



HAL
open science

Visual EKF-SLAM from Heterogeneous Landmarks

Jorge Esparza-Jiménez, Michel Devy, José Gordillo

► **To cite this version:**

Jorge Esparza-Jiménez, Michel Devy, José Gordillo. Visual EKF-SLAM from Heterogeneous Landmarks. *Sensors*, 2016, 16 (4), 10.3390/s16040489 . hal-01354880

HAL Id: hal-01354880

<https://hal.science/hal-01354880>

Submitted on 19 Aug 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Article

Visual EKF-SLAM from Heterogeneous Landmarks [†]

Jorge Othón Esparza-Jiménez ^{1,*}, Michel Devy ^{2,‡} and José L. Gordillo ^{1,‡}

¹ Center for Robotics and Intelligent Systems, Tecnológico de Monterrey, Monterrey 64849, Mexico; jlgordillo@itesm.mx

² CNRS, LAAS, Université de Toulouse, 7 Avenue du Colonel Roche, Toulouse F-31400, France; michel.devy@laas.fr

* Correspondence: jo.esparza@itesm.mx; Tel.: +52-81-8358-2000 (ext. 5141)

† This paper is an extended version of our paper published in Esparza-Jimenez, J.O.; Devy, M.; Gordillo, J.L. EKF-based SLAM fusing heterogeneous landmarks. In Proceedings of the IEEE 2014 17th International Conference on Information Fusion (FUSION), Salamanca, Spain, 7–10 July 2014; pp. 1–8

‡ These authors contributed equally to this paper.

Academic Editors: Lianqing Liu and Yajing Shen

Received: 31 December 2015; Accepted: 30 March 2016; Published: date

Abstract: Many applications require the localization of a moving object, e.g., a robot, using sensory data acquired from embedded devices. Simultaneous localization and mapping from vision performs both the spatial and temporal fusion of these data on a map when a camera moves in an unknown environment. Such a SLAM process executes two interleaved functions: the front-end detects and tracks features from images, while the back-end interprets features as landmark observations and estimates both the landmarks and the robot positions with respect to a selected reference frame. This paper describes a complete visual SLAM solution, combining both point and line landmarks on a single map. The proposed method has an impact on both the back-end and the front-end. The contributions comprehend the use of heterogeneous landmark-based EKF-SLAM (the management of a map composed of both point and line landmarks); from this perspective, the comparison between landmark parametrizations and the evaluation of how the heterogeneity improves the accuracy on the camera localization, the development of a front-end active-search process for linear landmarks integrated into SLAM and the experimentation methodology.

Keywords: SLAM; EKF; computer vision; landmarks; points; lines

1. Introduction

Simultaneous localization and mapping (SLAM) is an essential functionality required on a moving object for many applications where the localization or the motion estimation of this object must be determined from sensory data acquired by embedded sensors. The object is typically a robot or a vehicle, the position of which is required to deal with robust navigation in a cluttered environment. A SLAM module could also be required on smart tools (phones, glasses) to offer new services, e.g., augmented reality [1–3].

The robot or smart tool could be equipped with a global navigation satellite system (GNSS) receiver for outdoor applications to obtain directly a position with respect to the Earth reference frame [4]; at present, indoor localization with respect to a building reference frame could also be provided using ultra-wide band (UWB) [5], WiFi [6] or RF devices [7], on the condition that a hotspot or antenna network has been previously installed and calibrated. However, the direct localization is not always available (*i.e.*, occlusions, bad propagation, multiple paths); so generally, they are combined using loose or tie fusion strategies, with motion estimates provided by an inertial measurement unit (IMU), integrating successive accelerometer and gyro data [8–10]. Nevertheless, even GPS-IMU fusion could fail or be too inaccurate. Depending on the context, *a priori* knowledge could be exploited; a map

matching function can be sufficient, as in the GPS-based navigation systems available on commercial vehicles.

Considering mobile robots or the emerging autonomous vehicles, it is necessary to also make use of data acquired on the environment with embedded exteroceptive sensors, e.g., laser range finders [11], 3D sensors (ToF cameras [12], Kinect [13]) or vision with many possible modalities (mono, stereo, omni). Here, only visual SLAM is considered, due to the fact that it could be integrated both in low-cost unmanned ground and aerial vehicles and on smart tools equipped with cameras. Many visual SLAM methods have been proposed during the last decade [3,14].

A SLAM method combines two interleaved functionalities shown in Figure 1: the front-end detects and tracks features from images acquired from the moving robot, while the back-end, interpreting these feature and landmark observations, estimates both the landmark and robot positions with respect to the selected reference frame.

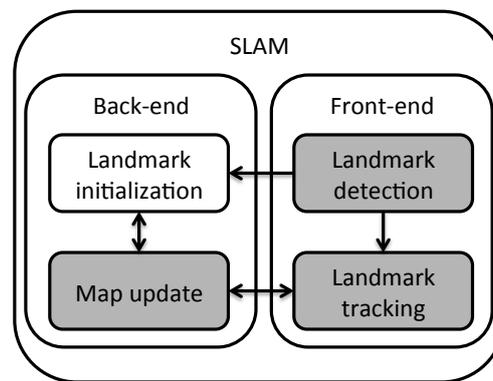


Figure 1. General block diagram of the SLAM solution. The blocks in gray represent the contributions of the present method.

The back-end can be based either on estimation (Kalman [15], information [16], particle filters [17]) or optimization (bundle adjustment) frameworks [18]. The more classic landmarks are 3D points, detected as interest points (SIFT [19], SURF [20], FAST [21]), matched by using their descriptors (binary robust independent elementary features (BRIEF) [22]), or tracked by using Kanade-Lucas-Tomasi (KLT) feature tracker [23], or an active-search strategy [24]. Generally, the set of 3D points extracted from an image does not give any semantic information, unlike 3D lines, which correspond generally to sharp 3D edges in the environment. This is the reason why segment-based SLAM from either an estimation [25] or optimization [26] back-end, has been proposed. The main challenge of these methods concerns the front-end, *i.e.*, the robustness of the line detection and tracking in successive images.

The initialization of such landmarks with their minimal Euclidean parameters requires more than one observation. One way to solve this problem was the delayed initialization [3,27], in which a landmark was added to the map only when it was known in the Euclidean space. This does not allow use of landmarks that are very far from the robot. An alternative solution is to add them to the map, as soon as they are observed (*i.e.*, undelayed initialization), and it has been proposed for point [28,29] or line [25] landmarks. The pros and cons of several representations for 3D points and 3D lines have been analyzed in [30].

This article is devoted to the analysis of a visual SLAM solution using a heterogeneous map, as a more complete approach where points and lines are both included from features extracted in images acquired by a camera moving in the environment, and with undelayed initialization. Therefore, the contributions of the proposed method comprehend the use of heterogeneous landmark-based SLAM (the management of a map composed of heterogeneous landmarks), and from this perspective, the comparison between landmark parametrizations, the development of a front-end active-search process for linear landmarks integrated into SLAM (the processing of linear landmarks by the inclusion of

detection and tracking algorithms taken from the state of the art) and the experimentation methodology. These contributions correspond to the gray blocks in Figure 1. Only the first two contributions were covered in [31], which dealt with the information fusion back-end for the map computation. The present article extends this concept to its application on real images, by developing the proposed front-end. Even if optimization-based methods make it possible to avoid the possible divergence of methods based on estimation due to linearization of the observation model, the fusion is performed from an extended Kalman filter, as a very light approach that can be integrated into a dedicated architecture to be used on small aerial vehicles.

Section 2 covers the techniques and parametrizations for the initialization and update of point and line landmarks on the map. Section 3 describes the detection and tracking methods selected and developed for the front-end. Section 4 focuses on experiments, including a simulation part that compares landmark representations, a real image part that recalls the integration of segments with the already existing point-based front-end and an integral experiment. The results of the experiments are presented and discussed in Section 5. Finally, Section 6 offers some conclusions.

2. SLAM Back-End

The SLAM back-end deals with the initialization of observed features as landmarks on the map and the estimation or update of both the landmarks and the robot positions with respect to a selected reference frame. This section deepens the description of each landmark parametrization, the initialization and update algorithms and it derives from the theoretical study presented in [31].

The undelayed landmark initialization (ULI) is presented in [30] for different point and line parametrizations. The main idea is to model the uncertainty derived from unmeasured degrees of freedom by a Gaussian prior that handles unbounded limits, in a manner that can be handled by EKF.

The implications of uncertainty are different for point and line landmarks. For points, depth is initially unknown, and uncertainty is present along the visual ray until infinity. In the case of infinite straight lines, uncertainty is present in two degrees of freedom, corresponding to a depth that should be covered up to infinity and all possible orientations.

2.1. 3D Point Parametrizations

In this section, a Euclidean point is described as a reference, for subsequent covering of the ones used for initialization purposes (*i.e.*, homogeneous point and anchored homogeneous point). Each description includes camera projection and back-projection.

Point parametrizations used for initialization are shown in Figure 2.

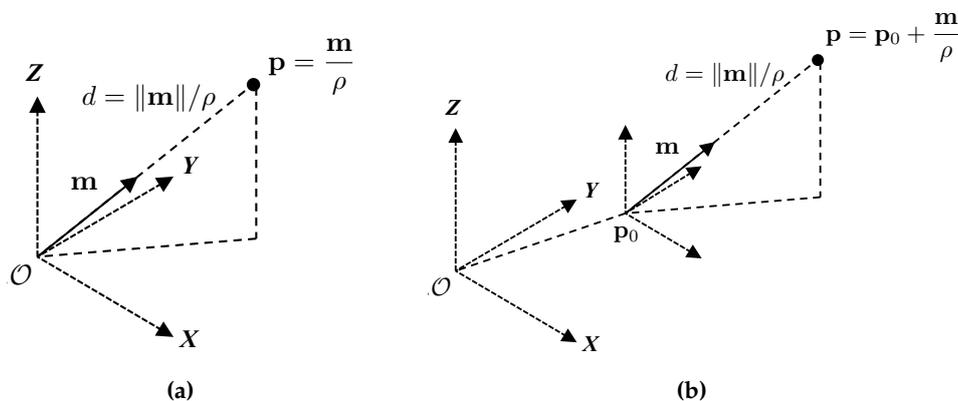


Figure 2. Point parametrizations used for initialization. (a) Homogeneous point parametrization; (b) anchored homogeneous point parametrization.

2.1.1. Euclidean Point

A Euclidean point is parametrized by its Cartesian coordinates.

$$\mathcal{L}_{EP} = \mathbf{p} = [x \ y \ z]^T \in \mathbb{R}^3$$

The projection to the camera frame is performed as follows:

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

where,

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

is the intrinsic parameter matrix and \mathbf{R} and \mathbf{T} are the rotation matrix and translation vector that define the camera C . Homogeneous coordinates are represented by underlined vectors, like $\underline{\mathbf{u}}$.

2.1.2. Homogeneous Point

Homogeneous points are four-vector composed of the 3D vector \mathbf{m} and scalar ρ , as introduced in [32].

$$\mathcal{L}_{HP} = \underline{\mathbf{p}} = \begin{bmatrix} \mathbf{m} \\ \rho \end{bmatrix} = [m_x \ m_y \ m_z \ \rho]^T \in \mathbb{P}^3 \subset \mathbb{R}^4$$

The vector \mathbf{m} gives the direction from the origin \mathcal{O} to the point \mathbf{p} , while ρ serves as a scale factor for providing the magnitude for each coordinate of the point.

The conversion from homogeneous to Euclidean coordinates is given by the following equation:

$$\mathbf{p} = \frac{\mathbf{m}}{\rho} \quad (2)$$

Depending on the characteristics of the parameters \mathbf{m} and ρ , there are three different canonical representations for a homogeneous point. The original Euclidean point refers to the case when $\rho = 1$, inverse-depth has $m_z = 1$ and in inverse-distance $\|\mathbf{m}\| = 1$.

In the camera frame, \mathbf{m} is the director vector of the optical ray, and ρ has a linear dependence on the inverse of the distance d defined from the optical center to the point,

$$\rho = \frac{\|\mathbf{m}\|}{d}.$$

The unbounded distance of a point along the optical ray from zero to infinity can then be expressed in the bounded interval in parameter space $\rho \in (0, \|\mathbf{m}\|/d_{min}]$.

The frame transformation of a homogeneous point is performed according to the next equation:

$$\underline{\mathbf{p}} = \mathbf{H}\underline{\mathbf{p}}^C = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \underline{\mathbf{p}}^C \quad (3)$$

where super-index C indicates the frame to which the point is referred and matrix \mathbf{H} specifies the frame to which the point is transformed.

The projection of a homogeneous point into the image frame is performed with the following equation:

$$\underline{\mathbf{u}} = \mathbf{K}\mathbf{R}^T (\mathbf{m} - \mathbf{T}\rho) \in \mathbb{P}^2 \quad (4)$$

By expressing a homogeneous point in the camera frame, the projected image point is $\underline{\mathbf{u}} = \mathbf{K}\mathbf{m}^C$, where super-index C indicates the frame to which the point is referred. In this case, ρ^C is not measurable. Back-projection is then:

$$\mathbf{m}^C = \mathbf{K}^{-1}\underline{\mathbf{u}}.$$

The complete homogeneous point parametrization is the following:

$$\mathcal{L}_{HP} = \underline{\mathbf{p}} = \begin{bmatrix} \mathbf{m} \\ \rho \end{bmatrix} = \mathbf{H} \begin{bmatrix} \mathbf{K}^{-1}\underline{\mathbf{u}} \\ \rho^C \end{bmatrix} = \begin{bmatrix} \mathbf{R}\mathbf{K}^{-1}\underline{\mathbf{u}} + \mathbf{T}\rho^C \\ \rho^C \end{bmatrix} \quad (5)$$

where ρ^C must be given as a prior and represents the inverse-distance from the origin of the coordinates, that is the scalar value that makes $\|\mathbf{m}\| = 1$.

2.1.3. Anchored Homogeneous Point

Linearity is supposed to be improved by the addition of an anchor that serves as a reference to the optical center at the initialization time of the landmark. The landmark is then composed of seven elements that include Cartesian coordinates of the anchor, the point with respect to the anchor and an inverse-depth scalar.

$$\mathcal{L}_{AHP} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{m} \\ \rho \end{bmatrix} = [x_0 \ y_0 \ z_0 \ m_x \ m_y \ m_z \ \rho]^T \in \mathbb{R}^7.$$

The conversion from the anchored homogeneous point to Euclidean coordinates can be obtained by the following equation:

$$\mathbf{p} = \mathbf{p}_0 + \frac{\mathbf{m}}{\rho} \quad (6)$$

The projection and frame transformation process is given below:

$$\underline{\mathbf{u}} = \mathbf{K}\mathbf{R}^T (\mathbf{m} - (\mathbf{T} - \mathbf{p}_0)\rho) \in \mathbb{P}^2 \quad (7)$$

The anchor is chosen to be the position of the optical center at the initialization time, given by \mathbf{T} . That way, the term multiplying the unmeasured degree of freedom ρ (i.e., $(\mathbf{T} - \mathbf{p}_0)\rho$) is small after initialization. This helps to decouple the uncertainty of the most uncertain parameter ρ . The complete anchored homogeneous point parametrization for back projection and transformation is the following:

$$\mathcal{L}_{AHP} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{m} \\ \rho \end{bmatrix} = \begin{bmatrix} \mathbf{T} \\ \mathbf{R}\mathbf{K}^{-1}\underline{\mathbf{u}} \\ \rho^C \end{bmatrix} \quad (8)$$

where ρ^C must be given as the prior.

2.2. 3D Line Parametrizations

The line parametrization includes the projection to the image frame and back-projection to 3D. The Plücker line and anchored homogeneous point line are shown in Figure 3.

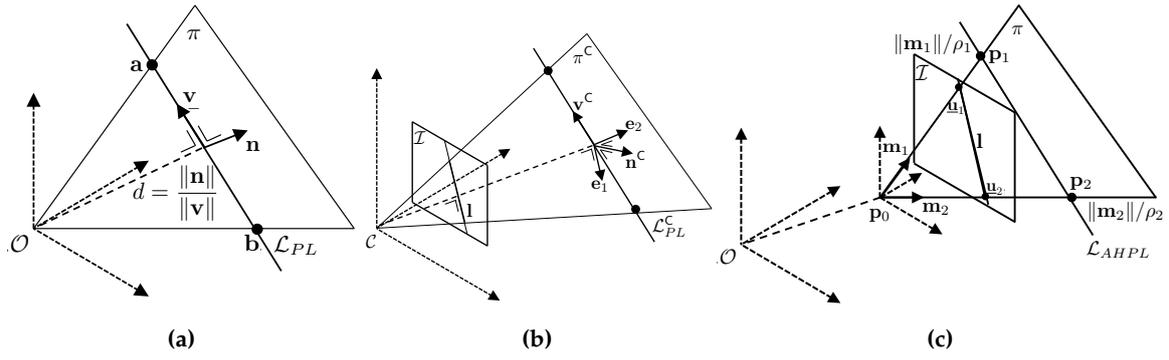


Figure 3. Line parametrizations used for initialization. (a) Plücker line geometrical representation; (b) Plücker line back-projection; (c) anchored homogeneous-points line parametrization.

2.2.1. Plücker Line

Plücker coordinates are conformed by a six-vector that represents a line in \mathbb{P}^3 defined by two points $\underline{\mathbf{a}} = [\mathbf{a} \ a]^T$ and $\underline{\mathbf{b}} = [\mathbf{b} \ b]^T$:

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

where $\mathbf{n} = \mathbf{a} \times \mathbf{b}$, $\mathbf{v} = \mathbf{a}\mathbf{b} - \mathbf{b}\mathbf{a}$, $\mathbf{n}, \mathbf{v} \in \mathbb{R}^3$, with the Plücker constraint: $\mathbf{n}^T \mathbf{v} = 0$.

In terms of geometry, \mathbf{n} is the vector normal to the plane π containing the line and the origin and \mathbf{v} is the director vector from $\underline{\mathbf{a}}$ to $\underline{\mathbf{b}}$. The Euclidean orthogonal distance from the line to the origin is given by $\|\mathbf{n}\|/\|\mathbf{v}\|$. Hence, $\|\mathbf{v}\|$ is the inverse-distance, analogous to ρ of homogeneous points. Plücker line geometrical representation is shown in Figure 3a.

Expressions for transformation and inverse-transformation of Plücker coordinates from and to the camera frame are as shown next:

$$\mathcal{L}_{PL} = \mathcal{H} \cdot \mathcal{L}_{PL}^C = \begin{bmatrix} \mathbf{R} & [\mathbf{T}]_{\times} \mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{n}^C \\ \mathbf{v}^C \end{bmatrix} \quad (9)$$

$$\mathcal{L}_{PL}^C = \mathcal{H}^{-1} \cdot \mathcal{L}_{PL} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T [\mathbf{T}]_{\times} \\ \mathbf{0} & \mathbf{R}^T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{n} \\ \mathbf{v} \end{bmatrix} \quad (10)$$

The transformation and projection process in terms of \mathbf{R} , \mathbf{T} , \mathbf{n} and \mathbf{v} is as follows:

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

where the intrinsic projection Plücker matrix \mathcal{K} is defined as:

$$\mathcal{K} = \begin{bmatrix} \alpha_v & 0 & 0 \\ 0 & \alpha_u & 0 \\ -\alpha_v u_0 & \alpha_u v_0 & \alpha_u \alpha_v \end{bmatrix}.$$

When Plücker coordinates are expressed in the camera frame, projection is obtained by:

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

The range and orientation of the line are included in \mathbf{v}^C and are not measurable.

For Plücker line back projection, vectors \mathbf{n}^C and \mathbf{v}^C are obtained with these expressions:

$$\begin{aligned}\mathbf{n}^C &= \mathcal{K}^{-1} \cdot \mathbf{l} \\ \mathbf{v}^C &= \beta_1 \cdot \mathbf{e}_1 + \beta_2 \cdot \mathbf{e}_2\end{aligned}$$

where $\beta_1, \beta_2 \in \mathbb{R}$ and $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{n}^C\}$ are mutually orthogonal.

Defining $\beta = (\beta_1, \beta_2) \in \mathbb{R}^2$, vector \mathbf{v}^C can also be expressed as:

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

where $\mathbf{v}^C \in \pi^C$ for any value of β . Plücker line back projection is shown in Figure 3b.

The complete Plücker line parametrization for back projection and transformation is given in the following equation:

$$\mathcal{L}_{PL} = \mathcal{H} \begin{bmatrix} \mathbf{n}^C \\ \mathbf{v}^C \end{bmatrix} = \mathcal{H} \begin{bmatrix} \mathcal{K}^{-1} \mathbf{l} \\ \mathbf{E} \beta \end{bmatrix} = \begin{bmatrix} \mathbf{R} \mathcal{K}^{-1} \mathbf{l} + \mathbf{T} \times \mathbf{R} \mathbf{E} \beta \\ \mathbf{R} \mathbf{E} \beta \end{bmatrix} \quad (13)$$

where β must be provided as a prior.

2.2.2. Anchored Homogeneous Points Line

A line can also be represented by the end points defining it. With the application of the anchored homogeneous point parametrization, shown in Figure 3c, an anchored homogeneous-points line is an eleven-vector defined as follows:

$$\mathcal{L}_{AHPL} = [\mathbf{p}_0 \ \mathbf{m}_1 \ \rho_1 \ \mathbf{m}_2 \ \rho_2]^T \in \mathbb{R}^{11}$$

For each point, the transformation and projection of a pinhole camera is as previously stated in Equation (4).

A homogeneous 2D line is obtained by the cross product of two points lying on it, $\mathbf{l} = \mathbf{u}_1 \times \mathbf{u}_2$, giving:

$$\mathbf{l} = \mathcal{K} \mathbf{R}^T ((\mathbf{m}_1 \times \mathbf{m}_2) - (\mathbf{T} - \mathbf{p}_0) \times (\rho_1 \mathbf{m}_2 - \rho_2 \mathbf{m}_1)) \quad (14)$$

In comparison to the result obtained for Plücker coordinates, the product $\mathbf{m}_1 \times \mathbf{m}_2$ is a vector orthogonal to the plane π , analogous to the Plücker sub-vector \mathbf{n} . Furthermore, the term $(\rho_1 \mathbf{m}_2 - \rho_2 \mathbf{m}_1)$ is a vector that gives the direction between the points of the line, therefore related to Plücker sub-vector \mathbf{v} .

The complete anchored homogeneous point line parametrization for back projection and transformation is the following:

$$\mathcal{L}_{AHPL} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{m}_1 \\ \rho_1 \\ \mathbf{m}_2 \\ \rho_2 \end{bmatrix} = \begin{bmatrix} \mathbf{T} \\ \mathbf{R} \mathcal{K}^{-1} \mathbf{u}_1 \\ \rho_1^C \\ \mathbf{R} \mathcal{K}^{-1} \mathbf{u}_2 \\ \rho_2^C \end{bmatrix} \quad (15)$$

where ρ_1^C and ρ_2^C must be given as priors.

2.3. Landmark Initialization

The process of the initialization of a landmark consists of the detection of a feature in the image, retro-projection to 3D and inclusion into the map. There are three important concepts that are involved in the landmark initialization: the 3D landmark itself \mathbf{x} , the 2D measurement of the landmark in the

image \mathbf{z} and the unmeasured degree of freedom π . All of these are modeled as Gaussian variables, whose notation is $\mathcal{N}\{\mu, \sigma^2\}$. Thus, the cited concepts are expressed as $\mathbf{x} \sim \mathcal{N}\{\bar{\mathbf{x}}, \mathbf{P}\}$, $\mathbf{z} \sim \mathcal{N}\{\bar{\mathbf{z}}, \mathbf{R}\}$ and $\pi \sim \mathcal{N}\{\bar{\pi}, \Pi\}$, respectively. The 3D landmarks \mathbf{x} considered for this study are points and lines, already described in Sections 2.1 and 2.2. The parametrizations for landmark 2D measurements \mathbf{z} and unmeasured degrees of freedom π , as well as the description of the initialization algorithm are covered in the following sections.

2.3.1. Landmark 2D Measurements in the Image

Points are represented as a two-vector containing Cartesian coordinates in pixel space, leading to the following:

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

where \mathbf{U} is the covariance matrix of the position of the point.

In homogeneous coordinates,

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

Lines can be expressed by a four-vector that represents the coordinates of their end points, also with a Gaussian probability density function.

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

The probability density function for infinite lines like Plücker, $\mathcal{N}\{\bar{\mathbf{I}}, \mathbf{L}\}$, is composed of the homogeneous line representation and the covariance matrix defined as follows:

$$\begin{aligned} \bar{\mathbf{I}} &= \bar{\mathbf{u}}_1 \times \bar{\mathbf{u}}_2, \text{ and} \\ \mathbf{L} &= [\bar{\mathbf{u}}_1]_{\times} \mathbf{U} [\bar{\mathbf{u}}_1]_{\times}^T + [\bar{\mathbf{u}}_2]_{\times} \mathbf{U} [\bar{\mathbf{u}}_2]_{\times}^T \end{aligned}$$

2.3.2. Unmeasured Degrees of Freedom

The uncertainty in 3D points and lines coming from projection is represented by inverse-distance variables ρ^C and β^C , which are modeled as Gaussian variables. The origin of each of these priors must be inside the 2σ of their probability density functions.

For points and end-point-based lines, the minimum distance must match the upper 2σ boundary, hence:

$$\begin{aligned} \rho - n\sigma_{\rho} &= 0, & 0 \leq n < 2 \\ \rho + 2\sigma_{\rho} &= 1/d_{\min} \end{aligned}$$

Then, $n = 1$ leads to,

$$\bar{\rho} = 1/3d_{\min} \text{ and } \sigma_{\rho} = 1/3d_{\min}$$

The probability density function of a point based line is defined as $\mathbf{t}^C \sim \mathcal{N}\{\bar{\mathbf{t}}; \mathbf{T}\}$, with:

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

The Plücker lines prior $\beta^C \sim \mathcal{N}\{\bar{\beta}; \mathbf{B}\}$ take the following values:

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

This initializes lines at the front of the camera.

2.3.3. Undelayed Landmark Initialization Algorithm

The ULI algorithm was presented in [30] for the construction of landmark-based stochastic maps, including a single type of landmark \mathcal{L} . The approach presented in this paper includes heterogeneous parametrizations of landmarks \mathcal{L}_* on the same map, where \mathcal{L}_* can be a point or line. The resulting algorithm is composed of the following steps:

1. Identify mapped magnitudes $\mathbf{x} \sim \mathcal{N}\{\bar{\mathbf{x}}, \mathbf{P}\}$.
2. Identify measurements $\mathbf{z} \sim \mathcal{N}\{\bar{\mathbf{z}}, \mathbf{R}\}$, where \mathbf{z} is either a point or a line (*i.e.*, \mathbf{u} or \mathbf{s} , respectively).
3. Define Gaussian prior $\pi \sim \mathcal{N}\{\bar{\pi}; \Pi\}$ for the unmeasured degree of freedom. π can either be ρ^C , \mathbf{t}^C or β^C .
4. Back-project the Gaussian measurement, and get the landmark mean and Jacobians.

$$\bar{\mathcal{L}}_* = g(\bar{\mathbf{C}}, \bar{\mathbf{z}}, \bar{\pi})$$

$$\mathbf{G}_C = \left. \frac{dg}{d\mathbf{C}} \right|_{\bar{\mathbf{C}}, \bar{\mathbf{z}}, \bar{\pi}}, \mathbf{G}_z = \left. \frac{dg}{d\mathbf{z}} \right|_{\bar{\mathbf{C}}, \bar{\mathbf{z}}, \bar{\pi}}, \mathbf{G}_\pi = \left. \frac{dg}{d\pi} \right|_{\bar{\mathbf{C}}, \bar{\mathbf{z}}, \bar{\pi}},$$

where $g()$ is the back projection and transformation function for the corresponding landmark. $\mathbf{C} = (\mathbf{T}, \mathbf{Q})$ is the camera frame expressed in terms of its position \mathbf{T} and orientation \mathbf{Q} in quaternion nomenclature.

5. Compute landmarks' co- and cross-variances.

$$\mathbf{P}_{\mathcal{L}_* \mathcal{L}_*} = \mathbf{G}_C \mathbf{P}_{CC} \mathbf{G}_C^T + \mathbf{G}_z \mathbf{R} \mathbf{G}_z^T + \mathbf{G}_\pi \Pi \mathbf{G}_\pi^T$$

$$\mathbf{P}_{\mathcal{L}_* \mathbf{x}} = \mathbf{G}_C \mathbf{P}_{C\mathbf{x}} = \mathbf{G}_C [\mathbf{P}_{CC} \mathbf{P}_{CM}]$$

6. Augment the SLAM map.

$$\bar{\mathbf{x}} \leftarrow \begin{bmatrix} \bar{\mathbf{x}} \\ \mathcal{L}_* \end{bmatrix}$$

$$\mathbf{P} \leftarrow \begin{bmatrix} \mathbf{P} & \mathbf{P}_{\mathcal{L}_* \mathbf{x}}^T \\ \mathbf{P}_{\mathcal{L}_* \mathbf{x}} & \mathbf{P}_{\mathcal{L}_* \mathcal{L}_*} \end{bmatrix}$$

2.4. Landmark Update

The purpose of the landmark update process is to recalculate the parameters of the elements on the map, (*i.e.*, the robot and landmark poses), given the observation of the already mapped landmarks in the current frame. This process starts by projecting all of the observable landmarks to the image plane and selecting those with higher uncertainty for correction. For points, the observation function $h()$ applies a homogeneous to Euclidean transformation $h2e()$ once having performed the projection process previously explained, as follows:

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

Innovation mean \mathbf{y} and covariance \mathbf{Y} is then obtained as shown next:

$$\mathbf{y} = \mathbf{z} - h(\bar{\mathbf{x}})$$

$$\mathbf{Y} = \mathbf{R} + \mathbf{H} \cdot \mathbf{P} \cdot \mathbf{H}^T,$$

where $\mathbf{R} = \mathbf{U}$ is the measurement noise covariance and Jacobian $\mathbf{H} = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}}$.

For lines, the innovation function computes the orthogonal distances from the detected end-points \mathbf{u}_i to a line \mathbf{l} , as shown in Figure 4, leading to the following:

$$\mathbf{z} = \begin{bmatrix} \mathbf{l}^T \cdot \mathbf{u}_1 / \sqrt{l_1^2 + l_2^2} \\ \mathbf{l}^T \cdot \mathbf{u}_2 / \sqrt{l_1^2 + l_2^2} \end{bmatrix} \in \mathbb{R}^2 \quad (16)$$

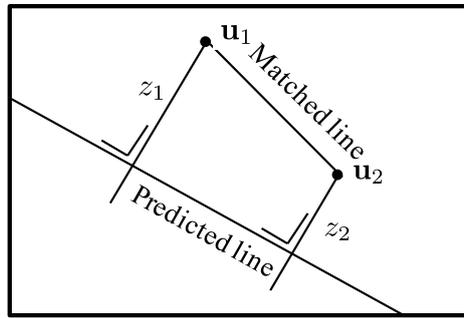


Figure 4. Measurement of the orthogonal distances from the detected end-points to the expected (or predicted) line.

Since the EKF innovation is the difference between the actual measurement and the expectation and \mathbf{z} is the orthogonal distance previously described, the line innovation function is:

$$\mathbf{y} = \mathbf{0} - \mathbf{z},$$

as the desired orthogonal distance from the predicted line to the matched end-points is zero.

A landmark is found consistent if the squared Mahalanobis distance $MD2$ of innovation is smaller than a threshold $MD2th$.

$$MD2 = \mathbf{y}^T \cdot \mathbf{Y}^{-1} \cdot \mathbf{y} < MD2th$$

As that is true, the landmark is updated:

$$\text{Kalman gain: } \mathbf{K} = \mathbf{P} \cdot \mathbf{H} \cdot \mathbf{Y}^{-1}$$

$$\text{State update: } \bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}} + \mathbf{K} \cdot \mathbf{y}$$

$$\text{Covariance update: } \mathbf{P} \leftarrow \mathbf{P} - \mathbf{K} \cdot \mathbf{H} \cdot \mathbf{P}$$

Point and line parametrizations are modeled as Gaussian variables in [25,29,30], validating the use of Mahalanobis distance as compared to a chi-squared distribution. Kalman gain is assumed to be optimal. Since this process is intended to be developed as a light approach that could be integrated into a dedicated architecture on small vehicles, the selected covariance update formula is used instead of the Joseph form, which has such a high complexity that it may compromise performance. Successful results of this formulation are presented in [30].

3. SLAM Front-End

To obtain the geometrical representation of landmarks given by the SLAM back-end, it is necessary to process the information coming from the sensors embedded in the moving agent (*i.e.*, cameras mounted on the mobile robot). The front-end deals with the detection of new landmarks and the matching of already existent ones in subsequent images.

This section covers the image processing algorithms used for detecting and matching points and lines. Points have been widely studied and implemented as SLAM landmarks [3,24,33,34]. In the case of lines, a different front-end strategy was integrated for the detection and tracking of line segments.

3.1. Point Landmarks

A point landmark is modeled as an appearance descriptor composed of a patch of pixels around the point in the image. Once detected, the patch is used for the matching of the feature on incoming images.

3.1.1. Point Detection

An active-search approach [24,34] can ensure that the point landmarks are equally distributed in the image by dividing it into a number of equal regions, in which it is expected to have a landmark (Figure 5a). At each iteration, an empty region is randomly selected, and a corner point is chosen to be the strongest Harris point [35] (Figure 5b). This point is used for the landmark initialization, and its appearance and the current position and orientation of the camera $C_0 = (T_0, R_0)$ are saved. The appearance of the point is given by the patch of pixels surrounding it, as seen in Figure 5c.

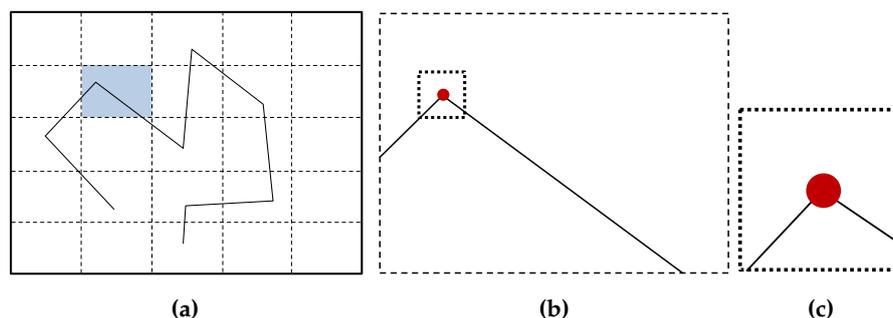


Figure 5. Point detection process. (a) Image divided into regions; one is randomly selected; (b) selected region, processed to find a corner point; (c) appearance of the point.

3.1.2. Point Matching

When there are point landmarks already mapped, the matching process searches for a point landmark x in the frame captured at camera pose C_i . This point has been initialized in the frame captured with camera pose C_0 (Figure 6a,b).

The saved appearance patch of the landmark is warped by applying a homography transformation. This transformation takes into account the rotation and translation in the camera position and orientation, with respect to its pose when the landmark was detected. The transformed coordinates of pixel j of the patch at camera pose i (*i.e.*, \mathbf{q}_i^j) is computed as follows:

$$\mathbf{q}_i^j = H\mathbf{q}_0^j \quad (17)$$

where:

$$H = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \mathbf{K}^{-1}$$

and \mathbf{q}_0^j are the coordinates of pixel j of the original patch, as \mathbf{R} and \mathbf{T} are the rotation matrix and translation vector between camera pose C_0 and camera pose C_i . Once the patch is warped, it is cropped to be squared in order to maintain the same dimensions of the original. This warping process is shown in Figure 6c.

The matching process performs the projection of point landmark \mathbf{x} into the image at camera pose C_i to get a point expected position \mathbf{u}_i , given the current pose i of the camera:

$$\mathbf{u}_i = h(C_i, \mathbf{x}).$$

The 2D covariance matrix \mathbf{U} of this point is obtained from the 3D covariance matrix $\mathbf{P}_{\mathcal{L}\mathcal{L}}$ corresponding to landmark \mathcal{L} , as follows:

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_{\text{RF}} & \mathbf{U}_{\text{SF}} & \mathbf{U}_{\mathcal{L}} \end{bmatrix} \mathbf{P}_{\mathcal{L}\mathcal{L}} \begin{bmatrix} \mathbf{U}_{\text{RF}} & \mathbf{U}_{\text{SF}} & \mathbf{U}_{\mathcal{L}} \end{bmatrix}^T$$

where \mathbf{U}_{RF} , \mathbf{U}_{SF} and $\mathbf{U}_{\mathcal{L}}$ are the Jacobians of the projection \mathbf{u} with respect to the robot frame, the sensor frame and the landmark, respectively.

Then, the **zero mean normalized cross-correlation (ZNCC)** test [36] is applied to the warped patch and a region of pixels surrounding the expected point in the image (Figure 6d). The rectangular search region is based on the projection mean \mathbf{u} and the covariance ellipse \mathbf{U} . The mean is the center of the search box, and the square roots of the diagonals of the covariance are the standard deviations, σ_u and σ_v . The search region goes $\pm 3\sigma$ at each side of the center. If the ZNCC score is over a threshold, the point is said to be matched.

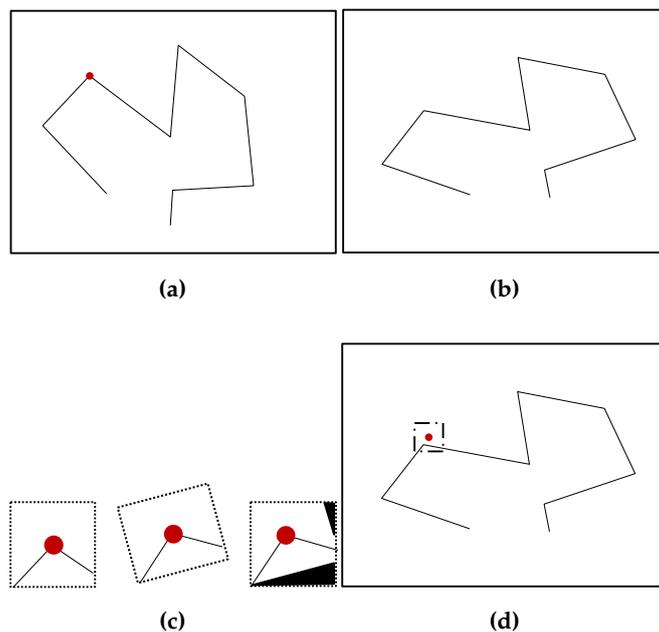


Figure 6. Point matching process. (a) Frame captured at camera pose C_0 , where point \mathbf{x} was initialized; (b) frame captured at camera pose C_i , where point \mathbf{x} is intended to be matched; (c) patch warping process, showing the original, the transformed and the cropped patches; (d) projection of landmark point \mathbf{x} into image point \mathbf{u}_i on the image at camera pose C_i . The search area is indicated by the rectangle surrounding the point.

3.2. Line landmarks

Many methods have been proposed to extract lines in the image processing community, generally starting from the detection of intensity discontinuities (gradient, Laplacian, Canny filter). The first method, introduced in the 1980s, was the chaining method [37], based on the polygonal approximation of extracted contours. This method is efficient, but the result is too dependent on its parametrizations (gradient threshold, contour thinning). This is why the Hough transform became so popular [38]; a recent variant, the kernel-based Hough transform [39], has been implemented and evaluated for line detection. However, this approach, when working with infinite lines rather than segments, makes its performance less than optimal for the intended purposes. In [40], the Dseg algorithm is proposed, close to the chaining method, but using an iterative filtering approach to integrate a contour point in the processed segment. The Dseg algorithm was compared to the chaining method, with the Hough transform and with the line segment detector (LSD) detector [41]; it was found that it allows extraction of a greater number of segments of various lengths.

This section covers the description of a front-end line segment active-search process, developed for segment-based SLAM. The selected techniques for working with segments are the LSD for the detection and moving edges (ME) [42] for matching.

An active-search approach was developed and implemented in order to handle the line segment landmarks, similar to the one previously described for points.

3.2.1. Segment Detection

The process starts by building a grid that divides the image into rectangular cells. A 3×3 grid was chosen, as shown in Figure 7a. There are two different ways of detecting lines: the one applied in the first frame of the sequence, and the one used in all other frames. In the first case, the segment detection algorithm is run for the whole image, and the longest segments found are selected. The cells containing a whole or partial segment are marked as “occupied”. This is shown in Figure 7b. The detection process, applied in a subsequent frame (Figure 7c), departs from the assumption that there are line landmarks already on the map. Once the EKF back-end computes the 3D position of the robot and the landmarks seen so far, landmarks are projected to the image. The projection of line landmark x into the image at camera pose C_i to get a line segment’s expected position s_i , given the current pose i of the camera, is performed as follows:

$$s_i = h(C_i, x) \quad (18)$$

Each projection is taken into account for updating the grid. The occupied cells are not considered, and one empty cell is chosen randomly. The image patch delimited by this cell is used to run the segment detector and to find the longest segment on it for initializing a new landmark. The line detected is extended to the other cells, and they are marked as “occupied” when this is the case. This process is shown in Figure 7d.

The patch of the cells where lines were detected in the present frame is saved for the line matching process, as well as the current position and orientation of the camera $C_0 = (T_0, R_0)$. Each line is defined by its end points, and only the pixels surrounding it are used for matching, as can be seen in Figure 7e.

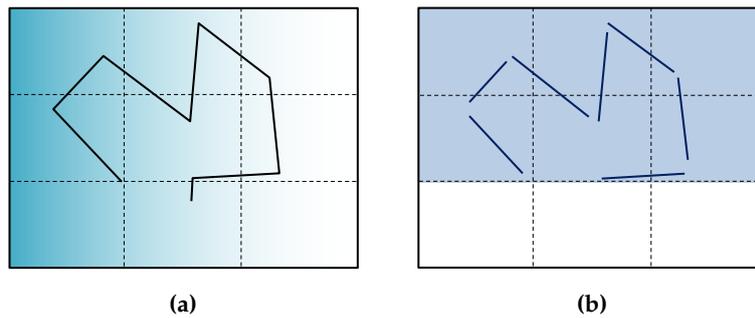


Figure 7. Cont.

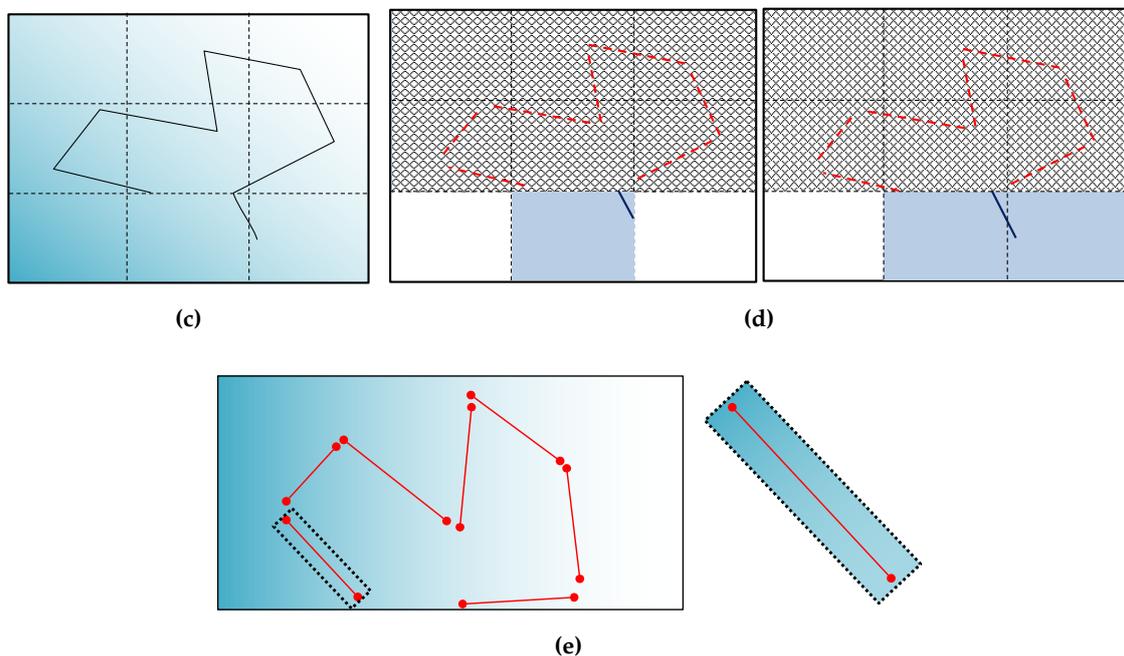


Figure 7. Line detection process. (a) Initial image divided in a 3×3 grid of cells. (b) line segments detected. Only the ones that are longer than a threshold are selected. The cells that contain them are marked as “occupied”. (c) Later image of the sequence. The map already contains line landmarks. (d) Existing landmarks are projected into the image, and the occupation grid is updated. Occupied grids are marked with a textured pattern. The first image also shows the randomly-selected empty cell and the segment detected on it. The second image shows the result of extending the detected segment. Cells occupied by this segment are colored in blue. (e) Appearance of the detected lines. Each line is parametrized by its end points, the pixels surrounding the line are used for matching.

3.2.2. Segment Matching

When there are line landmarks already mapped, the matching process searches for a line landmark x in the frame captured at camera pose C_i . This line has been initialized in the frame captured with camera pose C_0 (Figure 8a,b).

The saved appearance patch of the landmark is warped by applying a homography transformation. This transformation takes into account the rotation and translation in the camera position and orientation, with respect to its pose when the landmark was detected. The transformed coordinates

of pixel j of the patch at camera pose i (i.e., \mathbf{q}_i^j) are computed applying Equation (17). This warping process is shown in Figure 8c.

The matching process performs the projection of the line landmark from Equation (18) to estimate its position of line segment \mathbf{s}_i in the image. For updating the current position of each line landmark, their estimated position in the image is used for initializing the matching algorithm, and a search for a match is performed in the region surrounding the line estimation (Figure 8d). The errors between the estimated position and the position of the match found are used by the back-end to update the state and uncertainties of the landmarks and robot poses on the map.

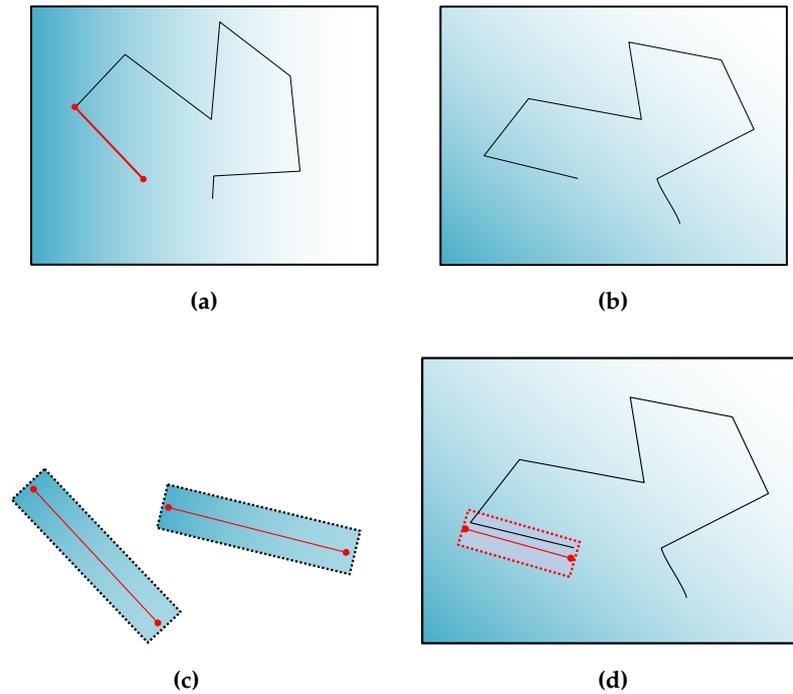


Figure 8. Line matching process. (a) Frame captured at camera pose C_0 , where line x was initialized; (b) frame captured at camera pose C_i , where line x is intended to be matched; (c) patch warping process, showing the original and the transformed patches; (d) projection of landmark line x into image segment \mathbf{s}_i on the image at camera pose C_i . The search area is indicated by the rectangle surrounding the segment.

To perform the tracking points that make up part of line segments, the moving edges algorithm is implemented as discussed in [43].

The algorithm consists of searching the correspondent point p^{t+1} on line $\mathbf{l}(\mathbf{r})^{t+1}$ in image I^{t+1} of point p^t in line $\mathbf{l}(\mathbf{r})^t$. The search for a match is performed in the direction normal to the line $\mathbf{l}(\mathbf{r})^t$, given by δ . For each point p^t , a search interval $\{Q_j, j \in [-J, J]\}$ is defined. Each sample Q_j is evaluated by the criterion ζ_j . This evaluation consists in computing the convolution value between an image patch at Q_j , and the mask M_δ , which is a function of the orientation of line $\mathbf{l}(\mathbf{r})^t$. The algorithm is shown in Figure 9.

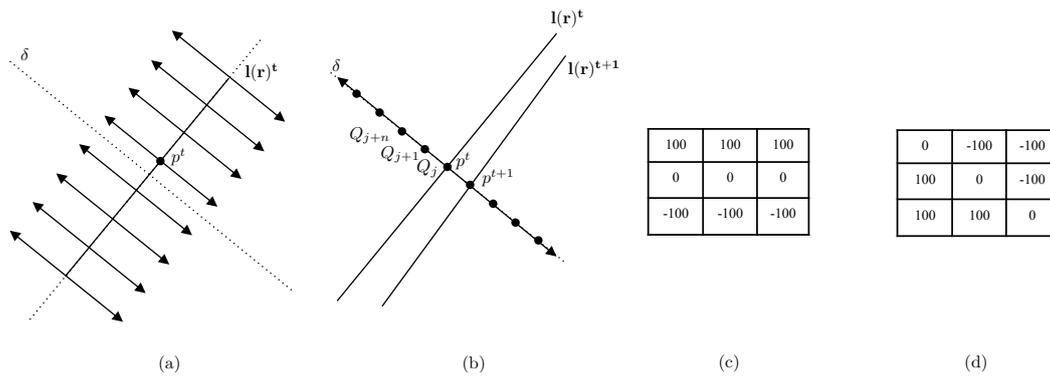


Figure 9. Moving edges (ME) algorithm to find correspondent points in image sequences. **(a)** Calculation of the direction δ normal to the line $I(r)^t$; **(b)** search sampling $\{Q_j, j \in [-J, J]\}$ along the normal direction; **(c)** M_δ mask at 180° and **(d)** M_δ mask at 45°.

Thus, the position of point p^{t+1} on line $I(r)^{t+1}$ in image I^{t+1} is given by:

$$Q_{j^*} = \arg \max_{j \in [-J, J]} \zeta_j \text{ with } \zeta_j = \left| I_{v(p^t)}^t * M_\delta + I_{v(Q_j)}^{t+1} * M_\delta \right| \quad (19)$$

A list of k points is produced, from which the segment extremities $\mathbf{s} = (\mathbf{u}_1, \mathbf{u}_2)$ are extracted.

One way to express the measurement \mathbf{z} of the matched segment with respect to the expected prediction of it is to compute the orthogonal distance of the matched end points \mathbf{u}_1 and \mathbf{u}_2 to the predicted line \mathbf{l} , as shown in Equation (16) and Figure 4.

By defining line measurement in this way, the matching can be accomplished regardless of which points of the corresponding line were detected by the tracker and of the segment length.

4. Experiments

This section includes the experimental part that tests the three main contributions of this article. The first part deals with the back-end and consists of a comparative evaluation between different landmark parametrizations. A set of simulations tests the benefits of the combination of point and line landmarks in the same map. The following part deals with the implementation of the developed segment-based SLAM front-end that includes the line segment active-search process presented in this paper. Finally, a complete heterogeneous landmark-based SLAM experiment that integrates the contributions to back-end and front-end is included.

4.1. Simulation of the Back-End for Heterogeneous SLAM

The point and line parametrizations previously presented have been tested independently in previous studies, such as [25,29,30]. This section offers a comparison of different heterogeneous approaches, including combinations of distinct landmarks on the same map. The purpose is to show the benefits of working with a heterogeneous parametrization that combines points and lines in a single map. The combinations performed are enumerated below:

1. Anchored homogeneous point (AHP)
2. Plücker line (PL)
3. Anchored homogeneous-points line (AHPL)
4. AHP + PL
5. AHP + AHPL

The MATLAB® EKF-SLAM toolbox [44] was extended with the heterogeneous functionality to perform the simulations.

Figure 10a shows the simulation environment. It consists of a house conformed by 23 lines and an array of 16 points distributed uniformly among the walls.

The robot performs two different trajectories. The first one is a circular path of 5 m in diameter, with a pose step of 8 cm and 0.09° . The second is a motion of 70 steps of 4 cm, each taken towards the scene. The linear noise is 0.5 cm and the angular noise 0.05° .

Besides the heterogeneous landmark capability of the toolbox, the transparency of the objects in the scene was also considered. By default, objects in the simulation environment of the toolbox are transparent, so landmarks are visible on almost every image frame. To work in a more realistic manner, an aspect graph was implemented to only observe visible surfaces of the house at each camera pose.

Both transparent and opaque object visualizations are shown in Figure 11. An example of a heterogeneous map constructed after a complete turn of the robot around the house is shown in Figure 10b. The parametrization used is AHP + PL. It displays in green the line landmarks estimated and in blue the point landmarks. Real, predicted and estimated robot trajectories are displayed in blue, red and green, respectively.

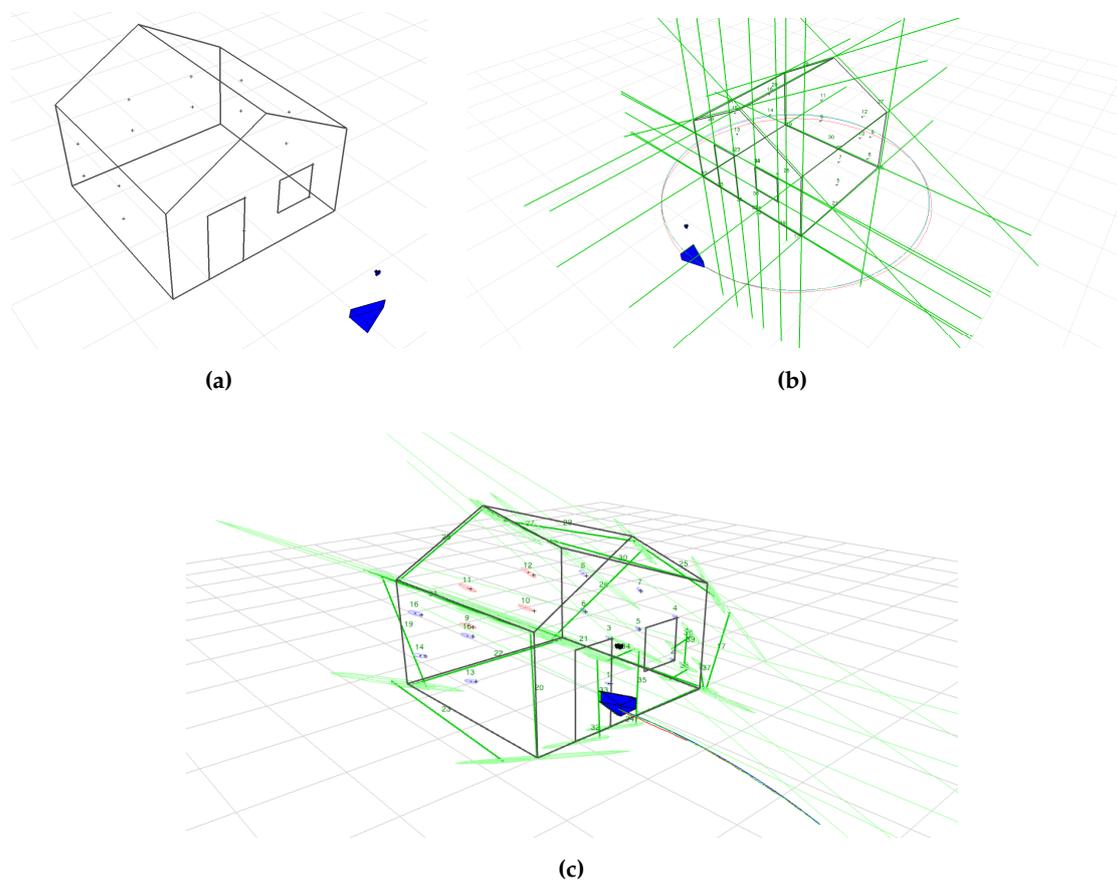


Figure 10. Environment world used for the simulation experiments of heterogeneous point and line SLAM. (a) Initial state of the environment world; (b) environment after performing circular trajectory with anchored homogeneous point (AHP) + Plücker line (PL) parametrization; (c) environment after performing approaching trajectory with AHP + AHPL parametrization.

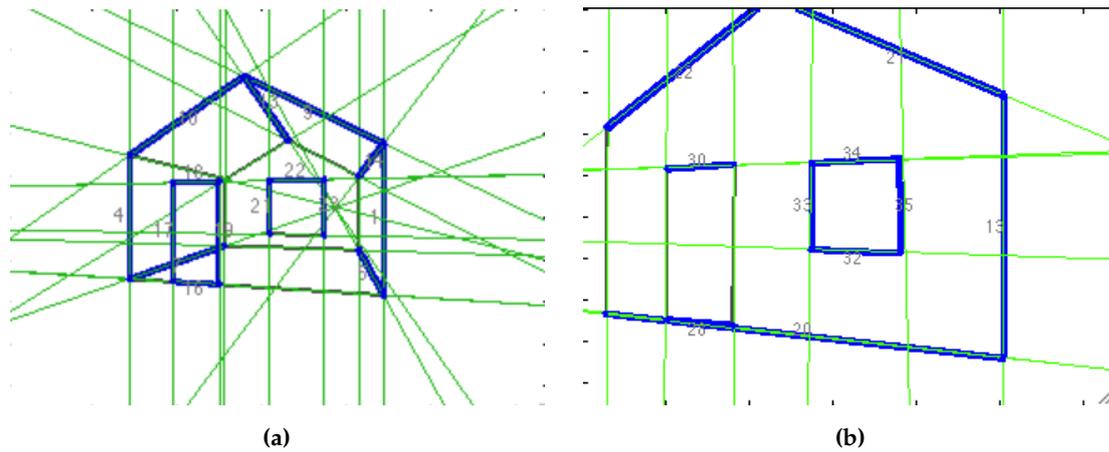


Figure 11. Different visualization modes of objects in the simulation environment. (a) Sensor view considering transparent objects; (b) sensor view considering opaque objects.

For the circular path, a trajectory of five turns, considering transparent and opaque objects, was performed for each parametrization.

4.2. Integration of Line Segment Active-Search to the SLAM Front-End

The line-based SLAM front-end that was developed and that implements the line segment active-search presented in this article is covered in this section. The LSD and ME algorithms were applied to an image sequence showing a piece of furniture inside a room.

Figure 12 shows the operation of this segment-based front-end of an EKF-SLAM process. Infinite thin lines represent the estimated position of the landmark in the current image, while thicker segments show the match found. It can be observed that the match corresponds to the estimation in most of the cases.

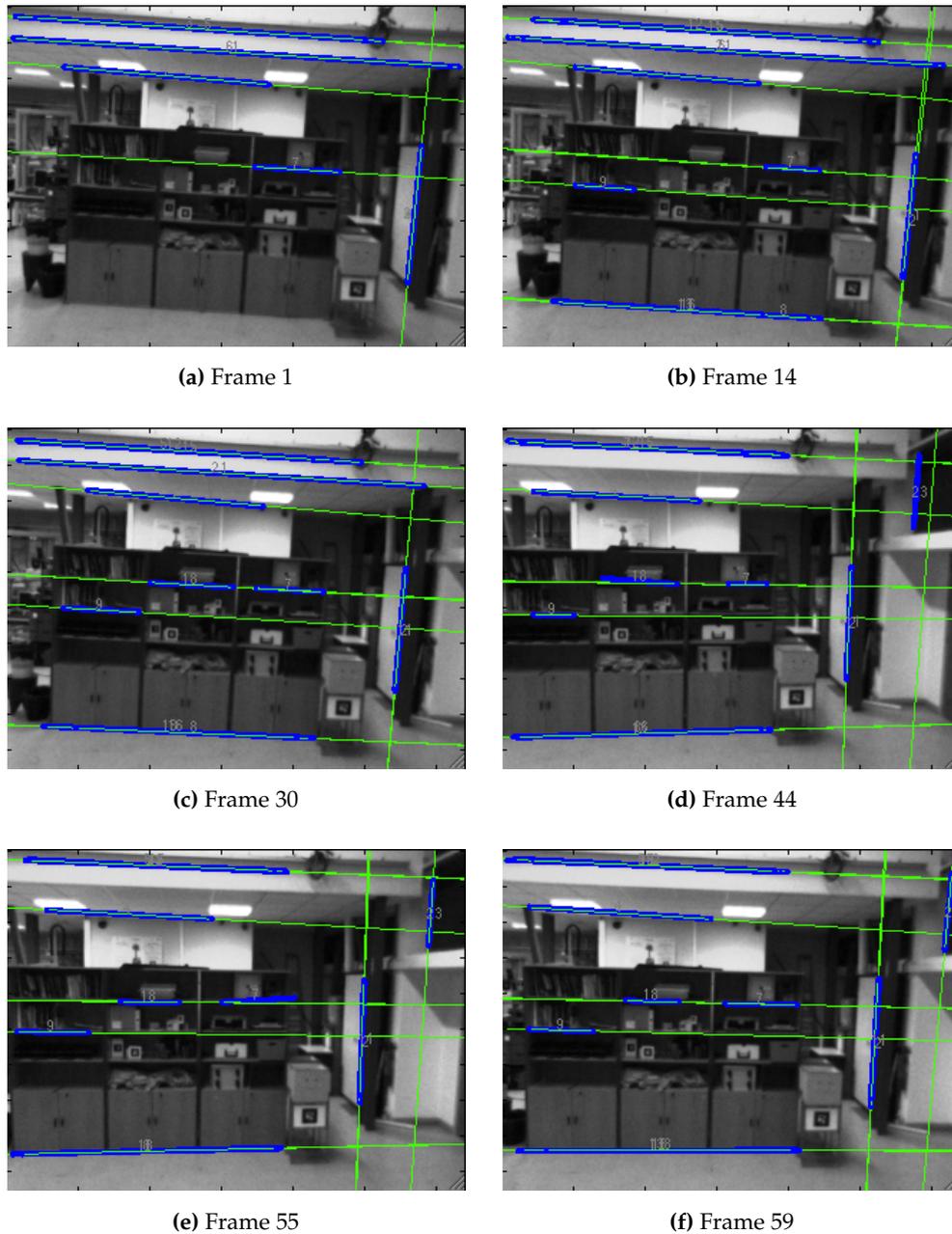


Figure 12. Front end operation of segment-based EKF-SLAM in an indoor sequence, showing the prediction and observation of landmarks. Infinite thin lines represent the estimated position of the landmark in the current image, while thicker segments show the match found.

4.3. Heterogeneous SLAM Experiment

The complete heterogeneous SLAM solution was tested with the experiment described in this section.

The mobile robot used was a CRS Robotics F3 system, with a Microsoft LifeCam Studio camera mounted on it, along with a 9DOF Razor IMU that provided the robot state estimation at each frame. A total of 401 images with a 1280×720 resolution form the sequence. From this robot, it was possible to get the ground truth information with a repeatability of ± 0.05 mm. The robot described a total trajectory of 0.4653 m. To get a prediction of the motion, the information provided by the IMU was

used in a constant acceleration motion model. The inclusion of this additional sensor made it possible to cope with the inherent scale ambiguity of monocular systems.

Figure 13 presents certain frames of the sequence, showing the landmarks used to update the state of the map. The AHP and AHPL were the parametrizations selected for the experiments, as they were the ones that provided better simulation results.

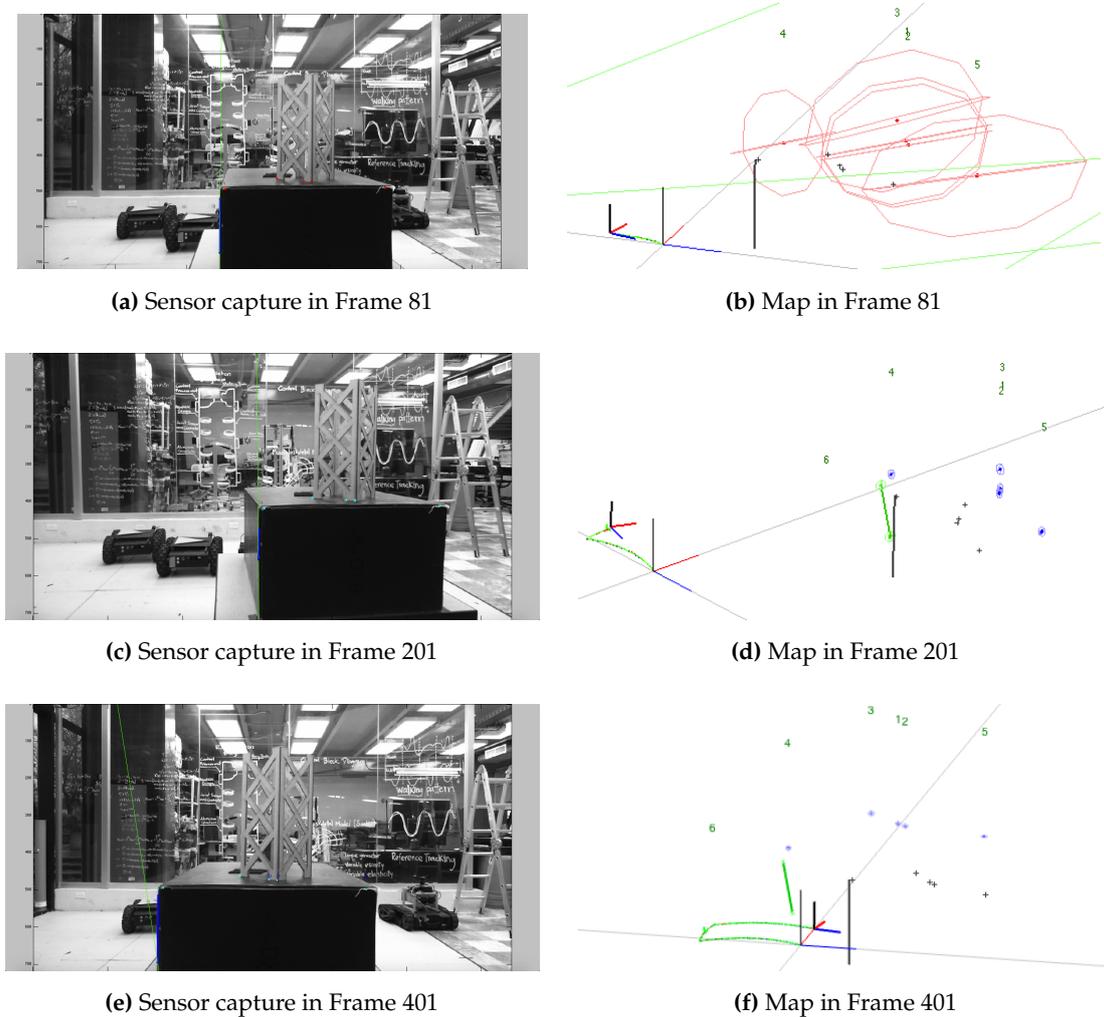


Figure 13. Point and line heterogeneous EKF-SLAM frames. (a,c,e) The front-end part; infinite thin lines represent the estimated position of the landmark in the current image, while thicker segments show the match found; circles represent the prediction of points, and points themselves are the match found. (b,d,f) The maps created on each frame, with the landmarks' estimation, the current pose of the camera represented by the small reference frame and the trajectory described.

5. Results and Discussion

For the analysis, root mean square error (RMSE) evaluation is used to compute errors. At each instant k , the estimated position (x_k, y_k, z_k) is compared to the ground truth position $(\hat{x}_k, \hat{y}_k, \hat{z}_k)$.

$$\epsilon_k = \sqrt{((x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2 + (z_k - \hat{z}_k)^2)} \quad (20)$$

From the previous results, the mean and standard deviation of the error are computed as follows:

$$\mu_{\epsilon} = \frac{1}{N} \sum_{k=1}^N \epsilon_k \quad (21)$$

$$\sigma_{\epsilon} = \sqrt{\frac{1}{N} \sum_{k=1}^N (\epsilon_k - \mu_{\epsilon})^2} \quad (22)$$

Simulation of the back-end for heterogeneous SLAM was intended to compare the different parametrizations and to show the benefits of landmark heterogeneity.

The parametrization with the highest error is the Plücker line. Anchored parametrizations achieved the best performance, for both points and lines. There is an improvement effect in line parametrizations by the addition of points. Even for the anchored cases, already having a relative good performance while working independently, the heterogeneity improves the results, in such a way that the combination of both AHP and AHPL is the one with the least error along the simulated trajectories.

The position error of the robot in the case of a circular trajectory with transparent and opaque objects is shown in Figure 14; these results are summarized in Table 1. For the case of the approaching trajectory, the results are shown and summarized in Figure 15 and Table 2.

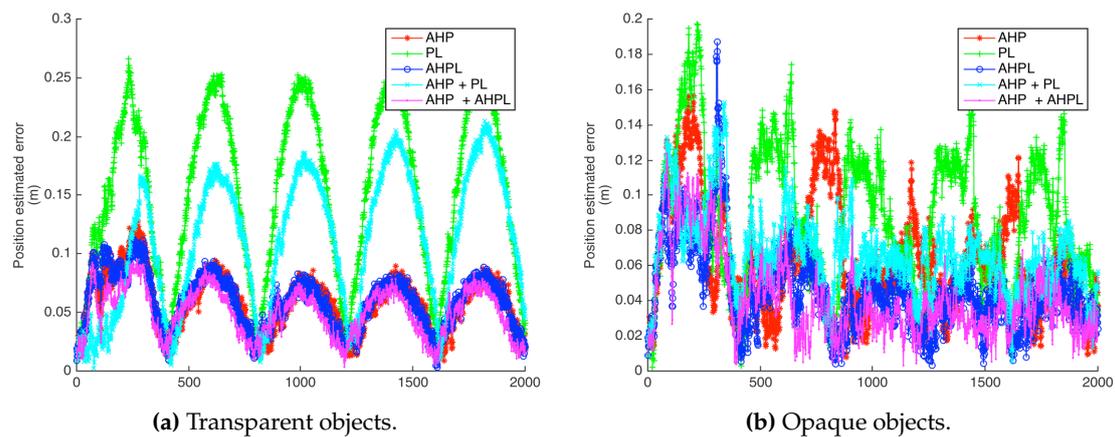


Figure 14. Robot position estimation errors for the circular trajectory.

Table 1. Robot position error mean and standard deviation, for transparent and opaque object visualization, for the different heterogeneous landmark parametrizations in the circular trajectory in the simulated environment.

		AHP	PL	AHPL	AHP + PL	AHPL + AHPL
Mean	Transparent objects	0.05856	0.1532	0.05861	0.1109	0.04982
	Opaque objects	0.06332	0.08806	0.04561	0.06634	0.04114
Standard Deviation	Transparent objects	0.02426	0.07139	0.02268	0.05733	0.02025
	Opaque objects	0.0316	0.03961	0.0236	0.02298	0.02368

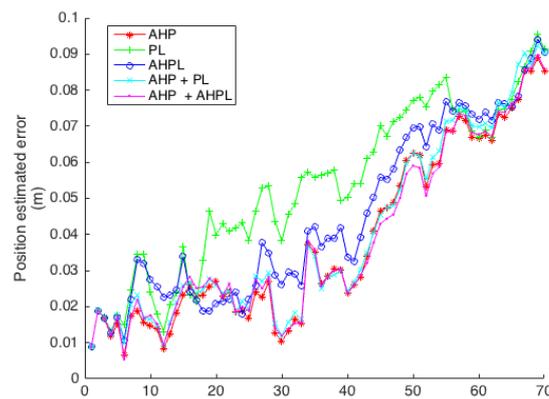


Figure 15. Robot position estimation errors for the approach trajectory.

Table 2. Robot position error mean and standard deviation for the different heterogeneous landmark parametrizations in the approach trajectory in the simulated environment.

	AHP	PL	AHPL	AHP + PL	AHP + AHPL
Mean	0.038	0.053	0.044	0.040	0.038
Standard Deviation	0.024	0.023	0.024	0.025	0.023

The complete SLAM experiment integrating the back-end with the developed front-end is used to compare the heterogeneous approach with a classic point-based SLAM applied to the same sequence. The ground truth and estimated trajectories for each SLAM approach tested are shown in Figure 16.

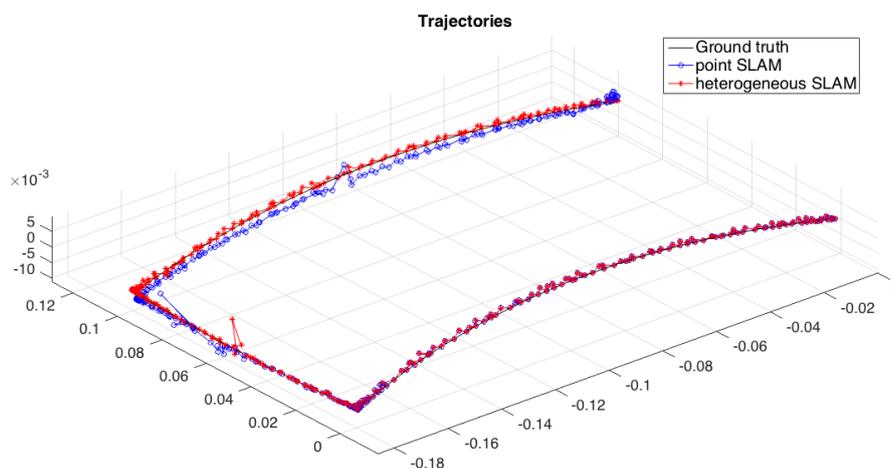


Figure 16. Ground truth, point SLAM estimated and heterogeneous SLAM estimated trajectories.

Figure 17 presents results in terms of the robot position estimation error, comparing the IMU estimation to both point and heterogeneous SLAM. As can be observed, the heterogeneous approach results in lower errors, as previously presented in the simulation part. Near Frame 200, there was a change in the motion direction of the robot, which can be seen as a peak in the position estimation error graph. Even in this case, heterogeneous SLAM achieved a better performance than the point-based SLAM. Table 3 shows a comparative summary of the errors from the three cases.

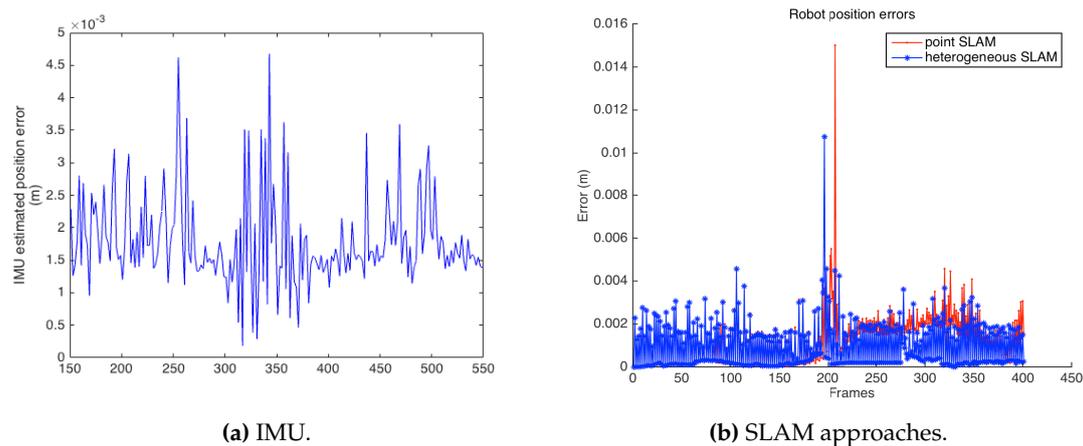


Figure 17. Robot position estimation error comparison between IMU, point SLAM and heterogeneous SLAM approaches.

Table 3. Robot position error mean, maximum, minimum and standard deviation, for IMU estimation point and heterogeneous SLAM solutions in the real environment.

	Mean	Max	Min	SD
IMU	0.0018	0.0047	0.0002	0.0007
Point SLAM	0.0011	0.1500	0.0000	0.0012
Heterogeneous SLAM	0.0008	0.0107	0.0000	0.0011

6. Conclusions

The purpose of this paper is to prove the benefits of including heterogeneous landmarks when building a map from an EKF-based visual SLAM method. Several authors have shown interest in the use of heterogeneous landmarks. For the front-end, the interest for joint tracking of points and lines is found in [45], while for the back-end, a theoretical study is presented in [31], and preliminary results of an EKF-based SLAM method, based on heterogeneous landmarks, are presented in [46]. The experiments performed that the authors describe have shown that the robot localization or the SLAM stability can be improved by combining several landmarks, *i.e.*, points and lines. The use of just monocular vision provides only partial observations of landmarks by features extracted from images; here, undelayed initialization of landmarks is used, as was proposed initially by Solá *et al.* [25,29] for points and lines. The use of simulated data has shown how the choice of the landmark representation has an impact on the accuracy of the map. The best ones, considering the construction of a map with heterogeneous landmarks, are anchored homogeneous points and anchored homogeneous points lines. These parametrizations were used in a complete heterogeneous SLAM experiment that produced better results than the classic point-based case, by reducing the camera position estimation error.

Another contribution of this paper is a method proposed for a segment-based SLAM front-end. This method relies on the line segment active-search process presented in this article and on state-of-the-art line detection and matching processes. The methods that compose the developed front-end that resulted were discussed, recalling, first, their theoretical background and, then, presenting some experimental evaluations on image sequences that show the stability of the process.

Finally, a complete heterogeneous landmark-based SLAM experiment was presented, integrating the contributions with the back-end and the front-end and confirming the results obtained independently.

In future work, constraints will be exploited in the map, typically when points and lines are extracted from known portions of the scene.

Acknowledgments: This study was performed during a long-term stay of Jorge Othón Esparza-Jiménez, PhD student in Tecnológico de Monterrey, Mexico, in the Robot, Action and Perception team of LAAS-CNRS in France. The authors would like to thank Mexico's National Council of Science and Technology (Consejo Nacional de Ciencia y Tecnología (CONACyT)), the Focus Group on Robotics at Tecnológico de Monterrey and the Laboratorio de Robótica del Área Noreste y Centro de México (Robotics Laboratory of Northeastern and Central Mexico) founded by CONACyT-Tecnológico de Monterrey, for supporting this research. This study contributes to the French FUI project AIR-COBOT. The authors also acknowledge Susan Fortenbaugh for the English language revision of the paper.

Author Contributions: The authors contributed equally to the development of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Carmigniani, J.; Furht, B.; Anisetti, M.; Ceravolo, P.; Damiani, E.; Ivkovic, M. Augmented reality technologies, systems and applications. *Multimedia Tools Appl.* **2011**, *51*, 341–377.
2. Chekhlov, D.; Gee, A.P.; Calway, A.; Mayol-Cuevas, W. Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 1–4.
3. Davison, A.J. **Real-time simultaneous localisation and mapping with a single camera.** In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; pp. 1403–1410.
4. Dow, J.M.; Neilan, R.; Rizos, C. The international GNSS service in a changing landscape of global navigation satellite systems. *J. Geodesy* **2009**, *83*, 191–198.
5. Zhang, C.; Kuhn, M.; Merkl, B.; Fathy, A.E.; Mahfouz, M. Accurate UWB indoor localization system utilizing time difference of arrival approach. In Proceedings of the 2006 IEEE Radio and Wireless Symposium, San Diego, CA, USA, 17–19 October 2006.
6. Lim, C.H.; Wan, Y.; Ng, B.P.; See, C.M.S. A real-time indoor WiFi localization system utilizing smart antennas. *IEEE Trans. Consum. Electron.* **2007**, *53*, 618–622.
7. Bulusu, N.; Heidemann, J.; Estrin, D. GPS-less low-cost outdoor localization for very small devices. *IEEE Pers. Commun.* **2000**, *7*, 28–34.
8. Yi, J.; Zhang, J.; Song, D.; Jayasuriya, S. IMU-based localization and slip estimation for skid-steered mobile robots. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA, 29 October–2 November 2007; pp. 2845–2850.
9. Karras, G.C.; Kyriakopoulos, K.J. Localization of an underwater vehicle using an IMU and a laser-based vision system. In Proceedings of the IEEE Mediterranean Conference on Control & Automation (MED'07), Athens, Greece, 21–24 June 2007; pp. 1–6.
10. Ndjeng, A.N.; Gruyer, D.; Glaser, S.; Lambert, A. Low cost IMU–Odometer–GPS ego localization for unusual maneuvers. *Inf. Fusion* **2011**, *12*, 264–274.
11. Surmann, H.; Nüchter, A.; Hertzberg, J. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robot. Auton. Syst.* **2003**, *45*, 181–198.
12. Hussmann, S.; Liepert, T. Robot vision system based on a 3D-ToF camera. In Proceedings of the IEEE Instrumentation and Measurement Technology Conference Proceedings (IMTC 2007), Warsaw, Poland, 1–3 May 2007; pp. 1–5.
13. Endres, F.; Hess, J.; Engelhard, N.; Sturm, J.; Cremers, D.; Burgard, W. An evaluation of the RGB-D SLAM system. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA), St. Paul, MN, USA, 14–18 May 2012; pp. 1691–1696.
14. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007), Nara, Japan, 13–16 November 2007; pp. 225–234.
15. Dissanayake, M.; Newman, P.; Clark, S.; Durrant-Whyte, H.F.; Csorba, M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Autom.* **2001**, *17*, 229–241.
16. Walter, M.R.; Eustice, R.M.; Leonard, J.J. Exactly sparse extended information filters for feature-based SLAM. *Int. J. Robot. Res.* **2007**, *26*, 335–359.

17. Sim, R.; Elinas, P.; Griffin, M.; Little, J.J. Vision-based SLAM using the Rao-Blackwellised particle filter. In Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics, Edinburgh, UK, 30 July–5 August 2005; Volume 14, pp. 9–16.
18. Strasdat, H.; Montiel, J.; Davison, A.J. Real-time monocular slam: Why filter? In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, Alaska, 3–8 May 2010; pp. 2657–2664.
19. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157.
20. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In *Computer vision—ECCV 2006*; Springer: Berlin, Germany, 2006; pp. 404–417.
21. Rosten, E.; Drummond, T. Machine learning for high-speed corner detection. In *Computer Vision—ECCV 2006*; Springer: Berlin, Germany, 2006; pp. 430–443.
22. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. Brief: Binary robust independent elementary features. In *Computer Vision—ECCV 2010*; Springer: Berlin, Germany, 2010; pp. 778–792.
23. Tomasi, C.; Kanade, T. *Detection and Tracking of Point Features*; Technical Report CMU-CS-91-132; Carnegie Mellon Univ.: Pittsburgh, PA, USA, April 1991.
24. Davison, A.J. Active search for real-time vision. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV 2005), Beijing, China, 17–20 October 2005; Volume 1, pp. 66–73.
25. Sola, J.; Vidal-Calleja, T.; Devy, M. Undelayed initialization of line segments in monocular SLAM. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009), St. Louis, MO, USA, 11–15 October 2009; pp. 1553–1558.
26. Lee, J.H.; Zhang, G.; Lim, J.; Suh, I.H. Place recognition using straight lines for vision-based SLAM. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 3799–3806.
27. Lemaire, T.; Lacroix, S.; Sola, J. A practical 3D bearing-only SLAM algorithm. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, AB, Canada, 2–6 August 2005; pp. 2449–2454.
28. Montiel, J.M.M. Unified inverse depth parametrization for monocular SLAM. In Proceedings of the Robotics: Science and Systems (RSS), Philadelphia, PA, USA, 16–19 August 2006.
29. Sola, J.; Monin, A.; Devy, M.; Lemaire, T. Undelayed initialization in bearing only SLAM. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Edmonton, AB, Canada, 2–6 August 2005.
30. Sola, J.; Vidal-Calleja, T.; Civera, J.; Montiel, J.M.M. Impact of landmark parametrization on monocular EKF-SLAM with points and lines. *Int. J. Comput. Vis.* **2012**, *97*, 339–368.
31. Esparza-Jimenez, J.O.; Devy, M.; Gordillo, J. EKF-based SLAM fusing heterogeneous landmarks. In Proceedings of the 2014 17th International Conference on Information Fusion (FUSION), Salamanca, Spain, 7–10 July 2014; pp. 1–8.
32. Smith, D. *History of Modern Mathematics*; Cornell University Library Historical Math Monographs; John Wiley & Sons: Hoboken, NJ, USA, 1906.
33. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067.
34. Civera, J.; Grasa, O.G.; Davison, A.J.; Montiel, J. 1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry. *J. Field Robot.* **2010**, *27*, 609–631.
35. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988.
36. Di Stefano, L.; Mattocchia, S.; Tombari, F. ZNCC-based template matching using bounded partial correlation. *Pattern Recognit. Lett.* **2005**, *26*, 2129–2134.
37. Etemadi, A. Robust segmentation of edge data. In Proceedings of the IET International Conference on Image Processing and its Applications, Maastricht, The Netherlands, 7–9 April 1992; pp. 311–314.
38. Ballard, D.H. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognit.* **1981**, *13*, 111–122.

39. Fernandes, L.A.; Oliveira, M.M. Real-time line detection through an improved Hough transform voting scheme. *Pattern Recognit.* **2008**, *41*, 299–314.
40. Berger, C.; Lacroix, S. *DSeg: Direct Line Segments Detection*; Technical Report; LAAS/CNRS: Toulouse, France, September 2009.
41. Von Gioi, R.G.; Jakubowicz, J.; Morel, J.M.; Randall, G. LSD: A line segment detector. *Image Process. Line* **2012**, *2*, 35–55.
42. Bouthemy, P. A maximum likelihood framework for determining moving edges. *IEEE Trans. Pattern Anal. Mach. Intell.* **1989**, *11*, 499–511.
43. Comport, A.I.; Marchand, E.; Pressigout, M.; Chaumette, F. **Real-time markerless tracking for augmented reality: the virtual visual servoing framework.** *IEEE Transaction on Visualization and Computer Graphics.* **2006**, *12*, 615–628.
44. Sola, J.; Marquez, D.; Codol, J.; Vidal-Calleja, T. An EKF-SLAM toolbox for MATLAB. Available online: <https://github.com/joansola/slamtb> (accessed on September 24, 2013).
45. Rosten, E.; Drummond, T. Fusing points and lines for high performance tracking. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV 2005), Beijing, China, 17–21 October 2005; Volume 2, pp. 1508–1515.
46. Vidal, T.; Berger, C.; Sola, J.; Lacroix, S. Environment modeling for cooperative aerial/ground robotic systems. In *Robotics Research*; Springer: Berlin, Germany, 2011; pp. 681–696.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons by Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).