International Journal of Computer Vision 74(3), 343–364, 2007
© 2007 Springer Science + Business Media, LLC. Manufactured in the United States.
DOI: 10.1007/s11263-007-0042-3

Vision-Based SLAM: Stereo and Monocular Approaches

THOMAS LEMAIRE*, CYRILLE BERGER, IL-KYUN JUNG AND SIMON LACROIX
LAAS/CNRS 7, Ave du Colonel Roche, F-31077 Toulouse Cedex 4, France
Thomas.Lemaire@laas.fr

Received November 18, 2005; Accepted January 23, 2007

First online version published in February, 2007

Abstract. Building a spatially consistent model is a key functionality to endow a mobile robot with autonomy. Without an initial map or an absolute localization means, it requires to concurrently solve the localization and mapping problems. For this purpose, vision is a powerful sensor, because it provides data from which stable features can be extracted and matched as the robot moves. But it does not directly provide 3D information, which is a difficulty for estimating the geometry of the environment. This article presents two approaches to the SLAM problem using vision: one with stereovision, and one with monocular images. Both approaches rely on a robust interest point matching algorithm that works in very diverse environments. The stereovision based approach is a classic SLAM implementation, whereas the monocular approach introduces a new way to initialize landmarks. Both approaches are analyzed and compared with extensive experimental results, with a rover and a blimp.

Keywords: bearing only SLAM, interest point matching, 3D SLAM

1. Introduction

Autonomous navigation is a fundamental ability for mobile robots, be it for ground rovers, indoor robots, flying or underwater robots. It requires the integration of a wide variety of processes, from low-level actuator control, to higher-level strategic decision making, via environment mapping and path planning. Among these various functionalities, self-localization is an essential one, as it is required at various levels in the whole system, from mission supervision to fine trajectory execution control:

- The missions to be achieved by the robot are often expressed in localization terms, explicitly (e.g. "reach that position" or "explore this area") or more implicitly, such as "return to the initial position".
- Autonomous navigation calls for the building of *global maps* of the environment, to compute trajectories or paths and to enable mission supervision. A good estimate of the robot position is required to guarantee the *spatial consistency* of such maps.
- Finally, the correct execution of the geometric trajectories provided by the planners relies on the precise knowledge of robot motions.

It is well known that dead reckoning techniques generate position estimates with unbounded error growth, as they compute the robot position from the composition of elementary noisy motion estimates. Such approaches can fulfill the localization requirements for local trajectory execution control, but do not allow global localization. Visual motion estimation techniques from monocular sequences (Heeger and Jepson, 1992; Vidal et al., 2001) or stereovision sequences (Zhang and Faugeras, 1992; Mallet et al., 2000; Olson et al., 2000) provide precise motion estimates between successive data acquisitions, but they are akin to dead reckoning.

The only solution to guarantee bounded errors on the position estimates is to rely on stable environment features. If a spatially consistent map of the environment is available, map-based localization can be applied: a number of successful approaches have been reported, e.g. (Borges and Aldon, 2002; Dellaert et al., 1999).[1] On the other hand, if the robot position is perfectly known, building an environment map with the perceived data is quite trivial, the only difficulty being to deal with the

*Corresponding author.

uncertainty of the geometric data to fuse them in a geometric representation.

Simultaneous Localization and Mapping. In the absence of an *a priori* map of the environment, the robot is facing a kind of "chicken and egg problem": it makes observations on the environment that are corrupted by noise, from positions which estimates are also corrupted with noise. These errors in the robot's pose have an influence on the estimate of the observed environment feature locations, and similarly, the use of the observations of previously perceived environment features to locate the robot provide position estimates that inherits from both errors: the errors of the robot's pose and the map features estimates are correlated.

It has been understood early in the robotic community that the mapping and the localization problems are intimately tied together (Chatila and Laumond, 1985; Smith et al., 1987), and that they must therefore be concurrently solved in a unified manner, turning the chicken and egg problem into a virtuous circle. The approaches that solve this problem are commonly denoted as "Simultaneous Localization and Mapping", and have now been thoroughly studied. In particular, stochastic approaches have proved to solve the SLAM problem in a consistent way, because they explicitly deal with sensor noise (Thrun, 2002; Dissanayake et al., 2001).

Functionalities Required by SLAM. The implementation of a feature-based SLAM approach encompasses the following four basic functionalities:

- **Environment feature selection.** It consists in detecting in the perceived data, features of the environment that are salient, easily observable and whose relative position to the robot can be estimated. This process depends on the kind of environment and on the sensors the robot is equipped with: it is a *perception process*, that represents the features with a specific data structure.
- **Relative measures estimation.** Two processes are involved here:
  - Estimation of the feature location relatively to the robot pose from which it is observed: this is the *observation*.
  - Estimation of the robot motion between two feature observations: this is the *prediction*. This estimate can be provided by sensors, by a dynamic model of robot evolution fed with the motion control inputs, or thanks to simple assumptions, such as a constant velocity model.
- **Data association.** The observations of landmarks are useful to compute robot position estimates only if they are perceived from different positions: they must imperatively be properly associated (or matched), otherwise the robot position can become totally inconsistent.
- **Estimation.** This is the core of the solution to SLAM: it consists in integrating the various relative measurements to estimate the robot and landmarks positions in a common global reference frame. The stochastic approaches incrementally estimate a posterior probability distribution over the robot and landmarks positions, with all the available relative estimates up to the current time. The distribution can be written as

  $$p(X_r(k), \{X_f(k)\}|\mathbf{z}, \mathbf{u}) \tag{1}$$

  where $X_r(k)$ is the current robot's state (at time k) and $\{X_f(k)\}$ is the set of landmark positions, conditioned on all the feature relative observations $\mathbf{z}$ and control inputs $\mathbf{u}$ that denote the robot motion estimates.

Besides these essential functionalities, one must also consider the *map management* issues. To ensure the best position estimates as possible and to avoid high computation time due to the algorithmic complexity of the estimation process, an active way of selecting and managing the various landmarks among all the detected ones is desirable. The latter requirement has lead to various essential contributions on the estimation function in the literature: use of the information filter (Thrun et al., 2002), splitting the global map into smaller sub-maps (Leonard and Feder, 2001), delayed integration of observations (Guivant and Nebot, 2001) (in turn, it happens that the use of sub-maps can help to cope with the non linearities that can hinder the convergence of a Kalman filter solution (Castellanos et al., 2004)).

Vision-Based SLAM. Besides obvious advantages such as lightness, compactness and power saving that make cameras suitable to embed in any robot, vision allows the development of a wide set of essential functionalities in robotics (obstacle detection, people tracking, visual servoing...). When it comes to SLAM, vision also offers numerous advantages: first, it provides data perceived in a solid angle, allowing the development of 3D SLAM approaches in which the robot state is expressed by 6 parameters. Second, visual motion estimation techniques can provide very precise robot motion estimates. Finally and more importantly, very stable features can be detected in the images, yielding the possibility to derive algorithms that allow to match them under significant viewpoint changes: such algorithms provide robust data association for SLAM.

Most SLAM approaches rely on the position estimates of the landmarks and the robot to associate the landmarks: the landmark observations are predicted from their positions and the current robot position estimate, and compared to the current observations. When the errors on

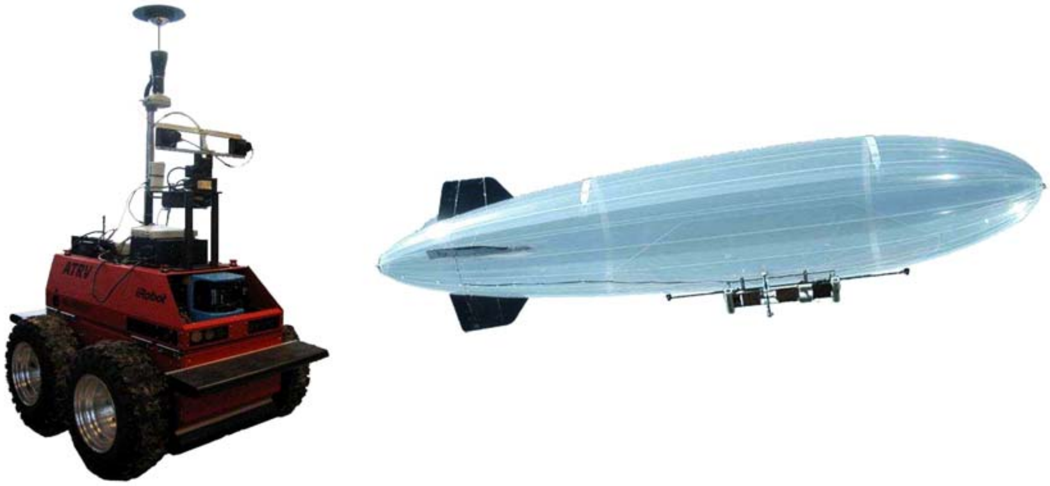


Figure 1. The ATRV rover Dala and the 10 m long blimp Karma. Both robots are equipped with a stereovision bench.

some of these positions are large, e.g. when the robot re-perceives landmarks after having traveled along a long loop trajectory for instance, the associations can become ambiguous. This is all the more difficult when the robot is evolving in 3D, the errors in the prediction of the 6 parameters of the robot position having rapidly a drastic influence on the predicted landmark positions. A robust way to solve the data association problem is to *recognize* the landmarks, independently from their position estimate: a good visual feature detection and matching algorithm can provide this ability.

Paper Outline. Section 2 presents a robust interest points matching algorithm that fulfills the feature selection and data association functionalities. The process matches Harris points detected in images by combining the image signal information and geometric constraints between the detected points. Section 3 presents two vision-based SLAM approaches that use 3D points as landmarks: an approach that relies on stereovision, in which the landmark positions are fully observed from a single position, and a *bearing-only* approach that exploits monocular sequences. The latter is emphasized, as it must deal with the fact that the landmark 3D state can not be observed from a single position. Section 4 then presents and analyses results obtained with an outdoor rover and a blimp (Fig. 1).

2. Feature Detection and Matching

To fulfill the data association functionality in a SLAM approach, a feature detection and matching algorithm should provide matches robust with respect to the variations in the images due to noise and illumination conditions, to viewpoint changes and to variations in the perceived scene itself.

Any solution to the image feature matching problem calls for three steps (Brown, 1992): definition of the feature space, definition of a similarity measure over the feature space, and match search strategy. The definition of the features to match is of course essential, as it conditions the whole process. Features can be directly the image signal, or edges, contours, lines, regions detected on the image, up to higher level semantic information. Using lower level features avoids the use of fragile segmentation algorithms: many contributions have therefore focused on the matching problem using directly the image signal as the feature space. The literature abounds with contributions on matching methods based on local gray values similarity scores (Shi and Tomasi, 1994; Zabih and Woodfill, 1994; Martin and Crowley, 1995). But in order to generate reliable matches, these approaches require to focus the match search (e.g. assuming the transformation between the two images is close to identity, or using a known epipolar constraint). In a SLAM context, such approaches can help to match features from consecutive positions, but they can hardly provide data associations when the robot has made large motions.

To establish matches when several unknown changes occur in the image, one must consider features that are as much invariant as possible with respect to any image transformation. Point features, often denoted as "interest points", are salient in images, have good invariant properties, and can be extracted with much less computation. A comparison of various interest points detectors is presented in Schmid et al. (1998): it introduces a modified version of the Harris detector (Harris and Stephens, 1988) which uses Gaussian functions to compute the two-dimensional image derivatives, and that gives the best *repeatability* under rotation and scale changes (the repeatability being defined as the percentage of repeated interest points between two images). However the repeatability steeply decreases with significant scale changes: in such cases, a scale adaptive version of the Harris detector is required to allow point matching (Dufournaud et al., 2004). When no information on scale change is available, matching features becomes quite time consuming, scale being an additional dimension to search through. To avoid this, scale invariant feature detection algorithms have been proposed (Lowe, 1999). However, these methods generate much less features than the standard or scale adaptive detectors, especially in unstructured or highly textured environments, and require more computing time.

2.1. *Interest Points*

To locate points in the image where the signal changes bi-directionally, the Harris corner detector computes the local moment matrix M of two normal gradients of intensity for each pixel $\mathbf{x} = (u, v)$ in the image (Harris and Stephens, 1988):

$$M(\mathbf{x}, \tilde{\sigma}) = G(\mathbf{x}, \tilde{\sigma}) \otimes \begin{pmatrix} I_u(\mathbf{x})^2 & I_u(\mathbf{x})I_v(\mathbf{x}) \\ I_u(\mathbf{x})I_v(\mathbf{x}) & I_v(\mathbf{x})^2 \end{pmatrix} \quad (2)$$

where $G(., \tilde{\sigma})$ is the Gaussian kernel of standard deviation $\tilde{\sigma}$, and $I_u(.)$ and $I_v(.)$ are the first order derivatives of the intensity respectively in the u and v directions. The eigenvalues (λ_1, λ_2) of $M(\mathbf{x}, \tilde{\sigma})$ are the principal curvatures of the auto-correlation function: the pixels for which they are locally maximum are declared as interest points. It has been shown in Schmid et al. (1998) that interest points are more stable when the derivatives are computed by convolving the image with Gaussian derivatives:

$$\begin{aligned} I_u(\mathbf{x}, \sigma) &= G_u(\mathbf{x}, \sigma) \otimes I(\mathbf{x}) \\ I_v(\mathbf{x}, \sigma) &= G_v(\mathbf{x}, \sigma) \otimes I(\mathbf{x}) \end{aligned}$$

where $G_u(., \sigma)$, $G_v(., \sigma)$ are the first order derivatives of the Gaussian kernel of standard deviation σ along the u and v directions. The auto-correlation matrix is then denoted $M(\mathbf{x}, \sigma, \tilde{\sigma})$. Note that to maintain the derivatives stable with respect to the image scale change s, the Gaussian functions can be normalized with respect to s—the auto-correlation matrix is then $M(\mathbf{x}, \sigma, \tilde{\sigma}, s)$ (Dufournaud et al., 2004).

Point Similarity. If the geometric transformation $\mathcal{T}$ between two images $\mathcal{I}$ and $\mathcal{I}'$ is strictly equal to a scale change s and rotation change θ, the following equality is satisfied for two matching points $(\mathbf{x}, \mathbf{x}')$ in the images:

$$\begin{pmatrix} I_u(\mathbf{x}, \sigma, \theta) \\ I_v(\mathbf{x}, \sigma, \theta) \end{pmatrix} = R(\theta) \begin{pmatrix} I_u(\mathbf{x}, \sigma) \\ I_v(\mathbf{x}, \sigma) \end{pmatrix} = \begin{pmatrix} I'_{u'}(\mathbf{x}', s\sigma) \\ I'_{v'}(\mathbf{x}', s\sigma) \end{pmatrix}$$

where $R(\theta)$ is the rotation and $I_u(\mathbf{x}, \sigma, \theta)$ and $I_v(\mathbf{x}, \sigma, \theta)$ are the steered Gaussian derivatives of the image in the direction θ (Freeman and Adelson, 1991). As a consequence, we can write:

$$R(\theta)M(\mathbf{x}, \sigma, \tilde{\sigma})R(\theta)^T = M(\mathbf{x}', \sigma, \tilde{\sigma}, s)$$

Since

$$M(\mathbf{x}, \sigma, \tilde{\sigma}) = U \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} U^T$$

and

$$M(\mathbf{x}', \sigma, \tilde{\sigma}, s) = U' \begin{pmatrix} \lambda'_1 & 0 \\ 0 & \lambda'_2 \end{pmatrix} U'^T$$

where the columns of U and U' are the eigenvectors. The principal curvatures of the two matched points are therefore equal: $\lambda_1 = \lambda'_1$ and $\lambda_2 = \lambda'_2$.

For two matching points in two images of real 3D scenes, this equality is of course not strictly verified, because of signal noise, and especially because the true transformation of the image is seldom strictly equal to a rotation and scale change. We define the *point similarity* $\mathcal{S}_p$ between two interest points on the basis of their eigenvalues and their intensity:

$$\mathcal{S}_P(\mathbf{x}, \mathbf{x}') = \frac{\mathcal{S}_{p1}(\mathbf{x}, \mathbf{x}') + \mathcal{S}_{p2}(\mathbf{x}, \mathbf{x}') + \mathcal{S}_{pI}(\mathbf{x}, \mathbf{x}')}{3}$$

where

$$\begin{aligned} \mathcal{S}_{p1}(\mathbf{x}, \mathbf{x}') &= \frac{min(\lambda_1, \lambda'_1)}{max(\lambda_1, \lambda'_1)}, \\ \mathcal{S}_{p2}(\mathbf{x}, \mathbf{x}') &= \frac{min(\lambda_2, \lambda'_2)}{max(\lambda_2, \lambda'_2)}, \\ \text{and } \mathcal{S}_{pI}(\mathbf{x}, \mathbf{x}') &= \frac{min(I(\mathbf{x}), I'(\mathbf{x}'))}{max(I(\mathbf{x}), I'(\mathbf{x}'))} \end{aligned}$$

The maximum similarity is 1.0. Statistics show that the evolution of $\mathcal{S}_{p1}$ and $\mathcal{S}_{p2}$ for matched points is hardly affected by rotation and scale changes, and is always larger than 0.8 (Fig. 2).

2.2. *Matching Interest Points*

To match interest points, a cross-correlation measure of the signal can be used (Lhuillier and Quan, 2003), but this requires a precise knowledge of the search area. To cope with this, local grayvalue invariants can be used,

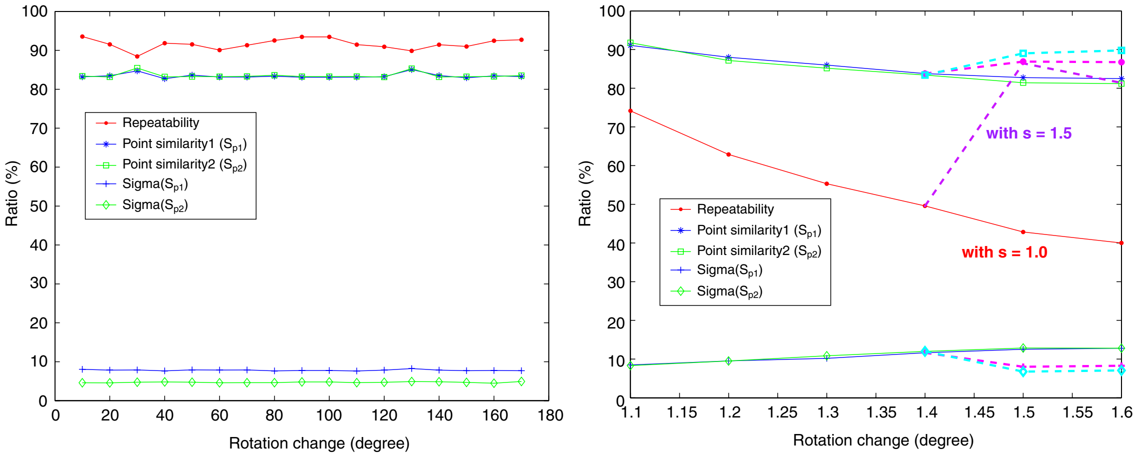


Figure 2. Evolution of the mean and standard deviation of matching points similarity with known rotation (left) and scale changes (right). On the right curve, the solid lines represent the evolution of the similarity when the scale of the detector is set to 1: the similarity decreases when the scale change between the images increases. The dashed lines show the values of the similarity when the scale of the detector is set to 1.5: the similarity is then closer to one when the actual scale change between the images is of the same order.

as in Schmid and Mohr (1997). The approach we propose here imposes a combination of geometric and signal similarity constraints, thus being more robust than approaches solely based on point signal characteristics (a simpler version of this algorithm has been presented in Jung and Lacroix (2001)). It relies on *interest point group* matching: an interest point group is a small set of neighbor interest points, that represent a small region of the image. With groups composed of a small number of interest points, the corresponding region is small enough to ensure that a simple rotation θ approximates fairly well the actual region transformation between the images—the translation being ignored here. The estimate of this rotation is essential in the algorithm, as a group match hypothesis (i.e. a small set of point matches) is assessed on both the signal similarity between interest points and the point matches compliance with the rotation. The matching procedure is a seed-and-grow algorithm initiated by a reliable group match (see Algorithm 1).

Algorithm 1 Overview of the interest point matching algorithm

```
Given two images $\mathcal{I}$ and $\mathcal{I}'$:

  1. Extract the interest points $\{\mathbf{x}\}$ and $\{\mathbf{x}'\}$ in both images
  2. In both images, establish the groups of extracted points. This defines two sets of groups $\mathbf{G} = \{\mathcal{G}\}$ and $\mathbf{G}' = \{\mathcal{G}'\}$, and the neighborhood relations establish a graph between the detected points—this procedure is depicted in Section 2.2.1.
  3. **While** $\mathbf{G} \neq \emptyset$:
     (a) Establish an initial group match $\mathcal{M}(\mathcal{G}_i, \mathcal{G}'_j)$, which defines a rotation $\theta_{i,j}$, and remove $\mathcal{G}_i$ from $\mathbf{G}$—this procedure is depicted in Section 2.2.2.
     (b) Recursive propagation: starting from the neighbors of $\mathcal{G}_i$, explore the neighboring points to find group matches compliant with $\theta_{i,j}$. Remove the new matched groups from $\mathbf{G}$, and iterate until no matches compliant with $\theta_{i,j}$ can be found—this procedure is depicted in Section 2.2.3.
  4. Check the validity of the propagated group matches
```

2.2.1. Grouping Process. The sets of interest points $\{\mathbf{x}\}$, $\{\mathbf{x}'\}$ detected respectively in the two images $\mathcal{I}$, $\mathcal{I}'$ are structured in *local groups*, formed by a pivot point $\mathbf{g}_0$ and its n closest neighbors $\{\mathbf{g}_1, \ldots, \mathbf{g}_n\}$ (Fig. 3). To ensure that the image region covered by the points of a group is small enough, n is rather small (e.g. we use $n = 5$). The groups are generated by studying the neighborhood

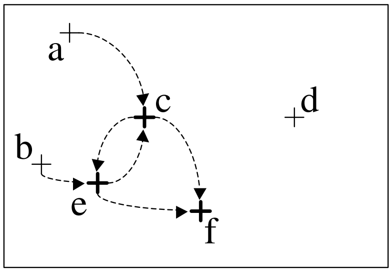


Figure 3. Illustration of the point grouping procedure, with $n = 2$ for readability purposes. Groups have not been generated around points **a** and **b** as they are too close to the image border, and neither around **d** as no neighbor is close enough. Three groups have been generated, with points **c**, **e** and **f** as a pivot. **b** → **e** means "**b** is a neighbor of **e**", which defines a graph relation between the points.

of each point following a spiral pattern: the grouping process is stopped if the spiral meets the image border before n neighbors are found. Also, a maximum threshold on the distance between the neighbor points and the pivot is applied, to avoid the formation of groups that cover a too large image region (in low textured areas for instance, where there are scarce interest points). This implies that a few points do not belong to any group: their matching is processed individually (see Section 2.2.3).

After the grouping process, we end up with two group sets $\mathbf{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_N\}$ and $\mathbf{G}' = \{\mathcal{G}'_1, \dots, \mathcal{G}'_M\}$, $\mathcal{G}_i$ denoting the local group generated with the point $\mathbf{x}_i$ as a pivot:

$$\mathcal{G}_i = \{\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_n\},\ \mathbf{g}_0 = \mathbf{x}_i$$

The neighbors $\{\mathbf{g}_1, \dots, \mathbf{g}_n\}$ are ordered by their distance to the pivot:

$$\parallel \mathbf{v}_1 \parallel < \cdots < \parallel \mathbf{v}_n \parallel$$

where the vectors $\mathbf{v}_i$ are defined as $\mathbf{v}_i = \mathbf{g}_i - \mathbf{g}_0$ and $\parallel . \parallel$ is the norm operator. For each neighbor of the group, we also compute its angle, defined as:

$$\theta_{\mathbf{g}_p} = tan^{-1}(\mathbf{v}_p \cdot \mathbf{v}, \mathbf{v}_p \cdot \mathbf{u})$$

where $\mathbf{u}$ and $\mathbf{v}$ are the image axes.

2.2.2. Group Matching. The procedure to establish a group match is essential in our approach: in particular, a wrong initial group match would cause the algorithm to fail. The procedure consists in three steps depicted in the next paragraphs:

1. Given a group $\mathcal{G}_i$ in $\mathcal{I}$ with $\mathbf{g}_0$ as a pivot, all the groups $\mathcal{G}'_j$ in $\mathcal{I}'$ whose pivot $\mathbf{g}'_0$ is *similar* to $\mathbf{g}_0$ are candidate group matches.
2. For each candidate group match $\mathcal{G}'_j$, determine all the *group match hypotheses* $H(\mathcal{G}_i, \mathcal{G}'_j)$ on the basis of the possible individual neighbor matches, and select the best one $H^*(\mathcal{G}_i, \mathcal{G}'_j)$.
3. Select the best group match among all the $H^*(\mathcal{G}_i, \mathcal{G}'_j)$, and apply a validation criteria.

Point Similarity. Two points $\mathbf{x}$, $\mathbf{x}'$ are defined as *similar* if their similarity measure is above a threshold $T_{\mathcal{S}_p}$:

$$\mathcal{S}_P(\mathbf{x}, \mathbf{x}') > T_{\mathcal{S}_P}$$

This test is used to asses the similarity of points in steps 1 and 2 of the group matching procedure.

Building Group Match Hypotheses. Given two groups $(\mathcal{G}_i, \mathcal{G}'_j)$ whose pivots have passed the point similarity test, one must evaluate all the possible associated group match hypotheses, i.e. the various combinations of matches between the neighbor points of the groups. A group match hypothesis $H(\mathcal{G}_i, \mathcal{G}'_j)$ is defined as:

- a rotation θ
- a set $M(\mathcal{G}_i, \mathcal{G}'_j)$ of interest point matches which respect the rotation θ and whose similarity score is above the threshold $T_{\mathcal{S}_P}$:

$$M(\mathcal{G}_i, \mathcal{G}'_j) = \{(\mathbf{g}_p, \mathbf{g}'_q) \in \mathcal{G}_i \times \mathcal{G}'_j \mid \mathcal{S}_P(\mathbf{g}_p, \mathbf{g}'_q) > T_{\mathcal{S}_P} \text{ and } |\theta_{\mathbf{g}_p} - \theta_{\mathbf{g}'_q}| < T_\theta\} \cup \{(\mathbf{g}_0, \mathbf{g}'_0)\}$$

- a group hypothesis similarity score $\mathcal{S}_G$, defined as the sum of the similarity of the corresponding matched interest points:

$$\mathcal{S}_G(H(\mathcal{G}_i, \mathcal{G}'_j)) = \sum_{(\mathbf{g}_p, \mathbf{g}'_q) \in M(\mathcal{G}_i, \mathcal{G}'_j)} \mathcal{S}_P(\mathbf{g}_p, \mathbf{g}'_q)$$

The best group match hypothesis among all the ones that can be defined on the basis of two candidate groups $(\mathcal{G}_i, \mathcal{G}'_j)$ is determined according to Algorithm 2: this provides the best group match hypothesis H^*, if it exists, between $\mathcal{G}_i$ and $\mathcal{G}'_j$. Note in this procedure the introduction of a threshold $\Delta_{\mathcal{S}_G}$ in the comparison of hypotheses, to ensure that the best hypotheses has a much better score than the second best: this is useful to avoid wrong group matches for images with repetitive patterns, in which many points are similar.

Algorithm 2 Determination of the best match hypothesis for two groups

```
• Init: $\mathcal{S}_G^* = 0$
• For $p = 1$ to $|\mathcal{G}_i|$, For $q = 1$ to $|\mathcal{G}'_j|$:
• if $\mathcal{S}_P(\mathbf{g}_p, \mathbf{g}'_q) > T_{\mathcal{S}_P}$ then create and evaluate a group match hypothesis $H_{p,q}(\mathcal{G}_i, \mathcal{G}'_j)$:
    – Set $M(\mathcal{G}_i, \mathcal{G}'_j) = \{(\mathbf{g}_0, \mathbf{g}'_0), (\mathbf{g}_p, \mathbf{g}'_q)\}$. This defines the rotation $\theta$ for this hypothesis: $\theta = \theta_{\mathbf{g}_p} - \theta_{\mathbf{g}'_q}$
    – complete $M(\mathcal{G}_i, \mathcal{G}'_j)$ with the other points in $\mathcal{G}_i$ that are similar to points in $\mathcal{G}'_j$, such that:
        $\forall s > p$ and $t > q$, $\mathcal{S}_P(\mathbf{g}_s, \mathbf{g}'_t) < T_{\mathcal{S}_P}$
        and $|\theta - (\theta_{\mathbf{g}_s} - \theta_{\mathbf{g}'_t})| < T_\theta$
      Note here that the fact that the neighbor are ordered by their distance to the pivot reduces the search for additional point matches—see Fig. 4.
    – Evaluate the hypothesis $H_{p,q}(\mathcal{G}_i, \mathcal{G}'_j)$:
        if $\mathcal{S}_G(H_{p,q}(\mathcal{G}_i, \mathcal{G}'_j)) > \mathcal{S}_G^* + \Delta_{\mathcal{S}_G}$,
        then $H^* = H_{p,q}(\mathcal{G}_i, \mathcal{G}'_j)$ and $\mathcal{S}_G^* = \mathcal{S}_G(H^*)$.
```

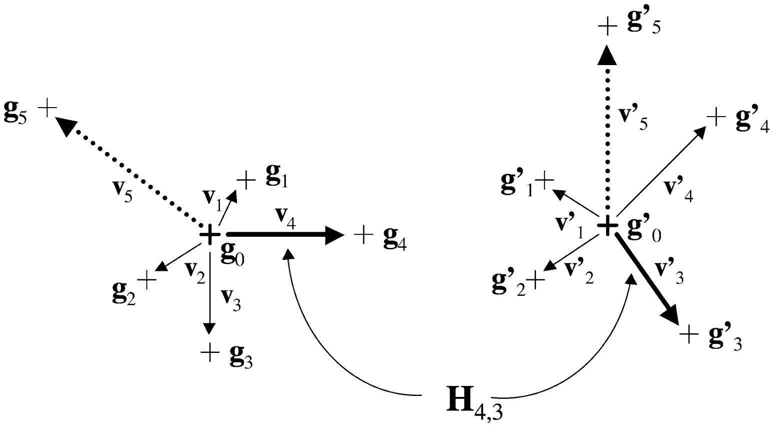


Figure 4. Completion of a group match hypothesis. Given the hypothesis $H_{4,3}$ defined by the point matches $(\mathbf{g}_0, \mathbf{g}'_0)$ and $(\mathbf{g}_4, \mathbf{g}'_3)$, the best potential match for $\mathbf{g}_5$ is determined by evaluating geometric and point similarity constraints. The indexes of the neighbors being ordered according to their distance to the pivot, only the matches $(\mathbf{g}_5, \mathbf{g}'_4)$, and $(\mathbf{g}_5, \mathbf{g}'_5)$ are evaluated—on this example, $(\mathbf{g}_5, \mathbf{g}'_5)$ is the sole valid match.

Selection of the Best Group Match Hypothesis. Now that we have determined the best group match hypothesis H^* for each candidate group match $\mathcal{G}'_j$ for the group $\mathcal{G}_i$, one must determine the one that actually corresponds to a true group match. This is simply done by comparing their similarity $\mathcal{S}_G$, applying the same threshold $\Delta_{\mathcal{S}_G}$ as above to make sure the best match is not ambiguous.

Finally, the validity of the found group match is confirmed by evaluating the zero-mean normalized correlation score (ZNCC) between windows centered on the pivots $(\mathbf{g}_0, \mathbf{g}'_0)$ of the groups. This score can be computed thanks to the knowledge of the rotation θ defined by the group match hypothesis, which is applied to the pixels of the window centered on $\mathbf{g}'_0$.

2.2.3. Finding Further Matches

Propagation Process. Once a reliable group match hypothesis is established, a propagation process searches for new matches. The principle of the propagation is to exploit the graph defined by the grouping process and the estimated rotation associated to the current hypothesis: additional point matches consistent with the current rotation estimate are searched in the neighborhood of the current group match. This process is depicted in Algorithm 3 .

Algorithm 3 Propagation process

```
Given a group match $(\mathcal{G}_i, \mathcal{G}'_j)$ and the associated rotation $\theta$:
• Init: set $M^{propage} = M(\mathcal{G}_i, \mathcal{G}'_j) \setminus \{(\mathbf{g}_0, \mathbf{g}'_0)\}$.
• While $M^{propage} \neq \emptyset$:
    – Select a point match $(\mathbf{g}_p, \mathbf{g}'_q) \in M^{propage}$. $\mathbf{g}_p$ and $\mathbf{g}'_q$ are respectively the pivots of the groups $\mathcal{G}_p$ and $\mathcal{G}'_q$.
    – For $s = 1$ to $|\mathcal{G}_p|$, For $t = 1$ to $|\mathcal{G}'_q|$:
      if $\mathcal{S}_P(\mathbf{g}_s, \mathbf{g}'_t) < T_{\mathcal{S}_P}$ and $|\theta - (\theta_{\mathbf{g}_s} - \theta_{\mathbf{g}'_t})| < T_\theta$,
      add $(\mathbf{g}_s, \mathbf{g}'_t)$ to $M^{propage}$
    – Remove $(\mathbf{g}_p, \mathbf{g}'_q)$ from $M^{propage}$
```

During the propagation, the translation between matched points is computed: when the propagation ends, this allow to focus the search for new matches, as illustrated in Fig. 5.

Propagation Monitoring. Repetitive patterns with a size similar to the group size can lead to false matches, although the initial group match has passed the tests described in Section 2.2.2. The occurrence of such cases can be detected by checking whether the propagation process succeeds or not around the first group match: if it fails, it is very likely that the initial group match hypothesis is a wrong one, and it is then discarded (Fig. 6). Note that this test also eliminates group matches if a group is isolated or if the overlap between the two images $\mathcal{I}$ and $\mathcal{I}'$ is restricted to the size of the group: these are degenerated cases in which the algorithm does not match the groups.

Non-Grouped Points Matching. As mentioned in Section 2.2.1, some points are not associated to groups after the grouping procedure, mainly near the image borders. Once the propagation procedure is achieved, for each non grouped point $\mathbf{x}_b$ of $\mathcal{I}$, matches are searched among the set of points X_b in the image $\mathcal{I}'$:

$$\mathbf{X}_c = \{\mathbf{x} | \mathbf{x} \in W(\hat{\mathbf{x}}'_b)\}$$

where $\hat{\mathbf{x}}'_b$ is the estimated position of $\mathbf{x}_b$ in $\mathcal{I}'$ provided by the application of the transformation defined by the mean of the rotations and translations estimated so far, and $W(\hat{\mathbf{x}}'_b)$ is a search window centered on $\hat{\mathbf{x}}'_b$. The points

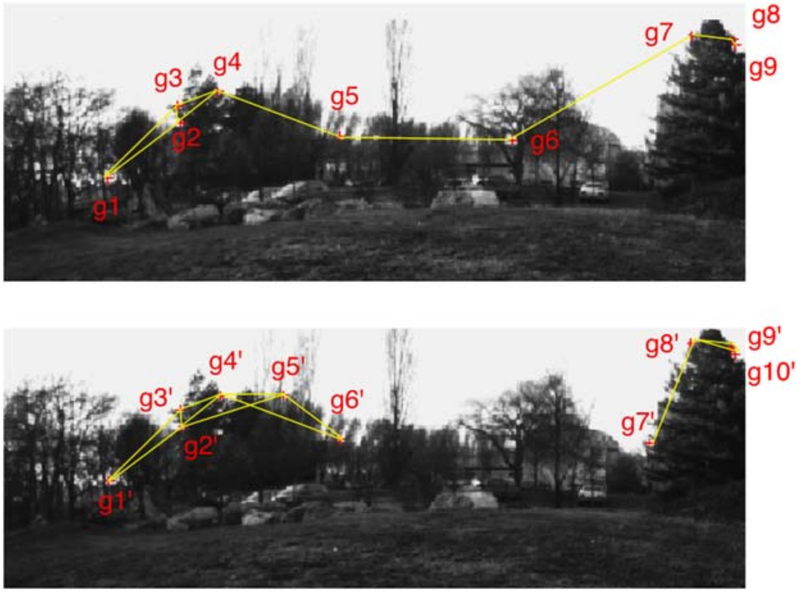


Figure 5. Illustration of the propagation process. Red crosses are interest points, yellow lines indicate neighborhood relations defined by the grouping process. Here, $\mathbf{g}_2$ and $\mathbf{g}'_2$ are the pivots of the initial group hypothesis $H(\mathcal{G}_2, \mathcal{G}'_2)$, and the corresponding list of individual points matches is $M(\mathcal{G}_2, \mathcal{G}'_2) = \{(\mathbf{g}_2, \mathbf{g}'_2), (\mathbf{g}_1, \mathbf{g}'_1), (\mathbf{g}_3, \mathbf{g}'_3), (\mathbf{g}_4, \mathbf{g}'_4)\}$. During the propagation, matches for points neighboring the ones of $M(\mathcal{G}_2, \mathcal{G}'_2)$ are evaluated—here the match $(\mathbf{g}_5, \mathbf{g}'_6)$ is added and the propagation stops. Thanks to the estimate of the translation between the points matched so far, the group match hypothesis $H(\mathcal{G}_7, \mathcal{G}'_8)$ can be evaluated, and new matches are added for a little computational cost.

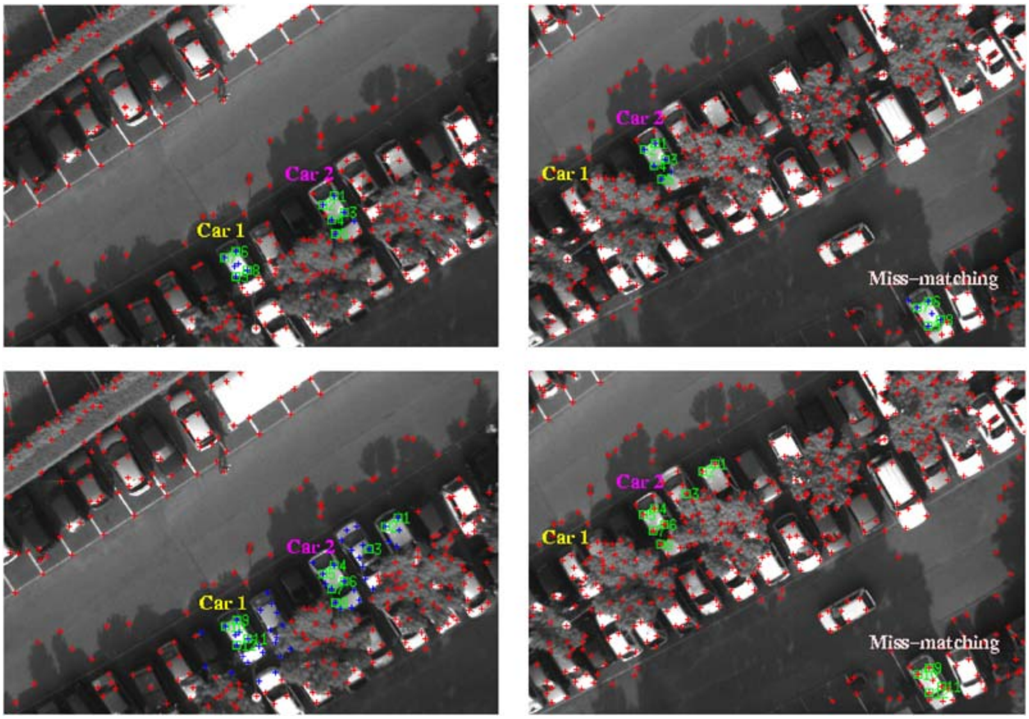


Figure 6. Illustration of the propagation monitoring. The top images show two group matches independently established according to the process of Section 2.2.2: the "Car 2" group is properly matched, whereas "Car 1" has been incorrectly matched. The bottom images show the additional matches established by the propagation process: no additional matches have been determined around the "Car 1" group, whereas other matches around the "Car 2" have been determined: the "Car 1" group match is a false one.

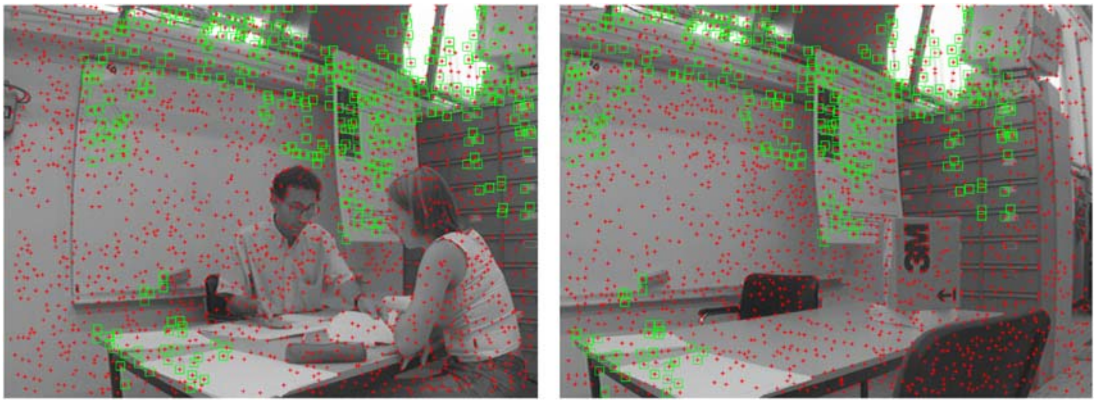


Figure 7. Points matched with a small viewpoint change and significant scene modifications (throughout the paper, red crosses shows the interest points, and green squares indicate successful matches). 349 matches are found in 200 ms (115 ms for the points detection, 85 ms for the matching process.

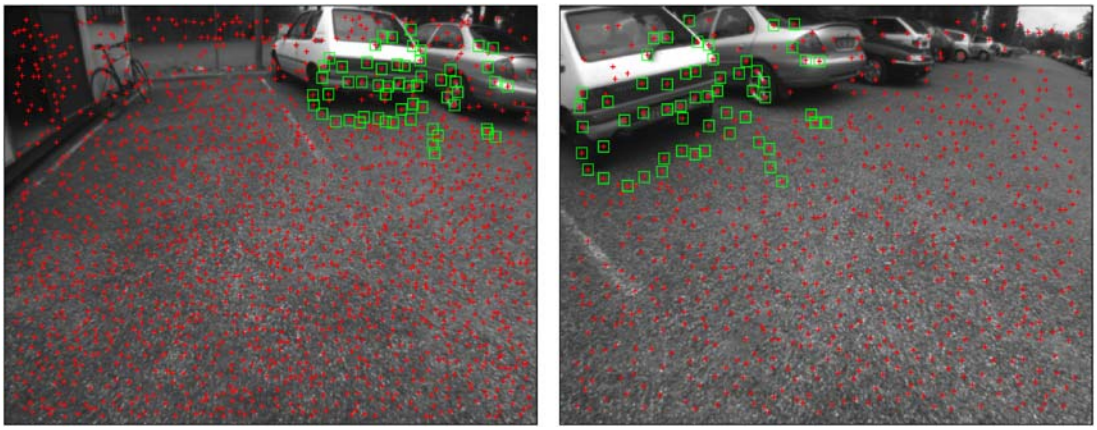


Figure 8. Points matched with a significant viewpoint change, that induces a 1.5 scale change. 57 points are matched in 80 ms.

comprised in $W(\hat{\mathbf{x}}'_b)$ are evaluated according to the hypothesis pruning process presented in Section 2.2.2: test on the point similarity measure $\mathcal{S}_P$ and verification with the computation of the ZNCC coefficient.

2.3. *Results*

The algorithm provides numerous good matches while keeping the number of outliers very small, in different kinds of scenes and in a wide variety of conditions, tolerating noticeable scene modifications and viewpoint changes. Figures 7 to 9 present matches obtained in various conditions, with the computational time required—the processed image size is 512 × 384, and time measures have been obtained on 3.2 GHz Pentium IV). The algorithm does not explore various scale changes: when a scale change greater than half a unity occurs, it must be provided as a parameter to the interest point detection routine. This is a limitation as compared to scale invariant point features, but a coarse knowledge of the scale change is sufficient: in a SLAM context, such an estimate is readily obtained.

Table 1 lists the parameters required by the matching algorithm and their values. These values are used for all the results presented throughout the paper, and during our everyday use of the algorithm: no parameter tuning is required.

Three parameters are used during the grouping process presented Section 2.2.1: *Minimum group size*, *Maximum group size* and *Maximum distance between pivot and group member* (the size parameters do not include

Table 1. Thresholds and parameters of the matching algorithm.

Maximum group size	5
Minimum group size	2
Size of correlation window for ZNCC	9 × 9
Threshold $T_{\mathcal{S}_P}$ on point similarity $\mathcal{S}_P$	0.7
Threshold on ZNCC T_{ZNCC}	0.6
Threshold $\Delta_{\mathcal{S}_G}$ to discriminate group hypotheses	0.1
Threshold on rotation change, T_θ	$0.2rad$

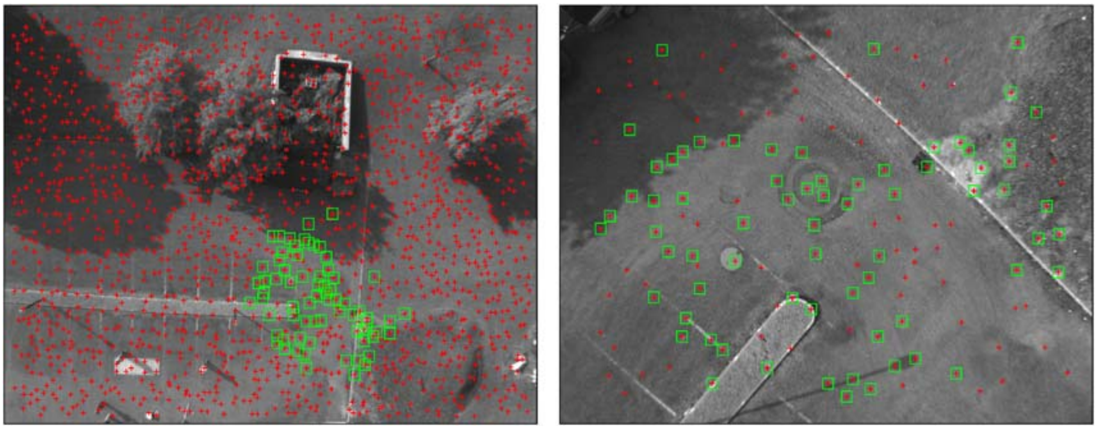


Figure 9. Points matched with a scale change of 3. 70 points are matched in 270 ms.

the pivot). The *Minimum group size* is naturally set to 2, the minimum size that allows to run the group matches determination presented Section 2.2.2. The *Maximum group size* is a compromise: on the one hand, the more members in a group, the more reliable are the group match hypotheses. On the other hand, a big number of points in a group tends to violate the hypothesis that a simple rotation approximates its transformation between the two images: empirical tests show that a value of 5 offers a good balance. Finally, the *Maximum distance between pivot and group member* threshold is set to $3\sqrt{\mathcal{D}}$, where $\mathcal{D}$ is the density of interest points in the image.

The threshold T_{S_P} on the similarity measure is used to evaluate if two points match: its value is set to 0.7, according to the variations of the point similarities presented in Section 2.1. The threshold T_{ZNCC} on the correlation score to confirm a point match is set to 0.6, a value smaller than usually used for this score (e.g. in dense stereovision): this is due to the fact that a rotation is imposed to one of the correlation window before computing the score, which smooths the signal in the window, and also to the fact that we aim at matching points seen from different viewpoints. Finally, the threshold on rotation change T_θ is set to $0.2rad$, a quite large value that is necessary to cope with the errors on the interest points detection, that can reach at most 1.5 pixel (Schmid et al., 1998).

3. Vision-Based SLAM

The vision algorithms of Section 2 provide a solution for two of the four basic SLAM functionalities introduced in Section 1: the observed features are the interest points, and data association is performed by the interest point matching process.

There are however two important cases to distinguish regarding the observation function, depending on whether the robot is endowed with stereovision or not:

- With stereovision, the 3D coordinates of the features with respect to the robot are simply provided by matching points in the stereoscopic image pair,
- But if the robot is endowed with a single camera, only the bearings of the features are observed: this requires a dedicated landmark initialization procedure, that integrates several observations over time.

This section presents how we set up a SLAM solution in both cases, focusing on the monocular case, which raises specific difficulties. Note that regarding the overall SLAM estimation process, vision does not raise any particular problem—we use a classical implementation of the Extended Kalman Filter in our developments.

3.1. SLAM with Stereovision

A vast majority of existing SLAM approaches rely on data that directly convey the landmark 3D state (e.g. using laser range finders or millimeter wave radars). Stereovision falls in this category, but it is only recently that this sensor has been used to develop SLAM approaches. In Se et al. (2002), an approach that uses scale invariant features (SIFT) is depicted, with results obtained by a robot evolving in 2D in a $10 \times 10\,\mathrm{m}^2$ laboratory environment, in which about 3500 landmarks are mapped in 3D. In Kim and Sukkarieh (2003), rectangular patches are spread on a horizontal ground and used as landmarks. This is not strictly speaking a stereovision approach, but the distance to the landmarks is readily observable as the landmarks have a known size. Our first contribution to this problem has been published in Jung and Lacroix (2003), with some preliminary results obtained on a stereovision bench mounted on a blimp.

With a stereovision bench, the state of the observed features can readily be estimated from a single observation: a feature (interest point) is transformed into a landmark (3D point) thanks to the matching of the feature in the two images provided by the stereoscopic bench. A SLAM solution can then be readily developed using the EKF scheme, in which the state X of the filter is composed of the 3D position parameters of the stereovision bench (or the robot) $X_r = [x_r, y_r, z_r, yaw_r, pitch_r, roll_r]$, and of a set of landmark 3D coordinates $X_f^k = [x_k, y_k, z_k]$:

$$X = (X_r, X_f^1 \cdots X_f^k \cdots)^T$$

The associated state covariance has the following form:

$$P = \begin{pmatrix} P_{X_r} & P_{X_r, X_f^1} & \cdots & P_{X_r, X_f^k} & \cdots \\ P_{X_f^1, X_r} & P_{X_f^1, X_f^1} & \cdots & P_{X_f^1, X_f^k} & \cdots \\ \vdots & \ddots & \vdots & \vdots & \cdots \\ P_{X_f^k, X_r} & P_{X_f^k, X_f^1} & \cdots & P_{X_r^k, X_f^k} & \cdots \\ \vdots & \cdots & \vdots & \vdots & \ddots \end{pmatrix}$$

where P_{X_i} refers to covariances of sub-state X_i and P_{X_i, X_j} refers to cross covariance of sub-states X_i and X_j. Thanks to the interest point detection and matching processes, the usual state prediction and state update processes can readily be applied (the equations of the extended Kalman filter are not recalled here, as they can be found in numerous articles in the literature).

3.2. Bearings-Only SLAM

3.2.1. Related Work. The bearings-only SLAM problem is an instance of the more general *partially observable* SLAM, in which the sensor does not give enough information to compute the full state of a landmark from a single observation. Using sonar sensors for example, raises the problem of *range-only* SLAM. A solution to this problem has been proposed in Leonard et al. (2002): since a single observation is not enough to estimate a feature, multiple observations are combined from multiple poses.

Several contributions propose different solutions for *delayed* initial state estimation in bearings-only SLAM. In Bailey (2003), an estimation is computed using observations from two robot poses, and is determined to be Gaussian using the Kullback distance. The complexity of the sampling method proposed to evaluate this distance is quite high. In Deans and Hebert (2000), a combination of a Bundle Adjustment for feature initialization and a Kalman filter is proposed. The complexity of the initialization step is greater than a Kalman filter but theoretically gives more optimal results. A method based on a particle filter to represent the initial depth of a feature is proposed in Davison (2003) and Davison et al. (2004). However its application in large environments is not straightforward, as the required number of particles is linear with the initialization range. In Lemaire et al. (2005) the initial PDF of a feature is approximated by a sum of Gaussians, bad members are pruned until only a single Gaussian remains, that is then simply added to the Kalman stochastic map.

A first *un-delayed* feature initialization method was proposed in Kwok and Dissanayake (2004). The initial state is approximated with a sum of Gaussians and is explicitly added to the state of the Kalman filter. The sum of Gaussians is not described and the convergence of the filter when updating a multi-Gaussian feature is not proved. This algorithm has been recently extended in Kwok et al. (2005) using Gaussian Sum Filter. Also a method based on a Kalman federate filtering technique is described in Solà et al. (2005).

Bearings-only SLAM using vision is also very similar to the well known structure from motion (SFM) problem. Recent work by Nister (2003) show very nice results. The main difference is that robotic applications require an incremental and computationally tractable solution whereas SFM algorithm can run in a time consuming batch process. Links between non linear optimization algorithms and standard Kalman filter for SLAM and bearings-only SLAM are studied in Konolige (2005).

The approach presented here is in the *delayed* category. Figure 10 depicts it: when a new feature is observed, a full Gaussian estimate of its state cannot be computed from the measure, since the bearings-only observation function cannot be inverted. We initialize the representation of this feature with a sum of Gaussians (Section 3.2.3). Then, a process updates this initial state representation, until the feature can be declared as a landmark whose full state is estimated (Section 3.2.4). Once estimated, the landmark is introduced in the stochastic map, which is managed by the usual EKF.

The main characteristics of our approach are the following:

- the initial probability density of a feature is approximated with a particular weighted sum of Gaussians,
- this initial state is expressed in the robot frame, and not in the global map frame, so that it is de-correlated from the stochastic map, until it is declared as a landmark and added to the map,
- many features can enter the initial estimation process at a low computational cost, and the delay can be used to select the best features.

In order to add the landmark to the map, and to compute its state in the map frame along with the correlations in a consistent way, the pose where the robot was when the feature was first seen has to be estimated in the filter. All observations of the feature are also memorized along the corresponding robot pose estimations, so that all available information is added to the filter at initialization.

3.2.2. Structure of the Kalman Filter. The state of the EKF is composed of the landmarks estimates, the current robot pose, and as previously pointed out, some past poses of the robot. For simplicity, let's consider that the k last poses of the robot are kept in the filter state. The Kalman state is:

$$X = \begin{pmatrix} X_r^0 \\ \vdots \\ X_r^k \\ X_f^0 \\ \vdots \end{pmatrix} \quad P = \begin{pmatrix} P_{X_r^0} & \cdots & P_{X_r^0, X_r^k} & P_{X_r^0, X_f^0} & \cdots \\ \vdots & \ddots & \vdots & \vdots & \cdots \\ P_{X_r^k, X_r^0} & \cdots & P_{X_r^k} & P_{X_r^k, X_f^0} & \cdots \\ P_{X_f^0, X_r^0} & \cdots & P_{X_f^0, X_r^k} & P_{X_f^0} & \cdots \\ \vdots & \cdots & \vdots & \vdots & \ddots \end{pmatrix}$$

In our case the prediction step must be conducted with special care since a whole part of the trajectory is estimated. All the poses but the current one are static states, so only the current pose is affected by prediction. *Before* applying the prediction equations, all the past poses are re-numbered, so that the robot trajectory looks like: $X_r = [X_r^0, X_r^2, \ldots, X_r^k, X_r^{k+1}]$. The oldest robot pose X_r^{k+1} is forgotten because we don't want the size of the filter to increase. X_r^{k+1} is used to *back up* the current robot pose and becomes X_r^1 (*ring buffer* mechanism):

$$\begin{gathered} X_r^1 \leftarrow X_r^0 \quad P_{X_r^1} \leftarrow P_{X_r^0} \quad \forall j\, P_{X_r^1, X_f^j} \leftarrow P_{X_r^0, X_f^j} \\ P_{X_r^1, X_r^i} \leftarrow P_{X_r^0, X_r^i}\; i \neq 0\, i \neq 1 \quad P_{X_r^1, X_r^0} \leftarrow P_{X_r^0} \end{gathered}$$

3.2.3. Feature Initialization. From now on we consider 3D point features represented by their Cartesian coordinates $X_f = (x, y, z)$ and the associated bearings-only observation function $z = \mathbf{h}(X_f)$:

$$\begin{pmatrix} \theta \\ \phi \end{pmatrix} = \begin{pmatrix} \arctan(y/x) \\ -\arctan(z/\sqrt{x^2 + y^2}) \end{pmatrix}$$

Actually $z = (\theta, \phi)$ represents the direction of a pixel (u, v) which corresponds to an interest point.

In our notation, the observation model $\mathbf{h}()$, as well as the inverse observation model $\mathbf{g}()$ do not include frame composition with the robot pose, instead these transformations are formalized in **to**() and **from**() functions: **to**(f, v) computes vector v in frame f, and **from**(f, v) computes vector v in frame f^{-1}. This eases the following developments, and is general with respect to the underlying representation of a 3D pose (using Euler angles, quaternions,...). This also makes the implementation more modular, and observation models easier to implement.

In the sensor polar coordinate system (ρ, θ, ϕ), the density probability of the feature state is already jointly Gaussian on (θ, ϕ), since the measure (interest point location estimate) is considered Gaussian. The measure itself does not give any information about the *depth*, but we generally have *a priori* knowledge. For indoor robots, the maximal depth can for instance be bounded to several meters. For outdoor robots the maximal range is theoretically infinity, but in general only the surrounding environment may be of interest for the robot. This gives us for ρ an *a priori* uniform distribution in the range $[\rho_{min}, \rho_{max}]$.

The Kalman filter assumes Gaussian PDF. A Gaussian is represented with only two values. So we choose to approximate this *a priori* knowledge on the depth with a sum of Gaussians:

$$\begin{aligned} p(\theta, \phi, s) &= \Gamma(\theta, \sigma_\theta) \cdot \Gamma(\phi, \sigma_\phi) \cdot p(\rho) \\ &= \Gamma(\theta, \sigma_\theta) \cdot \Gamma(\phi, \sigma_\phi) \cdot \sum_i w_i \Gamma_i(\rho_i, \sigma_{\rho_i}) \end{aligned}$$

Considering the invariant scale of the PDF, the following geometric series for $\sum_i w_i \Gamma_i(\rho_i, \sigma_{\rho_i})$ is proposed:

$$\begin{gathered} \rho_0 = \rho_{min}/(1 - \alpha) \\ \rho_i = \beta^i \cdot \rho_0 \quad \sigma_{\rho_i} = \alpha \cdot \rho_i \quad w_i \propto \rho_i \\ \rho_{n-2} < \rho_{max}/(1 - \alpha) \quad \rho_{n-1} \geq \rho_{max}/(1 - \alpha) \end{gathered}$$

Figure 10 shows a plot of this distribution for typical values of α and β. The constant ratio α between the mean and the variance defines the width of the Gaussians. The rate β of geometric series defines the density of Gaussians to fill in the depth range. α and β are chosen so as to meet the following constraints:

- nearly constant distribution in the range $[\rho_{min}, \rho_{max}]$,
- the covariance of each Gaussian must be compatible with non-linearity of the observation function around the mean of this Gaussian, so that it will be acceptable to update it in the EKF,
- the number of Gaussians should be kept as low as possible for computational efficiency purposes.

In Peach (1995) it is proved that for a scale invariant observation model the choice of α guarantees the linearization to be acceptable by the EKF, a discussion about the value of α is also given. A value of about 0.25 is empirically good. The same ratio can also be found in Davison (2003).

Each Gaussian $\{\mu_i^p = (\rho_i, \theta, \phi), \Sigma_i^p = (\sigma_{\rho_i}^2, \sigma_\theta^2, \sigma_\phi^2)\}$ is then converted to $\{\mu_i^c, \sigma_i^c)\}$ in Cartesian coordinates in the current robot frame, which is the reference frame for this feature $X_r^{t_f}$ (Fig. 11):

$$\mu_i^c = \mathbf{g}(z) = \begin{pmatrix} \rho_i \cos\phi \cos\theta \\ \rho_i \cos\phi \sin\theta \\ -\rho_i \sin\phi \end{pmatrix} \qquad \Sigma_i^c = G \Sigma_i^p G^T$$

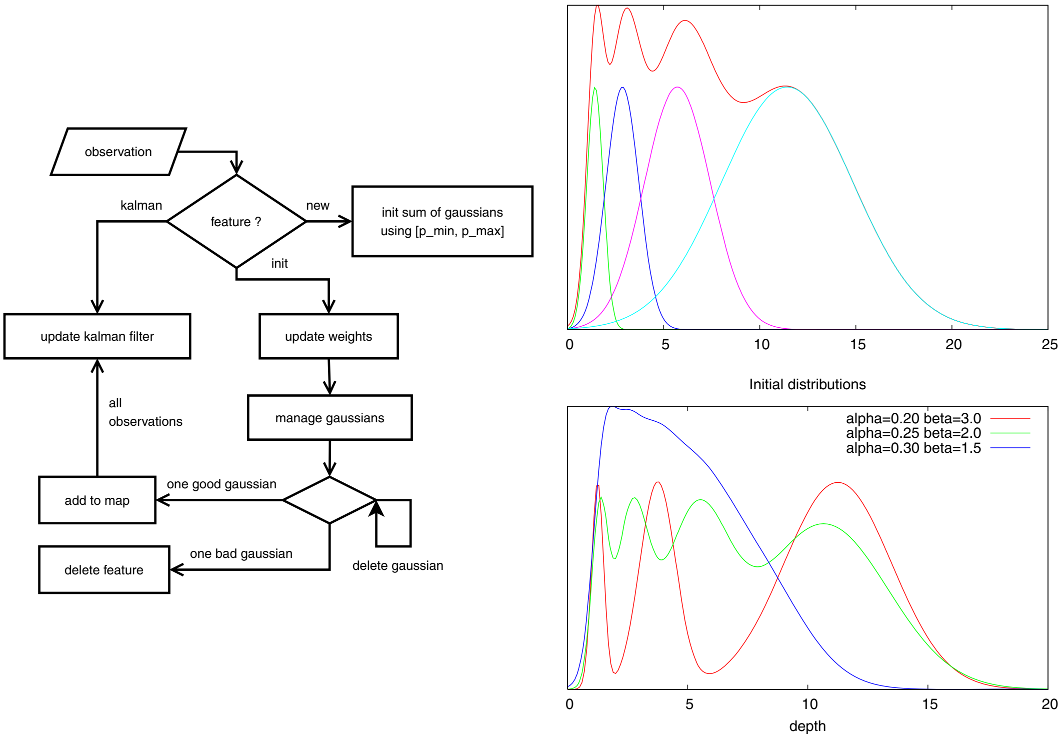


Figure 10. Left: Our approach to the bearings-only SLAM problem. Right-top: Gaussian sum approximating the initial distribution over depth. Right-bottom: different initial distributions.

where $G = \partial\mathbf{g}/\partial z|_{(\rho_i,\theta,\phi)}$. Since we do not project this distribution in the map frame, the distribution is for now kept uncorrelated with the current map. As a consequence the sum of Gaussians is not added to the state of the Kalman filter and initialization is done at a low computational cost.

3.2.4. Initial State Update. The rest of the initialization step consists in choosing the Gaussian which best approximates the feature pose—the feature being thrown away if no consistent Gaussian is found. This process is illustrated in Fig. 11.

Subsequent observations are used to compute the likelihood of each Gaussian i. At time t, given observation z_t with covariance R_t, the likelihood of Γ_i to be an estimation of the observed feature is:

$$L_i^t = \frac{1}{2\pi\sqrt{|S_i|}} \exp\left(-\frac{1}{2}(z_t - \hat{z}_i)^T S_i^{-1}(z_t - \hat{z}_i)\right)$$

where S_i is the covariance of the innovation $z_t - \hat{z}_i$. And the normalized likelihood for the hypothesis i is the product of likelihoods obtained for Γ_i:

$$\Lambda_i = \frac{\prod_t L_i^t}{\sum_j \prod_t L_j^t}$$

The prediction of the observation $\hat{z}_i$ must be done considering each Gaussian in the robot frame. For clarity, let $\mathcal{H}()$ be the full observation function. We have:

$$\begin{aligned}
\hat{z}_i &= \mathbf{h}\big(\mathbf{to}\big(\hat{X}_r^0, \mathbf{from}\big(\hat{X}_r^{t_f}, \mu_i^c\big)\big)\big)\\
&= \mathcal{H}\big(\hat{X}_r^0, \hat{X}_r^{t_f}, \mu_i^c\big)\\
S_i &= H_1 P_{X_r^0} H_1^T + H_2 P_{X_r^{t_f}} H_2^T\\
&\quad + H_1 P_{X_r^0, X_r^{t_f}} H_2^T + H_2 P^T_{X_r^0, X_r^{t_f}} H_1^T\\
&\quad + H_3 \Sigma_i^c H_3^T + R_t
\end{aligned}$$

where $H_1 = \partial\mathcal{H}/\partial X_r^0|_{\hat{X}_r^0, \hat{X}_r^{t_f}, \mu_i^c}$ $H_2 = \partial\mathcal{H}/\partial X_r^{t_f}|_{\hat{X}_r^0, \hat{X}_r^{t_f}, \mu_i^c}$ and $H_3 = \partial\mathcal{H}/\partial\mu_i^c|_{\hat{X}_r^0, \hat{X}_r^{t_f}, \mu_i^c}$

Then we can select the bad hypotheses and prune the associated Gaussian. Bad hypotheses are those whose likelihood Λ_i is low. When observing the evolution of the likelihoods Λ_i computed with simulated or with real data, we see that the likelihood of a hypothesis which is getting unlikely dramatically drops. The likelihood of n equally likely hypotheses is $1/n$: we take $1/n$ as a reference value, and simply prune a hypothesis if its likelihood is under a certain threshold τ/n.

When only a single Gaussian remains, the feature is a candidate for addition to the map. We check that this Gaussian is consistent with the last measure using the

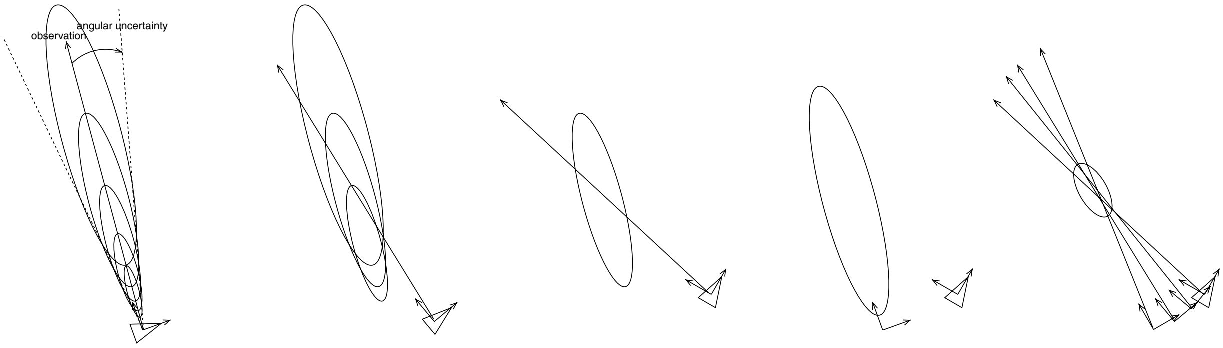


Figure 11. From an observed feature in the images to a landmark in the map. From left to right: the sum of Gaussians is initialized in the robot frame; some Gaussians are pruned based on their likelihood after additional observations of the feature; when a single hypothesis remains, the feature is declared as a landmark and it is projected into the map frame; and finally past observations are used to update the landmark estimate.

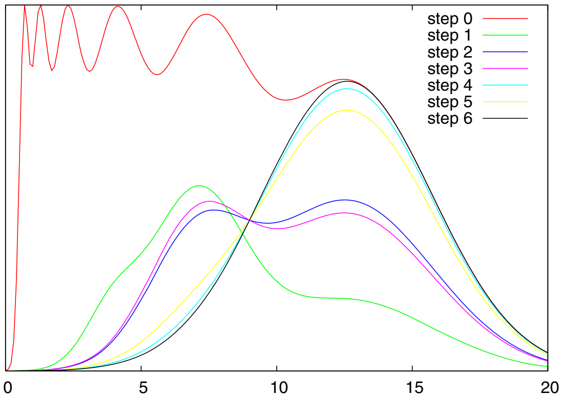


Figure 12. Evolution of the weighted sum of Gaussians through initialization steps.

χ^2 test. Such a convergence is plotted step by step in Fig. 12. If the test does not pass, it means that our *a priori* distribution did not include the feature, in other words that the feature is not in the range $[\rho_{min}, \rho_{max}]$: in this case the feature is rejected.

3.2.5. Landmark Initialization. When a Gaussian $\Gamma_i(\mu_i^c, \Sigma_i^c)$ is chosen, the corresponding feature j is declared as a landmark, and is added to the stochastic map:

$$X^+ = \begin{pmatrix} X^- \\ X_f^j \end{pmatrix} \quad P^+ = \begin{pmatrix} P^- & P_{X^-, X_f^j} \\ P_{X_f^j, X^-} & P_{X_f^j} \end{pmatrix}$$

$$\hat{X}_f^j = \mathbf{from}\left(\hat{X}_r^{t_f^j}, \mu_i^c\right)$$

$$P_{X_f^j} = F_1 P_{X_r^{t_f^j}} F_1^T + F_2 \Sigma_i^c F_2^T$$

$$P_{X_f^j, X^-} = F_2 P^-$$

where $F_1 = \partial\ \mathbf{from}/\partial f|_{\hat{X}_r^{t_f^j}, \mu_i^c}$ and $F_2 = \partial \mathbf{from}/\partial v|_{\hat{X}_r^{t_f^j}, \mu_i^c}$

Remember that for all steps since the feature was first seen, we kept the feature observations, and the corresponding poses of the robot have been estimated by the filter. Up to now the observations were used only to compute the likelihood of the hypotheses, but we can now use this information to *update* the filter state. In our algorithm, all available information in the initial step is added to the stochastic map just after the feature is added as a landmark.

4. Experiments

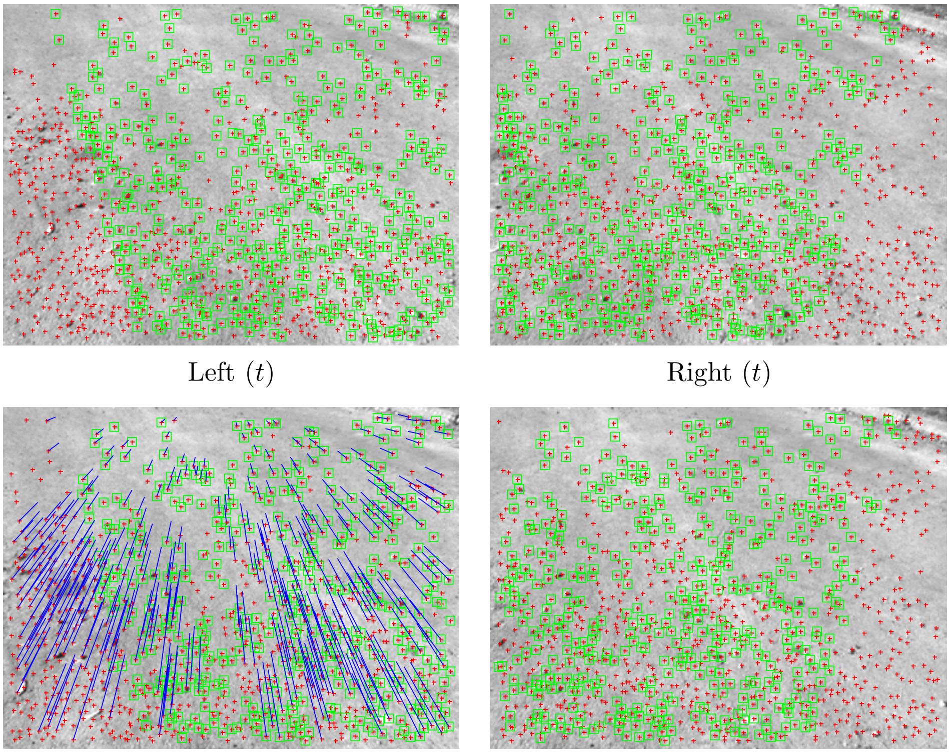
Besides the essential issues discussed in the former sections, the implementation of a complete SLAM solution calls for additional developments: the prediction function that estimates the relative robot motions between consecutive observations, an error model on this prediction and an error model of the feature observations are required. Also, an active way to select the features to be mapped in the filter state helps to control the overall system evolution. Finally, the availability of a ground truth is necessary to quantitatively assess the precision of the estimates.

These issues are presented in the following section for both the stereovision and bearings-only cases, and Sections 4.2 and 4.3 present results obtained with a ground rover and an airship.

4.1. SLAM Setup

4.1.1. Robot Motion Prediction.

3D Odometry. With a robot equipped with an inertial measurement unit, an estimate of the 3D elementary motions $\mathbf{u}(k+1)$ can be provided by integrating the odometry data on the plane defined by the pitch and roll angles of the robot. An actual error model on odometry is difficult to establish: since the rover Dala experiences



Left (t)

Right (t)

Left $(t+1)$

Right $(t+1)$

Figure 13. Illustration of the matches used to estimate robot motion with stereovision images, on a parking lot scene. The points matched between the stereo images are surrounded by green rectangles. The robot moved about half a meter forward between t and $t+1$, and the matches obtained between the two left images are represented by blue segments.

important slippage and is equipped with a cheap inertial measurement unit, we defined the following conservative error model:

- The standard deviation on the translation parameters Δt_x, Δt_y, Δt_z is set to 8% of the traveled distance,
- The standard deviation on $\Delta\Phi$ (yaw) is set to 1.0 degree per traveled meter, and to 1.0 degree for each measure on $\Delta\Theta$, $\Delta\Psi$ (pitch and roll).

Visual Motion Estimation (*VME*). With a stereo-vision bench, the motion between two consecutive frames can easily be estimated using the interest point matching algorithm (Mallet et al., 2000; Olson et al., 2000). Indeed, the interest points matched between the image provided by one camera at times t and $t+1$ can also be matched with the points detected in the other image at both times (Fig. 13): this produces a set of 3D point matches between time t and $t+1$, from which an estimate of the 6 displacement parameters can be obtained (we use the least square minimization technique presented in Arun et al. (1987) for that purpose).

The important point here is to get rid of the wrong matches, as they considerably corrupt the minimization result. Since the interest point matching algorithm generates only very scarce false matches, we do not need to use a robust statistic approach, and the outliers are therefore simply eliminated as follows:

1. A 3D transformation is determined by least-square minimization. The mean and standard deviation of the residual errors are computed.
2. A threshold is defined as k times the residual error standard deviation—k should be at least greater than 3.
3. The 3D matches whose error is over the threshold are eliminated.
4. k is set to $k-1$ and the procedure is re-iterated until $k=3$.

This approach to estimate motions yields precise results: with the rover Dala, the mean estimated standard deviations on the rotations and translations are of the order of 0.3° and 0.01 m for about half-meter motions (the error covariance on the computed motion parameters is determined using a first order approximation of the Jacobian of the minimized function (Haralick, 1994)).

In the scarce cases where VME fails to provide a motion estimate (e.g. when the perceived area is not textured enough to provide enough point matches), the 3D odometry estimate is used.

4.1.2. Observation Error Models

Landmark Observed in a Single Image. With the modified Harris detector we use, the error on the interest point location never exceeds 1.5 pixel (Schmid et al., 1998): we set a conservative value of $\sigma_p = 1.0$ pixel.

Landmark Observed with Stereovision. Given two matched interest points in a stereovision image pair, the corresponding 3D point coordinates are computed according to the usual triangulation equations:

$$z = \frac{b\alpha}{d} \qquad x = \beta_u z \qquad y = \gamma_v z$$

where z is the depth, b is the stereo baseline, α, β_u and γ_v are calibration parameters (the two latter depending on (u, v), the position of the considered pixel in the image), and d is the disparity between the matched points. Using a first order approximation, we have (Matthies, 1992):

$$\sigma_z^2 \simeq \left(\frac{\partial z}{\partial d}\right)^2 \sigma_d^2 = \frac{(b\alpha)^2}{d^4}\sigma_d^2$$

substituting the definition of z defined in (4.1.2), it follows:

$$\sigma_z = \frac{\sigma_d}{b\alpha} z^2$$

which is a well known property of stereovision, i.e. that the errors on the depth grow quadratically with the depth, and are inversely proportional to the stereo baseline. The covariance matrix of the point coordinates is then:

$$\begin{bmatrix} 1 & \beta_u & \gamma_v \\ \beta_u & \beta_u^2 & \beta_u \gamma_v \\ \gamma_v & \beta_u \gamma_v & \gamma_v^2 \end{bmatrix} \left(\frac{\sigma_d}{b\alpha} z^2\right)^2$$

All the parameters of this error model are conditioned on the estimate of σ_d, the standard deviation on the disparity. The point positions are observed with a standard deviation of 1 pixel, so we set $\sigma_d = \sqrt{2.0}$ pixel.

4.1.3. Feature Selection and Map Management. One of the advantages of using interest points as features is that they are very numerous. However, keeping many landmarks in the map is costly, the filter update stage having a quadratic complexity with the size of the state vector. It is therefore desirable to actively select among all the interest points the ones that will be kept as landmarks.

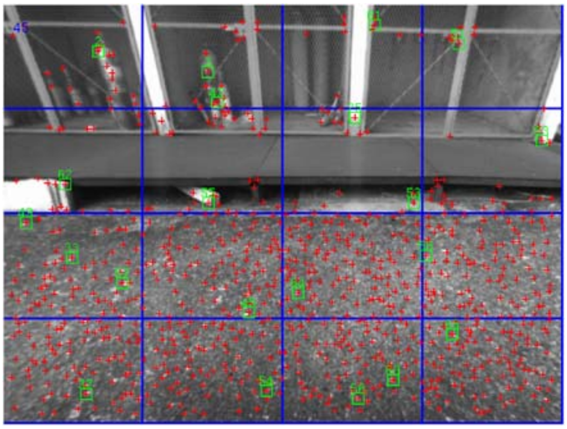


Figure 14. Selection of the points that will be kept as landmarks (green squares). Some cells here contain more than one landmark: indeed, when a landmark leaves a cell, it can move to a cell where there are already landmarks (a new landmark is then generated in the old cell).

Landmark Initialization. A good landmark should easily be observable (matched), and landmarks should be regularly dispatched in the environment. The strategy to select the landmarks is the following: the image is regularly sampled in cells (Fig. 14). If there is at least one mapped landmark in a cell, no new landmark is selected; if not, the point that has the highest Harris low eigenvalue λ_2 is selected as a landmark. This ensures a quite good regularity in the observation space (the image plane).

Map Management. In a first stage, all the landmarks are integrated in the map. The corresponding observations are helpful to estimate the pose of the robot in the short-term. Once a feature is not matched anymore (either because it is out of the field of view or is not matched), the associated landmark is only valuable for a future loop closing: in the meantime it only consumes computational resources. A successful loop closing does not require many landmarks, only a few good ones are necessary: our strategy is to keep the landmark density under a given threshold (one landmark per 0.8^3 m^3 in the experiments), the least observed landmarks being simply thrown away.

Loop Closing. Since no appearance model of the landmarks is memorized, an image database is built to achieve loop-closings: single images are stored every time the robot travels a given distance, along with the corresponding estimated robot pose and feature IDs. This database is periodically searched for a possible loop-closing on the basis of the current robot estimated position. For that purpose, a loose test between the robot and stored images positions is sufficient: one of the strengths of the matching algorithm is that it is able to match points with no knowledge of the relative transformation between the images

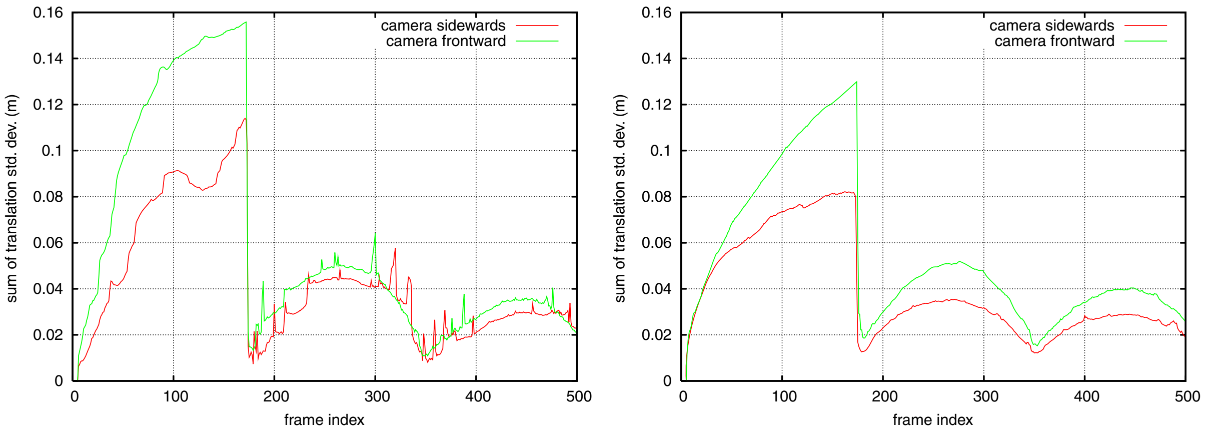


Figure 15. Comparison of the estimated uncertainty on the robot pose with cameras looking forward and sideward (left: stereovision SLAM, right: bearings-only SLAM). The robot follows a circular 6 m diameter circular trajectory during 3 laps.

(only a coarse estimate of the scale change is required, up to a factor of 0.5). When a loop-closing detection occurs, the corresponding image found in the database and the current image are fed to the matching algorithm, and successful matches are used to update the stochastic map.

4.1.4. Ground Truth. In the absence of precise devices that provide a reference position (such as a centimeter accuracy differential GPS), it is difficult to obtain a ground truth for the robot position along a whole trajectory. However, one can have a fairly precise estimate of the true rover position with respect to its starting position when it comes back near its starting position at time t, using the VME technique applied to the stereo pairs acquired at the first and current positions. VME then provides the current position with respect to the starting point, independently from the achieved trajectory: this position estimate can be used as a "ground truth" to estimate SLAM errors at time t.

4.2. Results with a Ground Rover

This section presents results obtained using data acquired with the rover Dala, equipped with a 0.35 m wide stereo bench mounted on a pan-tilt unit (the images are down-sampled to a resolution of 512 × 384). The Stereo SLAM setup is straightforward, the motion estimate computed with VME is used as the prediction input. For the bearings-only SLAM, only the left images of the stereo pairs are used, and odometry is used as the prediction input.

4.2.1. Qualitative Results. Data acquired along a simple 3-loop circular trajectory with a diameter of 6 m provide insights on the behavior of the algorithms. Two sets of data are compared here: one with the cameras looking forward, and one with the cameras heading sideward.

Figures 15 and 16-left respectively illustrate the evolution of the estimated robot pose standard deviations and

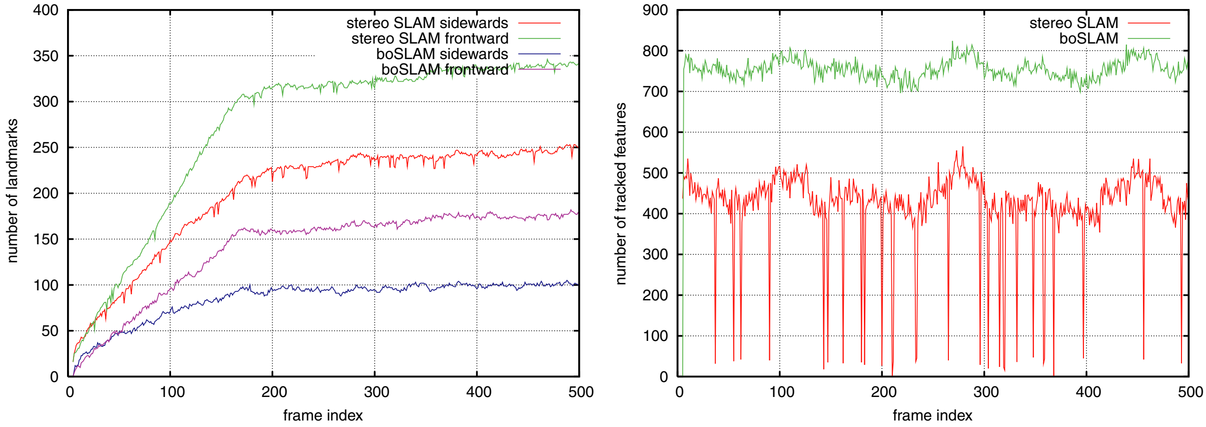


Figure 16. Number of mapped landmarks (left) and number of tracked features (right) during stereovision SLAM and bearings-only SLAM (sideward camera setup).

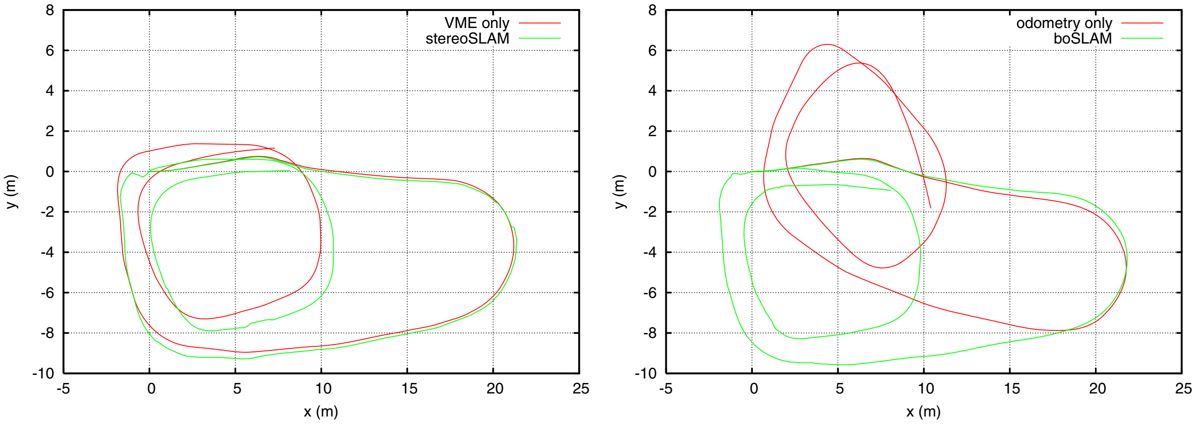


Figure 17. Left: trajectories obtained with VME integration and stereo SLAM. Right: trajectories obtained with odometry integration and bearings-only SLAM (sideward camera setup).

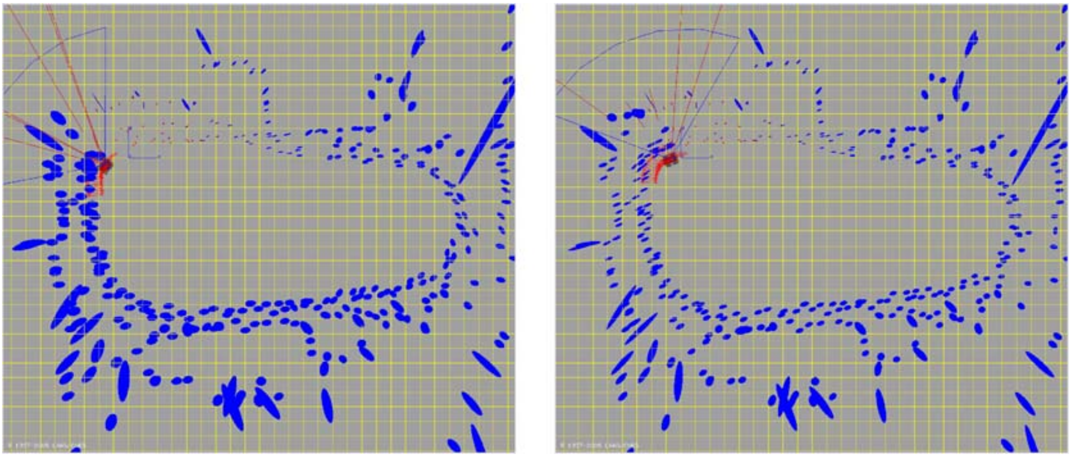


Figure 18. Bearings-only SLAM loop closing: orthogonal projection of the landmarks (3σ ellipses) just before closing the first loop (left) and just after the first loop (right). The grid step is 1 meter.

number of mapped landmarks. They exhibit the usual loop-closing effects of SLAM: the uncertainty on the robot pose dramatically drops after the first lap, and the number of mapped landmarks stops growing once the frist loop is closed.

The plots also exhibit better performance with the sideward setup than with the forward setup. Indeed, when the cameras are looking sideward, the features are tracked on more frames: the mapped landmarks are more often observed, and fewer landmarks are selected. And in the bearings-only case, the sideward setup yields a greater baseline between consecutive images, the landmarks are therefore initialized faster.

Finally, according to these plots bearings-only SLAM seems to perform better than stereo SLAM, in terms of robot position precision and number of mapped landmarks. This is actually due to the fact that in stereo SLAM, a feature needs to be matched between images $t-1$ and t *and* between the left and right images. As can be seen on Fig. 16-right, there are some frames on which few features are matched, which reduces the number of observations—note that the frame indices where few features are matched correspond to the frame indices of Fig. 15-left where the robot pose uncertainty raises.

4.2.2. Quantitative Results. Results are now presented on data acquired on a 100 m long trajectory during which two loop closures occur, for both the stereovision and bearings-only cases, the cameras being oriented sideward (Fig. 17).

Loop-Closing Analysis. Figure 18 presents an overview of the map obtained with bearings-only SLAM at different points of the trajectory. The landmarks uncertainties

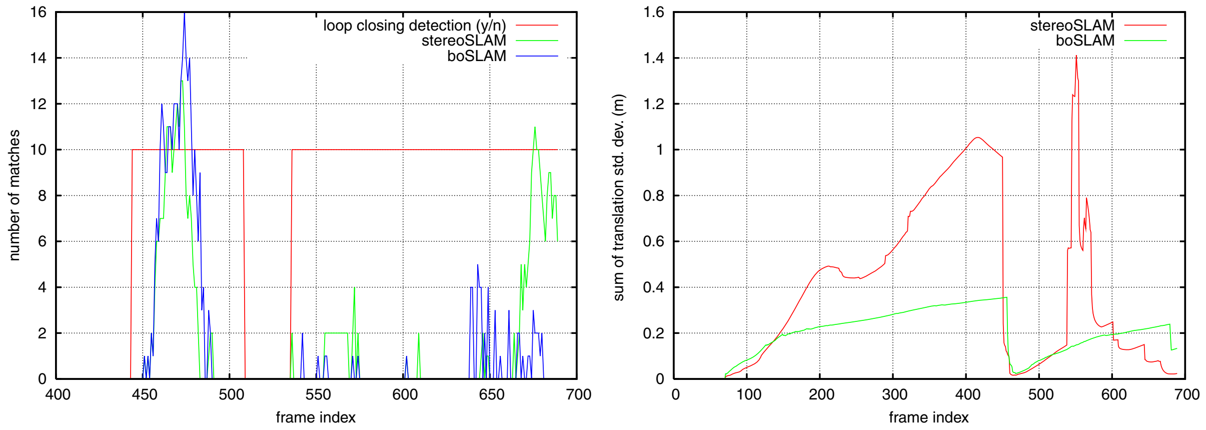


Figure 19. Left: number of mapped features that are matched during the loop-closing process. Right: uncertainty of the estimated robot pose with stereo and bearings-only SLAM.

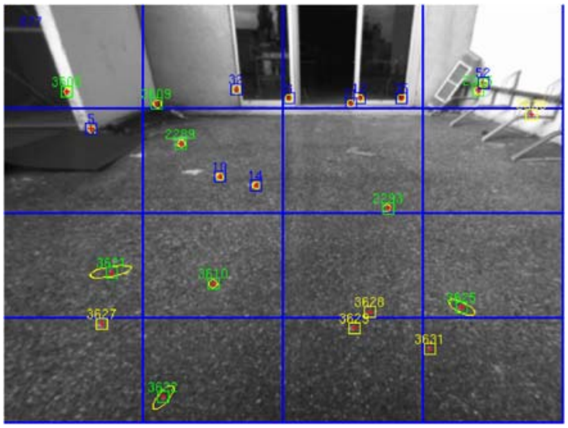


Figure 20. Stereovision SLAM: left image of the frame 677 after the second loop is closed. The overlay shows new features (*yellow*), tracked features (*green*), and features matched during the loop-closing process (*blue*), the yellow ellipses are the 3-σ bounds of the predicted observations.

obviously drop after the loop-closing process has successfully matched already mapped features: Fig. 19-left shows when a loop-closing event is detected and the number of landmarks successfully matched, and Fig. 19-right show the evolution of the robot pose uncertainty. Figure 20 shows the features used by SLAM, after the second loop has been closed. Blue features are the ones that have been mapped during the beginning of the trajectory and that are "recovered" by the loop-closing process: note that the predicted observations are consistent with the real observations, the detected features lie in the ellipses that represents their prediction.

Comparison Between Stereo SLAM and Bearings-Only SLAM. As presented in Section 4.1.4, using VME between the first frame (70) and frame 463, we obtain an accurate estimate of the actual robot pose at frame 463. Based on this reference, Table 2 compares the errors in the poses estimated by the integration of VME, stereo SLAM, the integration of odometry and bearings-only SLAM. In both the stereo and bearings-only cases, SLAM considerably reduces the errors made by a simple integration of the prediction data.

Both algorithms perform well on this data-set, but bearings-only SLAM does not produce a consistent pose estimate. Bearings-only SLAM is very sensitive to the prediction input: this is the only metric data in the system, that sets the scale of the robot trajectory and of the estimated map. On the ATRV Dala, the non-suspensed non-orientable wheels are often slipping, which yields poor odometry motion estimates (Fig. 17). A Gaussian error model for this odometry is not well-adapted: this is certainly the main source of inconsistency here.

4.3. *Results with a Blimp*

We ran the algorithms on data acquired by the blimp Karma equipped with a 2.1 m baseline stereoscopic bench

Table 2. Errors made by VME, stereo SLAM and bearings-only SLAM between frames 70 and 463 (distances in meter, angles in degree).

Loop	VME	stereoSLAM		Odometry	boSLAM	
70/463	Error	Error	Std. dev.	Error	Error	Std. dev.
x	0.26	0.003	0.006	3.71	0.25	0.01
y	1.20	0.007	0.005	1.55	0.14	0.03
z	0.65	0.012	0.006	0.16	0.03	0.02
yaw	10.1	0.1	0.1	43.2	2.3	0.2
pitch	2.0	0.3	0.1	1.7	0.4	0.3
roll	7.3	0.4	0.2	0.03	0.08	0.2

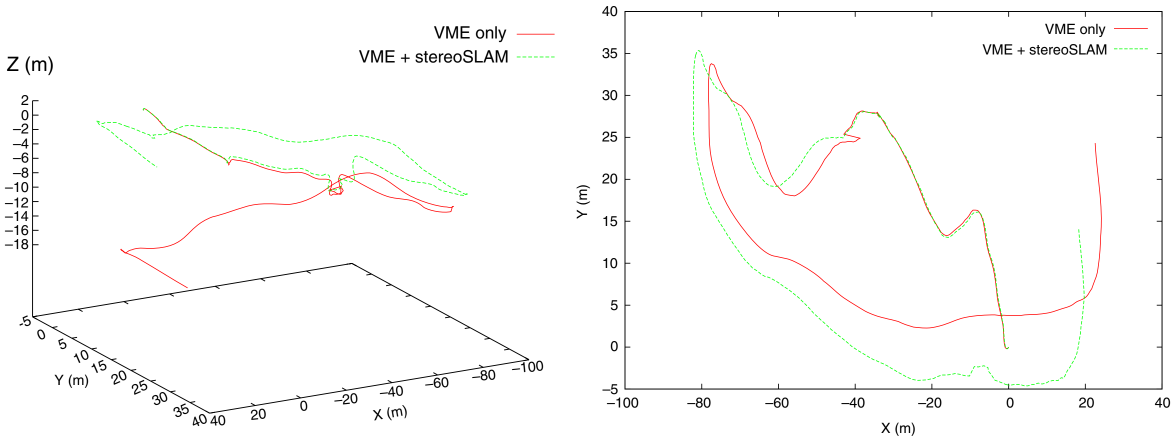


Figure 21. Comparison of the trajectory estimated by VME and stereo SLAM (3D view and projection on the ground). The trajectory is about 280 m long, and about 400 image frames have been processed.

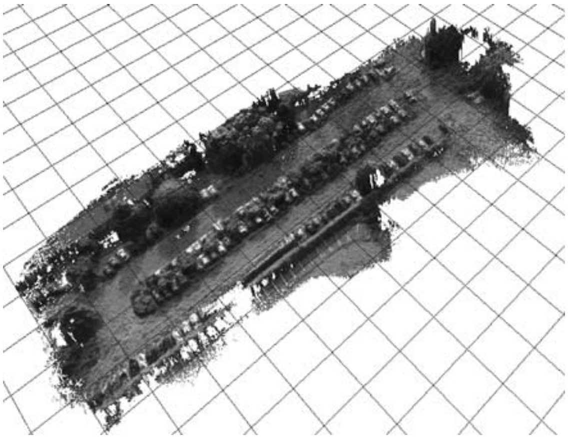


Figure 22. The digital elevation map built with dense stereovision and stereo SLAM trajectory estimate (the grid step is 10 meters).

(Fig. 1), using VME estimates as prediction for both the stereovision and bearings-only cases (Karma is not equipped with any ego-motion sensor). Figure 21 shows the trajectories estimated with VME only and with the stereo SLAM approach, and Fig. 22 shows a digital terrain map reconstructed using dense stereovision and the positions estimates provided by SLAM.

Table 3 compares the errors in the poses estimated by the integration of VME, stereo SLAM, and bearings-only SLAM, with the "ground truth" provided by the application of VME between frames 1650 and 1961 (Fig. 23). Here, the position resulting from the integration of VME exhibits a serious drift in the elevation estimate, a drift that is properly corrected by SLAM. Nevertheless, both SLAM approaches do not provide a consistent pose estimate: this is a well known problem of SLAM-EKF, that is sensitive to modeling and linearization errors (Castellanos et al., 2004). For large scale SLAM, one should switch to one of the existing algorithm that copes with it, such as hierarchical approaches which can use a different optimization framework than EKF for closing large loops (Estrada et al., 2005).

Table 3. Loop closing results with the blimp (distances in meter, angles in degree).

Loop	VME	stereoSLAM		boSLAM	
1650/1961	Error	Error	Std. dev.	Error	Std. dev.
x	1.01	9.65	0.15	9.11	0.27
y	4.11	1.35	0.13	1.87	0.16
z	13.97	0.64	0.07	1.68	0.20
yaw	4.26	1.95	0.22	4.49	0.26
pitch	5.70	0.56	0.42	1.24	0.47
roll	7.66	0.2	0.16	0.59	0.24

5. Conclusion

We presented two complete vision based SLAM solutions: a classic EKF-SLAM using stereovision observations, and a bearing-only SLAM algorithm that relies on monocular vision. Both solutions are built upon the same perception algorithm, which has interesting properties for SLAM: it can match features between images only with a coarse estimation of scale change, which enables to successfully close large loops. This would not be possible with a classic Mahalanobis data association for stereovision, and even more difficult for monocular vision.

As compared to Se et al. (2002), not so many landmarks are stored in the map, and loop closing is successfully achieved with a low number of landmarks (see Fig. 19): a good practical strategy for map management is essential, as it reduces the computation requirements of several parts of the whole SLAM process.

The presented bearing-only algorithm shows a quite good behavior with real world long range data. In particular, the delayed approach is well adapted to a

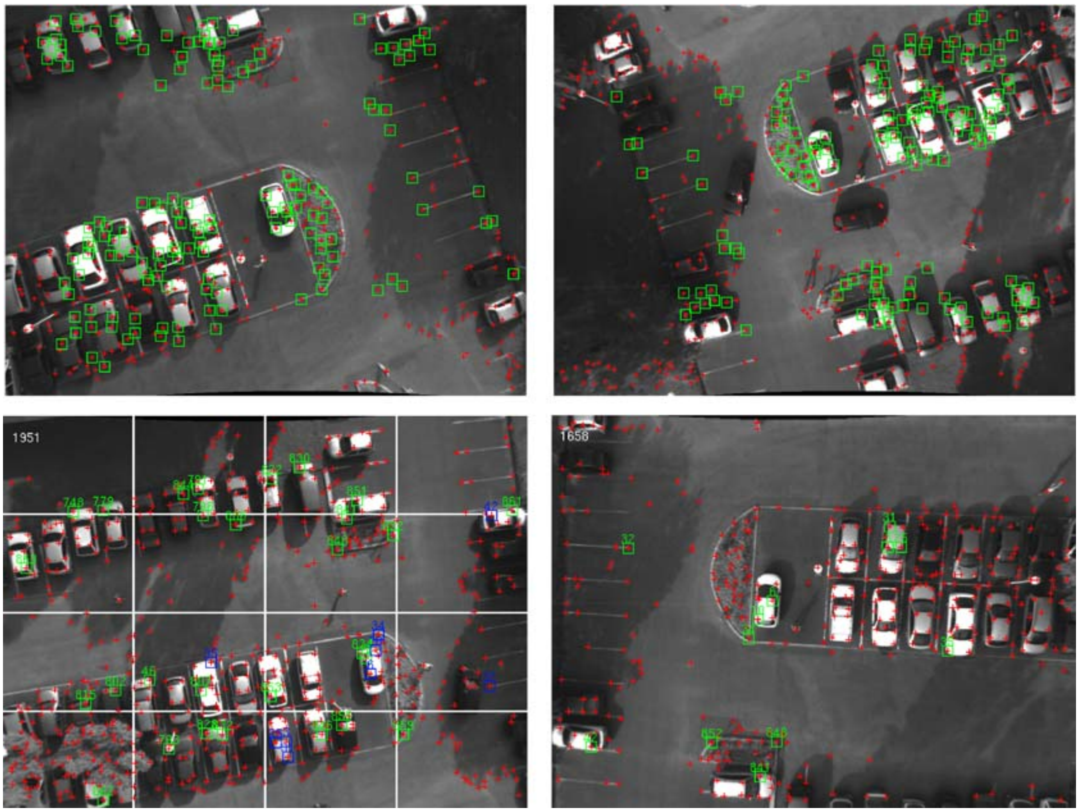


Figure 23. Top: images used to establish the ground truth by applying VME between the reference frame 1650 and the frame 1961 of the trajectory shown in Fig. 21. Green squares are the points matched between the two frames. Bottom: landmarks matched just after the loop closing. Left: the current processed image, with the landmarks currently tracked (green squares), and the ones that have been perceived again and matched (blue squares) with the stored image shown on the right.

real time implementation since it does not perform useless computations for unstable features which are rapidly lost. Experimental results confirm that it is more advantageous to have the camera oriented sideward the direction of travel: the best solution would definitely be to use a panoramic camera. Also, a good precision of the prediction function is of essential importance to initialize consistent and precise landmarks positions: for a ground rover, an efficient setup would be to use a stereovision bench to provide motion predictions, and a panoramic camera to detect and map the landmarks.

Finally, a weakness of our approach is that the loop closing mechanism requires to store a lot of images. A good solution would be to add a high level model to the landmarks, that could be matched with the perceived images—we believe the approach depicted in Rothganger et al. (2003) is very relevant for instance.

Note

1. Localization based on radioed beacons, such as GPS, fall in this category of approaches, the beacons playing the role of the a *priori* map.

References

1. Arun, K.S., Huang, T.S., and Blostein, S.D. 1987. Least-squares fitting of two 3d-points sets. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 9(5):698–700.
2. Bailey, T. 2003. Constrained initialisation for bearing-only slam. In *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan.
3. Borges, G.A. and Aldon, M-J. 2002. Optimal mobile robot pose estimation using geometrical maps. *IEEE Transactions on Robotics and Automation*, 18(1):87–94.
4. Brown, L.G. 1992. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376.
5. Castellanos, J.A., Neira, J., and Tardos, J.D. 2004. Limits to the consistency of EKF-based SLAM. In *5th symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal.
6. Chatila, R. and Laumond, J.-P. 1985. Position referencing and consistent world modeling for mobile robots. In *IEEE International Conference on Robotics and Automation*, St Louis (USA), pp. 138–145.
7. Davison, A.J. 2003. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the International Conference on Computer Vision, Nice*.
8. Davison, A.J., Cid, Y.G., and Kita, N. 2004. Real-time 3d slam with wide-angle vision. In *Proceedings of the IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon.
9. Deans, M. and Hebert, M. 2000. Experimental comparison of techniques for localization and mapping using a bearings only sensor. In

Proceedings of the ISER '00 7th International Symposium on Experimental Robotics.

10. Dellaert, F., Fox, D., Burgard, W., and Thrun, S. 1999. Monte carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA99)*.
11. Dissanayake, G., Newman, P.M., Durrant-Whyte, H.-F., Clark, S., and Csorba, M. 2001. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transaction on Robotic and Automation*, 17(3):229–241.
12. Dufournaud, Y., Schmid, C., and Horaud, R. 2004. Image matching with scale adjustment. *Computer Vision and Image Understanding*, 93(2):175–194.
13. Estrada, C., Neira, J., and Tardos, J.D. 2005. Hierarchical slam: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*.
14. Freeman, W.T. and Adelson, E.H. 1991. The design and use of steerable filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(9):891–906.
15. Guivant, J. and Nebot, E. 2001. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257.
16. Haralick, R.M. 1994. Propagating covariances in computer vision. In *International Conference on Pattern Recognition*, pp. 493–498.
17. Harris, C. and Stephens, M. 1988. A combined corner and edge detector. In *4th Alvey Vision Conference*, Manchester (UK), pp. 147–151.
18. Heeger, D.J. and Jepson, A.D. 1992. Subspace methods for recognition rigid motion i: Algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117.
19. Jung, I.-K. and Lacroix, S. 2001. A robust interest point matching algorithm. In *8th International Conference on Computer Vision*, Vancouver (Canada).
20. Jung, I.-K. and Lacroix, S. 2003. Simultaneous localization and mapping with stereovision. In *International Symposium on Robotics Research*, Siena (Italy).
21. Kim, J.H. and Sukkarieh, S. 2003. Airborne simultaneous localisation and map building. In *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan.
22. Knight, J., Davison, A., and Reid, I. 2001. Towards constant time SLAM using postponement. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems, Maui, HI*. IEEE Computer Society Press, vol. 1, pp. 406–412.
23. Konolige, K. 2005. Constraint maps: A general least squares method for slam. submited for publication.
24. Kwok, N.M. and Dissanayake, G. 2004. An efficient multiple hypothesis filter for bearing-only slam. In *IROS 2004*.
25. Kwok, N.M., Dissanayake, G., and Ha, Q.P. 2005. Bearing-only slam using a sprt based gaussian sum filter. In *ICRA 2005*.
26. Lemaire, T., Lacroix, S., and Solà, J. 2005. A practical 3d bearing only slam algorithm. In *IEEE International Conference on Intelligent Robots and Systems*.
27. Leonard, J.J. and Feder, H.J.S. 2001. Decoupled stochastic mapping. *IEEE Journal of Oceanic Engineering*, pp. 561–571.
28. Leonard, J., Rikoski, R., Newman, P., and Bosse, M. 2002. Mapping partially observable features from multiple uncertain vantage points. *International Journal of Robotics Research*.
29. Lhuillier, M. and Quan, L. 2003. Match propagation for image-based modeling and rendering. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 24(8):1140–1146.
30. Lowe, D.G. 1999. Object recognition from local scale-invariant features. In *7th International Conference on Computer Vision*, Kerkyra, Corfu (Greece), pp. 1150–1157.
31. Mallet, A., Lacroix, S., and Gallo, L. 2000. Position estimation in outdoor environments using pixel tracking and stereovision. In *IEEE International Conference on Robotics and Automation*, San Francisco, CA (USA), pp. 3519–3524.
32. Martin, J. and Crowley, J. 1995. Comparison of correlation techniques. In *International Conference on Intelligent Autonmous Systems*, Karlsruhe (Germany), pp. 86–93.
33. Matthies, L. 1992. Toward stochastic modeling of obstacle detectability in passive stereo range imagery. In *IEEE International Conference on Computer Vision and Pattern Recognition*, Champaign, Illinois (USA), pp. 765–768.
34. Nister, D. 2003. Preemptive ransac for live structure and motion estimation. In *Ninth IEEE International Conference on Computer Vision (ICCV'03)*, vol. 1, p. 199.
35. Olson, C., Matthies, L., Schoppers, M., and Maimone, M. 2000. Robust stereo ego-motion for long distance navigation. In *IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, SC (USA). JPL.
36. Peach, N. 1995. Bearing-only tracking using a set of range-parametrised extended kalman filters. *IEEE Proceedings on Control Theory Applications*, 142(1):73–80.
37. Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J. 2003. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, Madison, WI (USA), pp. 272–277.
38. Schmid, C. and Mohr, R. 1997. Local greyvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5).
39. Schmid, C., Mohr, R., and Bauckhage, C. 1998. Comparing and evaluating interest points. In *International Conference on Computer Vision*.
40. Se, S., Lowe, D., and Little, J. 2002. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758.
41. Shi, J. and Tomasi, C. 1994. Good features to track. In *IEEE International Conference on Computer Vision and Pattern Recognition*, Seattle (USA), pp. 593–600.
42. Smith, R., Self, M., and Cheeseman, P. 1987. A stochastic map for uncertain spatial relationships. In *Robotics Research: The Fourth International Symposium*, Santa Cruz (USA), pp. 468–474.
43. Solà, J., Devy, M., Monin, A., and Lemaire, T. 2005. Undelayed initialization in bearing only slam. In *IEEE International Conference on Intelligent Robots and Systems*.
44. Thrun, S. 2002. Robotic mapping: A survey. In *Exploring Artificial Intelligence in the New Millenium* G. Lakemeyer and B. Nebel (eds.), Morgan Kaufmann.
45. Thrun, S., Koller, D., Ghahramani, Z., Durrant-Whyte, H., and Ng, A.Y. 2002. Simultaneous mapping and localization with sparse extended information filters. In *Proceedings of the Fifth International Workshop on Algorithmic Foundations of Robotics*, Nice, France.
46. Thrun, S., Liu, Y., Koller, D., Ng, A.Y., Ghahramani, Z., and Durrant-Whyte, H. 2004. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, Submitted for journal publication.
47. Vidal, R., Ma, Y., Hsu, S., and Sastry, S. 2001. Optimal motion estimation from multiview normalized epipolar constraint. In *8th International Conference on Computer Vision*, Vancouver (Canada), pp. 34–41.
48. Zabih, R. and Woodfill, J. 1994. Non-parametric local transforms for computing visual correspondence. In *Third European Conference on Computer Vision*, Stockholm, (Sweden).
49. Zhang, Z. and Faugeras, O. 1992. Estimation of displacements from two 3-D frames obtained from stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(12):1141–1156.