

Undelayed initialization of line segments in monocular SLAM

Joan Solà, Teresa Vidal-Calleja and Michel Devy

Abstract— This paper presents 6-DOF monocular EKF-SLAM with undelayed initialization using linear landmarks with extensible endpoints, based on the Plücker parametrization. A careful analysis of the properties of the Plücker coordinates, defined in the projective space \mathbb{P}^5 , permits their direct usage for undelayed initialization. Immediately after detection of a segment in the image, a Plücker line is incorporated in the map. A single Gaussian *pdf* includes inside its 2-sigma region all possible lines given the observed segment, from arbitrarily close up to the infinity range, and in any orientation. The lines converge to stable 3D configurations as the moving camera gathers observations from new viewpoints. The line's endpoints, maintained out of the map, are constantly retro-projected from the image onto the line's local reference frame. An extending-only policy is defined to update them. We validate the method via Monte Carlo simulations and with real imagery data.

I. INTRODUCTION

In the last years we have seen an important development of visual Simultaneous Localization And Mapping (SLAM) techniques. Most of these works have made use of point features because points are easy to extract, match, and represent. However, there is wide consensus on the fact that punctual world representations cannot provide satisfactory mapping results. Indeed, a map consisting of a sparse set of 3D points is far from describing the structure of the surrounding world. This is inherent to the dimensionless character of points, which contain no notion of *neighborhood* by themselves. Instead, segment-based landmarks include the one-dimensional notion of *connectivity*, and the two-dimensional notion of *boundary*, providing the map with a much richer representativeness of reality (Fig. 1). Connectivity and boundary information can be exploited to establish useful metrical and topological descriptions of the environment. This is effectively an important step forward for structured scenarios with plenty of straight lines. This work is relevant for SLAM applications intended for such scenarios.

We present a method for undelayed initialization (UI, otherwise named *partial initialization*) of straight lines in monocular extended Kalman filter (EKF) SLAM. UI allows low parallax landmarks, *i.e.*, those that are remote or close to the motion axis of the camera, to contribute to SLAM from the first observation. This allows the exploitation of the full field of view of the camera up to the infinity range, which results in accurate localizations with very low angular drifts. We encourage the reader to consult [1]–[4] for discussions on delayed/undelayed initializations and their importance in monocular SLAM.

For points, UI means that landmarks must be initialized so that the uncertainty in distance – the only non-measurable degree of freedom (DOF) – covers all the visual ray up

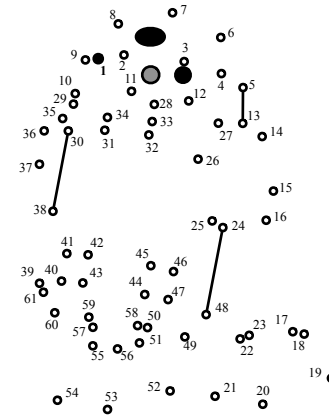


Fig. 1. What is this? To reveal it, join the dots with a piecewise line, starting from the black dot labeled “1” and proceeding in order. While there are (approximately) the same number of points and lines, only the line segments succeed in revealing a meaningful structure.

to infinity. EKF also requires that the observation function be reasonably linear within the whole range of uncertainty. These two severe requirements can be elegantly fulfilled by parametrizing distance by its inverse, leading to the method known as Inverse Depth Parametrization (IDP) [3], which can be considered the standard to this date.

The problems of lines and points are somewhat similar, and one of the aims of this paper is to make this similarity evident. Lines, however, present additional difficulties. For lines, UI requires the initial uncertainty to cover 2 non-measurable DOF: distance up to infinity, and all possible orientations. Unlike points, lines can be partially occluded, and most edge detectors produce very unstable endpoints. Therefore, the 3D segment's endpoints cannot be established from single observations. Also, line observations suffer from the aperture problem, which means that only the measured components that are orthogonal to the line are practicable. And still, in the Euclidean space a well defined measure of distance between lines is missing, making the definition of the innovation in EKF a delicate matter.

This work retakes the track started with [1], about undelayed initialization of points in monocular EKF-SLAM, which was improved by [2], [3] with the inverse depth parametrization (IDP). Some of these studies have since been extended to deal with segments, with either delayed [5] or undelayed [6] initialization. They share the remarkable asset of real-time operation. The points supporting the lines are coded using inverse depth parameters during the initialization phase, and converted to Cartesian representations after convergence. Being extensions of point-based systems, both methods suffer in our opinion from a somewhat poor

representation of the linear landmarks. Ref. [5] represents segments with their two endpoints, which are determined at the first observation and never updated. The inverse depth parameters are initially estimated by an external EKF until convergence, thus delaying initialization. Ref. [6] makes use of small edge landmarks, named ‘*edgelets*’, associated to a 3D point. Edgelets are typically 15 pixels long, and longer lines must be represented by several edgelets, compromising precisely the representativeness that we are seeking. Finally, during the initialization phase, [5] uses 11 parameters to represent the line, and [6] uses 9 parameters. After convergence, both systems switch to 6 parameters.

We represent infinite 3D lines with the Plücker coordinates, a 6-parameter parametrization well known in projective geometry. Plücker coordinates do not rely on point definitions, and are very easily manipulated: rotation, translation, join, incidence, intersection and pin-hole projection are resolved with exclusively additive, cross-product and dot-product operations. Using these operations we obtain a linear projection function that transforms Plücker lines in 3D space into homogeneous lines in the 2D image plane.

Our approach draws on [7], which performs SLAM with segments using the Euclidean Plücker coordinates (a subset of the Plücker coordinates where the direction vector is normalized) to map infinite lines. The drawback of [7] is a *delayed* initialization. The most remarkable insight that we provide is that, by using the Plücker coordinates alone (not the Euclidean Plücker), we naturally achieve inverse-depth behavior. This allows us to use Plücker lines in undelayed monocular SLAM directly. We keep track of the segment’s endpoints outside the map, and define for them an extending-only policy that adapts the method described in [7] to undelayed operation.

This paper is organized as follows. In Section II we highlight the most relevant properties of the Plücker coordinates. In Section III we present the initialization algorithm. We detail in Section IV the EKF updating procedures. In Section V the whole algorithm is benchmarked via simulations, and we present results with real images. We conclude in Section VI with a discussion.

II. THE PLÜCKER LINE COORDINATES

A. Notation

We deal with quite a few different representations for points, lines, motion matrices and projection matrices. The notation used is as follows. For readability, we write vector expressions within the text, *e.g.* $[\mathbf{A}^\top, \mathbf{B}^\top]^\top$, with the lighter form (\mathbf{A}, \mathbf{B}) . Points in Euclidean space \mathbb{E}^3 are specified with bold capitals, \mathbf{A} . Points in projective space \mathbb{P}^3 are underlined, $\underline{\mathbf{A}} = (\mathbf{A} : A) \sim (k\mathbf{A} : kA)$, where the colon ($:$) delimits the non-homogeneous part $\mathbf{A} \in \mathbb{R}^3$, A is a scalar and \sim denotes equivalence up to a scale factor k . Matrices are also in bold capitals (Plücker \mathbf{L} , motion \mathbf{H} , intrinsic \mathbf{K} , projection \mathbf{P}). In the projective image plane \mathbb{P}^2 we use non-capitalized fonts; points are underlined, $\underline{\mathbf{a}} = (\mathbf{a} : a)$, whereas lines are not, $\mathbf{l} = (l_1, l_2, l_3)$. In \mathbb{P}^5 we use calligraphic fonts; the Plücker line coordinates, \mathcal{L} , consist of two subvectors, $\mathcal{L} = (\mathbf{n} : \mathbf{v})$. The Plücker matrices: motion \mathcal{H} , intrinsic \mathcal{K} ,

and projection \mathcal{P} . Reference frames are also in calligraphic font, $\mathcal{F}, \mathcal{C}, \mathcal{O}$. Points and lines in a particular frame \mathcal{F} take this as a super-index, $\underline{\mathbf{A}}^{\mathcal{F}}, \mathcal{L}^{\mathcal{F}}$, except when it is irrelevant or it corresponds to the global frame. Vectors and matrices for frame transformation are in sans serif font: translation is \mathbf{T} , and we use quaternions \mathbf{Q} to represent rotation, so the rotation matrix is $\mathbf{R} = \mathbf{R}(\mathbf{Q})$. A frame pose is encoded as $\mathcal{F} = (\mathbf{T}, \mathbf{Q})$. We finally use $[\mathbf{a}]_\times$ for the 3×3 skew-symmetric matrix associated with the cross product, *i.e.*, $[\mathbf{a}]_\times \mathbf{b} \triangleq \mathbf{a} \times \mathbf{b}$.

B. Reminder. Homogeneous point manipulations in vision

The motion matrix \mathbf{H} transforms homogeneous points $\underline{\mathbf{A}}$ from camera frame \mathcal{C} to the global frame with

$$\underline{\mathbf{A}} = \mathbf{H} \cdot \underline{\mathbf{A}}^{\mathcal{C}} \triangleq \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \underline{\mathbf{A}}^{\mathcal{C}}. \quad (1)$$

In a pin-hole model of a perspective camera, points in the external \mathbb{P}^3 space are projected into the \mathbb{P}^2 image with

$$\underline{\mathbf{a}} = \mathbf{P} \cdot \underline{\mathbf{A}} = \mathbf{K} \cdot \mathbf{P}_0 \cdot \mathbf{H}^{-1} \cdot \underline{\mathbf{A}}, \quad (2)$$

with \mathbf{H}^{-1} the transformation to camera frame, \mathbf{P}_0 the canonical projection matrix, and \mathbf{K} the intrinsic matrix. Notice that when the point is expressed in camera frame, $\underline{\mathbf{A}}^{\mathcal{C}}$, only the non-homogeneous part appears in the projection expression,

$$\underline{\mathbf{a}} = \mathbf{K} \cdot \mathbf{A}^{\mathcal{C}}, \quad (3)$$

meaning that 1 DOF, the point’s range contained in $A^{\mathcal{C}}$, is not measurable.

C. The Plücker coordinates

A line in \mathbb{P}^3 can be defined from two points $\{\underline{\mathbf{A}}, \underline{\mathbf{B}}\} \in \mathbb{P}^3$ of the line by the *Plücker matrix*,

$$\mathbf{L} = \underline{\mathbf{A}} \cdot \underline{\mathbf{B}}^\top - \underline{\mathbf{B}} \cdot \underline{\mathbf{A}}^\top \in \mathbb{R}^{4 \times 4}. \quad (4)$$

This is a 4×4 skew symmetric matrix, with 6 independent entries $l_{ij} = -l_{ji}$, subject to the *Plücker constraint*,

$$\det(\mathbf{L}) = 0. \quad (5)$$

The Plücker matrix is independent of the two selected points of the line (more exactly, any two points of the same line give place to a matrix $\mathbf{L}' \sim \mathbf{L}$, *i.e.*, equivalent up to scale).

This line is coded as a homogeneous 6-vector $\mathcal{L} \in \mathbb{P}^5$ with the so called *Plücker coordinates*. These coordinates are any selection of the entries $\pm l_{ij}$, and have been defined in the literature in a number of different ways, some of them more fortunate (intuitive, easy to understand or manipulate) than others. It is handy to choose a representation $\mathcal{L} = (\mathbf{n} : \mathbf{v})$ [8] that corresponds to writing \mathbf{L} as

$$\mathbf{L} = \begin{bmatrix} [\mathbf{n}]_\times & -\mathbf{v} \\ \mathbf{v}^\top & 0 \end{bmatrix}, \quad \mathbf{n}, \mathbf{v} \in \mathbb{R}^3. \quad (6)$$

This yields $\mathcal{L} = (\mathbf{n} : \mathbf{v}) = (l_{32}, l_{13}, l_{21} : l_{41}, l_{42}, l_{43}) \in \mathbb{P}^5$, with 6 DOF. The Plücker constraint leads

$$\mathbf{n}^\top \mathbf{v} = 0, \quad (7)$$

leaving 5 DOF. Another DOF accounts for the irrelevant scale factor of \mathcal{L} . The remaining 4 DOF are required to minimally encode a line in 3D space [8].

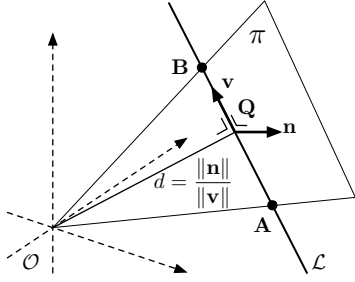


Fig. 2. Geometrical representation of the Plücker coordinates and sub-vectors \mathbf{n} and \mathbf{v} . Sub-vector \mathbf{v} exhibits an inverse-depth behavior which, proceeding with care, allows Plücker lines to be used directly for undelayed initialization.

The Plücker coordinates, when defined as in (6), admit a straightforward geometrical interpretation (in the Euclidean sense, Fig. 2):

- The vector \mathbf{n} is a vector normal to the plane π containing the line \mathcal{L} and the origin \mathcal{O} .
- The vector \mathbf{v} is a director vector of the line, oriented from \mathbf{A} to \mathbf{B} .
- The ratio $\|\mathbf{n}\|/\|\mathbf{v}\|$ is the Euclidean distance d from the origin to the line.
- The Plücker constraint trivially says that $\mathbf{n} \perp \mathbf{v}$.
- The point of the line closest to the origin is given by $\mathbf{Q} = (\mathbf{v} \times \mathbf{n})/\|\mathbf{v}\|^2$.

Remark 1 (Plücker and inverse depth): The third property above, saying $d = \|\mathbf{n}\|/\|\mathbf{v}\|$, is crucial for undelayed initialization in SLAM, notably because of the inverse depth behavior of the sub-vector \mathbf{v} . This is not possible with the Euclidean Plücker coordinates $\mathcal{L}^{\mathbb{E}} = (\mathbf{n} : \mathbf{u})$ in [7] because the director vector \mathbf{u} is normalized, $\|\mathbf{u}\| = 1$. See also Remark 2.

D. Plücker line manipulations

It is easy to see, via (1) and (4), that the Plücker matrix is transformed according to

$$\mathbf{L} = \mathbf{H} \cdot \mathcal{L}^c \cdot \mathbf{H}^T.$$

This expression is linear in the components of \mathcal{L}^c and therefore a linear expression exists for its vector counterpart \mathcal{L} . Having defined $\mathcal{L} = (\mathbf{n} : \mathbf{v})$, the expression of the transformation is amazingly simple [8]:

$$\mathcal{L} = \mathcal{H} \cdot \mathcal{L}^c \triangleq \begin{bmatrix} \mathbf{R} & [\mathbf{T}]_{\times} \mathbf{R} \\ 0 & \mathbf{R} \end{bmatrix} \cdot \mathcal{L}^c. \quad (8)$$

The inverse transformation is performed with

$$\mathcal{L}^c = \mathcal{H}^{-1} \cdot \mathcal{L} \sim \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T [\mathbf{T}]_{\times} \\ \mathbf{0} & \mathbf{R}^T \end{bmatrix} \cdot \mathcal{L}. \quad (9)$$

Similarly, the Plücker matrix is projected into a pin-hole camera according to

$$[\mathbf{l}]_{\times} = \mathbf{P} \cdot \mathbf{L} \cdot \mathbf{P}^T,$$

which is again linear in \mathbf{L} . The corresponding linear expression for the projected line in homogeneous coordinates, $\mathbf{l} \in \mathbb{P}^2$, is also very simple:

$$\mathbf{l} = \mathcal{P} \cdot \mathcal{L} = \mathcal{K} \cdot \mathcal{P}_0 \cdot \mathcal{H}^{-1} \cdot \mathcal{L}, \quad (10)$$

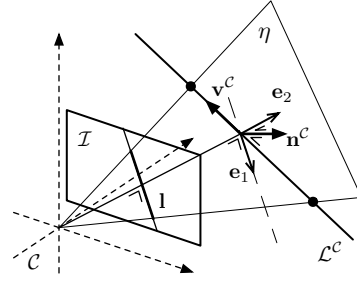


Fig. 3. Retro-projection of a segment in the image. The orthonormal base $\{\mathbf{e}_1, \mathbf{e}_2\}$ of the representation plane η permits the isolation of the 2DOF of the line's director vector \mathbf{v}^c . The base vector \mathbf{e}_1 is arbitrarily defined to be parallel to the image plane.

with intrinsic and canonical projection Plücker matrices

$$\mathcal{K} = \begin{bmatrix} \alpha_v & 0 & 0 \\ 0 & \alpha_u & 0 \\ -\alpha_v u_0 & -\alpha_u v_0 & \alpha_u \alpha_v \end{bmatrix}, \quad \mathcal{P}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

The whole transformation and projection process (10) can be expressed in terms of \mathbf{T} , \mathbf{Q} , \mathbf{n} and \mathbf{v} ,

$$\mathbf{l} = \mathcal{K} \cdot \mathbf{R}(\mathbf{Q})^T \cdot (\mathbf{n} - \mathbf{T} \times \mathbf{v}). \quad (11)$$

Notice that when the line is expressed in camera frame, \mathcal{L}^c , only the plane's normal \mathbf{n}^c appears in the projection expression,

$$\mathbf{l} = \mathcal{K} \cdot \mathbf{n}^c, \quad (12)$$

meaning that 2 DOF, the line's range and orientation contained in \mathbf{v}^c , are not measurable.

It is worth comparing (8) with (1), (10) with (2), and (12) with (3).

III. LINE INITIALIZATION

A. Line and uncertainty representations in \mathbb{E}^2 and \mathbb{P}^2

In order to specify the measurement noise in pixel units, we initially express segment measurements in the image plane with their two \mathbb{E}^2 endpoints \mathbf{p}_1 and \mathbf{p}_2 , with additive Gaussian noise $\mathcal{N}\{0; \mathbf{R}_{\mathbf{p}}\}$ each. When needed, the segment's *pdf* can be converted to homogeneous coordinates, $\mathbf{l} \sim \mathcal{N}\{\hat{\mathbf{l}}; \mathbf{\Lambda}\} \in \mathbb{P}^2$, with

$$\hat{\mathbf{l}} = \underline{\mathbf{p}}_1 \times \underline{\mathbf{p}}_2 \quad (13)$$

$$\mathbf{\Lambda} = \left[\underline{\mathbf{p}}_1 \right]_{\times} \cdot \underline{\mathbf{R}}_{\mathbf{p}} \cdot \left[\underline{\mathbf{p}}_1 \right]_{\times}^T + \left[\underline{\mathbf{p}}_2 \right]_{\times} \cdot \underline{\mathbf{R}}_{\mathbf{p}} \cdot \left[\underline{\mathbf{p}}_2 \right]_{\times}^T. \quad (14)$$

where $\underline{\mathbf{p}}_i = (\mathbf{p}_i : 1)$ and $\underline{\mathbf{R}}_{\mathbf{p}} = \text{diag}(\mathbf{R}_{\mathbf{p}}, 0)$.

B. Retro-projection of a line

A segment \mathbf{l} detected in an image \mathcal{I} uniquely determines the plane η containing the 3D line and the optical center \mathcal{C} (Fig. 3). This is named the *representation plane*. The plane's normal in camera frame is obtained by simply inverting (12),

$$\mathbf{n}^c = \mathcal{K}^{-1} \cdot \mathbf{l}. \quad (15)$$

The non-measured vector \mathbf{v}^c is meant to lie on the plane η . With the aim of isolating its two non-measurable DOF, we consider \mathbf{v}^c to be generated by a linear combination of the

vectors of an orthonormal base $\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2\}$ of the plane η , i.e.,

$$\mathbf{v}^c = \beta_1 \cdot \mathbf{e}_1 + \beta_2 \cdot \mathbf{e}_2, \quad \beta_1, \beta_2 \in \mathbb{R},$$

with $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{n}\}$ mutually orthogonal and $\|\mathbf{e}_1\| = \|\mathbf{e}_2\| = 1$. Doing $\mathbf{E} \triangleq [\mathbf{e}_1 \ \mathbf{e}_2] \in \mathbb{R}^{3 \times 2}$ and $\beta = (\beta_1, \beta_2)$ we get the matrix form

$$\mathbf{v}^c = \mathbf{E} \cdot \beta, \quad (16)$$

and $\mathbf{v}^c \subset \eta$ for any value of β . The base \mathcal{E} spans the null space of \mathbf{n}^c , thus the Plücker constraint is satisfied by construction.

For convenience, we arbitrarily build the base \mathcal{E} so that \mathbf{e}_1 is parallel to the image plane. This yields

$$\mathbf{e}_1 = \frac{[n_2^c \quad -n_1^c \quad 0]^\top}{\sqrt{(n_1^c)^2 + (n_2^c)^2}} \quad \text{and} \quad \mathbf{e}_2 = \frac{\mathbf{n}^c}{\|\mathbf{n}^c\|} \times \mathbf{e}_1. \quad (17)$$

The vector β admits the following geometrical interpretation (Fig. 4(a)):

- $\beta = (\beta_1, 0)$ is a line parallel to \mathbf{l} , thus to the image plane.
- $\beta = (0, \beta_2)$ is a line perpendicular to \mathbf{l} (but generally not to the image plane).
- The Euclidean distance from the line to the optical center \mathcal{C} is given by $\|\mathbf{n}^c\|/\|\beta\|$.

Remark 2 (Role of β): When $\beta \rightarrow (0, 0)$, the line tends to infinity. Its orientation is given by the relative strength of β_1 with respect to β_2 , and it easily covers the full circumference $[-\pi, \pi]$. Also, when $\|\mathbf{n}^c\| = 1$, the value $\|\beta\|$ is the inverse of the Euclidean distance from the line to the origin. See also Fig. 4 on initializing the *pdf* of β , together with step 2) in the algorithm that follows.

C. Initialization algorithm

Suppose we have a camera at location $\mathcal{C} = (\mathbf{T}, \mathbf{Q})$, with intrinsic Plücker matrix \mathcal{K} . \mathcal{C} has uncertainties encoded in the map, while \mathcal{K} is assumed to be deterministic. A segment $\{\mathbf{p}_1, \mathbf{p}_2\}$ is detected, then transformed to homogeneous coordinates with (13–14), and we wish to initialize it in a SLAM map. We proceed as follows.

1) *Identify the known magnitudes:*

- The SLAM map $X = (\mathcal{C}, \mathcal{M}) \sim \mathcal{N}\{\hat{X}; \mathbf{P}\}$ consists of

$$\hat{X} = \begin{bmatrix} \hat{\mathcal{C}} \\ \hat{\mathcal{M}} \end{bmatrix} \quad \text{and} \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathcal{C}\mathcal{C}} & \mathbf{P}_{\mathcal{C}\mathcal{M}} \\ \mathbf{P}_{\mathcal{M}\mathcal{C}} & \mathbf{P}_{\mathcal{M}\mathcal{M}} \end{bmatrix},$$

where \mathcal{C} is the camera pose and \mathcal{M} the set of landmarks. We define $\mathbf{P}_{\mathcal{C}X} \triangleq [\mathbf{P}_{\mathcal{C}\mathcal{C}} \ \mathbf{P}_{\mathcal{C}\mathcal{M}}]$.

- From (13–14), the measured line's *pdf* is $\mathbf{l} \sim \mathcal{N}\{\hat{\mathbf{l}}; \mathbf{\Lambda}\}$.

2) *Define the pdf of the non-measurable DOFs:* From arbitrary priors, build $\beta \sim \mathcal{N}\{\hat{\beta}; \mathbf{B}\}$, as follows (Fig. 4(b)):

- Initialize the line's mean at infinity with $\hat{\beta} \rightarrow (0, 0)$.
- Make sure the *pdf* is isotropic and contains, at e.g. 2σ , the lines at minimal distance, i.e. $2\sigma_\beta = 1/d_{min}$. Build covariances matrix $\mathbf{B} = \text{diag}(\sigma_\beta^2, \sigma_\beta^2)$.

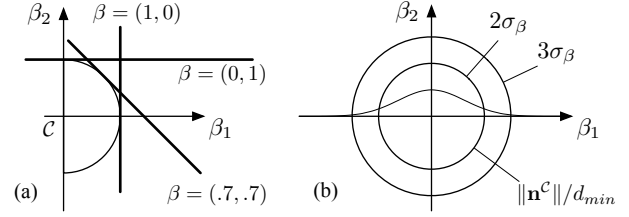


Fig. 4. Defining a *pdf* for β . (a) Different lines in the representation plane η (see Fig. 3) in camera frame \mathcal{C} , as a function of β . (b) The Gaussian *pdf* contains all possible lines at a minimum distance of d_{min} : it is isotropic in orientation, it includes the origin which represents the line at infinity, and d_{min} is at 2σ . For reference, a Gaussian shape is superimposed on the horizontal axis to evaluate the probability values at 2σ and 3σ .

3) *Compute the line estimate, $\hat{\mathcal{L}}$, and its Jacobians:*

- Using (15), compute the Plücker sub-vector in camera frame $\hat{\mathbf{n}}^c$ from the measured line $\hat{\mathbf{l}}$. Normalize it to a unit vector $\hat{\mathbf{n}}^{cn}$.
- Using (16–17), compute $\hat{\mathbf{v}}^c$ from $\hat{\mathbf{n}}^{cn}$ and $\hat{\beta}$.
- Build $\hat{\mathcal{L}}^c = (\hat{\mathbf{n}}^{cn}; \hat{\mathbf{v}}^c)$.
- Using (8), transform $\hat{\mathcal{L}}^c$ to the global frame. Obtain the line estimate $\hat{\mathcal{L}}$.
- Compute Jacobians $\mathbf{L}_c = \frac{\partial \mathcal{L}}{\partial \mathcal{C}}$, $\mathbf{L}_1 = \frac{\partial \mathcal{L}}{\partial \mathbf{l}}$ and $\mathbf{L}_\beta = \frac{\partial \mathcal{L}}{\partial \beta}$.

4) *Initialize the line in the map:*

- Compute the line's co- and cross-variances,

$$\begin{aligned} \mathbf{P}_{\mathcal{L}\mathcal{L}} &= \mathbf{L}_c \cdot \mathbf{P}_{\mathcal{C}\mathcal{C}} \cdot \mathbf{L}_c^\top + \mathbf{L}_1 \cdot \mathbf{\Lambda} \cdot \mathbf{L}_1^\top + \mathbf{L}_\beta \cdot \mathbf{B} \cdot \mathbf{L}_\beta^\top \\ \mathbf{P}_{\mathcal{L}X} &= \mathbf{L}_c \cdot \mathbf{P}_{\mathcal{C}X}. \end{aligned}$$

- Augment the SLAM map,

$$\hat{X} \leftarrow \begin{bmatrix} \hat{X} \\ \hat{\mathcal{L}} \end{bmatrix} \quad \text{and} \quad \mathbf{P} \leftarrow \begin{bmatrix} \mathbf{P} & \mathbf{P}_{\mathcal{L}X}^\top \\ \mathbf{P}_{\mathcal{L}X} & \mathbf{P}_{\mathcal{L}\mathcal{L}} \end{bmatrix}.$$

IV. LINE UPDATES

A. Plücker line update

The quality of the Plücker line update in EKF strongly depends on the degree of linearity of the observation function used. We already tackled the largest linearity issues by adopting the Plücker coordinates, which exhibit inverse depth behavior and linearity in the homogeneous observation functions. Our problem is that homogeneous coordinates are defined up to scale, preventing the innovation needed in EKF to be established by a simple subtraction. Converting the line to more suitable representations comes at the price of losing linearity, thus compromising EKF performance, and we need to proceed carefully.

The segment tracker returns two segment endpoints \mathbf{p}_1 and \mathbf{p}_2 . Due to the aperture problem, only the measurement components that are orthogonal to the expected line projection can be used for correction. Therefore, we define the innovation as a 2-vector containing the signed orthogonal distances from the endpoints to the expected line (Fig. 5). The signed distance z from a point $\underline{\mathbf{p}} = (u, v : 1)$ to a line $\mathbf{l} = (l_1, l_2, l_3)$ is given by

$$z = \mathbf{l}^\top \cdot \underline{\mathbf{p}} / \sqrt{l_1^2 + l_2^2},$$

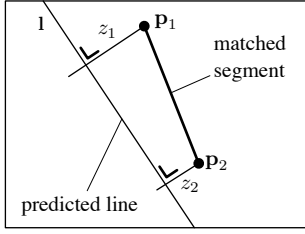


Fig. 5. Plücker line observation update. Direct measurement of the innovation $\mathbf{z} = (z_1, z_2)$ as the two signed orthogonal distances from the detected endpoints to the expected (or predicted) line.

so the innovation vector is

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{I}}^\top \cdot \mathbf{P}_1 / \sqrt{\hat{l}_1^2 + \hat{l}_2^2} \\ \hat{\mathbf{I}}^\top \cdot \mathbf{P}_2 / \sqrt{\hat{l}_1^2 + \hat{l}_2^2} \end{bmatrix} \in \mathbb{R}^2. \quad (18)$$

The above expression reveals the fact that the innovation can no longer be obtained from the subtraction $\mathbf{z} = \mathbf{y} - \mathbf{h}(\hat{X})$ we are familiar with in EKF, but from an implicit function $\mathbf{z} = \mathbf{g}(\hat{X}, \mathbf{y})$. This function is obtained by composing line transformation and projection (11) and the innovation measurements (18). The result is a somewhat complicated expression with a generic form

$$\mathbf{z} = \mathbf{g}(\hat{X}, \mathbf{y}) = \mathbf{g}(\hat{C}, \hat{L}, \mathbf{y}),$$

where \hat{C} and \hat{L} are the camera and line estimates from the map \hat{X} , and $\mathbf{y} = (u_1, v_1, u_2, v_2) \in \mathbb{R}^4$ is the measurement vector, with covariance $\mathbf{R} = \text{diag}(\mathbf{R}_p, \mathbf{R}_p)$. One detail that is worth noticing is the sign change in the Jacobians with respect to the habitual, explicit EKF innovation definition $\mathbf{g}(\hat{X}, \mathbf{y}) = \mathbf{y} - \mathbf{h}(\hat{X})$. It is clear that $\frac{\partial \mathbf{g}}{\partial \hat{X}} = -\frac{\partial \mathbf{h}}{\partial \hat{X}}$. With this sign change, and the non trivial contribution of \mathbf{y} in the innovation, the EKF correction equations have to be modified accordingly,

$$\mathbf{Z} = \mathbf{G}_X \cdot \mathbf{P} \cdot \mathbf{G}_X^\top + \mathbf{G}_y \cdot \mathbf{R} \cdot \mathbf{G}_y^\top \quad (19)$$

$$\mathbf{K} = -\mathbf{P} \cdot \mathbf{G}_X^\top \cdot \mathbf{Z}^{-1} \quad (20)$$

$$\hat{X} \leftarrow \hat{X} + \mathbf{K} \cdot \mathbf{z} \quad (21)$$

$$\mathbf{P} \leftarrow \mathbf{P} + \mathbf{K} \cdot \mathbf{G}_X \cdot \mathbf{P} \quad (= \mathbf{P} - \mathbf{K} \cdot \mathbf{Z} \cdot \mathbf{K}^\top), \quad (22)$$

with $\mathbf{G}_X = \frac{\partial \mathbf{g}}{\partial \hat{X}}$ and $\mathbf{G}_y = \frac{\partial \mathbf{g}}{\partial \mathbf{y}}$. An equivalent solution that permits reusing existing EKF code is to define $\mathbf{H} = -\mathbf{G}_X$ and $\mathbf{R}' = \mathbf{G}_y \mathbf{R} \mathbf{G}_y^\top$, and apply the regular EKF equations with the Jacobian \mathbf{H} and the new measurement noise \mathbf{R}' .

B. Segment endpoints update

The line's endpoints in 3D space are maintained out of the filter via two abscissas defined in the local 1D reference frame of the line, whose origin is at the point \mathbf{Q} (see Section II-C and Fig. 2). These abscissas are determined by back-projecting the detected segment's endpoints at each frame (Fig. 6(a)). They are updated following a different logic depending on the line having already converged or not. This is because, in an undelayed approach, the line estimates vary enormously (as much as from infinity to some close distance), and the abscissas are not stable upon large line modifications. While the line has not yet converged

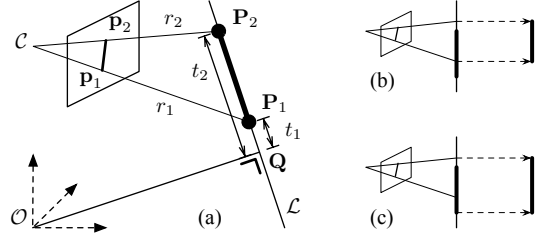


Fig. 6. Back-projection of the segments' endpoints. (a) The detected points \mathbf{p}_i and the optical center \mathcal{C} define two semi-infinite rays r_i . The closest points to these rays in \mathcal{L} are the 3D endpoints \mathbf{P}_i satisfying $\mathbf{P}_i = \mathbf{Q} + t_i \mathbf{v} / \|\mathbf{v}\|$. Only the abscissas t_i are stored. (b) Endpoints are systematically updated while the line is in convergence phase. (c) Once converged, endpoints are only updated if this lengthens the 3D segment.

(Fig. 6(b)), the abscissas are systematically updated, simply reflecting the last observation. Once the line has converged (Fig. 6(c)), an extending-only policy is applied: the abscissa is updated only if this lengthens the 3D segment. Line convergence is tested by projecting onto the image plane the current 3D endpoints' uncertainties. If the projected uncertainty in the line direction is lower than a certain threshold, the line is considered converged to a reasonably stable 3D position. The algebra behind all these operations is of no relevant interest and is not included here for space reasons. It can be found almost entirely in [7].

V. RESULTS

A. Simulations

We use simulations to benchmark the algorithms under controlled conditions. Simulation allows us to compare the estimated values against perfect ground truth and therefore to conclude on the consistency of the solution.

The simulated scenario consists of a wireframe of a house built with 27 segments (Fig. 7 shows the house being reconstructed). A robot with one perspective camera (90° FOV, 640×480 pix resolution, 0.5 pix error) looking forward approaches from a certain distance at 3 m/s, gathering images at 30 fps. In order to observe the scale factor, the robot takes noisy odometry readings with $1 \text{ cm}/\sqrt{\text{m}}$ and $0.25^\circ/\sqrt{\text{m}}$ error. The video attachment shows the reconstruction process in different conditions.

To evaluate the estimation's consistency, we perform a Monte Carlo analysis of the normalized estimation error squared (NEES) of the robot position, as explained in [9] (see Fig. 8). The averaged NEES after 50 runs shows good consistency up to frame 100 and a riskier behavior from then on. This is in accordance with [9], which concludes that long-term EKF-SLAM is always inconsistent, providing evidence of the necessity of approaches using multiple local maps (see [10]–[12] for solutions based on points). Our solution is therefore practicable within each one of these local maps. Possible improvements are discussed in Section VI.

B. Experiments

To evaluate our method with real images, we incorporated into our algorithms the segment detector and tracker described in [13]. The scene contains a 90° dihedral with

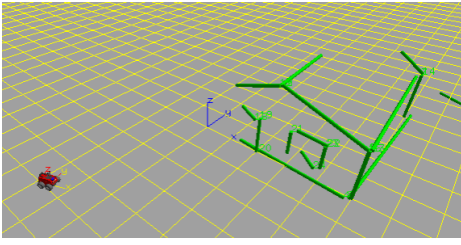


Fig. 7. The simulated environment consists of a robot approaching a wireframe of a house. The figure shows the partially reconstructed house, after approximately 40 frames have been processed. Despite the preliminary state of the map, the structure is already visible. With this fair amount of data, a map made of points would convey absolutely no information.

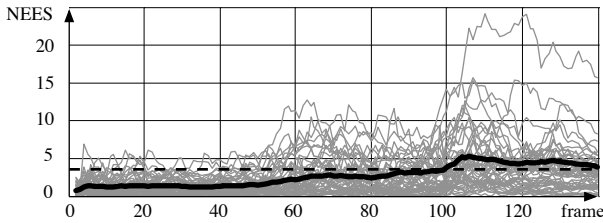


Fig. 8. 50 Monte Carlo runs (thin gray lines) showing the NEES measures of the robot position. The averaged NEES is in thick line. The dashed line corresponds to a NEES value of 3.59, which is the single-sided consistency limit of 95% confidence for 3 DOF and $N=50$ runs, *i.e.*, $\chi^2_{(3-50)}(1 - 0.95)/50 = 3.59$, see [9].

several segments on its planes (Fig. 9 left). The camera, controlled by a robotic arm, performs a 30×30 cm square trajectory perpendicularly to its optical axis. The position increments given by the arm are corrupted and used as odometry inputs to the system, thus providing the metrics for scale observability. The reconstructed scene is shown in Fig. 9 right, and in the third part of the video.

We evaluate the accuracy of the resulting map. We identify the two planes of the dihedral by optimally fitting them on the segments endpoints, and compute two different coplanarity errors. The first one is defined by the distances from the segments centers to their support plane, and is of 4.5mm STD (standard deviation). The second is defined by the angles between the segments and their support plane, of 0.56° STD. Finally, the angle between the two planes is of 89.72° .

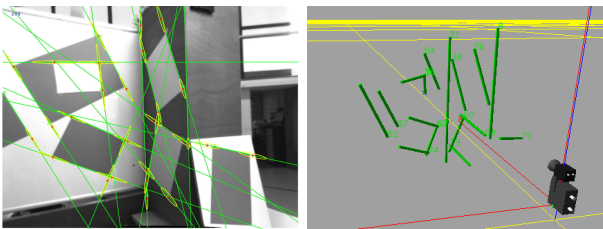


Fig. 9. Experimental setup. The camera executes a perfectly square trajectory in a plane perpendicular to its optical axis. A sample image (left) and the reconstructed structure (right).

VI. CONCLUSION AND DISCUSSION

We have presented an undelayed method to initialize infinite lines in monocular SLAM based on the Plücker coordinates. We highlighted the parallelisms between homogeneous points and Plücker lines, and saw that it is the inverse depth behavior of the homogeneous part what makes undelayed initialization possible. We defined the way to specify the priors of the line's two non-measured DOFs. We used an EKF update formulation that uses direct measurements of the lines innovations. Finally, we described a method to update the segment's endpoints. We presented a Monte Carlo analysis to evaluate the consistency of the estimated solution, and experimental results with real images.

The Plücker coordinates showed excellent capabilities for holding the line uncertainties during the first steps of undelayed initialization, when huge uncertainties combined with non-linearity challenge EKF. Further work must be done to investigate up to which point the Plücker coordinates, defined in the map frame, are able to withstand non-linearity that arises due to imprecise knowledge on the camera orientation, specially when the camera is far from the map origin. One way of improving linearity is by "anchoring" the Plücker lines at the point of first observation, as it is done with point landmarks in IDP [3]. This would increase the parametrization to a 9-vector, which could be switched back to a regular un-anchored 6-vector after passing a certain linearity test, also as in IDP.

REFERENCES

- [1] J. Solà, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only SLAM," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, Canada, 2005, pp. 2499–2504.
- [2] E. Eade and T. Drummond, "Scalable monocular SLAM," *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 469–476, 2006.
- [3] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. on Robotics*, vol. 24, no. 5, 2008.
- [4] J. Solà, A. Monin, M. Devy, and T. Vidal-Calleja, "Fusing monocular information in multi-camera SLAM," *IEEE Trans. on Robotics*, vol. 24, no. 5, pp. 958–968, 2008.
- [5] P. Smith, I. Reid, and A. Davison, "Real-time monocular SLAM with straight lines," in *British Machine Vision Conf.*, vol. 1, 2006, pp. 17–26.
- [6] E. Eade and T. Drummond, "Edge landmarks in monocular SLAM," in *British Machine Vision Conf.*, Edinburgh, Scotland, September 2006.
- [7] T. Lemaire and S. Lacroix, "Monocular-vision based SLAM using line segments," in *IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, 2007, pp. 2791–2796.
- [8] A. Bartoli and P. Sturm, "The 3D line motion matrix and alignment of line reconstructions," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 287–292.
- [9] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China, October 2006, pp. 3562–3568.
- [10] M. Bosse, P. Newman, J. Leonard, and S. Teller, "An Atlas framework for scalable mapping," in *IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 1899–1906.
- [11] C. Estrada, J. Neira, and J. Tardós, "Hierarchical SLAM: real-time accurate mapping of large environments," *IEEE Trans. on Robotics*, vol. 21, no. 4, pp. 588–596, 2005.
- [12] L. M. Paz, P. Piniés, J. Tardós, and J. Neira, "Large scale 6DOF SLAM with stereo-in-hand," *IEEE Trans. on Robotics*, vol. 24, no. 5, 2008.
- [13] P. Neubert, P. Protzel, T. Vidal-Calleja, and S. Lacroix, "A fast visual line segment tracker," in *IEEE Int. Conf. on Emerging Technologies and Factory Automation, Hamburg (Germany)*, Sept. 2008.