$$

where  $f$  is the focal length of the camera,  $(cx_0, cy_0)$  is the displacement of the origin of the camera.

### 3.4.3 Complete observation model

Putting all items together,  $H(\mathbf{x})$  can be written as:

$$\begin{aligned}
 H(\mathbf{x}) &= (H_{camera}(\mathbf{x}), H_{IMU_{int}}(\mathbf{x})) \\
 &= \overbrace{(u_{11}, v_{11}, \dots, u_{M1}, v_{M1}, \dots, u_{1N}, v_{1N}, \dots, u_{MN}, v_{MN})}^{M \times N \times 2}, \\
 &\quad \overbrace{(\Delta \mathbf{p}_2^+, \Delta \mathbf{v}_2, \Delta \mathbf{A}_2, \dots, \Delta \mathbf{p}_N^+, \Delta \mathbf{v}_N, \Delta \mathbf{A}_N)}^{(N-1) \times 9}
 \end{aligned} \tag{20}$$

Details of Jacobian for  $H$  are given in appendix B.

## 4 Experimental results

### 4.1 Comparison of Naive VIN vs Pre-Integration

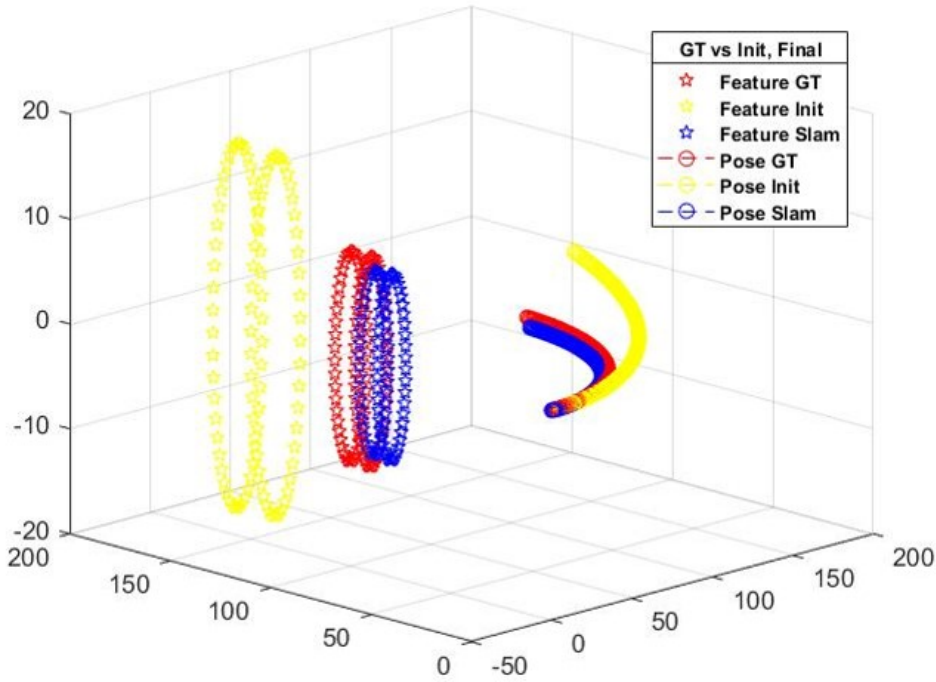


Figure 5: SimuNpose Naive vs Preintegration VIN

A matlab routine *Main\_simuNpose.m* was prepared to simulate VIN system and evaluate the performance of VIN with and without Pre-Integration. Table 1 shows performance results of Naive VIN vs Pre-integration. The later is much more efficient.

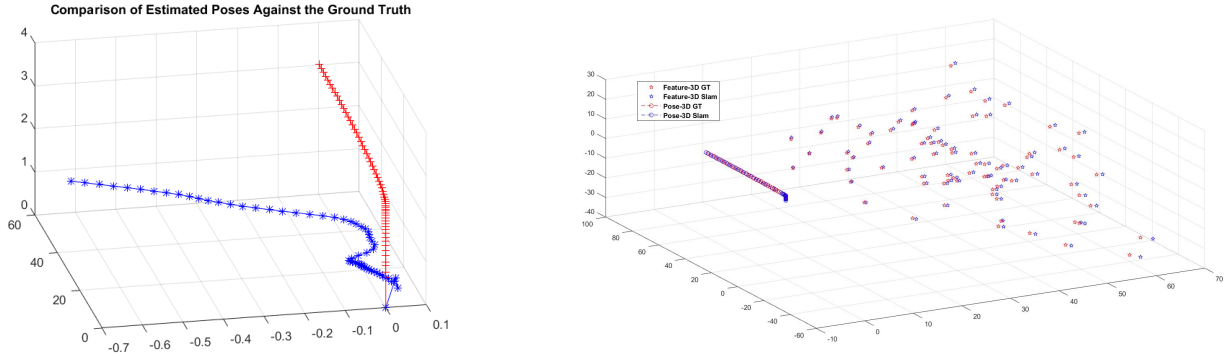
A simulation run of Pre-integration is shown in Figure 5.

Table 1: Caption for the table.

Num image frames		Naive VIN	Pre-Integration
10	Total time	39.1 [sec]	4.4 [sec]
	$\delta X$	4.2	15.7
20	Total time	145.1 [sec]	9.1 [sec]
	$\delta X$	7.69	3.87
100	Total time	NIL	44.5 [sec]
	$\delta X$	NIL	12

## 4.2 Incremental initialization of states

Direct initialization may lead to errors for long tests. This is probably due to non-linearity in transforming from pre-integrated inertial back to global frame, see equation (3 - 5). The problem is solved by incremental introduction of new camera poses. Starting with limited number of frames, obtain optimized robot poses and features. Now introduce new observation, using previously obtained values as initial guess in new optimization, expand state vectors if necessary. Repeat process until all observations are covered. A matlab routine *Main\_inc.m* was developed to simulate the process, see Fig 6.b for illustration. Note: this is in effect similar to iSam2.

Figure 6: *a)* One off initialization , *b)* Incremental initialization

## 4.3 Future improvements

- Change parameterization with Manifold [Forster and et al.(2015)]
- Merge Parallax angles into feature representation [Zhao and Huang(2015)]

## A Rotation representation – Euler angles

### A.1 Rotation matrix

$$\begin{aligned}
 R &= R_x(\alpha)R_y(\beta)R_z(\gamma) \\
 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \begin{pmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

### A.2 Rotation Rate Matrix

$$E = \begin{pmatrix} 1 & 0 & -\sin(\beta) \\ 0 & \cos(\alpha) & \cos(\beta)\sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\beta)\cos(\alpha) \end{pmatrix}$$

### A.3 Camera frame to Global transform

Using the IMU's coordinates at the 1st pose as the global reference frame, the relative position of these two sensors at that time can be related by  $\mathbf{A}_{u2c}$  and  $\mathbf{T}_{u2c}$  :

$$\begin{aligned}
 \mathbf{A}_{u2c} &= (\alpha_{u2c}, \beta_{u2c}, \gamma_{u2c}), \\
 \mathbf{T}_{u2c} &= (x_{u2c}, y_{u2c}, z_{u2c})
 \end{aligned}$$

And at the following poses, given IMU's states ( $\mathbf{R}_i^u$  and  $\mathbf{T}_i^u$ ), the camera's states ( $\mathbf{R}_i^c$  and  $\mathbf{T}_i^c$ ) can be obtained according to this formula:

$$\begin{aligned}
 \mathbf{R}_i^c &= \mathbf{R}_{u2c} \mathbf{R}_i^u \\
 \mathbf{p}_i^c &= \mathbf{p}_i^u + \mathbf{R}_i^{u'} \mathbf{T}_{u2c}
 \end{aligned}$$

## B Preintegration VIN Jacobian

### B.1 Jacobian of Inertial Delta to X

Based on the composition of  $H(\mathbf{x})$ , the corresponding Jacobian matrix can be calculated.

For camera observations of  $(u_{ij}, v_{ij}, d_{ij})$  which represent the observation of the  $i$ th feature at

the  $j$ th camera pose,

$$\frac{\partial u_{ij}}{\partial \mathbf{F}_{ij}} = [f/z_{ij}, 0, -fx_{ij}/z_{ij}^2] \quad (21)$$

$$\frac{\partial v_{ij}}{\partial \mathbf{F}_{ij}} = [0, f/z_{ij}, -fy_{ij}/z_{ij}^2] \quad (22)$$

$$\frac{\partial d_{ij}}{\partial \mathbf{F}_{ij}} = [0, 0, 1] \quad (23)$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{A}_j} = R_{u2c} \frac{\partial R_j}{\partial \mathbf{A}_j} (\mathbf{F}_{i1} - \mathbf{p}_j) \quad (24)$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{T}_j} = -R_{u2c} R_j \quad (25)$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{A}_{u2c}} = \frac{\partial R_{u2c}}{\partial \mathbf{A}_{u2c}} R_j (\mathbf{F}_{i1} - R'_j \mathbf{T}_{u2c} - \mathbf{p}_j) \quad (26)$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{T}_{u2c}} = -R_{u2c} \quad (27)$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{F}_{i1}} = R_{u2c} R_j \quad (28)$$

For  $d\mathbf{p}_i^+$ ,

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{A}_i} = \frac{\partial R_i}{\partial \mathbf{A}_i} (\mathbf{p}_{i+1} - \mathbf{p}_i - \mathbf{v}_i \Delta t - \frac{1}{2} \mathbf{g}^n (\Delta t)^2) \quad (29)$$

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{p}_i} = -R_i \quad (30)$$

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{p}_{i+1}} = R_i \quad (31)$$

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{v}_i} = -R_i \Delta t \quad (32)$$

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{g}} = -\frac{1}{2} R_i \Delta t^2 \quad (33)$$

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{b}_f} = -\frac{\partial \Delta \mathbf{p}_t^+}{\partial \mathbf{b}_f} \quad (34)$$

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{b}_\omega} = -\frac{\partial \Delta \mathbf{p}_t^+}{\partial \mathbf{b}_\omega} \quad (35)$$

For  $dv_i$ ,

$$\frac{\partial d\mathbf{v}_i}{\partial R_i} = \frac{\partial R_i}{\partial \mathbf{A}_i} (\mathbf{v}_{i+1} - \mathbf{v}_i - \mathbf{g}^n \Delta t) \quad (36)$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{v}_i} = -R_i \quad (37)$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{v}_{i+1}} = R_i \quad (38)$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{g}} = -R_i \Delta t \quad (39)$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{b}_f} = -\frac{\partial \Delta \mathbf{v}_t}{\partial \mathbf{b}_f} \quad (40)$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{b}_\omega} = -\frac{\partial \Delta \mathbf{v}_t}{\partial \mathbf{b}_\omega} \quad (41)$$

For  $d\mathbf{A}_i$ ,

$$\frac{\partial \Delta \mathbf{A}_i}{\partial \mathbf{A}_i} = \frac{\partial f^n}{\partial R} R_{i+1} \frac{\partial R_i}{\partial \mathbf{A}_i} \quad (42)$$

$$\frac{\partial \Delta \mathbf{A}_i}{\partial \mathbf{A}_{i+1}} = \frac{\partial f^n}{\partial R} \frac{\partial R_{i+1}}{\partial \mathbf{A}_{i+1}} R_1 \quad (43)$$

$$\frac{\partial \Delta \mathbf{A}_i}{\partial \mathbf{b}_\omega} = -\frac{\partial \Delta \mathbf{A}_t}{\partial \mathbf{b}_\omega} \quad (44)$$

where  $d\mathbf{P}_i = (dx_i, dy_i, dz_i)$ ,  $d\mathbf{v}_i = (dv_{x_i}, dv_{y_i}, dv_{z_i})$ , and  $d\mathbf{A}_i = (d\alpha_i, d\beta_i, d\gamma_i)$ .

## C Naive VIN

The measurements model  $H(\mathbf{x})$  in Naive VIN can be broken into the following three parts:

$$\omega_{ij} = E_{ij}(\mathbf{A}_{(i+1)j} - \mathbf{A}_{ij})/\Delta t + \mathbf{b}_w \quad (45)$$

$$\mathbf{f}_{ij} = R_{ij}((\mathbf{v}_{(i+1)j} - \mathbf{v}_{ij})/\Delta t - \mathbf{g}^n) + \mathbf{b}_f \quad (46)$$

$$\mathbf{bZeros} = \mathbf{p}_{(i+1)j} - \mathbf{p}_{ij} - \mathbf{v}_{ij} \Delta t \quad (47)$$

where  $i = 0, \dots, K-1$ , and  $R_{ij}, E_{ij} (= \begin{pmatrix} 1 & 0 & -\sin(\beta_{ij}) \\ 0 & \cos(\alpha_{ij}) & \cos(\beta_{ij})\sin(\alpha_{ij}) \\ 0 & -\sin(\alpha_{ij}) & \cos(\beta_{ij})\cos(\alpha_{ij}) \end{pmatrix})$  correspond to the

rotation matrix and rotation rate matrix for the IMU at the time step  $i$  since the  $j$ th key camera frame respectively.

Putting all items together,  $H(\mathbf{x})$  can be written as:

$$\begin{aligned}
H(\mathbf{x}) &= (H_{camera}(\mathbf{x}), H_{IMUraw}(\mathbf{x}), H_{Tv}(\mathbf{x})) \\
&= (\overbrace{u_{11}, v_{11}, \dots, u_{M1}, v_{M1}, \dots, u_{1N}, v_{1N}, \dots, u_{MN}, v_{MN}}^{M \times N \times 2}, \\
&\quad \overbrace{\omega \mathbf{f}_{01}, \omega \mathbf{f}_{11}, \dots, \omega \mathbf{f}_{(K-1)1}, \dots, \omega \mathbf{f}_{0(N-1)}, \omega \mathbf{f}_{1(N-1)}, \dots, \omega \mathbf{f}_{(K-1)(N-1)}}^{K \times (N-1) \times 6}, \\
&\quad \overbrace{\mathbf{p}_2 - \mathbf{p}_1 - \mathbf{v}_1 \Delta t, \mathbf{p}_3 - \mathbf{p}_2 - \mathbf{v}_2 \Delta t, \dots, \mathbf{p}_{(N-1)K+1} - \mathbf{p}_{(N-1)K} - \mathbf{v}_{(N-1)K} \Delta t}^{K \times (N-1) \times 3}) \\
&= (\overbrace{f * x_{11}/z_{11} + cx_0, f * y_{11}/z_{11} + cy_0, \dots, f * x_{MN}/z_{MN} + cx_0, f * y_{MN}/z_{MN} + cy_0,}^{M \times N \times 2}, \\
&\quad \overbrace{E_i * (\mathbf{A}_{11} - \mathbf{A}_{01})/\Delta t + \mathbf{b}_w, \dots, R_{(K-1)(N-1)} * ((\mathbf{v}_{0N} - \mathbf{v}_{(K-1)(N-1)})/\Delta t - \mathbf{g}^n) + \mathbf{b}_f}^{K \times (N-1) \times 6}, \\
&\quad \overbrace{\mathbf{p}_2 - \mathbf{p}_1 - \mathbf{v}_1 \Delta t, \mathbf{p}_3 - \mathbf{p}_2 - \mathbf{v}_2 \Delta t, \dots, \mathbf{p}_{(N-1)K+1} - \mathbf{p}_{(N-1)K} - \mathbf{v}_{(N-1)K} \Delta t}^{K \times (N-1) \times 3}) \quad (48)
\end{aligned}$$

## C.1 Jacobian Matrix

Based on the composition of  $H(\mathbf{x})$ , the corresponding Jacobian matrix can be calculated.

For camera observations of  $(u_{ij}, v_{ij}, d_{ij})$  which represent the observation of the  $i$ th feature at the  $j$ th camera pose,

$$\frac{\partial u_{ij}}{\partial \mathbf{F}_{ij}} = [f/z_{ij}, 0, -fx_{ij}/z_{ij}^2] \quad (49)$$

$$\frac{\partial v_{ij}}{\partial \mathbf{F}_{ij}} = [0, f/z_{ij}, -fy_{ij}/z_{ij}^2] \quad (50)$$

$$\frac{\partial d_{ij}}{\partial \mathbf{F}_{ij}} = [0, 0, 1] \quad (51)$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{A}_{0j}} = R_{u2c} \frac{\partial R_{0j}}{\partial \mathbf{A}_{0j}} (\mathbf{F}_{i1} - \mathbf{p}_{0j}) \quad (52)$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{p}_{0j}} = -R_{u2c} R_{0j} \quad (53)$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{A}_{u2c}} = \frac{\partial R_{u2c}}{\partial \mathbf{A}_{u2c}} R_{0j} (\mathbf{F}_{i1} - R'_{0j} \mathbf{T}_{u2c} - \mathbf{p}_{0j}) \quad (54)$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{T}_{u2c}} = -R_{u2c} \quad (55)$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{F}_{fi}} = R_{u2c} R_{0j} \quad (56)$$

For  $\omega_{ij}$ ,

$$\begin{aligned}\frac{\partial \omega_{ij}}{\partial \mathbf{A}_{ij}} &= \frac{\partial E_{ij}}{\partial \mathbf{A}_{ij}} (\mathbf{A}_{(i+1)j} - \mathbf{A}_{ij}) / \Delta t + E_{ij} \left( -\frac{\partial \mathbf{A}_{ij}}{\partial \mathbf{A}_{ij}} \right) / \Delta t \\ &= \left( \frac{\partial E_{ij}}{\partial \mathbf{A}_{ij}} (\mathbf{A}_{(i+1)j} - \mathbf{A}_{ij}) - E_{ij} \right) / \Delta t\end{aligned}\quad (57)$$

$$\frac{\partial E_{ij}}{\partial \mathbf{A}_{ij}} = \left[ \frac{\partial E_{ij}}{\partial \alpha_{ij}}, \frac{\partial E_{ij}}{\partial \beta_{ij}}, \frac{\partial E_{ij}}{\partial \gamma_{ij}} \right] \quad (58)$$

$$\frac{\partial \omega_{ij}}{\partial \mathbf{A}_{(i+1)j}} = E_{ij} \frac{\partial \mathbf{A}_{(i+1)j}}{\partial \mathbf{A}_{(i+1)j}} / \Delta t \quad (59)$$

$$= E_{ij} / \Delta t \quad (60)$$

$$\frac{\partial \omega_{ij}}{\partial b_{\omega}} = I_{3 \times 3} \quad (61)$$

For  $\mathbf{f}_{ij}$ ,

$$\frac{\partial \mathbf{f}_{ij}}{\partial \mathbf{A}_{ij}} = \frac{\partial R_{ij}}{\partial \mathbf{A}_{ij}} ((\mathbf{v}_{i+1} - \mathbf{v}_i) / \Delta t - \mathbf{g}) \quad (62)$$

$$\frac{\partial R_{ij}}{\partial \mathbf{A}_{ij}} = \left[ \frac{\partial R_{ij}}{\partial \alpha_{ij}}, \frac{\partial R_{ij}}{\partial \beta_{ij}}, \frac{\partial R_{ij}}{\partial \gamma_{ij}} \right] \quad (63)$$

$$\frac{\partial \mathbf{f}_{ij}}{\partial \mathbf{v}_{(i+1)j}} = R_{ij} / \Delta t \quad (64)$$

$$\frac{\partial \mathbf{f}_{ij}}{\partial \mathbf{v}_{ij}} = -R_{ij} / \Delta t \quad (65)$$

$$\frac{\partial \mathbf{f}_{ij}}{\partial \mathbf{g}} = -R_{ij} \quad (66)$$

$$\frac{\partial \mathbf{f}_{ij}}{\partial b_f} = I_{3 \times 3} \quad (67)$$

For  $\mathbf{bZeros}_{ij}$ ,

$$\frac{\partial \mathbf{bZeros}_{ij}}{\partial \mathbf{p}_{(i+1)j}} = I_{3 \times 3} \quad (68)$$

$$\frac{\partial \mathbf{bZeros}_{ij}}{\partial \mathbf{p}_{ij}} = -I_{3 \times 3} \quad (69)$$

$$\frac{\partial \mathbf{bZeros}_{ij}}{\partial \mathbf{v}_{ij}} = -I_{3 \times 3} \Delta t \quad (70)$$



## 5 Bibliography

### References

- [Lupton and Sukkarieh(2012)] Todd Lupton and Salah Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics*, 28(1):61–76, 2012. ISSN 15523098. doi: 10.1109/TRO.2011.2170332.
- [Forster and et al.(2015)] Christian Forster and et al. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. *Robotics: Science and Systems*, 11(6), 2015. doi: 10.15607/rss.2015.xi.006.
- [Hartley and Zisserman(2004)] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [Zhao and Huang(2015)] Liang Zhao and Shoudong Huang. ParallaxBA: bundle adjustment using parallax angle feature parametrization. *International Journal of Robotics Research*, 2015.