Visual Inertial SLAM using Inertial Preintegration

Liyang Liu

May 30, 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 state, observation, covariance, observation model and Jacobians. We provide an implementation, obtained results of VIN SLAM with and without pre-integration. We also include a discussion of various methods of initialization 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.

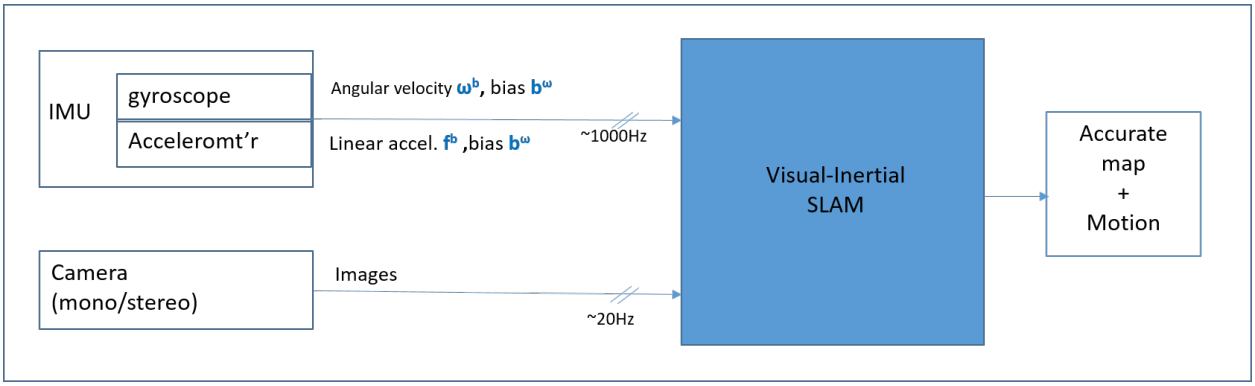


Figure 1: VIN SLAM

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.

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:

$$\begin{aligned}\mathbf{x} =&(\overbrace{\mathbf{A}_{10}, \mathbf{T}_{10}, \mathbf{A}_{20}, \mathbf{T}_{20}, ..., \mathbf{A}_{K-1,0}, \mathbf{T}_{K-1,0}, \mathbf{A}_{0,1}, \mathbf{T}_{0,1}, ..., \mathbf{A}_{K-1,1}, \mathbf{T}_{K-1,1}, ...\mathbf{A}_{K-1,N-1}, \mathbf{T}_{K-1,N-1}, \mathbf{A}_{0N}, \mathbf{T}_{0N}}^{K(N-1)\times 6},\\ &\overbrace{\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_M}^{M\times 3},\\ &\overbrace{\mathbf{v}_{00}, \mathbf{v}_{10}..., \mathbf{v}_{K-1,0}, ..., \mathbf{v}_{0,N-1}, ..., \mathbf{v}_{K-1,N-1}, \mathbf{v}_{0N}}^{(K(N-1)+1)\times 3},\\ &\mathbf{g}, \mathbf{A}_{u2c}, \mathbf{T}_{u2c}, \mathbf{b}_f, \mathbf{b}_w)'\end{aligned}$$

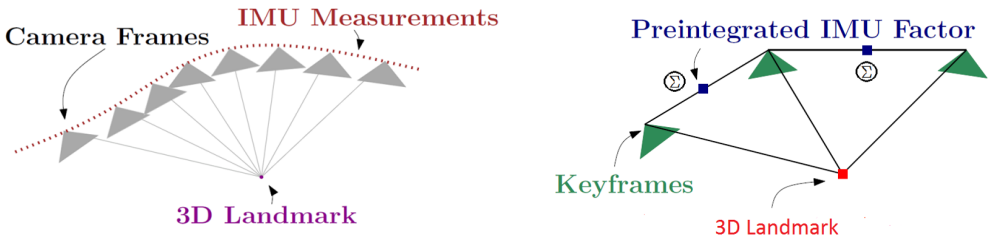
Observation Vector The Observation vector $\mathbf{Z}$ is defined as:

$$\begin{aligned}\mathbf{z}_{raw} &= (\mathbf{z}_{camera}, \mathbf{z}_{IMUraw}, \mathbf{z}_{Tv})'\\ &=(\overbrace{\mathbf{uv}_{11}, \mathbf{uv}_{21}, ..., \mathbf{uv}_{M1}, ..., \mathbf{uv}_{1N}, \mathbf{uv}_{2N}, ..., \mathbf{uv}_{MN}}^{M\times N\times 2},\\ &\overbrace{\omega\mathbf{a}_{01}, \omega\mathbf{a}_{11}, ..., \omega\mathbf{a}_{(K-1)1}, ..., \omega\mathbf{a}_{0(N-1)}, \omega\mathbf{a}_{1(N-1)}, ..., \omega\mathbf{a}_{(K-1)(N-1)}}^{K\times(N-1)\times 6}, \qquad (1)\\ &\overbrace{\mathbf{0}, \mathbf{0}, ..., \mathbf{0}}^{K\times(N-1)\times 3})' \qquad (2)\end{aligned}$$

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 propose 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.



(a) (b)

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

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

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

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

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

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

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

end for

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

$$\mathbf{x} = (\overbrace{\mathbf{A}_2^u, \mathbf{T}_2^u, ..., \mathbf{A}_N^u, \mathbf{T}_N^u}^{(N-1)\times 6}, \overbrace{\mathbf{F}_1, ..., \mathbf{F}_M}^{M\times 3}, \overbrace{\mathbf{v}_1, ..., \mathbf{v}_N}^{N\times 3}, \mathbf{g}^{\mathrm{n}}, \mathbf{A}_{u2c}, \mathbf{T}_{u2c}, \mathbf{b}_f, \mathbf{b}_w)'$$

where $\mathbf{A}_i^u = (\alpha_i^u, \beta_i^u, \gamma_i^u)$, $\mathbf{T}_i^u = (x_i^u, y_i^u, z_i^u)$, $\mathbf{F}_i = (x_i, y_i, z_i)$, $d\mathbf{P}_i = (dx_i, dy_i, dz_i)$, $d\mathbf{v}_i = (dvx_i, dvy_i, dvz_i)$, and $d\mathbf{A}_i = (d\alpha_i, d\beta_i, d\gamma_i)$.

Observation vector becomes:

$$\mathbf{z} = (\overbrace{\mathbf{uvd}_1, ..., \mathbf{uvd}_N, ..., \mathbf{uvd}_{1N}, ..., \mathbf{uvd}_{MN}}^{M\times N\times 3}, \overbrace{d\mathbf{p}_2, d\mathbf{v}_2, d\mathbf{A}_2, ..., d\mathbf{p}_N, d\mathbf{v}_N, d\mathbf{A}_N}^{(N-1)\times 9})'$$

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

3.1 Bias-correction

Algorithm 1 give the pre-integration process model, it is therefore possible to compute is uncertainty from that of IMU measurements. Also, the IMU reading contains biase values, so do not reflect the true motion's angular rate and acceleration. This is tackled in two steps. We first assume $\boldsymbol{b}$ is known. Now let the difference between true bias and observed bias be $\delta\boldsymbol{b} = \boldsymbol{b}^{obs} - \boldsymbol{b}^{est}$ use first order expansion to get inertial delta's modified observation function.

$$\begin{aligned}
\triangle \boldsymbol{p}^+(\boldsymbol{b}^{obs}) &= \triangle \boldsymbol{p}^+(\boldsymbol{b}^{est}) - \frac{\partial \triangle \boldsymbol{p}^+(\boldsymbol{b}^{obs})}{\partial \boldsymbol{b}} * \delta \boldsymbol{b} \\
\triangle \boldsymbol{v}(\boldsymbol{b}^{obs}) &= \triangle \boldsymbol{v}(\boldsymbol{b}^{est}) - \frac{\partial \triangle \boldsymbol{v}(\boldsymbol{b}^{obs})}{\partial \boldsymbol{b}} * \delta \boldsymbol{b} \\
\triangle \boldsymbol{A}(\boldsymbol{b}^{obs}) &= \triangle \boldsymbol{A}(\boldsymbol{b}^{est}) - \frac{\partial \triangle \boldsymbol{\phi}(\boldsymbol{b}^{obs})}{\partial \boldsymbol{b}} * \delta \boldsymbol{b} \\
\boldsymbol{b}^{obs} &= \boldsymbol{b}^{est} - I_3 * \delta \boldsymbol{b}
\end{aligned}$$

This gives a new process model base on Algorithm 1 to compute the inertial deltas and bias, is illustrated in algorithm 2

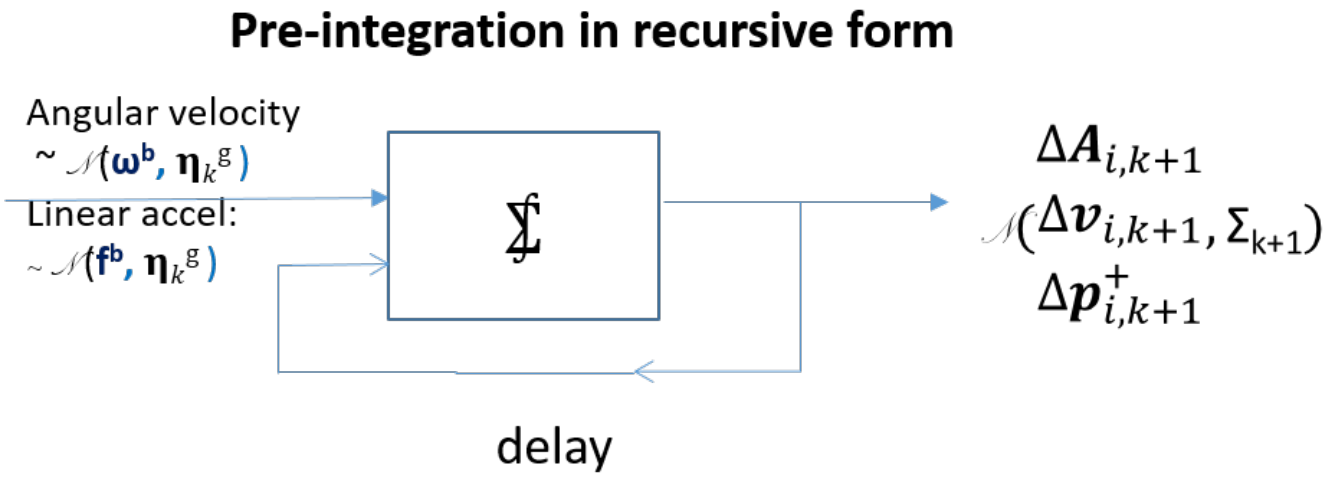


Figure 3: Pre-Integration

3.2 Inertial delta Σ and Jacobian for Bias

Now, it should be easy to compute inertial delta's uncertainty and Jacobian using the newly defined Algorithm 2. Algorithm 2 is recursive, see Figure 3 therefore the inertial delta's uncertainty and Jacobian against Bias terms can be computed recursively, starting from zero uncertainty and unity gain at beginning of preintegration. This is illustrated in algorithm 3. In each iteration stage, State covariance Σ_t is related to 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.

3.3 Feature observation model

For the features observed through the camera, the following formulas hold:

$$\mathbf{F}_{ij} = (x_{ij}, y_{ij}, z_{ij}) = R_{cj}(\mathbf{F}_{fi} - \mathbf{T}_{c0j}) \tag{3}$$
$$u_{ij} = f * x_{ij}/z_{ij} + cx_0 \tag{4}$$
$$v_{ij} = f * y_{ij}/z_{ij} + cy_0 \tag{5}$$
$$d_{ij} = z_{ij} \tag{6}$$

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

3.4 Jacobian of Inertial Delta to rest of 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

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

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

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

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

$\mathbf{f}_t^{bt1} = R_{bt}^{bt1}(\mathbf{f}_t^b - \mathbf{b}_f^{obs}) = R_{bt}^{bt1}(\mathbf{f}_t^b - \mathbf{b}_f^{est} + \delta \boldsymbol{b}_f)$

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

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

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

$\mathbf{b}_f^{obs} = \mathbf{b}_a^{est} - \delta \boldsymbol{b}_f$

$\mathbf{b}_\omega^{obs} = \mathbf{b}_\omega^{est} - \delta \boldsymbol{b}_\omega$

end for

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**

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

$\alpha = \frac{dR_{bt}^{bt1}(\mathbf{f}_t - \mathbf{b}_f)}{d\mathbf{A}_t}$

$\beta = \frac{dE_{bt}^{bt1}(\omega_t - \mathbf{b}_\omega)}{d\mathbf{A}_t}$

$$F_t = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3 \triangle t & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \alpha \triangle t & -R_{bt}^{bt1} \triangle t & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 + \beta \triangle t & \mathbf{0}_3 & -E_{bt}^{bt1} \triangle t \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}$$

$$G_t = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ R_{bt}^{bt1} \triangle t & \mathbf{0}_3 \\ \mathbf{0}_3 & E_{bt}^{bt1} \triangle t \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$$

$J_{t+1} = F_t J_t$

$\Sigma_{t+1} = F_t \Sigma_t F_t' + G_t Q_t G_t'$

end for

$J_{t1}^{t2} = J_t$

$\Sigma_{t1}^{t2} = \Sigma_t$

the jth camera pose,

$$\frac{\partial u_{ij}}{\partial \mathbf{F}_{ij}} = [f/z_{ij}, 0, -fx_{ij}/z_{ij}^2] \tag{7}$$

$$\frac{\partial v_{ij}}{\partial \mathbf{F}_{ij}} = [0, f/z_{ij}, -fy_{ij}/z_{ij}^2] \tag{8}$$

$$\frac{\partial d_{ij}}{\partial \mathbf{F}_{ij}} = [0, 0, 1] \tag{9}$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{A}_j} = R_{u2c}\frac{\partial R_j}{\partial \mathbf{A}_j}(\mathbf{F}_{i1} - \mathbf{T}_j) \tag{10}$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{T}_j} = -R_{u2c}R_j \tag{11}$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{A}_{u2c}} = \frac{\partial R_{u2c}}{\partial \mathbf{A}_{u2c}}R_j(\mathbf{F}_{i1} - R_j'\mathbf{T}_{u2c} - \mathbf{T}_j) \tag{12}$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{T}_{u2c}} = -R_{u2c} \tag{13}$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{F}_{i1}} = R_{u2c}R_j \tag{14}$$

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

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{A}_i} = \frac{\partial R_i}{\partial \mathbf{A}_i}(\mathbf{T}_{i+1} - \mathbf{T}_i - \mathbf{v}_i\triangle t - \frac{1}{2}\mathbf{g}(\triangle t)^2) \tag{15}$$

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{T}_i} = -R_i \tag{16}$$

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{T}_{i+1}} = R_i \tag{17}$$

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{v}_i} = -R_i\triangle t \tag{18}$$

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{g}} = -\frac{1}{2}R_i\triangle t^2 \tag{19}$$

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{b}_f} = -\frac{\partial \triangle \mathbf{p}_t^+}{\partial \mathbf{b}_f} \tag{20}$$

$$\frac{\partial d\mathbf{p}_i^+}{\partial \mathbf{b}_\omega} = -\frac{\partial \triangle \mathbf{p}_t^+}{\partial \mathbf{b}_\omega} \tag{21}$$

For $d\mathrm{v}_i$,

$$\frac{\partial d\mathbf{v}_i}{\partial R_i} = \frac{\partial R_i}{\partial \mathbf{A}_i}(\mathbf{v}_{i+1} - \mathbf{v}_i - \mathbf{g}\triangle t) \tag{22}$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{v}_i} = -R_i \tag{23}$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{v}_{i+1}} = R_i \tag{24}$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{g}} = -R_i\triangle t \tag{25}$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{b}_f} = -\frac{\partial \triangle \mathbf{v}_t}{\partial \mathbf{b}_f} \tag{26}$$

$$\frac{\partial d\mathbf{v}_i}{\partial \mathbf{b}_\omega} = -\frac{\partial \triangle \mathbf{v}_t}{\partial \mathbf{b}_\omega} \tag{27}$$

For $d\mathbf{A}_i$,

$$\frac{\partial \triangle \mathbf{A}_i}{\partial \mathbf{A}_i} = \frac{\partial fn}{\partial R} R_{i+1} \frac{\partial R_i}{\partial \mathbf{A}_i} \tag{28}$$

$$\frac{\partial \triangle \mathbf{A}_i}{\partial \mathbf{A}_{i+1}} = \frac{\partial fn}{\partial R} \frac{\partial R_{i+1}}{\partial \mathbf{A}_{i+1}} R_1 \tag{29}$$

$$\frac{\partial \triangle \mathbf{A}_i}{\partial \mathbf{b}_\omega} = -\frac{\partial \triangle \mathbf{A}_t}{\partial \mathbf{b}_\omega} \tag{30}$$

4 Experimental results

4.1 Comparison of Naive VIN vs Pre-Integration

Refer to table 1 for a comparison of Naive VIN vs Pre-integration. The later is shown to be much more efficient.

Table 1: Caption for the table.

Num image frames		Naive VIN	Pre-Integration
10	Total time	1644.2	17.1
	δX	4.76	17.6
150	Total time	NIL	171.4
	δX	NIL	11.3

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

4.2 Incremental implementation of Pre-Integration

Direct initialization may lead to errors for long tests. This issue can be resolved by incremental introduction of new camera poses.

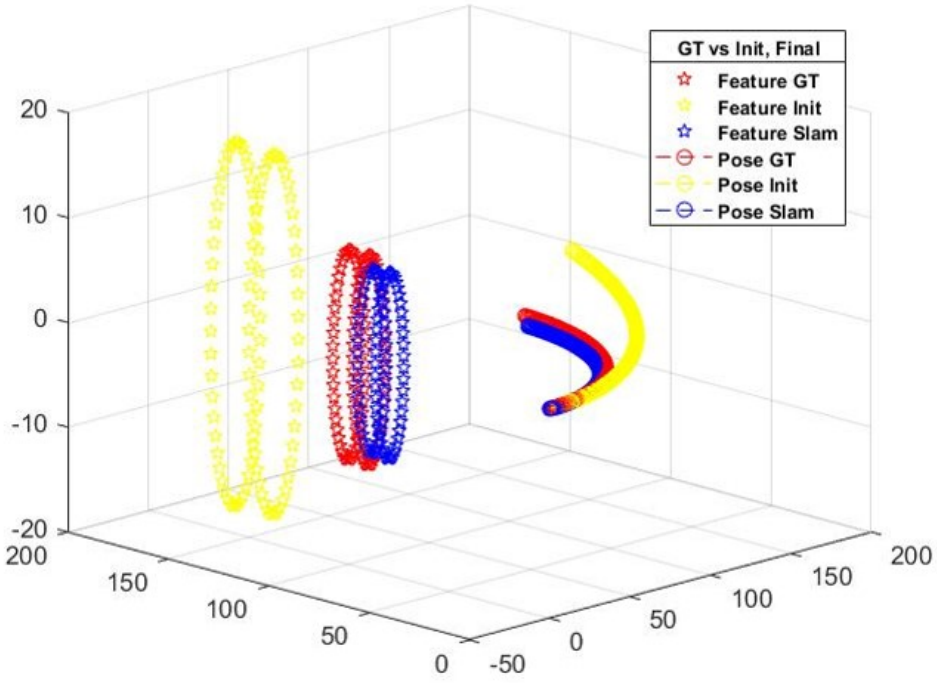


Figure 4: SimuNpose Naive vs Preintegration VIN

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

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

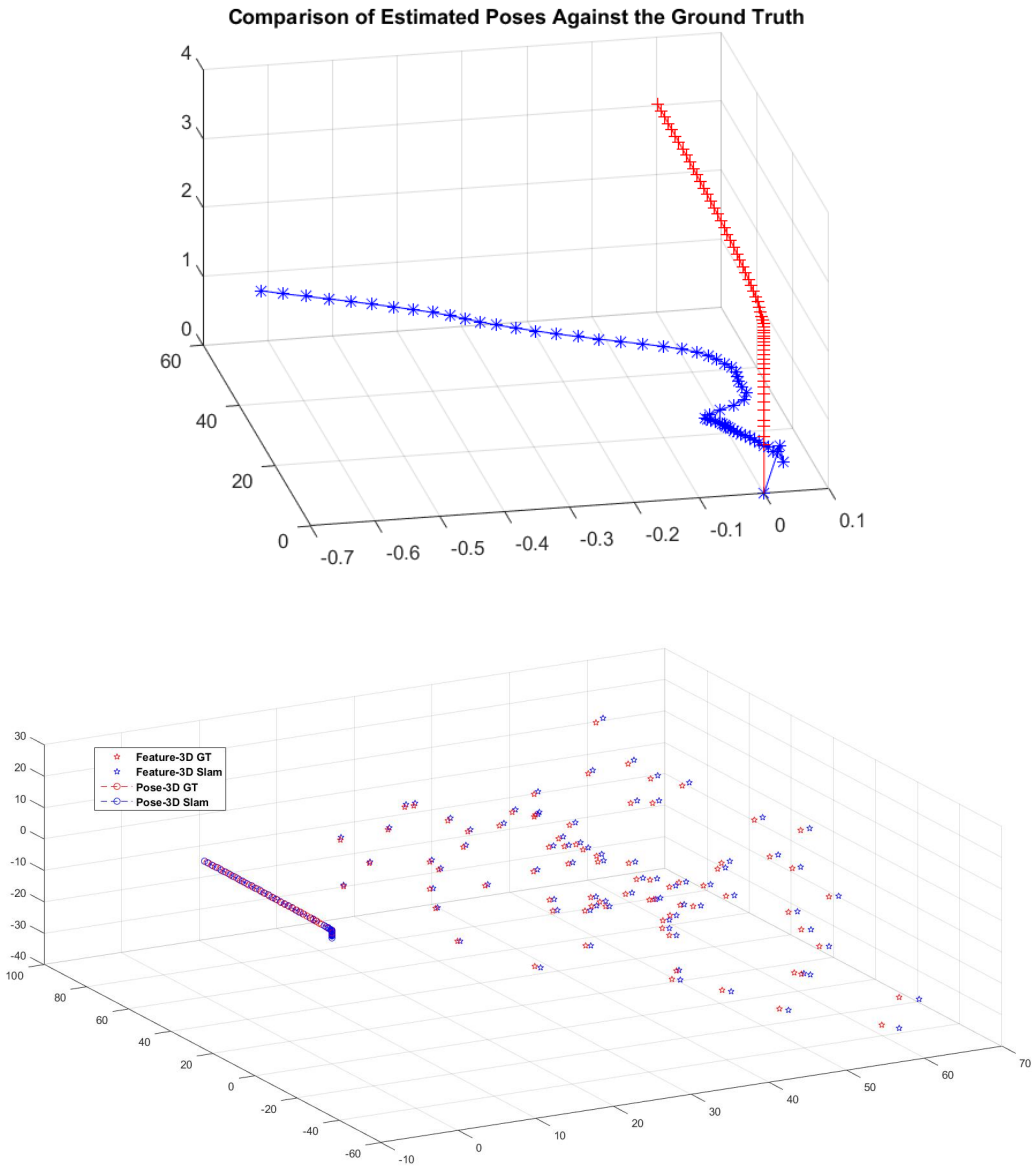


Figure 5: *a*) One off initialization , *b*) Incremental initialization

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:

$$\mathbf{R}_i^c = \mathbf{R}_{u2c}\mathbf{R}_i^u$$
$$\mathbf{T}_i^c = \mathbf{T}_i^u + \mathbf{R}_i^{u\prime}\mathbf{T}_{u2c}$$

B Naive VIN

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

$$\omega_{ij} = E_{ij}(\mathbf{A}_{(i+1)j} - \mathbf{A}_{ij})/\triangle t + \mathbf{b}_w \tag{31}$$
$$\mathbf{a}_{ij} = R_{ij}((\mathbf{v}_{(i+1)j} - \mathbf{v}_{ij})/\triangle t - \mathbf{g}) + \mathbf{b}_f \tag{32}$$
$$\mathbf{bZeros} = \mathbf{T}_{(i+1)j} - \mathbf{T}_{ij} - \mathbf{v}_{ij}\triangle t \tag{33}$$

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:

$$\begin{aligned}
H(\mathbf{x}) &= (H_{camera}(\mathbf{x}), H_{IMUraw}(\mathbf{x}), H_{Tv}(\mathbf{x})) \\
&= (\overbrace{u_{11}, v_{11}, ..., u_{M1}, v_{M1}, ..., u_{1N}, v_{1N}, ..., u_{MN}, v_{MN}}^{M\times N\times 2}, \\
&\quad \overbrace{\omega\mathbf{a}_{01}, \omega\mathbf{a}_{11}, ..., \omega\mathbf{a}_{(K-1)1}, ..., \omega\mathbf{a}_{0(N-1)}, \omega\mathbf{a}_{1(N-1)}, ..., \omega\mathbf{a}_{(K-1)(N-1)}}^{K\times(N-1)\times 6}, \\
&\quad \overbrace{\mathbf{T}_2 - \mathbf{T}_1 - \mathbf{v}_1\triangle t, \mathbf{T}_3 - \mathbf{T}_2 - \mathbf{v}_2\triangle t, ..., \mathbf{T}_{(N-1)K+1} - \mathbf{T}_{(N-1)K} - \mathbf{v}_{(N-1)K}\triangle t}^{K\times(N-1)\times 3}) \\
&= (\overbrace{f * x_{11}/z_{11} + cx_0, f * y_{11}/z_{11} + cy_0, ..., f * x_{MN}/z_{MN} + cx_0, f * y_{MN}/z_{MN} + cy_0}^{M\times N\times 2}, , \\
&\quad \overbrace{E_i * (\mathbf{A}_{11} - \mathbf{A}_{01})/\triangle t + \mathbf{b}_w, ..., R_{(K-1)(N-1)} * ((\mathbf{v}_{0N} - \mathbf{v}_{(K-1)(N-1)})/\triangle t - \mathbf{g}) + \mathbf{b}_f}^{K\times(N-1)\times 6}, \\
&\quad \overbrace{\mathbf{T}_2 - \mathbf{T}_1 - \mathbf{v}_1\triangle t, \mathbf{T}_3 - \mathbf{T}_2 - \mathbf{v}_2\triangle t, ..., \mathbf{T}_{(N-1)K+1} - \mathbf{T}_{(N-1)K} - \mathbf{v}_{(N-1)K}\triangle t}^{K\times(N-1)\times 3})
\end{aligned} \tag{34}$$

B.0.1 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,

$$\frac{\partial u_{ij}}{\partial \mathbf{P}_{ij}} = [f/z_{ij}, 0, -fx_{ij}/z_{ij}^2] \tag{35}$$

$$\frac{\partial v_{ij}}{\partial \mathbf{P}_{ij}} = [0, f/z_{ij}, -fy_{ij}/z_{ij}^2] \tag{36}$$

$$\frac{\partial d_{ij}}{\partial \mathbf{P}_{ij}} = [0, 0, 1] \tag{37}$$

$$\frac{\partial \mathbf{P}_{ij}}{\partial \mathbf{A}_{0j}} = R_{u2c}\frac{\partial R_{0j}}{\partial \mathbf{A}_{0j}}(\mathbf{P}_{i1} - \mathbf{T}_{0j}) \tag{38}$$

$$\frac{\partial \mathbf{P}_{ij}}{\partial \mathbf{T}_{0j}} = -R_{u2c}R_{0j} \tag{39}$$

$$\frac{\partial \mathbf{P}_{ij}}{\partial \mathbf{A}_{u2c}} = \frac{\partial R_{u2c}}{\partial \mathbf{A}_{u2c}}R_{0j}(\mathbf{P}_{i1} - R'_{0j}\mathbf{T}_{u2c} - \mathbf{T}_{0j}) \tag{40}$$

$$\frac{\partial \mathbf{P}_{ij}}{\partial \mathbf{T}_{u2c}} = -R_{u2c} \tag{41}$$

$$\frac{\partial \mathbf{P}_{ij}}{\partial \mathbf{P}_{fi}} = R_{u2c}R_{0j} \tag{42}$$

For ω_{ij},

$$\begin{aligned}\frac{\partial \omega_{ij}}{\partial \mathbf{A}_{ij}} &= \frac{\partial E_{ij}}{\partial \mathbf{A}_{ij}}(\mathbf{A}_{(i+1)j} - \mathbf{A}_{ij})/\triangle t + E_{ij}(-\frac{\partial \mathbf{A}_{ij}}{\partial \mathbf{A}_{ij}})/\triangle t \\ &= (\frac{\partial E_{ij}}{\partial \mathbf{A}_{ij}}(\mathbf{A}_{(i+1)j} - \mathbf{A}_{ij}) - E_{ij})/\triangle t \end{aligned} \tag{43}$$

$$\frac{\partial E_{ij}}{\partial \mathbf{A}_{ij}} = [\frac{\partial E_{ij}}{\partial \alpha_{ij}}, \frac{\partial E_{ij}}{\partial \beta_{ij}}, \frac{\partial E_{ij}}{\partial \gamma_{ij}}] \tag{44}$$

$$\frac{\partial \omega_{ij}}{\partial \mathbf{A}_{(i+1)j}} = E_{ij}\frac{\partial \mathbf{A}_{(i+1)j}}{\partial \mathbf{A}_{(i+1)j}}/\triangle t \tag{45}$$

$$= E_{ij}/\triangle t \tag{46}$$

$$\frac{\partial \omega_{ij}}{\partial b_\omega} = I_{3\times 3} \tag{47}$$

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

$$\frac{\partial \mathbf{a}_{ij}}{\partial \mathbf{A}_{ij}} = \frac{\partial R_{ij}}{\mathbf{A}_{ij}}((\mathbf{v}_{i+1} - \mathbf{v}_i)/\triangle t - \mathbf{g}) \tag{48}$$

$$\frac{\partial R_{ij}}{\partial \mathbf{A}_{ij}} = [\frac{\partial R_{ij}}{\partial \alpha_{ij}}, \frac{\partial R_{ij}}{\partial \beta_{ij}}, \frac{\partial R_{ij}}{\partial \gamma_{ij}}] \tag{49}$$

$$\frac{\partial \mathbf{a}_{ij}}{\partial \mathbf{v}_{(i+1)j}} = R_{ij}/\triangle t \tag{50}$$

$$\frac{\partial \mathbf{a}_{ij}}{\partial \mathbf{v}_{ij}} = -R_{ij}/\triangle t \tag{51}$$

$$\frac{\partial \mathbf{a}_{ij}}{\partial \mathbf{g}} = -R_{ij} \tag{52}$$

$$\frac{\partial \mathbf{a}_{ij}}{\partial b_f} = I_{3\times 3} \tag{53}$$

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

$$\frac{\partial \mathbf{bZeros}_{ij}}{\partial \mathbf{T}_{(i+1)j}} = I_{3\times 3} \tag{54}$$

$$\frac{\partial \mathbf{bZeros}_{ij}}{\partial \mathbf{T}_{ij}} = -I_{3\times 3} \tag{55}$$

$$\frac{\partial \mathbf{bZeros}_{ij}}{\partial \mathbf{v}_{ij}} = -I_{3\times 3}\triangle t \tag{56}$$

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.