**International Journal on Computer Vision manuscript No.**
(will be inserted by the editor)

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

**Joan Solà · Teresa Vidal-Calleja · Javier Civera · J. M. M. Montiel**

the date of receipt and acceptance should be inserted later

**Abstract** This paper explores the impact that landmark parametrization has in the performance of monocular, EKF-based, 6-DOF simultaneous localization and mapping (SLAM) in the context of undelayed landmark initialization.

Undelayed initialization in monocular SLAM challenges EKF because of the combination of non-linearity with the large uncertainty associated with the unmeasured degrees of freedom. In the EKF context, the goal of a good landmark parametrization is to improve the model's linearity as much as possible, improving the filter consistency, achieving robuster and more accurate localization and mapping.

This work compares the performances of eight different landmark parametrizations: three for points and five for straight lines. It highlights and justifies the keys for satisfactory operation: the use of parameters behaving proportionally to inverse-distance, and landmark anchoring. A unified EKF-SLAM framework is formulated as a benchmark for points and lines that is independent of the parametrization used. The paper also defines a generalized linearity index suited for the EKF, and uses it to compute and compare the degrees of linearity of each parametrization. Finally, all eight parametrizations are benchmarked employing analytical tools (the linearity index) and statistical tools (based on Monte Carlo error and consistency analysis), with simulations and real imagery data, using the standard and the robocentric EKF-SLAM formulations.

J. Solà
1. CNRS; LAAS; 7 avenue du Colonel Roche, F-31077 Toulouse, France
Université de Toulouse; UPS, INSA, INP, ISAE; LAAS-CNRS : F-31077 Toulouse, France,
2. Ictineu Submarins SL, Barcelona, Catalonia,
jsola@ictineu.net.

T. Vidal-Calleja
University of Sydney, ACFR, Australia.
t.vidal@acfr.usyd.edu.au

J. Civera and J. M. M. Montiel
I3A, Universidad de Zaragoza, Spain.
{josemari, jcivera}@unizar.es

## 1 Introduction

Simultaneous localization and mapping (SLAM) is the problem of concurrently estimating in real time the structure of the surrounding world (the *map*), perceived by moving exteroceptive sensors, while simultaneously getting *localized* in it. The seminal solution to the problem by Smith and Cheeseman (1987) employs an extended Kalman filter (EKF) as the central estimator, and has been used extensively. In EKF-SLAM, the map is a large vector stacking camera and landmark states, and it is modeled by a Gaussian variable. This map, usually called the *stochastic map*, is maintained by the EKF through the processes of prediction (the camera moves) and correction (the camera observes the landmarks in the environment that had been previously mapped).

In order to achieve true exploration, the EKF machinery is enriched with an extra step of *landmark initialization*, where newly discovered landmarks are added to the map. Landmark initialization is performed by inverting the observation function and using it and its Jacobians to compute, from the camera pose and the measurements, the observed landmark state and its necessary co- and cross- correlations with the rest of the map. These relations are then appended to the state vector and the covariances matrix.

Monocular SLAM refers to the case where the exteroceptive sensing means are limited to a single projective camera. Monocular SLAM gained popularity back in 2003 thanks to the first full real-time implementation by Davison (2003), based on Smith and Cheeseman's EKF solution. Davison's technique elegantly solved a great number

of problems, but there still remained one that occupied researchers on visual SLAM for some years (Chiuso et al, 2002; Bailey, 2003; Kwok and Dissanayake, 2003; Lemaire et al, 2005): the problem of landmark initialization.

Landmark initialization in monocular SLAM is difficult because, due to the projective nature of the sensor, this cannot provide the distance to the perceived landmarks: the measurements are rank-deficient and the observation functions are not invertible. This means that a full 3D estimate of the landmarks just discovered is not available for mapping, because the uncertainty in the unobserved DOF is infinite and, being the measurement equations non-linear, EKF cannot deal with it. Early approaches (Davison, 2003; Bailey, 2003; Lemaire et al, 2005) took advantage of the sensor motion to achieve fully 3D estimates before actually initializing the landmarks. This family of methods introduces a *delay* until the sensor motion has gained enough parallax for triangulation, during which the landmarks, not yet mapped, cannot provide any information for localization.

Monocular EKF-SLAM reached maturity with the advent of *undelayed landmark initialization* techniques (ULI, explained in more detail in the next section), a need of partial initialization firstly stated by Solà et al (2005). The key to ULI is to substitute the unmeasured DOF by a Gaussian prior: the objective then is to find a way to allow this prior to possess an infinite uncertainty while still being manageable by the EKF. Solà et al describe a preliminary solution based on an exponentially distributed multi-hypotheses depth parametrization, which was inspired on a previous work by Kwok and Dissanayake (2004). The problem was successfully solved for the first time with the inverse-depth landmark parametrization (IDP) by Montiel et al (2006), which has become very popular. More recently, Marzorati et al (2008) and Haner and Heyden (2010) have presented new parametrizations for which the authors claim better performances than IDP. Solà (2010) presents a comparative study of three parametrizations for point landmarks.

A smaller number of works incorporate line landmarks or *segments* to the EKF-SLAM framework. Gee and Mayol (2006), Smith et al (2006) and Lemaire and Lacroix (2007) use delayed techniques for initialization. Solà et al (2009b) reports the only ULI solution for infinite lines we are aware of, which uses the Plücker line. Edgelets (very short line fragments associated to a 3D point) were introduced by Eade and Drummond (2006a), also in an undelayed manner, using IDP as the supporting point type.

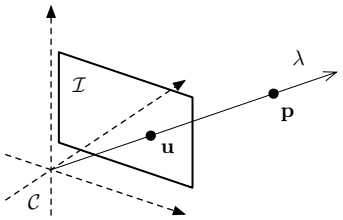
Overall, the methods here cited have many points in common. Unfortunately, their differences lie in many parts of the algorithm other than landmark parametrization, in the evaluation methods and/or in the heterogeneity of the experimental setups. This makes it difficult to tell which aspects of the proposed solutions are at the base of the observed differences in performance. In this work, we fix the algorithmic and experimental aspects of the problem and center our attention to the effect that landmark parametrization has, by its own right, on monocular EKF-SLAM performance. For this, the paper retakes the problem from a unified perspective that considers points and lines alike (edgelets are not covered), and presents and analyzes a compendium of eight different parametrizations, three for points and five for lines, among which three are innovative to the best of our knowledge.

### 1.1 Undelayed landmark initialization (ULI)

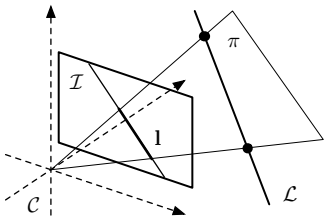
To overcome the drawbacks of delayed initialization, *undelayed landmark initialization* (ULI, also known as *partial initialization*, Solà et al, 2005) incorporates the partially measured landmarks at the first observation, that is, before all of their DOF are determined (or *sufficiently estimated*). In bearings-only systems (*e.g.* a monocular camera, see Fig. 1), ULI allows landmarks showing low parallax (those that are at remote distances or close to the motion direction of the camera) to contribute to SLAM from the first observation, providing precious bearing information that helps constrain the camera orientation. In other words, ULI allows us to exploit the full field of view of the camera up to the infinity range, regardless of the sensor trajectory, which results in accurate localization with very low angular drifts.

For example, when turning a corner in a corridor, a visual SLAM system with ULI can immediately initialize a point or two at the other end of the corridor, which have just become visible and will most likely remain visible along the whole corridor. During this time, the conditions for triangulation are bad, as there is no significant increase in parallax. Without ULI, these landmarks must be ignored with the consequence of the robot accumulating angular errors that after a few meters may become the primary source of filter failure due to inconsistency. Thanks to ULI, observing these landmarks serves to constrain the camera orientation, meaning that the robot can reach the end of the corridor without accumulating angular drift. The total angular drift for a whole loop closure (say, 4 corridors and 4 corners) is thus limited to only the drift accumulated during the transitions in the corners. We encourage the reader to consult (Solà et al, 2005; Civera et al, 2008; Solà et al, 2008) for discussions on delayed/undelayed initializations and their importance in monocular SLAM, and (Bailey et al, 2006; Huang and Dissanayake, 2007; Huang et al, 2008) for insights on the sources of inconsistency in EKF-SLAM.

ULI is an interesting challenge in EKF because the filter needs to cope with naturally non-linear equations and huge uncertainty levels associated with the unmeasured DOF (Fig. 1). The best solutions accepted so far require some degree of over-parametrization of the landmarks' states. Two aspects have been identified as being beneficial (Civera et al, 2008; Solà et al, 2009b): first, the enormous (potentially infinite)




(a) The back-projection of a point $\mathbf{u}$ gives place to a semi-infinite line $\lambda$ where the point landmark $\mathbf{p}$ must lie. There is 1 unmeasured DOF: the point's depth or distance. To observe it, the camera needs to gain parallax by moving away from the line $\lambda$.




(b) The back-projection of a segment $\mathbf{l}$ gives place to a semi-infinite plane $\pi$ where the line landmark $\mathcal{L}$ must lie. There are 2 unmeasured DOF: the line's depth and its orientation in $\pi$. The camera gains parallax by moving away from the plane $\pi$.

**Fig. 1** The problem of undelayed initialization. Back-projection of detected features in a monocular image at their first observation. The unmeasured DOF's have infinite uncertainty and need to be properly modeled by Gaussian shapes, and manipulated using reasonably linear functions.

uncertainty must be represented by a single and well-defined (*i.e.*, bounded) Gaussian. Second, the observation functions must be reasonably linear inside all this uncertainty range. These two severe requirements can be elegantly fulfilled by using parametrizations incorporating the non-observable DOFs proportionally to inverse-distance, as it is done with *e.g.* IDP (Civera et al, 2008), homogeneous points (Marzorati et al, 2008) and Plücker lines (Solà et al, 2009b). This is because, on one hand, a bounded Gaussian in inverse-distance including the origin of coordinates naturally maps onto an unbounded uncertainty region including the infinity, and on the other hand, the inverse-distance is key in projective geometry and the projection equations exploiting it become quasi-linear precisely with respect to these highly uncertain DOF.

A third aspect that has proved positive is landmark anchoring. Although not explicitly stated, anchoring was already used in the delayed method of Davison (2003), and later in IDP. Recently, it has been explicitly evaluated by Solà (2010), who compares three different point parametrizations. Anchoring allows the landmark uncertainty to be referenced to a point close-by (the *anchor*), which is chosen to be the optical center at initialization time. This allows the system to get rid of many linearization errors accumulated since the start of the map, and to consider instead mainly the local motion since the initialization of each particular landmark. More complex anchoring uses the whole camera pose (position and orientation), achieving a higher degree of decoupling between global and local motions (Gee and Mayol, 2006; Gee et al, 2008). These last schemes use shared anchoring to keep the map size small, and thus require that the landmarks be initialized in groups.

### 1.2 Points and straight lines

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

For *points*, Fig. 1(a), ULI means that landmarks must be initialized so that the uncertainty in distance – the only unmeasurable DOF – covers all the visual ray up to infinity.

For *infinite straight lines*, Fig. 1(b), ULI requires the initial uncertainty to cover 2 unmeasurable DOF: distance up to infinity, and all possible orientations.

Bounded lines or *segments* present additional difficulties. Unlike points, lines can be partially occluded, and the edge detectors in use return therefore unstable endpoints. This means that the endpoints of a 3D segment cannot be established from single observations, and that they are generally not re-observable. For these (and other possible) reasons, it is common practice to employ the stochastic map to estimate just the infinite lines supporting the segments, and to keep track of the segment's endpoints separately. In this paper, we focus mainly on the estimation of infinite lines supporting arbitrarily long segments (not edgelets), and only general guidelines are given about the management of the segments' endpoints.

### 1.3 Alternative approaches to monocular EKF-SLAM

There exist a significant amount of research investigating the possibilities of using estimation techniques other than EKF. We find IDP used in Bayesian frameworks such as FastSLAM2.0 (Eade and Drummond, 2006b) and the unscented Kalman filter (UKF, Sunderhauf et al, 2007; Holmes et al, 2008). Very recently, methods based on Bundle Adjustment optimization (BA, Triggs et al, 2000; Engels et al, 2006) over a sparse set of keyframes on the sequence are gaining popularity (Klein and Murray, 2007; Konolige and Agrawal, 2008). Real-time operation has been achieved by dividing the SLAM operation into a Bundle Adjustment thread, using mainly the software in (SBA, Lourakis and Argyros, 2004) and a camera tracking thread using pairwise geometry. Those keyframe approaches have also been successfully used with edgelets (Klein and Murray, 2008).

Very recently, (Strasdat et al, 2010) has proved a clear advantage of keyframe SLAM algorithms: while the dominant computation for EKF-SLAM (a complexity $\mathcal{O}(n^2)$ in the state covariance update) has to be performed at every step, the cost of the non-linear optimization in keyframe SLAM is amortized among several frames. As a consequence, keyframe algorithms are able to include and measure more features in their maps, hence improving the general accuracy of the keyframe estimation with respect to the EKF one. Then, while there exist EKF-based algorithms with performances comparable to keyframe-based ones (Paz et al, 2008; Civera, 2009), they present a higher computational cost per map landmark.

Still, EKF-SLAM (or other similar approaches based on filtering and Gaussians such as UKF-SLAM or extended information filter (EIF)-SLAM) is still widely used by major robotics and vision laboratories and is at the core of other localization, mapping or modeling systems, with points (Paz et al, 2008; Civera, 2009) (with performances comparable to those of SBA), lines (Gee and Mayol, 2006), and even introducing planes (Gee et al, 2008). The opinion of the authors is that EKF-SLAM can have an important niche of applications: as stated in (Strasdat et al, 2010), EKF monocular SLAM presents computational advantages in cases where the computational budget is low. This particular case could be of importance now that smart mobile devices are populating our lives. Also, the EKF keeps an uncertainty estimation for the map features that would be expensive to extract from a keyframe algorithm. This is especially valuable in situations where only a few landmarks are visible, as the filter keeps a coherent estimate thanks to the prediction stage of EKF, which is missing in non-linear optimization schemes. Finally, and apart from the fact that EKF-SLAM is the implementation with the longest tradition, two other technical 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: one is the mentioned computational burden; the other is filter consistency, presented in the following paragraphs.

### 1.4 Linearity and EKF consistency

The consistency issues of EKF-SLAM are well known and have been the subject of numerous studies in the last years. Castellanos et al (2004) showed that inconsistency appears even before the computational burden of the problem becomes prohibitive, and proposed in (Castellanos et al, 2007) robocentric SLAM where the local operation of the filter results in significant linearity improvements. A more concise study of inconsistency is given by Bailey et al (2006), where the normalized estimation error squared (NEES) is averaged over a number of conditionally independent Monte Carlo runs and used to evaluate consistency. This work shows that using ground truth Jacobians guarantees filter consistency, and thus that inconsistency comes from the unavoidable errors produced when linearizing the system. More theoretically sound insights have been provided by a remarkable work by Huang et al (2008), where it is shown that, using the authors' words, "the observable subspace of the linearized system is of higher dimension than that of the actual, non-linear one, leading to covariance reductions in directions of the state where no information is available, which is a primary cause of inconsistency".

All the studies mentioned above assume 2D implementations using range-and-bearing sensing and Euclidean point parametrizations, exactly as they appear in the original EKF-SLAM solution. Our case of study differs from them in at least four aspects. The first one is 3D operation (*i.e.*, 6 DOF motion). The second one is that we are dealing with monocular observations, which convey rank-deficient information about the landmark locations. The third aspect, which is a consequence of the previous one, is that landmark parametrization can no longer be the trivial, minimal, Euclidean one, but something more or less complicated and redundant that seeks an improvement of linearity. The fourth and last aspect is that we also incorporate lines.

Our aim with this paper, however, is not a theoretical mathematical analysis (in the style especially of (Huang et al, 2008)) but a performance comparison that visualizes the impact that landmark parametrization has on linearity, estimation error and filter consistency. We show that inconsistency comes mostly from covariance over-estimation rather than error magnitude, which corroborates Huang's conclusions.

Because inconsistency has its roots in non-linearity, we correlate our evaluation with measurements of the degree of linearity of each parametrization. We define for this purpose a linearity index that on one hand is pertinent to EKF (*i.e.*, it accounts for non-linearity *and* uncertainty), and on the other hand it defines its metric in the measurement space and therefore allows us to compare parametrizations having state representations of different sizes and natures.

The choice of the classical EKF engine for SLAM is not casual: as a well-known algorithm, it serves the purpose of a standard workbench through which to evaluate performance differences that have their roots in non-linearity. As an interesting counterpoint, we additionally show with large-scale experiments that algorithms robuster to non-linearity such as robocentric EKF-SLAM also benefit from the linearity improvements of the landmark parametrizations proposed in this article.

### 1.5 Contributions

We provide several contributions:

1. A compendium of eight landmark parametrizations for ULI, three for points (homogeneous points HP, anchored homogeneous points AHP, and anchored modified-polar points AMPP) and five for lines (Plücker lines PL, anchored Plücker lines APL, homogeneous-points lines HPL, anchored homogeneous-points lines AHPL, and anchored modified-polar-points lines AMPPL). Three of these parametrizations (APL, HPL and AMPPL) are innovative to the best of our knowledge.
2. A unified methodology to tackle all eight parametrizations emphasizing the two keys to satisfactory ULI, namely landmark anchoring and inverse-distance behavior.
3. An analytical measure of linearity of multi-dimensional functions that takes into account the time-varying support of probability.
4. A statistical evaluation of root mean squared (RMS) errors and average normalized estimation error squared (NEES) consistency, based on Monte-Carlo simulation runs.
5. A benchmark with real outdoors imagery of the point parametrizations on a robocentric SLAM implementation, showing that our proposed solutions achieve error levels typical of state-of-the-art SLAM based on non-linear optimization.

### 1.6 Outline

This paper is organized as follows. In Section 2 we describe three parametrizations for points and give details on the necessary algebra to support them. In Section 3 we repeat the process with five types of infinite lines. Section 4 describes the initialization and updating procedures. Section 5 describes the methods we use for linearity and consistency evaluation, with simulation results in Section 6. Further results with real images are presented in Section 7. The paper continues with a discussion in Section 8 and the conclusions in Section 9. A final appendix gives accessory details on segments endpoints management.

## 2 Parametrizations for 3D points

This section presents some parametrizations for 3D points, with their projection and back-projection operations needed for EKF-SLAM initialization and updates. We start with *Euclidean points* (EP, not suited for ULI) 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 refer to here as *anchored modified-polar points* (AMPP) for reasons that will be explained soon.

### 2.1 Euclidean points (EP)

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

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

where we use $\mathcal{L}_{NAME}$ to represent a landmark of type *NAME*.

Transformation to camera frame and perspective (pinhole) 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 in projective spaces $\mathbb{P}^n$; $\mathbf{K}$ is the intrinsic matrix,

$$\mathbf{K} \triangleq \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \;; \tag{2}$$

$\mathsf{R} = \mathsf{R}(\mathsf{Q})$ and $\mathsf{T}$ are the rotation matrix and the translation vector defining the camera frame $\mathsf{C}$, which is coded by the vector $\mathsf{C} = (\mathsf{T}, \mathsf{Q})$, $\mathsf{Q}$ being an orientation representation of our choice suitable for EKF filtering.[1]

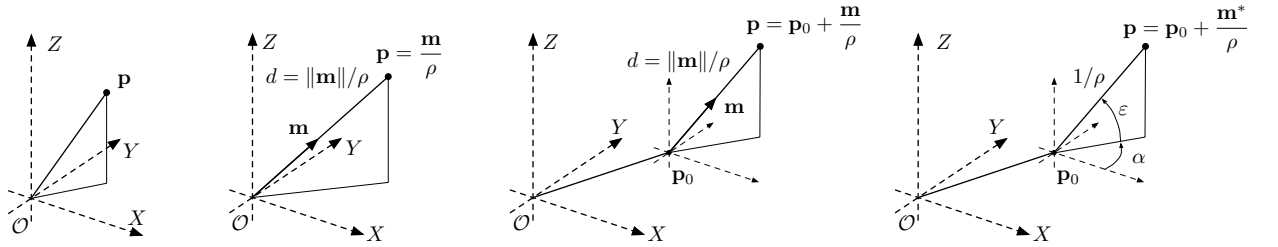
Euclidean points lead to severely non-linear observation functions in bearings-only systems and are not suited for undelayed initialization, as it has been extensively demonstrated, (Chiuso et al, 2002; Bailey, 2003; Davison, 2003; Kwok and Dissanayake, 2004; Eade and Drummond, 2006b; Solà et al, 2008) and most particularly (Solà et al, 2005; Civera et al, 2008). In brief, the problem of ULI with EP can be explained as follows. In EKF, the requirements of function linearity must hold inside the whole uncertainty region of the state variable. Because in Euclidean parametrizations the uncertainty region of partially observed landmarks is unbounded (it reaches the infinity in parameter space), the observation functions' linear approximation should hold for a whole unbounded interval, and this is impossible.

### 2.2 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 $\rho$ (usually referred to as the *homogeneous part*),

$$\mathcal{L}_{HP} = \underline{\mathbf{p}} = \begin{bmatrix} \mathbf{m} \\ \rho \end{bmatrix} = \begin{bmatrix} m_x & m_y & m_z & \rho \end{bmatrix}^\top \in \mathbb{P}^3 \subset \mathbb{R}^4 \,. \tag{3}$$

[1] We use normalized quaternions for encoding orientation, mainly because of the absence of gimbal lock, and because the bilinear relations appearing in the expression of the rotation matrix make the computation of Jacobians very easy.




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

**Fig. 2** Point parametrizations. (a) EP is minimal but not suited for ULI. (b,c) HP and AHP do not require $\mathbf{m}$ to be a unit vector. (d) In AMPP the observed ray is coded by two angles: the derived direction vector is unitary and hence $\rho$ is exactly inverse-distance. The anchor point $\mathbf{p}_0$ in AHP and AMPP corresponds to the optical center at initialization time.

A HP refers to the following EP:

$$\mathbf{p} = \mathbf{m}/\rho \,. \tag{4}$$

Although HP have been only recently introduced for monocular EKF-SLAM by Marzorati et al (2008), they have been extensively used in computer vision for years. In homogeneous coordinates, a point in 3D space is represented by an equivalence class, under proportionality transforms, of a 4-vector $(\mathbf{m}, \rho)$. Based on this equivalence several canonical choices are possible. The choice $\rho = 1$ is the original Euclidean point representation; the choice $m_z = 1$ is the conventional inverse-depth choice; and the choice $\|\mathbf{m}\| = 1$ is the inverse-distance choice. Note that the last two are not absolutely equivalent, although they are very similar. Unlike depth (or inverse-depth), which is defined with respect to a particular direction in space, inverse-distance has the advantage of being isotropic, that is, its properties are independent of the orientation of the reference frame.

In this paper, however, we do not make a canonical choice, and let the four parameters of $\underline{\mathbf{p}}$ free to move to the values determined by the different steps of the EKF estimator.[2] In HP we rather exploit the fact that the scalar $\rho$ is *proportional to* inverse-distance, as stressed in the following remark.

*Remark 1 (Inverse-distance)* When the point is expressed in the camera frame, the vector $\mathbf{m}$ corresponds to a director vector of the observed optical ray, and the scalar $\rho$ depends linearly with the inverse of the distance $d$ from the optical center to the 3D point,

$$\rho = \|\mathbf{m}\|/d \,. \tag{5}$$

The unbounded distance uncertainty, which spans from a minimal distance $d_{min}$ to infinity, is transformed into a bounded interval $\rho \in (0, \|\mathbf{m}\|/d_{min}]$ in parameter space. This is of central importance as this is precisely the factor that will allow us to use such parametrization for ULI (see Section 4.2 for further justification and details). The same concept of inverse-distance is found in all the parametrizations described here (except of course EP), even the ones for lines.

[2] In fact, we make use of other kinds of redundancy in our parametrizations, with very positive results. Refer to Section 8 for further discussion on redundant parametrizations in EKF.

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

$$\underline{\mathbf{p}} = \mathbf{H}\underline{\mathbf{p}}^{\mathsf{C}} \triangleq \begin{bmatrix} \mathsf{R} & \mathsf{T} \\ 0 & 1 \end{bmatrix} \underline{\mathbf{p}}^{\mathsf{C}} \,, \tag{6}$$

where the super-index $\bullet^{\mathsf{C}}$ indicates the local frame $\mathsf{C}$ where the point is referenced to, and $\mathbf{H}$ is the homogeneous motion matrix specifying this frame.

Homogeneous points project into perspective cameras according to

$$\underline{\mathbf{u}} = \mathbf{P}\underline{\mathbf{p}} = \mathbf{K}\mathbf{P}_0\mathbf{H}^{-1}\underline{\mathbf{p}} \,, \tag{7}$$

with $\mathbf{P} \triangleq \mathbf{K}\mathbf{P}_0\mathbf{H}^{-1}$, and where $\mathbf{P}_0$ is 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 $\mathsf{T}$, $\mathsf{R}$, $\mathbf{m}$ and $\rho$,

$$\underline{\mathbf{u}} = \mathbf{K}\mathsf{R}^{\top}(\mathbf{m} - \mathsf{T}\rho) \in \mathbb{P}^2 \,, \tag{8}$$

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

$$\underline{\mathbf{u}} = \mathbf{K}\cdot\mathbf{m}^{\mathsf{C}} \,, \tag{9}$$

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

On back-projection, the observed part $\mathbf{m}^{\mathsf{C}}$ in camera frame is obtained by just inverting (9),

$$\mathbf{m}^{\mathsf{C}} = \mathbf{K}^{-1}\underline{\mathbf{u}} \,. \tag{10}$$

The non-observed part $\rho^{\mathsf{C}}$ cannot be obtained from any data in the system, and must be provided as prior (see Section 4.2 about defining Gaussian priors appropriate for EKF). Overall, the back-projection and frame-transformation composition necessary for landmark initialization (see Section 4.3 for the initialization algorithm) is performed with

$$\mathcal{L}_{HP} = \underline{\mathbf{p}} = \begin{bmatrix}\mathbf{m}\\ \rho\end{bmatrix} = \mathbf{H}\begin{bmatrix}\mathbf{K}^{-1}\underline{\mathbf{u}}\\ \rho^{\mathsf{C}}\end{bmatrix} = \begin{bmatrix}\mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}} + \mathsf{T}\rho^{\mathsf{C}}\\ \rho^{\mathsf{C}}\end{bmatrix} \,, \tag{11}$$

where $\rho^{\mathsf{C}}$ depends inversely with the distance $d^{\mathsf{C}}$ to the camera, via $\rho^{\mathsf{C}} = \|\mathbf{K}^{-1}\underline{\mathbf{u}}\|/d^{\mathsf{C}}$. It must be provided as prior.

*Remark 2 (Inverse-distance and frame transformation)* In HP, the interpretation of $\rho$ as the inverse-distance from the point to the camera is lost after frame transformation (6), as $\rho$ becomes an inverse-distance to the origin of coordinates. Due to the bilinear character of this transformation, this might have more or less adverse effects on the performance of tools such as the EKF (which demand reasonably linear systems). One one hand, while $\mathbf{H}$ (that is, rotation $\mathsf{R}$ and translation $\mathsf{T}$) is accurately estimated, *i.e.*, after small camera motions, bilinearity can be considered quasi-linearity and the system is expected to work. On the other hand, when $\mathbf{H}$ is no longer accurate, *i.e.*, after large camera motions, the system is prone to failure. See also Remark 3.

### 2.3 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 (IDP, Civera et al, 2008), which we will see later. Anchoring the HP means referencing 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, Solà, 2010, Fig. 2(c)), parametrized with the 7-vector

$$\mathcal{L}_{AHP} = \begin{bmatrix}\mathbf{p}_0\\ \mathbf{m}\\ \rho\end{bmatrix} = \begin{bmatrix}x_0 & y_0 & z_0 & m_x & m_y & m_z & \rho\end{bmatrix}^\top \in \mathbb{R}^7 \,. \tag{12}$$

An AHP refers to the following EP:

$$\mathbf{p} = \mathbf{p}_0 + \mathbf{m}/\rho \,. \tag{13}$$

Transformation to camera frame and projection resume to

$$\underline{\mathbf{u}} = \mathbf{K}\mathsf{R}^\top\big(\mathbf{m} - (\mathsf{T} - \mathbf{p}_0)\rho\big) \in \mathbb{P}^2 \,. \tag{14}$$

*Remark 3 (Landmark anchoring)* Anchoring the landmarks at the optical center at initialization time has the effect of decoupling the uncertainty of the term multiplying the most uncertain parameter, the inverse-distance $\rho$. This term was $\mathsf{T}$ in HP and has become $(\mathsf{T} - \mathbf{p}_0)$ in AHP – see (8) and (14). It is easily seen that the uncertainty of $(\mathsf{T} - \mathbf{p}_0)$ is small after initialization, while the current camera pose $\mathsf{T}$ is not far from the anchor $\mathbf{p}_0$ and their cross-correlation is significant. See Remark 2 for the unanchored case, and Section 5.1 for the impact that uncertainty has on the degree of linearity seen by EKF.

The back-projection and transformation composition is performed with

$$\mathcal{L}_{AHP} = \begin{bmatrix}\mathbf{p}_0\\ \mathbf{m}\\ \rho\end{bmatrix} = \begin{bmatrix}\mathsf{T}\\ \mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}}\\ \rho^{\mathsf{C}}\end{bmatrix} \,, \tag{15}$$

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

### 2.4 Anchored modified-polar points (AMPP)

We lighten the previous AHP from 7 to 6 parameters by encoding the direction vector $\mathbf{m}$ with just elevation and azimuth angles $(\varepsilon, \alpha)$ of the observed optical ray joining $\mathbf{p}_0$ to $\mathbf{p}$. When these angles are appended with the inverse of the distance $\rho = 1/d$, 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, Civera et al, 2008, Fig. 2(d)), coded by the 6-vector

$$\mathcal{L}_{AMPP} = \begin{bmatrix}\mathbf{p}_0\\ (\varepsilon, \alpha)\\ \rho\end{bmatrix} = \begin{bmatrix}x_0 & y_0 & z_0 & \varepsilon & \alpha & \rho\end{bmatrix}^\top \in \mathbb{R}^6 \,. \tag{16}$$

*Remark 4 (Inverse-depth points)* In this article we refer to the originally named "inverse depth points" (IDP) in (Civera et al, 2008; Eade and Drummond, 2006b) 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 (Aidala and Hammel, 1983, and possibly earlier). 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:

$$\mathbf{p} = \mathbf{p}_0 + \mathbf{m}^*(\varepsilon,\alpha)/\rho \,, \tag{17}$$

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

$$\mathbf{m}^*(\varepsilon,\alpha) = \begin{bmatrix} \cos(\varepsilon)\cos(\alpha) \\ \cos(\varepsilon)\sin(\alpha) \\ \sin(\varepsilon) \end{bmatrix} \,. \tag{18}$$

Transformation to camera frame and pin-hole projection to the homogeneous plane are composed to give

$$\underline{\mathbf{u}} = \mathbf{K}\mathsf{R}^\top \big(\mathbf{m}^*(\varepsilon,\alpha) - (\mathsf{T} - \mathbf{p}_0)\rho\big) \,. \tag{19}$$

The back-projection and transformation composition is performed with

$$\mathcal{L}_{AMPP} = \begin{bmatrix} \mathbf{p}_0 \\ (\varepsilon,\alpha) \\ \rho \end{bmatrix} = \begin{bmatrix} \mathsf{T} \\ \mu^*(\mathsf{R}\mathbf{K}^{-1}\underline{\mathbf{u}}) \\ \rho^{\mathsf{C}} \end{bmatrix} \,, \tag{20}$$

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

$$\begin{bmatrix} \varepsilon \\ \alpha \end{bmatrix} = \mu^*(m_x, m_y, m_z) = \begin{bmatrix} \arctan(m_z/\sqrt{m_x^2 + m_y^2}) \\ \arctan(m_y/m_x) \end{bmatrix} \,. \tag{21}$$

The parameter $\rho^{\mathsf{C}}$ is now exactly the inverse-distance $1/d$ because $\mathbf{m}^*$ is unitary. It must be provided as prior.

## 3 Parametrizations for infinite straight 3D lines

This section mimics the structure of Section 2, now for the case of infinite straight lines. 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), which 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).

### 3.1 Plücker lines (PL)

This sub-section devoted to the Plücker line is long. We decided to include all this material because, for the sake of providing a coherent picture, it is important to highlight many interesting connections between homogeneous points (HP) and Plücker lines (PL), notably the existence of bilinear transformation and projection equations reproducing the structure of those of HP, and the inverse-distance behavior of the homogeneous part of the Plücker vector. These connections 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 of lines in monocular EKF-SLAM. Most of the material here can be found in (Solà et al, 2009b). The geometry of the Plücker line is taken from (Bartoli and Sturm, 2001).

#### *3.1.1 The Plücker coordinates*

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

$$\mathbf{L} = \underline{\mathbf{b}}\cdot\underline{\mathbf{a}}^\top - \underline{\mathbf{a}}\cdot\underline{\mathbf{b}}^\top \in \mathbb{R}^{4\times4} \,, \tag{22}$$

with $\underline{\mathbf{a}} = (\mathbf{a}, a) \in \mathbb{P}^3$ and the same for $\underline{\mathbf{b}}$. This is a $4\times4$ skew-symmetric matrix (with 12 off-diagonal entries $l_{ij} = -l_{ji}$) subject to the *Plücker constraint*,

$$\det(\mathbf{L}) = 0 \,. \tag{23}$$

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 $\mathcal{L}_{PL} \in \mathbb{P}^5$ 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 by Bartoli and Sturm (2001), that we name here the *Plücker line* (PL, introduced to monocular EKF-SLAM by Lemaire and Lacroix (2007), and then by Solà et al (2009b) implementing ULI, Fig. 3(a)),

$$\mathcal{L}_{PL} = \begin{bmatrix} \mathbf{n} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} n_x & n_y & n_z & v_x & v_y & v_z \end{bmatrix}^\top \in \mathbb{P}^5 \subset \mathbb{R}^6 \,, \tag{24}$$

which corresponds to writing the Plücker matrix $\mathbf{L}$ as

$$\mathbf{L} = \begin{bmatrix} [\mathbf{n}]_\times & \mathbf{v} \\ -\mathbf{v}^\top & 0 \end{bmatrix}, \quad \mathbf{n},\mathbf{v} \in \mathbb{R}^3 \,, \tag{25}$$

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

$$[\mathbf{n}]_\times \triangleq \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix} \,. \tag{26}$$

This choice and the definition (22) allow us to write

$$\mathbf{n} = \mathbf{a}\times\mathbf{b} \tag{27}$$

$$\mathbf{v} = a\mathbf{b} - b\mathbf{a} \,, \tag{28}$$

with which the Plücker constraint becomes the orthogonality condition $\mathbf{n}^\top\mathbf{v} = 0$.


(a) Plücker line (PL). The line $\mathcal{L}$ and the origin $\mathcal{O}$ define the support plane $\pi$.

(b) Back-projection of a Plücker line. The prior $\beta$ for initialization is expressed in the base $\{\mathbf{e}_1, \mathbf{e}_2\}$.

**Fig. 3** Geometrical interpretations of the Plücker line, with back-projection details. The 3-vector $\mathbf{v}$ is not observable at initialization time. Its initial covariance, however, must be defined in the plane $\pi^{\mathsf{C}}$ by means of a 2D Gaussian prior $\beta$. See Fig. 10 for further details.

The Plücker coordinates, when defined as in (27–28), admit a comprehensible geometrical interpretation (in the Euclidean sense, Fig. 3(a)):

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

*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 (Lemaire and Lacroix, 2007) because its director vector $\mathbf{u}$ is normalized, *i.e.*, $\|\mathbf{u}\| = 1$ and hence $d = \|\mathbf{n}\|$. Instead of normalizing $\mathbf{v}$ (or $\mathbf{u}$), it would have been more interesting to normalize $\mathbf{n}$, yielding an exact inverse-distance $\|\mathbf{v}\| = 1/d$. Anyway, normalization is not really necessary: as we will see in this paper, just proportionality to inverse-distance is enough for achieving ULI. See also Remark 6.

### *3.1.2 Frame transformations and projection*

It is easy to see, via (6) and (22), that the Plücker matrix is transformed according to

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

This expression is linear in the components of $\mathbf{L}^{\mathsf{C}}$ and therefore a linear expression exists for its vector counterpart $\mathcal{L}_{PL}$. Having defined $\mathcal{L}_{PL} = (\mathbf{n}, \mathbf{v})$, the expression of the transformation is amazingly simple (Bartoli and Sturm, 2001):

$$\mathcal{L}_{PL} = \mathcal{H}\cdot\mathcal{L}^{\mathsf{C}}_{PL} \triangleq \begin{bmatrix} \mathsf{R} & [\mathsf{T}]_\times \mathsf{R} \\ 0 & \mathsf{R} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{n}^{\mathsf{C}} \\ \mathbf{v}^{\mathsf{C}} \end{bmatrix} \ . \tag{29}$$

The inverse transformation is performed with

$$\mathcal{L}^{\mathsf{C}}_{PL} = \mathcal{H}^{-1}\cdot\mathcal{L}_{PL} = \begin{bmatrix} \mathsf{R}^\top & -\mathsf{R}^\top[\mathsf{T}]_\times \\ \mathbf{0} & \mathsf{R}^\top \end{bmatrix} \cdot \begin{bmatrix} \mathbf{n} \\ \mathbf{v} \end{bmatrix} \ . \tag{30}$$

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 $\mathbf{L}$ (see (26) for the meaning of $[\mathbf{l}]_\times$). The corresponding linear expression for the projected line in homogeneous coordinates, $\mathbf{l} \in \mathbb{P}^2$, is also very simple:

$$\mathbf{l} = \mathcal{P}\cdot\mathcal{L}_{PL} = \mathcal{K}\cdot\mathcal{P}_0\cdot\mathcal{H}^{-1}\cdot\mathcal{L}_{PL} \ , \tag{31}$$

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

$$\mathbf{l} = \mathcal{K}\cdot\mathsf{R}^\top\cdot(\mathbf{n} - \mathsf{T}\times\mathbf{v}). \tag{32}$$

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

$$\mathbf{l} = \mathcal{K}\cdot\mathbf{n}^{\mathsf{C}} \ , \tag{33}$$

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

We can now fully observe the revealing parallelisms between PL and HP by comparing equations (29) with (6), (31) with (7), (32) with (8) and (33) with (9). Roughly speaking, the vector $\mathbf{n}$ in PL plays the role of $\mathbf{m}$ in HP, and $\mathbf{v}$ plays the role of $\rho$. We will exploit this fact to achieve ULI operation.

*3.1.3 Pin-hole back-projection*

A segment $\mathbf{l}$ detected in an image $\mathcal{I}$ uniquely determines the plane $\pi^{\mathsf{C}}$ containing the 3D line and the optical center (Fig. 3(b)). The plane's normal in camera frame, $\mathbf{n}^{\mathsf{C}}$, constitutes the measured part; it is obtained by simply inverting (33),

$$\mathbf{n}^{\mathsf{C}} = \mathcal{K}^{-1}\cdot\mathbf{l}\,. \tag{34}$$

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

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

with $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{n}^{\mathsf{C}}\}$ mutually orthogonal. Doing $\beta = (\beta_1, \beta_2) \in \mathbb{R}^2$ we get the matrix form

$$\mathbf{v}^{\mathsf{C}} = \mathbf{E}\cdot\beta\,, \tag{35}$$

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

The mutual orthogonality condition between $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{n}^{\mathsf{C}}\}$ gives us some freedom of choice for the base $\mathbf{E}$. For convenience, we arbitrarily build $\mathbf{E}$ so that $\|\beta\|$ is exactly inverse-distance and $\mathbf{e}_1$ is parallel to the image plane. This yields

$$\mathbf{e}_1 = \frac{\begin{bmatrix} n_2^{\mathsf{C}} & -n_1^{\mathsf{C}} & 0 \end{bmatrix}^\top}{\sqrt{(n_1^{\mathsf{C}})^2 + (n_2^{\mathsf{C}})^2}}\cdot\|\mathbf{n}^{\mathsf{C}}\| \quad \text{and} \quad \mathbf{e}_2 = \frac{\mathbf{n}^{\mathsf{C}}\times\mathbf{e}_1}{\|\mathbf{n}^{\mathsf{C}}\|}\,. \tag{36}$$

With this base choice the vector $\beta$ admits the following geometrical interpretation:

- $\beta = (\beta_1, 0)$ 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 the 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 $\mathsf{C}$ is given by $d = 1/\|\beta\|$.

Fig. 4 shows some examples of parameters $\beta$ and their corresponding lines in the representation plane $\pi^{\mathsf{C}}$.

*Remark 6 (Role of $\beta$)* The planar $\beta$-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 $\beta_1$ with respect to $\beta_2$, and it easily covers the full circumference. The value $\|\beta\|$ is the inverse of the Euclidean distance from the line to the origin. When assigning a prior *pdf* to $\beta$ at initialization time (see Fig. 10 in Section 4 on initializing the *pdf* of $\beta$), this will be properly mapped to the 3D space as a planar *pdf* on the plane $\pi^{\mathsf{C}}$. The support of high probability of this *pdf* covers from a specified minimal distance to infinity.

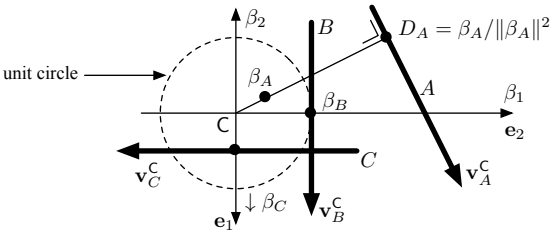



**Fig. 4** Different lines in the representation plane $\pi^{\mathsf{C}}$, defined by the base $\{\mathbf{e}_1, \mathbf{e}_2\}$ in camera frame $\mathsf{C}$, as a function of $\beta$. The direction $\mathbf{e}_1$ is parallel to the image plane. Given $\beta$, obtain the point $D = \beta/\|\beta\|^2$ and pass a line over it in the direction orthogonal to $\beta$. Three examples: first, the line $A$ is defined by $\beta_A = (0.4, 0.2)$; its closest point to $\mathsf{C}$ is $D_A = (2, 1)$, at a distance $1/\|\beta_A\| = \|D_A\| = \sqrt{5}$; it has direction $\mathbf{v}_A^{\mathsf{C}} = (0.4\,\mathbf{e}_1, 0.2\,\mathbf{e}_2)$. Second, the line $B$ is parallel to the image plane, at a distance of $1/\|\beta_B\| = \|D_B\| = 1$ from the optical center $\mathsf{C}$. And third, the line $C$ is orthogonal to the detected segment in the image (the image plane is not shown in this figure, please refer to Fig. 3(b)). Notice that the line $\mathcal{L}_C$ is generally not orthogonal to the image plane, because the plane $\pi^{\mathsf{C}}$ is generally not orthogonal to it.

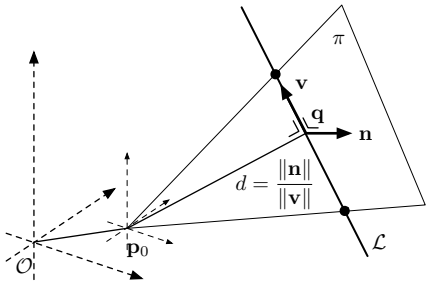



**Fig. 5** Anchored Plücker line (APL).

Summarizing, back-projection and transformation is performed by composing (29), (34) and (35), yielding

$$\mathcal{L}_{PL} = \mathcal{H}\begin{bmatrix}\mathbf{n}^{\mathsf{C}}\\ \mathbf{v}^{\mathsf{C}}\end{bmatrix} = \mathcal{H}\begin{bmatrix}\mathcal{K}^{-1}\mathbf{l}\\ \mathbf{E}\beta\end{bmatrix} = \begin{bmatrix}\mathsf{R}\mathcal{K}^{-1}\mathbf{l} + \mathsf{T}\times\mathsf{R}\mathbf{E}\beta\\ \mathsf{R}\mathbf{E}\beta\end{bmatrix}\,, \tag{37}$$

where $\beta$ must be provided as prior.

## 3.2 Anchored Plücker lines (APL)

As we did with points, we add an anchor to the Plücker parametrization to improve linearity. The *anchored Plücker line* (APL, introduced here for the first time, Fig. 5) is then

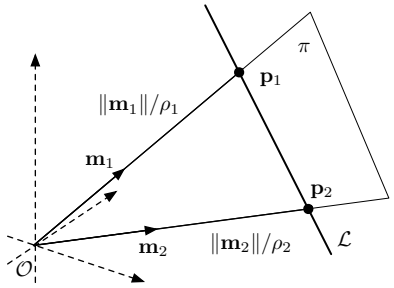



**Fig. 6** Homogeneous-points line (HPL) defined with two HP that support it.

the 9-vector:

$$\mathcal{L}_{APL} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{n} \\ \mathbf{v} \end{bmatrix} \in \mathbb{R}^9 \,. \tag{38}$$

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

$$\mathbf{l} = \mathcal{K} \cdot \mathsf{R}^\top \cdot (\mathbf{n} - (\mathsf{T} - \mathbf{p}_0) \times \mathbf{v}) \in \mathbb{P}^2 \,, \tag{39}$$

in which we 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.

Back-projection and transformation resume to

$$\mathcal{L}_{APL} = \begin{bmatrix} \mathsf{T} \\ \mathsf{R}\mathcal{K}^{-1}\mathbf{l} \\ \mathsf{R}\mathbf{E}\beta \end{bmatrix} \,, \tag{40}$$

where $\beta$ must be provided as prior.

### 3.3 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 2 to build lines, in the hope that this will preserve most of the properties of the formers.

A *homogeneous-points line* (HPL, introduced here for the first time, Fig. 6) is coded by two HP that support it:

$$\mathcal{L}_{HPL} = \begin{bmatrix} \mathbf{m}_1 \\ \rho_1 \\ \mathbf{m}_2 \\ \rho_2 \end{bmatrix} \in \mathbb{R}^8 \,. \tag{41}$$

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

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

This expression (which is obviously equal to HP's (8)) 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,

$$\mathbf{l} = \underline{\mathbf{u}}_1 \times \underline{\mathbf{u}}_2 \,. \tag{42}$$

This yields after a few arrangements[3]

$$\mathbf{l} = \mathcal{K}\mathsf{R}^\top \Big( (\mathbf{m}_1 \times \mathbf{m}_2) - \mathsf{T} \times (\rho_1 \mathbf{m}_2 - \rho_2 \mathbf{m}_1) \Big) \,. \tag{43}$$

This last expression is important in the sense that it allows us to observe the parallelisms between parametrizations. Comparing HPL (43) against PL (32), and remembering equations (27–28) defining the Plücker sub-vectors, we observe that:

- The product $\mathbf{m}_1 \times \mathbf{m}_2$ is a vector orthogonal to the plane $\pi$, 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 (32) and (43) coincide (using (27–28) this coincidence can be easily proved to hold exactly).

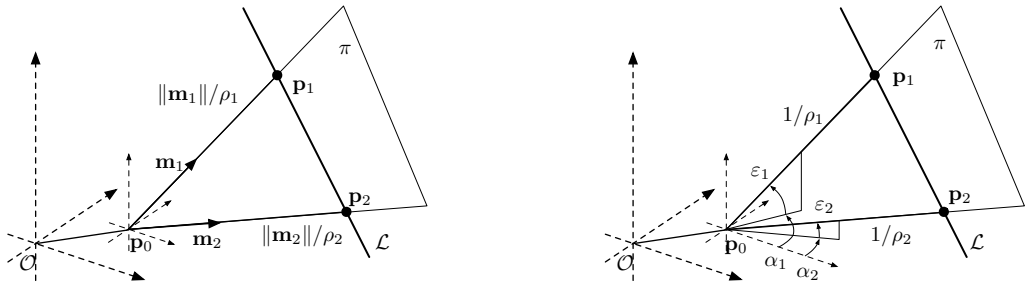
Obtaining the expression for back-projection and transformation should be trivial after the one used for HP. See Table 1 in page 13 for details.

### 3.4 Anchored homogeneous-points lines (AHPL)

The *anchored homogeneous-points line* (AHPL, used by Smith et al (2006) with delayed initialization, and introduced here for the first time implementing ULI, Fig. 7(a)) can be built either by adding an anchor to the HPL or by joining two AHP with a shared anchor:

$$\mathcal{L}_{AHPL} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{m}_1 \\ \rho_1 \\ \mathbf{m}_2 \\ \rho_2 \end{bmatrix} \in \mathbb{R}^{11} \,. \tag{44}$$

[3] To prove (43) we use the distributive property of the cross-product, the identity $(\mathbf{M}\mathbf{a}) \times (\mathbf{M}\mathbf{b}) = \det(\mathbf{M})\mathbf{M}^{-\top}(\mathbf{a} \times \mathbf{b})$, 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 therefore it remains equivalent under proportionality transforms.




(a) Anchored homogeneous-points line (AHPL)

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

**Fig. 7** Anchored point-supported lines. The lines are defined by two support points like the ones in Section 2. The anchor is common to both points.

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 line, $\mathbf{l} = \underline{\mathbf{u}}_1 \times \underline{\mathbf{u}}_2$. As before, this can be rearranged as

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

See Table 1 in page 13 for the back-projection and transformation equation.

### 3.5 Anchored modified-polar-points lines (AMPPL)

The *anchored modified-polar-points line* (AMPPL, introduced here for the first time, Fig. 7(b)) is coded by two AMPP that support it, which share a common anchor:

$$\mathcal{L}_{AMPPL} = \begin{bmatrix} \mathbf{p}_0 \\ (\varepsilon_1, \alpha_1) \\ \rho_1 \\ (\varepsilon_2, \alpha_2) \\ \rho_2 \end{bmatrix} \in \mathbb{R}^9 \ . \quad (46)$$

Transformation and projection resume to

$$\mathbf{l} = \mathcal{K}\mathsf{R}^\top \Big( (\mathbf{m}_1^* \times \mathbf{m}_2^*) - (\mathsf{T} - \mathbf{p}_0) \times (\rho_1 \mathbf{m}_2^* - \rho_2 \mathbf{m}_1^*) \Big) \ , \quad (47)$$

where we used the shortcut $\mathbf{m}_i^* \triangleq \mathbf{m}^*(\varepsilon_i, \alpha_i)$, which corresponds to the trigonometric transform (18).

See Table 1 in page 13 for the back-projection and transformation equation.

### 3.6 Final comment - points and lines

We summarize in Table 1 all points and lines parametrizations with their main manipulation expressions. On completion of their descriptions we have seen many parallelisms that should help building a coherent picture of a number of parametrizations suited for undelayed initialization in monocular EKF-SLAM. These relations are represented in Fig. 8. We have seen anchored and unanchored representations. We have seen the surprising similarities between homogeneous 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 for further justification.

## 4 Landmark initialization and updates

Undelayed landmark initialization with partial measurements mimics the algorithm for full measurements and incorporates the unmeasured magnitudes as Gaussian priors. We first detail the way we express physical measurements on the image plane, and the way to define the unmeasured priors. We finally proceed with details on the initialization and updating procedures related to the EKF machinery. For the initialization and updates of the segments endpoints, out of the Kalman filter, please refer to App. A. (For details on camera motion models, refer to *e.g.* (Davison, 2003, for constant velocity), (Solà, 2007, for odometry) or (Piniés et al, 2007, for inertial aiding).)

### 4.1 2D measurements in the image plane

The previous discourse assumed homogeneous parametrizations of points and lines in the projective image plane $\mathbb{P}^2$. We detail here how to obtain them from real point and line measurements defined in the Euclidean pixels image. Unfortunately, going from homogeneous space to Euclidean can only be made at the price of some linearity loss. We decoupled the two stages of projection to the homogeneous plane and transformation to Euclidean because only projection depends on the landmark parametrization. Transformation to Euclidean only depends on the generic type of landmark, that is, if it is a point or a line.

**Table 1** Summary of landmark parametrizations with their main manipulations

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

*Points:* A 2D point is measured as two Cartesian coordinates in pixel space, and modeled as a Gaussian variable. Please note that the numeric value of the measurement corresponds to the mean value $\bar{\mathbf{u}}$ of the distribution:

$$\mathbf{u} = \begin{bmatrix}u\\ v\end{bmatrix} \sim \mathcal{N}\{\bar{\mathbf{u}}, \mathbf{U}\}\ . \tag{48}$$

Its homogeneous counterpart is built with

$$\underline{\mathbf{u}} = \begin{bmatrix}\mathbf{u}\\ 1\end{bmatrix} \sim \mathcal{N}\{\underline{\bar{\mathbf{u}}}, \underline{\mathbf{U}}\} = \mathcal{N}\left\{\begin{bmatrix}\bar{\mathbf{u}}\\ 1\end{bmatrix}, \begin{bmatrix}\mathbf{U} & \mathbf{0}\\ \mathbf{0} & 0\end{bmatrix}\right\}\ . \tag{49}$$

*Lines:* A bounded 2D segment is measured as a 4-vector stacking its two endpoints:

$$\mathbf{s} = \begin{bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\end{bmatrix} \sim \mathcal{N}\{\bar{\mathbf{s}}, \mathbf{S}\} = \mathcal{N}\left\{\begin{bmatrix}\bar{\mathbf{u}}_1\\ \bar{\mathbf{u}}_2\end{bmatrix}, \begin{bmatrix}\mathbf{U} & \mathbf{0}\\ \mathbf{0} & \mathbf{U}\end{bmatrix}\right\}\ . \tag{50}$$

The segments homogeneous endpoints $\underline{\mathbf{u}}_i$, used for initialization of point-supported lines, are built like the regular points (Eq. (49)). The homogeneous line, used for initialization of Plücker lines, is built with (42), yielding a Gaussian *pdf* $\mathcal{N}\{\bar{\mathbf{l}}, \mathbf{L}\}$ with

$$\bar{\mathbf{l}} = \underline{\bar{\mathbf{u}}}_1 \times \underline{\bar{\mathbf{u}}}_2 \tag{51}$$

$$\mathbf{L} = [\underline{\bar{\mathbf{u}}}_1]_\times\, \underline{\mathbf{U}}\, [\underline{\bar{\mathbf{u}}}_1]_\times^\top + [\underline{\bar{\mathbf{u}}}_2]_\times\, \underline{\mathbf{U}}\, [\underline{\bar{\mathbf{u}}}_2]_\times^\top\ . \tag{52}$$

### 4.2 Defining the unmeasured Gaussian priors

Two basic rules apply to the definition of the prior, be it $\rho^{\mathsf{C}}$ for points or $\beta^{\mathsf{C}}$ for Plücker lines: the origin must be well inside the $2\sigma$ support of the *pdf*, and the minimum considered distance $d_{min}$ must (approximately) match the upper $2\sigma$ bound. For points and point-supported lines, this resumes to (see Fig. 9)

$$\bar{\rho} - n\sigma_\rho = 0, \qquad 0 \le n < 2 \tag{53}$$

$$\bar{\rho} + 2\sigma_\rho = 1/d_{min}\ . \tag{54}$$

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

$$\bar{\rho} = 1/3d_{min}, \quad \sigma_\rho = 1/3d_{min}\ . \tag{55}$$

For point-supported lines HPL, AHPL and AMPPL, we just need to stack two stochastically independent $\rho^{\mathsf{C}}$ priors, *i.e.*, if we note such prior with $\mathbf{t}^{\mathsf{C}} \sim \mathcal{N}\{\bar{\mathbf{t}}; \mathbf{T}\}$, we have

$$\bar{\mathbf{t}} = \begin{bmatrix}\bar{\rho}\\ \bar{\rho}\end{bmatrix}\ , \quad \mathbf{T} = \begin{bmatrix}\sigma_\rho^2 & 0\\ 0 & \sigma_\rho^2\end{bmatrix}. \tag{56}$$

Defining the 2D Gaussian prior $\beta \sim \mathcal{N}\{\bar{\beta}; \mathbf{B}\}$ for Plücker lines PL and APL is a bit trickier, as it is difficult to

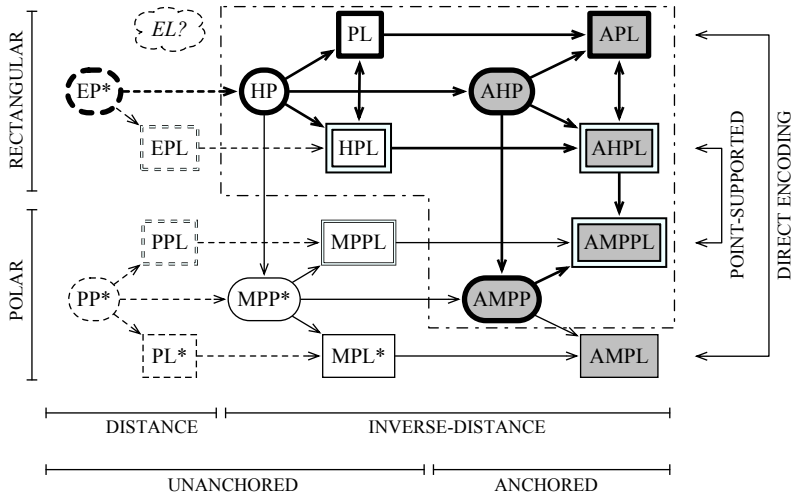



**Fig. 8** Links between all proposed parametrizations and more. Round boxes are points; square boxes are lines. Single-stroke square boxes are directly-coded lines. Double-stroke square boxes are point-supported lines. Gray boxes are anchored parametrizations. Arrows indicate the links that we established 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 lines: 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 lines: 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). Elements in the first column (with dashed boxes) do not benefit from the inverse-distance property and are not suited for undelayed initialization. Minimal parametrizations are marked with an asterisk.

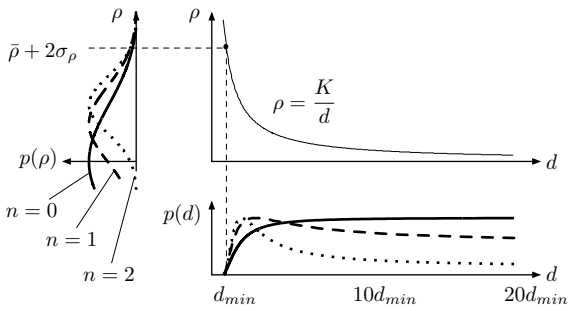
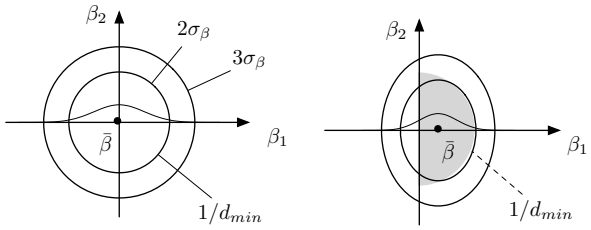



**Fig. 9** Inverse-distance *pdf* for HP, AHP, AMPP, HPL, AHPL and AMPPL. A Gaussian $p(\rho) = \mathcal{N}(\rho - \bar{\rho}, \sigma_\rho^2)$ is defined in inverse-distance (vertical axes). We have ample choice: in one extreme (solid, $n = 0$) we may define it so that $\bar{\rho} = 0$; the other extreme (dotted, $n = 2$) takes $\bar{\rho} - 2\sigma_\rho = 0$. 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 $\rho$ is exactly equal to inverse-distance.




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

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

**Fig. 10** Defining a prior $\beta \sim \mathcal{N}\{\bar{\beta}; \mathbf{B}\}$ for PL and APL. (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 $1/d_{min}$ is at $2\sigma$. For reference, a Gaussian shape is superimposed on the horizontal axis to evaluate the probability values at $2\sigma$ and $3\sigma$. (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 shadowed) 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 $\sigma_{\beta_2} = 1/2d_{min}$.

express the conditions as straightforward equations like (53) and (54). We prefer to refer the reader directly to the explanations of Fig. 10. For all the implementations presented in this paper we use the solution in Fig. 10(b),

$$\bar{\beta} = \begin{bmatrix} 1/3d_{min} \\ 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} (1/3d_{min})^2 & 0 \\ 0 & (1/2d_{min})^2 \end{bmatrix}. \tag{57}$$

### 4.3 Landmark initialization

The ULI algorithm valid for all parametrizations is detailed below.

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

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

with $\mathsf{C} = (\mathsf{T}, \mathsf{Q})$ the camera frame and $\mathsf{M} = (\mathcal{L}_1, \ldots, \mathcal{L}_N)$ the set of mapped landmarks (points, lines or a mixture of them).
2. Identify the measurement $\mathbf{z} \sim \mathcal{N}\{\bar{\mathbf{z}}, \mathbf{R}\}$ (Section 4.1; $\mathbf{z}$ is either $\mathbf{u}$ or $\mathbf{s}$).
3. Define a Gaussian prior $\pi \sim \mathcal{N}\{\bar{\pi}; \mathbf{\Pi}\}$ for the unmeasured DOFs (Section 4.2; $\pi$ is either $\rho^{\mathsf{C}}$, $\mathbf{t}^{\mathsf{C}}$ or $\beta^{\mathsf{C}}$).
4. Back-project the Gaussian measurement; get landmark mean and Jacobians

$$\bar{\mathcal{L}} = g(\bar{\mathsf{C}}, \bar{\mathbf{z}}, \bar{\pi})$$
$$\mathbf{G}_{\mathsf{C}} = \left.\frac{dg}{d\mathsf{C}}\right|_{\bar{\mathsf{C}},\bar{\mathbf{z}},\bar{\pi}}, \; \mathbf{G}_{\mathbf{z}} = \left.\frac{dg}{d\mathbf{z}}\right|_{\bar{\mathsf{C}},\bar{\mathbf{z}},\bar{\pi}}, \; \mathbf{G}_{\pi} = \left.\frac{dg}{d\pi}\right|_{\bar{\mathsf{C}},\bar{\mathbf{z}},\bar{\pi}}$$

with $g()$ the composition of the measurement-to-homogeneous transforms (Section 4.1) with the back-projection and transformation function (functions $\underline{g}()$ in Table 1).
5. Compute landmark co- and cross-variances

$$\mathbf{P}_{\mathcal{L}\mathcal{L}} = \mathbf{G}_{\mathsf{C}}\mathbf{P}_{\mathsf{C}\mathsf{C}}\mathbf{G}_{\mathsf{C}}^{\top} + \mathbf{G}_{\mathbf{z}}\mathbf{R}\mathbf{G}_{\mathbf{z}}^{\top} + \mathbf{G}_{\pi}\mathbf{\Pi}\mathbf{G}_{\pi}^{\top}$$
$$\mathbf{P}_{\mathcal{L}\mathbf{x}} = \mathbf{G}_{\mathsf{C}}\mathbf{P}_{\mathsf{C}\mathbf{x}} = \mathbf{G}_{\mathsf{C}}[\mathbf{P}_{\mathsf{C}\mathsf{C}} \; \mathbf{P}_{\mathsf{C}\mathsf{M}}] \; .$$

6. Augment the SLAM map

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

## 4.4 Landmark updates

### *4.4.1 Point updates*

The observation function $h()$ is the composition of the ones in Table 1 with the homogeneous-to-Euclidean transform $h2e()$,

$$\mathbf{z} = h2e(\underline{\mathbf{u}}) = \begin{bmatrix} u_1/u_3 \\ u_2/u_3 \end{bmatrix} \in \mathbb{R}^2 \; . \tag{58}$$

The complete observation function is therefore $h(\mathbf{x}) = h2e(\underline{h}(\mathbf{x}))$. Point updates follow the standard EKF-SLAM formulation,

$$\text{Innovation mean:} \quad \mathbf{y} = \mathbf{z} - h(\bar{\mathbf{x}}) \tag{59}$$
$$\text{Innovation covariance:} \quad \mathbf{Y} = \mathbf{R} + \mathbf{H}\cdot\mathbf{P}\cdot\mathbf{H}^{\top} \tag{60}$$
$$\text{Kalman gain:} \quad \mathbf{K} = \mathbf{P}\cdot\mathbf{H}^{\top}\cdot\mathbf{Y}^{-1} \tag{61}$$
$$\text{State update:} \quad \bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}} + \mathbf{K}\cdot\mathbf{y} \tag{62}$$
$$\text{Covariance update:} \quad \mathbf{P} \leftarrow \mathbf{P} - \mathbf{K}\cdot\mathbf{H}\cdot\mathbf{P} \; , \tag{63}$$

with $\mathbf{R} = \mathbf{U}$ the measurement noise covariance (see (48)), and the Jacobian $\mathbf{H} = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\bar{\mathbf{x}}}$.

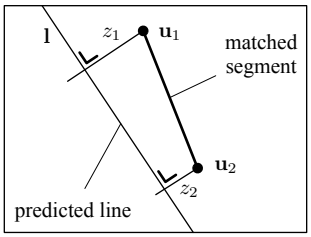



**Fig. 11** Plücker line observation update. Direct measurement of the two signed orthogonal distances from the detected endpoints to the expected (or predicted) line.

### *4.4.2 Line updates*

It is convenient to represent the matched segment by its two endpoints, $\mathbf{s} = (\mathbf{u}_1, \mathbf{u}_2) \in \mathbb{R}^4$. Due to the aperture problem, only the measurement components that are orthogonal to the expected line projection can be used for correction. Therefore, a proper measurement space that accounts for this orthogonality and these distances needs to be defined.

We define the measurement space as the set of 2-vectors containing the signed orthogonal distances from the detected endpoints $\mathbf{u}_i$ to a line $\mathbf{l}$. This leads to the measurement function

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \mathbf{l}^{\top}\cdot\underline{\mathbf{u}}_1/\sqrt{l_1^2 + l_2^2} \\ \mathbf{l}^{\top}\cdot\underline{\mathbf{u}}_2/\sqrt{l_1^2 + l_2^2} \end{bmatrix} \in \mathbb{R}^2 \; , \tag{64}$$

which is in pixels units. If we name this function $h_1(\mathbf{l}, \mathbf{s})$, the full observation function is its composition with the projection functions $\underline{h}()$ in Table 1,

$$\mathbf{z} = h(\mathbf{x}, \mathbf{s}) = h_1(\underline{h}(\mathbf{x}), \mathbf{s}) \; . \tag{65}$$

The EKF innovation $\mathbf{y}$ is defined as the difference between the actual measurement and the expectation,

$$\mathbf{y} = \mathbf{z} - h(\bar{\mathbf{x}}, \mathbf{s}) \; .$$

For the measurement $\mathbf{z}$, this corresponds to the distances from the detected endpoints to the detected line $\mathbf{l} = \mathbf{u}_1 \times \mathbf{u}_2$. Because this line $\mathbf{l}$ is precisely defined by the two endpoints, the measured vector is zero by definition, and we just need to consider a covariance $\mathbf{R} = \mathbf{U} \in \mathbb{R}^2$ (see (48)) representing the pixel noise in just two of the four dimensions.[4] The expectation corresponds to the distances (64) to the expected line $\bar{\mathbf{l}} = \underline{h}(\bar{\mathsf{C}}, \bar{\mathbf{x}})$ (Fig. 11). This yields an innovation $\mathbf{y} = 0 - h(\bar{\mathbf{x}}, \mathbf{s})$ with covariance $\mathbf{Y} = \mathbf{R} + \mathbf{H}\cdot\mathbf{P}\cdot\mathbf{H}^{\top}$. The rest of the EKF update is as before.

[4] The expression $\mathbf{R} = \mathbf{U}$ is only valid if the pixel noise is defined isotropic via $\mathbf{U} = \sigma^2\mathbf{I}_2$, which is most generally the case. Otherwise we need to compute $\mathbf{R} = \mathbf{H}_{\mathbf{s}}\mathbf{S}\mathbf{H}_{\mathbf{s}}^{\top}$ with $\mathbf{H}_{\mathbf{s}}$ the Jacobian of (64) with respect to the measured segment $\mathbf{s}$. In fact, $\mathbf{H}_{\mathbf{s}}$ is such that if $\mathbf{S} = \mathrm{diag}(\mathbf{U}, \mathbf{U}) = \sigma^2\mathbf{I}_4$ then $\mathbf{R} = \mathbf{U} = \sigma^2\mathbf{I}_2$.

*4.4.3 A first comment about the Plücker constraint*

When dealing with Plücker lines PL or APL we do not apply any kind of correction to enforce the Plücker constraint. We ensured its satisfaction during landmark initialization, with the specification of the initial covariance in the $\beta$-plane, Section 3.1.3, and its validity at any later time is only approximately guaranteed through cross-correlations. Although this is of course not the optimal way to proceed, we decided to leave the method as parallel as possible with the others presented here, so that we can impute the differences in performance exclusively to landmark parametrization – thus not to algorithmic aspects. Refer to Section 8.2 for further discussion.

### 4.5 Landmark re-parametrization

Landmark over-parametrization, which we have defended for EKF performance so far, is expensive and should only be used when justified. Landmarks should be reparametrized to their minimal forms after convergence, that is, when the observation functions of these minimal forms (the *destination* forms) are judged linear enough.

For points, the natural choice is to reparametrize to EP. The reparametrization is triggered by the linearity test described in Civera et al (2008), which is very cheap to compute and can be easily adapted to HP, AHP and AMPP.

For lines, and because of the need of endpoints, it may be convenient to choose a non-minimal two-points representation $\mathcal{L} = (\mathbf{p}_1, \mathbf{p}_2)$ (EPL, see Fig. 8), with 6 parameters. In this case we can use the test for points in Civera et al (2008), which must hold for both support points. We can also use any of the minimal representations, which are of size 4 (see also Fig. 8). Tests for these other line representations might be defined from the linearity indices described in the next section, although these indices are not conceived for speed. A compromise that would probably lead to satisfactory operation is to use the test for EPL, which is simple and does indicate that the line has already converged, and then reparametrize to any other form of our convenience. We have not explored these last possibilities.

## 5 Linearity and performance evaluation tools

We present here the analytical and statistical tools used in this article to evaluate the performances of all parametrizations.

### 5.1 Analytical measure of linearity

The EKF requires high degrees of linearity in the measurement and dynamic model equations. Defining an analytic measure of linearity allowing us to compare the degrees of linearity of the observation functions for different parametrizations is therefore of clear importance. In (Civera et al, 2008), an analytic linearity index is proposed, based on the variation in the first derivative of the function inside the $95\%$ probability interval of the most uncertain state variable: the inverse-distance parameter. This measure is restricted, thanks to the particular symmetries of the problem, to just 1 DOF, and it is difficult to generalize to our amalgam of parametrizations. Very related to this work, the trace of the Hessian of the measurement model is proposed in (Eade and Drummond, 2007) as a measure 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.

In EKF, linearity must always be evaluated with respect to the extension of the probability concentration region of the input variable, which is specified by the covariances matrix. We introduce an analytical linearity index for multiple-input/multiple-output (MIMO) functions which accounts for this probability region. As a desirable additional quality, the proposed index is defined in the measurement space and therefore allows us to compare parametrizations having different state sizes, and even to compare the degree of linearity of points against lines. As in (Eade and Drummond, 2007), it involves the computation of the Hessian which concentrates the local degree of non-linearity of a function.

We are interested in the complete observation functions $\mathbf{z} = h(\mathbf{x})$, *i.e.*, the composition of the transformation and pin-hole projection functions, $\underline{h}()$ in Table 1, with the appropriate measurement functions, (58) for points and (64) for lines. For concision, we define the state $\mathbf{x} = (\mathsf{C}, \mathcal{L})$, *i.e.*, only the pair camera-landmark under consideration, whose estimate in the map is a Gaussian $\mathcal{N}\{\bar{\mathbf{x}}, \mathbf{P}\}$. We denote the measurement and state dimensions with $m = \dim(\mathbf{z})$ and $n = \dim(\mathbf{x})$. In our case we have a fixed $m = 2$, and a variable $10 \leq n \leq 18$ depending on the selected parametrization.

Our linearity index is based on the error in the filter innovation $\mathbf{y} = \mathbf{z} - h(\bar{\mathbf{x}})$ due to linearization. For any measurement $\mathbf{z}$, this error corresponds to the propagation error through $h()$ of the state's mean (Fig. 12), given by

$$\epsilon \triangleq \mathbb{E}[h(\mathbf{x})] - h(\mathbb{E}[\mathbf{x}]) \in \mathbb{R}^m \,, \tag{66}$$

where $\mathbb{E}[\bullet]$ is the expectation operator, *i.e.*, $\bar{\mathbf{x}} \equiv \mathbb{E}[\mathbf{x}]$. This error is a magnitude expressed in the measurement space. To obtain a computable approximation of $\epsilon$ we use the Taylor-Young expansion for multi-variate functions applied to each

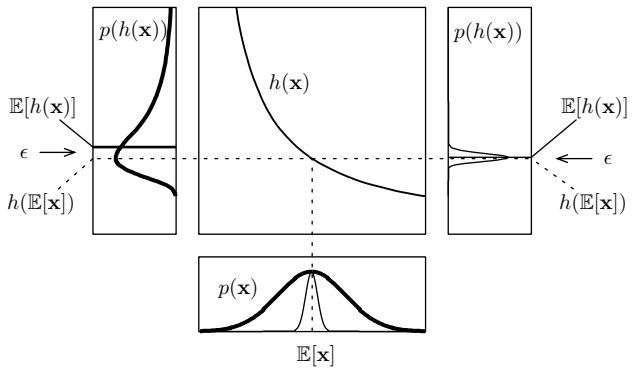



**Fig. 12** Propagation error of the Gaussian mean due to non-linearity. For a given non-linear function $h()$, the propagation error $\epsilon = \mathbb{E}[h(\mathbf{x})] - h(\mathbb{E}[\mathbf{x}])$ is large for Gaussians with large variance (*thick line, left*) and unnoticeable for narrow Gaussians (*thin line, right*).

component $i$ of $h(\mathbf{x})$,

$$h_i(\mathbf{x}) = h_i(\bar{\mathbf{x}}) + \sum_{j=1}^{n} J_{ij} \Delta x_j + \sum_{j=1}^{n}\sum_{k=1}^{n} \frac{1}{2} H_{ijk} \Delta x_j \Delta x_k + o(\|\Delta \mathbf{x}\|^2) \,, \quad (67)$$

where $\Delta\mathbf{x} \triangleq \mathbf{x} - \bar{\mathbf{x}}$, $\Delta x_j$ are the components of $\Delta\mathbf{x}$, $h_i$ are the components of $h()$, $J_{ij} \triangleq \frac{\partial h_i}{\partial x_j}(\bar{\mathbf{x}})$ are the components of the $m \times n$ Jacobian matrix of first derivatives, and $H_{ijk} \triangleq \frac{\partial^2 h_i}{\partial x_j x_k}(\bar{\mathbf{x}})$ are the components of the $m \times n \times n$ Hessian tensor of second derivatives. Inserting (67) into (66) and ignoring $o(\|\Delta\mathbf{x}\|^2)$ yields the first-order approximation

$$\begin{aligned} \epsilon_i &\approx \mathbb{E}\left[ h_i(\bar{\mathbf{x}}) + \sum_{j=1}^{n} J_{ij}\Delta x_j + \sum_{j=1}^{n}\sum_{k=1}^{n} \frac{1}{2} H_{ijk} \Delta x_j \Delta x_k \right] \\ &\quad - h_i(\bar{\mathbf{x}}) \\ &= \sum_{j=1}^{n} J_{ij} \mathbb{E}[\Delta x_j] + \sum_{j=1}^{n}\sum_{k=1}^{n} \frac{1}{2} H_{ijk} \mathbb{E}[\Delta x_j \Delta x_k] \,. \end{aligned} \quad (68)$$

Having $\mathbb{E}[\Delta\mathbf{x}] = 0$ and knowing that the covariance of $\mathbf{x}$ is a $n \times n$ matrix given by $\mathbf{P} \triangleq \mathbb{E}[\Delta\mathbf{x}\Delta\mathbf{x}^\top]$, with components $P_{jk} \triangleq \mathbb{E}[\Delta x_j \Delta x_k]$, we obtain

$$\epsilon = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_m \end{bmatrix} \,, \quad \epsilon_i \approx \frac{1}{2} \sum_{j=1}^{n}\sum_{k=1}^{n} H_{ijk} P_{jk} \,. \quad (69)$$

Finally, taking the norm yields the scalar index

$$L = \|\epsilon\|_2 \in \mathbb{R}^+ \,. \quad (70)$$

The $m$-dimensional vector $\epsilon$ belongs to the measurement space and is therefore expressed, in our case, in pixel units. It can be interpreted as the bias introduced in the EKF innovation by the non-linearity of the measurement equation. Its norm, the proposed index $L$, also in pixels, 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. 13 the fitness of this index for a 2-input, 1-output function. Observe that the function is always 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.

## 5.2 Monte Carlo RMS errors and consistency evaluation

For practical reasons and because the full SLAM state vector is of varying size, we restrict the error and consistency analyses to the state variables representing the robot (or camera) pose $\mathsf{C}$, knowing that consistent localization indicates consistent mapping (Huang and Dissanayake, 2007). We systematically transform mean and covariances matrix of the pose to a minimal representation (the orientation is transformed to the Euler angles) to avoid singularities in the covariance.

### *5.2.1 RMSE evaluation*

We perform a number $N$ of Monte Carlo runs. At each time instant $k$, we evaluate the root mean square error (RMSE) of each component $i$ of the camera pose,

$$\epsilon_{i,k} = \sqrt{\frac{1}{N}\sum_{j=1}^{N} (\mathsf{C}_{i,k} - \hat{\mathsf{C}}^j_{i,k})^2} \,, \quad (71)$$

where $\mathsf{C}_{i,k}$ is the $i$-th component ($x$, $y$, $z$, roll $\phi$, pitch $\theta$ and yaw $\psi$) of the true camera pose at time $k$, and $\hat{\mathsf{C}}^j_{i,k}$ is its EKF estimate's mean corresponding to the $j$-th among the $N$ Monte Carlo runs.

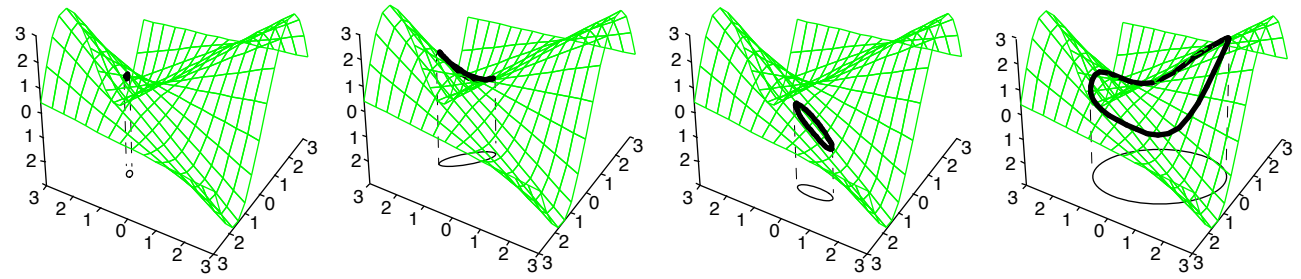
For visualization purposes, these errors are compared against the estimated error given by the filter. We take its average over all the Monte Carlo runs,

$$\bar{\sigma}_{i,k} = \frac{1}{N}\sum_{j=1}^{N} \sqrt{P^j_{ii,k}} \,, \quad (72)$$

where $P^j_{ii,k}$ is the $i$-th diagonal component of the estimated covariances matrix of the camera pose, for run $j$ and at time $k$. The RMSE plots in the Results section will show the true error $\epsilon_{i,k}$ against the 2-sigma bound given by $2\bar{\sigma}_{i,k}$.

### *5.2.2 Average NEES evaluation*

We use the average normalized estimation error squared (NEES) for evaluating consistency. We follow strictly (Bailey et al, 2006), which is in turn following (Bar-Shalom et al,



(a) $L = 0.0032$ (b) $L = 0.2269$ (c) $L = 0.0014$ (d) $L = 1.5263$

**Fig. 13** Linearity index $L$ of the MISO function $z = h(x, y) = x \cdot \sin(y)$ 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 $\mathcal{S} = \{(x, y, z) \,/\, z = h(x, y)\}$ (*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.

2001, pp. 234–235). After a number $N$ of Monte Carlo runs, the averaged NEES value is defined by

$$\eta_k \triangleq \frac{1}{N} \sum_{j=1}^{N} (\mathsf{C}_k - \hat{\mathsf{C}}_k^j)^\top \mathbf{P}_k^{j\,-1} (\mathsf{C}_k - \hat{\mathsf{C}}_k^j) \,, \tag{73}$$

where $\mathsf{C}_k$ is the true camera pose at time $k$ and $\{\hat{\mathsf{C}}_k^j, \mathbf{P}_k^j\}$ is its Gaussian estimate corresponding to the $j$-th among $N$ Monte Carlo runs. For 6 DOF and $N = 25$ runs, the upper and lower bounds of the double-sided 95% probability concentration region are given by:

$$\overline{\eta} = \chi^2_{(25\times 6)}(1 - 0.975)/25 = 7.432$$
$$\underline{\eta} = \chi^2_{(25\times 6)}(1 - 0.025)/25 = 4.719 \,.$$

If $\eta_k < \underline{\eta}$ for some significant amount of time (more than 2.5% of the time), the filter is conservative. If $\eta_k > \overline{\eta}$ (also by more that 2.5%), the filter is optimistic and therefore inconsistent.

## 6 Simulation results

### 6.1 Software and SLAM algorithm

We have made available the software used for simulations (Solà et al, 2009a). It consists in a 6 DOF EKF-SLAM system written in MATLAB®, with simulation and 3D graphics capabilities.

The algorithm is organized as an EKF-SLAM with active features search (see Davison et al, 2007, for the active search), 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. Data association errors are not simulated and therefore data association is perfect.

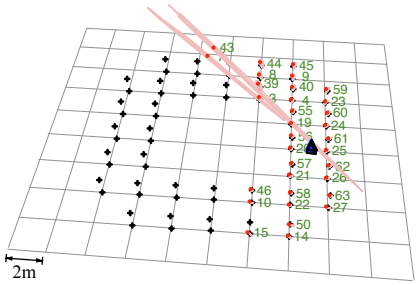



**Fig. 14** Simulated 3D environment for benchmarking point parametrizations.

### 6.2 Evaluation of point parametrizations

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.

#### *6.2.1 Simulated scenario*

We simulate a robot performing a circular trajectory in an area of $12\,\mathrm{m} \times 12\,\mathrm{m}$ populated with 72 landmarks forming a cloister (Fig. 14). 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. Three sets of parameters have been used for the tests (see Table 2). 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). Set 3 is like Set 2 but with a different inverse-distance prior.

**Table 2** Simulation parameters

| Concept | Set 1 | Set 2 | Set 3 |
|---|---|---|---|
| Img. size | $640\times480$ pix | | |
| Focal | 320 pix, $HFOV = 90°$ | | |
| Pix. noise | 1 pix | | |
| Pose step | $(8\text{cm}, 0.9°)$ | $(4\text{cm}, 0.45°)$ | |
| Lin. noise, $1\sigma$ | 0.5cm | 0.25cm | |
| Ang. noise, $1\sigma$ | $0.05°$ | $0.025°$ | |
| $\rho^{\mathsf{C}}$ prior | $(\bar{\rho}, \sigma_\rho) = (0.01, 0.5)\ \text{m}^{-1}$ | | $(1.0, 1.0)\ \text{m}^{-1}$ |

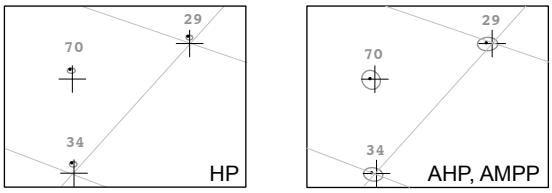



**Fig. 15** 3D view of some landmark $3\sigma$ estimates at the end of the first loop. Inconsistency comes mostly from covariance overestimation rather than mean errors, as can be seen by the too small ellipses in the HP case (*left*). See the accompanying video.

### *6.2.2 Visual evaluation*

We provide the accompanying video `points.mov` showing the three methods running in parallel under the conditions of Set 1.

The differences in behavior are not easily visible in the 3D movies, and we need to zoom in to appreciate incorrect operation (Fig. 15). We see that HP estimates have too small covariances, a clear sign of overconfidence, and therefore inconsistency. This over-estimation, which is in accordance with (Huang et al, 2008), is attributed exclusively to parametrization differences because the information provided to the filter for HP is exactly the same for all methods. Of the 25 HP runs, one diverged, and 35 landmarks had to be deleted due to inconsistent observations (22 of which during the divergent run).

We do not observe any significant difference between AHP and AMPP. No landmarks were declared inconsistent in any of the 25 runs of AHP and AMPP.

### *6.2.3 Linearity measures*

The linearity index in section 5.1 has been computed for each measured landmark and for the three parametrizations of interest (HP, AHP and AMPP), using the parameters of Set 1.

Fig. 16 shows the linearity indices of one particular landmark in order to illustrate the typical behavior. The chosen landmark corresponds to the first landmark initialized after the camera has completed one quarter of a turn (100 frames). This is to ensure that the camera uncertainty at initialization

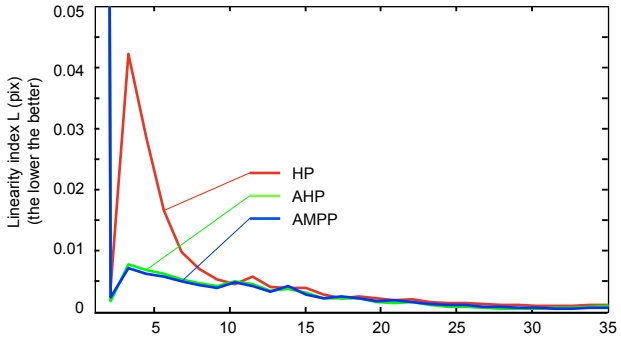



**Fig. 16** Linearity index for the three point parametrizations during the first 35 frames of a landmark's life.

time is not null, and therefore that the effect of anchoring can be observed. The index starts very high (bad linearity) due to the huge uncertainty region. It decays rapidly and reaches a minimum at about frame 25. After this point it stabilizes to very small values (high linearity).

The index of HP is clearly higher than those of AHP and AMPP, indicating a poorer linearity. The reason, as previously mentioned, is that setting an anchor propagates the camera uncertainty only from the anchor to the current location, while HP propagates a wider uncertainty with respect to a world reference frame.

It is worth remarking in this figure the low values of the proposed index, two or three orders of magnitude less than typical image noise or EKF innovations (which we recall have been set to 1 pixel in these simulations). Although it is true that linearization errors introduce bias in an EKF estimation, especially because they are of systematic origin, our experiments show that the value for this bias is small enough to guarantee a good behavior of the EKF filtering for local monocular SLAM. The same conclusion can be applied to line-based EKF-SLAM, as the quantitative results for the lines index are similar (see the linearity measures for lines in Fig. 21).

Another aspect that is worth remarking is that the linearity index refers to the source of the estimation error at a given moment. This error accumulates over time following two mechanisms. The first one is just linear integration. The second effect is the effect that heading errors at a given time have on the position at a later time, due to translation with inaccurate heading. These accumulated errors are visible as errors in the camera pose, as illustrated in the next section.

### *6.2.4 Error and consistency evaluation*

Error and consistency evaluations are based on the root mean square error (RMSE) of the camera pose, and the average normalized estimation error squared (NEES), both described in Section 5. We use $N = 25$ runs for each experiment, each run with a different seed for the random

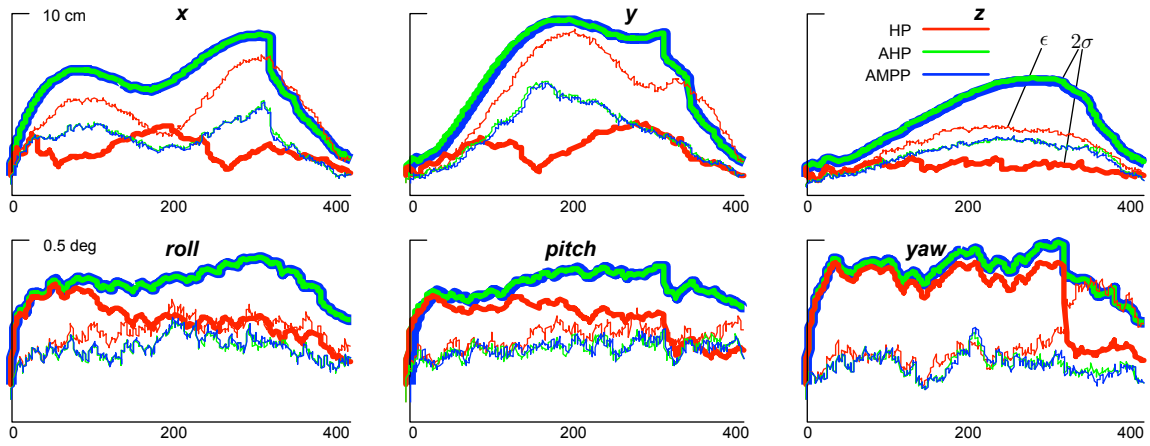



**Fig. 17** RMS errors $\epsilon$ of the three point parametrizations HP (red), AHP (green) and AMPP (blue), averaged over 25 runs. The $2\sigma$ estimated bounds are plotted in thicker line. AHP and AMPP have the largest estimated bound and the lowest error, leading to the best results. See Fig. 18 for the corresponding consistency plots.

generator. The random generator affects several aspects of the algorithm, namely the process noises and the measurement noises. In parallel, one simulated mobile camera without process noise, gathering noiseless images of the environment, is used to generate the "ideal" or "perfect" run against which the other "noisy" runs are compared for errors. This ideal run is often referred to in the SLAM literature as *ground truth*.

The RMSE and average NEES plots in Figs. 17 and 18 (please notice the logarithmic vertical scales in the NEES plots) confirm the results seen for the linearity indices. HP behaves poorly, and there is no significant difference between AHP and AMPP, except for a tiny but appreciable difference in favor of AMPP. Both AHP and AMPP behave 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.

It is now clearly visible that HP inconsistency comes mostly from covariance overestimation: in the RMSE plots there is a significant decay of the estimated sigma-values, while the error magnitude is indeed larger but to a smaller extent. As all methods process the same amount of information, it must be concluded that overestimation comes from the effect that linearization errors have over the Kalman gains.

We tuned the algorithms with the second set of parameters in order to improve the conditions for linearity: odometry steps and noise are cut in half, reducing the measurements innovation, 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. 19 show no significant improvement with respect to those of Set 1 (these 200 frames correspond to the

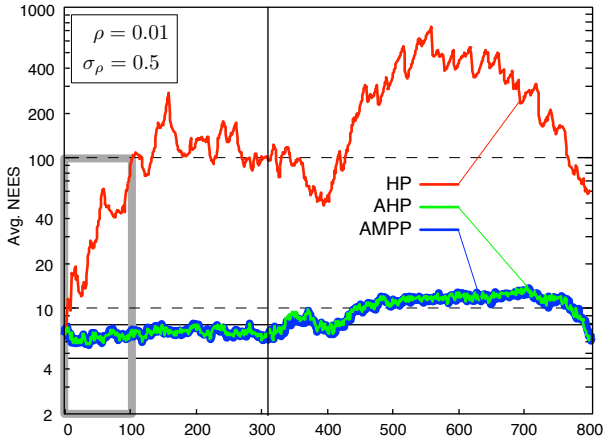



**Fig. 18** Consistency of HP (top plot), AHP (bottom) and AMPP (bottom). Average normalized estimation error squared (NEES) of the 6 DOF vehicle pose $[x, y, z, \phi, \theta, \psi]^\top$ over 25 runs for 800 frames (2 turns) and parameters of Set 1. The dotted horizontal band between abscissas $\underline{\eta} = 4.719$ and $\overline{\eta} = 7.432$ mark the $95\%$ consistency region: if the average NEES is greater than the upper limit $\overline{\eta}$ for more than $2,5\%$ of the time, the filter estimate is considered inconsistent. The vertical line marks the loop closure at frame 308. The framed area corresponds to the area covered by Fig. 19.

first 100 frames of Set 1, which have been boxed in Fig. 18): HP is not good, largely inconsistent, and AHP and AMPP are again the ones that behave consistently. Interestingly, AHP and AMPP have the same average NEES values as the ones observed in the previous test, showing an important robustness against varying operating conditions.

A third test consisted in selecting a different prior for the unmeasurable inverse-distance. The dashed plots in Fig. 19 show that AMPP and AHP are almost insensitive to 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

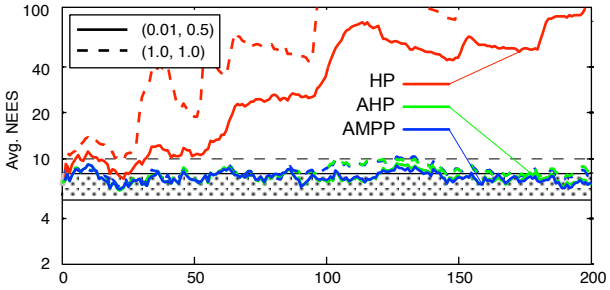
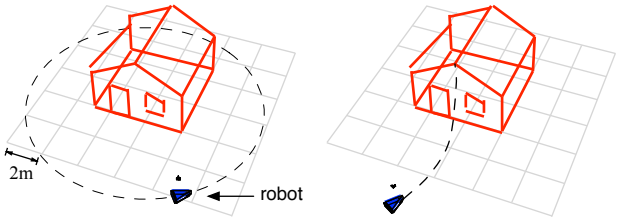



**Fig. 19** Consistency of HP, AHP and AMPP. Average NEES over 25 runs for 200 frames (1/4 turn) and 10 initializations in the first frame. *Solid:* parameters of Set 2 with prior $(\bar{\rho}, \sigma_\rho) = (0.01, 0.5)$. *Dashed:* parameters of Set 3 with an alternative prior $(\bar{\rho}, \sigma_\rho) = (1.0, 1.0)$.




(a) Circular trajectory. (b) Arc trajectory.

**Fig. 20** Simulated 3D environments for benchmarking the 5 line parametrizations. The robot's trajectory and the house being reconstructed are shown. (a) Circular trajectory, camera looking sideways to the house. (b) Arc trajectory, camera looking forward to the house.

initialized at (or close to) infinity ($\bar{\rho}^{\mathsf{C}} = 0.01\mathrm{m}^{-1}$) than at some close distance ($\bar{\rho}^{\mathsf{C}} = 1\mathrm{m}^{-1}$).

### 6.3 Evaluation of line parametrizations

We benchmark PL, APL, HPL, AHPL and AMPPL for linearity, RMS errors and average NEES consistency, in two different scenarios.

The first scenario (Fig. 20(a)) consists of a robot making a turn around a wireframe model of a house. Occlusions are not simulated and all the house's edges are visible. 400 frames are processed, and again only the 10 most informative segments are processed at each frame. The camera is looking sideways to the house and, the house being always visible, there is no loop closure. The simulation parameters are equivalent to the ones we used for points in Set 1.

We provide the accompanying video `lines.mov` showing the five systems running in parallel. At first sight all parametrizations seem to work correctly. As we did with points, we use the analytical and numerical tools to reveal the differences in performance between parametrizations.

The linearity indices are shown in Fig. 21. All indices follow essentially the same pattern as we saw for points. Moreover, their numerical values are similar for points and lines (compare Figs. 16 and 21), suggesting that the index can be used for comparing points against lines. Again, unanchored parametrizations are the ones showing the poorest linearity. APL is better than all unanchored ones but not good enough, probably because the Plücker constraint is only applied at initialization time and not enforced on subsequent updates (see Section 8 for a more detailed discussion).

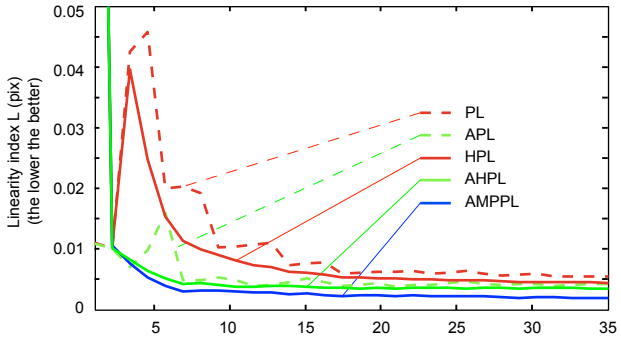



**Fig. 21** Linearity index for the five line parametrizations during the first 35 frames of a landmark's life, showing the superior linearity of anchored point-supported lines. Compare with Fig. 16.

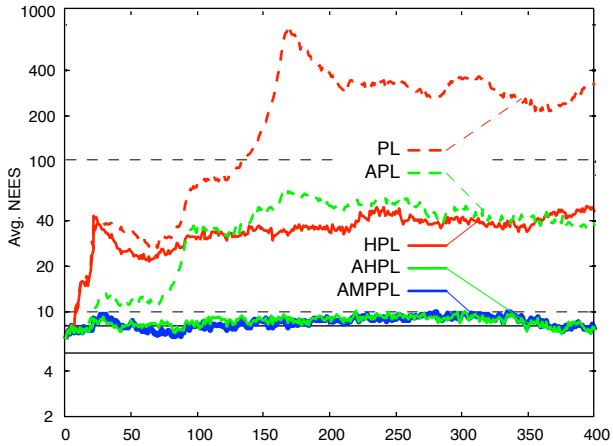



**Fig. 22** Consistency of PL, APL, HPL, AHPL and AMPPL. Average NEES over 25 runs for 400 frames (one turn around the house). Plücker-based lines (PL and APL) do not behave consistently, even when anchored. Lines based on homogeneous points (HPL) neither, as expected from the HP behavior. Anchored point-supported lines (AHPL and AMPPL) behave similarly and close to consistency.

The average NEES results are shown in Fig. 22 – please notice the logarithmic vertical scales. We observe that the only parametrizations that behave consistently are the anchored, point-supported lines AHPL and AMPPL.

The RMSE results are shown in Figs. 23 and 24. The Plücker-based lines behave poorly, especially if not anchored. Among the point-supported lines, anchored parametrizations exhibit both smaller errors and larger error estimates, indicating better consistency. We can say that they inherit the properties of the point parametrizations they are based on.

The second scenario (Fig. 20(b)) corresponds to a forward motion, a situation that is more challenging for mo-

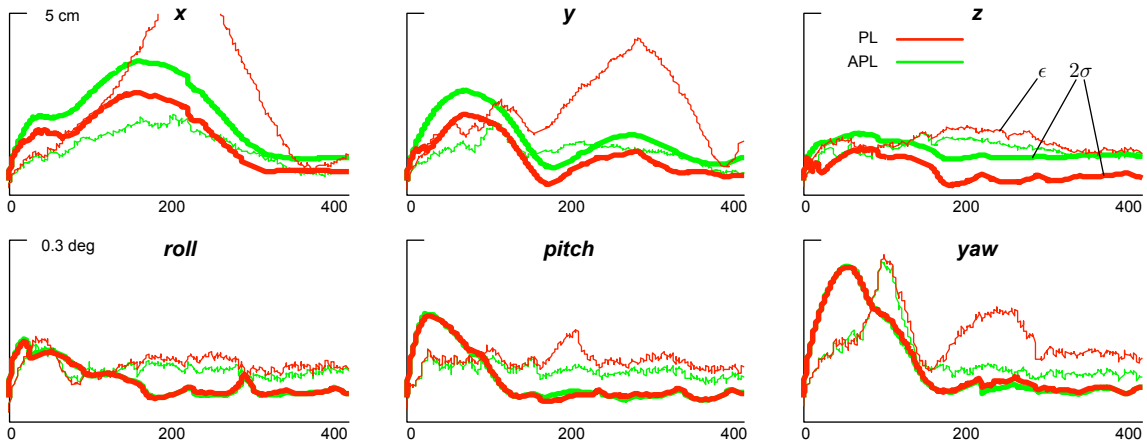



**Fig. 23** Averaged RMS errors $\epsilon$ of the Plücker-based line parametrizations PL and APL against the $2\sigma$ estimated bounds. Anchoring the Plücker line helps improving the estimates, but none of these Plücker parametrizations seem to work correctly. See discussion in Section 8 about the Plücker constraint.

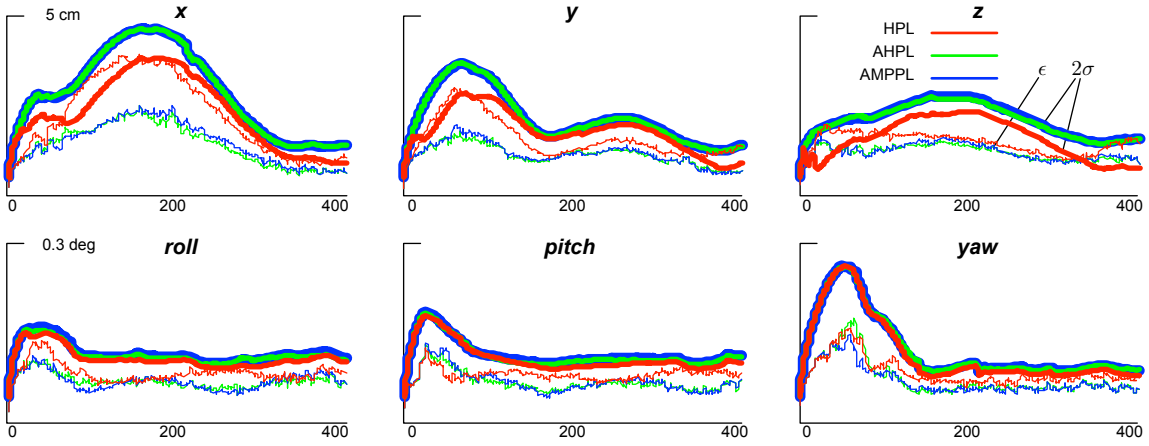



**Fig. 24** Averaged RMS errors $\epsilon$ of the three point-based line parametrizations HPL, AHPL and AMPPL against the $2\sigma$ estimated bounds. Anchoring has produced both a larger $2\sigma$ bound and a smaller error $\epsilon$. AHPL and AMPPL behave almost exactly.

nocular SLAM as the parallax increase is slow and therefore the scene observability is weak. 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 the average NEES results for the anchored parametrizations (Fig. 25), namely APL, AHPL and AMPPL – the rest are clearly inconsistent. The results are equivalent to those of the first scenario (compare to the three corresponding plots in Fig. 22), showing an important robustness in face of large variations of the operating conditions.

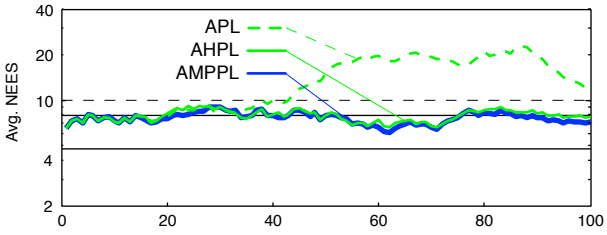
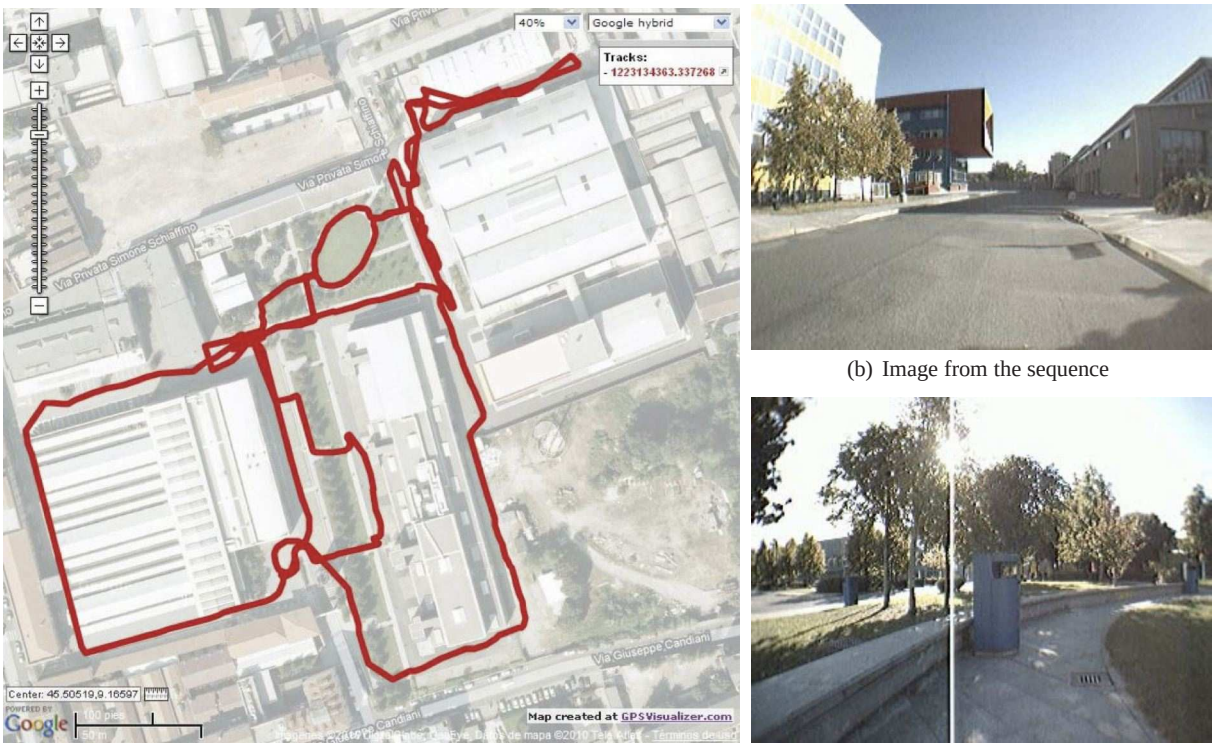



**Fig. 25** Average NEES over 25 runs for 100 frames (frontal trajectory) for APL, AHPL and AMPPL. Again, only anchored point-supported lines behave close to consistency.

## 7 Experimental results

### 7.1 Robocentric EKF-SLAM with points

An interesting alternative to the algorithm here benchmarked is robocentric EKF-SLAM (Castellanos et al, 2007; Marzorati et al, 2008; Civera, 2009). Robocentric EKF-SLAM




(a) RTK-GPS ground truth over an aerial Google Maps view

(b) Image from the sequence

(c) Image from the sequence

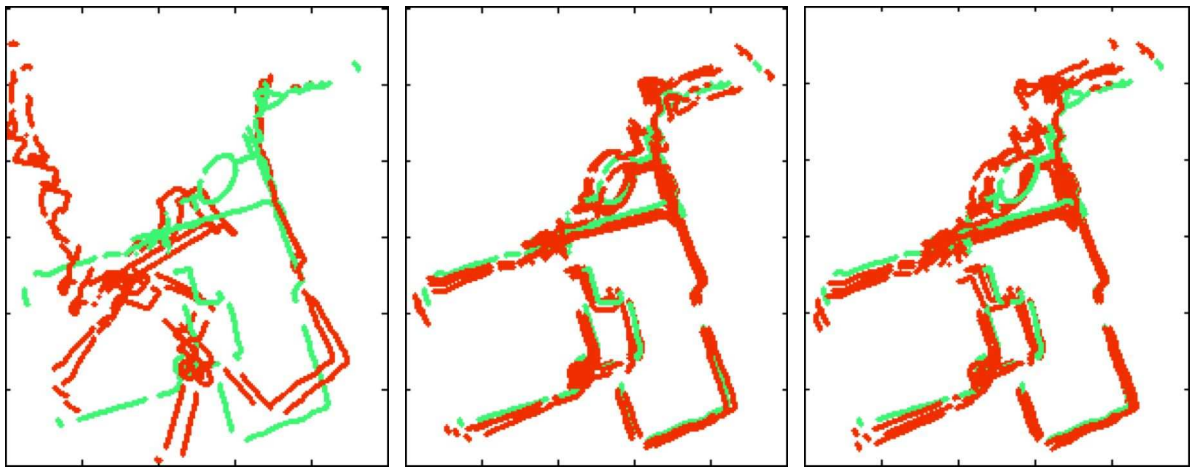
**Fig. 26** Bovisa urban image sequence data. (a) Ground truth of the 1.6 km trajectory. (b, c) Two representative images from the sequence.

performs the composition of the current frame and the local motion after the landmarks update. This greatly helps reducing linearization errors, improving accuracy and consistency (Castellanos et al, 2007; Huang et al, 2008). The transformation affects the full landmarks map, with the consequence of making the robot pose $\mathsf{T}$ become the origin at every step, with null covariance, and hence the *robocentric* term. Immediately after, at initialization time, we have that the anchor to generate is $\mathbf{p}_0 = \mathsf{T} \equiv \mathbf{0}$, with null covariance. We could then think of dropping it from the parametrization, obtaining in *e.g.* the AHP case, pure homogeneous points HP. This combination of HP and robocentric SLAM constitutes exactly the algorithm proposed by Marzorati et al (2008).

We have run robocentric SLAM using HP, AHP and AMPP on a sequence of more than 68.000 images taken during an outdoors run of over 1600 m, covering an area of some $250\,\mathrm{m}\times 250\,\mathrm{m}$ (the Bovisa dataset from Rawseeds (Bonarini et al, 2006; Ceriani et al, 2009)). Figure 26 shows an aerial view of the covered trajectory, along with two representative images of the sequence. The algorithm is set to visual odometry mode, meaning that landmarks exiting the field of view are deleted. This way, one single EKF can be used for the whole run of 1.600 m. Furthermore, the algorithm incorporates a 1-point RANSAC outlier rejection step (Civera et al, 2009) that discards the negative effect introduced by outlier correspondences. With all these features we can attribute the outcome differences uniquely to landmark parametrization. Three different runs have been made for each parameterization, each one of them initializing and measuring different features of the sequence, randomly chosen in order to increase independence between runs.

The results in Fig. 27 show that HP performs much worse than AHP and AMPP, agreeing with our previous simulation results. Using HP, the trajectory is off by 167 m in mean (80% of the area dimensions) with respect to RTK-GPS (with centimetric accuracy). AMPP derived 20 m, and AHP only 13 m averaging the three runs. We can draw the same conclusions as in the previous simulations: anchoring is the major factor of improvement, and the difference between modified-polar and homogeneous representations can be considered negligible.

We observe with this evaluation that the analysis performed in this paper is valid also for this improved EKF-based estimation algorithm. Known algorithms improving the degree of linearity with respect to classic EKF-SLAM are *submapping*, where a map is divided into a set of local maps, and *robocentric*, where the map is always referred to the sensor frame. In both cases the covariance of the local motion $(\mathsf{T} - \mathbf{p}_0)$ is small and the effect of anchoring must be smaller than in the standard formulation. We see with the current experiment that even in these cases anchoring is still necessary.



(a) Average error HP: 167m (b) Average error AHP: 13m (c) Average error AMPP: 20m

**Fig. 27** Robocentric EKF visual odometry (red) against RTK-GPS (green), over an outdoors run of around 1.600 m and more than 68.000 images. The figure shows 3 runs for each parameterization. (a) HP exhibits poor performances, and the estimated trajectory is off by 167 m in average. (b) AHP completes the run successfully with an averaged drift of 13m, an order of magnitude less. (c) AMPP, with an averaged drift of 20 m, is only slightly outperformed by AHP. Notice that these results are in agreement with the simulation results (figs. 17, 18 and 19), where AMPP and AHP show a very similar performance and clearly outperform HP.

### 7.2 Classical EKF-SLAM with segments

We have implemented EKF-SLAM with real images for the line parametrizations PL, APL, HPL, AHPL and AMPPL, using the segment detector and tracker of Berger and Lacroix (2010). The scene contains a $90^\circ$ dihedral with several segments on its planes (Fig. 28). The camera, controlled by a robotic arm, performs a $30\times30$ cm square trajectory perpendicularly to its optical axis. The position increments given by the arm are corrupted and used as odometry inputs to the system, thus providing the metrics for scale observability.

The videos `plucker-based-lines.mov` and `point-supported-lines.mov` show the methods PL, APL, HPL and AHPL running in parallel (the AMPPL video, indistinguishable from the AHPL one, is not shown for space reasons). A selection of snapshots of the AMPPL run is shown in Fig. 28. It is worth mentioning the enormous size of the uncertainty ellipsoids (in yellow color in the movies) shortly after initialization, a consequence of the undelayed initialization of unobserved DOF. As it happened with simulations, to the naked eye there are not big differences between PL and APL, or between HPL and AHPL. However, Plücker-based lines behave differently from point-supported lines. The most remarkable difference is the smaller size of the uncertainty ellipsoids for point-supported lines, due to the superior representativeness of these parametrizations. This issue is discussed in more detail in Section 8.

To evaluate the accuracy of the resulting maps we identify the two planes of the dihedral by optimally fitting them on the segments endpoints, and compute two different coplanarity errors. The first one is defined by the standard deviation of the distances from the segments midpoints to their support plane. The second one is defined by the angles between the segments and their support plane. Finally, we report the angle between the two planes. The results are summarized in Table 3. We observe a progressive improvement of all of the map accuracy indicators as we adopt point-supported parametrizations and incorporate anchors. AHPL and AMPPL, the two parametrizations incorporating both features, exhibit an equivalent performance, the best of all, with coplanarity errors of as low as 1.5 mm (notice that the segments are at some 120 cm from the camera). Even when anchored, the Plücker lines exhibit a poorer performance than point-supported lines. Refer to Section 8 for a discussion.

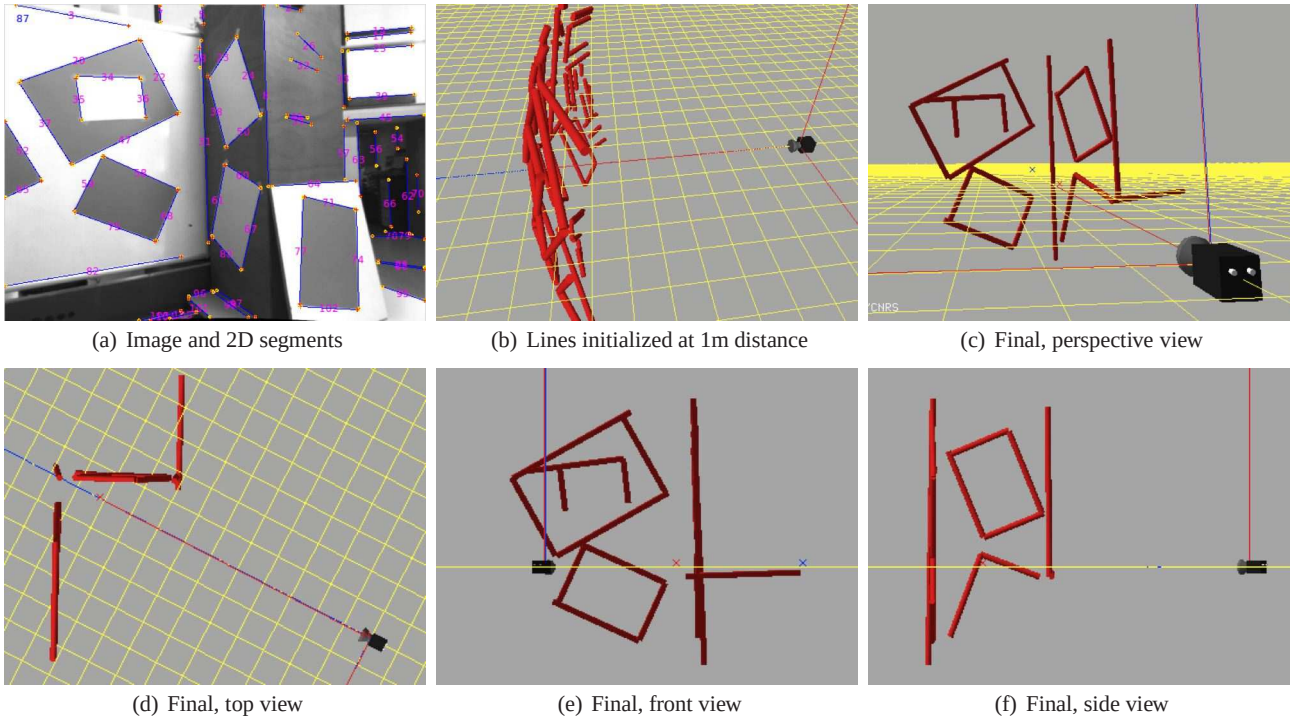
To further emphasize the effect of anchoring, we have repeated PL, APL and AHPL runs with the segment detector set to initialize lines only at the first frame. This situa-

**Table 3** Reconstruction accuracy of the segments maps. DTP: distance to plane; ATP: angle to plane; ABP: angle between planes.

| | PL | APL | HPL | AHPL | AMPPL |
|---|---|---|---|---|---|
| DTP ($1\sigma$, mm) | 7.5 | 2.7 | 3.1 | 1.6 | 1.5 |
| ATP ($1\sigma$, deg) | 1.02 | 0.54 | 0.58 | 0.19 | 0.20 |
| ABP (deg) | 88.26 | 89.89 | 88.95 | 90.00 | 90.00 |

**Table 4** Reconstruction accuracy of the segments maps with all initializations at the origin.

| | PL | APL | AHPL |
|---|---|---|---|
| DTP ($1\sigma$, mm) | 2.0 | 2.1 | 1.5 |
| ATP ($1\sigma$, deg) | 0.49 | 0.49 | 0.20 |
| ABP (deg) | 89.34 | 89.52 | 90.00 |



(a) Image and 2D segments (b) Lines initialized at 1m distance (c) Final, perspective view

(d) Final, top view (e) Final, front view (f) Final, side view

**Fig. 28** Monocular EKF-SLAM with ULI of AMPPL segments. (a) A sample image with the tracked 2D segments. (b) The set of initialized lines, all at a distance of 1 m from the camera, corresponding to the selected prior, thus defining a spherical distribution. (c) The final map, from a viewpoint close to the camera (compare with (a)). (d, e, f) Top, front and side views of the final map, showing the correct reconstruction of the scene, with the dihedral planes at precise right angles. The grid in the 3D views has 10 cm steps.

tion is generally unrealistic because it assumes that all the world is visible from the first sensor location, but when the hypothesis is valid (as is the case for this experiment) it constitutes for this precise fact the ideal situation, from which the best possible mapping results have to be expected. It produces anchors that are exactly zero, $\mathbf{p}_0 \equiv \mathsf{T} = \mathbf{0}$, and with null covariance. Results in Table 4 show, when compared to Table 3, that the effect of anchoring disappears to make unanchored parametrizations equivalent to anchored ones. It is worth noticing that only point-supported anchored parametrizations perform in the general situation similar than in this ideal one, and therefore that anchoring contributes to keeping a performance comparable to the best case.

## 8 Elements of discussion

### 8.1 Redundancy and constraints in the EKF

There exist recurrent discussions on whether estimators should employ minimal state parametrizations or not, and the effects that redundancy and constraints have in EKF estimates. It is not our aim now to provide a detailed analysis of these issues here, but this paper has clearly showed that redundancy can be exploited to our benefit. With a little insight we discover that not all redundancies are the same, neither the constraints, and that they do not always come together. Some ideas to situate these concepts follow.

1. Using redundant parametrizations is possible in EKF because of its Bayesian character. Bayesian estimators use predictions to generate priors that constrain the redundant DOF that otherwise would make convergence difficult or even impossible (we think especially on iterative optimizers such as BA where prediction is not present and a good canonical choice of the parametrizations used is crucial for a quick convergence – see for example (Engels et al, 2006)).
2. Homogeneous vectors are redundant or *equivalent* under proportionality transforms. This equivalence has continuity in all dimensions of the state space, and thus it imposes no constraint to the filter: the new states resulting from EKF updates are always valid homogeneous vectors.
3. Quaternions are redundant only with respect to symmetry: a quaternion and its negative are equivalent. They are also constrained by a unity norm which defines a unit spheroid in $\mathbb{R}^4$. In EKF, this normalization constraint can be applied explicitly, via $\mathsf{Q} \leftarrow \mathsf{Q}/\|\mathsf{Q}\|$, its Jacobian $J$ and the EKF prediction equations, resulting in a *projection* of the covariance onto the hyperplane $\mathcal{S} = null(J)$ tangent to the unit spheroid. Subsequent EKF updates, constrained to $\mathcal{S}$, result in quater-

nions escaping the spheroid and violating the spherical constraint, and thus renormalization is needed.
4. Anchored landmarks are redundant in the sense that landmarks with different anchors may be equivalent. There exists a continuity of solutions, in this case not related to proportionality, with no constraints. As we have seen, this redundancy allows us to arbitrarily select the anchor with the most beneficial effects.
5. Plücker lines are defined in the projective space $\mathbb{P}^5$ and are therefore equivalent under proportionality transformations. This DOF is not constrained. However, they contain a second redundant DOF affected by the Plücker constraint $\mathbf{n} \perp \mathbf{v}$. The Plücker constraint is more delicate than the normalization constraint in the quaternion because it can only be applied implicitly, via $\mathbf{n}^\top \mathbf{v} = 0$ and the EKF correction equations. The application of the EKF correction equations means that the covariance is *intersected with* (not *projected to*, as it was in the quaternion) the constraint manifold, with the subsequent risk of collapse of the covariances matrix. This does have a noticeable impact on the filter and is further discussed in the next section.

### 8.2 The Plücker constraint

We have seen in Section 3.1 that for a pair $(\mathbf{n}, \mathbf{v})$ to be a Plücker line the Plücker constraint $\mathbf{n}^\top \mathbf{v} = 0$ is mandatory. We have ensured its satisfaction at initialization time by defining the inverse-distance prior in the $\beta$-plane, but we have not enforced it further during landmark updates, for several reasons. One reason is our desire to use a common algorithm for all parametrizations so that the differences in performance can be better interpreted. A second reason is that we did not find a clean and convincing method for enforcing such constraints in the EKF framework. In the linear case, enforcing implicit equality constraints $\mathbf{Hx} = 0$ can be done by performing a KF update with a synthetic measurement $0 = \mathbf{z} = \mathbf{Hx}$ with infinite information. This 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, creating a lifelong guarantee of the constraint satisfaction. Unfortunately, in EKF the combination of uncertainty and non-linearity prevents non-linear implicit equality constraints $h(\mathbf{x}) = 0$ from being enforced this straightforwardly. This problem has been treated and solved by Lemaire and Lacroix (2007) for the Euclidean Plücker lines (*i.e.*, with normalized director vector) using the smoothly constrained Kalman Filter (Geeter et al, 1997). The idea is to apply a number of relaxed constraints over time, with an EKF update $0 = h(\mathbf{x}) + n$, where $n$ is a noise vector with a variance decreasing with time, to make the filter gradually converge to a state satisfying the equality constraint. This method is directly applicable to PL and APL. However, it requires several tuning parameters (initial covariance of $n$, rate of decay of $n$, at which times and/or under which conditions to apply it, when to stop) and, for this reason, we do not feel the solution to be satisfactory enough.

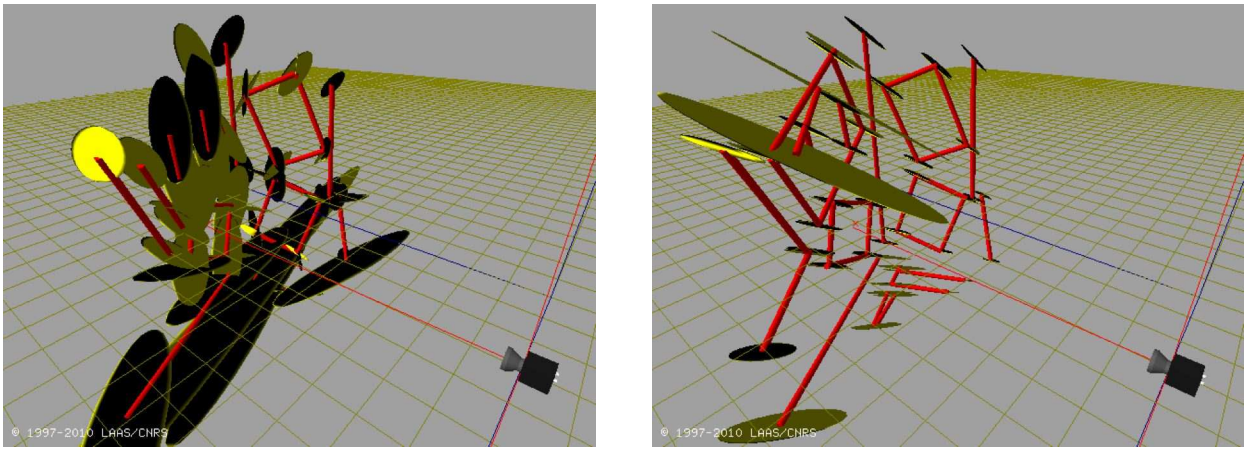
These facts might very well be at the base of the poor results of APL, which otherwise would be expected to perform similarly to its point-counterpart AHP. According to our experience, the improvements produced by enforcing the constraint with the methods here explained are small, and in any case not sufficient. This is possibly due to tuning issues, or is maybe a matter of the method itself: the constraint is only truly enforced at the end of the process, when the noise $n$ has converged to zero, and the errors produced during convergence must most possibly have adverse effects. We have not investigated this hypothesis fully, mainly because there exist other strong reasons to prefer point-supported lines over Plücker-based lines, as we discuss in the following section.

### 8.3 Endpoints management in Plücker-based and point-supported lines

In addition to an accurate estimation of the infinite lines supporting the segments, a proper endpoints management is crucial to produce meaningful maps of segments. The methods for endpoints management require some information to be stored out of the map. We limit this to the two abscissas of the endpoints expressed in a local reference frame of the line (see Appendix A for details on endpoints management).

Plücker-based lines PL and APL condensate all the information of the initial observation in the plane normal, via $\mathbf{n} = \mathbf{m}_1 \times \mathbf{m}_2$, Eq. (27), and all other information on the endpoints' initial view is lost. This constitutes an important drawback: the local line origin (point $\mathbf{q}$, Fig. 3), where the abscissas are referenced to, moves with the line's orientation, which is initially unobserved and therefore undergoes large variations during the convergence phase. Fig. 29(a) shows that not even the cross-correlations in the covariances matrix are able to account for this information. In other words, the endpoints cannot be assumed to remain stable from one frame to the next one. Because choosing an alternative local origin for the line with better properties does not seem to be trivial, the only reasonable strategy for managing endpoints is to reset them at each frame using the current observation, potentially losing information about the segment extension observed in previous frames.

On the contrary, anchored point-supported line parametrizations AHPL and AMPPL have the ability to respect the initial view of the two endpoints via the anchor $\mathbf{p}_0$ and the director vectors $(\mathbf{m}_1, \mathbf{m}_2)$, as it can be observed in Fig. 29(b). This information is part of the state and is available even after large updates thanks to the role that cross-correlations play in the EKF update. This allows us to use




(a) APL (b) AMPPL

**Fig. 29** Endpoints covariance ellipsoids in Plücker-based and point-supported lines, showing the superior representativity of the latter. The snapshots are taken 30 frames after initialization, before the end of the convergence phase. In APL, the ellipsoids are "discs" in the plane defined by the segment and the landmark anchor. In AMPPL, the ellipsoids are "pencils" pointing to the anchor.

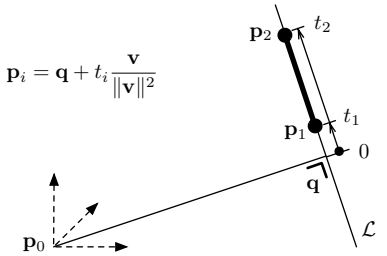
more elaborate ways of updating the segments endpoints – see Appendix A for further reference.

## 9 Conclusions

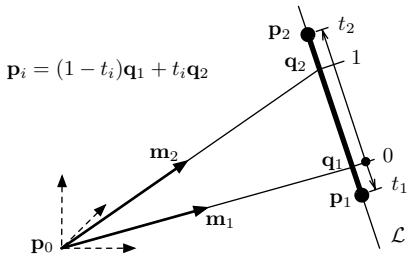
This paper was initially conceived as a compendium of landmark parametrizations for monocular SLAM. Our very first aim was to show that all these methods are very intimately related, as we have exposed amply. As the work evolved, we realized that the material and insights provided should also be a good basis for establishing good practices for approaching a more general problem, that is, the problem of accurately estimating high-dimensional dynamic systems of non-linear nature and huge uncertainty levels with the use of relatively simple analytic tools such as the EKF.

We summarize here the main concepts and results presented in this work:

1. Undelayed landmark initialization (ULI) is fundamental in the sense that it is the way we can make use of all the geometrical information provided by the landmarks: from the first observation, up to the infinity range, and independently of the sensor trajectory.
2. Implementing ULI within EKF is difficult because of the combination of non-linearity and unbounded uncertainty regions.
   - Linearity and Gaussiannity are the two keys to satisfactory EKF operation. Astute transformations and redundancy in the parametric descriptions of the system produce analytic expressions and probability densities that are well adapted to the estimator in use.
   - Inverse-distance behavior of the unmeasurable parameters is the key that makes undelayed initialization of landmarks possible when they are perceived from projective sensors.
   - Once ULI is achieved, landmark anchoring has shown to be the major actor in further improving linearity to reach satisfactory levels.
   - Manipulations on the measurable parameters, such as the use of rectangular or polar coordinates for the director vectors, have shown to produce no remarkable effects. This is because, being these parameters measured with good accuracy, their degree of uncertainty is small and the functions in which they appear are regarded by EKF as being linear.
   - Therefore, AMPP and AMPPL parametrizations are preferred over AHP and AHPL because of their lower computation cost for an equivalent performance.
3. Because of the higher representativeness of anchored point-supported lines over unanchored and Plücker-based lines, and because of the absence of constraints to be guaranteed, the anchored point-supported lines constitute the preferred choice for undelayed monocular EKF-SLAM. Therefore, AMPPL is the preferred parametrization for infinite straight lines.
4. A great number of parametrizations can be regarded as a sequence of small modifications of ones with respect to the others (Fig. 8). We have traversed a family of eight parametrizations and established quasi-trivial links between them. We have shown that estimating points and lines is fundamentally the same problem and admits fundamentally the same solutions. Estimating other parametric entities such as conic sections, splines or planes should also be feasible as long as we can take good advantage of the principles exposed here.
5. Measuring non-linearity for high-dimensional MIMO functions is an interesting but difficult task, especially




(a) Plücker-based lines PL and APL.




(b) Point-supported lines HL, AHL and AMPPL.

**Fig. 30** Segment endpoints in the local ordinate frame of Plücker-based and point-supported lines.

if we want the measure to be useful for comparing parametrizations of different dimensionality.
  - We defined a linearity index that incorporates the local knowledge of the uncertainty region, which is well suited for EKF usage.
  - The index is expressed in the measurement space, which is common to all parametrizations and allows us to compare them with each other.
6. Visual inspection, linearity analysis, RMS errors and average NEES consistency, all give a coherent picture of the performance of each parametrization.
7. Using more evolved algorithms such as robocentric EKF-SLAM has not altered the relative performances of the parametrizations. This is because the superiority of some parameterizations over others comes from the severe linearity constraints that the EKF imposes, and hence the conclusions drawn in this paper could be extended to any EKF-based visual SLAM algorithm. It would be interesting to see if this also applies to iterative optimizers such as SBA, and in such case if the improvements (*e.g.* fewer optimizer iterations) are sufficient to compensate for the extra amount of computational power (*e.g.* more operations per iteration) that would be required.

## A Segment endpoints management

The segment's endpoints in 3D space are maintained out of the filter via two abscissas $(t_1, t_2)$ defined in the local 1D reference frame of the line.

– In Plücker-based lines (Fig. 30(a)) 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 $\mathbf{u} = \mathbf{v}/\|\mathbf{v}\|^2$ providing the unit length (we make the norm of $\mathbf{u}$ proportional to distance to improve its projective behavior). Each endpoint $\mathbf{p}_i$ is specified by an abscissa $t_i$ such that

$$\mathbf{p}_i = \mathbf{q} + t_i\mathbf{v}/\|\mathbf{v}\|^2 \qquad = \mathbf{p}_0 + \frac{\mathbf{v}\times\mathbf{n} + t_i\mathbf{v}}{\|\mathbf{v}\|^2} \,. \tag{74}$$

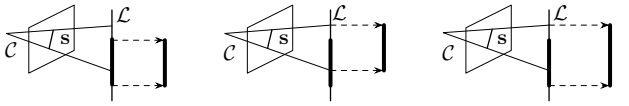
– In point-supported lines (Fig. 30(b)) the endpoints are defined with respect to the support points $\mathbf{q}_i = \mathbf{p}_0 + \mathbf{m}_i/\rho_i$. The origin is at $\mathbf{q}_1$ and the unit length is defined by $\mathbf{q}_2 - \mathbf{q}_1$, leading to the endpoints

$$\mathbf{p}_i = (1-t_i)\mathbf{q}_1 + t_i\mathbf{q}_2 = \mathbf{p}_0 + (1-t_i)\frac{\mathbf{m}_1}{\rho_1} + t_i\frac{\mathbf{m}_2}{\rho_2} \,. \tag{75}$$

The initial abscissas are defined trivially with $(t_1, t_2) = (0, 1)$.

Before updating, we need to back-project the currently observed endpoints onto the 3D line. This is done by triangulating the 3D line with the optical rays of the two currently observed 2D endpoints. To avoid aberrant results it is advised to update each abscissa only if the triangulation angle between the ray and the line is greater than a certain value (we use $15°$). The result is a couple of candidate abscissas $(t'_1, t'_2)$ that are assigned to $(t_1, t_2)$ depending on the the following rules:

– During convergence the abscissas are either not updated (Fig. 31(a), used for AHPL and AMPPL), thus reflecting the initial observation, or systematically updated (Fig. 31(b), used for PL, APL and HPL), simply reflecting the last observation.
– Once the line has converged, an extending-only policy is applied (Fig. 31(c)): the abscissa is updated only if this lengthens the 3D segment. It is to note that a converged line is ready for reparametrization to any minimal (or at least more economical) form.




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

**Fig. 31** 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.

## References

Aidala V, Hammel S (1983) Utilization of modified polar coordinates for bearings-only tracking. IEEE Transactions on Automatic Control 28(3):283–294

Bailey T (2003) Constrained initialisation for bearing-only SLAM. In: Int. Conf. on Robotics and Automation, vol 2, pp 1966–1971

Bailey T, Nieto J, Guivant J, Stevens M, Nebot E (2006) Consistency of the EKF-SLAM algorithm. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Beijing, China, pp 3562–3568

Bar-Shalom Y, Li XR, Kirubarajan T (2001) Estimation with applications to tracking and navigation. John Wiley and Sons

Bartoli A, Sturm P (2001) The 3D line motion matrix and alignment of line reconstructions. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol 1, pp 287–292

Berger C, Lacroix S (2010) DSeg: Détection directe de segments dans une image. In: Reconnaissance des Formes et Intelligence Artificielle

Bonarini A, Burgard W, Fontana G, Matteucci M, Sorrenti DG, Tardos JD (2006) RAWSEEDS: Robotics Advancement through Web-publishing of Sensorial and Elaborated Extensive Data Sets. In: In proceedings of IROS'06 Workshop on Benchmarks in Robotics Research

Castellanos JA, Neira J, Tardós JD (2004) Limits to the consistency of the EKF-based SLAM. In: 5th IFAC Symp. on Intelligent Autonomous Vehicles, Lisboa, PT

Castellanos JA, Martinez-Cantin R, Tardós JD, Neira J (2007) Robocentric map joining: improving the consistency of EKF-SLAM. In: Robotics and Autonomous Systems, vol 55, pp 21–29

Ceriani S, Fontana G, Giusti A, Marzorati D, Matteucci M, Migliore D, Rizzi D, Sorrenti DG, Taddei P (2009) Rawseeds ground truth collection systems for indoor self-localization and mapping. Autonomous Robots 27(4):353–371

Chiuso A, Favaro P, Jin H, Soatto S (2002) Structure from motion causally integrated over time. In: IEEE Trans. on Pattern Analysis and Machine Intelligence, vol 24, pp 523–535

Civera J (2009) Real-time EKF-Based Structure from Motion. PhD thesis, Universidad de Zaragoza

Civera J, Davison AJ, Montiel JMM (2008) Inverse depth parametrization for monocular SLAM. IEEE Trans on Robotics 24(5)

Civera J, Grasa OG, Davison AJ, Montiel JMM (2009) 1-point RANSAC for EKF-based structure from motion. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems

Davison AJ (2003) Real-time simultaneous localisation and mapping with a single camera. In: Int. Conf. on Computer Vision, Nice, vol 2, pp 1403–1410

Davison AJ, Reid ID, Molton ND, Stasse O (2007) MonoSLAM: Real-time single camera SLAM. Trans on Pattern Analysis and Machine Intelligence 29(6):1052–1067

Eade E, Drummond T (2006a) Edge landmarks in monocular SLAM. In: British Machine Vision Conf., Edimburgh, Scotland

Eade E, Drummond T (2006b) Scalable monocular SLAM. IEEE Int Conf on Computer Vision and Pattern Recognition 1:469–476, DOI http://doi.ieeecomputersociety.org/10.1109/CVPR.2006.263

Eade E, Drummond T (2007) Monocular SLAM as a graph of coalesced observations. In: IEEE Int. Conf. on Computer Vision

Engels C, Stewénius H, Nistér D (2006) Bundle adjustment rules. In: Photogrammetric Computer Vision

Gee AP, Mayol W (2006) Real-time model-based SLAM using line segments. In: LNCS proceedings of the 2nd International Symposium on Visual Computing

Gee AP, Chekhlov D, Calway A, Mayol-Cuevas W (2008) Discovering higher level structure in visual SLAM. In: IEEE Trans. on Robotics special issue on Visual SLAM, vol 24, pp 980–990

Geeter JD, Brussel HV, Schutter JD, Decréton M (1997) A smoothly constrained Kalman filter. In: IEEE Trans. on Pattern Analysis and Machine Intelligence, vol 19, pp 1171–1177

Haner S, Heyden A (2010) On-line structure and motion estimation based on an novel parameterized extended Kalman filter. In: Int. Conf. on Pattern Recognition, Istambul, Turkey

Holmes SA, Klein G, Murray DW (2008) A square root UKF for visual monoSLAM. In: IEEE Int. Conf. on Robotics and Automation, Pasadena

Huang G, Mourikis A, Roumeliotis S (2008) Analysis and improvement of the consistency of extended Kalman filter based SLAM. In: IEEE Int. Conf. on Robotics and Automation, pp 473–479

Huang S, Dissanayake G (2007) Convergence and consistency analysis for extended Kalman filter based SLAM. In: IEEE Transactions on Robotics, vol 23, pp 1036–1049

Klein G, Murray D (2007) 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, pp 1–10

Klein G, Murray D (2008) Improving the agility of keyframe-based SLAM. In: 10th European Conference on Computer Vision, Marseille, pp 802–815

Konolige K, Agrawal M (2008) FrameSLAM: From bundle adjustment to real-time visual mapping. IEEE Trans on Robotics 24(5):1066–1077

Kwok NM, Dissanayake G (2003) Bearing-only SLAM in indoor environments using a modified particle filter. In: Australasian Conf. on Robotics and Automation (ACRA), Brisbane, Australia

Kwok NM, Dissanayake G (2004) An efficient multiple hypothesis filter for bearing-only SLAM. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Sendai, Japan

Lemaire T, Lacroix S (2007) Monocular-vision based SLAM using line segments. In: IEEE Int. Conf. on Robotics and Automation, Rome, Italy, pp 2791–2796

Lemaire T, Lacroix S, Solà J (2005) A practical 3D bearing only SLAM algorithm. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Edmonton, Canada

Lourakis M, Argyros A (2004) The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Tech. Rep. 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, available from http://www.ics.forth.gr/ lourakis/sba

Marzorati D, Matteucci M, Migliore D, Sorrenti DG (2008) Monocular SLAM with inverse scaling parametrization. In: Proc. of the British Machine Vision Conference, Leeds

Montiel JMM, Civera J, Davison AJ (2006) Unified inverse depth parametrization for monocular SLAM. In: Robotics: Science and Systems, Philadelphia, USA

Paz LM, Piniés P, Tardós JD, Neira J (2008) Large scale 6DOF SLAM with stereo-in-hand. IEEE Trans on Robotics 24(5)

Piniés P, Lupton T, Sukkarieh S, Tardós JD (2007) Inertial aiding of inverse depth SLAM using a monocular camera. In: Int. Conf. on Robotics and Automation

Smith P, Reid I, Davison AJ (2006) Real-time monocular SLAM with straight lines. In: British Machine Vision Conf., vol 1, pp 17–26

Smith R, Cheeseman P (1987) On the representation and estimation of spatial uncertainty. Int Journal of Robotics Research 5(4):56–68

Solà J (2007) Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach. PhD thesis, Institut National Polytechnique de Toulouse

Solà J (2010) Consistency of the monocular EKF-SLAM algorithm for 3 different landmark parametrizations. In: IEEE Int. Conf. on Robotics and Automation, Anckorage, USA

Solà J, Monin A, Devy M, Lemaire T (2005) Undelayed initialization in bearing only SLAM. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Edmonton, Canada, pp 2499–2504

Solà J, Monin A, Devy M, Vidal-Calleja T (2008) Fusing monocular information in multi-camera SLAM. IEEE Trans on Robotics 24(5):958–968

Solà J, Marquez D, Codol JM, Vidal-Calleja T (2009a) An EKF-SLAM toolbox for MATLAB. URL http://homepages.laas.fr/jsola/JoanSola/eng/toolbox.html

Solà J, Vidal-Calleja T, Devy M (2009b) Undelayed initialization of line segments in monocular SLAM. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Saint Louis, USA, pp 1553–1558

Strasdat H, Montiel JMM, Davison AJ (2010) Real-time monocular SLAM: Why filter? In: Int. Conf. on Robotics and Automation, Anckorage, USA

Sunderhauf N, Lange S, Protzel P (2007) 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

Triggs B, McLauchlan P, Hartley R, Fitzgibbon A (2000) Bundle adjustment – A modern synthesis. In: Triggs W, Zisserman A, Szeliski R (eds) Vision Algorithms: Theory and Practice, LNCS, Springer

Verlag, pp 298–375