On-Manifold Preintegration for Real-Time Visual-Inertial Odometry

4 authors:

Christian Forster
University of Zurich
20 PUBLICATIONS   **3,473** CITATIONS

SEE PROFILE

Luca Carlone
Massachusetts Institute of Technology
130 PUBLICATIONS   **3,726** CITATIONS

SEE PROFILE

Frank Dellaert
Georgia Institute of Technology
252 PUBLICATIONS   **18,092** CITATIONS

SEE PROFILE

Davide Scaramuzza
University of Zurich
247 PUBLICATIONS   **14,739** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Decentralized Visual SLAM View project

Event-based Vision View project

On-Manifold Preintegration for Real-Time Visual-Inertial Odometry

Christian Forster, Luca Carlone, Frank Dellaert, Davide Scaramuzza

Abstract—Current approaches for visual-inertial odometry (VIO) are able to attain highly accurate state estimation via nonlinear optimization. However, real-time optimization quickly becomes infeasible as the trajectory grows over time; this problem is further emphasized by the fact that inertial measurements come at high rate, hence leading to fast growth of the number of variables in the optimization. In this paper, we address this issue by preintegrating inertial measurements between selected keyframes into single relative motion constraints. Our first contribution is a *preintegration theory* that properly addresses the manifold structure of the rotation group. We formally discuss the generative measurement model as well as the nature of the rotation noise and derive the expression for the *maximum a posteriori* state estimator. Our theoretical development enables the computation of all necessary Jacobians for the optimization and a-posteriori bias correction in analytic form. The second contribution is to show that the preintegrated IMU model can be seamlessly integrated into a visual-inertial pipeline under the unifying framework of factor graphs. This enables the application of incremental-smoothing algorithms and the use of a *structureless* model for visual measurements, which avoids optimizing over the 3D points, further accelerating the computation. We perform an extensive evaluation of our monocular VIO pipeline on real and simulated datasets. The results confirm that our modelling effort leads to accurate state estimation in real-time, outperforming state-of-the-art approaches.

Supplementary Material

- Video of the experiments: https://youtu.be/CsJkci5lfco
- Source-code for preintegrated IMU and structureless vision factors https://bitbucket.org/gtborg/gtsam.

I. Introduction

The use of cameras and inertial sensors for three-dimensional structure and motion estimation has received considerable attention from the robotics community. Both sensor types are cheap, ubiquitous, and complementary. A single moving camera is an exteroceptive sensor that allows us to measure appearance and geometry of a three-dimensional scene, up to an unknown metric scale; an inertial measurement unit (IMU) is a proprioceptive sensor that renders metric scale

C. Forster is with the Robotics and Perception Group, University of Zurich, Switzerland. E-mail: forster@ifi.uzh.ch

L. Carlone is with the Laboratory for Information & Decision Systems, Massachusetts Institute of Technology, USA. E-mail: lcarlone@mit.edu

F. Dellaert is with the College of Computing, Georgia Institute of Technology, USA. E-mail: dellaert@cc.gatech.edu

D. Scaramuzza is with the Robotics and Perception Group, University of Zurich, Switzerland. E-mail: sdavide@ifi.uzh.ch

of monocular vision and gravity observable [1] and provides robust and accurate inter-frame motion estimates. Applications of VIO range from autonomous navigation in GPS-denied environments, to 3D reconstruction, and augmented reality.

The existing literature on VIO imposes a trade-off between accuracy and computational efficiency (a detailed review is given in Section II). On the one hand, filtering approaches enable fast inference, but their accuracy is deteriorated by the accumulation of linearization errors. On the other hand, full smoothing approaches, based on nonlinear optimization, are accurate, but computationally demanding. Fixed-lag smoothing offers a compromise between accuracy for efficiency; however, it is not clear how to set the length of the estimation window so to guarantee a given level of performance.

In this work we show that it is possible to overcome this trade-off. We design a VIO system that enables fast incremental smoothing and computes the optimal *maximum a posteriori* (MAP) estimate in real time. An overview of our approach is given in Section IV.

The first step towards this goal is the development of a novel preintegration theory. The use of *preintegrated IMU measurements* was first proposed in [2] and consists of combining many inertial measurements between two keyframes into a single relative motion constraint. We build upon this work and present a preintegration theory that properly addresses the manifold structure of the rotation group SO(3). Our preintegration theory is presented in Sections V-VI. Compared with [2], our theory offers a more formal treatment of the rotation noise, and avoids singularities in the representation of rotations. Furthermore, we are able to derive all necessary Jacobians in analytic form: specifically, we report the analytic Jacobians of the residuals, the noise propagation, and the a-posteriori bias correction in the appendix of this paper.

Our second contribution is to frame the IMU preintegration theory into a factor graph model. This enables the application of incremental smoothing algorithms, as iSAM2 [3], which avoid the accumulation of linearization errors and offer an elegant way to trade-off accuracy with efficiency. Inspired by [4, 5], we also adopt a *structureless* model for visual measurements, which allows eliminating a large number of variables (*i.e.*, all 3D points) during incremental smoothing, further accelerating the computation (Section VII). In contrast to [5], we use the structureless model in an incremental smoothing framework. This has two main advantages: we do not need to delay the processing of visual measurements, and we can relinearize the visual measurements multiple times.

In order to demonstrate the effectiveness of our model, we integrated the proposed IMU preintegration in a state-of-the-art VIO pipeline and tested it on real and simulated datasets

(Sections VIII). Our theoretical development leads to tangible practical advantages: an implementation of the approach proposed in this paper performs full-smoothing at a rate of 100 Hz and achieves superior accuracy with respect to competitive state-of-the-art filtering and optimization approaches.

Besides the technical contribution, the paper also provides a tutorial contribution for practitioners. In Section III and across the paper, we provide a short but concise summary of uncertainty representation on manifolds and exemplary derivations for uncertainty propagation and Jacobian computation. The complete derivation of all equations and Jacobians – necessary to implement our model – are given in the appendix.

This paper is an extension of our previous work [6] with additional experiments, an in-depth discussion of related work, and comprehensive technical derivations. The results of the new experiments highlight the accuracy of bias estimation, demonstrate the consistency of our approach, and provide comparisons against full batch estimation. We release our implementation of the preintegrated IMU and structureless vision factors in the GTSAM 4.0 optimization toolbox [7].

II. RELATED WORK

Related work on visual-inertial odometry can be sectioned along three main dimensions. The first dimension is the number of camera-poses involved in the estimation. While *full smoothers* (or *batch nonlinear least-squares* algorithms) estimate the complete history of poses, *fixed-lag smoothers* (or *sliding window estimators*) consider a window of the latest poses, and *filtering* approaches only estimate the latest state. Both fixed-lag smoothers and filters marginalize older states and absorb the corresponding information in a Gaussian prior.

The second dimension regards the representation of the uncertainty for the measurements and the Gaussian priors: the *Extended Kalman Filter* (EKF) represents the uncertainty using a covariance matrix; instead, *information filters* and smoothers resort to the information matrix (the inverse of the covariance) or the square-root of the information matrix [3, 8].

Finally, the third dimension distinguishes existing approaches by looking at the number of times in which the measurement model is linearized. While a standard EKF (in contrast to the *iterated* EKF) processes a measurement only once, a smoothing approach allows linearizing multiple times.

While the terminology is vast, the underlying algorithms are tightly related. For instance, it can be shown that the iterated Extended Kalman filter equations are equivalent to the Gauss-Newton algorithm, commonly used for smoothing [9].

A. Filtering

Filtering algorithms enable efficient estimation by restricting the inference process to the latest state of the system. The complexity of the EKF grows quadratically in the number of estimated landmarks, therefore, a small number of landmarks (in the order of 20) are typically tracked to allow real-time operation [10–12]. An alternative is to adopt a "structureless" approach where landmark positions are marginalized out of the state vector. An elegant example of this strategy is the

Multi-State Constraint Kalman filter (MSC-KF) [5]. The structureless approach requires to keep previous poses in the state vector, by means of *stochastic cloning* [13].

A drawback of using a structureless approach for filtering, is that the processing of landmark measurements needs to be delayed until all measurements of a landmark are obtained [5]. This hinders accuracy as the filter cannot use all current visual information. Marginalization is also a source of errors as it locks in linearization errors and erroneous outlier measurements. Therefore, it is particularly important to filter out spurious measurements as a single outlier can irreversibly corrupt the filter [14]. Further, linearization errors introduce drift in the estimate and render the filter *inconsistent*. An effect of inconsistency is that the estimator becomes over-confident, resulting in non-optimal information fusion. Generally, the VIO problem has four unobservable directions: the global position and the orientation around the gravity direction (yaw) [15, 16]. In [16] it is shown that linearization at the wrong estimate results in only three unobservable directions (the global position); hence, erroneous linearization adds spurious information in yaw direction to the Gaussian prior, which renders the filter inconsistent. This problem was addressed with the *first-estimates jacobian* approach [17], which ensures that a state is not updated with different linearization points — a source of inconsistency. In the *observability-constrained* EKF (OC-EKF) an estimate of the unobservable directions is maintained which allows to update the filter only in directions that are observable [16, 18]. A thorough analysis of VIO observability properties is given in [1, 15, 19].

B. Fixed-lag Smoothing

Fixed-lag smoothers estimate the states that fall within a given time window, while marginalizing out older states [20–24]. In a maximum likelihood estimation setup, fixed-lag smoothers lead to an optimization problem over a set of recent states. For nonlinear problems, smoothing approaches are generally more accurate than filtering, since they relinearize past measurements [25]. Moreover, these approaches are more resilient to outliers, which can be discarded a posteriori (i.e., after the optimization), or can be alleviated by using robust cost functions. On the downside, the marginalization of the states outside the estimation window leads to dense Gaussian priors which hinder efficient inference. For this reason, it has been proposed to drop certain measurements in the interest of sparsity [24]. Furthermore, due to marginalization, fixed-lag smoothers share part of the issues of filtering (consistency, build-up of linearization errors) [18, 22, 26].

C. Full Smoothing

Full smoothing methods estimate the entire history of the states (camera trajectory and 3D landmarks), by solving a large nonlinear optimization problem [27–31]. Full smoothing guarantees the highest accuracy; however, real-time operation quickly becomes infeasible as the trajectory and the map grow over time. Therefore, it has been proposed to discard frames except selected *keyframes* [24, 32–34] or to run the optimization in a parallel thread, using a tracking

and mapping dual architecture [20, 35]. A breakthrough has been the development of *incremental smoothing techniques* (iSAM [36], iSAM2 [3]), which leverage the expressiveness of *factor graphs* to maintain sparsity and to identify and update only the typically small subset of variables affected by a new measurement.

Nevertheless, the high rate of inertial measurements (usually 100 Hz to 1 kHz) still constitutes a challenge for smoothing approaches. A naive implementation would require adding a new state at every IMU measurement, which quickly becomes impractically slow [37]. Therefore, inertial measurements are typically integrated between frames to form relative motion constraints [24, 30, 38–40]. For standard IMU integration between two frames, the initial condition is given by the state estimate at the first frame. However, at every iteration of the optimization, the state estimate changes, which requires to repeat the IMU integration between all frames [24]. Lupton and Sukkarieh [2] show that this repeated integration can be avoided by a reparametrization of the relative motion constraints. Such reparametrization is called *IMU preintegration*.

In the present work, we build upon the seminal work [2] and bring the theory of IMU preintegration to maturity by properly addressing the manifold structure of the rotation group SO(3). The work [2] adopted Euler angles as global parametrization for rotations. Using Euler angles and applying the usual averaging and smoothing techniques of Euclidean spaces for state propagation and covariance estimation is not properly invariant under the action of rigid transformations [41, 42]. Moreover, Euler angles are known to have singularities. Our work, on the other hand, provides a formal treatment of the rotation measurements (and the corresponding noise), and provides a complete derivation of the maximum a posteriori estimator. We also derive analytic expressions for the Jacobians (needed for the optimization), which, to the best of our knowledge, have not been previously reported in the literature. In the experimental section, we show that a proper representation of the rotation manifold results in higher accuracy and robustness, leading to tangible advantages over the original proposal [2].

III. PRELIMINARIES

In this paper we formulate VIO in terms of MAP estimation. In our model, MAP estimation leads to a nonlinear optimization problem that involves quantities living on smooth manifolds (e.g., rotations, poses). Therefore, before delving into details, we conveniently review some useful geometric concepts. This section can be skipped by the expert reader.

We structure this section as follows: Section III-A provides useful notions related to two main Riemannian manifolds: the Special Orthogonal Group SO(3) and the Special Euclidean Group SE(3). Our presentation is based on [43, 44]. Section III-B describes a suitable model to describe uncertain rotations in SO(3). Section III-C reviews optimization on manifolds, following standard references [45].

A. Notions of Riemannian geometry

a) Special Orthogonal Group: SO(3) describes the group of 3D rotation matrices and it is formally defined as
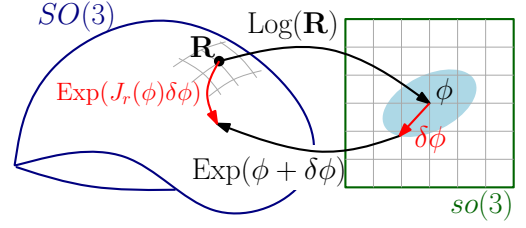

Fig. 1: The right Jacobian J_r relates an additive perturbation $\delta\phi$ in the tangent space to a multiplicative perturbation on the manifold SO(3), as per Eq. (7).

$SO(3) \doteq \{R \in \mathbb{R}^{3\times3} : R^\mathsf{T}R = I, \det(R) = 1\}$. The group operation is the usual matrix multiplication, and the inverse is the matrix transpose. The group SO(3) also forms a smooth manifold. The tangent space to the manifold (at the identity) is denoted as $\mathfrak{so}(3)$, which is also called the *Lie algebra* and coincides with the space of 3×3 skew symmetric matrices. We can identify every skew symmetric matrix with a vector in $\mathbb{R}^3$ using the *hat* operator:

$$\boldsymbol{\omega}^\wedge = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathfrak{so}(3). \quad (1)$$

Similarly, we can map a skew symmetric matrix to a vector in $\mathbb{R}^3$ using the *vee* operator $(\cdot)^\vee$: for a skew symmetric matrix $S = \boldsymbol{\omega}^\wedge$, the vee operator is such that $S^\vee = \boldsymbol{\omega}$. A property of skew symmetric matrices that will be useful later on is:

$$\mathbf{a}^\wedge \mathbf{b} = -\mathbf{b}^\wedge \mathbf{a}, \quad \forall\, \mathbf{a}, \mathbf{b} \in \mathbb{R}^3. \quad (2)$$

The *exponential map* (at the identity) $\exp : \mathfrak{so}(3) \to SO(3)$ associates an element of the Lie Algebra to a rotation and coincides with standard matrix exponential (Rodrigues' formula):

$$\exp(\boldsymbol{\phi}^\wedge) = I + \frac{\sin(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}\|}\boldsymbol{\phi}^\wedge + \frac{1-\cos(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}\|^2}\left(\boldsymbol{\phi}^\wedge\right)^2. \quad (3)$$

A first-order approximation of the exponential map that we will use later on is:

$$\exp(\boldsymbol{\phi}^\wedge) \approx I + \boldsymbol{\phi}^\wedge. \quad (4)$$

The *logarithm map* (at the identity) associates a matrix $R \neq I$ in SO(3) to a skew symmetric matrix:

$$\log(R) = \frac{\varphi \cdot (R - R^\mathsf{T})}{2\sin(\varphi)} \text{ with } \varphi = \cos^{-1}\left(\frac{\mathrm{tr}\,(R) - 1}{2}\right). \quad (5)$$

Note that $\log(R)^\vee = \mathbf{a}\varphi$, where $\mathbf{a}$ and φ are the rotation axis and the rotation angle of R, respectively. If $R = I$, then $\varphi = 0$ and $\mathbf{a}$ is undetermined and can therefore be chosen arbitrarily.

The exponential map is a bijection if restricted to the open ball $\|\boldsymbol{\phi}\| < \pi$, and the corresponding inverse is the logarithm map. However, if we do not restrict the domain, the exponential map becomes surjective as every vector $\boldsymbol{\phi} = (\varphi + 2k\pi)\mathbf{a}$, $k \in \mathbb{Z}$ would be an admissible logarithm of R.

For notational convenience, we adopt "vectorized" versions of the exponential and logarithm map:

$$\begin{array}{llllll} \mathrm{Exp}: & \mathbb{R}^3 & \to & SO(3) & ; & \boldsymbol{\phi} & \mapsto & \exp(\boldsymbol{\phi}^\wedge) \\ \mathrm{Log}: & SO(3) & \to & \mathbb{R}^3 & ; & R & \mapsto & \log(R)^\vee, \end{array} \quad (6)$$

which operate directly on vectors, rather than on skew symmetric matrices in $\mathfrak{so}(3)$.

Later, we will use the following first-order approximation:

$$\text{Exp}(\phi + \delta\phi) \approx \text{Exp}(\phi)\,\text{Exp}(\text{J}_r(\phi)\delta\phi). \qquad (7)$$

The term $\text{J}_r(\phi)$ is the *right Jacobian* of SO(3) [43, p.40] and relates additive increments in the tangent space to multiplicative increments applied on the right-hand-side (Fig. 1):

$$\text{J}_r(\phi) = \mathbf{I} - \frac{1 - \cos(\|\phi\|)}{\|\phi\|^2}\phi^\wedge + \frac{\|\phi\| - \sin(\|\phi\|)}{\|\phi^3\|}(\phi^\wedge)^2. \qquad (8)$$

A similar first-order approximation holds for the logarithm:

$$\text{Log}\big(\,\text{Exp}(\phi)\,\text{Exp}(\delta\phi)\,\big) \approx \phi + \text{J}_r^{-1}(\phi)\delta\phi. \qquad (9)$$

Where the inverse of the right Jacobian is

$$\text{J}_r^{-1}(\phi) = \mathbf{I} + \frac{1}{2}\phi^\wedge + \left(\frac{1}{\|\phi\|^2} + \frac{1 + \cos(\|\phi\|)}{2\|\phi\|\sin(\|\phi\|)}\right)(\phi^\wedge)^2.$$

The right Jacobian $\text{J}_r(\phi)$ and its inverse $\text{J}_r^{-1}(\phi)$ reduce to the identity matrix for $\|\phi\| = 0$.

Another useful property of the exponential map is:

$$\text{R}\,\text{Exp}(\phi)\,\text{R}^\mathsf{T} = \exp(\text{R}\phi^\wedge\text{R}^\mathsf{T}) = \text{Exp}(\text{R}\phi) \qquad (10)$$

$$\Leftrightarrow \quad \text{Exp}(\phi)\,\text{R} = \text{R}\,\text{Exp}(\text{R}^\mathsf{T}\phi). \qquad (11)$$

b) Special Euclidean Group: SE(3) describes the group of rigid motion in 3D, which is the semi-direct product of SO(3) and $\mathbb{R}^3$, and it is defined as SE(3) $\doteq \{(\text{R}, \mathbf{p}) : \text{R} \in$ SO(3), $\mathbf{p} \in \mathbb{R}^3\}$. Given $\text{T}_1, \text{T}_2 \in$ SE(3), the group operation is $\text{T}_1 \cdot \text{T}_2 = (\text{R}_1\text{R}_2\ ,\ \mathbf{p}_1 + \text{R}_1\mathbf{p}_2)$, and the inverse is $\text{T}_1^{-1} = (\text{R}_1^\mathsf{T}\ ,\ -\text{R}_1^\mathsf{T}\mathbf{p}_1)$. The *exponential map* and the *logarithm map* for SE(3) are defined in [44]. However, these are not needed in this paper for reasons that will be clear in Section III-C.

B. Uncertainty Description in SO(3)

A natural definition of uncertainty in SO(3) is to define a distribution in the tangent space, and then map it to SO(3) via the exponential map (6) [44, 46, 47]:

$$\tilde{\text{R}} = \text{R}\,\text{Exp}(\epsilon), \qquad \epsilon \sim \mathcal{N}(0, \Sigma), \qquad (12)$$

where R is a given noise-free rotation (the *mean*) and ϵ is a small normally distributed perturbation with zero mean and covariance Σ.

To obtain an explicit expression for the distribution of $\tilde{\text{R}}$, we start from the integral of the Gaussian distribution in $\mathbb{R}^3$:

$$\int_{\mathbb{R}^3} p(\epsilon)d\epsilon = \int_{\mathbb{R}^3} \alpha e^{-\frac{1}{2}\|\epsilon\|_\Sigma^2}d\epsilon = 1, \qquad (13)$$

where $\alpha = 1/\sqrt{(2\pi)^3 \det(\Sigma)}$ and $\|\epsilon\|_\Sigma^2 \doteq \epsilon^\mathsf{T}\Sigma^{-1}\epsilon$ is the squared Mahalanobis distance with covariance Σ. Then, applying the change of coordinates $\epsilon = \text{Log}(\text{R}^{-1}\tilde{\text{R}})$ (this is the inverse of (12) when $\|\epsilon\| < \pi$), the integral (13) becomes:

$$\int_{\text{SO}(3)} \beta(\tilde{\text{R}})\,e^{-\frac{1}{2}\|\text{Log}(\text{R}^{-1}\tilde{\text{R}})\|_\Sigma^2}\,d\tilde{\text{R}} = 1, \qquad (14)$$

where $\beta(\tilde{\text{R}})$ is a normalization factor. The normalization factor assumes the form $\beta(\tilde{\text{R}}) = \alpha/|\det(\mathcal{J}(\tilde{\text{R}}))|$, where $\mathcal{J}(\tilde{\text{R}}) \doteq \text{J}_r(\text{Log}(\text{R}^{-1}\tilde{\text{R}}))$ and $\text{J}_r(\cdot)$ is the right Jacobian (8); $\mathcal{J}(\tilde{\text{R}})$ is a by-product of the change of variables, see [46] for a derivation.

From the argument of (14) we can directly read our "Gaussian" distribution in SO(3):

$$p(\tilde{\text{R}}) = \beta(\tilde{\text{R}})\,e^{-\frac{1}{2}\|\text{Log}(\text{R}^{-1}\tilde{\text{R}})\|_\Sigma^2}. \qquad (15)$$

For small covariances we can approximate $\beta \simeq \alpha$, as $\text{J}_r(\text{Log}(\text{R}^{-1}\tilde{\text{R}}))$ is well approximated by the identity matrix when $\tilde{\text{R}}$ is close to R. Note that (14) already assumes relatively a small covariance Σ, since it "clips" the probability tails outside the open ball of radius π (this is due to the re-parametrization $\epsilon = \text{Log}(\text{R}^{-1}\tilde{\text{R}})$, which restricts ϵ to $\|\epsilon\| < \pi$). Approximating β as a constant, the negative log-likelihood of a rotation R, given a measurement $\tilde{\text{R}}$ distributed as in (15), is:

$$\mathcal{L}(\text{R}) = \frac{1}{2}\left\|\text{Log}(\text{R}^{-1}\tilde{\text{R}})\right\|_\Sigma^2 + \text{const} = \frac{1}{2}\left\|\text{Log}(\tilde{\text{R}}^{-1}\text{R})\right\|_\Sigma^2 + \text{const}, \qquad (16)$$

which geometrically can be interpreted as the squared angle (geodesic distance in SO(3)) between $\tilde{\text{R}}$ and R weighted by the inverse uncertainty Σ^{-1}.

C. Gauss-Newton Method on Manifold

A standard Gauss-Newton method in Euclidean space works by repeatedly optimizing a quadratic approximation of the (generally non-convex) objective function. Solving the quadratic approximation reduces to solving a set of linear equations (*normal equations*), and the solution of this local approximation is used to update the current estimate. Here we recall how to extend this approach to (unconstrained) optimization problems whose variables belong to some manifold $\mathcal{M}$.

Let us consider the following optimization problem:

$$\min_{x \in \mathcal{M}} f(x), \qquad (17)$$

where the variable x belongs to a manifold $\mathcal{M}$; for the sake of simplicity we consider a single variable in (17), while the description easily generalizes to multiple variables.

Contrarily to the Euclidean case, one cannot directly approximate (17) as a quadratic function of x. This is due to two main reasons. First, working directly on x leads to an over-parametrization of the problem (e.g., we parametrize a rotation matrix with 9 elements, while a 3D rotation is completely defined by a vector in $\mathbb{R}^3$) and this can make the *normal equations* under-determined. Second, the solution of the resulting approximation does not belong to $\mathcal{M}$ in general.

A standard approach for optimization on manifold [45, 48], consists of defining a *retraction* $\mathcal{R}_x$, which is a bijective map between an element δx of the tangent space (at x) and a neighborhood of $x \in \mathcal{M}$. Using the retraction, we can re-parametrize our problem as follows:

$$\min_{x \in \mathcal{M}} f(x) \quad \Rightarrow \quad \min_{\delta x \in \mathbb{R}^n} f(\mathcal{R}_x(\delta x)). \qquad (18)$$

The re-parametrization is usually called *lifting* [45]. Roughly speaking, we work in the tangent space defined at the current estimate, which locally behaves as an Euclidean space. The use of the retraction allows framing the optimization problem over an Euclidean space of suitable dimension (e.g., $\delta x \in \mathbb{R}^3$ when we work in SO(3)). We can now apply standard optimization
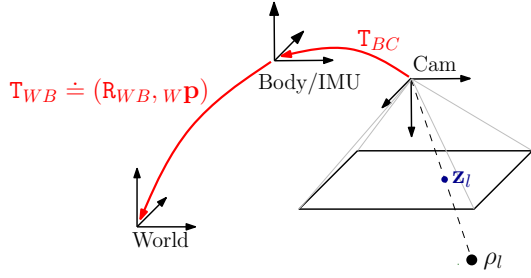
Fig. 2: $\mathsf{T}_{\mathrm{WB}} \doteq (\mathsf{R}_{\mathrm{WB}}, {}_\mathrm{W}\mathbf{p})$ is the pose of the body frame B w.r.t. the world frame W. We assume that the body frame coincides with the IMU frame. T_{BC} is the pose of the camera in the body frame, known from prior calibration.

techniques to the problem on the right-hand side of (18). In the Gauss-Newton framework, we square the cost around the current estimate. Then we solve the quadratic approximation to get a vector $\delta x^\star$ in the tangent space. Finally, the current guess on the manifold is updated as

$$\hat{x} \leftarrow \mathcal{R}_{\hat{x}}(\delta x^\star). \qquad (19)$$

This "lift-solve-retract" scheme can be generalized to any trust-region method [45]. Moreover, it provides a grounded and unifying generalization of the *error state model*, commonly used in aerospace literature for filtering [49] and recently adopted in robotics for optimization [23, 34].

We conclude this section by discussing the choice of the retraction $\mathcal{R}_x$. A possible retraction is the exponential map. It is known that, computationally, this may not be the most convenient choice, see [50].

In this work, we use the following retraction for SO(3),

$$\mathcal{R}_{\mathsf{R}}(\boldsymbol{\phi}) = \mathsf{R}\,\mathrm{Exp}(\delta\boldsymbol{\phi}), \qquad \delta\boldsymbol{\phi} \in \mathbb{R}^3, \qquad (20)$$

and for SE(3), we use the retraction at $\mathsf{T} \doteq (\mathsf{R}, \mathbf{p})$:

$$\mathcal{R}_{\mathsf{T}}(\delta\boldsymbol{\phi}, \delta\mathbf{p}) = (\mathsf{R}\,\mathrm{Exp}(\delta\boldsymbol{\phi}),\ \mathbf{p} + \mathsf{R}\,\delta\mathbf{p}), \qquad [\delta\boldsymbol{\phi}\ \ \delta\mathbf{p}] \in \mathbb{R}^6, \qquad (21)$$

which explains why in Section III-A we only defined the exponential map for SO(3): with this choice of retraction we never need to compute the exponential map for SE(3).

IV. MAXIMUM A POSTERIORI VISUAL-INERTIAL STATE ESTIMATION

We consider a VIO problem in which we want to track the state of a *sensing system* (e.g., a mobile robot, a UAV, or a hand-held device), equipped with an IMU and a monocular camera. We assume that the IMU frame "B" coincides with the body frame we want to track, and that the transformation between the camera and the IMU is fixed and known from prior calibration (Fig. 2). Furthermore, we assume that a *front-end* provides image measurements of 3D landmarks at unknown position. The front-end also selects a subset of images, called *keyframes* [32], for which we want to compute a pose estimate. Section VIII-B1 discusses implementation aspects, including the choice of the front-end in our experiments.

A. The State

The state of the system at time i is described by the IMU orientation, position, velocity and biases:

$$\mathbf{x}_i \doteq [\mathsf{R}_i, \mathbf{p}_i, \mathbf{v}_i, \mathbf{b}_i]. \qquad (22)$$

The pose $(\mathsf{R}_i, \mathbf{p}_i)$ belongs to SE(3), while velocities live in a vector space, i.e., $\mathbf{v}_i \in \mathbb{R}^3$. IMU biases can be written as $\mathbf{b}_i = [\mathbf{b}_i^g\ \ \mathbf{b}_i^a] \in \mathbb{R}^6$, where $\mathbf{b}_i^g, \mathbf{b}_i^a \in \mathbb{R}^3$ are the gyroscope and accelerometer bias, respectively.

Let $\mathcal{K}_k$ denote the set of all keyframes up to time k. In our approach we estimate the state of all keyframes:

$$\mathcal{X}_k \doteq \{\mathbf{x}_i\}_{i \in \mathcal{K}_k}. \qquad (23)$$

In our implementation, we adopt a structureless approach (*cf.*, Section VII), hence the 3D landmarks are not part of the variables to be estimated. However, the proposed approach generalizes in a straightforward manner to also estimating the landmarks and the camera intrinsic and extrinsic calibration parameters.

B. The Measurements

The input to our estimation problem are the measurements from the camera and the IMU. We denote with $\mathcal{C}_i$ the image measurements at keyframe i. At time i, the camera can observe multiple landmarks l, hence $\mathcal{C}_i$ contains multiple image measurements $\mathbf{z}_{il}$. With slight abuse of notation we write $l \in \mathcal{C}_i$ when a landmark l is seen at time i.

We denote with $\mathcal{I}_{ij}$ the set of IMU measurements acquired between two consecutive keyframes i and j. Depending on the IMU measurement rate and the frequency of selected keyframes, each set $\mathcal{I}_{ij}$ can contain from a small number to hundreds of IMU measurements. The set of measurements collected up to time k is

$$\mathcal{Z}_k \doteq \{\mathcal{C}_i, \mathcal{I}_{ij}\}_{(i,j) \in \mathcal{K}_k}. \qquad (24)$$

C. Factor Graphs and MAP Estimation

The posterior probability of the variables $\mathcal{X}_k$, given the available visual and inertial measurements $\mathcal{Z}_k$ and priors $p(\mathcal{X}_0)$ is:

$$p(\mathcal{X}_k | \mathcal{Z}_k) \propto\ p(\mathcal{X}_0)p(\mathcal{Z}_k | \mathcal{X}_k) \overset{(a)}{=} p(\mathcal{X}_0) \prod_{(i,j) \in \mathcal{K}_k} p(\mathcal{C}_i, \mathcal{I}_{ij} | \mathcal{X}_k)$$

$$\overset{(b)}{=} p(\mathcal{X}_0) \prod_{(i,j) \in \mathcal{K}_k} p(\mathcal{I}_{ij} | \mathbf{x}_i, \mathbf{x}_j) \prod_{i \in \mathcal{K}_k} \prod_{l \in \mathcal{C}_i} p(\mathbf{z}_{il} | \mathbf{x}_i). \qquad (25)$$

The factorizations (a) and (b) follow from a standard independence assumption among the measurements. Furthermore, the Markovian property is applied in (b) (e.g., an image measurement at time i only depends on the state at time i).

As the measurements $\mathcal{Z}_k$ are known, we are free to eliminate them as variables and consider them as parameters of the joint probability factors over the actual unknowns. This naturally leads to the well known factor graph representation, a class of bipartite graphical models that can be used to represent such factored densities [51, 52]. A schematic representation of the connectivity of the factor graph underlying the VIO problem is given in Fig. 3 (the connectivity of the structureless vision factors will be clarified in Section VII). The factor graph is composed of nodes for unknowns and nodes for the probability factors defined on them, and the graph structure expresses which unknowns are involved in each factor.
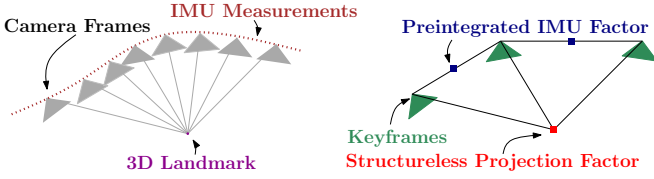
Fig. 3: Left: visual and inertial measurements in VIO. Right: factor graph in which several IMU measurements are summarized in a single preintegrated IMU factor and a structureless vision factor constraints keyframes observing the same landmark.

The MAP estimate $\mathcal{X}_k^\star$ corresponds to the maximum of (25), or equivalently, the minimum of the negative log-posterior. Under the assumption of zero-mean Gaussian noise, the negative log-posterior can be written as a sum of squared residual errors:

$$\mathcal{X}_k^\star \doteq \arg\min_{\mathcal{X}_k} -\log_e \ p(\mathcal{X}_k|\mathcal{Z}_k) \tag{26}$$

$$= \arg\min_{\mathcal{X}_k} \|\mathbf{r}_0\|_{\boldsymbol{\Sigma}_0}^2 + \sum_{(i,j)\in\mathcal{K}_k} \|\mathbf{r}_{\mathcal{I}_{ij}}\|_{\boldsymbol{\Sigma}_{ij}}^2 + \sum_{i\in\mathcal{K}_k}\sum_{l\in\mathcal{C}_i} \|\mathbf{r}_{\mathcal{C}_{il}}\|_{\boldsymbol{\Sigma}_\mathcal{C}}^2$$

where $\mathbf{r}_0$, $\mathbf{r}_{\mathcal{I}_{ij}}$, $\mathbf{r}_{\mathcal{C}_{il}}$ are the residual errors associated to the measurements, and $\boldsymbol{\Sigma}_0$, $\boldsymbol{\Sigma}_{ij}$, and $\boldsymbol{\Sigma}_\mathcal{C}$ are the corresponding covariance matrices. Roughly speaking, the residual error is a function of $\mathcal{X}_k$ that quantifies the mismatch between a measured quantity and the predicted value of this quantity given the state $\mathcal{X}_k$ and the priors. The goal of the following sections is to provide expressions for the residual errors and the covariances.

V. IMU MODEL AND MOTION INTEGRATION

An IMU commonly includes a 3-axis accelerometer and a 3-axis gyroscope and allows measuring the rotation rate and the acceleration of the sensor with respect to an inertial frame. The measurements, namely $_B\tilde{\mathbf{a}}(t)$, and $_B\tilde{\boldsymbol{\omega}}_{WB}(t)$, are affected by additive white noise $\boldsymbol{\eta}$ and a slowly varying sensor bias $\mathbf{b}$:

$$_B\tilde{\boldsymbol{\omega}}_{WB}(t) = {_B\boldsymbol{\omega}}_{WB}(t) + \mathbf{b}^g(t) + \boldsymbol{\eta}^g(t) \tag{27}$$

$$_B\tilde{\mathbf{a}}(t) = \mathtt{R}_{WB}^\mathsf{T}(t)\left(_W\mathbf{a}(t) - {_W\mathbf{g}}\right) + \mathbf{b}^a(t) + \boldsymbol{\eta}^a(t), \tag{28}$$

In our notation, the prefix B denotes that the corresponding quantity is expressed in the frame B (*c.f.,* Fig. 2). The pose of the IMU is described by the transformation $\{\mathtt{R}_{WB}, {_W\mathbf{p}}\}$, which maps a point from sensor frame B to W. The vector $_B\boldsymbol{\omega}_{WB}(t)\in\mathbb{R}^3$ is the instantaneous angular velocity of B relative to W expressed in coordinate frame B, while $_W\mathbf{a}(t)\in\mathbb{R}^3$ is the acceleration of the sensor; $_W\mathbf{g}$ is the gravity vector in world coordinates. We neglect effects due to earth's rotation, which amounts to assuming that W is an inertial frame.

The goal now is to infer the motion of the system from IMU measurements. For this purpose we introduce the following kinematic model [49, 53]:

$$\dot{\mathtt{R}}_{WB} = \mathtt{R}_{WB}\, {_B\boldsymbol{\omega}}_{WB}^\wedge, \qquad {_W\dot{\mathbf{v}}} = {_W\mathbf{a}}, \qquad {_W\dot{\mathbf{p}}} = {_W\mathbf{v}}, \tag{29}$$

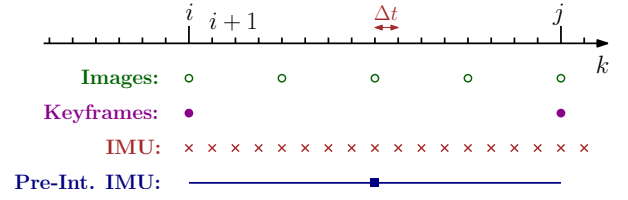which describes the evolution of the pose and velocity of B.

Fig. 4: Different rates for IMU and camera.

The state at time $t+\Delta t$ is obtained by integrating Eq. (29):

$$\mathtt{R}_{WB}(t+\Delta t) = \mathtt{R}_{WB}(t)\,\mathrm{Exp}\left(\int_t^{t+\Delta t} {_B\boldsymbol{\omega}}_{WB}(\tau)d\tau\right)$$

$$_W\mathbf{v}(t+\Delta t) = {_W\mathbf{v}}(t) + \int_t^{t+\Delta t} {_W\mathbf{a}}(\tau)d\tau$$

$$_W\mathbf{p}(t+\Delta t) = {_W\mathbf{p}}(t) + \int_t^{t+\Delta t} {_W\mathbf{v}}(\tau)d\tau + \iint_t^{t+\Delta t} {_W\mathbf{a}}(\tau)d\tau^2.$$

Assuming that $_W\mathbf{a}$ and $_B\boldsymbol{\omega}_{WB}$ remain constant in the time interval $[t, t+\Delta t]$, we can write:

$$\mathtt{R}_{WB}(t+\Delta t) = \mathtt{R}_{WB}(t)\,\mathrm{Exp}\left(_B\boldsymbol{\omega}_{WB}(t)\Delta t\right)$$

$$_W\mathbf{v}(t+\Delta t) = {_W\mathbf{v}}(t) + {_W\mathbf{a}}(t)\Delta t$$

$$_W\mathbf{p}(t+\Delta t) = {_W\mathbf{p}}(t) + {_W\mathbf{v}}(t)\Delta t + \frac{1}{2}{_W\mathbf{a}}(t)\Delta t^2. \tag{30}$$

Using Eqs. (27)–(28), we can write $_W\mathbf{a}$ and $_B\boldsymbol{\omega}_{WB}$ as a function of the IMU measurements, hence (30) becomes

$$\mathtt{R}(t+\Delta t) = \mathtt{R}(t)\,\mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}(t) - \mathbf{b}^g(t) - \boldsymbol{\eta}^{gd}(t)\right)\Delta t\right)$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \mathbf{g}\Delta t + \mathtt{R}(t)\left(\tilde{\mathbf{a}}(t) - \mathbf{b}^a(t) - \boldsymbol{\eta}^{ad}(t)\right)\Delta t$$

$$\mathbf{p}(t+\Delta t) = \mathbf{p}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{g}\Delta t^2$$
$$+ \frac{1}{2}\mathtt{R}(t)\left(\tilde{\mathbf{a}}(t) - \mathbf{b}^a(t) - \boldsymbol{\eta}^{ad}(t)\right)\Delta t^2, \tag{31}$$

where we dropped the coordinate frame subscripts for readability (the notation should be unambiguous from now on). This numeric integration of the velocity and position assumes a constant orientation $\mathtt{R}(t)$ for the time of integration between two measurements, which is not an exact solution of the differential equation (29) for measurements with non-zero rotation rate. In practice, the use of a high-rate IMU mitigates the effects of this approximation. We adopt the integration scheme (31) as it is simple and amenable for modeling and uncertainty propagation. While we show that this integration scheme performs very well in practice, we remark that for slower IMU measurement rates one may consider using higher-order numerical integration methods [54–57].

The covariance of the discrete-time noise $\boldsymbol{\eta}^{gd}$ is a function of the sampling rate and relates to the continuous-time spectral noise $\boldsymbol{\eta}^g$ via $\mathrm{Cov}(\boldsymbol{\eta}^{gd}(t)) = \frac{1}{\Delta t}\mathrm{Cov}(\boldsymbol{\eta}^g(t))$. The same relation holds for $\boldsymbol{\eta}^{ad}$ (*cf.*, [58, Appendix]).

VI. IMU PREINTEGRATION ON MANIFOLD

While Eq. (31) could be readily seen as a probabilistic constraint in a factor graph, it would require to include states in the factor graph at high rate. Intuitively, Eq. (31) relates states at time t and $t+\Delta t$, where Δt is the sampling period

of the IMU, hence we would have to add new states in the estimation at every new IMU measurement [37].

Here we show that all measurements between two keyframes at times $k = i$ and $k = j$ (see Fig. 4) can be summarized in a single compound measurement, named *preintegrated IMU measurement*, which constrains the motion between consecutive keyframes. This concept was first proposed in [2] using Euler angles and we extend it, by developing a suitable theory for preintegration on the manifold SO(3).

We assume that the IMU is synchronized with the camera and provides measurements at discrete times k (*cf.*, Fig. 4).[1] Iterating the IMU integration (31) for all Δt intervals between two consecutive keyframes at times $k = i$ and $k = j$ (*c.f.*, Fig. 4), we find:

$$\mathtt{R}_j = \mathtt{R}_i \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_k^g - \boldsymbol{\eta}_k^{gd}\right)\Delta t\right),$$

$$\mathbf{v}_j = \mathbf{v}_i + \mathbf{g}\Delta t_{ij} + \sum_{k=i}^{j-1} \mathtt{R}_k \left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t \qquad (32)$$

$$\mathbf{p}_j = \mathbf{p}_i + \sum_{k=i}^{j-1}\left[\mathbf{v}_k\Delta t + \frac{1}{2}\mathbf{g}\Delta t^2 + \frac{1}{2}\mathtt{R}_k\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t^2\right]$$

where we introduced the shorthands $\Delta t_{ij} \doteq \sum_{k=i}^{j-1}\Delta t$ and $(\cdot)_i \doteq (\cdot)(t_i)$ for readability. While Eq. (32) already provides an estimate of the motion between time t_i and t_j, it has the drawback that the integration in (32) has to be repeated whenever the linearization point at time t_i changes [24] (intuitively, a change in the rotation $\mathtt{R}_i$ implies a change in all future rotations $\mathtt{R}_k$, $k = i, \dots, j-1$, and makes necessary to re-evaluate summations and products in (32)).

We want to avoid to recompute the above integration whenever the linearization point at time t_i changes. Therefore, we follow [2] and *define* the following relative motion increments that are independent of the pose and velocity at t_i:

$$\Delta\mathtt{R}_{ij} \doteq \mathtt{R}_i^\mathsf{T}\mathtt{R}_j = \prod_{k=i}^{j-1}\mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_k^g - \boldsymbol{\eta}_k^{gd}\right)\Delta t\right)$$

$$\Delta\mathbf{v}_{ij} \doteq \mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) = \sum_{k=i}^{j-1}\Delta\mathtt{R}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t$$

$$\Delta\mathbf{p}_{ij} \doteq \mathtt{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\sum_{k=i}^{j-1}\mathbf{g}\Delta t^2\right)$$
$$= \sum_{k=i}^{j-1}\left[\Delta\mathbf{v}_{ik}\Delta t + \frac{1}{2}\Delta\mathtt{R}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t^2\right] \quad (33)$$

where $\Delta\mathtt{R}_{ik} \doteq \mathtt{R}_i^\mathsf{T}\mathtt{R}_k$ and $\Delta\mathbf{v}_{ik} \doteq \mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_k - \mathbf{v}_i - \mathbf{g}\Delta t_{ik}\right)$. We highlight that, in contrast to the "delta" rotation $\Delta\mathtt{R}_{ij}$, neither $\Delta\mathbf{v}_{ij}$ nor $\Delta\mathbf{p}_{ij}$ correspond to the true *physical* change in velocity and position but are defined in a way that make the right-hand side of (33) independent from the state at time i as well as gravitational effects. Indeed, we will be able to compute the right-hand side of (33) directly from the inertial measurements between the two keyframes.

---

[1] We calibrate the IMU-camera delay using the *Kalibr* toolbox [59]. An alternative is to add the delay as a state in the estimation process [60].

Unfortunately, summations and products in (33) are still function of the bias estimate. We tackle this problem in two steps. In Section VI-A, we assume $\mathbf{b}_i$ is known; then, in Section VI-C we show how to avoid repeating the integration when the bias estimate changes.

In the rest of the paper, we assume that the bias remains constant between two keyframes:

$$\mathbf{b}_i^g = \mathbf{b}_{i+1}^g = \dots = \mathbf{b}_{j-1}^g, \quad \mathbf{b}_i^a = \mathbf{b}_{i+1}^a = \dots = \mathbf{b}_{j-1}^a. \quad (34)$$

A. Preintegrated IMU Measurements

Equation (33) relates the states of keyframes i and j (left-hand side) to the measurements (right-hand side). In this sense, it can be already understood as a measurement model. Unfortunately, it has a fairly intricate dependence on the measurement noise and this complicates a direct application of MAP estimation; intuitively, the MAP estimator requires to clearly define the densities (and their log-likelihood) of the measurements. In this section we manipulate (33) so to make easier the derivation of the measurement log-likelihood. In practice, we isolate the noise terms of the individual inertial measurements in (33). As discussed above, across this section assume that the bias at time t_i is known.

Let us start with the rotation increment $\Delta\mathtt{R}_{ij}$ in (33). We use the first-order approximation (7) (rotation noise is "small") and rearrange the terms, by "moving" the noise to the end, using the relation (11):

$$\Delta\mathtt{R}_{ij} \overset{\mathrm{eq.(7)}}{\simeq} \prod_{k=i}^{j-1}\left[\mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_i^g\right)\Delta t\right)\mathrm{Exp}\left(-\mathtt{J}_r^k\,\boldsymbol{\eta}_k^{gd}\,\Delta t\right)\right]$$

$$\overset{\mathrm{eq.(11)}}{=} \Delta\tilde{\mathtt{R}}_{ij}\prod_{k=i}^{j-1}\mathrm{Exp}\left(-\Delta\tilde{\mathtt{R}}_{k+1j}^\mathsf{T}\,\mathtt{J}_r^k\,\boldsymbol{\eta}_k^{gd}\,\Delta t\right)$$

$$\doteq \Delta\tilde{\mathtt{R}}_{ij}\mathrm{Exp}\left(-\delta\boldsymbol{\phi}_{ij}\right) \qquad (35)$$

with $\mathtt{J}_r^k \doteq \mathtt{J}_r^k((\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_i^g)\Delta t)$. In the last line of (35), we defined the *preintegrated rotation measurement* $\Delta\tilde{\mathtt{R}}_{ij} \doteq \prod_{k=i}^{j-1}\mathrm{Exp}\left((\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_i^g)\Delta t\right)$, and its noise $\delta\boldsymbol{\phi}_{ij}$, which will be further analysed in the next section.

Substituting (35) back into the expression of $\Delta\mathbf{v}_{ij}$ in (33), using the first-order approximation (4) for $\mathrm{Exp}\left(-\delta\boldsymbol{\phi}_{ij}\right)$, and dropping higher-order noise terms, we obtain:

$$\Delta\mathbf{v}_{ij} \overset{\mathrm{eq.(4)}}{\simeq} \sum_{k=i}^{j-1}\Delta\tilde{\mathtt{R}}_{ik}\left(\mathbf{I} - \delta\boldsymbol{\phi}_{ik}^\wedge\right)\left(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a\right)\Delta t - \Delta\tilde{\mathtt{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t$$

$$\overset{\mathrm{eq.(2)}}{=} \Delta\tilde{\mathbf{v}}_{ij} + \sum_{k=i}^{j-1}\left[\Delta\tilde{\mathtt{R}}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a\right)^\wedge\delta\boldsymbol{\phi}_{ik}\Delta t - \Delta\tilde{\mathtt{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t\right]$$

$$\doteq \Delta\tilde{\mathbf{v}}_{ij} - \delta\mathbf{v}_{ij} \qquad (36)$$

where we defined the *preintegrated velocity measurement* $\Delta\tilde{\mathbf{v}}_{ij} \doteq \sum_{k=i}^{j-1}\Delta\tilde{\mathtt{R}}_{ik}(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a)\Delta t$ and its noise $\delta\mathbf{v}_{ij}$.

Similarly, substituting (35) and (36) in the expression of $\Delta\mathbf{p}_{ij}$ in (33), and using the first-order approximation (4), we

obtain:

$$\Delta \mathbf{p}_{ij} \overset{\text{eq.(4)}}{\simeq} \sum_{k=i}^{j-1} \Big[(\Delta \tilde{\mathbf{v}}_{ik} - \delta \mathbf{v}_{ik}) \Delta t + \frac{1}{2} \Delta \tilde{\mathtt{R}}_{ik} (\mathbf{I} - \delta \boldsymbol{\phi}_{ik}^{\wedge}) (\tilde{\mathbf{a}}_k - \mathbf{b}_i^a) \Delta t^2$$
$$- \frac{1}{2} \Delta \tilde{\mathtt{R}}_{ik} \boldsymbol{\eta}_k^{ad} \Delta t^2 \Big]$$
$$\overset{\text{eq.(2)}}{=} \Delta \tilde{\mathbf{p}}_{ij} + \sum_{k=i}^{j-1} \Big[- \delta \mathbf{v}_{ik} \Delta t + \frac{1}{2} \Delta \tilde{\mathtt{R}}_{ik} (\tilde{\mathbf{a}}_k - \mathbf{b}_i^a)^{\wedge} \delta \boldsymbol{\phi}_{ik} \Delta t^2$$
$$- \frac{1}{2} \Delta \tilde{\mathtt{R}}_{ik} \boldsymbol{\eta}_k^{ad} \Delta t^2 \Big]$$
$$\doteq \Delta \tilde{\mathbf{p}}_{ij} - \delta \mathbf{p}_{ij}, \tag{37}$$

where we defined the *preintegrated position measurement* $\Delta \tilde{\mathbf{p}}_{ij}$ and its noise $\delta \mathbf{p}_{ij}$.

Substituting the expressions (35), (36), (37) back in the original definition of $\Delta \mathtt{R}_{ij}, \Delta \mathbf{v}_{ij}, \Delta \mathbf{p}_{ij}$ in (33), we finally get our *preintegrated measurement model* (remember $\mathrm{Exp} \left(-\delta \boldsymbol{\phi}_{ij} \right)^{\mathsf{T}} = \mathrm{Exp} \left(\delta \boldsymbol{\phi}_{ij} \right)$):

$$\Delta \tilde{\mathtt{R}}_{ij} = \mathtt{R}_i^{\mathsf{T}} \mathtt{R}_j \mathrm{Exp} \left(\delta \boldsymbol{\phi}_{ij} \right)$$
$$\Delta \tilde{\mathbf{v}}_{ij} = \mathtt{R}_i^{\mathsf{T}} \left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g} \Delta t_{ij} \right) + \delta \mathbf{v}_{ij}$$
$$\Delta \tilde{\mathbf{p}}_{ij} = \mathtt{R}_i^{\mathsf{T}} \left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i \Delta t_{ij} - \frac{1}{2} \mathbf{g} \Delta t_{ij}^2 \right) + \delta \mathbf{p}_{ij} \tag{38}$$

where our compound measurements are written as a function of the (to-be-estimated) state "plus" a random noise, described by the random vector $[\delta \boldsymbol{\phi}_{ij}^{\mathsf{T}}, \delta \mathbf{v}_{ij}^{\mathsf{T}}, \delta \mathbf{p}_{ij}^{\mathsf{T}}]^{\mathsf{T}}$.

To wrap-up the discussion in this section, we manipulated the measurement model (33) and rewrote it as (38). The advantage of Eq. (38) is that, for a suitable distribution of the noise, it makes the definition of the log-likelihood straightforward. For instance the (negative) log-likelihood of measurements with zero-mean additive Gaussian noise (last two lines in (38)) is a quadratic function. Similarly, if $\delta \boldsymbol{\phi}_{ij}$ is a zero-mean Gaussian noise, we compute the (negative) log-likelihood associated with $\Delta \tilde{\mathtt{R}}_{ij}$. The nature of the noise terms is discussed in the following section.

B. Noise Propagation

In this section we derive the statistics of the noise vector $[\delta \boldsymbol{\phi}_{ij}^{\mathsf{T}}, \delta \mathbf{v}_{ij}^{\mathsf{T}}, \delta \mathbf{p}_{ij}^{\mathsf{T}}]^{\mathsf{T}}$. While we already observed that it is convenient to approximate the noise vector to be zero-mean Normally distributed, it is of paramount importance to accurately model the noise covariance. Indeed, the noise covariance has a strong influence on the MAP estimator (the inverse noise covariance is used to weight the terms in the optimization (26)). In this section, we therefore provide a derivation of the covariance $\mathbf{\Sigma}_{ij}$ of the preintegrated measurements:

$$\boldsymbol{\eta}_{ij}^{\Delta} \doteq [\delta \boldsymbol{\phi}_{ij}^{\mathsf{T}}, \delta \mathbf{v}_{ij}^{\mathsf{T}}, \delta \mathbf{p}_{ij}^{\mathsf{T}}]^{\mathsf{T}} \sim \mathcal{N}(\mathbf{0}_{9 \times 1}, \mathbf{\Sigma}_{ij}). \tag{39}$$

We first consider the preintegrated rotation noise $\delta \boldsymbol{\phi}_{ij}$. Recall from (35) that

$$\mathrm{Exp} \left(-\delta \boldsymbol{\phi}_{ij} \right) \doteq \prod_{k=i}^{j-1} \mathrm{Exp} \left(-\Delta \tilde{\mathtt{R}}_{k+1j}^{\mathsf{T}} \mathtt{J}_r^k \boldsymbol{\eta}_k^{gd} \Delta t \right). \tag{40}$$

Taking the Log on both sides and changing signs, we get:

$$\delta \boldsymbol{\phi}_{ij} = -\mathrm{Log} \left(\prod_{k=i}^{j-1} \mathrm{Exp} \left(-\Delta \tilde{\mathtt{R}}_{k+1j}^{\mathsf{T}} \mathtt{J}_r^k \boldsymbol{\eta}_k^{gd} \Delta t \right) \right). \tag{41}$$

Repeated application of the first-order approximation (9) (recall that $\boldsymbol{\eta}_k^{gd}$ as well as $\delta \boldsymbol{\phi}_{ij}$ are small rotation noises, hence the right Jacobians are close to the identity) produces:

$$\delta \boldsymbol{\phi}_{ij} \simeq \sum_{k=i}^{j-1} \Delta \tilde{\mathtt{R}}_{k+1j}^{\mathsf{T}} \mathtt{J}_r^k \boldsymbol{\eta}_k^{gd} \Delta t \tag{42}$$

Up to first order, the noise $\delta \boldsymbol{\phi}_{ij}$ is zero-mean and Gaussian, as it is a linear combination of zero-mean noise terms $\boldsymbol{\eta}_k^{gd}$. This is desirable, since it brings the rotation measurement model (38) exactly in the form (12).

Dealing with the noise terms $\delta \mathbf{v}_{ij}$ and $\delta \mathbf{p}_{ij}$ is now easy: these are linear combinations of the acceleration noise $\boldsymbol{\eta}_k^{ad}$ and the preintegrated rotation noise $\delta \boldsymbol{\phi}_{ij}$, hence they are also zero-mean and Gaussian. Simple manipulation leads to:

$$\delta \mathbf{v}_{ij} \simeq \sum_{k=i}^{j-1} \Big[-\Delta \tilde{\mathtt{R}}_{ik} (\tilde{\mathbf{a}}_k - \mathbf{b}_i^a)^{\wedge} \delta \boldsymbol{\phi}_{ik} \Delta t + \Delta \tilde{\mathtt{R}}_{ik} \boldsymbol{\eta}_k^{ad} \Delta t \Big] \tag{43}$$

$$\delta \mathbf{p}_{ij} \simeq \sum_{k=i}^{j-1} \Big[\delta \mathbf{v}_{ik} \Delta t - \frac{1}{2} \Delta \tilde{\mathtt{R}}_{ik} (\tilde{\mathbf{a}}_k - \mathbf{b}_i^a)^{\wedge} \delta \boldsymbol{\phi}_{ik} \Delta t^2 + \frac{1}{2} \Delta \tilde{\mathtt{R}}_{ik} \boldsymbol{\eta}_k^{ad} \Delta t^2 \Big]$$

where the relations are valid up to the first order.

Eqs. (42)-(43) express the preintegrated noise $\boldsymbol{\eta}_{ij}^{\Delta}$ as a linear function of the IMU measurement noise $\boldsymbol{\eta}_k^d \doteq [\boldsymbol{\eta}_k^{gd}, \boldsymbol{\eta}_k^{ad}]$, $k = 1, \ldots, j-1$. Therefore, from the knowledge of the covariance of $\boldsymbol{\eta}_k^d$ (given in the IMU specifications), we can compute the covariance of $\boldsymbol{\eta}_{ij}^{\Delta}$, namely $\mathbf{\Sigma}_{ij}$, by a simple linear propagation.

In Appendix IX-A, we provide a more clever way to compute $\mathbf{\Sigma}_{ij}$. In particular, we show that $\mathbf{\Sigma}_{ij}$ can be conveniently computed in iterative form: as a new IMU measurement arrive we only update $\mathbf{\Sigma}_{ij}$, rather than recomputing it from scratch. The iterative computation leads to simpler expressions and is more amenable for online inference.

C. Incorporating Bias Updates

In the previous section, we assumed that the bias $\{\bar{\mathbf{b}}_i^a, \bar{\mathbf{b}}_i^g\}$ that is used during preintegration between $k = i$ and $k = j$ is correct and does not change. However, more likely, the bias estimate changes by a small amount $\delta \mathbf{b}$ during optimization. One solution would be to recompute the delta measurements when the bias changes; however, that is computationally expensive. Instead, given a bias update $\mathbf{b} \leftarrow \bar{\mathbf{b}} + \delta \mathbf{b}$, we can update the delta measurements using a first-order expansion:

$$\Delta \tilde{\mathtt{R}}_{ij}(\mathbf{b}_i^g) \simeq \Delta \tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g) \, \mathrm{Exp} \Big(\frac{\partial \Delta \bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g} \delta \mathbf{b}^g \Big) \tag{44}$$
$$\Delta \tilde{\mathbf{v}}_{ij}(\mathbf{b}_i^g, \mathbf{b}_i^a) \simeq \Delta \tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i^g, \bar{\mathbf{b}}_i^a) + \frac{\partial \Delta \bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}^g} \delta \mathbf{b}_i^g + \frac{\partial \Delta \bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}^a} \delta \mathbf{b}_i^a$$
$$\Delta \tilde{\mathbf{p}}_{ij}(\mathbf{b}_i^g, \mathbf{b}_i^a) \simeq \Delta \tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i^g, \bar{\mathbf{b}}_i^a) + \frac{\partial \Delta \bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}^g} \delta \mathbf{b}_i^g + \frac{\partial \Delta \bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}^a} \delta \mathbf{b}_i^a$$

This is similar to the bias correction in [2] but operates directly on SO(3). The Jacobians $\{ \frac{\partial \Delta \bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}, \frac{\partial \Delta \bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}^g}, \ldots \}$ (computed at $\bar{\mathbf{b}}_i$, the bias estimate at integration time) describe how the measurements change due to a change in the bias estimate. The Jacobians remain constant and can be precomputed during the preintegration. The derivation of the Jacobians is very similar to the one we used in Section VI-A to express the measurements as a large value *plus* a small perturbation and is given in Appendix IX-B.

D. Preintegrated IMU Factors

Given the preintegrated measurement model in (38) and since measurement noise is zero-mean and Gaussian (with covariance $\boldsymbol{\Sigma}_{ij}$) up to first order (39), it is now easy to write the residual errors $\mathbf{r}_{\mathcal{I}_{ij}} \doteq [\mathbf{r}_{\Delta\mathsf{R}_{ij}}^{\mathsf{T}}, \mathbf{r}_{\Delta\mathbf{v}_{ij}}^{\mathsf{T}}, \mathbf{r}_{\Delta\mathbf{p}_{ij}}^{\mathsf{T}}]^{\mathsf{T}} \in \mathbb{R}^9$, where

$$
\begin{aligned}
\mathbf{r}_{\Delta\mathsf{R}_{ij}} &\doteq \mathrm{Log}\left(\left(\Delta\tilde{\mathsf{R}}_{ij}(\bar{\mathbf{b}}_i^g)\mathrm{Exp}\left(\tfrac{\partial\Delta\bar{\mathsf{R}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}^g\right)\right)^{\mathsf{T}}\mathsf{R}_i^{\mathsf{T}}\mathsf{R}_j\right) \\
\mathbf{r}_{\Delta\mathbf{v}_{ij}} &\doteq \mathsf{R}_i^{\mathsf{T}}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) \\
&\quad - \left[\Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i^g,\bar{\mathbf{b}}_i^a) + \tfrac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}^g + \tfrac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^a}\delta\mathbf{b}^a\right] \\
\mathbf{r}_{\Delta\mathbf{p}_{ij}} &\doteq \mathsf{R}_i^{\mathsf{T}}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \tfrac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) \\
&\quad - \left[\Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i^g,\bar{\mathbf{b}}_i^a) + \tfrac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}^g + \tfrac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}_a}\delta\mathbf{b}^a\right], \quad (45)
\end{aligned}
$$

in which we also included the bias updates of Eq. (44).

According to the "lift-solve-retract" method (Section III-C), at each Gauss-Newton iteration we need to re-parametrize (45) using the retraction (21). Then, the "solve" step requires to linearize the resulting cost around the current estimate. For the purpose of linearization, it is convenient to compute analytic expressions of the Jacobians of the residual errors, which we derive in the Appendix IX-C.

E. Bias Model

When presenting the IMU model (27), we said that biases are slowly time-varying quantities. Hence, we model them with a "Brownian motion", *i.e.*, integrated white noise:

$$
\dot{\mathbf{b}}^g(t) = \boldsymbol{\eta}^{bg}, \qquad \dot{\mathbf{b}}^a(t) = \boldsymbol{\eta}^{ba}. \tag{46}
$$

Integrating (46) over the time interval $[t_i, t_j]$ between two consecutive keyframes i and j we get:

$$
\mathbf{b}_j^g = \mathbf{b}_i^g + \boldsymbol{\eta}^{bgd}, \qquad \mathbf{b}_j^a = \mathbf{b}_i^a + \boldsymbol{\eta}^{bad}, \tag{47}
$$

where, as done before, we use the shorthand $\mathbf{b}_i^g \doteq \mathbf{b}^g(t_i)$, and we define the discrete noises $\boldsymbol{\eta}^{bgd}$ and $\boldsymbol{\eta}^{bad}$, which have zero mean and covariance $\boldsymbol{\Sigma}^{bgd} \doteq \Delta t_{ij}\mathrm{Cov}(\boldsymbol{\eta}^{bg})$ and $\boldsymbol{\Sigma}^{bad} \doteq \Delta t_{ij}\mathrm{Cov}(\boldsymbol{\eta}^{ba})$, respectively (*cf.* [58, Appendix]).

The model (47) can be readily included in our factor graph, as a further additive term in (26) for all consecutive keyframes:

$$
\|\mathbf{r}_{\mathbf{b}_{ij}}\|^2 \doteq \|\mathbf{b}_j^g - \mathbf{b}_i^g\|_{\boldsymbol{\Sigma}^{bgd}}^2 + \|\mathbf{b}_j^a - \mathbf{b}_i^a\|_{\boldsymbol{\Sigma}^{bad}}^2 \tag{48}
$$

VII. STRUCTURELESS VISION FACTORS

In this section we introduce our structureless model for vision measurements. The key feature of our approach is the linear elimination of landmarks. Note that the elimination is repeated at each Gauss-Newton iteration, hence we are still guaranteed to obtain the optimal MAP estimate.

Visual measurements contribute to the cost (26) via the sum:

$$
\sum_{i\in\mathcal{K}_k}\sum_{l\in\mathcal{C}_i}\|\mathbf{r}_{\mathcal{C}_{il}}\|_{\boldsymbol{\Sigma}_{\mathcal{C}}}^2 = \sum_{l=1}^{L}\sum_{i\in\mathcal{X}(l)}\|\mathbf{r}_{\mathcal{C}_{il}}\|_{\boldsymbol{\Sigma}_{\mathcal{C}}}^2 \tag{49}
$$

which, on the right-hand-side, we rewrote as a sum of contributions of each landmark $l = 1, \ldots, L$. In (49), $\mathcal{X}(l)$ denotes the subset of keyframes in which l is seen.

A fairly standard model for the residual error of a single image measurement $\mathbf{z}_{il}$ is the reprojection error:

$$
\mathbf{r}_{\mathcal{C}_{il}} = \mathbf{z}_{il} - \pi(\mathsf{R}_i, \mathbf{p}_i, \rho_l), \tag{50}
$$

where $\rho_l \in \mathbb{R}^3$ denotes the position of the l-th landmark, and $\pi(\cdot)$ is a standard perspective projection, which also encodes the (known) IMU-camera transformation T_{BC}.

Direct use of (50) would require to include the landmark positions ρ_l, $l = 1, \ldots, L$ in the optimization, and this impacts negatively on computation. Therefore, in the following we adopt a *structureless* approach that avoids optimization over the landmarks, thus ensuring to retrieve the MAP estimate.

As recalled in Section III-C, at each GN iteration, we *lift* the cost function, using the retraction (21). For the vision factors this means that the original residuals (49) become:

$$
\sum_{l=1}^{L}\sum_{i\in\mathcal{X}(l)}\|\mathbf{z}_{il} - \check{\pi}(\delta\boldsymbol{\phi}_i, \delta\mathbf{p}_i, \delta\rho_l)\|_{\boldsymbol{\Sigma}_{\mathcal{C}}}^2 \tag{51}
$$

where $\delta\boldsymbol{\phi}_i, \delta\mathbf{p}_i, \delta\rho_l$ are now Euclidean corrections, and $\check{\pi}(\cdot)$ is the lifted cost function. The "solve" step in the GN method is based on linearization of the residuals:

$$
\sum_{l=1}^{L}\sum_{i\in\mathcal{X}(l)}\|\mathbf{F}_{il}\delta\mathbf{T}_i + \mathbf{E}_{il}\delta\rho_l - \mathbf{b}_{il}\|^2, \tag{52}
$$

where $\delta\mathbf{T}_i \doteq [\delta\boldsymbol{\phi}_i\ \delta\mathbf{p}_i]^{\mathsf{T}}$; the Jacobians $\mathbf{F}_{il}, \mathbf{E}_{il}$, and the vector $\mathbf{b}_{il}$ (both normalized by $\boldsymbol{\Sigma}_{\mathcal{C}}^{1/2}$) result from the linearization. The vector $\mathbf{b}_{il}$ is the residual error at the linearization point.

Writing the second sum in (52) in matrix form we get:

$$
\sum_{l=1}^{L}\|\mathbf{F}_l\ \delta\mathbf{T}_{\mathcal{X}(l)} + \mathbf{E}_l\ \delta\rho_l - \mathbf{b}_l\|^2 \tag{53}
$$

where $\mathbf{F}_l, \mathbf{E}_l, \mathbf{b}_l$ are obtained by stacking $\mathbf{F}_{il}, \mathbf{E}_{il}, \mathbf{b}_{il}$, respectively, for all $i \in \mathcal{X}(l)$.

Since a landmark l appears in a single term of the sum (53), for any given choice of the pose perturbation $\delta\mathbf{T}_{\mathcal{X}(l)}$, the landmark perturbation $\delta\rho_l$ that minimizes the quadratic cost $\|\mathbf{F}_l\ \delta\mathbf{T}_{\mathcal{X}(l)} + \mathbf{E}_l\ \delta\rho_l - \mathbf{b}_l\|^2$ is:

$$
\delta\rho_l = -(\mathbf{E}_l^{\mathsf{T}}\mathbf{E}_l)^{-1}\mathbf{E}_l^{\mathsf{T}}(\mathbf{F}_l\ \delta\mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l) \tag{54}
$$

Substituting (54) back into (53) we can *eliminate* the variable $\delta\rho_l$ from the optimization problem:

$$
\sum_{l=1}^{L}\|(\mathbf{I} - \mathbf{E}_l(\mathbf{E}_l^{\mathsf{T}}\mathbf{E}_l)^{-1}\mathbf{E}_l^{\mathsf{T}})\left(\mathbf{F}_l\ \delta\mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l\right)\|^2, \tag{55}
$$

where $\mathbf{I} - \mathbf{E}_l(\mathbf{E}_l^{\mathsf{T}}\mathbf{E}_l)^{-1}\mathbf{E}_l^{\mathsf{T}}$ is an orthogonal projector of $\mathbf{E}_l$. In Appendix IX-D we show that the cost (55) can be further manipulated, leading to a more efficient implementation.

This approach is well known in the bundle adjustment literature as the *Schur complement trick*, where a standard practice is to update the linearization point of ρ_l via *back-substitution* [61]. In contrast, we obtain the updated landmark positions from the linearization point of the poses using a fast linear triangulation. Using this approach, we reduced a large set of factors (51) which involve poses and landmarks into a smaller set of L factors (55), which only involve poses. In particular, the factor corresponding to landmark l only involves the states $\mathcal{X}(l)$ observing l, creating the connectivity pattern of Fig. 3. The same approach is also used in MSC-KF [5] to avoid the inclusion of landmarks in the state vector.

Fig. 6: Left: CPU time required for inference, comparing batch estimation against iSAM2. Right: histogram plot of CPU time for the proposed approach.
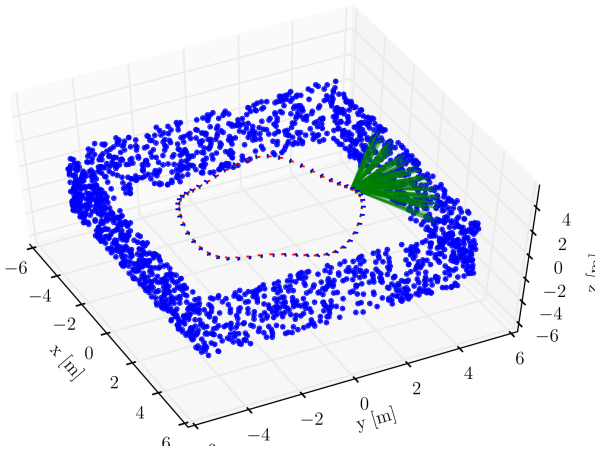

Fig. 5: Simulation setup: The camera moves along a circular trajectory while observing features (green lines) on the walls of a square environment.

However, since MSC-KF can only linearize and absorb a measurement once, the processing of measurements needs to be delayed until all measurements of the same landmark are observed. This does not apply to the proposed optimization-based approach, which allows for multiple relinearizations and the incremental inclusion of new measurements.

VIII. EXPERIMENTAL ANALYSIS

We tested the proposed approach on both simulated and real data. Section VIII-A reports simulation results, showing that our approach is accurate, fast, and consistent. Section VIII-B compares our approach against the state-of-the-art, confirming its superior accuracy in real indoor and outdoor experiments.

A. Simulation Experiments

We simulated a camera following a circular trajectory of three meter radius with a sinusoidal vertical motion. The total length of the trajectory is 120 meters. While moving, the camera observes landmarks as depicted in Fig. 5. The number of landmark observations per frame is limited to 50. To simulate a realistic feature-tracker, we corrupt the landmark measurements with isotropic Gaussian noise with standard deviation $\sigma_{\mathrm{px}} = 1$ pixel. The camera has a focal length of 315 pixels and runs at a rate of 2.5 Hz (simulating keyframes). The simulated acceleration and gyroscope measurements are computed from the analytic derivatives of the parametric trajectory and additionally corrupted by white noise and a slowly time-varying bias terms, according to the IMU model in Eq. (27).[2] To evaluate our approach, we performed a Monte Carlo analysis with 50 simulation runs, each with different realizations of process and measurement noise. In each run we compute the MAP estimate using the IMU and the vision models presented in this paper. The optimization (whose solution is the MAP estimate) is solved using the incremental smoothing algorithm iSAM2 [3]. iSAM2 uses the Bayes tree [62] data structure to obtain efficient variable
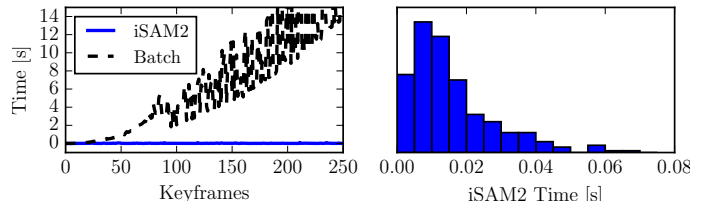
ordering that minimizes fill-in in the square-root information matrix and, thus, minimizes computation time. Further, iSAM2 exploits the fact that new measurements often have only local effect on the MAP estimate, hence applies incremental updates directly to the square-root information matrix, only re-solving for the variables affected by a new measurement.

In the following we present the results of our experiments, organized in four subsections: 1) pose estimation accuracy and timing, 2) consistency, 3) bias estimation accuracy, and 4) first-order bias correction. Then, in Section VIII-A5 we compare our approach against the original proposal of [2].

1) Pose Estimation Accuracy and Timing: The optimal MAP estimate is given by the *batch* nonlinear optimization of the least-squares objective in Eq. (26). However, as shown on the left in Fig. 6, the computational cost of batch optimization quickly increases as the trajectory length grows. A key ingredient that makes our approach extremely efficient is the use of the incremental smoothing algorithm iSAM2 [3], which performs close-to-optimal inference, while preserving real-time capability. Fig. 7 shows that the accuracy of iSAM2 is practically the same as the batch estimate. In odometry problems, the iSAM2 algorithm results in approximately constant update time per frame (Fig. 6, left), which in our experiment is approximately 10 milliseconds per update (Fig. 6, right).

2) Consistency: For generic motion, the VIO problem has four unobservable degrees of freedom, three corresponding to the global translation and one to the global orientation around the gravity direction (yaw), see [16]. A VIO algorithm must preserve these observability properties and avoid inclusion of spurious information along the unobservable directions, which would result in inconsistency [16]. Fig. 8 reports orientation and position errors with the corresponding 3σ bounds, confirming that our approach is consistent. In the VIO problem, the gravity direction is observable, hence the uncertainty on roll and pitch remains bounded. In contrast, global yaw and position cannot be measured and the uncertainty slowly grows over time.

To present more substantial evidence of the fact that our estimator is consistent, we recall a standard measure of consistency, the *average* Normalized Estimation Error Squared (NEES) [63]. The NEES is the squared estimation error ϵ_k normalized by the estimator-calculated covariance $\boldsymbol{\Sigma}_k$:

$$\eta_k \doteq \boldsymbol{\epsilon}_k^{\mathsf{T}} \hat{\boldsymbol{\Sigma}}_k^{-1} \boldsymbol{\epsilon}_k \qquad \text{(NEES)} \qquad (56)$$

The error in estimating the current pose is computed as:

$$\boldsymbol{\epsilon}_k \doteq \left[\mathrm{Log}(\hat{\mathtt{R}}_k^{\mathsf{T}} \mathtt{R}_k^{\mathrm{gt}}), \ \hat{\mathtt{R}}_k^{\mathsf{T}}(\hat{\mathbf{p}}_k - \mathbf{p}_k^{\mathrm{gt}}) \right]^{\mathsf{T}} \qquad (57)$$

where the exponent "gt" denotes ground-truth states and $(\hat{\mathtt{R}}_k, \hat{\mathbf{p}}_k)$ denotes the estimated pose at time k. Note that the

---

[2]We used the following IMU parameters: Gyroscope and accelerometer continuous-time noise density: $\sigma^g = 0.0007$ [rad/(s$\sqrt{\mathrm{Hz}}$)], $\sigma^a = 0.019$ [m/(s$^2\sqrt{\mathrm{Hz}}$)]. Gyroscope and accelerometer *bias* continous-time noise density: $\sigma^{bg} = 0.0004$ [rad/(s$^2\sqrt{\mathrm{Hz}}$)], $\sigma^{ba} = 0.012$ [m/(s$^3\sqrt{\mathrm{Hz}}$)].
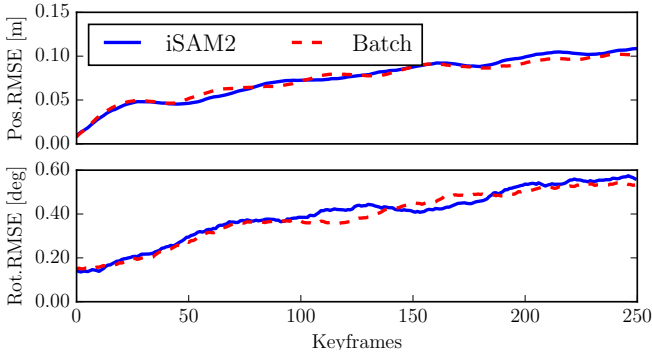
Fig. 7: Root Mean Squared Error (RMSE) averaged over 50 Monte Carlo experiments, comparing batch nonlinear optimization and iSAM2.
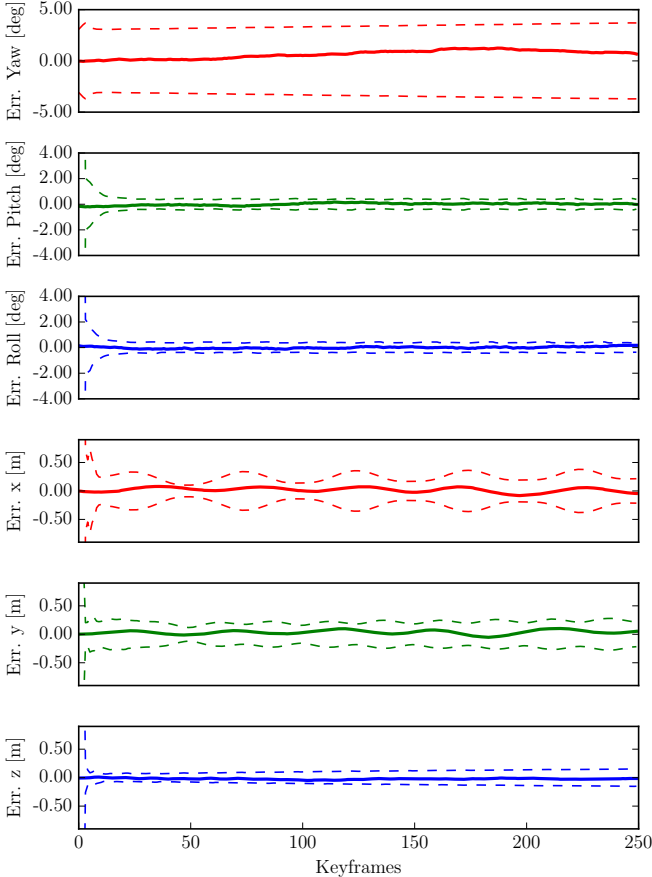

Fig. 8: Orientation and position errors with 3σ bounds (single simulation).

error (57) is expressed in the body frame and it is consistent with our choice of the retraction in Eq. (21) (intuitively, the retraction applies the perturbation in the body frame).

The *average* NEES over N independent Monte Carlo runs, can be computed by averaging the NEES values:

$$\bar{\eta}_k = \frac{1}{N} \sum_{i=1}^{N} \eta_k^{(i)} \qquad \text{(average NEES)} \qquad (58)$$

where $\eta_k^{(i)}$ is the NEES computed at the i-th Monte Carlo run. If the estimator is consistent, then $N\bar{\eta}_k$ is χ_n^2 chi-square distributed with $n = \dim(\boldsymbol{\epsilon}_k) \cdot N$ degrees of freedom [63, pp. 234]. We evaluate this hypothesis with a χ_n^2 acceptance test [63, pp. 235]. For a significance level $\alpha = 2.5\%$ and $n = \dim(\boldsymbol{\epsilon}_k) \cdot N = 6 \cdot 50$, the acceptance region of the

test is given by the two-sided probability concentration region $\bar{\eta}_k \in [5.0, 7.0]$. If $\bar{\eta}_k$ rises significantly higher than the upper bound, the estimator is overconfident, if it tends below the lower bound, it is conservative. In VIO one usually wants to avoid overconfident estimators: the fact that $\bar{\eta}_k$ exceeds the upper bound is an indicator of the fact that the estimator is including spurious information in the inference process.

In Fig. 9 we report the average NEES of the proposed approach. The average NEES approaches the lower bound but, more importantly, it remains below the upper bound at 7.0 (black dots), which assures that the estimator is not overconfident. We also report the average rotational and translational NEES to allow a comparison with the observability-constrained EKF in [16, 18], which obtains similar results by enforcing explicitly the observability properties in EKF.

3) Bias Estimation Accuracy: Our simulations allow us to compare the estimated gyroscope and accelerometer bias with the true biases that were used to corrupt the simulated inertial measurements. Fig. 10 shows that biases estimated by our approach (in blue) correctly track the ground truth biases (in red). Note that, since we use a smoothing approach, at each step, we potentially change the entire history of the bias estimates, hence we visualize the bias estimates using multiple curves. Each curve represents the history of the estimated biases from time zero (left-most extreme of the blue curve) to the current time (right-most extreme of the blue curve).

4) First-Order Bias Correction: We performed an additional Monte-Carlo analysis to evaluate the a-posteriori bias correction proposed in Section VI-C. The preintegrated measurements are computed with the bias estimate at the time of integration. However, as seen in Fig. 10, the bias estimate for an older preintegrated measurement may change when more information becomes available. To avoid repeating the integration when the bias estimate changes, we perform a first-order correction of the preintegrated measurement according to Eq. (44). The accuracy of this first order bias correction is reported in Fig. 11. To compute the statistics, we integrated 100 random IMU measurements with a given bias estimate $\bar{\mathbf{b}}_i$ which results in the preintegrated measurements $\Delta\tilde{\mathsf{R}}_{ij}(\bar{\mathbf{b}}_i), \Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i)$ and $\Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i)$. Subsequently, a random perturbation $\delta\mathbf{b}_i$ with magnitude between 0.04 and 0.2 was applied to both the gyroscope and accelerometer bias. We repeated the integration at $\bar{\mathbf{b}}_i + \delta\mathbf{b}_i$ to obtain $\Delta\tilde{\mathsf{R}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i), \Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$ and $\Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$. This ground-truth result was then compared against the first-order correction in (44) to compute the error of the approximation. The errors resulting from the first-order approximation are negligible, even for relatively large bias perturbations.

5) Advantages over the Euler-angle-based formulation: In this section we compare the proposed IMU preintegration with the original formulation of [2], based on Euler angles. We observe three main problems with the preintegration using Euler angles, which are avoided in our formulation.

The first drawback is that, in contrast to the integration using the exponential map in Eq. (30), the rotation integration based on Euler angles is only exact up to the first order. For the interested reader, we recall the rotation rate integration using Euler angles in Appendix IX-E. On the left of Fig. 12,
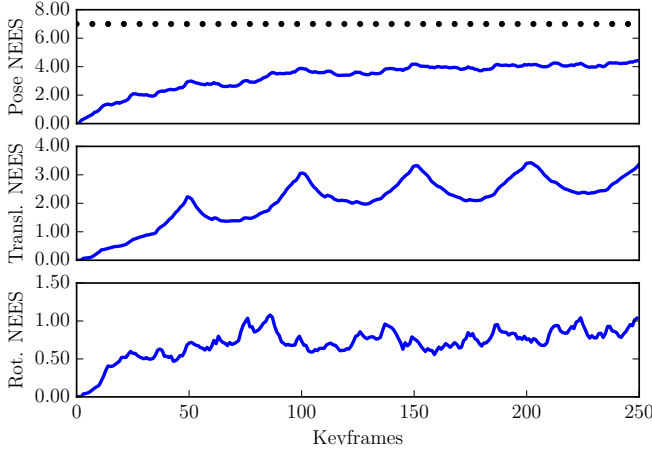
Fig. 9: Normalized Estimation Error Squared (NEES) averaged over 50 Monte Carlo runs. The average NEES is reported for the current pose (top), current position (middle), and current rotation (bottom).
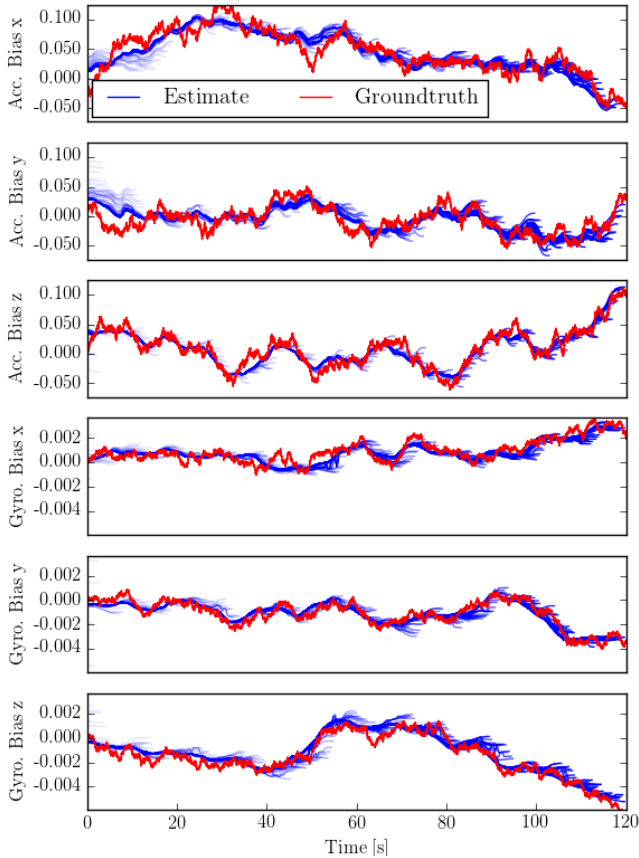

Fig. 10: Comparison between ground truth bias (red line) and estimated bias (blue lines) in a Monte Carlo run.
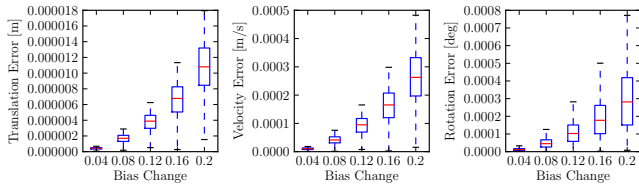

Fig. 11: Error committed when using the first-order approximation (44) instead of repeating the integration, for different bias perturbations. Left: $\Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$ error; Center: $\Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$ error; Right: $\Delta\tilde{\mathbf{R}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$ error. Statistics are computed over 1000 Monte Carlo runs.

we report the integration errors commited by the Euler angle parametrization when integrating angular rates with randomly selected rotation axes and magnitude in the range from 1 to 3 rad/s. Integration error in Euler angles accumulates quickly when the sampling time Δt or the angular rate $\tilde{\omega}$ are large. On the other hand, the proposed approach, which performs integration directly on the rotation manifold, is exact, regardless the values of Δt and $\tilde{\omega}$.

The second drawback is that the Euler parametrization is not *fair* [64], which means that, given the preintegrated Euler angles $\tilde{\theta}$, the negative log-likelihood $\mathcal{L}(\theta) = \frac{1}{2}\|\tilde{\theta} - \theta\|_\Sigma^2$ is not invariant under the action of rigid body transformations. On the right of Fig. 12 we show experimentally how the log-likelihood changes when the frame of reference is rotated around randomly selected rotation axes. This essentially means that an estimator using Euler angles may give different results for different choices of the world frame (*cf.* with Fig. 2). On the other hand, the SO(3) parametrization can be easily seen to be fair (the negative likelihood (16) can be promptly seen to be left invariant), and this is confirmed by Fig. 12 (right).

The third drawback is the existence of so-called *gimball lock* singularities. For a zyx Euler angle parametrization, the singularity is reached at pitch values of $\theta = \frac{\pi}{2} + n\pi$, for $n \in \mathbb{Z}$. To evaluate the effect of the singularity and how it affects the computation of preintegrated measurement noise, we performed the following Monte Carlo analysis. We simulated a set of trajectories that reach maximum pitch values $\theta_{\max}$ of increasing magnitude. For each trajectory, we integrate the rotation uncertainty using the Euler parametrization and the proposed on-manifold approach. The ground-truth covariance is instead obtained through sampling. We use the Kullback-Leibler (KL) divergence to quantify the mismatch between the estimated covariances and the ground-truth one. The results of this experiment are shown in Fig. 13, where we observe that the closer we get to the singularity, the worse is the noise propagation using Euler angles. On the other hand, the proposed approach can accurately estimate the measurement covariance, independently on the motion of the platform.

B. Real Experiments

We integrated the proposed inertial factors in a monocular VIO pipeline to benchmark its performance against the state of the art. In the following, we first discuss our implementation, and then present results from an indoor experiment with motion-capture ground-truth. Finally, we show results from longer trajectories in outdoor experiments. The results confirm that our approach is more accurate than state-of-the-art filtering and fixed-lag smoothing algorithms, and enables fast inference in real-world problems.

1) Implementation: Our implementation consists of a high frame rate tracking front-end based on SVO[3] [65] and an optimization back-end based on iSAM2 [3][4]. The front-end tracks salient features in the image at camera rate while the back-end optimizes in parallel the state of selected *keyframes* as described in this paper.
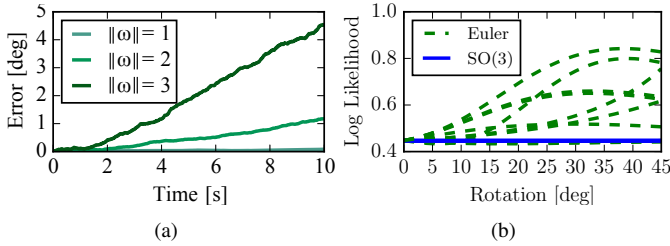
---

[3] http://github.com/uzh-rpg/rpg_svo
[4] http://borg.cc.gatech.edu

Fig. 12: (a) Integration errors committed with the Euler angle parametrization for angular velocities ω of increasing magnitude [rad/s]. (b) Negative log-likelihood of a rotation measurement under the action of random rigid body transformations.
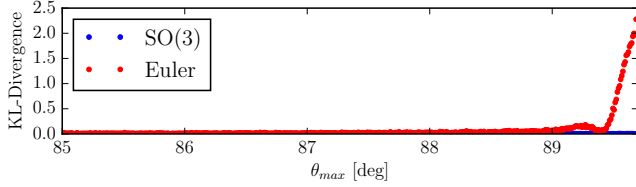

Fig. 13: Kullback-Leibler divergence between the preintegrated rotation covariance –computed using Euler angles (red) and the proposed approach (blue)– and the ground-truth covariance. The Euler angle parametrization degrades close to the singularity at $\theta_{\max} = 90$ deg while the proposed on-manifold approach is accurate regardless of the motion.

SVO [65] is a precise and robust monocular visual odometry system that employs *sparse image alignment* to estimate incremental motion and tracks features by minimizing the photometric error between subsequent frames. The difference to tracking features individually, as in standard Lucas-Kanade tracking, is that we exploit the known depth of features from previous triangulations. This allows us to track all features as a bundle in a single optimization that satisfies epipolar constraints; hence, outliers only originate from erroneous triangulations. In the visual-inertial setting, we further exploit the availability of accurate rotation increments, obtained by integrating angular velocity measurements from the gyroscope. These increments are used as rotation priors in the sparse-image-alignment algorithm, and this increases the overall robustness of the system. The motion estimation is combined with an outlier resistant probabilistic triangulation method that is implemented with a recursive Bayesian filter. The high frame-rate motion estimation combined with the robust depth estimation results in increased robustness in scenes with repetitive and high frequency texture (*e.g.*, asphalt). The output of SVO are selected keyframes with feature-tracks corresponding to triangulated landmarks. This data is passed to the back-end that computes the visual-inertial MAP estimate in Eq. (26) using iSAM2 [3].

We remark that our approach does not marginalize out past states. Therefore, while the approach is designed for fast visual-inertial odometry, if desired, it could be readily extended to incorporate loop closures.

2) Indoor Experiments: The indoor experiment shows that the proposed approach is more accurate than two competitive state-of-the-art approaches, namely OKVIS[5] [24], and MSCKF [5]. The experiment is performed on the 430m-long indoor trajectory of Fig. 14. The dataset was recorded

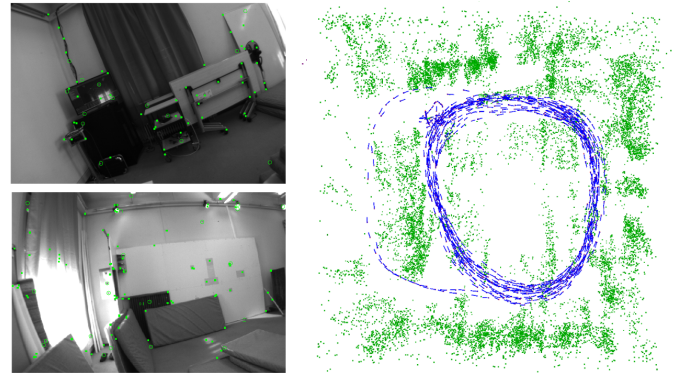[5]https://github.com/ethz-asl/okvis

Fig. 14: Left: two images from the indoor trajectory dataset with tracked features in green. Right: top view of the trajectory estimate produced by our approach (blue) and 3D landmarks triangulated from the trajectory (green).
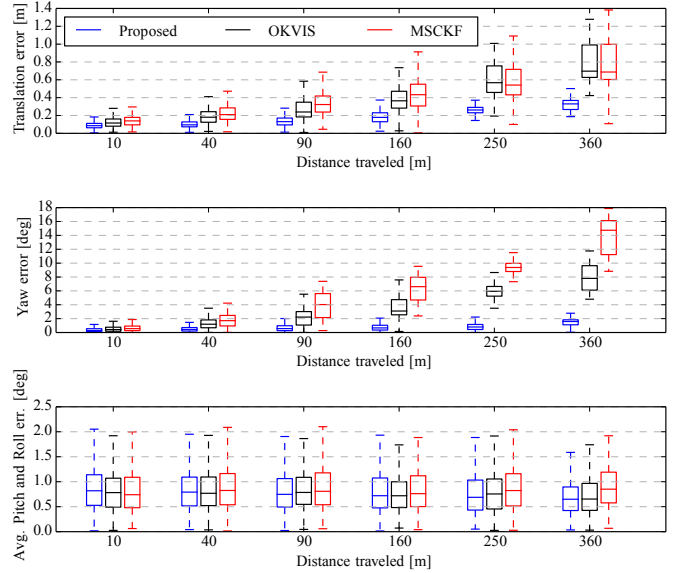

Fig. 15: Comparison of the proposed approach versus the OKVIS algorithm [24] and an implementation of the MSCKF filter [5]. Relative errors are measured over different segments of the trajectory, of length $\{10, 40, 90, 160, 250, 360\}$m, according to the odometric error metric in [68].
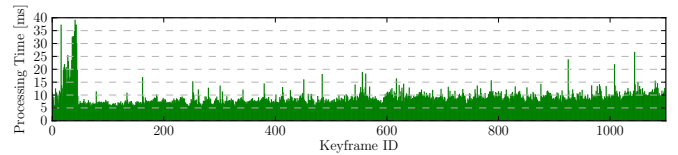

Fig. 16: Processing-time per keyframe for the proposed VIO approach.

with a forward-looking VI-Sensor [66] that consists of an ADIS16448 MEMS IMU and two embedded WVGA monochrome cameras (we only use the left camera). Intrinsic and extrinsic calibration was obtained using [59]. The camera runs at 20Hz and the IMU at 800Hz. Ground truth poses are provided by a Vicon system mounted in the room; the *hand-eye* calibration between the Vicon markers and the camera is computed using a least-squares method [67].

Fig. 15 compares the proposed system against the OKVIS algorithm [24], and an implementation of the MSCKF filter [5]. Both these algorithms currently represent the state-of-the-art in VIO, OKVIS for optimization-based approaches, and MSCKF for filtering methods. We obtained the datasets as
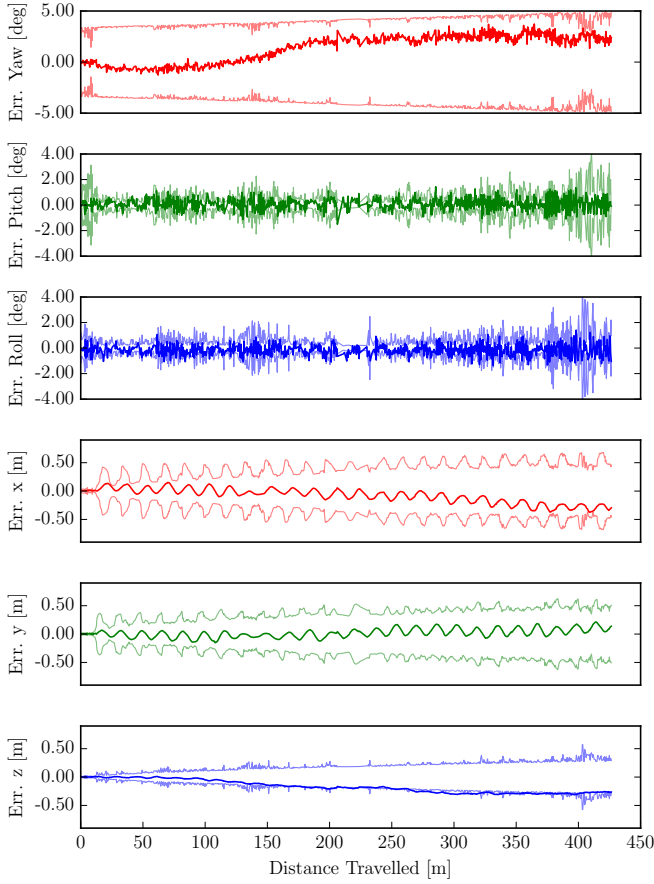
Fig. 17: Orientation and position errors with 3σ bounds for the real indoor experiment in Fig. 14.

Fig. 18: Outdoor trajectory (length: 300m) around a building with identical start and end point at coordinates $(0,0,0)$. The end-to-end error of the proposed approach is 1.0m. Google Tango accumulated 2.2m drift. The green dots are the 3D points triangulated from our trajectory estimate.

Fig. 19: Real test comparing the proposed VIO approach against Google Tango. The 160m-long trajectory starts at $(0,0,0)$ (ground floor), goes up till the 3rd floor of a building, and returns to the initial point. The figure shows a side view (left) and a top view (right) of the trajectory estimates for our approach (blue) and Tango (red). Google Tango accumulates 1.4m error, while the proposed approach only has 0.5m drift. 3D points triangulated from our trajectory estimate are shown in green for visualization purposes.

well as the trajectories computed with OKVIS and MSCKF from the authors of [24]. We use the relative error metrics proposed in [68] to obtain error statistics. The metric evaluates the relative error by averaging the drift over trajectory segments of different length ($\{10, 40, 90, 160, 250, 360\}$m in Fig. 15). Our approach exhibits less drift than the state-of-the-art, achieving 0.3m drift on average over 360m traveled distance; OKVIS and MSCKF accumulate an average error of 0.7m. We observe significantly less drift in yaw direction in the proposed approach while the error in pitch and and roll direction is constant for all methods due to the observability of the gravity direction.

We highlight that these three algorithms use different front-end feature tracking systems, which influence the overall performance of the approach. Therefore, while in Section VIII-A we discussed only aspects related to the preintegration theory, in this section we evaluate the proposed system as a whole (SVO, preintegration, structureless vision factors, iSAM2).

Evaluating consistency in real experiments by means of analysing the average NEES is difficult as one would have to evalate and average the results of multiple runs of the same trajectory with different realizations of sensor noise. In Figure 17 we show the error plots with the 3-sigma bounds for a single run. The result is consistent as the estimation errors remain within the bounds of the estimated uncertainty. In this experiment, we aligned only the first frame of the trajectory with the vicon trajectory. Therefore, analyzing the drift over
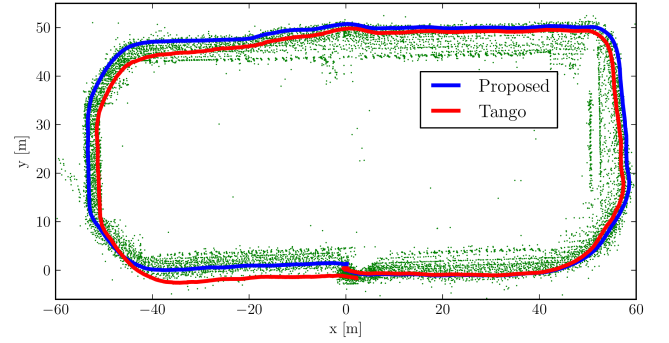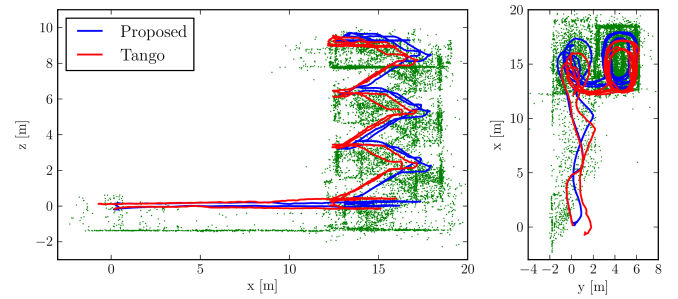
400 meters is very prone to errors in the initial pose from the ground-truth or errors in the hand-eye calibration of the system.

Figure 16 illustrates the time required by the back-end to compute the full MAP estimate, by running iSAM2 with 10 optimization iterations. The experiment was performed on a standard laptop (Intel i7, 2.4 GHz). The average update time for iSAM2 is 10ms. The peak corresponds to the start of the experiment in which the camera was not moving. In this case the number of tracked features becomes very large making the back-end slightly slower. The SVO front-end requires approximately 3ms to process a frame on the laptop while the back-end runs in a parallel thread and optimizes only keyframes. Although the processing times of OKVIS were not reported, the approach is described as computationally demanding [24]. OKVIS needs to repeat IMU integration at every change of the linearization point, which we avoid by using the preintegrated IMU measurements.

3) Outdoor Experiments: The second experiment is performed on an outdoor trajectory, and compares the proposed approach against the *Google* Tango *Peanut* sensor (mapper version 3.15), which is an *engineered* VIO system. We rigidly attached the VI-Sensor to a Tango device and walked around an office building. Fig. 18 depicts the trajectory estimates for our approach and Google Tango. The trajectory starts and ends

at the same location, hence we can report the end-to-end error which is 1.5m for the proposed approach and 2.2m for the Google Tango sensor.

In Fig. 18 we also show the estimated landmark positions (in green). 3D points are not estimated by our approach (which uses a structureless vision model), but are triangulated from our trajectory estimate for visualization purposes.

The third experiment is the one in Fig. 19. The trajectory goes across three floors of an office building and eventually returns to the initial location on the ground floor. Also in this case the proposed approach guarantees a very small end-to-end error (0.5m), while Tango accumulates 1.4m error.

We remark that Tango and our system use different sensors, hence the reported end-to-end errors only allow for a qualitative comparison. However, the IMUs of both sensors exhibit similar noise characteristics [69, 70] and the Tango camera has a significantly larger field-of-view and better shutter speed control than our sensor. Therefore, the comparison is still valuable to assess the accuracy of the proposed approach.

A video demonstrating the execution of our approach for the real experiments discussed in this section can be viewed at https://youtu.be/CsJkci5lfco

IX. Conclusion

This paper proposes a novel preintegration theory, which provides a grounded way to model a large number of IMU measurements as a single motion constraint. Our proposal improves over related works that perform integration in a global frame, e.g., [5, 23], as we do not commit to a linearization point during integration. Moreover, it leverages the seminal work on preintegration [2], bringing to maturity the preintegration and uncertainty propagation in $SO(3)$.

As a second contribution, we discuss how to use the preintegrated IMU model in a VIO pipeline; we adopt a structureless model for visual measurements which avoids optimizing over 3D landmarks. Our VIO approach uses iSAM2 to perform constant-time incremental smoothing.

An efficient implementation of our approach requires 10ms to perform inference (back-end), and 3ms for feature tracking (front-end). Experimental results also confirm that our approach is more accurate than state-of-the-art alternatives, including filtering and optimization-based techniques.

We release the source-code of the IMU preintegration and the structureless vision factors in the GTSAM 4.0 optimization toolbox [7] and provide additional theoretical derivations and implementation details in the Appendix of this paper.

Appendix

A. Iterative Noise Propagation

In this section we show that the computation of the preintegrated noise covariance, discussed in Section VI-B, can be carried out in iterative form, which leads to simpler expressions and is more amenable for online inference.

Let us start from the preintegrated rotation noise in (42). To write $\delta\phi_{ij}$ in iterative form, we simply take the last term ($k = j-1$) out of the sum and rearrange the terms:

$$\delta\phi_{ij} \simeq \sum_{k=i}^{j-1} \Delta\tilde{\mathsf{R}}_{k+1j}^{\mathsf{T}} \mathsf{J}_r^k \boldsymbol{\eta}_k^{gd} \Delta t \tag{59}$$

$$= \sum_{k=i}^{j-2} \Delta\tilde{\mathsf{R}}_{k+1j}^{\mathsf{T}} \mathsf{J}_r^k \boldsymbol{\eta}_k^{gd} \Delta t + \overbrace{\Delta\tilde{\mathsf{R}}_{jj}^{\mathsf{T}}}^{=\mathbf{I}_{3\times3}} \mathsf{J}_r^{j-1} \boldsymbol{\eta}_{j-1}^{gd} \Delta t$$

$$= \sum_{k=i}^{j-2} (\overbrace{\Delta\tilde{\mathsf{R}}_{k+1j-1}\Delta\tilde{\mathsf{R}}_{j-1j}}^{=\Delta\tilde{\mathsf{R}}_{k+1j}})^{\mathsf{T}} \mathsf{J}_r^k \boldsymbol{\eta}_k^{gd} \Delta t + \mathsf{J}_r^{j-1} \boldsymbol{\eta}_{j-1}^{gd} \Delta t$$

$$= \Delta\tilde{\mathsf{R}}_{j-1j}^{\mathsf{T}} \sum_{k=i}^{j-2} \Delta\tilde{\mathsf{R}}_{k+1j-1}^{\mathsf{T}} \mathsf{J}_r^k \boldsymbol{\eta}_k^{gd} \Delta t + \mathsf{J}_r^{j-1} \boldsymbol{\eta}_{j-1}^{gd} \Delta t$$

$$= \Delta\tilde{\mathsf{R}}_{j-1j}^{\mathsf{T}} \delta\phi_{ij-1} + \mathsf{J}_r^{j-1} \boldsymbol{\eta}_{j-1}^{gd} \Delta t.$$

Repeating the same process for $\delta\mathbf{v}_{ij}$ in (43):

$$\delta\mathbf{v}_{ij} = \sum_{k=i}^{j-1} \left[-\Delta\tilde{\mathsf{R}}_{ik}(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a)^\wedge \delta\phi_{ik}\Delta t + \Delta\tilde{\mathsf{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t \right] \tag{60}$$

$$= \sum_{k=i}^{j-2} \left[-\Delta\tilde{\mathsf{R}}_{ik}(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a)^\wedge \delta\phi_{ik}\Delta t + \Delta\tilde{\mathsf{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t \right]$$

$$\qquad - \Delta\tilde{\mathsf{R}}_{ij-1}(\tilde{\mathbf{a}}_{j-1} - \mathbf{b}_i^a)^\wedge \delta\phi_{ij-1}\Delta t + \Delta\tilde{\mathsf{R}}_{ij-1}\boldsymbol{\eta}_{j-1}^{ad}\Delta t$$

$$= \delta\mathbf{v}_{ij-1} - \Delta\tilde{\mathsf{R}}_{ij-1}(\tilde{\mathbf{a}}_{j-1} - \mathbf{b}_i^a)^\wedge \delta\phi_{ij-1}\Delta t + \Delta\tilde{\mathsf{R}}_{ij-1}\boldsymbol{\eta}_{j-1}^{ad}\Delta t$$

Doing the same for $\delta\mathbf{p}_{ij}$ in (43), and noting that $\delta\mathbf{p}_{ij}$ can be written as a function of $\delta\mathbf{v}_{ij}$ (*cf.* with the expression of $\delta\mathbf{v}_{ij}$ in (43)):

$$\delta\mathbf{p}_{ij} = \sum_{k=i}^{j-1} \left[\delta\mathbf{v}_{ik}\Delta t - \frac{1}{2}\Delta\tilde{\mathsf{R}}_{ik}(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a)^\wedge \delta\phi_{ik}\Delta t^2 + \frac{1}{2}\Delta\tilde{\mathsf{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t^2 \right]$$

$$= \sum_{k=i}^{j-2} \left[\delta\mathbf{v}_{ik}\Delta t - \frac{1}{2}\Delta\tilde{\mathsf{R}}_{ik}(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a)^\wedge \delta\phi_{ik}\Delta t^2 + \frac{1}{2}\Delta\tilde{\mathsf{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t^2 \right]$$

$$+ \delta\mathbf{v}_{ij-1}\Delta t - \frac{1}{2}\Delta\tilde{\mathsf{R}}_{ij-1}(\tilde{\mathbf{a}}_{j-1} - \mathbf{b}_i^a)^\wedge \delta\phi_{ij-1}\Delta t^2 + \frac{1}{2}\Delta\tilde{\mathsf{R}}_{ij-1}\boldsymbol{\eta}_{j-1}^{ad}\Delta t^2$$

$$= \delta\mathbf{p}_{ij-1} + \delta\mathbf{v}_{ij-1}\Delta t - \frac{1}{2}\Delta\tilde{\mathsf{R}}_{ij-1}(\tilde{\mathbf{a}}_{j-1} - \mathbf{b}_i^a)^\wedge \delta\phi_{ij-1}\Delta t^2$$

$$+ \frac{1}{2}\Delta\tilde{\mathsf{R}}_{ij-1}\boldsymbol{\eta}_{j-1}^{ad}\Delta t^2 \tag{61}$$

Recalling that $\boldsymbol{\eta}_{ik}^\Delta \doteq [\delta\phi_{ik}, \delta\mathbf{v}_{ik}, \delta\mathbf{p}_{ik}]$, and defining the IMU measurement noise $\boldsymbol{\eta}_k^d \doteq [\boldsymbol{\eta}_k^{gd}\ \boldsymbol{\eta}_k^{ad}]$,[6] we can finally write Eqs. (59)-(61) in compact matrix form as:

$$\boldsymbol{\eta}_{ij}^\Delta = \mathbf{A}_{j-1}\boldsymbol{\eta}_{ij-1}^\Delta + \mathbf{B}_{j-1}\boldsymbol{\eta}_{j-1}^d, \tag{62}$$

From the linear model (62) and given the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\eta}} \in \mathbb{R}^{6\times6}$ of the raw IMU measurements noise $\boldsymbol{\eta}_k^d$, it is now possible to compute the preintegrated measurement covariance iteratively:

$$\boldsymbol{\Sigma}_{ij} = \mathbf{A}_{j-1}\boldsymbol{\Sigma}_{ij-1}\mathbf{A}_{j-1}^{\mathsf{T}} + \mathbf{B}_{j-1}\boldsymbol{\Sigma}_{\boldsymbol{\eta}}\mathbf{B}_{j-1}^{\mathsf{T}} \tag{63}$$

starting from initial conditions $\boldsymbol{\Sigma}_{ii} = \mathbf{0}_{9\times9}$.

---

[6]Both $\boldsymbol{\eta}_{ij}^\Delta$ and $\boldsymbol{\eta}_k^d$ are column vectors: we omit the transpose in the definition to keep notation simple.

B. Bias Correction via First-Order Updates

In this section we provide a complete derivation of the first-order bias correction proposed in Section VI-C.

Let us assume that we have computed the preintegrated variables at a given bias estimate $\bar{\mathbf{b}}_i \doteq [\bar{\mathbf{b}}_i^g \quad \bar{\mathbf{b}}_i^a]$, and let us denote the corresponding preintegrated measurements as

$$\Delta\bar{\mathsf{R}}_{ij} \doteq \Delta\tilde{\mathsf{R}}_{ij}(\bar{\mathbf{b}}_i), \; \Delta\bar{\mathbf{v}}_{ij} \doteq \Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i), \; \Delta\bar{\mathbf{p}}_{ij} \doteq \Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i). \quad (64)$$

In this section we want to devise an expression to "update" $\Delta\bar{\mathsf{R}}_{ij}$, $\Delta\bar{\mathbf{v}}_{ij}$, $\Delta\bar{\mathbf{p}}_{ij}$ when our bias estimate changes.

Consider the case in which we get a new estimate $\hat{\mathbf{b}}_i \leftarrow \bar{\mathbf{b}}_i + \delta\mathbf{b}_i$, where $\delta\mathbf{b}_i$ is a *small* correction w.r.t. the previous estimate $\bar{\mathbf{b}}_i$.

We start with the bias correction for the preintegrated rotation measurement. The key idea here is to write $\Delta\tilde{\mathsf{R}}_{ij}(\hat{\mathbf{b}}_i)$ (the preintegrated measurement at the new bias estimate) as a function of $\Delta\tilde{\mathsf{R}}_{ij}$ (the preintegrated measurement at the old bias estimate), "plus" a first-order correction. Recalling Eq. (35), we write $\Delta\tilde{\mathsf{R}}_{ij}(\hat{\mathbf{b}}_i)$ as:

$$\Delta\tilde{\mathsf{R}}_{ij}(\hat{\mathbf{b}}_i) = \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \hat{\mathbf{b}}_i^g\right)\Delta t\right) \quad (65)$$

Substituting $\hat{\mathbf{b}}_i = \bar{\mathbf{b}}_i + \delta\mathbf{b}_i$ in the previous expression and using the first-order approximation (4) in each factor (we assumed small $\delta\mathbf{b}_i$):

$$\Delta\tilde{\mathsf{R}}_{ij}(\hat{\mathbf{b}}_i) = \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - (\bar{\mathbf{b}}_i^g + \delta\mathbf{b}_i^g)\right)\Delta t\right) \quad (66)$$

$$\simeq \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \bar{\mathbf{b}}_i^g\right)\Delta t\right) \mathrm{Exp}\left(-\mathsf{J}_r^k \, \delta\mathbf{b}_i^g \, \Delta t\right).$$

Now, we rearrange the terms in the product, by "moving" the terms including $\delta\mathbf{b}_i^{gd}$ to the end, using the relation (11):

$$\Delta\tilde{\mathsf{R}}_{ij}(\hat{\mathbf{b}}_i) = \Delta\bar{\mathsf{R}}_{ij} \prod_{k=i}^{j-1} \mathrm{Exp}\left(-\Delta\tilde{\mathsf{R}}_{k+1j}(\bar{\mathbf{b}}_i)^\mathsf{T} \mathsf{J}_r^k \, \delta\mathbf{b}_i^g \, \Delta t\right), \quad (67)$$

where we used the fact that by definition it holds that $\Delta\bar{\mathsf{R}}_{ij} = \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \bar{\mathbf{b}}_i^g\right)\Delta t\right)$. Repeated application of the first-order approximation (7) (recall that $\delta\mathbf{b}_i^g$ is small, hence the right Jacobians are close to the identity) produces:

$$\Delta\tilde{\mathsf{R}}_{ij}(\hat{\mathbf{b}}_i) \simeq \Delta\bar{\mathsf{R}}_{ij}\mathrm{Exp}\left(\sum_{k=i}^{j-1}-\Delta\tilde{\mathsf{R}}_{k+1j}(\bar{\mathbf{b}}_i)^\mathsf{T}\mathsf{J}_r^k\delta\mathbf{b}_i^g\Delta t\right)$$

$$= \Delta\bar{\mathsf{R}}_{ij}\mathrm{Exp}\left(\frac{\partial\Delta\bar{\mathsf{R}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g\right) \quad (68)$$

Using (68) we can now update the preintegrated rotation measurement $\Delta\tilde{\mathsf{R}}_{ij}(\bar{\mathbf{b}}_i)$ to get $\Delta\tilde{\mathsf{R}}_{ij}(\hat{\mathbf{b}}_i)$ without repeating the integration.

Let us now focus on the bias correction of the preintegrated velocity $\Delta\tilde{\mathbf{v}}_{ij}(\hat{\mathbf{b}}_i)$:

$$\Delta\tilde{\mathbf{v}}_{ij}(\hat{\mathbf{b}}_i) = \sum_{k=i}^{j-1} \Delta\tilde{\mathsf{R}}_{ik}(\hat{\mathbf{b}}_i)\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a - \delta\mathbf{b}_i^a\right)\Delta t \quad (69)$$

$$\overset{(68)}{\simeq} \sum_{k=i}^{j-1} \Delta\bar{\mathsf{R}}_{ik}\mathrm{Exp}\left(\frac{\partial\Delta\bar{\mathsf{R}}_{ik}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g\right)\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a - \delta\mathbf{b}_i^a\right)\Delta t$$

$$\overset{(4)}{\simeq} \sum_{k=i}^{j-1} \Delta\bar{\mathsf{R}}_{ik}\left(\mathbf{I} + \left(\frac{\partial\Delta\bar{\mathsf{R}}_{ik}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g\right)^\wedge\right)\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a - \delta\mathbf{b}_i^a\right)\Delta t$$

$$\overset{(a)}{\simeq} \Delta\bar{\mathbf{v}}_{ij} - \sum_{k=i}^{j-1}\Delta\bar{\mathsf{R}}_{ik}\Delta t\delta\mathbf{b}_i^a + \sum_{k=i}^{j-1}\Delta\bar{\mathsf{R}}_{ik}\left(\frac{\partial\Delta\bar{\mathsf{R}}_{ik}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g\right)^\wedge\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a\right)\Delta t$$

$$\overset{(2)}{=} \Delta\bar{\mathbf{v}}_{ij} - \sum_{k=i}^{j-1}\Delta\bar{\mathsf{R}}_{ik}\Delta t\delta\mathbf{b}_i^a - \sum_{k=i}^{j-1}\Delta\bar{\mathsf{R}}_{ik}\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a\right)^\wedge\frac{\partial\Delta\bar{\mathsf{R}}_{ik}}{\partial\mathbf{b}^g}\Delta t \, \delta\mathbf{b}_i^g$$

$$= \Delta\bar{\mathbf{v}}_{ij} + \frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^a}\delta\mathbf{b}_i^a + \frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g$$

Where for (a), we used $\Delta\bar{\mathbf{v}}_{ij} = \sum_{k=i}^{j-1}\Delta\bar{\mathsf{R}}_{ik}\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a\right)\Delta t$. Exactly the same derivation can be repeated for $\Delta\tilde{\mathbf{p}}_{ij}(\hat{\mathbf{b}}_i)$. Summarizing, the Jacobians used for the a posteriori bias update in Eq. (44) are:

$$\begin{aligned}
\frac{\partial\Delta\bar{\mathsf{R}}_{ij}}{\partial\mathbf{b}^g} &= -\sum_{k=i}^{j-1}\left[\Delta\bar{\mathsf{R}}_{k+1j}(\bar{\mathbf{b}}_i)^\mathsf{T} \mathsf{J}_r^k \, \Delta t\right] \\
\frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^a} &= -\sum_{k=i}^{j-1}\Delta\bar{\mathsf{R}}_{ik}\Delta t \\
\frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^g} &= -\sum_{k=i}^{j-1}\Delta\bar{\mathsf{R}}_{ik}\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a\right)^\wedge\frac{\partial\Delta\bar{\mathsf{R}}_{ik}}{\partial\mathbf{b}^g}\Delta t \\
\frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^a} &= \sum_{k=i}^{j-1}\frac{\partial\Delta\bar{\mathbf{v}}_{ik}}{\partial\mathbf{b}^a}\Delta t - \frac{1}{2}\Delta\bar{\mathsf{R}}_{ik}\Delta t^2 \\
\frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^g} &= \sum_{k=i}^{j-1}\frac{\partial\Delta\bar{\mathbf{v}}_{ik}}{\partial\mathbf{b}^g}\Delta t - \frac{1}{2}\Delta\bar{\mathsf{R}}_{ik}\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a\right)^\wedge\frac{\partial\Delta\bar{\mathsf{R}}_{ik}}{\partial\mathbf{b}^g}\Delta t^2
\end{aligned}$$

Note that the Jacobians can be computed incrementally, as new measurements arrive.

C. Jacobians of Residual Errors

In this section we provide analytic expressions for the Jacobian matrices of the residual errors in Eq. (45). These Jacobians are crucial when using iterative optimization techniques (e.g., the Gauss-Newton method of Section III-C) to minimize the cost in Eq. (26).

"Lifting" the cost function (see Section III-C) consists in substituting the following retractions:

$$\begin{aligned}
\mathsf{R}_i &\leftarrow \mathsf{R}_i \mathrm{Exp}(\delta\boldsymbol{\phi}_i), & \mathsf{R}_j &\leftarrow \mathsf{R}_j \mathrm{Exp}(\delta\boldsymbol{\phi}_j), \\
\mathbf{p}_i &\leftarrow \mathbf{p}_i + \mathsf{R}_i\delta\mathbf{p}_i, & \mathbf{p}_j &\leftarrow \mathbf{p}_j + \mathsf{R}_j\delta\mathbf{p}_j, \\
\mathbf{v}_i &\leftarrow \mathbf{v}_i + \delta\mathbf{v}_i, & \mathbf{v}_j &\leftarrow \mathbf{v}_j + \delta\mathbf{v}_j, \\
\delta\mathbf{b}_i^g &\leftarrow \delta\mathbf{b}_i^g + \tilde{\delta}\mathbf{b}_i^g, & \delta\mathbf{b}_i^a &\leftarrow \delta\mathbf{b}_i^a + \tilde{\delta}\mathbf{b}_i^a,
\end{aligned} \quad (70)$$

The process of lifting makes the residual errors a function defined on a vector space, on which it is easy to compute Jacobians. Therefore, in the following sections we derive the Jacobians w.r.t. the vectors $\delta\boldsymbol{\phi}_i, \delta\mathbf{p}_i, \delta\mathbf{v}_i, \delta\boldsymbol{\phi}_j, \delta\mathbf{p}_j, \delta\mathbf{v}_j, \tilde{\delta}\mathbf{b}_i^g, \tilde{\delta}\mathbf{b}_i^a$.

1) Jacobians of $\mathbf{r}_{\Delta\mathbf{p}_{ij}}$: Since $\mathbf{r}_{\Delta\mathbf{p}_{ij}}$ is linear in $\delta\mathbf{b}_i^g$ and $\delta\mathbf{b}_i^a$, and the retraction is simply a vector sum, the Jacobians of $\mathbf{r}_{\Delta\mathbf{p}_{ij}}$ w.r.t. $\tilde{\delta}\mathbf{b}_i^g, \tilde{\delta}\mathbf{b}_i^a$ are simply the matrix coefficients of $\delta\mathbf{b}_i^g$ and $\delta\mathbf{b}_i^a$. Moreover, R_j and $\mathbf{v}_j$ do not appear in $\mathbf{r}_{\Delta\mathbf{p}_{ij}}$, hence the Jacobians w.r.t. $\delta\boldsymbol{\phi}_j, \delta\mathbf{v}_j$ are zero. Let us focus on the remaining Jacobians:

$$\mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathbf{p}_i + \mathsf{R}_i\delta\mathbf{p}_i) = \mathsf{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathsf{R}_i\delta\mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) - C$$
$$= \mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathbf{p}_i) + (-\mathbf{I}_{3\times 1})\delta\mathbf{p}_i \quad (71)$$

$$\mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathbf{p}_j + \mathsf{R}_j\delta\mathbf{p}_j) = \mathsf{R}_i^\mathsf{T}\left(\mathbf{p}_j + \mathsf{R}_j\delta\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) - C$$
$$= \mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathbf{p}_j) + (\mathsf{R}_i^\mathsf{T}\mathsf{R}_j)\delta\mathbf{p}_j \quad (72)$$

$$\mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathbf{v}_i + \delta\mathbf{v}_i) = \mathsf{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \delta\mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) - C$$
$$= \mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathbf{v}_i) + (-\mathsf{R}_i^\mathsf{T}\Delta t_{ij})\delta\mathbf{v}_i \quad (73)$$

$$\mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathsf{R}_i \, \mathrm{Exp}(\delta\boldsymbol{\phi}_i)) \quad (74)$$
$$= (\mathsf{R}_i\mathrm{Exp}(\delta\boldsymbol{\phi}_i))^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) - C$$
$$\overset{(4)}{\simeq} (\mathbf{I} - \delta\boldsymbol{\phi}_i^\wedge)\mathsf{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) - C$$
$$\overset{(2)}{=} \mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathsf{R}_i) + \left(\mathsf{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right)\right)^\wedge\delta\boldsymbol{\phi}_i.$$

Where we used the shorthand $C \doteq \Delta\tilde{\mathbf{p}}_{ij} + \frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}_i^g}\delta\mathbf{b}_i^g + \frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}_i^a}\delta\mathbf{b}_i^a$. Summarizing, the Jacobians of $\mathbf{r}_{\Delta\mathbf{p}_{ij}}$ are:

$$\begin{aligned}
\frac{\partial\mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial\delta\boldsymbol{\phi}_i} &= (\mathsf{R}_i^\mathsf{T}(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2))^\wedge & \frac{\partial\mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial\delta\boldsymbol{\phi}_j} &= \mathbf{0} \\
\frac{\partial\mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial\delta\mathbf{p}_i} &= -\mathbf{I}_{3\times 1} & \frac{\partial\mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial\delta\mathbf{p}_j} &= \mathsf{R}_i^\mathsf{T}\mathsf{R}_j \\
\frac{\partial\mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial\delta\mathbf{v}_i} &= -\mathsf{R}_i^\mathsf{T}\Delta t_{ij} & \frac{\partial\mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial\delta\mathbf{v}_j} &= \mathbf{0} \\
\frac{\partial\mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial\tilde{\delta}\mathbf{b}_i^a} &= -\frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}_i^a} & \frac{\partial\mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial\tilde{\delta}\mathbf{b}_i^g} &= -\frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}_i^g}
\end{aligned}$$

2) Jacobians of $\mathbf{r}_{\Delta \mathbf{v}_{ij}}$: As in the previous section, $\mathbf{r}_{\Delta \mathbf{v}_{ij}}$ is linear in $\delta \mathbf{b}_i^g$ and $\delta \mathbf{b}_i^a$, hence the Jacobians of $\mathbf{r}_{\Delta \mathbf{v}_{ij}}$ w.r.t. $\tilde{\delta}\mathbf{b}_i^g, \tilde{\delta}\mathbf{b}_i^a$ are simply the matrix coefficients of $\delta \mathbf{b}_i^g$ and $\delta \mathbf{b}_i^a$. Moreover, $\mathtt{R}_j$, $\mathbf{p}_i$, and $\mathbf{p}_j$ do not appear in $\mathbf{r}_{\Delta \mathbf{v}_{ij}}$, hence the Jacobians w.r.t. $\delta \boldsymbol{\phi}_j, \delta \mathbf{p}_i, \delta \mathbf{p}_j$ are zero. The remaining Jacobias are computed as:

$$
\begin{aligned}
\mathbf{r}_{\Delta \mathbf{v}_{ij}}(\mathbf{v}_i + \delta \mathbf{v}_i) &= \mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \delta \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) - D \\
&= \mathbf{r}_{\Delta \mathbf{v}}(\mathbf{v}_i) - \mathtt{R}_i^\mathsf{T}\delta \mathbf{v}_i
\end{aligned} \tag{75}
$$

$$
\begin{aligned}
\mathbf{r}_{\Delta \mathbf{v}_{ij}}(\mathbf{v}_j + \delta \mathbf{v}_j) &= \mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j + \delta \mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) - D \\
&= \mathbf{r}_{\Delta \mathbf{v}}(\mathbf{v}_j) + \mathtt{R}_i^\mathsf{T}\delta \mathbf{v}_j
\end{aligned} \tag{76}
$$

$$
\begin{aligned}
\mathbf{r}_{\Delta \mathbf{v}_{ij}}(\mathtt{R}_i \operatorname{Exp}(\delta \boldsymbol{\phi}_i)) &= (\mathtt{R}_i \operatorname{Exp}(\delta \boldsymbol{\phi}_i))^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) - D \\
&\overset{(4)}{\simeq} (\mathbf{I} - \delta \boldsymbol{\phi}_i^\wedge)\mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) - D \\
&\overset{(2)}{=} \mathbf{r}_{\Delta \mathbf{v}}(\mathtt{R}_i) + \left(\mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right)\right)^\wedge \delta \boldsymbol{\phi}_i,
\end{aligned} \tag{77}
$$

with $D \doteq \left[\Delta \tilde{\mathbf{v}}_{ij} + \frac{\partial \Delta \bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}_i^g}\delta \mathbf{b}_i^g + \frac{\partial \Delta \bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}_i^a}\delta \mathbf{b}_i^a\right]$. Summarizing, the Jacobians of $\mathbf{r}_{\Delta \mathbf{v}_{ij}}$ are:

$$
\begin{array}{ll}
\frac{\partial \mathbf{r}_{\Delta \mathbf{v}_{ij}}}{\partial \delta \boldsymbol{\phi}_i} = \left(\mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right)\right)^\wedge & \frac{\partial \mathbf{r}_{\Delta \mathbf{v}_{ij}}}{\partial \delta \boldsymbol{\phi}_j} = \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta \mathbf{v}_{ij}}}{\partial \delta \mathbf{p}_i} = \mathbf{0} & \frac{\partial \mathbf{r}_{\Delta \mathbf{v}_{ij}}}{\partial \delta \mathbf{p}_j} = \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta \mathbf{v}_{ij}}}{\partial \delta \mathbf{v}_i} = -\mathtt{R}_i^\mathsf{T} & \frac{\partial \mathbf{r}_{\Delta \mathbf{v}_{ij}}}{\partial \delta \mathbf{v}_j} = \mathtt{R}_i^\mathsf{T} \\
\frac{\partial \mathbf{r}_{\Delta \mathbf{v}_{ij}}}{\partial \delta \mathbf{b}_i^a} = -\frac{\partial \Delta \bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}_i^a} & \frac{\partial \mathbf{r}_{\Delta \mathbf{v}_{ij}}}{\partial \delta \mathbf{b}_i^g} = -\frac{\partial \Delta \bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}_i^g}
\end{array}
$$

3) Jacobians of $\mathbf{r}_{\Delta \mathtt{R}_{ij}}$: The derivation of the Jacobians of $\mathbf{r}_{\Delta \mathtt{R}_{ij}}$ is slightly more involved. We first note that $\mathbf{p}_i, \mathbf{p}_j, \mathbf{v}_i, \mathbf{v}_j, \delta \mathbf{b}_i^a$ do not appear in the expression of $\mathbf{r}_{\Delta \mathtt{R}_{ij}}$, hence the corresponding Jacobians are zero. The remaining Jacobians can be computed as follows:

$$
\begin{aligned}
\mathbf{r}_{\Delta \mathtt{R}_{ij}}(\mathtt{R}_i \operatorname{Exp}(\delta \boldsymbol{\phi}_i)) &= \operatorname{Log}\left(\left(\Delta \tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)E\right)^\mathsf{T}(\mathtt{R}_i \operatorname{Exp}(\delta \boldsymbol{\phi}_i))^\mathsf{T}\mathtt{R}_j\right) \\
&= \operatorname{Log}\left(\left(\Delta \tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)E\right)^\mathsf{T}\operatorname{Exp}(-\delta \boldsymbol{\phi}_i)\mathtt{R}_i^\mathsf{T}\mathtt{R}_j\right) \\
&\overset{(11)}{=} \operatorname{Log}\left(\left(\Delta \tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)E\right)^\mathsf{T}\mathtt{R}_i^\mathsf{T}\mathtt{R}_j \operatorname{Exp}(-\mathtt{R}_j^\mathsf{T}\mathtt{R}_i \delta \boldsymbol{\phi}_i)\right) \\
&\overset{(9)}{\simeq} \mathbf{r}_{\Delta \mathtt{R}}(\mathtt{R}_i) - \mathtt{J}_r^{-1}(\mathbf{r}_{\Delta \mathtt{R}}(\mathtt{R}_i))\mathtt{R}_j^\mathsf{T}\mathtt{R}_i \delta \boldsymbol{\phi}_i
\end{aligned} \tag{78}
$$

$$
\begin{aligned}
\mathbf{r}_{\Delta \mathtt{R}_{ij}}(\mathtt{R}_j \operatorname{Exp}(\delta \boldsymbol{\phi}_j)) &= \operatorname{Log}\left(\left(\Delta \tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)E\right)^\mathsf{T}\mathtt{R}_i^\mathsf{T}(\mathtt{R}_j \operatorname{Exp}(\delta \boldsymbol{\phi}_j))\right) \\
&\overset{(9)}{\simeq} \mathbf{r}_{\Delta \mathtt{R}}(\mathtt{R}_j) + \mathtt{J}_r^{-1}(\mathbf{r}_{\Delta \mathtt{R}}(\mathtt{R}_j))\delta \boldsymbol{\phi}_j
\end{aligned} \tag{79}
$$

$$
\mathbf{r}_{\Delta \mathtt{R}_{ij}}(\delta \mathbf{b}_i^g + \tilde{\delta}\mathbf{b}_i^g) \tag{80}
$$
$$
\begin{aligned}
&= \operatorname{Log}\left(\left(\Delta \tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)\operatorname{Exp}\left(\frac{\partial \Delta \bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}(\delta \mathbf{b}_i^g + \tilde{\delta}\mathbf{b}_i^g)\right)\right)^\mathsf{T}\mathtt{R}_i^\mathsf{T}\mathtt{R}_j\right) \\
&\overset{(7)}{\simeq} \operatorname{Log}\left(\left(\Delta \tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)\ E\ \operatorname{Exp}\left(\mathtt{J}_r^b \frac{\partial \Delta \bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}\tilde{\delta}\mathbf{b}_i^g\right)\right)^\mathsf{T}\mathtt{R}_i^\mathsf{T}\mathtt{R}_j\right) \\
&= \operatorname{Log}\left(\operatorname{Exp}\left(-\mathtt{J}_r^b \frac{\partial \Delta \bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}\tilde{\delta}\mathbf{b}_i^g\right)\left(\Delta \tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)\ E\right)^\mathsf{T}\mathtt{R}_i^\mathsf{T}\mathtt{R}_j\right) \\
&= \operatorname{Log}\left(\operatorname{Exp}\left(-\mathtt{J}_r^b \frac{\partial \Delta \bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}\tilde{\delta}\mathbf{b}_i^g\right)\operatorname{Exp}\left(\mathbf{r}_{\Delta \mathtt{R}_{ij}}(\delta \mathbf{b}_i^g)\right)\right) \\
&\overset{(11)}{=} \operatorname{Log}\left(\operatorname{Exp}\left(\mathbf{r}_{\Delta \mathtt{R}_{ij}}(\delta \mathbf{b}_i^g)\right)\right. \\
&\qquad\qquad \left.\cdot \operatorname{Exp}\left(-\operatorname{Exp}\left(\mathbf{r}_{\Delta \mathtt{R}_{ij}}(\delta \mathbf{b}_i^g)\right)^\mathsf{T}\mathtt{J}_r^b \frac{\partial \Delta \bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}\tilde{\delta}\mathbf{b}_i^g\right)\right) \\
&\overset{(9)}{\simeq} \mathbf{r}_{\Delta \mathtt{R}_{ij}}(\delta \mathbf{b}_i^g) \\
&\qquad - \mathtt{J}_r^{-1}\left(\mathbf{r}_{\Delta \mathtt{R}_{ij}}(\delta \mathbf{b}_i^g)\right)\operatorname{Exp}\left(\mathbf{r}_{\Delta \mathtt{R}_{ij}}(\delta \mathbf{b}_i^g)\right)^\mathsf{T}\mathtt{J}_r^b \frac{\partial \Delta \bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}\ \tilde{\delta}\mathbf{b}_i^g,
\end{aligned}
$$

where we used the shorthands $E \doteq \operatorname{Exp}\left(\frac{\partial \Delta \bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}\delta \mathbf{b}^g\right)$ and $\mathtt{J}_r^b \doteq \mathtt{J}_r\left(\frac{\partial \Delta \bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}\delta \mathbf{b}_i^g\right)$. In summary, the Jacobians of $\mathbf{r}_{\Delta \mathtt{R}_{ij}}$ are:

$$
\begin{array}{ll}
\frac{\partial \mathbf{r}_{\Delta \mathtt{R}_{ij}}}{\partial \delta \boldsymbol{\phi}_i} = -\mathtt{J}_r^{-1}(\mathbf{r}_{\Delta \mathtt{R}}(\mathtt{R}_i))\mathtt{R}_j^\mathsf{T}\mathtt{R}_i & \frac{\partial \mathbf{r}_{\Delta \mathtt{R}_{ij}}}{\partial \delta \mathbf{p}_i} = \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta \mathtt{R}_{ij}}}{\partial \delta \mathbf{v}_i} = \mathbf{0} & \frac{\partial \mathbf{r}_{\Delta \mathtt{R}_{ij}}}{\partial \delta \boldsymbol{\phi}_j} = \mathtt{J}_r^{-1}(\mathbf{r}_{\Delta \mathtt{R}}(\mathtt{R}_j)) \\
\frac{\partial \mathbf{r}_{\Delta \mathtt{R}_{ij}}}{\partial \delta \mathbf{p}_j} = \mathbf{0} & \frac{\partial \mathbf{r}_{\Delta \mathtt{R}_{ij}}}{\partial \delta \mathbf{v}_j} = \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta \mathtt{R}_{ij}}}{\partial \delta \mathbf{b}_i^a} = \mathbf{0} & \frac{\partial \mathbf{r}_{\Delta \mathtt{R}_{ij}}}{\partial \delta \mathbf{b}_i^g} = \alpha
\end{array} \tag{81}
$$

with $\alpha = -\mathtt{J}_r^{-1}\left(\mathbf{r}_{\Delta \mathtt{R}_{ij}}(\delta \mathbf{b}_i^g)\right)\operatorname{Exp}\left(\mathbf{r}_{\Delta \mathtt{R}_{ij}}(\delta \mathbf{b}_i^g)\right)^\mathsf{T}\mathtt{J}_r^b \frac{\partial \Delta \bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}$.

D. Structureless Vision Factors: Null Space Projection

In this section we provide a more efficient implementation of the structureless vision factors, described in Section VII.

Let us consider Eq. (55). Recall that $\mathbf{Q} \doteq (\mathbf{I} - \mathbf{E}_l(\mathbf{E}_l^\mathsf{T}\mathbf{E}_l)^{-1}\mathbf{E}_l^\mathsf{T}) \in \mathbb{R}^{2n_l \times 2n_l}$ is an *orthogonal projector* of $\mathbf{E}_l$, where n_l is the number of cameras observing landmark l. Roughly speaking, $\mathbf{Q}$ projects any vector in $\mathbb{R}^{2n_l}$ to the null space of the matrix $\mathbf{E}_l$. Since $\mathbf{E}_l \in \mathbb{R}^{2n_l \times 3}$ has rank 3, the dimension of its null space is $2n_l - 3$. Any basis $\mathbf{E}_l^\perp \in \mathbb{R}^{2n_l \times 2n_l - 3}$ of the null space of $\mathbf{E}_l$ satisfies the following relation [71]:

$$
\mathbf{E}_l^\perp \left((\mathbf{E}_l^\perp)^\mathsf{T}\mathbf{E}_l^\perp\right)^{-1}(\mathbf{E}_l^\perp)^\mathsf{T} = \mathbf{I} - \mathbf{E}_l(\mathbf{E}_l^\mathsf{T}\mathbf{E}_l)^{-1}\mathbf{E}_l^\mathsf{T}. \tag{82}
$$

A basis for the null space can be easily computed from $\mathbf{E}_l$ using SVD. Such basis is *unitary*, i.e., satisfies $(\mathbf{E}_l^\perp)^\mathsf{T}\mathbf{E}_l^\perp = \mathbf{I}$. Substituting (82) into (55), and recalling that $\mathbf{E}_l^\perp$ is a unitary matrix, we obtain:

$$
\sum_{l=1}^L \|\mathbf{E}_l^\perp (\mathbf{E}_l^\perp)^\mathsf{T}\left(\mathbf{F}_l\ \delta \mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l\right)\|^2 \tag{83}
$$
$$
= \sum_{l=1}^L \left(\mathbf{E}_l^\perp (\mathbf{E}_l^\perp)^\mathsf{T}(\mathbf{F}_l\ \delta \mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l)\right)^\mathsf{T}\left(\mathbf{E}_l^\perp (\mathbf{E}_l^\perp)^\mathsf{T}(\mathbf{F}_l\ \delta \mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l)\right)
$$
$$
= \sum_{l=1}^L \left(\mathbf{F}_l\ \delta \mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l\right)^\mathsf{T}\mathbf{E}_l^\perp \overbrace{(\mathbf{E}_l^\perp)^\mathsf{T}\mathbf{E}_l^\perp}^{=\mathbf{I}_{3\times 3}}(\mathbf{E}_l^\perp)^\mathsf{T}\left(\mathbf{F}_l\ \delta \mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l\right)
$$
$$
= \sum_{l=1}^L \|(\mathbf{E}_l^\perp)^\mathsf{T}\left(\mathbf{F}_l\ \delta \mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l\right)\|^2
$$

which is an alternative representation of the cost function (55). This representation is usually preferable from a computational standpoint, as it does not include matrix inversion and can be computed using a smaller number of matrix multiplications.

E. Rotation Rate Integration Using Euler Angles

In this section, we recall how to integrate rotation rate measurements using the Euler angle parametrization. Let $\tilde{\boldsymbol{\omega}}_k$ be the rotation rate measurement at time k and $\boldsymbol{\eta}_k^g$ be the corresponding noise. Then, given the vector of Euler angles at time k, namely $\boldsymbol{\theta}_k \in \mathbb{R}^3$, we can integrate the rotation rate measurement $\tilde{\boldsymbol{\omega}}_k$ and get $\boldsymbol{\theta}_{k+1}$ as follows:

$$
\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + [E'(\boldsymbol{\theta}_k)]^{-1}(\tilde{\boldsymbol{\omega}}_k - \boldsymbol{\eta}_k^g)\Delta t, \tag{84}
$$

where the matrix $E'(\boldsymbol{\theta}_k)$ is the *conjugate Euler angle rate matrix* [72]. The covariance of $\boldsymbol{\theta}_{k+1}$ can be approximated by a first-order propagation as:

$$
\boldsymbol{\Sigma}_{k+1}^{\text{Euler}} = \mathbf{A}_k \boldsymbol{\Sigma}_k^{\text{Euler}}\mathbf{A}_k^\mathsf{T} + \mathbf{B}_k \boldsymbol{\Sigma}_{\boldsymbol{\eta}}\mathbf{B}_k^\mathsf{T} \tag{85}
$$

where $\mathbf{A}_k \doteq \mathbf{I}_{3\times 3} + \frac{\partial [E'(\boldsymbol{\theta}_k)]^{-1}}{\partial \boldsymbol{\theta}_k}\Delta t$, $\mathbf{B}_k = -[E'(\boldsymbol{\theta}_k)]^{-1}\Delta t$, and $\boldsymbol{\Sigma}_{\boldsymbol{\eta}}$ is the covariance of the measurement noise $\boldsymbol{\eta}_k^{gd}$.

References

[1] A. Martinelli. Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Trans. Robotics*, 28(1):44–60, 2012.

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

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

[4] L. Carlone, Z. Kira, C. Beall, V. Indelman, and F. Dellaert. Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014.

[5] A.I. Mourikis and S.I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3565–3572, April 2007.

[6] 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 (RSS)*, 2015.

[7] Frank Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology, September 2012.

[8] K. J. Wu, A. M. Ahmed, G. A. Georgiou, and S. I. Roumeliotis. A square root inverse filter for efficient vision-aided inertial navigation on mobile devices. In *Robotics: Science and Systems (RSS)*, 2015.

[9] Bradley M. Bell and Frederick W. Cathey. The iterated kalman filter update as a gauss-newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, 1993.

[10] A.J. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Machine Intell.*, 29(6):1052–1067, Jun 2007.

[11] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust visual inertial odometry using a direct EKF-based approach. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2015.

[12] E.S. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *Intl. J. of Robotics Research*, 30(4), Apr 2011.

[13] S. I. Roumeliotis and J. W. Burdick. Stochastic cloning: A generalized framework for processing relative state measurements. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2002.

[14] K. Tsotsos, A. Chiuso, and S. Soatto. Robust inference for visual-inertial sensor fusion. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015.

[15] A. Martinelli. Observability properties and deterministic algorithms in visual-inertial structure from motion. *Foundations and Trends in Robotics*, pages 1–75, 2013.

[16] Dimitrios G. Kottas, Joel A. Hesch, Sean L. Bowman, and Stergios I. Roumeliotis. On the consistency of vision-aided inertial navigation. In *Intl. Sym. on Experimental Robotics (ISER)*, 2012.

[17] G.P. Huang, A.I. Mourikis, and S.I. Roumeliotis. A first-estimates jacobian EKF for improving SLAM consistency. In *Intl. Sym. on Experimental Robotics (ISER)*, 2008.

[18] J.A. Hesch, D.G. Kottas, S.L. Bowman, and S.I. Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *Intl. J. of Robotics Research*, 33(1):182–201, 2014.

[19] J. Hernandez, K. Tsotsos, and S. Soatto. Observability, identifiability and sensitivity of vision-aided inertial navigation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015.

[20] 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 at CVPR*, Anchorage, Alaska, June 2008.

[21] G. Sibley, L. Matthies, and G. Sukhatme. Sliding window filter with application to planetary landing. *J. of Field Robotics*, 27(5):587–608, 2010.

[22] T-C. Dong-Si and A.I. Mourikis. Motion tracking with fixed-lag smoothing: Algorithm consistency and analysis. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.

[23] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart. Keyframe-based visual-inertial slam using nonlinear optimization. In *Robotics: Science and Systems (RSS)*, 2013.

[24] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual-inertial slam using nonlinear optimization. *Intl. J. of Robotics Research*, 2015.

[25] P. Maybeck. *Stochastic Models, Estimation and Control*, volume 1. Academic Press, New York, 1979.

[26] G.P. Huang, A.I. Mourikis, and S.I. Roumeliotis. An observability-constrained sliding window filter for SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 65–72, 2011.

[27] S-H. Jung and C.J. Taylor. Camera trajectory estimation using inertial sensor measurements and structure fom motion results. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[28] D. Sterlow and S. Singh. Motion estimation from image and inertial measurements. 2004.

[29] M. Bryson, M. Johnson-Roberson, and S. Sukkarieh. Airborne smoothing and mapping using vision and inertial sensors. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3143–3148, 2009.

[30] V. Indelman, S. Wiliams, M. Kaess, and F. Dellaert. Information fusion in navigation systems via factor graph based incremental smoothing. *Robotics and Autonomous Systems*, 61(8):721–738, August 2013.

[31] A. Patron-Perez, S. Lovegrove, and G. Sibley. A spline-based trajectory representation for sensor fusion and rolling shutter cameras. *Intl. J. of Computer Vision*,

February 2015.

[32] H. Strasdat, J.M.M. Montiel, and A.J. Davison. Real-time monocular SLAM: Why filter? In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2657–2664, 2010.

[33] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, 2009.

[34] E.D. Nerurkar, K.J. Wu, and S.I. Roumeliotis. C-KLAM: Constrained keyframe-based localization and mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014.

[35] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, pages 225–234, Nara, Japan, Nov 2007.

[36] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378, Dec 2008.

[37] V. Indelman, S. Wiliams, M. Kaess, and F. Dellaert. Factor graph based incremental smoothing in inertial navigation systems. In *Intl. Conf. on Information Fusion, FUSION*, 2012.

[38] V. Indelman, A. Melim, and F. Dellaert. Incremental light bundle adjustment for robotics navigation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, November 2013.

[39] S. Shen. *Autonomous Navigation in Complex Indoor and Outdoor Environments with Micro Aerial Vehicles*. PhD Thesis, University of Pennsylvania, 2014.

[40] N. Keivan, A. Patron-Perez, and G. Sibley. Asynchronous adaptive conditioning for visual-inertial SLAM. In *Intl. Sym. on Experimental Robotics (ISER)*, 2014.

[41] J. Hornegger and C. Tomasi. Representation issues in the ML estimation of camera motion. In *Intl. Conf. on Computer Vision (ICCV)*, pages 640–647, Kerkyra, Greece, September 1999.

[42] M. Moakher. Means and averaging in the group of rotations. *SIAM Journal on Matrix Analysis and Applications*, 24(1):1–16, 2002.

[43] G. S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications (Applied and Numerical Harmonic Analysis)*. Birkhauser, 2012.

[44] Y. Wang and G.S. Chirikjian. Nonparametric second-order theory of error propagation on motion groups. *Intl. J. of Robotics Research*, 27(11–12):1258–1273, 2008.

[45] P.A. Absil, C.G. Baker, and K.A. Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007.

[46] T. D. Barfoot and P. T. Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Trans. Robotics*, 30(3):679–693, 2014.

[47] Y. Wang and G.S. Chirikjian. Error propagation on the euclidean group with applications to manipulator kinematics. *IEEE Trans. Robotics*, 22(4):591–602, 2006.

[48] S. T. Smith. Optimization techniques on Riemannian manifolds. *Hamiltonian and Gradient Flows, Algorithms and Control, Fields Inst. Commun., Amer. Math. Soc.*, 3: 113–136, 1994.

[49] J.A. Farrell. *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill, 2008.

[50] J. H. Manton. Optimization algorithms exploiting unitary constraints. *IEEE Trans. Signal Processing*, 50:63–650, 2002.

[51] F.R. Kschischang, B.J. Frey, and H-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2), February 2001.

[52] F. Dellaert. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. Technical Report GIT-GVU-05-11, Georgia Institute of Technology, 2005.

[53] R.M. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[54] P. E. Crouch and R. Grossman. Numerical integration of ordinary differential equations on manifolds. *J. Nonlinear Sci.*, 3:1–22, 1993.

[55] H. Munthe-Kaas. Higher order runge-kutta methods on manifolds. *Appl. Numer. Math.*, 29(1):115–127, 1999.

[56] J. Park and W.-K. Chung. Geometric integration on euclidean group with application to articulated multibody systems. *IEEE Trans. Robotics*, 2005.

[57] M. S. Andrle and J. L. Crassidis. Geometric integration of quaternions. *Journal of Guidance, Control, and Dynamics*, 36(6):1762–1757, 2013.

[58] J.L. Crassidis. Sigma-point Kalman filtering for integrated GPS and inertial navigation. *IEEE Trans. Aerosp. Electron. Syst.*, 42(2):750–756, 2006.

[59] P. Furgale, J. Rehder, and R. Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[60] M. Li and A.I. Mourikis. Online temporal calibration for camera-imu systems: Theory and algorithms. *Intl. J. of Robotics Research*, 33(6), 2014.

[61] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[62] M. Kaess, V. Ila, R. Roberts, and F. Dellaert. The Bayes tree: Enabling incremental reordering and fluid relinearization for online mapping. Technical Report MIT-CSAIL-TR-2010-021, Computer Science and Artificial Intelligence Laboratory, MIT, Jan 2010.

[63] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications To Tracking and Navigation*. John Wiley and Sons, 2001.

[64] J. Hornegger. Statistical modeling of relations for 3-D object recognition. In *Intl. Conf. Acoust., Speech, and Signal Proc. (ICASSP)*, volume 4, pages 3173–3176, Munich, April 1997.

[65] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast Semi-Direct Monocular Visual Odometry. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014. doi: 10.1109/ICRA.2014.6906584.

[66] J. Nikolic, J., Rehderand M. Burri, P. Gohl, S. Leutenegger, P. Furgale, and R. Siegwart. A Synchronized Visual-

Inertial Sensor System with FPGA Pre-Processing for Accurate Real-Time SLAM. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014.

[67] F.C. Park and B.J. Martin. Robot sensor calibration: Solving AX=XB on the euclidean group. 10(5), 1994.

[68] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, Providence, USA, June 2012.

[69] Tango IMU specifications. URL http://ae-bst. resource.bosch.com/media/products/dokumente/bmx055/ BST-BMX055-FL000-00_2013-05-07_onl.pdf.

[70] Adis IMU specifications. URL http://www.analog. com/media/en/technical-documentation/data-sheets/ ADIS16448.pdf.

[71] C.D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

[72] J. Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. Technical report, Stanford University, 2006.

Christian Forster (1986, Swiss) obtained his Ph.D. in Computer Science (2016) at the University of Zurich under the supervision of Davide Scaramuzza. Previously, he received a B.Sc. degree in Mechanical Engineering (2009) and a M.Sc. degree in Robotics, Systems and Control (2012) at ETH Zurich, Switzerland. In 2011, he was a visiting researcher at CSIR (South Africa), and in 2014 at Georgia Tech in the group of Frank Dellaert. He is broadly interested in developing real-time computer vision algorithms that enable robots to perceive the three dimensional environment.

Luca Carlone is a research scientist in the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. Before joining MIT, he was a postdoctoral fellow at Georgia Tech (2013-2015), a visiting researcher at the University of California Santa Barbara (2011), and a visiting researcher at the University of Zaragoza (2010). He got his Ph.D. from Politecnico di Torino, Italy, in 2012. His research interests include nonlinear estimation, optimization, and control applied to robotics, with special focus on localization, mapping, and decision making for navigation in single and multi robot systems.

Frank Dellaert is Professor in the School of Interactive Computing at Georgia Tech. He graduated from Carnegie Mellon in 2001 with a Ph.D. in Computer Science, after obtaining degrees from Case Western Reserve University and K.U. Leuven before that. Professor Dellaert's research focuses on large-scale inference for autonomous robot systems, on land, air, and in water. He pioneered the use of several probabilistic methods in both computer vision and robotics. With Dieter Fox and Sebastian Thrun, he has introduced the Monte Carlo localization method for estimating and tracking the pose of robots, which is now a standard and popular tool in mobile robotics. Most recently, he has investigated 3D reconstruction in large-scale environments by taking a graph-theoretic view, and pioneered the use of sparse factor graphs as a representation in robot navigation and mapping. He and his group released an open-source large-scale optimization toolbox, GTSAM.

Davide Scaramuzza (1980, Italian) is Professor of Robotics at the University of Zurich, where he does research at the intersection of robotics, computer vision, and neuroscience. He did his PhD in robotics and computer Vision at ETH Zurich and a postdoc at the University of Pennsylvania. From 2009 to 2012, he led the European project sFly, which introduced the worlds first autonomous navigation of micro drones in GPS-denied environments using visual-inertial sensors as the only sensor modality. For his research contributions, he was awarded an SNSF-ERC Starting Grant, the IEEE Robotics and Automation Early Career Award, and a Google Research Award. He coauthored the book Introduction to Autonomous Mobile Robots (published by MIT Press).