Invariant Kalman Filtering for Visual Inertial SLAM

Martin BROSSARD*, Silvère BONNABEL* and Axel BARRAU†
*MINES ParisTech, PSL Research University, Centre for Robotics, 60 Boulevard Saint-Michel, 75006 Paris, France
†Safran Tech, Groupe Safran, Rue des Jeunes Bois-Châteaufort, 78772, Magny Les Hameaux Cedex, France

Abstract—**Combining visual information with inertial measurements is a popular approach to achieve robust and autonomous navigation in robotics, specifically in GPS-denied environments. In this paper, building upon both the recent theory of Unscented Kalman Filtering on Lie Groups (UKF-LG) and more generally the theory of invariant Kalman filtering (IEKF), an innovative UKF is derived for the monocular visual simultaneous localization and mapping (SLAM) problem. The body pose, velocity, and the 3D landmarks' positions are viewed as a single element of a (high dimensional) Lie group $SE_{2+p}(3)$, which constitutes the state, and where the accelerometers' and gyrometers' biases are appended to the state and estimated as well. The efficiency of the approach is validated both on simulations and on five real datasets.**

Index Terms—**Lie groups, invariant Kalman filtering, unscented Kalman filter, visual inertial SLAM, sensor fusion, localization**

I. INTRODUCTION

Over the last decades, tremendous progresses have been achieved in visual simultaneous localization and mapping frameworks (SLAM), see e.g., [1]. Most approaches include data fusion using filters [2]–[5], or optimization/bundle adjustment techniques, e.g., [6]–[8]. Optimization based methods are more efficient but generally come with higher computational demands than more basic filtering algorithms which are generally more suited to real-time applications.

In this paper, we tackle the problem of fusing Inertial Measurement Unit (IMU) signals with monocular vision for SLAM for Micro Aerial Vehicles (MAVs). We propose a novel Unscented Kalman Filter (UKF) that mainly builds upon two components. First, the recent Lie group structure of SLAM advocated in the field of invariant filtering, see [9]–[11]. Secondly, the UKF on Lie Groups (UKF-LG), whose general methodology has been recently introduced in [12]. The effectiveness of our algorithm is tested both on simulations and on real data [13]. The method, an UKF-LG visual SLAM, favorably compares to some state-of-the-art Kalman filter based solutions.

Note that, in the present paper we do not specifically address Visual Inertial Odometry (VIO) which is a powerful technique where the features (i.e., the map) are not included in the state, saving execution time. As in SLAM, VIO estimates the sequential changes of the robot over time using an IMU and cameras, but there is no attempt to build a map [14]. Here, by contrast, we explicitly consider the probabilistic visual SLAM problem, where a consistent map of the environment is also pursued. Note that, even for navigation purposes only, building a map allows loop closures, a powerful method to drastically reduce uncertainty on the state, when applicable.

A. Contributions and Links with Previous Literature

In robotics, it has been long recognized that the Lie group structure of the space of poses $SE(3)$ plays an important role, see e.g., [15,16]. More recently, probability distributions on $SE(3)$, and their role for control and estimation, have been well studied, see e.g., [17]–[19], and the monographs [20,21].

In [9], we proposed the Right Invariant Extended Kalman Filter (RIEKF) based SLAM. Letting the output function $h(.)$ of [9] be the projection onto the camera frame, we readily obtain a RIEKF for visual SLAM, and this filter has been described and also advocated recently in [4] for 3D Visual Inertial Navigation Systems (VINS), owing to its consistency properties. However, in the implementation section of the latter paper, the authors rather opt for a multi-state constrained RIEKF. The resulting RI-MSCKF does not address the SLAM problem since landmarks are removed from the state (see eq. 21 therein). Very recently [22] also proposed a multi-state constrained Right invariant EKF (RIEKF) for 3D VINS. In Section VI dedicated to experiments, we also apply the framework [9] to the inertial and vision fusion, to implement a RIEKF where the landmarks are part of the state. To our best knowledge this is the first published implementation of an RIEKF for visual inertial SLAM.

Our main contribution is a Right-UKF-LG, which can be viewed as an unscented-based transform alternative to the RIEKF, but which has the advantage of being much more versatile than the RIEKF. Indeed, it spares the user the computation of Jacobians, that can prove difficult, especially in the Invariant EKF framework where Jacobians are defined with respect to the Lie structure, see e.g., [10]. As a result,

- the practitioner can *readily* implement our algorithm when using, e.g., a different camera model, or if one wants to add additional measurements such as GPS measurements outdoors, or a complementary depth sensor;
- should additional parameters/variables be estimated, such as IMU's scale factors and/or harmonization angles, the algorithm is straightforward to adapt following the state augmentation technique of Section III-C3.

Note that, the present paper presents some new developments on the UKF-LG methodology, that are as follows: the state augmentation technique of Section III-C3 allows dealing with state spaces that are not Lie groups, a square-root form implementation detailed in Appendix A and a modification to deal with large updates described in Appendix B.

In [5], the authors consider the same visual inertial fusion problem, and devise an UKF that takes advantage of the Lie group structure of the body pose $SE(3)$. The main differences are threefold. First, the Lie group we use $SE_{2+p}(3)$, introduced in [9,23], is much bigger than $SE(3)$, and includes the pose but also the velocity and the landmarks' positions. Then, and more generally, the UKF-LG [12] generates sigma points in the Lie algebra, and then uses concentrated Gaussian distributions (as in e.g., [18]) to map them onto the group. In contrast, [5] uses a probability distribution directly defined on the group [20] to generate the sigma points, which is akin to the general unscented Kalman filtering on manifolds of [24]. Moreover, while [5] uses parallel transport operations based on left multiplications, we explore two variants based both on left and right multiplications, and the right one proves to be actually much better.

B. Paper's Organization

The paper is organized as follows. Section II formulates the fusion problem. Section III contains mathematical preliminaries on unscented Kalman filtering on Lie groups. Section IV describes the two proposed UKFs for monocular visual and inertial SLAM. Section V and VI illustrate the performances of the proposed filters based both on Monte-Carlo simulations and on real datasets.

II. Visual Inertial SLAM Problem Modeling

We recall in this section the standard dynamic model for flying devices equipped with an IMU. We then detail the visual measurement model, and we finally pose the SLAM problem.

A. Variables of Interest and Dynamical Model

Let us consider an aerial body equipped with an IMU whose biases are modeled as random walks. Assume moreover that p fixed landmarks of the scene can be tracked visually, and that they constitute the map. The state we want to estimate consists of the position $\mathbf{x} \in \mathbb{R}^3$, velocity $\mathbf{v} \in \mathbb{R}^3$, orientation $\mathbf{R} \in SO(3)$ of the body, the IMU biases $\mathbf{b}_{\boldsymbol{\omega}} \in \mathbb{R}^3$ and $\mathbf{b}_{\mathbf{a}} \in \mathbb{R}^3$, as well as the 3D positions $\mathbf{p}_1, \ldots, \mathbf{p}_p \in \mathbb{R}^3$ of the landmarks in the global frame. The dynamics of the system read:

$$\text{body state dynamics} \begin{cases} \dot{\mathbf{R}} = \mathbf{R}\left(\boldsymbol{\omega} - \mathbf{b}_{\boldsymbol{\omega}} + \mathbf{w}_{\boldsymbol{\omega}}\right)_\times \\ \dot{\mathbf{v}} = \mathbf{R}\left(\mathbf{a} - \mathbf{b}_{\mathbf{a}} + \mathbf{w}_{\mathbf{a}}\right) + \mathbf{g} \\ \dot{\mathbf{x}} = \mathbf{v} \end{cases}, \quad (1)$$

$$\text{IMU biases dynamics} \begin{cases} \dot{\mathbf{b}}_{\boldsymbol{\omega}} = \mathbf{w}_{\mathbf{b}_{\boldsymbol{\omega}}} \\ \dot{\mathbf{b}}_{\mathbf{a}} = \mathbf{w}_{\mathbf{b}_{\mathbf{a}}} \end{cases}, \quad (2)$$

$$\text{landmarks dynamics} \left\{ \dot{\mathbf{p}}_i = \mathbf{0}, \ i = 1, \ldots, p \right. \quad , \quad (3)$$

where $(\boldsymbol{\omega})_\times$ denotes the skew symmetric matrix associated with the cross product with vector $\boldsymbol{\omega} \in \mathbb{R}^3$. The various white Gaussian continuous time noises can be stacked as

$$\mathbf{w} = \begin{bmatrix}\mathbf{w}_{\boldsymbol{\omega}}^T \ \mathbf{w}_{\mathbf{a}}^T \ \mathbf{w}_{\mathbf{b}_{\boldsymbol{\omega}}}^T \ \mathbf{w}_{\mathbf{b}_{\mathbf{a}}}^T\end{bmatrix}^T, \quad (4)$$

where $\mathbf{w}$ is centered with autocorrelation $\mathbb{E}(\mathbf{w}(t)\mathbf{w}(s)) = \mathbf{W}\delta(t-s)$. These equations correspond to the equations of navigation, provided the earth is considered as locally flat. They model the dynamics of most of MAVs such as quadrotors where the IMU measurements $\boldsymbol{\omega}$ and $\mathbf{a}$ in (1) are considered as noisy and biased inputs of the system.

B. Measurement Model

In addition to the IMU measurements used as input for the dynamics, the vehicle gets visual information from a calibrated monocular camera. It observes and tracks the p landmarks through a standard pinhole model [25]. Landmark $\mathbf{p}_i$ is observed through the camera as

$$\mathbf{y}_i = \frac{1}{y_w^i}\begin{bmatrix} y_u^i \\ y_v^i \end{bmatrix} + \mathbf{n}_{\mathbf{y}}^i, \quad (5)$$

where $\mathbf{y}_i$ is the measured normalized pixel location of the landmark in the camera frame, that is,

$$\begin{bmatrix} y_u^i \\ y_v^i \\ y_w^i \end{bmatrix} = \mathbf{R}_C^T\left(\mathbf{R}^T\left(\mathbf{p}_i - \mathbf{x}\right) - \mathbf{x}_C\right), \quad (6)$$

where the right term corresponds to the distance from the landmark to the camera expressed in the camera frame. $\mathbf{R}_C$ and $\mathbf{x}_C$ are the known rotation matrix and the translation mapping from the body frame onto the camera frame. Finally, $\mathbf{n}_{\mathbf{y}} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{N}\right)$ represents the pixel image noise.

C. Estimation/Fusion Problem

Our goal is to compute the probability distribution of the high dimensional system's state $(\mathbf{R}, \mathbf{x}, \mathbf{v}, \mathbf{b}_{\boldsymbol{\omega}}, \mathbf{b}_{\mathbf{a}}, \mathbf{p}_1, \ldots, \mathbf{p}_p)$ defined through an initial Gaussian prior and the probabilistic dynamic model (1)-(3), *conditionally* on the visual landmarks measurements of the form (5) for $1 \leq i \leq p$. This is the standard probabilistic formulation of the visual 3D SLAM problem with an IMU.

III. Unscented Kalman Filtering on Lie Groups

In this section we provide the reader with the bare minimum about the UKF on Lie Groups (UKF-LG) introduced in [12].

A. Matrix Lie Groups

A matrix Lie group $G \subset \mathbb{R}^{N \times N}$ is a subset of square invertible matrices such that the following properties hold

$$\mathbf{I} \in G \quad (7)$$

$$\forall \boldsymbol{\chi} \in G, \boldsymbol{\chi}^{-1} \in G \quad (8)$$

$$\forall \boldsymbol{\chi}_1, \boldsymbol{\chi}_2 \in G, \boldsymbol{\chi}_1\boldsymbol{\chi}_2 \in G. \quad (9)$$

Locally about the identity matrix $\mathbf{I}$, the group G can be identified with an Euclidean space $\mathbb{R}^q$ using the matrix exponential map $\exp_m(.)$, where $q = \dim G$. Indeed, to any $\boldsymbol{\xi} \in \mathbb{R}^q$ one can associate a matrix $\mathcal{L}_{\mathfrak{g}}(\boldsymbol{\xi}) = \boldsymbol{\xi}^\wedge$ of the tangent space of G at $\mathbf{I}$, called the Lie algebra $\mathfrak{g}$. We then define the exponential map $\exp : \mathbb{R}^q \to G$ for Lie groups as

$$\exp\left(\boldsymbol{\xi}\right) = \exp_m\left(\boldsymbol{\xi}^\wedge\right). \quad (10)$$

Locally, it is a bijection, and one can define the Lie logarithm map $\log : G \to \mathbb{R}^q$ as the exponential inverse, leading to

$$\log\left(\exp\left(\boldsymbol{\xi}\right)\right) = \boldsymbol{\xi}. \quad (11)$$

B. Uncertainty on Lie Groups

To define random variables on Lie groups, we cannot apply the usual approach of additive noise for $\mathcal{X} \in G$ as G is not a vector space. In contrast, we define the probability distribution $\mathcal{X} \sim \mathcal{N}_L(\bar{\mathcal{X}}, \mathbf{P})$ for the random variable $\mathcal{X} \in G$ as [18,26]

$$\mathcal{X} = \bar{\mathcal{X}} \exp\left(\boldsymbol{\xi}\right), \ \boldsymbol{\xi} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{P}\right), \tag{12}$$

where $\mathcal{N}\left(.,.\right)$ is the classical Gaussian distribution in Euclidean space $\mathbb{R}^q$ and $\mathbf{P} \in \mathbb{R}^{q \times q}$ is a covariance matrix. In (12), the original Gaussian $\boldsymbol{\xi}$ of the Lie algebra is moved over by left multiplication to be centered at $\bar{\mathcal{X}} \in G$, hence the letter L which stands for "left", this type of uncertainty being also referred to as left-equivariant [12]. We similarly define the distribution $\mathcal{X} \sim \mathcal{N}_R(\bar{\mathcal{X}}, \mathbf{P})$ for right multiplication of $\bar{\mathcal{X}}$, as

$$\mathcal{X} = \exp\left(\boldsymbol{\xi}\right) \bar{\mathcal{X}}, \ \boldsymbol{\xi} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{P}\right). \tag{13}$$

In (12) and (13), $\bar{\mathcal{X}}$ may represent a large, noise-free and deterministic value, whereas $\mathbf{P}$ is the covariance of the small, noisy perturbation $\boldsymbol{\xi}$. We stress that $\boldsymbol{\xi} \in \mathbb{R}^q$ is Gaussian, but $\mathcal{N}_L\left(.,.\right)$ and $\mathcal{N}_R\left(.,.\right)$ are not.

Remark 1: defining Gaussian distributions on Lie groups through (12) and (13) is advocated notably in [18,26], and the corresponding distribution is sometimes referred to as concentrated Gaussian on Lie groups, see [27]. An alternative approach, introduced to our best knowledge in [20], and used in [19], consists of defining a (Gaussian) density directly on the group using the Haar measure. In the latter case, the group needs be unimodular, but such a requirement is in fact unnecessary to define the random variables (12) and (13).

C. Unscented Kalman Filtering on Lie Groups

By representing the state error as a variable $\boldsymbol{\xi}$ of the Lie algebra, we can build two alternative unscented filters for any state living in a Lie group, along the lines of [12]. Let us consider a discrete time dynamical system of the form

$$\mathcal{X}_{n+1} = f\left(\mathcal{X}_n, \mathbf{u}_n, \mathbf{w}_n\right), \tag{14}$$

where the state $\mathcal{X}_n \in G$, $\mathbf{u}_n$ is a known input variable and $\mathbf{w}_n \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}_n\right)$ is a white Gaussian noise. Consider discrete measurements of the form

$$\mathbf{y}_n = h\left(\mathcal{X}_n, \mathbf{n}_n\right), \tag{15}$$

where $\mathbf{n}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{N}_n)$ is a white Gaussian noise. Two different UKFs follow from the above uncertainty representation.

1) Left-UKF-LG: the state is modeled as $\mathcal{X}_n \sim \mathcal{N}_L(\bar{\mathcal{X}}_n, \mathbf{P}_n)$, that is, using the left-equivariant formulation (12) of the uncertainties. The mean state is thus encoded in $\bar{\mathcal{X}}_n$ and dispersion in $\boldsymbol{\xi} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{P}_n\right)$. The sigma points are generated based on the variable $\boldsymbol{\xi}$, and mapped to the group through the model (12). Note that, this is in slight contrast with [5,19], which generate sigma points through a distribution defined directly on the group. The filter consists of two steps along the lines of the conventional UKF: propagation and update, and compute estimates $\bar{\mathcal{X}}_n$ and $\mathbf{P}_n$ at each n.

2) Right-UKF-LG: the state is alternatively modeled as $\mathcal{X}_n \sim \mathcal{N}_R(\bar{\mathcal{X}}_n, \mathbf{P}_n)$, that is, using the right-equivariant formulation (13) of the uncertainties.

3) UKF-LG with state augmentation: the paper [12] is dedicated to the case where the state is a Lie group. However, when the state consists of a matrix $\mathcal{X}$ belonging to a Lie group, and an additional vector, say $\mathbf{b}$, the methodology is straightforward to apply by augmenting the state space: vector $\mathbf{b}$ is appended to the state which then becomes the couple $(\mathcal{X}, \mathbf{b})$. The Lie group variable $\mathcal{X}$ is treated using the UKF-LG methodology, whereas the vector variable is treated as in the conventional UKF. We use this approach in the sequel, where the vector variable corresponds to the IMU biases.

D. The Special Euclidean Group $SE_{2+p}(3)$

In [23], the author noticed there is a natural Lie group structure underlying the (wheeled robot based) SLAM problem. The corresponding Lie group was named $SE_{1+p}(3)$ in [9], and leveraged therein to design an Invariant EKF, which resolves some well-known consistency issues of the conventional EKF based SLAM. Some alternative properties of the Invariant EKF based SLAM have also recently been proved in [4,11].

Any matrix $\mathcal{X} \in SE_{2+p}(3)$ is defined as

$$\mathcal{X} = \left[\begin{array}{cccccc} \mathbf{R} & \mathbf{v} & \mathbf{x} & \mathbf{p}_1 & \cdots & \mathbf{p}_p \\ \multicolumn{2}{c}{\mathbf{0}_{2+p \times 3}} & & \multicolumn{3}{c}{\mathbf{I}_{2+p \times 2+p}} \end{array} \right]. \tag{16}$$

The dimension of the group, and thus of the Lie algebra, is $3 + 3(2 + p)$. The uncertainties, defined as $\boldsymbol{\xi} = \left[\boldsymbol{\xi}_{\mathbf{R}}^T \boldsymbol{\xi}_{\mathbf{v}}^T \boldsymbol{\xi}_{\mathbf{x}}^T \boldsymbol{\xi}_{\mathbf{p}_1}^T \cdots \boldsymbol{\xi}_{\mathbf{p}_p}^T\right]^T \in \mathbb{R}^{9+3p}$, are mapped to the Lie algebra through the transformation $\mathcal{L}_{\mathfrak{g}} : \boldsymbol{\xi} \mapsto \mathcal{L}_{\mathfrak{g}}(\boldsymbol{\xi}) = \boldsymbol{\xi}^\wedge$ as

$$\boldsymbol{\xi}^\wedge = \left[\begin{array}{c} (\boldsymbol{\xi}_{\mathbf{R}})_\times \ \boldsymbol{\xi}_{\mathbf{v}} \ \boldsymbol{\xi}_{\mathbf{x}} \ \boldsymbol{\xi}_{\mathbf{p}_1} \ \cdots \ \boldsymbol{\xi}_{\mathbf{p}_p} \\ \mathbf{0}_{2+p \times 5+p} \end{array} \right]. \tag{17}$$

The closed form expressions for exponential, logarithm and Jacobian can be deduced along the lines of [9,26].

IV. Proposed Algorithms

To apply the methodology of UKF on Lie groups, the dynamics must first be discretized, and the state space must be (partly) embedded in a matrix Lie group.

A. Time Discretization

Equations (1) are standard navigation equations, and their discretization is well established. In this paper, we implemented the method of pre-integration on manifolds of [8].

B. Lie Group Embedding

The state space can be partially embedded into a Lie group, by letting $\mathcal{X}_n$ be the matrix of the group $G = SE_{2+p}(3)$ that represents the state variables $(\mathbf{R}, \mathbf{v}, \mathbf{x}, \mathbf{p}_1, \cdots, \mathbf{p}_p)$ at time step n through representation (16). Using this embedding, the state at time n can in turn be represented as $(\mathcal{X}_n, \mathbf{b}_n)$, letting the bias vector be $\mathbf{b} = \left[\mathbf{b}_{\boldsymbol{\omega}}^T \ \mathbf{b}_{\mathbf{a}}^T\right]^T \in \mathbb{R}^6$. The dispersion on $\mathcal{X}_n$ can be encoded using the left uncertainty (12) or the right one (13), leading to two alternative filters (see Section III-C). In the following, we detail the Right-UKF-LG which adopts the right-equivariant uncertainties (13) of $\mathcal{X}_n$ and and conventional additive uncertainties on the biases $\mathbf{b}$. We leave to the reader the derivation of the Left-UKF-LG, based upon left-equivariant uncertainties (12).

C. Final Retained Model and Filter Architecture

Defining the input vector $\mathbf{u} = \begin{bmatrix}\boldsymbol{\omega}^T & \mathbf{a}^T\end{bmatrix}^T$, and gathering the results of the two preceding subsections, we obtain the following uncertainty representation and discrete time model associated to the Right-UKF-LG:

$$\text{uncertainty rep.} \begin{cases} \mathcal{X}_n = \exp\left(\boldsymbol{\xi}\right)\bar{\mathcal{X}}_n \\ \mathbf{b}_n = \bar{\mathbf{b}}_n + \tilde{\mathbf{b}} \end{cases}, \begin{bmatrix}\boldsymbol{\xi} \\ \tilde{\mathbf{b}}\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{P}_n\right), \tag{18}$$

$$\text{dynamics } \left\{ \mathcal{X}_n, \mathbf{b}_n = f\left(\mathcal{X}_{n-1}, \mathbf{u}_n - \mathbf{b}_{n-1}, \mathbf{w}_n\right), \right. \tag{19}$$

$$\text{observations} \begin{cases} \mathbf{Y}_n = \begin{bmatrix}\mathbf{y}_1^T & \cdots & \mathbf{y}_p^T\end{bmatrix}^T := \mathbf{Y}\left(\mathcal{X}_n, \mathbf{n}_n\right) \\ \mathbf{y}_i \text{ given in (5)},\ i = 1, \ldots, p \end{cases}, \tag{20}$$

such that $\left(\bar{\mathcal{X}}_n, \bar{\mathbf{b}}_n\right) \in \mathbb{R}^{15+3p}$ represents the mean (estimated) state at time n, $\mathbf{P}_n \in \mathbb{R}^{(15+3p)\times(15+3p)}$ is the covariance matrix that defines the state uncertainties $\left(\boldsymbol{\xi}, \tilde{\mathbf{b}}\right)$, and the vector $\mathbf{Y}_n$ contains the observations of the p landmarks with associated discrete Gaussian noise $\mathbf{w}_n \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}\right)$. The filter consists of two steps: propagation and update; as shown in Algorithm 1. We detail these two main steps in the following with the formalism of square-root implementation [28] where $\mathbf{S}$ is the Cholesky decomposition of the covariance, sparing the computation of covariance matrices and being numerically more stable.

Remark 2: for the Left-UKF-LG, we define $\mathcal{X}_n = \bar{\mathcal{X}}_n \exp\left(\boldsymbol{\xi}\right)$ and substitute it in (18). This results in quite different filters, though. In particular, consistence properties for EKF SLAM are characteristics of the right-invariant formalism, see [9] and also [4,11].

Algorithm 1: Left and Right UKF on Lie groups

```
  Input: $\bar{\mathcal{X}}, \bar{\mathbf{b}}, \mathbf{S} = \mathrm{chol}\left(\mathbf{P}\right), \mathbf{u}, \mathbf{Q}, \mathbf{Y}, \mathbf{N}$;
1 $\bar{\mathcal{X}}, \bar{\mathbf{b}}, \mathbf{S} \leftarrow$ Propagation $\left(\bar{\mathcal{X}}, \bar{\mathbf{b}}, \mathbf{S}, \mathbf{u}, \mathbf{Q}\right)$;
  if received measurement then
2     $\bar{\mathcal{X}}, \bar{\mathbf{b}}, \mathbf{S} \leftarrow$ Update $\left(\bar{\mathcal{X}}, \bar{\mathbf{b}}, \mathbf{S}, \mathbf{Y}, \mathbf{N}\right)$;
  Output: $\bar{\mathcal{X}}, \bar{\mathbf{b}}, \mathbf{S}$;
```

D. Propagation Step

The propagation step is described in Algorithm 2 and operates as follow. The filter first computes the propagated mean state, and then the $2J$ sigma points obtained at line 5 are propagated at lines 6-7. It is then convenient to view the propagated Cholesky factors $\mathbf{S}$ as an output of the function qr(.). Details are provided in Appendix A along with the definitions of J and γ. Although more details on the methodology can be found in [12] (see also [18] regarding propagation), line 7 deserves a few explanations for the paper to be self-contained. According to uncertainty model (13), dispersion around the mean is modeled as $\exp(\boldsymbol{\xi})\mathcal{X}$, so if $\bar{\mathcal{X}}$ denotes the propagated mean, and $\mathcal{X}_j$ denotes a propagated sigma point, then the corresponding sigma point in the Lie algebra is defined through $\exp(\boldsymbol{\xi}_j)\bar{\mathcal{X}} = \mathcal{X}_j$, i.e., $\boldsymbol{\xi}_j = \log\left(\mathcal{X}_j\bar{\mathcal{X}}^{-1}\right)$.

Algorithm 2: Propagation function for Right-UKF-LG

```
  Input: $\bar{\mathcal{X}}, \bar{\mathbf{b}}, \mathbf{S}, \mathbf{u}, \mathbf{Q}$;
1 $\mathbf{u} \leftarrow \mathbf{u} - \bar{\mathbf{b}}$ ;                    // unbiased input
2 $\mathbf{S}^a = \mathrm{blkdiag}\left(\mathbf{S}, \mathrm{chol}\left(\mathbf{Q}\right)\right)$;
3 $\mathcal{X} = \bar{\mathcal{X}}$ ;      // save non propagated state
4 $\bar{\mathcal{X}}, \bar{\mathbf{b}} = f\left(\mathcal{X}, \mathbf{u}, \mathbf{0}\right)$ ;            // propagate mean
  // step 5: sigma points generation
5 $\begin{bmatrix}\boldsymbol{\xi}_j^{\mp} & \mathbf{b}_j^{\mp} & \mathbf{n}_j^{\mp}\end{bmatrix} = \mp\gamma\,\mathrm{col}_j\left(\mathbf{S}^a\right),\ j = 1, \ldots, J$;
  // steps 6-7: sigma point propagation
6 $\mathcal{X}_j^{\mp}, \mathbf{b}_j^{\mp} \leftarrow f\left(\exp\left(\boldsymbol{\xi}_j^{\mp}\right)\mathcal{X}, \mathbf{u} - \mathbf{b}_j^{\mp}, \mathbf{n}_j^{\mp}\right),\ j = 1, \ldots, J$;
7 $\boldsymbol{\xi}_j^{\mp} \leftarrow \log\left(\mathcal{X}_j^{\mp}\bar{\mathcal{X}}^{-1}\right),\ j = 1, \ldots, J$ ;
8 $\mathbf{S} \leftarrow$ qr $\left(\boldsymbol{\xi}_j^{\mp}, \mathbf{b}_j^{\mp}, j = 1, \ldots, J, \mathbf{Q}\right)$;
  // see Appendix A for definition of qr
  // the notation $\mathbf{x}^{\mp}$ is used to denote
     the two variables $+\mathbf{x}$ and $-\mathbf{x}$
  Output: $\bar{\mathcal{X}}, \bar{\mathbf{b}}, \mathbf{S}$;
```

Algorithm 3: Update function for the Right-UKF-LG

```
  Input: $\bar{\mathcal{X}}, \bar{\mathbf{b}}, \mathbf{S}, \mathbf{Y}, \mathbf{N}$;
1 $\mathbf{Y}_0 = \mathbf{Y}\left(\bar{\mathcal{X}}, \mathbf{0}\right)$; // see (20) and (5)-(6)
2 $\begin{bmatrix}\boldsymbol{\xi}_j^{\mp} & \mathbf{b}_j^{\mp}\end{bmatrix} = \mp\gamma\,\mathrm{col}_j\left(\mathbf{S}\right),\ j = 1, \ldots, J'$;
3 $\mathcal{X}_j^{\mp} = \exp\left(\boldsymbol{\xi}_j^{\mp}\right)\bar{\mathcal{X}},\ j = 1, \ldots, J'$;
4 $\mathbf{Y}_j = \mathbf{Y}\left(\mathcal{X}_j^{\mp}, \mathbf{w}_j^{\mp}\right),\ j = 1, \ldots, J'$;
5 $\delta\bar{\boldsymbol{\xi}}, \delta\bar{\mathbf{b}}, \mathbf{S} \leftarrow$ qr′ $\left(\mathbf{Y}_n, \mathbf{Y}_0, \mathbf{Y}_j^{\mp}, \boldsymbol{\xi}_j^{\mp}, j = 1, \ldots, J', \mathbf{N}\right)$;
6 $\bar{\mathcal{X}} \leftarrow \exp\left(\delta\bar{\boldsymbol{\xi}}\right)\bar{\mathcal{X}}, \bar{\mathbf{b}} \leftarrow \bar{\mathbf{b}} + \delta\bar{\mathbf{b}}$ ;      // update mean
  // See Appendix A for definition of qr′
  Output: $\bar{\mathcal{X}}, \bar{\mathbf{b}}, \mathbf{S}$;
```

E. Update Step

The update step incorporates the observation of the p landmarks at time n and is described in Algorithm 3. It operates as follow. The sigma points generated in the Lie algebra at line 2 are mapped to the group through model (13) at line 3, and used to compute $2J' + 1$ measurement sigma points at line 4. The function qr′(.) then evaluates the updated Cholesky factors and the correction term $\left(\delta\bar{\boldsymbol{\xi}}, \delta\bar{\mathbf{b}}\right)$ used to update the mean state, along the lines of the conventional UKF methodology, and it is detailed in the Appendix A. Line 6 is the update of [12] as concerns the Lie group part of the state, and conventional update as concerns teh biases, see next subsection for more details.

Remark 3: following [29], the square-root implementation can add or remove landmarks, initializing landmark position as inverse depth point and allows computationally efficient propagation steps.

F. Discussion on the Final Update

Let $\bar{\mathcal{X}}$ denote the *propagated* mean, and $\mathbf{P} = \mathbf{S}\mathbf{S}^T$ the *propagated* covariance matrix of the state error after the propagation step, i.e. the outputs of Algorithm 2. According to model (13), it means the propagated state is described by

$$\mathcal{X} \approx \exp\left(\boldsymbol{\xi}\right)\bar{\mathcal{X}}, \ \ \boldsymbol{\xi} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{P}\right)$$

before measurement $\mathbf{Y}_n$. At the update step, the UKF methodology takes into account the observation $\mathbf{Y}_n$ to update the element $\boldsymbol{\xi} \in \mathbb{R}^{9+3p}$ as $\boldsymbol{\xi} \sim \mathcal{N}(\delta\bar{\boldsymbol{\xi}}, \mathbf{P}^+)$, i.e., $\boldsymbol{\xi} = \delta\bar{\boldsymbol{\xi}} + \boldsymbol{\xi}^+$ with $\boldsymbol{\xi}^+ \sim \mathcal{N}(\mathbf{0}, \mathbf{P}^+)$. Back to the Lie group this implies

$$\mathcal{X} \approx \exp\left(\boldsymbol{\xi}^+ + \delta\bar{\boldsymbol{\xi}}\right)\bar{\mathcal{X}}, \ \text{where } \boldsymbol{\xi}^+ \sim \mathcal{N}\left(\mathbf{0}, \mathbf{P}^+\right). \qquad (21)$$

Following [12], and assuming both correction $\delta\bar{\boldsymbol{\xi}}$ and uncertainty $\boldsymbol{\xi}^+$ in (21) are small, we end up with the following posterior that matches with the uncertainty representation (13):

$$\mathcal{X} \simeq \exp\left(\boldsymbol{\xi}^+\right)\bar{\mathcal{X}}^+, \ \text{where} \qquad (22)$$

$$\boldsymbol{\xi}^+ \sim \mathcal{N}\left(\mathbf{0}, \mathbf{P}^+\right), \qquad (23)$$

$$\bar{\mathcal{X}}^+ = \exp\left(\delta\bar{\boldsymbol{\xi}}\right)\bar{\mathcal{X}}. \qquad (24)$$

This approximation is based indeed upon the Baker-Campbell-Hausdorff (BCH) formula that states that $\exp\left(\boldsymbol{\xi}^+ + \delta\bar{\boldsymbol{\xi}}\right) = \exp\left(\boldsymbol{\xi}^+\right)\exp\left(\delta\bar{\boldsymbol{\xi}}\right) + O\left(\delta\bar{\boldsymbol{\xi}}^+, (\boldsymbol{\xi}^+)^2, \boldsymbol{\xi}^+\delta\bar{\boldsymbol{\xi}}\right)$.

Remark 4: when the correction terms are large and the BCH based approximation does not hold true, we propose an alternative method in Appendix B.

V. Simulation Results

Five different filters are compared on Monte-Carlo simulations:

- an UKF that considers the attitude as an element $SO(3)$ and the remaining variables as a vector space;
- the $SE(3)$-based UKF recently introduced in [5]. This filter is an UKF which encodes body attitude and position in $SE(3)$ and uses parallel transport associated to left-invariant vector fields of $SE(3)$;
- the Right-Invariant visual EKF SLAM (RIEKF) of [4,9] (where the biases are appended to the state and treated as in the conventional EKF) whose first numerical evaluation appears to our knowledge in the present paper;
- the proposed Right-UKF-LG described in Section IV;
- the proposed Left-UKF-LG, as an alternative to Right-UKF-LG based on the left uncertainty representation (12).

A. Simulation Setting

We generate a noise-free trajectory displayed on Figure 1. This trajectory is realistic since it is inspired by true quadrotor trajectories from [13]. Noises and slowly drifting small biases are added, and a standard deviation of 2 pixels is set for the observation noise. We define the number of landmarks in the state as $p = 60$ and at each observation, we observe a subset of 10 of these landmarks. 100 Monte Carlo simulations are then run.

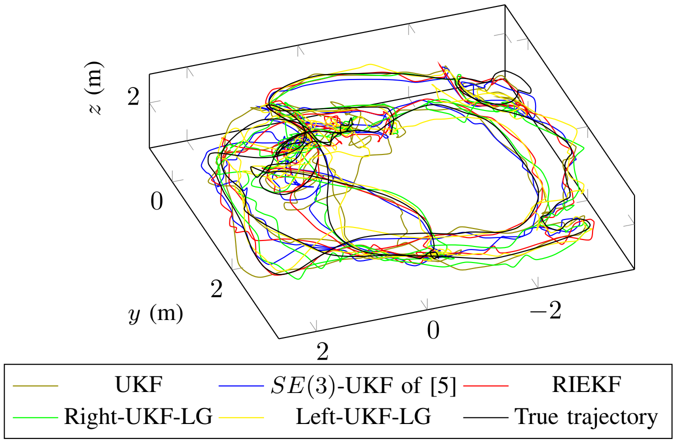

Fig. 1. Simulation trajectory used in Section V, and trajectories estimated by the various filters.

	$\mathbf{x}$ (cm)	$\mathbf{R}$ (°)
Conventional UKF	9.3	1.3
$SE(3)$-UKF of [5]	7.8	1.2
Left-UKF-LG	7.5	1.2
RIEKF	6.8	1.1
Right-UKF-LG	6.7	1.1

Fig. 2. Root Mean Squared Error averaged on 100 Monte Carlo simulations, on the body position and attitude, for the various filters, as described in Section V. The proposed Right-UKF-LG and RIEKF achieve the best results.

B. Results

The Root Mean Squared Error (RMSE) for the entire trajectory, averaged over 100 Monte Carlo runs, is displayed in Figure 2. From these results, we observe that:

- three groups appear: the RIEKF and Right-UKF-LG achieve the best results. This confirms that the right-invariant errors on $SE_{2+p}(3)$ are best suited to SLAM, as explained in [9,11]. Then, the Left-UKF-LG and the $SE(3)$-based UKF of [5] run second, and the conventional UKF runs last;
- The discrepancy between RIEKF and Right-UKF-LG is low at this noise level. Both algorithms are based on the right-invariant error on the Lie group $SE_{2+p}(3)$, but the first one uses the EKF methodology and the second one the UKF methodology.

VI. Experimental Results

To further validate then the two proposed filters (Right and Left UKF-LG) on real data, we evaluate them on the EuRoC dataset [13]. The five compared filters are the same as in the previous section. We selected five sequences in [13] in which landmarks can be well tracked in order to minimize the influence of the frontend image processor.

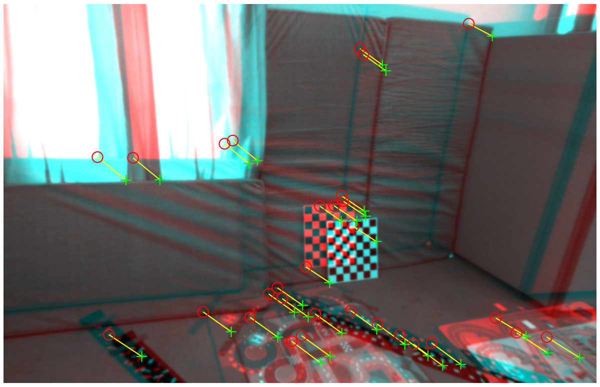

Fig. 3. Landmark tracking in the experiment. Green crosses are the current pixel locations of the landmarks and red circles are the pixel locations of the landmarks five images (i.e., 1 s) earlier. Picture comes from the EuRoC dataset [13].

A. Experimental Setting

Owing to the number of landmarks that keeps growing, the state may grow unboundedly and the filters become intractable for real time implementation. We thus propose to marginalize out landmarks that are not seen anymore, and add new landmarks to the state as they arrive, along the lines of [5]. This way, we conserve a constant number of 30 observed landmarks in the state, and the experimental results to come can be viewed as preliminary regarding our visual SLAM algorithm.

In our implementation, the filter tracks features via the KLT tracker using minimum eigenvalue feature detection [30] for its efficiency, and points are undistorted with the furnished camera parameters. The different filters are configured with the same parameters, where we set 2 pixels standard deviation for the landmark observations and IMU noise provided by [13]. The initial state corresponds to the ground truth.

B. Results

The different filters are thus launched on the real data and we plot the position errors with respect to ground truth for five experiments in Figure 4. On this set of experiments, we see that as for the previous section, two groups appear: the RIEKF and Right-UKF-LG achieve the best results. However, the differences between the approaches are smaller than in the simulation section. This is mainly due to the small time presence of each landmark in the state, such that the different error representations have less influence on the results. This preliminary experiment confirms the potential of Right-UKF-LG and RIEKF over conventional UKF and $SE(3)$-UKF.

C. Comparison of Execution Times

We compare in this section the execution time of the filters, both for the propagation and update steps. Figure 5 summarizes the results (frontend execution time is excluded). From this table, we observe that

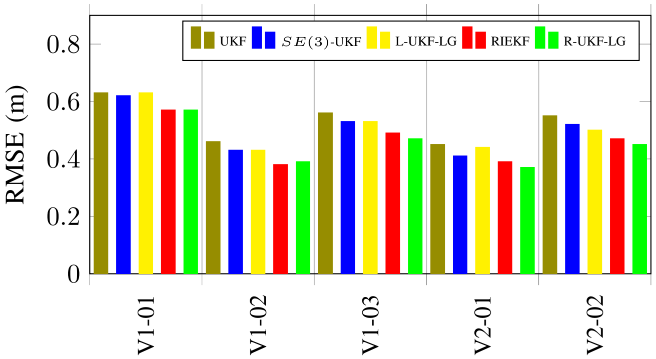

Fig. 4. Root Mean Squared Error on position with respect to ground trugh, on five different experiments.

	propagation (s)	update (s)
Conventional UKF	28	2.5
$SE(3)$-UKF of [5]	33	4.2
Left-UKF-LG	35	3.4
RIEKF	2.1	2.2
Right-UKF-LG	36	3.4

Fig. 5. Execution times of the different filters, both for the propagation and the update step. The indicated times correspond to the time execution of all sequences averaged for 10 s of flight.

- UKFs approaches require much more computational power than RIEKF during the propagation step. This is reinforced by the computation of logarithm at line 7 of Algorithm 2;
- the propagation necessitates much more calculus than the update for each UKF solutions, since the IMU (propagation) frequency (200 Hz) is ten times the camera (update) frequency (20 Hz);
- the differences between UKF-based approaches and RIEKF solutions for the update step appear as negligible compared to the propagation step.

Note that, the various UKFs' propagation step seems particularly long, owing in part to the non-optimal use of Matlab. However, an alternative solution we advocate for low powerful devices is merely to implement an hybrid R-UKF-LG that combines the RIEKF propagation and the R-UKF-LG update, in which we preserve the versatility (and fast prototyping benefits) of the Unscented approach with respect to the addition of other sensors' measurements (such as GPS) and/or variations in the measurement camera model. We implemented this solution, and we obtained similar results as the (full) R-UKF-LG on those datasets. Finally, in practice, we note that the front image processing is anyway generally much higher than the execution time of the filter.

VII. Conclusion

Two novel UKFs for data fusion of inertial sensors and monocular vision in the context of visual SLAM have been proposed. They build upon the very recent theory of UKF on Lie groups of [12], and have the merit of exploiting the full Lie group structure underlying the SLAM problem introduced in [9,23]. Another advantage is that the UKF approach spares the user the computation of Jacobians inherent to EKF implementation, and thus the proposed filters can be readily adapted to small modifications in the model, estimation of additional parameters, and/or addition of one or several sensors. Results from simulations and real experimental data have shown the relevance of the approach based on invariance, and notably the Right-UKF-LG. Future works will explore the theoretical consistency properties that the proposed Right-UKF-LG might possess along the lines of [9]. Regarding more practical aspects, sigma point selections [3] is also an interesting issue left for future work.

Appendix A

We give here the details of parameters and functions used for L-UKF-LG and R-UKF-LG. We set the scale parameters γ and γ' with the scaled unscented transform [31], such that they depend on the augmented covariance size $J = 27 + 3p$, $J' = 15 + 3p$ and

$$\gamma = \sqrt{J/(1 - W_0)}, W_0 = 1 - J/3, W_j = \frac{1 - W_0}{2J}, \quad (25)$$

$$\gamma' = \sqrt{J'/(1 - W_0')}, W_0' = 1 - J'/3, W_j' = \frac{1 - W_0'}{2J'}. \quad (26)$$

The function $\mathtt{qr}(.)$ operates as taking the $\mathcal{QR}$ decomposition of

$$\mathcal{QR} \leftarrow \sqrt{W_1} \begin{bmatrix} \boldsymbol{\xi}_1^+ & \cdots & \boldsymbol{\xi}_J^+ & \boldsymbol{\xi}_1^- & \cdots & \boldsymbol{\xi}_J^- \\ \mathbf{b}_1^+ & \cdots & \mathbf{b}_J^+ & \mathbf{b}_1^- & \cdots & \mathbf{b}_J^- \\ \mathbf{0} & \mathrm{chol}(\mathbf{Q}) & \mathbf{0} & -\mathrm{chol}(\mathbf{Q}) \end{bmatrix}, \quad (27)$$

from which we can extract the Cholesky factor as

$$\mathcal{R} = \begin{bmatrix} \mathbf{S} \\ \mathbf{0} \end{bmatrix}. \quad (28)$$

The function $\mathtt{qr}'(.)$ operates as follow: first, compute the mean measurement and weighted deviation

$$\bar{\mathbf{Y}} = W_0'\mathbf{Y}_0 + \sum_{j=1}^{J'} W_i' \left(\mathbf{Y}_j^+ + \mathbf{Y}_j^-\right), \quad (29)$$

$$\mathbf{e}_0 = \sqrt{|W_0'|}\left(\mathbf{Y}_0 - \bar{\mathbf{Y}}\right), \quad (30)$$

$$\mathbf{e}_j^{\mp} = \sqrt{W_j'}\left(\mathbf{Y}_j^{\mp} - \bar{\mathbf{Y}}\right), \ j = 1, \ldots, J', \quad (31)$$

and compute the Cholesky factors of the measurement covariance and the cross covariance as

$$\mathcal{QR} \leftarrow \begin{bmatrix} \mathbf{e}_1^+ & \cdots & \mathbf{e}_{J'}^+ & \mathbf{e}_1^- & \cdots & \mathbf{e}_{J'}^- & \mathbf{R}' \end{bmatrix}, \quad (32)$$

$$\mathcal{R} = \begin{bmatrix} \mathbf{S}' \\ \mathbf{0} \end{bmatrix}, \quad (33)$$

$$\mathbf{S}' \leftarrow \mathtt{CholUpdate}\left(\mathbf{S}', \mathrm{sign}\left(W_0'\right), \mathbf{e}_0\right), \quad (34)$$

$$\mathbf{P}' = \sum_{j=1}^{J'} \sqrt{W_j'} \left(\begin{bmatrix} \boldsymbol{\xi}_j^+ \\ \mathbf{b}_j^+ \end{bmatrix}^T \mathbf{e}_j^+ + \begin{bmatrix} \boldsymbol{\xi}_j^- \\ \mathbf{b}_j^- \end{bmatrix}^T \mathbf{e}_j^- \right), \quad (35)$$

$\mathbf{R}'$ is a block diagonal matrix containing p times the matrix $\mathrm{chol}(\mathbf{N})$ along its diagonal, and then compute gain, innovation and covariance as

$$\mathbf{K} = \mathbf{P}' \left(\mathbf{S}'^T\mathbf{S}'\right)^{-1} \quad (36)$$

$$= \mathbf{P}'\mathbf{S}'^{-1}\mathbf{S}'^{-T} \quad (37)$$

$$\begin{bmatrix} \delta\bar{\boldsymbol{\xi}} \\ \delta\bar{\mathbf{b}} \end{bmatrix} = \mathbf{K}\left(\mathbf{Y} - \bar{\mathbf{Y}}\right), \quad (38)$$

$$\mathbf{S} \leftarrow \mathtt{SeqCholUpdate}\left(\mathbf{S}, -1, \mathbf{K}\mathbf{S}'^T\right), \quad (39)$$

where $\mathtt{SeqCholUpdate}$ denotes repeated Cholesky updating $\mathtt{CholUpdate}$ using successive columns of $\mathbf{K}\mathbf{S}'^T$ as the updating vector [28]. To finally consider the Jacobian (see Section IV-F), we compute

$$\mathbf{S} \leftarrow \mathbf{S}\mathbf{J}^T, \quad (40)$$

letting $\mathbf{S}$ no longer triangular, but $\mathbf{S}$ keeps a valid matrix square root which could be used to define the next set of sigma points [29].

Remark 5: since we consider observation $\mathbf{Y}$ that lives in vector space, (38) is always valid and we do not have to compute any logarithm operation.

Appendix B

As concerns the update step, when the innovation $\delta\bar{\boldsymbol{\xi}}$ is important, we propose in the present paper to possibly use the more accurate approximation of [20,32]

$$\exp\left(\boldsymbol{\xi}^+ + \delta\bar{\boldsymbol{\xi}}\right) = \exp\left(\mathbf{J}\boldsymbol{\xi}^+\right)\exp\left(\delta\bar{\boldsymbol{\xi}}\right) + O\left(\boldsymbol{\xi}^+\right), \quad (41)$$

where $\mathbf{J}$ is the left Jacobian. In this case we compute the updated parameters as

$$\bar{\boldsymbol{\chi}}^+ = \exp\left(\delta\bar{\boldsymbol{\xi}}\right)\bar{\boldsymbol{\chi}}, \quad (42)$$

$$\mathbf{P}^+ \leftarrow \mathbf{J}\mathbf{P}^+\mathbf{J}^T, \quad (43)$$

When $\delta\bar{\boldsymbol{\xi}}$ remains small, $\mathbf{J} \approx \mathbf{I}$ such that we can discard $\mathbf{J}$ in (43) for computational efficiency, recovering the update [12].

References

[1] R. Mur-Artal, J. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[2] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Observability-constrained vision-aided inertial navigation," *University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Tech. Rep*, vol. 1, 2012.

[3] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "A quadratic-complexity observability-constrained unscented Kalman filter for SLAM," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1226–1243, 2013.

[4] K. Wu, T. Zhang, D. Su, S. Huang, and G. Dissanayake, "An Invariant-EKF VINS algorithm for improving consistency," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[5] G. Loianno, M. Watterson, and V. Kumar, "Visual inertial odometry for quadrotors on SE(3)," in *IEEE International Conference On Robotics and Automation (ICRA)*, pp. 1544–1551, IEEE, 2016.

[6] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision*, pp. 834–849, Springer, 2014.

[7] R. Mur-Artal and J. D. Tardos, "Visual-Inertial monocular SLAM with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[8] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

[9] A. Barrau and S. Bonnabel, "An EKF-SLAM algorithm with consistency properties," *arXiv:1510.06263*, 2015.

[10] A. Barrau and S. Bonnabel, "Invariant Kalman filtering," *Annual Reviews of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 237–257, 2018.

[11] T. Zhang, K. Wu, J. Song, S. Huang, and G. Dissanayake, "Convergence and consistency analysis for a 3-D Invariant-EKF SLAM," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 733–740, 2017.

[12] M. Brossard, S. Bonnabel, and J.-P. Condomines, "Unscented Kalman filtering on Lie groups," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[13] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. Achtelik, and R. Siegwart, "The EuRoC MAV datasets," *The International Journal of Robotics Research*, 2015.

[14] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[15] F. C. Park, J. E. Bobrow, and S. R. Ploen, "A lie group formulation of robot dynamics," *The International Journal of Robotics Research*, vol. 14, no. 6, pp. 609–618, 1995.

[16] F. Bullo and A. D. Lewis, *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems*, vol. 49. Springer Science & Business Media, 2004.

[17] Y. Wang and G. S. Chirikjian, "Error propagation on the euclidean group with applications to manipulator kinematics," *IEEE Transactions on Robotics*, vol. 22, pp. 591–602, Aug 2006.

[18] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.

[19] M. Zefran, V. Kumar, and C. Croke, "Metrics and connections for rigid-body kinematics," *International Journal of Robotics Research*, vol. 18, no. 2, pp. 243–258, 1999.

[20] G. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Applied and Numerical Harmonic Analysis, Birkhäuser Boston, 2011.

[21] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[22] S. Heo and C. G. Park, "Consistent ekf-based visual-inertial odometry on matrix lie group," *IEEE Sensors Journal*, vol. 18, no. 9, pp. 3780–3788, 2018.

[23] S. Bonnabel, "Symmetries in observer design: Review of some recent results and applications to ekf-based slam," in *Robot Motion and Control*, pp. 3–15, Springer, 2011.

[24] S. Hauberg, F. Lauze, and K. S. Pedersen, "Unscented kalman filtering on riemannian manifolds," *Journal of mathematical imaging and vision*, vol. 46, no. 1, pp. 103–120, 2013.

[25] D. Forsyth and J. Ponce, *Computer vision: a modern approach.* Upper Saddle River, NJ; London: Prentice Hall, 2011.

[26] A. Barrau and S. Bonnabel, "The Invariant Extended Kalman Filter as a stable observer," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2017.

[27] G. Bourmaud, R. Mégret, M. Arnaudon, and A. Giremus, "Continuous-discrete extended Kalman filter on matrix lie groups using concentrated Gaussian distributions," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 1, pp. 209–228, 2015.

[28] R. Van Der Merwe and E. A. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 6, pp. 3461–3464, 2001.

[29] S. A. Holmes, G. Klein, and D. W. Murray, "An $O(N^2)$ square root unscented Kalman filter for visual simultaneous localization and mapping," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 7, pp. 1251–1263, 2009.

[30] J. Shi *et al.*, "Good features to track," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593–600, IEEE, 1994.

[31] S. J. Julier, "The scaled unscented transformation," in *Proceedings of the American Control Conference*, vol. 6, pp. 4555–4559, IEEE, 2002.

[32] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.