Line-based Monocular Graph SLAM

Dong Ruifang, Vincent Frémont, Simon Lacroix, Isabelle Fantoni, Liu Changan

Line-based Monocular Graph SLAM

Dong Ruifang[1], Vincent Fremont[2], Simon Lacroix[3], Isabelle Fantoni[2], Liu Changan[1]

***Abstract*— This paper presents a new line based 6-DOF monocular algorithm that uses the iSAM2, a point-based Graph SLAM approach. We extend iSAM2 to minimize the reprojection error of the line features to solve the line-based SLAM problem. A specific line representation is exploited that combines the Plücker Coordinates and the Cayley representation. The Plücker Coordinates are used for the 3D line projection function, and the Cayley representation helps to update the lines parameters during the non-linear optimization process. An undelayed initialization method with inverse depth parameters is also adopted. Both simulation and real experiments are carried out showing that the approach achieves high accuracy and consistency, and outperforms EKF-based SLAM method.**

I. INTRODUCTION

Monocular SLAM is a SLAM problem where the only onboard perception sensor is a single camera [1]. It is more challenging than the SLAM problems using laser sensors and/or RGB-D cameras because of the lack of scale-free depth information. Works on visual SLAM focuse on mapping landmarks points. Point landmarks have interesting properties in the context of visual SLAM: Point feature selection, description and tracking have been well-studied in the literature [2]. However, a map consisting of a sparse set of 3D points is far from describing the structure of the surrounding world. Instead, line shape structures exist in most of the man-built environments. Then a line-based map can give higher level relevant information about the observed environment. This is especially true for scenes with abundant line elements such as transmission towers, buildings, etc. Furthermore, edge matching can be achieved even when important viewpoints changes occur, like in loop closing, or when registering aerial with ground data. For all these reasons, in this work, we focus on the representation, the camera motion estimation and the landmark localization for line feature based monocular SLAM.

A. Related work

Previous work on line representations can be divided in two approaches: The non-linear minimal 4-parameters formalism, and the linear over-parametrized representation. For the first one, the authors of [3] propose to use two cosines direction with 2D coordinates to create the direction and position of a line. In [4], a line is represented as two planes intersection that are parallel to the X-axis and the Y-axis respectively. Several conventions coexist to represent all possible lines with a satisfactory numerical precision. More recently, lines used in [5] are parametrized by two pairs of elevation and angles with the two corresponding camera centers as anchors. The Cayley representation is used in [6] to model lines. These representations use only four parameters that give the 4 degrees of freedom of a line in 3D space. Therefore there is no internal gauge freedom nor any consistency constraint. This feature makes the minimal representations well adapted for non-linear optimization. It is however difficult to express the line projection function because of the intrinsic non linearity.

For the linear over-parametrized approaches, [7] and [8] represent a line with its two endpoints. In [9], the midpoint and the direction of the segment are used to represent a 3D line. For these two representations, neither the endpoints nor the midpoint of segments are observable. In [10] a 3D line is represented by its closest point to the origin together with its direction. In [11] and [12], the authors use the Plücker line representation that includes the moment of a line to the origin and the line direction in space. The Plücker coordinates are well adapted to the projection through a pinhole camera, but the internal gauge freedom may induce numerical instabilities during the non-linear optimization step. This is mainly due to the fact that the Plücker coordinates have two degrees of internal gauge freedoms and are subject to the Plücker constraints.

In this paper, the Plücker Coordinates and the Cayley representation are combined: The Plücker Coordinates are employed to build the projection function for the 3D lines and the Cayley representation is applied to update their parameters during the non-linear optimization process.

For camera motion estimation and landmark localization, there mainly exist two frameworks in monocular visual SLAM. One is based on filtering, and represents all camera poses and the map as a single state vector. The other one uses non-linear optimization methods, or smoothing approaches, to compute the estimate of all poses and the map, given an initial guess. For line-based monocular SLAM, most of previous works focus on filtering-based solutions. The works of [7], [8], [11], [12], [13] use Extended Kalman Filter (EKF) as the line-based SLAM framework. While in [7], [8], [12] delayed techniques are used for initialization, and in [11] an undelayed solution is proposed. We encourage the reader to consult [14], [15], [16] for discussions on delayed/undelayed initializations and their importance in monocular SLAM.

The main drawback of EKF-based algorithms is associated

Authors[1] are with the Control and Computer Engineering School, North China Electric Power University, Beijing, China dongruifang123@gmail.com

Authors[2] are with Sorbonne Universités, Université de Technologie de Compiègne, CNRS, Heudiasyc UMR 7253, CS 60319, 60203 Compiègne cedex, France, firstname.name@hds.utc.fr

Author[3] is with the LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France, simon.lacroix@laas.fr

to the limited number of update parameters, it is well known that the computational complexity of the EKF becomes intractable fairly quickly, hence the number of features in the map will be severly limited. Moreover, filtering has been shown to be inconsistent when applied to the inherently nonlinear SLAM problem [17]. On the other side, smoothing approaches estimate the full robot trajectory from a full set of measurements. These methods solve the so-called full SLAM problem, and they mainly rely on least squares error minimization techniques. To the best of our knowledge, few line-based monocular SLAM works using nonlinear optimization methods exist. Only recently, [6] extended the sparse bundle adjustment algorithm to solve the line-based SFM (Structure-From-Motion) problem.

In this work, we also focus on the nonlinear optimization process, the incremental Smoothing And Mapping (iSAM2) [18] for SLAM. The iSAM2 approach provides an efficient and exact solution to a sparse nonlinear optimization problem in an incremental setting that is employed for nonlinear estimation. The iSAM2 implementation was originally designed for point-based reconstruction: in this paper, we extend the iSAM2 approach to make it appropriate for updating and optimizing line parameters.

B. Contributions and outline

The primary contributions of this paper are the following: First, two representations of lines are combined during the monocular SLAM process: the Plücker Coordinates are employed to build the projection function while the Cayley representation is applied to update the line parameters during the iSAM2 optimization. Second, the iSAM2 approach is extended to solve the line-based visual monocular SLAM problem, thus build more structured maps than point-based approaches.

The paper is organized as follows. Section II gives a complete description of the 3D line representations and manipulations used in this paper. Then the back-end optimization using the modified iSAM2 approach is described in Section III. In Section IV, simulations and experiments on real image sequences are presented. Finally, Section V concludes the paper and presents some future work.

II. 3D Line Representation for SLAM

A. Plücker Coordinates

A line in the 3D projective space $\mathbb{P}^3$, can be defined from two 3-vector points $A, B \in \mathbb{R}^3$ belonging to the line. This line can be coded as a homogeneous 6-vector $\mathcal{L}$, with the so called Plücker coordinates $\mathcal{L} = (\mathbf{v}; \mathbf{n})^\top$. Assume O is the origin of the world frame, its Plücker coordinates are obtained from:

$$\begin{cases} \mathbf{v} = \overrightarrow{OB} - \overrightarrow{OA} \\ \mathbf{n} = \overrightarrow{OA} \times \overrightarrow{OB} \text{ or } \mathbf{n} = \overrightarrow{OA} \times \mathbf{v} \text{ or } \mathbf{n} = \overrightarrow{OB} \times \mathbf{v} \end{cases} \tag{1}$$

where $\mathbf{n}$ is called the 3-vector line moment and is normal to the plane containing the line $\mathcal{L}$ and the origin O, with a magnitude equal to the distance from the line to origin, and $\mathbf{v}$ is the 3-vector direction vector from A to B.

1) Plücker line manipulations: Having defined $\mathcal{L} = (\mathbf{v}; \mathbf{n})^\top$, the expression of the transformation from camera frame to world frame is:

$$\mathcal{L}^w = \mathbf{H} \cdot \mathcal{L}^c \triangleq \begin{bmatrix} \mathbf{R} & [\mathbf{t}]_\times \mathbf{R} \\ 0 & \mathbf{R} \end{bmatrix} \cdot \mathcal{L}^c \tag{2}$$

where $\mathbf{R}$, $\mathbf{t}$ represent respectively the 3×3 rotation matrix and the 3-vector translation vector. The element $[\mathbf{t}]_\times$ represents the skew-symmetric matrix of the vector $\mathbf{t}$, $\mathbf{H}$ is the motion matrix for Plücker lines. The inverse transformation is performed with:

$$\mathcal{L}^c = \mathbf{H}^{-1} \cdot \mathcal{L}^w \sim \begin{bmatrix} \mathbf{R}^\top & -\mathbf{R}^\top [\mathbf{t}]_\times \\ 0 & \mathbf{R}^\top \end{bmatrix} \cdot \mathcal{L}^w \tag{3}$$

where $\mathcal{L}^c = (\mathbf{v}^c; \mathbf{n}^c)^\top$ represents the Plücker Coordinates of line in the camera frame, while $\mathcal{L}^w = (\mathbf{v}^w; \mathbf{n}^w)^\top$ represents the Plücker coordinates of line in world frame.

The corresponding linear expression for the 3-vector projected Plücker line $\mathbf{l}$ in image is expressed as:

$$\mathbf{l} = \mathbf{K} \cdot \mathbf{R}^\top \cdot (\mathbf{n}^w - \mathbf{t} \times \mathbf{v}^w) \text{ or } \mathbf{l} = \mathbf{K} \cdot \mathbf{n}^c \tag{4}$$

where the intrinsic matrix $\mathbf{K}$ for the Plücker line is:

$$\mathbf{K} = \begin{bmatrix} \alpha_v & 0 & 0 \\ 0 & \alpha_u & 0 \\ -\alpha_v u_0 & -\alpha_u v_0 & \alpha_u \alpha_v \end{bmatrix} \tag{5}$$

while α_v, α_u are the focal lengths in X and Y directions respectively, u_0, v_0 are the coordinates of the principal point in the image plane, the derivation of $\mathbf{K}$ is given in [6].

Notice that when the line is expressed in camera frame, only the plane's normal $\mathbf{n}^c$ appears in the projection equation, meaning that 2 degrees of freedom (DOF), the line's range and orientation contained in $\mathbf{v}^c$, are not observable.

2) Plücker Line Initialization: Monocular SLAM is a partially observable problem, in which the sensor does not give sufficient information to estimate the full state of a landmark given a single observation. It then raises to a landmark initialization problem. In [14], the authors presented the undelayed initialization method with inverse depth parameters to initialize the line segments for Monocular SLAM, which we adopt in our work. The initialization means obtaining the 3D position of landmarks from measurements defined in images. The following details the retro-projection process from the measurement lines to Euclidean space lines.

1) Compute the Plücker line representation 3-vector $\mathbf{l}$ in the image;
   The line measurement $z = (p_1, p_2)$ in the image plane is represented with two 2-vector endpoints $p_1, p_2 \in \mathbb{R}^2$. The associated Plücker line representation $\mathbf{l}$ in image can be computed as:

$$\mathbf{l} = \underline{p_1} \times \underline{p_2} \tag{6}$$

where $\underline{p_1}$ and $\underline{p_2}$ represent the homogeneous 3-vector coordinates of p_1 and p_2.

2) Compute the Plücker sub-vector $\mathbf{n}^c$ from the measured line $\mathbf{l}$, and normalize it to unit vector $\hat{\mathbf{n}}^c$.
The segment $\mathbf{l}$ detected in image uniquely determines the plane η containing the 3D line and the optical center C (see Fig. 1). It is known as the representation plane. The line moment $\mathbf{n}^c$ in the camera frame is obtained by simply inverting Eq. (4):

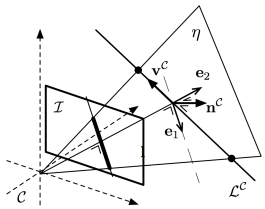


Fig. 1: Retro-projection of a segment in the image (excerpt from [14]).

$$\mathbf{n}^c = \mathbf{K}^{-1} \cdot \mathbf{l}. \tag{7}$$

The line moment $\mathbf{n}^c$ needs to be normalized into $\hat{\mathbf{n}}^c$ for a convenient implementation of inverse depth parameters-based initialization.

3) Compute the Plücker sub-vector $\mathbf{v}^c$ from $\hat{\mathbf{n}}^c$.
The non-measured direction vector $\mathbf{v}^c$ is meant to lie on the plane η, then $\mathbf{v}^c$ is initialized as:

$$\mathbf{v}^c = \beta_1 \cdot \mathbf{e_1} + \beta_2 \cdot \mathbf{e_2}, \quad \beta_1, \beta_2 \in \mathbb{R} \tag{8}$$

$$\mathbf{e_1} = \frac{(\hat{n}_y^c, \hat{n}_x^c, 0)^\top}{\sqrt{(\hat{n}_x^c)^2 + (\hat{n}_y^c)^2}}, \mathbf{e_2} = \frac{\hat{\mathbf{n}}^c}{\|\hat{\mathbf{n}}^c\|} \times \mathbf{e_1} \tag{9}$$

where $(\mathbf{e_1}, \mathbf{e_2})$ is a pair of 3-vector orthonormal basis of the plane η, $\boldsymbol{\beta} = (\beta_1, \beta_2)$ is provided as prior. It is possible to get $\mathbf{v}^c \subset \eta$ for any values of $\boldsymbol{\beta}$, and importantly, the Plücker constraint $\mathbf{v} \perp \mathbf{n}$ is satisfied by construction. Note that the distance from the line to the optical center is given by $d = \|\hat{\mathbf{n}}^c\|/\mathbf{v}^c = 1/\|\boldsymbol{\beta}\|$, this is an inverse-depth parameterization.

4) Transform $\mathcal{L}^c = (\hat{\mathbf{n}}^c, \mathbf{v}^c)$ to $\mathcal{L}^w$ using Eq. (2).

B. Cayley Representation of 3D Lines

The Plücker line representation must satisfy the constraint $\mathbf{v} \perp \mathbf{n}$, which is not easy to enforce in the optimization process. In [11], the Plücker constraint is guaranteed at initialization time but is not enforced further during the landmarks updates. So, in order to guarantee that the lines parameters satisfy the Plücker constraint, the Cayley representation is adopted to update the line during the nonlinear optimization process.

The Cayley representation is defined as a 4-vector $\mathbf{w} = (d, \mathbf{s})$, where d represents the distance from the line to the optical center, and $\mathbf{s}$ is a 3-vector which encodes the rotation information as illustrated in Fig. 2.

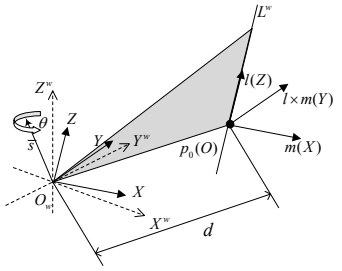


Fig. 2: Interpretation of the Cayley representation of a 3D line, excerpt from [6]. d is the distance from the origin of the world frame $O^w X^w Y^w Z^w$ to the 3D line. The rotation between $O^w X^w Y^w Z^w$ and $OXYZ$ the object frame is determined by a rotation angle $\theta = 2 \arctan \|\bar{s}\|$ around the rotation axis $\bar{s} = s/\|s\|$.

To transform the Plücker line to a Cayley line, an orthogonal matrix $\mathbf{Q}$ is constructed from the Plücker line as follows:

1) Compute the closest point P_0 in the line $\mathcal{L} = (\mathbf{v}; \mathbf{n})^\top$ to origin:

$$P_0 = (\mathbf{v} \times \mathbf{n})/\|\mathbf{v}\|^2, P_0 \in \mathbb{R}^3. \tag{10}$$

2) Compute the related line moment $\mathbf{m}$ for P_0. We need to normalize the direction vector $\mathbf{v}$ first, that is $\mathbf{v}^n = \mathbf{v}/\|\mathbf{v}\|^2$, then:

$$\mathbf{m} = P_0 \times \mathbf{v}^n, d = \|\mathbf{m}\| \tag{11}$$

3) Construct the 3×3 orthogonal matrix $\mathbf{Q}$:

$$\mathbf{Q} = \left[\mathbf{v}^n, \frac{\mathbf{m}}{\|\mathbf{m}\|}, \frac{\mathbf{v}^n \times \mathbf{m}}{\|\mathbf{v}^n \times \mathbf{m}\|} \right]. \tag{12}$$

Then a skew-symmetric matrix formed by the vector $\mathbf{s} = (s_x, s_y, s_z)^\top$ can be computed from $\mathbf{Q}$ and the identity matrix $\mathbf{I}_{3\times3}$ as :

$$[\mathbf{s}]_\times = (\mathbf{Q} - \mathbf{I})(\mathbf{Q} + \mathbf{I})^{-1}. \tag{13}$$

Thus, the Cayley representation $\mathbf{w} = (d, \mathbf{s})$ can be obtained. However, most of 4-parameter non-linear representations of lines have some singularities, and [6] presented a modified Cayley representation to solve the singularity problem as:

$$\mathbf{Q} = \begin{cases} [\mathbf{v}^n, \mathbf{i}_1, \mathbf{i}_2] & \text{if } \|\mathbf{m}\| \leq \tau \\ \left[\mathbf{v}^n, \frac{\mathbf{m}}{\|\mathbf{m}\|}, \frac{\mathbf{v}^n \times \mathbf{m}}{\|\mathbf{v}^n \times \mathbf{m}\|}\right] & \text{otherwise} \end{cases} \tag{14}$$

where $\mathbf{i}_1$ and $\mathbf{i}_2$ are one pair of the 3-vector orthogonal base of the nullspace formed by the linear system $(\mathbf{v}^n)^\top \cdot \mathbf{i} = 0$, and τ is a small number close to 0.

Transferring back from the Cayley representation to the Plücker coordinates is trivial:

$$\begin{aligned} \mathbf{Q} = [\mathbf{q_1}, \mathbf{q_2}, \mathbf{q_3}] &= (\mathbf{I} - [\mathbf{s}]_\times)^{-1}(\mathbf{I} + [\mathbf{s}]_\times) \\ &= \frac{(1-\|\mathbf{s}\|^2)\mathbf{I} + 2[\mathbf{s}]_\times + 2\mathbf{s}\mathbf{s}^\top}{1+\|\mathbf{s}\|^2}. \end{aligned} \tag{15}$$

According to the definition of $\mathbf{Q}$ in Eq. (12):

$$\mathbf{v}^n = \mathbf{q_1} \text{ and } \mathbf{m} = d \cdot \mathbf{q_2}. \tag{16}$$

III. Back-end Optimization using Incremental Smoothing and Mapping

In [18], the incremental smoothing and mapping (iSAM2) has been originally implemented for feature point based SLAM. In this work, their implementation is extended to solve the line feature based SLAM. So far, the robot poses are parameterized by their rotation matrices and translation vectors as $\mathbf{X}_i = (\mathbf{R}_i, \mathbf{t}_i)$, with $i \in 0 \dots M$; and the positions of the 3D line landmarks are represented by their Plücker coordinates $\mathcal{L}_j = (\mathbf{n}_j; \mathbf{v}_j)^\top$, with $j \in 1 \dots N$, and are updated by the 4-vector Cayley representation $\mathbf{w}_j = (d_j, \mathbf{s}_j)$ as described in the previous section.

A. The Error Model

Let $\mathbf{u}_i$ denote the control input or measured odometry, with $i \in 1 \dots M$. The robot motion prediction error model is defined as:

$$e(\mathbf{X}_i, \mathbf{X}_{i-1}) = \mathbf{X}_i - f(\mathbf{X}_{i-1}, \mathbf{u}_i), \tag{17}$$

where $\mathbf{X}_i, \mathbf{X}_{i-1}$ is the robot state at time i and $i-1$, $f(\mathbf{X}_{i-1}, \mathbf{u}_i)$ represents the motion model of the robot. It is generally a non-linear function.

Let $z_k(p_1, p_2)$ represent the line measurement in the image, with $k \in 1 \dots K$. Unlike point features whose error function is simply the distance between the observed location and the projected location in the image plane, the line measurement prediction error model is defined in this work as a 2-vector $\mathbf{e}$ containing the signed orthogonal distances from the endpoints to the re-projected line (see Fig. 3).

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} \mathbf{l}^\top \cdot \underline{p_1} / \sqrt{l_1^2 + l_2^2} \\ \mathbf{l}^\top \cdot \underline{p_2} / \sqrt{l_1^2 + l_2^2} \end{bmatrix} \in \mathbb{R}^2, \tag{18}$$

where, $\underline{p_1}$ and $\underline{p_2}$ represent the 3-vector homogeneous coordinates of endpoint p_1 and p_2, $\mathbf{l} = (l_1, l_2, l_3)^\top$ represents the predicted line using the line projection function of Eq. (4).

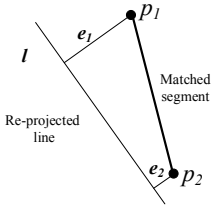


Fig. 3: Illustration of the error model of a re-projected line.

The re-projection error γ of the estimated lines can be computed as:

$$\gamma^2 = e_1^2 + e_2^2. \tag{19}$$

We only consider lines of sufficient length, as very short line segment yields an artificially small error, in this work, the line segment with the length less than 30 pixels is unaccepted.

B. The Optimization Process

Smoothing And Mapping (SAM) aims to recover the maximum a posteriori estimate for all unknown poses $\mathbf{X}_i$ of the robot and landmark positions $\mathcal{L}_j$, given the measurements z_k and control inputs $\mathbf{u}_i$. This leads to the following non-linear least-squares problem:

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{M} \left\| e\left(\mathbf{X}_{i-1}, \mathbf{X}_i\right) \right\|_{\Omega_i}^2 + \sum_{k=1}^{K} \left\| e\left(\mathbf{X}_{ik}, \mathcal{L}_{jk}\right) \right\|_{\Lambda_k}^2 \right\} \tag{20}$$

where, $\Theta \triangleq (\mathbf{X}, \mathbf{LK})$, while $\mathbf{X} \triangleq \{\mathbf{X}_i\}$,with $\mathbf{LK} \triangleq \{\mathcal{L}_j\}$; $e(\mathbf{X}_{i-1}, \mathbf{X}_i)$ is the motion prediction error shown in (17), $e(\mathbf{X}_{ik}, \mathcal{L}_{jk})$ is the measurement prediction error shown in Eq. (18), the correspondences $(\mathbf{X}_{ik}, \mathcal{L}_{jk})$ indicate that line measurement z_k is derived from pose $\mathbf{X}_i$ and landmark position $\mathcal{L}_j$; Ω_i and Λ_k represent the covariance matrix of the two errors. The detailed derivation behind Eq. (20) can be found almost entirely in [17].

Next, the terms in Eq. (20) are linearized as follows. First the terms of the motion prediction error are obtained as:

$$\begin{aligned} e(\mathbf{X}_i, \mathbf{X}_{i-1}) &\approx \left\{ {\mathbf{X}^0}_i + \delta\mathbf{X}_i \right\} \\ &\quad - \left\{ f_i({\mathbf{X}^0}_{i-1}, \mathbf{u}_i) + {\mathbf{F}_i}^{i-1} \delta\mathbf{X}_{i-1} \right\} \\ &\approx \left\{ \delta\mathbf{X}_i - {\mathbf{F}_i}^{i-1} \delta\mathbf{X}_{i-1} \right\} + \boldsymbol{\alpha}_i \end{aligned} \tag{21}$$

where, $\boldsymbol{\alpha}_i = \mathbf{X}_i^0 - f_i\left(\mathbf{X}_{i-1}^0, \mathbf{u}_i\right)$, $\mathbf{F}_i^{i-1} = \frac{\partial f_i(\mathbf{X}_{i-1}, \mathbf{u}_i)}{\partial \mathbf{X}_{i-1}}$.

For the measurement prediction error $e(\mathbf{X}_{ik}, \mathcal{L}_{jk})$, Eq. (18) reveals that the error can no longer be obtained from the subtraction $e(\mathbf{X}_{ik}, \mathcal{L}_{jk}) = h(\mathbf{X}_{ik}, \mathcal{L}_{jk}) - z_k$, but from an implicit function that is obtained by composing the line transformation and projection from Eq. (4) and the error measurements from Eq. (18). The result is a somewhat complicated expression with a generic form $e(\mathbf{X}_{ik}, \mathcal{L}_{jk}) = g_k(\mathbf{X}_{ik}, \mathcal{L}_{jk}, z_k)$. Linearizing the terms of measurement prediction error gives:

$$e\left(\mathbf{X}_{ik}, \mathcal{L}_{jk}\right) \approx g_k\left(\mathbf{X}_{ik}^0, \mathcal{L}_{jk}^0, z_k\right) + \mathbf{H}_k^{ik} \delta\mathbf{X}_{ik} + \mathbf{J}_k^{jk} \delta\mathbf{w}_{jk} \tag{22}$$

where, $\mathbf{H}_k^{ik} = \frac{\partial g_k(\mathbf{X}_{ik}, \mathcal{L}_{jk}, z_k)}{\partial \mathbf{X}_{ik}}$, $\mathbf{J}_k^{jk} = \frac{\partial g_k(\mathbf{X}_{ik}, \mathcal{L}_{jk}, z_k)}{\partial \mathbf{w}_{jk}}$.

It is worth noticing the significant change of Jacobians and increment with respect to lines. This is due to the Cayley representation used in the update process where the increment of line changes into $\delta\mathbf{w}_{jk}$ instead of $\delta\mathcal{L}_{jk}$.

Using the linearized process from Eq. (21) and Eq. (22) in Eq. (20) gives:

$$\begin{aligned} \delta^* = \underset{\delta}{\operatorname{argmin}} \Bigg(& \sum_{i=1}^{M} \left\| \mathbf{G}_i^i \delta\mathbf{X}_i - \mathbf{F}_i^{i-1} \delta\mathbf{X}_{i-1} + \boldsymbol{\alpha}_i \right\|_{\Omega_i}^2 \\ & + \sum_{k=1}^{N} \left\| \mathbf{H}_k^{ik} \delta\mathbf{X}_{ik} + \mathbf{J}_k^{jk} \delta\mathbf{w}_{jk} + \mathbf{g}_k \right\|_{\Lambda_k}^2 \Bigg). \end{aligned} \tag{23}$$

To avoid treating $\delta\mathbf{X}_i$ in a special way, the matrix $\mathbf{G}_i^i = \mathbf{I}_{h \times h}$ is introduced, with h the dimension of $\mathbf{X}_i$.

Following what SAM performs, dropping the Mahalanobis notation and collecting all the Jacobian matrices into a matrix $\mathbf{A}$, and the vectors $\boldsymbol{\alpha}_i$ and $\mathbf{g}_k$ into a right-hand side vector $\mathbf{b}$, the following standard least-squares problem (24) is obtained.

$$\delta^* = \underset{\delta}{\operatorname{argmax}} \|\mathbf{A}\delta + \mathbf{b}\|^2_{\Sigma} \tag{24}$$

while $\mathbf{A} = \Sigma^{-\top/2}\mathbf{Y}$,$\mathbf{b} = \Sigma^{-\top/2}\mathbf{B}$. Σ is a diagonal matrix consists of each Ω_i and Λ_k, $\mathbf{Y}$ is a sparse Jacobian matrix, the non-zero blocks are only at states affected by the constraints imposed by the measurements. We illustrate the sparsity of $\mathbf{Y}_i$ (linking $\mathbf{X}_i$ and $\mathbf{X}_{i-1}$) and $\mathbf{Y}_k$ (linking $\mathbf{X}_i$ and $\mathcal{L}_j$)as:

$$\mathbf{Y}_i = \begin{bmatrix} \cdots & \mathbf{F}_i^{i-1} & \mathbf{G}_i^i & \cdots \end{bmatrix}$$

$$\mathbf{Y}_k = \begin{bmatrix} \cdots & \mathbf{H}_k^{ik} & \cdots & \mathbf{J}_k^{jk} & \cdots \end{bmatrix}.$$

C. 3D Line Endpoints Update

The line endpoints in 3D space are defined via two abscissas that need to be updated while the related line is updated. The update process is defined as follows:

- Transform $\mathcal{L}^w$ to $\mathcal{L}^c$;
- Compute two semi-infinite line equations $\mathbf{sl}_1$ and $\mathbf{sl}_2$ defined by the optical center C and the detected endpoint P_1 and P_2;
- Calculate the closest points on the Plücker line $\mathcal{L}^c$ to $\mathbf{sl}_1$ and $\mathbf{sl}_2$ so that we can get $P^c_{\mathcal{L}1}$ and $P^c_{\mathcal{L}2}$;
- Transform $P^c_{\mathcal{L}1}$ and $P^c_{\mathcal{L}2}$ to $P^w_{\mathcal{L}1}$ and $P^w_{\mathcal{L}2}$;
- Use $P^w_{\mathcal{L}i} = P^w_0 + t_i \cdot \mathbf{v}^w / \|\mathbf{v}^w\|$ to find t_i $(i = 1, 2)$, P^w_0 being the closest point from $\mathcal{L}^w$ to the origin of the world frame.

IV. Experimental Results

A. Simulations

Simulation allows to compare the estimated values against perfect ground truth and therefore to assess the errors of the estimations, and also allows comparisons with the EKF-based approach.

The scenario is shown in Fig. 4 that consists of a model of transmission tower built with 38 segments, and the height of tower is $7m$. Occlusions are not simulated and all the edges are visible. 90 frames are processed. A robot with one perspective camera (90°FOV, 640×480 pix resolution, 0.5 pix error) looking inwards to the model from a given distance of $5m$, gathering images. In order to observe the scale factor, the robot takes noisy odometry readings with errors. There is no loop closure in the simulation experiment. The end points of all lines are projected in all views, where their positions are corrupted by an additive Gaussian noise.

Fig. 4 shows the transmission tower being reconstructed by our proposed approach. It can be seen that the estimated lines are consistent with the true model.

In all the simulation experiments, the accuracy of the estimated camera pose is measured by computing the error of each component $(x, y, z, roll, pitch, yaw)$ between the estimated camera pose and the ground truth. Besides, the quality of reconstruction results is evaluated by computing the estimation error from (19) that describes the error between re-projected lines and measurement lines.

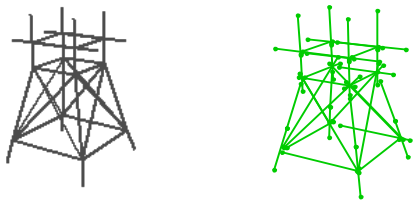


Fig. 4: Left: a simulated 3D transmission tower made of 38 segments; right: the reconstructed model.

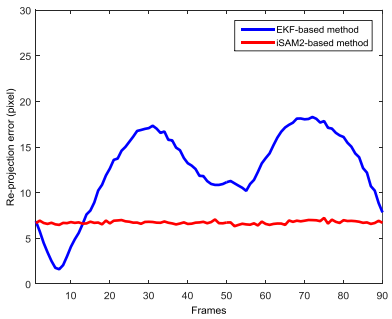


Fig. 5: The results of re-projection errors

Fig. 5 shows the re-projection errors of our proposed algorithm and EKF-based method. It gives the error for each frame. It can be seen that the re-projected error of our algorithm is less than the EKF-based method overall, and the curve of our method is much smoother than the curve of EKF-based method. So it demonstrates that the reconstruction accuracy of our algorithm outperforms the EKF-based method. But at the start, the error of EKF-based method is less, it is because in the simulation process, the number of line features be initialized increases gradually, on the other hand, at the begining of 38 frames, each frame adds a new initialized line to the map, the result curve indicates that the EKF-based SLAM might be beneficial when processing the small number of variables, then with the poses and landmarks increase, the inaccuracy accumulates.

Fig. 6 shows the comparison of the camera pose of the ground truth, our proposed SLAM, EKF-based SLAM and odometry. For the translation part, it can be seen that the differences between odometry estimation and ground truth is obvious as the green curve shows, the EKF-based SLAM is much better than the odometry result, whereas our iSAM2 based result is the closest to the ground truth one as the red curve shows. For the rotation part, the differences in Yaw is negligible, the estimated values of all methods are virtually identical, while the differences in $Roll$ and $Pitch$ indicate that our proposed algorithm works better than EKF-based method and odometry estimation.

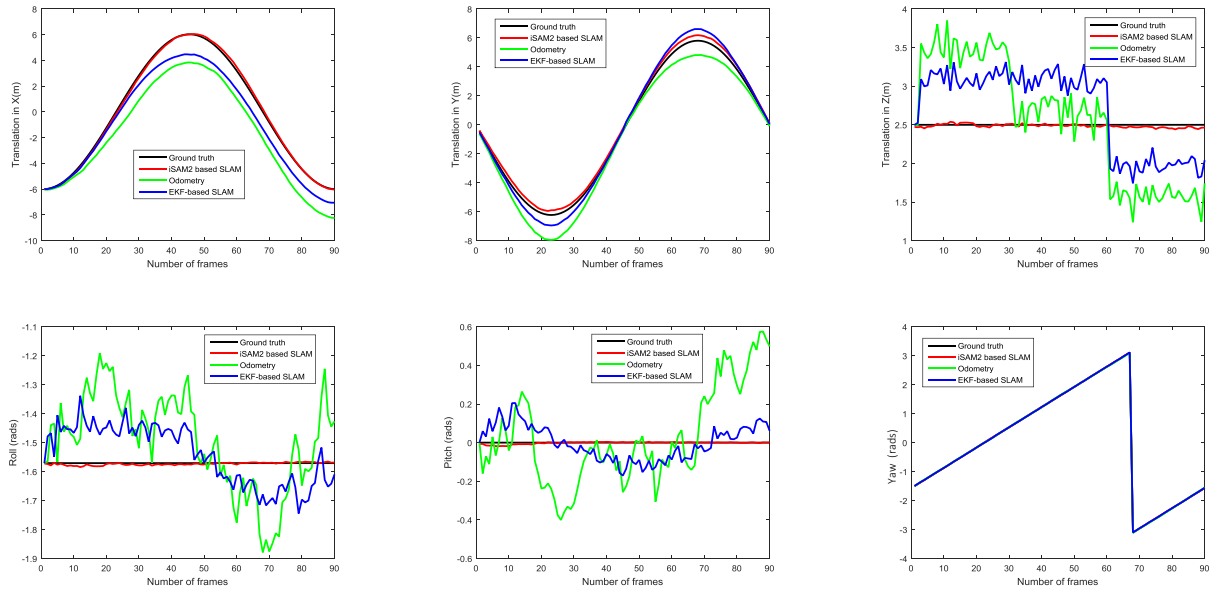
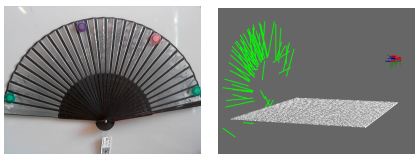


Fig. 6: Comparison of the ground truth, our proposed line-based SLAM, EKF-based SLAM, and odometry estimation

B. *Experiments on real image sequences*

To evaluate our proposed method with real images, the segments detector LSD (Line Segment Detector) [19] and the descriptor LBD (Line Band Descriptor)[20] have been used. LSD is a popular line detecting method, designed to work on any digital image without parameter tuning, while LBD works under various situations: Low-texture scenes, partial occlusion, rotation changes, blurred images, moderate viewpoint changes, and scale change. LSD and LBD are combined to ensure robust and reliable segment matches.

Two image sequences are tested in this paper. First image sequence is acquired with a web camera (Logitech Webcam C270, 640×480 pixels). A constant motion model is used to feed the prediction step of our SLAM approach since a really slow motion is operated with the camera. The scene contains a fan shown Fig. 7a: the reason why this scene has been chosen is that the lines of fan lie on the same plane. Fig. 7b shows the mapped 3D line segments and the estimated camera poses. The lines of the fan appear precisely reconstructed. In this experiment, there are 26 poses, 68 lines are reconstructed in 3D, and a total of 687 constraints or measurements. The proposed approach took about $20s$ to get the full results.



(a) Scene with a fan (b) Reconstructed fan

Fig. 7: Scene with a fan and reconstructed lines and poses.

The second image sequence comes from [21], it is recorded by UAV in a room of an approximate size of $8m \times 8.4m \times 4m$, which contains different objects, e.g. ladder, boxes and cabinets. The IMU data provided by the dataset is used to feed the prediction step of our SLAM approach using a strapdown inertial calculation. Fig. 8 shows the example images and the snapshot of the reconstructed scenes. The scene structures seem to be well recovered. In addition, the estimated trajectory is compared with the ground truth as Fig. 9 shows. It can be seen that it is close to the true path. The high reconstruction and estimation accuracy benefits from the unconstrained Cayley representation of line and the appropriate modified iSAM2 approach. During this experiment, there are 1074 poses estimated, 2229 lines are reconstructed in 3D, and a total of 11640 constraints or measurements. The proposed approach took about $840s$ to get the total results.

V. Conclusions and Future Works

This paper describes a novel line-based 6-DOF monocular SLAM algorithm which uses the iSAM2 approach. The difficult issue of representing the lines has been addressed by a combination of the Plücker Coordinates and the Cayley representation. An undelayed initialization method has been adopted to initialize the line features. Finally, the iSAM2 approach, originally designed for the point-based SLAM problem, has been extended to solve the line-based SLAM problem, which greatly improves the accuracy. Simulation experiments have been carried out to assess the accuracy of the proposed algorithm. The accuracy of the estimated pose result has been evaluated, and the reconstruction quality has been measured and shows the good accuracy of our algorithm. Meanwhile, the results have been compared with the EKF-based method and shows that the proposed algorithm outperforms the EFK-based methods. Experiments

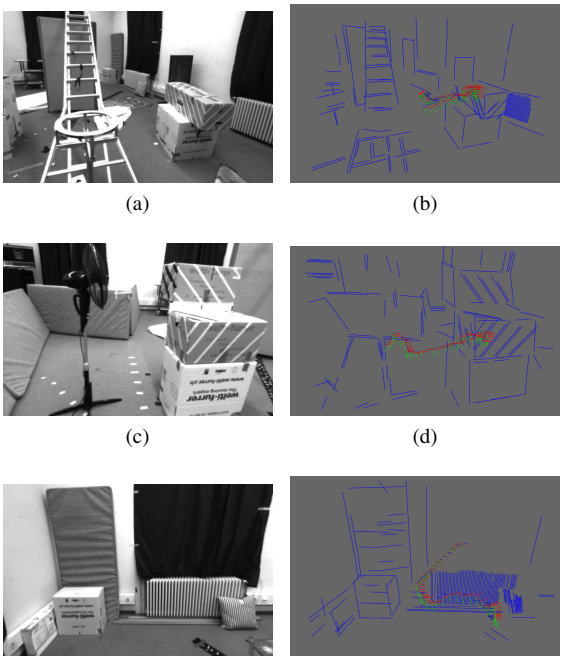


Fig. 8: The mapped scene. The first column are example images from the sequences and the second column shows the snapshots of the mapped scene.

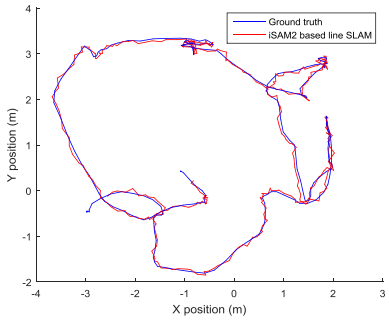


Fig. 9: The estimated trajectory

on real image sequences have also been performed, and promising result have been obtained. However, this work still needs some improvements on some aspects: First, the lines initialization must be handled carefully, and some initial depth information may help; Second, global loop-closure detection method should be added to enhance the robustness of the approach; Third, real-time requirements are crucial for SLAM applications, so the processing time of the proposed approach has to be optimized. In addition, we will focus on specific line-based SLAM applications like power tower inspection, railway analysis, or factory building inspection.

ACKNOWLEDGMENT

This work is part of the DIVINA challenge team, and was carried out in the framework of the Labex MS2T, which was funded by the French Government, through the program "Investments for the Future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02)

REFERENCES

[1] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1403–1410.

[2] E. Eade and T. Drummond, "Edge landmarks in monocular slam," *Image and Vision Computing*, vol. 27, no. 5, pp. 588–596, 2009.

[3] K. S. Roberts, "A new representation for a line," in *Computer Vision and Pattern Recognition, 1988. Proceedings CVPR'88., Computer Society Conference on*. IEEE, 1988, pp. 635–640.

[4] M. Ohwovoriole, "An extension of screw theory and its application to the automation of industrial assemblies," Ph.D. dissertation, Thesis, Mechanical Engineering, Standford University, 1980.

[5] L. Zhao, S. Huang, L. Yan, and G. Dissanayake, "A new feature parametrization for monocular slam using line features," *Robotica*, vol. 33, no. 03, pp. 513–536, 2015.

[6] L. Zhang and R. Koch, "Structure and motion from line correspondences: Representation, projection, initialization and sparse bundle adjustment," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 904–915, 2014.

[7] P. Smith, I. D. Reid, and A. J. Davison, "Real-time monocular slam with straight lines." in *BMVC*, vol. 6, 2006, pp. 17–26.

[8] A. P. Gee and W. Mayol-Cuevas, "Real-time model-based slam using line segments," in *International Symposium on Visual Computing*. Springer, 2006, pp. 354–363.

[9] J. Montiel, J. D. Tardós, and L. Montano, "Structure and motion from straight line segments," *Pattern Recognition*, vol. 33, no. 8, pp. 1295–1307, 2000.

[10] J. Weng, T. S. Huang, and N. Ahuja, "Motion and structure from line correspondences; closed-form solution, uniqueness, and optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 3, pp. 318–336, 1992.

[11] J. Sola, T. Vidal-Calleja, J. Civera, and J. M. M. Montiel, "Impact of landmark parametrization on monocular ekf-slam with points and lines," *International journal of computer vision*, vol. 97, no. 3, pp. 339–368, 2012.

[12] T. Lemaire and S. Lacroix, "Monocular-vision based slam using line segments," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 2791–2796.

[13] J. O. Esparza-Jiménez, M. Devy, and J. L. Gordillo, "Visual ekf-slam from heterogeneous landmarks," *Sensors*, vol. 16, no. 4, p. 489, 2016.

[14] J. Sola, T. Vidal-Calleja, and M. Devy, "Undelayed initialization of line segments in monocular slam," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 1553–1558.

[15] E. Eade and T. Drummond, "Scalable monocular slam," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1. IEEE, 2006, pp. 469–476.

[16] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular slam," *IEEE transactions on robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[17] F. Dellaert and M. Kaess, "Square root sam: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[18] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, p. 0278364911430419, 2011.

[19] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: a line segment detector," *Image Processing On Line*, vol. 2, pp. 35–55, 2012.

[20] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.

[21] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, 2016.