Environment modeling for cooperative aerial/ground robotic systems

Teresa Vidal, Cyrille Berger, Joan Sola, and Simon Lacroix

Abstract This paper addresses the cooperative localization and visual mapping problem for multiple aerial and ground robots. We propose the use of heterogeneous visual landmarks, points and line segments. A large-scale SLAM algorithm is generalized to manage multiple robots, in which a global graph maintains the topological relationships between a series of local sub-maps built by the different robots. Only single camera setups are considered: in order to achieve undelayed initialization, we present a novel parametrization for lines based on anchored Plücker coordinates, to which we add extensible endpoints to enhance their representativeness. The built maps combine such lines with 3D points parametrized in inverse-depth. The overall approach is evaluated with real-data taken with a helicopter and a ground rover in an abandoned village.

1 Introduction

In a aerial / ground multi-robot context, as in any multi-robot context, the ability to build and share environment models among the robots is an essential pre-requisite 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

LAAS/CNRS
University of Toulouse
7, Ave du Colonel Roche
31077 Toulouse Cedex 4, France
Firstname.Name@laas.fr

in the three dimensions, but the determination of vantage points calls for visibility computations in the 3D space. Also, vision is here the primary environment 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 is the only sensor that can be exploited on-board micro-drones – that have undoubtedly a promising future.

Approach. Our aerial / ground context requires the resolution of the two following issues: on the one hand the mapping approach must be distributed so as to cope with the communications constraints, and on the other hand the map structure must allow data association and fusion, coping with the fact that the sensors or viewpoints can be very different among the different robots. The contribution of this paper is therefore twofold.

We first propose a distributed 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 work stems from 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, various events can trigger loop closures and later map merging, namely rendezvous between robots, landmark correspondences ("map matching") and absolute localizations provided by GPS fixes or by matches with an a priori existing map. These events exhibit a cycle on the graph of the map poses, and thus define constraints that allows the system to refine the estimates of the sub-maps origins.

Among the three possibilities to close loops in the overall graph of maps, map matching is the most difficult to achieve when maps have been built by heterogeneous robots, *i.e.* with different kinds of sensors or vantage points. To be able to match maps, we need to represent in maps characteristics of the environment that are invariant with respect to large viewpoint changes. For this purpose, we exploit line segments detected in the images in order to build 3D wireframe maps that can be matched among the robots. This calls for a dedicated implementation of the EKF-based SLAM approach, which relies on the undelayed initialization of anchored Plücker 3D lines.

Outline. Sections 2 and 3 present the localization and mapping approach to deal with multiple robots. The approach is based on the scalable SLAM approaches, known as hybrid or hierarchical, that consider sub-maps and graph levels. Section 4 is devoted to the building of a 3D wireframe model on the basis of line segments detected in monocular imagery. It introduces the anchored Plücker line parametrization to use segments as landmarks in a monocular EKF SLAM approach. Finally, Section 5 present results obtained with data gathered by a helicopter and a ground rover.

2 Hierarchical SLAM

The multiple robot localization and mapping problem is formulated using sub-maps in a similar manner to hierarchical SLAM in [1] or hybrid metric-topological SLAM in [2], where there are two levels: metric (local sub-maps), and topological (global graph).

The topological (global) level represents the relationships $\mathbf{s}_i^j$ between local maps i and j. The metric level contains the local maps, composed of the set of landmarks $\mathbf{m}_i$ and the current robot pose $\mathbf{x}_i$. Each local map stores information in its own *lrf*. 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 $\mathbf{s}_i^{i+1} = \mathbf{x}_i$. 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 $\mathbf{M}_i$ if it is required.

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

$$\mathbf{x}_{\mathbf{m}}^i = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{m}_i \end{bmatrix}, \tag{1}$$

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

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

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. Let us consider for now that no information is shared between these sub-maps, thus the new map starts in a *lrf* with robot's pose and error covariance equal to zero.

Topological Level. The global or topological 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 $\mathbf{s}_i^{i+1}$. 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 local maps, namely:

$$\hat{\mathbf{s}} = \begin{bmatrix} \hat{\mathbf{s}}_0^1 \\ \vdots \\ \hat{\mathbf{s}}_i^{i+1} \end{bmatrix}, \qquad \mathbf{P}_{\mathbf{s}} = \begin{bmatrix} \mathbf{P}_{\mathbf{s}_0^1} & 0 & 0 \\ 0 & . & 0 \\ 0 & 0 & \mathbf{P}_{\mathbf{s}_i^{i+1}} \end{bmatrix}. \tag{3}$$

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, $\mathbf{S}_{i+1} = \mathbf{S}_i \oplus \mathbf{s}_i^{i+1}$. The current position of the robot in *wrf* is computed as $\mathbf{X}_i = \mathbf{S}_i \oplus \mathbf{x}_i$. Also, the global map can be obtained through,

$$\mathbf{M}_i = \mathbf{S}_i \oplus \mathbf{m}_i \,. \tag{4}$$

Mean and covariances of the Gaussian estimates are obtained by regular linear-Gaussian propagation using the Jacobians of $\oplus$ and $\ominus$.

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 [3], where local maps are like landmarks hanging from robot poses in *wrf*. The main difference is associated to the fact that the state-space in our case contains relative poses $\mathbf{x}_i$, instead of absolute poses $\mathbf{S}_i$.

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:

$$\mathbf{h}(\mathbf{s}) = \mathbf{s}_0^1 \oplus \mathbf{s}_1^2 \cdots \oplus \mathbf{s}_i^0 = 0 \tag{5}$$

$$= \mathbf{S}_i \oplus \mathbf{s}_i^0 = 0 \,. \tag{6}$$

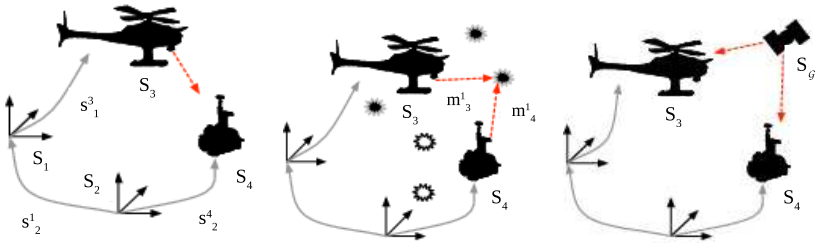
Given that $\mathbf{h}(\mathbf{s})$ is not linear due to the angular terms, the enforcement of this constraint can be formulated as a nonlinear constrained optimization problem. A solution for instance could be based on the Iterative EKF [1]: the part of the state involved in the loop closure at global level then becomes correlated, resulting in a non-sparse covariance matrix $\mathbf{P_s}$.

Note that for a single robot, local maps are obtained sequentially, hence the relative transformation between the local maps is given by the last robot pose in the current *lrf*, $\mathbf{s}_i^{i+1} = \mathbf{x}_i$.

3 Multiple robots

A hierarchical/hybrid SLAM approach in the multi-robot case seems a priori straightforward: each robot manages a set of sub-maps and a global graph of poses. But the interests of multi-robot mapping arise of course when the robots exchange mapping or position information, which allows to enhance the spatial consistency and to build up a *multi-robot global graph* of map poses (origins of local sub-maps).

In general when multiple robots are exploring the same area, they meet sooner or later or their maps partially overlap: these events will allow the system to establish connections between robot locations [4]. The events we



(a) Rendezvous event (b) Common landmark event (c) GPS fix

Fig. 1 Loop-closing events for multiple robots.

have identified are: *(i)* robots rendezvous (Figure 1(a)), *(ii)* common information match within sub-maps (Figure 1(b)), and *(iii)* receiving external information that provide absolute localizations (*e.g.* a GPS fix (Figure 1(c)), or feature matches with an existing environment model. The latter is not exactly a multi-robot loop closure, it provides a link between a *lrf* and a global geo-referenced frame: this yields the possibility to establish a link with another robot that has already been absolutely localized once.

All these events create a link between the robots' global levels. Whereas in a single robot case a loop closure only occurs when the robot revisits a previously mapped place, in a multi-robot case these events trigger loop closures: any cycle that appears in the overall graph defined by the concatenation of each robot graph (the *multi-robot graph*) is a loop closure. The compounding of all relative transformations that define a cycle is equal to zero as in Eq. 5, and a batch 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.

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. We will however see that in the *map-matching* case, the sub-maps that match must also be communicated: but they have naturally already been exchanged to establish the matches. Most importantly is that in the general case, only marginal distributions of each node has to be communicated, as opposed to the full joint distributions of the graph. As we will see later, each robot performs its own optimization based on the minimal cycle.

Robot Rendezvous. The event occurs when a robot observes another robot (partial rendezvous) or when both robots observe each other (full rendezvous). We will focus on the case when the relative transformation between two robots it is fully recovered, from the information obtained through a partial or full rendezvous.

New local maps are created in the instant of the rendezvous, then the current robot poses are promoted to the global level. 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, thus $\mathbf{z} = \mathbf{s}_j^i$.

Matching common information. There exist at least two different ways to match common information. For example, using information completely independent to the SLAM, *e.g.* image's descriptors matching (SIFT, SURF), image indexing techniques or scan matching (ICP). A common way to produce a map of poses [3] is to find the rotation and translation between two robot poses using one of these techniques, as opposed to tracking features. A second way to match common information is using the available information of the maps (position and uncertainty in global level). This is usually done using the current position of the robot or robots in the *wrf*.

- The "image to image" association produces a link directly between images, that are associated to certain poses $\mathbf{S}_i$ and $\mathbf{S}_j$. Note, the robot poses have to be part of the global graph when this event occurs.
  This is a simple, but effective manner to obtain the relationship between two robots, or even to close a loop with a single robot. Note that the observations are independent from the previous mapped information. As in the rendezvous case, this event produces the missing link $\mathbf{z} = \mathbf{s}_j^i$. In practical terms, image to image matching is equivalently to rendezvous.
- The "map to map" kind of data association requires both local maps to be transformed to a common frame, *e.g.* promoted to the global level as $\mathbf{M}_i$ and $\mathbf{M}_j$ using Eq. 4. In other words, the matching process happens in a common frame, being the *wrf* or one of the *lrf*. The disadvantage of this method is that in the worst case the absolute position of the two 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.

Absolute Localization. In an aerial/ground context, it is reasonable to assume that both kind of robots receive GPS fixes from time to time. The relative transformation provided by a GPS fix for vehicle i is simply $\mathbf{s}_i^{\mathcal{G}}$, where $\mathcal{G}$ is the geo-referenced frame. Such information provides a link between a *lrf* and a global geo-referenced frame, and generates a loop at the graph level for an individual robot.

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 sub-maps built by different robots, but an approximation for the sub-maps that are built by the same robot, the origin of the local sub-

map is the only state that changes after the constraint is applied. It can be easily shown that $\mathbf{m}_{i-1} \perp \mathbf{m}_i \mid \mathbf{x}_{i-1}$, *i.e.*, given the relative transformations, the consecutive local sub-maps are independent. Notice that the global poses $\mathbf{S}$ are d-separated of all possible paths between any pair of sub-maps $\mathbf{m}$ or even $\mathbf{M}$. Note that the approximation can be palliated using conditionally independent local maps as proposed in [5].

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*. In the case of receiving GPS fixes once in a while, the fact of starting a new local map at the time t when the fix is received, removes the dynamics aspects of the internal local maps setting a fix pose at global level.

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 prerequisite 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.

4 Line segments for monocular EKF-SLAM

We explore the possibility of using linear landmarks or *line segments*, which provide an improved semantic over points: lines inherently contain the notions of connectivity and boundary, which open the door to potential automatic interpretations of the environment, both at the metrical and topological levels. A 3D model based on lines, akin to a wireframe model, can further allow the possibility to build higher level entities (planes, closed regions, objects).

Plucker lines (PL). Plücker lines have been used in major vision works with straight 3D lines [6, 7]. These works, and other ones referenced therein, are based on Structure From Motion approaches. We showed in [8], drawing on previous work in [9], the way to employ Plücker lines to achieve undelayed initialization of lines in monocular SLAM.

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[1], $\mathcal{L} = (\mathbf{n} : \mathbf{v})$, with $\{\mathbf{n}, \mathbf{v}\} \in \mathbb{R}^3$, with which an intuitive geometrical interpretation of the line in 3D Euclidean space is possible (Fig. 2):

- The sub-vector $\mathbf{n}$ is a vector normal to the plane π supporting the line $\mathcal{L}$ and the origin $\mathcal{O}$.
- The sub-vector $\mathbf{v}$ is a director vector of the line.

[1] We use a colon (:) to separate the non-homogeneous and homogeneous parts in projective space.

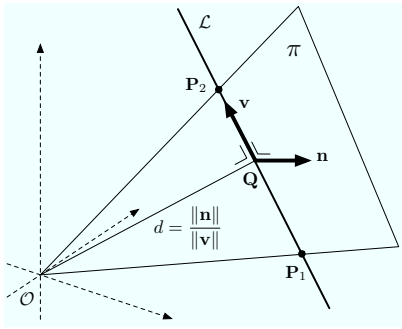


Fig. 2 Geometrical representation of the Plücker coordinates and sub-vectors $\mathbf{n}$ and $\mathbf{v}$. The line $\mathcal{L}$ and the origin $\mathcal{O}$ define the support plane π. The line's sub-vector $\mathbf{n} \in \mathbb{R}^3$ is orthogonal to π. The sub-vector $\mathbf{v} \in \mathbb{R}^3$ is a director vector of the line, and lies on π. This implies $\mathbf{n} \perp \mathbf{v}$. The closest point to $\mathcal{O}$ is $\mathbf{Q} = (\mathbf{v} \times \mathbf{n} : \mathbf{v}^\top \mathbf{v}) \in \mathbb{P}^3$. The distance from $\mathcal{L}$ to $\mathcal{O}$ is $d = \|\mathbf{n}\|/\|\mathbf{v}\|$, showing that $\mathbf{v}$ acts as the homogeneous part of $\mathcal{L}$, thus exhibiting inverse-depth properties.

- The distance from the line to the origin is given by $d = \|\mathbf{n}\|/\|\mathbf{v}\|$.

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 $\mathbf{v}$, something that will allow us to design appropriate initialization methods for EKF. For details on the use of Plücker lines in monocular EKF-SLAM see [8].

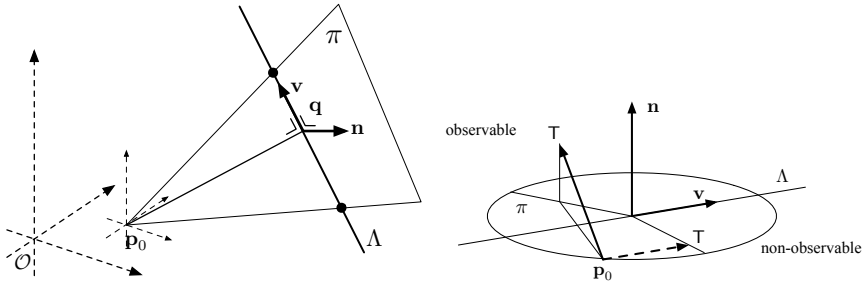
Anchored Plücker lines (APL). Now, we add an anchor to the parametrization to improve linearity, as it is done for points in the inverse-depth parametrization [10]. Anchoring the Plücker line means referring it to a point $\mathbf{p}_0$ in 3D space different from the origin (Fig. 3). The anchor point $\mathbf{p}_0$ is chosen to be the optical center at initialization time. The effect of such anchoring is that, on subsequent EKF updates, only the accumulated errors from the anchor $\mathbf{p}_0$ to the current camera position 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. 3) is then the 9-vector:

$$\Lambda = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{n} \\ \mathbf{v} \end{bmatrix} \in \mathbb{R}^9 \tag{7}$$

Transformation and projection expressions are as follows:

Frame transformation: Given a (camera) reference frame $\mathcal{C}$ specified by a rotation R and a translation T, an anchored Plücker line Λ in global frame is obtained from a line $\Lambda^{\mathcal{C}}$ in frame $\mathcal{C}$ with the affine transformation



(a) Geometrical interpretation of anchor $\mathbf{p}_0$ and sub-vectors $\mathbf{n}$ and $\mathbf{v}$. The closest point to the anchor is $\mathbf{q} = \mathbf{p}_0 + (\mathbf{v} \times \mathbf{n})/(\mathbf{v}^\top \mathbf{v})$

(b) Observability conditions. The line is 3D-observable if and only if the motion vector $(\mathsf{T} - \mathbf{p}_0)$ does not lie on the line's support plane π.

Fig. 3 The anchored Plücker line.

$$\Lambda = \begin{bmatrix} \mathsf{R} & 0 & 0 \\ 0 & \mathsf{R} & 0 \\ 0 & 0 & \mathsf{R} \end{bmatrix} \cdot \Lambda^{\mathcal{C}} + \begin{bmatrix} \mathsf{T} \\ 0 \\ 0 \end{bmatrix}. \tag{8}$$

The inverse transformation is performed with

$$\Lambda^{\mathcal{C}} = \begin{bmatrix} \mathsf{R} & 0 & 0 \\ 0 & \mathsf{R} & 0 \\ 0 & 0 & \mathsf{R} \end{bmatrix}^\top \cdot \Big(\Lambda - \begin{bmatrix} \mathsf{T} \\ 0 \\ 0 \end{bmatrix} \Big). \tag{9}$$

Un-anchoring: given an anchored Plücker line $\Lambda = (\mathbf{p}_0, \mathbf{n} : \mathbf{v})$, its corresponding (un-anchored) Plücker line $\mathcal{L}$ is computed with

$$\mathcal{L} = \begin{bmatrix} \mathbf{n} + \mathbf{p}_0 \times \mathbf{v} \\ \mathbf{v} \end{bmatrix} \tag{10}$$

Pin-hole projection: Projection is better expressed for regular Plücker lines. Given a perspective camera defined by the intrinsic parameters $\mathbf{k} = (u_0, v_0, \alpha_u, \alpha_v)$, a Plücker line $\mathcal{L}^{\mathcal{C}} = (\mathbf{n}^{\mathcal{C}} : \mathbf{v}^{\mathcal{C}})$, expressed in the camera's coordinate system, projects into a homogeneous line $\mathbf{l} \in \mathbb{P}^2$ in the image plane with the linear expression [6, 11]

$$\mathbf{l} = \mathcal{K} \cdot \mathbf{n}^{\mathcal{C}} \triangleq \begin{bmatrix} \alpha_v & 0 & 0 \\ 0 & \alpha_u & 0 \\ -\alpha_v u_0 & -\alpha_u v_0 & \alpha_u \alpha_v \end{bmatrix} \cdot \mathbf{n}^{\mathcal{C}} \in \mathbb{P}^2. \tag{11}$$

where $\mathcal{K}$ is called the *Plücker intrinsic matrix*.

Transformation and projection: Transformation and projection are accomplished with a transformation to the camera frame (9), un-anchoring (10), and projection (11). This can be composed in one single expression with:

$$\mathbf{l} = \mathcal{K} \cdot \mathsf{R}^\top \cdot (\mathbf{n} - (\mathsf{T} - \mathbf{p}_0) \times \mathbf{v}) \in \mathbb{P}^2, \tag{12}$$

in which we will notice:

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

Expression (12) gives us the means to analyze line observability as a function of the camera motion (Fig. 3(b)). Let $(\mathsf{T} - \mathbf{p}_0)$ be the camera motion since initialization time. Because the projected line $\mathbf{l}$ is expressed in projective space, any vector $\mathbf{l}' = \alpha\mathbf{l}$, with $\alpha \in \mathbb{R}$, represents the same line and the line sub-vector $\mathbf{v}$ is only observable if the vector $(\mathsf{T} - \mathbf{p}_0) \times \mathbf{v}$ is not proportional to $\mathbf{n}$. If we remind the Plücker constraint stating that $\mathbf{v} \perp \mathbf{n}$, this resumes to a motion $(\mathsf{T} - \mathbf{p}_0)$ that does not belong to the plane π. As the anchor $\mathbf{p}_0$ belongs to this plane, we conclude that the new camera position T must escape the plane π in order to fully observe the line.

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 $\mathbf{q} = \mathbf{p}_0 + \frac{\mathbf{v}\times\mathbf{n}}{\mathbf{v}^\top\mathbf{v}}$, the closest point to the anchor (see Fig. 3(a)). Given the line $\Lambda = (\mathbf{p}_0, \mathbf{n} : \mathbf{v})$ and abscissas $\{t_1, t_2\}$, the 3D Euclidean endpoints are obtained with

$$\mathbf{p}_i = \mathbf{q} + t_i \cdot \frac{\mathbf{v}}{\|\mathbf{v}\|} \in \mathbb{E}^3 \,,\ i \in \{1, 2\}. \tag{13}$$

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

Back-projection of a Plücker line: In the camera frame, the Plücker sub-vector $\mathbf{n}^\mathcal{C}$ resulting from the observation $\mathbf{l}$ is simply

$$\mathbf{n}^\mathcal{C} = \mathcal{K}^{-1}\mathbf{l}. \tag{14}$$

The sub-vector $\mathbf{v}^\mathcal{C}$ is not measured and must be obtained by injecting prior information. We give here the formulation and refer the reader to [8] where full explanations and justifications are provided. The sub-vector $\mathbf{v}^\mathcal{C}$ is obtained with

$$\mathbf{v}^\mathcal{C} = \mathbf{E}\beta, \tag{15}$$

where $\mathbf{E} \in \mathbb{R}^{3\times2}$ is a matrix transforming vectors $\beta \in \mathbb{R}^2$ in the Cartesian plane into the plane in $\mathbb{R}^3$ supporting the line, defined by the optical center and $\mathbf{n}^\mathcal{C}$. It is constructed as a base spanning the plane orthogonal to $\mathbf{n}^\mathcal{C}$, *i.e.*, the two non-measured DOFs,

$$\mathbf{E} = \begin{bmatrix}\mathbf{e}_1 \ \mathbf{e}_2\end{bmatrix}, \quad \mathbf{n}^\mathcal{C} \perp \mathbf{e}_1 \perp \mathbf{e}_2 \perp \mathbf{n}^\mathcal{C}, \tag{16}$$

the base vectors $\mathbf{e}_i$ being chosen so that $\mathbf{e}_1$ is parallel to the image plane,

$$\mathbf{e}_1 = \frac{\begin{bmatrix} n_2^{\mathcal{C}} & -n_1^{\mathcal{C}} & 0 \end{bmatrix}^\top}{\sqrt{(n_1^{\mathcal{C}})^2 + (n_2^{\mathcal{C}})^2}} \quad \text{and} \quad \mathbf{e}_2 = \frac{\mathbf{n}^{\mathcal{C}}}{\|\mathbf{n}^{\mathcal{C}}\|} \times \mathbf{e}_1. \tag{17}$$

The vector $\beta = (\beta_1, \beta_2)$ must be provided as prior. To help selecting an appropriate one, we give here some intuitive notions about β:

- A vector $\beta = (1, 0)$ is a line parallel to the image plane at a distance $d = \|\mathbf{n}^{\mathcal{C}}\|$ from the optical center.
- A vector $\beta = (0, 1)$ is a line perpendicular to the detected segment $\mathbf{l}$ in the image.
- The distance from the optical center to the line is given by $d = \|\mathbf{n}^{\mathcal{C}}\|/\|\beta\|$.

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,

$$\Lambda^{\mathcal{C}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{n}^{\mathcal{C}} \\ \mathbf{v}^{\mathcal{C}} \end{bmatrix}. \tag{18}$$

Back-projection and transformation: The operations above plus the transformation to the global frame (8) can be composed and written as a single-step function of R, T, $\mathbf{l}$ and β,

$$\Lambda = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{n} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathsf{T} \\ \mathsf{R}\mathcal{K}^{-1}\mathbf{l} \\ \mathsf{R}\mathbf{E}\beta \end{bmatrix}. \tag{19}$$

ALP initialization and update in EKF-SLAM. We now have expressed all the basic relations to deal with APL, that are useful to exploit them in an EKF framework. The line initialization and update equations are not depicted here, details can be found in [8].

5 Experimental results

The environment 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, see Figures 4.

The ground robot Dala is an iRobot's 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 [12], and is also equipped with a calibrated stereo vision bench made of two 1024×768 cameras, with a 0.9m baseline (Figure 5).

Fig. 4 Aerial view of the experiment area (obtained on www.geoportail.fr). The robot Dala evolves in the north-west group of buildings, while the helicopter Ressac is flying along a swathing pattern oriented diagonally between the north-west and south-east groups of buildings, at an elevation of about 40m. The red rectangle approximately represents the field of view of the image acquired by Ressac shown on top-right of the figure. On this latter image, the red angular sector shows approximately the field of view of Dala when taking the bottom-right image.

(a) Dala (b) Ressac

Fig. 5 Ground and aerial robots used for the experimental validation.

Involved processes. Unfortunately, because of engineering issues encountered during the data collection, no inertial or odometric motion estimates are available[2]. As a consequence, we use a visual odometry approach based on stereo vision for the motion prediction steps of the EKF SLAM algorithms – all the results presented hereafter have therefore been obtained using *exclusively* visual data.

The SLAM algorithms integrate two types of observations from only one camera; image points (parametrized as inverse-depth points) and image line

[2] GPS ground truth could neither be recorded.

segments (parametrized as anchored Plücker line segments). 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. A heuristic is used to select the points that will be used as landmarks: the image is regularly partitioned in 3×3 regions, in which one ensures that at least 2 landmarks are being tracked. As for the lines, all the ones whose length is grater than 60 pixels are retained as landmarks.

Point landmarks are Harris interest points, that are matched from one view to the other with the group based matching procedure described in [13]. We use different initialization parameters for inverse depth point parametrization with Dala and Ressac. Dala's parameters are $\rho_{init} = 0.1$ m^{-1} and $\sigma_\rho = 0.2$ m^{-1}, while Ressac's parameters are $\rho_{init} = 0.025$ m^{-1} and $\sigma_\rho = 0.0125$ m^{-1} (the points are initialized at 40m, which is its the helicopter average elevation over the terrain.

We use the algorithms presented in [14] to detect and track segments. 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 for the segment tracker, is done using the projection of the 3D line segment into the image frame.

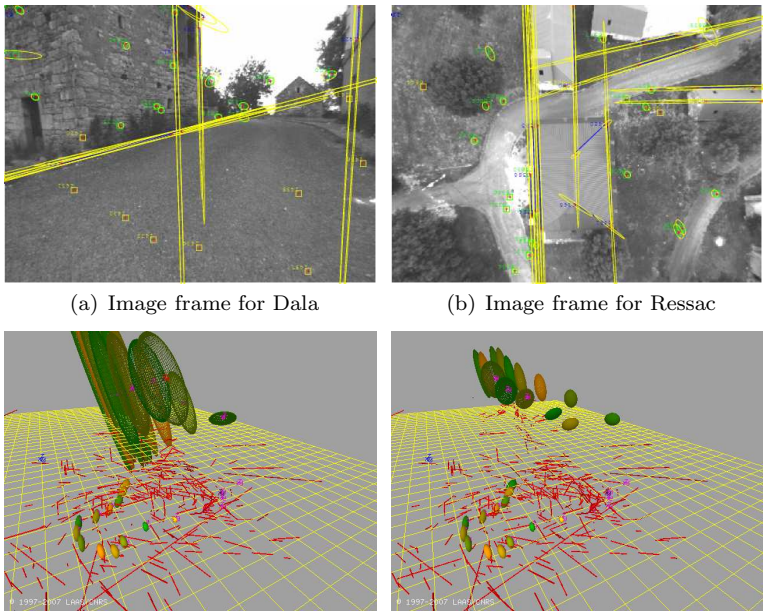
New local maps are created when 100 landmarks (combining points and line segments) are in the map. Immediately after, the current robot's pose is the new relative transformation in the global graph.

Enforcing loop closures. In the experiments, we have two types of events: rendezvous and image to image matching.

- 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 in each robot frame are obtained by stereo vision: as a result, the point matches yields an estimate of the relative robot position.
- The image matching event recovers the relative transformation between the current robot's pose and a past pose from a different robot (origin of a local sub-map), using also matches of interest points between their respective frames.

Figure 7 shows 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 to image matching event are shown in Figure 6. The figure also shows the image



(a) Image frame for Dala (b) Image frame for Ressac

(c) Before an event (d) After an event

Fig. 6 Top: Image frames from both robots before the event. 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.

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 motion estimates are 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 image to image matching event, the elevation of Ressac and the origins of all the built maps are strongly corrected.

6 Conclusion

We have explored the use of a multiple local maps technique for multi-robots according to a hierarchical SLAM approach, in which loop closures are integrated through an optimization process (an Iterative EKF). Loop closures are triggered using multi- or single-robot events such as finding information

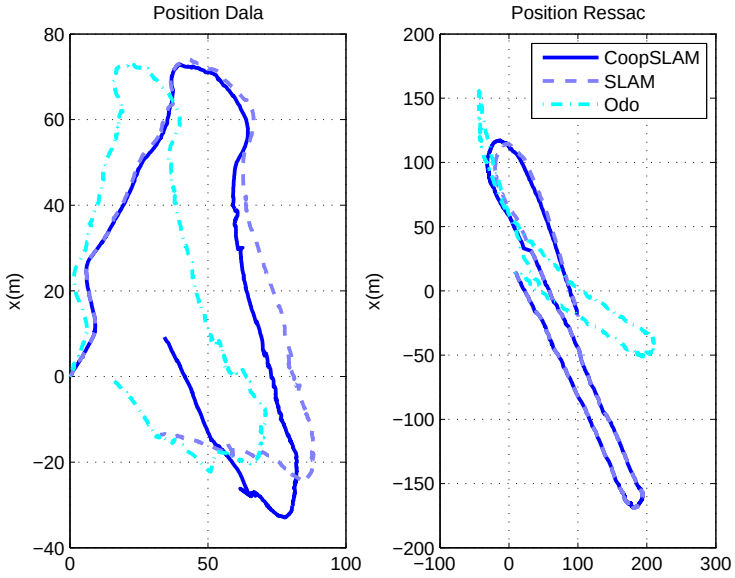


Fig. 7 Comparative trajectory plots: visual odometry in dash-dot line, open loop run in dashed line and cooperative run for Dala (left) and Ressac (right).

correspondences between unconnected local maps, rendezvous between two robots or GPS fixes. Therefore, the mapping problem is relegated within the local sub-maps, and is decorrelated from the global localization problem, making our approach akin to a cooperative localization approach. The approach is distributed, and the graph level is the sole information that must be exchanged between the robots. It 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.

In order to build more meaningful landmark maps and to be able to match data acquired from very different vantage points (or even different sensors), we have proposed to use line segments to build a wireframe model. An important contribution of this paper is the new line segment parametrization for undelayed initialization. We add an anchor to our previously proposed parametrization [8] to improve the linearity, exactly as it is done with inverse-depth points. Thanks to the cross-correlations stored in the EKF, the anchor allows the filter to account for accumulated errors only from the anchor to the current position, not from the origin of coordinates. Ongoing work is a more detailed analysis of different line parametrization in view of their linearity behavior.

Heterogeneous visual landmarks are proposed to map semi-structured environments. We combine inverse-depth points and anchored Plücker line seg-

ments. Our experiments show that inverse-depth points, that are numerous in the scene but not very robust to large viewpoint variations, play a crucial role for robot localization, while Plücker line segments, that are seldom in semi-structured environments but allow detection and matching from disparate viewpoints, are well suited to build a 3D model that exhibit some of the environment structure.

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.

References

1. C. Estrada, J. Neira, and J.D. Tardós. Hierarchical SLAM: Real-time accurate mapping of large environments. *IEEE Trans. Robot.*, 21(4):588–596, Aug. 2005.
2. José-Luis Blanco, Javier González, and Juan-Antonio Fernández-Madrigal. Subjective local maps for hybrid metric-topological SLAM. *Robotics and Autonomous Systems*, 57(1):64–74, 2009.
3. R.M. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Trans. Robot.*, 22(6):1100–1114, Dec. 2006.
4. Teresa Vidal-Calleja, Cyrille Berger, and Simon Lacroix. Even-driven loop closure in multi-robot mapping. In *IEEE Int. Conf. on Intelligent Robots and Systems*, Saint Louis, USA, 2009. To appear.
5. Pedro Piniés and J. D. Tardós. Scalable slam building conditionally independent local maps. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Diego, CA, Oct 29-Nov 2 2007.
6. Adrien Bartoli and Peter Sturm. The 3D line motion matrix and alilgnment of line reconstructions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 287–292, 2001.
7. Behzad Kamgar-Parsi and Behrooz Kamgar-Parsi. Algorithms for matching 3D line sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):582–593, 2004.
8. Joan Solà, Teresa Vidal-Calleja, and Michel Devy. Undelayed initialization of line segments in monocular SLAM. In *IEEE Int. Conf. on Intelligent Robots and Systems*, Saint Louis, USA, 2009. To appear.
9. T. Lemaire and S. Lacroix. Monocular-vision based SLAM using line segments. In *IEEE International Conference on Robotics and Automation, Roma (Italy)*, April 2007.
10. J. Civera, A.J. Davison, and J.M.M Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Trans. on Robotics*, 24(5), 2008.
11. J. Solà, André Monin, Michel Devy, and T. Vidal-Calleja. Fusing monocular information in multi-camera SLAM. *IEEE Trans. on Robotics*, 24(5):958–968, 2008.
12. P. Fabiani, V. Fuertes, A. Piquereau, R. Mampey, and F. Teichteil-Knigsbuch. Autonomous flight and navigation of vtol uavs: from autonomy demonstrations to out-of-sight flights. *Aerospace Science and Technology*, 11(2-3):183 – 193, 2007.
13. T. Lemaire, C. Berger, I-K. Jung, and S. Lacroix. Vision-based slam: Stereo and monocular approaches. *IJCV - IJRR*, 2006.
14. C. Berger and S Lacroix. DSeg: Direct line segments detection. Technical report, LAAS/CNRS, 2009.