IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation

Christian Forster*, Luca Carlone†, Frank Dellaert†, and Davide Scaramuzza*
*Robotics and Perception Group, University of Zurich, Switzerland. Email: {forster,sdavide}@ifi.uzh.ch
†School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA.
Email: luca.carlone@gatech.edu, frank@cc.gatech.edu

***Abstract*—Recent results in monocular visual-inertial navigation (VIN) have shown that optimization-based approaches outperform filtering methods in terms of accuracy due to their capability to relinearize past states. However, the improvement comes at the cost of increased computational complexity. In this paper, we address this issue by preintegrating inertial measurements between selected keyframes. The preintegration allows us to accurately summarize hundreds of inertial measurements into a single relative motion constraint. Our first contribution is a *preintegration theory* that properly addresses the manifold structure of the rotation group and carefully deals with uncertainty propagation. The measurements are integrated in a *local frame*, which eliminates the need to repeat the integration when the linearization point changes while leaving the opportunity for belated bias corrections. The second contribution is to show that the preintegrated IMU model can be seamlessly integrated in a visual-inertial pipeline under the unifying framework of factor graphs. This enables the use of a *structureless* model for visual measurements, further accelerating the computation. The third contribution is an extensive evaluation of our monocular VIN pipeline: experimental results confirm that our system is very fast and demonstrates superior accuracy with respect to competitive state-of-the-art filtering and optimization algorithms, including off-the-shelf systems such as *Google Tango* [1].**

I. Introduction

The fusion of cameras and inertial sensors for three-dimensional structure and motion estimation has received considerable interest in 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 of monocular vision and gravity observable [2] and provides robust and accurate inter-frame motion estimates. Applications of VIN range from autonomous navigation in GPS-denied environments, to 3D reconstruction, and augmented reality.

Although impressive results have been achieved in VIN, state-of-the-art algorithms require trading-off computational efficiency with accuracy. Batch non-linear optimization, which has become popular for visual-inertial fusion [3–15], allows one to compute an optimal estimate; 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* [9, 16–18] or to run the optimization in a parallel thread, using a tracking and mapping dual architecture [5, 19]. Another approach is to maintain a local map of fixed size and to marginalize old states [6, 7, 9], which is also termed *fixed-lag smoothing*. To that extreme, if only the latest sensor state is maintained, we speak of *filtering*, which amounts the vast majority of related work in VIN [20, 21]. Although filtering and fixed-lag smoothing enable fast computation, they commit to a linearization point when marginalizing; the gradual build-up of linearization errors leads to drift and possible inconsistencies [22]. A breakthrough in the direction of reconciling filtering and batch optimization has been the development of *incremental smoothing techniques* (iSAM [23], iSAM2 [24]), which leverage the expressiveness of *factor graphs* to identify and update only the typically small subset of variables affected by a new measurement. Although this results in constant update time in odometry problems, previous VIN applications still work at low frame rates [25].

In this work, we present a system that uses incremental smoothing for fast computation of the optimal *maximum a posteriori* (MAP) estimate. The first step towards this goal is the development of a novel preintegration theory. The use of *preintegrated IMU measurements* was first proposed in [26] and consists of combining many inertial measurements

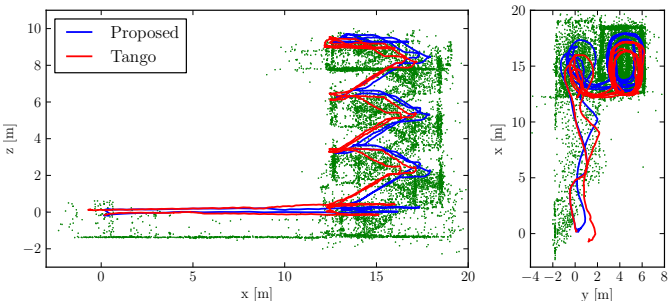

Fig. 1: Real test comparing the proposed VIN approach against Google Tango. The 160m-long trajectory starts at $(0, 0, 0)$ (ground floor), goes up till the 3rd floor of a building, and returns to the initial point. The figure shows a side view (left) and a top view (right) of the trajectory estimates 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.

This research was partially funded by the Swiss National Foundation (project number 200021-143607, "Swarm of Flying Cameras"), the National Center of Competence in Research Robotics (NCCR), the UZH Forschungskredit, the NSF Award 11115678, and the USDA NIFA Award GEOW-2014-09160.

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 and allows us to analytically derive all Jacobians. This is in contrast to [26], which 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 [27]. Moreover, Euler angles are known to have singularities. Our theoretical derivation in Section V also advances previous works [10, 12, 13, 25] that used preintegrated measurements but did not develop the corresponding theory for uncertainty propagation and a-posteriori bias correction. Besides these improvements, our model still benefits from the pioneering insight of [26]: the integration is performed in a *local frame*, which does not require to repeat the integration when the linearization point changes.

As a second contribution, we frame our preintegrated IMU model into a factor graph perspective. This enables the design of a constant-time VIN pipeline based on iSAM2 [24]. Our incremental-smoothing solution avoids the accumulation of linearization errors and provides an appealing alternative to using an adaptive support window for optimization [10].

Inspired by [20, 28], we adopt a *structureless* model for visual measurements, which allows one to eliminate large numbers of variables (*i.e.*, all 3D points) during incremental smoothing, further accelerating the computation.

The third contribution is an efficient implementation and extensive evaluation of our system. Experimental results highlight that our back-end requires an average CPU time of 10ms to compute the full MAP estimate and achieves superior accuracy with respect to competitive state-of-the-art approaches. The paper is accompanied by supplementary material [29] that reports extra details of our derivation. Furthermore, we release our implementation of the IMU preintegration and the structurless vision factors in the GTSAM 4.0 optimization toolbox [30]. A video showing an example of the execution of our system is available at https://youtu.be/CsJkci5lfco

II. Preliminaries

A. Notions of Riemannian geometry

This section recalls useful concepts related to the Special Orthogonal Group $\mathrm{SO}(3)$ and the Special Euclidean Group $\mathrm{SE}(3)$. Our presentation is based on [31, 32].

a) Special Orthogonal Group: $\mathrm{SO}(3)$ describes the group of 3D rotation matrices and it is formally defined as $\mathrm{SO}(3) \doteq \{\mathtt{R} \in \mathbb{R}^{3\times 3} : \mathtt{R}^\mathsf{T}\mathtt{R} = \mathbf{I}, \det(\mathtt{R}) = 1\}$. The group operation is the usual matrix multiplication, and the inverse is the matrix transpose. The group $\mathrm{SO}(3)$ also forms a smooth manifold. The tangent space to the manifold (at the identity) is denoted $\mathfrak{so}(3)$, 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

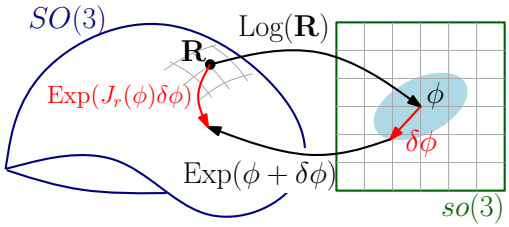

Fig. 2: The right Jacobian $\mathtt{J}_r$ relates an additive perturbation $\delta\boldsymbol{\phi}$ in the tangent space to a multiplicative perturbation on the manifold $\mathrm{SO}(3)$, as per Eq. (7).

in $\mathbb{R}^3$ using the *hat* operator:

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

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 $\mathtt{S} = \boldsymbol{\omega}^\wedge$, the vee operator is such that $\mathtt{S}^\vee = \boldsymbol{\omega}$. A property of skew symmetric matrices that will be useful later on is:

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

The *exponential map* (at the identity) $\exp : \mathfrak{so}(3) \to \mathrm{SO}(3)$ associates an element of the Lie Algebra to a rotation:

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

A first-order approximation of the exponential map is:

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

The *logarithm map* (at the identity) associates a matrix $\mathtt{R}$ in $\mathrm{SO}(3)$ to a skew symmetric matrix:

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

Note that $\log(\mathtt{R})^\vee = \mathbf{a}\varphi$, where $\mathbf{a}$ and φ are the rotation axis and the rotation angle of $\mathtt{R}$, respectively.

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 $\mathtt{R}$.

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

$$\begin{array}{rccccccc}\mathrm{Exp}: & \mathbb{R}^3 & \ni & \boldsymbol{\phi} & \to & \exp(\boldsymbol{\phi}^\wedge) & \in & \mathrm{SO}(3),\\ \mathrm{Log}: & \mathrm{SO}(3) & \ni & \mathtt{R} & \to & \log(\mathtt{R})^\vee & \in & \mathbb{R}^3,\end{array} \tag{6}$$

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

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

$$\mathrm{Exp}(\boldsymbol{\phi}+\delta\boldsymbol{\phi}) \approx \mathrm{Exp}(\boldsymbol{\phi})\,\mathrm{Exp}(\mathtt{J}_r(\boldsymbol{\phi})\delta\boldsymbol{\phi}). \tag{7}$$

The term $\mathtt{J}_r(\boldsymbol{\phi})$ is the *right Jacobian* of $\mathrm{SO}(3)$ [31, p.40] and relates additive increments in the tangent space to multiplicative increments applied on the right-hand-side (Fig. 2):

$$\mathtt{J}_r(\boldsymbol{\phi}) = \mathbf{I} - \tfrac{1-\cos(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}\|^2}\boldsymbol{\phi}^\wedge + \tfrac{\|\boldsymbol{\phi}\|-\sin(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}^3\|}(\boldsymbol{\phi}^\wedge)^2. \tag{8}$$

A similar first-order approximation holds for the logarithm:

$$\mathrm{Log}\big(\,\mathrm{Exp}(\boldsymbol{\phi})\,\mathrm{Exp}(\delta\boldsymbol{\phi})\,\big) \approx \boldsymbol{\phi} + \mathtt{J}_r^{-1}(\boldsymbol{\phi})\delta\boldsymbol{\phi}. \tag{9}$$

An explicit expression for the inverse of the right Jacobian is given in the supplementary material [29]. $\mathtt{J}_r(\boldsymbol{\phi})$ reduces to the identity for $\|\boldsymbol{\phi}\|=0$.

Another useful property of the exponential map that follows directly from the *Adjoint* representation is:

$$\mathtt{R}\ \mathrm{Exp}(\boldsymbol{\phi})\ \mathtt{R}^{\mathsf{T}} = \exp(\mathtt{R}\boldsymbol{\phi}^{\wedge}\mathtt{R}^{\mathsf{T}}) = \mathrm{Exp}(\mathtt{R}\boldsymbol{\phi}) \tag{10}$$

$$\Leftrightarrow \quad \mathrm{Exp}(\boldsymbol{\phi})\ \mathtt{R} = \mathtt{R}\ \mathrm{Exp}(\mathtt{R}^{\mathsf{T}}\boldsymbol{\phi}) \tag{11}$$

b) Special Euclidean Group: $\mathrm{SE}(3)$ describes the group of rigid motion in 3D and it is defined as $\mathrm{SE}(3) \doteq \{(\mathtt{R},\mathbf{p}) : \mathtt{R}\in\mathrm{SO}(3), \mathbf{p}\in\mathbb{R}^3\}$. The group operation is $\mathtt{T}_1\cdot\mathtt{T}_2 = (\mathtt{R}_1\mathtt{R}_2\ ,\ \mathtt{R}_1\mathbf{p}_2+\mathbf{p}_1)$, and the inverse is $\mathtt{T}_1^{-1} = (\mathtt{R}_1^{\mathsf{T}}\ ,\ -\mathtt{R}_1^{\mathsf{T}}\mathbf{p}_1)$. The *exponential map* and the *logarithm map* for $\mathrm{SE}(3)$ are defined in [32]. However, these are not needed in this paper for reasons that will be clear in Section II-C.

B. Uncertainty Description in $\mathrm{SO}(3)$

A fairly natural definition of uncertainty in $\mathrm{SO}(3)$ is to define a distribution in the tangent space, and then map it to $\mathrm{SO}(3)$ via the exponential map (6) [32–34]:

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

where $\mathtt{R}$ is a given noise-free rotation (the *mean*) and ϵ is a small normally distributed perturbation with zero mean.

The distribution of the random variable $\tilde{\mathtt{R}}\in\mathrm{SO}(3)$ can be computed explicitly, as shown in [33], leading to:

$$p(\tilde{\mathtt{R}}) = \beta(\tilde{\mathtt{R}})\ e^{-\frac{1}{2}\left\|\mathrm{Log}(\mathtt{R}^{-1}\tilde{\mathtt{R}})\right\|_{\Sigma}^2} \tag{13}$$

where $\beta(\tilde{\mathtt{R}})$ is a normalization factor that can be safely approximated as $\beta(\tilde{\mathtt{R}})\simeq 1/\sqrt{2\pi\det(\Sigma)}$ when Σ is small. If we approximate β as a constant, the negative log-likelihood of a rotation $\mathtt{R}$ given a measurement $\tilde{\mathtt{R}}$ distributed as in (13) is:

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

C. Gauss-Newton Method on Manifold

Let us consider the optimization problem $\min_{x\in\mathcal{M}} f(x)$, where $f(\cdot)$ is a given cost function, and the variable x belongs to a manifold $\mathcal{M}$; for simplicity we consider a single variable, while the description can be easily generalized.

A standard approach for optimization on manifold [35, 36], 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:

$$\min_{x\in\mathcal{M}} f(x) \quad \Rightarrow \quad \min_{\delta x\in\mathbb{R}^n} f(\mathcal{R}_x(\delta x)) \tag{14}$$

The re-parametrization is usually called *lifting* [35]. Roughly speaking, we work in the tangent space defined at the current estimate, which locally behaves as an Euclidean space. We can now apply standard optimization techniques to the problem on the right in (14). In the GN 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 $\hat{x}\leftarrow\mathcal{R}_{\hat{x}}(\delta x^{\star})$.

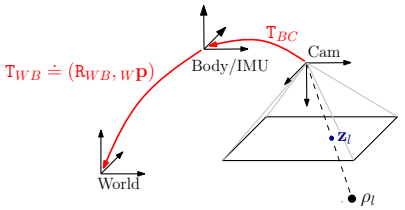

Fig. 3: $\mathtt{T}_{\mathrm{WB}} \doteq (\mathtt{R}_{\mathrm{WB}}, {}_{\mathrm{W}}\mathbf{p})$ is the pose of the body frame B w.r.t. the world frame W. We assume that the body frame coincides with the IMU frame. $\mathtt{T}_{\mathrm{BC}}$ is the pose of the camera in the body frame, known from prior calibration.

A possible retraction is the exponential map. However, computationally, this may be not the most convenient, see [37].

For $\mathrm{SE}(3)$, we use the following retraction at $\mathtt{T}\doteq(\mathtt{R},\mathbf{p})$:

$$\mathcal{R}_{\mathtt{T}}(\delta\boldsymbol{\phi},\delta\mathbf{p}) = (\mathtt{R}\ \mathrm{Exp}(\delta\boldsymbol{\phi}),\ \ \mathbf{p}+\mathtt{R}\ \delta\mathbf{p}), \qquad [\delta\boldsymbol{\phi}\ \ \ \delta\mathbf{p}]\in\mathbb{R}^6 \tag{15}$$

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

III. Maximum a Posteriori Visual-Inertial State Estimation

System and assumptions. We consider a VIN problem in which we want to track the state of a *sensing system* (e.g., a mobile robot 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. 3). Furthermore, we assume that a *front-end* provides pixel measurements of 3D landmarks at unknown position. The front-end also selects a subset of images, called *keyframes* [16], for which we want to compute a pose estimate. Section VII discusses implementation aspects, including the choice of the front-end in our experiments.

The state. The state of the system at time i is described by the IMU orientation, position, velocity and biases: $\mathbf{x}_i \doteq [\mathtt{R}_i, \mathbf{p}_i, \mathbf{v}_i, \mathbf{b}_i]$. Recall that $(\mathtt{R}_i,\mathbf{p}_i)\in\mathrm{SE}(3)$, 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:

$$\mathcal{X}_k \doteq \{\mathbf{x}_i\}_{i\in\mathcal{K}_k}. \tag{16}$$

We adopt a structureless approach (*cf.*, Section VI), hence the 3D landmarks are not part of the variables to be estimated.

The measurements. The input to our estimation problem are the measurements from the camera and the IMU. We denote with $\mathcal{C}_i$ the camera measurements at keyframe i. At time i, the camera can observe multiple landmarks l, hence $\mathcal{C}_i$ contains multiple pixel observations $\mathbf{z}_{il}$. 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. Usually, each set $\mathcal{I}_{ij}$ contains hundreds of IMU measurements.

The set of measurements collected up to time k is

$$\mathcal{Z}_k \doteq \{\mathcal{C}_i, \mathcal{I}_{ij}\}_{(i,j)\in\mathcal{K}_k}, \tag{17}$$

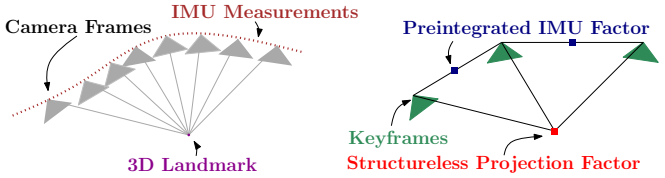

Fig. 4: Left: visual and inertial measurements in VIN. Right: factor graph in which several IMU measurements are summarized in a single preintegrated IMU factor and a structureless vision factor constraints keyframes observing the same landmark.

Factor graphs and MAP estimation. A *factor graph* encodes the posterior probability of the variables $\mathcal{X}_k$, given the available measurements $\mathcal{Z}_k$ and priors $p(\mathcal{X}_0)$:

$$p(\mathcal{X}_k|\mathcal{Z}_k) \propto p(\mathcal{X}_0)p(\mathcal{Z}_k|\mathcal{X}_k) = p(\mathcal{X}_0) \prod_{(i,j)\in\mathcal{K}_k} p(\mathcal{C}_i, \mathcal{I}_{ij}|\mathcal{X}_k)$$
$$= p(\mathcal{X}_0) \prod_{(i,j)\in\mathcal{K}_k} p(\mathcal{I}_{ij}|\mathbf{x}_i, \mathbf{x}_j) \prod_{i\in\mathcal{K}_k} \prod_{l\in\mathcal{C}_i} p(\mathbf{z}_{il}|\mathbf{x}_i). \quad (18)$$

The terms in the factorization (18) are called *factors*. A schematic representation of the connectivity of the factor graph underlying the problem is given in Fig. 4 (the connectivity of the structureless vision factors will be clarified in Section VI).

The MAP estimate corresponds to the maximum of (18), or equivalently, the minimum of the negative log-posterior. Under the assumption of zero-mean Gaussian noise, the negative log-posterior can be written a sum of squared residual errors:

$$\mathcal{X}_k^\star \doteq \arg\min_{\mathcal{X}_k} -\log_e p(\mathcal{X}_k|\mathcal{Z}_k) \quad (19)$$
$$= \arg\min_{\mathcal{X}_k} \|\mathbf{r}_0\|^2_{\mathbf{\Sigma}_0} + \sum_{(i,j)\in\mathcal{K}_k} \|\mathbf{r}_{\mathcal{I}_{ij}}\|^2_{\mathbf{\Sigma}_{ij}} + \sum_{i\in\mathcal{K}_k}\sum_{l\in\mathcal{C}_i} \|\mathbf{r}_{\mathcal{C}_{il}}\|^2_{\mathbf{\Sigma}_\mathcal{C}}$$

where $\mathbf{r}_0$, $\mathbf{r}_{\mathcal{I}_{ij}}$, $\mathbf{r}_{\mathcal{C}_{il}}$ are the residual errors associated to the measurements, and $\mathbf{\Sigma}_0$, $\mathbf{\Sigma}_{ij}$, and $\mathbf{\Sigma}_\mathcal{C}$ are the corresponding covariance matrices. The goal of the following sections is to provide expressions for the residual errors.

IV. IMU Model and Motion Integration

An IMU measures the rotation rate and the acceleration of the sensor with respect to an inertial frame. The measurements, namely ${}_\mathrm{B}\tilde{\mathbf{a}}(t)$, and ${}_\mathrm{B}\tilde{\boldsymbol{\omega}}_{\mathrm{WB}}(t)$, are affected by additive white noise $\boldsymbol{\eta}$ and a slowly varying sensor bias $\mathbf{b}$:

$${}_\mathrm{B}\tilde{\boldsymbol{\omega}}_{\mathrm{WB}}(t) = {}_\mathrm{B}\boldsymbol{\omega}_{\mathrm{WB}}(t) + \mathbf{b}^g(t) + \boldsymbol{\eta}^g(t) \quad (20)$$
$${}_\mathrm{B}\tilde{\mathbf{a}}(t) = \mathtt{R}^\mathsf{T}_{\mathrm{WB}}(t)\,({}_\mathrm{W}\mathbf{a}(t) - {}_\mathrm{W}\mathbf{g}) + \mathbf{b}^a(t) + \boldsymbol{\eta}^a(t), \quad (21)$$

In our notation, the prefix B denotes that the corresponding quantity is expressed in the frame B (*c.f.,* Fig. 3). The pose of the IMU is described by the transformation $\{\mathtt{R}_{\mathrm{WB}}, {}_\mathrm{W}\mathbf{p}\}$, which maps a point from sensor frame B to W. The vector ${}_\mathrm{B}\boldsymbol{\omega}_{\mathrm{WB}}(t) \in \mathbb{R}^3$ is the instantaneous angular velocity of B relative to W expressed in coordinate frame B, while ${}_\mathrm{W}\mathbf{a}(t) \in \mathbb{R}^3$ is the acceleration of the sensor; ${}_\mathrm{W}\mathbf{g}$ 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 [38, 39]:

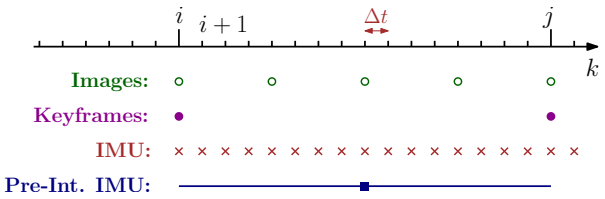

Fig. 5: Different rates for IMU and camera.

$$\dot{\mathtt{R}}_{\mathrm{WB}} = \mathtt{R}_{\mathrm{WB}}\,{}_\mathrm{B}\boldsymbol{\omega}^\wedge_{\mathrm{WB}}, \qquad {}_\mathrm{W}\dot{\mathbf{v}} = {}_\mathrm{W}\mathbf{a}, \qquad {}_\mathrm{W}\dot{\mathbf{p}} = {}_\mathrm{W}\mathbf{v}, \quad (22)$$

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

The state at time $t+\Delta t$ is obtained by integrating Eq. (22). Applying Euler integration, which is exact assuming that ${}_\mathrm{W}\mathbf{a}$ and ${}_\mathrm{B}\boldsymbol{\omega}_{\mathrm{WB}}$ remain constant in the interval $[t, t+\Delta t]$, we get:

$$\mathtt{R}_{\mathrm{WB}}(t+\Delta t) = \mathtt{R}_{\mathrm{WB}}(t)\,\mathrm{Exp}\left({}_\mathrm{B}\boldsymbol{\omega}_{\mathrm{WB}}(t)\Delta t\right)$$
$${}_\mathrm{W}\mathbf{v}(t+\Delta t) = {}_\mathrm{W}\mathbf{v}(t) + {}_\mathrm{W}\mathbf{a}(t)\Delta t$$
$${}_\mathrm{W}\mathbf{p}(t+\Delta t) = {}_\mathrm{W}\mathbf{p}(t) + {}_\mathrm{W}\mathbf{v}(t)\Delta t + \frac{1}{2}{}_\mathrm{W}\mathbf{a}(t)\Delta t^2. \quad (23)$$

Using Eqs. (20)–(21), we can compute ${}_\mathrm{W}\mathbf{a}$ and ${}_\mathrm{B}\boldsymbol{\omega}_{\mathrm{WB}}$ as a function of the IMU measurements, hence (23) becomes

$$\begin{aligned}
\mathtt{R}(t+\Delta t) &= \mathtt{R}(t)\,\mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}(t) - \mathbf{b}^g(t) - \boldsymbol{\eta}^{gd}(t)\right)\Delta t\right)\\
\mathbf{v}(t+\Delta t) &= \mathbf{v}(t) + \mathbf{g}\Delta t + \mathtt{R}(t)\left(\tilde{\mathbf{a}}(t) - \mathbf{b}^a(t) - \boldsymbol{\eta}^{ad}(t)\right)\Delta t\\
\mathbf{p}(t+\Delta t) &= \mathbf{p}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{g}\Delta t^2\\
&\quad + \frac{1}{2}\mathtt{R}(t)\left(\tilde{\mathbf{a}}(t) - \mathbf{b}^a(t) - \boldsymbol{\eta}^{ad}(t)\right)\Delta t^2, \quad (24)
\end{aligned}$$

where we dropped the coordinate frame subscripts for readability (the notation should be unambiguous from now on). 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 $\mathrm{Cov}(\boldsymbol{\eta}^{gd}(t)) = \frac{1}{\Delta t}\mathrm{Cov}(\boldsymbol{\eta}^g(t))$, and the same relation holds for $\boldsymbol{\eta}^{ad}$ (*cf.*, [40, Appendix]).

V. IMU Preintegration on Manifold

This section contains the first key contribution of the paper. While Eq. (24) 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. Here we show that all measurements between two keyframes at times $k = i$ and $k = j$ 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 [26] using Euler angles and we extend it, by developing a suitable theory for preintegration on manifold.

We assume that the IMU is synchronized with the camera and provides measurements at discrete times k (*cf.*, Fig. 5).[1]

[1]We calibrate the IMU-camera delay using the *Kalibr* toolbox [41]. An alternative is to add the delay as a state in the estimation process [42].

Iterating the IMU integration (24) for all Δt intervals between $k=i$ and $k=j$ (*c.f.*, Fig. 5), we find:

$$\begin{aligned}
\mathtt{R}_j &= \mathtt{R}_i \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_k^g - \boldsymbol{\eta}_k^{gd}\right)\Delta t\right), \\
\mathbf{v}_j &= \mathbf{v}_i + \mathbf{g}\Delta t_{ij} + \sum_{k=i}^{j-1} \mathtt{R}_k\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t \qquad (25)\\
\mathbf{p}_j &= \mathbf{p}_i + \sum_{k=i}^{j-1} \mathbf{v}_k\Delta t + \frac{1}{2}\mathbf{g}\Delta t_{ij}^2 + \frac{1}{2}\sum_{k=i}^{j-1} \mathtt{R}_k\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t^2
\end{aligned}$$

where we introduced the shorthands $\Delta t_{ij} \doteq \sum_{k=i}^{j}\Delta t$ and $(\cdot)_i \doteq (\cdot)(t_i)$ for readability.

While Eq. (25) already provides an estimate of the motion between time t_i and t_j, it has the drawback that the integration in (25) has to be repeated whenever the linearization point at time t_i changes (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 (25)).

Our goal is to avoid repeated integrations. For this purpose, we define the following relative motion increments that are independent of the pose and velocity at t_i:

$$\begin{aligned}
\Delta\mathtt{R}_{ij} &\doteq \mathtt{R}_i^\mathsf{T}\mathtt{R}_j = \textstyle\prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_k^g - \boldsymbol{\eta}_k^{gd}\right)\Delta t\right)\\
\Delta\mathbf{v}_{ij} &\doteq \mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) = \textstyle\sum_{k=i}^{j-1} \Delta\mathtt{R}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t\\
\Delta\mathbf{p}_{ij} &\doteq \mathtt{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \tfrac{1}{2}\mathbf{g}\Delta t_{ij}^2\right)\\
&= \textstyle\sum_{k=i}^{j-1}\left[\Delta\mathbf{v}_{ik}\Delta t + \frac{1}{2}\Delta\mathtt{R}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t^2\right]\\
&= \textstyle\sum_{k=i}^{j-1}\left[\frac{3}{2}\Delta\mathtt{R}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t^2\right] \qquad (26)
\end{aligned}$$

where we defined $\Delta\mathtt{R}_{ik} \doteq \mathtt{R}_i^\mathsf{T}\mathtt{R}_k$ and $\Delta\mathbf{v}_{ik} \doteq \mathbf{v}_k - \mathbf{v}_i$.

Unfortunately, summations and products in (26) are still function of the bias. We tackle this problem in two steps. In Section V-A, we assume $\mathbf{b}_i$ is known; then, in Section V-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:

$$\mathbf{b}_i^g = \mathbf{b}_{i+1}^g = \ldots = \mathbf{b}_{j-1}^g, \quad \mathbf{b}_i^a = \mathbf{b}_{i+1}^a = \ldots = \mathbf{b}_{j-1}^a. \quad (27)$$

A. Preintegrated IMU Measurements

In this section, we assume that the bias at time t_i is known. We want to isolate the noise in (26). Therefore, starting with the rotation increment $\Delta\mathtt{R}_{ij}$, 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):

$$\begin{aligned}
\Delta\mathtt{R}_{ij} &\overset{\text{eq.(7)}}{\simeq} \prod_{k=i}^{j-1}\left[\mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_i^g\right)\Delta t\right)\mathrm{Exp}\left(-\mathtt{J}_r^k\,\boldsymbol{\eta}_k^{gd}\,\Delta t\right)\right]\\
&\overset{\text{eq.(11)}}{=} \Delta\tilde{\mathtt{R}}_{ij}\prod_{k=i}^{j-1}\mathrm{Exp}\left(-\Delta\tilde{\mathtt{R}}_{k+1j}^\mathsf{T}\,\mathtt{J}_r^k\,\boldsymbol{\eta}_k^{gd}\,\Delta t\right)\\
&\doteq \Delta\tilde{\mathtt{R}}_{ij}\mathrm{Exp}\left(-\delta\boldsymbol{\phi}_{ij}\right) \qquad (28)
\end{aligned}$$

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

Substituting (28) back into the expression of $\Delta\mathbf{v}_{ij}$ in (26), using the approximation (4) for $\mathrm{Exp}\left(-\delta\boldsymbol{\phi}_{ij}\right)$, and dropping higher-order noise terms, we obtain:

$$\begin{aligned}
\Delta\mathbf{v}_{ij} &\overset{\text{eq.(4)}}{\simeq} \sum_{k=i}^{j-1}\Delta\tilde{\mathtt{R}}_{ik}(\mathbf{I} - \delta\boldsymbol{\phi}_{ik}^\wedge)\left(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a\right)\Delta t - \Delta\tilde{\mathtt{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t\\
&\overset{\text{eq.(2)}}{=} \Delta\tilde{\mathbf{v}}_{ij} + \sum_{k=i}^{j-1}\left[\Delta\tilde{\mathtt{R}}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a\right)^\wedge\delta\boldsymbol{\phi}_{ik}\Delta t - \Delta\tilde{\mathtt{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t\right]\\
&\doteq \Delta\tilde{\mathbf{v}}_{ij} - \delta\mathbf{v}_{ij} \qquad (29)
\end{aligned}$$

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 (28) in the expression of $\Delta\mathbf{p}_{ij}$ in (26), and using the first-order approximation (4), we obtain:

$$\begin{aligned}
\Delta\mathbf{p}_{ij} &\overset{\text{eq.(4)}}{\simeq} \sum_{k=i}^{j-1}\frac{3}{2}\Delta\tilde{\mathtt{R}}_{ik}(\mathbf{I} - \delta\boldsymbol{\phi}_{ik}^\wedge)\left(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a\right)\Delta t^2 - \sum_{k=i}^{j-1}\frac{3}{2}\Delta\tilde{\mathtt{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t^2\\
&\overset{\text{eq.(2)}}{=} \Delta\tilde{\mathbf{p}}_{ij} + \sum_{k=i}^{j-1}\left[\frac{3}{2}\Delta\tilde{\mathtt{R}}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a\right)^\wedge\delta\boldsymbol{\phi}_{ik}\Delta t^2 - \frac{3}{2}\Delta\tilde{\mathtt{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t^2\right]\\
&\doteq \Delta\tilde{\mathbf{p}}_{ij} - \delta\mathbf{p}_{ij} \qquad (30)
\end{aligned}$$

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

Substituting the expressions (28), (29), (30) back in the original definition of $\Delta\mathtt{R}_{ij}, \Delta\mathbf{v}_{ij}, \Delta\mathbf{p}_{ij}$ in (26), we finally get our *preintegrated measurement model*:

$$\begin{aligned}
\Delta\tilde{\mathtt{R}}_{ij} &= \mathtt{R}_i^\mathsf{T}\mathtt{R}_j\mathrm{Exp}\left(\delta\boldsymbol{\phi}_{ij}\right)\\
\Delta\tilde{\mathbf{v}}_{ij} &= \mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) + \delta\mathbf{v}_{ij}\\
\Delta\tilde{\mathbf{p}}_{ij} &= \mathtt{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) + \delta\mathbf{p}_{ij} \qquad (31)
\end{aligned}$$

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

B. Noise Propagation

Let us start with the rotation noise:

$$\mathrm{Exp}\left(-\delta\boldsymbol{\phi}_{ij}\right) \doteq \textstyle\prod_{k=i}^{j-1}\mathrm{Exp}\left(-\Delta\tilde{\mathtt{R}}_{k+1j}^\mathsf{T}\mathtt{J}_r^k\,\boldsymbol{\eta}_k^{gd}\,\Delta t\right). \quad (32)$$

Taking the Log at both members and changing signs, we get:

$$\delta\boldsymbol{\phi}_{ij} = -\mathrm{Log}\left(\textstyle\prod_{k=i}^{j-1}\mathrm{Exp}\left(-\Delta\tilde{\mathtt{R}}_{k+1j}^\mathsf{T}\mathtt{J}_r^k\,\boldsymbol{\eta}_k^{gd}\,\Delta t\right)\right). \quad (33)$$

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:

$$\delta\boldsymbol{\phi}_{ij} \simeq \textstyle\sum_{k=i}^{j-1}\Delta\tilde{\mathtt{R}}_{k+1j}^\mathsf{T}\,\mathtt{J}_r^k\,\boldsymbol{\eta}_k^{gd}\,\Delta t \qquad (34)$$

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 (31) 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.

Therefore, we can fully characterize the noise as:

$$[\delta\boldsymbol{\phi}_{ij}^\mathsf{T}, \delta\mathbf{v}_{ij}^\mathsf{T}, \delta\mathbf{p}_{ij}^\mathsf{T}]^\mathsf{T} \sim \mathcal{N}(\mathbf{0}_{9\times 1}, \boldsymbol{\Sigma}_{ij}). \tag{35}$$

The expression for the covariance $\boldsymbol{\Sigma}_{ij}$ is provided in the supplementary material [29], where we also show that both the preintegrated measurements $\Delta\tilde{\mathtt{R}}_{ij}, \Delta\tilde{\mathbf{v}}_{ij}, \Delta\tilde{\mathbf{p}}_{ij}$, and the covariance $\boldsymbol{\Sigma}_{ij}$ can be computed incrementally.

C. Incorporating Bias Updates

In the previous section, we assumed that the bias $\mathbf{b}_i$ used to compute the preintegrated measurements is given. However, more likely, the bias estimate changes 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:

$$\begin{aligned}
\Delta\tilde{\mathtt{R}}_{ij}(\mathbf{b}_i^g) &\simeq \Delta\tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)\,\mathrm{Exp}\Big(\tfrac{\partial\Delta\bar{\mathtt{R}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}^g\Big)\\
\Delta\tilde{\mathbf{v}}_{ij}(\mathbf{b}_i^g, \mathbf{b}_i^a) &\simeq \Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i^g, \bar{\mathbf{b}}_i^a) + \tfrac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g + \tfrac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^a}\delta\mathbf{b}_i^a\\
\Delta\tilde{\mathbf{p}}_{ij}(\mathbf{b}_i^g, \mathbf{b}_i^a) &\simeq \Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i^g, \bar{\mathbf{b}}_i^a) + \tfrac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g + \tfrac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^a}\delta\mathbf{b}_i^a
\end{aligned} \tag{36}$$

This is similar to the bias correction in [26] but operates directly on $\mathrm{SO}(3)$. 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 $\bar{\mathbf{b}}_i$) describe how the measurements change due to a change in the bias estimate. The derivation of the Jacobians is very similar to the one we used in Section V-A to express the measurements as a large value *plus* a small perturbation; hence, we omit the complete derivation, which can be found in the supplementary material [29]. Note that the Jacobians remain constant and can be precomputed during the preintegration.

D. Preintegrated IMU Factors

Given the preintegrated measurement model in (31) and since measurement noise is zero-mean and Gaussian up to first order (35), it is now easy to write the residual errors $\mathbf{r}_{\mathcal{I}_{ij}} \doteq [\mathbf{r}_{\Delta\mathtt{R}_{ij}}^\mathsf{T}, \mathbf{r}_{\Delta\mathbf{v}_{ij}}^\mathsf{T}, \mathbf{r}_{\Delta\mathbf{p}_{ij}}^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^9$, where

$$\begin{aligned}
\mathbf{r}_{\Delta\mathtt{R}_{ij}} &\doteq \mathrm{Log}\left(\left(\Delta\tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)\mathrm{Exp}\left(\tfrac{\partial\Delta\bar{\mathtt{R}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}^g\right)\right)^\mathsf{T}\mathtt{R}_i^\mathsf{T}\mathtt{R}_j\right)\\
\mathbf{r}_{\Delta\mathbf{v}_{ij}} &\doteq \mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right)\\
&\quad - \left[\Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i^g, \bar{\mathbf{b}}_i^a) + \tfrac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}^g + \tfrac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^a}\delta\mathbf{b}^a\right]\\
\mathbf{r}_{\Delta\mathbf{p}_{ij}} &\doteq \mathtt{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \tfrac{1}{2}\mathbf{g}\Delta t_{ij}^2\right)\\
&\quad - \left[\Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i^g, \bar{\mathbf{b}}_i^a) + \tfrac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}^g + \tfrac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}_a}\delta\mathbf{b}^a\right],
\end{aligned} \tag{37}$$

in which we also included the bias updates of Eq. (36).

According to the "lift-solve-retract" method (Section II-C), at each GN iteration we need to re-parametrize (37) using the retraction (15). Then, the "solve" step requires to linearize the resulting cost. For the purpose of linearization, it is convenient to compute analytic expressions of the Jacobians of the residual errors, which we derive in [29].

E. Bias Model

When presenting the IMU model (20), we said that biases are slowly time-varying quantities. Hence, we model them with a "Brownian motion", *i.e.*, integrated white noise:

$$\dot{\mathbf{b}}^g(t) = \boldsymbol{\eta}^{bg}, \qquad \dot{\mathbf{b}}^a(t) = \boldsymbol{\eta}^{ba}. \tag{38}$$

Integrating (38) over the time interval $[t_i, t_j]$ between two consecutive keyframes i and j we get:

$$\mathbf{b}_j^g = \mathbf{b}_i^g + \boldsymbol{\eta}^{bgd}, \qquad \mathbf{b}_j^a = \mathbf{b}_i^a + \boldsymbol{\eta}^{bad}, \tag{39}$$

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.* [40, Appendix]).

The model (39) can be readily included in our factor graph, as a further additive term in (19) for all consecutive keyframes:

$$\|\mathbf{r}_{\mathbf{b}_{ij}}\|^2 \doteq \|\mathbf{b}_j^g - \mathbf{b}_i^g\|^2_{\boldsymbol{\Sigma}^{bgd}} + \|\mathbf{b}_j^a - \mathbf{b}_i^a\|^2_{\boldsymbol{\Sigma}^{bad}} \tag{40}$$

VI. 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 (19) via the sum:

$$\textstyle\sum_{i\in\mathcal{K}_k}\sum_{l\in\mathcal{C}_i}\|\mathbf{r}_{\mathcal{C}_{il}}\|^2_{\boldsymbol{\Sigma}_\mathcal{C}} = \sum_{l=1}^{L}\sum_{i\in\mathcal{X}(l)}\|\mathbf{r}_{\mathcal{C}_{il}}\|^2_{\boldsymbol{\Sigma}_\mathcal{C}} \tag{41}$$

which, on the right-hand-side, we rewrote as a sum of contributions of each landmark $l = 1, \ldots, L$. In (41), $\mathcal{X}(l)$ denotes the subset of keyframes in which l is seen.

A fairly standard model for the residual error of a single pixel measurement $\mathbf{z}_{il}$ is [13]:

$$\mathbf{r}_{\mathcal{C}_{il}} = \mathbf{z}_{il} - \pi(\mathtt{R}_i, \mathbf{p}_i, \rho_l), \tag{42}$$

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 $\mathtt{T}_{\mathrm{BC}}$.

Direct use of (42) would require to include the landmark positions $\rho_l, l = 1, \ldots, L$ 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 II-C, at each GN iteration, we *lift* the cost function, using the retraction (15). For the vision factors this means that the original residuals (41) become:

$$\textstyle\sum_{l=1}^{L}\sum_{i\in\mathcal{X}(l)}\|\mathbf{z}_{il} - \check{\pi}(\delta\boldsymbol{\phi}_i, \delta\mathbf{p}_i, \delta\rho_l)\|^2_{\boldsymbol{\Sigma}_\mathcal{C}} \tag{43}$$

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:

$$\textstyle\sum_{l=1}^{L}\sum_{i\in\mathcal{X}(l)}\|\mathbf{F}_{il}\delta\mathbf{T}_i + \mathbf{E}_{il}\delta\rho_l - \mathbf{b}_{il}\|^2, \tag{44}$$

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 (44) in matrix form we get:

$$\sum_{l=1}^{L} \|\mathbf{F}_l \, \delta\mathbf{T}_{\mathcal{X}(l)} + \mathbf{E}_l \, \delta\rho_l - \mathbf{b}_l\|^2 \tag{45}$$

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 $i \in \mathcal{X}(l)$. We can eliminate the variable $\delta\rho_l$ by projecting the residual into the null space of $\mathbf{E}_l$:

$$\sum_{l=1}^{L} \left\|\mathbf{Q}(\mathbf{F}_l \, \delta\mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l)\right\|^2 \tag{46}$$

where $\mathbf{Q} \doteq \mathbf{I} - \mathbf{E}_l(\mathbf{E}_l^\top \mathbf{E}_l)^{-1}\mathbf{E}_l^\top$ is an orthogonal projector of $\mathbf{E}_l$ as shown in the supplementary material [29]. Using this approach, we reduced a large set of factors (43) which involve poses and landmarks into a smaller set of L factors (46), 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. 4.

VII. Implementation

Our implementation consists of a high frame rate tracking front-end based on SVO[2] [43] and an optimization back-end based on iSAM2[3] [24]. The front-end tracks the pose at camera rate while the back-end optimizes in parallel the state of selected *keyframes* as described in this paper.

SVO [43] is a precise and robust monocular visual odometry system that employs *sparse image alignment*, which estimates incremental motion and tracks features by minimizing the photometric error between subsequent images. Thereby, SVO avoids every-frame feature extraction, resulting in high-frame-rate motion estimation. Combined with an outlier resistant probabilistic triangulation method, SVO provides increased robustness in scenes with repetitive and high frequency texture.

The computation of the MAP estimate in Eq. (19) is based on iSAM2 [24], which is a state-of-the-art incremental smoothing approach. 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 the new measurements. In odometry problems, the use of iSAM2 results in constant-time updates.

VIII. Experiments

The first experiment shows that the proposed approach is more accurate than two competitive state-of-the-art approaches, namely ASLAM [9], and MSCKF [20]. The experiment is performed on the indoor trajectory of Fig. 6. The dataset was recorded with a forward-looking VI-Sensor [44] 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 [41]. 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 [45].

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

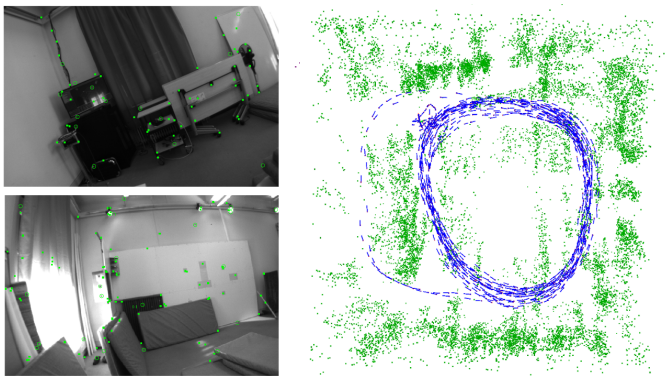

Fig. 6: Left: two images from the indoor trajectory dataset with tracked features in green. Right: top view of the trajectory estimate produced by our approach (blue) and 3D landmarks triangulated from the trajectory (green).

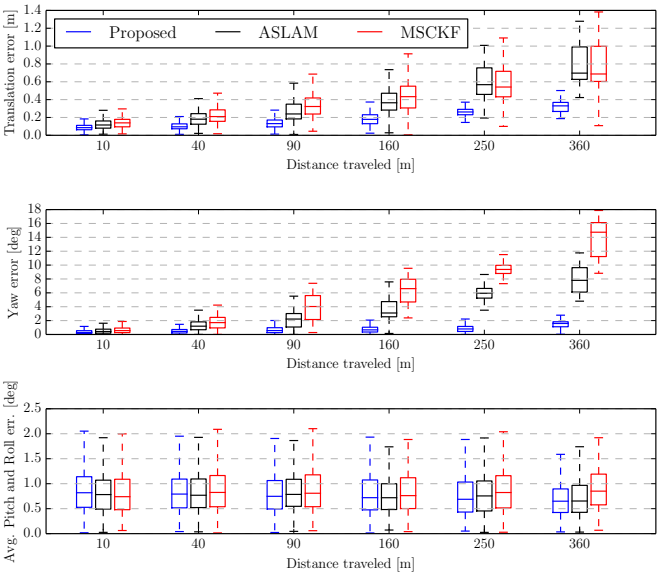

Fig. 7: Comparison of the proposed approach versus the ASLAM algorithm [9] and an implementation of the MSCKF filter [20]. Relative errors are measured over different segments of the trajectory, of length $\{10, 40, 90, 160, 250, 360\}$m, according to the odometric error metric in [46].

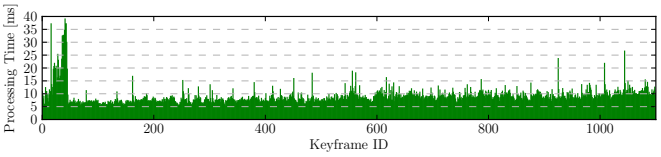

Fig. 8: Processing-time per keyframe for the proposed VIN approach.

Fig. 7 compares the proposed system against the ASLAM algorithm [9], and an implementation of the MSCKF filter [20]. Both these algorithms currently represent the state-of-the-art in VIN, ASLAM for optimization-based approaches, and MSCKF for filtering methods. We obtained the datasets as well as the trajectories computed with ASLAM and MSCKF from the authors of [9]. We use the relative error metrics proposed in [46] 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. 7). Our approach exhibits less drift than the state-of-the-art, achieving 0.3m drift on average over 360m travelled

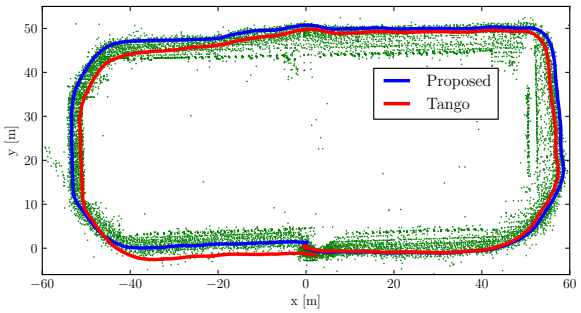

Fig. 9: Outdoor trajectory (length: 300m) around a building with identical start and end point at coordinates $(0, 0, 0)$. The end-to-end error of the proposed approach is 1.0m. Google Tango accumulated 2.2m drift. The green dots are the 3D points triangulated from our trajectory estimate.

distance; ASLAM 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.

Figure 8 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 ASLAM were not reported, the approach is described as computationally demanding [9]. ASLAM needs to repeat IMU integration at every change of the linearization point, which we avoid by using the preintegrated IMU measurements.

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* VIN system. We rigidly attached the VI-Sensor to a Tango device and walked around an office building. Fig. 9 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.

The third experiment is the one in Fig. 1. 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 [47, 48] 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.

The fourth experiment evaluates a specific aspect of our

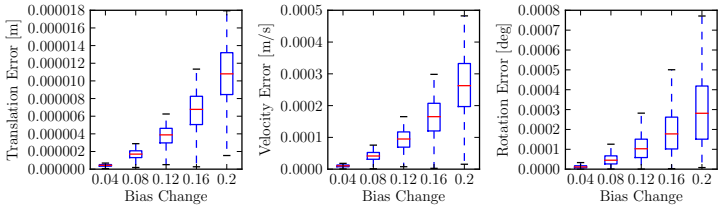

Fig. 10: Error committed when using the first-order approximation (36) instead of repeating the integration, for different bias perturbations. Left: $\Delta\tilde{\mathbf{R}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$ error; Center: $\Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$ error; Right: $\Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$ error. Statistics are computed over 1000 Monte Carlo runs.

approach: the a-posteriori bias correction of Section V-C. To evaluate the quality of the first-order approximation (36), we performed the following Monte Carlo analysis. First, we computed the preintegrated measurements $\Delta\tilde{\mathbf{R}}_{ij}(\bar{\mathbf{b}}_i)$, $\Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i)$ and $\Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i)$ over 100 IMU measurements at a given bias estimate $\bar{\mathbf{b}}_i$. Then, we applied a perturbation $\delta\mathbf{b}_i$ (in a random direction with magnitude between 0.04 and 0.2) to both the gyroscope and accelerometer bias. We repeated the integration at $\bar{\mathbf{b}}_i + \delta\mathbf{b}_i$, and we obtained $\Delta\tilde{\mathbf{R}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$, $\Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$ and $\Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$. Finally, we compared the result of the integration against the first-order correction in (36). Fig. 10 reports statistics of the errors in the preintegrated variables between the re-integrated variables and the first-order correction, over 1000 Monte Carlo runs. The order of magnitude of the errors suggests that the first-order approximation captures very well the bias change, and can be safely used for relatively large bias fluctuations.

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

We propose a novel preintegration theory that 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., [8, 20], as we do not commit to a linearization point during integration. Moreover, it leverages the seminal work on preintegration [26], bringing to maturity the preintegration and uncertainty propagation in $\mathrm{SO}(3)$. We also discuss how to use the preintegrated IMU model in a VIN pipeline; we adopt a structureless model for visual measurements which avoids optimizing over 3D landmarks. Our VIN 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). We provide comparisons against state-of-the-art alternatives, including filtering and optimization-based techniques. We release the source-code of the IMU preintegration and the structurless vision factors in the GTSAM 4.0 optimization toolbox [30] and provide additional theoretical derivations and implementation details in the supplementary material [29].

Acknowledgments The authors gratefully acknowledge Stephen Williams and Richard Roberts for helping with an early implementation in GTSAM, and Simon Lynen and Stefan Leutenegger for providing datasets and results of their algorithms.

REFERENCES

[1] Google. Project tango. URL https://www.google.com/atap/projecttango/.

[2] 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.

[3] 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.

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

[5] 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.

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

[7] 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.

[8] 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.

[9] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual-inertial slam using nonlinear optimization. *Intl. J. of Robotics Research*, 2014.

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

[11] D.-N. Ta, K. Ok, and F. Dellaert. Vistas and parallel tracking and mapping with wall-floor features: Enabling autonomous flight in man-made environments. *Robotics and Autonomous Systems*, 62(11):1657–1667, 2014. doi: http://dx.doi.org/10.1016/j.robot.2014.03.010. Special Issue on Visual Control of Mobile Robots.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] E.D. Nerurkar, K.J. Wu, and S.I. Roumeliotis. C-KLAM: Constrained keyframe-based localization and mapping. In *Robotics: Science and Systems (RSS)*, 2013.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

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

[28] 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.

[29] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. Supplementary material to: IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. Technical Report GT-IRIM-CP&R-2015-001, Georgia Institute of Technology, 2015.

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

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

[32] 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.

[33] 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.

[34] 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.

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

[36] 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.

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

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

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

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

[41] 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.

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

[43] 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.

[44] 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.

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

[46] 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.

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

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