Visual Inertial SLAM using Inertial Preintegration

Liyang Liu

June 2, 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, obtained results of VIN SLAM 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

1 Introduction

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

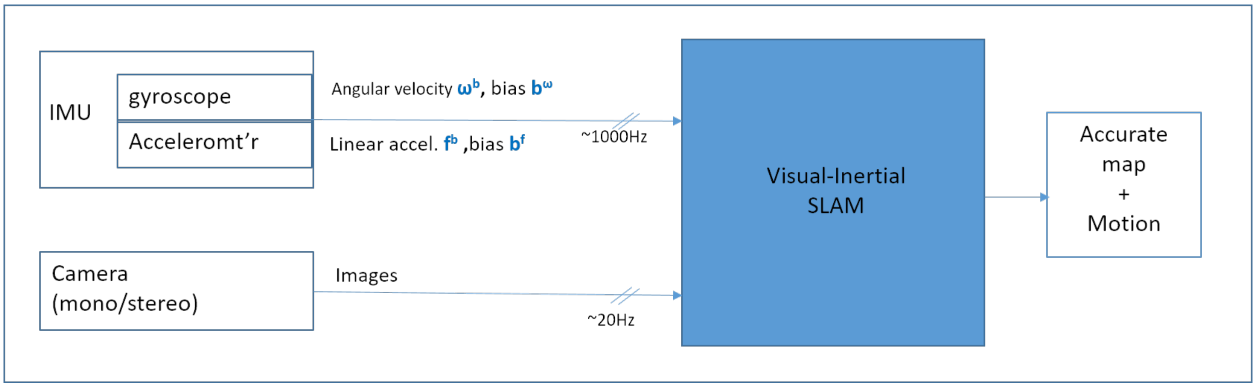

Figure 1: VIN SLAM

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

Sensor data The IMU readings include 3-D linear acceleration $\boldsymbol{f}^{\mathrm{b}}$ and angular rate $\boldsymbol{\omega}^{\mathrm{b}}$, both given in body frame and come with non-zero bias : $\boldsymbol{b}_f$ and $\boldsymbol{b}_\omega$.
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
$\boldsymbol{f}^{\mathrm{n}} = R_b^{\mathrm{n}}(\boldsymbol{f}^{\mathrm{b}} - \boldsymbol{b}_f) + \boldsymbol{g}^{\mathrm{n}}$
R_b^{n} and E_b^{n} are the rotation and rotation rate matrices. Superscript refers to the reference frame, $^{\mathrm{n}}$ is the navigation frame.

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

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

2 The original VIN problem

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

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

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

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

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

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

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 referred to as Pre-Integration.
The navigation frame within the Preintegration framework is defined as the body frame at initial robot pose, instead of the traditional globally referenced frame. Todd Lupton also suggests it is possible to recover global frame after 3 images of preintegration with a stereocamera. However this is not possible with monocular camera setup.

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

$$\mathbf{X}_{prn} = (\overbrace{\mathbf{A}^u_2, \mathbf{p}^u_2, ..., \mathbf{A}^u_N, \mathbf{p}^u_N}^{N-1\ \text{poses}}, \overbrace{\mathbf{F}_1, ..., \mathbf{F}_M}^{M\ \text{features}}, \overbrace{\mathbf{v}_1, ..., \mathbf{v}_N}^{N\ \text{velocities}}, \mathbf{g}^{\mathrm{n}}, \mathbf{A}_{u2c}, \mathbf{T}_{u2c}, \mathbf{b}_f, \mathbf{b}_w)'$$

where $\mathbf{A}^u_i = (\alpha^u_i, \beta^u_i, \gamma^u_i)$, $\mathbf{p}^u_i = (x^u_i, y^u_i, z^u_i)$, $\mathbf{F}_i = (x_i, y_i, z_i)$, $d\mathbf{P}_i = (dx_i, dy_i, dz_i)$, $d\mathbf{v}_i = (dvx_i, dvy_i, dvz_i)$, and $d\mathbf{A}_i = (d\alpha_i, d\beta_i, d\gamma_i)$.

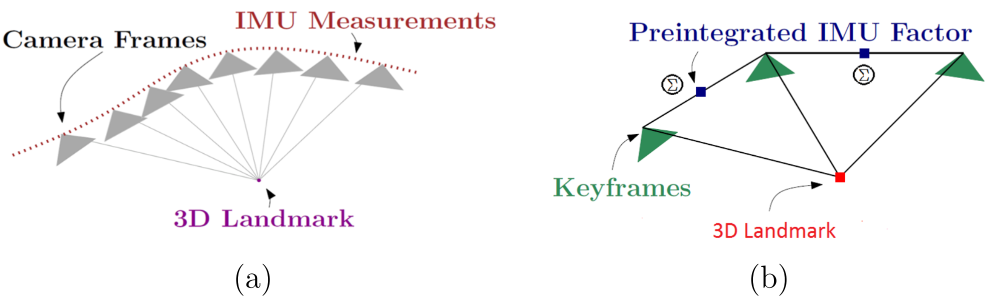

Figure 2: *a*) Samples: Camera+IMU *b*) Inertial-delta: preintegrated information [Forster and et al.(2015)]

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

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

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

end for

Observation vector becomes:

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

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

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

$$\boldsymbol{p}_{t2}^{\mathrm{n}} = \boldsymbol{p}_{t1}^{\mathrm{n}} + (t2 - t1)\boldsymbol{v}_{t1}^{\mathrm{n}} + \frac{1}{2}(t2 - t1)^2 \boldsymbol{g}^{\mathrm{n}} + R_{bt1}^{\mathrm{n}} \triangle \boldsymbol{p}_{t2}^{t1+} \tag{3}$$

$$\boldsymbol{v}_{t2}^{\mathrm{n}} = \boldsymbol{v}_{t1}^{\mathrm{n}} + (t2 - t1)\boldsymbol{g}^{\mathrm{n}} + R_{bt1}^{\mathrm{n}} \triangle \boldsymbol{v}_{t2}^{t1} \tag{4}$$

$$\boldsymbol{A}_{t2}^{\mathrm{n}} = EulerFromDCM(R_{bt1}^{\mathrm{n}} \triangle R_{bt2}^{\mathrm{bt1}}) \tag{5}$$

3.1 Bias-correction

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 biase values, so do not reflect the true motion's angular rate and acceleration. We tackle this problem in two steps, we assume bi is known; then show how to avoid repeating the integration when the bias estimate changes.

3.1.1 Modified process model for Inertial-delta + Bias

To analyze the effect of the unknown bias term on Inertial Delta, we come up with a new process model based on Algorithm 1, it involves evolution of Inertial Delta plus bias, as illustrated in algorithm 2.

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

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

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

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

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

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

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

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

$\mathbf{b}_f = \mathbf{b}_f$

$\mathbf{b}_\omega = \mathbf{b}_\omega$

end for

3.2 Inertial delta Σ and Jacobian for Bias

Now, it should be easy to compute inertial delta's uncertainty and Jacobian recursively based on definition of discrete integration process, see Figure 3.

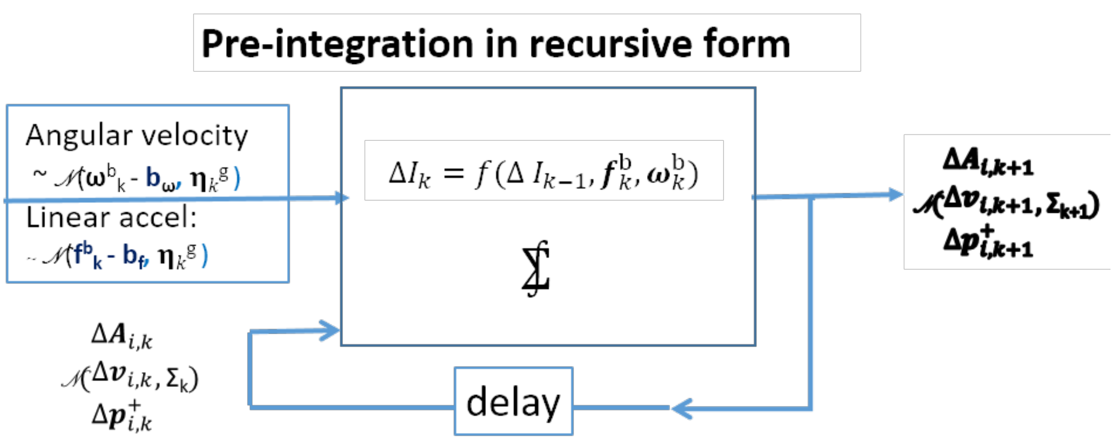

Figure 3: Pre-Integration

The algorithm 1 is modified to include $\boldsymbol{b}$ as part of process vector, as shown in Algorithm 2.

The integration process is recursive, therefore the inertial delta's uncertainty Σ and Jacobian J (now against both inertial delta and Bias terms) are computed recursively, starting from zero uncertainty and unity gain at the beginning, as shown in equation below.

$$\triangle \boldsymbol{I}_k^+ = \boldsymbol{f}(\triangle \boldsymbol{I}_{\boldsymbol{k-1}}^+, \boldsymbol{f}_k^{\mathrm{b}}, \boldsymbol{\omega}_k^{\mathrm{b}}) \tag{6}$$

In each discrete summation step, state covariance Σ_t is related to the uncertainty Σ_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 w.r.t the state vector inertial delta and the noise input $\boldsymbol{\eta}_{\mathrm{k}}$ respectively.

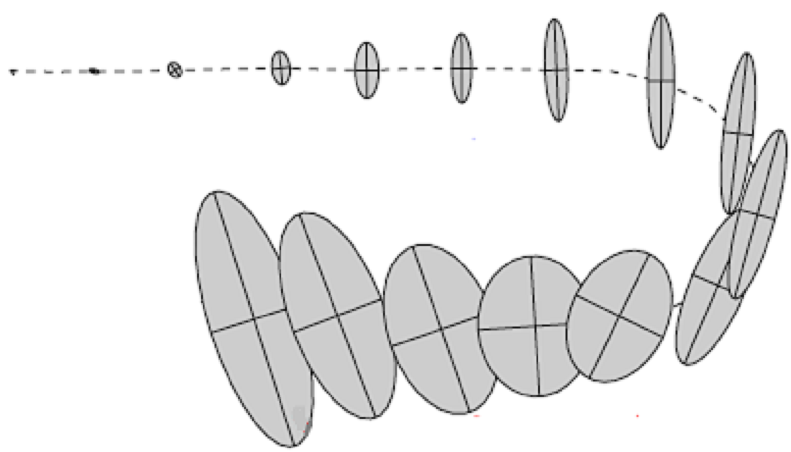

Figure 4: Evolution of uncertainty during Pre-Integration

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}. The Jacobian's structure is shown in Equation 7. The bias component of J (last 2 columns of Eq. 7) will be needed for Preintegration optimization process to account for Inertial Delta's Jacobian on bias in later section. This process is listed in algorithm 3.

$$J = \begin{bmatrix} \frac{\partial \triangle \boldsymbol{p}_{t2}^+}{\partial \boldsymbol{p}_{t1}^{t1}} & \frac{\partial \triangle \boldsymbol{p}_{t2}^+}{\partial \boldsymbol{v}_{t1}^{t1}} & \frac{\partial \triangle \boldsymbol{p}_{t2}^+}{\partial \boldsymbol{A}_{t1}^{t1}} & \frac{\partial \triangle \boldsymbol{p}_{t2}^+}{\partial \boldsymbol{b}_f} & \frac{\partial \triangle \boldsymbol{p}_{t2}^+}{\partial \boldsymbol{b}_\omega} \\ \mathbf{0}_3 & \frac{\partial \triangle \boldsymbol{v}_{t2}}{\partial \boldsymbol{v}_{t1}^{t1}} & \frac{\partial \triangle \boldsymbol{v}_{t2}}{\partial \boldsymbol{A}_{t1}^{t1}} & \frac{\partial \triangle \boldsymbol{v}_{t2}}{\partial \boldsymbol{b}_f} & \frac{\partial \triangle \boldsymbol{v}_{t2}}{\partial \boldsymbol{b}_\omega} \\ \mathbf{0}_3 & \mathbf{0}_3 & \frac{\partial \triangle \boldsymbol{A}_{t2}}{\partial \boldsymbol{A}_{t1}^{t1}} & \mathbf{0}_3 & \frac{\partial \triangle \boldsymbol{A}_{t2}}{\partial \boldsymbol{b}_\omega} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \frac{\partial \boldsymbol{b}_f^{obs}}{\partial \boldsymbol{b}_f} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \frac{\partial \boldsymbol{b}_\omega^{obs}}{\partial \boldsymbol{b}_\omega} \end{bmatrix} \tag{7}$$

3.3 Pre-Integration observation model

3.3.1 Bias treatment

In previous section, the bias $\boldsymbol{b}$ is assumed to be known. This is tackled in two steps. We first assume $\boldsymbol{b}$ is known. Now let the difference between true bias and observed bias be $\delta \boldsymbol{b} = \boldsymbol{b}^{est} - \boldsymbol{b}^{obs}$,

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

use first order expansion to get inertial delta's modified observation function.

$$\triangle \boldsymbol{p}^+(\boldsymbol{b}^{obs}) = \triangle \boldsymbol{p}^+(\boldsymbol{b}^{est}) - \frac{\partial \triangle \boldsymbol{p}^+(\boldsymbol{b}^{obs})}{\partial \boldsymbol{b}} * \delta \boldsymbol{b} \tag{8}$$

$$\triangle \boldsymbol{v}(\boldsymbol{b}^{obs}) = \triangle \boldsymbol{v}(\boldsymbol{b}^{est}) - \frac{\partial \triangle \boldsymbol{v}(\boldsymbol{b}^{obs})}{\partial \boldsymbol{b}} * \delta \boldsymbol{b} \tag{9}$$

$$\triangle \boldsymbol{A}(\boldsymbol{b}^{obs}) = \triangle \boldsymbol{A}(\boldsymbol{b}^{est}) - \frac{\partial \triangle \boldsymbol{A}(\boldsymbol{b}^{obs})}{\partial \boldsymbol{b}} * \delta \boldsymbol{b} \tag{10}$$

$$\boldsymbol{b}^{obs} = \boldsymbol{b}^{est} - I_3 * \delta \boldsymbol{b} \tag{11}$$

Note this $\delta \boldsymbol{b}$ 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 8 - 10)is:

$$\triangle \mathbf{p}_i^+ = R_i(\mathbf{p}_{i+1} - \mathbf{p}_i - \mathbf{v}_i \triangle t - \frac{1}{2}\mathbf{g}^{\mathrm{n}}(\triangle t)^2) - \frac{\partial \triangle \mathbf{p}_t^+}{\partial \mathbf{b}_f}(\mathbf{b}_f - \mathbf{b}_{f0}) - \frac{\partial \triangle \mathbf{p}_t^+}{\partial \mathbf{b}_\omega}(\mathbf{b}_\omega - \mathbf{b}_{\omega 0}) \tag{12}$$

$$\triangle \mathbf{v}_i = R_i(\mathbf{v}_{i+1} - \mathbf{v}_i - \mathbf{g}\triangle t) - \frac{\partial \triangle \mathbf{v}_t}{\partial \mathbf{b}_f}(\mathbf{b}_f - \mathbf{b}_{f0}) - \frac{\partial \triangle \mathbf{v}_t}{\partial \mathbf{b}_\omega}(\mathbf{b}_\omega - \mathbf{b}_{\omega 0}) \tag{13}$$

$$\triangle \mathbf{A}_i = fn\_ABGFromR(R_{i+1} * R_i') - \frac{\partial \triangle \mathbf{A}_t}{\partial \mathbf{b}_\omega}(\mathbf{b}_\omega - \mathbf{b}_{\omega 0}) \tag{14}$$

where $[\mathbf{b}_\omega, \mathbf{b}_f]'$ is aforementioned $\mathbf{b}^{obs}$, and $[\mathbf{b}_{\omega 0}, \mathbf{b}_{f0}]'$ is aforementioned $\mathbf{b}^{est}$. $i = 2, ..., N$, $fn\_ABGFromR$ 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.

3.3.2 Feature observation model

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

$$\mathbf{F}_{ij} = (x_{ij}, y_{ij}, z_{ij}) = R_{cj}(\mathbf{F}_{fi} - \mathbf{p}_{c0j}) \tag{15}$$

$$u_{ij} = f * x_{ij}/z_{ij} + cx_0 \tag{16}$$

$$v_{ij} = f * y_{ij}/z_{ij} + cy_0 \tag{17}$$

$$d_{ij} = z_{ij} \tag{18}$$

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

3.3.3 Complete observation model

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

$$\begin{aligned} H(\mathbf{x}) &= (H_{camera}(\mathbf{x}), H_{IMUint}(\mathbf{x})) \\ &= (\overbrace{u_{11}, v_{11}, ..., u_{M1}, v_{M1}, ..., u_{1N}, v_{1N}, ..., u_{MN}, v_{MN}}^{M\times N\times 2}, \\ &\quad \overbrace{\triangle\mathbf{p}_2^+, \triangle\mathbf{v}_2, \triangle\mathbf{A}_2, ..., \triangle\mathbf{p}_N^+, \triangle\mathbf{v}_N, \triangle\mathbf{A}_N}^{(N-1)\times 9}) \end{aligned} \tag{19}$$

Details of Jacobian for H are given in appendix B.

3.3.4 Bias Jacobian

Equation 7 shows the contents of inertial delta Jacobian. We will borrow the parts related to Bias, i.e. last 2 column blocks of J_{t1}^{t2} for inclusion in the pre-integration observation model.

4 Experimental results

4.1 Comparison of Naive VIN vs Pre-Integration

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.

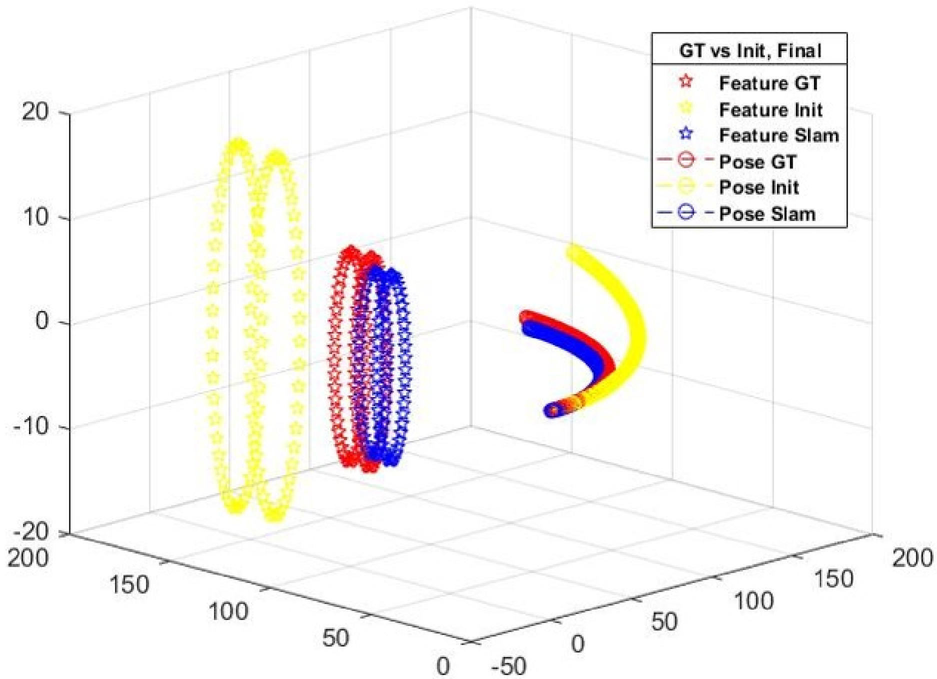

Figure 5: SimuNpose Naive vs Preintegration VIN

Table 1: Caption for the table.

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

4.2 Incremental implementation of Pre-Integration

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 figure **??**. 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 for illustration. Note: this is in effect similar to iSam2.

4.3 Future improvements

- Change parameterization with Manifold [Forster and et al.(2015)]

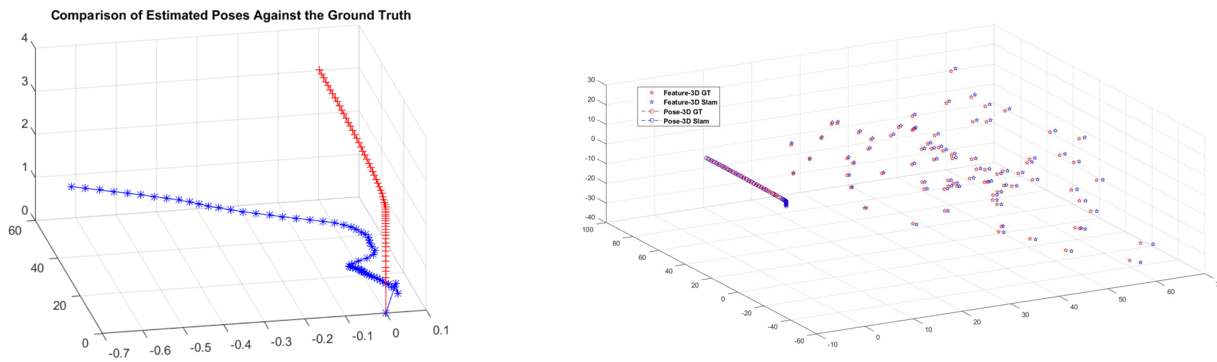

Figure 6: *a*) One off initialization , *b*) Incremental initialization

- 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:

$$\begin{aligned} \mathbf{R}_i^c &= \mathbf{R}_{u2c}\mathbf{R}_i^u \\ \mathbf{p}_i^c &= \mathbf{p}_i^u + \mathbf{R}_i^{u\prime}\mathbf{T}_{u2c} \end{aligned}$$

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 ith feature at the jth camera pose,

$$\frac{\partial u_{ij}}{\partial \mathbf{F}_{ij}} = [f/z_{ij}, 0, -fx_{ij}/z_{ij}^2] \tag{20}$$
$$\frac{\partial v_{ij}}{\partial \mathbf{F}_{ij}} = [0, f/z_{ij}, -fy_{ij}/z_{ij}^2] \tag{21}$$
$$\frac{\partial d_{ij}}{\partial \mathbf{F}_{ij}} = [0, 0, 1] \tag{22}$$
$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{A}_j} = R_{u2c}\frac{\partial R_j}{\partial \mathbf{A}_j}(\mathbf{F}_{i1} - \mathbf{p}_j) \tag{23}$$
$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{T}_j} = -R_{u2c}R_j \tag{24}$$
$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{A}_{u2c}} = \frac{\partial R_{u2c}}{\partial \mathbf{A}_{u2c}}R_j(\mathbf{F}_{i1} - R_j'\mathbf{T}_{u2c} - \mathbf{p}_j) \tag{25}$$
$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{T}_{u2c}} = -R_{u2c} \tag{26}$$
$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{F}_{i1}} = R_{u2c}R_j \tag{27}$$

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

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{A}_i} = \frac{\partial R_i}{\partial \mathbf{A}_i}(\mathbf{p}_{i+1} - \mathbf{p}_i - \mathbf{v}_i\triangle t - \frac{1}{2}\mathbf{g}^{\mathrm{n}}(\triangle t)^2) \tag{28}$$
$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{p}_i} = -R_i \tag{29}$$
$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{p}_{i+1}} = R_i \tag{30}$$
$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{v}_i} = -R_i\triangle t \tag{31}$$
$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{g}} = -\frac{1}{2}R_i\triangle t^2 \tag{32}$$
$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{b}_f} = -\frac{\partial \triangle \mathbf{p}_t^+}{\partial \mathbf{b}_f} \tag{33}$$
$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{b}_\omega} = -\frac{\partial \triangle \mathbf{p}_t^+}{\partial \mathbf{b}_\omega} \tag{34}$$

For $d\mathrm{v}_i$,

$$\frac{\partial d\mathbf{v}_i}{\partial R_i} = \frac{\partial R_i}{\partial \mathbf{A}_i}(\mathbf{v}_{i+1} - \mathbf{v}_i - \mathbf{g}^{\mathrm{n}} \triangle t) \tag{35}$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{v}_i} = -R_i \tag{36}$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{v}_{i+1}} = R_i \tag{37}$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{g}} = -R_i \triangle t \tag{38}$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{b}_f} = -\frac{\partial \triangle \mathbf{v}_t}{\partial \mathbf{b}_f} \tag{39}$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{b}_\omega} = -\frac{\partial \triangle \mathbf{v}_t}{\partial \mathbf{b}_\omega} \tag{40}$$

For $d\mathbf{A}_i$,

$$\frac{\partial \triangle \mathbf{A}_i}{\partial \mathbf{A}_i} = \frac{\partial fn}{\partial R} R_{i+1} \frac{\partial R_i}{\partial \mathbf{A}_i} \tag{41}$$

$$\frac{\partial \triangle \mathbf{A}_i}{\partial \mathbf{A}_{i+1}} = \frac{\partial fn}{\partial R} \frac{\partial R_{i+1}}{\partial \mathbf{A}_{i+1}} R_1 \tag{42}$$

$$\frac{\partial \triangle \mathbf{A}_i}{\partial \mathbf{b}_\omega} = -\frac{\partial \triangle \mathbf{A}_t}{\partial \mathbf{b}_\omega} \tag{43}$$

C Naive VIN

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

$$\omega_{ij} = E_{ij}(\mathbf{A}_{(i+1)j} - \mathbf{A}_{ij})/\triangle t + \mathbf{b}_w \tag{44}$$

$$\mathbf{f}_{ij} = R_{ij}((\mathbf{v}_{(i+1)j} - \mathbf{v}_{ij})/\triangle t - \mathbf{g}^{\mathrm{n}}) + \mathbf{b}_f \tag{45}$$

$$\mathbf{bZeros} = \mathbf{p}_{(i+1)j} - \mathbf{p}_{ij} - \mathbf{v}_{ij} \triangle t \tag{46}$$

where $i = 0, ..., K-1$, 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 jth key camera frame respectively.

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

$$
\begin{aligned}
H(\mathbf{x}) &= (H_{camera}(\mathbf{x}), H_{IMUraw}(\mathbf{x}), H_{Tv}(\mathbf{x})) \\
=&(\overbrace{u_{11}, v_{11}, ..., u_{M1}, v_{M1}, ..., u_{1N}, v_{1N}, ..., u_{MN}, v_{MN}}^{M\times N\times 2}, \\
&\overbrace{\omega\mathbf{f}_{01}, \omega\mathbf{f}_{11}, ..., \omega\mathbf{f}_{(K-1)1}, ..., \omega\mathbf{f}_{0(N-1)}, \omega\mathbf{f}_{1(N-1)}, ..., \omega\mathbf{f}_{(K-1)(N-1)}}^{K\times(N-1)\times 6}, \\
&\overbrace{\mathbf{p}_2 - \mathbf{p}_1 - \mathbf{v}_1\triangle t, \mathbf{p}_3 - \mathbf{p}_2 - \mathbf{v}_2\triangle t, ..., \mathbf{p}_{(N-1)K+1} - \mathbf{p}_{(N-1)K} - \mathbf{v}_{(N-1)K}\triangle t}^{K\times(N-1)\times 3}) \\
=&(\overbrace{f * x_{11}/z_{11} + cx_0, f * y_{11}/z_{11} + cy_0, ..., f * x_{MN}/z_{MN} + cx_0, f * y_{MN}/z_{MN} + cy_0,}^{M\times N\times 2}, \\
&\overbrace{E_i * (\mathbf{A}_{11} - \mathbf{A}_{01})/\triangle t + \mathbf{b}_w, ..., R_{(K-1)(N-1)} * ((\mathbf{v}_{0N} - \mathbf{v}_{(K-1)(N-1)})/\triangle t - \mathbf{g}^{\mathrm{n}}) + \mathbf{b}_f}^{K\times(N-1)\times 6}, \\
&\overbrace{\mathbf{p}_2 - \mathbf{p}_1 - \mathbf{v}_1\triangle t, \mathbf{p}_3 - \mathbf{p}_2 - \mathbf{v}_2\triangle t, ..., \mathbf{p}_{(N-1)K+1} - \mathbf{p}_{(N-1)K} - \mathbf{v}_{(N-1)K}\triangle t}^{K\times(N-1)\times 3})
\end{aligned} \tag{47}
$$

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 ith feature at the jth camera pose,

$$\frac{\partial u_{ij}}{\partial \mathbf{F}_{ij}} = [f/z_{ij}, 0, -fx_{ij}/z_{ij}^2] \tag{48}$$

$$\frac{\partial v_{ij}}{\partial \mathbf{F}_{ij}} = [0, f/z_{ij}, -fy_{ij}/z_{ij}^2] \tag{49}$$

$$\frac{\partial d_{ij}}{\partial \mathbf{F}_{ij}} = [0, 0, 1] \tag{50}$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{A}_{0j}} = R_{u2c}\frac{\partial R_{0j}}{\partial \mathbf{A}_{0j}}(\mathbf{F}_{i1} - \mathbf{p}_{0j}) \tag{51}$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{p}_{0j}} = -R_{u2c}R_{0j} \tag{52}$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{A}_{u2c}} = \frac{\partial R_{u2c}}{\partial \mathbf{A}_{u2c}}R_{0j}(\mathbf{F}_{i1} - R'_{0j}\mathbf{T}_{u2c} - \mathbf{p}_{0j}) \tag{53}$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{T}_{u2c}} = -R_{u2c} \tag{54}$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{F}_{fi}} = R_{u2c}R_{0j} \tag{55}$$

For ω_{ij},

$$
\begin{aligned}
\frac{\partial \omega_{ij}}{\partial \mathbf{A}_{ij}} &= \frac{\partial E_{ij}}{\partial \mathbf{A}_{ij}}(\mathbf{A}_{(i+1)j} - \mathbf{A}_{ij})/\triangle t + E_{ij}(-\frac{\partial \mathbf{A}_{ij}}{\partial \mathbf{A}_{ij}})/\triangle t \\
&= (\frac{\partial E_{ij}}{\partial \mathbf{A}_{ij}}(\mathbf{A}_{(i+1)j} - \mathbf{A}_{ij}) - E_{ij})/\triangle t && (56) \\
\frac{\partial E_{ij}}{\partial \mathbf{A}_{ij}} &= [\frac{\partial E_{ij}}{\partial \alpha_{ij}}, \frac{\partial E_{ij}}{\partial \beta_{ij}}, \frac{\partial E_{ij}}{\partial \gamma_{ij}}] && (57) \\
\frac{\partial \omega_{ij}}{\partial \mathbf{A}_{(i+1)j}} &= E_{ij}\frac{\partial \mathbf{A}_{(i+1)j}}{\partial \mathbf{A}_{(i+1)j}}/\triangle t && (58) \\
&= E_{ij}/\triangle t && (59) \\
\frac{\partial \omega_{ij}}{\partial b_\omega} &= I_{3\times 3} && (60)
\end{aligned}
$$

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

$$
\begin{aligned}
\frac{\partial \mathbf{f}_{ij}}{\partial \mathbf{A}_{ij}} &= \frac{\partial R_{ij}}{\mathbf{A}_{ij}}((\mathbf{v}_{i+1} - \mathbf{v}_i)/\triangle t - \mathbf{g}) && (61) \\
\frac{\partial R_{ij}}{\partial \mathbf{A}_{ij}} &= [\frac{\partial R_{ij}}{\partial \alpha_{ij}}, \frac{\partial R_{ij}}{\partial \beta_{ij}}, \frac{\partial R_{ij}}{\partial \gamma_{ij}}] && (62) \\
\frac{\partial \mathbf{f}_{ij}}{\partial \mathbf{v}_{(i+1)j}} &= R_{ij}/\triangle t && (63) \\
\frac{\partial \mathbf{f}_{ij}}{\partial \mathbf{v}_{ij}} &= -R_{ij}/\triangle t && (64) \\
\frac{\partial \mathbf{f}_{ij}}{\partial \mathbf{g}} &= -R_{ij} && (65) \\
\frac{\partial \mathbf{f}_{ij}}{\partial b_f} &= I_{3\times 3} && (66)
\end{aligned}
$$

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

$$
\begin{aligned}
\frac{\partial \mathbf{bZeros}_{ij}}{\partial \mathbf{p}_{(i+1)j}} &= I_{3\times 3} && (67) \\
\frac{\partial \mathbf{bZeros}_{ij}}{\partial \mathbf{p}_{ij}} &= -I_{3\times 3} && (68) \\
\frac{\partial \mathbf{bZeros}_{ij}}{\partial \mathbf{v}_{ij}} &= -I_{3\times 3}\triangle t && (69)
\end{aligned}
$$

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. 2015.