remote sensing

Article

Lie Group Modelling for an EKF-Based Monocular SLAM Algorithm

Samy Labsir [1,2], Gaël Pages [2] and Damien Vivet [2,*]

[1] Thales LAS France, Rue Pierre Gilles de Gennes, 91470 Limours, France; samy.labsir@isae-supaero.fr
[2] Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-SUPAERO), University of Toulouse, 31400 Toulouse, France; gael.pages@isae-supaero.fr
* Correspondence: damien.vivet@isae-supaero.fr

Abstract: This paper addresses the problem of monocular Simultaneous Localization And Mapping on Lie groups using fiducial patterns. For that purpose, we propose a reformulation of the classical camera model as amodel on matrix Lie groups. Thus, we define an original-state vector containing the camera pose and the set of transformations from the world frame to each pattern, which constitutes the map's state. Each element of the map's state, as well as the camera pose, are intrinsically constrained to evolve on the matrix Lie group $SE(3)$. Filtering is then performed by an extended Kalman filter dedicated to matrix Lie groups to solve the visual SLAM process (LG-EKF-VSLAM). This algorithm has been evaluated in different scenarios based on simulated data as well as real data. The results show that the LG-EKF-VSLAM can improve the absolute position and orientation accuracy, compared to aclassical EKF visual SLAM (EKF-VSLAM).

Keywords: visual SLAM; coded patterns; camera model; Kalman filtering; optimization; Lie groups

check for updates

Citation: Labsir, S.; Pages, G.; Vivet, V. Lie Group Modelling for an EKF-Based Monocular SLAM Algorithm. *Remote Sens.* **2022**, *14*, 571. https://doi.org/10.3390/rs14030571

Academic Editor: Andrzej Stateczny

Received: 29 November 2021
Accepted: 20 January 2022
Published: 25 January 2022

1. Introduction

The problem of simultaneously localizing amobile robot and constructing amap from sensor measurements has arisen in many research works during the last decades [1,2]. Simultaneous Localization And Mapping (SLAM) can be seen as an estimation problem, where the state includes the pose of the robot and the position of the detected landmarks. This estimation problem is usually solved by one of the following two methods:

- the bundle adjustment approach: this formulates the SLAM as an optimization problem, where the set of all measurements (or a part of) is used to estimate the desired system state [3,4];
- the filtering approach: this is classically based on aKalman filter and its variants (such as the extended Kalman filter, i.e., EKF). It provides an estimate of the state and its associated covariance matrix at each epoch.

When avisual sensor is used for SLAM, the approach is usually referred to as visual-SLAM (VSLAM) [5]. In this context, the classical landmarks are provided by well-known corner-detection algorithms, such as Harris [6], FAST [7], SIFT, BRIEF, SURF, or ORB. The use of such 2D features admit two main disadvantages:

(1) the number of detected landmarks in each image can be high (≥500), making the map huge for big environments. This impacts directly the size of the state vector, which increases dramatically, making afilter-based method with 2D features not well-suited for long-term navigation;
(2) each measurement must be associated to each detection with adata association algorithm [8]. This step remains achallenge and false matching can cause the filter to diverge. Moreover, it can be difficult to treat it from acomputational point of view.

To overcome these difficulties, we propose in this study the use of fiducial or coded markers as landmarks [9–11]. The main advantage of such patterns is that they reduce the number of landmarks while solving the association step, thanks to labelled detections. Moreover, even if it is required to instrument the environment with coded patterns, a priori knowledge of the absolute position of one of the artificial landmarks would allow precise absolute localization.

For this study, the chosen patterns are constituted of aconstellation of circular shapes (see Figure 1) but can be changed for other rigid patterns such as ARTag [12] without loss of generality. By assuming that the patterns have aknown given size L, the pose of apattern in the global frame can be represented as a 3D affine transformation. The latter is composed of arotation and atranslation and can be associated to the set of matrices defined in the special Euclidean group $SE(3)$, which has the structure of aLie group (LG). The aim of this article is to rewrite the camera model in order to express the camera pose, as well as the pose of each landmark, in $SE(3)$ to reformulate the VSLAM problem as arecursive estimation problem on LGs.

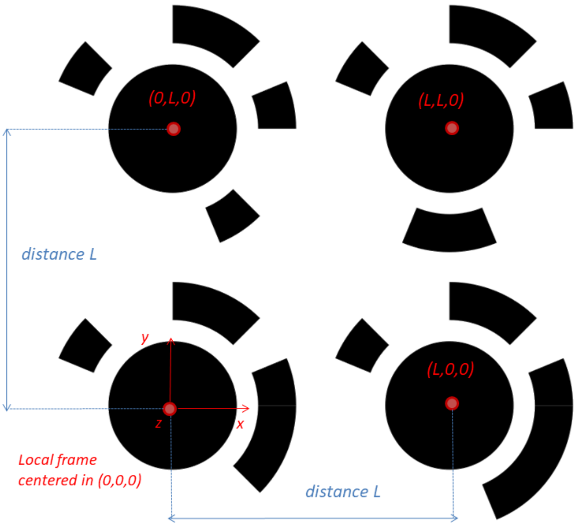

Figure 1. Representation of the coded pattern and its local frame. The detection algorithm detects the red points $\mathbf{q}_1 = [0,0,0]^\top$, $\mathbf{q}_2 = [0,L,0]^\top$, $\mathbf{q}_3 = [L,0,0]^\top$, and $\mathbf{q}_4 = [L,L,0]^\top$.

1.1. Formulating VSLAM on Lie Groups

The problem of filtering on LGs has recieved alot of attention in recent years, particularly in applications such as visual navigation and robotics to estimate the rotation and transformation matrices of arigid body [13,14]. The interest of the LG framework is to overcome acomplex Euclidean parametrization (as Euler angles or quaternions) by using aunified and elegant formalism, taking into account the geometric properties of these matrices. Consequently, the LG structure of the state vector admits advantages, compared to the classical VSLAM extended Kalman filter (EKF-VSLAM):

- We can rewrite acompact camera model, allowing us to overcome the high non-linearities created by rotation matrices parametrization;
- The analytical development of quantities of interest, such as Jacobian matrices, are intrinsic to LGs and are consequently less difficult to compute and to implement.

To apply filtering on LGs, several dedicated analytical algorithms have been developed in the literature, such as the Lie Group-Extended Kalman Filter (LG-EKF) [15], the Invariant Kalman filter [16,17], the Unscented Kalman filter (LG-UKF) [18], the information Kalman filter (LG-IKF) [19], but also Monte-Carlo filtering, such as particle filters [20]. In the context of SLAM, several works have been made. In [21], aVSLAM on LG is proposed and based on aLG-UKF. The unknown state is assumed to be lying on the LG $SE_{2+p}(3)$. In [22], aSLAM on LGs is developed that combines LG-EKF and LG-IKF. In [23], arobust hybrid VSLAM is proposed to detect the loop closures: the map and the pose are constructed in aEuclidean framework, but the loop closures are determined by estimating 3D map similarities with aLG modelling. All of these techniques have in common the assumption that the map is constructed by estimating Euclidean parameters (3D points).

In this study, we propose aLG-EKF-VSLAM which estimates both the state and the map on LGs. The advantages of this method are that it intrinsically takes into account the geometry of the pose, and it copes with the high non-linearities introduced by the camera model. Consequently, the proposed approach should theoretically improve the pose-estimation performance.

1.2. Contribution

Our contribution is two-fold:

- Theoretical: we define anew matrix state containing the camera pose and the pattern transformations on $SE(3)$, which constitutes the map. Consequently, the associated state space is $(SE(3))^{1+K}$ ($1+K$ direct products of $SE(3)$), where K is the number of current patterns;
- Algorithmic: the newly detected patterns are initialized by a$SE(3)$ transformation, obtained by minimizing acriterion through aGauss–Newton algorithm on LGs.

1.3. Organization

This paper is organized as follows: the EKF-VSLAM associated with the camera observation model in the context of coded patterns is detailed in Section 2. In Section 3, abackground on LGs is introduced, and the considered camera model is detailed and rewritten on LG $SE(3)$. The proposed LG-VSLAM, based on LG-EKF, is then explained in Section 4. Section 5 is dedicated to the performance evaluation. The proposed algorithm is implemented and assessed with simulated and real data and compared to aclassical Euclidean EKF-SLAM. Finally, conclusions and remarks are drawn in Section 6.

2. The Euclidean Approach

In this section, we present the classical Euclidean pinhole camera model. Particularly, we provide the different Euclidean parametrizations of unknown variables and the way the EKF-VSLAM is applied.

2.1. Coded Patterns

Consider amobile robot equipped with amonocular camera which tries to localize itself by using coded patterns spread out in its close environment. In our case, each coded pattern is a2D square surface, composed of four circles of the same size (see Figure 1). We define the center of the bottom left ellipse as the origin of the pattern's local frame. In this frame, the $3D$ coordinates $\{\mathbf{q}_i\}_{i=1}^4$ of the centers of each ellipse can be computed, knowing the inter-circle distance L. Consequently, every pattern j can be fully characterized by:

- arotation matrix between the local frame and aworld frame, $\mathbf{R}_{WP}^{(j)}$;
- the position of the origin of the local frame expressed in this world frame, $\mathbf{p}_{WP}^{(j)}$;
- the distance L between the center of two ellipses;
- an ID coded by the circular bar-codes.

2.2. Camera Observation Model

At each instant k, afiducial pattern detector provides the 2D coordinates of the center of each ellipse, $\mathbf{u}_{k,i}^{(j)}$, associated to the 3D point $\mathbf{q}_i$ of pattern j in the world frame. According to the pinhole camera model, the reprojection function provides the mathematical relation linking $\{\mathbf{u}_{k,i}^{(j)}\}_{i=1}^4$ and $\{\mathbf{q}_{i,C}^{(j)}\}_{i=1}^4$, such as:

$$\mathbf{u}_{k,i}^{(j)} = \frac{f}{(\mathbf{q}_{i,C}^{(j)})_3}\left[\nu_x\left(\mathbf{q}_{i,C}^{(j)}\right)_1 + x_0, \nu_y\left(\mathbf{q}_{i,C}^{(j)}\right)_2 + y_0\right] + \mathbf{n}_{k,i}^{(j)}, \tag{1}$$

with f being the focal distance of the camera, (x_0, y_0) being the principal point of the image and parameters ν_x and ν_y, the pixel densities on the horizontal and the vertical axes, respectively. $(\mathbf{v})_l$ denotes the l^{th} component of vector $\mathbf{v}$. The random process, $\mathbf{n}_{k,i}^{(j)}$, is considered as white Gaussian noise on $\mathbb{R}^2$ with covariance $\mathbf{N}$, $\mathbf{n}_{k,i}^{(j)} \sim \mathcal{N}_{\mathbb{R}^2}(\mathbf{0}, \mathbf{N})$. As illustrated in Figure 2, $\mathbf{q}_{i,C}^{(j)}$ explicitly depends on the rotation matrix $\mathbf{R}_{WC}^{(k)}$ and the rotation between the frame of the coded pattern and the world frame $\mathbf{R}_{WP}^{(j)}$. In aclassical Euclidean framework, they can be parametrized by:

- **Euler angles**: the unknown variables are six angles $\{\theta^{(k)}, \phi^{(k)}, \psi^{(k)}, \theta^{(j)} \phi^{(j)}, \psi^{(j)}\}$, such as:

$$\mathbf{R}_{WC}^{(k)} = \mathbf{R}_x(\theta^{(k)})\,\mathbf{R}_y(\phi^{(k)})\,\mathbf{R}_z(\psi^{(k)}) \tag{2}$$

$$\mathbf{R}_{WP}^{(j)} = \mathbf{R}_x(\theta^{(j)})\,\mathbf{R}_y(\phi^{(j)})\,\mathbf{R}_z(\psi^{(j)}), \tag{3}$$

  where $\mathbf{R}_u$ denotes an elementary 3D rotation according to the axis u;
- **Quaternion**: In this case $\mathbf{R}_{WC}^{(k)}$ and $\mathbf{R}_{WP}^{(j)}$ can be written as acomplex combination of quaternion coefficients with the form $\mathbf{z} = a + b\,\mathbf{i} + c\,\mathbf{j} + d\,\mathbf{k}$.

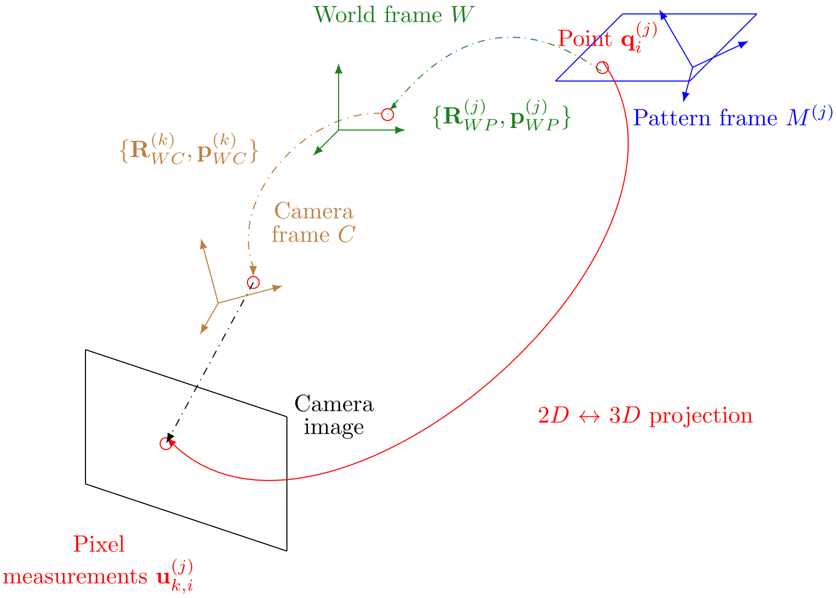
World frame W
Point $\mathbf{q}_i^{(j)}$
Pattern frame $M^{(j)}$
$\{\mathbf{R}_{WP}^{(j)}, \mathbf{p}_{WP}^{(j)}\}$
$\{\mathbf{R}_{WC}^{(k)}, \mathbf{p}_{WC}^{(k)}\}$
Camera frame C
Camera image
$2D \leftrightarrow 3D$ projection
Pixel measurements $\mathbf{u}_{k,i}^{(j)}$

Figure 2. Geometrical representation of the projection camera model.

Through this observation model and agiven dynamic model, it is possible to estimate the global state $\mathbf{x}_k$ containing:

- the pose of the camera $\left[\theta^{(k)}, \phi^{(k)}, \psi^{(k)}, \mathbf{p}_{WC}^{(k)}\right] \in \mathbb{R}^6$;
- the set of parameters of the patterns $\{\theta^{(j)}, \phi^{(j)}, \psi^{(j)}, \mathbf{p}_{WP}^{(j)}\}_{j=1}^{K} \in \mathbb{R}^{6K}$, where K is the number of detected patterns,

by performing aclassical EKF-VSLAM [24].

2.3. Why EKF-SLAM on Lie Groups?

Using the EKF-VSLAM with coded patterns and classical camera model has some drawbacks due to the high non-linearity of the observation model. Indeed, we observe that the unknown variables are linked to the observations by the composition of multiple inverse and trigonometric functions for the Euler parametrization. For the quaternion parametrization, the non-linearity is introduced by the composition of inverse and multiple square functions. Thus, the linearization in the update step of the EKF introduces errors. This can induce some numerical problems in the Jacobian calculation and degrade the performance of the estimation.

In the following, we propose to resolve the SLAM problem by performing the filtering on LGs. Indeed, this formalism enables us to overcome these problems. Particularly:

- the camera observation model can be written in acompact way and depends "linearly" on the rotation matrix. Consequently, the approximation at order 1 is more valid;
- The Jacobian of the observation model can be computed analytically by considering the geometrical structure of the rotation matrices and does not depend on any parametrization.

3. Background on Lie Groups

3.1. Elementary Notions

A LG G is aset which has the properties of a group and a smooth manifold [25,26]. In this work, we deal with matrix LGs; thus, every element of G is amatrix belonging to $\mathbb{R}^{n\times n}$.

→ Due to its group structure, G has an internal composition law which is associative. Furthermore, there is an identity element defined, so that each element of the group has an inverse;

→ Due to its manifold structure, it is possible to compute the derivative and the integral of two elements, or the inverse of one element.

Furthermore, the properties of Lie Groups allow us to define atangent vector space $T_{\mathbf{M}}G$ at every point $\mathbf{M} \in G$. It is then possible to associate any point of $T_{\mathbf{I}}G$ to any point of $T_{\mathbf{X}}G$, thanks to the operator called *left tangent application* (see Figure 3). The latter is defined, between two tangent spaces, by:

$$\begin{array}{cccc} L^{L}_{\mathbf{X}\,\mathbf{Y}} & : & T_{\mathbf{X}}G & \rightarrow & T_{\mathbf{Y}}G \\ & & \mathbf{V} & \mapsto & \mathbf{Y}\mathbf{X}^{-1}\,\mathbf{V} \end{array} \tag{4}$$

Consequently, we can work with the tangent space to the identity $\mathbf{I}_{n\times n}$, also called the Lie algebra, denoted as $\mathfrak{g}$. This space is avector space with adimension of m and defines the dimension of G as manifold.

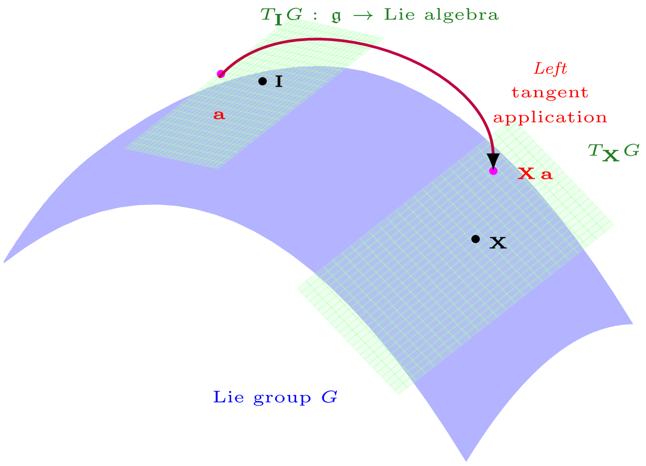

Figure 3. Geometrical relationship between $T_{\mathbf{I}}G$ and $T_{\mathbf{X}}G$ through the left tangent application.

We can establish arelation between G and $\mathfrak{g}$ through the exponential mapping, $\text{Exp}_G : \mathfrak{g} \rightarrow G$, and its reciprocal, the logarithm mapping, $\text{Log}_G : G \rightarrow \mathfrak{g}$. They are locally bijective. The mathematical expressions of these operators can be found in [25,27,28].

On the other hand, $\mathfrak{g}$ can be directly linked to the Euclidean space $\mathbb{R}^m$ through an isomorphism $[.]^{\wedge}_{G}$ and its inverse $[.]^{\vee}_{G}$, such as $\mathfrak{a} = [\mathbf{a}]^{\wedge}_{G}$ and $\mathbf{a} = [\mathfrak{a}]^{\vee}_{G}$, $\forall \mathbf{a} \in \mathbb{R}^m, \forall \mathfrak{a} \in \mathfrak{g}$. To condense these notations , we define $\mathbf{a} = \text{Log}^{\vee}_{G}(\mathbf{X}) = \left[\text{Log}_G(\mathbf{X})\right]^{\vee}$ and $\mathbf{X} = \text{Exp}^{\wedge}_{G}(\mathbf{a}) =$

$\text{Exp}_G\left([\mathbf{a}]_G^\wedge\right)$, with $\mathbf{X} \in G$ and $\mathbf{a} \in \mathbb{R}^m$. Figure 4 shows, geometrically, the relationship between aLG G, its Lie algebra $\mathfrak{g}$, and $\mathbb{R}^m$.

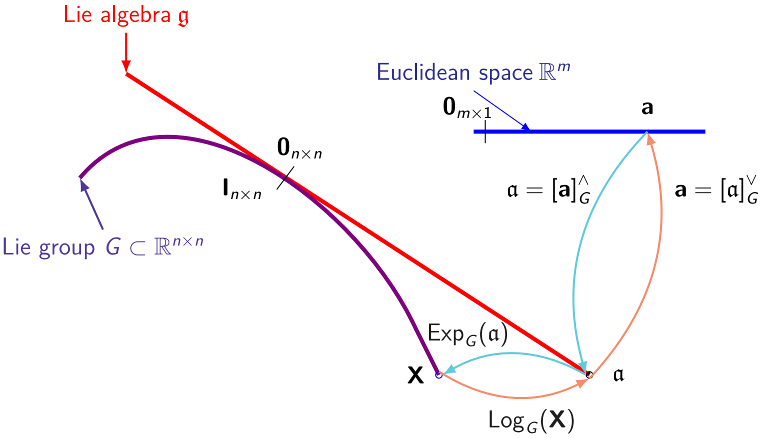
Lie algebra $\mathfrak{g}$
Euclidean space $\mathbb{R}^m$
$\mathbf{0}_{m\times 1}$
$\mathbf{a}$
$\mathbf{0}_{n\times n}$
$\mathbf{I}_{n\times n}$
$\mathfrak{a} = [\mathbf{a}]_G^\wedge$
$\mathbf{a} = [\mathfrak{a}]_G^\vee$
Lie group $G \subset \mathbb{R}^{n\times n}$
$\text{Exp}_G(\mathfrak{a})$
$\mathbf{X}$
$\mathfrak{a}$
$\text{Log}_G(\mathbf{X})$

Figure 4. Relation between the Lie group, the Lie algebra, and the Euclidean space. For each element $\mathbf{a} \in \mathbb{R}^m$, we associate an element $\mathfrak{a}$ in Lie algebra, which is converted in aelement $\mathbf{X}$ of LG G, thanks to the operator $\text{Exp}_G^\wedge(.)$.

3.2. Non-Commutativity and Jacobian

3.2.1. Non-Commutativity

In general, aLG is non-commutative and $\forall\,(\mathbf{a},\mathbf{b}) \in (\mathbb{R}^m)^2$:

$$\text{Log}_G^\vee(\text{Exp}_G^\wedge(\mathbf{a})\ \ \text{Exp}_G^\wedge(\mathbf{b})) \neq \mathbf{a} + \mathbf{b}. \tag{5}$$

The adjoint representation allows one to take into account the non-commutativity of LGs. It is an application $Ad_G(.) : G \to \mathbb{R}^{m\times m}$, such that $\forall \mathbf{X} \in G, \forall \mathbf{a} \in \mathbb{R}^m, \mathbf{X}\,\text{Exp}_G^\wedge(\mathbf{a}) = \text{Exp}_G^\wedge(Ad_G(\mathbf{X})\,\mathbf{a})\ \ \mathbf{X}$.

The non-commutativity of LGs can be also explicited through the Baker–Campbell–Hausdorff formula (BCH):

$$\text{Log}_G^\vee(\text{Exp}_G^\wedge(-\mathbf{a})\ \text{Exp}_G^\wedge(\mathbf{a}+\mathbf{b}))\ \ = \boldsymbol{\phi}_G(-\mathbf{a})\,\mathbf{b} + O(\|\mathbf{b}\|^2).,\tag{6}$$

and depends of the left Jacobian matrix, which is written:

$$\boldsymbol{\phi}_G(\mathbf{a}) = \sum_{n=0}^{+\infty} \frac{ad_G(\mathbf{a})^n}{(n+1)!},\tag{7}$$

where $ad_G : \mathbb{R}^m \to \mathbb{R}^{m\times m}$ is an adjoint operator characterizing the non-commutativity directly on the Lie algebra.

3.2.2. Jacobian on LGs

We can define the notion of Jacobian according to afunction. Let us consider $f : G \to \mathbb{R}$ an integrable function. We define LG Jacobian of f by the following expression:

$$D_{\mathbf{X}} f = \left.\frac{\partial f(\mathbf{X}\,\text{Exp}_G^\wedge(\delta)\ )}{\partial \delta}\right|_{\delta=\mathbf{0}},\tag{8}$$

which can be seen as ageneralization of the classical directional derivative.

3.3. Lie Groups of Interest

Two useful LGs in our application are introduced in this part.

3.3.1. The Special Orthogonal LG $SO(n)$

The LG $SO(n)$ is the group of rotation matrices of dimension n. Every element $\mathbf{R}$ of $SO(n)$ belongs to the following set:

$$SO(n) = \{\mathbf{R} \in \mathbb{R}^{n\times n} | \mathbf{R}\,\mathbf{R}^{\top} = \mathbf{I}, |\mathbf{R}| = 1\}, \tag{9}$$

where $|\mathbf{R}|$ is the determinant of matrix $\mathbf{R}$. Its Lie algebra is the set of skew-symmetric matrices:

$$\mathfrak{so}(n) = \{[\mathbf{w}]_\times | \mathbf{w} \in \mathbb{R}^m\}, \tag{10}$$

where $\mathbf{w}$ is the vector of $\mathbb{R}^m$ isomorph to the element of $[\mathbf{w}]_\times$. The notation $[.]_\times$ refers to the skew-symmetric matrix operator. In the case of $n = 3$, every element of $SO(3)$ is a3D rotation matrix, which can be classically parametrized in different ways (quaternions or Euler angles), as seen previously. With the LG formalism, we may parametrize it with an angle vector $\boldsymbol{\theta} \in \mathbb{R}^m$ $(m = 3)$. It is written, according to the Rodrigues formula [29], as:

$$\mathbf{R} = \mathbf{I}_{3\times3} + \frac{[\boldsymbol{\theta}]_\times}{\|\boldsymbol{\theta}\|}\sin(\|\boldsymbol{\theta}\|) + \frac{[\boldsymbol{\theta}]_\times^2}{\|\boldsymbol{\theta}\|^2}(1 - \cos(\|\boldsymbol{\theta}\|)). \tag{11}$$

Every element of its Lie algebra has the following structure:

$$[\boldsymbol{\theta}]_\times = \begin{bmatrix} 0 & -(\boldsymbol{\theta})_3 & (\boldsymbol{\theta})_2 \\ (\boldsymbol{\theta})_3 & 0 & -(\boldsymbol{\theta})_1 \\ -(\boldsymbol{\theta})_2 & (\boldsymbol{\theta})_1 & 0 \end{bmatrix}. \tag{12}$$

3.3.2. The Special Euclidean LG $SE(n)$

The LG $SE(n)$ corresponds to the semi-direct product between $SO(n)$ and $\mathbb{R}^n$:

$$SE(n) = \left\{ \mathbf{A} \ = \begin{bmatrix} \mathbf{R} & \mathbf{x} \\ \mathbf{0}_{1\times n} & 1 \end{bmatrix} \middle| \mathbf{R} \in SO(n), \mathbf{x} \in \mathbb{R}^n \right\}. \tag{13}$$

Its Lie algebra is:

$$\mathfrak{se}(n) = \left\{ \begin{bmatrix} [\mathbf{w}]_\times & \mathbf{u} \\ \mathbf{0}_{1\times n} & 0 \end{bmatrix} \middle| \mathbf{w} \in \mathbb{R}^{\frac{n\,(n-1)}{2}}, \mathbf{u} \in \mathbb{R}^n \right\}. \tag{14}$$

3.4. Camera Observation Model on LGs

The observation camera model (1) can be written in acompact way on the LG $SE(3)$. Indeed, the camera rotation $\mathbf{R}_{WC}^{(k)}$ and the associated translation $\mathbf{p}_{WC}^{(k)}$ constitute the rigid-body transformation of the camera and has the following form:

$$\mathbf{X}_{WC}^{(k)} = \begin{bmatrix} \mathbf{R}_{WC}^{(k)} & \mathbf{p}_{WC}^{(k)} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \in SE(3). \tag{15}$$

In asimilar way for the landmarks, the rotation matrix $\mathbf{R}_{WP}^{(j)}$ and the associated translation $\mathbf{p}_{WP}^{(j)}$ are written as:

$$\mathbf{X}_{WP}^{(j)} = \begin{bmatrix} \mathbf{R}_{WP}^{(j)} & \mathbf{p}_{WP}^{(j)} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \in SE(3). \tag{16}$$

According to Figure 2, we can express the successive $SE(3)$ transformations between the camera frame C and the pattern frame P in the following way:

$$\left[\left(\mathbf{q}_{i,W}^{(j)}\right)^{\top}, 1\right] = \mathbf{X}_{WP}^{(j)} \left[\mathbf{q}_i^{\top}, 1\right] \tag{17}$$

$$\left[\left(\mathbf{q}_{i,C}^{(j)}\right)^{\top}, 1\right]^{\top} = \left(\mathbf{X}_{WC}^{(k)}\right)^{-1} \left[\left(\mathbf{q}_{i,W}^{(j)}\right)^{\top}, 1\right]^{\top}. \tag{18}$$

Consequently, the pinhole model (1) can be written as:

$$\left[\left(\mathbf{u}_{k,i}^{(j)}\right)^{\top}, 1\right]^{\top} = \mathbf{h}(\mathbf{X}_{WC}^{(k)}, \mathbf{X}_{WP}^{(j)}) + \left[\left(\mathbf{n}_{k,i}^{(j)}\right)^{\top}, 0\right]^{\top}, \tag{19}$$

where:

$$\mathbf{h}(\mathbf{X}_{WC}^{(k)}, \mathbf{X}_{WP}^{(j)}) = \frac{\mathbf{K}\,\mathbf{I}_{3\times4}\left(\mathbf{X}_{WC}^{(k)}\right)^{-1}\mathbf{X}_{WP}^{(j)}\left[\mathbf{q}_i^{\top}, 1\right]^{\top}}{\left(\mathbf{I}_{3\times4}\left(\mathbf{X}_{WC}^{(k)}\right)^{-1}\mathbf{X}_{WP}^{(j)}\left[\mathbf{q}_i^{\top}, 1\right]^{\top}\right)_3}, \tag{20}$$

and $\mathbf{K}$ corresponds to the camera's calibration matrix:

$$\mathbf{K} = \begin{bmatrix} \nu_x f & 0 & x_0 \\ 0 & \nu_y f & y_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{21}$$

It is also interesting to note that the model can be written by substituting L:

$$\left[\left(\mathbf{u}_{k,i}^{(j)}\right)^{\top}, 1\right]^{\top} = \mathbf{h}^{L}(\mathbf{X}_{WC}^{(k)}, \mathbf{X}_{WP}^{(j)}; L) + \left[\left(\mathbf{n}_{k,i}^{(j)}\right)^{\top}, 0\right]^{\top}, \tag{22}$$

where:

$$\mathbf{h}^{L}(\mathbf{X}_{WC}^{(k)}, \mathbf{X}_{WP}^{(j)}, L) = \frac{\mathbf{K}\,\mathbf{I}_{3\times4}\left(\mathbf{X}_{WC}^{(k)}\right)^{-1}\mathbf{X}_{WP}^{(j)}\left[L\,\mathbf{e}_i, 1\right]^{\top}}{\left(\mathbf{I}_{3\times4}\left(\mathbf{X}_{WC}^{(k)}\right)^{-1}\mathbf{X}_{WP}^{(j)}\left[L\mathbf{e}_i, 1\right]^{\top}\right)_3} \tag{23}$$

$$\begin{cases} \mathbf{e}_1 = [0, 0, 0] & (24) \\ \mathbf{e}_2 = [1, 0, 0] & (25) \\ \mathbf{e}_3 = [0, 1, 0] & (26) \\ \mathbf{e}_4 = [1, 1, 0]. & (27) \end{cases}$$

This reformulation allows us to define an inverse problem for L. Thus, when the latter is unknown, it is also possible to estimate it conjointly with the camera pose.

4. SLAM Filtering Problem on Lie Groups

In the state-of-the-art VSLAM approaches (Euclidean or Lie groups), the aim is to obtain arecursive estimation of both the pose of the camera and the pose of each pattern in the world frame. The unknown state contains, at each instant, the pose parameters (on $SE(3)$ or on $\mathbb{R}^6$) and the set of 3D points on $\mathbb{R}^3$. Contrary to the Euclidean VSLAM approach, we propose, in the same way as [21], to define the pose parameters as a$SE(3)$ matrix, but instead of estimating the positions of the 3D points that compose each pattern, we take advantage of the coded patterns approach in order to estimate the $SE(3)$ transformation $\mathbf{X}_{WP}^{(j)}$. Consequently, if the pattern size is known, the estimation of this transformation provides directly the estimation of the 3D position of the center of each ellipse in that pattern. In this way, the global state belongs to several direct products of $SE(3)$. To achieve this estimation, we perform SLAM-extended Kalman filtering on LGs (LG-EKF-VSLAM) based on the LG-EKF proposed in [23]. Classically, on aEuclidean space, an extended Kalman filter is arecursive filtering algorithm which enables the approximation of the posterior distribution of parameters of interest by aGaussian distribution. The LG-EKF can be seen as ageneralization of the latter in the case where the parameters are lying on LG.

The main difference with the algorithm proposed in [23] is that we express the landmark initialization phase (mean and covariance) using aconcentrated Gaussian on LG.

4.1. Gaussian Distribution on LGs

To perform filtering on LGs, probability distributions on these manifolds have to be defined. Some have already been introduced in the literature, in particular for smooth manifolds [30,31]. We focus, in this paper, on the Gaussian distributions (CGD) on LGs, introduced in [32], which generalizes the concept of Euclidean Gaussian multivariate probability distribution functions (pdf). Indeed, they have similar properties, such as minimizing the entropy under some constraints.

It is possible to obtain arealization from aCGD thanks to aEuclidean Gaussian sample. Indeed, if $\boldsymbol{\epsilon}$ is aGaussian vector with covariance matrix $\mathbf{P}$, and $\boldsymbol{\mu}$ is amatrix defined on aLG G of intrinsic dimension m, then $\mathbf{X} = \boldsymbol{\mu}\,\mathrm{Exp}_G^{\wedge}(\boldsymbol{\epsilon})$ is distributed according to aleft-concentrated Gaussian pdf (CGD) (in asimilar way, we can define aright CGD where sample $\mathbf{X}$ can be written $\mathbf{X} = \mathrm{Exp}_G^{\wedge}(\boldsymbol{\epsilon})\ \boldsymbol{\mu}$) on G, with parameters $\boldsymbol{\mu} \in G$ and $\mathbf{P} \in \mathbb{R}^{m \times m}$. We note that $\mathbf{X} \sim \mathcal{N}_G^L(\mathbf{X}; \boldsymbol{\mu}, \mathbf{P})$. If $\boldsymbol{\mu}$ is close to $\mathbf{I}$, the expression of this pdf can be approximated by:

$$p(\mathbf{X}) \simeq \frac{1}{\sqrt{(2\,\pi)^m\,|\mathbf{P}|}}\ \exp^{-\frac{1}{2}\,||\mathrm{Log}_G^{\vee}(\boldsymbol{\mu}^{-1}\mathbf{X})||^2_{\mathbf{P}}}, \tag{28}$$

where $|\mathbf{P}|$ is the determinant of the matrix $\mathbf{P}$ and $\|.\|^2_{\mathbf{P}}$ is the Mahalanobis distance.

4.2. Unknown State and Evolution Model

The unknown state is constituted of the camera pose $\mathbf{X}_{WC}^{(k)} \in SE(3)$ and the transformation $\mathbf{X}_{WP}^{(j)}$ of every pattern j, $\forall j \in [\![1, m_k]\!]$, where m_k is the number of already-detected patterns from an initial instant to instant k.

Due to the dynamics of the robot, some other Euclidean parameters (for instance, the vehicle velocity or the ellipse size) may have to be integrated in the state. Thus, to keep the LG compact in form, such additional parameters can be added into avector $\mathbf{g}^{(k)} \in \mathbb{R}^p$. Thus, the global state (without landmarks) of our VSLAM problem can be written as:

$$\mathbf{X}_a^{(k)} = \left[\begin{array}{c|cc} \mathbf{X}_{WC}^{(k)} & \multicolumn{2}{c}{\mathbf{0}_{4\times(p+1)}} \\ \hline \mathbf{0}_{(p+1)\times(p+1)} & \mathbf{I}_{p\times p} & \mathbf{g}^{(k)} \\ & \mathbf{0}_{1\times p} & 1 \end{array}\right] \in SE(3) \times \mathbb{R}^p.$$

As aconsequence, the LG unknown global state $\mathbf{X}^{(k)}$, containing $\mathbf{X}_a^{(k)}$ and the set of pattern transformations, can be written as:

$$\mathbf{X}^{(k)} = \begin{bmatrix} \mathbf{X}_a^{(k)} & \mathbf{0}_{(5+p)\times 4} & \cdots & \mathbf{0}_{(5+p)\times 4} \\ \mathbf{0}_{4\times(5+p)} & \mathbf{X}_{WP}^{(1)} & & \mathbf{0}_{4\times 4} \\ \vdots & & \ddots & \vdots \\ \mathbf{0}_{4\times(5+p)} & \mathbf{0}_{4\times 4} & \cdots & \mathbf{X}_{WP}^{(m_k)} \end{bmatrix}, \tag{29}$$

with $\mathbf{X}^{(k)} \in G = SE(3) \times \mathbb{R}^p \times SE(3)^{m_k}$ being asquare matrix with size $d_k^X = 5 + p + 4\,m_k$.

To perform the filtering, we have to define the evolution model of this state. By assuming that the patterns are static, the global evolution model can be written as:

$$\boldsymbol{\Pi}(\mathbf{X}^{(k)}) {=} \boldsymbol{\Pi}(\mathbf{X}^{(k-1)})\ \mathrm{Exp}_{G'}^{\wedge}(\boldsymbol{\Omega}(\mathbf{X}^{(k-1)}, \mathbf{w}^{(k-1)}) {+} \mathbf{n}^{(k)})\ , \tag{30}$$

where $\boldsymbol{\Pi}$ is afunction which expresses the state on the LG $G' = SO(3) \times \mathbb{R}^3 \times \mathbb{R}^p \times SE(3)^{m_k}$ and defines two separated dynamic models on the rotation $\mathbf{R}_{WC}^{(k)}$ and the position $\mathbf{p}_{WC}^{(k)}$.

- $\mathbf{\Omega} : G' \rightarrow \mathbb{R}^{\dim(G')} = \left[\mathbf{\Omega}_{\mathbf{X}}(\mathbf{X}_a^{(k)})^\top, \mathbf{0}_{1\times 6\,m_k}\right]^\top$ is adynamic evolution function (where $\dim(G')$ corresponds to the dimension of G');
- $\mathbf{w}^{(k-1)}$ is acontrol input;
- $\mathbf{n}^{(k)} = \left[\left(\mathbf{n}_a^{(k)}\right)^\top, \mathbf{0}_{1\times 6\,m_k}\right]^\top$, where $\mathbf{n}_a^{(k)}$ is a white Gaussian noise on $\mathbb{R}^{\dim(G')}$, with covariance matrix $\mathbf{Q}_a^{(k)}$.

4.3. Principle of LG-EKF

The aim of the LG-EKF is to recursively approximate the posterior distribution of the unknown parameters defined on LG by aleft CGD (note that it also possible to define the LG-EKF-SLAM with aright CGD). Precisely, in our case, we want to approach $p(\mathbf{X}^{(k)}|\mathbf{u}_{(1:k)})$ by:

$$p(\mathbf{X}^{(k)}|\mathbf{u}_{(1:k)}) \simeq \mathcal{N}_G^L(\mathbf{X}^{(k)}; \widehat{\mathbf{X}}^{(k|k)}, \mathbf{P}^{(k|k)}), \tag{31}$$

where $\mathbf{u}_{(1:k)}$ is the set of all collected measurements from instant 1 to k. In the same way as the classical EKF, the estimation is done in two steps: prediction and update.

4.3.1. The Prediction Step

The objective of this step is to estimate the distribution $p(\mathbf{X}^{(k)}|\mathbf{u}_{(1:k-1)})$ (i.e., estimating the values $\{\widehat{\mathbf{X}}^{(k|k-1)}, \mathbf{P}^{(k|k-1)}\}$), given aprior on the distribution $p(\mathbf{X}^{(k-1)}|\mathbf{u}_{(1:k-1)})$ (defined by $\{\widehat{\mathbf{X}}^{(k-1|k-1)}, \mathbf{P}^{(k-1|k-1)}\}$). As $\mathbf{n}_a^{(k)}$ is zero-mean, the prediction of $\hat{\mathbf{X}}^{(k-1|k-1)}$ is given, thanks to the noise-free dynamic model (30) on LG, by:

$$\mathbf{\Pi}(\widehat{\mathbf{X}}^{(k|k-1)}) = \mathbf{\Pi}(\widehat{\mathbf{X}}^{(k-1|k-1)})\,\mathrm{Exp}_{G'}^{\wedge}(\mathbf{\Omega}(\widehat{\mathbf{X}}^{(k-1|k-1)})\,) \tag{32}$$

and $\mathbf{P}^{(k|k-1)}$ is given by:

$$\begin{aligned}\mathbf{P}^{(k|k-1)} &= \mathbf{F}^{(k)}\,\mathbf{P}^{(k-1|k-1)}\left(\mathbf{F}^{(k)}\right)^\top + \\ &\boldsymbol{\phi}_G(-\mathbf{\Omega}(\widehat{\mathbf{X}}^{(k-1|k-1)}))\,\mathbf{Q}^{(k)}\,\boldsymbol{\phi}_G(-\mathbf{\Omega}(\widehat{\mathbf{X}}^{(k-1|k-1)}))^\top,\end{aligned} \tag{33}$$

where:

$$\mathbf{F}^{(k)} = Ad_G(\mathrm{Exp}_G^{\wedge}(\mathbf{\Omega}(-\widehat{\mathbf{X}}^{(k-1|k-1)})\,) + \boldsymbol{\phi}_G(\mathbf{\Omega}(-\widehat{\mathbf{X}}^{(k-1|k-1)}))\,\mathbf{C}^{(k-1)} \tag{34}$$

$$\mathbf{C}^{(k-1)} = D_{\widehat{\mathbf{X}}^{(k-1|k-1)}}\mathbf{\Omega} \tag{35}$$

$$\mathbf{Q}^{(k)} = \begin{bmatrix}\mathbf{Q}_a^{(k)} & \mathbf{0}_{9\times 6\,m_k} \\ \mathbf{0}_{6\,m_k\times 9} & \mathbf{0}_{6\,m_k\times 6\,m_k}\end{bmatrix}. \tag{36}$$

4.3.2. The Update Step

The update step takes advantage of the camera observation $\mathbf{u}_{k,i}^{(j)}$ provided at instant k of the jth already-known pattern, in order to approximate:

$$p(\mathbf{X}^{(k)}|\mathbf{u}_{(1:k)}) \simeq \mathcal{N}_G^L(\mathbf{X}^{(k)}; \widehat{\mathbf{X}}^{(k|k)}, \mathbf{P}^{(k|k)}). \tag{37}$$

The parameters $\{\widehat{\mathbf{X}}^{(k|k)}, \mathbf{P}^{(k|k)}\}$ are written as:

$$\widehat{\mathbf{X}}^{(k|k)} = \widehat{\mathbf{X}}^{(k|k-1)}\,\mathrm{Exp}_G^{\wedge}(\mathbf{K}^{(k)}\,\mathbf{m}^{(k)})\,, \tag{38}$$

$$\mathbf{P}^{(k|k)} = \boldsymbol{\phi}_G(-\mathbf{K}^{(k)}\,\mathbf{m}^{(k)})\,\mathbf{P}^{(k|k,-)}\,\boldsymbol{\phi}_G(-\mathbf{K}^{(k)}\,\mathbf{m}^{(k)})^\top, \tag{39}$$

where:

$$\mathbf{m}^{(k)} = \overbrace{\left(\tilde{\mathbf{u}}_k^{(j)} - \tilde{\mathbf{h}}(\widehat{\mathbf{X}}_a^{(k|k-1)}, \widehat{\mathbf{X}}_{WP}^{(j)})\right)}^{\text{Innovation vector}} \tag{40}$$

$$\mathbf{K}^{(k)} = \mathbf{P}^{(k|k-1)} \left(\mathbf{J}^{(k)}\right)^\top \times \left(\mathbf{J}^{(k)}\, \mathbf{P}^{(k|k-1)} \left(\mathbf{J}^{(k)}\right)^\top + \tilde{\mathbf{N}}\right)^{-1} \tag{41}$$

$$\mathbf{P}^{(k|k,-)} = (\mathbf{I} - \mathbf{K}^{(k)}\, \mathbf{J}^{(k)})\, \mathbf{P}^{(k|k-1)}, \tag{42}$$

and:

- $$\tilde{\mathbf{N}} = \begin{bmatrix} \mathbf{N} & 0 \\ 0 & 0 \end{bmatrix} \tag{43}$$
- $$\tilde{\mathbf{u}}_k^{(j)} = \left[\tilde{\mathbf{u}}_{k,1}^{(j)\,\top}, \ldots, \tilde{\mathbf{u}}_{k,4}^{(j)\,\top}\right]^\top \tag{44}$$

  with $$\tilde{\mathbf{u}}_{k,i}^{(j)} = \left[\left(\mathbf{u}_{k,i}^{(j)}\right)^\top, 1\right]^\top \tag{45}$$
- $$\tilde{\mathbf{h}}(\widehat{\mathbf{X}}_a^{(k|k-1)}, \widehat{\mathbf{X}}_{WP}^{(j)}) = \underbrace{\left[\mathbf{h}(\widehat{\mathbf{X}}_a^{(k|k-1)}, \widehat{\mathbf{X}}_{WP}^{(j)})^\top, \ldots, \mathbf{h}(\widehat{\mathbf{X}}_a^{(k|k-1)}, \widehat{\mathbf{X}}_{WP}^{(j)})^\top\right]^\top}_{\text{repeated 4 times for the 4 measurements}} \tag{46}$$
- $$\mathbf{J}^{(k)} = \left[\mathbf{J}_c^{(k)} \quad \mathbf{0}_{2\times 6\,(j-1)} \quad \mathbf{J}_p^{(j)} \quad \mathbf{0}_{2\,\times 6\,(m_k-j)}\right] \odot \mathbf{I}_{4\times 1} \tag{47}$$

is the Jacobian matrix of $\tilde{\mathbf{h}}$, according to $\widehat{\mathbf{X}}^{(k|k-1)}$, with:

$$\mathbf{J}_c^{(k)} = D_{\widehat{\mathbf{X}}_a^{(k|k-1)}}\, \mathbf{h} \tag{48}$$

$$\mathbf{J}_p^{(j)} = D_{\widehat{\mathbf{X}}_{WP}^{(j)}}\, \mathbf{h}, \tag{49}$$

and $\odot$ defines the Kronecker product.

Remark 1. *It is important to note that if several patterns are detected at the same instant, the update step is recursively realized for each of them.*

4.4. Initialization of Pattern on $SE(3)$

If anew pattern is detected by the camera, the step is called the mapping step. As anew object $\mathbf{X}_{WP}^{(j)}$ is detected, the current state $\mathbf{X}^{(k)}$ is augmented as follows:

$$\mathbf{X}_p^{(k)} = \begin{bmatrix} \mathbf{X}^{(k)} & \mathbf{0}_{d_{k-1}^X \times 4} \\ \mathbf{0}_{4\times d_{k-1}^X} & \mathbf{X}_{WP}^{(j)} \end{bmatrix}. \tag{50}$$

Consequently, the associated covariance to this state can be written as:

$$\mathbf{P}_a^{(k|k)} = \begin{bmatrix} \mathbf{P}^{(k|k)} & \mathbf{P}_{CP}^{(k|k)} \\ \left(\mathbf{P}_{CP}^{(k|k)}\right)^\top & \mathbf{P}_{WP}^{(k|k)} \end{bmatrix}, \tag{51}$$

where:

- $\mathbf{P}_{CP}^{(k|k)}$ corresponds to the correlation between the new pattern and the previous state;

- $\mathbf{P}_{WP}^{(k|k)}$ is the covariance associated to the new pattern j.

At this step, we need to have an initial estimation of the pose of the pattern and of its associated covariance matrix. Classically, in aEuclidean framework, the initialization of apattern is determined by finding the solution of the inverse of the observation model, and the covariance matrix is approximated by computing correlations using associated Jacobians [33].

In the case of the camera model, this approach is not feasible due to its analytical complexity, and numeric approaches must be considered. For instance, it is possible to find the initial state by implementing alocal bundle adjustment, which optimizes acriterion built from the observation model. The covariance matrix of pattern j can be obtained by aGauss–Laplace approximation [34], and then the global covariance matrix of the new state is updated according to Equation (39). In this work, we propose an adaptation of this method, using the LG formalism.

The initialization of $\mathbf{X}_{WP}^{(j)}$ can be obtained by minimizing the anti-logarithm (opposite of the logarithm) of the posterior distribution $p(\mathbf{X}_{WP}^{(j)}|\{\mathbf{u}_{k,i}^{(j)}\}_{i=1}^{4}, \mathbf{u}_{(1:k-1)}, \widehat{\mathbf{X}}_{WC}^{(k|k-1)})$, associated to the measurement j. Due to its non-tractability, we propose approaching it by aleft CGD with parameters $\{\hat{\mathbf{X}}_{WP}, \mathbf{P}_{WP}\}$:

$$\begin{aligned} &p(\mathbf{X}_{WP}^{(j)}|\{\mathbf{u}_{k,i}^{(j)}\}_{i=1}^{4}, \mathbf{u}_{(1:k-1)}, \widehat{\mathbf{X}}_{WC}^{(k|k-1)}) \\ &\simeq \mathcal{N}_{SE(3)}^{L}(\mathbf{X}_{WP}^{(j)}; \widehat{\mathbf{X}}_{WP}, \mathbf{P}_{WP}), \end{aligned} \tag{52}$$

where $\widehat{\mathbf{X}}_{WP}$ is the estimated pose of the new pattern and $\mathbf{P}_{WP}$ its associated covariance.

4.4.1. Computation of $\widehat{\mathbf{X}}_{WP}$

$\widehat{\mathbf{X}}_{WP}$ can be determined by maximizing the posterior distribution (52). Due to the Bayes rule, we show that:

$$\begin{aligned} &p(\mathbf{X}_{WP}^{(j)}|\{\mathbf{u}_{k,i}^{(j)}\}_{i=1}^{4}, \mathbf{u}_{(1:k-1)}, \widehat{\mathbf{X}}_{WC}^{(k|k-1)}) \\ &\propto \prod_{i=1}^{4} p(\mathbf{u}_{k,i}^{(j)}|\widehat{\mathbf{X}}_{WC}^{(k|k-1)}, \mathbf{u}_{(1:k-1)}, \mathbf{X}_{WP}^{(j)}) \qquad (53) \\ &\simeq \prod_{i=1}^{4} \mathcal{N}_{\mathbb{R}^2}\left(\mathbf{u}_{k,i}^{(j)}; \mathbf{h}_p(\widehat{\mathbf{X}}_{WC}^{(k|k-1)}, \mathbf{X}_{WP}^{(j)}), \mathbf{N}\right), \qquad (54) \end{aligned}$$

where $\mathbf{h}_p$ is afunction conserving the two first components of $\mathbf{h}$. Consequently, we seek to find the minima of the following criterion corresponding to the anti-logarithm of (54):

$$\widehat{\mathbf{X}}_{WP} = \underset{\mathbf{X}_{WP} \in SE(3)}{\text{argmin}} \ \frac{1}{2} \|\boldsymbol{\phi}(\mathbf{X}_{WP})\|_{\boldsymbol{\Sigma}}^{2}, \tag{55}$$

where:

$$\begin{cases} [\boldsymbol{\phi}(\mathbf{X}_{WP})]_i = \mathbf{u}_{k,i}^{(j)} - \mathbf{h}_p(\widehat{\mathbf{X}}_{WC}^{(k|k-1)}, \mathbf{X}_{WP}) \quad \forall i \in \{1, \dots, 4\} & (56) \\ \boldsymbol{\Sigma} = \mathbf{N} \odot \mathbf{I}_{4\times 4} & (57) \end{cases}$$

and defines an optimization problem on $SE(3)$. A local minima is obtained thanks to aGauss–Newton algorithm on LG [35]. At each iteration l, the updated recursive formula is written as:

$$\mathbf{X}_{WP}^{(l)} = \mathbf{X}_{WP}^{(l-1)} \, \text{Exp}_{SE(3)}^{\wedge}(\delta^{(l)}) \,, \tag{58}$$

where $\delta^{(l)}$ is adescent direction, computed in the following way:

$$\delta^{(l)} = -\left(\left(\mathbf{J}_{\boldsymbol{\phi}}^{(l-1)}\right)^{\top} \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\boldsymbol{\phi}}^{(l-1)}\right)^{-1} \left(\mathbf{J}_{\boldsymbol{\phi}}^{(l-1)}\right)^{\top} \boldsymbol{\Sigma}^{-1} \boldsymbol{\phi}^{(l-1)}, \tag{59}$$

with:

$$\mathbf{J}_{\boldsymbol{\phi}}^{(l-1)} = D_{\mathbf{X}_{WP}^{(l-1)}} \boldsymbol{\phi} \tag{60}$$

$$\boldsymbol{\phi}^{(l-1)} = \boldsymbol{\phi}(\mathbf{X}_{WP}^{(l-1)}). \tag{61}$$

4.4.2. Computation of $\mathbf{P}_{WP}$

The initial covariance of $\mathbf{X}_{WP}$ is obtained by using aGauss–Laplace approximation on LG. It consists to approximate $\boldsymbol{\phi}$ around $\widehat{\mathbf{X}}_{WP}$ under the form:

$$\boldsymbol{\phi}(\mathbf{X}_{WP}) \simeq \boldsymbol{\phi}(\widehat{\mathbf{X}}_{WP}) + \mathbf{J}_{\boldsymbol{\phi}} \operatorname{Log}_{SE(3)}^{\vee}(\mathbf{X}_{WP}^{-1} \widehat{\mathbf{X}}_{WP}) \,. \tag{62}$$

As $\widehat{\mathbf{X}}_{WP}$ is acritical point of $\|\boldsymbol{\phi}(\mathbf{X})\|_{\boldsymbol{\Sigma}}^2$, we obtain:

$$p(\mathbf{X}_{WP}^{(j)} | \mathbf{u}_{(1:k-1)}, \widehat{\mathbf{X}}_{WC}^{(k|k-1)}) \simeq \tag{63}$$

$$\exp^{-\frac{1}{2} \operatorname{Log}_{SE(3)}^{\vee}(\mathbf{X}_{WP}^{-1} \widehat{\mathbf{X}}_{WP})^{\top} \mathbf{J}_{\boldsymbol{\phi}}^{\top} \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\boldsymbol{\phi}} \operatorname{Log}_{SE(3)}^{\vee}(\mathbf{X}_{WP}^{-1} \widehat{\mathbf{X}}_{WP})} \,. \tag{64}$$

By identification, with the expression of the CGD, it implies that $\mathbf{P}_{WP}$ is approximated by:

$$\mathbf{P}_{WP} = \mathbf{J}_{\boldsymbol{\phi}}^{\top} \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\boldsymbol{\phi}}. \tag{65}$$

To obtain the cross-correlation matrix $\mathbf{P}_{CP}^{(k|k)}$, as in (51), we perform an update step of the increased state (50) with the measurement $\mathbf{u}_{k,i}^{(j)}$.

5. Experiments

In this section, we compare our proposed approach, the LG-EKF-VSLAM, to the standard EKF-VSLAM algorithm while using coded patterns. We first carried out two Monte-Carlo simulations in order to evaluate the usefulness of performing LG modelling for monocular visual SLAM. In the first scenario, we considered that the distance between the center of two ellipse on the coded patterns, L, is perfectly known. For the second simulation, L is added to the state vector as aparameter to be estimated.

We then evaluated the LG-EKF-VSLAM with real data obtained within our laboratory, since, to the best of our knowledge, there are no available public benchmarks with coded patterns.

5.1. Experiments with Simulated Data

In the simulated scenarios, we considered awheeled robot following atrajectory generated according to aconstant velocity model. The control input is given by asimulated angular velocity $\mathbf{w}^{(k)}$. The unknown state is constituted of the camera pose $\mathbf{X}_{WC}^{(k)}$ and the linear velocity $\mathbf{v}^{(k)}$:

$$\mathbf{X}_a^{(k)} = \left[\begin{array}{c|cc} \mathbf{X}_{WC}^{(k)} & \multicolumn{2}{c}{\mathbf{0}_{4\times 4}} \\ \hline \mathbf{0}_{4\times 4} & \begin{array}{cc} \mathbf{I}_{3\times 3} & \mathbf{v}^{(k)} \\ \mathbf{0}_{1\times 3} & 1 \end{array} \end{array}\right] \in SE(3) \times \mathbb{R}^3.$$

The discrete dynamic model is written as:

$$\mathbf{v}^{(k)} = \mathbf{v}^{(k-1)} + \mathbf{n}_v\, \delta_t \tag{66}$$

$$\mathbf{p}_{WC}^{(k)} = \mathbf{p}_{WC}^{(k-1)} + \mathbf{v}^{(k-1)}\delta t + \mathbf{n}_p\, \delta_t \tag{67}$$

$$\mathbf{R}_{WC}^{(k)} = \mathbf{R}_{WC}^{(k-1)}\, \mathrm{Exp}_{SO(3)}^{\wedge}(\mathbf{w}^{(k-1)}\, \delta t + \mathbf{n}_r\, \delta_t)\ , \tag{68}$$

where δt is the sampling rate. $\mathbf{n}_v, \mathbf{n}_w$, $\mathbf{n}_p$, and $\mathbf{n}_r$ are four white Gaussian noises on linear and angular velocity, position, and orientation, respectively. These equations can be written on G with the form (30), where:

$$\boldsymbol{\Omega}_{\mathbf{X}}(\mathbf{X}_a^{(k-1)}) = \left[\left(\mathbf{w}^{(k-1)}\right)^{\top}\delta t, \left(\mathbf{v}^{(k-1)}\right)^{\top}\delta t, \mathbf{0}_{1\times 6\,m_k}\right]^{\top} \tag{69}$$

$$\mathbf{n}_a^{(k)} = \left[\mathbf{n}_r^{\top}\, \delta t, \mathbf{n}_p\, \delta t^{\top}, \mathbf{n}_v\, \delta t^{\top}\right]^{\top}. \tag{70}$$

The obtained trajectory is generated during $T = 885\,s$ with $\delta t = 1\,s$, and is shown in Figure 5. The positions of each coded pattern are defined with anull z-axis and are randomly distributed all along the ground-truth trajectory. Their orientations are obtained by sampling from aCGD on $SO(3)$, with mean $\mathbf{I}_3$ and covariance $0.2^2\,\mathbf{I}_3$.

The camera measurements are simulated according to the model (22), with:

$$\mathbf{K} = \begin{bmatrix} 200 & 0 & 240 \\ 0 & 200 & 320 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{N} = \sigma_n^2\, \mathbf{I}_2$$

$$\sigma_n = 10^{-1}\text{pixel}, \quad L = 5\,\text{m}.$$

The Jacobian matrices $\mathbf{C}^{(k-1)}$, $\mathbf{J}^{(k)}$, and $\boldsymbol{\phi}_G(.)$ of the LG-EKF-VSLAM (from Equations (7), (35), and (47), respectively) have been implemented according to the formulation given in Appendices A.1, B, and C.

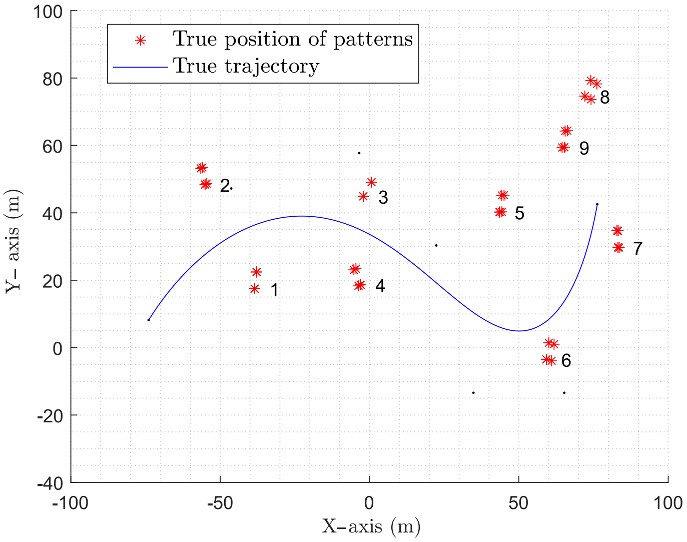

Figure 5. Simulated true trajectory and ellipse center positions of 9 coded labelled patterns.

The performance of the method must be evaluated on astate belonging to $SE(3)$. Consequently, we studied it by separating the estimator associated with the rotation matrix $\widehat{\mathbf{R}}_{WC}^{(k|k)}$ and the estimator associated with the position $\widehat{\mathbf{p}}_{WC}^{(k|k)}$.

(1) The quality of $\widehat{\mathbf{R}}_{WC}^{(k|k)}$ can be evaluated by using two intrinsic metrics on LGs, which enables us to compute the rotation error directly on $SO(3)$:

- aRMSE (Root Mean Square Error) metric, which can be seen as ageneralization of the classical Euclidean RMSE. The classical distance is substituted by the geodesic distance on $SO(3)$ [36]:

$$RMSE_R = \sqrt{\frac{1}{N_r\,T}\sum_{k=1}^{T}\sum_{i=1}^{Nr}\|\mathrm{Log}^{\vee}_{SO(3)}(\left(\mathbf{R}_{WC}^{(k)}\right)^{-1}\widehat{\mathbf{R}}_{WC}^{(k|k),i})\,\|^2}, \tag{71}$$

  where $\widehat{\mathbf{R}}^{(k|k),i}$ corresponds to the Monte-Carlo realization i of the rotation matrix estimator at epoch k. N_r is the number of realizations;

- amean RPE (Relative Pose Error) metric to evaluate the relative pose error between two consecutive epochs for several realizations of the algorithm.

$$RPE_R = \tag{72}$$

$$\frac{1}{N_r\,T}\sum_{k=1}^{T}\sum_{i=1}^{N_r}\Big|\|\mathrm{Log}^{\vee}_{SO(3)}(\left(\mathbf{R}_{WC}^{(k-1)}\right)^{-1}\mathbf{R}_{WC}^{(k)})\,\|-$$

$$\|\mathrm{Log}^{\vee}_{SO(3)}(\left(\widehat{\mathbf{R}}_{WC}^{(k-1|k-1),i}\right)^{-1}\widehat{\mathbf{R}}_{WC}^{(k|k),i})\,\|\Big|; \tag{73}$$

(2) As the position parameter $\mathbf{p}_{WC}^{(k)}$ is Euclidean, the quality of its estimator is determined with the classical Euclidean RPE and RMSE.

These performances are compared to aEuclidean EKF-VSLAM. In the latter, the rotation matrix is parametrized by Euler angles $\left[\theta^{(k)},\psi^{(k)},\phi^{(k)}\right]^{\top}$. Consequently, the unknown state is avector which belongs to $\mathbb{R}^{6+p+6\,m_k}$. The estimation error on rotation and position are directly evaluated by aclassical RPE and RMSE between the vector $\left[\theta^{(k)},\psi^{(k)},\phi^{(k)},\left(\mathbf{p}_{WC}^{(k)}\right)^{\top}\right]^{\top}$ and the estimated vector $\left[\widehat{\theta}^{(k)},\widehat{\psi}^{(k)},\widehat{\phi}^{(k)},\left(\widehat{\mathbf{p}}_{WC}^{(k)}\right)^{\top}\right]^{\top}$.

5.1.1. Case 1: L Perfectly Known

According to Table 1, we observe that the proposed method slightly outperforms the classical EKF-VSLAM. Two arguments can explain these results:

- Firstly, the LG modelling takes into account the intrinsic properties of the unknown state belonging to $SE(3)$. Consequently, the indeterminacy of the Euler angle estimates is deleted;
- Secondly, through this modelling, the observation model is freed from the non-linearity introduced by the cosine and sine functions of the rotation matrix in the Euclidean modelling. In this way, the approximation of the covariance estimation error is more precise.

Table 1. Simulation results with known L. Obtained performance estimation for $N_r = 500$ realizations.

	LG-EKF-VSLAM	EKF-VSLAM
RMSE position (m)	0.298	0.427
RMSE orientation (rad)	0.0048	0.0055
RPE position (m)	0.0172	0.0205
RPE orientation (rad)	5.12×10^{-4}	6×10^{-4}

5.1.2. Case 2: L Is Misspecified

In order to challenge the proposed algorithm, we assume that the distance between two ellipses of apattern is incorrectly specified. Consequently, L is initialized with an initial error on the true value (5 m) equal to 1 m, and added to the state:

$$\begin{bmatrix} \mathbf{X}_a^{(k)} & 0 \\ 0 & L \end{bmatrix} \in SE(3) \times \mathbb{R}^4. \tag{74}$$

In order to estimate L with the LG-EKF-VSLAM, the observation model that explicitly involves L is considered. From anumerical point of view, we also have to compute the Jacobian of the observation model according to L.

From the obtained results given in Table 2 and Figure 6, we can observe that the LG-EKF-VSLAM achieves the best results. Indeed, compared to the EKF-VSLAM, it is able to asymptotically converge to the true value of L, in spite of high initialization errors (here, 1 m). Indeed, the estimation error becomes inferior to 0.1 m. Moreover, the precision of the estimator, after convergence, is better than for the Euclidean EKF-VSLAM (see Table 2).

Table 2. Simulation results with misspecified L. Obtained performance estimation for $N_r = 500$ realizations.

	LG-EKF-VSLAM	**EKF-VSLAM**
RMSE position (m)	0.758	1.257
RMSE orientation (rad)	0.0158	0.065
RPE position (m)	0.0398	0.0924
RPE orientation (rad)	7.2×10^{-4}	7.7×10^{-4}

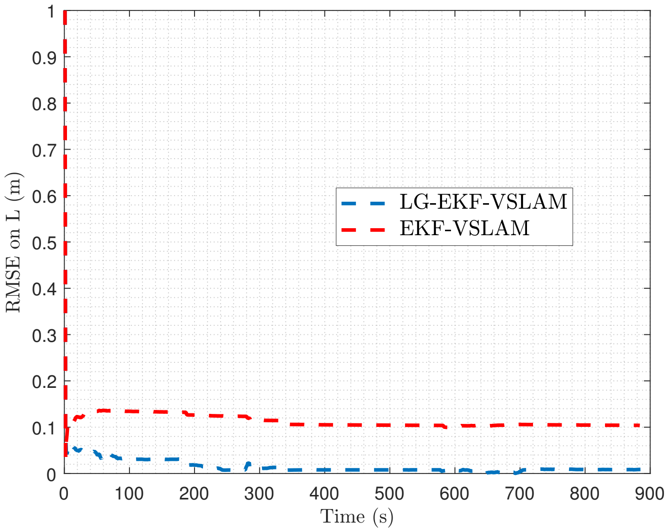

Figure 6. Evolution of the RMSE on L for $Nr = 500$ realizations.

5.2. Experiment with Real Data

5.2.1. Experiment Setup

To validate the methodology presented in the previous section (Section 4), we performed an experiment with amonocular camera (PointGrey Blackfly from Flir @7fps), with a3.8–13-mm Varifocal lens, embedded on amobile cart. Additionally, an Optitrack motion-capture system provided the ground truth for the pose of the camera and the coded patterns. The camera's reference trajectory and the true positions of the coded patterns (i.e., the center of the bottom-left ellipse) are represented in Figure 8. The camera's calibration matrix is given by:

$$\mathbf{K} = \begin{bmatrix} 1250.94 & 0 & 667.14 \\ 0 & 1251.77 & 532.89 \\ 0 & 0 & 1 \end{bmatrix}. \tag{75}$$

In this experiment, 12 coded patterns, each printed on a$A3$ paper, with $L = 0.1415$ m, were placed in different places of the room, as illustrated in Figures 7 and 8. The camera moved in acircular motion and the fiducial pattern-detection algorithm provided observations for each frame. The state $\mathbf{X}_a^{(k)}$ contains the pose, but also the linear and angular velocity $\mathbf{w}^{(k)}$. Indeed, in our experiment, no angular sensor is available; thus, we used aconstant velocity model. Consequently, we have:

$$\mathbf{X}_a^{(k)} = \left[\begin{array}{c|c|c} \mathbf{X}_{WC}^{(k)} & \mathbf{0}_{4\times4} & \mathbf{0}_{4\times4} \\ \hline \mathbf{0}_{4\times4} & \begin{array}{cc}\mathbf{I}_{3\times3} & \mathbf{v}^{(k)} \\ \mathbf{0}_{1\times3} & 1\end{array} & \mathbf{0}_{4\times4} \\ \hline \mathbf{0}_{4\times4} & \mathbf{0}_{4\times4} & \begin{array}{cc}\mathbf{I}_{3\times3} & \mathbf{w}^{(k)} \\ \mathbf{0}_{1\times3} & 1\end{array} \end{array}\right] \in SE(3) \times \mathbb{R}^6. \tag{76}$$

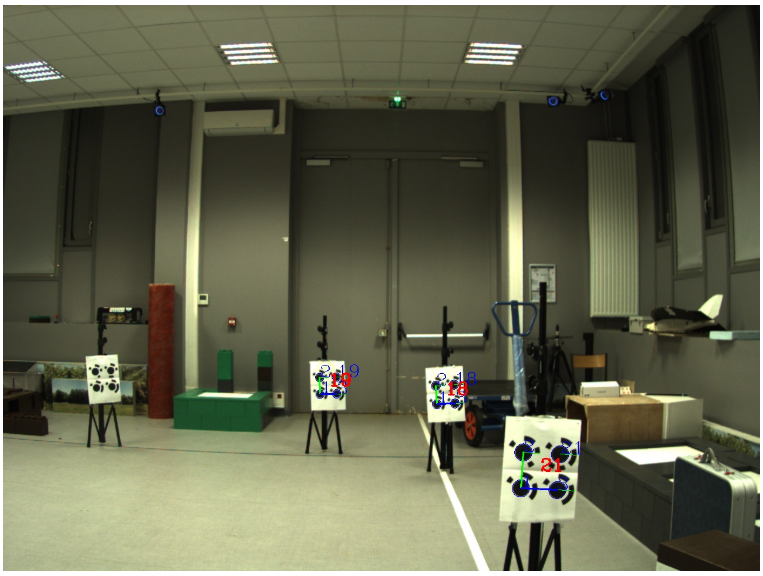

Figure 7. First image frame integrated in the detector: it computes the ellipse coordinates of the pattern with labels 18, 19, and 21.

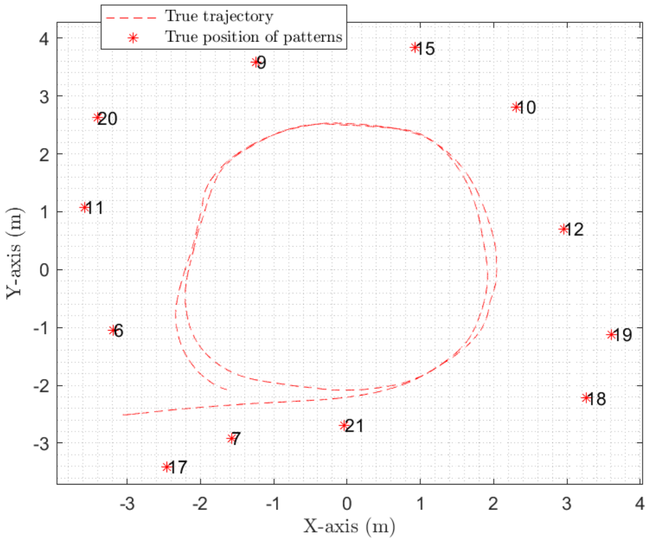

Figure 8. Ground truth obtained by an Optitrack motion capture system in the XY–plane, superimposed to the position of the center of the bottom-left ellipse of each coded pattern.

To take into account the circular motion of the mobile in the filter, we propose leveraging aposition-prediction model correlating its position and orientation. Indeed, the correlations between these two variables are especially high in the turns.

The proposed model has the following structure:

$$\mathbf{p}_{WC}^{(k)} = \mathbf{p}_{WC}^{(k-1)} + \mathbf{R}_{WC}^{(k-1)} \left[\left(\mathbf{v}^{(k-1)} \right)_1, 0, 0 \right]^\top \delta t + \mathbf{n}_p \, \delta_t. \tag{77}$$

The translation velocity and rotation matrix are assumed to follow the evolution models (66) and (68). Moreover, we suppose that the angular velocity follows arandom walk process with aGaussian white noise $\mathbf{n}_w$. Consequently, the function $\mathbf{\Omega_X}(\mathbf{X}_a^{(k-1)})$ is written as:

$$\mathbf{\Omega_X}(\mathbf{X}_a^{(k-1)}) = \left[(\mathbf{w}^{(k-1)})^\top \delta t, \left(\mathbf{R}_{WC}^{(k-1)} \left[\left(\mathbf{v}^{(k-1)} \right)_1, 0, 0 \right]^\top \right)^\top \delta t, \mathbf{0}_{1\times 6} \right]^\top. \tag{78}$$

The computation of the Jacobian of $\mathbf{\Omega_X}$ is provided in Appendix A.2.

5.2.2. Obtained Results

In order to use the method with real data, we first have to define the noise parameters. Concerning the observation model, the covariance noise is taken to be sufficiently high in order to take into account the uncertainties introduced by non-analytical phenomenons. Consequently, we take $\mathbf{N} = \sigma_n^2 \mathbf{I}_3$ with $\sigma_n = 2$ pixels. Concerning the dynamic model, the standard deviation of noises $\mathbf{n}_r$, $\mathbf{n}_p$, $\mathbf{n}_v$, and $\mathbf{n}_w$ are, respectively, fixed at $\sigma_p = \frac{10^{-4}}{\delta_t}$ m.s^{-1} $\sigma_r = \frac{10^{-2}}{\delta_t}$ rad.s^{-2}, $\sigma_v = \frac{10^{-6}}{\delta_t}$ m.s^{-2}, and $\sigma_w = \frac{10^{-4}}{\delta_t}$ rad.s^{-2}.

In order to quantify the estimation error of the trajectory, we use the Optitrack reference system and compare it with the estimated trajectory given by each of the two methods. This is done by computing the absolute error on the position at each instant of the trajectory, as represented in Figure 9.

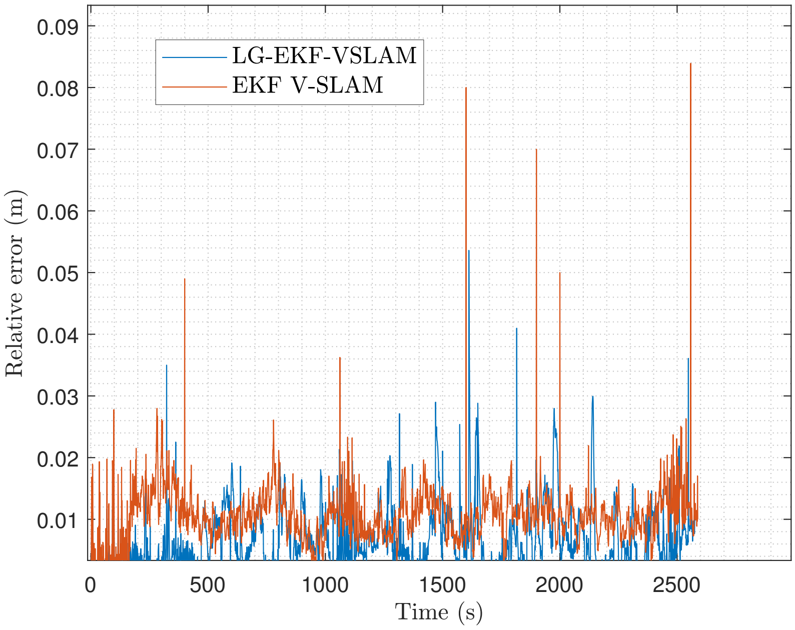

Figure 9. Absolute position error obtained for the two methods.

In the two approaches, we first observe, through the Figures 10 and 11, that the estimated position is particularly degraded in some parts of the trajectory, especially between the coded patterns 9 and 11 and 6 and 7. This is due to the pattern detector, which does not provide any detection because of blur and specular reflection on the objects. Consequently, during these instants, the estimation is only realized by dead-reckoning navigation, and the non-linearities of the models increases the pose-estimation error. In Figure 9, we observe that the performance of the two approaches are quite similar in terms of absolute error estimation, even if the LG-EKF-VSLAM performs alittle bit better all over the trajectory. This is confirmed through Figure 11. Indeed, we observe that during the last turn of the trajectory, the EKF-VSLAM tends to deviate from the ground truth, implying ahigher estimation error.

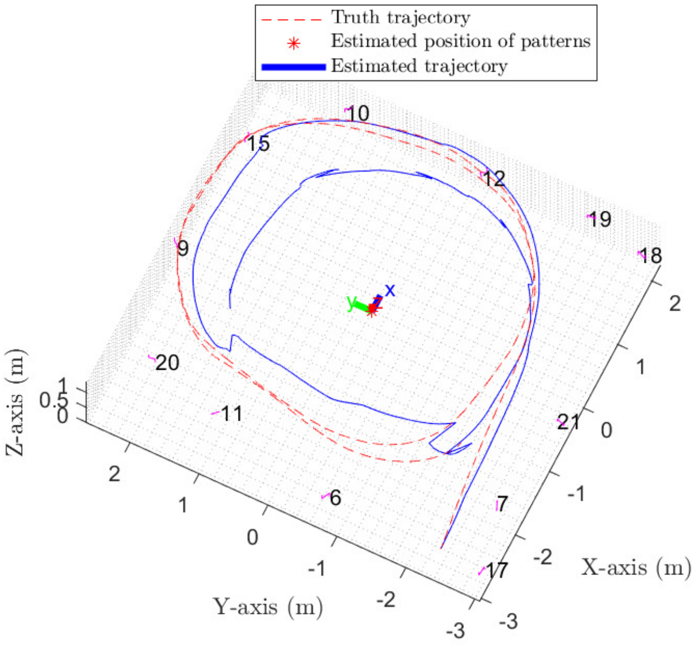

Figure 10. Estimated trajectory for **EKF–VSLAM**.

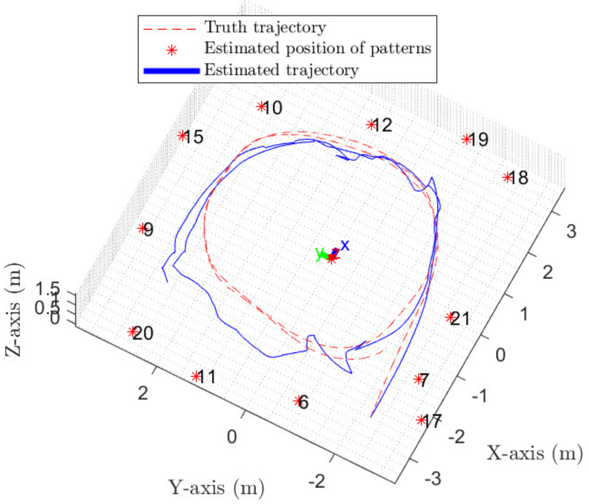

Figure 11. Estimated trajectory for **LG–EKF–VSLAM**.

6. Conclusions

In this paper, we propose an original modelling on LGs in order to perform monocular VSLAM. After reformulating the classical pinhole on LGs, we propose aEKF-VSLAM algorithm on LG for coded-pattern landmarks. We take into account the particularity of the SLAM problem by proposing aLG method to initialize new patterns. Our algorithm has been compared with aclassical EKF-VSLAM using aEuler parametrization on simulated data as well as on real-world data. Both the simulation and the experiments showed the validity and the performance improvement provided by the proposed LG-EKF-VSLAM, compared to the state-of-the-art solution. In addition, we have shown that amisspecified value on the size of the coded pattern has alarge impact on the estimation solution. When this parameter is taken into account as anew state in the estimation process, the EKF-VSLAM reduces the estimation error, but is outperformed by the LG-EKF-VSLAM, which converges asymptotically to the true value of L. This is mostly due to the LG formalism, which copes better with system non-linearities.

A pertinent perspective would be to propose aLG-based approach by taking into account the lack of knowledge of non-estimated parameters models. In our case, L could be considered uncertain, but not estimated. There exist Euclidean filtering approaches dedicated to this problem [37]; consequently, it would be pertinent to adapt them for parameters lying on LGs.

Moreover, in ageneral way, it would be interesting to adapt the formalism of LG (especially the optimization method) to other applications in computer vision and signal processing. For instance, we can quote the well-known problems of Structure of Motion (SfM) [38] and image registration [39].

From an application point of view, another perspective would be to fuse the camera measurements with, for instance, information from Ultra-Wide-Band sensors or with a Global Navigation Satellite System for outdoor navigation, in order to improve the performance of the algorithm, especially in situations where camera frames can be lost or where no patterns are present in the close environment of the vehicle.

Author Contributions: Conceptualization, S.L. and D.V.; methodology, S.L.; S.L. designed the models; writing—original draft preparation, S.L.; writing—review and editing, S.L., G.P. and D.V.; experiments were designed by G.P. and D.V., and were equally realized with S.L.; supervision, G.P. and D.V.; project administration, G.P.; funding acquisition, G.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the DGA/AID (French Secretary of Defense) under grant n° 2018.60.0072.00.470.75.01.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request to the corresponding author. The data belong to ISAE-SUPAERO and are not publicly available.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A. Computation of $C^{(k-1)}$

It should be remembered that:

$$\mathbf{C}^{(k-1)} = \left. \frac{\partial\, \boldsymbol{\Omega}(\hat{\mathbf{X}}^{(k-1|k-1)}\ \mathrm{Exp}_G^{\wedge}(\delta)\ )}{\partial \delta} \right|_{\delta=\mathbf{0}} . \tag{A1}$$

Appendix A.1. Case of $\Omega_{\mathbf{X}}\left(\mathbf{X}_a^{(k)}\right)$ from (69)

We know that $\Omega_{\mathbf{X}} : G \to \mathbb{R}^9$ is defined by:

$$\boldsymbol{\Omega}_{\mathbf{X}}(\mathbf{X}^{(k-1)}) = \left[\left(\mathbf{w}^{(k-1)}\right)^\top \delta t, \left(\mathbf{v}^{(k-1)}\right)^\top \delta t, \mathbf{0}_{1\times 3} \right]^\top . \tag{A2}$$

Consequently, the Jacobian of $\boldsymbol{\Omega}$, according to $\mathbf{X}^{(k-1)}$, is written as:

$$\mathbf{C}^{(k-1)} = \left[\mathbf{J}_R, \mathbf{J}_p, \mathbf{J}_v, \mathbf{0}_{9\times 6\, m_{k-1}}\right] \in \mathbb{R}^{9\times 9}, \tag{A3}$$

with:

$$\mathbf{J}_R = \left.\frac{\partial\, \boldsymbol{\Omega}_{\mathbf{X}}(\hat{\mathbf{R}}_{WC}^{(k-1|k-1)}\, \mathrm{Exp}^{\wedge}_{SO(3)}(\delta_R)\,, ., .}{\partial\, \delta_R}\right|_{\delta_R=\mathbf{0}} \tag{A4}$$

$$\mathbf{J}_p = \left.\frac{\partial\, \boldsymbol{\Omega}_{\mathbf{X}}(., \hat{\mathbf{p}}_{WC}^{(k-1|k-1)} + \delta_p, .)}{\partial\, \delta_p}\right|_{\delta_p=\mathbf{0}} \tag{A5}$$

$$\mathbf{J}_v = \left.\frac{\partial\, \boldsymbol{\Omega}_{\mathbf{X}}(., ., \hat{\mathbf{v}}^{(k-1|k-1)} + \delta_v)}{\partial\, \delta_v}\right|_{\delta_v=\mathbf{0}}. \tag{A6}$$

Due to the structure of $\boldsymbol{\Omega}_{\mathbf{X}}$, we show that:

$$\mathbf{J}_R = \mathbf{0}_{9\times 3} \tag{A7}$$

$$\mathbf{J}_p = \mathbf{0}_{9\times 3} \tag{A8}$$

$$\mathbf{J}_v = \left[\mathbf{0}_{6\times 3}^{\top}, \delta t\, \mathbf{I}_{3\times 3}\right]^{\top}. \tag{A9}$$

Appendix A.2. Case of $\Omega_{\mathbf{X}}\left(\mathbf{X}_a^{(k)}\right)$ *from* (78)

We know that $\boldsymbol{\Omega}_{\mathbf{X}} : G \to \mathbb{R}^{12}$ is defined by:

$$\Omega_{\mathbf{X}}(\mathbf{X}_a^{(k-1)}) = \left[(\mathbf{w}^{(k-1)})^{\top}, \left(\mathbf{R}^{(k-1)}\left[\left(\mathbf{v}^{(k-1)}\right)_1, 0, 0\right]^{\top}\right)^{\top} \delta t, \mathbf{0}_{1\times 6}\right]^{\top}. \tag{A10}$$

The Jacobian has the following structure:

$$\mathbf{C}^{(k-1)} = \left[\mathbf{J}_R, \mathbf{J}_p, \mathbf{J}_v, \mathbf{J}_w, \mathbf{0}_{12\times 6\, m_{k-1}}\right] \in \mathbb{R}^{12\times 12}, \tag{A11}$$

where $\mathbf{J}_R, \mathbf{J}_p$, and $\mathbf{J}_v$ are defined by (A4), (A5), and (A6). We show that:

$$\mathbf{J}_R = \left[\mathbf{0}_{3\times 3}, \left[\mathbf{G}^{(1)}_{\mathfrak{so}(3)}\, \mathbf{u}^{(k-1)}\, \delta t \ldots, \mathbf{G}^{(6)}_{\mathfrak{so}(3)}\, \mathbf{u}^{(k-1)}\, \delta t\right]^{\top}, \mathbf{0}_{3\times 6}\right]^{\top} \tag{A12}$$

$$\mathbf{u}^{(k-1)} = \left[\left(\hat{\mathbf{v}}^{(k-1|k-1)}\right)_1, 0, 0\right]^{\top} \tag{A13}$$

$$\mathbf{J}_p = \mathbf{0}_{12\times 3} \tag{A14}$$

$$\mathbf{J}_v = \left[\mathbf{0}_{3\times 3}, \left(\widehat{\mathbf{R}}_{WC}^{(k-1|k-1)}\, [1, 0, 0]^{\top}\right)^{\top} \delta t, \mathbf{0}_{3\times 6}\right]^{\top} \tag{A15}$$

$$\mathbf{J}_w = \left[\mathbf{0}_{3\times 9}, \mathbf{I}_{3\times 3}\, \delta t\right]^{\top}, \tag{A16}$$

where $\{\mathbf{G}^{(l)}_{\mathfrak{so}(3)}\}_{l=1}^{3}$ $\left(\mathbf{G}^{(l)}_{\mathfrak{so}(3)} \in \mathbb{R}^{3\times 3}\right)$ is avector basis of Lie algebra $\mathfrak{so}(3)$.

Appendix B. Computation of $J^{(k)}$

It should be remembered that:

$$\mathbf{J}^{(k)} = \left[\mathbf{J}_c^{(k)} \quad \mathbf{0}_{2\times 6\,(j-1)} \quad \mathbf{J}_p^{(j)} \quad \mathbf{0}_{2\times 6\,(m_k-j)}\right] \odot \mathbf{I}_{4\times 1}. \tag{A17}$$

Consequently, we have to compute two quantities: $\mathbf{J}_c^{(k)}$ and $\mathbf{J}_p^{(j)}$.

Appendix B.1. Computation of $\mathbf{J}_c^{(k)}$

$\mathbf{J}_c^{(k)}$ corresponds to the Jacobian of the observation, according to $\widehat{\mathbf{X}}_a^{(k|k-1)}$. Consequently:

$$\mathbf{J}_c^{(k)} = \left.\frac{\partial \mathbf{h}(\widehat{\mathbf{X}}_a^{(k|k-1)}\,\mathrm{Exp}_G^{\wedge}(\delta_c)\,,\widehat{\mathbf{X}}_{WP}^{(j)})}{\partial \delta}\right|_{\delta=\mathbf{0}} \tag{A18}$$

$$= \left.\left[\frac{\partial \mathbf{h}(\widehat{\mathbf{X}}_{WC}^{(k|k-1)}\,\mathrm{Exp}_{SE(3)}^{\wedge}(\delta_c))}{\partial \delta_c}, \underbrace{\frac{\partial \mathbf{h}(.,\mathbf{g}^{(k)}+\delta_g)}{\partial \delta_g}}_{\mathbf{0}_{p\times 1}}\right]\right|_{\delta_c=\mathbf{0},\delta_g=\mathbf{0}}. \tag{A19}$$

Furthermore:

$$\left.\frac{\partial \mathbf{h}(\widehat{\mathbf{X}}_{WC}^{(k|k-1)}\,\mathrm{Exp}_{SE(3)}^{\wedge}(\delta_c),\widehat{\mathbf{X}}_{WP}^{(j)})}{\partial \delta_c}\right|_{\delta_c=\mathbf{0}} =$$

$$\left.\frac{\partial\, \mathbf{K}\,\mathbf{I}_{3\times 4}\left(\widehat{\mathbf{X}}_{WC}^{(k|k-1)}\,\mathrm{Exp}_{SE(3)}^{\wedge}(\delta_c)\right)^{-1}\widehat{\mathbf{X}}_{WP}^{(j)}\,[\mathbf{q}_i^{\top},1]^{\top}}{\partial \delta_c\left(\mathbf{I}_{3\times 4}\left(\widehat{\mathbf{X}}_{WC}^{(k)}\,\mathrm{Exp}_{SE(3)}^{\wedge}(\delta_c)\right)^{-1}\widehat{\mathbf{X}}_{WP}^{(j)}\,[\mathbf{q}_i^{\top},1]^{\top}\right)_3}\right|_{\delta_c=\mathbf{0}}. \tag{A20}$$

In order to compute this Jacobian, we need to know:

- the classical derivative product rule according to $\delta_c \in \mathbb{R}^6$;
- the fact that $\left.\frac{\partial\,\mathrm{Exp}_{SE(3)}^{\wedge}(\delta_c)}{\partial(\delta_c)_l}\right|_{\delta_c=\mathbf{0}} = \mathbf{G}_{\mathfrak{se}(3)}^{(l)}\ \forall l \in [\![1,6]\!]$, where $\mathbf{G}_{\mathfrak{se}(3)}^{(l)} \in \mathbb{R}^{4\times 4}$ is the lth vector basis of Lie algebra $\mathfrak{se}(3)$.

Consequently, by computing:

$$\mathbf{d}^{(l)} = \left.\frac{\partial\,\mathbf{h}(\mathbf{X}_{WC}^{(k)}\,\mathrm{Exp}_{SE(3)}^{\wedge}(\delta_c)\,,\mathbf{X}_{WP}^{(j)})}{\partial(\delta_c)_l}\right|_{\delta_c=\mathbf{0}} \in \mathbb{R}^3,$$

$\mathbf{J}_c^{(k)}$ can be obtained in the following way:

$$\mathbf{J}_c^{(k)} = \left[\mathbf{d}^{(1)},\ldots,\mathbf{d}^{(6)},\mathbf{0}_{p\times 1}\right]. \tag{A21}$$

$\mathbf{d}^{(l)}$ is, thus, obtained thanks to the product rule:

$$\mathbf{d}^{(l)} = \frac{1}{\alpha^2}\{-\mathbf{K}\,\mathbf{I}_{3\times 4}\,\mathbf{G}_{\mathfrak{se}(3)}^{(l)}\left(\mathbf{X}_{WC}^{(k)}\right)^{-1}\mathbf{X}_{WP}^{(j)}\left[\mathbf{q}_i^{\top},1\right]^{\top}\alpha +$$
$$\left(\mathbf{I}_{3\times 4}\,\mathbf{G}_{\mathfrak{se}(3)}^{(l)}\left(\mathbf{X}_{WC}^{(k)}\right)^{-1}\mathbf{X}_{WP}^{(j)}\left[\mathbf{q}_i^{\top},1\right]^{\top}\right)_3\beta\}, \tag{A22}$$

with:

$$\begin{cases} \alpha = \left(\mathbf{I}_{3\times 4}\left(\widehat{\mathbf{X}}^{(k)}_{WC}\right)^{-1}\widehat{\mathbf{X}}^{(j)}_{WP}\left[\mathbf{q}_i^{\top},1\right]^{\top}\right)_3 \in \mathbb{R} & \text{(A23)} \\ \boldsymbol{\beta} = \mathbf{K}\,\mathbf{I}_{3\times 4}\left(\widehat{\mathbf{X}}^{(k)}_{WC}\right)^{-1}\widehat{\mathbf{X}}^{(j)}_{WP}\left[\mathbf{q}_i^{\top},1\right]^{\top} \in \mathbb{R}^3. & \text{(A24)} \end{cases}$$

Appendix B.2. Computation of $\mathbf{J}_p^{(j)}$

$\mathbf{J}_p^{(j)}$ corresponds to the Jacobian of the observation, according to $\widehat{\mathbf{X}}^{(j)}_{WP}$. Thus:

$$\mathbf{J}_p^{(j)} = \left.\frac{\partial \mathbf{h}(\widehat{\mathbf{X}}_a^{(k|k-1)}, \widehat{\mathbf{X}}^{(j)}_{WP}\,\mathrm{Exp}^{\wedge}_{SE(3)}(\delta_p))}{\partial \delta_p}\right|_{\delta_p=\mathbf{0}} \tag{A25}$$

$$= \left.\frac{\partial\,\mathbf{K}\,\mathbf{I}_{3\times 4}\left(\widehat{\mathbf{X}}^{(k)}_{WC}\right)^{-1}\widehat{\mathbf{X}}^{(j)}_{WP}\,\mathrm{Exp}^{\wedge}_{SE(3)}(\delta_p)\ \left[\mathbf{q}_i^{\top},1\right]^{\top}}{\partial \delta_p\left(\mathbf{I}_{3\times 4}\left(\widehat{\mathbf{X}}^{(k)}_{WC}\right)^{-1}\widehat{\mathbf{X}}^{(j)}_{WP}\,\mathrm{Exp}^{\wedge}_{SE(3)}(\delta_p)\ \left[\mathbf{q}_i^{\top},1\right]^{\top}\right)_3}\right|_{\delta_p=\mathbf{0}}. \tag{A26}$$

By using the classical derivative product rule, according to $\delta_p \in \mathbb{R}^6$, we obtain:

$$\begin{aligned} \mathbf{f}^{(l)} = \left.\frac{\partial \mathbf{h}(\widehat{\mathbf{X}}^{(k)}_{WC}, \widehat{\mathbf{X}}^{(j)}_{WP}\,\mathrm{Exp}^{\wedge}_{SE(3)}(\delta_p))}{\partial (\delta_p)_l}\right|_{\delta_p=\mathbf{0}} = \\ \frac{1}{\alpha^2}\{\mathbf{K}\,\mathbf{I}_{3\times 4}\left(\widehat{\mathbf{X}}^{(k)}_{WC}\right)^{-1}\widehat{\mathbf{X}}^{(j)}_{WP}\,\mathbf{G}^{(l)}_{\mathfrak{se}(3)}\left[\mathbf{q}_i^{\top},1\right]^{\top}\alpha - \\ \left(\mathbf{I}_{3\times 4}\left(\widehat{\mathbf{X}}^{(k)}_{WC}\right)^{-1}\widehat{\mathbf{X}}^{(j)}_{WP}\,\mathbf{G}^{(l)}_{\mathfrak{se}(3)}\left[\mathbf{q}_i^{\top},1\right]^{\top}\right)_3 \boldsymbol{\beta}\}. \end{aligned} \tag{A27}$$

Thus, $\mathbf{J}_p^{(j)}$ is given by:

$$\mathbf{J}_p^{(j)} = \left[\mathbf{f}^{(1)}, \dots, \mathbf{f}^{(6)}\right]. \tag{A28}$$

Appendix C. Expression of $\boldsymbol{\phi}_G(.)$

Appendix C.1. Case of $SO(3)$

The left Jacobian of $SO(3)$ is given by the following expression [26]:

$$\boldsymbol{\phi}_{SO(3)}(\mathbf{a}) = \mathbf{I} + \frac{1-\cos(\|\mathbf{a}\|)}{\|a\|^2}\,[\mathbf{a}]_{\times} + \frac{\|\mathbf{a}\| - \sin(\|\mathbf{a}\|)}{\|\mathbf{a}\|^3}\,[\mathbf{a}]^2_{\times}. \tag{A29}$$

Appendix C.2. Case of $SE(3)$

The left Jacobian of $SO(3)$ is given by the following expression [32]:

$$\boldsymbol{\phi}_{SE(3)}(\mathbf{a}) = \begin{bmatrix} \boldsymbol{\phi}_{SO(3)}(\mathbf{a}_r) & \mathbf{Q}(\mathbf{a}) \\ \mathbf{0}_{3\times 3} & \boldsymbol{\phi}_{SO(3)}(\mathbf{a}_p) \end{bmatrix}, \tag{A30}$$

with $\mathbf{a} = \begin{bmatrix} \underbrace{\overset{\top}{\mathbf{a}_r}}_{\in\mathbb{R}^3}, \underbrace{\mathbf{a}_p}_{\in\mathbb{R}^3} \end{bmatrix}^\top$ and

$$\begin{aligned}\mathbf{Q}(\mathbf{a}) = \frac{1}{2}\left[\mathbf{a}_p\right]_\times + \left(\frac{\|\mathbf{a}_r\| - \sin(\|\mathbf{a}_r\|)}{\|\mathbf{a}_r\|^3}\right) \times \\ \left([\mathbf{a}_r]_\times \left[\mathbf{a}_p\right]_\times + \left[\mathbf{a}_p\right]_\times [\mathbf{a}_r]_\times + [\mathbf{a}_r]_\times \left[\mathbf{a}_p\right]_\times [\mathbf{a}_r]_\times\right) - \\ \left(\frac{1 - \frac{\|\mathbf{a}_r\|^1}{2} - \cos(\|\mathbf{a}_r\|)}{\|\mathbf{a}_r\|^4}\right) \times \\ \left([\mathbf{a}_r]_\times^2 \left[\mathbf{a}_p\right] + \left[\mathbf{a}_p\right] [\mathbf{a}_r]_\times^2 - 3\,[\mathbf{a}_r]_\times [\mathbf{a}_r]_\times [\mathbf{a}_r]_\times^2\right) - \\ \frac{1}{2}\left(\frac{1 - \|\mathbf{a}_r\| - \cos(\|\mathbf{a}_r\|)}{\|\mathbf{a}_r\|^4} - 3\,\frac{\|\mathbf{a}_r\| - \sin(\|\mathbf{a}_r\|) - \frac{\|\mathbf{a}_r\|^3}{6}}{\|\mathbf{a}_r\|^5}\right) \times \\ \left([\mathbf{a}_r]_\times \left[\mathbf{a}_p\right]_\times [\mathbf{a}_r]_\times^2 + [\mathbf{a}_r]_\times^2 \left[\mathbf{a}_p\right]_\times [\mathbf{a}_r]_\times\right). \end{aligned} \tag{A31}$$

Appendix C.3. Case of G

In our case, the LG of interest is written as $G = SE(3) \times \mathbb{R}^p \times SE(3)^{m_k}$, such that:

$$\boldsymbol{\phi}_G(\mathbf{a}) = \begin{bmatrix} \boldsymbol{\phi}_{SE(3)}(\mathbf{a}_1) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\phi}_{\mathbb{R}^p}(\mathbf{a}_2) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\phi}_{SE(3)^{m_k}}(\mathbf{a}_3) \end{bmatrix}, \tag{A32}$$

with $\mathbf{a} = \begin{bmatrix} \underbrace{\overset{\top}{\mathbf{a}_1}}_{\in\mathbb{R}^6}, \underbrace{\overset{\top}{\mathbf{a}_2}}_{\in\mathbb{R}^3}, \underbrace{\overset{\top}{\mathbf{a}_3}}_{\in\mathbb{R}^{6\,m_k}} \end{bmatrix}^\top$. As $\mathbb{R}^p$ is acommutative LG, $\boldsymbol{\phi}_{\mathbb{R}^p}(\mathbf{a}_2) = \mathbf{I}_{p\times p}$. Furthermore, $\boldsymbol{\phi}_{SE(3)^{m_k}}(\mathbf{a}_3)$ is the block diagonal concatenation of m_k repetitions of $\boldsymbol{\phi}_{SE(3)}(.)$.

References

1. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Computer Vision—ECCV 2014, Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Springer International Publishing: Cham, Switzerland, 2014; pp. 834–849.
2. Khairuddin, A.R.; Talib, M.S.; Haron, H. Review on simultaneous localization and mapping (SLAM). In Proceedings of the IEEE International Conference on Control System, Computing and Engineering, Penang, Malaysia, 27–29 November 2015; pp. 85–90.
3. Triggs, B.; McLauchlan, P.; Hartley, R.; Fitzgibbon, A. Bundle Adjustment—A Modern Synthesis. In Proceedings of the International Workshop on Vision Algorithms, Corfu, Greece, 21–22 September 1999.
4. Mouragnon, E.; Lhuillier, M.; Dhome, M.; Dekeyser, F.; Sayd, P. 3D reconstruction of complex structures with bundle adjustment: an incremental approach. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 3055–3061.
5. Guivant, J.; Nebot, E. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Trans. Robot. Autom.* **2001**, *17*, 242–257. [CrossRef]
6. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Fourth Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; pp. 147–151.
7. Rosten, E.; Drummond, T. Machine learning for high-speed corner detection. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006.
8. Bowman, S.L.; Atanasov, N.; Daniilidis, K.; Pappas, G.J. Probabilistic data association for semantic SLAM. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1722–1729.
9. Lim, H.; Lee, Y.S. Real-time single camera SLAM using fiducial markers. In Proceedings of the ICCAS-SICE, Fukuoka, Japan, 18–21 August 2009; pp. 177–182.
10. Houben, S.; Droeschel, D.; Behnke, S. Joint 3D laser and visual fiducial marker based SLAM for amicro aerial vehicle. In Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Baden-Baden, Germany, 19–21 September 2016; pp. 609–614.

11. Kalaitzakis, M.; Cain, B.; Carroll, S.; Ambrosi, A.; Whitehead, C.; Vitzilaios, N. Fiducial Markers for Pose Estimation Overview, Applications and Experimental Comparison of the ARTag, AprilTag, ArUco and STag Markers. *J. Intell. Robot. Syst.* **2021**, *101*, 71. [CrossRef]
12. Fiala, M. ARTag, afiducial marker system using digital techniques. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; Volume 2, pp. 590–596.
13. Pilté, M.; Bonnabel, S.; Barbaresco, F. Tracking the Frenet-Serret frame associated to ahighly maneuvering target in 3D. In Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, VIC, Australia, 12–15 December 2017; pp. 1969–1974.
14. Mika, D.; Jozwik, J. Lie groups methods in blind signal processing. *Sensors* **2020**, *20*, 440. [CrossRef] [PubMed]
15. Bourmaud, G.; Giremus, A.; Berthoumieu, Y.; Mégret, R. Discrete Extended Kalman Filter on Lie groups. In Proceedings of the EUSIPCO, Marrakech, Morocco, 9–13 September 2013.
16. Bonnabel, S.; Martin, P.; Salaün, E. Invariant Extended Kalman Filter: Theory and application to a velocity-aided attitude estimation problem. In Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, Shanghai, China, 15–18 December 2009; pp. 1297–1304.
17. Wang, J.; Zhang, C.; Wu, J.; Liu, M. An Improved Invariant Kalman Filter for Lie Groups Attitude Dynamics with Heavy-Tailed Process Noise. *Machines* **2021**, *9*, 182. [CrossRef]
18. Brossard, M.; Bonnabel, S.; Condomines, J.P. Unscented Kalman filtering on Lie groups. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 2485–2491.
19. Ćesić, J.; Marković, I.; Bukal, M.; Petrović, I. Extended information filter on matrix Lie groups. *Automatica* **2017**, *82*, 226–234. [CrossRef]
20. Marjanovic, G.; Solo, V. An engineer's guide to particle filtering on matrix Lie groups. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 3969–3973.
21. Brossard, M.; Bonnabel, S.; Barrau, A. Invariant Kalman Filtering for Visual Inertial SLAM. In Proceedings of the 2018 21st International Conference on Information Fusion (FUSION), Cambridge, UK, 10–13 July 2018; pp. 2021–2028.
22. Lenac, K.; Marković, I.; Petrović, I. Exactly sparse delayed state filter on Lie groups for long-term pose graph SLAM. *Int. J. Robot. Res.* **2018**, *37*, 585–610. [CrossRef]
23. Bourmaud, G.; Mégret, R. Robust large scale monocular visual SLAM. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1638–1647.
24. Hui, C.; Shiwei, M. Visual SLAM based on EKF filtering algorithm from omnidirectional camera. In Proceedings of the 2013 IEEE 11th International Conference on Electronic Measurement Instruments, Harbin, China, 16–19 August 2013; Volume 2, pp. 660–663.
25. Selig, J.M. *Geometrical Methods in Robotics*; Springer: Berlin/Heidelberg, Germany, 1996.
26. Solà, J.; Deray, J.; Atchuthan, D. A micro Lie theory for state estimation in robotics. *arXiv* **2018**, arXiv:1812.01537.
27. Faraut, J. *Analysis on Lie Groups: An Introduction*; Cambridge University Press: Cambridge, UK, 2008.
28. Chirikjian, G. Information-Theoretic Inequalities On Unimodular Lie Groups. *J. Geom. Mech.* **2010**, *2*, 119–158. [CrossRef] [PubMed]
29. Koks, D. A Roundabout Route to Geometric Algebra. In *Explorations in Mathematical Physics: The Concepts Behind an Elegant Language*; Springer Science: Berlin/Heidelberg, Germany, 2006; Chapter 4, pp. 147–184.
30. Said, S.; Bombrun, L.; Berthoumieu, Y.; Manton, J.H. Riemannian Gaussian Distributions on the Space of Symmetric Positive Definite Matrices. *IEEE Trans. Inf. Theory* **2017**, *63*, 2153–2170. [CrossRef]
31. Pennec, X. Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. *J. Math. Imaging Vis.* **2006**, *25*, 127–155. [CrossRef]
32. Barfoot, T.; Furgale, P.T. Associating Uncertainty With Three-Dimensional Poses For Use In Estimation Problem. *IEEE Trans. Robot.* **2014**, *30*, 679–693. [CrossRef]
33. Sola, J.; Monin, A.; Devy, M.; Lemaire, T. Undelayed initialization in bearing only SLAM. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 2499–2504.
34. Bishop, C. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
35. Bourmaud, G.; Mégret, R.; Giremus, A.; Berthoumieu, Y. From Intrinsic Optimization to Iterated Extended Kalman Filtering on Lie Groups. *J. Math. Imaging Vis.* **2016**, *55*, 284–303. [CrossRef]
36. Huynh, D.Q. Metrics for 3D Rotations: Comparison and Analysis. *J. Math. Imaging Vis.* **2009**, *35*, 155–164. [CrossRef]
37. Chauchat, P.; Vilà-Valls, J.; Chaumette, E. Robust Linearly Constrained Invariant Filtering for aClass of Mismatched Nonlinear Systems. *IEEE Control Syst. Lett.* **2022**, *6*, 223–228. [CrossRef]
38. Ozyesil, O.; Voroninski, V.; Basri, R.; Singer, A. A Survey of Structure from Motion. *Acta Numerica* **2017**, *26*, 305–364. [CrossRef]
39. Zitová, B.; Flusser, J. Image registration methods: A survey. *Image Vis. Comput.* **2003**, *21*, 977–1000. [CrossRef]