Fusing Monocular Information in Multicamera SLAM

Joan Solà, André Monin, Michel Devy, and Teresa Vidal-Calleja

***Abstract*—This paper explores the possibilities of using monocular simultaneous localization and mapping (SLAM) algorithms in systems with more than one camera. The idea is to combine in a single system the advantages of both monocular vision (bearings-only, infinite range observations but no 3-D instantaneous information) and stereovision (3-D information up to a limited range). Such a system should be able to instantaneously map nearby objects while still considering the bearing information provided by the observation of remote ones. We do this by considering each camera as an independent sensor rather than the entire set as a monolithic supersensor. The visual data are treated by monocular methods and fused by the SLAM filter. Several advantages naturally arise as interesting possibilities, such as the desynchronization of the firing of the sensors, the use of several unequal cameras, self-calibration, and cooperative SLAM with several independently moving cameras. We validate the approach with two different applications: a stereovision SLAM system with automatic self-calibration of the rig's main extrinsic parameters and a cooperative SLAM system with two independent free-moving cameras in an outdoor setting.**

***Index Terms*—Calibration, image sequence analysis, Kalman filtering, machine vision, robot vision systems, stereovision.**

I. Introduction

THE SIMULTANEOUS localization and mapping (SLAM) problem, as formulated by the robotics community, is that of creating a *map* of the perceived environment while *localizing* oneself in it. The two tasks are coupled in such a way so as to benefit each other; a good localization is crucial to create good maps, and a good map is necessary for localization. For this reason, the two tasks must be performed *simultaneously*, and hence, the full acronym SLAM. In recent years, the maturity of both online SLAM algorithms, together with fast and reliable image processing tools from the computer vision literature, has crystallized into a considerable quantity of real-time demonstrations of visual SLAM.

In this paper, we insist on the quality of the achieved localization, which will impact in turn the map quality. The key to good localization is to ensure the correct processing of the geometrical information gathered by the cameras. In this long introduction, we present an overview of visual SLAM and related techniques to show that visual SLAM systems have historically discarded precious sensory information. We present a novel approach that uses the SLAM filter as a classical fusion engine that incorporates the full monocular information coming from multiple cameras.

Manuscript received June 15, 2007; revised May 8, 2008. First published xxx; current version published xxx. This paper was recommended for publication by Associate Editor J. Tardos (with approval of the Guest Editors) and Editor L. Parker upon evaluation of the reviewers' comments.

The authors are with the Laboratoire d'Analyse et d'Architecture des System, Centre National de la Recherche Scientifique (LAAS-CNRS), University of Toulouse, Toulouse 31077, France (e-mail: jsola@laas.fr; monin@laas.fr; michel@laas.fr; tvidal@laas.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2008.2004640

A. Monocular SLAM

Possibly, the best example of the aforementioned technological crystallization is monocular SLAM, a particular case of bearings-only (BO) SLAM (where the sensor does not provide any range or depth). It is well known that the reduction in system observability due to BO measurements has two main drawbacks: the loss of the scale factor and the delay in obtaining good 3-D estimates. Previous works either added some metric measurement to observe the scale factor, such as odometry [1] or the size of known perceived objects [2], [3], or have considered it irrelevant [4]. The delay in getting good 3-D estimates comes from the fact that such estimates require several BO observations from different viewpoints. This makes landmark initialization in BO-SLAM difficult, to the point that satisfactory methods able to exploit all the geometrical information provided by the cameras have only recently become available. We have witnessed an evolution of the algorithms as follows. First, *delayed landmark initialization* methods attempted to obtain a full 3-D estimate before initialization via several observations from different viewpoints. Davison [3] showed real-time feasibility of monocular SLAM with affordable hardware, using the original extended Kalman filter (EKF) SLAM algorithm for all but the unmeasured landmark's depth, and a separate particle filter to estimate this depth. Initialization was *deferred* to the moment when the depth estimate was good enough. The consequence of a delayed scheme is that we can only initialize landmarks with enough parallax, i.e., those that are close to the camera and situated perpendicularly to its trajectory, and therefore, the need to operate in room-size scenarios with lateral motions. Second, Solà *et al.* [1] showed that *undelayed landmark initialization* (mapping the landmarks from their first, partial observation) was needed when considering low parallax landmarks, i.e., those that are remote and/or situated close to the motion axis. This permits mapping larger scenes while performing frontal trajectories. Third, Civera *et al.* [5] have recently achieved the mapping of landmarks up to infinity, due to an undelayed initialization via an *inverse depth parameterization* (IDP). IDP has also been developed by Eade *et al.* [6] in a FastSLAM2.0 context. Today, the monocular SLAM systems exploit the geometrical information in its entirety: from the first observation, independently of the sensor's trajectory, and up to the infinity range.

B. Structure From Motion (SFM)

Monocular SLAM compares to a similar problem solved by the vision community: the structure from motion problem (SFM). In SFM, the goal is to determine, from a collection of images and up to an unrecoverable scale factor, the 3-D structure of the perceived scene and all 6-D camera poses from where the images were captured. When compared to SLAM, the structure plays the role of the map, while the set of camera poses defines all the successive observer's localizations.

Roboticists often claim that the main difference between SFM and SLAM is that the former is solved offline via the iterative nonlinear optimization method known as bundle adjustment (BA) [7], while the latter must be incrementally solved online, thus making use of stochastic estimators or *filters* that naturally provide incremental operation. This has been true for some years (today, SLAM is also solved online with iterative optimization [8]), but does not tell the whole story. The differences between SFM and SLAM are not only in the methods but also in the objectives, meaning that similar aspects of similar problems are given different priorities.

In particular, SFM exploits the visual information in its entirety without the difficulties encountered in monocular SLAM. Let us try to understand this curious fact. SFM puts the structure as a final objective, i.e., as a result of the whole process, and the emphasis is placed on minimizing the errors in the *measurement space*, thus using all the measured information. On the other hand, the SLAM map has a central role, with some of the operations (and particularly landmark initialization) being performed in map space, which is the system's *state space*. The fact that this state space is not statically observable, because it is of higher dimension than the observation space, leads to the difficulties exposed before. As an informal attempt to fill this gap, we could say that modern undelayed methods for monocular SLAM, with partial landmark initialization and partial updates, are almost equivalent to an operation in the measurement space: the information is initialized in the map space *partially*, i.e., exactly as it comes from the measurement space. A similar point of view over this concept can be found in [9].

C. Stereovision SLAM

Stereovision SLAM has also received considerable attention. The ability of a stereo assembly to directly and immediately provide 3-D landmark estimates allows us to use the best available SLAM algorithms and rapidly obtain good results with little effort in the conceptual parts. Such SLAM systems consider the stereo assembly as being a single monolithic sensor, capable of gathering 3-D geometrical information from the robot's surroundings, e.g. [10]. This fact, which appears perfectly reasonable, is the main paradigm that this paper questions. By considering two linked cameras as a single 3-D sensor, SLAM is unable to face the following two issues.

1) Limited 3-D Estimability Range: While cameras are capable of sensing visible objects that are potentially at infinity, a stereo rig provides only reasonably good 3-D estimates up to a limited range, typically from 3 m to a few tens of meters depending on the baseline. Because classical, nonmonocular SLAM algorithms expect full 3-D estimates for landmark initialization (i.e., they are reasoned in the map space), information belonging to only this limited region can be used for SLAM. This is really a pity; it is like if, having our two eyes, we were obliged to neglect everything outside a certain range from us, what we could call "*walking inside dense fog.*" Without remote landmarks, it is easy to lose spacial references, to become disoriented, and finally, find ourselves lost. Therefore, stereovision, as it is classically conceived, is a bad starting point for visual SLAM.

2) Mechanical Fragility: If we aim at extending the 3-D estimability range beyond these few tens of meters, we need to increase the stereo baseline while keeping or improving the overall sensor precision. This is obviously a contradiction: larger assemblies are less precise when using the same mechanical solutions. In order to maintain accuracy with a larger assembly, we must use more complex structures that will be either heavier or more expensive, if not both. The result for moderately large baselines ($>$1 m) is a sensor that is very easily decalibrated, and therefore, almost useless. Large rigs, however, are very interesting in outdoor applications because they allow farther objects to be positioned, thus making them contribute to the observability of the overall scale factor. This is especially true in aerial and underwater settings where, without nearby objects to observe, a small stereo rig provides no significant gain with respect to a single camera. Self-calibration can compensate for the inherent lack of stability of large camera rigs. It also allows multicamera platforms to start operation without undergoing a previous calibration phase, making on-field system deployment and maintenance easier.

To our knowledge, the only SLAM work that goes beyond the current stereoparadigm (apart from our conference paper [11]) is the one by Paz *et al.* [12], which uses a small-baseline, fully calibrated stereo rig. Matched features presenting significant disparity are initialized as classical Euclidean landmarks, while those presenting low disparities are treated with the inverse depth algorithm.

D. Visual Odometry (VO)

One could say that, in terms of methodology, visual odometry (VO) is to stereovision SLAM what SFM is to monocular SLAM. VO is conceived to obtain the robot's ego motion from a sequence of stereo images [13]. Visual features are matched across two or more pairs of stereo images taken during the robot motion. An iterative minimization algorithm, usually based on BA, is run to recover the stereo rig motion, which is then transformed into robot motion. For this, the algorithm needs to recover the structure of the 3-D points that correspond to the matched features. This structure is not exploited for other tasks and can be usually discarded. Remarkably, when the structure is coded in the measurement space (u, v, d), a disparity $d \to 0$ allows points at infinity to be properly handled [14]. This is also accomplished by using homogeneous coordinates [7]. VO must work in real time because robot localization is needed online.

Advanced VO solutions achieve very low drift levels after long distances by making use of: 1) hardware-based image processing with real-time construction and querying of large feature databases [15]; 2) dense image information matching via planar homographies and the use of the quadrifocal tensor [16]; or 3) bundle adjusting the set of N recent key frames together with additional fusion with an inertial measurement unit (IMU) [14].

E. Sensor Fusion in SLAM

The fact of SLAM being solved by filters allows us to envision SLAM systems as sensor fusion engines. Let us highlight some of the assets of filtering in sensor fusion.

1) *Multisensor operation:* Any number of differing sensors can be operated together in a consistent framework.
2) *Sensors self-calibration:* Unknown biases, gains, and other sensor's parameters can be estimated provided that they are observable [17].
3) *Desynchronized operation:* The data rates of all these sensors do not need to be synchronized.
4) *Decentralized operation:* Advanced filter formulations such as those using channel filters [18] achieve a decentralized operation that should permit live connection and disconnection of sensors without the need for filter reprogramming or reparameterization.

This paper explores the first three points for the case of multiple cameras.

SLAM systems naturally fuse information from both proprioceptive (odometry, GPS, and IMU) and exteroceptive (range scanners, sonar, and vision) sensors into the map. But our interest here is in fusing several exteroceptive sensors. We can distinguish two cases.

1) *Sensors of different kind:* When using differing sensors (e.g., laser plus vision), the main problem is in finding a map representation well adapted to the different kinds of sensory data (i.e., the data association problem).
2) *Sensors of the same kind:* The perceived information is of the same nature. This makes appearance-based matching possible, and therefore, makes map building easier. Nevertheless, most of such SLAM systems do not take advantage of fusion. Instead, the extrinsic parameters linking the sensors are calibrated offline, and the set of sensors is treated as a single supersensor. This is the case for two 180° range scanners simulating a 360° one, and for the previously mentioned stereo rig simulating a 3-D sensor. A sensor-fusion approach in these cases should naturally bring the aforementioned advantages to the SLAM system.

F. Multicamera SLAM and the Aim of This Paper

The key idea of this paper is very simple: by employing the SLAM filter as a fusion engine, we will be able to use any number of cameras in any configuration. And, by treating them as BO sensors with the modern undelayed initialization methods, we will extract the entire geometrical information provided by the images. The filter—not the sensor—will be responsible for making the 3-D properties of the perceived world arise.

Applications may vary from the simplest stereo system, through robots with several differing cameras (e.g., a panoramic one for localization and a perspective one looking forward for reactive navigation), to multirobot cooperative SLAM where BO observations from different robots are used to determine the 3-D locations of very distant landmarks. Although there certainly exist issues concerning multicamera management, the main ideas we want to convey may be demonstrated with systems of just two cameras. In this paper, we will illustrate two cases: first, the case of a robot equipped with a stereo rig, with its cameras being treated as two individual monocular sensors and second, two cameras moving independently and mapping together an outdoors scene.

This paper draws on previous work published in the conference paper [11] and the author's Ph.D. thesis [19]. These two works use the federated information sharing algorithm (FIS) in [1] to initialize the landmarks, which has been surpassed by the inverse depth methods (IDP) [5]. The present paper takes and extends all this research by developing a better founded justification (providing a wider scope to the proposed concepts), by improving on the implementation with the incorporation of IDP in the algorithms, and by extending the experimental validation to a cooperative monocular SLAM setup.

This paper is organized as follows. Section II presents the main ideas that will be exploited later and revises some background material for monocular SLAM. Section III explains how to set up multicamera SLAM, an application for stereo benches with self-calibration, and an application for two collaborative cameras. Section IV presents the perception and map management techniques used. Sections V and VI show the experimental results, and finally, Section VII gives conclusions and future directions.

II. 3-D Estimability in Visual SLAM

In this section, we present the ideas that support our approach to visual SLAM. We make use of the concept of estimability, which will help understand the abilities of vision for observing 3-D structure in the presence of uncertainty. We clarify the key properties of undelayed initialization in monocular SLAM, and remark its importance in multicamera SLAM. We also remind the key aspects of IDP-SLAM.

A. Geometrical Approach to 3-D Estimability

We are interested in finding the shape and dimensions of the 3-D-estimable region defined by two monocular views.

For this, we start with a couple of ideas to help understanding the concept of estimability used. When a new feature is detected in an image, the backprojection of its noisy-measured position defines a conic-shaped *pdf* for the landmark position, called *ray*, which extends to infinity (see Fig. 1). Let us consider two features extracted and matched from a pair of images, corresponding to the same landmark: their backprojections are two conic rays A and B that extend to infinity. The angular

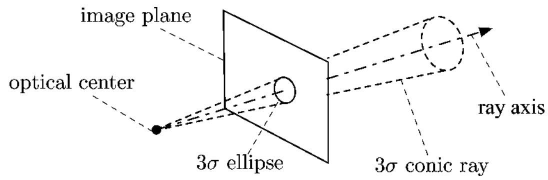

Fig. 1. Conic ray backprojects the elliptic representation of the Gaussian 2-D measure. It extends to infinity.

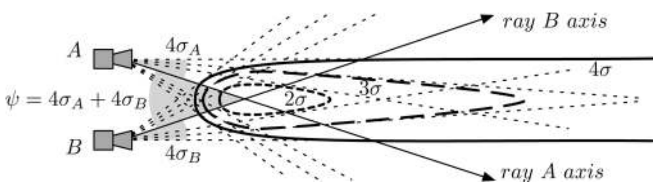

Fig. 2. Different regions of intersection for (solid) 4σ, (dashed) 3σ, and (dotted) 2σ ray widths when the outer 4σ bounds are, parallel. (Shaded) The parallax or angle between rays axes A and B is $\psi = 4\,\sigma_A + 4\,\sigma_B$.

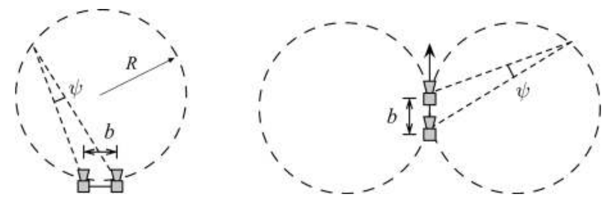

Fig. 3. Simplified depth estimability regions in a (left) stereo rig and (right) a camera traveling forward. The angle ψ is the one that assures estimability via triangulation from different viewpoints. The maximum range is $2R = b/\sin(\psi/2)$.

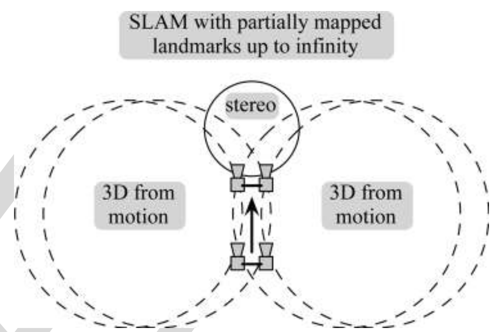

Fig. 4. Simplified depth estimability for a stereo rig moving forward. On both sides, estimability depends on the baseline gained by motion. In front, by stereo. Out of these bounds and up to infinity, landmarks are mapped partially. SLAM keeps incorporating the visual information due to the undelayed monocular methods, i.e., IDP in our case.

widths of these rays can be defined as a multiple of the standard deviations σ_A and σ_B of the angular errors (a composition of the cameras extrinsic and intrinsic parameters errors, and of the image processing algorithms accuracy). Informally speaking, we may say that the landmark's depth is fully estimated if the region of intersection of these rays is both *closed* and *sufficiently small*. If we consider, for example, the case where the two external 4σ bounds of the rays are parallel (see Fig. 2), then we can assure that the 3σ intersection region (which covers 98% probability) is *closed* and that the 2σ one (covering 74%) is *closed and small*. The ratio between the depth's standard deviation and its mean (a measure of linearity in monocular EKF-SLAM [1], [3]) is then better than 0.25. The *parallax* angle ψ between the two rays axes is therefore $\psi = 4(\sigma_A + \sigma_B) = \text{constant}$. This is the minimum parallax for full estimability.

In 2-D, we can plot the locus of constant estimability. In the case, where σ_A and σ_B can be considered constant, ψ is constant too, and from the inscribed angle theorem, the locus is then circular (Fig. 3, see also [19]). Landmarks inside this circle are considered *fully estimable*—and *partially* outside. In 3-D, the *fully 3-D estimable* region is obtained by revolution of this circle around the axis joining both cameras, producing a torus-shaped region with a degenerated central hole. This shape admits the following interpretations.

1) In a stereo configuration or for a lateral motion of a moving camera (see Fig. 3, left), the estimable region is located in front of the sensor. Beyond the region's border stereo provides no profit: if we want to consider distant landmarks, we have to use undelayed monocular techniques.
2) Depth recovery is impossible in the motion axis of a single camera moving forward (Fig. 3, right). Close to this axis, estimability is possible only if the region's radius becomes very large. This implies the necessity of very large displacements of the camera during the initialization process. Again, this can be accomplished only with undelayed initializations.
3) By combining both monocular and stereovision, we get an instant estimability of close frontal objects while still utilizing the information of distant ones (see Fig. 4). Landmarks lying outside the estimability regions are not 3-D-estimable but, when initialized using undelayed monocular methods, they will contribute to constrain the camera orientation. Ideally, long-term observations of stable distant landmarks would completely cancel orientation drift (visual compass).

B. Monocular IDP-SLAM

The core algorithm of this paper is an EKF-SLAM with an IDP of landmarks during the initialization phase, as described in [5]. In IDP-SLAM, partially observed landmarks are coded as a 6-D-vector,

$$\mathbf{i} = [\mathbf{x}_0, \theta, \psi, \rho] \tag{1}$$

where $\mathbf{x}_0$ is the 3-D position of the camera at initialization time, (θ, ψ) are the elevation and azimuth angles in global frame defining the direction of the landmark's ray, and ρ is the inverse of the Euclidean distance from $\mathbf{x}_0$ to the landmark's position (notice that ρ is usually known as *inverse depth* but it is rather an inverse distance). After the first observation, all parameters of $\mathbf{i}$ except ρ are immediately observable, and their values and covariances are obtained by proper inversion and linearization of the observation functions. The inverse depth ρ is initialized with a Gaussian $\mathcal{N}(\rho - \bar{\rho}; \sigma_\rho^2)$ such that in the depth dimension $s = 1/\rho$, we have

$$s_{(-n\sigma)} = \frac{1}{\bar{\rho} - n\sigma_\rho} = \infty \tag{2}$$

$$s_{(+n\sigma)} = \frac{1}{\bar{\rho} + n\sigma_\rho} = s_{\min} \tag{3}$$

with $s_{\min}$ the minimum considered depth and n the inverse depth shape factor. This gives $\bar{\rho} = 1/(2s_{\min})$ and, more remarkably

$$n\,\sigma_\rho = \bar{\rho}. \tag{4}$$

Importantly, values of $1 \le n \le 2$ assure from (2) that the infinity range is included in the parametrization with ample probability.

On subsequent updates, IDP achieves correct EKF operation (i.e., quasi-linear behavior) along the whole ray as long as the parallax shown by the new viewpoint is not too large. The linearity test in [20] is regularly evaluated. If passed, the landmark can be safely transformed into a 3-D Euclidean parametrization.

III. Multicamera SLAM

The general scheme for the multicamera SLAM system is presented in this section. This scheme is particularized to deal with two different problems. The first one is the automatic self-calibration of a stereo rig while performing SLAM. The second one is a master–lave solution to cooperative monocular SLAM. Both setups are explained here, and their corresponding experiments are presented in Sections V and VI.

A. System Overview

We implement the multicamera SLAM system as follows. A central EKF-SLAM will hold the stochastic representation of the set of all cameras $\mathcal{C}_i$ plus the set of landmarks $\mathcal{L}_j$

$$X^\top = [\mathcal{C}_1^\top \quad \cdots \quad \mathcal{C}_N^\top \quad \mathcal{L}_1^\top \quad \cdots \quad \mathcal{L}_M^\top] \tag{5}$$

where the cameras states contain position and orientation quaternion $[\mathcal{C}_i = (\mathbf{r}_i, \mathbf{q}_i) \in \mathbb{R}^7]$, and landmarks can be coded either in inverse depth ($\mathcal{L}_j = \mathbf{i}_j \in \mathbb{R}^6$) or in Euclidean coordinates ($\mathcal{L}_j = \mathbf{p}_j \in \mathbb{R}^3$). Any number of cameras can be considered this way. As each camera needs to remain localized properly, it needs to observe a minimum number of landmarks at each frame. The algorithm's complexity increases linearly with the number of cameras if this number is small with respect to the map.

For camera motions, we consider two possible models. In the first one, a simple odometer provides motion predictions $[\Delta x, \Delta y, \Delta \psi]$ in the robot's local 2-D plane. Gaussian uncertainties are added to the 6-DOF linear and angular components $[x, y, z, \phi, \theta, \psi]$ with a variance proportional to the measured forward motion Δx

$$\{\sigma_x^2, \sigma_y^2, \sigma_z^2\} = k_L^2 \cdot \Delta x \tag{6}$$

$$\{\sigma_\phi^2, \sigma_\theta^2, \sigma_\psi^2\} = k_A^2 \cdot \Delta x. \tag{7}$$

The variance in $[\phi, \theta, \psi]$ is mapped to the quaternion space using the corresponding Jacobians.

The second model is a 6-DOF constant velocity model

$$\begin{aligned} \mathbf{r}^+ &= \mathbf{r} + \mathbf{v}\,\Delta t \\ \mathbf{q}^+ &= \mathbf{q} \times \mathrm{v2q}(\omega\,\Delta t) \\ \mathbf{v}^+ &= \mathbf{v} + \eta_v \\ \omega^+ &= \omega + \eta_\omega \end{aligned}$$

where $(\,)^+$ means the updated value, $\times$ is the quaternions product, and $\mathrm{v2q}(\omega\,\Delta t)$ transforms the local incremental rotation vector $\omega\,\Delta t$ into a quaternion (quaternions are systematically normalized). This way, the camera state vector $\mathcal{C}_i$ is augmented to $\mathcal{C}_i = (\mathbf{r}_i, \mathbf{q}_i, \mathbf{v}_i, \omega_i) \in \mathbb{R}^{13}$. At each time step, perturbations $\{\eta_v, \eta_\omega\} \sim \mathcal{N}(0; \{\sigma_v^2, \sigma_\omega^2\})$ add variances to the linear and angular velocities proportionally to the elapsed time Δt

$$\sigma_v^2 = k_v^2 \cdot \Delta t \tag{8}$$

$$\sigma_w^2 = k_\omega^2 \cdot \Delta t. \tag{9}$$

The events of camera motion, landmark initialization, and landmark observation are handled as in regular IDP-SLAM by just selecting the appropriate block elements from the SLAM state vector and covariances matrix, and applying the corresponding motion or observation models. For example, at the observation of landmark j from camera i, we would use the function $\mathbf{u}_j^i = \mathbf{h}(\mathcal{C}_i, \mathcal{L}_j)$, which will be explained later for the case of an IDP ray [see 11]. Before transforming IDP rays into points, the linearity test in [20] needs to hold for all cameras.

B. Stereo SLAM With Extrinsic Self-Calibration

Our approach is relevant to fully calibrated stereo rigs if they are small (10–20 cm, as in [12]) or if, having long baselines, their main extrinsic parameters can be continuously self-calibrated.

Not all of the six extrinsic parameters of a stereo rig (three for translation, three for orientation) need to be calibrated. In fact, the notion of *self-calibration* inherently requires the system to possess its own gauge. In our case, the metric dimensions or *scale factor* of the whole world–robot system can only be obtained either from the stereo rig baseline, which is one of the extrinsic parameters (then, it makes no sense to self-calibrate the gauge), or from odometry, which is often much less accurate than any coarse measurement we could make of this baseline. Additionally, as cameras are actually angular sensors, vision measurements are much more sensitive to the cameras orientations than to any translation parameter. This means that vision measurements will contain little information about these translation parameters. In consequence, self-calibration may concern only orientation, and more precisely, the orientation of one camera with respect to the other. The error of the reconstructed map's scale factor will be the same as the relative error of the baseline measurement.

With these assumptions, our self-calibration solution is straightforward: for the second camera, we just include its orientation in the map and let EKF make the rest. The state vector (5) is modified and written as

$$X^\top = [\mathcal{R}^\top \quad \mathbf{q}_R^\top \quad \mathcal{L}_1^\top \quad \cdots \quad \mathcal{L}_M^\top]$$

where $\mathcal{R}$ and $\mathcal{L}_1 \cdots \mathcal{L}_M$ are the robot pose and landmarks map. The left camera pose $\mathcal{C}_L$ has a fixed transformation with respect to the robot, and $\mathbf{q}_R$ is the orientation part of the right-hand camera $\mathcal{C}_R$ in the robot frame. The time-evolution function of the angular extrinsic parameters is simply $\mathbf{q}_R^+ = \mathbf{q}_R + \gamma$, where γ is a white, Gaussian, low-energy process noise that accounts for eventual decalibrations, e.g., due to vibrations. For short-duration experiments, we set $\gamma = 0$. A coarse analysis of the stereo structure's mechanical precision will be enough to set the initial uncertainty to a value of the order of 1° or 2° per axis. This can be reduced to a few tenths of degree in cases where we dispose of previous calibrated values about which we are not confident anymore.

C. *Cooperative Multicamera SLAM*

The ideal, most general case of cooperative SLAM (5), corresponds to a (not too large) number of cameras moving independently. Each camera is able to manage its own measurements and communicates directly with the map. The aim of this communication is to obtain information about existing landmarks to get localized, and provide information about new or reobserved landmarks. This way, the algorithms to be executed by each camera are absolutely symmetrical, without any kind of hierarchy. A simplified implementation considers cameras with different privileges.

In our particular case, the cooperative SLAM system considers two cameras. One of them takes the role of *master,* and is responsible for all landmarks detection and initialization. The second one acts as the *slave*. It follows the master at a close distance and reobserves the SLAM map that is being built by the master. By doing so, it provides a second viewpoint to landmarks just initialized, accelerating the convergence of the map. The master and slave trajectories are highly independent, and for instance, they can cross paths. The only requirement is to look in the same direction. A trivial extension to more than two cameras consists in including additional slaves.

IV. Perception and Map Management

Active search (AS, nicely described in [21] and also referred to as *top-down* in [6]) is a powerful framework for real-time image processing within SLAM. It has been successfully used in several monocular SLAM works [3], [5], [11], using a diversity of techniques for landmark initialization. The idea of AS is to exploit the information contained in the map to predict a number of characteristics of the landmarks to observe. AS is helpful in solving the following issues:

1) selecting interesting image regions for initialization;
2) selecting the most informative landmarks to measure;
3) predicting where in the image they may be found, and with which probability;
4) predicting the current landmark's appearance to maximize the chances of a successful match.

A. *Feature Detection and Initialization*

Based on the projection of the map information into the master image, a heuristic strategy is used to select a region of interest for a new initialization: we divide the image with a grid and randomly select a grid element with no landmarks inside. We extract the strongest Harris point [22] in this region and validate it if its strength is above a predefined threshold. We store a small rectangular region or *patch* of 15×15 pixels around the point as the landmark's appearance descriptor, together with the pose of the camera. Finally, we initialize the IDP ray in the SLAM map.

B. *Expectations: The Active Search Regions*

Some considerations about AS can be made for its usage in multicamera IDP–SLAM to improve performance. We use for this the $\mathcal{E}_1$ and $\mathcal{E}_\infty$ ellipses, defined and explained as follows.

1) $\mathcal{E}_1$ Ellipse: Expectation of the Inverse Depth Ray: The inverse depth ray (1) is easily projected into a camera. We take the transformation to camera frame given in [5]:

$$\mathbf{h}_1^{\mathcal{C}} = \mathbf{R}(\mathbf{q})^\top \left(\rho \left(\mathbf{x}_0 - \mathbf{r}\right) + \mathbf{m}(\theta, \psi)\right) \tag{10}$$

where $\mathbf{R}(\,)$ is the rotation matrix corresponding to the camera orientation $\mathbf{q}$ and $\mathbf{r}$ is the current camera position. This value is then projected into the camera, described by intrinsic and distortion parameters $\mathbf{k}$ and $\mathbf{d}$ (we use a classical radial distortion model of up to three parameters, which is inverted as explained in [19]). Let us call $\mathcal{K} = (\mathbf{k}, \mathbf{d})$ the camera parameters, $\mathcal{C} = (\mathbf{r}, \mathbf{q})$ the camera pose, and $\mathbf{i} = (\mathbf{x}_0, \theta, \psi, \rho)$ the IDP ray. The observation function is

$$\mathbf{u} = \mathbf{h}_1(\mathcal{C}, \mathcal{K}, \mathbf{i}) + \eta = \text{project}(\mathbf{h}_1^{\mathcal{C}}, \mathcal{K}) + \eta \tag{11}$$

where project () takes into account the camera model (we use perspective cameras) and η is the pixel Gaussian noise, with covariance $\mathbf{R}$.

We define the $\mathcal{E}_1$ ellipse as the Gaussian expectation $\mathcal{E}_1(\mathbf{u}) \triangleq \mathcal{N}(\mathbf{u} - \bar{\mathbf{e}}_1; \mathbf{E}_1)$, with $\mathbf{u}$ being the pixel position, and with mean and covariances matrix

$$\bar{\mathbf{e}}_1 = \mathbf{h}_1(\bar{\mathcal{C}}, \mathcal{K}, \bar{\mathbf{i}}) \tag{12}$$

$$\mathbf{E}_1 = [\mathbf{H}_{\mathcal{C}}\, \mathbf{H}_{\mathbf{i}}]\, \mathbf{P}_{\mathcal{C},\mathbf{i}}\, [\mathbf{H}_{\mathcal{C}}\, \mathbf{H}_{\mathbf{i}}]^\top + \mathbf{R}. \tag{13}$$

Here, $\mathbf{H}_{\mathcal{C}}$ and $\mathbf{H}_{\mathbf{i}}$ are the Jacobians of $\mathbf{h}_1$ with respect to the uncertain parameters $\mathcal{C}$ and $\mathbf{i}$, $\bar{\bullet}$ are variable estimates from the SLAM map, and $\mathbf{P}_{\mathcal{C},\mathbf{i}}$ is the joint covariances matrix (all correlations and cross correlations) of $\mathcal{C}$ and $\mathbf{i}$, also from the map. In AS, $\mathcal{E}_1$ is usually gated at 3σ, giving place to an elliptic region in the image where the landmark must project with 98% probability. However, this is not necessarily true in cases of noticeable parallax, as we examine now.

At landmark initialization, its inverse depth ρ is initialized according to (2)–(4). When considering 3σ uncertainty regions, (4) implies that ρ can go negative with a nonnegligible probability, meaning that the coded landmarks might be situated *behind the camera*. This becomes evident when projecting the IDP ray into a second camera presenting some parallax: the projected 3σ $\mathcal{E}_1$ ellipse contains a region with negative disparity (see

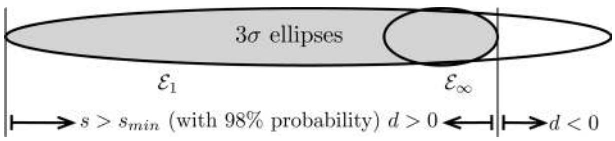

Fig. 5. 3σ search region defined by the $\mathcal{E}_1$ ellipse contains a significant part that corresponds to negative disparities $d < 0$, where the feature should not be searched. The final 3σ search region (gray) is defined by the $\mathcal{E}_1$ and $\mathcal{E}_\infty$ ellipses. The rightmost 3σ border of $\mathcal{E}_\infty$ is where the probability to find the projection of the infinity point has fallen below 2%.

Q1

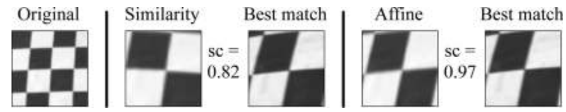

Fig. 6. Similarity and affine warping on a sample patch. From left to right: original patch; similarity warped patch (~180% scale, 10° rotation); best match in a later image affected by distortion and its zero mean normalized cross correlation (ZNCC) score (0.82); affine warped patch; best match and score (0.97). The affine warping contains a significant skew component mainly due to image distortion. The improvement in the ZNCC score is very important.

Fig. 5). It is desirable to limit the search area to values of only positive disparity for two reasons: the correlation-based search (one of the most time-consuming processes) is faster and the possibility of including false matches as outliers is diminished. With nonrectified images and/or camera sets with uncertain extrinsic parameters, determining the null disparity bound is not straightforward. One solution is to use the $\mathcal{E}_\infty$ ellipse, which we introduce in the following paragraph.

2) $\mathcal{E}_\infty$ Ellipse: Expectation of the Infinity Point: The infinity point is easily projected by considering the transformation (10) with $\rho \to 0$

$$\mathbf{h}_\infty^{\mathcal{C}} \approx \mathbf{R}(\mathbf{q})^\top \mathbf{m}(\theta, \psi) \tag{14}$$

where only the camera orientation $\mathbf{q}$ and the ray's direction angles (θ, ψ) are present (the visual compass). Proceeding as before, we obtain the definition of the ellipse $\mathcal{E}_\infty(\mathbf{u}) \triangleq \mathcal{N}(\mathbf{u} - \bar{\mathbf{e}}_\infty; \mathbf{E}_\infty)$ as

$$\bar{\mathbf{e}}_\infty = \mathbf{h}(\bar{\mathbf{q}}, \mathcal{K}, \bar{\theta}, \bar{\psi}) \tag{15}$$

$$\mathbf{E}_\infty = [\mathbf{H_q}\, \mathbf{H}_\theta\, \mathbf{H}_\psi]\, \mathbf{P}_{\{\mathbf{q},\theta,\psi\}}\, [\mathbf{H_q}\, \mathbf{H}_\theta\, \mathbf{H}_\psi]^\top + \mathbf{R} \tag{16}$$

where $\mathbf{P}_{\{\mathbf{q},\theta,\psi\}}$ is the joint covariances matrix of the uncertain parameters. The $\mathcal{E}_\infty$ 3σ region is composed of the previous $\mathcal{E}_1$ region, as indicated in Fig. 5, to define the search area.

C. Selection of the Best Map Updates

Following the AS approach in [23], a predefined number of landmarks with the biggest $\mathcal{E}_1$ ellipse surfaces are selected in each camera as those being the most interesting to be measured. For each camera, we organize all candidates (visible landmarks) in descending order of expectation surfaces, without caring if they are points or rays. We update at each frame a predefined number of them (usually around 10, and no more than 20). Updates are processed sequentially, with all Jacobians being recalculated each time to minimize the effect of linearization errors.

D. Feature Matching: Affine Patch Warping

AS continues by *warping* the stored patch and searching for a correlation peak inside the search area earlier. The objective of warping is to predict the landmark's current appearance, maximizing the chances for a good match. In the absence of distortion, a planar homography $\mathbf{H} \in \mathbb{R}^{3\times3}$, defined in the homogeneous spaces, would be desirable [24]. This type of warping requires the online estimation of the patch normal in the 3-D space, and may become very time-consuming. A good simplification considers this normal fixed at the initial visual axis [23]. Further simplification applies just a similarity transformation $\mathbf{T} = s\mathbf{R} \in \mathbb{R}^{2\times2}$ in the image Euclidean plane [19]. This accounts only for scale changes s and rotations $\mathbf{R}$ obtained from the stored information (landmark position, camera initial, and current poses). However, in the presence of distortion, features lying close to the image borders suffer from additional deformations. We developed a warping approach that easily adds a skew component to the operator $\mathbf{T}$ (thus achieving fully affine warping, but not perspective warping; Fig. 6), based on the Jacobian of the function linking the first observation to the current one. Let us consider the backward observation model $\mathbf{g}(\,)$ for a camera A at initialization time $t = 0$, and the observation model $\mathbf{h}(\,)$ for a different camera B at current time $t \geq 0$

$$\begin{aligned}\mathbf{p} &= \mathbf{g}(\mathcal{C}_A(0), \mathcal{K}_A, \mathbf{u}_A(0), s_A)\\ \mathbf{u}_B(t) &= \mathbf{h}(\mathcal{C}_B(t), \mathcal{K}_B, \mathbf{p}).\end{aligned}$$

Here, $\mathbf{p}$ is the landmark's position, $\mathcal{K}_i = (\mathbf{k}_i, \mathbf{d}_i)$ are the intrinsic and distortion parameters of camera i, $\mathbf{u}_i(t)$ is the measured pixel, and s_A is the landmark's depth with respect to the initial camera. We can compose these functions to obtain the expression linking the initial and the current pixels

$$\mathbf{u}_B(t) = \mathbf{h}\left[\mathcal{C}_B(t), \mathcal{K}_B, \mathbf{g}(\mathcal{C}_A(0), \mathcal{K}_A, \mathbf{u}_A(0), s_A)\right]. \tag{17}$$

When all but the pixel positions are fixed, this represents an invertible mapping $\mathbb{R}^2 \mapsto \mathbb{R}^2$ from the pixels in the first image to the pixels in the current one. The local linearization around the initially measured pixel defines an affine warping expressed by the Jacobian matrix

$$\mathbf{T} = \left.\frac{\partial \mathbf{u}_B}{\partial \mathbf{u}_A}\right|_{(\mathcal{C}_A(0), \mathcal{C}_B(t), \mathcal{K}_A, \mathcal{K}_B, \mathbf{u}_A(0), s_A)}. \tag{18}$$

By defining $\tilde{\mathbf{u}}_i$ as the coordinates of the patch in camera i, with the central pixel $\mathbf{u}_i$ as the origin, we have $\tilde{\mathbf{u}}_B(t) = \mathbf{T}\,\tilde{\mathbf{u}}_A(0)$. Based on this mapping, we use linear interpolation of the pixels' luminosity to construct the warped patch.

V. Experiment 1: Stereo SLAM With Self-Calibration

The "White-board" indoor experiment aims at demonstrating stereovision SLAM with self-calibration. A robot with a stereo head looking forward is run for about 10 m in straight line inside the robotics laboratory at the LAAS (see Fig. 7). Over 500 image pairs are taken at approximately 5-Hz frequency. The robot

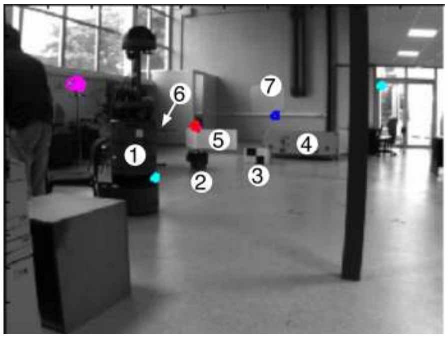

Fig. 7. Laboratoire d'Analyse et d'Architecture des System (LAAS) robotics laboratory. The robot will approach the scene in a straightforward trajectory. We notice in the scene the presence of a robot ①, a bin ②, a box ③, a trunk ④, a fence ⑤, a table ⑥ (hidden by the robot in this image), and the white board ⑦ at the end wall.

TABLE I
STEREO RIG PARAMETERS IN THE "WHITE-BOARD" EXPERIMENT

Scope	Parameters = Values
Dimensions	Baseline = 33 cm
Orientation - Euler	$\{\phi, \theta, \psi\} = \{0°, 5°, 0°\}$
Cameras	{resolution,FOV} = $\{512 \times 384\,\text{pix}, 55°\}$
Right camera uncertainties	$\{\sigma_\phi, \sigma_\theta, \sigma_\psi\} = \{1°, 1°, 1°\}$

moves towards the objects to be mapped at 0.15 m/s. The stereo rig consists of two intrinsically calibrated cameras arranged as indicated in Table I. The orientations of both cameras are specified with respect to the robot frame. The left camera is taken as reference, thus deterministically specified, and the orientation of the right one is initialized with an uncertainty of 1° standard deviation. We use the odometry model (Section III-A) with $k_L = 0.1$ m/$\sqrt{\text{m}}$ and $k_A = 0.05$ rad/$\sqrt{\text{m}}$.

We show details and results on the self-calibration procedure and the metric accuracy of the resulting map. The mapping process can be appreciated in the movie `whiteboard.mov` in the multimedia section.

A. Self-Calibration

We plot in Fig. 8 left the evolution of the three self-calibrated angles. We have also used the shape of the $\mathcal{E}_\infty$ ellipses to provide additional qualitative evidence of the calibration process (Fig. 9 and movie `whiteboard − einf.mov`). We observe the following behavior.

1) Pitch θ: The pitch angle (cameras tilt, 5° nominal value) is observable from the first matched landmark. It rapidly converges to an angle of 4.77° and remains very stable during the whole experiment.

2) Roll ϕ: Roll angle is observable after at least two landmarks are observed from the right camera. Once this condition holds, convergence occurs relatively fast.

3) Yaw ψ: Yaw angle is very weakly observable because it is coupled with the landmarks depths: both yaw angle and landmark depth variations produce a similar uncertainty growth in the right image. For this reason, yaw converges slowly, only showing reasonable convergence after some 50 frames.

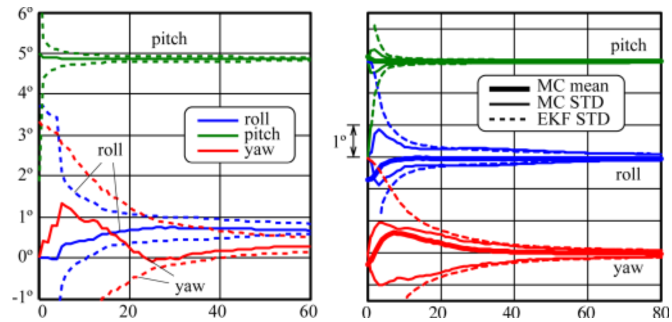

Fig. 8. Extrinsic self-calibration. (Left) The three Euler angles of the right camera orientation with respect to the robot during the first 60 frames. The 3σ bounds are plotted in dotted line showing consistent estimation. (Right) Error analysis after 100 MC runs using 200 frames per run (only the first 80 frames are shown). The thick solid lines represent the means over the 100 runs. The 3σ bounds for each angle are plotted using thin solid lines. The dotted lines represent the averaged 3σ bounds estimated by the EKF, showing consistent calibration.

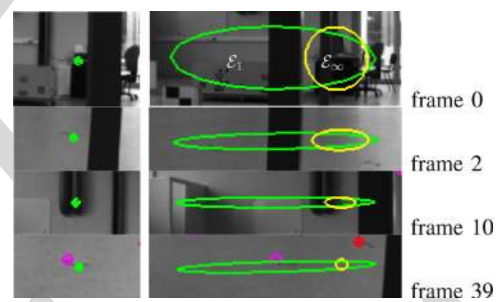

Fig. 9. Evolution of the $\mathcal{E}_1$ and $\mathcal{E}_\infty$ ellipses during calibration. On the left column, newly detected pixels in the left image. On the right, expectations in the right image (green) $\mathcal{E}_1$ and (yellow) $\mathcal{E}_\infty$ of the newly initialized IDP rays (i.e., still with the full initial uncertainty in ρ). At frame 0, initial uncertainties of 1° result in a big, round $\mathcal{E}_\infty$ ellipse. After the first updated landmark from the left camera (frame 2), the uncertainty in pitch decreases and $\mathcal{E}_\infty$ becomes flat. Successive updates further refine the calibrated angles. The yaw angle takes longer to converge, but the tiny $\mathcal{E}_\infty$ in frame 39 shows that the calibration is already finished. The portion of the green ellipse on the right side of the yellow one corresponds to negative disparities and is not searched for matches. This portion is larger as parallax increases.

TABLE II
MC ANALYSIS OF THE SELF-CALIBRATION

Angle	MC mean	STD	EKF STD	Offline	STD
roll	0.69°	0.028°	0.018°	0.61°	0.013°
pitch	4.77°	0.003°	0.005°	4.74°	0.099°
yaw	0.33°	0.021°	0.016°	0.51°	0.109°

In Fig. 8 right, we plot results of a Monte Carlo (MC) analysis, run over the data of this experiment, for the mean and standard deviation of the Euler angles of the right camera. Because all MC runs are extracted from the same sequence, we tried to maximize their independence by using a different *random seed* in the algorithm (acting in the random selection of the initialization region, Section IV-A), and by starting each run at a different frame. The figure shows that the dynamic estimation is consistent (the EKF estimated sigmas are larger than the MC ones). After 200 frames, we compare these values with those of the offline calibration [25]. Table II summarizes these results, showing MC [(means and standard deviations (STD)] and Kalman Filter (EKF, showing the estimated STD). All

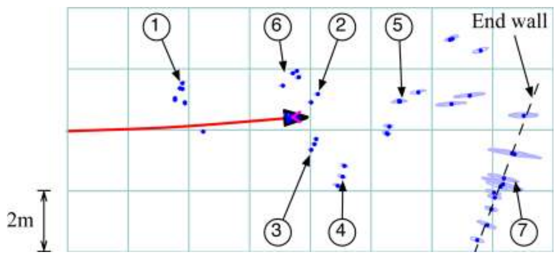

Fig. 10. Map produced during the "white board" experiment. We marked the mapped robot ①, the bin ②, the box ③, the trunk ④, the fence ⑤, the table ⑥, and the white board ⑦ at the end wall.

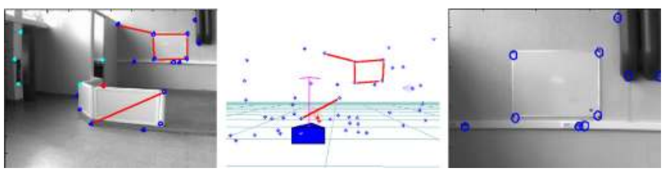

Fig. 11. Metric mapping. The magnitudes of some segments in the real laboratory are compared to those in the map (*red lines*). Ground truth corresponds to metric measurements of the distances between landmarks that are identified by zooming in the last image of the experiment (right) and translated to the real world. Thirteen points on the end wall are tested for coplanarity.

TABLE III
WHITE BOARD: MAP TO GROUND TRUTH TOMPARISON

segment	board	board	board	board	wall	fence
real (cm)	116	86	117	88	136	124
mapped	116.6	87.2	115.8	87.0	135.1	125.5
STD	0.91	0.81	1.21	0.52	1.06	1.32

self-calibrated values lie within the 3σ bounds defined by the offline mean and STD values.

B. Metric Accuracy

We show in Fig. 10 a top view of the map generated during this experiment. To contrast this map against reality, two tests are performed: planarity and metric scale (see Fig. 11): 1) the four corners of the white board are taken together with nine other points at the end wall to test coplanarity: the 13 mapped points are found to be coplanar within 4.9 cm STD; 2) the lengths of the real and mapped segments marked in Fig. 11 are summarized in Table III. The white board has a physical size of 120 cm $\times$ 90 cm, but we take real measurements from the approximated corners where the features are detected. We observe errors in the order of 1 cm for landmarks that are still about 4 m away from the robot.

VI. EXPERIMENT 2: COOPERATIVE MONOCULAR SLAM

This experiment shows independent cameras collaborating to build a 3-D map using exclusively bearings-only observations. Two independent cameras are placed on top of two bicycles looking forward, moving on different trajectories in the parking of the LAAS (see Fig. 12). Over 1000 images are taken by each camera at 15-Hz frequency, 512×384 pixel resolution, 100° field of view (FOV), and are processed offline. The cameras travel approximately 28 m observing landmarks beyond 60 m. As in the previous experiment, the left camera is the master. The two trajectories start parallel to each other, separated 75 cm perpendicularly to the motion direction. The reference frame is defined by the master camera initial position and orientation, which are initialized with null uncertainty. The scale factor is determined by the initial baseline of 75 cm, meaning that the position of the slave camera in the lateral Y-axis is also initialized with null uncertainty. The orientations of the slave camera start with an uncertainty of 2° STD, and its position in the frontal Y- and vertical Z-axes with $75\,\text{cm}\cdot\sin(2^\circ) = 2.6\,\text{cm}$ STD. With these uncertainties, the experiment's initial configuration can be set up manually by just observing the images and centering the projections of some distant object. We use two independent constant-velocity models with $k_v = 0.3$ m/s $\cdot \sqrt{\text{s}}$ and $k_w = 0.3$ rad/s $\cdot \sqrt{\text{s}}$. The measurement noise is 1 pixel.

Fig. 12. Snapshots of master and slave sequences in cooperative SLAM. Faraway landmarks (e.g., black arrowed), still initialized as rays (red), are the ones fixing the orientation. Nearby landmarks, usually as Euclidean points (blue), maintain the metric. A virtual model of the master camera is visible from the slave camera (white arrowed). See `cooperativeSLAM.mov`.

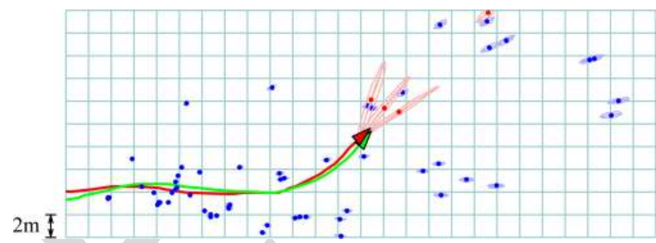

Fig. 13. Top view of the map produced by cooperative SLAM of two independent cameras, and their crossing trajectories. The grid spacing is 2 m.

Landmarks at infinity, illumination changes and few salient features are some characteristics of this outdoors scene. It presents relatively few stable landmarks, something that makes the correct operation of the SLAM system difficult. In the case of having crossing trajectories, the problem of one camera occluding the other could appear and severely affect the image processing. To avoid this, we decided to take both image sequences shifted in time, i.e., one after the other, and make them overlap for processing. The mapping process is presented in the movie `cooperativeSLAM.mov` in the multimedia section. Fig. 13 shows the top view of the map and the camera trajectories generated during this experiment.

A proper metrical evaluation of this experiment is difficult; having a variable baseline does not allow us to compare the results, because there is no knowledge of the ground truth. In order to evaluate this approach, we consider the setup in experiment

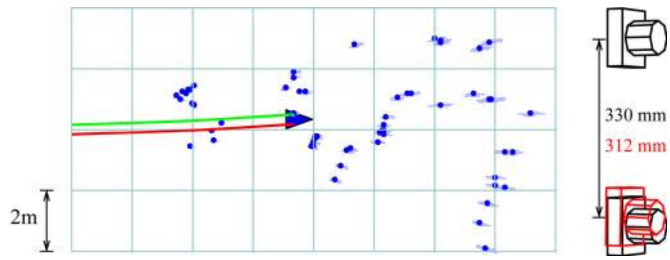

Fig. 14. Final map in the "white board" setup using the cooperative monocular SLAM algorithm. The cameras are modeled as being entirely independent using the same data and initial configuration as in Experiment 1. The stereo rig on the right shows (red) the final estimated relative position compared with (black) ground truth.

1 and apply the same algorithm. The new experiment consists of recovering the full extrinsic calibration, which is fixed in reality, considering both cameras as independent. Again, we use a constant-velocity model for each camera. The initial setup including uncertainties is as in experiment 1.

Fig. 14 shows the obtained map. We see that it compares very well to the map obtained in experiment 1 (see Fig. 10), where the motions of the two cameras were constrained by the stereo rig and a common motion was predicted using odometry. Fig. 14 bottom shows a detail of the cameras in their final relative position. We measure an error along the baseline of less than 2 cm. The orientation errors are less than 0.7°.

VII. Conclusion

We showed in this paper that fusing the visual information with monocular methods while performing multicamera SLAM provides several advantages: the ability to consider points at infinity, desynchronization of the different cameras, the use of any number of cameras of different types, sensor self-calibration, and the possibility to conceive decentralized schemes that will make realistic multirobot monocular SLAM possible. Except for decentralization, these advantages have been explored with the inverse depth monocular SLAM algorithm, and applied to two different problems: stereovision SLAM with an extrinsically decalibrated stereo rig and cooperative SLAM of two independently moving cameras.

Both demonstrations employed a *master–slave* approach, which made solving some of the issues of map and image management easier, and we are now improving on this by implementing a fully symmetrical approach. This approach should easily permit the extension of the presented applications to cases with more than two cameras. In parallel to these activities, we started new work on landmark parametrization to improve EKF linearity in cases of increasing parallax. Also, as parallax increases, landmarks appearances may change too much as to guarantee a stable operation with the matching methods presented here. We believe that wide baseline feature matching will be the bottleneck of visual SLAM for some time to come. As for decentralization, we note that it demands a full reformulation of the fusion engines we use in this paper (one central EKF), for example, via channel filters, and is currently a subject of intense research at LAAS and other laboratories.

References

[1] J. Solà, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, Aug. 2–6, 2005, pp. 2499–2504.

[2] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "Structure from motion causally integrated over time," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 523–535, Apr. 2002.

[3] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. Int. Conf. Comput. Vis.*, Nice, France, Oct. 2003, vol. 2, pp. 1403–1410.

[4] J. Civera, A. J. Davison, and J. M. M. Montiel, "Dimensionless monocular SLAM," in *Proc. Iberian Conf. Pattern Recognit. Image Anal.*, Jun, 2007, pp. 412–419.

[5] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, Oct. 2008.

[6] E. Eade and T. Drummond, "Scalable monocular SLAM," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun.17–22, 2006, vol. 1, pp. 469–476.

[7] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment—A modern synthesis," in *Vision Algorithms: Theory and Practice,* ser. LNCS, W. Triggs, A. Zisserman, and R. Szeliski, Eds. New York: Springer-Verlag, 2000, pp. 298–375.

[8] K. Konolige, "SLAM via variable reduction from constraints maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 18–22, 2005, pp. 667–672.

[9] J. Folkesson, P. Jensfelt, and H. I. Christensen, "Vision SLAM in the measurement subspace," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 18–22, 2005, pp. 30–35.

[10] J. Diebel, K. Reutersward, S. Thrun, and R. G. J. Davis, "Simultaneous localization and mapping with active stereo vision," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sendai, Japan, Oct. 2004, vol. 4, pp. 3436–3443.

[11] J. Solà, A. Monin, and M. Devy, "BiCamSLAM: Two times mono is more than stereo," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 4795–4800.

[12] L. M. Paz, P. Piniés, J. Tardós, and J. Neira, "Large scale 6 DOF SLAM with stereo-in-hand," *IEEE Trans. Robot.*, vol. 24, no. 5, Oct. 2008.

[13] A. Mallet, S. Lacroix, and L. Gallo, "Position estimation in outdoor environments using pixel tracking and stereovision," in *Proc. Int. Conf. Robot. Autom.*, San Francisco, CA, 2000, vol. 4, pp. 3519–3524.

[14] K. Konolige, M. Agrawal, and J. Solà, "Large-scale visual odometry for rough terrain," presented at the Int. Symp. Res. Robot., Hiroshima, Japan, Nov. 2007.

[15] T. D. Barfoot, "Online visual motion estimation using FastSLAM with SIFT features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2–6, 2005, pp. 579–585.

[16] A. I. Comport, E. Malis, and P. Rives, "Accurate quadrifocal tracking for robust 3D visual odometry," in *Proc. Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 10–14, 2007, pp. 40–45.

[17] E. M. Foxlin, "Generalized architecture for simultaneous localization, auto-calibration, and map-building," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Lausanne, Switzerland, 2002, vol. 1, pp. 527–533.

[18] E. Nettleton, H. Durrant-Whyte, and S. Sukkarieh, "A robust architecture for decentralised data fusion," presented at the Int. Conf. Adv. Robot., Coimbra, Portugal, 2003.

[19] J. Solà, "Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach" Ph.D. dissertation, Inst. Nat. Polytech. de Toulouse, Toulouse, France, 2007.

[20] J. Civera, A. Davison, and J. Montiel, "Inverse depth to depth conversion for monocular SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 10–14, 2007, pp. 2778–2783.

[21] A. J. Davison, "Active search for real-time vision," in *Proc. Int. Conf. Comput. Vis.*, 2005, vol. 1, pp. 66–73.

[22] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vis. Conf.*, Manchester, U.K., 1988, pp. 189–192.

[23] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[24] N. Molton, A. J. Davison, and I. Reid, "Locally planar patch features for real-time structure from motion," presented at the Brit. Mach. Vis. Conf., Kingston, U.K., 2004.

[25] K. Strobl, W. Sepp, S. Fuchs, C. Paredes, and K. Arbter. (2006). Camera calibration toolbox for Matlab. Inst. Robot. Mechatronics, Wessling, Germany, Tech. Rep. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

Joan Solà was born in Barcelona, Spain, in 1969. He received the B.Sc. degree in telecommunications and electronic engineering from the Universitat Politècnica de Catalunya, Barcelona, in 1995, the M.Sc. degree in control systems from the École Doctorale Systèmes, Toulouse, France, in 2003, and the Ph.D. degree in control systems from the Institut National Polytechnique de Toulouse in 2007, where he was hosted by the Laboratoire d'Analyse et d'Architecture des System (LAAS), Centre National de la Recherche Scientifique (CNRS).

He was a Postdoctoral Fellow at SRI International, Menlo Park, CA. He is currently at LAAS-CNRS, where he is engaged in research on visual localization and mapping. His current research interests include estimation and data fusion applied to off-road navigation, mainly using vision.

André Monin was born in Le Creusot, France, in 1958. He received the Graduate degree from the Ecole Nationale Supérieure dIngénieurs Electriciens de Grenoble, Grenoble, France, in 1980, the Ph.D. degree in nonlinear systems representation from the University Paul Sabatier, Toulouse, France, in 1987, and the Habilitation pour Diriger des Recherches degree from the University Paul Sabatier in 2002.

From 1981 to 1983, he was a Teaching Assistant with the Ecole Normale Supérieure de Marrakech, Marrakech, Morocco. Since 1985, he has been with the Laboratoire Automatique et d'Analyse des Systèmes, Centre Nationale de la Recherche Scientifique (LAAS-CNRS), Toulouse, as the "Chargé de Recherche." His current research interests include the areas of nonlinear filtering, systems realization, and identification.

Michel Devy received the degree in computer science engineering from the Ecole Nationale Supérieure dIngénieurs Electriciens de Grenoble, Grenoble, France, in 1976 and the Ph.D. degree from the Laboratoire Automatique et d'Analyse des Systèmes, Centre Nationale de la Recherche Scientifique (LAAS-CNRS), Toulouse, France, in 1980.

Since 1980, he has been with the Department of Robotics and Artificial Intelligence, LAAS-CNRS, where he is the Research Director and the Head of the Research Group Robotics, Action, and Perception. His current research interests include computer vision for automation and robotics applications. He has also been involved in numerous national and international projects concerning, about field and service robots, 3-D vision for intelligent vehicles, 3-D metrology, and others. He has authored or coauthored about 150 scientific communications.

Teresa Vidal-Calleja received the B.Sc. degree in mechanical engineering from the Universidad Nacional Autónoma de México, México City, México, in 2000, the M.Sc. degree in mechatronics from CINVESTAV-IPN, México City, in 2002, and the Ph.D. degree in robotics, automatic control, and computer vision from the Universitat Politèctica de Catalunya, Barcelona, Spain, in 2007.

She was a Visiting Research Student with the University of Oxford's Robotics Research Group, the Australian Centre for Field Robotics, and the University of Sydney. She is currently a Postdoctoral Fellow with the Robotics and Artificial Intelligence Group, Laboratoire Automatique et d'Analyse des Systèmes, Centre Nationale de la Recherche Scientifique (LAAS-CNRS), Toulouse, France. Her current research interests include autonomous vehicles, perception, control, and cooperation.

QUERIES

Q1: Author: Please reframe the references to colors in the caption of Figs. 5, 8, 9, 11, 12, and 14, if the artwork is not being produced in color

Q2. Author: Please check if the details of the academic degrees received by A. Monin are OK as edited.

Q3. Author: Please provide the title of first degree. Also, provide the subject (physics, mathematics, electrical engineering, etc.) in which M. Dey received the Ph. D. degree.

Fusing Monocular Information in Multicamera SLAM

Joan Solà, André Monin, Michel Devy, and Teresa Vidal-Calleja

***Abstract*—This paper explores the possibilities of using monocular simultaneous localization and mapping (SLAM) algorithms in systems with more than one camera. The idea is to combine in a single system the advantages of both monocular vision (bearings-only, infinite range observations but no 3-D instantaneous information) and stereovision (3-D information up to a limited range). Such a system should be able to instantaneously map nearby objects while still considering the bearing information provided by the observation of remote ones. We do this by considering each camera as an independent sensor rather than the entire set as a monolithic supersensor. The visual data are treated by monocular methods and fused by the SLAM filter. Several advantages naturally arise as interesting possibilities, such as the desynchronization of the firing of the sensors, the use of several unequal cameras, self-calibration, and cooperative SLAM with several independently moving cameras. We validate the approach with two different applications: a stereovision SLAM system with automatic self-calibration of the rig's main extrinsic parameters and a cooperative SLAM system with two independent free-moving cameras in an outdoor setting.**

***Index Terms*—Calibration, image sequence analysis, Kalman filtering, machine vision, robot vision systems, stereovision.**

I. Introduction

THE SIMULTANEOUS localization and mapping (SLAM) problem, as formulated by the robotics community, is that of creating a *map* of the perceived environment while *localizing* oneself in it. The two tasks are coupled in such a way so as to benefit each other; a good localization is crucial to create good maps, and a good map is necessary for localization. For this reason, the two tasks must be performed *simultaneously*, and hence, the full acronym SLAM. In recent years, the maturity of both online SLAM algorithms, together with fast and reliable image processing tools from the computer vision literature, has crystallized into a considerable quantity of real-time demonstrations of visual SLAM.

In this paper, we insist on the quality of the achieved localization, which will impact in turn the map quality. The key to good localization is to ensure the correct processing of the geometrical information gathered by the cameras. In this long introduction, we present an overview of visual SLAM and related techniques to show that visual SLAM systems have historically discarded precious sensory information. We present a novel approach that uses the SLAM filter as a classical fusion engine that incorporates the full monocular information coming from multiple cameras.

Manuscript received June 15, 2007; revised May 8, 2008. First published xxx; current version published xxx. This paper was recommended for publication by Associate Editor J. Tardos (with approval of the Guest Editors) and Editor L. Parker upon evaluation of the reviewers' comments.

The authors are with the Laboratoire d'Analyse et d'Architecture des System, Centre National de la Recherche Scientifique (LAAS-CNRS), University of Toulouse, Toulouse 31077, France (e-mail: jsola@laas.fr; monin@laas.fr; michel@laas.fr; tvidal@laas.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2008.2004640

A. Monocular SLAM

Possibly, the best example of the aforementioned technological crystallization is monocular SLAM, a particular case of bearings-only (BO) SLAM (where the sensor does not provide any range or depth). It is well known that the reduction in system observability due to BO measurements has two main drawbacks: the loss of the scale factor and the delay in obtaining good 3-D estimates. Previous works either added some metric measurement to observe the scale factor, such as odometry [1] or the size of known perceived objects [2], [3], or have considered it irrelevant [4]. The delay in getting good 3-D estimates comes from the fact that such estimates require several BO observations from different viewpoints. This makes landmark initialization in BO-SLAM difficult, to the point that satisfactory methods able to exploit all the geometrical information provided by the cameras have only recently become available. We have witnessed an evolution of the algorithms as follows. First, *delayed landmark initialization* methods attempted to obtain a full 3-D estimate before initialization via several observations from different viewpoints. Davison [3] showed real-time feasibility of monocular SLAM with affordable hardware, using the original extended Kalman filter (EKF) SLAM algorithm for all but the unmeasured landmark's depth, and a separate particle filter to estimate this depth. Initialization was *deferred* to the moment when the depth estimate was good enough. The consequence of a delayed scheme is that we can only initialize landmarks with enough parallax, i.e., those that are close to the camera and situated perpendicularly to its trajectory, and therefore, the need to operate in room-size scenarios with lateral motions. Second, Solà *et al.* [1] showed that *undelayed landmark initialization* (mapping the landmarks from their first, partial observation) was needed when considering low parallax landmarks, i.e., those that are remote and/or situated close to the motion axis. This permits mapping larger scenes while performing frontal trajectories. Third, Civera *et al.* [5] have recently achieved the mapping of landmarks up to infinity, due to an undelayed initialization via an *inverse depth parameterization* (IDP). IDP has also been developed by Eade *et al.* [6] in a FastSLAM2.0 context. Today, the monocular SLAM systems exploit the geometrical information in its entirety: from the first observation, independently of the sensor's trajectory, and up to the infinity range.

B. Structure From Motion (SFM)

Monocular SLAM compares to a similar problem solved by the vision community: the structure from motion problem (SFM). In SFM, the goal is to determine, from a collection of images and up to an unrecoverable scale factor, the 3-D structure of the perceived scene and all 6-D camera poses from where the images were captured. When compared to SLAM, the structure plays the role of the map, while the set of camera poses defines all the successive observer's localizations.

Roboticists often claim that the main difference between SFM and SLAM is that the former is solved offline via the iterative nonlinear optimization method known as bundle adjustment (BA) [7], while the latter must be incrementally solved online, thus making use of stochastic estimators or *filters* that naturally provide incremental operation. This has been true for some years (today, SLAM is also solved online with iterative optimization [8]), but does not tell the whole story. The differences between SFM and SLAM are not only in the methods but also in the objectives, meaning that similar aspects of similar problems are given different priorities.

In particular, SFM exploits the visual information in its entirety without the difficulties encountered in monocular SLAM. Let us try to understand this curious fact. SFM puts the structure as a final objective, i.e., as a result of the whole process, and the emphasis is placed on minimizing the errors in the *measurement space*, thus using all the measured information. On the other hand, the SLAM map has a central role, with some of the operations (and particularly landmark initialization) being performed in map space, which is the system's *state space*. The fact that this state space is not statically observable, because it is of higher dimension than the observation space, leads to the difficulties exposed before. As an informal attempt to fill this gap, we could say that modern undelayed methods for monocular SLAM, with partial landmark initialization and partial updates, are almost equivalent to an operation in the measurement space: the information is initialized in the map space *partially*, i.e., exactly as it comes from the measurement space. A similar point of view over this concept can be found in [9].

C. Stereovision SLAM

Stereovision SLAM has also received considerable attention. The ability of a stereo assembly to directly and immediately provide 3-D landmark estimates allows us to use the best available SLAM algorithms and rapidly obtain good results with little effort in the conceptual parts. Such SLAM systems consider the stereo assembly as being a single monolithic sensor, capable of gathering 3-D geometrical information from the robot's surroundings, e.g. [10]. This fact, which appears perfectly reasonable, is the main paradigm that this paper questions. By considering two linked cameras as a single 3-D sensor, SLAM is unable to face the following two issues.

1) Limited 3-D Estimability Range: While cameras are capable of sensing visible objects that are potentially at infinity, a stereo rig provides only reasonably good 3-D estimates up to a limited range, typically from 3 m to a few tens of meters depending on the baseline. Because classical, nonmonocular SLAM algorithms expect full 3-D estimates for landmark initialization (i.e., they are reasoned in the map space), information belonging to only this limited region can be used for SLAM. This is really a pity; it is like if, having our two eyes, we were obliged to neglect everything outside a certain range from us, what we could call "*walking inside dense fog.*" Without remote landmarks, it is easy to lose spacial references, to become disoriented, and finally, find ourselves lost. Therefore, stereovision, as it is classically conceived, is a bad starting point for visual SLAM.

2) Mechanical Fragility: If we aim at extending the 3-D estimability range beyond these few tens of meters, we need to increase the stereo baseline while keeping or improving the overall sensor precision. This is obviously a contradiction: larger assemblies are less precise when using the same mechanical solutions. In order to maintain accuracy with a larger assembly, we must use more complex structures that will be either heavier or more expensive, if not both. The result for moderately large baselines (>1 m) is a sensor that is very easily decalibrated, and therefore, almost useless. Large rigs, however, are very interesting in outdoor applications because they allow farther objects to be positioned, thus making them contribute to the observability of the overall scale factor. This is especially true in aerial and underwater settings where, without nearby objects to observe, a small stereo rig provides no significant gain with respect to a single camera. Self-calibration can compensate for the inherent lack of stability of large camera rigs. It also allows multicamera platforms to start operation without undergoing a previous calibration phase, making on-field system deployment and maintenance easier.

To our knowledge, the only SLAM work that goes beyond the current stereoparadigm (apart from our conference paper [11]) is the one by Paz *et al.* [12], which uses a small-baseline, fully calibrated stereo rig. Matched features presenting significant disparity are initialized as classical Euclidean landmarks, while those presenting low disparities are treated with the inverse depth algorithm.

D. Visual Odometry (VO)

One could say that, in terms of methodology, visual odometry (VO) is to stereovision SLAM what SFM is to monocular SLAM. VO is conceived to obtain the robot's ego motion from a sequence of stereo images [13]. Visual features are matched across two or more pairs of stereo images taken during the robot motion. An iterative minimization algorithm, usually based on BA, is run to recover the stereo rig motion, which is then transformed into robot motion. For this, the algorithm needs to recover the structure of the 3-D points that correspond to the matched features. This structure is not exploited for other tasks and can be usually discarded. Remarkably, when the structure is coded in the measurement space (u, v, d), a disparity $d \to 0$ allows points at infinity to be properly handled [14]. This is also accomplished by using homogeneous coordinates [7]. VO must work in real time because robot localization is needed online.

Advanced VO solutions achieve very low drift levels after long distances by making use of: 1) hardware-based image processing with real-time construction and querying of large feature databases [15]; 2) dense image information matching via planar homographies and the use of the quadrifocal tensor [16]; or 3) bundle adjusting the set of N recent key frames together with additional fusion with an inertial measurement unit (IMU) [14].

E. Sensor Fusion in SLAM

The fact of SLAM being solved by filters allows us to envision SLAM systems as sensor fusion engines. Let us highlight some of the assets of filtering in sensor fusion.

1) *Multisensor operation:* Any number of differing sensors can be operated together in a consistent framework.
2) *Sensors self-calibration:* Unknown biases, gains, and other sensor's parameters can be estimated provided that they are observable [17].
3) *Desynchronized operation:* The data rates of all these sensors do not need to be synchronized.
4) *Decentralized operation:* Advanced filter formulations such as those using channel filters [18] achieve a decentralized operation that should permit live connection and disconnection of sensors without the need for filter reprogramming or reparameterization.

This paper explores the first three points for the case of multiple cameras.

SLAM systems naturally fuse information from both proprioceptive (odometry, GPS, and IMU) and exteroceptive (range scanners, sonar, and vision) sensors into the map. But our interest here is in fusing several exteroceptive sensors. We can distinguish two cases.

1) *Sensors of different kind:* When using differing sensors (e.g., laser plus vision), the main problem is in finding a map representation well adapted to the different kinds of sensory data (i.e., the data association problem).
2) *Sensors of the same kind:* The perceived information is of the same nature. This makes appearance-based matching possible, and therefore, makes map building easier. Nevertheless, most of such SLAM systems do not take advantage of fusion. Instead, the extrinsic parameters linking the sensors are calibrated offline, and the set of sensors is treated as a single supersensor. This is the case for two 180° range scanners simulating a 360° one, and for the previously mentioned stereo rig simulating a 3-D sensor. A sensor-fusion approach in these cases should naturally bring the aforementioned advantages to the SLAM system.

F. Multicamera SLAM and the Aim of This Paper

The key idea of this paper is very simple: by employing the SLAM filter as a fusion engine, we will be able to use any number of cameras in any configuration. And, by treating them as BO sensors with the modern undelayed initialization methods, we will extract the entire geometrical information provided by the images. The filter—not the sensor—will be responsible for making the 3-D properties of the perceived world arise.

Applications may vary from the simplest stereo system, through robots with several differing cameras (e.g., a panoramic one for localization and a perspective one looking forward for reactive navigation), to multirobot cooperative SLAM where BO observations from different robots are used to determine the 3-D locations of very distant landmarks. Although there certainly exist issues concerning multicamera management, the main ideas we want to convey may be demonstrated with systems of just two cameras. In this paper, we will illustrate two cases: first, the case of a robot equipped with a stereo rig, with its cameras being treated as two individual monocular sensors and second, two cameras moving independently and mapping together an outdoors scene.

This paper draws on previous work published in the conference paper [11] and the author's Ph.D. thesis [19]. These two works use the federated information sharing algorithm (FIS) in [1] to initialize the landmarks, which has been surpassed by the inverse depth methods (IDP) [5]. The present paper takes and extends all this research by developing a better founded justification (providing a wider scope to the proposed concepts), by improving on the implementation with the incorporation of IDP in the algorithms, and by extending the experimental validation to a cooperative monocular SLAM setup.

This paper is organized as follows. Section II presents the main ideas that will be exploited later and revises some background material for monocular SLAM. Section III explains how to set up multicamera SLAM, an application for stereo benches with self-calibration, and an application for two collaborative cameras. Section IV presents the perception and map management techniques used. Sections V and VI show the experimental results, and finally, Section VII gives conclusions and future directions.

II. 3-D Estimability in Visual SLAM

In this section, we present the ideas that support our approach to visual SLAM. We make use of the concept of estimability, which will help understand the abilities of vision for observing 3-D structure in the presence of uncertainty. We clarify the key properties of undelayed initialization in monocular SLAM, and remark its importance in multicamera SLAM. We also remind the key aspects of IDP-SLAM.

A. Geometrical Approach to 3-D Estimability

We are interested in finding the shape and dimensions of the 3-D-estimable region defined by two monocular views.

For this, we start with a couple of ideas to help understanding the concept of estimability used. When a new feature is detected in an image, the backprojection of its noisy-measured position defines a conic-shaped *pdf* for the landmark position, called *ray*, which extends to infinity (see Fig. 1). Let us consider two features extracted and matched from a pair of images, corresponding to the same landmark: their backprojections are two conic rays A and B that extend to infinity. The angular

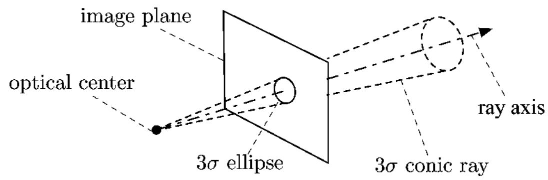

Fig. 1. Conic ray backprojects the elliptic representation of the Gaussian 2-D measure. It extends to infinity.

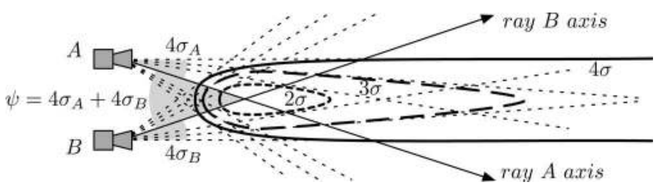

Fig. 2. Different regions of intersection for (solid) 4σ, (dashed) 3σ, and (dotted) 2σ ray widths when the outer 4σ bounds are, parallel. (Shaded) The parallax or angle between rays axes A and B is $\psi = 4\,\sigma_A + 4\,\sigma_B$.

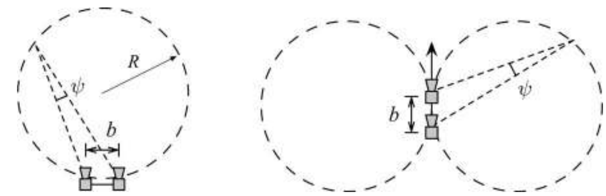

Fig. 3. Simplified depth estimability regions in a (left) stereo rig and (right) a camera traveling forward. The angle ψ is the one that assures estimability via triangulation from different viewpoints. The maximum range is $2R = b/\sin(\psi/2)$.

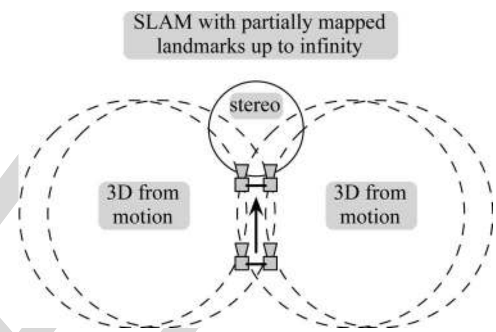

Fig. 4. Simplified depth estimability for a stereo rig moving forward. On both sides, estimability depends on the baseline gained by motion. In front, by stereo. Out of these bounds and up to infinity, landmarks are mapped partially. SLAM keeps incorporating the visual information due to the undelayed monocular methods, i.e., IDP in our case.

widths of these rays can be defined as a multiple of the standard deviations σ_A and σ_B of the angular errors (a composition of the cameras extrinsic and intrinsic parameters errors, and of the image processing algorithms accuracy). Informally speaking, we may say that the landmark's depth is fully estimated if the region of intersection of these rays is both *closed* and *sufficiently small*. If we consider, for example, the case where the two external 4σ bounds of the rays are parallel (see Fig. 2), then we can assure that the 3σ intersection region (which covers 98% probability) is *closed* and that the 2σ one (covering 74%) is *closed and small*. The ratio between the depth's standard deviation and its mean (a measure of linearity in monocular EKF-SLAM [1], [3]) is then better than 0.25. The *parallax* angle ψ between the two rays axes is therefore $\psi = 4(\sigma_A + \sigma_B) = \text{constant}$. This is the minimum parallax for full estimability.

In 2-D, we can plot the locus of constant estimability. In the case, where σ_A and σ_B can be considered constant, ψ is constant too, and from the inscribed angle theorem, the locus is then circular (Fig. 3, see also [19]). Landmarks inside this circle are considered *fully estimable*—and *partially* outside. In 3-D, the *fully 3-D estimable* region is obtained by revolution of this circle around the axis joining both cameras, producing a torus-shaped region with a degenerated central hole. This shape admits the following interpretations.

1) In a stereo configuration or for a lateral motion of a moving camera (see Fig. 3, left), the estimable region is located in front of the sensor. Beyond the region's border stereo provides no profit: if we want to consider distant landmarks, we have to use undelayed monocular techniques.
2) Depth recovery is impossible in the motion axis of a single camera moving forward (Fig. 3, right). Close to this axis, estimability is possible only if the region's radius becomes very large. This implies the necessity of very large displacements of the camera during the initialization process. Again, this can be accomplished only with undelayed initializations.
3) By combining both monocular and stereovision, we get an instant estimability of close frontal objects while still utilizing the information of distant ones (see Fig. 4). Landmarks lying outside the estimability regions are not 3-D-estimable but, when initialized using undelayed monocular methods, they will contribute to constrain the camera orientation. Ideally, long-term observations of stable distant landmarks would completely cancel orientation drift (visual compass).

B. Monocular IDP-SLAM

The core algorithm of this paper is an EKF-SLAM with an IDP of landmarks during the initialization phase, as described in [5]. In IDP-SLAM, partially observed landmarks are coded as a 6-D-vector,

$$\mathbf{i} = [\mathbf{x}_0, \theta, \psi, \rho] \tag{1}$$

where $\mathbf{x}_0$ is the 3-D position of the camera at initialization time, (θ, ψ) are the elevation and azimuth angles in global frame defining the direction of the landmark's ray, and ρ is the inverse of the Euclidean distance from $\mathbf{x}_0$ to the landmark's position (notice that ρ is usually known as *inverse depth* but it is rather an inverse distance). After the first observation, all parameters of $\mathbf{i}$ except ρ are immediately observable, and their values and covariances are obtained by proper inversion and linearization of the observation functions. The inverse depth ρ is initialized with a Gaussian $\mathcal{N}(\rho - \bar{\rho}; \sigma_\rho^2)$ such that in the depth dimension $s = 1/\rho$, we have

$$s_{(-n\sigma)} = \frac{1}{\bar{\rho} - n\sigma_\rho} = \infty \tag{2}$$

$$s_{(+n\sigma)} = \frac{1}{\bar{\rho} + n\sigma_\rho} = s_{\min} \tag{3}$$

with $s_{\min}$ the minimum considered depth and n the inverse depth shape factor. This gives $\bar{\rho} = 1/(2s_{\min})$ and, more remarkably

$$n\,\sigma_\rho = \bar{\rho}. \tag{4}$$

Importantly, values of $1 \leq n \leq 2$ assure from (2) that the infinity range is included in the parametrization with ample probability.

On subsequent updates, IDP achieves correct EKF operation (i.e., quasi-linear behavior) along the whole ray as long as the parallax shown by the new viewpoint is not too large. The linearity test in [20] is regularly evaluated. If passed, the landmark can be safely transformed into a 3-D Euclidean parametrization.

III. MULTICAMERA SLAM

The general scheme for the multicamera SLAM system is presented in this section. This scheme is particularized to deal with two different problems. The first one is the automatic self-calibration of a stereo rig while performing SLAM. The second one is a master–lave solution to cooperative monocular SLAM. Both setups are explained here, and their corresponding experiments are presented in Sections V and VI.

A. System Overview

We implement the multicamera SLAM system as follows. A central EKF-SLAM will hold the stochastic representation of the set of all cameras $\mathcal{C}_i$ plus the set of landmarks $\mathcal{L}_j$

$$X^\top = [\mathcal{C}_1^\top \quad \cdots \quad \mathcal{C}_N^\top \quad \mathcal{L}_1^\top \quad \cdots \quad \mathcal{L}_M^\top] \tag{5}$$

where the cameras states contain position and orientation quaternion $[\mathcal{C}_i = (\mathbf{r}_i, \mathbf{q}_i) \in \mathbb{R}^7]$, and landmarks can be coded either in inverse depth ($\mathcal{L}_j = \mathbf{i}_j \in \mathbb{R}^6$) or in Euclidean coordinates ($\mathcal{L}_j = \mathbf{p}_j \in \mathbb{R}^3$). Any number of cameras can be considered this way. As each camera needs to remain localized properly, it needs to observe a minimum number of landmarks at each frame. The algorithm's complexity increases linearly with the number of cameras if this number is small with respect to the map.

For camera motions, we consider two possible models. In the first one, a simple odometer provides motion predictions $[\Delta x, \Delta y, \Delta\psi]$ in the robot's local 2-D plane. Gaussian uncertainties are added to the 6-DOF linear and angular components $[x, y, z, \phi, \theta, \psi]$ with a variance proportional to the measured forward motion Δx

$$\{\sigma_x^2, \sigma_y^2, \sigma_z^2\} = k_L^2 \cdot \Delta x \tag{6}$$

$$\{\sigma_\phi^2, \sigma_\theta^2, \sigma_\psi^2\} = k_A^2 \cdot \Delta x. \tag{7}$$

The variance in $[\phi, \theta, \psi]$ is mapped to the quaternion space using the corresponding Jacobians.

The second model is a 6-DOF constant velocity model

$$\begin{aligned}
\mathbf{r}^+ &= \mathbf{r} + \mathbf{v}\,\Delta t \\
\mathbf{q}^+ &= \mathbf{q} \times \mathrm{v2q}(\omega\,\Delta t) \\
\mathbf{v}^+ &= \mathbf{v} + \eta_v \\
\omega^+ &= \omega + \eta_\omega
\end{aligned}$$

where $(\,)^+$ means the updated value, $\times$ is the quaternions product, and $\mathrm{v2q}(\omega\,\Delta t)$ transforms the local incremental rotation vector $\omega\,\Delta t$ into a quaternion (quaternions are systematically normalized). This way, the camera state vector $\mathcal{C}_i$ is augmented to $\mathcal{C}_i = (\mathbf{r}_i, \mathbf{q}_i, \mathbf{v}_i, \omega_i) \in \mathbb{R}^{13}$. At each time step, perturbations $\{\eta_v, \eta_\omega\} \sim \mathcal{N}(0; \{\sigma_v^2, \sigma_\omega^2\})$ add variances to the linear and angular velocities proportionally to the elapsed time Δt

$$\sigma_v^2 = k_v^2 \cdot \Delta t \tag{8}$$

$$\sigma_w^2 = k_\omega^2 \cdot \Delta t. \tag{9}$$

The events of camera motion, landmark initialization, and landmark observation are handled as in regular IDP-SLAM by just selecting the appropriate block elements from the SLAM state vector and covariances matrix, and applying the corresponding motion or observation models. For example, at the observation of landmark j from camera i, we would use the function $\mathbf{u}_j^i = \mathbf{h}(\mathcal{C}_i, \mathcal{L}_j)$, which will be explained later for the case of an IDP ray [see 11]. Before transforming IDP rays into points, the linearity test in [20] needs to hold for all cameras.

B. Stereo SLAM With Extrinsic Self-Calibration

Our approach is relevant to fully calibrated stereo rigs if they are small (10–20 cm, as in [12]) or if, having long baselines, their main extrinsic parameters can be continuously self-calibrated.

Not all of the six extrinsic parameters of a stereo rig (three for translation, three for orientation) need to be calibrated. In fact, the notion of *self-calibration* inherently requires the system to possess its own gauge. In our case, the metric dimensions or *scale factor* of the whole world–robot system can only be obtained either from the stereo rig baseline, which is one of the extrinsic parameters (then, it makes no sense to self-calibrate the gauge), or from odometry, which is often much less accurate than any coarse measurement we could make of this baseline. Additionally, as cameras are actually angular sensors, vision measurements are much more sensitive to the cameras orientations than to any translation parameter. This means that vision measurements will contain little information about these translation parameters. In consequence, self-calibration may concern only orientation, and more precisely, the orientation of one camera with respect to the other. The error of the reconstructed map's scale factor will be the same as the relative error of the baseline measurement.

With these assumptions, our self-calibration solution is straightforward: for the second camera, we just include its orientation in the map and let EKF make the rest. The state vector (5) is modified and written as

$$X^\top = [\mathcal{R}^\top \quad \mathbf{q}_R^\top \quad \mathcal{L}_1^\top \quad \cdots \quad \mathcal{L}_M^\top]$$

where $\mathcal{R}$ and $\mathcal{L}_1 \cdots \mathcal{L}_M$ are the robot pose and landmarks map. The left camera pose $\mathcal{C}_L$ has a fixed transformation with respect to the robot, and $\mathbf{q}_R$ is the orientation part of the right-hand camera $\mathcal{C}_R$ in the robot frame. The time-evolution function of the angular extrinsic parameters is simply $\mathbf{q}_R^+ = \mathbf{q}_R + \gamma$, where γ is a white, Gaussian, low-energy process noise that accounts for eventual decalibrations, e.g., due to vibrations. For short-duration experiments, we set $\gamma = 0$. A coarse analysis of the stereo structure's mechanical precision will be enough to set the initial uncertainty to a value of the order of 1° or 2° per axis. This can be reduced to a few tenths of degree in cases where we dispose of previous calibrated values about which we are not confident anymore.

C. *Cooperative Multicamera SLAM*

The ideal, most general case of cooperative SLAM (5), corresponds to a (not too large) number of cameras moving independently. Each camera is able to manage its own measurements and communicates directly with the map. The aim of this communication is to obtain information about existing landmarks to get localized, and provide information about new or reobserved landmarks. This way, the algorithms to be executed by each camera are absolutely symmetrical, without any kind of hierarchy. A simplified implementation considers cameras with different privileges.

In our particular case, the cooperative SLAM system considers two cameras. One of them takes the role of *master,* and is responsible for all landmarks detection and initialization. The second one acts as the *slave*. It follows the master at a close distance and reobserves the SLAM map that is being built by the master. By doing so, it provides a second viewpoint to landmarks just initialized, accelerating the convergence of the map. The master and slave trajectories are highly independent, and for instance, they can cross paths. The only requirement is to look in the same direction. A trivial extension to more than two cameras consists in including additional slaves.

IV. Perception and Map Management

Active search (AS, nicely described in [21] and also referred to as *top-down* in [6]) is a powerful framework for real-time image processing within SLAM. It has been successfully used in several monocular SLAM works [3], [5], [11], using a diversity of techniques for landmark initialization. The idea of AS is to exploit the information contained in the map to predict a number of characteristics of the landmarks to observe. AS is helpful in solving the following issues:

1) selecting interesting image regions for initialization;
2) selecting the most informative landmarks to measure;
3) predicting where in the image they may be found, and with which probability;
4) predicting the current landmark's appearance to maximize the chances of a successful match.

A. *Feature Detection and Initialization*

Based on the projection of the map information into the master image, a heuristic strategy is used to select a region of interest for a new initialization: we divide the image with a grid and randomly select a grid element with no landmarks inside. We extract the strongest Harris point [22] in this region and validate it if its strength is above a predefined threshold. We store a small rectangular region or *patch* of 15×15 pixels around the point as the landmark's appearance descriptor, together with the pose of the camera. Finally, we initialize the IDP ray in the SLAM map.

B. *Expectations: The Active Search Regions*

Some considerations about AS can be made for its usage in multicamera IDP–SLAM to improve performance. We use for this the $\mathcal{E}_1$ and $\mathcal{E}_\infty$ ellipses, defined and explained as follows.

1) $\mathcal{E}_1$ Ellipse: Expectation of the Inverse Depth Ray: The inverse depth ray (1) is easily projected into a camera. We take the transformation to camera frame given in [5]:

$$\mathbf{h}_1^{\mathcal{C}} = \mathbf{R}(\mathbf{q})^\top \left(\rho\left(\mathbf{x}_0 - \mathbf{r}\right) + \mathbf{m}(\theta, \psi)\right) \tag{10}$$

where $\mathbf{R}(\,)$ is the rotation matrix corresponding to the camera orientation $\mathbf{q}$ and $\mathbf{r}$ is the current camera position. This value is then projected into the camera, described by intrinsic and distortion parameters $\mathbf{k}$ and $\mathbf{d}$ (we use a classical radial distortion model of up to three parameters, which is inverted as explained in [19]). Let us call $\mathcal{K} = (\mathbf{k}, \mathbf{d})$ the camera parameters, $\mathcal{C} = (\mathbf{r}, \mathbf{q})$ the camera pose, and $\mathbf{i} = (\mathbf{x}_0, \theta, \psi, \rho)$ the IDP ray. The observation function is

$$\mathbf{u} = \mathbf{h}_1(\mathcal{C}, \mathcal{K}, \mathbf{i}) + \eta = \text{project}(\mathbf{h}_1^{\mathcal{C}}, \mathcal{K}) + \eta \tag{11}$$

where project () takes into account the camera model (we use perspective cameras) and η is the pixel Gaussian noise, with covariance $\mathbf{R}$.

We define the $\mathcal{E}_1$ ellipse as the Gaussian expectation $\mathcal{E}_1(\mathbf{u}) \triangleq \mathcal{N}(\mathbf{u} - \bar{\mathbf{e}}_1; \mathbf{E}_1)$, with $\mathbf{u}$ being the pixel position, and with mean and covariances matrix

$$\bar{\mathbf{e}}_1 = \mathbf{h}_1(\bar{\mathcal{C}}, \mathcal{K}, \bar{\mathbf{i}}) \tag{12}$$

$$\mathbf{E}_1 = [\mathbf{H}_{\mathcal{C}}\, \mathbf{H}_{\mathbf{i}}]\, \mathbf{P}_{\mathcal{C},\mathbf{i}}\, [\mathbf{H}_{\mathcal{C}}\, \mathbf{H}_{\mathbf{i}}]^\top + \mathbf{R}. \tag{13}$$

Here, $\mathbf{H}_{\mathcal{C}}$ and $\mathbf{H}_{\mathbf{i}}$ are the Jacobians of $\mathbf{h}_1$ with respect to the uncertain parameters $\mathcal{C}$ and $\mathbf{i}$, $\bar{\bullet}$ are variable estimates from the SLAM map, and $\mathbf{P}_{\mathcal{C},\mathbf{i}}$ is the joint covariances matrix (all correlations and cross correlations) of $\mathcal{C}$ and $\mathbf{i}$, also from the map. In AS, $\mathcal{E}_1$ is usually gated at 3σ, giving place to an elliptic region in the image where the landmark must project with 98% probability. However, this is not necessarily true in cases of noticeable parallax, as we examine now.

At landmark initialization, its inverse depth ρ is initialized according to (2)–(4). When considering 3σ uncertainty regions, (4) implies that ρ can go negative with a nonnegligible probability, meaning that the coded landmarks might be situated *behind the camera*. This becomes evident when projecting the IDP ray into a second camera presenting some parallax: the projected 3σ $\mathcal{E}_1$ ellipse contains a region with negative disparity (see

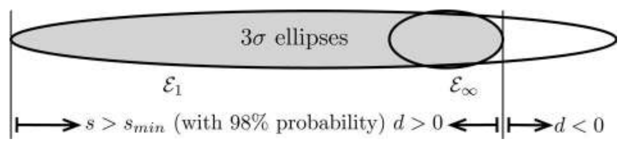

Fig. 5. 3σ search region defined by the $\mathcal{E}_1$ ellipse contains a significant part that corresponds to negative disparities $d < 0$, where the feature should not be searched. The final 3σ search region (gray) is defined by the $\mathcal{E}_1$ and $\mathcal{E}_\infty$ ellipses. The rightmost 3σ border of $\mathcal{E}_\infty$ is where the probability to find the projection of the infinity point has fallen below 2%.

Q1

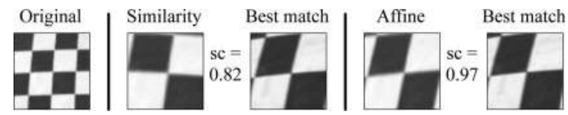

Fig. 6. Similarity and affine warping on a sample patch. From left to right: original patch; similarity warped patch (~180% scale, 10° rotation); best match in a later image affected by distortion and its zero mean normalized cross correlation (ZNCC) score (0.82); affine warped patch; best match and score (0.97). The affine warping contains a significant skew component mainly due to image distortion. The improvement in the ZNCC score is very important.

Fig. 5). It is desirable to limit the search area to values of only positive disparity for two reasons: the correlation-based search (one of the most time-consuming processes) is faster and the possibility of including false matches as outliers is diminished. With nonrectified images and/or camera sets with uncertain extrinsic parameters, determining the null disparity bound is not straightforward. One solution is to use the $\mathcal{E}_\infty$ ellipse, which we introduce in the following paragraph.

2) $\mathcal{E}_\infty$ Ellipse: Expectation of the Infinity Point: The infinity point is easily projected by considering the transformation (10) with $\rho \to 0$

$$\mathbf{h}_\infty^{\mathcal{C}} \approx \mathbf{R}(\mathbf{q})^\top \mathbf{m}(\theta, \psi) \tag{14}$$

where only the camera orientation $\mathbf{q}$ and the ray's direction angles (θ, ψ) are present (the visual compass). Proceeding as before, we obtain the definition of the ellipse $\mathcal{E}_\infty(\mathbf{u}) \triangleq \mathcal{N}(\mathbf{u} - \bar{\mathbf{e}}_\infty; \mathbf{E}_\infty)$ as

$$\bar{\mathbf{e}}_\infty = \mathbf{h}(\bar{\mathbf{q}}, \mathcal{K}, \bar{\theta}, \bar{\psi}) \tag{15}$$

$$\mathbf{E}_\infty = [\mathbf{H}_\mathbf{q}\, \mathbf{H}_\theta\, \mathbf{H}_\psi]\, \mathbf{P}_{\{\mathbf{q},\theta,\psi\}}\, [\mathbf{H}_\mathbf{q}\, \mathbf{H}_\theta\, \mathbf{H}_\psi]^\top + \mathbf{R} \tag{16}$$

where $\mathbf{P}_{\{\mathbf{q},\theta,\psi\}}$ is the joint covariances matrix of the uncertain parameters. The $\mathcal{E}_\infty$ 3σ region is composed of the previous $\mathcal{E}_1$ region, as indicated in Fig. 5, to define the search area.

C. Selection of the Best Map Updates

Following the AS approach in [23], a predefined number of landmarks with the biggest $\mathcal{E}_1$ ellipse surfaces are selected in each camera as those being the most interesting to be measured. For each camera, we organize all candidates (visible landmarks) in descending order of expectation surfaces, without caring if they are points or rays. We update at each frame a predefined number of them (usually around 10, and no more than 20). Updates are processed sequentially, with all Jacobians being recalculated each time to minimize the effect of linearization errors.

D. Feature Matching: Affine Patch Warping

AS continues by *warping* the stored patch and searching for a correlation peak inside the search area earlier. The objective of warping is to predict the landmark's current appearance, maximizing the chances for a good match. In the absence of distortion, a planar homography $\mathbf{H} \in \mathbb{R}^{3\times3}$, defined in the homogeneous spaces, would be desirable [24]. This type of warping requires the online estimation of the patch normal in the 3-D space, and may become very time-consuming. A good simplification considers this normal fixed at the initial visual axis [23]. Further simplification applies just a similarity transformation $\mathbf{T} = s\mathbf{R} \in \mathbb{R}^{2\times2}$ in the image Euclidean plane [19]. This accounts only for scale changes s and rotations $\mathbf{R}$ obtained from the stored information (landmark position, camera initial, and current poses). However, in the presence of distortion, features lying close to the image borders suffer from additional deformations. We developed a warping approach that easily adds a skew component to the operator $\mathbf{T}$ (thus achieving fully affine warping, but not perspective warping; Fig. 6), based on the Jacobian of the function linking the first observation to the current one. Let us consider the backward observation model $\mathbf{g}(\,)$ for a camera A at initialization time $t = 0$, and the observation model $\mathbf{h}(\,)$ for a different camera B at current time $t \geq 0$

$$\mathbf{p} = \mathbf{g}(\mathcal{C}_A(0), \mathcal{K}_A, \mathbf{u}_A(0), s_A)$$
$$\mathbf{u}_B(t) = \mathbf{h}(\mathcal{C}_B(t), \mathcal{K}_B, \mathbf{p}).$$

Here, $\mathbf{p}$ is the landmark's position, $\mathcal{K}_i = (\mathbf{k}_i, \mathbf{d}_i)$ are the intrinsic and distortion parameters of camera i, $\mathbf{u}_i(t)$ is the measured pixel, and s_A is the landmark's depth with respect to the initial camera. We can compose these functions to obtain the expression linking the initial and the current pixels

$$\mathbf{u}_B(t) = \mathbf{h}\left[\mathcal{C}_B(t), \mathcal{K}_B, \mathbf{g}(\mathcal{C}_A(0), \mathcal{K}_A, \mathbf{u}_A(0), s_A)\right]. \tag{17}$$

When all but the pixel positions are fixed, this represents an invertible mapping $\mathbb{R}^2 \mapsto \mathbb{R}^2$ from the pixels in the first image to the pixels in the current one. The local linearization around the initially measured pixel defines an affine warping expressed by the Jacobian matrix

$$\mathbf{T} = \left.\frac{\partial \mathbf{u}_B}{\partial \mathbf{u}_A}\right|_{(\mathcal{C}_A(0), \mathcal{C}_B(t), \mathcal{K}_A, \mathcal{K}_B, \mathbf{u}_A(0), s_A)}. \tag{18}$$

By defining $\tilde{\mathbf{u}}_i$ as the coordinates of the patch in camera i, with the central pixel $\mathbf{u}_i$ as the origin, we have $\tilde{\mathbf{u}}_B(t) = \mathbf{T}\,\tilde{\mathbf{u}}_A(0)$. Based on this mapping, we use linear interpolation of the pixels' luminosity to construct the warped patch.

V. Experiment 1: Stereo SLAM With Self-Calibration

The "White-board" indoor experiment aims at demonstrating stereovision SLAM with self-calibration. A robot with a stereo head looking forward is run for about 10 m in straight line inside the robotics laboratory at the LAAS (see Fig. 7). Over 500 image pairs are taken at approximately 5-Hz frequency. The robot

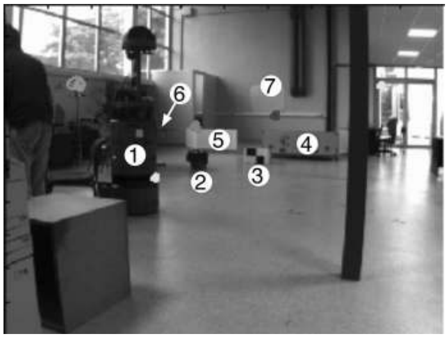

Fig. 7. Laboratoire d'Analyse et d'Architecture des System (LAAS) robotics laboratory. The robot will approach the scene in a straightforward trajectory. We notice in the scene the presence of a robot ①, a bin ②, a box ③, a trunk ④, a fence ⑤, a table ⑥ (hidden by the robot in this image), and the white board ⑦ at the end wall.

TABLE I
STEREO RIG PARAMETERS IN THE "WHITE-BOARD" EXPERIMENT

Scope	Parameters = Values
Dimensions	Baseline = 33 cm
Orientation - Euler	$\{\phi, \theta, \psi\} = \{0^\circ, 5^\circ, 0^\circ\}$
Cameras	{resolution,FOV} = $\{512 \times 384\,\text{pix}, 55^\circ\}$
Right camera uncertainties	$\{\sigma_\phi, \sigma_\theta, \sigma_\psi\} = \{1^\circ, 1^\circ, 1^\circ\}$

moves towards the objects to be mapped at 0.15 m/s. The stereo rig consists of two intrinsically calibrated cameras arranged as indicated in Table I. The orientations of both cameras are specified with respect to the robot frame. The left camera is taken as reference, thus deterministically specified, and the orientation of the right one is initialized with an uncertainty of 1° standard deviation. We use the odometry model (Section III-A) with $k_L = 0.1\ \text{m}/\sqrt{\text{m}}$ and $k_A = 0.05\ \text{rad}/\sqrt{\text{m}}$.

We show details and results on the self-calibration procedure and the metric accuracy of the resulting map. The mapping process can be appreciated in the movie `whiteboard.mov` in the multimedia section.

A. Self-Calibration

We plot in Fig. 8 left the evolution of the three self-calibrated angles. We have also used the shape of the $\mathcal{E}_\infty$ ellipses to provide additional qualitative evidence of the calibration process (Fig. 9 and movie `whiteboard − einf.mov`). We observe the following behavior.

1) Pitch θ: The pitch angle (cameras tilt, 5° nominal value) is observable from the first matched landmark. It rapidly converges to an angle of 4.77° and remains very stable during the whole experiment.

2) Roll ϕ: Roll angle is observable after at least two landmarks are observed from the right camera. Once this condition holds, convergence occurs relatively fast.

3) Yaw ψ: Yaw angle is very weakly observable because it is coupled with the landmarks depths: both yaw angle and landmark depth variations produce a similar uncertainty growth in the right image. For this reason, yaw converges slowly, only showing reasonable convergence after some 50 frames.

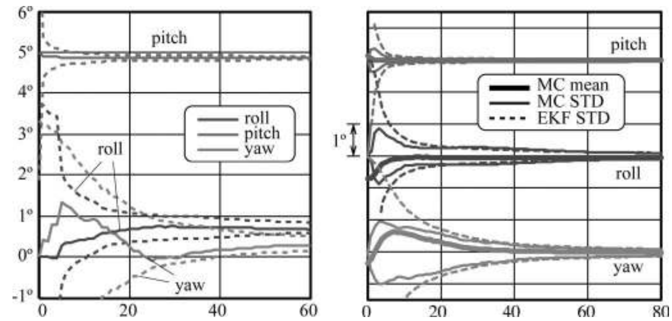

Fig. 8. Extrinsic self-calibration. (Left) The three Euler angles of the right camera orientation with respect to the robot during the first 60 frames. The 3σ bounds are plotted in dotted line showing consistent estimation. (Right) Error analysis after 100 MC runs using 200 frames per run (only the first 80 frames are shown). The thick solid lines represent the means over the 100 runs. The 3σ bounds for each angle are plotted using thin solid lines. The dotted lines represent the averaged 3σ bounds estimated by the EKF, showing consistent calibration.

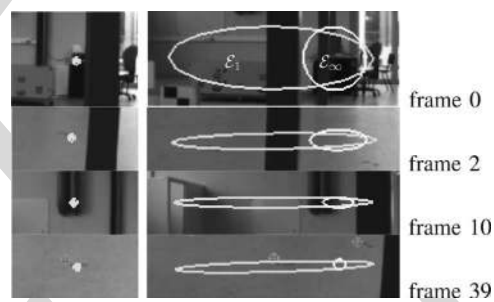

Fig. 9. Evolution of the $\mathcal{E}_1$ and $\mathcal{E}_\infty$ ellipses during calibration. On the left column, newly detected pixels in the left image. On the right, expectations in the right image (green) $\mathcal{E}_1$ and (yellow) $\mathcal{E}_\infty$ of the newly initialized IDP rays (i.e., still with the full initial uncertainty in ρ). At frame 0, initial uncertainties of 1° result in a big, round $\mathcal{E}_\infty$ ellipse. After the first updated landmark from the left camera (frame 2), the uncertainty in pitch decreases and $\mathcal{E}_\infty$ becomes flat. Successive updates further refine the calibrated angles. The yaw angle takes longer to converge, but the tiny $\mathcal{E}_\infty$ in frame 39 shows that the calibration is already finished. The portion of the green ellipse on the right side of the yellow one corresponds to negative disparities and is not searched for matches. This portion is larger as parallax increases.

TABLE II
MC ANALYSIS OF THE SELF-CALIBRATION

Angle	MC mean	STD	EKF STD	Offline	STD
roll	0.69°	0.028°	0.018°	0.61°	0.013°
pitch	4.77°	0.003°	0.005°	4.74°	0.099°
yaw	0.33°	0.021°	0.016°	0.51°	0.109°

In Fig. 8 right, we plot results of a Monte Carlo (MC) analysis, run over the data of this experiment, for the mean and standard deviation of the Euler angles of the right camera. Because all MC runs are extracted from the same sequence, we tried to maximize their independence by using a different *random seed* in the algorithm (acting in the random selection of the initialization region, Section IV-A), and by starting each run at a different frame. The figure shows that the dynamic estimation is consistent (the EKF estimated sigmas are larger than the MC ones). After 200 frames, we compare these values with those of the offline calibration [25]. Table II summarizes these results, showing MC [(means and standard deviations (STD)] and Kalman Filter (EKF, showing the estimated STD). All

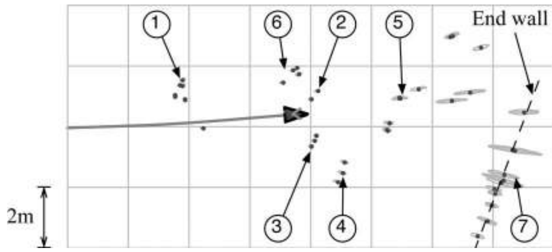

Fig. 10. Map produced during the "white board" experiment. We marked the mapped robot ①, the bin ②, the box ③, the trunk ④, the fence ⑤, the table ⑥, and the white board ⑦ at the end wall.

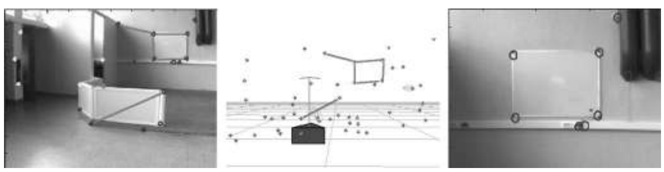

Fig. 11. Metric mapping. The magnitudes of some segments in the real laboratory are compared to those in the map (*red lines*). Ground truth corresponds to metric measurements of the distances between landmarks that are identified by zooming in the last image of the experiment (right) and translated to the real world. Thirteen points on the end wall are tested for coplanarity.

TABLE III
WHITE BOARD: MAP TO GROUND TRUTH TOMPARISON

segment	board	board	board	board	wall	fence
real (cm)	116	86	117	88	136	124
mapped	116.6	87.2	115.8	87.0	135.1	125.5
STD	0.91	0.81	1.21	0.52	1.06	1.32

self-calibrated values lie within the 3σ bounds defined by the offline mean and STD values.

B. Metric Accuracy

We show in Fig. 10 a top view of the map generated during this experiment. To contrast this map against reality, two tests are performed: planarity and metric scale (see Fig. 11): 1) the four corners of the white board are taken together with nine other points at the end wall to test coplanarity: the 13 mapped points are found to be coplanar within 4.9 cm STD; 2) the lengths of the real and mapped segments marked in Fig. 11 are summarized in Table III. The white board has a physical size of 120 cm $\times$ 90 cm, but we take real measurements from the approximated corners where the features are detected. We observe errors in the order of 1 cm for landmarks that are still about 4 m away from the robot.

VI. EXPERIMENT 2: COOPERATIVE MONOCULAR SLAM

This experiment shows independent cameras collaborating to build a 3-D map using exclusively bearings-only observations. Two independent cameras are placed on top of two bicycles looking forward, moving on different trajectories in the parking of the LAAS (see Fig. 12). Over 1000 images are taken by each camera at 15-Hz frequency, 512×384 pixel resolution, $100°$ field of view (FOV), and are processed offline. The cameras travel approximately 28 m observing landmarks beyond 60 m. As in the previous experiment, the left camera is the master. The two trajectories start parallel to each other, separated 75 cm perpendicularly to the motion direction. The reference frame is defined by the master camera initial position and orientation, which are initialized with null uncertainty. The scale factor is determined by the initial baseline of 75 cm, meaning that the position of the slave camera in the lateral Y-axis is also initialized with null uncertainty. The orientations of the slave camera start with an uncertainty of $2°$ STD, and its position in the frontal Y- and vertical Z-axes with $75\,\mathrm{cm}\cdot\sin(2°) = 2.6\,\mathrm{cm}$ STD. With these uncertainties, the experiment's initial configuration can be set up manually by just observing the images and centering the projections of some distant object. We use two independent constant-velocity models with $k_v = 0.3$ m/s $\cdot \sqrt{\mathrm{s}}$ and $k_w = 0.3$ rad/s $\cdot \sqrt{\mathrm{s}}$. The measurement noise is 1 pixel.

Fig. 12. Snapshots of master and slave sequences in cooperative SLAM. Faraway landmarks (e.g., black arrowed), still initialized as rays (red), are the ones fixing the orientation. Nearby landmarks, usually as Euclidean points (blue), maintain the metric. A virtual model of the master camera is visible from the slave camera (white arrowed). See `cooperativeSLAM.mov`.

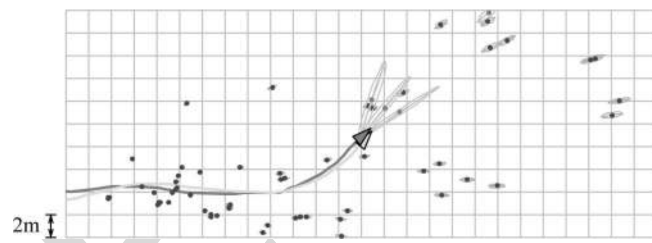

Fig. 13. Top view of the map produced by cooperative SLAM of two independent cameras, and their crossing trajectories. The grid spacing is 2 m.

Landmarks at infinity, illumination changes and few salient features are some characteristics of this outdoors scene. It presents relatively few stable landmarks, something that makes the correct operation of the SLAM system difficult. In the case of having crossing trajectories, the problem of one camera occluding the other could appear and severely affect the image processing. To avoid this, we decided to take both image sequences shifted in time, i.e., one after the other, and make them overlap for processing. The mapping process is presented in the movie `cooperativeSLAM.mov` in the multimedia section. Fig. 13 shows the top view of the map and the camera trajectories generated during this experiment.

A proper metrical evaluation of this experiment is difficult; having a variable baseline does not allow us to compare the results, because there is no knowledge of the ground truth. In order to evaluate this approach, we consider the setup in experiment

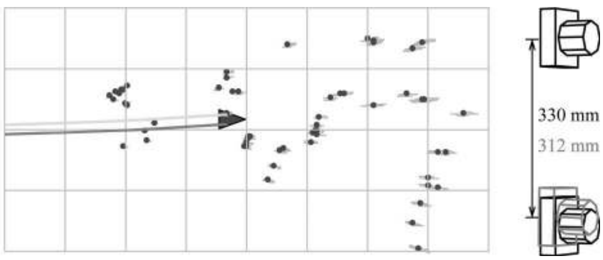

Fig. 14. Final map in the "white board" setup using the cooperative monocular SLAM algorithm. The cameras are modeled as being entirely independent using the same data and initial configuration as in Experiment 1. The stereo rig on the right shows (red) the final estimated relative position compared with (black) ground truth.

1 and apply the same algorithm. The new experiment consists of recovering the full extrinsic calibration, which is fixed in reality, considering both cameras as independent. Again, we use a constant-velocity model for each camera. The initial setup including uncertainties is as in experiment 1.

Fig. 14 shows the obtained map. We see that it compares very well to the map obtained in experiment 1 (see Fig. 10), where the motions of the two cameras were constrained by the stereo rig and a common motion was predicted using odometry. Fig. 14 bottom shows a detail of the cameras in their final relative position. We measure an error along the baseline of less than 2 cm. The orientation errors are less than 0.7°.

VII. Conclusion

We showed in this paper that fusing the visual information with monocular methods while performing multicamera SLAM provides several advantages: the ability to consider points at infinity, desynchronization of the different cameras, the use of any number of cameras of different types, sensor self-calibration, and the possibility to conceive decentralized schemes that will make realistic multirobot monocular SLAM possible. Except for decentralization, these advantages have been explored with the inverse depth monocular SLAM algorithm, and applied to two different problems: stereovision SLAM with an extrinsically decalibrated stereo rig and cooperative SLAM of two independently moving cameras.

Both demonstrations employed a *master–slave* approach, which made solving some of the issues of map and image management easier, and we are now improving on this by implementing a fully symmetrical approach. This approach should easily permit the extension of the presented applications to cases with more than two cameras. In parallel to these activities, we started new work on landmark parametrization to improve EKF linearity in cases of increasing parallax. Also, as parallax increases, landmarks appearances may change too much as to guarantee a stable operation with the matching methods presented here. We believe that wide baseline feature matching will be the bottleneck of visual SLAM for some time to come. As for decentralization, we note that it demands a full reformulation of the fusion engines we use in this paper (one central EKF), for example, via channel filters, and is currently a subject of intense research at LAAS and other laboratories.

References

[1] J. Solà, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, Aug. 2–6, 2005, pp. 2499–2504.

[2] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "Structure from motion causally integrated over time," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 523–535, Apr. 2002.

[3] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. Int. Conf. Comput. Vis.*, Nice, France, Oct. 2003, vol. 2, pp. 1403–1410.

[4] J. Civera, A. J. Davison, and J. M. M. Montiel, "Dimensionless monocular SLAM," in *Proc. Iberian Conf. Pattern Recognit. Image Anal.*, Jun, 2007, pp. 412–419.

[5] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, Oct. 2008.

[6] E. Eade and T. Drummond, "Scalable monocular SLAM," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun.17–22, 2006, vol. 1, pp. 469–476.

[7] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment—A modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. LNCS, W. Triggs, A. Zisserman, and R. Szeliski, Eds. New York: Springer-Verlag, 2000, pp. 298–375.

[8] K. Konolige, "SLAM via variable reduction from constraints maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 18–22, 2005, pp. 667–672.

[9] J. Folkesson, P. Jensfelt, and H. I. Christensen, "Vision SLAM in the measurement subspace," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 18–22, 2005, pp. 30–35.

[10] J. Diebel, K. Reuterswärd, S. Thrun, and R. G. J. Davis, "Simultaneous localization and mapping with active stereo vision," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sendai, Japan, Oct. 2004, vol. 4, pp. 3436–3443.

[11] J. Solà, A. Monin, and M. Devy, "BiCamSLAM: Two times mono is more than stereo," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 4795–4800.

[12] L. M. Paz, P. Piniés, J. Tardós, and J. Neira, "Large scale 6 DOF SLAM with stereo-in-hand," *IEEE Trans. Robot.*, vol. 24, no. 5, Oct. 2008.

[13] A. Mallet, S. Lacroix, and L. Gallo, "Position estimation in outdoor environments using pixel tracking and stereovision," in *Proc. Int. Conf. Robot. Autom.*, San Francisco, CA, 2000, vol. 4, pp. 3519–3524.

[14] K. Konolige, M. Agrawal, and J. Solà, "Large-scale visual odometry for rough terrain," presented at the Int. Symp. Res. Robot., Hiroshima, Japan, Nov. 2007.

[15] T. D. Barfoot, "Online visual motion estimation using FastSLAM with SIFT features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2–6, 2005, pp. 579–585.

[16] A. I. Comport, E. Malis, and P. Rives, "Accurate quadrifocal tracking for robust 3D visual odometry," in *Proc. Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 10–14, 2007, pp. 40–45.

[17] E. M. Foxlin, "Generalized architecture for simultaneous localization, auto-calibration, and map-building," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Lausanne, Switzerland, 2002, vol. 1, pp. 527–533.

[18] E. Nettleton, H. Durrant-Whyte, and S. Sukkarieh, "A robust architecture for decentralised data fusion," presented at the Int. Conf. Adv. Robot., Coimbra, Portugal, 2003.

[19] J. Solà, "Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach" Ph.D. dissertation, Inst. Nat. Polytech. de Toulouse, Toulouse, France, 2007.

[20] J. Civera, A. Davison, and J. Montiel, "Inverse depth to depth conversion for monocular SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 10–14, 2007, pp. 2778–2783.

[21] A. J. Davison, "Active search for real-time vision," in *Proc. Int. Conf. Comput. Vis.*, 2005, vol. 1, pp. 66–73.

[22] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vis. Conf.*, Manchester, U.K., 1988, pp. 189–192.

[23] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[24] N. Molton, A. J. Davison, and I. Reid, "Locally planar patch features for real-time structure from motion," presented at the Brit. Mach. Vis. Conf., Kingston, U.K., 2004.

[25] K. Strobl, W. Sepp, S. Fuchs, C. Paredes, and K. Arbter. (2006). Camera calibration toolbox for Matlab. Inst. Robot. Mechatronics, Wessling, Germany, Tech. Rep. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

Joan Solà was born in Barcelona, Spain, in 1969. He received the B.Sc. degree in telecommunications and electronic engineering from the Universitat Politècnica de Catalunya, Barcelona, in 1995, the M.Sc. degree in control systems from the École Doctorale Systèmes, Toulouse, France, in 2003, and the Ph.D. degree in control systems from the Institut National Polytechnique de Toulouse in 2007, where he was hosted by the Laboratoire d'Analyse et d'Architecture des System (LAAS), Centre National de la Recherche Scientifique (CNRS).

He was a Postdoctoral Fellow at SRI International, Menlo Park, CA. He is currently at LAAS-CNRS, where he is engaged in research on visual localization and mapping. His current research interests include estimation and data fusion applied to off-road navigation, mainly using vision.

André Monin was born in Le Creusot, France, in 1958. He received the Graduate degree from the Ecole Nationale Supérieure dIngénieurs Electriciens de Grenoble, Grenoble, France, in 1980, the Ph.D. degree in nonlinear systems representation from the University Paul Sabatier, Toulouse, France, in 1987, and the Habilitation pour Diriger des Recherches degree from the University Paul Sabatier in 2002.

From 1981 to 1983, he was a Teaching Assistant with the Ecole Normale Supérieure de Marrakech, Marrakech, Morocco. Since 1985, he has been with the Laboratoire Automatique et d'Analyse des Systèmes, Centre Nationale de la Recherche Scientifique (LAAS-CNRS), Toulouse, as the "Chargé de Recherche." His current research interests include the areas of nonlinear filtering, systems realization, and identification.

Michel Devy received the degree in computer science engineering from the Ecole Nationale Supérieure dIngénieurs Electriciens de Grenoble, Grenoble, France, in 1976 and the Ph.D. degree from the Laboratoire Automatique et d'Analyse des Systèmes, Centre Nationale de la Recherche Scientifique (LAAS-CNRS), Toulouse, France, in 1980.

Since 1980, he has been with the Department of Robotics and Artificial Intelligence, LAAS-CNRS, where he is the Research Director and the Head of the Research Group Robotics, Action, and Perception. His current research interests include computer vision for automation and robotics applications. He has also been involved in numerous national and international projects concerning, about field and service robots, 3-D vision for intelligent vehicles, 3-D metrology, and others. He has authored or coauthored about 150 scientific communications.

Teresa Vidal-Calleja received the B.Sc. degree in mechanical engineering from the Universidad Nacional Autónoma de México, México City, México, in 2000, the M.Sc. degree in mechatronics from CINVESTAV-IPN, México City, in 2002, and the Ph.D. degree in robotics, automatic control, and computer vision from the Universitat Politèctica de Catalunya, Barcelona, Spain, in 2007.

She was a Visiting Research Student with the University of Oxford's Robotics Research Group, the Australian Centre for Field Robotics, and the University of Sydney. She is currently a Postdoctoral Fellow with the Robotics and Artificial Intelligence Group, Laboratoire Automatique et d'Analyse des Systèmes, Centre Nationale de la Recherche Scientifique (LAAS-CNRS), Toulouse, France. Her current research interests include autonomous vehicles, perception, control, and cooperation.

QUERIES

Q1: Author: Please reframe the references to colors in the caption of Figs. 5, 8, 9, 11, 12, and 14, if the artwork is not being produced in color

Q2. Author: Please check if the details of the academic degrees received by A. Monin are OK as edited.

Q3. Author: Please provide the title of first degree. Also, provide the subject (physics, mathematics, electrical engineering, etc.) in which M. Dey received the Ph. D. degree.