Impact of landmark parametrization on monocular EKF-SLAM with points and lines

Joan Solà, Teresa Vidal-Calleja, Javier Civera

Joan Solà, Teresa Vidal-Calleja, Javier Civera. Impact of landmark parametrization on monocular EKF-SLAM with points and lines. 2010. hal-00451778v2

HAL Id: hal-00451778
https://hal.science/hal-00451778v2

Preprint submitted on 5 Jun 2010 (v2), last revised 17 Sep 2011 (v4)

Impact of landmark parametrization on monocular EKF-SLAM with points and lines

Joan Solà, Teresa Vidal-Calleja and Javier Civera

***Abstract*— This paper aims at providing answers to a variety of questions regarding undelayed initialization of point- and line-landmarks in monocular EKF-SLAM. It does so by stating the problem and the necessary conditions for a successful solution, and by offering a compendium of landmark parametrizations, including three different methods for points and five other methods for straight lines. Some of these methods are already well-known, others are presented here for the first time. The discourse of the text is chosen to highlight the similarities and the differences between them, and how all these parametrizations can be seen as incremental variations of ones with respect to others. We present a new linearity measure to predict how well a parametrization will perform in an EKF framework. To confirm these predictions, we benchmark all the methods by running Monte-Carlo tests on all algorithms, using the same data and comparing their performances on filter RMS errors and NEES consistency. Finally, some results with real imagery are presented as a means to further analyze their possibilities in view of real implementations.**

TODO: *Remove TOC - check intro and abstract for real contents - check paper outline description -*

I. INTRODUCTION

Monocular simultaneous localization and mapping (SLAM) gained popularity back in 2003 thanks to a real-time implementation due to Davison [1], based on Smith *et al.*'s original extended Kalman filter (EKF) solution to SLAM [2]. Davison's technique elegantly solved a great number of problems, but there still remained one that occupied researchers on visual SLAM for some years [3]–[5]: the problem of landmark initialization. Monocular EKF-SLAM reached maturity with the advent of undelayed landmark initialization techniques (ULI), a need of partial initialization firstly stated in 2005 by Solà *et al.* [6], with a preliminary solution based on a multi-hipothesized depth parametrization, which was inspired on a previous work in 2004 by Kwok *et al.* [7]. The problem was successfully solved for the first time in 2006 with the inverse-depth landmark parametrization (IDP) due to Montiel *et al.* [8].

The present work aims at contributing several concepts related to landmark parametrization in monocular, EKF-based SLAM implementing ULI. The goal is to improve the system's linearity, and thus filter consistency, to achieve more robust and accurate localization and mapping. We present eight different methods, three for points and five for lines; we highlight their similarities and differences, analyze their degree of linearity, and compare their performances using both analytical and statistical tools, in both simulated and real setups.

A. Undelayed landmark initialization (ULI)

ULI (also known as *partial initialization*) is the process by which landmarks that have been partially measured (*e.g.* with bearings-only —see Fig. 1— or range-only sensors) are incorporated into the SLAM map at the time of the first observation, that is, before their full 3D state can be determined. In bearings-only systems such as those based on a monocular camera, ULI allows landmarks showing low parallax (those that are at remote distances or close to the motion axis of the camera) to contribute to SLAM from the first observation, contributing precious bearing information that helps constraining the camera orientation. This allows the exploitation of the full field of view of the camera up to the infinity range, regardless of the sensor trajectory, which results in accurate localizations with very low angular drifts. We encourage the reader to consult [6], [9], [10] for discussions on delayed/undelayed initializations and their importance in monocular SLAM.

In brief, ULI in EKF frameworks must be conceived with two key properties, namely

1) The full uncertainty range of the non-measured degrees of freedom (DOF) must be properly represented by a Gaussian *pdf*.
2) The subsequent updating procedures must be robust and accurate when performed with standard EKF machinery.

In order to simultaneously fulfill these requirements with a simple EKF, some degree of landmark over-parametrization is required. On one hand, a non-linear mapping is needed for the DOF encoding distance to transform a bounded Gaussian in parametrization space into an unbounded density in map space. In bearings-only systems this leads to parametrizations incorporating these DOFs proportional to inverse-distance. On the other hand, we need to make the observation functions as linear as possible. The inverse-distance helps here too, but additional over-parametrization may also show beneficial as we will see.

B. Points and straight lines

The problems of lines and points are somewhat similar, and one of the aims of this paper is to make this similarity evident.

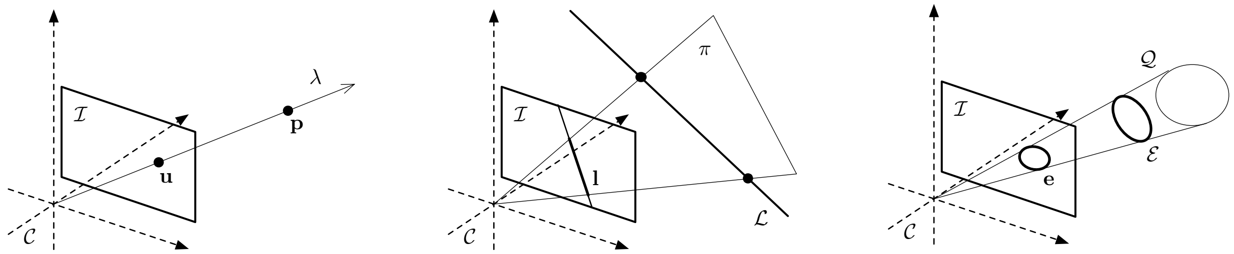
For points, ULI means that landmarks must be initialized so that the uncertainty in distance – the only non-measurable

---

J. Solà is at University of Toulouse, LAAS-CNRS, France. `jsola@laas.fr`

T. VIdal-Calleja is at University of Sydney, ACFR, Australia. `tvidal@gmail.com`

J. CIvera is at Universidad de Zaragoza, DIIS-CPS, Spain. `jcivera@unizar.es`



(a) The retro-projection of a point $\mathbf{u}$ gives place to a semi-infinite line λ where the point landmark $\mathbf{p}$ must lie. There is 1 unmeasured DOF: the point's depth.

(b) The retro-projection of a segment $\mathbf{l}$ gives place to a semi-infinite plane π where the line landmark $\mathcal{L}$ must lie. There are 2 unmeasured DOF: the line's depth and its orientation in π.

(c) The retro-projection of an ellipse $\mathbf{e}$ gives place to a semi-infinite cone $\mathcal{Q}$ where the elliptic landmark $\mathcal{E}$ must lie. There are 3 unmeasured DOF: the ellipse's plane depth and its two attitude angles.

Figure 1. The problem of undelayed initialization. Retro-projection of detected features in a monocular image $\mathcal{I}$ at their first observation. The unmeasured DOF's have infinite uncertainty and need to be properly modeled and handled by Gaussian shapes and reasonably linear functions. This paper explores the possibilities of undelayed initialization of points and straight lines, but not of ellipses.

DOF – covers all the visual ray up to infinity. The problem knows today two main solutions: *inverse-depth points* (IDP, [8], [9]), and *homogeneous points* (HP, [11]).

Lines present additional difficulties. For lines, ULI 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. The only ULI solution for infinite lines we are aware of is [12], which uses the Plücker line. Delayed EKF approaches include [13]–[15]. Edgelets (very short line fragments) are used in [16] in an undelayed manner. Edgelets are also used in [17], but not with the EKF but with bundle adjustment techniques.

C. Conics and planes

Points and straight lines are not the only geometrical primitives that one might use. In perspective cameras, points project into points, straight lines into straight lines, and conics such as ellipses project into conics (Fig. 1). In panoramic cameras governed by the Barreto model [18], points project into points and straight lines into ellipses. It is convenient in all these cases to base our observation models in the functions that relate the geometric parameters defining the shapes in 3D space to the ones in the 2D images. But a fundamental difference appears when trying to use planes as primitives. In perspective cameras a plane does project into a plane, but this plane is absolutely coincident with the image plane: the observed plane parameters are always the same and therefore non-informative. This seems a counter-intuitive paradox saying "we cannot observe 3D planes with a projective device". It is not: planes are observable, but they must be observed indirectly, by observing either primitives of the previous kinds (points, lines, curves) that lie on their surface or that delimit them, or some dense information such as planar textured patches. The observation models for planes must be based on indirect methods and cannot be conceived with the same class of algorithms as the ones for points or curves. We leave the problem of planes, and the question of whether the ideas presented in this paper are extrapolable to them, out of consideration. Besides, our aim is not to be exhaustive, and we further restrict our study to the cases of points and straight lines projected into perspective cameras, which we judge to be the ones with the clearest practical applications.

D. Alternative approaches

Other authors investigated the possibilities of using different estimation techniques. We have seen IDP used in FastSLAM2.0 [19] and UKF [20], [21] frameworks; and methods based on bundle adjustment [22], [23], which is at this moment the technique that shows the clearest future perspectives [24]. These works are often motivated by inconsistency and computational burden issues associated with EKF-SLAM. However, EKF-SLAM (or other similar approaches based on filtering and Gaussians such as UKF- or EIF-SLAM) is still widely used by major robotics and vision laboratories and is at the core of other localization, mapping or modeling systems [15], [25]–[27]. Two reasons in our opinion keep it alive: its (relative) simplicity of implementation, and the fact that large maps are usually being built by means of small sub-maps, thus circumventing most of the EKF drawbacks.

E. Contributions

We claim several contributions. First, a compendium of eight landmark parametrizations especially conceived for ULI, three for points and five for lines, where five of them are innovative to our best knowledge. Second, these parametrizations are presented in a unified discourse that emphasizes the two keys to satisfactory ULI, namely landmark anchoring and inverse-distance behavior. These two keys have already been proved positive, *e.g.* in [9], [11], but they have not been purposely conceptualized and explored to the limit. Third, an analytical measure of linearity for EKF that takes into account the time-varying support of probability. Fourth, a statistical analysis of root mean squared (RMS) errors and normalized estimation error squared (NEES) consistency based on Monte-Carlo simulation runs. The analytical and statistical approaches presented here should allow the reader to correlate the discourse of the paper with the linearity indices and with the performance of each parametrization. We illustrate the two most performing methods with a couple of experiments using real imagery. A final discussion serves to relocate the EKF-SLAM algorithm among other close-by solutions, namely robo-centric monocular EKF-SLAM [11], [26], and boards some secondary but important aspects regarding one or other solution.

F. Outline

In Section II we describe three parametrizations for points and give details on the necessary algebra to support them. In Section III we repeat the process with five types of infinite lines. Section IV describes the initialization and updating procedures in standard EKF-SLAM with ULI using points or lines. Section V introduces the analytical and statistical methods we use for linearity and consistency evaluation, with results presented in Section VI. The paper continues with a discussion in Section VII and the conclusions in Section VIII.

II. POINTS

This section presents some parametrizations for 3D points, with their projection and back-projection operations needed for EKF-SLAM initialization and updating. We start with Euclidean points (EP) just as a matter of completeness, and to introduce some notation. The discourse evolves through homogeneous points (HP), anchored homogeneous points (AHP), and inverse-distance points (IDP), which we will refer to here as anchored modified-polar points (AMPP) for reasons that will be explained soon.

A. Euclidean points (EP)

A *Euclidean point* $\mathbf{p}$ (EP, Fig. 2(a)) is trivially coded with three Cartesian coordinates

$$\mathcal{P}_{\mathbb{E}} = \mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^\top \in \mathbb{R}^3$$

Transformation to camera frame and perspective (pin-hole) projection are performed with the well-known expression

$$\underline{\mathbf{u}} = \mathbf{K}\mathsf{R}^\top(\mathbf{p} - \mathsf{T}) \in \mathbb{P}^2, \tag{1}$$

which we use to introduce some notation. Underlined fonts $\underline{\bullet}$ indicate homogeneous coordinates; $\mathbf{K}$ is the intrinsic matrix,

$$\mathbf{K} \triangleq \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix};$$

$\mathsf{R} = \mathsf{R}(\mathsf{Q})$ and T are the rotation matrix and the translation vector defining the camera frame $\mathcal{C}$, which is coded by the vector $\mathcal{C} = (\mathsf{T}, \mathsf{Q})$, Q being an orientation representation of our choice suitable for EKF filtering (we use quaternions).

Euclidean points lead to severely non-linear observation functions in bearings-only systems and are not suited for undelayed initialization, as it has been extensively demonstrated, [3], [7], [10], [11], [19] and most particularly [6], [9]. In brief, the problem can be described as follows. In EKF, the requirements of function linearity must be stated with respect to the uncertainty region. Because in Euclidean parametrizations the uncertainty region of partially observed landmarks is of infinite size, the observation functions' linearity should hold for an infinite variation of the non-measured DOF, which is not the case. The parametrizations that we present mitigate this problem by transforming the unbounded uncertainty region into a bounded one, and can be employed for ULI with just a few precautions.

B. Homogeneous points (HP)

A *homogeneous point* (HP, Fig. 2(b)) is coded by a 4-vector in projective space $\mathbb{P}^3$. It is composed of a 3D vector $\mathbf{m}$ and a scalar ρ, usually referred to as the *homogeneous part*,

$$\mathcal{P}_{\mathbb{H}} = \underline{\mathbf{p}} = \begin{bmatrix} \mathbf{m} \\ \rho \end{bmatrix} = \begin{bmatrix} u & v & w & \rho \end{bmatrix}^\top \in \mathbb{R}^4. \tag{2}$$

A HP refers to the following EP:

$$\mathbf{p} = \mathbf{m}/\rho. \tag{3}$$

Remark 1 (Inverse-distance): The scalar ρ depends linearly with the inverse of the distance d from the origin to the 3D point,

$$\rho = \|\mathbf{m}\|/d. \tag{4}$$

The unbounded distance uncertainty, which spans from a minimal distance d_{min} to infinity, is coded by a bounded interval in $\rho \in (0, \|\mathbf{m}\|/d_{min}]$. This is of central importance as this is precisely the factor that will allow us to use such parametrization for undelayed initialization in EKF-SLAM (see Section IV-B for further justification and details). The same concept of inverse-distance is found in absolutely all the parametrizations we present (except of course EP), even the ones for lines.

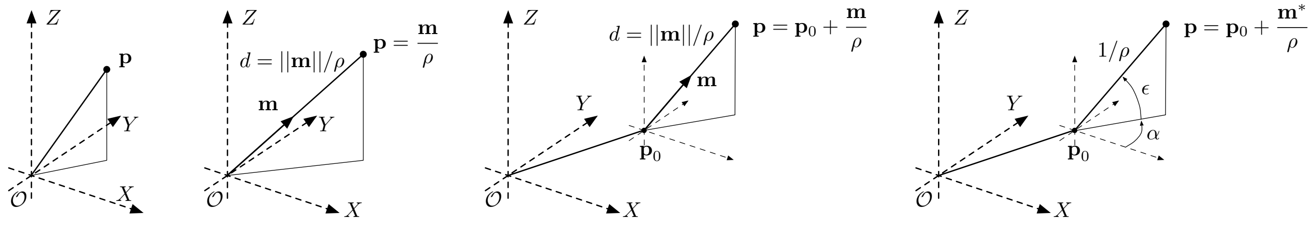
Homogeneous points have the additional interesting property of presenting a bi-linear transformation equation:

$$\underline{\mathbf{p}} = \mathbf{H}\underline{\mathbf{p}}^{\mathcal{C}} \triangleq \begin{bmatrix} \mathsf{R} & \mathsf{T} \\ 0 & 1 \end{bmatrix} \underline{\mathbf{p}}^{\mathcal{C}}, \tag{5}$$

where the super-index $\bullet^{\mathcal{C}}$ indicates the frame $\mathcal{C}$ where the point is referred to, and $\mathbf{H}$ is the homogeneous motion matrix. Bi-linearity becomes linearity when the localization information (T, R) is known perfectly. A strict linearity would allow us to transform covariances from and to different reference frames with absolutely no degradation, contributing to good EKF performances. If we dispose of a non-perfect but accurate estimate of (T, R), *e.g.* in cases of small localization uncertainty, the resulting quasi-linearity makes the EKF solution to SLAM practicable.

Homogeneous points project to perspective cameras according to the linear expression

$$\underline{\mathbf{u}} = \mathbf{P}\underline{\mathbf{p}} = \mathbf{K}\mathbf{P}_0\mathbf{H}^{-1}\underline{\mathbf{p}}, \tag{6}$$



(a) Euclidean point (b) Homogeneous point (HP) (c) Anchored homogeneous point (AHP) (d) Anchored modified-polar point (AMPP)

Figure 2. Point parametrizations. Homogeneous and anchored homogeneous do not require $\mathbf{m}$ to be a unit vector. In modified-polar point, the observed ray is coded by two angles: the derived direction vector is unitary and hence ρ is exactly inverse-distance. The anchor point $\mathbf{p}_0$ in AHP and AMPP corresponds to the optical center at initialization time.

with $\mathbf{P} \triangleq \mathbf{K}\mathbf{P}_0\mathbf{H}^{-1}$, and $\mathbf{P}_0$ the canonical projection matrix

$$\mathbf{P}_0 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

This can be expressed in terms of T, R, $\mathbf{m}$ and ρ,

$$\underline{\mathbf{u}} = \mathbf{K}\mathsf{R}^\top(\mathbf{m} - \mathsf{T}\rho) \in \mathbb{P}^2. \tag{7}$$

Notice that when the point is expressed in camera frame, $\underline{\mathbf{p}}^{\mathcal{C}} = (\mathbf{m}^{\mathcal{C}}, \rho^{\mathcal{C}})$, only the non-homogeneous part $\mathbf{m}^{\mathcal{C}}$ appears in the projection expression,

$$\underline{\mathbf{u}} = \mathcal{K}\cdot\mathbf{m}^{\mathcal{C}}, \tag{8}$$

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

The back-projection and transformation composition necessary for landmark initialization is done with

$$\mathcal{P}_{\mathbb{H}} = \underline{\mathbf{p}} = \begin{bmatrix} \mathbf{m} \\ \rho \end{bmatrix} = \mathbf{H} \begin{bmatrix} \mathbf{K}^{-1}\underline{\mathbf{u}} \\ \rho^{\mathcal{C}} \end{bmatrix}, \tag{9}$$

where $\rho^{\mathcal{C}}$ depends inversely with the distance $d^{\mathcal{C}}$ to the camera, via $\rho^{\mathcal{C}} = \|\mathbf{K}^{-1}\underline{\mathbf{u}}\|/d^{\mathcal{C}}$. Being $\rho^{\mathcal{C}}$ not measured, it must be provided as prior (see Section IV-B). Once transformed to the global frame with $\mathbf{H}$, this meaning of $\rho^{\mathcal{C}}$ is lost and ρ becomes the inverse-distance to the global origin $\mathcal{O}$.

C. Anchored homogeneous points (AHP)

We add an anchor to the HP parametrization to improve linearity, as it is done in the well-known inverse-depth parametrization [9], which we will see later. Anchoring the HP means referring it to a point $\mathbf{p}_0$ in 3D space different from the origin (Fig. 2(c)). The *anchor point* $\mathbf{p}_0$ is chosen to be the optical center at initialization time. This leads to the *anchored homogeneous point* (AHP, Fig. 2(c)), parametrized with the 7-vector

$$\mathcal{P}_{\mathbb{AH}} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{m} \\ \rho \end{bmatrix} = \begin{bmatrix} x_0 & y_0 & z_0 & u & v & w & \rho \end{bmatrix}^\top \in \mathbb{R}^7 \tag{10}$$

Remark 2 (Landmark anchor): The effect of anchoring is that, on subsequent EKF updates, only the accumulated errors from the anchor $\mathbf{p}_0$ to the current camera position T will be considered, in contrast with regular HP where the error accounts for the absolute motion of the sensor from the origin of coordinates. This results in lower linearization errors and therefore to more consistent filters, which is one of the conclusions of this work.

An AHP refers to the following EP:

$$\mathbf{p} = \mathbf{p}_0 + \mathbf{m}/\rho. \tag{11}$$

Remark 3 (Direction vector): The homogeneous point of the AHP, $(\mathbf{m}, \rho) \in \mathbb{P}^3$, does not require $\mathbf{m}$ to be a unit vector. If it is not, the parametrization is absolutely valid but ρ is then not the inverse distance $1/d$ but proportional to it, $\rho = \|\mathbf{m}\|/d$.

Transformation to camera frame and projection resume to

$$\underline{\mathbf{u}} = \mathbf{K}\mathsf{R}^\top\big(\mathbf{m} - (\mathsf{T} - \mathbf{p}_0)\rho\big) \in \mathbb{P}^2. \tag{12}$$

The back-projection and transformation composition is done with

$$\mathcal{P}_{\mathbb{AH}} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{m} \\ \rho \end{bmatrix} = \begin{bmatrix} \mathsf{T} \\ \mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}} \\ \rho^{\mathcal{C}} \end{bmatrix}, \tag{13}$$

where $\rho^{\mathcal{C}}$ must be provided as prior; its relation to distance d is given by $\rho^{\mathcal{C}} = \|\mathbf{K}^{-1}\underline{\mathbf{u}}\|/d$.

D. Anchored modified-polar points (AMPP)

We lighten the previous AHP parametrization by encoding the direction vector $\mathbf{m}$ with just elevation and azimuth angles (ε, α) of the observed optical ray joining $\mathbf{p}_0$ to $\mathbf{p}$. When these angles are appended with the inverse of the distance ρ, the result is a 3D point in modified-polar coordinates, $(\varepsilon, \alpha, 1/d)$. Adding the anchor $\mathbf{p}_0$ leads to the *anchored modified-polar point* (AMPP, Fig. 2(d)), coded by the 6-vector

$$\mathcal{P}_{\mathbb{AMP}} = \begin{bmatrix} \mathbf{p}_0 \\ (\varepsilon, \alpha) \\ \rho \end{bmatrix} = \begin{bmatrix} x_0 & y_0 & z_0 & \varepsilon & \alpha & \rho \end{bmatrix}^\top \in \mathbb{R}^6 \tag{14}$$

Remark 4 (Inverse-depth points): In this article we will refer to the originally named “inverse depth” points (IDP) in [9] as *anchored modified-polar points* (AMPP). There is absolutely no difference between IDP and AMPP, and the name change is justified by two facts: on one hand, our name better explains the nature of the parametrization as it recalls the previously existing “modified polar coordinates” term[1]. On the other hand, all our parametrizations share the concept of inverse-depth (or inverse-distance), rendering the term “IDP” ambiguous and non-informative.

An AMPP refers to the following EP:

$$\mathbf{p} = \mathbf{p}_0 + \mathbf{m}^*(\varepsilon, \alpha)/\rho \tag{15}$$

where $\mathbf{m}^*(\varepsilon, \alpha)$ is a unit vector in the direction of (ε, α),

$$\mathbf{m}^*(\varepsilon, \alpha) = \begin{bmatrix}\cos(\varepsilon)\cos(\alpha) & \cos(\varepsilon)\sin(\alpha) & \sin(\varepsilon)\end{bmatrix}^\top. \tag{16}$$

Transformation to camera frame and pin-hole projection operations resume to

$$\underline{\mathbf{u}} = \mathbf{K}\mathsf{R}^\top\big(\mathbf{m}^*(\varepsilon, \alpha) - (\mathsf{T} - \mathbf{p}_0)\rho\big). \tag{17}$$

The back-projection and transformation composition is performed with

$$\mathcal{P}_{\mathbb{AMP}} = \begin{bmatrix}\mathbf{p}_0 \\ (\varepsilon, \alpha) \\ \rho\end{bmatrix} = \begin{bmatrix}\mathsf{T} \\ \mu^*(\mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}}) \\ \rho^{\mathcal{C}}\end{bmatrix}, \tag{18}$$

where $\mu^*(\mathbf{m})$ gives elevation and azimuth angles (ε, α) of a director vector $\mathbf{m} = (m_x, m_y, m_z)$,

$$\begin{bmatrix}\varepsilon \\ \alpha\end{bmatrix} = \mu^*(m_x, m_y, m_z) = \begin{bmatrix}\arctan(m_z/\sqrt{m_x^2 + m_y^2}) \\ \arctan(m_y/m_x)\end{bmatrix}. \tag{19}$$

The parameter $\rho^{\mathcal{C}}$ is now exactly the inverse-distance $1/d$ because in AMPP the vector $\mathbf{m}^*$ is always unitary. It is defined in the camera frame at initialization time and, being non-measurable, must be provided as prior.

III. INFINITE STRAIGHT LINES

This section mimics the structure of Section II for the case of infinite straight lines (refer to Section IV-E for details on defining segment endpoints). We remark the numerous parallelisms that can be established among them, and also between points and lines. We start with a quite exhaustive introduction to the Plücker line (PL), that behaves surprisingly similar to HP, and where the concept of inverse-distance is associated to a 3D vector instead of a scalar. The discourse evolves through the anchored Plücker line (APL), the homogeneous-points line (HPL), the anchored homogeneous-points line (AHPL), and the anchored modified-polar-points line (AMPPL).

A. Plucker lines (PL)

This sub-section devoted to the Plücker line is long. We decided to include all this material because it is important here to highlight several interesting connections between homogeneous points (HP) and Plücker lines (PL), notably the existence of linear transformation and projection equations reproducing the structure of those for HP, and the inverse-distance behavior of the homogeneous part of the Plücker vector. These connections will clearly arise with the adoption of a discourse that retraces the one we used for HP. They allow us to propose the Plücker line as an interesting starting candidate for undelayed initialization in monocular EKF-SLAM. Most of the material here can be found in [12], [28].

1) The Plücker coordinates: A line in $\mathbb{P}^3$ can be defined from two points of the line by the *Plücker matrix*,

$$\mathbf{L} = \underline{\mathbf{b}}\cdot\underline{\mathbf{a}}^\top - \underline{\mathbf{a}}\cdot\underline{\mathbf{b}}^\top \in \mathbb{R}^{4\times4}, \tag{20}$$

with $\underline{\mathbf{a}} = (\mathbf{a}, a) \in \mathbb{P}^3$ and the same for $\underline{\mathbf{b}}$. This is a 4×4 skew-symmetric matrix (with 12 non-null off-diagonal entries $l_{ij} = -l_{ji}$) subject to the *Plücker constraint*,

$$\det(\mathbf{L}) = 0. \tag{21}$$

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 $\mathcal{L}_{\mathbb{P}} \in \mathbb{P}^5$ with the so called *Plücker coordinates*. These coordinates are any linearly-independent 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. In order to make the similarities with HP visible, it is handy to choose the representation suggested in [28], that we will name here the *Plücker line* (PL, Fig. 3(a)),

$$\mathcal{L}_{\mathbb{P}} = \begin{bmatrix}\mathbf{n} \\ \mathbf{v}\end{bmatrix} = \begin{bmatrix}n_x & n_y & n_z & v_x & v_y & v_z\end{bmatrix}^\top \in \mathbb{P}^5 \subset \mathbb{R}^6, \tag{22}$$

which corresponds to writing $\mathbf{L}$ as

$$\mathbf{L} = \begin{bmatrix}[\mathbf{n}]_\times & \mathbf{v} \\ -\mathbf{v}^\top & 0\end{bmatrix}, \quad \mathbf{n}, \mathbf{v} \in \mathbb{R}^3, \tag{23}$$

with $[\mathbf{n}]_\times$ the skew-symmetric matrix associated with the cross-product (*i.e.*, $[\mathbf{n}]_\times \mathbf{m} \equiv \mathbf{n} \times \mathbf{m}$),

$$[\mathbf{n}]_\times \triangleq \begin{bmatrix}0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0\end{bmatrix}. \tag{24}$$

This choice and the definition (20) allow us to write

$$\mathbf{n} = \mathbf{a}\times\mathbf{b} \tag{25}$$

$$\mathbf{v} = a\mathbf{b} - b\mathbf{a}, \tag{26}$$

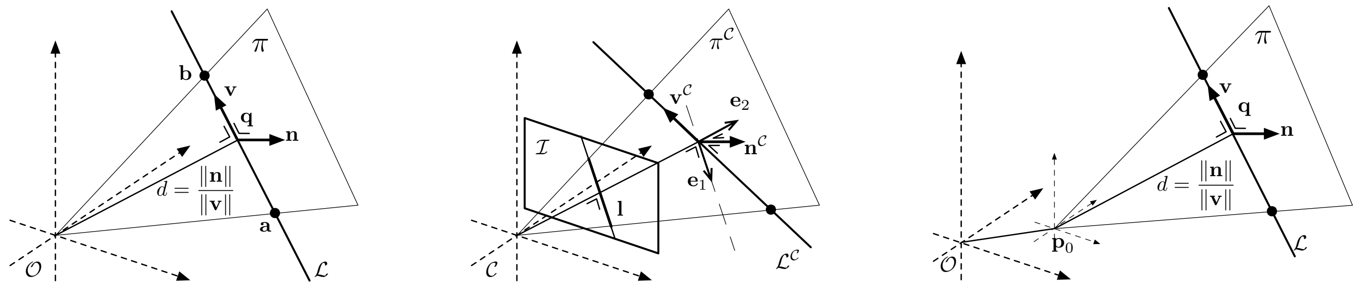
with which the Plücker constraint becomes

$$\mathbf{n}^\top\mathbf{v} = 0. \tag{27}$$

The Plücker coordinates, when defined as in (25–26), admit a comprehensible geometrical interpretation (in the Euclidean sense, Fig. 3(a)):

---

[1] We do not study here the modified-polar point (MPP) as we judge it uninteresting for EKF-SLAM. Refer to the discussion in Section VII for further justification.



(a) Plücker line (PL). The line $\mathcal{L}$ and the origin $\mathcal{O}$ define the support plane π. The line's sub-vector $\mathbf{n} \in \mathbb{R}^3$ is orthogonal to π. The sub-vector $\mathbf{v} \in \mathbb{R}^3$ is a director vector of the line, and lies on π.

(b) Back-projection of a Plücker line. The orthogonal base $\{\mathbf{e}_1, \mathbf{e}_2\}$ of $\pi^{\mathcal{C}}$ permits the isolation of the 2DOF of the line's director vector $\mathbf{v}^{\mathcal{C}}$. The base vector $\mathbf{e}_1$ is parallel to the image plane.

(c) Anchored Plücker line (APL). The closest point to the anchor is $\mathbf{q} = \mathbf{p}_0 + (\mathbf{v}\times\mathbf{n})/(\mathbf{v}^\top\mathbf{v})$.

Figure 3. Geometrical interpretations of Plücker-based lines with their back-projection details. The 3-vector $\mathbf{v}$ is not observable at initialization time. Its initial covariance, however, must be defined in the plane $\pi^{\mathcal{C}}$ by means of a 2D Gaussian prior β. See Figs. 8 and 9 for further details.

- The vector $\mathbf{n}$ is a vector normal to the plane π containing the line $\mathcal{L}_{\mathbb{P}}$ (hence the points $\mathbf{a}$ and $\mathbf{b}$) 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 orthogonal distance d from the line $\mathcal{L}$ to the origin $\mathcal{O}$.
- The Plücker constraint trivially says that $\mathbf{n} \perp \mathbf{v}$.
- The point of the line closest to the origin is given by $\mathbf{q} = (\mathbf{v}\times\mathbf{n})/\|\mathbf{v}\|^2 \in \mathbb{E}^3$ or $\underline{\mathbf{q}} = (\mathbf{v}\times\mathbf{n}, \mathbf{v}^\top\mathbf{v}) \in \mathbb{P}^3$.

Remark 5 (Plücker and inverse-distance): The third property above, saying $d = \|\mathbf{n}\|/\|\mathbf{v}\|$, is crucial for undelayed initialization in SLAM, notably because of the inverse-distance 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 [13] because the director vector $\mathbf{u}$ is normalized, $\|\mathbf{u}\| = 1$. See also Remark 6.

---

2) Frame transformations and projection: It is easy to see, via (5) and (20), that the Plücker matrix is transformed according to

$$\mathbf{L} = \mathbf{H}\cdot\mathbf{L}^{\mathcal{C}}\cdot\mathbf{H}^\top.$$

This expression is linear in the components of $\mathbf{L}^{\mathcal{C}}$ and therefore a linear expression exists for its vector counterpart $\mathcal{L}_{\mathbb{P}}$. Having defined $\mathcal{L}_{\mathbb{P}} = (\mathbf{n}, \mathbf{v})$, the expression of the transformation is amazingly simple [28]:

$$\mathcal{L}_{\mathbb{P}} = \mathcal{H}\cdot\mathcal{L}_{\mathbb{P}}^{\mathcal{C}} \triangleq \begin{bmatrix} \mathsf{R} & [\mathsf{T}]_\times \mathsf{R} \\ 0 & \mathsf{R} \end{bmatrix} \cdot \mathcal{L}_{\mathbb{P}}^{\mathcal{C}}. \tag{28}$$

The inverse transformation is performed with

$$\mathcal{L}_{\mathbb{P}}^{\mathcal{C}} = \mathcal{H}^{-1}\cdot\mathcal{L}_{\mathbb{P}} \sim \begin{bmatrix} \mathsf{R}^\top & -\mathsf{R}^\top [\mathsf{T}]_\times \\ \mathbf{0} & \mathsf{R}^\top \end{bmatrix} \cdot \mathcal{L}_{\mathbb{P}}. \tag{29}$$

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

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

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:

$$\mathbf{l} = \mathcal{P}\cdot\mathcal{L}_{\mathbb{P}} = \mathcal{K}\cdot\mathcal{P}_0\cdot\mathcal{H}^{-1}\cdot\mathcal{L}_{\mathbb{P}}, \tag{30}$$

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 (30) can be expressed in terms of T, R, $\mathbf{n}$ and $\mathbf{v}$,

$$\mathbf{l} = \mathcal{K}\cdot\mathsf{R}^\top\cdot(\mathbf{n} - \mathsf{T}\times\mathbf{v}). \tag{31}$$

Notice that when the line is expressed in camera frame, $\mathcal{L}_{\mathbb{P}}^{\mathcal{C}} = (\mathbf{n}^{\mathcal{C}}, \mathbf{v}^{\mathcal{C}})$, only the plane's normal $\mathbf{n}^{\mathcal{C}}$ appears in the projection expression,

$$\mathbf{l} = \mathcal{K}\cdot\mathbf{n}^{\mathcal{C}}, \tag{32}$$

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

We can now fully observe the revealing parallelisms between PL and HP by comparing equations (28) with (5), (30) with (6), (31) with (7) and (32) with (8). Roughly speaking, we see that PL's $\mathbf{n}$ plays the role of HP's $\mathbf{m}$, and $\mathbf{v}$ plays the role of ρ. We will exploit this fact to achieve ULI operation.

3) Pin-hole retro-projection: A segment $\mathbf{l}$ detected in an image $\mathcal{I}$ uniquely determines the plane $\pi^{\mathcal{C}}$ containing the 3D line and the optical center $\mathcal{C}$ (Fig. 3(b)). This is named the *representation plane*. The plane's normal in camera frame $\mathbf{n}^{\mathcal{C}}$ is obtained by simply inverting (32),

$$\mathbf{n}^{\mathcal{C}} = \mathcal{K}^{-1}\cdot\mathbf{l}. \tag{33}$$

The vector $\mathbf{v}^{\mathcal{C}}$ is meant to lie on the plane $\pi^{\mathcal{C}}$ and has therefore only 2DOF, which are not measured. We need to isolate them to be able to provide the necessary Gaussian prior for initialization. For this, we consider $\mathbf{v}^{\mathcal{C}}$ to be generated by a linear combination of the vectors of an orthogonal base $\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2\}$ of the plane $\pi^{\mathcal{C}}$, *i.e.*,

$$\mathbf{v}^{\mathcal{C}} = \beta_1\cdot\mathbf{e}_1 + \beta_2\cdot\mathbf{e}_2, \qquad \beta_1, \beta_2 \in \mathbb{R},$$

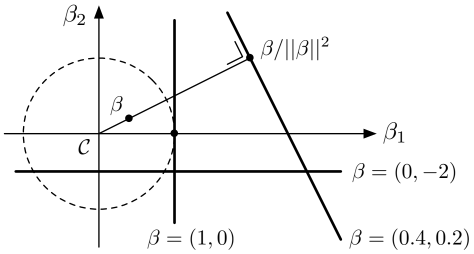


Figure 4. Different lines in the representation plane π (see Fig. 3(b)) in camera frame $\mathcal{C}$, as a function of β. The circle is of unit radius. Given β, the line is such that its closest point to $\mathcal{C}$ is $\beta/\|\beta\|^2$. The line is orthogonal to the vector β. See also Fig. 9.

with $\{\mathbf{e}_1, \mathbf{e}_2\} \perp \mathbf{n}^{\mathcal{C}}$. Doing $\mathbf{E} \triangleq [\mathbf{e}_1 \; \mathbf{e}_2] \in \mathbb{R}^{3\times2}$ and $\beta = (\beta_1, \beta_2) \in \mathbb{R}^2$ we get the matrix form

$$\mathbf{v}^{\mathcal{C}} = \mathbf{E}\cdot\beta, \tag{34}$$

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

For convenience, we arbitrarily build the base $\mathcal{E}$ so that $\|\beta\|$ is exactly inverse-distance and $\mathbf{e}_1$ is parallel to the image plane. This yields

$$\mathbf{e}_1 = \frac{\begin{bmatrix} n_2^{\mathcal{C}} & -n_1^{\mathcal{C}} & 0 \end{bmatrix}^\top}{\sqrt{(n_1^{\mathcal{C}})^2 + (n_2^{\mathcal{C}})^2}} \cdot \|\mathbf{n}^{\mathcal{C}}\| \quad \text{and} \quad \mathbf{e}_2 = \frac{\mathbf{n}^{\mathcal{C}} \times \mathbf{e}_1}{\|\mathbf{n}^{\mathcal{C}}\|}. \tag{35}$$

With this base choice the vector β admits the following geometrical interpretation (Fig. 4):

- $\beta = (\beta_1, 0)$ is a line parallel to $\mathbf{l}$, thus to the image plane, passing over the point $D = (1/\beta_1, 0)$.
- $\beta = (0, \beta_2)$ is a line perpendicular to $\mathbf{l}$ (but generally not to the image plane), passing over the point $D = (0, 1/\beta_2)$.
- $\beta = (\beta_1, \beta_2)$ is a line in he direction of $(\beta_2, -\beta_1)$ passing over the point $D = \beta/\|\beta\|^2$ which is the point of the line closest to the optical center.
- The orthogonal Euclidean distance from the line to the optical center $\mathcal{C}$ is given by $1/\|\beta\|$.

Remark 6 (Role of β): The planar β-space is well-suited for defining our Gaussian prior. When $\beta \to (0, 0)$, 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 $[-\pi, \pi]$. The value $\|\beta\|$ is the inverse of the Euclidean distance from the line to the origin. When assigning a prior *pdf* to β at initialization time (see Fig. 8 in Section IV on initializing the *pdf* of β), this will be properly mapped to the 3D space as a planar *pdf* on the plane $\pi^{\mathcal{C}}$. The support of high probability of this *pdf* covers from a specified minimal distance to infinity.

Summarizing, retro-projection and transformation are performed with

$$\mathcal{L}_{\mathbb{P}} = \mathcal{H}\begin{bmatrix} \mathcal{K}^{-1}\mathbf{l} \\ \mathbf{E}\beta \end{bmatrix} = \begin{bmatrix} \mathsf{R}\mathcal{K}^{-1}\mathbf{l} + [\mathsf{T}]_\times \mathsf{R}\mathbf{E}\beta \\ \mathsf{R}\mathbf{E}\beta \end{bmatrix}, \tag{36}$$

where β must be provided as prior.

B. Anchored Plucker lines (APL)

As we did with points, we add an anchor to the Plücker parametrization to improve linearity. The *anchored Plücker line* (APL, Fig. 3(c)) is then the 9-vector:

$$\mathcal{L}_{\mathbb{AP}} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{n} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} x_0 \; y_0 \; z_0, \; n_x \; n_y \; n_z, \; v_x \; v_y \; v_z \end{bmatrix}^\top \in \mathbb{R}^9 \tag{37}$$

Transformation and projection are accomplished by transforming the line to the camera frame, un-anchoring it, and projecting it into the pin-hole camera. This can be done in one single expression with:

$$\mathbf{l} = \mathcal{K}\cdot\mathsf{R}^\top\cdot(\mathbf{n} + (\mathbf{p}_0 - \mathsf{T})\times\mathbf{v}) \in \mathbb{P}^2, \tag{38}$$

in which we will notice:

- The linear character with respect to $\mathbf{n}$.
- For accurate estimates of $(\mathsf{T} - \mathbf{p}_0)$, which is true for observations shortly after initialization, the linear character also with respect to the non-observed $\mathbf{v}$, which additionally exhibits inverse-distance behavior.

Retro-projection and transformation resume to

$$\mathcal{L}_{\mathbb{A}P} = \begin{bmatrix} \mathsf{T} \\ \mathsf{R}\mathcal{K}^{-1}\mathbf{l} \\ \mathsf{R}\mathbf{E}\beta \end{bmatrix}, \tag{39}$$

where β must be provided as prior.

C. Homogeneous-points lines (HPL)

This and the following parametrizations are based on the fact that a line in 3D space can be represented by two points supporting it. We will use the point parametrizations explored in Section II to build lines, in the hope that this will preserve most of the properties of the formers.

A *homogeneous-points line* (HPL, Fig. 5(a)) is coded by two HP that support it:

$$\mathcal{L}_{\mathbb{H}} = \begin{bmatrix} \mathbf{m}_1 \\ \rho_1 \\ \mathbf{m}_2 \\ \rho_2 \end{bmatrix} = \begin{bmatrix} u_1 \; v_1 \; w_1 \; \rho_1 \; u_2 \; v_2 \; w_2 \; \rho_2 \end{bmatrix}^\top \in \mathbb{R}^8 \tag{40}$$

Transformation and pin-hole projection require the projection of the two support points, *i.e.* for $i \in \{1, 2\}$,

$$\underline{\mathbf{u}}_i = \mathbf{K}\mathsf{R}^\top\big(\mathbf{m}(\varepsilon_i, \alpha_i) - \mathsf{T}\rho_i\big).$$

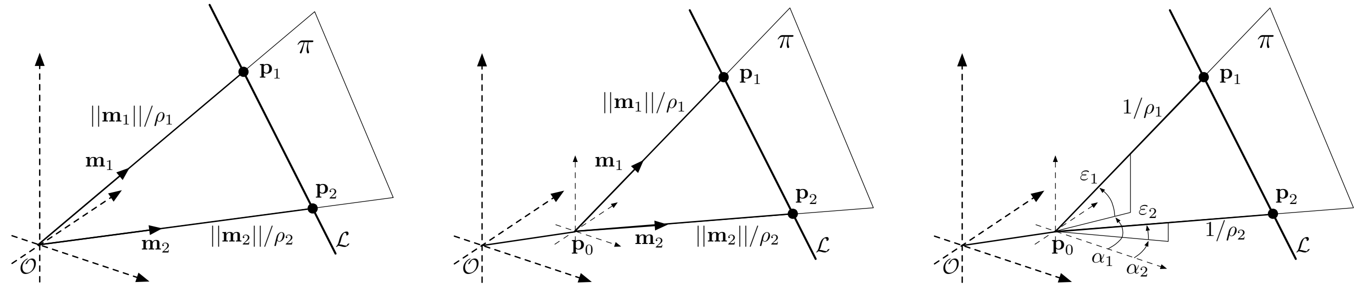
This expression (which is obviously equal to (7)) may be practical to design appropriate updating algorithms as it contains information about the segment's support points in the image. However, for the sake of comparing HPL against other line parametrizations, we join the two projected points into a homogeneous 2D line,

$$\mathbf{l} = \underline{\mathbf{u}}_1 \times \underline{\mathbf{u}}_2. \tag{41}$$

This yields[2]

$$\mathbf{l} = \mathcal{K}\mathsf{R}^\top\Big((\mathbf{m}_1 \times \mathbf{m}_2) - \mathsf{T}\times(\rho_1\mathbf{m}_2 - \rho_2\mathbf{m}_1)\Big). \tag{42}$$

[2] To prove (42) we use the distributive property of the cross-product, the identity $(\mathbf{Ma})\times(\mathbf{Mb}) = \det(\mathbf{M})\mathbf{M}^{-\top}(\mathbf{a}\times\mathbf{b})$, the fact that regular and Plücker intrinsic matrices are related by $\mathcal{K} \propto \mathbf{K}^{-\top}$, and remind that $\mathbf{l} \in \mathbb{P}^2$ and is therefore invariant to proportionality transforms.



(a) Homogeneous-points line (HPL)

(b) Anchored homogeneous-points line (AHPL)

(c) Anchored modified-polar-points line (AMPPL)

Figure 5. Point-supported lines. The lines are defined by two support points like the ones in Section II. The anchor, if it exists, is common to both points.

This last expression is important in the sense that it allows us to observe the parallelisms between parametrizations. Comparing HPL (42) against PL (31), and remembering equations (25–26) defining the Plücker sub-vectors, we observe that:

- The product $\mathbf{m}_1 \times \mathbf{m}_2$ is a vector orthogonal to the plane π, and it can be identified with the PL sub-vector $\mathbf{n}$.
- The term $(\rho_1 \mathbf{m}_2 - \rho_2 \mathbf{m}_1)$ is a vector joining the two support points of the line. It is therefore its director vector and can be identified with the PL sub-vector $\mathbf{v}$.
- With these two identifications, equations (31) and (42) coincide (using (25–26) this coincidence can be easily proved to hold exactly).

Retro-projection and transformation resume to

$$\mathcal{L}_{\mathbb{H}} = \begin{bmatrix} \mathbf{H} \begin{pmatrix} \mathbf{K}^{-1}\underline{\mathbf{u}}_1 \\ \rho^{\mathcal{C}} \end{pmatrix} \\ \mathbf{H} \begin{pmatrix} \mathbf{K}^{-1}\underline{\mathbf{u}}_2 \\ \rho^{\mathcal{C}} \end{pmatrix} \end{bmatrix}, \tag{43}$$

where $\rho^{\mathcal{C}}$ must be provided as prior.

D. Anchored homogeneous-points lines (AHPL)

The *anchored homogeneous-points line* (AHPL, Fig. 5(b)) can be built either by adding an anchor to HPL or by joining two AHP with a shared anchor to support it:

$$\mathcal{L}_{\mathbb{AHP}} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{m}_1 \\ \rho_1 \\ \mathbf{m}_2 \\ \rho_2 \end{bmatrix} = [x_0, y_0, z_0,\ u_1, v_1, w_1, \rho_1,\ u_2, v_2, w_2, \rho_2]^\top \in \mathbb{R}^{11} \tag{44}$$

Transformation and pin-hole projection require the projection of the two support points $\underline{\mathbf{u}}_1$ and $\underline{\mathbf{u}}_2$, which are joined into a homogeneous-points line, $\mathbf{l} = \underline{\mathbf{u}}_1 \times \underline{\mathbf{u}}_2$. As before, this can be rearranged as

$$\mathbf{l} = \mathcal{K}\mathsf{R}^\top \Big((\mathbf{m}_1 \times \mathbf{m}_2) - (\mathsf{T} - \mathbf{p}_0) \times (\rho_1 \mathbf{m}_2 - \rho_2 \mathbf{m}_1) \Big) \tag{45}$$

where the same parallelisms that we highlighted between PL and HPL can now be observed between APL and AHPL: equations (38) and (45) are equivalent after identifying $(\mathbf{m}_1 \times \mathbf{m}_2)$ with $\mathbf{n}$ and $(\rho_1 \mathbf{m}_2 - \rho_2 \mathbf{m}_1)$ with $\mathbf{v}$.

Retro-projection and transformation resume to

$$\mathcal{L}_{\mathbb{AHP}} = \begin{bmatrix} \mathsf{T} \\ \mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}}_1 \\ \rho^{\mathcal{C}} \\ \mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}}_2 \\ \rho^{\mathcal{C}} \end{bmatrix}, \tag{46}$$

where $\rho^{\mathcal{C}}$ must be provided as prior.

E. Anchored modified-polar-points lines (AMPPL)

The *anchored modified-polar-points line* (AMPPL, Fig. 5(c)) is coded by two AMPP that support it, which share a common anchor:

$$\mathcal{L}_{\mathbb{ID}} = \begin{bmatrix} \mathbf{p}_0 \\ (\varepsilon_1, \alpha_1) \\ \rho_1 \\ (\varepsilon_2, \alpha_2) \\ \rho_2 \end{bmatrix} = \begin{bmatrix} x_0 & y_0 & z_0 & \varepsilon_1 & \alpha_1 & \rho_1 & \varepsilon_2 & \alpha_2 & \rho_2 \end{bmatrix}^\top \tag{47}$$

Transformation and projection resume to

$$\mathbf{l} = \mathcal{K}\mathsf{R}^\top \Big((\mathbf{m}_1^* \times \mathbf{m}_2^*) - (\mathsf{T} - \mathbf{p}_0) \times (\rho_1 \mathbf{m}_2^* - \rho_2 \mathbf{m}_1^*) \Big) \tag{48}$$

where we used the shortcut $\mathbf{m}_i^* \triangleq \mathbf{m}(\varepsilon_i, \alpha_i)$ which corresponds to the trigonometric transform (16).

Retro-projection and transformation resume to

$$\mathcal{L}_{\mathbb{AMP}} = \begin{bmatrix} \mathsf{T} \\ \mu^*(\mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}}_1) \\ \rho^{\mathcal{C}} \\ \mu^*(\mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}}_2) \\ \rho^{\mathcal{C}} \end{bmatrix}, \tag{49}$$

where $\mu^*()$ is the trigonometric transform given in (19), and $\rho^{\mathcal{C}}$ must be provided as prior.

F. Final comment - points and lines

We summarize in Table I all points and lines parametrizations with their main manipulation expressions. On completion of their descriptions we have seen a number of parallelisms that should help building a coherent picture of the parametrizations suited for undelayed initialization in monocular EKF-SLAM. We have summarized these relations in Fig. 6. We have seen anchored and un-anchored representations. We have seen the surprising similarities between homogeneous

Table I
SUMMARY OF LANDMARK PARAMETRIZATIONS AND THEIR MAIN MANIPULATIONS

Lmk	**parameters**	**size**	**transformation + projection** $\underline{h}()$	**back-projection + transform** $g()$
EP	$\mathcal{P} = \mathbf{p}$	3	$\underline{\mathbf{u}} = \mathbf{K}\mathsf{R}^\top(\mathbf{p} - \mathsf{T})$	$\mathbf{p} = t\mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}} + \mathsf{T}$
HP	$\mathcal{P} = (\mathbf{m}, \rho)$	4	$\underline{\mathbf{u}} = \mathbf{K}\mathsf{R}^\top(\mathbf{m} - \mathsf{T}\rho)$	$\mathcal{P}_{\mathbb{H}} = \underline{\mathbf{p}} = \mathbf{H}\begin{pmatrix}\mathbf{K}^{-1}\underline{\mathbf{u}} \\ \rho^{\mathcal{C}}\end{pmatrix}$
AHP	$\mathcal{P} = (\mathbf{p}_0, \mathbf{m}, \rho)$	7	$\underline{\mathbf{u}} = \mathbf{K}\mathsf{R}^\top(\mathbf{m} - (\mathsf{T} - \mathbf{p}_0)\rho)$	$\mathcal{P}_{\mathbb{AH}} = \begin{pmatrix}\mathsf{T} \\ \mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}} \\ \rho^{\mathcal{C}}\end{pmatrix}$
AMPP	$\mathcal{P} = (\mathbf{p}_0, \varepsilon, \alpha, \rho)$	6	$\underline{\mathbf{u}} = \mathbf{K}\mathsf{R}^\top(\mathbf{m}^* - (\mathsf{T} - \mathbf{p}_0)\rho)$	$\mathcal{P}_{\mathbb{AMP}} = \begin{pmatrix}\mathsf{T} \\ \mu^*(\mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}}) \\ \rho^{\mathcal{C}}\end{pmatrix}$
PL	$\mathcal{L} = (\mathbf{n}, \mathbf{v})$	6	$\mathbf{l} = \mathcal{K}\mathsf{R}^\top(\mathbf{n} - \mathsf{T}\times\mathbf{v})$	$\mathcal{L}_{\mathbb{P}} = \mathcal{H}\begin{pmatrix}\mathcal{K}^{-1}\mathbf{l} \\ \mathbf{E}\beta\end{pmatrix}$
APL	$\mathcal{L} = (\mathbf{p}_0, \mathbf{n}, \mathbf{v})$	9	$\mathbf{l} = \mathcal{K}\mathsf{R}^\top(\mathbf{n} - (\mathsf{T} - \mathbf{p}_0)\times\mathbf{v})$	$\mathcal{L}_{\mathbb{A}P} = \begin{pmatrix}\mathsf{T} \\ \mathsf{R}\mathcal{K}^{-1}\mathbf{l} \\ \mathsf{R}\mathbf{E}\beta\end{pmatrix}$
HPL	$\mathcal{L} = (\mathbf{m}_1, \rho_1, \mathbf{m}_2, \rho_2)$	8	$\mathbf{l} = \mathcal{K}\mathsf{R}^\top\big((\mathbf{m}_1\times\mathbf{m}_2) - \mathsf{T}\times(\mathbf{m}_2\rho_1 - \mathbf{m}_1\rho_2)\big)$	$\mathcal{L}_{\mathbb{HP}} = \begin{pmatrix}\mathbf{H}\begin{pmatrix}\mathbf{K}^{-1}\underline{\mathbf{u}}_1 \\ \rho^{\mathcal{C}}\end{pmatrix} \\ \mathbf{H}\begin{pmatrix}\mathbf{K}^{-1}\underline{\mathbf{u}}_2 \\ \rho^{\mathcal{C}}\end{pmatrix}\end{pmatrix}$
AHPL	$\mathcal{L} = (\mathbf{p}_0, \mathbf{m}_1, \rho_1, \mathbf{m}_2, \rho_2)$	11	$\mathbf{l} = \mathcal{K}\mathsf{R}^\top\big((\mathbf{m}_1\times\mathbf{m}_2) - (\mathsf{T} - \mathbf{p}_0)\times(\mathbf{m}_2\rho_1 - \mathbf{m}_1\rho_2)\big)$	$\mathcal{L}_{\mathbb{AHP}} = \begin{pmatrix}\mathsf{T} \\ \mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}}_1 \\ \rho^{\mathcal{C}} \\ \mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}}_2 \\ \rho^{\mathcal{C}}\end{pmatrix}$
AMPPL	$\mathcal{L} = (\mathbf{p}_0, \varepsilon_1, \alpha_1, \rho_1, \varepsilon_2, \alpha_2, \rho_2)$	9	$\mathbf{l} = \mathcal{K}\mathsf{R}^\top\big((\mathbf{m}_1^*\times\mathbf{m}_2^*) - (\mathsf{T} - \mathbf{p}_0)\times(\mathbf{m}_2^*\rho_1 - \mathbf{m}_1^*\rho_2)\big)$	$\mathcal{L}_{\mathbb{AMPP}} = \begin{pmatrix}\mathsf{T} \\ \mu^*(\mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}}_1) \\ \rho^{\mathcal{C}} \\ \mu^*(\mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}}_2) \\ \rho^{\mathcal{C}}\end{pmatrix}$

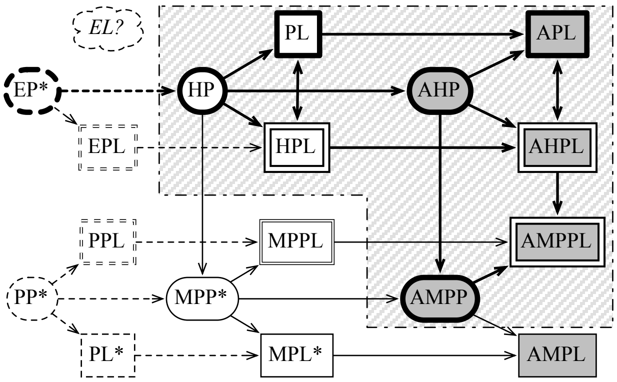


Figure 6. Links between all proposed parametrizations and more. Round boxes are points; square boxes are lines. Single-stroke square boxes are Plücker-based lines. Double-stroke square boxes are point-supported lines. Gray boxes are anchored parametrizations. Arrows indicate the links that we established within the discourse. The dashed area encloses all parametrizations benchmarked in this paper. Some other possible parametrizations, in thin line, have not been studied here (there are some repeated acronyms): polar point (PP, *i.e.*, $[\varepsilon, \alpha, d]$), modified-polar point (MPP, $[\varepsilon, \alpha, \rho]$); the point-supported Euclidean-points line (EPL, $[x_1, y_1, z_1, x_2, y_2, z_2]$), polar-points line (PPL, $[\varepsilon_1, \alpha_1, d_1, \varepsilon_2, \alpha_2, d_2]$), and modified-polar-points line (MPPL); and the directly-coded polar line (PL, $[\phi, \varepsilon, \alpha, d]$), modified-polar line (MPL, $[\phi, \varepsilon, \alpha, \rho]$), and anchored modified-polar line (AMPL). There is no such thing as a directly-coded Euclidean line (EL). Dashed-stroke elements do not benefit from the inverse-distance property and are not suited for undelayed initialization. Minimal parametrizations are marked with an asterisk.

points and Plücker lines. We have highlighted the parallelisms between point-supported and Plücker-based lines. We have finally situated the modified-polar parametrizations as lightened versions of homogeneous entities. The figure shows further parametrizations that fall out of our interest – refer to the figure's caption and to the discussion in Section VII for further justification.

IV. LANDMARK INITIALIZATION AND UPDATES

Undelayed landmark initialization in EKF-SLAM with partial measurements (such as monocular measurements) mimics the algorithm for full measurements and incorporates the non-measured magnitudes as Gaussian priors. We first detail the way we express physical measurements on the image plane, and the way to define the non-measured priors. We finally proceed with details on the initialization and updating procedures related to the EKF-machinery. For the initialization and updates of parameters not in the Kalman filter, such as lines endpoints, please refer to Section IV-E.

A. 2D measurements in the image plane

The previous discourse assumed homogeneous parametrizations of points and lines in the image plane $\mathbb{P}^2$. We detail here how to obtain them from real point and line measurements defined in the Euclidean pixels image.

- A point is measured as two Cartesian coordinates, and modeled as a Gaussian variable. Please note that the measurement corresponds to the mean value $\bar{\mathbf{u}}$ of the distribution:

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} \sim \mathcal{N}\{\bar{\mathbf{u}}, \mathbf{U}\}. \tag{50}$$

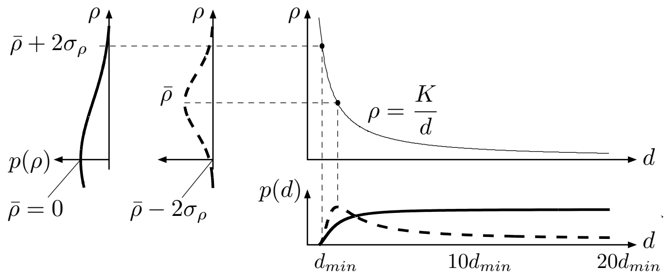


Figure 7. Inverse-distance *pdf*. A Gaussian $p(\rho) = \mathcal{N}(\rho - \bar\rho, \sigma_\rho^2)$ is defined in inverse-distance (vertical axes). We have ample choice: in one extreme (dashed) we may define it so that $\bar\rho - 2\sigma_\rho = 0$; the other extreme (solid) takes $\bar\rho = 0$. In all cases, we have $(\bar\rho + 2\sigma_\rho) = K/d_{min}$. They result in *pdfs* in distance (bottom) that cover from a minimal distance d_{min} to infinity. K is just a proportionality constant, *e.g.* $K = 1$ for AMPP, and $K = \|\mathbf{K}^{-1}\underline{\mathbf{u}}\|$ for AHP and HP. We can also normalize $\mathbf{K}^{-1}\underline{\mathbf{u}}$ at initialization time and take $K = 1$, in which case ρ is exactly equal to inverse-distance.

Its homogeneous counterpart is built with

$$\underline{\mathbf{u}} = \begin{bmatrix}\mathbf{u}\\1\end{bmatrix} \sim \mathcal{N}\left\{\begin{bmatrix}\bar{\mathbf{u}}\\1\end{bmatrix}, \begin{bmatrix}\mathbf{U} & 0\\ \mathbf{0} & 0\end{bmatrix}\right\}. \tag{51}$$

- A bounded segment is measured as a 4-vector stacking its two endpoints:

$$\mathbf{s} = \begin{bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\end{bmatrix} \sim \mathcal{N}\{\bar{\mathbf{s}}, \mathbf{S}\} = \mathcal{N}\left\{\begin{bmatrix}\bar{\mathbf{u}}_1\\ \bar{\mathbf{u}}_2\end{bmatrix}, \begin{bmatrix}\mathbf{U} & \mathbf{0}\\ \mathbf{0} & \mathbf{U}\end{bmatrix}\right\} \tag{52}$$

The segments homogeneous endpoints $\underline{\mathbf{u}}_i$, used for initialization of point-supported lines, are built like the regular points (Eq. (51)). The homogeneous line, used for initialization of Plücker lines, is built with

$$\mathbf{l} = \underline{\mathbf{u}}_1 \times \underline{\mathbf{u}}_2 \sim \mathcal{N}\{\bar{\mathbf{l}}, \mathbf{L}\} \tag{53}$$

with

$$\bar{\mathbf{l}} = \bar{\underline{\mathbf{u}}}_1 \times \bar{\underline{\mathbf{u}}}_2 \tag{54}$$

$$\mathbf{L} = [\bar{\underline{\mathbf{u}}}_1]_\times \, \mathbf{U} \, [\bar{\underline{\mathbf{u}}}_1]_\times^\top + [\bar{\underline{\mathbf{u}}}_2]_\times \, \mathbf{U} \, [\bar{\underline{\mathbf{u}}}_2]_\times^\top . \tag{55}$$

B. Defining the non-measured Gaussian priors

The key advantage of the inverse-distance parameters is that they allow the infinity range to be included in a Gaussian *pdf*, and at the same time allowing the observation functions to be linear with respect to them.

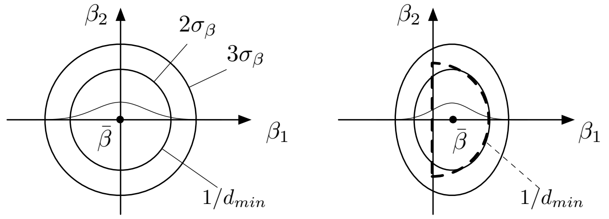
Two basic rules apply to the definition of the prior, be it $\rho^{\mathcal{C}}$ for points or $\beta^{\mathcal{C}}$ for Plücker lines: the origin must be well inside the 2σ support of the *pdf*, and the minimum considered distance d_{min} must (approximately) match the upper 2σ bound. For points and point-supported lines, this resumes to (see Fig. 7)

$$\bar\rho^{\mathcal{C}} - n\sigma_\rho = 0, \qquad 0 \le n < 2 \tag{56}$$

$$\bar\rho^{\mathcal{C}} + 2\sigma_\rho = 1/d_{min}. \tag{57}$$

A good practice is to choose $n = 1$, although this choice is not critical as it will be revealed by the benchmarking.

Defining the prior for Plücker lines is a bit trickier, as it is difficult to express the conditions as straightforward equations like (56) and (57). We prefer to refer the reader directly to Figs. 8 and 9. We use the solution $\beta \sim \mathcal{N}\{\bar\beta; \mathbf{B}\}$ in Fig. 8(b),



(a) Isotropic Gaussian *pdf* with line's mean at infinity.

(b) Non-isotropic *pdf* penalizing lines at negative depths.

Figure 8. Defining a *pdf* for $\beta \sim \mathcal{N}\{\bar\beta; \mathbf{B}\}$. (a) The isotropic Gaussian with $\bar\beta = (0,0)$ and $\mathbf{B} = \sigma_\beta^2\mathbf{I}$ contains all possible lines at a minimum distance of d_{min}: it has central symmetry, 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. (b) An interesting alternative that penalizes lines at the back of the camera is to approximate just the right-hand half of the *pdf* in (a) (here dashed) by a new Gaussian. A good fit is obtained with $\bar\beta = (1/3d_{min}, 0)$ and an anisotropic covariance $\mathbf{B} = \mathrm{diag}(\sigma_{\beta_1}^2, \sigma_{\beta_2}^2)$ with $\sigma_{\beta_1} = 1/3d_{min}$ and $\sigma_{\beta_2} = 1/2d_{min}$. See Fig. 9 for the result in inverse-β space.

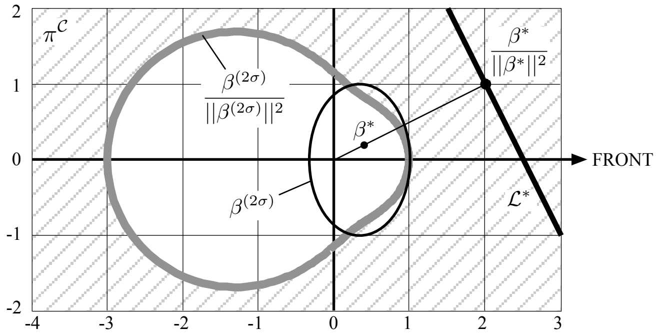


Figure 9. β and inverse-β planes. The inverse-distance application $\beta \to D = \beta/\|\beta\|^2$ maps the interior of the 2σ ellipse of β defined in Fig. 8(b) for $d_{min} = 1$ (black ellipse, noted here $\beta^{(2\sigma)}$) to the exterior of the gray shape (dashed area, extending to infinity). It is a funny curiosity that the figure resembles the section of an eye looking in the correct direction. The minimal distance in FRONT of the camera is well around 1, and lines on the back are penalized with feebler probability. As an example, a value $\beta^* = (0.4, 0.2)$ results in $D^* = \beta^*/\|\beta^*\|^2 = (2, 1)$, which represents the line $\mathcal{L}^*$.

$$\bar\beta = \begin{bmatrix}1/3d_{min}\\ 0\end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix}(1/3d_{min})^2 & 0\\ 0 & (1/2d_{min})^2\end{bmatrix} \tag{58}$$

C. Landmark initialization

Initialization with partial measurements mimics the regular EKF-SLAM initialization algorithm with the addition of the non-measured DOF as the Gaussian priors just defined. Fig. 10 serves as a reference for all initialization algorithm variants.

1) Identify the mapped magnitudes $\mathbf{x} \sim \mathcal{N}\{\bar{\mathbf{x}}, \mathbf{P}\}$, where

$$\mathbf{x} = \begin{bmatrix}\mathcal{C}\\ \mathcal{M}\end{bmatrix}, \quad \bar{\mathbf{x}} = \begin{bmatrix}\bar{\mathcal{C}}\\ \bar{\mathcal{M}}\end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix}\mathbf{P}_{\mathcal{CC}} & \mathbf{P}_{\mathcal{CM}}\\ \mathbf{P}_{\mathcal{MC}} & \mathbf{P}_{\mathcal{MM}}\end{bmatrix},$$

   with $\mathcal{C} = (\mathsf{T}, \mathsf{Q})$ the camera frame and $\mathcal{M} = (\mathcal{L}_1, \ldots, \mathcal{L}_N)$ the set of mapped landmarks.

2) Identify the measurement,
   - $\mathbf{u} \sim \mathcal{N}\{\bar{\mathbf{u}}, \mathbf{U}\}$ for points,

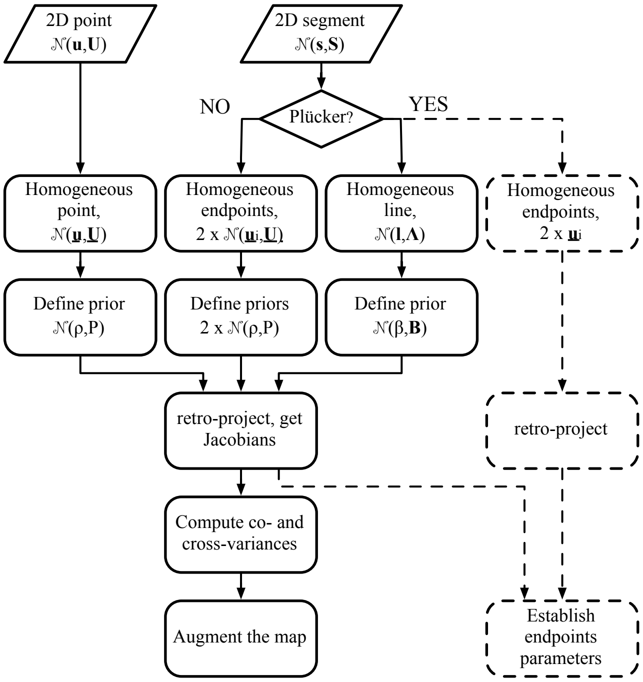


Figure 10. General initialization algorithm, valid for points (leftmost thread), point-supported lines (center left), and Plücker-based lines (center right). The right dashed thread is for segments endpoints.

- $\mathbf{s} \sim \mathcal{N}\{\bar{\mathbf{s}}, \mathbf{S}\}$ for line segments.

and convert it to the homogeneous space (Section IV-A).

3) Define a Gaussian prior for the non-measured DOFs (Section IV-B),
   - inverse-distance for points and for point-supported lines, $\rho^{\mathcal{C}} \sim \mathcal{N}\{\bar{\rho}^{\mathcal{C}}, \sigma^2_{\rho^{\mathcal{C}}}\}$, see Fig. 7,
   - Plücker parameters for Plücker-based lines, $\beta \sim \mathcal{N}\{\bar{\beta}; \mathbf{B}\}$, see Fig. 8.
4) Back-project the Gaussian measurement; get landmark mean and Jacobians

$$\begin{aligned}\bar{\mathcal{P}} &= g(\bar{\mathcal{C}}, \bar{\mathbf{u}}, \bar{\rho}^{\mathcal{C}})\\ \mathbf{G}_{\mathcal{C}} &= \left.\frac{dg}{d\mathcal{C}}\right|_{\bar{\mathcal{C}},\bar{\mathbf{u}},\bar{\rho}^{\mathcal{C}}}, \mathbf{G}_{\mathbf{u}} = \left.\frac{dg}{d\mathbf{u}}\right|_{\bar{\mathcal{C}},\bar{\mathbf{u}},\bar{\rho}^{\mathcal{C}}}, \mathbf{G}_{\rho} = \left.\frac{dg}{d\rho}\right|_{\bar{\mathcal{C}},\bar{\mathbf{u}},\bar{\rho}^{\mathcal{C}}}\end{aligned}$$

with $\mathcal{P} = g(\mathcal{C}, \mathbf{u}, \rho^{\mathcal{C}})$ (conversely $\mathcal{L} = g(\mathcal{C}, \mathbf{s}, \beta)$ or $\mathcal{L} = g(\mathcal{C}, \mathbf{s}, \rho^{\mathcal{C}})$) one of the back-projection functions in Table I, with $\mathcal{C} = (\mathsf{T}, \mathsf{Q})$, $\mathsf{R} = \mathsf{R}(\mathsf{Q})$, $\underline{\mathbf{u}} = [\mathbf{u}^\top \; 1]^\top$, $\mathbf{s} = [\mathbf{u}_1^\top \; \mathbf{u}_2^\top]^\top$, and $\mathbf{l} = \underline{\mathbf{u}}_1 \times \underline{\mathbf{u}}_2$.

5) Compute landmark co- and cross-variances

$$\begin{aligned}\mathbf{P}_{\mathcal{P}\mathcal{P}} &= \mathbf{G}_{\mathcal{C}}\mathbf{P}_{\mathcal{C}\mathcal{C}}\mathbf{G}_{\mathcal{C}}^\top + \mathbf{G}_{\mathbf{u}}\mathbf{S}\mathbf{G}_{\mathbf{u}}^\top + \mathbf{G}_{\rho}\sigma^2_{\rho^{\mathcal{C}}}\mathbf{G}_{\rho}^\top\\ \mathbf{P}_{\mathcal{P}\mathbf{x}} &= \mathbf{G}_{\mathcal{C}}\mathbf{P}_{\mathcal{C}\mathbf{x}}\end{aligned}$$

with $\mathbf{P}_{\mathcal{C}\mathbf{x}} = [\mathbf{P}_{\mathcal{C}\mathcal{C}} \; \mathbf{P}_{\mathcal{C}\mathcal{M}}]$.

6) Augment the SLAM map

$$\bar{\mathbf{x}} \leftarrow \begin{bmatrix}\bar{\mathbf{x}}\\ \bar{\mathcal{P}}\end{bmatrix}, \qquad \mathbf{P} \leftarrow \begin{bmatrix}\mathbf{P} & \mathbf{P}_{\mathcal{P}\mathbf{x}}^\top\\ \mathbf{P}_{\mathcal{P}\mathbf{x}} & \mathbf{P}_{\mathcal{P}\mathcal{P}}\end{bmatrix}.$$

D. Landmark updates

1) Point updates: Point updates follow the standard EKF-SLAM formulation. The observation functions $\mathbf{u} = h(\mathcal{C}, \mathcal{P})$ are the composition of the homogeneous implementations $\underline{h}()$ in Table I with the homogeneous-to-Euclidean transform $\underline{\mathbf{u}} = [u, v, w]^\top \rightarrow \mathbf{u} = h2e(\underline{\mathbf{u}}) = [u/w, v/w]^\top$, with covariance $\mathbf{U}$. In other words, we have $h(\bullet) \triangleq h2e(\underline{h}(\bullet))$. We apply regular EKF:

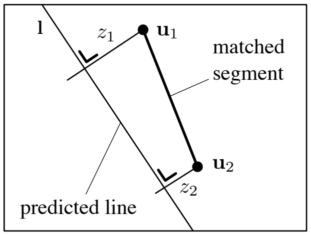


Figure 11. Plücker line observation update. Direct measurement of the innovation $\mathbf{z} = (z_1, z_2)$ as the two signed orthogonal distances from the detected endpoints to the expected (or predicted) line.

$$\begin{aligned}\mathbf{z} &= \mathbf{u} - h(\bar{\mathbf{x}}) \qquad \big(= \mathbf{u} - h(\bar{\mathcal{C}}, \bar{\mathcal{P}})\big) && (59)\\ \mathbf{Z} &= \mathbf{H}{\cdot}\mathbf{P}{\cdot}\mathbf{H}^\top + \mathbf{U} && (60)\\ \mathbf{K} &= \mathbf{P}{\cdot}\mathbf{H}^\top{\cdot}\mathbf{Z}^{-1} && (61)\\ \bar{\mathbf{x}} &\leftarrow \bar{\mathbf{x}} + \mathbf{K}{\cdot}\mathbf{z} && (62)\\ \mathbf{P} &\leftarrow \mathbf{P} - \mathbf{K}{\cdot}\mathbf{H}{\cdot}\mathbf{P} \quad (= \mathbf{P} - \mathbf{K}{\cdot}\mathbf{Z}{\cdot}\mathbf{K}^\top), && (63)\end{aligned}$$

with $\mathbf{H} = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\bar{\mathbf{x}}}$.

2) Line updates: It is convenient to represent the matched segment by its two endpoints, $\mathbf{s} = (\mathbf{u}_1, \mathbf{u}_2) \in \mathbb{R}^4$, with covariance $\mathbf{S} = \mathrm{diag}(\mathbf{U}, \mathbf{U})$. 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 $\mathbf{u}_i$ to the expected line $\bar{\mathbf{l}} = \underline{h}(\bar{\mathcal{C}}, \bar{\mathbf{x}})$ (Fig. 11). The signed distance z from a point $\underline{\mathbf{u}} = (u, v, 1)$ to a line $\mathbf{l} = (l_1, l_2, l_3)$ is given by

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

so the innovation vector is

$$\mathbf{z} = \begin{bmatrix}z_1\\ z_2\end{bmatrix} = \begin{bmatrix}\bar{\mathbf{l}}^\top{\cdot}\underline{\mathbf{u}}_1/\sqrt{\bar{l}_1^2 + \bar{l}_2^2}\\ \bar{\mathbf{l}}^\top{\cdot}\underline{\mathbf{u}}_2/\sqrt{\bar{l}_1^2 + \bar{l}_2^2}\end{bmatrix} \in \mathbb{R}^2. \qquad (64)$$

The above expression reveals the fact that the innovation can no longer be obtained from the subtraction $\mathbf{z} = \mathbf{s} - h(\bar{\mathbf{x}})$ we are familiar with in EKF, like in (59), but from a non-linear function $\mathbf{z} = j(\bar{\mathbf{x}}, \mathbf{s})$. This function is obtained by composing line transformation and projection (in Table I) and the innovation measurements (64). The result is a somewhat complicated expression with a generic form

$$\mathbf{z} = j(\bar{\mathbf{x}}, \mathbf{s}) = j(\bar{\mathcal{C}}, \bar{\mathcal{L}}, \mathbf{s}),$$

where $\bar{\mathcal{C}}$ and $\bar{\mathcal{L}}$ are the camera and line estimates from the map $\bar{\mathbf{x}}$, and $\mathbf{s}$ is the measurement vector. One detail that is worth noticing is the sign change in the Jacobians with respect to the habitual, explicit EKF innovation definition $j(\bar{\mathbf{x}}, \mathbf{s}) = \mathbf{s} - h(\bar{\mathbf{x}})$. It is clear that $\frac{\partial j}{\partial \mathbf{x}} = -\frac{\partial h}{\partial \mathbf{x}}$. With this sign change and the non-trivial contribution of $\mathbf{s}$ in the innovation, the EKF correction

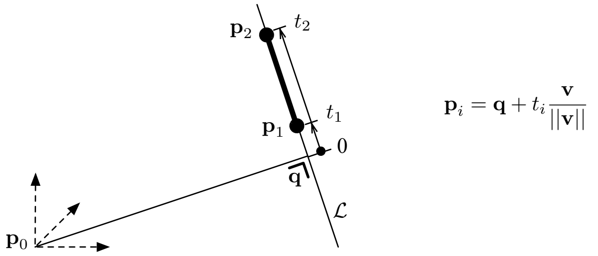


Figure 12. Segment endpoints in the local ordinate frame of Plücker-based lines PL and APL. The closest point $\mathbf{q}$ to the anchor is the local origin, and the unit length and positive orientation are defined by a normalized director vector $\mathbf{u} = \mathbf{v}/\|\mathbf{v}\|$. The endpoint $\mathbf{p}_i$ is coded by the scalar abscissa t_i, satisfying $\mathbf{p}_i = \mathbf{q} + t_i\mathbf{v}/\|\mathbf{v}\|$. Only the abscissas t_i are stored.

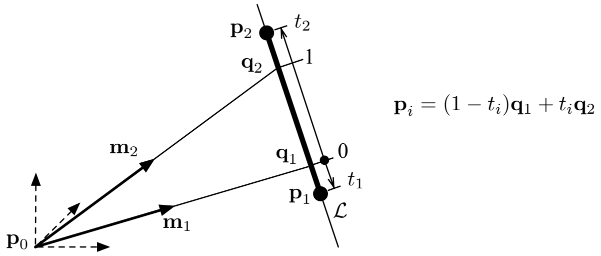


Figure 13. Segment endpoints in the local ordinate frame of point-defined lines HL, AHL and AMPL. The line is supported by two points $\mathbf{q}_i = \mathbf{p}_0 + \mathbf{m}_i/\rho_i$. The local frame has its origin at $\mathbf{q}_1$ and the unit length and positive orientation are defined by $\mathbf{q}_2 - \mathbf{q}_1$. The endpoint $\mathbf{p}_i$ is coded by the scalar abscissa t_i, satisfying $\mathbf{p}_i = (1 - t_i)\mathbf{q}_1 + t_i\mathbf{q}_2$. Only the abscissas t_i are stored.

equations have to be modified accordingly,

$$\begin{aligned}
\mathbf{Z} &= \mathbf{J_x}\cdot\mathbf{P}\cdot\mathbf{J_x}^\top + \mathbf{J_s}\cdot\mathbf{S}\cdot\mathbf{J_s}^\top && (65)\\
\mathbf{K} &= -\mathbf{P}\cdot\mathbf{J_x}^\top\cdot\mathbf{Z}^{-1} && (66)\\
\bar{\mathbf{x}} &\leftarrow \bar{\mathbf{x}} + \mathbf{K}\cdot\mathbf{z} && (67)\\
\mathbf{P} &\leftarrow \mathbf{P} + \mathbf{K}\cdot\mathbf{J_x}\cdot\mathbf{P} \quad (= \mathbf{P} - \mathbf{K}\cdot\mathbf{Z}\cdot\mathbf{K}^\top), && (68)
\end{aligned}$$

with $\mathbf{J_x} = \frac{\partial \mathbf{z}}{\partial \mathbf{x}}$ and $\mathbf{J_s} = \frac{\partial \mathbf{z}}{\partial \mathbf{s}}$. An equivalent solution that permits reusing existing EKF code is to define $\mathbf{H} = -\mathbf{J_x}$ and $\mathbf{R} = \mathbf{J_s}\mathbf{S}\mathbf{J_s}^\top$, and apply the regular EKF equations (59–63) with the Jacobian $\mathbf{H}$ and the new measurement noise $\mathbf{R}$.[3]

We do not apply any kind of correction to enforce the Plücker constraint. We ensured it during initialization and its validity at any later time is only approximately guaranteed through cross-correlations. Refer to Section VII for further discussion.

E. Segment endpoints

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. In Plücker-based lines (Fig. 12) the local frame is defined by a single axis with the origin at the point $\mathbf{q} = \mathbf{p}_0 + (\mathbf{v}\times\mathbf{n})/\|\mathbf{v}\|^2$, the closest to the anchor, and the director vector $\mathbf{u} = \mathbf{v}/\|\mathbf{v}\|$ providing the unit length. Each endpoint $\mathbf{p}_i$ is specified by an abscissa t_i such that

$$\mathbf{p}_i = \mathbf{q} + t_i\mathbf{v}/\|\mathbf{v}\|. \tag{69}$$

In point-supported lines (Fig. 13) the endpoints are defined with respect to the support points $\mathbf{q}_i$,

$$\mathbf{q}_i = \mathbf{p}_0 + \mathbf{m}_i/\rho_i. \tag{70}$$

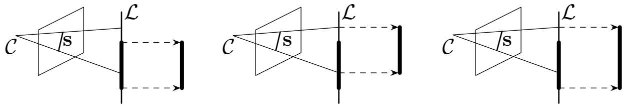
The origin is at $\mathbf{q}_1$ and the unit length defined by $\mathbf{q}_2 - \mathbf{q}_1$, leading to the endpoints expression

$$\mathbf{p}_i = (1 - t_i)\mathbf{q}_1 + t_i\mathbf{q}_2, \tag{71}$$

so that the initial abscissas are

$$(t_1, t_2) = (0, 1). \tag{72}$$

The endpoints are updated according to a different rule 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. This is especially dramatic in Plücker-based lines because the local origin $\mathbf{q}$ depends non linearly on the line's orientation (refer to Section VII for further discussion). For convenience, we considered the line converged at the moment when it is ready for re-parametrization (see Section IV-F). Before this happens the abscissas are either not updated (Fig. 14(a)), thus reflecting the initial observation, or systematically updated (Fig. 14(b)), simply reflecting the last observation. Once the line has converged an extending-only policy is applied (Fig. 14(c)): the abscissa is updated only if this lengthens the 3D segment. The algebra expressing the current endpoints projection onto the line is of no relevant interest for the purposes of this work, and is not included here for space reasons. As a reference, an example valid for PL can be found in [13].



(a) Keep initial (b) Keep current (c) Lengthen segment

Figure 14. The three general strategies for updating 3D segment endpoints. (a) Endpoints are defined at initialization time and never updated. (b) Endpoints are systematically updated according to the current observation. (c) Endpoints are updated only if this lengthens the 3D segment.

F. Landmark re-parametrization

The necessary over-parametrization need only be maintained during the period of convergence, while the landmark keeps some of its DOF at high uncertainty levels. After this period it is convenient for computational reasons to switch to cheaper representations. One can choose any minimal or quasi-minimal representation, in a trade-off between economy of resources and representational power. For points, the natural choice is the minimal Euclidean parametrization EP [9]. For lines, and because of the need of endpoints, we feel convenient to choose a non-minimal two-points representation $\mathcal{L} = (\mathbf{p}_1, \mathbf{p}_2)$ with 6 parameters (the Euclidean-points line EPL, see Fig. 6 – the minimal representations for 3D lines are of size 4). The reparametrization is performed when the

---

[3] It is handy to realize that $\mathbf{J_s}$ is such that, if the pixel noise is isotropic, $\mathbf{U} = \sigma^2\mathbf{I}_2$, we have $\mathbf{S} = \mathrm{diag}(\mathbf{U}, \mathbf{U}) = \sigma^2\mathbf{I}_4$, and then $\mathbf{R} = \mathbf{J_s}\mathbf{S}\mathbf{J_s}^\top = \sigma^2\mathbf{I}_2 = \mathbf{U}$.

linearity test of the destination representation is passed [9]. See Section V-A for details on the linearity measures and tests.

Landmark re-parametrization requires a passage function $\mathcal{L}' = k(\mathcal{L})$ and its Jacobian $\partial\mathcal{L}'/\partial\mathcal{L}|_{\bar{\mathcal{L}}}$. It is performed in the standard Gaussian propagation fashion just taking care of deleting or de-activating the unused states after re-parametrization. Some of these $k()$ functions can be found in this paper, *e.g.* (3) or (11) for points. For lines, and we refer now to Section IV-E and Figs. 12–13, we have two options. One option is to take the current endpoints $\mathbf{p}_i$ (*i.e.*, $\mathcal{L}_{\mathbb{EPL}} = (\mathbf{p}_1, \mathbf{p}_2)$ using (69) or (71)) and reset the abscissas pair to the value $(0, 1)$. The other option is to recover the previously existing support points $\mathbf{q}_i$ (*i.e.*, $\mathcal{L}_{\mathbb{EPL}} = (\mathbf{q}_1, \mathbf{q}_2)$ using (70)) and keep the abscissas at their value (t_1, t_2).

V. LINEARITY AND PERFORMANCE ANALYSIS

A. *Analytical measure of linearity*

The EKF requirements for high degrees of linearity in the measurement and dynamic model equations makes the use of an analytic linearity measurement interesting. In the already referenced [9], an analytic linearity index was proposed to emphasize the improvement in linearity when the unmeasured range is parametrized as inverse-distance instead of distance. This linearity measure is based on the variation in the first derivative of the function inside the 95% probability interval. The measure is restricted, thanks to the particular symmetries of the problem, to just 1DOF. Very related to this work, the trace of the Hessian of the measurement model is proposed in [29] as a measurement of the degree of linearity in several nodes of a multi-map SLAM. This second measure has the drawback of not incorporating the dimensions of the uncertainty region.

We introduce an analytical linearity index for multiple-input/multiple-output (MIMO) functions which accounts for the probability region of the input variable. As a desirable quality, the proposed measurement allows us to compare parametrizations having different sizes, and even compare the degree of linearity of points against lines. As in [29], the index involves the computation of the Hessian; this is composed with the covariances matrix, incorporating this way all the information on the uncertainty region.

Consider the transformation plus pinhole projection functions in Table I: $\underline{\mathbf{u}} = \underline{h}(\mathcal{C}, \mathcal{P})$ for points and $\mathbf{l} = \underline{h}(\mathcal{C}, \mathcal{L})$ for lines. Compose them with the appropriate innovation functions: $\mathbf{z} = \mathbf{y} - h(\mathcal{C}, \mathcal{P})$ for points and (64) for lines, to obtain the generic innovation functions $\mathbf{z} = j(\mathcal{C}, \mathcal{L}, \mathbf{y})$. For concision, let us indistinctly denote landmark parameters by $\mathcal{L}$, and define the state $\mathbf{x} = (\mathcal{C}, \mathcal{L})$. Let us further consider the measurement $\mathbf{y}$ as a given magnitude acting as a parameter and drop it from the notation. In what is to follow, we write simply $\mathbf{z} = j(\mathbf{x})$ to mean $\mathbf{z} = j(\mathcal{C}, \mathcal{L}, \mathbf{y})$, with $\mathbf{z}$ the innovation and $\mathbf{x}$ the state. Let us finally note $m = \dim(\mathbf{z})$ and $n = \dim(\mathbf{x})$ —in our case we have $m = 2$, and $10 \leq n \leq 18$ depending on the selected parametrization.

We are interested in analyzing the linearity of $j(\mathbf{x})$ inside the region of high probability of $\mathbf{x}$ described by the Gaussian $\mathcal{N}\{\bar{\mathbf{x}}; \mathbf{P}\}$. For linear functions, the Jacobian matrix $\mathbf{J} = \frac{\partial j}{\partial \mathbf{x}}$ is constant for all $\mathbf{x}$. For non-linear functions, the Jacobian matrix depends on the particular evaluation point $\mathbf{x}^*$, *i.e.*,

$$\mathbf{J}(\mathbf{x}^*) = \left.\frac{\partial j}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} \in \mathbb{R}^{m\times n}. \tag{73}$$

Assuming our function $j()$ is reasonably linear inside the probability region, which is after all a requisite for the Extended Kalman Filter to operate properly, the Jacobian can be well approximated in the vicinity of the mean $\bar{\mathbf{x}}$ by its Taylor series truncated at the linear terms,

$$\mathbf{J}(\bar{\mathbf{x}} + \Delta\mathbf{x}) \approx \mathbf{J}(\bar{\mathbf{x}}) + \mathbf{H}(\bar{\mathbf{x}}) \cdot \Delta\mathbf{x} \ , \tag{74}$$

with $\Delta\mathbf{x} \triangleq \mathbf{x} - \bar{\mathbf{x}}$ and where

$$\mathbf{H}(\bar{\mathbf{x}}) \triangleq \left.\frac{\partial \mathbf{J}}{\partial \mathbf{x}}\right|_{\bar{\mathbf{x}}} = \left.\frac{\partial^2 j}{\partial \mathbf{x}^2}\right|_{\bar{\mathbf{x}}} \in \mathbb{R}^{m\times n\times n} \tag{75}$$

is the Hessian tensor of $j()$ evaluated at the mean $\bar{\mathbf{x}}$.

Let us refer to the rightmost term in (74) with the matrix $\Delta\mathbf{J}$,

$$\Delta\mathbf{J} \triangleq \mathbf{H}(\bar{\mathbf{x}}) \cdot \Delta\mathbf{x} \in \mathbb{R}^{m\times n} \ , \tag{76}$$

defined as (the linear approximation of) the error in the Jacobian due to the state error $\Delta\mathbf{x}$. It is computed as the product of the tensor $\mathbf{H}$, with components H_{ijk}, times the vector $\Delta\mathbf{x}$ with components Δx_k, giving the matrix $\Delta\mathbf{J}$ with components $\Delta J_{ij} = \sum_{k=1}^{n} H_{ijk}\Delta x_k$. The Einstein Summation Convention (ESC) [30] allows us to not write the summation signs each time a given index (i, j or k) appears twice, yielding a practical scalar expression for each component,

$$\Delta J_{ij} = H_{ijk}\Delta x_k. \tag{77}$$

The error $\Delta\mathbf{x}$ is obviously unknown but we dispose in the SLAM map of its covariances matrix $\mathbf{P} \triangleq \mathbb{E}[\Delta\mathbf{x} \cdot \Delta\mathbf{x}^\top]$, with $\mathbb{E}[\bullet]$ the expectation operator. We may incorporate this knowledge to our measure by constructing the matrix

$$\mathbf{Q} \triangleq \mathbb{E}[\Delta\mathbf{J} \cdot \Delta\mathbf{J}^\top] \in \mathbb{R}^{m\times m} \ , \tag{78}$$

which we can develop in terms of $\mathbf{H}$ and $\mathbf{P}$ using the ESC,

$$\begin{aligned} Q_{ij} &= \mathbb{E}[\Delta J_{ik}\Delta J_{jk}] \\ &= \mathbb{E}[H_{ikl}\Delta x_l H_{jkm}\Delta x_m] \\ &= H_{ikl}\mathbb{E}[\Delta x_l \Delta x_m] H_{jkm} \\ &= H_{ikl}P_{lm}H_{jkm} \ , \end{aligned} \tag{79}$$

leading to[4]

$$\mathbf{Q} = \sum_{k=1}^{n} \mathbf{H}_k \cdot \mathbf{P} \cdot \mathbf{H}_k^\top \ , \tag{80}$$

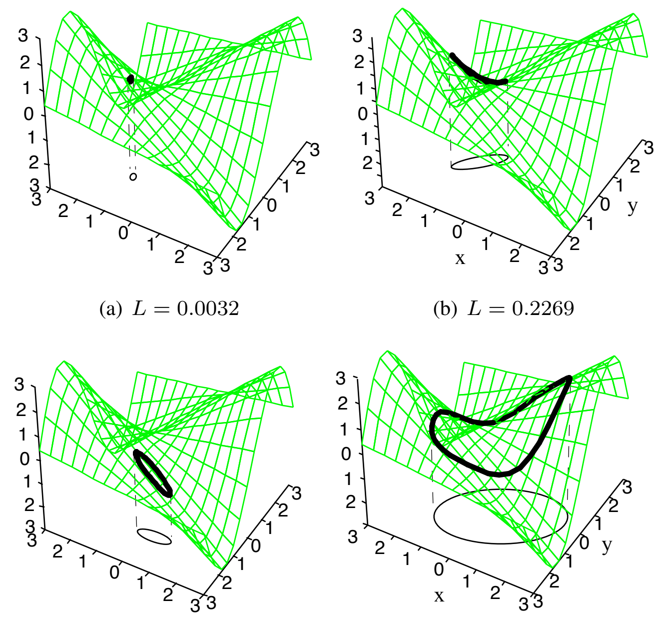
where the $n\times m$ matrices $\mathbf{H}_k \triangleq \frac{\partial \mathbf{J}}{\partial x_k} = \frac{\partial}{\partial x_k}\frac{\partial j}{\partial \mathbf{x}}$ are the n slices of the tensor $\mathbf{H}$. We may note this with $\mathbf{H}_k = [H_{ij}]_k$.

The matrix $\mathbf{Q}$ may be viewed as the error in linearity relative to (or normalized by) the shape and dimensions of the uncertainty region. It has the following properties:

1) It is symmetric and nonnegative.

---

[4] To obtain (80) we can also use Kronecker expansions of the tensors into planar matrices, write $\Delta\mathbf{J} = \mathbf{H} \cdot \Delta\mathbf{x} = [\mathbf{H}_1, \cdots, \mathbf{H}_k, \cdots, \mathbf{H}_n] \cdot (\mathbf{I}_n \otimes \mathbf{x})$, and inject it in (78), using the Kronecker product $\otimes$ properties to conclude on the same result.



(a) $L = 0.0032$ (b) $L = 0.2269$

(c) $L = 0.0014$ (d) $L = 1.5263$

Figure 15. Linearity index L of the MISO function $z = f(x, y) = x{\cdot}\sin(y)$ for different probability regions. We illustrate the projection (thick black) of the 2-sigma elliptical bound of the probability region (thin black) onto the surface $\mathcal{S} = \{(x, y, z) \ s.t. \ z = f(x, y)\}$ (mesh). The more elliptic the projected shape, the more linear is the function and smaller the index L. (a) A tiny probability region gives good linearity and a very small index. (b) A large probability region usually obliges the ellipse to bend over the surface, meaning high non-linearity and resulting in a large index. (c) If such a large ellipse falls on a planar region of the surface, the index drops to show good linearity. (d) An extreme case of very high non-linearity.

2) It has a fixed and compact size $m \times m$ independent of the state dimension n.
3) It is exactly zero for linear functions ($\mathbf{H} = 0$) or for particle distributions ($\mathbf{P} = 0$, immune to non-linearity).
4) It grows when either $\mathbf{H}$ or $\mathbf{P}$ grow.

For all these reasons, $\mathbf{Q}$ is very pertinent for evaluating the fitness of $j()$ in EKF. In order to obtain a scalar index, we define the ultimate linearity index L as a convenient norm of $\mathbf{Q}$, *i.e.*,

$$L = \|\mathbf{Q}\| \in \mathbb{R}^+. \tag{81}$$

We use the Frobenius norm $\|\mathbf{Q}\|_F \triangleq \sqrt{\sum_{ij} |q_{ij}|^2}$, which being $\mathbf{Q}$ symmetric with $m = 2$ yields the scalar linearity index

$$L = \sqrt{|q_{11}|^2 + 2|q_{12}|^2 + |q_{22}|^2}. \tag{82}$$

The index L is zero for functions showing a linear behavior inside the probability region, and positively increasing as the validity of this hypothesis vanishes. As an example, we illustrate in Fig. 15 the fitness of this index for a 2-input, 1-output function. Observe that the function is the same but the evaluation region changes position (the evaluation point $\bar{\mathbf{x}}$) and dimensions (the covariance $\mathbf{P}$), greatly affecting the linearity index.

B. Statistical consistency analysis

TODO: *Review literature on EKF-SLAM consistency and maybe some other.*

Until very recently many SLAM works have limited the illustration of their performance to providing some figures showing some snapshots of the final map. It is true that providing more consequent and informative evaluations is costly in robotics because of the difficulties of accessing reliable ground-truth information. This is where simulation is useful. But even with simulation, many works have failed to provide accurate evaluations, mainly because their conclusions about filter consistency were based on one single run. Because of the stochastic nature of the processes we are dealing with, it is usually easy to obtain a consistent realization of an inconsistent filter, and also to find an inconsistent realization of a consistent one. This is particularly critical in non-linear systems because of the potential that inconsistency has in such cases to lead to fatal divergence. As it was already indicated by Bar-Shalom [31, pp. 394] for the general filtering case and reminded to the SLAM community by Bailey [32], consistency evaluation should be performed statistically via Monte-Carlo analysis. This is what we provide here: a benchmarking of all the proposed parametrizations via a chi-square consistency test that evaluates the averaged normalized estimation error squared (NEES) over a number of Monte-Carlo runs. Here we follow strictly [32], which is in turn based on [31, pp. 234–235].

Although consistency is not the ultimate measure of performance [33], it is a necessary condition that all filters should meet, and it is intimately related to the degree of linearity-Gaussianity of the system. An interesting outcome of our evaluation is therefore the possibility of correlating our proposed linearity measure with the conclusions on filter consistency.

When ground truth about a variable $\mathbf{x}_k$ is known, the NEES of its estimate $\mathcal{N}\{\hat{\mathbf{x}}_k, \mathbf{P}_k\}$ at each time k can be defined by

$$\epsilon_k = (\mathbf{x}_k - \hat{\mathbf{x}}_k)^\top \mathbf{P}_k^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k). \tag{83}$$

Under the hypothesis of consistent filtering of a linear-Gaussian system, ϵ_k obeys a χ^2 (chi-square) distribution with $\dim(\mathbf{x}_k)$ DOF, noted $\chi^2_{\dim(\mathbf{x})}$, whose expectation over an increasing number of runs converges to the state dimension, $\mathbb{E}[\epsilon_k] = \dim(\mathbf{x}_k)$. Then, the linear-Gaussian hypothesis can be statistically evaluated by means of a χ^2 acceptance test over a set of $N < \infty$ Monte-Carlo runs.

Given N Monte-Carlo runs, $\sum_{i=1}^{N} \epsilon_{ik}$ obeys a $\chi^2_{N \dim(\mathbf{x})}$ distribution. The bounds of the *double-sided* 95% *probability concentration region* are given by the $\chi^2_{N \dim(\mathbf{x})}$ values corresponding to tail probabilities of 2.5% and 97.5%.

For practical reasons and because the full SLAM state vector is of varying size, we restrict our analysis to the state variables representing the robot (or sensor) pose, hoping that consistent localization leads to consistent mapping. For 6-DOF SLAM and 25 runs, we have $\dim(\mathbf{x}) = 6$ and $N = 25$, with which the lower and upper bounds are computed,

$$\underline{\nu} = \chi^2_{(25\times 6)}(1 - 0.025) = 117.985$$
$$\overline{\nu} = \chi^2_{(25\times 6)}(1 - 0.975) = 185.800.$$

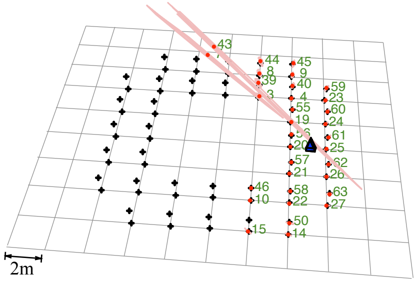


Figure 16. Simulated 3D environment for benchmarking 3 point parametrizations.

The 6-vector pose $\mathbf{x}_k$ and covariance $\mathbf{P}_k$ are computed from our 7-vector pose $\mathcal{C}_k \sim \mathcal{N}\{\overline{\mathcal{C}}_k, \mathbf{P}_{\mathcal{C}\mathcal{C}k}\}$ via a classical uncertainty propagation using the Jacobians of the passage function (the 3-DOF orientation can be chosen to be expressed with the Euler angles or with the rotation vector). The NEES value (83) is computed in this 6-space.

The average NEES is defined as

$$\bar{\epsilon}_k \triangleq \frac{1}{N} \sum_{i=1}^{N} \epsilon_{ik}. \tag{84}$$

We compare the average NEES against $\underline{\epsilon} = \underline{\nu}/N = 4.719$ and $\overline{\epsilon} = \overline{\nu}/N = 7.432$. If the average NEES is below the lower bound for some significant amount of time (more than 2.5% of the time), the filter is conservative. If it is above the upper bound (also by more that 2.5%), the filter is optimistic and therefore inconsistent.

VI. RESULTS

A. *Software and SLAM algorithm*

We have made available the software used for simulations [34]. It consists in a 6-DOF EKF-SLAM system written in MATLAB®, with simulation and 3D graphics capabilities.

The algorithm is organized as an active-search-based SLAM [35], which allows us to optimize information gain with a limited number of updates per frame. At each frame, we perform updates to the 10 most informative landmarks. We also attempt to initialize one landmark per frame. Unstable and inconsistent landmarks are deleted from the map to avoid map overpopulation and corruption.

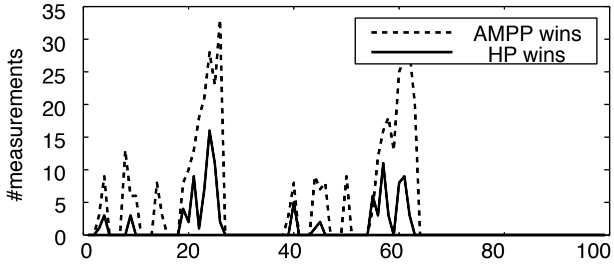
B. *Points*

We benchmark HP, AHP and AMPP using the same simulated scenario, the same software and the same seeds for the random generator. We start with a description of the simulation conditions, then proceed with the results of the (analytical) linearity and (statistical) error and consistency analyses.

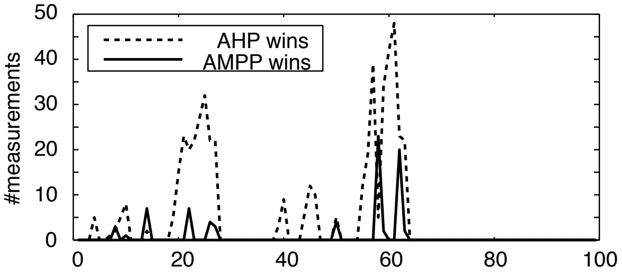
1) Simulated scenario: We simulate a robot performing a circular trajectory in an area of $12m \times 12m$, populated with 72 landmarks forming a cloister (Fig. 16). The robot receives noisy control inputs which are used for the prediction stage of the EKF, fixing the scale factor. One noisy image per control step is gathered with a single camera heading forward. Two sets of parameters have been used for the tests (the nominal and perturbation levels of all these magnitudes, together with the inverse-distance priors used, are all summarized in Table II). In the first set, the robot makes two turns to the cloister (800 frames are processed). The second set uses smaller odometry increments and perturbations, and the trajectory is limited to one quarter of a turn (200 frames).

Table II
SIMULATION PARAMETERS FOR THE POINTS SIMULATIONS

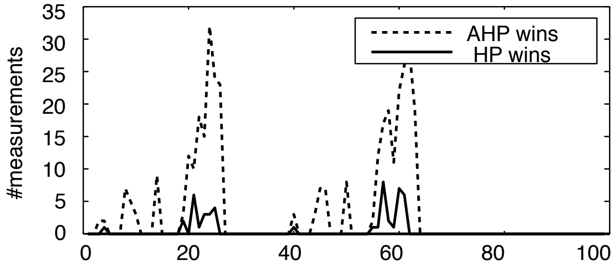
Concept	**Param.**	**Set 1**	**Set 2**
Pose step	$(\Delta X, \Delta\psi)$	$(8\text{cm}, 0.9°)$	$(4\text{cm}, 0.45°)$
Lin. noise	$(\sigma_X, \sigma_Y, \sigma_Z)$	1cm	0.5cm
Ang. noise	$(\sigma_\phi, \sigma_\theta, \sigma_\psi)$	0.1°	0.05°
Img. size		640×480 pix	
Focal	(α_u, α_v)	320 pix, $FOV = 90°$	
Pix. noise	$\sigma_{\mathbf{u}}$	1 pix	
$\rho^{\mathcal{C}}$ prior	$(\bar{\rho}^{\mathcal{C}}, \sigma_{\rho^{\mathcal{C}}})$	$(0.01, 0.5)\ \text{m}^{-1}$	id. + $(1, 1)\ \text{m}^{-1}$



(a) AMPP outperforms HP



(b) AHP outperforms AMPP



(c) AHP outperforms HP

Figure 17. Bi-lateral comparisons of the linearity index for HP, AHP and AMPP. Each plot shows, at each frame, the number of landmarks of one type showing lower linearity index (better linearity) than the other type, and vice versa.

2) Linearity measures: The analysis of the linearity index proposed in section V-A, displayed in figures 17(a) to 17(c), confirms the results from the consistency analysis. The linearity index has been computed for each measured point

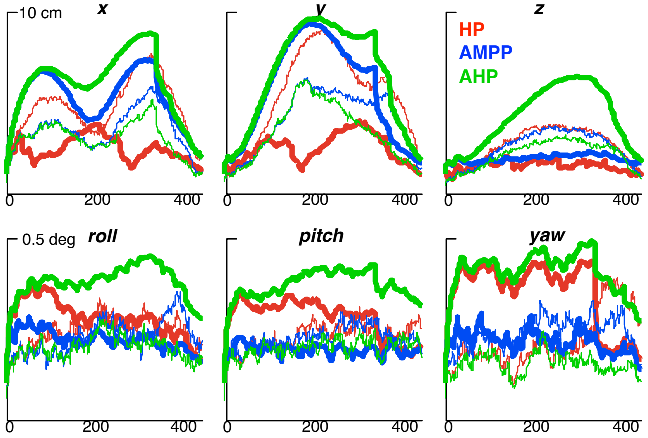


Figure 18. RMS errors of the three point parametrizations HP (red), AMPP (blue) and AHP (green), averaged over 25 runs. The 2-sigma estimated bounds are plotted in thicker line. AHP has the largest estimated bound and the lowest error, leading to the best results. See Fig. 20 for the corresponding consistency plots.

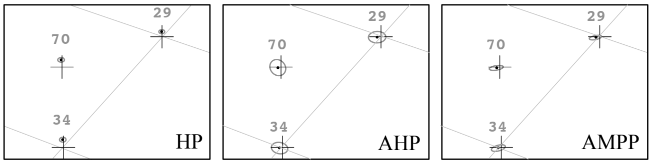


Figure 19. 3D view of some landmark 3σ estimates at the end of the first loop. Inconsistency comes from covariance overestimation rather than mean errors. See the accompanying video.

in the above experiments and for the three parametrizations of interest: HP, AMPP and AHP. Fig. 17(a) shows that, in every frame, there are always more features showing lower linearity index –and hence higher degree of linearity– when coded as AMPP instead of HP and vice versa. The reason, as previously mentioned, is that setting an anchor only propagates the camera uncertainty from the anchor to the current location, while HP propagates a wider uncertainty with respect to a world reference frame.

Fig. 17(b) analyses the linearity index for AMPP and AHP parametrizations. The extra non-linearities introduced by the computation of trigonometric functions from azimuth-elevation angles to direction ray translates in a large number of features showing higher degree of linearity when coded in AHP instead of AMPP.

Fig. 17(c) confirms the superiority of AHP and the non-suitability of HP by showing an even wider advantage of AHP with respect to HP than those showed in the previous plots.

3) Error and consistency analysis: We provide an accompanying video showing the three methods running in parallel. The differences in behavior are not easily visible in the 3D movies, and we need to zoom in to appreciate incorrect operation (Fig. 19: AMPP and HP estimate too small covariances). However, their NEES behavior is radically different (Fig. 20, please note the logarithmic vertical scales):

- HP behaves poorly. Of the 25 runs, one diverged, and 35 landmarks had to be deleted due to inconsistent observations (22 of which during the divergent run).

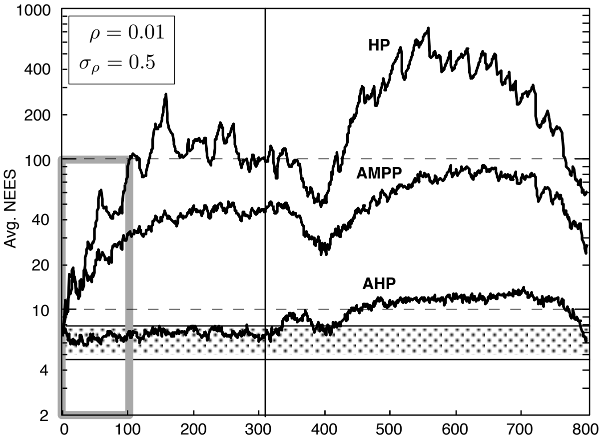


Figure 20. Consistency of HP, AMPP and AHP. Average NEES of the 6-DOF vehicle pose $[x,y,z,\phi,\theta,\psi]^\top$ over 25 runs for 800 frames (2 turns) and parameters of Set 1. The dotted horizontal band between abscissas $\underline{\epsilon} = 4.719$ and $\overline{\epsilon} = 7.432$ mark the 95% consistency region: the filter is consistent if the average NEES is inside this band. The vertical line marks the loop closure at frame 308. The framed area corresponds to the area covered by Fig. 21.

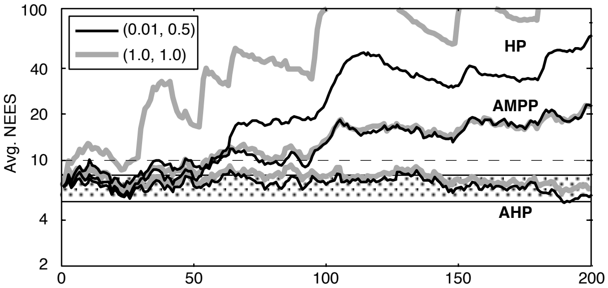


Figure 21. Consistency of HP, AMPP and AHP. Average NEES over 25 runs for 200 frames (1/4 turn) with parameters of Set 2 and 10 initializations in the first frame. *Thick gray:* initialization with an alternative prior $(\bar{\rho}, \sigma_\rho) = (1, 1)$.

- AHP behaves consistently, certainly with a slight tendency to inconsistency, until shortly after the first loop closure. During the second turn the filter is inconsistent but it does not seem to degrade too quickly. No landmarks were declared inconsistent.
- AMPP shows better performance than HP but also escapes consistency very quickly. No run diverged but inconsistent observations triggered landmark deletion in two occasions.

We tuned the algorithms with the second set of parameters in order to improve linearity: odometry steps and noise are cut in half, and the filter is bootstrapped with 10 landmarks being initialized at the first frame. Here, we focus on the first quarter of the first loop (1/8 of the first run's length) to see the moment when the filters loose consistency. The results in Fig. 21 show no significant improvement with respect to those of Set 1 (these 200 frames correspond to the first 100 frames of Set 1, which have been boxed in Fig. 20): HP is just not good, AMPP starts fine but only keeps track until frame 50, and AHP is again the only one to behave consistently.

A third test consisted in selecting a different prior for the unmeasurable inverse-distance. The gray superimposed plots in Fig. 21 show that AMPP and AHP are not very sensitive to

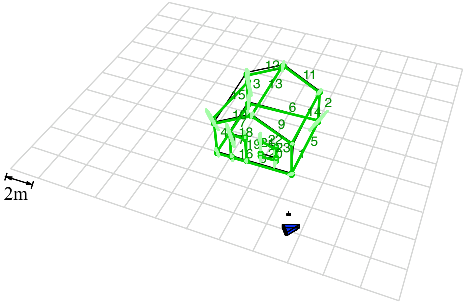


Figure 22. Simulated 3D environment for benchmarking 5 line parametrizations.

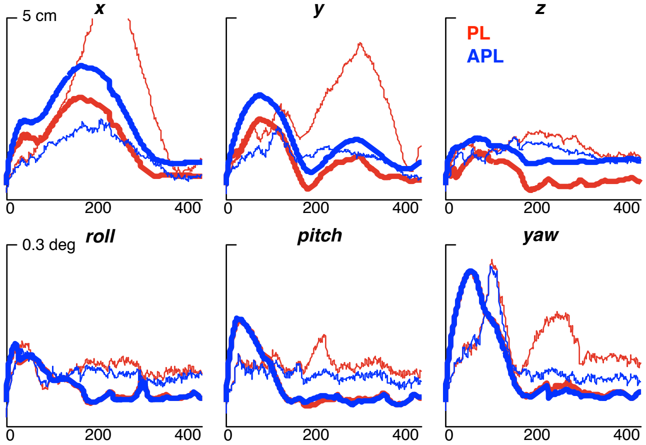


Figure 23. RMS errors of the two Plücker-based line parametrizations PL (red) and APL (blue), averaged over 25 runs. The 2-sigma estimated bounds are plotted in thicker line. Anchoring the Plücker line helps in improving the estimates, but none of these Plücker parametrizations seem to work correctly. See discussion in Section VII.

large variations of these parameters, while the contrary must be said for HP. It seems, even if for AHP and AMPP the difference is small, that the filter behaves better with landmarks initialized at (or close to) infinity ($\bar{\rho}^{\mathcal{C}} = 0.01\mathrm{m}^{-1}$) than at some close distance ($\bar{\rho}^{\mathcal{C}} = 1\mathrm{m}^{-1}$).

C. Lines

We benchmark PL, APL, HPL, AHPL and AMPPL for NEES consistency. The first scenario (shown in Fig. 22) consists of a robot making a turn around a wireframe model of a house. 400 frames are processed. The camera is looking sideways to the house and there is no loop closure. The simulation parameters in terms of sensor calibration and odometry noise levels are equivalent to the ones we used for points. The results are shown in Fig. 25 – please notice the logarithmic vertical scales.

The second scenario corresponds to a frontal trajectory, a situation that is more challenging for monocular SLAM as the scene observability becomes weaker. The camera looks forward and the robot performs an arc of a circle towards the house. The sequence is stopped after 100 frames when the robot is actually inside the house and no more segments are in the field of view. In this case we just show results for the two winning parametrizations (Fig. 26), namely AHPL and AMPPL. The rest are clearly inconsistent, and in fact the results are pretty much the same as in the first scenario (compare to the two lower plots in Fig. 25).

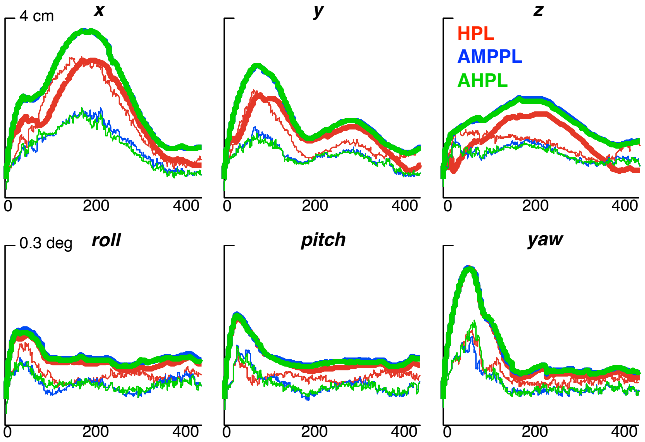


Figure 24. RMS errors of the three point-based line parametrizations HPL (red), AMPPL (blue) and AHPL (green), averaged over 25 runs. The 2-sigma estimated bounds are plotted in thicker line. Anchoring has produced both a larger 2-sigma bound error estimate, and a smaller error. AMPPL and AHPL behave almost exactly. See the corresponding NEES consistency plots in Fig. 25.

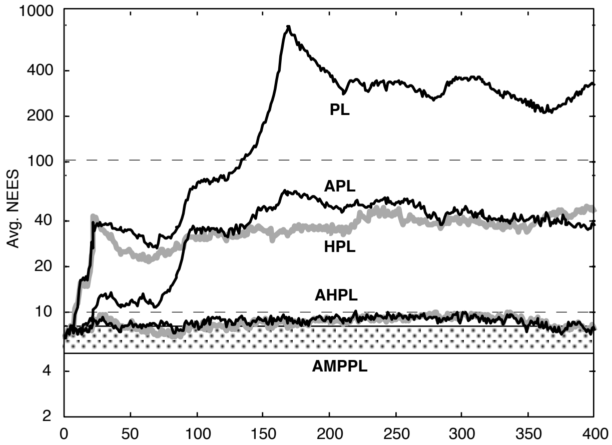


Figure 25. Consistency of PL, APL, HPL, AHPL and AMPPL. Average NEES over 25 runs for 400 frames (one turn around the house). Plücker-based lines (PL and APL) do not behave consistently, even when anchored. Lines based on homogeneous points (HPL) neither, as expected from the HP behavior. Anchored point-supported lines (AHPL and AMPPL) behave similarly and close to consistency.

VII. DISCUSSION

A. Anchors and non re-measurable states

We have shown in this paper the benefits of anchoring landmarks as a way to improve linearity and therefore EKF performance. Here we give some more insight to the subject.

The anchor constitutes a part of the SLAM state that is not directly re-measurable. In this regard, it stays in the map like any other landmark that is not being measured. Importantly, as long as the filter is run under conditions of sufficient linearity, the anchor benefits from the ability of fully-correlated SLAM to keep non-measured states up-to-date and consistent with

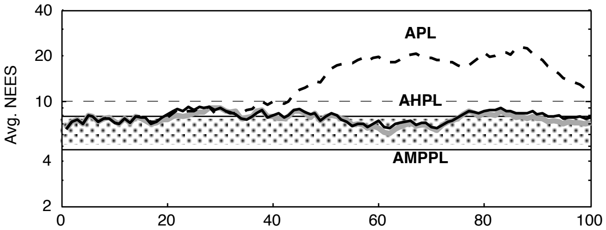


Figure 26. Average NEES over 25 runs for 100 frames (frontal trajectory). Again, only anchored point-supported lines behave close to consistency.

their neighbors. This is particularly interesting during loop closures where estimates may vary abruptly after one single EKF update.

The anchor also allows the observation function to operate at a reduced uncertainty level, allowing higher linearity as we have seen in Section V-A. We illustrate this in very general terms. Consider that, due to motion, an uncertainty $\Delta\mathbf{P}$ is added to the sensor position T over a period of time Δt. If the position covariance at initialization time t_0 is $\mathbf{P}_0$, at a later observation time $t_1 = t_0 + \Delta t$ we have $\mathbf{P}_1 = \mathbf{P}_0 + \Delta\mathbf{P}$. For observations shortly after initialization, we have Δt small and $\Delta\mathbf{P} << \mathbf{P}_0$. Now consider the general forms of the observation functions (see Table I) of one un-anchored landmark $\mathcal{L}_u = (A_u, b_u)$ and one anchored landmark $\mathcal{L}_a = (\mathbf{p}_0, A_a, b_a)$,

$$\mathbf{y}_u = K\mathsf{R}^\top (A_u - \mathsf{T} b_u) \tag{85}$$

$$\mathbf{y}_a = K\mathsf{R}^\top (A_a - (\mathsf{T} - \mathbf{p}_0) b_a)\ . \tag{86}$$

The first function (un-anchored) works with the uncertainty level of T, which is $\mathbf{P}_1$. The terms A_u and b_u had been composed with $\mathbf{P}_0$ at initialization time t_0, via some Jacobian matrices computed at inaccurate estimates and therefore containing linearization errors. Due to cross-correlations, the two contributions come actually close to cancellation and the uncertainty in the innovation space is theoretically close to that produced by just $\Delta\mathbf{P}$, but the linearization errors are unrecoverable. The second function (anchored) works with the uncertainty of $(\mathsf{T} - \mathbf{p}_0)$ which is $\Delta\mathbf{P} << \mathbf{P}_1$. This is directly mapped to the innovation space. The terms A_a and b_a are correlated to the rest of the map only through the anchor $\mathbf{p}_0$, which appears explicitly in the formula. This anchor was established at initialization time via $\mathbf{p}_0 = \mathsf{T}(t_0)$, *i.e.*, without linearization errors. The EKF update is free from the linearization errors of the translational composition above.

B. Anchored landmarks versus robocentric SLAM

An interesting alternative to the algorithm here benchmarked is robocentric SLAM [11], [26], [36]. Robocentric SLAM adds a step to the estimation loop, as follows:

1) Motion prediction with $f()$ (not described in this work).
2) Landmark observations with $h()$, innovation with $j()$, EKF correction, and re-parametrization with $k()$.
3) Map transformation to robot frame with $m()$ (not described in this work).
4) Landmark initialization with $g()$.

Item 3 above has the consequence of making the robot pose to be the origin at every step. This has interesting implications. For what matters here, we notice that the robot position becomes $\mathsf{T} \equiv \mathbf{0}$, with null covariance. Therefore, at initialization time (Step 4), we have that the anchor to generate is $\mathbf{p}_0 = \mathsf{T} = \mathbf{0}$. We can then drop it from the parametrization, obtaining, in *e.g.* the AHP case, simple homogeneous points HP – this is exactly the work presented in [11]. In any case (HP, HPL or PL), we are curious to see if robocentric SLAM can achieve the linearity levels of anchored parametrizations (or even improve on them because also the rotational part and its covariance are set to zero) at lower parametrization costs. The price to pay is the cost of Step 3 which is quadratic with the map size [36].

TODO: *Add robocentric results from J. Civera.*

C. The Plücker constraint

We have seen that for a pair $(\mathbf{n}, \mathbf{v})$ to be a Plücker line the Plücker constraint $\mathbf{n} \perp \mathbf{v}$ is mandatory. We have ensured its satisfaction at initialization time, but we have not enforced it further during landmark updates. The reason is that we did not find a clean and convincing method for doing so in the EKF framework. Enforcing equality constraints (*i.e.*, with infinite information) in EKF has the consequence of producing singular covariances matrices. The directions of the state space being affected by this singularity become blocked and no more evolution on them can be expected. If the constraint could be applied exactly, and if our system were linear, this would create a lifelong guarantee of the constraint satisfaction. But due to linearization errors, the produced singularity is not exactly at the right point (*i.e.*, the eigenvector corresponding to the null singular value does not have the correct direction). This problem has been treated in [13] for the Plücker lines using the smoothly constrained Kalman Filter [37]. The idea is to apply a number of relaxed constraints (with gradually increasing information) to make the filter converge to the true constraint. This method requires several tuning parameters and, as mentioned, we do not feel the solution to be very elegant.

This neglected step might very well be at the base of the poor consistency results of APL. We have not investigated the validity of this hypothesis, mainly because there are other reasons to prefer point-supported lines over Plücker lines, as we see next.

D. Endpoints in Plücker-based and point-supported lines

The methods described here for endpoint management require some information to be stored out of the map. We limited this to two abscissas. Plücker lines condensate all the information of the initial observation to the plane normal, via $\mathbf{n} = \mathbf{m}_1 \times \mathbf{m}_2$, and any information on the endpoints' initial view is lost. The two abscissas representation reveals very limited for Plücker lines: the local line origin moves with the line's orientation, and the segment length does not change upon distance variations. During the initial convergence phase, the only reasonable strategy for managing endpoints is to reset them at each frame using the current observation (Fig. 14(b)), potentially losing information about the segment extension observed in previous frames.

On the contrary, Anchored, point-supported line parametrizations have the ability to respect the initial view of the two segment endpoints via the anchor $\mathbf{p}_0$ and the two director vectors $(\mathbf{m}_1, \mathbf{m}_2)$. Because of this higher representativeness of point-supported lines, and because of the absence of constraints, we consider point-supported lines preferable to Plücker-based lines for undelayed monocular SLAM. The additional cost is marginal (size 11 instead of 9) and needs to be paid only during the convergence phase. Reparametrization is triggered with the same criterion described in [9], which must hold for each endpoint.

E. Modified Polar Coordinates

We notice here that the AMPP's un-anchored counterpart, *Modified Polar Point* (MPP – refer to Fig. 6), not studied in this paper, had already been treated in the 80's in the bearing-only tracking literature with similar problematic and justification [39], where it was referred to as Modified Polar Coordinates (MPC). Its use in monocular EKF-SLAM is not recommended: it presents a singularity at the origin, and if we draw the correct conclusions from the present paper, it should behave even worse than HP, as it happens to AMPP with respect to AHP. The same thing happens if we try to use MPC for lines.

VIII. CONCLUSION

REFERENCES

[1] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Int. Conf. on Computer Vision*, vol. 2, Nice, October 2003, pp. 1403–1410.
[2] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1987.
[3] T. Bailey, "Constrained initialisation for bearing-only SLAM," in *Int. Conf. on Robotics and Automation*, vol. 2, 2003, pp. 1966–1971.
[4] K. N.M. and D. G., "Bearing-only SLAM in indoor environments using a modified particle filter," in *Australasian Conf. on Robotics and Automation (ACRA)*, Brisbane, Australia, Dec. 2003.
[5] T. Lemaire, S. Lacroix, and J. Solà, "A practical 3D bearing only SLAM algorithm," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, Canada, August 2005.
[6] 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.
[7] N. M. Kwok and G. Dissanayake, "An efficient multiple hypothesis filter for bearing-only SLAM," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sendai, Japan, 2004.
[8] J. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular SLAM," in *Robotics: Science and Systems*, Philadelphia, USA, August 2006.
[9] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. on Robotics*, vol. 24, no. 5, 2008.
[10] 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.
[11] D. Marzorati, M. Matteucci, D. Migliore, and D. G. Sorrenti, "Monocular SLAM with inverse scaling parametrization," in *Proc. of the British Machine Vision Conference*, Leeds, 2008.
[12] J. Solà, T. Vidal-Calleja, and M. Devy, "Undelayed initialization of line segments in monocular SLAM," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Saint Louis, USA, Oct. 2009, pp. 1553–1558.
[13] 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.
[14] 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.
[15] A. P. Gee and W. Mayol, "Real-time model-based SLAM using line segments," in *LNCS proceedings of the 2nd International Symposium on Visual Computing*, Nov. 2006.
[16] E. Eade and T. Drummond, "Edge landmarks in monocular SLAM," in *British Machine Vision Conf.*, Edimburgh, Scotland, September 2006.
[17] G. Klein and D. Murray, "Improving the agility of keyframe-based SLAM," in *Proc. 10th European Conference on Computer Vision (ECCV'08)*, Marseille, October 2008, pp. 802–815.
[18] J. P. Barreto, "A unifying geometric representation for central projection systems," *Comput. Vis. Image Underst.*, vol. 103, no. 3, pp. 208–217, 2006.
[19] E. Eade and T. Drummond, "Scalable monocular SLAM," *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 469–476, 2006.
[20] N. Sunderhauf, S. Lange, and P. Protzel, "Using the unscented kalman filter in mono-SLAM with inverse depth parametrization for autonomous airship control," in *IEEE Int. Workshop on Safety, Security and Rescue Robotics*, Rome, 2007.
[21] S. A. Holmes, G. Klein, and D. W. Murray, "A square root UKF for visual monoSLAM," in *IEEE Int. Conf. on Robotics and Automation*, Pasadena, 2008.
[22] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 2007, pp. 1–10.
[23] K. Konolige and M. Agrawal, "FrameSLAM: From bundle adjustment to real-time visual mapping," *IEEE Trans. on Robotics*, vol. 24, no. 5, pp. 1066–1077, Oct. 2008.
[24] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?" in *Int. Conf. on Robotics and Automation*, Anckorage, AK, May 2010.
[25] 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.
[26] J. Civera, "Real-time EKF-Based Structure from Motion," Ph.D. dissertation, Universidad de Zaragoza, Sept. 2009.
[27] A. P. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas, "Discovering higher level structure in visual SLAM," in *IEEE Trans. on Robotics special issue on Visual SLAM*, vol. 24, no. 5, Oct. 2008, pp. 980–990.
[28] 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.
[29] E. Eade and T. Drummond, "Monocular SLAM as a graph of coalesced observations," in *IEEE Int. Conf. on Computer Vision*, 2007.
[30] A. Einstein, *The Foundation of the General Theory of Relativity*. Annalen der Physik, 1916, English translation. [Online]. Available: http://www.alberteinstein.info/gallery/gtext3.html
[31] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. John Wiley and Sons, 2001.
[32] 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.
[33] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. D. Tardós, "A comparison of SLAM algorithms based on a graph of relations," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Saint Louis, USA, Oct. 2009.
[34] J. Solà, D. Marquez, J. M. Codol, and T. Vidal-Calleja. (2009) An EKF-SLAM toolbox for MATLAB. [Online]. Available: http://homepages.laas.fr/jsola/JoanSola/eng/toolbox.html
[35] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.
[36] J. A. Castellanos, R. Martinez-Cantin, J. Tardós, and J. Neira, "Robocentric map joining: improving the consistency of EKF-SLAM," in *Robotics and Autonomous Systems*, vol. 55, no. 1, 2007, pp. 21–29.
[37] J. D. Geeter, H. V. Brussel, J. D. Schutter, and M. Decréton, "A smoothly constrained Kalman filter," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 10, Oct. 1997, pp. 1171–1177.
[38] V. Aidala and S. Hammel, "Utilization of modified polar coordinates for bearings-only tracking," *IEEE Transactions on Automatic Control*, vol. 28, no. 3, pp. 283 – 294, March 1983.