



# Large scale multiple robot visual mapping with heterogeneous landmarks in semi-structured terrain

Teresa Vidal-Calleja, Cyrille Berger, Joan Solà, Simon Lacroix

## ► To cite this version:

Teresa Vidal-Calleja, Cyrille Berger, Joan Solà, Simon Lacroix. Large scale multiple robot visual mapping with heterogeneous landmarks in semi-structured terrain. *Robotics and Autonomous Systems*, 2011, 59 (9), pp.654-674. 10.1016/j.robot.2011.05.008 . hal-04037082

**HAL Id: hal-04037082**

**<https://laas.hal.science/hal-04037082>**

Submitted on 20 Mar 2023

**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.



Distributed under a Creative Commons Attribution 4.0 International License

# Large Scale Multiple Robot Visual Mapping with Heterogeneous Landmarks in Semi-structured Terrain

Teresa A. Vidal-Calleja<sup>a</sup>, t.vidal@acfr.usyd.edu.au,  
Cyrille Berger<sup>b</sup>, cberger@laas.fr,  
Joan Solà<sup>b</sup>, jsola@laas.fr,  
Simon Lacroix<sup>b</sup>, Simon.Lacroix@laas.fr

<sup>a</sup> Australian Centre for Field Robotics  
University of Sydney, NSW 2006, Australia

<sup>b</sup> CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France  
Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France

---

## Abstract

This paper addresses the cooperative localization and visual mapping problem with multiple heterogeneous robots. The approach is designed to deal with the challenging large semi-structured outdoors environments in which aerial/ground ensembles are to evolve. We propose the use of heterogeneous visual landmarks, points and line segments, to achieve effective cooperation in such environments. A large-scale SLAM algorithm is generalized to handle multiple robots, in which a global graph maintains the relative relationships between a series of local sub-maps built by the different robots. The key issue when dealing with multiple robots is to find the link between them, and to integrate these relations to maintain the overall geometric consistency: the events that introduce these links on the global graph are described in detail. Monocular cameras are considered as the primary exteroceptive sensor. In order to achieve undelayed initialization required by the bearing-only observations, the well-known inverse-depth parametrization is adopted to estimate 3D points. Similarly, to estimate 3D line segments, we present a novel parametrization based on anchored Plücker coordinates, to which extensible endpoints are added. Extensive simulations show the proposed developments, and the overall approach is illustrated using real-data taken with a helicopter and a ground rover.

**Keywords:** multi-robots cooperation, visual SLAM

---

## 1. Introduction

There are several cooperation schemes in which the complementarity of aerial and ground robots can be exploited to enhance the efficiency of autonomous robotics operations. For instance, aerial robots can provide the ground robots with information related to the environment, *e.g.* traversability maps or landmarks, they can localize the ground robots by perceiving them, or provide them communication links with a remote station. Aerial and ground robots can also cooperate in a more symmetric way to achieve a task, such as exploration, surveillance or target detection and tracking. All these tasks can benefit from the enhanced observation capabilities brought by both kinds of robots.

In this context, the ability to build and share environment models among the robots is an essential prerequisite to the development of cooperation schemes. Be it for exploration, surveillance or intervention missions, environment models are indeed necessary to plan and coordinate paths, but also to determine the utility of vantage points, to assess whether robots will be able to communicate or not, and to localize the robots in a common frame. In particular, 3D information on the environment is required: not only the robots evolve in the three dimensions, but the determination of vantage points calls for visibility computations in the 3D space. Also, vision is here the primary sensor to build environment representations: besides the fact that images carry a lot of information on the environment, vision is passive, it has the main advantage to perceive features that are arbitrarily far away, and it can be embedded on-board any kind of drone, even the smallest ones.

Similarly to the single robot case, one can not take for granted the fact that the robots are at all times perfectly



Figure 1: Three aerial images of a village acquired at about 40m altitude (top) and three images of the same area acquired by a ground rover. The red angular sector in the aerial images approximately represents the position and field of view of the ground rover camera. How could these data be registered and fused in a spatially consistent data structure?

localized. Centimeter accuracy RTK-GPS is often disturbed by signal outages<sup>1</sup>, and no on-board sensors can ensure a robust long term localization. Nevertheless, the maps must be *spatially consistent*: their spatial uncertainties must be compatible with the actual environment’s spatial characteristics, and the process of mapping must tolerate the fact that the robots positions are not always perfectly known. This naturally leads to exploit a SLAM approach, in which uncertainties on the environment features and on the robots poses are explicitly managed over time.

But mapping in a multi-robot system brings forth the following additional issues, that have to be tackled in our aerial/ground context:

- The mapping algorithms must be distributed among the robots. They can not rely on a complete and permanent communication infrastructure between the robots, as this would impose strong requirements on communication coverage and bandwidth, which can hardly be satisfied in most field applications.
- The structure of the maps built by the various robots must allow to match, register or fuse them, so that they can be shared among the robots. For these purposes, the information on the environment they exhibit must be common to the various robots. In our context the 3D geometry of the environment is the only *intrinsic environment characteristic* on which one can rely to tackle data association, as it can cope with the fact that the sensors or viewpoints can be very different among the different robots (see example data in Figure 1).

*Approach.* To tackle these two issues, we aim at defining a *3D multiple robot mapping framework* using vision, that is able to handle *any localization means*, including matching perceived data with pre-existing 3D maps. The produced environment representation must also allow the incorporation of range data such as those provided by stereo vision or Lidars, these sensors being widely used on ground robots. For these purposes, we propose to use 3D points and 3D line segments to represent the environment from sequences of aerial and ground images. These representations should gather most of the environment geometric structure, and is the basis to register and fuse the data acquired from different vantage points, different sensors, and prior 3D information on the environment.

In the direction towards the achievement of such a mapping framework, the contribution of this paper is twofold:

---

<sup>1</sup>some applications contexts even preclude its usage.

- We propose a mapping approach in which robots build series of sub-maps using a classical EKF-based SLAM paradigm. The overall spatial consistency of the maps among the robots is ensured by an optimization process, that takes into account various inter-robot and absolute localization estimates. This approach takes after the work on hierarchical SLAM proposed by Estrada *et al* in [1], which relies on a hierarchical representation of the map: the global level is an adjacency graph, where nodes are local maps (or “sub-maps”), and edges are relative locations between local maps. In a multi-robot context, the sub-maps are not necessarily built in a sequential manner, and various events can trigger loop closures and later map merging, namely rendezvous between robots, feature correspondences (“matching”) and absolute localizations, provided either by GPS fixes or by matches with an a priori existing map [2]. These events produce loop closures: the global graph exhibits a cycle, and thus define constraints that allow the system to refine the estimates of the sub-maps origins. We analyze the impact of these events on the overall map structure and propose a way to handle them in a distributed context.
- We exploit 3D points and 3D line segments detected in the images in order to create a environment representation that is invariant to the vantage points. This environment representation is built using bearing-only information. Bearing-only information calls for a dedicated implementation of the EKF-based SLAM approach, which relies on the undelayed initialization. We use the well-known inverse-depth parametrization (IDP) for 3D points and we propose a new parametrization based on anchored Plücker 3D lines. Plücker coordinates are well known in computer vision for their nice projective properties, similar to what happens with homogeneous points. To improve their behavior within EKF-SLAM, we add an anchor to the parametrization as it is done in IDP.

*Outline.* Section 2 deals with the overall localization and mapping approach to deal with multiple robots. It presents the principle of hierarchical SLAM by means of a probabilistic graph model, and its extension in the multi-robot case, analyzing in particular the various loop closures that can occur between the robots. It also discusses how the mapping process could be distributed among the robots, and simulation results are depicted. Section 3 is devoted to the line segments detected in monocular imagery. It introduces the way such segments can be used as landmarks in a monocular EKF SLAM approach: the anchored Plücker line parametrization is used, the initialization and update processes are depicted, the map-matching with lines is presented, and simulation results are analyzed. Finally, Section 4 presents results obtained with data gathered by a helicopter and a ground rover in a village-like environment, and a discussion concludes the paper.

## 2. Managing multiple maps among multiple robots

### 2.1. Related work

The multi-robot SLAM problem has recently gained a lot of attention. Several approaches have been presented to solve this problem: based on extended Kalman filters [3, 4, 5], information filters [6, 7], maximum likelihood estimators [8, 9] and particle filters [10, 11].

Early work on cooperative localization and mapping extended the well-studied single robot SLAM problem to multiple robots. This extension is straightforward when the robots start from a known location and the information is centralized. In [8] a maximum likelihood algorithm is applied to find the maps that are maximally consistent with sensor data and odometry. They assume the robots start from a known location and a central mapper improves the map by iteratively combining data from the robots. Ref. [10] also assume known location to fuse maps from different robots combining a maximum likelihood with an a-posteriori pose estimator. The robots poses are propagated using a particle filter representation of their belief functions, while the map is compiled in a distributed manner among robots using laser range data. In [3], the authors do not address implementation issues but the theoretically attainable estimation accuracy, by providing bounds for the steady state covariance of the position estimates.

Further developments have been achieved for the case when the robots do not have prior knowledge about their relative locations. In [6] a sparse extended information filter is presented with no knowledge of initial locations. The alignment of local maps into a single global map is achieved by a tree-based algorithm for searching similar-looking local landmark configurations, paired with a hill climbing algorithm that maximizes the overall likelihood by searching in the space of correspondences. Ref. [12] deals with completely unknown start locations by using an adapted particle

filter in combination with a predictive model of indoor environments in order to sequentially determine whether and how the partial maps of two robots overlap.

The authors in [4] proposed an algorithm to align maps built by different vehicles that not necessarily overlap. They used the information provided by the robot-to-robot measurements (rendezvous) to obtain the transformation between their coordinate frames. They used an Extended Kalman Filter (EKF) to estimate the robots and landmarks positions. For maps that do overlap, [5] presents an incremental minimum cycle basis algorithm that combined with hierarchical SLAM [1] manages the global level after a map merging is performed. The work of [13] uses a maximum likelihood approach for merging maps with initial unknown correspondences with manifold operations and scan matching. This work subdivides the multi-robot problem into three sub-problems: incremental localization and mapping, loop closure and island merging. This latter work is related to our work in the sense that we also consider this subdivision, although our approach relies on events such as rendezvous, landmark correspondences or GPS fixes to first trigger a loop closure in global level and at the end if required do the map merging.

The theoretical solution for the centralized multi-robot SLAM algorithm is achieved by appending robots together with landmarks to the same state-space, using the same filter. The practical solution requires a central station where all the information is concentrated [8, 10]. Some efforts have been done in terms of decentralization, where robots broadcast the gathered information and, independently, each robot computes its own pose and full map. A decentralized SLAM approach was proposed by [7] using channel filters. In order to reduce the communication bandwidth the authors proposed the use of the covariance intersection algorithm. A interesting distributed SLAM approach is presented in [14], where communication is limited to an upper trapezoidal matrix which condenses the entire measurement history on the individual robots. More feasible algorithms can achieve distributed mapping using cooperative localization [15, 16]. Distributed mapping is accomplished by the fact that multiple robots build their own maps. Our solution is related with cooperative localization in the sense that robots only communicate some past robot poses (origins of local maps) to each other.

Range and bearing observation characterized most of the approaches in the literature for multi-robot SLAM [8, 10, 12, 17]. The work in [18] is one of the very few work on multi-robot SLAM using vision. The multi-robot terrain mapping algorithm uses localization information to combine vision-based range estimates with an elevation profile across the terrain. Also, in [11] the proposed approach considers the use of a stereo vision system, where again range and bearing observations are acquired. It relies on a fully centralized Rao-Blackwellized particle filter, and only presents simulation results.

## 2.2. Hierarchical SLAM

Different scalable SLAM approaches in which a single vehicle builds multiple local maps have been proposed, mainly to reduce computational complexity and to delay linearization errors until the map merging [19, 20, 1, 21, 22]. When the maps are merged into a single one, on the basis of either common landmarks between local maps, or simply the sequential constraint, a fully correlated map of the environment is obtained. Successful fast implementations exploiting the topology of the representation to systematically join and fuse local map have been proposed for the single vehicle case, such as trees [23], or binary trees [21].

We formulate the multiple robot localization and mapping problem using sub-maps, in a similar manner to hierarchical SLAM in [1] or hybrid metric-topological SLAM in [22], where there are two levels: local level (sub-maps), and global level (adjacency graph). We make use of the probabilistic graphical models representation to describe our approach.

The global level represents the relationships  $\mathbf{s}_i^j$  between local maps  $i$  and  $j$ . The local level contains the sub-maps, composed of the set of landmarks  $\mathbf{m}^i$  and the current robot pose  $\mathbf{x}_k^i$  (at instant  $k$ ). At a certain point a new local map is generated with the robot pose acting as the new local reference frame (*lrf*). Thus the robot pose  $\mathbf{x}^i$  truly represents the relation between the previous map and the new one, and one can set  $\mathbf{s}_{i+1}^i = \mathbf{x}^i$ . Other non-correlative relations  $\mathbf{s}_i^j$  may be established between maps as we will see. Based on simple frame compositions, information in the world reference frame (*wrf*) is also available for the origins  $\mathbf{S}_i$  of each map, and for the map itself  $(\mathbf{X}, \mathbf{M})^i$  if it is required. The Bayesian network in Figure 2 shows the representation of this hierarchical/hybrid SLAM.

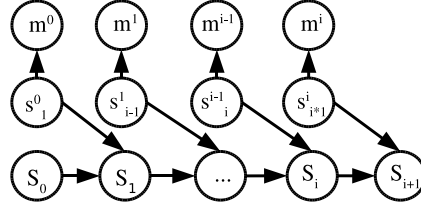


Figure 2: Graphical model as a simplified Bayesian network that represents the multiple local sub-maps. Sub-maps constitute the local level ( $\mathbf{x}^i, \mathbf{m}^i$ ) and the global level on top links the local sub-maps ( $\mathbf{S}_i$ ). In this representation there is no information shared between local sub-maps. The map building here is sequential, corresponding to one robot exploring and not closing loops, having always  $\mathbf{s}_{i+1}^i = \mathbf{x}^i$ . Loop closures will establish non-sequential links  $\mathbf{s}_i^j$  that are not present in this figure (see Figs. 5–7 for examples).

*Local Level.* The local level contains the feature-based locally referred stochastic maps, built with the standard EKF-SLAM. The  $i$ -th local map is defined by

$$\mathbf{x}_m^i = \begin{bmatrix} \mathbf{x}^i \\ \mathbf{m}^i \end{bmatrix}, \quad (1)$$

where  $\mathbf{x}^i$  is the current pose of the robot, and  $\mathbf{m}^i = [\mathbf{l}_1^i \dots \mathbf{l}_m^i]^\top$  is the set of  $m$  mapped landmarks, both with respect to the  $i$ -th *lrf*. EKF-SLAM keeps a Gaussian estimate  $\mathbf{x}_m^i \sim \mathcal{N}\{\hat{\mathbf{x}}_m^i, \mathbf{P}_m^i\}$  of this map, namely

$$\hat{\mathbf{x}}_m^i = \begin{bmatrix} \hat{\mathbf{x}}^i \\ \hat{\mathbf{m}}^i \end{bmatrix}, \quad \mathbf{P}_m^i = \begin{bmatrix} \mathbf{P}_{\mathbf{x}^i \mathbf{x}^i} & \mathbf{P}_{\mathbf{x}^i \mathbf{m}^i} \\ (\mathbf{P}_{\mathbf{x}^i \mathbf{m}^i})^\top & \mathbf{P}_{\mathbf{m}^i \mathbf{m}^i} \end{bmatrix}. \quad (2)$$

The maps are built sequentially as mentioned above. Once a threshold is passed, either in number of landmarks or in robot's uncertainty, a new map is created. No information is shared between these sub-maps, thus the new map starts in a *lrf* with the robot's pose and error covariances equal to zero. Each local map stores information in its own *lrf*.

We assume the control input of the robot and landmark's observation have been already introduced to obtain the local sub-maps. The joint distribution resulting of the local map SLAM process is  $p(\mathbf{x}^i, \mathbf{m}^i)$ . Thus the  $\mathbf{m}^i$  conditional probability, as shown in Figure 2, is given by,

$$p(\mathbf{m}^i | \mathbf{x}^i) = \frac{p(\mathbf{x}^i, \mathbf{m}^i)}{p(\mathbf{x}^i)} = \mathcal{N}(\hat{\mathbf{m}}^i, \hat{\mathbf{P}}_{\mathbf{x}^i}'). \quad (3)$$

*Global Level.* The global level is represented as an adjacency graph in which origins of local maps  $\mathbf{S}_i$  in *wrf* are nodes, and the links between them are the relative transformations  $\mathbf{s}_{i+1}^i$ . Let us define the global level as the Gaussian state  $\mathbf{s} \sim \mathcal{N}\{\hat{\mathbf{s}}; \mathbf{P}_s\}$  of relative transformations between sequentially built local maps, namely

$$\hat{\mathbf{s}} = \begin{bmatrix} \hat{\mathbf{s}}_1^0 \\ \vdots \\ \hat{\mathbf{s}}_{i+1}^i \end{bmatrix}, \quad \mathbf{P}_s = \begin{bmatrix} \mathbf{P}_{\mathbf{s}_1^0} & 0 & 0 \\ 0 & \cdot & 0 \\ 0 & 0 & \mathbf{P}_{\mathbf{s}_{i+1}^i} \end{bmatrix}. \quad (4)$$

The global origins of the maps in the *wrf* are computed as the compounding of the previous global origin with the relative transformation between sub-maps,  $\mathbf{S}_{i+1} = \mathbf{S}_i \oplus \mathbf{s}_{i+1}^i$  (a detailed description of the compounding  $\oplus$  and inversion  $\ominus$  operations for 2D and 3D can be found in [24]). The current position of the robot in *wrf* is computed as  $\mathbf{X}_k = \mathbf{S}_i \oplus \mathbf{x}_k^i$ . Also, the global map can be obtained through,

$$\mathbf{M}^i = \mathbf{S}_i \oplus \mathbf{m}^i. \quad (5)$$

Mean and covariances of the Gaussian estimates are obtained by regular linear-Gaussian propagation using the Jacobians of  $\oplus$  and  $\ominus$ . The conditional probability distribution of  $\mathbf{S}_i$ , as shown in Figure 2, is given by,

$$p(\mathbf{S}_i | \mathbf{S}_{i-1}, \mathbf{s}_i^{i-1}) = \frac{p(\mathbf{S}_i, \mathbf{S}_{i-1} | \mathbf{s}_i^{i-1})}{p(\mathbf{S}_{i-1} | \mathbf{s}_i^{i-1})}. \quad (6)$$

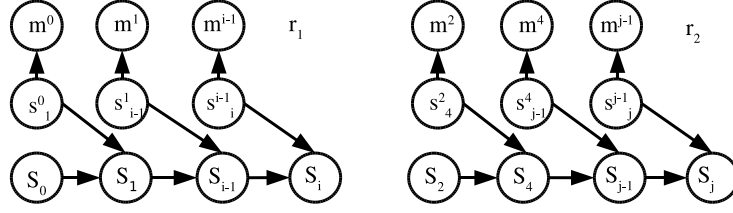


Figure 3: Bayesian network of the hierarchical SLAM for two robots. Two disconnected sub-graphs, one for each robot.

Considering the relative transformations between local maps as past robot poses, we note that the global level can be viewed as a sparse delayed-state pose-SLAM [25], where local maps are like landmarks hanging from robot poses in *wrf* as shown in Figure 2. The main difference is due to the fact that the state-space in our case contains relative poses  $s_i^{i-1}$ , instead of absolute poses  $S_i$  (that will be naturally correlated) as in the pose-SLAM case when the map features are marginalized out.

*Loop Closure.* At the global level, a loop closure corresponds to a cycle in the graph, that appears for instance when a relative position estimate between non-consecutive sub-maps is established by a map matching process. Such a cycle defines a constraint between a series of relative transformations:

$$\mathbf{h}(\mathbf{s}) = \mathbf{s}_1^0 \oplus \mathbf{s}_2^1 \cdots \oplus \mathbf{s}_0^i = 0 \quad (7)$$

$$= \mathbf{S}_i \oplus \mathbf{s}_0^i = 0. \quad (8)$$

Given that  $\mathbf{h}(\mathbf{s})$  is not linear due to the angular terms, the enforcement of this constraint can be formulated as a nonlinear constrained optimization problem such that,

$$\min_{\mathbf{s}} F(\mathbf{s}) = \min_{\mathbf{s}} (\mathbf{s} - \hat{\mathbf{s}})^\top \mathbf{P}_s^{-1} (\mathbf{s} - \hat{\mathbf{s}}) \quad (9)$$

$$\mathbf{h}(\mathbf{s}) = 0. \quad (10)$$

For multiple  $l$  loops  $\mathbf{h}(\mathbf{s})$  is formed by all the constraints  $\mathbf{h}(\mathbf{s}) = [\mathbf{h}_1(\mathbf{s}) \dots \mathbf{h}_l(\mathbf{s})]^\top$ . The maximum *a posteriori* likelihood solution for instance, could be based on the Iterative EKF as presented in [1]. As a consequence, the part of the state involved in the loop closure at global level becomes correlated, resulting in a non-sparse covariance matrix  $\mathbf{P}_s$ .

### 2.3. Multiple robots

A hierarchical/hybrid SLAM approach in the multi-robot case is quite straightforward: each robot manages a set of sub-maps and a global graph of poses. This approach lends itself nicely to the case of multiple robots as shown in the Bayesian network of Figure 3. But the interests of multi-robot mapping arise of course when the robots exchange mapping or position information, which permits the enhancement of the spatial consistency and the construction of a *multi-robot global graph* of map poses. Indeed, when robots meet or communicate, they can estimate their relative position or match maps that overlap: these kind of events generate connections between the individual global graph. The possible events are the following:

- robots rendezvous (Figure 4(a)),
- match common information within sub-maps (Figure 4(b)),
- receiving external information that provide absolute localizations (*e.g.* a GPS fix (Figure 4(c)), or feature matches with an existing environment model).

The latter is not exactly a direct multi-robot loop closure, but it provides a link between a *lrf* and a global geo-referenced frame, which in turn establishes a link with any other robot that has already been absolutely localized once.

Whereas in a single robot case a loop closure only occurs when the robot revisits and matches a previously mapped place, in a multi-robot case these events trigger loop closures: any cycle that appears in the overall graph defined by

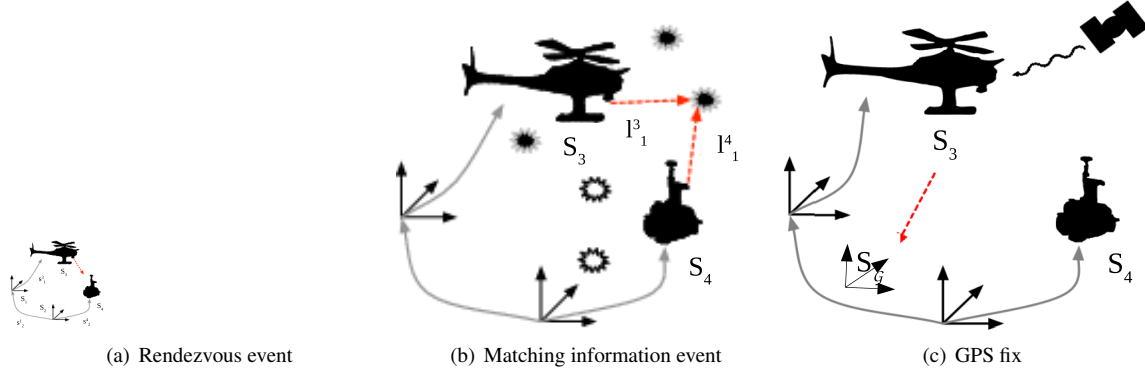


Figure 4: Loop-closing events for multiple robots.

the concatenation of each robot graph (the *multi-robot graph*) is a loop closure. The compounding of all relative transformations that defines a cycle is equal to zero as in (7), and a nonlinear optimization over the transformations can be performed. Note that to obtain a cycle in the graph defined by the concatenation of two robots global levels, at least two events between these robots are required.

**Robot Rendezvous.** The event occurs when a robot observes another robot (partial rendezvous) or when both robots observe each other (full rendezvous). We focus on the case when the relative transformation between two robots is fully recovered, from the information obtained through a partial or full rendezvous. That is, whenever this event occurs, it leads to an immediate observation of the relative transformation between locations of the two vehicles.

New local maps are created at the instant of the rendezvous, then the current robot poses are “promoted” to the global level, *i.e.*,  $\mathbf{s}_i = \mathbf{x}_k^{i-1}$  and  $\mathbf{s}_j = \mathbf{x}_k^{j-1}$ . In this way, the observed transformation  $\mathbf{z}$  naturally produces a link between the maps’ origins  $\mathbf{S}_i$  and  $\mathbf{S}_j$  on the global level;  $\mathbf{z} = \mathbf{s}_j^i$ . Figure 5 shows the Bayesian network representation of this event. Note that in most of the cases, multiple observations of the relative transformation between two robots would be needed to recovered the full pose (position and orientation). This will require a dedicated implementation of a delayed-state filter for instance.

**Matching common information.** There are two different ways in the literature to match common information. The first one uses signal information independent from the SLAM estimates, *e.g.* image’s descriptors matching (SIFT, SURF), image indexing techniques [26] or scan matching. A common way to produce a map of poses (Pose SLAM [25]) is to find the rotation and translation between two robot poses using one of these techniques, as opposed to tracking features. The second way to match common information is using the available information in the SLAM maps (landmarks’ position and uncertainty in global level using (5)), which is triggered usually based on the current position of the robot or robots in the *wrf*. We refer the first approach as “data-matching” (image-matching in the vision case), and the second as “map-matching”.

The image-matching produces a link directly between images, that are associated to certain poses  $\mathbf{S}_i$  and  $\mathbf{S}_j$ , as shown in Figure 6(a). This is a simple, but effective manner to obtain the relationship between two robots, or even to close a loop with a single robot. The observations are independent from the previous mapped information, but the robot poses have to be part of the global graph when this event occurs. As in the rendezvous case, this event produces directly the missing link  $\mathbf{z} = \mathbf{s}_j^i$ : practically, two current images matching is equivalent to a rendezvous.

The map-matching requires the transformation between *lrf* to be recovered based on previously built maps. It could require both local maps to be transformed to a common frame for data-association, *e.g.* promoted to the global level as  $\mathbf{M}^i$  and  $\mathbf{M}^j$  using (5) (see Figure 6(b)). The matching process happens in the 3D space, being the *wrf* or the



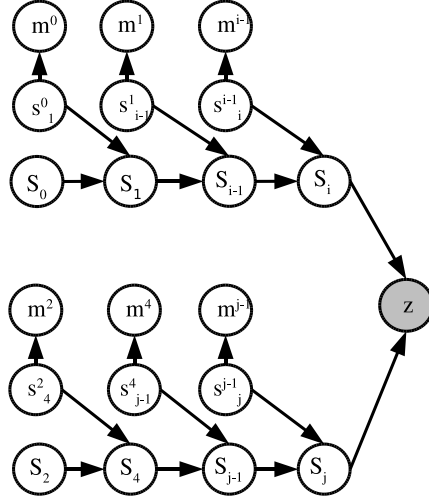


Figure 5: Bayesian network of rendezvous event. The observation  $\mathbf{z}$  is the relative transformation between the two nodes.

*lrfs*. The disadvantage of this method is that in the worst case the absolute position of the two sub-maps must be computed. Moreover, once the maps are matched, they have to be fused into a single one, otherwise it could lead into inconsistencies when merging all the maps. The reason is that the local maps involved turn out not to be conditionally independent<sup>2</sup>  $\mathbf{m}^i \perp\!\!\!\perp \mathbf{m}^j \mid \mathbf{z}$ , as shown in Figure 6(b).

*Absolute Localization.* In an aerial/ground context, it is reasonable to assume that one or both kind of robots might receive GPS fixes from time to time<sup>3</sup>. The relative transformation provided by a GPS fix for the pose  $\mathbf{x}_k^i$  instant  $k$  is simply  $\mathbf{s}_{i+1}^{\mathcal{G}}$ , where  $\mathcal{G}$  is the geo-referenced frame and new local map is started. Such information provides a link between a *lrfs* and a global geo-referenced frame, and can generate a loop at the graph level for an individual robot. Figure 7 shows a graphical model representation of this event.

*Map Merging.* Merging sub-maps into a global map is not necessary for the robots to operate, as this is not required to maintain the consistency on the graph. Should one require a global map, the map fusion could be delayed until all possible loop closures are performed, *e.g.* at the end of the mapping process. For that purpose, we exploit an approach similar to the Divide & Conquer algorithm [21]. Maps are changed to a common reference frame and merged two by two, and the fusion considers the common map information in covariance form.

*Impact on the sub-maps.* From the point of view of a hierarchical SLAM formulation, the hierarchical nature of this model manifests itself in the ability to apply a loop-consistency constraint to a subset of local transformations (*e.g.* a single loop) without taking into account the local sub-maps. Particularly, when no information is shared between sub-maps, which is the case between two sub-maps built by different robots with strictly independent observations but an approximation for the sub-maps with non-independent observations, the origin of the local sub-map is the only state that changes after the constraint is applied. We are interested in formulating this important property in terms of the properties of the graphical model. Let us consider the conditional independence property of the graph in Figure 2. It can be easily shown that  $\mathbf{m}^{i-1} \perp\!\!\!\perp \mathbf{S}^i \mid \mathbf{s}_i^{i-1}$  and therefore  $\mathbf{m}^{i-1} \perp\!\!\!\perp \mathbf{m}^i \mid \mathbf{s}_i^{i-1}$ , *i.e.*, given the relative transformations, the consecutive local sub-maps are independent, such that

$$p(\mathbf{m}^i, \mathbf{m}^{i-1} \mid \mathbf{s}_i^{i-1}) = p(\mathbf{m}^i \mid \mathbf{s}_i^{i-1})p(\mathbf{m}^{i-1} \mid \mathbf{s}_i^{i-1}), \quad (11)$$

where (3) is directly applied. Note that the global poses  $\mathbf{S}_i$  are d-separated [27] from all possible paths between the pair of sub-maps  $\mathbf{m}^{i-1}$  and  $\mathbf{m}^i$  or even  $\mathbf{M}^{i-1}$  and  $\mathbf{M}^i$ .

<sup>2</sup>We use  $\perp$  to express conditional dependence and  $\perp\!\!\!\perp$  conditional independence, for further details please refer to [27].

<sup>3</sup>Typically an aerial robot is often localized using GPS information.

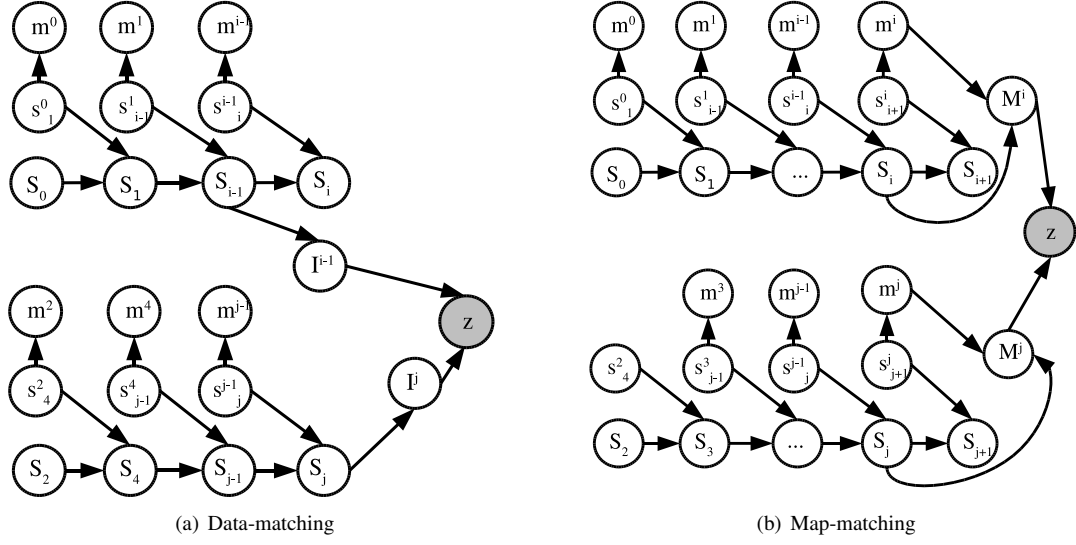


Figure 6: Bayesian network of information matching event. a) Independent information of the map is observed to establish the link between the nodes (e.g. images), without the need to recover the global maps. b) It requires to recover the global maps to perform the data association.

Also, in the multi-robot case it can happen that two different events create a link on the same node, *i.e.*, if a map-matching is established after a rendezvous. To avoid this problem, new sub-maps are started *after an event occurs* for the robots involved. In the case of receiving GPS fixes once in a while, the fact of starting a new local map at the instant  $k$  when the fix is received, removes the dynamics aspects of the internal local maps setting a fix pose at the global level. Note that in practice GPS fixes could be received at high frequencies. In this case not every measurement should create an event.

Similarly, to avoid counting information twice if one eventually wants to merge all the sub-maps, after a map-matching event both sub-maps should be fused into a single sub-map. This has the disadvantage that the sub-maps must be shared among the two robots, but on the one hand this is a pre-requisite for at least one robot to establish the matches, and on the other hand such events will occur when the robots are within communication range.

The main advantage to exploit a hierarchical map structure in multi-robot mapping is the low communication bandwidth required among the robots: only the individual graphs need to be exchanged to update the multi-robot graph. Most importantly is that in the general case, only marginal distributions of each node have to be communicated, as opposed to the full joint distributions of the graph.

#### 2.4. Distributed constraints enforcement

The presented approach is distributed in the sense that each robot builds its own local sub-maps. The only information that is shared between robots is the set of relative transformations between sub-maps  $s_{i+1}^i$  (the marginals that form the unconstrained global graph  $\mathbf{s} \sim \mathcal{N}(\hat{\mathbf{s}}, \mathbf{P}_s)$ ). With this information, each robot builds its own graph of global origins and enforces its own constraints separately.

In the centralized case (*e.g.* with a central server), the loop constraint can be enforced whenever a cycle in the graph is found. For the distributed case, if a constraint is enforced by one robot and not the others, further global information exchanges will lead to incompatible graphs among robots. Therefore, a loop constraint should only be enforced when it is known by all the robots (see Figure 8). Nevertheless, any constraint can be enforced locally, provided the graph state before the constraint application is memorized, which allows to backtrack to a globally consistent graph when new information is received from other robots.

Several cycles might appear in the graph when an event happens. Our algorithm deals with this issue enforcing only the constraints over the minimal cycle, as shown in Figure 8. This approximation is performed to enforce only one cycle at a time in order to accelerate the incremental online solution. The minimal cycle guarantees less linearization errors and focuses only in the current mapped area. The offline centralized solution should use instead (9) for all

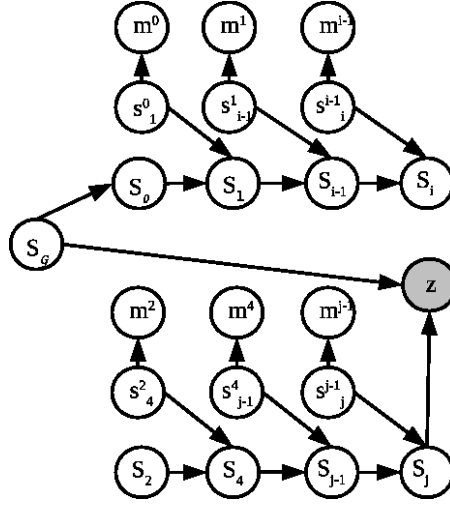


Figure 7: Bayesian network of GPS fix event. A link is created to the geo-referenced frame  $\mathcal{G}$  and the current pose.

cycles in the graph. The smallest cycle between the new connected nodes is searched right after the link is added. Not only the graph, but also a record of all the new links related with any event are sent to the robots. This is done especially for the robots that are not involved in the loop closure.

The sequence of steps of the algorithm is as follows:

1. Build local maps
2. When an event occurs
  - (a) Broadcast the unconstrained global graph<sup>4</sup>
  - (b) Append the unconstrained global graph with the new marginals
  - (c) Check if all robots' information have been received
  - (d) Check if there is a cycle
    - Perform optimization over the minimal cycle.

In the example of Figure 8, three robots are considered. Each robot searches for cycles in the graph until the information of the other two has been communicated. For the sake of clarity the robot  $n$  is indicated as the superscript of the global origin *i.e.*,  $S_i^n$ . Note that in Figure 8c)  $r_1$  does not have any cycle yet (step 2(b)), while  $r_2$  and  $r_3$  do, after having received the information of the other two robots previously (step 2(c)). It is not until Figure 8d), when  $r_1$  receives the information from the missing robot that can impose its own constraint, and therefore re-localize itself.

*Algorithms complexity.* Within the local maps, the EKF-SLAM is  $O(m^2)$ , with  $m$  the number of landmarks in each submap. At the global level, finding the minimum cycle is  $O(\log n)$ , with  $n$  the number of nodes of the graph, and the optimization of the minimum cycle of  $g$  nodes is  $O(g)$ , without the full computation of  $\mathbf{P}_s$  that is  $O(n^2)$ .

## 2.5. Simulation results

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

Simulation results are analyzed for three settings: rendezvous, map matching, and the distributed full collaboration of three robots with several loop closures triggered by the latter events.

The world has 300 landmarks spread along a 10000 m<sup>2</sup> surface. Aerial and ground robots build fixed size local maps with point landmarks. We show robots' trajectories and consistency plots before the final global map merging.

<sup>4</sup>in practice the new nodes can be communicated whenever the robots are in communication range, *e.g.*, every time a new local map is created.

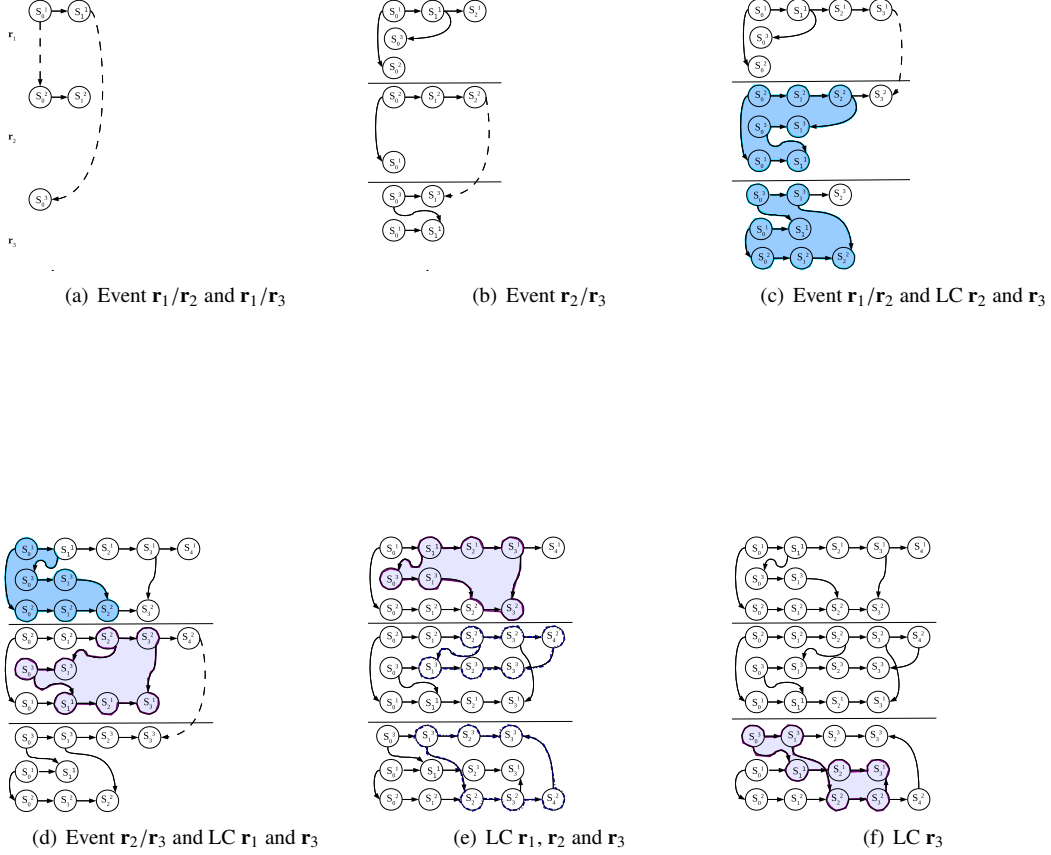


Figure 8: Example of the graph for the distributed algorithm with three robots. The dash-lines represent events between robots. For the sake of simplicity, we also consider dash-lines as communication links. Highlighted areas represent nodes involve in the optimization. The algorithm automatically searches for the minimum cycle between the new connected nodes after an event occurs.

In each sub-map, the poses and their uncertainties are expressed in the associated *lrfs*: in order to plot the global positions, we compound the poses and the covariances to obtain them in the *wrf*.

The robots make bearing-only observations: the inverse-depth parametrization (IDP) is used to map 3D point landmarks, with the parameters  $\rho_{init} = 0.5 \text{ m}^{-1}$  and  $\sigma_\rho = 0.5 \text{ m}^{-1}$  [28]. The bearing-only observation model has a  $0.2^\circ$  standard deviation, and the initial parameters for the three experiments are shown in Table 1. Robots are controlled by linear  $u$  and angular  $\omega$  velocities on their own planes with the following odometry noise model:  $\sigma_u = 0.01 \text{ m}/\sqrt{s}$ ,  $\sigma_\omega = 1^\circ/\sqrt{s}$ ,  $\sigma_{\omega u} = \sigma_{u\omega} = 0$ . The uncertainty of each robot is considered for the 6 DOF pose vector  $(x, y, z, \psi, \theta, \phi)$ .

*Rendezvous.* For this setting we use two robots,  $\mathbf{r}_1$  (aerial robot) and  $\mathbf{r}_2$  (ground robot). We choose to start with a large uncertainty for  $\mathbf{r}_2$  to see the impact of loop closures in the global localization – each local map is started with zero uncertainty, so this does not affect the local EKF-SLAM performance.

The robots meet in the middle of the environment where  $\mathbf{r}_1$  detects  $\mathbf{r}_2$  and computes the transformation between the two robots. This event triggers a loop closure because both robots' initial positions are expressed in the same *wrf*. The standard deviation of the relative measure between robots is  $(0.02, 0.02, 0.02, 0.005, 0.005, 0.005)$  in

Experiment	Rob	Real Initial Position ( $x, y, z, \psi, \theta, \phi$ ) (m,m,m,rad,rad,rad)	Slam Initial Position ( $x, y, z, \psi, \theta, \phi$ ) (m,m,m,rad,rad,rad)	Initial Standard deviation ( $\sigma_x, \sigma_y, \sigma_z, \sigma_\psi, \sigma_\theta, \sigma_\phi$ ) (m,m,m,rad,rad,rad)	Speed ( $u, \omega$ ) (m/s, rad/s)
Rendezvous	$\mathbf{r}_1$	(0, -25, 8, 0, 0, 0)	(0, -25, 8, 0, 0, 0)	(0, 0, 0, 0, 0, 0)	(0.1, 0.01)
	$\mathbf{r}_2$	(0, 25, 0, 0, 0, 0)	(0.2, 25.5, -0.3, 0.015, 0.015, 0.015)	(0.05, 0.05, 0.05, 0.0003, 0.0003, 0.0003)	(0.1, -0.01)
Map Matching	$\mathbf{r}_1$	(0, 0, 8, 0, 0, 0)	(0, 0, 8, 0, 0, 0)	(0, 0, 0, 0, 0, 0)	(0.1, 0.007)
	$\mathbf{r}_2$	(0, 25, 0, 0, 0, 0)	(-0.1, 24.7, 0.1, 0.015, 0.015, 0.015)	(0.05, 0.05, 0.05, 0.0003, 0.0003, 0.0003)	(0.1, 0.01)
Full Collaboration	$\mathbf{r}_1$	(25, 50, 0, 0, 0, 0)	(25, 50, 0, 0, 0, 0)	(0.1, 0.1, 0.1, 0, 0, 0)	(0.1, -0.01)
	$\mathbf{r}_2$	(-25, -50, 0, 0, 0, 0)	(-25, -50, 0, 0, 0, 0)	(0.1, 0.1, 0.1, 0, 0, 0)	(0.2, 0.01)
	$\mathbf{r}_3$	(0, -40, 8, 0, 0, 0)	(0, -40, 8, 0, 0, 0)	(0.1, 0.1, 0.1, 0, 0, 0)	(0.3, 0.008)

Table 1: Simulation parameters for the three robots settings.

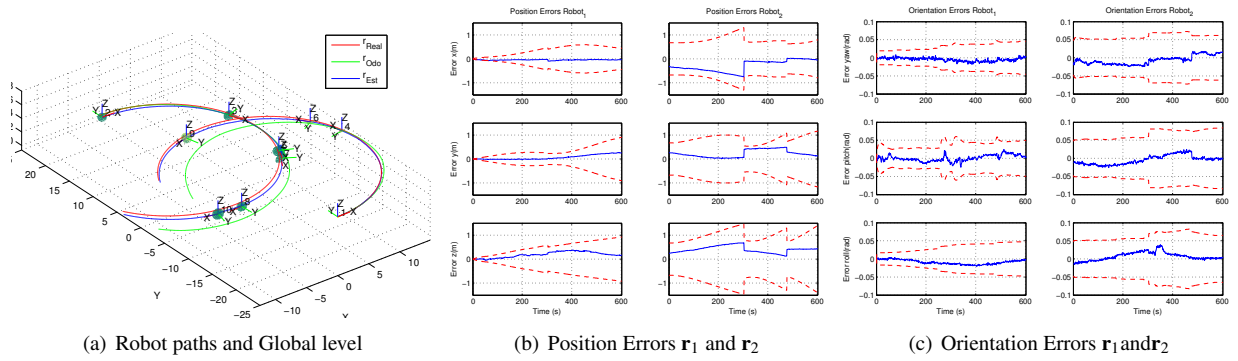


Figure 9: Rendezvous single run simulation results for one aerial and one ground robots. In a) the odometry is shown in green, real and estimated trajectories are shown in red and blue respectively.  $3\sigma$  ellipsoids are plotted on the basis of each  $lrf$ . b) shows the global position errors for each robot and their global  $3\sigma$  uncertainty bounds. c) shows the global orientation errors for each robot and their global  $3\sigma$  uncertainty bounds.

(m,m,m,rad,rad,rad). The robots meet twice, the first time about 300 s and then approximately at 440 s as shown Figure 9. The first rendezvous reduces the covariance on the position of  $\mathbf{r}_2$  and the error as well (see Figure 9(b) at 300 s). As one can see in the plot after the second rendezvous, again  $\mathbf{r}_2$  is re-localized.

To evaluate the consistency of the approach, we performed a 25-runs Monte Carlo analysis of the normalized estimation error squared (NEES) of the current robot pose in  $wrf$  as explained in [29]. Figure 11(a) shows the average NEES for the global pose of both robots and the single-sided 95% region for the  $N * 6$  DOF during this setting. The average NEES shows good consistency for the full simulation, thanks to the use of sub-maps. For robot  $\mathbf{r}_2$ , the average NEES is small compared with the consistency limits after the loop closure. This is because some uncertainty in the orientation was added by sensing each other (see Figure 9(c)), while the estimation error remain the same.

**Map Matching.** For this experiment the two robots move clockwise in a circle. Robot  $\mathbf{r}_1$  moves in the direction of the initial position of  $\mathbf{r}_2$ . Once  $\mathbf{r}_1$  reaches that position it is able to close a loop, as seen on Fig. 10(a). Data association is known. One of the main issues of this experiment is to compute the transformation between the two local maps which have common landmarks, because the least-square estimation is very sensitive to points whose localization is not accurate. The problem is even more difficult for bearing-only observations, when newly initialized points can be anywhere on the bearing line. The eigenvalues of each point's covariance matrix are computed, and points whose eigenvalues are below a threshold (e.g.  $0.1 \text{ m}^2$ ) are selected to estimate the transformation. The relative transformation is recovered using the approach in [30].

The results of a single run experiment are presented on Figure 10. A first loop is closed at 380 s, which increases the precision of the global localization of  $\mathbf{r}_2$ . At 520 s the two robots close another loop. Note that those two loops have mostly improved the global localization of  $\mathbf{r}_2$ . When  $\mathbf{r}_2$  reaches its origin, it closes a loop with its first map, at around 610 s: that map being topologically connected to the maps of  $\mathbf{r}_1$ , a slight improvement on the localization of  $\mathbf{r}_1$  can be seen through its error covariance (Figure 10(b)).

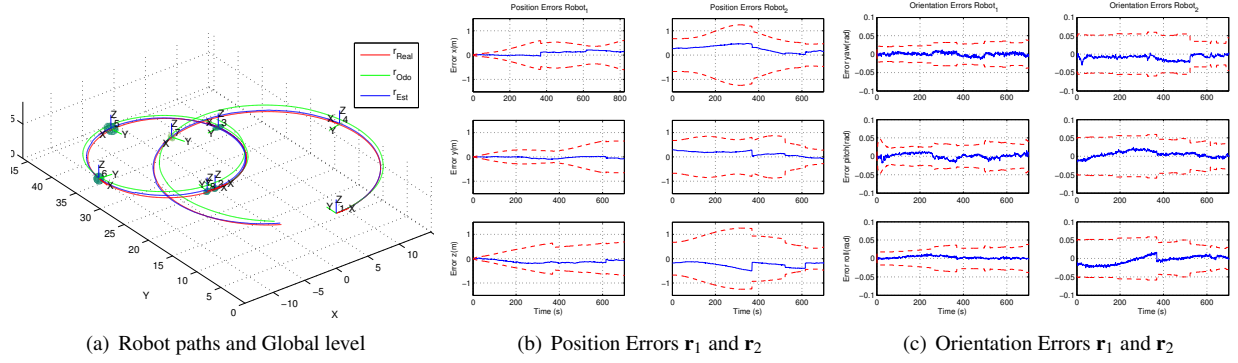


Figure 10: Map-matching simulation results for one aerial and one ground robots. See caption of Fig. 9.

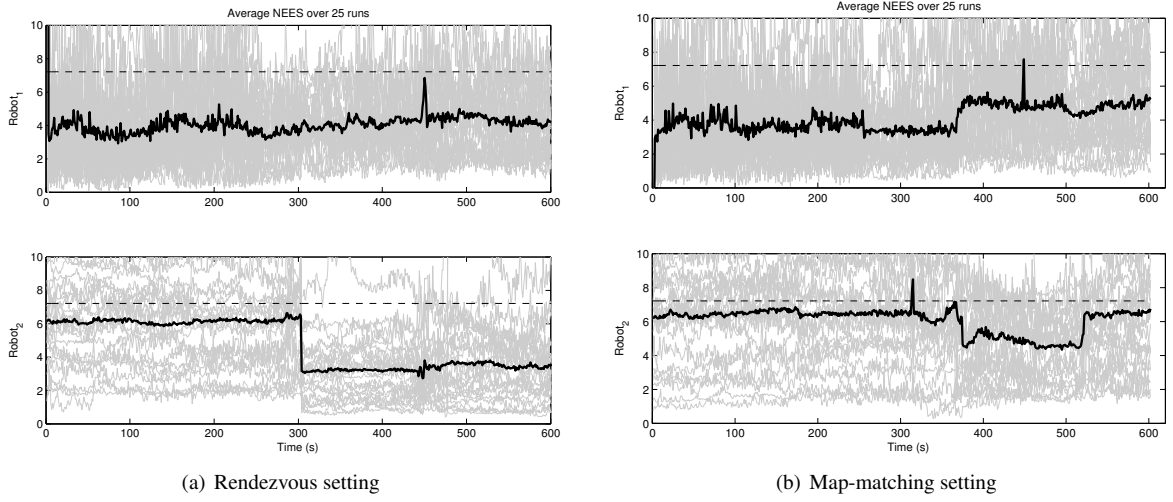


Figure 11: Average NEES of 25 Monte Carlo runs of the robot pose (position and orientation) for (a) the rendezvous setting and (b) the map-matching setting. The averaged NEES is in thick line. The thin lines represent the NEES for each run. The dashed line corresponds to a NEES value of 7.18, which is the single-sided consistency limit of 95% confidence for the 6-dimensional state vector.

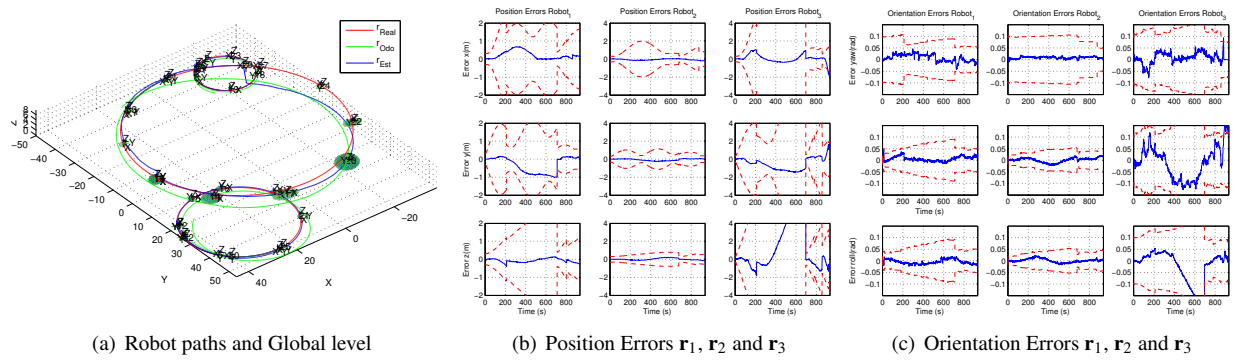


Figure 12: Simulation results for the 3 robots exploration; one aerial and two ground robots. See caption of Fig. 9.

Figure 11(b) shows the average NEES of the robots poses for this setting: after a loop closing the average NEES

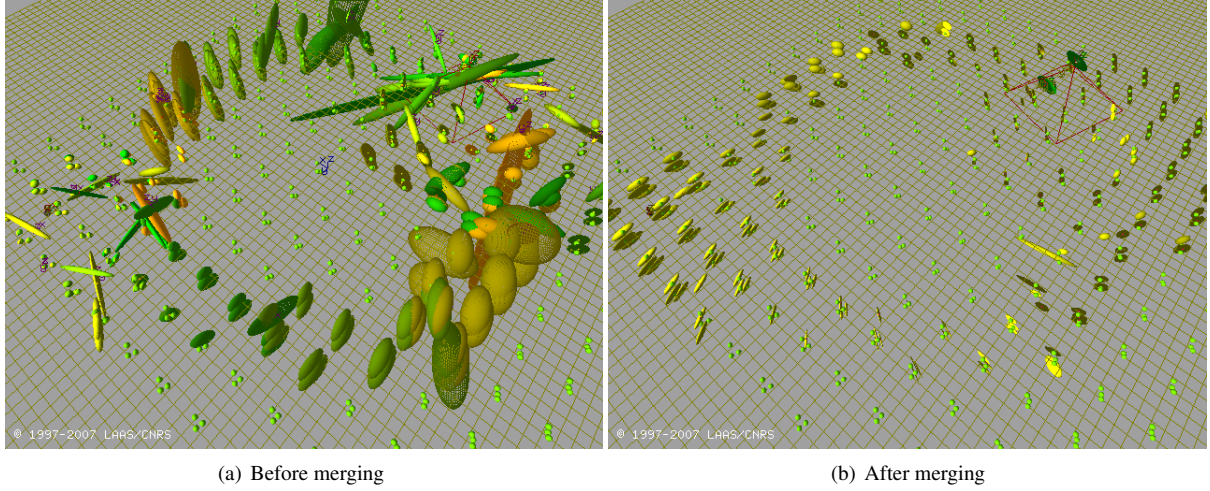


Figure 13: Final map resulting from the 3 robots exploration centralizing the information. The figure shows the 3D global map, local map origins and the final robots and landmarks location in *wrf*.

remains pretty much the same, showing consistency for both robots.

*Distributed collaboration between three robots.* We show here the effects of multiple loop closures between three robots that communicate their unconstrained global maps whenever an event occurs. Two ground robots  $\mathbf{r}_1$  and  $\mathbf{r}_2$  move along circles in different locations on the environment. They never meet, and their maps never overlap. The third robot  $\mathbf{r}_3$  is an aerial robot that moves in a circular trajectory which extends from the area covered by  $\mathbf{r}_1$  to the one covered by  $\mathbf{r}_2$ . Each robot simulation runs in its own process.

The three robots start at a well known location, then at 240 s,  $\mathbf{r}_1$  and  $\mathbf{r}_3$  have a rendezvous, later at 610 s and 700 s the robot  $\mathbf{r}_3$  detects a loop closure with two maps of  $\mathbf{r}_2$  (the video `slamnet.mp4`<sup>5</sup> shows a run of this simulation). The robots communicate the unconstrained global graph when a local map is created. Perfect communication is assumed. The uncertainties are expressed in the *wrf* so that the effects of the loop closure can be seen. Notice the robots only have information about their own landmarks. The consistency plots for a single run are shown in Figure 12. The final map and global graph before and after merging (at the end of the simulation) are shown in Figure 13.

### 3. Estimating line segments in the monocular EKF-SLAM framework

Most of the visual SLAM approaches rely on landmarks defined by interest points detected in the images, be they Harris, CenSurE, SIFT, SURF or other features. The estimated state of such landmarks is their 3D position, and the resulting maps are sparse collections of 3D points: even though they are spatially consistent and span the world's three dimensions, such maps do not represent useful information to assess volumes, to compute visibilities, or to establish matches either with maps built by another robot or with existing 3D maps now widely available.

To improve the map's representational power, one of the key points here is to use, in addition to sparse 3D points, linear landmarks or *line segments*. Lines provide an improved semantic over points: they inherently contain the notions of connectivity (they *span* from one position to another) and boundary (they *separate* one region from another), which open the door to potential automatic interpretations of the environment, both at the local and global levels. A 3D model based on lines can further allow the possibility to build higher level entities (planes, closed regions, objects).

By building such maps, our goal is to aim at building a meaningful 3D model, on the basis of which one can assess visibilities and match features or maps from disparate viewpoints. These goals, however, are difficult to achieve and lie beyond the scope of the work presented in this paper. As recognized by Smith *et. al.* [31], one of the difficult

<sup>5</sup> <http://www-personal.acfr.usyd.edu.au/tvidal/videosRAS/slamnet.mp4>



features to achieve satisfactory SLAM operation with lines is a good lines extractor and matcher algorithm, which is not one of our goals in this paper. We concentrate only on the estimation of the geometry of 3D lines from moving monocular platforms, which constitutes by its own right a problem that has not been successfully solved yet, as we expose in the following section. Wide baseline map matching based on lines is introduced, and demonstrated only in simulation. Real map matching based on lines and map interpretation are left for future work. The focus is therefore on the ability of building heterogeneous maps with points and lines.

### 3.1. Related work

Most approaches to visual SLAM with segments define the segments based on two endpoints that support them, fixing the segment's length so that partially occluded segments remain shorter than they actually are regardless of the evidence gathered with new observations. Additionally, the majority of these works make use of delayed initialization techniques, limiting the potential performances of monocular or long-range sensing SLAM. These remarks are detailed in the following paragraphs.

Ref. [32] proposes a model-based monocular SLAM system using line segments and the UKF, but the use of direct-distance parametrization for the unknown endpoints depths leads to delayed initialization. Ref. [31] represents segments with their two endpoints. The support endpoints are coded using inverse-depth parameters during the initialization phase, and are converted to Cartesian representations after convergence. The inverse depth parameters are initially estimated by an external EKF until convergence, thus delaying initialization. Ref. [33] makes use of small edge landmarks, named '*edgelets*', associated to a 3D point which is initialized in an undelayed manner. Edgelets are typically 15 pixels long, and longer lines must be represented by several edgelets. This intrinsically local definition of edgelets does not possess the connectivity property that we highlighted above (though it does possess local boundary), compromising precisely the representativeness that we are seeking. We look here for representations of arbitrarily long lines with the additional possibility of incrementally updating their endpoints according to new observations.

These works have been conceived and demonstrated in indoor setups, often using camera motions that have been purposely chosen to maximize observability (*i.e.*, motions with a significant sideways component looking at objects close by), thus achieving successful operation even with delayed initialization. Our case is radically different as we are dealing with outdoors robots whose trajectories are planned and executed with other priorities in mind, and where objects to be mapped might be at large distances. As we have demonstrated earlier [34, 35, 36], undelayed initialization allows the robots to use the partial landmark information for localization from the very first observation of a landmark, regardless of the sensor trajectory and the distance to the landmark, greatly helping to constrain the camera localization. In particular, this makes the use of remote landmarks beneficial for long term attitude estimation, minimizing angular drift which is the major cause of inconsistency in EKF-SLAM [29]. This constitutes a crucial feature in *e.g.* aerial vehicles as in our case, especially because, due to the long range sensing requirements and strict energy and payload constraints, only monocular vision seems to be practicable.

Our approach draws on [36], which is in turn based on [37]. Ref. [37] performs SLAM with segments using the Euclidean Plücker coordinates (a sub-set of the Plücker coordinates where the direction vector is normalized) to map infinite lines. The major drawback of [37] is a *delayed* initialization. In [36] we added undelayed operation by using the Plücker coordinates directly, where the Plücker direction vector's norm exhibits inverse-distance behavior. Here we add an anchor to improve linearity, equivalently to what it is shown in [38] for the case of points.

Plücker lines have also been used in many major vision works with straight 3D lines [39, 40, 41]. These works, and other ones referenced therein, are based on Structure From Motion approaches solved offline using non-linear optimization. Online, incremental SLAM methods based on these techniques would have to draw on *e.g.* [42] (which works for points and edgelets) but they have not yet been reported.

### 3.2. Plücker lines (PL): incorporating the inverse-depth concept to lines

The Plücker coordinates for a line consist of a homogeneous 6-vector in projective space,  $\mathcal{L} \in \mathbb{P}^5$ . In this vector, one can identify two sub-vectors<sup>6</sup>,  $\mathcal{L} = (\mathbf{n} : \mathbf{v})$ , with  $\{\mathbf{n}, \mathbf{v}\} \in \mathbb{R}^3$ , with which an intuitive geometrical interpretation of the line in 3D Euclidean space is possible [36] (Fig. 14(a)):

<sup>6</sup>We use a colon (:) to separate the non-homogeneous and homogeneous parts in projective space.



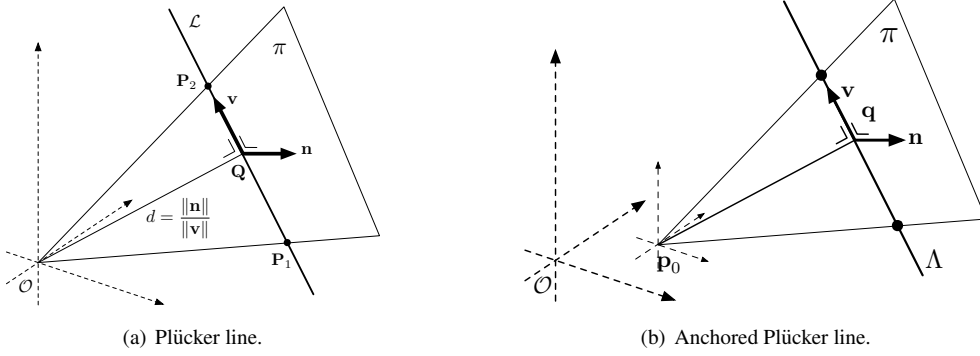


Figure 14: Geometrical representation of Plücker-based lines, the sub-vectors  $\mathbf{n}$  and  $\mathbf{v}$  and the anchor  $\mathbf{p}_0$ . (a) The Plücker line  $\mathcal{L}$  and the origin  $O$  define the support plane  $\pi$ . The line's sub-vector  $\mathbf{n} \in \mathbb{R}^3$  is orthogonal to  $\pi$ . The sub-vector  $\mathbf{v} \in \mathbb{R}^3$  is a director vector of the line, and lies on  $\pi$ . This implies  $\mathbf{n} \perp \mathbf{v}$ . The closest point to  $O$  is  $\mathbf{Q} = (\mathbf{v} \times \mathbf{n} : \mathbf{v}^\top \mathbf{v}) \in \mathbb{P}^3$ . The distance from  $\mathcal{L}$  to  $O$  is  $d = \|\mathbf{n}\|/\|\mathbf{v}\|$ , showing that  $\mathbf{v}$  acts as the homogeneous part of  $\mathcal{L}$ , thus exhibiting inverse-depth properties. (b) The anchored Plücker line  $\Lambda$  is a Plücker line referred to an anchor  $\mathbf{p}_0$ . The closest point of the line to the anchor is  $\mathbf{q} = \mathbf{p}_0 + (\mathbf{v} \times \mathbf{n})/(\mathbf{v}^\top \mathbf{v})$ .

1. The sub-vector  $\mathbf{n}$  is a vector normal to the plane  $\pi$  supporting the line  $\mathcal{L}$  and the origin  $O$ .
2. The sub-vector  $\mathbf{v}$  is a director vector of the line.
3. The distance from the line to the origin is given by  $d = \|\mathbf{n}\|/\|\mathbf{v}\|$ .

The basic operations needed to manipulate Plücker lines are detailed below:

**Frame transformation.** Given a (camera) reference frame  $C$  specified by a rotation matrix  $\mathbf{R}$  and a translation vector  $\mathbf{T}$ , a Plücker line  $\mathcal{L}$  in global frame is obtained from a line  $\mathcal{L}^C$  in frame  $C$  with the linear transformation [39]

$$\mathcal{L} = \begin{bmatrix} \mathbf{R} & [\mathbf{T}]_\times \mathbf{R} \\ 0 & \mathbf{R} \end{bmatrix} \cdot \mathcal{L}^C. \quad (12)$$

**Pin-hole projection.** Given a perspective camera defined by the intrinsic parameters  $\mathbf{k} = (u_0, v_0, \alpha_u, \alpha_v)$ , a Plücker line in camera frame  $\mathcal{L}^C = (\mathbf{n}^C : \mathbf{v}^C)$  projects into a homogeneous line  $\lambda \in \mathbb{P}^2$  in the image plane with the linear expression [39]

$$\lambda = \mathcal{K} \cdot \mathbf{n}^C \triangleq \begin{bmatrix} \alpha_v & 0 & 0 \\ 0 & \alpha_u & 0 \\ -\alpha_v u_0 & -\alpha_u v_0 & \alpha_u \alpha_v \end{bmatrix} \cdot \mathbf{n}^C \in \mathbb{P}^2. \quad (13)$$

where  $\mathcal{K}$  is called the *Plücker intrinsic matrix*. Its relation to the regular intrinsic matrix  $\mathbf{K}$  is  $\mathcal{K} \propto \mathbf{K}^{-\top}$ .

The two most remarkable properties of the Plücker line are its linear transformation and projection equations and the inverse-depth behavior of the sub-vector  $\mathbf{v}$  (recall property 3. above), something that allows us to design appropriate undelayed initialization methods for EKF. For further details on the use of Plücker lines in monocular EKF-SLAM see [36].

### 3.3. Anchored Plücker lines (APL): improving on PL linearity

Now, we add an anchor to the parametrization to improve linearity, as it is done for points in the inverse-depth parametrization [28]. Anchoring the Plücker line means referring it to a point  $\mathbf{p}_0$  in 3D space different from the origin (Fig. 14(b)). The anchor point  $\mathbf{p}_0$  is chosen to be the optical center at initialization time. Thanks to this, the Plücker line part  $(\mathbf{n} : \mathbf{v})$  is correlated to the rest of the map only through the anchor  $\mathbf{p}_0$ . As a consequence, on subsequent EKF updates, only the accumulated errors from  $\mathbf{p}_0$  to the current camera position  $\mathbf{T}$  are considered, in contrast with regular Plücker lines where the error accounts for the absolute motion of the sensor from the origin of coordinates.

The *anchored Plücker line* (APL, Fig. 14(b)) is then the 9-vector:

$$\Lambda = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{n} \\ \mathbf{v} \end{bmatrix} \in \mathbb{R}^9 \quad (14)$$

The operations needed to manipulate APL are as follows:

**Frame transformation.** Frame transformation reduces simply to transforming the point  $\mathbf{p}_0$  and rotating the vectors  $\mathbf{n}$  and  $\mathbf{v}$ . This is accomplished with the affine transformation

$$\Lambda = \begin{bmatrix} \mathbf{R} & 0 & 0 \\ 0 & \mathbf{R} & 0 \\ 0 & 0 & \mathbf{R} \end{bmatrix} \cdot \Lambda^C + \begin{bmatrix} \mathbf{T} \\ 0 \\ 0 \end{bmatrix}. \quad (15)$$

**Un-anchoring.** Given an anchored Plücker line  $\Lambda = (\mathbf{p}_0, \mathbf{n} : \mathbf{v})$ , its corresponding (un-anchored) Plücker line  $\mathcal{L}$  is obtained by transforming the Plücker part  $(\mathbf{n} : \mathbf{v})$  from a frame at the anchor  $(\mathbf{T}, \mathbf{R}) = (\mathbf{p}_0, \mathbf{I}_3)$  to the global frame with (12), and removing the anchor. This reduces to

$$\mathcal{L} = \begin{bmatrix} \mathbf{n} + \mathbf{p}_0 \times \mathbf{v} \\ \mathbf{v} \end{bmatrix}. \quad (16)$$

**Transformation and projection.** Transformation and projection are accomplished with a transformation to the camera frame (the inverse of (15)), un-anchoring (16), and Plücker projection (13). The three steps can be composed in one single expression with:

$$\lambda = \mathcal{K} \cdot \mathbf{R}^\top \cdot (\mathbf{n} - (\mathbf{T} - \mathbf{p}_0) \times \mathbf{v}) \in \mathbb{P}^2, \quad (17)$$

### 3.4. Segment endpoints

The line's endpoints in 3D space are maintained out of the filter via two abscissas defined in the local 1D reference frame of the line, whose origin is at the point  $\mathbf{q} = \mathbf{p}_0 + (\mathbf{v} \times \mathbf{n}) / (\mathbf{v}^\top \mathbf{v})$ , the closest point to the anchor (see Fig. 14(b)). Given the line  $\Lambda = (\mathbf{p}_0, \mathbf{n} : \mathbf{v})$  and abscissas  $\{t_1, t_2\}$ , the 3D Euclidean endpoints are obtained for  $i \in \{1, 2\}$  with

$$\mathbf{p}_i = \mathbf{q} + t_i \cdot \frac{\mathbf{v}}{\|\mathbf{v}\|} = \mathbf{p}_0 + \frac{\mathbf{v} \times \mathbf{n}}{\|\mathbf{v}\|^2} + t_i \cdot \frac{\mathbf{v}}{\|\mathbf{v}\|}. \quad (18)$$

### 3.5. Back-projection of an APL

APL back-projection consists in defining a Plücker line  $\mathcal{L}$  from a segment observation  $\lambda$ , and anchoring it at the camera position  $\mathbf{T}$  to obtain an APL  $\Lambda$ . These operations are detailed below.

**Back-projection of a Plücker line.** Here we briefly summarize the development in [36]. In the camera frame, the Plücker sub-vector  $\mathbf{n}^C$  resulting from the observation  $\lambda$  is simply the inverse of (13),

$$\mathbf{n}^C = \mathcal{K}^{-1} \lambda. \quad (19)$$

The second sub-vector  $\mathbf{v}^C$  is not measured and must be obtained by injecting prior information. This prior must specify exclusively the 2 DOF that are missing in the observation. It is defined in the 2-dimensional plane and mapped to the plane orthogonal to the observed  $\mathbf{n}^C$ . This plane is spanned by the base matrix  $\mathbf{E}$  which is obviously orthogonal to  $\mathbf{n}^C$ ,

$$\mathbf{E} = [\mathbf{e}_1 \quad \mathbf{e}_2], \quad \mathbf{n}^C \perp \mathbf{e}_1 \perp \mathbf{e}_2 \perp \mathbf{n}^C. \quad (20)$$

Then, given a prior  $\beta \in \mathbb{R}^2$ , the sub-vector  $\mathbf{v}^C$  is obtained linearly with

$$\mathbf{v}^C = \mathbf{E} \beta. \quad (21)$$

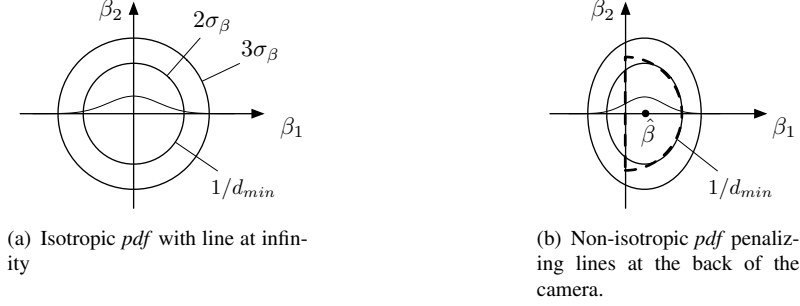


Figure 15: Defining a *pdf* for the prior  $\beta$ . (a) The Gaussian *pdf* contains all possible lines at a minimum distance of  $d_{min}$ : it is isotropic in orientation, it includes the origin which represents the line at infinity, and  $d_{min}$  is at  $2\sigma$ . For reference, a Gaussian shape is superimposed on the horizontal axis to evaluate the probability values at  $2\sigma$  and  $3\sigma$ . (b) An interesting alternative that penalizes lines in the back of the camera is to approximate just the right-hand half of the *pdf* in (a) (here dashed) by a new Gaussian. A good fit is obtained with  $\hat{\beta} = (1/3d_{min}, 0)$  and a non-isotropic covariance  $\mathbf{B} = \text{diag}([\sigma_{\beta_1}^2, \sigma_{\beta_2}^2])$  with  $\sigma_{\beta_1} = 1/3d_{min}$  and  $\sigma_{\beta_2} = 1/2d_{min}$ .

For convenience, we arbitrarily build  $\mathbf{E}$  so that the points in the  $\beta$ -plane correspond to lines easy to interpret. With this purpose of intuitiveness we choose the base vectors  $\{\mathbf{e}_1, \mathbf{e}_2\}$  so that  $\|\beta\|$  is exactly inverse-distance and  $\mathbf{e}_1$  is parallel to the image plane, leading to

$$\mathbf{e}_1 = \frac{\begin{bmatrix} n_2^C & -n_1^C & 0 \end{bmatrix}^\top}{\sqrt{(n_1^C)^2 + (n_2^C)^2}} \cdot \|\mathbf{n}^C\| \quad \text{and} \quad \mathbf{e}_2 = \frac{\mathbf{n}^C \times \mathbf{e}_1}{\|\mathbf{n}^C\|}. \quad (22)$$

**Anchoring.** This step is trivial as we have interest in making the anchor  $\mathbf{p}_0$  coincide with the current camera position, which is the origin when we are in camera frame,

$$\Lambda^C = \begin{bmatrix} \mathbf{0} \\ \mathbf{n}^C \\ \mathbf{v}^C \end{bmatrix}. \quad (23)$$

**Back-projection and transformation.** The operations above plus the transformation to the global frame (15) can be composed and written as a single-step function of  $\mathbf{R}$ ,  $\mathbf{T}$ ,  $\lambda$  and  $\beta$ ,

$$\Lambda = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{n} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{T} \\ \mathbf{R}\mathcal{K}^{-1}\lambda \\ \mathbf{R}\mathbf{E}\beta \end{bmatrix}. \quad (24)$$

We name this function  $g()$  and retain it as it is needed for landmark initialization. In EKF-SLAM, the pose orientation  $\mathbf{R}$  is often encoded by a quaternion or the Euler angles, which we denote indistinctly by  $\mathbf{Q}$ . The camera pose  $C$  is specified by the vector  $\mathbf{x} = (\mathbf{T}, \mathbf{Q})$  (we neglect here the super-index indicating the local map,  $\mathbf{x} \equiv \mathbf{x}^i$ , to make the notation easier to read), in which case the function above becomes

$$\Lambda = g(\mathbf{x}, \lambda, \beta), \quad (25)$$

which is simply a function of the current state  $\mathbf{x}$ , the current measurement  $\lambda$  and the provided prior  $\beta$ .

### 3.6. Undelayed APL initialization in EKF-SLAM

To initialize the APL, suppose we have a camera  $C$  at location  $\mathbf{x} = (\mathbf{T}, \mathbf{Q})$ , with intrinsic Plücker matrix  $\mathcal{K}$ .  $\mathbf{x}$  has uncertainties encoded in the map<sup>7</sup>, while  $\mathcal{K}$  is assumed to be deterministic. A newly detected segment, together with

<sup>7</sup>The camera pose  $C$  may be encoded in the map indirectly, *i.e.*, via a fixed transformation related to the robot pose,  $C = f(\mathbf{x})$ . Without loss of generality we consider this function here to be the identity. In case a non-trivial transformation is to be considered, we need to compose it with (25) to get the new function  $\Lambda = g(f(\mathbf{x}), \lambda, \beta) \triangleq g^*(\mathbf{x}, \lambda, \beta)$ .

its uncertainty encoded by a covariances matrix, needs to be transformed (via regular covariance propagation based on the Jacobians of the transformation) to a homogeneous line in  $\mathbb{P}^2$ . For example, if the segment is detected in the form of two endpoints  $\{\mathbf{u}_1, \mathbf{u}_2\}$ , with noise covariance  $\mathbf{R}_u = \text{diag}(\sigma_u^2, \sigma_u^2)$  each, we obtain the homogeneous line's *pdf*  $\lambda \sim \mathcal{N}\{\hat{\lambda}; \mathbf{L}\}$  by joining the two points

$$\hat{\lambda} = \underline{\mathbf{u}}_1 \times \underline{\mathbf{u}}_2 \quad (26)$$

$$\mathbf{L} = [\underline{\mathbf{u}}_1]_{\times} \cdot \underline{\mathbf{R}}_u \cdot [\underline{\mathbf{u}}_1]_{\times}^{\top} + [\underline{\mathbf{u}}_2]_{\times} \cdot \underline{\mathbf{R}}_u \cdot [\underline{\mathbf{u}}_2]_{\times}^{\top}, \quad (27)$$

where  $\underline{\mathbf{u}}_i = (\mathbf{u}_i : 1)$ ,  $\underline{\mathbf{R}}_u = \text{diag}(\mathbf{R}_u, 0)$ , and  $[\bullet]_{\times}$  is the skew-symmetric matrix associated to the cross-product (*i.e.*,  $[\mathbf{u}]_{\times} \mathbf{v} \equiv \mathbf{u} \times \mathbf{v}$ ),

$$[\mathbf{u}]_{\times} \triangleq \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}. \quad (28)$$

Initialization of the support APL is done with the classical EKF-SLAM method, by linearizing  $\Lambda = g(\mathbf{x}, \lambda, \beta)$ , Eq. (24–25), and providing  $\beta$  as a Gaussian prior. The camera  $C$  is part of the  $i$ -th local map vector  $\mathbf{x}_m^i$ , that we rename here simply  $\mathbf{x}_m$ . Its Gaussian *pdf*  $\mathbf{x}_m \sim \mathcal{N}\{\hat{\mathbf{x}}_m; \mathbf{P}_m\}$  is decomposed in (2), let us rename  $\mathbf{m} \equiv \mathbf{m}^i$  the set of existing landmarks in the current (the  $i$ -th) local map.

For the prior  $\beta$  we define a Gaussian *pdf*  $\beta \sim \mathcal{N}\{\hat{\beta}; \mathbf{B}\}$  as shown in Fig. 15. This Gaussian in the  $\beta$ -plane will be conveniently mapped to the 3D space via the transformation matrix  $\mathbf{E}$ , as we have seen in the previous section. We use the non-isotropic Gaussian shown in Fig. 15(b), described by

$$\hat{\beta} = \begin{bmatrix} 1/3d_{min} \\ 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} (1/3d_{min})^2 & 0 \\ 0 & (1/2d_{min})^2 \end{bmatrix}. \quad (29)$$

We further choose  $d_{min} = 0.75\text{m}$ .

We obtain with (24) the landmark's mean and Jacobians

$$\hat{\Lambda} = g(\hat{\mathbf{x}}, \hat{\lambda}, \hat{\beta}) \quad (30)$$

$$\mathbf{G}_x = \left. \frac{\partial g}{\partial \mathbf{x}} \right|_{(\hat{\mathbf{x}}, \hat{\lambda}, \hat{\beta})}, \quad \mathbf{G}_\lambda = \left. \frac{\partial g}{\partial \lambda} \right|_{(\hat{\mathbf{x}}, \hat{\lambda}, \hat{\beta})}, \quad \mathbf{G}_\beta = \left. \frac{\partial g}{\partial \beta} \right|_{(\hat{\mathbf{x}}, \hat{\lambda}, \hat{\beta})}. \quad (31)$$

with which, being  $\mathbf{x}$ ,  $\lambda$  and  $\beta$  mutually independent, we compute the co- and cross-variances

$$\mathbf{P}_{\Lambda\Lambda} = \mathbf{G}_x \cdot \mathbf{P}_{xx} \cdot \mathbf{G}_x^{\top} + \mathbf{G}_\lambda \cdot \mathbf{L} \cdot \mathbf{G}_\lambda^{\top} + \mathbf{G}_\beta \cdot \mathbf{B} \cdot \mathbf{G}_\beta^{\top} \quad (32)$$

$$\mathbf{P}_{\Lambda x_m} = \mathbf{G}_x \cdot [\mathbf{P}_{xx} \quad \mathbf{P}_{xm}] \quad (33)$$

to finally augment the map

$$\mathbf{x}_m \leftarrow \begin{bmatrix} \mathbf{x}_m \\ \Lambda \end{bmatrix}, \quad \hat{\mathbf{x}}_m \leftarrow \begin{bmatrix} \hat{\mathbf{x}}_m \\ \hat{\Lambda} \end{bmatrix}, \quad \mathbf{P}_m \leftarrow \begin{bmatrix} \mathbf{P}_m & \mathbf{P}_{\Lambda x_m}^{\top} \\ \mathbf{P}_{\Lambda x_m} & \mathbf{P}_{\Lambda\Lambda} \end{bmatrix}. \quad (34)$$

The endpoints are found by retroprojecting both image endpoints  $\{\mathbf{u}_1, \mathbf{u}_2\}$  onto the Plücker estimate's mean, expressing the solution  $\{\mathbf{p}_1, \mathbf{p}_2\}$  in terms of abscissas  $\{t_1, t_2\}$  with (18), and solving for them. Details on these operations and the necessary algebra can be found in [37, 36].

### 3.7. Segment EKF correction

The update of the APL is essentially a standard EKF update. The only difficulty, perhaps, is the fact that a proper measure of distance between two lines is not available in the Euclidean sense, and we need to find something practicable to define the innovation. We chose the technique described in [37] which defines the innovation space in polar coordinates (Fig. 16). The observation function  $h()$  is defined by composing (17) with the transformation from homogeneous lines  $\lambda = (\lambda_1, \lambda_2, \lambda_3)$  into polar coordinates  $(\rho, \theta)$ , given by

$$\begin{bmatrix} \rho \\ \theta \end{bmatrix} = \begin{bmatrix} -\lambda_3 / \sqrt{\lambda_1^2 + \lambda_2^2} \\ \arctan(\lambda_2, \lambda_1) \end{bmatrix}. \quad (35)$$

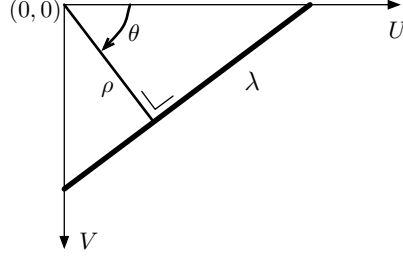


Figure 16: The polar coordinates of a line in 2D space

The measured lines are defined by 2 endpoints  $\{\mathbf{u}_1, \mathbf{u}_2\}$  with a fixed covariance  $\mathbf{R}_u$  each. Converting them to polar coordinates is just a matter of composing (26) with (35). We remind that an innovation measure  $\mathbf{y} = \mathbf{z} - h(\hat{\mathbf{x}}_m)$  based on a mapping (35) contains some singularities and discontinuities that we need to tackle. See [37] for details.

The currently measured segment's endpoints are retro-projected onto the 3D line, and its abscissas  $\{t_1, t_2\}$  computed (see Eq. 18). During the line's convergence phase, the segment endpoints abscissas are systematically replaced with the current ones. We use some heuristics to determine when the line estimate is stable enough, and then apply an extending-only policy: each segment's abscissa is updated with the new one only if this lengthens the segment.

### 3.8. Map-matching with 3D lines

Assuming the data-association problem for lines in the 3D Euclidean space is solved, the problem of recovering the transformation between two reference frames, based on a set of 3D lines, is approached using a closed form least-squares solution.

First the APLs are transformed to the Euclidean Plücker coordinates representation using (16) and normalized with respect to the line direction as  $\bar{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$  and  $\bar{\mathbf{n}} = \frac{\mathbf{n}}{\|\mathbf{n}\|}$ . When removing the anchor, the normal  $\bar{\mathbf{n}}$  is defined with respect to the origin of the reference frame. Then (12) is used to state the least-squares problem to determine  $\mathbf{R}$  and  $\mathbf{T}$ . As in the case of pose estimation from corresponding 3D points, the problem is decoupled to first recover the rotation matrix and latter the translation vector [30].

From corresponding Euclidean PLs the orientation is determined through the following criterion,

$$Err(\mathbf{R}) = \sum_{n=1}^N \|\bar{\mathbf{v}}_n^i - \mathbf{R}(\bar{\mathbf{v}}_n^j)\|^2 \quad (36)$$

where  $N$  is the number of matched lines between reference frames  $i$  and  $j$ . The rotation matrix that minimizes (36) is obtained through a singular value decomposition (SVD) algorithm as described in [30] for 3D points.

The translation vector  $\mathbf{T}$  is obtained by minimizing,

$$Err(\mathbf{T}) = \sum_{n=1}^N \|\bar{\mathbf{n}}_n^i - \mathbf{R}\bar{\mathbf{n}}_n^j - [\bar{\mathbf{v}}_n^i]_{\times}^T \mathbf{T}\|^2, \quad (37)$$

thus the explicit translation is recovered as follows,

$$\mathbf{T} = \left( \sum_{n=1}^N [\bar{\mathbf{v}}_n^i]_{\times}^T [\bar{\mathbf{v}}_n^i]_{\times} \right)^{-1} \left( \sum_{n=1}^N [\bar{\mathbf{v}}_n^i]_{\times} (\bar{\mathbf{n}}_n^i - \mathbf{R}\bar{\mathbf{n}}_n^j) \right). \quad (38)$$

### 3.9. Simulation results

We simulate a scenario consisting of a wireframe of a house built with 27 segments (Fig. 17 shows the house being reconstructed). A robot with one perspective camera (90° FOV, 640×480 pix resolution, 0.5 pix error) looking forward approaches from a certain distance at 1.2 m/s, gathering images at 30 fps. In order to observe the scale factor, the robot takes noisy odometry readings with  $0.01 \text{ m} / \sqrt{\text{m}}$  and  $0.25^\circ / \sqrt{\text{m}}$  error. In order to improve consistency, the measurement noise covariance is multiplied by a factor 2, i.e.,  $\mathbf{R}_u = 2 \times \text{diag}(0.5^2, 0.5^2) \text{pix}^2$ , as suggested in [29].

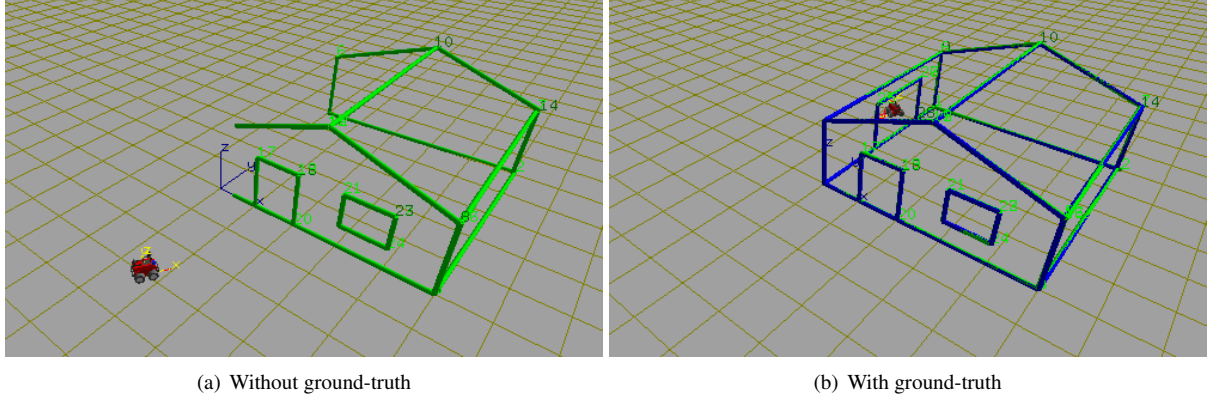


Figure 17: The simulated environment consists of a robot approaching a wireframe of a house. The figure shows a) the partially reconstructed house, after approximately 40 frames have been processed. Despite the preliminary state of the map, the structure is already visible. With this fair amount of data, a map made of points (with *e.g.* one point at each line intersection) would not convey very much information besides localization, because it lacks structural semantics such as connectivity and boundaries. b) Fully reconstructed house with the ground-truth in blue.

To evaluate the estimation’s consistency, a Monte Carlo analysis of the normalized estimation error squared (NEES) of the robot 6D-pose is made (Fig. 18(a)). The averaged NEES after 25 runs shows good consistency up to frame 100 and a riskier behavior from then on. This is in accordance with [29], which concludes that long-term EKF-SLAM is always inconsistent, providing evidence of the necessity of approaches using multiple local maps. Simulation results with multiple local maps are shown Figure 18(b), where the consistency has been clearly improved.

In order to evaluate the map-matching approach presented in Section 3.8, two robots are deployed in a simulation setting populated with line segments only (see adjoint video `mapMatchingLines.avi`<sup>8</sup>). The robots trajectories do not intersect until the end of the run, however two of their 50 lines sub-maps partially overlap. A map-matching event occurs at frame 377 (data association is known in this setup). The event takes place after 16 anchored Plücker lines from the aerial robot current frame ( $S_4$ ) are matched with a previous map built from the ground robot ( $S_2$ ). Lines with large uncertainties are not taken into account for the least square minimization. The transformation between the *lrfs* obtained with the proposed approach is  $\hat{s}_4^2 = (0.73, 14.42, -6.14, -1.178, -0.004, 0.006)^T$  in (m,m,m,rad,rad,rad). The real transformation between the reference 4th and 2th frames is  $s_4^2 = (0.66, 14.56, -6, -1.169, 0, 0)^T$ . Note that the orientation and the translation are well-estimated, but most importantly they are consistent. Figure 19 shows the 3D robots location and sub-maps before and after the map-matching event, and figure 20 shows the consistency plots for a single run of this setting.

## 4. Experimental Results

### 4.1. Setup

Outdoors data acquisitions in a large environment have been conducted to verify the performance of the proposed approach, with the ground robot “Dala” and the helicopter “Ressac” (Figure 21). The environment is an abandoned country village in the south of France, now used as a military training facility – see Figure 1. It is “semi-structured”, in the sense that it does not contain as many buildings as an urban area, and the building themselves do not contain many straight lines or perfect planar areas.

The ground robot Dala is an iRobot ATRV platform, equipped with a calibrated stereo-vision bench made of two 1024×768 cameras with a baseline of 0.35m. The helicopter Ressac is controlled by algorithms developed at Onera [43], and is also equipped with a calibrated stereo vision bench made of two 1024×768 cameras, with a 0.9m baseline.

Several thousands of images have been acquired while the helicopter Ressac is automatically achieving a swathing pattern at an altitude of about 40m, and Dala is making loop trajectories in the north-west group of buildings under manual control (Figure 24).

<sup>8</sup> <http://www-personal.acfr.usyd.edu.au/tvidal/videosRAS/mapMatchingLines.avi>

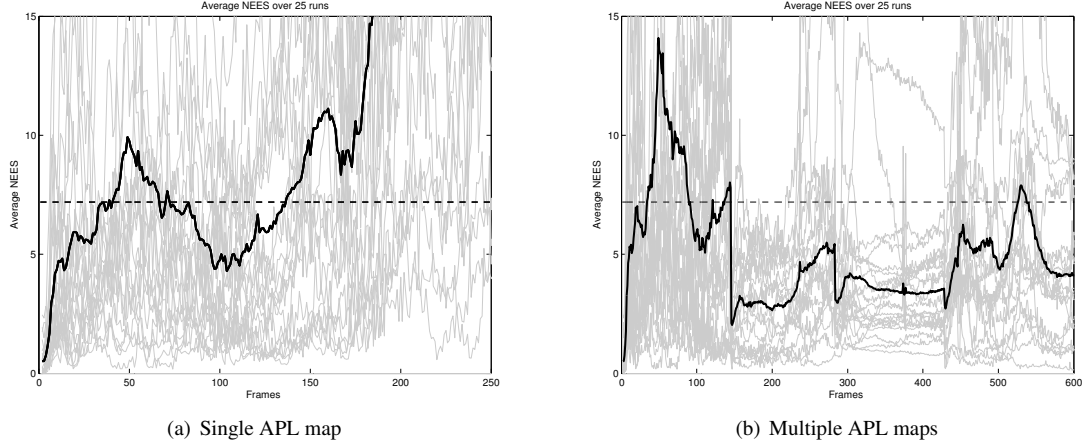


Figure 18: 25 Monte Carlo runs (thin gray lines) showing the NEES measures of the robot 6D-pose. The averaged NEES is in thick line. The dashed line corresponds to a NEES value of 7.18, which is the single-sided consistency limit of 95% confidence for 6 DOF and  $N=25$  runs, *i.e.*,  $\chi^2_{(6 \times 25)}(1 - 0.95)/25 = 7.18$ , see [29]. a) APL-SLAM without submapping. b) APL-SLAM using submapping; 4 maps are created.

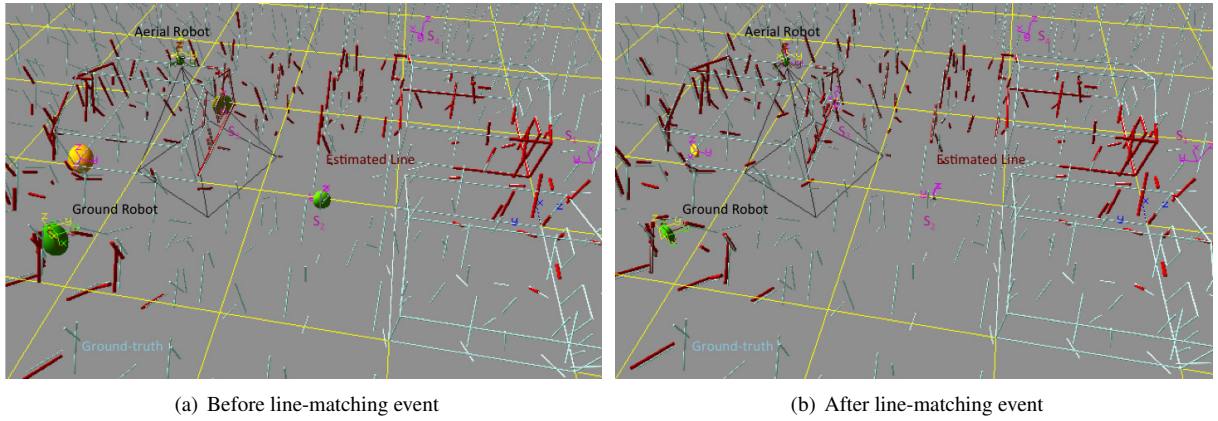


Figure 19: Event effect in the global map, the sub-maps origins expressed in the *wrf* are the large ellipsoids – 3D line segments landmarks are also shown here.

#### 4.2. Involved processes

The SLAM algorithms integrate two types of observations from only one camera; image points and image line segments. The mapped points are parametrized as inverse-depth points, and the mapped segments are parametrized as anchored Plücker line segments, as presented Section 3.

Unfortunately, because of engineering issues encountered during the data collection, no inertial or odometric motion estimates are available<sup>9</sup>. As a consequence, we use a visual odometry approach based on stereo vision for the motion prediction steps of the EKF SLAM algorithms – which in turn is the mean trough which the scale is recovered.

At each image acquisition, point observations are firstly processed: the resulting updated motion estimate is exploited by the line segment tracker, and line landmarks observations are then processed. Images are sub-sampled by a factor of 2 before being processed ( $640 \times 480$ ), and a heuristic is used to select the points that will be used as landmarks: the image is regularly partitioned in  $3 \times 3$  regions, in which one ensures that at least 2 landmarks are being tracked – one or two new interest points are selected as a landmark every time this is not satisfied. As for the lines, only the ones whose length is greater than 60 pixels are retained as landmarks.

<sup>9</sup>GPS ground truth could neither be recorded.

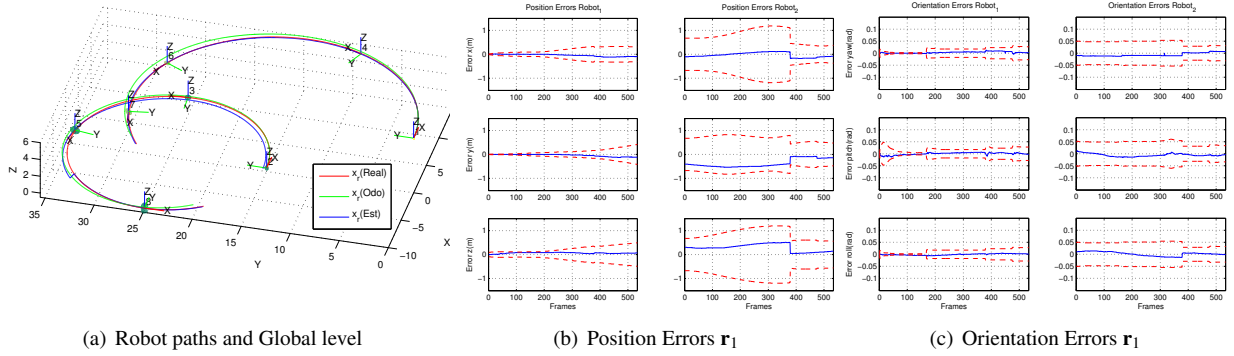


Figure 20: Multiple APL maps with line-matching event single run simulation results for a ground and an aerial robots. In a) the odometry is shown in green, real and estimated trajectories are shown in red and blue respectively.  $3\sigma$  ellipsoids are plotted on the basis of each  $lrf$ . b) shows the global position errors for each robot and their global  $3\sigma$  uncertainty bounds. c) shows the global orientation errors for each robot and their global  $3\sigma$  uncertainty bounds.



(a) Dala



(b) Ressac

Figure 21: Ground and aerial robots used for the experimental validation.

#### 4.3. Landmark detection, tracking and deletion

Point landmarks are Harris interest points that are matched from one view to the other with the group based matching procedure described in [44]: a first candidate match between two interest points is established using signal information (the two principal curvatures of the auto-correlation function), and confirmed if matches of neighboring points that satisfy geometric constraints on their location in the image are found.

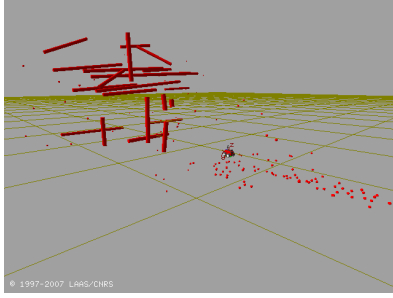
We use different initialization parameters for inverse depth point parametrization with Dala and Ressac. Dala's parameters are  $\rho_{init} = 0.1 \text{ m}^{-1}$  and  $\sigma_\rho = 0.2 \text{ m}^{-1}$ , while Ressac's parameters are  $\rho_{init} = 0.025 \text{ m}^{-1}$  and  $\sigma_\rho = 0.0125 \text{ m}^{-1}$  (the points are initialized at 40m, which is the helicopter average elevation over the terrain).

To extract and track line segments, we use a model-driven approach, that is very robust to scene and illumination variations, and that does not require the setting of any sensitive parameter (details can be found in [45]).

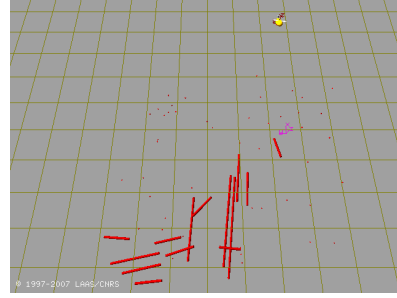
For the estimation part, the a priori parameters used in the experiment for the APL are  $\beta = (0.025, 0)$ ,  $\sigma_{\beta_1} = 0.025$  and  $\sigma_{\beta_2} = 0.0375$  for both robots. The prediction of the line segment position in the image, required by the segment tracker, is done using the projection of the 3D line segment into the image frame.

Smart landmark deletion is crucial for maintaining reliable maps consisting of only stable and consistent landmarks. We perform it at the local-map level, based on a test using three counters associated to each landmark:  $N_s$ , the number of match attempts (searches);  $N_m$ , the number of matches performed; and  $N_i$ , the number of validated matches (inliers). A landmark is deleted whenever the condition  $D(N_s, N_m, N_i) = (N_s > 10) \wedge [(N_m/N_s < 0.5) \vee (N_i/N_m < 0.5)]$





(a) Dala local map



(b) Ressac local map

Figure 22: Local maps built by the rober Dala (a) and the UAV Ressac (b).



(a) Dala local map



(b) Ressac local map



(c) Ressac/Dala combined map

Figure 23: Local maps superimposed on a aerial view of the scene: with the rover Dala (a), the UAV Ressac (b), and the combined submaps (c). Yellow ellipsoids represent the endpoints covariances. Note that only line segments with small covariances have been plotted.

holds true.  $D()$  basically ensures sufficient evidence with ( $N_s > 10$ ), and then checks if the landmark appearance OR visibility are unstable, with ( $N_m/N_s < 0.5$ ), or if the landmark estimate is inconsistent with the observations, with ( $N_i/N_m < 0.5$ ). The three thresholds (10, 0.5, 0.5) could be optimized in some way, however, with these initial values the impact on map quality and estimation consistency is notable.

#### 4.4. Local Maps

New local maps are created when 100 landmarks (combining points and line segments) are in the map. Immediately after, the current robot's pose is the new relative transformation in the global graph. Figures 22 and 23 shows examples of local maps built by Dala and Ressac: the maps consist of a set of 3D line segments and 3D points.

Figure 24 shows the estimated trajectories of the robots, superimposed to the aerial image of the area. For this run, no events that link the two robots graphs are considered.

#### 4.5. Enforcing loop closures

With this dataset, only very few line segments are mapped by both robots, which precludes the segment map matching. To define loop closing events, we therefore take advantage of the availability of stereovision data, and emulate with 3D point matches two types of events, rendezvous and data-matching (image to image):

- The rendezvous is emulated using matches of interest points perceived by the two robots using their current image frames. The 3D coordinates of the points obtained by stereovision yield the possibility to provide an estimate of the relative robot position.



(a) Dala trajectory in open loop

(b) Ressac trajectory in open loop

Figure 24: SLAM trajectories for Dala and Ressac in open loop, drawn on an aerial view of the experiment area (obtained on [www.geoportail.fr](http://www.geoportail.fr)). The actual trajectory of Dala fits the road, whereas here the trajectory estimate does not, although its origin as been set at the good position over the aerial view.

- The image-matching event recovers the relative transformation between the current robot pose and a past pose from a different robot (origin of a local sub-map, such as in Figure 6(a)), using also matches of interest points between their respective frames. Only key-frames corresponding to the origins of local sub-maps are used for the matching.

Once an event arises, a modified A-star algorithm is used to detect the occurrence of a cycle in the graph, and in particular the minimum cycle on which the loop constraint is applied (the algorithm searches for the minimal length path between the two nodes linked by the event, the existence of this link being ignored for the search).

Figures 25 to 28 show results obtained by the integration of events between Dala and Ressac. Dala starts at the entry of the north-west group of buildings, with no uncertainty in its local map, but also in the *wrf*: the first Dala sub-map is the origin of the world. Ressac starts above Dala, and heads towards south-east. A first rendezvous event occurs immediately after start, and Ressac is localized in Dala's reference frame.

A second event occurs after Ressac comes back from the south-east village, passing above a place previously visited by Dala. The effects of the image-matching event are shown in Figure 25. The figure also shows the image frames that were evaluated for the matching: new local maps are initiated afterward for both robots. Note that Ressac's uncertainty in height is pretty large, especially before the second event: the visual odometry used as predictions is indeed not very precise in the vertical direction, because Ressac stereo baseline is small with respect to the depth of the perceived points – and the integration of points and lines in the sub-maps does not reduces much the elevation estimates. However, after the data-matching event, the elevation of Ressac and the origins of all the built maps are strongly corrected. No map-merging is done. Figure 26 shows in the same plot the robots' trajectories as estimated by visual odometry, SLAM without any loop closure event integrated, and with the integration of the events.

The final global graph (origins of local maps with associated uncertainties in the *wrf*), and the robots' trajectory in *wrf* are shown Figure 27. This figure shows how the origins of Ressac's sub-maps around the same area (sub-map 5 and 27) have a similar uncertainty, as an effect of the loop closing event when it comes back. Note also that Ressac's 3D filtered path is off the trajectory defined by the sub-maps origins (mean of the ellipsoids): the path shown is the result of the EKF at each time instant, that is not corrected by the optimization algorithm. However, the full topology (origins of the local maps) is readjusted, as in a smoothing algorithm.

Finally, Figure 28 presents qualitative results of the proposed approach. Each robot has processed over 1200 images mapping approximately an area of  $300 \times 300$  m.

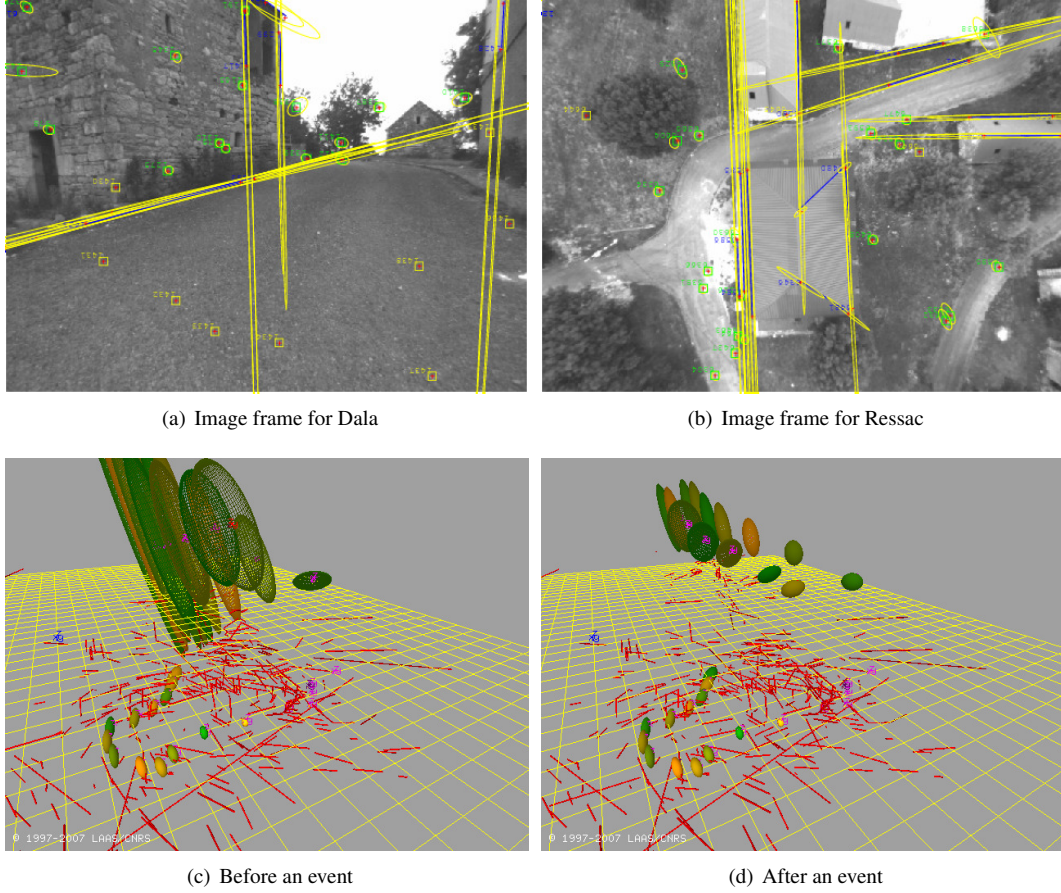


Figure 25: Top: Image frames from both robots before the event, approximately corresponding to the middle column images of figure 1. Green squares represent interest point currently considered as landmarks, yellow squares represent interest points just initialized as landmarks. The line segments are in blue, with endpoints in red. Yellow ellipses are the uncertainty in the image view. Bottom: Event effect in the global map, the sub maps origins expressed in the *wrf* are the large ellipsoids – only 3D line segments landmarks are shown here.

In the second part of the adjoint video `airGround.mov`<sup>10</sup>, the robots trajectory in *wrf* are shown. The south-east group of buildings is easy to spot in the video, it is the place where Ressac is turning 180°. Also, the video shows the processed image sequence for Dala and Ressac, along with the 3D landmarks, and local maps' origins in *wrf*.

## 5. Discussion

The contributions proposed in this paper fit well with the objective of deploying cooperative aerial / ground robotic systems: on the one hand, the distributed multi-map approach handles communication losses and can cope with large areas, and on the other hand, maps made of points and lines represent better the environment geometry than only points, and thus yield the possibility to match and fuse data acquired by both kind of robots.

The proposed solution to the localisation and mapping problem is based on a combination of filtering (within the local maps the robots poses are filtered) and smoothing (the origins of the local sub-maps are past robots' poses and whenever an event occurs the past poses are corrected). The mapping problem is therefore relegated within the local sub-maps, and is decorrelated from the global localization problem, making our approach akin to a cooperative localization approach, where robots improve their localization using the information provided by other robots [15].

<sup>10</sup> <http://www-personal.acfr.usyd.edu.au/tvidal/videosRAS/dalaRessacExp.mov>

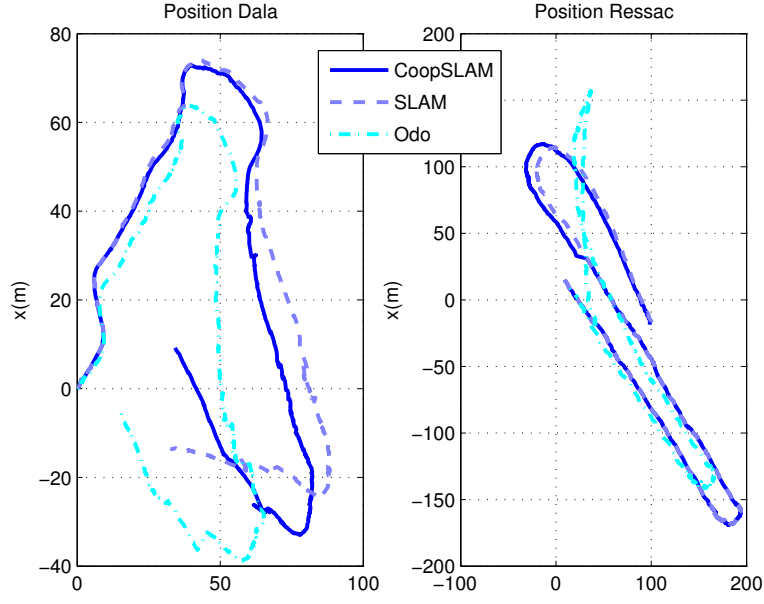


Figure 26: Comparative trajectory plots: odometry in dash-dot line, open loop run in dashed line and cooperative run for Dala (left) and Ressac (right).

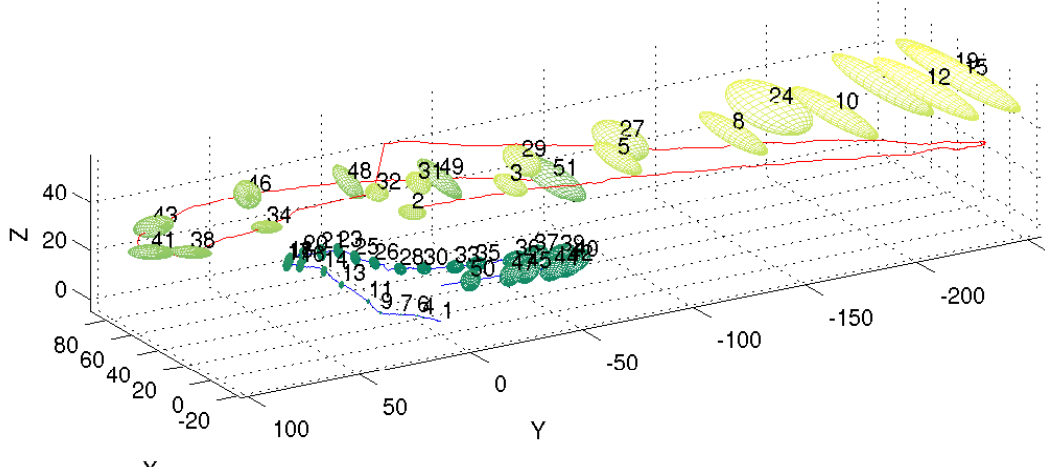


Figure 27: Final global graph (the global level) with Dala's trajectory in blue and Ressac's trajectory in red in the *wrf*.

In terms of cooperation, the marginals of graph level is the sole information that must be exchanged between the robots or to a central server. Future work considers the implementation of the distributed approach in a real scenario. The approach is well suited to a multi-robot context, and it can in particular handle *all* the possible localization means, from odometry to absolute localization with respect to an initial model. The fact that no information is shared between sub-maps leads to a loss of information, as faraway points and lines that are being tracked for long periods of time and between sub-map need to be re-initialized after a new sub-map is created. The conditional independent maps approach (CI-SLAM [46]) palliates this problem in the single robot case. But in the multi-robot case, this solution raises issues that remain to be studied – in particular, sharing information between the robots can not be straightforwardly done at the graph level.

In order to obtain the best localization for each robot in the incremental SLAM solution and to avoid any graph



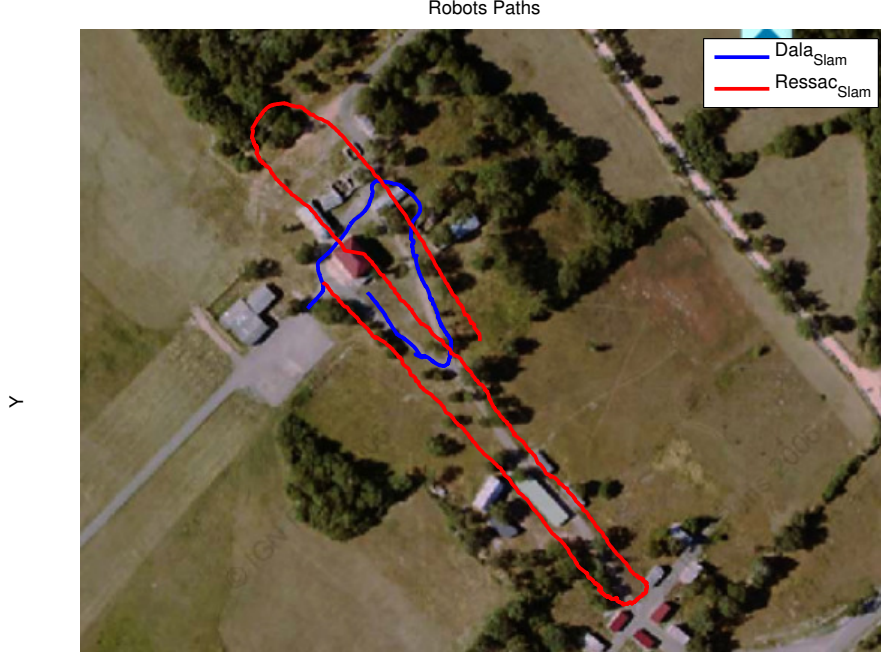


Figure 28: The thick-lines are the estimated trajectories for Dala and Ressac after the integration of two rendezvous/matching events. Ressac's trajectories are shown in red and magenta, and Dala's trajectories are shown in blue and cyan. Note that Dala's SLAM trajectory is now fitted the road.

incompatibility between robots, an alternative efficient implementation of a *posteriori* maximum likelihood solution should be considered instead of the minimal cycle optimization.

The map merging process is not required for the robots to operate, but should one require a global map, it can be done at the end of the mapping process. In our simulations, we applied a fusion in covariance form: fusion in information form [6] has however more interesting properties for decentralized systems, because information in the same reference frame is just added. Moreover, the sparseness of the information matrix's structure can be exploited to reduce communication bandwidth. Combining covariance and information forms might reduce the computation complexity, as shown in [47] which exploits the advantages of both methods.

In order to build landmark maps that are invariant to the vantage point, we have proposed to use 3D points and line segments. An important contribution of this paper is the new line segment parametrization for undelayed initialization, the anchored Plücker line, with which we exploited the two key concepts of inverse-distance and anchoring. The outcome is promising, but not fully satisfactory yet. Ongoing work is a more detailed analysis of different line parametrizations with the aim of further improving line estimation performances, most particularly with regard to filter consistency but also to other important aspects such as endpoints management [48].

The built maps combine inverse-depth points and anchored Plücker line segments. Essentially, as we stated the problem, detected line endpoints do not provide useful information as there is no guaranty that they represent stable 3D points. Therefore, one line in the image conveys the same amount of information as one point: 2DOF. In general, a set of  $N$  3D lines in general configuration provides the same amount of information as a set of the same number of points, also in general configuration. Having both points and lines means that such amount of information can be gathered with, say,  $N/2$  points and  $N/2$  lines. Within a complex scenario, there may be places where mostly points are visible, but other places where mostly lines are visible and trackable (structured scenarios for example). An algorithm that is able to exploit both types of landmarks is therefore in advantage in front of single-type ones.

The use of line segments is very promising, but yields a wireframe model that remains preliminary. Further post-processes can allow the building of a whole surfaces model, *e.g.* by collapsing point landmarks or by using homographies to verify plane hypotheses generated on the basis of pairs or triplets of coplanar segments.

Finally, much work remain to be done at the control level in order to efficiently deploy these mapping processes

among a fleet of heterogeneous robots. For instance, creating a new submap after every detected loop closure becomes totally inefficient when GPS is available, and besides sharing graphs every-time a connection between robots occurs, one must define *what* additional information has to be transmitted – *e.g.* which maps should be associated to trigger a loop closure? Mapping is an *active* process that must be controlled and supervised: this is even more necessary when it is distributed among several robots.

*Acknowledgments.* This work has been partially founded by the Action project (action.onera.fr). Cyrille Berger is founded by a Cifre contract with Thales Optronics. We would like to especially thank the Onera “Ressac team” for all their help for the experiments.

- [1] C. Estrada, J. Neira, J. Tardós, Hierarchical SLAM: Real-time accurate mapping of large environments, *IEEE Trans. Robot.* 21 (2005) 588–596.
- [2] T. Vidal-Calleja, C. Berger, S. Lacroix, Even-driven loop closure in multi-robot mapping, in: *IEEE Int. Conf. on Intelligent Robots and Systems*, Saint Louis, USA, 2009.
- [3] A. I. Mourikis, S. I. Roumeliotis, Predicting the accuracy of Cooperative Simultaneous Localization and Mapping (C-SLAM), *International Journal of Robotics Research* 25 (2006) 1273–1286.
- [4] X. Zhou, S. Roumeliotis, Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case, in: *IEEE Int. Conf. on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 1785–1792.
- [5] C. Estrada, J. Neira, J. Tardós, Finding good cycle constraints for large scale multi-robot SLAM, in: *In Proceedings of the International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 395–402.
- [6] S. Thrun, Y. Liu, Multi-robot slam with sparse extended information filters, in: *International Symposium on Robotics Research*, Siena, Italy, 2003.
- [7] E. Nettleton, S. Thrun, H. Durrant-Whyte, S. Sukkariéh, Decentralised slam with low-bandwidth communication for teams of vehicles, in: *4th international conference on Field and Service Robotics*, Coimbra, Portugal, 2006.
- [8] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, H. Younes, Coordination for multi-robot exploration and mapping, in: *Proceedings of the AAAI National Conference on Artificial Intelligence*, AAAI, Austin, TX, 2000.
- [9] A. Howard, Multi-robot simultaneous localization and mapping using particle filters, *Int. J. Robot. Res.* 25 (2006) 1243–1256.
- [10] S. Thrun, A probabilistic online mapping algorithm for teams of mobile robots, *International Journal of Robotics Research* 20 (2001) 335–363.
- [11] A. Gil, O. Reinoso, M. Ballesta, M. Juliá, Multi-robot visual slam using a rao-blackwellized particle filter, *International Journal of Robotics Research* 25 (2006) 1273–1286.
- [12] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, B. Stewart, Distributed multi-robot exploration and mapping, *Proceedings of the IEEE* (2006).
- [13] A. Howard, G. Sukhatme, M. Mataric, Multirobot simultaneous localization and mapping using manifold representations, *Proceedings of the IEEE - Special Issue on Multi-robot Systems* 94 (2006) 1360–1369.
- [14] F. Dellaert, A. Kipp, P. Krauthausen, A multifrontal QR factorization approach to distributed inference applied to multirobot localization and mapping, in: *Proceedings of the AAAI National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, 2005, pp. 1261–1266.
- [15] A. Bahr, M. R. Walter, J. J. Leonard, Consistent cooperative localisation, in: *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009.
- [16] S. I. Roumeliotis, G. A. Bekey, Distributed Multirobot Localization, *IEEE Trans. on Robotics and Automation* 18 (2002) 781–795.
- [17] H. J. Chang, C. G. Lee, Y. C. Hu, Y.-H. Lu, Multi-robot SLAM with topological/metric maps, in: *In Proceedings of the International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 395–402.
- [18] R. Madhavan, K. Fregene, L. Parker, Distributed cooperative outdoor multirobot localization and mapping, *Autonomous Robots* 17 (2004) 23–39.
- [19] S. B. Williams, H. Durrant-Whyte, G. Dissanayake, Constrained initialization of the simultaneous localization and mapping algorithm, *Int. J. Robot. Res.* 22 (2003) 541–564.
- [20] J. D. Tardós, J. Neira, P. M. Newman, J. J. Leonard, Robust mapping and localization in indoor environments using sonar data, *Int. J. Robot. Res.* 21 (2002) 311–330.
- [21] L. M. Paz, P. Jensfelt, J. D. Tardós, J. Neira, EKF SLAM updates in  $O(n)$  with Divide and Conquer SLAM, in: *IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, 2007.
- [22] J.-L. Blanco, J. González, J.-A. Fernández-Madriral, Subjective local maps for hybrid metric-topological SLAM, *Robotics and Autonomous Systems* 57 (2009) 64–74.
- [23] U. Frese, Treemap: An  $O(\log n)$  algorithm for indoor simultaneous localization and mapping, *Auton. Robot.* 21 (2006) 103–122.
- [24] R. C. Smith, P. Cheeseman, On the representation and estimation of spatial uncertainty, *Int. J. Robot. Res.* 5 (1986) 56–68.
- [25] R. Eustice, H. Singh, J. Leonard, Exactly sparse delayed-state filters for view-based SLAM, *IEEE Trans. Robot.* 22 (2006) 1100–1114.
- [26] M. Cummins, P. Newman, FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance, *The International Journal of Robotics Research* 27 (2008) 647–665.
- [27] M. Jordan, An introduction to Probabilistic Graphical Models, to appear, 2006.
- [28] J. Civera, A. Davison, J. Montiel, Inverse depth parametrization for monocular SLAM, *IEEE Trans. on Robotics* 24 (2008).
- [29] T. Bailey, J. Nieto, J. Guivant, M. Stevens, E. Nebot, Consistency of the EKF-SLAM algorithm, in: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 3562–3568.
- [30] R. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidya, M. Kim, Pose estimation from corresponding point data, *IEEE Transactions on Systems, Man, and Cybernetics* 19 (1989) 1426–1446.

- [31] P. Smith, I. Reid, A. Davison, Real-time monocular SLAM with straight lines, in: British Machine Vision Conf., volume 1, Edinburgh, 2006, pp. 17–26.
- [32] A. P. Gee, W. Mayol, Real-time model-based SLAM using line segments, in: LNCS proceedings of the 2nd International Symposium on Visual Computing, Lake Tahoe, USA, 2006.
- [33] E. Eade, T. Drummond, Edge landmarks in monocular SLAM, in: British Machine Vision Conf., Edimburgh, Scotland, 2006.
- [34] J. Solà, A. Monin, M. Devy, T. Lemaire, Undelayed initialization in bearing only SLAM, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Edmonton, Canada, 2005, pp. 2499–2504.
- [35] J. Solà, A. Monin, M. Devy, T. Vidal-Calleja, Fusing monocular information in multi-camera SLAM, IEEE Trans. on Robotics 24 (2008) 958–968.
- [36] J. Solà, T. Vidal-Calleja, M. Devy, Undelayed initialization of line segments in monocular SLAM, in: IEEE Int. Conf. on Intelligent Robots and Systems, Saint Louis, USA, 2009, pp. 1553–1558.
- [37] T. Lemaire, S. Lacroix, Monocular-vision based SLAM using line segments, in: IEEE International Conference on Robotics and Automation, Roma, Italy, 2007.
- [38] J. Solà, Consistency of the monocular EKF-SLAM algorithm for 3 different landmark parametrizations, in: IEEE Int. Conf. on Robotics and Automation, Anchorage, USA, 2010.
- [39] A. Bartoli, P. Sturm, The 3D line motion matrix and alignment of line reconstructions, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, Colorado Springs, USA, 2001, pp. 287–292.
- [40] B. Kamgar-Parsi, B. Kamgar-Parsi, Algorithms for matching 3D line sets, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 582–593.
- [41] A. Bartoli, P. Sturm, Structure from motion using lines: Representation, triangulation and bundle adjustment, Computer Vision and Image Understanding 100 (2005) 416–441.
- [42] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, in: Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, IEEE Computer Society, Nara, Japan, 2007, pp. 1–10.
- [43] P. Fabiani, V. Fuertes, A. Piquereau, R. Mampey, F. Teichteil-Kigsbuch, Autonomous flight and navigation of vtol uavs: from autonomy demonstrations to out-of-sight flights, Aerospace Science and Technology 11 (2007) 183 – 193.
- [44] T. Lemaire, C. Berger, I.-K. Jung, S. Lacroix, Vision-based slam: Stereo and monocular approaches, IJCV - IJRR (2006).
- [45] C. Berger, S. Lacroix, DSeg: Direct line segments detection, Technical Report, LAAS/CNRS, 2009.
- [46] P. Piniés, J. D. Tardós, Scalable slam building conditionally independent local maps, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, San Diego, CA, 2007.
- [47] C. Cadena, J. Neira, SLAM in  $O(\log n)$  with the combined kalman-information filter, in: IEEE Int. Conf. on Intelligent Robots and Systems, Saint Louis, USA, 2009.
- [48] J. Solà, T. Vidal-Calleja, J. Civera, Impact of landmark parametrization on monocular EKF-SLAM with points and lines, Technical Report, LAAS-CNRS, 2010. In preparation. Available: <http://hal.archives-ouvertes.fr/hal-00451778/fr/>.