2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Direct Visual-Inertial Navigation with Analytical Preintegration

Kevin Eckenhoff, Patrick Geneva, and Guoquan Huang

Abstract— **Recent advancements in the performance and affordability of cameras and inertial measurement units (IMUs) have caused demand for efficient, accurate visual-inertial navigation solutions. In this paper, we present a system for the fusion of preintegrated inertial measurements with highly informative direct alignment of images. In particular, our preintegration theory is based on *closed-form* solutions of the continuous-time IMU kinematic model, instead of discrete time. This allows for more accurate computation of preintegrated measurements and their uncertainty as well as bias Jacobians. These measurements are fused via graph-based methods with relative pose constraints obtained from direct image alignment from a stereo platform. The proposed system is validated on publicly-available real-world datasets.**

I. Introduction

Visual-inertial navigation systems (VINS) have become very popular in recent years due to their enormous potential in robotics. These low-cost and light-weight sensors are still highly informative, and are therefore ideal for providing navigation solutions for Unmanned Aerial Vehicles (UAVs) and mobile/wearable devices. The accuracy of these solutions is imperative for real-world applications, especially in the case of autonomous systems which require accurate information of their state for decision making. As such, a great deal of efforts have been placed in estimation via the processing of camera and IMU measurements [1], [2].

Traditionally, estimation algorithms for VINS have been based on the Extended Kalman Filter (EKF) [3], [4], where incoming measurements are linearized and processed sequentially, without the ability to correct past states. In contrast, for Simultaneous Localization and Mapping (SLAM), graph-based optimization techniques [5] process all measurements at once to estimate an entire trajectory. However, it has historically been difficult to incorporate IMU readings into a factor-graph due to the nature of acceleration measurements and biases. To deal with these issues, Lupton et al. [6] intoduced the theory of *preintegration*, where IMU measurements are processed in the IMU local frame of reference, while dependency of these measurements on biases is removed by linearizing about the current bias estimates. This reformation of the IMU processing allows for the creation of factors for graph-based SLAM [7]. Nevertheless, current preintegration methods are based on the discretization of the state dynamics, which incurs approximation.

In this work, however, by building upon our recently developed *analytical* preintegration theory [8], we propose a graph-based direct VINS algorithm that not only analytically preintegrates the IMU measurements between keyframes based on continuous state dynamics, but also directly tracks IMU/camera motion based on dense pixel values, *without* detecting and tracking visual features. In particular, we utilize the recent strategy of direct visual odometry [9] and find relative camera pose factors by minimizing the photometric error between two images, thereby using a much larger subset of pixels and thus more information than sparse methods [7]. The IMU and camera factors are then fused in a visual-inertial graph-based optimization framework. In this way we offer a "best of both worlds" approach, leveraging state-of-the-art advances in both visual and inertial sensor technologies.

This work was partially supported by the University of Delaware College of Engineering, UD Cybersecurity Initiative, the Delaware NASA/EPSCoR Seed Grant, the NSF (IIS-1566129), and the DTRA (HDTRA1-16-1-0039).

The authors are with the Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA. Email: {keck,pgeneva,ghuang}@udel.edu

II. Related Work

In the realm of filter-based VINS, Mourikis et al. [1] introduced the multi-state constraint Kalman filter (MSCKF), where the state vector consists of the current sensor state, as well as the poses of the past n states, allowing short-term correction of those states for a smoother path. Feature measurements were extracted from images and used in the update step. Naively, these measurements could be processed by adding the 3D feature points into the state vector, which causes an unbounded computational overhead. Instead, the authors projected all the information onto a lower dimensional subspace, such that the residual only constrains the *robot* states within the window. Recently, VINS system observability and filter consistency have been extensively studied [10]. On the other hand, batch optimization methods for use in SLAM process all measurements at once by solving for the Maximum A Posteriori (MAP) estimate through nonlinear optimization [5], [11]–[13].

Lupton, et al. [6] introduced the theory of preintegration. They showed that by integrating IMU measurements in a *local* frame of reference and linearizing about the current bias estimate, a measurement connecting the start and end states of the interval could be created. Forster et al. [7] extended this work to a stable Lie group representation of SO(3), thereby gauranteeing that all rotation matrices were valid. In both of these works, discrete versions of the measurement dynamics were used, which introduces approximation during integration. Yang and Shen [14] utilized preintegration while formulating the error dynamics in a continuous fashion. However, the measurement dynamics were sampled to compute covariances and measurement means. In addition, bias Jacobians were not utilized. By contrast, our method uses continuous measurement and error dynamics to compute, in closed form, the mean, covariance, and bias Jacobians of our

preintegrated measurements. This affords our method higher accuracy than its predecessors.

Different visual tracking methods have been used in recent literature. Direct methods for monocular systems (i.e., without IMU) have been successfully implemented by Engel et al. [9], which was extended onto rolling shutter cameras [16]. Direct methods with inertial preintegration have been studied previously [15], [17], which, while having sophisticated visual front-ends, were still based on the discrete version of preintegration. Lastly, the direct visual-inertial fusion of [18] fixed all past states and optimized only over the active state.

III. Batch Optimization

Given a set of measurements $\mathcal{Z}$, batch optimization seeks the estimation of state parameters, $\mathbf{x}$, to maximize the conditional distribution $p(\mathcal{Z}|\mathbf{x})$. This process is known as maximum likelihood estimation (MLE). The problem can be viewed as a graph, where nodes represent parameters to be estimated, while edges represent measurements relating adjacent nodes (see Figure 1). Under the assumption of Gaussian noise and independence of measurements, this optimization can be written as the following Nonlinear Least Squares (NLS) problem [5]:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_i ||\mathbf{r}_i(\mathbf{x})||^2_{\mathbf{W}_i} \tag{1}$$

We define $\mathbf{r}_i(\mathbf{x})$ as the residual of measurement i, $\mathbf{W}_i$ is the information matrix associated with the noise corrupting that measurement, and $||\mathbf{v}||^2_{\mathbf{A}} = \mathbf{v}^\top \mathbf{A}\mathbf{v}$ is the energy norm. This cost function is minimized using the Gauss-Newton method of iterative linearization of the residual about the current estimate. State variables may be restricted to a manifold (such as SO(3)) [19], and we therefore expand the state about the current estimate $\mathbf{x} = \hat{\mathbf{x}} \boxplus \Delta\mathbf{x}$. Here $\mathbf{x}$ and $\hat{\mathbf{x}}$ are elements of the manifold corresponding to the true and approximate states, $\Delta\mathbf{x}$ is a correction vector, and $\boxplus$ is an operation that maps a manifold element and a correction vector to a new manifold element. For the case of states in a vector space (such as position and velocity), this operation is simply vector addition. For unit quaternions (in JPL convention [20]), this operation is typically approximated as $\bar{q} \approx \begin{bmatrix} \frac{\Delta\boldsymbol{\theta}}{2} \\ 1 \end{bmatrix} \otimes \hat{\bar{q}}$, with $\otimes$ indicating quaternion multiplication [20]. The NLS problem (1) can then be expanded about this correction vector, and the optimization is reformatted to finding the optimal correction:

$$\Delta\mathbf{x}^* = \arg\min_{\Delta\mathbf{x}} \sum_i ||\mathbf{r}_i(\hat{\mathbf{x}} \boxplus \Delta\mathbf{x})||^2_{\mathbf{W}_i} \tag{2}$$

$$\approx \arg\min_{\Delta\mathbf{x}} \sum_i ||\mathbf{r}_i(\hat{\mathbf{x}}) + \mathbf{J}_i\Delta\mathbf{x}||^2_{\mathbf{W}_i} \tag{3}$$

where $\mathbf{J}_i = \frac{\partial \mathbf{r}_i(\hat{\mathbf{x}} \boxplus \Delta\mathbf{x})}{\partial \Delta\mathbf{x}}|_{\Delta\mathbf{x}=\mathbf{0}}$ is the Jacobian of the residual with respect to the error state. By taking the gradient of (3), the correction vector can be found in closed form:

$$\Delta\mathbf{x}^* = -\left(\sum_i \mathbf{J}_i^\top \mathbf{W}_i \mathbf{J}_i\right)^{-1} \left(\sum_i \mathbf{J}_i^\top \mathbf{W}_i \mathbf{r}_i\left(\hat{\mathbf{x}}\right)\right) \tag{4}$$

With this, the state estimate is updated, $\hat{\mathbf{x}}^+ = \hat{\mathbf{x}}^- \boxplus \Delta\mathbf{x}^*$. This process is iterated until convergence. After optimization,

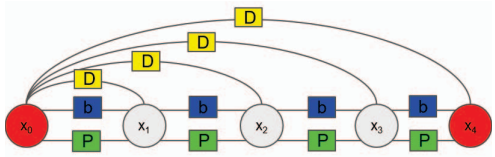


Fig. 1: Factor-graph representation used in the global optimization. Every state (denoted $\mathbf{x}_i$) is connected to the next state by a preintegration factor (green) and a bias drift factor (blue). Keyframe states (shown in red) are also connected to other states in the graph by direct alignment factors (yellow).

$\boldsymbol{\Sigma} = \left(\sum_i \mathbf{J}_i^\top \mathbf{W}_i \mathbf{J}_i\right)^{-1}$ serves as an approximation for the covariance of the zero-mean estimate error:

$$\mathbf{x} = \hat{\mathbf{x}} \boxplus \Delta\mathbf{x}, \qquad \Delta\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \tag{5}$$

For this work, the state of our sensor suite at step k and the corresponding correction vector can be written as:

$$\mathbf{x}_k = \begin{bmatrix} {}^k_G\bar{q}^\top & {}^G\mathbf{v}_k^\top & {}^G\mathbf{p}_k^\top \end{bmatrix}^\top \tag{6}$$

$$\Delta\mathbf{x}_k = \begin{bmatrix} \Delta^k\boldsymbol{\theta}_G^\top & \Delta^G\mathbf{v}_k^\top & \Delta^G\mathbf{p}_k^\top \end{bmatrix}^\top \tag{7}$$

${}^k_G\bar{q}$ is the JPL quaternion associated with a rotation from the global frame to the local frame, and ${}^G\mathbf{v}_k$ and ${}^G\mathbf{p}_k$ are the velocity and position in the global frame.

IV. Analytical Preintegration Constraints

Inertial measurements are processed in the form of local gyro and acceleration measurements, denoted ${}^\tau\boldsymbol{\omega}_m$ and ${}^\tau\mathbf{a}_m$, respectively. These measurements are received at step τ and are related to the true angular velocity, true acceleration, biases, and noises as follows [20]:

$${}^\tau\mathbf{a}_m = {}^\tau\mathbf{a} + {}^\tau_G\mathbf{R}\,{}^G\mathbf{g} + \mathbf{b}_a + \mathbf{n}_a \tag{8}$$

$${}^\tau\boldsymbol{\omega}_m = {}^\tau\boldsymbol{\omega} + \mathbf{b}_w + \mathbf{n}_w \tag{9}$$

$$\dot{\mathbf{b}}_a = \mathbf{n}_{ba} \tag{10}$$

$$\dot{\mathbf{b}}_w = \mathbf{n}_{bw} \tag{11}$$

We define ${}^\tau\boldsymbol{\omega}$ and ${}^\tau\mathbf{a}$ as the true angular velocity and acceleration, $\mathbf{b}_a$ and $\mathbf{b}_w$ as the acceleration and gyro biases, $\mathbf{n}_i \sim \mathcal{N}\left(\mathbf{0}_{3\times3}, \sigma_i^2\mathbf{I}_{3\times3}\right)$ as the process noises, ${}^\tau_G\mathbf{R}$ as the rotation matrix which rotates a vector from the global frame to the τ frame, and ${}^G\mathbf{g}$ as the gravity vector. In this work we use the convention of ${}^G\mathbf{g} \approx \begin{bmatrix} 0 & 0 & 9.81 \end{bmatrix}$. The biases that corrupt the measurements also need to be estimated, so we augment our state vector (6):

$$\mathbf{x}_k = \begin{bmatrix} {}^k_G\bar{q}^\top & {}^G\mathbf{v}_k^\top & {}^G\mathbf{p}_k^\top & \mathbf{b}_{a_k}^\top & \mathbf{b}_{w_k}^\top \end{bmatrix}^\top \tag{12}$$

Given a sequence of IMU readings, collected at steps τ between two states, $k \leq \tau \leq k+1$, we can integrate across these measurements to connect the beginning and end states

of the preintegration interval of total time ΔT [6], [21]:

$$
\begin{aligned}
{}^{G}\mathbf{p}_{k+1} &= {}^{G}\mathbf{p}_{k} + {}^{G}\mathbf{v}_{k}\Delta T - \frac{1}{2}{}^{G}\mathbf{g}\Delta T^{2} \\
&\quad + {}^{G}_{k}\mathbf{R}\underbrace{\int_{t_k}^{t_{k+1}}\int_{t_k}^{s}{}^{k}_{\tau}\mathbf{R}\left({}^{\tau}\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a\right)d\tau ds}_{{}^{k}\boldsymbol{\alpha}_{k+1}} \\
&=: {}^{G}\mathbf{p}_{k} + {}^{G}\mathbf{v}_{k}\Delta T - \frac{1}{2}{}^{G}\mathbf{g}\Delta T^{2} + {}^{G}_{k}\mathbf{R}\,{}^{k}\boldsymbol{\alpha}_{k+1} \quad (13) \\
{}^{G}\mathbf{v}_{k+1} &= {}^{G}\mathbf{v}_{k} - {}^{G}\mathbf{g}\Delta T \\
&\quad + {}^{G}_{k}\mathbf{R}\underbrace{\int_{t_k}^{t_{k+1}}{}^{k}_{\tau}\mathbf{R}\left({}^{\tau}\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a\right)d\tau}_{{}^{k}\boldsymbol{\beta}_{k+1}} \\
&=: {}^{G}\mathbf{v}_{k} - {}^{G}\mathbf{g}\Delta T + {}^{G}_{k}\mathbf{R}\,{}^{k}\boldsymbol{\beta}_{k+1} \quad (14) \\
{}^{k+1}_{G}\mathbf{R} &= {}^{k+1}_{k}\mathbf{R}\,{}^{k}_{G}\mathbf{R} \quad (15)
\end{aligned}
$$

We have collected the terms which are independent from the poses and velocities at the kth and $(k+1)$th steps as three preintegration measurements. ${}^{k}\boldsymbol{\alpha}_{k+1}$ and ${}^{k}\boldsymbol{\beta}_{k+1}$ describe how the positions and velocities evolve over a interval, while ${}^{k+1}_{k}\mathbf{R}$ is the relative rotation between the two states. By rearranging the dynamic equations, we derive the measurements as functions of the state variables at steps k and $k+1$:

$$
\begin{aligned}
{}^{k}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_{k+1} - {}^{G}\mathbf{p}_{k} - {}^{G}\mathbf{v}_{k}\Delta T + \frac{1}{2}{}^{G}\mathbf{g}\Delta T^{2}\right) &= {}^{k}\boldsymbol{\alpha}_{k+1}\left(\mathbf{b}_a, \mathbf{b}_w\right) \\
{}^{k}_{G}\mathbf{R}\left({}^{G}\mathbf{v}_{k+1} - {}^{G}\mathbf{v}_{k} + {}^{G}\mathbf{g}\Delta T\right) &= {}^{k}\boldsymbol{\beta}_{k+1}\left(\mathbf{b}_a, \mathbf{b}_w\right) \\
{}^{k+1}_{G}\mathbf{R}\,{}^{k}_{G}\mathbf{R}^{\top} &= {}^{k+1}_{k}\mathbf{R}\left(\mathbf{b}_w\right)
\end{aligned}
$$

The dependencies of the measurements on the biases are removed via expansions about the current linearization point for these biases, $\bar{\mathbf{b}}_a$ and $\bar{\mathbf{b}}_w$ [6]. These biases are approximated as remaining constant over the preintegration interval. By defining $\Delta\mathbf{b} = \mathbf{b} - \bar{\mathbf{b}}$, we have:

$$
\begin{aligned}
&{}^{k}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_{k+1} - {}^{G}\mathbf{p}_{k} - {}^{G}\mathbf{v}_{k}\Delta T + \frac{1}{2}{}^{G}\mathbf{g}\Delta T^{2}\right) \simeq \quad (16) \\
&\qquad {}^{k}\boldsymbol{\alpha}_{k+1}\left(\bar{\mathbf{b}}_a, \bar{\mathbf{b}}_w\right) + \frac{\partial\boldsymbol{\alpha}}{\partial\mathbf{b}_a}\Big|_{\bar{\mathbf{b}}_a}\Delta\mathbf{b}_a + \frac{\partial\boldsymbol{\alpha}}{\partial\mathbf{b}_w}\Big|_{\bar{\mathbf{b}}_w}\Delta\mathbf{b}_w \\
&{}^{k}_{G}\mathbf{R}\left({}^{G}\mathbf{v}_{k+1} - {}^{G}\mathbf{v}_{k} + {}^{G}\mathbf{g}\Delta T\right) \simeq \quad (17) \\
&\qquad {}^{k}\boldsymbol{\beta}_{k+1}\left(\bar{\mathbf{b}}_a, \bar{\mathbf{b}}_w\right) + \frac{\partial\boldsymbol{\beta}}{\partial\mathbf{b}_a}\Big|_{\bar{\mathbf{b}}_a}\Delta\mathbf{b}_a + \frac{\partial\boldsymbol{\beta}}{\partial\mathbf{b}_w}\Big|_{\bar{\mathbf{b}}_w}\Delta\mathbf{b}_w \\
&{}^{k+1}_{G}\mathbf{R}\,{}^{k}_{G}\mathbf{R}^{\top} \simeq \mathbf{R}\left(\frac{\partial\boldsymbol{R}}{\partial\mathbf{b}_w}\Big|_{\bar{\mathbf{b}}_w}\Delta\mathbf{b}_w\right){}^{k+1}_{k}\mathbf{R}\left(\bar{\mathbf{b}}_w\right) \quad (18)
\end{aligned}
$$

(16) and (17) are simple Taylor series expansions for the case of our ${}^{k}\boldsymbol{\alpha}_{k+1}$ and ${}^{k}\boldsymbol{\beta}_{k+1}$ measurements, while (18) models an induced extra rotation [7]. Note that because the preintegrated measurements are performed using only IMU data and current bias estimates, preintegration avoids costly reintegration as required by other IMU processing techniques [22]. With these definitions we can create a factor between the start and end states of the preintegration window (k and $k+1$ respectively) for use in batch optimization. The residual associated with this edge can then be written as:

$$
\begin{aligned}
\mathbf{r}_{k+1} &= \begin{bmatrix}\Delta^{k}\boldsymbol{\alpha}_{k+1} \\ \Delta^{k}\boldsymbol{\beta}_{k+1} \\ \Delta^{k+1}\boldsymbol{\theta}_{k}\end{bmatrix} \\
\Delta^{k}\boldsymbol{\alpha}_{k+1} &= {}^{k}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_{k+1} - {}^{G}\mathbf{p}_{k} - {}^{G}\mathbf{v}_{k}\Delta T + \frac{1}{2}{}^{G}\mathbf{g}\Delta T^{2}\right) \\
&\quad - \mathbf{J}_{\alpha}(\mathbf{b}_w - \bar{\mathbf{b}}_w) - \mathbf{H}_{\alpha}(\mathbf{b}_a - \bar{\mathbf{b}}_a) - {}^{k}\breve{\boldsymbol{\alpha}}_{k+1} \\
\Delta^{k}\boldsymbol{\beta}_{k+1} &= {}^{k}_{G}\mathbf{R}\left({}^{G}\mathbf{v}_{k+1} - {}^{G}\mathbf{v}_{k} + {}^{G}\mathbf{g}\Delta T\right) \\
&\quad - \mathbf{J}_{\beta}(\mathbf{b}_w - \bar{\mathbf{b}}_w) - \mathbf{H}_{\beta}(\mathbf{b}_a - \bar{\mathbf{b}}_a) - {}^{k}\breve{\boldsymbol{\beta}}_{k+1} \\
\Delta^{k+1}\boldsymbol{\theta}_{k} &= 2\mathrm{vec}\left({}^{k+1}_{G}\bar{q} \otimes {}^{k}_{G}\bar{q}^{-1} \otimes {}^{k+1}_{k}\breve{\bar{q}}^{-1} \otimes \mathbf{quat}(\mathrm{Exp}(-\lfloor\mathbf{J}_{q}(\mathbf{b}_w - \bar{\mathbf{b}}_w)\times\rfloor))\right)
\end{aligned} \quad (19)
$$

Here we define preintegration measurement means, ${}^{k}\breve{\boldsymbol{\alpha}}_{k+1}$, ${}^{k}\breve{\boldsymbol{\beta}}_{k+1}$, and ${}^{k+1}_{k}\breve{\bar{q}}$. For ease of notation we have also defined $\mathbf{J}_{\alpha} = \frac{\partial^{k}\boldsymbol{\alpha}_{k+1}}{\partial\mathbf{b}_w}, \mathbf{J}_{\beta} = \frac{\partial^{k}\boldsymbol{\beta}_{k+1}}{\partial\mathbf{b}_w}, \mathbf{H}_{\alpha} = \frac{\partial^{k}\boldsymbol{\alpha}_{k+1}}{\partial\mathbf{b}_a}$, and $\mathbf{H}_{\beta} = \frac{\partial^{k}\boldsymbol{\beta}_{k+1}}{\partial\mathbf{b}_a}$. $\mathbf{J}_q$ is a matrix which describes how the relative rotation changes with a change of $\mathbf{b}_w$. $\mathrm{Exp}(\cdot)$ refers to the matrix exponential, $\mathbf{quat}(\cdot)$ returns the quaternion associated with the argument rotation matrix, and $\mathrm{vec}(\cdot)$ returns the vector consisting of the first three elements of $\bar{q}$. Each of these bias Jacobians can be computed incrementally as new IMU measurements arrive, with *closed-form* expressions derived in [8], [23] (along with derivations of the *measurement* Jacobians for use in batch optimization). In addition to the preintegration factors described, states are also linked by factors constraining the drift of the biases across the interval [7] (see Figure 1).

A. Preintegration Mean

The preintegration measurement means, ${}^{k}\breve{\boldsymbol{\alpha}}_{k+1}$ and ${}^{k}\breve{\boldsymbol{\beta}}_{k+1}$, and ${}^{k+1}_{k}\breve{\bar{q}}$ are computed using the inertial measurements and bias linearization points. ${}^{k+1}_{k}\breve{\bar{q}}$ can be found using successive applications of the zeroth order quaternion integrator [20]. Based on the definitions of ${}^{k}\boldsymbol{\alpha}_{k+1}$ and ${}^{k}\boldsymbol{\beta}_{k+1}$, we have the following dynamics at every step τ:

$$
{}^{k}\dot{\boldsymbol{\alpha}}_{\tau} = {}^{k}\boldsymbol{\beta}_{\tau} \quad (20)
$$

$$
{}^{k}\dot{\boldsymbol{\beta}}_{\tau} = {}^{k}_{\tau}\mathbf{R}\left({}^{\tau}\mathbf{a}_m - \bar{\mathbf{b}}_a - \mathbf{n}_a\right) \quad (21)
$$

The rotation (quaternion) dynamics is given by [20]:

$$
{}^{\tau}_{k}\dot{\bar{q}} = \frac{1}{2}\boldsymbol{\Omega}({}^{\tau}\boldsymbol{\omega}_m - \bar{\mathbf{b}}_w - \mathbf{n}_w){}^{\tau}_{k}\bar{q} \quad (22)
$$

where $\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix}-\lfloor\boldsymbol{\omega}\times\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{\top} & 0\end{bmatrix}$ and $\lfloor\boldsymbol{\omega}\times\rfloor = \begin{bmatrix}0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0\end{bmatrix}$. From these definitions, rather than discretizing the measurement dynamics, we formulate the following linear system that describes the *continuous* evolution of the estimated states by taking the (approximate) expectation of the dynamic equations:

$$
\begin{bmatrix}{}^{k}\dot{\breve{\boldsymbol{\alpha}}}_{\tau} \\ {}^{k}\dot{\breve{\boldsymbol{\beta}}}_{\tau}\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{I}_{3\times3} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}{}^{k}\breve{\boldsymbol{\alpha}}_{\tau} \\ {}^{k}\breve{\boldsymbol{\beta}}_{\tau}\end{bmatrix} + \begin{bmatrix}\mathbf{0} \\ {}^{k}_{\tau}\breve{\mathbf{R}}\end{bmatrix}\left({}^{\tau}\mathbf{a}_m - \bar{\mathbf{b}}_a\right) \quad (23)
$$

Given a sampled ${}^{\tau}\boldsymbol{a}_m$ and ${}^{\tau}\boldsymbol{\omega}_m$ between times $[t_\tau, t_{\tau+1}]$, we solve this linear system to obtain the updated preintegration

means. By defining $\hat{\boldsymbol{\omega}} = {}^{\tau}\boldsymbol{\omega}_m - \bar{\mathbf{b}}_w$, $\hat{\boldsymbol{a}} = {}^{\tau}\boldsymbol{a}_m - \bar{\mathbf{b}}_a$, and $\Delta t = t_{\tau+1} - t_\tau$, we can integrate (23) using standard linear system techniques to obtain ${}^{k}\breve{\boldsymbol{\alpha}}_{\tau+1}$ and ${}^{k}\breve{\boldsymbol{\beta}}_{\tau+1}$:

$$\begin{bmatrix} {}^{k}\breve{\boldsymbol{\alpha}}_{\tau+1} \\ {}^{k}\breve{\boldsymbol{\beta}}_{\tau+1} \end{bmatrix} = \begin{bmatrix} {}^{k}\breve{\boldsymbol{\alpha}}_{\tau} + {}^{k}\breve{\boldsymbol{\beta}}_{\tau}\Delta t \\ {}^{k}\breve{\boldsymbol{\beta}}_{\tau} \end{bmatrix} + \begin{bmatrix} {}^{k}_{\tau+1}\breve{\mathbf{R}}(\frac{(\Delta t^2)}{2}\mathbf{I}_{3\times3} + \frac{|\hat{\boldsymbol{\omega}}|\Delta t\cos(|\hat{\boldsymbol{\omega}}|\Delta t) - \sin(|\hat{\boldsymbol{\omega}}|\Delta t)}{|\hat{\boldsymbol{\omega}}|^3}\lfloor\hat{\boldsymbol{\omega}}\times\rfloor \\ + \frac{(|\hat{\boldsymbol{\omega}}|\Delta t)^2 - 2\cos(|\hat{\boldsymbol{\omega}}|\Delta t) - 2(|\hat{\boldsymbol{\omega}}|\Delta t)\sin(|\hat{\boldsymbol{\omega}}|\Delta t) + 2}{2|\hat{\boldsymbol{\omega}}|^4}\lfloor\hat{\boldsymbol{\omega}}\times\rfloor^2)(\hat{\mathbf{a}}) \\ {}^{k}_{\tau+1}\breve{\mathbf{R}}(\Delta t\mathbf{I}_{3\times3} - \frac{1-\cos(|\hat{\boldsymbol{\omega}}|(\Delta t))}{|\hat{\boldsymbol{\omega}}|^2}\lfloor\hat{\boldsymbol{\omega}}\times\rfloor \\ + \frac{(|\hat{\boldsymbol{\omega}}|\Delta t) - \sin(|\boldsymbol{\omega}|\Delta t)}{|\hat{\boldsymbol{\omega}}|^3}\lfloor\hat{\boldsymbol{\omega}}\times\rfloor^2)(\hat{\mathbf{a}}) \end{bmatrix} \tag{24}$$

Note that this expression is evaluated every time an IMU measurement is received during the preintegration interval. At the end of this interval, the total preintegrated measurements will have been computed. These *closed-form* expressions allow the derivation of the bias Jacobians associated with our ${}^{k}\breve{\boldsymbol{\alpha}}_{k+1}$ and ${}^{k}\breve{\boldsymbol{\beta}}_{k+1}$ using the derivatives of the above equations with respect to the biases [23]. For a comparison of our method vs. the current, state-of-the-art discrete preintegration [7], the reader is referred to our previous work [8], where our method was shown to offer improved performance, in particular, in highly-dynamic trajectories.

B. Preintegration Covariance

For use in batch optimization, we also need to compute the covariance, $\mathbf{P}$, of the preintegration measurements, such that for the batch optimization problem (1), $\mathbf{W}_i = \mathbf{P}^{-1}$. This covariance begins at zero, and grows as noise from the IMU is injected into the system. To compute this covariance, we begin by defining a linear-system approximation of the dynamics of the measurement errors by linearizing about the current state estimates [21]:

$$\begin{bmatrix} \Delta^k\dot{\boldsymbol{\alpha}}_\tau \\ \Delta^k\dot{\boldsymbol{\beta}}_\tau \\ \Delta^\tau\dot{\boldsymbol{\theta}}_k \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -{}^{k}_{\tau}\breve{\mathbf{R}}\lfloor\hat{\mathbf{a}}\times\rfloor \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\lfloor\hat{\boldsymbol{\omega}}\times\rfloor \end{bmatrix} \begin{bmatrix} \Delta^k\boldsymbol{\alpha}_\tau \\ \Delta^k\boldsymbol{\beta}_\tau \\ \Delta^\tau\boldsymbol{\theta}_k \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -{}^{k}_{\tau}\breve{\mathbf{R}} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\mathbf{I}_{3\times3} \end{bmatrix} \begin{bmatrix} \mathbf{n}_a \\ \mathbf{n}_w \end{bmatrix} \tag{25}$$

$$\Rightarrow \quad \dot{\mathbf{r}} = \mathbf{F}\mathbf{r} + \mathbf{G}\mathbf{n} \tag{26}$$

This differential equation gives rise to a solution which describes the error (note that t_τ refers to the physical time corresponding to step τ):

$$\mathbf{r}(\tau+1) = \boldsymbol{\Phi}(t_{\tau+1}, t_\tau)\mathbf{r}(\tau) + \int_{t_\tau}^{t_{\tau+1}} \boldsymbol{\Phi}(t_{\tau+1}, u)\mathbf{G}(u)\mathbf{n}(u)du \tag{27}$$

$\boldsymbol{\Phi}(t_{\tau+1}, t_\tau)$ is the state-transition matrix from step τ to step $\tau+1$. The state-transition matrix is found using the equations:

$$\begin{aligned} \dot{\boldsymbol{\Phi}}(t_{\tau+s}, t_\tau) &= \mathbf{F}(t_{\tau+s})\boldsymbol{\Phi}(t_{\tau+s}, t_\tau) \\ \begin{bmatrix} \dot{\boldsymbol{\Phi}}_{11} & \dot{\boldsymbol{\Phi}}_{12} & \dot{\boldsymbol{\Phi}}_{13} \\ \dot{\boldsymbol{\Phi}}_{21} & \dot{\boldsymbol{\Phi}}_{22} & \dot{\boldsymbol{\Phi}}_{23} \\ \dot{\boldsymbol{\Phi}}_{31} & \dot{\boldsymbol{\Phi}}_{32} & \dot{\boldsymbol{\Phi}}_{33} \end{bmatrix} &= \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -{}^{k}_{\tau+s}\breve{\mathbf{R}}\lfloor\hat{\mathbf{a}}\times\rfloor \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\lfloor\hat{\boldsymbol{\omega}}\times\rfloor \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_{11} & \boldsymbol{\Phi}_{12} & \boldsymbol{\Phi}_{13} \\ \boldsymbol{\Phi}_{21} & \boldsymbol{\Phi}_{22} & \boldsymbol{\Phi}_{23} \\ \boldsymbol{\Phi}_{31} & \boldsymbol{\Phi}_{32} & \boldsymbol{\Phi}_{33} \end{bmatrix} \\ \boldsymbol{\Phi}(t_\tau, t_\tau) &= \mathbf{I}_{9\times9} \end{aligned} \tag{28}$$

This system of differential equations can be solved entry-wise:

$$\begin{aligned} \boldsymbol{\Phi}_{31}(t_{\tau+1}, t_\tau) &= \boldsymbol{\Phi}_{32}(t_{\tau+1}, t_\tau) = \boldsymbol{\Phi}_{21}(t_{\tau+1}, t_\tau) = \mathbf{0}_{3\times3} \\ \boldsymbol{\Phi}_{11}(t_{\tau+1}, t_\tau) &= \boldsymbol{\Phi}_{22}(t_{\tau+1}, t_\tau) = \mathbf{I}_{3\times3} \\ \boldsymbol{\Phi}_{33}(t_{\tau+1}, t_\tau) &= {}^{\tau+1}_{\tau}\breve{\mathbf{R}} \\ \boldsymbol{\Phi}_{12}(t_{\tau+1}, t_\tau) &= \mathbf{I}_{3\times3}\Delta t \\ \boldsymbol{\Phi}_{23}(t_{\tau+1}, t_\tau) &= -\lfloor({}^{k}\breve{\boldsymbol{\beta}}_{\tau+1} - {}^{k}\breve{\boldsymbol{\beta}}_{\tau})\times\rfloor {}^{k}_{\tau}\breve{\mathbf{R}} \\ \boldsymbol{\Phi}_{13}(t_{\tau+1}, t_\tau) &= -\lfloor({}^{k}\breve{\boldsymbol{\alpha}}_{\tau+1} - {}^{k}\breve{\boldsymbol{\alpha}}_{\tau} - {}^{k}\breve{\boldsymbol{\beta}}_{\tau}\Delta t)\times\rfloor {}^{k}_{\tau}\breve{\mathbf{R}} \end{aligned} \tag{29}$$

Using the state-transition matrix, we can incrementally compute the measurement covariance:

$$\begin{aligned} \mathbf{P}_k &= \mathbf{0}_{9\times9} \\ \mathbf{P}_{\tau+1} &= \boldsymbol{\Phi}(t_{\tau+1}, t_\tau)\mathbf{P}_\tau\boldsymbol{\Phi}(t_{\tau+1}, t_\tau)^\top + \mathbf{Q}_d \\ \mathbf{Q}_d &= \int_{t_\tau}^{t_{\tau+1}} \boldsymbol{\Phi}(t_{\tau+1}, u)\mathbf{G}(u)\mathbf{Q}_c\mathbf{G}^\top(u)\boldsymbol{\Phi}(t_{\tau+1}, u)^\top du \\ \mathbf{Q}_c &= \begin{bmatrix} \sigma_a^2\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \sigma_w^2\mathbf{I}_{3\times3} \end{bmatrix} \end{aligned} \tag{30}$$

V. Direct-Alignment Visual Factors

IMU preintegration factors accurately connect nodes that are time sequential. Over time, however, errors will propagate and cause divergence of the state estimate. As such, optimally fusing IMU data with different sensor measurements is essential for providing long-term accuracy. For visual-inertial navigation, this extra sensor data takes the form of camera measurements. These typically come in the form of features: 3D points in the environment corresponding to "interesting" pixel locations that are matched between images. These methods, however, fail to utilize the information contained in other pixels. We therefore leverage the recent progress in direct methods, which seek to estimate relative transformations using a large subset of the available pixels.

At every image time, the stereo camera records a pair of images of the surrounding environment. These images are rectified so that the epipolar line corresponds to the horizontal, thereby allowing for efficient depth computation. This yields a depth map for the image pair which describes the 3D scene in the local frame of reference corresponding to the base of the alignment. When the scene is recorded at another time, the environment is projected into this new image. Barring changes of lighting, we expect pixel intensities corresponding to the same 3D point to be similar when viewed in both images.

A. Cost Function

Given two images, I_1 and I_2, our goal is to estimate the transformation between the camera poses of the sensor suite at the two imaging times. I_1 and I_2 correspond to the left stereo images at two different times (denoted step 1 and step 2). This estimation is done by finding the relative quaternion ${}^{c2}_{c1}\bar{q}$ and position ${}^{c1}\mathbf{p}_{c2}$ that minimize the photometric error of the alignment [9], [15]:

$${}^{c2}_{c1}\breve{q}, {}^{c1}\breve{\mathbf{p}}_{c2} = \arg\min_{{}^{c2}_{c1}\bar{q}, {}^{c1}\mathbf{p}_{c2}} \sum_{j\in\mathcal{J}} \rho\left(\frac{1}{\sigma_{rj}^2}\left(I_2\left({}^{c1}\mathbf{p}_{c2}, {}^{c2}_{c1}\bar{q}, {}^{c1}\mathbf{p}_{f_j}\right) - I_{1_j}\right)^2\right) \tag{31}$$

By analyzing (31), we can see that our cost is the sum of the intensity differences between a pixel in image I_1 and the intensity of that pixel warped into image I_2 (see Figure 2). We sum only over a subset of pixels $\mathcal{J}$, called the valid pixel set. The warping between the images depends on the relative transformation parameters, as well as the 3D position of the pixel in the first frame of reference, ${}^{c1}\mathbf{p}_{f_j}$, which is found utilizing the first image's disparity map. The variance of this residual is denoted σ_{rj}^2, and the weighted photometric error of each pixel is passed through the Huber cost function ρ [24], defined as:

$$\rho(v) = \begin{cases} v & \text{if } v < k^2 \\ 2k\sqrt{v} - k^2 & \text{otherwise} \end{cases} \tag{32}$$

The purpose of the Huber cost (with scale k) is to down-weight large residuals which occur naturally in image alignment due to occlusions, and has been used extensively in previous direct alignment techniques [9].

B. Residual Covariance

The variance of each residual σ_{rj}^2 encodes the uncertainty due to errors in the intensity measurements as well as the disparity map [9]:

$$\sigma_{rj}^2 = 2\sigma_{int}^2 + \left(\frac{\partial r_j}{\partial d_j}\right)^2 \sigma_{dj}^2 \tag{33}$$

We define σ_{int}^2 as the variance of the intensity reading, $\frac{\partial r_j}{\partial d_j}$ as the Jacobian of the intensity residual with respect to the measured disparity, and σ_{dj}^2 as the variance on the disparity measurement, d_j. Defining $\mathbf{t}$ as the pixel coordinates, z as the pixel depth in I_1, and b as the baseline between the stereo pair, this Jacobian can be found using the chain rule:

$$\begin{aligned} \frac{\partial r_j}{\partial d_j} &= \frac{\partial I_2}{\partial \mathbf{t}} \frac{\partial \mathbf{t}}{\partial {}^{c2}\mathbf{p}_{f_j}} \frac{\partial {}^{c2}\mathbf{p}_{f_j}}{\partial {}^{c1}\mathbf{p}_{f_j}} \frac{\partial {}^{c1}\mathbf{p}_{f_j}}{\partial z} \frac{\partial z}{\partial d_j} \\ &= \begin{bmatrix} I_{2_x} & I_{2_y} \end{bmatrix} \begin{bmatrix} \frac{f_x}{{}^{c2}\mathbf{p}_{f_j}(3)} & 0 & -\frac{f_x {}^{c2}\mathbf{p}_{f_j}(1)}{{}^{c2}\mathbf{p}_{f_j}(3)^2} \\ 0 & \frac{f_y}{{}^{c2}\mathbf{p}_{f_j}(3)} & -\frac{f_y {}^{c2}\mathbf{p}_{f_j}(2)}{{}^{c2}\mathbf{p}_{f_j}(3)^2} \end{bmatrix} {}^{c2}_{c1}\mathbf{R} \frac{{}^{c1}\mathbf{p}_{f_j}}{z} \frac{-f_x b}{d_j^2} \end{aligned} \tag{34}$$

I_{2_x} and I_{2_y} are the image gradients in the x and y directions respectively, f_x and f_y are the focal lengths of the camera, and ${}^{c2}\mathbf{p}_{f_j}(i)$ refers to the ith entry in the ${}^{c2}\mathbf{p}_{f_j}$ vector. The variance of the pixel disparity, σ_{dj}^2, can be found by considering the disparity as the minimizer of the following cost:

$$d_j^* = \arg\min_d \frac{1}{\sigma_{rd}^2} \left(I_{1L}(v,u) - I_{1R}(v,u-d)\right)^2 \tag{35}$$

That is, this disparity is the maximum likelihood estimate for a single measurement graph, with the residual being the difference in intensity between the pixel in the left and right stereo pair. The variance associated with this residual r_d can be found as $\sigma_{rd}^2 = 2\sigma_{int}^2$, and comes from uncertainty in the intensity readings. I_{1L} and I_{1R} refer to the left and right images of the stereo pair at step 1. The variance on our disparity estimate can then be approximated as:

$$\sigma_{dj}^2 = \left(\frac{\partial r_d}{\partial d}^2 \frac{1}{\sigma_{rd}^2}\right)^{-1} = \sigma_{rd}^2 \left(\frac{1}{I_{1R_x}}\right)^2 \tag{36}$$

C. Direct-Alignment Optimization

At each iteration of our Gauss-Newton optimization, the update vector $\Delta\mathbf{x} = \begin{bmatrix} \Delta^{c2}\boldsymbol{\theta}_{c1}^\top & \Delta^{c1}\mathbf{p}_{c2}^\top \end{bmatrix}^\top$ can be computed by solving the normal equations:

$$\left(\sum_{j\in\mathcal{J}} w_j \mathbf{J}_j^\top \mathbf{J}_j\right) \Delta\mathbf{x} = -\sum_{j\in\mathcal{J}} w_j \mathbf{J}_j^\top r_j \tag{37}$$

where r_j is the residual, and w_j is a weight computed at each Gauss-Newton iteration as $w_j = \frac{\partial \rho(v_j)}{\partial v_j}\frac{1}{\sigma_{rj}^2}$:

$$\frac{\partial \rho(v_j)}{\partial v_j} = \begin{cases} 1 & \text{if } v_j < k^2 \\ \frac{k}{\sqrt{v_j}} & \text{otherwise} \end{cases} \tag{38}$$

Note that v_j is the argument of the Huber cost. $\mathbf{J}_j$ is the Jacobian of the dense residual with respect to the error state:

$$\begin{aligned} \mathbf{J}_j = & \\ & \begin{bmatrix} I_{2_x} & I_{2_y} \end{bmatrix} \begin{bmatrix} \frac{f_x}{{}^{c2}\mathbf{p}_{f_j}(3)} & 0 & -\frac{f_x {}^{c2}\mathbf{p}_{f_j}(1)}{{}^{c2}\mathbf{p}_{f_j}(3)^2} \\ 0 & \frac{f_y}{{}^{c2}\mathbf{p}_{f_j}(3)} & -\frac{f_y {}^{c2}\mathbf{p}_{f_j}(2)}{{}^{c2}\mathbf{p}_{f_j}(3)^2} \end{bmatrix} \begin{bmatrix} \lfloor {}^{c2}\mathbf{p}_{f_j} \times \rfloor & -{}^{c2}_{c1}\mathbf{R} \end{bmatrix} \end{aligned} \tag{39}$$

Note that we consider w_j as "fixed" during an optimization iteration and therefore ignore the terms that add to our Hessian due to the dependence of w_j on the current estimate. After convergence, we will be left with a distribution on the relative camera pose with covariance $\Sigma_c = \left(\sum_{j\in\mathcal{J}} w_j \mathbf{J}_j^\top \mathbf{J}_j\right)^{-1}$, which we wish to transfer onto one constraining the relative IMU states:

$$\begin{aligned} \begin{bmatrix} {}^{c2}_{c1}\bar{q} \\ {}^{c1}\mathbf{p}_{c2} \end{bmatrix} = \begin{bmatrix} {}^{c2}_{c1}\breve{q} \\ {}^{c1}\breve{\mathbf{p}}_{c2} \end{bmatrix} \boxplus \begin{bmatrix} \Delta^{c2}\boldsymbol{\theta}_{c1} \\ \Delta^{c1}\mathbf{p}_{c2} \end{bmatrix}, \begin{bmatrix} \Delta^{c2}\boldsymbol{\theta}_{c1} \\ \Delta^{c1}\mathbf{p}_{c2} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}_{6\times 1}, \boldsymbol{\Sigma}_c\right) \Rightarrow \\ \begin{bmatrix} {}^{2}_{1}\bar{q} \\ {}^{1}\mathbf{p}_{2} \end{bmatrix} = \begin{bmatrix} {}^{2}_{1}\breve{q} \\ {}^{1}\breve{\mathbf{p}}_{2} \end{bmatrix} \boxplus \begin{bmatrix} \Delta^{2}\boldsymbol{\theta}_{1} \\ \Delta^{1}\mathbf{p}_{2} \end{bmatrix}, \begin{bmatrix} \Delta^{2}\boldsymbol{\theta}_{1} \\ \Delta^{1}\mathbf{p}_{2} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}_{6\times 1}, \boldsymbol{\Sigma}_i\right) \end{aligned} \tag{40}$$

We then transform this distribution using the rigid calibration parameters between the IMU and camera, $({}^{I}_{c}\mathbf{R}, {}^{c}\mathbf{p}_I)$:

$$\begin{aligned} {}^{2}_{1}\breve{\mathbf{R}} &= {}^{I}_{c}\mathbf{R}\, {}^{c2}_{c1}\breve{\mathbf{R}}\, {}^{I}_{c}\mathbf{R}^\top \\ {}^{1}\breve{\mathbf{p}}_2 &= {}^{I}_{c}\mathbf{R}({}^{c2}_{c1}\breve{\mathbf{R}}^\top {}^{c}\mathbf{p}_I + {}^{c1}\breve{\mathbf{p}}_{c2} - {}^{c}\mathbf{p}_I) \end{aligned} \tag{41}$$

The covariance can be propogated by computing the derivative of the relative IMU residual with respect to the relative camera residual:

$$\begin{aligned} \boldsymbol{\Sigma}_i &= \mathbf{H}\boldsymbol{\Sigma}_c\mathbf{H}^\top \\ \mathbf{H} &= \begin{bmatrix} {}^{I}_{c}\mathbf{R} & \mathbf{0}_{3\times 3} \\ -{}^{I}_{c}\mathbf{R}\, {}^{c2}_{c1}\breve{\mathbf{R}}^\top \lfloor {}^{c}\mathbf{p}_I \times \rfloor & {}^{I}_{c}\mathbf{R} \end{bmatrix} \end{aligned} \tag{42}$$

Given the two transformation parameters, ${}^{2}_{1}\breve{q}$ and ${}^{1}\breve{\mathbf{p}}_2$, the residuals associated with these measurements are given by:

$$\begin{aligned} \Delta^2\boldsymbol{\theta}_1 &= 2\text{vec}\left({}^{2}_{G}\bar{q} \otimes {}^{1}_{G}\bar{q}^{-1} \otimes {}^{2}_{1}\breve{q}^{-1}\right) \\ \Delta^1\mathbf{p}_2 &= {}^{1}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_2 - {}^{G}\mathbf{p}_1\right) - {}^{1}\breve{\mathbf{p}}_2 \end{aligned} \tag{43}$$

These factors can then be inserted into the graph as an edge connecting the two IMU states, with the information matrix of this measurement being the inverse of the computed relative transformation covariance. Alternatively, we could have formulated the optimization problem (31) directly in terms of the relative transformation of the IMU poses. However, this would require more computation, as the warping function

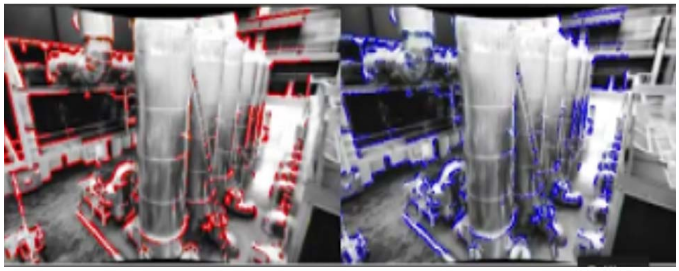


Fig. 2: Illustration of the direct alignment process. A 3D scene is projected onto two images corresponding to different sensor poses. The alignment process attempts to map a large subset of the pixels in the left image (red) to pixels in the right image (blue).

would include the rigid transformation. By parameterizing the problem in terms of the camera state, we need only compute the transformation at the end rather than at every Gauss-Newton iteration.

VI. Experimental Results

A. Implementation

Preintegration was performed across all the IMU measurements received between imaging times. Keyframes were created to serve as the base of the image constraints. When a new image pair arrived, the current predicted pose was used to find a close keyframe for alignment. This criterion was based on the relative position and orientation between the two poses. If no good keyframe was found, a new one was generated using the previous imaging time. Direct alignment was performed between the active keyframe and the newest image. The disparity map of each keyframe was determined with the OpenCV function *StereoSGBM* [25].

In order to speed up processing, we first subsampled the images into 376 x 240 pixel size. Alignment was performed across two pyramid levels. The solution for the relative pose at the first pyramid level served as the initial guess for the next level, as suggested in [9] and [15]. Intuitively, this corresponds to a coarse alignment across a subset of pixels with the highest visual frequencies removed, followed by a finer alignment on the largest image. In practice this leads to a higher basin of attraction for the alignment method. The set of valid pixels were chosen as those with successful depth estimates and gradients above a threshold. Batch optimization across all the image times was performed after every new image using the iSAM2 [26] implementation available within GTSAM [13]. This incremental smoothing approach allows for very fast, approximate updates as new measurements are added into a graph. To ensure long-term performance, full optimization was performed at periodic intervals.

B. Real-World Experiments

To verify the proposed dense-VINS with analytical preintegration, we tested our approach on several publicly available real-world, EuRoC MAV Datasets [27]. The three datasets tested were V1-02-medium, V1-03-difficult, and V2-02-medium (Vicon Room 1 02, Vicon Room 1 03, and Vicon Room 2 02). These datasets consist of an UAV equipped with an IMU and stereo camera exhibiting dynamic motion through an indoor environment. IMU readings and image pairs were recorded at 200 Hz and 20 Hz respectively. Note that we inflated the visual factor noise covariance to capture unmodeled errors. The trajectory for V1-02-medium is shown in Figure 3. For V1-02-medium, the position RMSE was approximately 0.07 m (0.098% of the path). For V1-03-difficult, the algorithm achieved an RMSE of 0.086 m (0.1% of path). Finally, for V2-02-medium, the RMSE was 0.14 m (0.17% of path). The error values across each of the paths are shown in Figure 4. These results clearly validate the proposed method.

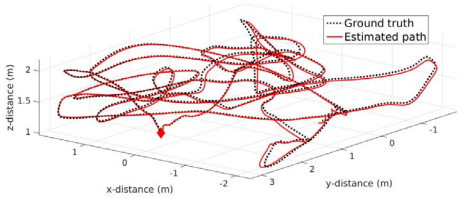


Fig. 3: Ground truth vs. the estimated trajectory of the proposed dense VINS in the experiment on the V1-02-medium dataset [27].

VII. Conclusions and Future Work

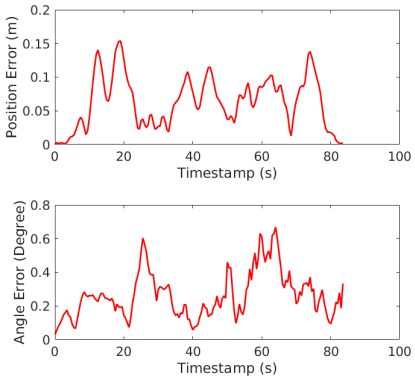
In this paper we have introduced a graph-based sensor fusion algorithm for visual-inertial navigation. In particular, IMU measurements are processed in a continuous fashion to provide high-accuracy preintegrated inertial factors. These are then combined with relative-pose constraints derived from direct alignment of images from a stereo system. The proposed direct-VINS method has been validated on real-world datasets and shown to attain good performance. Note that in this work, in order to prove the key concepts of fusing analytical IMU preintegration with direct visual alignments, our method still requires a periodic, full-batch solution of the global graph, which would prevent real-time performance. As such, we plan to leverage our prior work on marginalization techniques [28], [29] to optimally remove past states. In addition, we plan to extend our work to include better loop closure detection, improved depth map computation, and robustness to alignment failures.

References

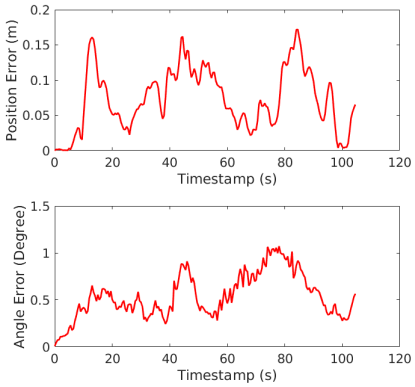
[1] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 10–14, 2007, pp. 3565–3572.

[2] J. W. Langelaan, "State estimation for autonomous flight in cluttered environments," Ph.D. dissertation, Stanford University, Department of Aeronautics and Astronautics, 2006.

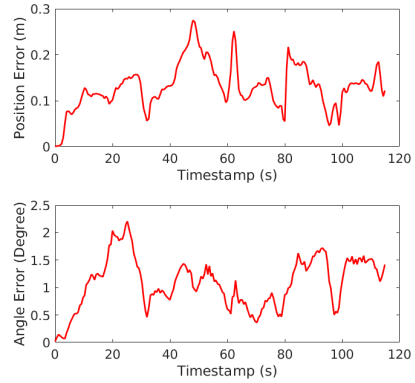
[3] M. Bryson and S. Sukkarieh, "Observability Analysis and Active Control for Airborne SLAM," *IEEE Transactions on Aerospace Electronic Systems*, vol. 44, pp. 261–280, Jan. 2008.



(a) V1-02-medium



(b) V1-03-difficult



(c) V2-02-medium

Fig. 4: Estimation results for the algorithm on three publicly available datasets. In each case our method was able to achieve good estimation performance.

[4] A. I. Mourikis and S. I. Roumeliotis, "A dual-layer estimator architecture for long-term localization," in *Proc. of the Workshop on Visual Localization for Mobile Platforms, IEEE Conference on Computer Vision and Pattern Recognition,*, Anchorage, AK, Jun. 28, 2008.

[5] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9–13, 2011, pp. 3607–3613.

[6] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 61–76, Feb 2012.

[7] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Robotics: Science and Systems XI*, no. EPFL-CONF-214687, 2015.

[8] K. Eckenhoff, P. Geneva, and G. Huang, "High-accuracy preintegration for visual inertial navigation," in *International Workshop on the Algorithmic Foundations of Robotics*, December 2016.

[9] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision (ECCV)*, September 2014.

[10] J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis, "Towards consistent vision-aided inertial navigation," in *Algorithmic Foundations of Robotics X*, ser. Springer Tracts in Advanced Robotics, E. Frazzoli, T. Lozano-Perez, N. Roy, and D. Rus, Eds. Springer Berlin Heidelberg, 2013, vol. 86, pp. 559–574.

[11] F. Dellaert, "Square root SAM," in *Proceedings of Robotics: Science and Systems*, Cambridge, MA, June 2005.

[12] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[13] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Tech, Tech. Rep. GT-RIM-CPR-2012-002, 2012.

[14] Z. Yang and S. Shen, "Monocular visual-inertial state estimation with online initialization and camera-imu extrinsic calibration," *IEEE Transactions on Automation Science and Engineering*, vol. PP, no. 99, pp. 1–13, 2016.

[15] Y. Ling, T. Liu, and S. Shen, "Aggressive quadrotor flight using dense visual-inertial fusion," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1499–1506.

[16] J. H. Kim, C. Cadena, and I. Reid, "Direct semi-dense SLAM for rolling shutter cameras," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2016.

[17] V. Usenko, J. Engel, J. Stueckler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *Int. Conf. on Robotics and Automation*, 2016.

[18] A. Concha, G. Loianno, V. Kumar, and J. Civera, "Visual-inertial direct SLAM," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1331–1338.

[19] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011, vol. 2.

[20] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep., Mar. 2005.

[21] S. Shen, N. Michael, and V. Kumar, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5303–5310.

[22] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Rob. Res.*, vol. 34, no. 3, pp. 314–334, Mar. 2015. [Online]. Available: http://dx.doi.org/10.1177/0278364914554813

[23] K. Eckenhoff, P. Geneva, and G. Huang, "High-accuracy preintegration for visual inertial navigation," University of Delaware, Tech. Rep. RPNG-2016-001, 2016, available: http://udel.edu/~ghuang/papers/tr_hapi.pdf.

[24] E. Eade, "Gauss-newton/levenberg-marquardt optimization," Tech. Rep., 2013, available: http://ethaneade.com/optimization.pdf.

[25] OpenCV Developers Team, "Open source computer vision (OpenCV) library," Available: http://opencv.org.

[26] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *International Journal of Robotics Research*, vol. 31, pp. 217–236, Feb. 2012.

[27] 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*, 2016. [Online]. Available: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract

[28] G. Huang, M. Kaess, and J. Leonard, "Consistent sparsification for graph optimization," in *Proc. of the European Conference on Mobile Robots*, Barcelona, Spain, Sep. 25–27, 2013, pp. 150–157.

[29] K. Eckenhoff, L. Paull, and G. Huang, "Decoupled, consistent node removal and edge sparsification for graph-based slam," in *in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, October 9-14 2016.