High-Accuracy Preintegration for Visual-Inertial Navigation

Patrick Geneva
University of Delaware

High-Accuracy Preintegration for Visual-Inertial Navigation

Kevin Eckenhoff, Patrick Geneva, and Guoquan Huang[⋆]

Department of Mechanical Engineering, University of Delaware, Newark, DE
{keck,pgeneva,ghuang}@udel.edu

Abstract. Visual-inertial navigation that is able to provide accurate 3D localization in GPS-denied environments has seen popularity in recent years due to the proliferation of cost-effective cameras and inertial measurement units (IMUs). While an extended Kalman filter (EKF) is often used for sensor fusion, factor graph-based optimization has recently revealed its superior performance, which, however, is still compromised by the lack of rigorous IMU preintegration (i.e., integrating IMU measurements in a local frame of reference). To address this issue, in this paper, we analytically derive the preintegration based on the closed-form solutions of the continuous integration equations of IMU measurements. These expressions allow us to analytically compute the mean, covariance, and bias Jacobians for a set of IMU preintegration factors. These accurate factors are subsequently fused with the visual information via visual-inertial factor graph optimization to provide high-precision trajectory estimates. The proposed method is validated on both Monte Carlo simulations and real-world experiments.

1 Introduction

Visual-inertial navigation systems (VINS) that fuse visual and inertial information to provide accurate localization, have become nearly ubiquitous in part because of their low cost and light weight (e.g., see [1, 2, 3]). IMUs provide local angular velocity and linear acceleration measurements, while cameras are a cheap yet informative means for sensing the surrounding environment and thus an ideal aiding source for inertial navigation. In particular, these benefits have made VINS popular in resource-constrained systems such as micro aerial vehicles (MAVs) [4]. Traditionally, navigation solutions have been achieved via extended Kalman Filters (EKFs), where incoming proprioceptive (IMU) and exteroceptive (camera) measurements are processed to propagate and update state estimates, respectively. These filtering methods do not update past state estimates that have been marginalized out, thus causing them to be susceptible to drift due to the compounding of errors.

Graph-based optimization methods, by contrast, process all measurements taken over a trajectory simultaneously to estimate a smooth history of sensor states. These methods achieve higher accuracy due to the ability to relinearize nonlinear measurement

[⋆] 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-15661293), and the DTRA (HDTRA1-16-1-0039).

functions and correct previous state estimates [5]. Recently, graph-based formulations have been introduced that allow the incorporation of IMU measurements into "preintegrated" factors by performing integration of the system dynamics in a *local* frame of reference [6, 7, 8]. However, these methods often simplify the required preintegration by resorting to discrete solutions under the approximation of piece-wise constant accelerations. To improve this IMU preintegration, in this paper, we instead model the IMU measurements as piece-wise constant and rigorously derive closed-form solutions of the integration equations. These solutions precisely model the underlying continuous dynamics of the preintegrated measurements. Based on these expressions, we offer analytical computations of the mean, covariance, and bias Jacobians, which have historically been solved using the discretized integrations.

After reviewing past literature in Section 2, we briefly review graph-based batch optimization. Following this, Section 4 presents our rigorous derivations of the analytical IMU preintegration and their respective Jacobians needed for graph-based VINS. In Section 5, we then explain the sliding window based visual tracking. We benchmark our analytical preintegration against the state-of-the-art discrete preintegration in Monte-Carlo simulations in Section 6, where we also offer real-world evaluations of the proposed VINS on a publicly available dataset. Finally, Section 7 concludes the paper as well as possible directions for future research.

2 Related Work

Filtering formulations for the VINS have been dominated by sliding window filters. Mourikis et al. [9] introduced the Multi-State Constraint Kalman Filter (MSCKF). This method was based on the idea of stochastic cloning [10], where the current state is augmented with copies of the past n sensor poses. This allowed for short term smoothing and past error correction, as well as processing exteroceptive sensor measurements without needing to store features in the state vector. This was achieved by projecting the measurement residuals onto the nullspace of the measurement Jacobians corresponding to features. Later filters [1, 3] improved the consistency of the MSCKF by enforcing the correct observability properties on the nonlinear system. In a similar vein, the Optimal-State Constraint (OSC)-EKF proposed in our prior work [11] also stores a sliding window of historical IMU poses. Rather than the projection method used in MSCKF, the OSC-EKF generates a multi-state constraint using a local batch optimization across the camera measurements in the window, and removes the dependency on the environment by marginalizing out features. This idea is also used in our visual tracking front-end in this work (see Section 5). These filtering formulations historically utilize the highly-accurate continuous view of the dynamics when performing propagation. This view of the nonlinear system motivates our derivation of closed-form expressions for preintegration. The error associated in discretization becomes especially detrimental when using low-cost sensors which cannot sample fast enough to mitigate linearization errors. These EKFs, while being computationally inexpensive, suffer from the inability to correct past states, leading to large estimation drift.

Graph-based formulations improve accuracy over their filtering counterparts by processing all measurements at once to estimate an entire trajectory [5, 12]. While these methods are well suited for the case of relative pose measurements, they have historically had difficulty processing IMU measurements. The theory of preintegration was first introduced by Lupton et al [6]. By integrating multiple IMU measurements in a *local* frame of reference, initial conditions (velocity and gravity) can be recovered. The dependency of these integrations on bias was removed via a first-order Taylor series expansion, linearized about the current bias estimate. These techniques allowed for IMU measurements to be processed with a graph in a tractable fashion. Note that the relative rotation uses the Euler angle parametrization, which is known to suffer from singularities. This was mitigated by Forster et al [7] who used the Lie algebra representation of rotation to analytically derive the involved Jacobians by exploiting properties of the manifold. However, both of these techniques discretize the preintegration equations. By contrast our solutions are based on the continuous dynamics which properly model the underlying system. Preintegration with quaternions has been extensively used by Ling and et al. [8], While these were also set up in a continuous manner, they were still sampled when computing means and covariances. In addition, bias updates are not used to correct preintegrated measurements. Other work that utilized preintegrated measurements without bias update can also be found in [13, 14]. Without using bias Jacobians, these methods are unable to correct their preintegrated measurements when the bias linearization point changes, thereby introducing avoidable errors into the system.

3 Graph-based Batch Optimization

Batch optimization techniques provide more accurate estimation over their filtering counterparts by processing all measurements taken during a trajectory simultaneously to build a consistent and smooth estimate over the entire path. Measurements are commonly modeled as being corrupted by Gaussian noise in the form:[1]

$$\mathbf{z}_i = \mathbf{h}_i\left(\mathbf{x}\right) + \mathbf{n}_i \ , \quad \mathbf{n}_i \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Lambda}_i^{-1}\right) \ , \tag{1}$$

where $\mathbf{z}_i$ is the measurement vector, $\mathbf{x}$ is the true value of the state that is being estimated, $\mathbf{h}_i$ is a function that maps our state into the measurement, and $\mathbf{n}_i$ is zero-mean Gaussian noise with information matrix $\boldsymbol{\Lambda}_i$. Maximum Likelihood Estimation (MLE) for our state can be formulated as the following Nonlinear Least-Squares (NLS) optimization problem [5, 12]:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \sum_i ||\mathbf{z_i} - \mathbf{h}_i\left(\mathbf{x}\right)||^2_{\boldsymbol{\Lambda}_i} = \arg\min_{\mathbf{x}} \sum_i ||\mathbf{r}_i\left(\mathbf{x}\right)||^2_{\boldsymbol{\Lambda}_i} \ , \tag{2}$$

where $||\mathbf{w}||^2_{\boldsymbol{\Lambda}} = \mathbf{w}^\top \boldsymbol{\Lambda}\mathbf{w}$. We define $\mathbf{r}_i$ as the residual associated with measurement i. This problem is often solved iteratively via Gauss-Newton approximations of the cost about the current estimate.

$$\delta\mathbf{x} = \arg\min_{\delta\mathbf{x}} \sum_i ||\mathbf{r}_i\left(\hat{\mathbf{x}} \boxplus \delta\mathbf{x}\right)||^2_{\boldsymbol{\Lambda}_i} \simeq \arg\min_{\delta\mathbf{x}} \sum_i ||\mathbf{r}_i\left(\hat{\mathbf{x}}\right) + \mathbf{J}_i\delta\mathbf{x}||^2_{\boldsymbol{\Lambda}_i} \ , \tag{3}$$

[1] Throughout this paper $\hat{x}$ is used to denote the estimate of a random variable x, while $\tilde{x} = x - \hat{x}$ is the error in this estimate. $\mathbf{I}_{n\times n}$ and $\mathbf{0}_{n\times n}$ are the $n \times n$ identity and zero matrices, respectively. $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_3 \in \mathbb{R}^3$ are the unit vectors along $x-$, $y-$ and $z-$axes. The left superscript denotes the frame of reference which the vector is expressed with respect to.

where $\mathbf{J}_i := \left.\frac{\partial \mathbf{r}_i(\hat{\mathbf{x}} \boxplus \delta\mathbf{x})}{\partial \delta\mathbf{x}}\right|_{\delta\mathbf{x}=\mathbf{0}}$ is the Jacobian of the residual with respect to the error state, evaluated at the error being zero. We also define a corrective operator, $\boxplus$, that applies a correction term to the current estimate to yield a new estimate. This is a generalization of the additive noise associated with vector spaces, and allows the use of states restricted to a manifold, such as that of the unit quaternions. By stacking our measurements, the total corrective term can be obtained via the normal equation:

$$\delta\mathbf{x} = -\left(\mathbf{J}^\top \boldsymbol{\Lambda}\mathbf{J}\right)^{-1}\mathbf{J}^\top \boldsymbol{\Lambda}\mathbf{r}\ . \tag{4}$$

Our estimate at the next iteration is updated as $\hat{\mathbf{x}}^{k+1} = \hat{\mathbf{x}}^k \boxplus \delta\mathbf{x}^k$. This process is repeated until convergence. The resulting distribution is then approximated with the following Gaussian:

$$\mathbf{x} = \hat{\mathbf{x}} \boxplus \delta\mathbf{x}\ , \quad \delta\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \left(\mathbf{J}^\top \boldsymbol{\Lambda}\mathbf{J}\right)^{-1}\right)\ . \tag{5}$$

That is, our true state is modeled as being achievable from our current state estimate with a random error correction which is pulled from a Gaussian distribution with zero mean and information matrix of $\mathbf{J}^\top \boldsymbol{\Lambda}\mathbf{J}$.

4 Analytical IMU Preintegration

An IMU typically measures the local angular velocity $\boldsymbol{\omega}$ and linear acceleration $\boldsymbol{a}$ of its body, which are assumed to be corrupted by the Gaussian white noise ($\mathbf{n}_w$ and $\mathbf{n}_a$) and the random-walk biases ($\mathbf{b}_w$ and $\mathbf{b}_a$) [15]:

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{b}_w + \mathbf{n}_w\ ,\ \mathbf{a}_m = \mathbf{a} + \mathbf{g} + \mathbf{b}_a + \mathbf{n}_a\ ,\ \dot{\mathbf{b}}_w = \mathbf{n}_{wg}\ ,\ \dot{\mathbf{b}}_a = \mathbf{n}_{wa}\ , \tag{6}$$

where $\mathbf{g}$ is the gravity vector in the local frame whose global counterpart is constant (e.g., ${}^G\mathbf{g} = [0\ \ 0\ 9.8]^T$). The navigation state at time-step k is given by:

$$\mathbf{x}_k = \begin{bmatrix} {}^{L_k}_{G}\bar{q}^T & \mathbf{b}_{w_k}^T & {}^G\mathbf{v}_k^T & \mathbf{b}_{a_k}^T & {}^G\mathbf{p}_k^T \end{bmatrix}^T\ , \tag{7}$$

where ${}^{L_k}_{G}\bar{q} := \begin{bmatrix}\mathbf{q}^T & q_4\end{bmatrix}^T$ is the JPL convention, [15], that describes the rotation from frame $\{G\}$ to frame $\{L_k\}$, and ${}^G\mathbf{v}_k$ and ${}^G\mathbf{p}_k$ are the velocity and position of the k-th local frame in the global frame, respectively. The corresponding error state and $\boxplus$ operation used in batch optimization can be written as (note that hereafter the transpose has been omitted for brevity):

$$\tilde{\mathbf{x}}_k = \begin{bmatrix} {}^{L_k}\delta\boldsymbol{\theta}_G & \tilde{\mathbf{b}}_{w_k} & {}^G\tilde{\mathbf{v}}_k & \tilde{\mathbf{b}}_{a_k} & {}^G\tilde{\mathbf{p}}_k \end{bmatrix}\ , \tag{8}$$

$$\mathbf{x}_k = \hat{\mathbf{x}}_k \boxplus \tilde{\mathbf{x}}_k = \begin{bmatrix} {}^{L_k}_{\hat{L}_k}\delta\bar{q} \otimes {}^{\hat{L}_k}_{G}\hat{\bar{q}} \\ \hat{\mathbf{b}}_{w_k} + \tilde{\mathbf{b}}_{w_k} \\ {}^G\hat{\mathbf{v}}_k +^G \tilde{\mathbf{v}}_k \\ \hat{\mathbf{b}}_{a_k} + \tilde{\mathbf{b}}_{a_k} \\ {}^G\hat{\mathbf{p}}_k +^G \tilde{\mathbf{p}}_k \end{bmatrix}\ , \tag{9}$$

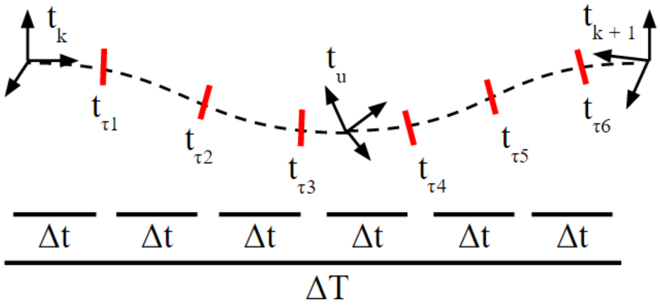

Fig. 1: IMU measurements are collected at discrete times t_τ with period Δt. This sampling occurs between image times t_k and t_{k+1}. We refer to the step k with a time of t_k, with t_u describing a generic time in the continuous domain.

where the operator $\otimes$ denotes quaternion multiplication, and ${}^{L_k}_{\hat{L}_k}\delta\bar{q}$ is the error quaternion whose vector portion is half the error angle, ${}^{L_k}\delta\boldsymbol{\theta}_G = 2\mathbf{vec}\left({}^{L_k}_{\hat{L}_k}\delta\bar{q}\right) := 2{}^{L_k}_{\hat{L}_k}\delta\mathbf{q}$.

Between two imaging steps, k to $k+1$, IMU measurements are collected and integrated at times $t_k < t_\tau < t_{k+1}$ without accessing the state estimates (in particular, the orientation). This is done by performing the following factorization of the *current* rotation matrix and integration of the measurements [16]:

$$\begin{aligned}
{}^{G}\mathbf{p}_{k+1} &= {}^{G}\mathbf{p}_k + {}^{G}\mathbf{v}_k \Delta T - \frac{1}{2}{}^{G}\mathbf{g}\Delta T^2 + {}^{G}_{k}\mathbf{R}\underbrace{\int_{t_k}^{t_{k+1}}\int_{t_k}^{s} {}^{k}_{u}\mathbf{R}\left({}^{u}\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a\right) du ds}_{{}^{k}\boldsymbol{\alpha}_{k+1}} \\
&=: {}^{G}\mathbf{p}_k + {}^{G}\mathbf{v}_k \Delta T - \frac{1}{2}{}^{G}\mathbf{g}\Delta T^2 + {}^{G}_{k}\mathbf{R}{}^{k}\boldsymbol{\alpha}_{k+1} , \qquad (10) \\
{}^{G}\mathbf{v}_{k+1} &= {}^{G}\mathbf{v}_k - {}^{G}\mathbf{g}\Delta T + {}^{G}_{k}\mathbf{R}\underbrace{\int_{t_k}^{t_{k+1}} {}^{k}_{u}\mathbf{R}\left({}^{u}\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a\right) du}_{{}^{k}\boldsymbol{\beta}_{k+1}} \\
&=: {}^{G}\mathbf{v}_k - {}^{G}\mathbf{g}\Delta T + {}^{G}_{k}\mathbf{R}{}^{k}\boldsymbol{\beta}_{k+1} , \qquad (11) \\
{}^{k+1}_{G}\mathbf{R} &= {}^{k+1}_{k}\mathbf{R}{}^{k}_{G}\mathbf{R} , \qquad (12)
\end{aligned}$$

where $\Delta T = t_{k+1} - t_k$, and s and u are dummy variables used for integration (see Figure 1). Following the notation of Ling et al. [8], it becomes clear that the above integrals have been collected into the *preintegrated* measurements, ${}^{k}\boldsymbol{\alpha}_{k+1}$ and ${}^{k}\boldsymbol{\beta}_{k+1}$, which are expressed in the k-th local frame. Rearrangement of these equations yields:

$$\begin{aligned}
{}^{k}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_{k+1} - {}^{G}\mathbf{p}_k - {}^{G}\mathbf{v}_k \Delta T + \frac{1}{2}{}^{G}\mathbf{g}\Delta T^2\right) &= {}^{k}\boldsymbol{\alpha}_{k+1}\left(\mathcal{I}, \mathbf{n}_a, \mathbf{n}_w, \mathbf{b}_a, \mathbf{b}_w\right) , \qquad (13) \\
{}^{k}_{G}\mathbf{R}\left({}^{G}\mathbf{v}_{k+1} - {}^{G}\mathbf{v}_k + {}^{G}\mathbf{g}\Delta T\right) &= {}^{k}\boldsymbol{\beta}_{k+1}\left(\mathcal{I}, \mathbf{n}_a, \mathbf{n}_w, \mathbf{b}_a, \mathbf{b}_w\right) , \qquad (14) \\
{}^{k+1}_{G}\mathbf{R}{}^{k}_{G}\mathbf{R}^{\top} &= {}^{k+1}_{k}\mathbf{R}\left(\mathcal{I}, \mathbf{n}_w, \mathbf{b}_w\right) , \qquad (15)
\end{aligned}$$

where $\mathcal{I}$ is the set of all discrete IMU measurements collected between times t_k and t_{k+1} i.e., $\{{}^{\tau}\mathbf{a}_m, {}^{\tau}\boldsymbol{\omega}_m\}$. For the remainder of this paper the biases will refer to the those of state k, and are approximated as constant over the preintegration interval. It is important to note that the above equations (13)–(15) are dependent on the true biases which will causes exact preintegration to be intractable. In particular, they naively require the re-computation of the preintegration terms every time the bias linearization point changes. To address this issue, we employ the following first-order Taylor series expansion with respect to the biases:

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

where the preintegration functions have been linearized about the current bias estimates, $\bar{\mathbf{b}}_w$ and $\bar{\mathbf{b}}_a$, and $\mathit{\Delta}\mathbf{b}_w := \mathbf{b}_w - \bar{\mathbf{b}}_w$ and $\mathit{\Delta}\mathbf{b}_a := \mathbf{b}_a - \bar{\mathbf{b}}_a$ are the difference between the true biases and their linearization points. Note that in the case of the relative rotations, a change in bias is modeled as inducing a further rotation on our preintegrated relative rotation. This linearization process allows for the computation of the preintegration *once*, while still allowing approximate updates when the bias linearization point changes. The corresponding residuals of these preintegrated measurements for use in graph optimization are given by:

$$
\begin{aligned}
\delta^{k}\boldsymbol{\alpha}_{k+1}&={}^{k}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_{k+1}-{}^{G}\mathbf{p}_{k}-{}^{G}\mathbf{v}_{k}\mathit{\Delta}T+\frac{1}{2}{}^{G}\mathbf{g}\mathit{\Delta}T^{2}\right)-\frac{\partial\boldsymbol{\alpha}}{\partial\mathbf{b}_{a}}\mathit{\Delta}\mathbf{b}_{a}-\frac{\partial\boldsymbol{\alpha}}{\partial\mathbf{b}_{w}}\mathit{\Delta}\mathbf{b}_{w}-{}^{k}\breve{\boldsymbol{\alpha}}_{k+1}\ ,\\
\delta^{k}\boldsymbol{\beta}_{k+1}&={}^{k}_{G}\mathbf{R}\left({}^{G}\mathbf{v}_{k+1}-{}^{G}\mathbf{v}_{k}+{}^{G}\mathbf{g}\mathit{\Delta}T\right)-\frac{\partial\boldsymbol{\beta}}{\partial\mathbf{b}_{a}}\mathit{\Delta}\mathbf{b}_{a}-\frac{\partial\boldsymbol{\beta}}{\partial\mathbf{b}_{w}}\mathit{\Delta}\mathbf{b}_{w}-{}^{k}\breve{\boldsymbol{\beta}}_{k+1}\ ,\\
{}^{k+1}\delta\boldsymbol{\theta}_{k}&=2\mathbf{vec}\left({}^{k+1}_{G}\bar{q}\otimes{}^{k}_{G}\bar{q}^{-1}\otimes{}^{k+1}_{k}\breve{\bar{q}}^{-1}\otimes\bar{q}\left(\mathit{\Delta}\mathbf{b}_{w}\right)^{-1}\right)\ , \qquad (19)
\end{aligned}
$$

where ${}^{k}\breve{\boldsymbol{\alpha}}_{k+1}$, ${}^{k}\breve{\boldsymbol{\beta}}_{k+1}$ and ${}^{k+1}_{k}\breve{\bar{q}}$ are the preintegrated measurements with the quaternion being associated with the preintegrated rotation.

4.1 Compute Preintegration Mean and Covariance via Linear Systems

Before we use the IMU preintegrated measurement residuals (19) in batch optimization, we need to find their mean and covariance. To this end, we first note that the rotation (quaternion) time evolution is given by:

$$
{}^{u}_{k}\dot{\bar{q}}=\frac{1}{2}\boldsymbol{\mathit{\Omega}}({}^{u}\boldsymbol{\omega}_{m}-\bar{\mathbf{b}}_{w}-\mathbf{n}_{w}){}^{u}_{k}\bar{q}\ ,\ \ \boldsymbol{\mathit{\Omega}}(\boldsymbol{\omega})=\begin{bmatrix}-\lfloor\boldsymbol{\omega}\times\rfloor & \boldsymbol{\omega}\\ -\boldsymbol{\omega}^{\top} & 0\end{bmatrix}\ . \qquad (20)
$$

This can be solved using the zeroth order quaternion integrator (see [15]). Based on (10) and (11), we have the following continuous measurement dynamics:

$$
{}^{k}\dot{\boldsymbol{\alpha}}_{u}={}^{k}\boldsymbol{\beta}_{u}\ , \qquad (21)
$$

$$ {}^{k}\dot{\boldsymbol{\beta}}_{u} = {}^{k}_{u}\mathbf{R}\left({}^{u}\mathbf{a}_{m} - \bar{\mathbf{b}}_{a} - \mathbf{n}_{a}\right) \ . \tag{22} $$

From these, we formulate the following *linear* system that describes the time evolution of the preintegrated measurements by taking the (approximate) expectation of the dynamic equations (21) and (22):

$$ \begin{bmatrix} {}^{k}\dot{\hat{\boldsymbol{\alpha}}}_{u} \\ {}^{k}\dot{\hat{\boldsymbol{\beta}}}_{u} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{3\times3} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} {}^{k}\hat{\boldsymbol{\alpha}}_{u} \\ {}^{k}\hat{\boldsymbol{\beta}}_{u} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ {}^{k}_{u}\hat{\mathbf{R}} \end{bmatrix} \left({}^{u}\mathbf{a}_{m} - \bar{\mathbf{b}}_{a}\right) . \tag{23} $$

To *analytically* compute the mean of the preintegration measurement, we preform direct integration between two sample steps τ and $\tau + 1$, which correspond to IMU measurement times t_{τ} to $t_{\tau+1}$. With a little abuse of notation, we define $\hat{\mathbf{a}} = {}^{\tau}\mathbf{a}_{m} - \bar{\mathbf{b}}_{a}$ and $\varDelta t = t_{\tau+1} - t_{\tau}$.

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

The entire preintegrated measurement can be computed incrementally by applying the above expression and the zeroth order quaternion integrator as new IMU measurements arrive. The complete derivations can be found in our tech report [17].

To find the covariance of the preintegrated measurements, we first write out the dynamics of the corresponding error-states as follows:

$$ {}^{k}\delta\dot{\boldsymbol{\alpha}}_{u} = {}^{k}\delta\boldsymbol{\beta}_{u} \ , \tag{25} $$

$$ \begin{aligned} {}^{k}\delta\dot{\boldsymbol{\beta}}_{u} &= {}^{k}_{u}\hat{\mathbf{R}}(\mathbf{I}_{3\times3} + \lfloor{}^{u}\delta\boldsymbol{\theta}_{k}\times\rfloor)\left({}^{u}\mathbf{a}_{m} - \bar{\mathbf{b}}_{a} - \mathbf{n}_{a}\right) - {}^{k}_{u}\hat{\mathbf{R}}\left({}^{u}\mathbf{a}_{m} - \bar{\mathbf{b}}_{a}\right) \\ &= {}^{k}_{u}\hat{\mathbf{R}}\left(-\mathbf{n}_{a}\right) + {}^{k}_{u}\hat{\mathbf{R}}\lfloor{}^{u}\delta\boldsymbol{\theta}_{k}\times\rfloor\left({}^{u}\mathbf{a}_{m} - \bar{\mathbf{b}}_{a}\right) \ , \end{aligned} \tag{26} $$

$$ {}^{u}\delta\dot{\boldsymbol{\theta}}_{k} = -\left\lfloor\left({}^{u}\boldsymbol{\omega}_{m} - \bar{\mathbf{b}}_{w}\right)\times\right\rfloor{}^{u}\delta\boldsymbol{\theta}_{k} - \mathbf{n}_{w} \ . \tag{27} $$

This immediately yields the following linearized system of the error states:

$$ \begin{aligned} &\begin{bmatrix} {}^{k}\delta\dot{\boldsymbol{\alpha}}_{u} \\ {}^{k}\delta\dot{\boldsymbol{\beta}}_{u} \\ {}^{u}\delta\dot{\boldsymbol{\theta}}_{k} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{3\times3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -{}^{k}_{u}\hat{\mathbf{R}}\left\lfloor\left({}^{u}\mathbf{a}_{m} - \bar{\mathbf{b}}_{a}\right)\times\right\rfloor \\ \mathbf{0} & \mathbf{0} & -\left\lfloor\left({}^{u}\boldsymbol{\omega}_{m} - \bar{\mathbf{b}}_{w}\right)\times\right\rfloor \end{bmatrix} \begin{bmatrix} {}^{k}\delta\boldsymbol{\alpha}_{u} \\ {}^{k}\delta\boldsymbol{\beta}_{u} \\ {}^{u}\delta\boldsymbol{\theta}_{k} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -{}^{k}_{u}\hat{\mathbf{R}} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{3\times3} \end{bmatrix} \begin{bmatrix} \mathbf{n}_{a} \\ \mathbf{n}_{w} \end{bmatrix} , \\ &\Rightarrow \ \dot{\mathbf{r}} = \mathbf{F}\mathbf{r} + \mathbf{G}\mathbf{n} \ . \end{aligned} \tag{28} $$

Therefore, the noise covariance, $\mathbf{P}$, can be found as follows:

$$ \mathbf{P}_{k} = \mathbf{0}_{9\times9} \tag{29} $$

$$ \mathbf{P}_{\tau+1} = \mathbf{\Phi}(t_{\tau+1}, t_{\tau})\mathbf{P}_{\tau}\mathbf{\Phi}(t_{\tau+1}, t_{\tau})^{\top} + \mathbf{Q}_{d} \tag{30} $$

$$\mathbf{Q}_d = \int_{t_\tau}^{t_{\tau+1}} \mathbf{\Phi}(t_{\tau+1},u)\mathbf{G}(u)\mathbf{Q}_c\mathbf{G}^\top(u)\mathbf{\Phi}^\top(t_{\tau+1},u)du \ , \ \mathbf{Q}_c = \begin{bmatrix} \sigma_a^2\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \sigma_w^2\mathbf{I}_{3\times3} \end{bmatrix} . \quad (31)$$

We want to stress that rather than using the discrete covariance approximation as in [8], we analytically compute the state-transition matrix, $\mathbf{\Phi}$, and discrete-time noise covariance, $\mathbf{Q}_d$, as derived in our tech report [17]. Although these expressions are more complicated than those found in discrete methods, we find that they do not prevent real-time processing of IMU data due their closed forms.

4.2 Compute Preintegration Jacobians w.r.t. Biases

As shown earlier, changes in bias are modeled as adding corrections to our preintegration measurements through the use of bias Jacobians [see (16) and (17)]. In particular, as seen from (24), each update term is linear in the estimated acceleration, $\hat{\mathbf{a}} =^\tau \mathbf{a}_m - \bar{\mathbf{b}}_a$, thus we find the bias Jacobians of ${}^k\boldsymbol{\alpha}_{k+1}$ and ${}^k\boldsymbol{\beta}_{k+1}$ with respect to $\mathbf{b}_a$ as follows:

$$\begin{bmatrix} \frac{\partial\boldsymbol{\alpha}}{\partial\mathbf{b}_a} \\ \frac{\partial\boldsymbol{\beta}}{\partial\mathbf{b}_a} \end{bmatrix} =: \begin{bmatrix} \mathbf{H}_\alpha(\tau+1) \\ \mathbf{H}_\beta(\tau+1) \end{bmatrix} = \begin{bmatrix} \mathbf{H}_\alpha(\tau) + \mathbf{H}_\beta(\tau)\Delta t \\ \mathbf{H}_\beta(\tau) \end{bmatrix} - \quad (32)$$

$$\begin{bmatrix} {}^{\;\;k}_{\tau+1}\mathbf{R}\left(\frac{\Delta t^2}{2}\mathbf{I}_{3\times3} + \frac{|w|\Delta t\cos(|w|\Delta t) - \sin(|w|\Delta t)}{|w|^3}\lfloor\hat{\boldsymbol{\omega}}\times\rfloor + \frac{(|w|\Delta t)^2 - 2\cos(|w|\Delta t) - 2(|w|\Delta t)\sin(|w|\Delta t) + 2}{2|w|^4}\lfloor\hat{\boldsymbol{\omega}}\times\rfloor^2\right) \\ {}^{\;\;k}_{\tau+1}\mathbf{R}\left(\Delta t\mathbf{I}_{3\times3} - \frac{1-\cos(|\boldsymbol{\omega}|(\Delta t))}{|w|^2}\lfloor\hat{\boldsymbol{\omega}}\times\rfloor + \frac{(|w|\Delta t) - \sin(|\boldsymbol{\omega}|\Delta t)}{|w|^3}\lfloor\hat{\boldsymbol{\omega}}\times\rfloor^2\right) \end{bmatrix} .$$

Similarly, the Jacobians with respect to the gyro bias $\frac{\partial\boldsymbol{\alpha}}{\partial\mathbf{b}_w} =: \mathbf{J}_\alpha$, $\frac{\partial\boldsymbol{\beta}}{\partial\mathbf{b}_w} =: \mathbf{J}_\beta$ can be found. The detailed derivations are omitted here for brevity but are included in our companion technical report [17].

Now consider the relative-rotation measurement, ${}^{k+1}_{\;\;k}\mathbf{R}$. The updated rotation can be approximated as [7]:[2]

$${}^{k+1}_{\;\;k}\mathbf{R}_\oplus = \exp\left(\lfloor\mathbf{J}_q(k+1)(\mathbf{b}_w - \bar{\mathbf{b}}_w)\times\rfloor\right){}^{k+1}_{\;\;k}\mathbf{R}_\ominus \ , \quad (33)$$

where $\exp(\cdot)$ is the matrix exponential. It should be pointed out that in the above expression, that the rotational bias Jacobian, $\mathbf{J}_q$, can be computed *incrementally* using the right Jacobian of SO(3) $\mathbf{J}_{r_{\tau+1}} = \mathbf{J}_r(\boldsymbol{\omega}_\tau(t_{\tau+1} - t_\tau))$ (see [7, 17, 18]):

$$\mathbf{J}_q(\tau+1) = {}^{\tau+1}_{\;\;\tau}\hat{\mathbf{R}}\sum_{u=k}^{\tau}{}^{\tau}_{u}\hat{\mathbf{R}}\mathbf{J}_{r_u}\Delta t + \mathbf{J}_{r_{\tau+1}}\Delta t = {}^{\tau+1}_{\;\;\tau}\hat{\mathbf{R}}\mathbf{J}_q(\tau) + \mathbf{J}_{r_{\tau+1}}\Delta t \ . \quad (34)$$

Then the angle measurement residual can be written as:

$$\begin{aligned} {}^{k+1}\delta\boldsymbol{\theta}_k &= 2\mathbf{vec}\left({}^{k+1}_{\;\;G}\bar{q} \otimes {}^{k}_{G}\bar{q}^{-1} \otimes {}^{k+1}_{\;\;k}\breve{\bar{q}}^{-1} \otimes \mathbf{quat}(\exp(-\lfloor\mathbf{J}_q(k+1)(\mathbf{b}_w - \bar{\mathbf{b}}_w)\times\rfloor))\right) , & (35) \\ &\simeq 2\mathbf{vec}\left({}^{k+1}_{\;\;G}\bar{q} \otimes {}^{k}_{G}\bar{q}^{-1} \otimes {}^{k+1}_{\;\;k}\breve{\bar{q}}^{-1} \otimes \begin{bmatrix} \frac{1}{2}(\mathbf{J}_q(\mathbf{b}_w - \bar{\mathbf{b}}_w) \\ 1 \end{bmatrix}\right) , & (36) \end{aligned}$$

where $\mathbf{quat}(\cdot)$ denotes the transformation of a rotation matrix to the corresponding quaternion. In the above expression, we have also used the common assumption that $(\mathbf{b}_w - \bar{\mathbf{b}}_w)$ is small. Note that we only use this approximation for the computation of Jacobians, while the more accurate (37) is used for the evaluation of actual residuals.

[2] We use the symbol $\oplus$ to denote an estimate after update and $\ominus$ before update.

4.3 Compute Preintegration Measurement Jacobians

Our total preintegrated measurement residuals can now be written as:

$$\mathbf{r}=\begin{bmatrix}{}^{k}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_{k+1}-{}^{G}\mathbf{p}_{k}-{}^{G}\mathbf{v}_{k}\mathit{\Delta} T+\frac{1}{2}{}^{G}\mathbf{g}\mathit{\Delta} T^{2}\right)-\mathbf{J}_{\alpha}(\mathbf{b}_{w}-\bar{\mathbf{b}}_{w})-\mathbf{H}_{\alpha}(\mathbf{b}_{a}-\bar{\mathbf{b}}_{a})-{}^{k}\breve{\boldsymbol{\alpha}}_{k+1}\\ {}^{k}_{G}\mathbf{R}\left({}^{G}\mathbf{v}_{k+1}-{}^{G}\mathbf{v}_{k}+{}^{G}\mathbf{g}\mathit{\Delta} T\right)-\mathbf{J}_{\beta}(\mathbf{b}_{w}-\bar{\mathbf{b}}_{w})-\mathbf{H}_{\beta}(\mathbf{b}_{a}-\bar{\mathbf{b}}_{a})-{}^{k}\breve{\boldsymbol{\beta}}_{k+1}\\ 2\mathrm{vec}\left({}^{k+1}_{G}\bar{q}\otimes{}^{k}_{G}\bar{q}^{-1}\otimes{}^{k+1}_{k}\breve{\bar{q}}^{-1}\otimes\mathbf{quat}(\mathrm{Exp}(-\lfloor\mathbf{J}_{q}(\mathbf{b}_{w}-\bar{\mathbf{b}}_{w})\times\rfloor)\right)\end{bmatrix} \tag{37}$$

In order to use these residuals in graph-based optimization (3), the corresponding Jacobians with respect to the optimization variables are necessary. To this end, we first rewrite the relative-rotation measurement residual as:

$${}^{k+1}\delta\boldsymbol{\theta}_{k}=2\mathbf{vec}\left({}^{k+1}_{G}\bar{q}\otimes{}^{k}_{G}\bar{q}^{-1}\otimes{}^{k+1}_{k}\breve{\bar{q}}^{-1}\otimes\bar{q}_{b}\right)\ , \tag{38}$$

where $\bar{q}_b$ is the quaternion induced by a change in gyro bias. The measurement Jacobian with respect to one element of the state vector can be found by perturbing the residual function by the corresponding element. For example, the relative-rotation measurement residual is perturbed by a change in gyro bias around the current estimate (i.e., $\mathbf{b}_w-\bar{\mathbf{b}}_w=\hat{\mathbf{b}}_w+\tilde{\mathbf{b}}_w-\bar{\mathbf{b}}_w$):

$$\begin{aligned}{}^{k+1}\delta\boldsymbol{\theta}_{k}&=2\mathbf{vec}\left({}^{k+1}_{G}\hat{\bar{q}}\otimes{}^{k+1}_{G}\hat{\bar{q}}^{-1}\otimes{}^{k+1}_{k}\breve{\bar{q}}^{-1}\otimes\begin{bmatrix}\frac{\mathbf{J}_{q}(\hat{\mathbf{b}}_{w}+\tilde{\mathbf{b}}_{w}-\bar{\mathbf{b}}_{w})}{2}\\ 1\end{bmatrix}\right)\\ &=2\mathbf{vec}\left(\begin{bmatrix}\hat{q}_{r,4}\mathbf{I}_{3\times3}-\lfloor\hat{\mathbf{q}}_{r}\times\rfloor & \hat{\mathbf{q}}_{r}\\ -\hat{\mathbf{q}}_{r}^{\top} & \hat{q}_{r,4}\end{bmatrix}\begin{bmatrix}\frac{\mathbf{J}_{q}(\hat{\mathbf{b}}_{w}+\tilde{\mathbf{b}}_{w}-\bar{\mathbf{b}}_{w})}{2}\\ 1\end{bmatrix}\right)\\ &=(\hat{q}_{r,4}\mathbf{I}_{3\times3}-\lfloor\hat{\mathbf{q}}_{r}\times\rfloor)\mathbf{J}_{q}(\hat{\mathbf{b}}_{w}+\tilde{\mathbf{b}}_{w}-\bar{\mathbf{b}}_{w})+\text{other terms}\ ,\\ \Rightarrow\quad\frac{\partial^{k+1}\delta\boldsymbol{\theta}_{k}}{\partial\tilde{\mathbf{b}}_{w}}&=(\hat{q}_{r,4}\mathbf{I}_{3\times3}-\lfloor\hat{\mathbf{q}}_{r}\times\rfloor)\mathbf{J}_{q}\ .\end{aligned} \tag{39}$$

Similarly, the Jacobian with respect to ${}^{k+1}\delta\boldsymbol{\theta}_G$ can be found as follows:

$$\begin{aligned}{}^{k+1}\delta\boldsymbol{\theta}_{k}&=2\mathbf{vec}\left(\begin{bmatrix}\frac{1}{2}{}^{k+1}\delta\boldsymbol{\theta}_{G}\\ 1\end{bmatrix}\otimes{}^{k+1}_{G}\hat{\bar{q}}\otimes{}^{k}_{G}\hat{\bar{q}}^{-1}\otimes{}^{k+1}_{k}\bar{q}^{-1}\otimes\hat{\bar{q}}_{b}\right)\\ &=2\mathbf{vec}\left(\begin{bmatrix}\hat{q}_{rb,4}\mathbf{I}_{3\times3}+\lfloor\hat{\mathbf{q}}_{rb}\times\rfloor & \hat{\mathbf{q}}_{rb}\\ -\hat{\mathbf{q}}_{rb}^{\top} & \hat{q}_{rb,4}\end{bmatrix}\begin{bmatrix}\frac{1}{2}{}^{k+1}\delta\boldsymbol{\theta}_{G}\\ 1\end{bmatrix}\right)\\ &=(\hat{q}_{rb,4}\mathbf{I}_{3\times3}+\lfloor\hat{\mathbf{q}}_{rb}\times\rfloor)^{k+1}\delta\boldsymbol{\theta}_{G}+\ \text{other terms}\ ,\\ \Rightarrow\quad\frac{\partial^{k+1}\delta\boldsymbol{\theta}_{k}}{\partial^{k+1}\delta\boldsymbol{\theta}_{G}}&=\hat{q}_{rb,4}\mathbf{I}_{3\times3}+\lfloor\hat{\mathbf{q}}_{rb}\times\rfloor\ .\end{aligned} \tag{40}$$

The Jacobian with respect to ${}^{k}\delta\boldsymbol{\theta}_G$ is given by:

$${}^{k+1}\delta\boldsymbol{\theta}_{k}=2\mathbf{vec}\left({}^{k+1}_{G}\hat{\bar{q}}\otimes{}^{k}_{G}\hat{\bar{q}}^{-1}\otimes\begin{bmatrix}-\frac{{}^{k}\delta\boldsymbol{\theta}_{G}}{2}\\ 1\end{bmatrix}\otimes{}^{k+1}_{k}\bar{q}^{-1}\otimes\hat{\bar{q}}_{b}\right)$$

$$
\begin{aligned}
&= 2\mathbf{vec}\left(\begin{bmatrix}\hat{q}_{n,4}\mathbf{I}_{3\times3}-\lfloor\hat{\mathbf{q}}_n\times\rfloor & \hat{\mathbf{q}}_n\\ -\hat{\mathbf{q}}_n^\top & \hat{q}_{n,4}\end{bmatrix}\begin{bmatrix}\bar{q}_{mb,4}\mathbf{I}_{3\times3}-\lfloor\bar{\mathbf{q}}_{mb}\times\rfloor & -\mathbf{q}_{mb}\\ \mathbf{q}_{mb}^\top & \bar{q}_{mb,4}\end{bmatrix}\begin{bmatrix}-\frac{{}^k\delta\boldsymbol{\theta}_G}{2}\\ 1\end{bmatrix}\right)\\
&= -((\hat{q}_{n,4}\mathbf{I}_{3\times3\times3}-\lfloor\hat{\mathbf{q}}_n\times\rfloor)(q_{mb,4}\mathbf{I}_{3\times3}-\lfloor\mathbf{q}_{mb}\times\rfloor)+\hat{\mathbf{q}}_n\mathbf{q}_{mb}^\top)^k\delta\boldsymbol{\theta}_G+\text{ other terms },\\
\Rightarrow\quad &\frac{\partial^{k+1}\delta\boldsymbol{\theta}_k}{\partial^k\delta\boldsymbol{\theta}_G} = -((\hat{q}_{n,4}\mathbf{I}_{3\times3}-\lfloor\hat{\mathbf{q}}_n\times\rfloor)(\bar{q}_{mb,4}\mathbf{I}_{3\times3}-\lfloor\mathbf{q}_{mb}\times\rfloor)+\hat{\mathbf{q}}_n\bar{\mathbf{q}}_{mb}^\top)\,. \qquad (41)
\end{aligned}
$$

Note than in the preceding Jacobians, we have defined several intermediate quaternions ($\hat{\bar{q}}_r$, $\hat{\bar{q}}_{rb}$, $\hat{\bar{q}}_n$, and $\hat{\bar{q}}_{mb}$) for ease of notation which can easily be interpreted from context. Following the same methodology, we can find the Jacobians of the ${}^k\boldsymbol{\alpha}_{k+1}$ measurement with respect to the position, velocity and bias.

$$
\begin{aligned}
{}^k\boldsymbol{\alpha}_{k+1} &= {}^k_G\mathbf{R}\left({}^G\mathbf{p}_{k+1}-{}^G\mathbf{p}_k-{}^G\mathbf{v}_k\Delta T+\frac{1}{2}{}^G\mathbf{g}\Delta T^2\right)-\mathbf{J}_\alpha(\mathbf{b}_w-\bar{\mathbf{b}}_w)-\mathbf{H}_\alpha(\mathbf{b}_a-\bar{\mathbf{b}}_a)\\
&\simeq \left(\mathbf{I}_{3\times3}-\lfloor{}^k\delta\boldsymbol{\theta}_G\times\rfloor\right){}^k_G\hat{\mathbf{R}}\Big({}^G\hat{\mathbf{p}}_{k+1}+{}^G\tilde{\mathbf{p}}_{k+1}-{}^G\hat{\mathbf{p}}_k-{}^G\tilde{\mathbf{p}}_k-{}^G\hat{\mathbf{v}}_k\Delta T-{}^G\tilde{\mathbf{v}}_k\Delta T\\
&\qquad +\frac{1}{2}{}^G\mathbf{g}\Delta T^2\Big)-\mathbf{J}_\alpha\left(\hat{\mathbf{b}}_w+\tilde{\mathbf{b}}_w-\bar{\mathbf{b}}_w\right)-\mathbf{H}_\alpha\left(\hat{\mathbf{b}}_a+\tilde{\mathbf{b}}_a-\bar{\mathbf{b}}_a\right)\,. \qquad (42)
\end{aligned}
$$

Then the following Jacobians immediately become available:

$$
\begin{aligned}
&\frac{\partial^k\boldsymbol{\alpha}_{k+1}}{\partial^k\delta\boldsymbol{\theta}_G} = \left\lfloor{}^k_G\hat{\mathbf{R}}\left({}^G\hat{\mathbf{p}}_{k+1}-{}^G\hat{\mathbf{p}}_k-{}^G\hat{\mathbf{v}}_k\Delta T+\frac{1}{2}{}^G\mathbf{g}\Delta T^2\right)\times\right\rfloor\,,\\
&\frac{\partial^k\boldsymbol{\alpha}_{k+1}}{\partial^G\mathbf{p}_k} = -{}^k_G\hat{\mathbf{R}}\,,\ \frac{\partial^k\boldsymbol{\alpha}_{k+1}}{\partial^G\mathbf{p}_{k+1}} = {}^k_G\hat{\mathbf{R}}\,,\ \frac{\partial^k\boldsymbol{\alpha}_{k+1}}{\partial^G\mathbf{v}_k} = -{}^k_G\hat{\mathbf{R}}\Delta T\,,\\
&\frac{\partial^k\boldsymbol{\alpha}_{k+1}}{\partial\tilde{\mathbf{b}}_w} = -\mathbf{J}_\alpha\,,\ \frac{\partial^k\boldsymbol{\alpha}_{k+1}}{\partial\tilde{\mathbf{b}}_a} = -\mathbf{H}_\alpha\,. \qquad (43)
\end{aligned}
$$

Similarly, we can write our ${}^k\boldsymbol{\beta}_{k+1}$ measurement with respect to the position, velocity and bias as:

$$
\begin{aligned}
{}^k\boldsymbol{\beta}_{k+1} &= {}^k_G\mathbf{R}\left({}^G\mathbf{v}_{k+1}-{}^G\mathbf{v}_k+{}^G\mathbf{g}\Delta T\right)-\mathbf{J}_\beta(\mathbf{b}_w-\bar{\mathbf{b}}_w)\\
&\simeq (\mathbf{I}_{3\times3}-\lfloor{}^k\delta\boldsymbol{\theta}_G\times\rfloor){}^k_G\hat{\mathbf{R}}\left({}^G\hat{\mathbf{v}}_{k+1}+{}^G\tilde{\mathbf{v}}_{k+1}-{}^G\hat{\mathbf{v}}_k-{}^G\tilde{\mathbf{v}}_k+{}^G\mathbf{g}\Delta T\right)\\
&\qquad -\mathbf{J}_\beta(\hat{\mathbf{b}}_w+\tilde{\mathbf{b}}_w-\bar{\mathbf{b}}_w)-\mathbf{H}_\beta(\hat{\mathbf{b}}_a+\tilde{\mathbf{b}}_a-\bar{\mathbf{b}}_a)\,, \qquad (44)
\end{aligned}
$$

which leads to the following Jacobians:

$$
\begin{aligned}
&\frac{\partial^k\boldsymbol{\beta}_{k+1}}{\partial^k\delta\boldsymbol{\theta}_G} = \left\lfloor{}^k_G\hat{\mathbf{R}}({}^G\hat{\mathbf{v}}_{k+1}-{}^G\hat{\mathbf{v}}_k+{}^G\mathbf{g}\Delta T)\times\right\rfloor\,,\\
&\frac{\partial^k\boldsymbol{\beta}_{k+1}}{\partial^G\mathbf{v}_k} = -{}^k_G\hat{\mathbf{R}}\,,\ \frac{\partial^k\boldsymbol{\beta}_{k+1}}{\partial^G\mathbf{v}_{k+1}} = {}^k_G\hat{\mathbf{R}}\,,\ \frac{\partial^k\boldsymbol{\beta}_{k+1}}{\partial\tilde{\mathbf{b}}_w} = -\mathbf{J}_\beta\,,\ \frac{\partial^k\boldsymbol{\beta}_{k+1}}{\partial\tilde{\mathbf{b}}_a} = -\mathbf{H}_\beta\,. \qquad (45)
\end{aligned}
$$

5 Sliding-Window Visual Tracking

Reliance on pure inertial measurements causes large drift over time due to the high noise factors, thus we rely on additional visual measurements from a camera. As an

IMU-camera sensor suite moves throughout the environment, images are taken from the mounted stereo camera and features are extracted and tracked over a window of historical camera poses. Naively, we could add these features into our state vector, thereby greatly increasing the computational burden. Instead, we seek to extract all the information contained in these measurements about the sensor suite's states. To this end, we add these features into a *local* graph containing a sliding window of states and the corresponding tracked features, and perform a local *marginalization* across these features.

In particular, the measurement model associated with feature factors involves a transformation into a camera frame, followed by a projection onto the corresponding image plane. This function for a feature j detected on an image i is given by:

$$\mathbf{z}_{ij} = \begin{bmatrix} \frac{{}^{C_i}\mathbf{p}_j(1)}{{}^{C_i}\mathbf{p}_j(3)} \\ \frac{{}^{C_i}\mathbf{p}_j(2)}{{}^{C_i}\mathbf{p}_j(3)} \end{bmatrix} + \mathbf{n}_{ij} \ , \tag{46}$$

$${}^{C_i}\mathbf{p}_j = {}^{C}_{I}\mathbf{R}\,{}^{I_i}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_j - {}^{G}\mathbf{p}_{I_i}\right) + {}^{C}\mathbf{p}_I \ , \tag{47}$$

where $\mathbf{n}_{ij} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Lambda}_{ij}^{-1}\right)$ is the zero-mean white Gaussian noise, and $\{{}^{C}_{I}\mathbf{R}, {}^{C}\mathbf{p}_I\}$ is the extrinsic calibration between the IMU and the camera, which is assumed to be known and constant over time.

Given the set of feature measurements, we seek the MLE for all the IMU poses in the window as well as the corresponding features (see Section 3). This optimization yields a normal distribution in the form (see [11, 19, 20]):

$$\begin{bmatrix} {}^{G}\mathbf{x} \\ {}^{G}\mathbf{p}_f \end{bmatrix} = \begin{bmatrix} {}^{G}\breve{\mathbf{x}} \\ {}^{G}\breve{\mathbf{p}}_f \end{bmatrix} \boxplus \begin{bmatrix} {}^{G}\delta\mathbf{x} \\ {}^{G}\delta\mathbf{p}_f \end{bmatrix}, \text{ with } \begin{bmatrix} {}^{G}\delta\mathbf{x} \\ {}^{G}\delta\mathbf{p}_f \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \boldsymbol{\Lambda}^{-1}\right) \ , \tag{48}$$

where ${}^{G}\mathbf{x}$ and ${}^{G}\mathbf{p}_f$ are the stacked sensor poses and feature positions respectively, while $(\breve{*})$ refers to the estimates achieved by the MLE optimization. Defining $\mathbf{J}_{ij}$ as the Jacobian of the ij-th measurement, the information matrix is computed as:

$$\boldsymbol{\Lambda} = \sum_{(i,j)} \mathbf{J}_{ij}^{\top} \boldsymbol{\Lambda}_{ij} \mathbf{J}_{ij} = \begin{bmatrix} \boldsymbol{\Lambda}_{ss} & \boldsymbol{\Lambda}_{sf} \\ \boldsymbol{\Lambda}_{fs} & \boldsymbol{\Lambda}_{ff} \end{bmatrix} \ . \tag{49}$$

Note that the information matrix is partitioned according to the dimensions of the states and features (s and f respectively). Marginalizing the features yields the following normal distribution of our error state:

$${}^{G}\delta\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, (\boldsymbol{\Lambda}_{ss} - \boldsymbol{\Lambda}_{sf}\boldsymbol{\Lambda}_{ff}^{-1}\boldsymbol{\Lambda}_{fs})^{-1}\right) \ . \tag{50}$$

It is important to note that this distribution encapsulates all the information in the measurements about the window nodes (up to linearization errors) [11]. Due to the lack of measurements anchoring the graph to the global frame, this optimization problem will typically be under-constrained. We therefore shift the frame of reference of the optimization problem into that of the oldest node in the window, and thus have the distribution on the *relative* states denoted by the left superscript "L" [see (48)]:

$$\begin{bmatrix} {}^{L}\mathbf{x} \\ {}^{L}\mathbf{p}_f \end{bmatrix} = \begin{bmatrix} {}^{L}\breve{\mathbf{x}} \\ {}^{L}\breve{\mathbf{p}}_f \end{bmatrix} \boxplus \begin{bmatrix} {}^{L}\delta\mathbf{x} \\ {}^{L}\delta\mathbf{p}_f \end{bmatrix}, \text{ with } \begin{bmatrix} {}^{L}\delta\mathbf{x} \\ {}^{L}\delta\mathbf{p}_f \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, {}^{L}\boldsymbol{\Lambda}^{-1}\right) \ . \tag{51}$$

Insertion back in the graph gives the following residuals for an example window of states $\{\mathbf{x}_k\}_{k=0}^{n}$, which will be used along with the IMU preintegration measurements in the graph optimization:

$$\mathbf{r}_f = \begin{bmatrix} 2\mathbf{vec}\left({}^{1}_{G}\bar{q} \otimes {}^{0}_{G}\bar{q}^{-1} \otimes {}^{1}_{0}\breve{\bar{q}}^{-1}\right) \\ {}^{0}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_1 - {}^{G}\mathbf{p}_0\right) - {}^{0}\breve{\mathbf{p}}_1 \\ \vdots \\ 2\mathbf{vec}\left({}^{n}_{G}\bar{q} \otimes {}^{0}_{G}\bar{q}^{-1} \otimes {}^{n}_{0}\breve{\bar{q}}^{-1}\right) \\ {}^{0}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_n - {}^{G}\mathbf{p}_0\right) - {}^{0}\breve{\mathbf{p}}_n \end{bmatrix} . \tag{52}$$

We therefore have extracted the information contained in the feature measurements about the states in the window while not storing features in the global graph.

6 Experimental Results

6.1 Monte-Carlo Simulations

The proposed method was implemented in C++ and tested on a MATLAB generated simulation in order to compare the proposed analytical preintegration against the discrete one [7]. A dynamic trajectory of approximately 107 meters traversed in 100 seconds, as well as a set of random 3D features were generated. Ten sets of noisy IMU and synthetic stereo image outputs were collected for Monte Carlo evaluation. Realistic noise levels and camera calibration parameters from the dataset below (see Section 6.2) were used, while feature projections were corrupted by one pixel noise. IMU measurements were generated at a rate of 100 Hz, while synthetic images were created at 10 Hz. New state nodes were created every time a synthetic image pair was collected with features being "tracked" across a sliding window of six images. The local graph problem was solved to compute the vision factors (see Section 5) when the window reached its full size. In addition to the vision and IMU preintegration measurements, bias drift factors [21] were added into the global graph to constrain the difference in biases between nodes. Both the local and full graph optimizations were performed using the GTSAM library [22] and the discrete preintegration [7] was implemented using the open source code available within GTSAM. Figure 2 shows the generated path, while the root mean square errors (RMSE) are shown in Figure 3. The proposed method achieved an ending position RMSE of 0.64m (0.6% of the distance traveled) with an average RMSE of 0.36 m and 0.35 deg across the entire path. By contrast, the same system using discrete preintegration achieved an ending RMSE of 0.74 m (0.7% of the total path) with an average of 0.41 m and 0.38 deg across the entire path. This clearly demonstrates the improvements offered by our analytical preintegration.

6.2 Real-World Experiments

The proposed algorithm was validated on one of the "EuRoC MAV Datasets" that are publicly available [23]. The datasets use two Aptina MT9V034 global shutter cameras

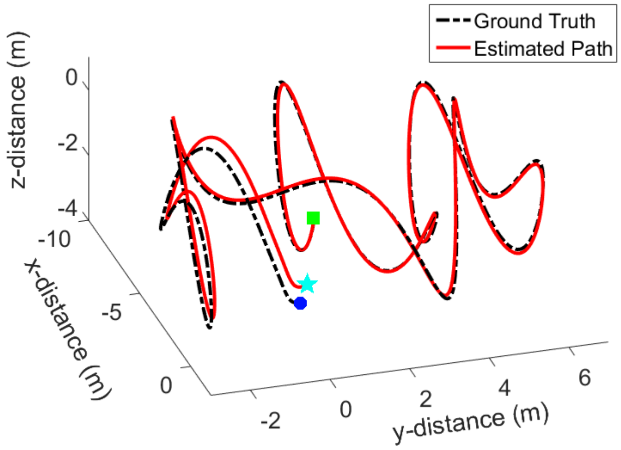

Fig. 2: Simulation results: Estimated trajectory versus the ground truth for an example Monte Carlo run. The initial start is show with a green square, with ground truth ending with a blue circle, and the estimation a teal star. Note that the discrete trajectory is not shown.

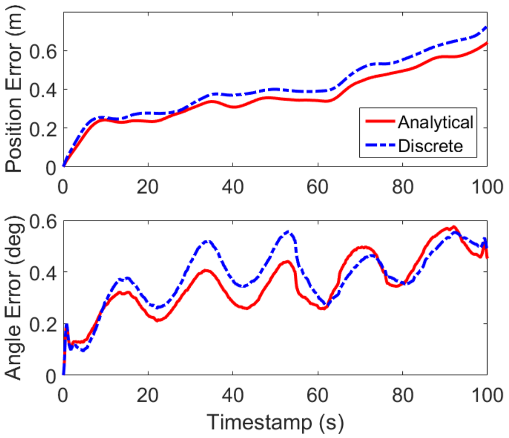

Fig. 3: Simulation results: Position and orientation RMSE for 10 Monte-Carlo simulations.

at 20 FPS with an image resolution of 752 × 480 pixels and the MEMS ADIS16448 IMU at a rate of 200 Hz. A MAV where the sensor suite is mounted, flies through the environment in a dynamic motion. The left to right images and IMU timestamps come synchronized, allowing for the inherent sensor time delay to be ignored. All extrinsic calibrations are provided including the camera to camera and IMU to camera transformations.

All images are histogram equalized to allow for better feature extraction and rectified using the OpenCV library [24]. New IMU measurements were preintegrated by “stacking” the readings over time [see (24)]. The current state was stored in a node and linked with this preintegration edge when a new stereo image pair is received. After ten images, a new feature factor was created through matching SIFT features across the first stereo pair and matching to older images using the KLT tracking method. To reject outliers, epipolar constraints are then enforced in each of the stereo pairs. In this experiment, g2o [12] was used as the graph solver, and the sliding window of ten images was chosen. The rest of the implementation was performed as explained in Section 6.1.

The Machine Hall 01 recording is 140 seconds long and provides dynamic aerial motion in an indoor area. The estimated trajectory verses the ground truth is shown in Figure 4, and the estimation errors are depicted in Figure 5. In this test, the proposed approach attains the position error of 0.5m, approximately 0.7% of the total distance traveled. While this is a similar level to the simulations, the angle estimation was worse in the real world-experiments. This is likely due to weakness in the visual front-end: improved robustness of image processing before local estimation is expected to greatly improve performance. Nevertheless, along with the Monte-Carlo simulation results, these real-world experiments clearly validate the proposed analytical preintegration for graph-based VINS.

7 Conclusions and Future Work

We have introduced a high-accuracy preintegration theory based on the *closed-form* solutions of the IMU integration equations, which allow us to accurately compute the mean, covariance, and bias Jacobians for each of the preintegration factors. This theory was integrated into graph-based VINS system and validated on both synthetic and real data. As currently our system implementation does not include loop closures and thus does not fully gain the benefits of the batch optimization, in the future, we will include these constraints, which would allow us to further improve accuracy by taking advantage of the graph formulation. In addition, we will seek to reduce the computational complexity of our system by intelligently sparsifying the graph so as to enable long-term and large-scale robot navigation.

References

[1] Hesch, J., Kottas, D., Bowman, S., Roumeliotis, S.: Camera-IMU-based localization: Observability analysis and consistency improvement. International Journal of Robotics Research **33** (2014) 182–201

[2] Li, M., Mourikis, A.: High-precision, consistent EKF-based visual-inertial odometry. International Journal of Robotics Research **32**(6) (2013) 690–711

[3] Huang, G., Kaess, M., Leonard, J.: Towards consistent visual-inertial navigation. In: Proc. of the IEEE International Conference on Robotics and Automation, Hong Kong, China (May 31–June 7, 2014) 4926–4933

[4] Kumar, V., Michael, N.: Opportunities and challenges with autonomous micro aerial vehicles. International Journal of Robotics Research **31**(11) (September 2012) 1279–1291

[5] Dellaert, F., Kaess, M.: Square root SAM: Simultaneous localization and mapping via square root information smoothing. International Journal of Robotics Research **25**(12) (Dec. 2006) 1181–1203

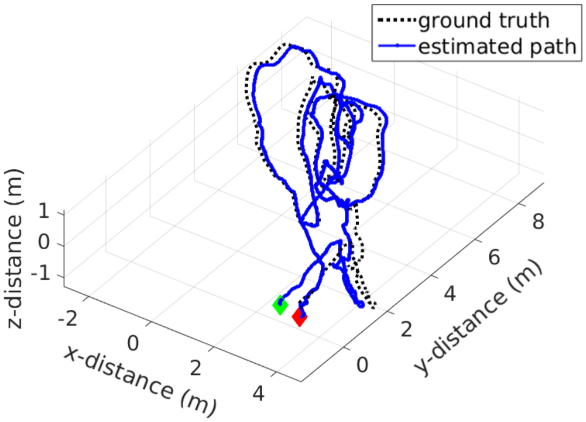

Fig. 4: Experimental results: Estimated trajectory versus the ground truth. The initial start is show with a red diamond and the ending location is shown with a green diamond.

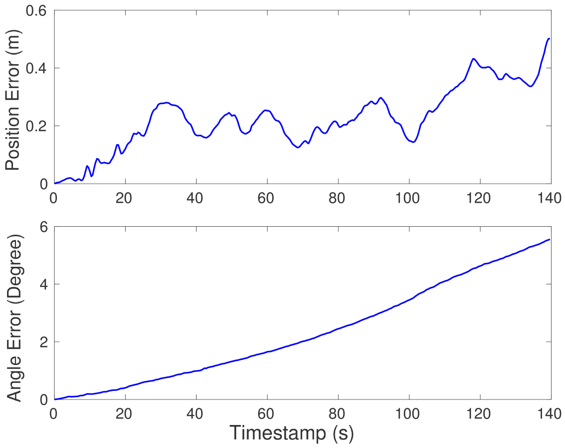

Fig. 5: Experimental results: Position and orientation estimation errors over the Machine Hall 01 dataset.

[6] Lupton, T., Sukkarieh, S.: Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. IEEE Transactions on Robotics **28**(1) (February 2012) 61 –76

[7] Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D.: Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In: Robotics: Science and Systems XI. Number EPFL-CONF-214687 (2015)

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

[9] Mourikis, A.I., Roumeliotis, S.I.: A multi-state constraint Kalman filter for vision-aided

inertial navigation. In: Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy (April 10–14, 2007) 3565–3572

[10] Roumeliotis, S.I., Burdick, J.W.: Stochastic cloning: A generalized framework for processing relative state measurements. In: Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC (May 11-15 2002) 1788–1795

[11] Huang, G., Eckenhoff, K., Leonard, J.: Optimal-state-constraint EKF for visual-inertial navigation. In: Proc. of the International Symposium on Robotics Research, Sestri Levante, Italy (September 12–15, 2015)

[12] Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g2o: A general framework for graph optimization. In: Proc. of the IEEE International Conference on Robotics and Automation, Shanghai, China (May 9–13, 2011) 3607–3613

[13] Indelman, V., Williams, S., Kaess, M., Dellaert, F.: Factor graph based incremental smoothing in inertial navigation systems. In: Proc. of the International Conference on Information Fusion, Singapore (July 2012)

[14] Indelman, V., Melim, A., Dellaert, F.: Incremental light bundle adjustment for robotics navigation. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE (2013) 1952–1959

[15] Trawny, N., Roumeliotis, S.I.: Indirect Kalman filter for 3D attitude estimation. Technical report, University of Minnesota, Dept. of Comp. Sci. & Eng. (March 2005)

[16] Lupton, T., Sukkarieh, S.: Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. IEEE Transactions on Robotics **28**(1) (Feb 2012) 61–76

[17] Eckenhoff, K., Geneva, P., Huang, G.: High-accuracy preintegration for visual inertial navigation. Technical Report RPNG-2016-001, University of Delaware (2016) Available: `http://udel.edu/~ghuang/papers/tr_hapi.pdf`.

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

[19] Huang, G.P., Mourikis, A.I., Roumeliotis, S.I.: An observability constrained sliding window filter for SLAM. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA (September 25–30, 2011) 65–72

[20] Huang, G., Kaess, M., Leonard, J.: Consistent sparsification for graph optimization. In: Proc. of the European Conference on Mobile Robots, Barcelona, Spain (September 25–27, 2013) 150–157

[21] Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: Fast semi-direct monocular visual odometry. In: Proc. of the IEEE International Conference on Robotics and Automation, Hong Kong, China (May 2014)

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

[23] Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., Siegwart, R.: The euroc micro aerial vehicle datasets. The International Journal of Robotics Research (2016)

[24] OpenCV Developers Team: Open source computer vision (OpenCV) library. Available: `http://opencv.org`