Abstract

This article presents an approach to SLAM that takes advantage of panoramic images. Landmarks are interest points detected and matched in the images and mapped according to a bearings-only SLAM approach. As they are acquired and processed, the panoramic images are also indexed and stored into a database. A database query procedure, independent of the robot and landmark position estimates, is able to detect loop closures by retrieving memorized images that are close to the current robot position. The bearings-only estimation process is described, and results over a trajectory of a few hundreds of meters are presented and discussed.

1 Introduction

A large amount of research has been devoted to the various problems raised by SLAM, yielding operational approaches that work for hours in indoor structured environments. Much less large scale SLAM achievements can be found in field robotics, notable exceptions being abandoned mine mapping (Ferguson et al., 2003) and SLAM in urban-like environments performed at the University of Sydney (Guivant et al., 2003). In these contexts, the robot moves on a plane, and landmarks are mapped within this plane thanks to range sensors. Fewer contributions tackle 3D motion: robust and reliable large scale 3D SLAM in field applications is still a challenging problem.

The difficulties to achieving large scale SLAM are threefold: (i) the algorithmic complexity, inherent to any large scale application, (ii) the consistency issue, the estimation techniques being challenged by the large distances and the consideration of 3D motions, and (iii) the loop closing detection and feature matching issue, which are difficult to solve using the landmark and robot position estimates as they tend to be inconsistent over long distances.

The literature provides numerous solutions to the first difficulty and workarounds to
the second, relying on the use of well-suited estimation techniques and algorithmic implementations (e.g., using and information filter (Thrun et al., 2004), delaying the integration of observations (Guivant and Nebot, 2001), or splitting the global map into smaller sub-maps (Leonard and Feder, 2001)).

But loop closing detection is still challenging in field environments. The contributions that present the most advanced solutions to this problem are mainly applied in indoor-like 2D environments with range sensors, either according to an “estimation theory” point of view (e.g. by carefully controlling the estimation process (Hahnel et al., 2003b) or by allowing the revision of previous data associations (Hahnel et al., 2003a)), or according to a “perception” point of view, by exploiting perceptual data to ease the data association process rather than the position estimates (e.g. (Guivant et al., 2000; Jung and Lacroix, 2003)). To reliably detect loop closures regardless of the consistency of the position estimates, an ideal solution would be to build landmark representations that allow to quickly and unambiguously recognize them. Vision is definitely a powerful sensor for this purpose: it provides rich information on the environment, and improvements in the domain of visual object modeling and recognition are very promising. But the complex characteristics of field environments remain very challenging. Another possibility is to rely on a place recognition technique, not necessarily based on the landmark representation, that would guide the association process, yielding robust data associations. Such an approach has been proposed in (Newman and Ho, 2005) for instance, where loop closing is detected thanks to salient visual features, the estimation process being achieved on the basis of 2D laser range data. Our approach follows this principle, using indexed panoramic images.

**Vision-based SLAM.** Cameras are light, low-power sensors that provide rich sensory data and provide a mean to tackle 3D motions in space. A lot of attention has been paid to vision-based SLAM these last years, using feature points detected in images as landmarks. Stereo-vision based approaches can readily be applied with standard estimation techniques, as they are able to observe the full state of the landmarks (Jung and Lacroix, 2003; Se et al., 2005), and various contributions propose solutions to monocular vision SLAM, in which landmarks are partially observed from a single camera position. Such approaches are often referred to as “bearings-only SLAM” (Deans and Hebert, 2000; Davison, 2003; Lemaire et al., 2005) Nevertheless, the detection of loop closing on the basis of visual point features requires evaluation of matches between the current perceived features and many mapped ones, which can become very time consuming in long range applications in which thousands of images are processed.

**Proposed approach and outline.** Our SLAM approach relies on panoramic vision. Landmarks are salient point features detected and matched in the panoramic images, and their positions are estimated thanks to a bearings-only SLAM approach. The panoramic images are indexed in a database, then this database is used to efficiently detect loop closures, independently of the robot and landmark position
estimates.

Thanks to their 360° field of view, panoramic cameras have various advantages over perspective cameras for the SLAM problem. Features can be tracked over long distances: the estimation process is very well conditioned by these numerous observation of the same landmark, this also gives the possibility to map far away landmarks, which stereo-vision cannot achieve for instance. Finally, the literature abounds with contributions on the place recognition problem with panoramic images – these contributions often being referred to as “view-based navigation” or “qualitative localization”.

Besides the use of panoramic images for both landmark matching and loop closure detection, the main contribution of the paper is a bearings-only SLAM estimation process, using an efficient and optimal landmark initialization algorithm. The next section is devoted to this initialization process. Section 3 then summarizes the two vision algorithms used to match landmarks and detect loop closures. Practical (but nevertheless crucial) issues are discussed in section 4, and section 5 discusses the obtained results.

2 Bearings-Only SLAM

Monocular vision based SLAM is a partially observable SLAM problem, in which the sensor does not give enough information to compute the full state of a landmark from a single observation, thus raising a landmark initialization problem. Another instance of this problem, with sonar sensors, yields a range-only SLAM problem (a solution 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.

Bearings-only SLAM algorithms can be divided into two groups:

- In the delayed algorithms, a feature observed at time $t$ is added to the map at a subsequent time step $t + k$. This delay allows the angular baseline between observations of this landmark to grow, and the triangulation operation to become well conditioned.

- On the other hand, the un-delayed algorithms take advantage of the landmark to localize the robot at time $t$. But the update of the stochastic map has to be computed carefully.

Several contributions propose different solutions for 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 accurate results. In (Strelow and Singh, 2003; Strelow, 2004), a similar batch process is used for landmark initialization but the correlations with the current map of the newly added landmarks are not properly computed. A method based on a particle filter to represent the initial depth of a feature is proposed in (Davison, 2003; Davison et al., 2004). However its application in large environments is not straightforward, as it would require a huge number of particles.

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 extended in (Kwok et al., 2005) using a Gaussian Sum Filter, but the number of required filters can grow exponentially. This method is similar to what has been proposed in (Peach, 1995) for the bearings-only tracking. In (Solà et al., 2005), a single Kalman filter is necessary: a rigorous method based on the Federate Kalman filtering technique is introduced. Moreover, the initial PDF is defined using a geometric sum of Gaussians, this approximation is also used in our work. These techniques are especially well suited when using a camera looking towards the front of the robot. Delayed initialization methods would hardly initialize landmarks in this special case.

Bearings-only SLAM using vision is also very similar to the well known structure from motion (SFM) problem. The main difference is that robotic applications require an incremental and computationally tractable solution whereas SFM algorithm usually run in a time consuming batch process. Nevertheless incremental SFM algorithms were proposed in (McLauchlan and Murray, 1995; Broïda et al., 1990; McLauchlan, 2000) and recent works show real-time results on small data-set (Nister, 2003), and on large data-set (Mouragnon et al., 2006). The initialisation problem is adressed using a batch process involving non linear optimization algorithms such that the Levenberg-Marquardt algorithm. Also these algorithms do not consider the loop-closing problem. Links between non linear optimization algorithms usually applied in SFM and standard Kalman filter used in SLAM and bearings-only SLAM are studied in (Konolige, 2005).

2.1 Principle of the approach

The approach presented here is in the delayed category. This is a simple initialization algorithm which does not involve any batch process, also the initialization baseline is theoretically not limited. The figure 1 depicts the algorithm: when a new feature is observed, a full Gaussian estimate of its state cannot be computed from this single measure, since the bearings-only observation function cannot be inverted. The representation of this feature is initialized outside of the map with a sum of Gaussians (section 2.3). Then, a process updates this initial state representation (section 2.4), until the feature can be declared as a landmark whose full
state is is introduced in the stochastic map (section 2.5), 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 not correlated to 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 introduced by this step can be used to select the best features.

In order to add landmarks to the map, and to compute their 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. Some observations of the feature are also stored along with the corresponding robot pose estimates, so that all available information can be added to the filter when landmark initialization occurs.
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. Current pose at time $t$ is $X_r^0$, and past pose at time $t-i$ is $X_r^i$. The Kalman state is:

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

In the associated covariance matrix $P$, $P_{X_i}$ refers to covariances of sub-state $X_i$ and $P_{X_i,X_j}$ refers to cross covariance of sub-states $X_i$ and $X_j$.

In our case, the prediction step must be conducted with special care. First, it is only applied if the robot has moved a distance long enough to yield a good observability. Second, a whole part of the trajectory is estimated in the filter state: 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 backup the current robot pose and becomes $X_r^1$:

$$X_r^1 \leftarrow X_r^0, \quad P_{X_r^1} \leftarrow P_{X_r^0}, \quad \forall j P_{X_r^1,X_r^j} \leftarrow P_{X_r^0,X_r^j}$$

Then usual prediction is applied to $X_r^0$ sub-state.

Smarter approaches can be applied here which keep any past poses in the filter; the number of poses must still be bounded. If these poses are well chosen, arbitrary baseline can be obtained for the initialization.

### 2.3 Feature initialization

3D point landmarks represented by their Cartesian coordinates $X_l = (x, y, z)^t$ are now considered.

- the observation function is $z = h(X_{l/R})$,
- the inverse observation function is $X_{b/R} = g(z)$

$X_{l/R}$ is the state of the feature expressed in the robot frame, and $X_{b/R}$ is the bearing of the feature in the robot frame represented by a unit vector.
In our notation, the observation model $h()$, as well as the inverse observation model $g()$ do not include frame composition with the robot frame, 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 functions and their Jacobian matrix easier to write. Also, the same formalism can be used to take into account other frame transformations such as robot to sensor transformation. For the sake of clarity, the sensor frame is here supposed to be the same as the robot frame.

Given the first observation $z$ of the feature with covariance $P_z$, the probability density of $X_{b/R}$ is already jointly Gaussian since the measure $z$ is considered to be Gaussian.

The measure itself does not give any information about the depth, but we generally have an 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 this infinity can be bounded. This gives us for the depth $\rho$ an a priori uniform distribution in the range $[\rho_{min}, \rho_{max}]$.

This a priori PDF is approximated with a sum of $n$ Gaussians $\Gamma_i$. On one hand, it is a convenient way to approximate a PDF, and on the other hand, once a single hypothesis is selected, it is straightforward to incorporate this Gaussian in the Kalman filter.

$$p(X_{b/R}, \rho) \approx \Gamma(X_{b/R}, P_{X_{b/R}}).p(\rho)$$
$$\approx \Gamma(X_{b/R}, P_{X_{b/R}}).\sum_{i=0}^{n-1} w_i \Gamma(\rho_i, \sigma_{\rho_i}) \quad (1)$$

As shown in (Peach, 1995), the stability of a Extended Kalman filter initialized with the Gaussian $\Gamma(\rho_i, \sigma_{\rho_i})$ is governed by the ratio $\alpha = \sigma_{\rho_i}/\rho_i$. If $\alpha$ is properly chosen, the linearisation of a bearings-only observation function around $\rho_i$ is valid for this Gaussian. This gives us the definition of $\sigma_{\rho_i}$.

The means $\rho_i$ of the Gaussians are now defined. The distance between two consecutive means $\rho_i$ and $\rho_{i+1}$ is set to be proportional to $\sigma_{\rho_i} + \sigma_{\rho_{i+1}}$:

$$\rho_{i+1} - \rho_i = k_\alpha (\sigma_i + \sigma_{i+1})$$

It traduces the fact that each Gaussian fills up a depth interval which is proportional to the standard deviation of this Gaussian (see figure 1).
In fact these formula define a geometric progression for the means of the Gaussians, with the common ratio $\beta = \frac{\rho_{i+1}}{\rho_i}$:

$$
\rho_{i+1} - \rho_i = k_\sigma(\sigma_{\rho_i} + \sigma_{\rho_{i+1}}) \\
\frac{\rho_{i+1}}{\rho_i} - 1 = k_\sigma\left(\frac{\sigma_{\rho_{i+1}}}{\rho_{i+1}} + \frac{\sigma_{\rho_{i+1}}}{\rho_{i+1}}\right) \\
\beta - 1 = k_\sigma\alpha + k_\sigma\alpha\beta \\
\beta(1 - k_\sigma\alpha) = 1 + k_\sigma\alpha \\
\beta = \frac{1 + k_\sigma\alpha}{1 - k_\sigma\alpha}
$$

The sum of Gaussians which approximates the PDF of the depth $\rho$ is controlled by two parameters $\alpha$ and $k_\sigma$ or equivalently by $\alpha$ and $\beta$:

$$
\rho_0 = \frac{\rho_{\text{min}}}{1 - k_\sigma\alpha} \\
\rho_i = \beta^i\rho_0 \\
\sigma_{\rho_i} = \alpha_\rho_i \\
w_i = 1/n \\
\rho_{n-2} < \frac{\rho_{\text{max}}}{(1 + k_\sigma\alpha)} \\
\rho_{n-1} \geq \frac{\rho_{\text{max}}}{(1 + k_\sigma\alpha)}
$$

Figure 1 shows a plot of individual Gaussian members and the resultant sum for $\alpha = 0.2$ and $k_\sigma = 1.0$.

Now, let’s rewrite equation (1) under the form of a compact sum of Gaussians:

$$
p(X_{b/R}, \rho) = \sum_i w_i \Gamma(X_{b/R}, P_{X_{b/R}}), \Gamma_i(\rho_i, \sigma_{\rho_i}) \\
= \sum_i w_i \Gamma(X_{i/R}, P_{X_{i/R}}^i)
$$

where:

$$
X_{i/R} = \rho_i X_{b/R} \\
P_{X_{i/R}} = \rho_i^2 P_{X_{b/R}} + X_{b/R}\sigma_{\rho_i} X_{b/R}^T \\
P_{X_{b/R}} = GPzGT \\
G = \partial g/\partial z|_z
$$

Each $\Gamma(X_{i/R}, P_{X_{i/R}})$ represent a Gaussian hypothesis for the landmark state. Since it is kept in the robot frame, the distribution is uncorrelated with the current map. As a consequence the sum of Gaussians is not added to the state of the Kalman filter and this step of our algorithm is done at a low computational cost.

2.4 Initial state update

The sequel 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 figure 2.
Figure 2: From an observed feature 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.

Subsequent observations are used to compute the likelihood of each Gaussian $\Gamma_i$ given observation $z_t$ at time $t$. The likelihood of $\Gamma_i$ 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 $\Gamma_i$:

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

For each hypothesis, $\hat{z}_i$ and $S_i$ have to be computed. Since the feature was first seen at time $t_{\text{ref}}$, the Gaussians are expressed in the past robot frame $X_{r_{\text{ref}}}^t$, they first need to be expressed into the map frame. For clarity, let $H()$ be the full observation function, we have:

$$\hat{z}_i = h(\text{to}(\hat{X}_r^0, \text{from}(\hat{X}_{r_{\text{ref}}}^i, \hat{X}_{l/R}^i)))$$

$$= H(\hat{X}_r^0, \hat{X}_{r_{\text{ref}}}^i, \hat{X}_{l/R}^i)$$

$$S_i = H_1 P_{X_r^0} H_1^T + H_2 P_{X_{r_{\text{ref}}}^{i_{\text{ref}}}} H_2^T$$

$$+ H_1 P_{X_r^0, X_{r_{\text{ref}}}^{i_{\text{ref}}}} H_2^T + H_2 P_{X_r^0, X_{r_{\text{ref}}}^{i_{\text{ref}}}} H_1^T$$

$$+ H_3 P_{X_{l/R}^i} H_3^T + R_t$$

where

$$H_1 = \partial H / \partial X_r^0 \bigg|_{\hat{X}_r^0, X_{r_{\text{ref}}}^{i_{\text{ref}}}, X_{l/R}^i} \quad H_2 = \partial H / \partial X_{r_{\text{ref}}}^{i_{\text{ref}}} \bigg|_{\hat{X}_r^0, X_{r_{\text{ref}}}^{i_{\text{ref}}}, X_{l/R}^i} \quad H_3 = \partial H / \partial X_{l/R}^i \bigg|_{\hat{X}_r^0, X_{r_{\text{ref}}}^{i_{\text{ref}}}, X_{l/R}^i}$$
Then the bad hypotheses are selected and the associated Gaussian is pruned. Bad hypotheses are those whose likelihood $\Lambda_i$ falls under the threshold $\tau / n_t$, where $n_t$ is the number of hypotheses which remain at time $t$, and $\tau$ is a fixed value parameter smaller than 1.

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 measures using the $\chi^2$ test. An example of such a convergence is plotted step by step in figure 3. 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.

2.5 Map augmentation

When a Gaussian $\Gamma(X_{l/R}^i, P_{X_{l/R}^i})$ is chosen, the corresponding feature $j$ is declared as a landmark, and is added to the stochastic map:

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

$$\hat{X}_j^+ = \text{from}(\hat{X}_{i}^{t_{\text{ref}}}, X_{l/R}^i)$$

$$P_{X_j^+} = F_1 P_{X_{i}^{t_{\text{ref}}}, X_{l/R}^i} F_1^T + F_2 P_{X_{l/R}^i} F_2^T$$

$$P_{X_j^+,X_i^-} = F_2 P^-$$

where $F_1 = \partial \text{from} / \partial f|_{\hat{X}_{i}^{t_{\text{ref}}},X_{l/R}^i}$ and $F_2 = \partial \text{from} / \partial v|_{\hat{X}_{i}^{t_{\text{ref}}},X_{l/R}^i}$

Remember that for some steps since the feature was first seen, the feature observations were kept, 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, now this information is used to update the filter state: once a feature is added as a landmark, all available information regarding it is fused in the stochastic map.
Figure 4: Left: the robot trajectory. Right: evolution of the standard deviation on $(x, y, z)$

2.6 First results

Parameters setting. A discussion on the various parameters and their influence on the overall algorithm is given in (Lemaire et al., 2005), on the basis of simulations.

- $\alpha$ defines the size of each Gaussian: the subsequent liberalizations of the observation function must be valid around each member,
- $k_\sigma$ defines the density of Gaussians: each member must not overlap too much with its neighbors so that a single hypothesis remains after a small number of observations,
- $\tau$ is the threshold to prune bad hypotheses, if it is too high good hypotheses can be pruned, and if it is too low the convergence can be slow.

In our current setup, we use the following values: $\alpha = 0.25$, $\beta = 1.8$ and $\tau = 0.01$.

This algorithm was first applied using feature points tracked in standard perspective images. Figure 4 shows a trajectory of about 100m closing two loops. As shown by the plot of the standard deviations, the uncertainty on the robot pose is dramatically reduced during the loop closing.

3 Vision tools

3.1 Feature point landmarks

A good visual landmark should be located with precision in the image to allow a precise mapping. It should also easily be tracked in consecutive images, and matched from significantly different points of view, so that loops can be closed
without necessarily revisiting exactly the same positions. Interest points, often denoted as “corner points” have these properties: they are salient in the images and have good invariant properties. A comparison of various detectors of interest points is presented in (Schmid et al., 1998) (a more recent and complete comparison can also be found in (Mikolajczyk and Schmid, 2005)): it introduces a modified version of the Harris detector (Harris and Stephens, 1988) which relies on the computation of Gaussian image derivatives, and that gives the best repeatability under rotation and small scale changes – the repeatability being defined as the percentage of repeated interest points between two images. The detection of points that remain stable under significant scale changes is more challenging: the point detection can be adapted to an initial scale change estimate (Dufournaud et al., 2004), and scale independent feature points have been proposed (Mikolajczyk and Schmid, 2004). The SIFT features (Lowe, 1999) fall in this latter category, however their computation is quite time consuming. Nevertheless stereo-vision SLAM approaches based on SIFT features has been proposed in (Se et al., 2002) and (Barfoot, 2005).

In our approach, we use the modified Harris detector to detect features, and a previously developed matching algorithm that proved to be robust in a wide variety of image acquisition conditions (Jung and Lacroix, 2001). To establish point matches, the algorithm uses a combination of the points’ signal information computed during their extraction and of the geometric constraints between detected points.

Because of their highly non-linear image formation model, panoramic images challenge the feature matching algorithms, as small translations and rotations cause important scale changes in the images: scale independent features would be welcomed here. Nevertheless, our interest point matching algorithm behaves well with panoramic images: it produces hundreds of matches in consecutive images and enough matches to allow loop closing in images from significant different positions (figure 5).

In the worst cases, a small percentage of the established matches are outliers. (The matching results are most of the time outlier-free, and when outliers occur, their number is actually extremely low, of the order of 1 %). With classic perspective cameras, they can easily be discarded by checking the epipolar constraint. But with panoramic images, this constraint is more tricky to establish, its estimate being very unstable with respect to the matched points precision: a consistency check performed at the estimation level allows elimination of the scarce wrong matches (see section 4.5).

### 3.2 Loop closing detection

Because large scale SLAM is prone to yield inconsistent position estimates, it is of crucial importance to have a means to detect loop closing situations that is independent from the position estimates (Newman and Ho, 2005). Panoramic im-

---

1 Equiangular mirrors are less “pathological” from this point of view, but are not single view point.
Figure 5: Results of the interest point matching algorithm applied on panoramic images in the case of consecutive images (top), and in the case of significantly different points of view (bottom). The red crosses are the extracted Harris points, and the green rectangles surround points which are matched.

Images are very well suited for this purpose, as they easily allow the application of any image indexing technique to solve this problem. In the vision community, a large number of well founded approaches to image indexing have recently been proposed\(^2\), and many of them have been exploited with panoramic images in robotics to tackle the place recognition problem—a problem often associated with qualitative or topological localization. Dozens of approaches can be found in the literature: some algorithms rely on space changes to index the images (often using principal component analysis, e.g. (Jogan and Leonardis, 2000; J. Gaspar, 2001; Matej Artac and Leonardis, 2002)), the ones that exploit local image attributes, e.g. (Toshiro Matsui and Thompson., 2000; Lamon et al., 2001), and the ones that use

\(^2\)Progress in this area is mainly driven by image database management applications.
global image signatures, *e.g.* (Ishiguro and Tsuji, 1996; S. Li and Hayashi, 1996). The latest approaches have the advantage of being simple to develop, are not computationally demanding, and are nevertheless very effective: we use such an approach, that we previously developed in (Gonzalez and Lacroix, 2002).

The principle of our approach is the following: as it navigates, the rover continuously collects panoramic views of its environment and builds a database of *image indexes*, *i.e.* a set of characteristics computed on the images (learning phase). The characteristics computed are statistics (sets of histograms) of the local characteristics defined by the first and second Gaussian derivatives responses of the images, that proves to have good invariance properties. After a long traverse, when the rover arrives nearby an already crossed area, a newly perceived image is matched with the stored ones using a distance in the index space (query phase – the distance is based on $\chi^2$ statistics between the sets of histograms, considered as probability density functions).

To reduce both the storage memory and the recognition computation time, it is worthwhile to reduce the number of stored images: a simple way to achieve this is to discard the images that are similar, using the distance between image index as a criterion (see details in (Gonzalez and Lacroix, 2002)). Also, when used in a SLAM context, the search process can be focused on the basis of the current robot pose estimate – even if it is a coarse one.

The approach can efficiently find among a database of hundreds of images the closest image, “closest” being defined here in the index space. It happens that for Euclidean distances on the order of a few meters, there is a quite strong correlation between the index distance and the point of view distance. Figure 6 presents statistics obtained with thousands of image pairs that exhibit this fact, which is exploited in the whole SLAM process to determine whether a loop has been closed or not (see section 4).

![Figure 6](image_url)

**Figure 6:** Evolution of the distance in the images index space as a function of the Euclidean distance between their viewpoints (mean and standard deviation, computed on thousands of panoramic image pairs).

Figure 7 summarizes the processes that are applied to each panoramic image every
time they are acquired: the Harris interest points are extracted, and the histograms that index the image database are computed. Of course, the fact that both computations rely on the determination of Gaussian derivatives computed on the image allows efficient processing of the images.

![Diagram of processes](image)

**Figure 7:** Processes involved every time a panoramic image is acquired.

### 4 Practical issues

Achieving an operational SLAM setup requires the consideration of various “annex” issues, that can hinder the behavior of the estimation algorithm if they are not properly tackled. This section describes how we solved these issues with the objective to put the whole algorithms on-board the ATRV rover Dala (figure 8). Dala is equipped with a catadioptric camera made of a parabolic mirror and a telecentric lens, whose observation model $h()$ is presented in section 4.1. In bearings-only SLAM, the motion of the robot has to be precisely predicted: erroneous predictions yield bad landmark initial position estimates, which eventually cause the EKF to diverge. The 6 parameters of the motion have to be estimated in field robotics: the integration of odometry and inertial data provide such estimates, but they can sometimes be highly erroneous, especially when odometry experiences slippages. For this reason, we use a “visual odometry” method that exploits the stereoscopic
bench of Dala to provide the estimation process with motion estimates (section 4.2). Note that the stereo-vision bench is not only used for the purpose of SLAM: its data are used to model the perceived environment (the proposed setup of the panoramic camera and the stereo bench is effective for a navigation application). The frame transformation that links both sensors has to be known: section 4.3 presents how this could be achieved. Finally, loop closure detection and the management of the landmark map (i.e. selecting the landmarks to add or discard) are presented in sections 4.4 and 4.5, respectively.

4.1 Panoramic camera perception model

EKF based SLAM requires the observation model \( h() \) and its inverse \( g() \) of the panoramic camera, as well as the Jacobian matrix of these functions. Only Analytical expressions of these functions are given in the following, the Jacobian matrix being computed using classic calculus methods.

The “general central projection systems model” proposed by Barreto in (Barreto, 2003) is used in this work. A non-linear transformation depending on the mirror of the catadioptric system is first applied on the incoming rays, followed by a standard perspective projection\(^3\):

\[
\begin{align*}
\mathbf{h}(x, y, z) &= PM_c \begin{pmatrix}
x \\
y \\
z
\end{pmatrix} - \xi = \begin{pmatrix}
\frac{x}{\sqrt{x^2+y^2+z^2}} \\
\frac{y}{\sqrt{x^2+y^2+z^2}} \\
\frac{z}{\sqrt{x^2+y^2+z^2}} - \xi
\end{pmatrix} \\
\mathbf{g}(u, v) &= \begin{pmatrix}
\frac{-\xi \sqrt{x^2+(1-\xi^2)(y^2+z^2)}}{x^2+y^2+z^2} u \\
\frac{\xi \sqrt{x^2+(1-\xi^2)(y^2+z^2)}}{x^2+y^2+z^2} v \\
\frac{-\xi \sqrt{x^2+(1-\xi^2)(y^2+z^2)}}{x^2+y^2+z^2} \sqrt{u^2+v^2} + \xi
\end{pmatrix}
\end{align*}
\]

\(^3\text{Camera lens distortions are omitted here.}\)
with \((x, y, z)^T\) a 3D point in the camera frame, \((u, v)^T\) a 2D point in the image plane, and:

\[
P = \begin{pmatrix}
\alpha_u & 0 & u_0 \\
0 & \alpha_v & v_0 \\
0 & 0 & 1
\end{pmatrix}
\quad M_c = \begin{pmatrix}
\xi - \phi & 0 & 0 \\
0 & \xi - \phi & 0 \\
0 & 0 & 1
\end{pmatrix}
\quad \begin{pmatrix}
\bar{x} \\
\bar{y} \\
\bar{z}
\end{pmatrix} = M_c^{-1} P^{-1} \begin{pmatrix}
u \\
v \\
1
\end{pmatrix}
\]

Our mirror is a parabola with equation \(\sqrt{x^2 + y^2 + z^2} = z + 2p\), in this case the Barreto model gives \(\xi = 1\) and \(\phi = 1 + 2p\). The parameters \(p, u_0, v_0, \alpha_u, \alpha_v\) are obtained after a calibration process.

To complete the observation model, the covariance of the detected Harris corners is set to 0.5 pixel (Schmid et al., 1998).

### 4.2 Prediction using visual motion estimation (VME)

With a stereo-vision bench, the motion \(\mathbf{u}(k+1)\) and its covariance between two consecutive frames can easily be estimated using our interest point matching algorithm (Mallet et al., 2000; Olson et al., 2000). The interest points matched between the image provided by one camera at times \(k\) and \(k + 1\) can also be matched with the points detected in the other image at both times (figure 9): this produces a set of 3D point matches between time \(k\) and \(k + 1\), from which an estimate of the 6 displacement parameters can be obtained (we use the least square fitting technique presented in (Arun et al., 1987) for that purpose).

The important point here is to get rid of the outliers (wrong matches), as they considerably corrupt the minimization result. As mentioned in section 3.1, the interest point matching outliers could be rejected using the epipolar constraint defined by the fundamental matrix computed on the basis of the matches. However, the computation of this matrix is very sensitive to the small errors in the positions of the points and to the outliers themselves, and such an outlier removal technique will not address the stereo-vision matching errors, as the ones that occur along depth discontinuities for instance (matches in the image plane might become outliers when considering the corresponding 3D coordinates). Since the interest point matching algorithm generates very few 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 \(k = 3\).

The error covariance on the computed motion parameters is determined using a first order approximation of the Jacobian of the minimized function (Haralick, 1994).
Figure 9: Illustration of the matches used to estimate the robot motion with stereo-vision images, on a parking lot scene. Top: images of the stereo pair at time $k$, the matched points between the two images are denoted by green “+” signs. Bottom: images of the stereo pair at time $k+1$. The robot moved about half a meter forward between $k$ and $k+1$, and the matches obtained between the two left images at these positions are denoted by red squares.

Then, knowing the rigid transformation between the stereo bench and the reference frame (see section 4.3), the EKF prediction model is a simple frame composition.

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), we fall back on usual odometry prediction: $(v, \omega)$ is integrated in 3D with the assumption of a locally planar ground surface. Also, the distance threshold below which no filter prediction and update are applied is set to 0.2$m$ in our experiments.

### 4.3 Calibration

The Panoramic camera is calibrated thanks to the Matlab toolbox developed by Christopher Mei (Mei and Rives, 2006) (we have used the biased calibration procedure). The calibration process is very similar to the one of a simple perspective camera: images of a known planar chessboard pattern are first acquired, and a semi-automated process extracts the corners of the chessboards from the set of images. The parameter $p$ that defines the mirror is given by the manufacturer and is usually
known very precisely, and the detection of the panoramic image circular boundary gives a first estimate of $u_0, v_0$ and of the focal length. Then a minimization procedure refines their estimation.

**Inter-frames calibration.** The motion predictions being provided by one sensor (the stereo-vision bench) and the landmarks being observed by another one (the panoramic camera), it is of essential importance to have a precise estimate of the transformation between the associated frames: errors in this transformation would indeed bias the SLAM estimation process. The precise knowledge of this transformation is often taken for granted, although most SLAM applications do rely on different sensors for the prediction and the observation.

Thanks to camera calibration tools, this transformation can be estimated with good precision. Images of the chessboard are acquired simultaneously by the stereo-vision bench and the panoramic camera (figure 10). Knowing the intrinsic parameters of both cameras, the chessboard to camera transformations are computed, from which the transformation from the stereo-vision cameras to the panoramic cameras is estimated (we actually estimate the transformation between the left camera of the stereo bench and the panoramic camera, as VME produces motion estimates in this camera frame).

Figure 10 presents the frames used: the ref frame position is the one estimated by the SLAM process. In our case, the ref frame is not the usual robot frame but the sensor frame (the panoramic camera). As a consequence, there is no need for a robot to sensor frame composition at each landmark observation. The required frame composition is only applied to the predictions: since there are fewer prediction steps than observation ones, this is computationally more efficient.

Figure 10: Images used for the inter-sensor calibration. Right: in black the frames usually involved in SLAM, in red the frame we use.

### 4.4 Panoramic data flow for SLAM

Once acquired, a panoramic image is processed according to the scheme presented in figure 7 on page 15. The rest of the process for SLAM is presented in figure 11:
first, feature points detected at time $t - 1$ are tracked\(^4\) in the new set of points $t$. The histogram database is updated with the new sets of histograms, and is queried: it returns the index $k$ and the index distance $d$ of the most similar panoramic image in the database, omitting a fixed number of recent frames $k_{\text{min}}$ (the query could also be guided by the current robot pose estimate – but this would make the loop closing detection dependent from the estimation process). $d < d_t$ triggers a potential loop detection: the points detected in image $k$ and the current points are sent to the matching algorithm. The matched points between images $t$ and $k$ yield observations that are used to update the map, and new features are selected in the image areas where no corner has been tracked or matched (see next section).

---

\(^4\)We use the following terminology here: points are tracked in consecutive images, and matched in non-consecutive images.
with landmarks (*i.e.* re-observe them) that are not currently being tracked. This redefines the notion of “loop closure”: a loop-closure occurs here when a landmark that is not currently being tracked is re-observed. Loop-closure is then a “perceptual event” that does not necessarily correspond to a topological situation. Section 5 shows a plot of such “events”.

### 4.5 Feature selection and landmark management

The selection process consists in choosing among all the detected Harris points the ones that will be used in the SLAM estimation process. The panoramic images are partitioned in areas as shown in figure 12: each area should contain at least one feature tracked for SLAM, either a feature which is being initialized, or a feature corresponding to an already mapped landmark. If an area is empty, then a feature is picked up (the Harris point with the highest lower eigenvalue is chosen, as it is very likely to be further matched). This ensures a good distribution of the feature points in the image, and in our case all around the robot.

Figure 12: Partition of the panoramic image used to select the features to be used by SLAM.

When a feature is lost by the tracking algorithm, it remains in the Kalman state vector, and consumes computational power during updates. It will only be useful for future observations in the case of loop-closing. Nevertheless, a useful loop-closing does not require many landmarks, so the density of the landmarks in the map is limited. The following process is applied every time a feature is lost by the tracking process:

- if the landmark does not fall into the vicinity of others, then it remains in the map,
- if it is close to another one, then the landmark which has been observed the fewer times is removed from the map, its state sub-vector and covariance sub-matrix being garbage-collected for future landmarks.

Also, before applying a filter update, the Mahalanobis distance $d$ between the observation and the predicted observation is computed. If the observation is not
consistent $d > d_{i1}$, the update is not applied, and if the observation is too much inconsistent $d > d_{i2} > d_{i1}$ the landmark is removed from the map. These simple checks have proved to robustly eliminate the few outliers that are eventually generated by the feature matching algorithm when processing thousands of images.

5 Results

5.1 On a small loop

Figure 13 shows results obtained while the rover is following a path of about 25 meters. The top-right figure indicates the occurrence of loop-closing detection events: mapped landmarks are re-observed before the rover actually comes back to a previously visited position. As a consequence, the robot position estimate is smoothly corrected by these observations of “loop-closing” features. Here the drop in the uncertainty of the robot pose is less abrupt than with a sensor with a limited field of view (compare e.g. with figure 4 on page 11, obtained with a classic perspective camera). More abrupt loop closing causes huge Kalman gain and large innovation, which is one of the causes of divergence.
5.2 On a longer trajectory

Figure 14 shows an about 200m long trajectory, that exhibits three different loops, and during which the robot is driven in a rough area, in which the robot’s pitch and roll angles reach values of the order of 5 to 10 degrees.

Figure 15 gathers various data illustrating the behavior of the SLAM process. The top figures compare the trajectory estimated by integration of the VME local motion estimates: the $x - y$ plot shows a particular good behavior of VME, that has been “lucky” here, its errors being well compensated. However, the robot elevation estimate (top right) shows that VME is actually drifting. The middle row figures exhibit the behavior of the loop detection process. The three loops can be seen on the left curve that plots the robot pose “uncertainty volume” (the determinant of the covariance matrix): topological loop closing occur around frames 150, 320 and 1070. The right plot indicates the loop closure detections and the number of landmarks that are re-observed: this number exactly looks as expected. Indeed, the path followed by the robot revisits many known places on its way back from the rough area (around frame 1000), but no loop closing feature is matched there. The database search can sometimes give a wrong answer (i.e. the returned image has not been acquired near the robot current position), and sometimes the feature matching algorithm does not find enough matches on quite similar images. The hesitating behavior of the database search is illustrated on the figure 15-bottom-left. Nevertheless, landmarks mapped at the beginning of the trajectory are eventually re-observed, and the robot uncertainty is then considerably reduced – still in a rather smooth manner. The bottom right plot shows the evolution of the robot attitude (note the rough terrain traversed between frames 800 and 1000).
Figure 15: Top: comparison of SLAM pose estimate versus pose estimate obtained with VME integration, left: on the $x - y$ plane, right: on the $z$ axis. Middle: illustration of the loop-closing. Bottom: Pitch and roll angles during the trajectory.
### 5.2.1 Estimation errors

Although no ground truth is available along the whole trajectory, the estimated robot pose errors can be computed at some points of the trajectory: the 3D transformation between two close robot positions after an actual loop closure can be computed using VME applied between the corresponding images, and compared to the position provided by SLAM. The VME estimate is very accurate, and can be taken as ground truth. Table 1 demonstrates the precision of our SLAM algorithm:

- After an about 200 meter long trajectory, the robot pose estimate has a translation error of less than $0.15m$, and orientation errors of less than a degree.
- Nevertheless, this estimate is not strictly consistent since some translation and angular errors exceed the $3\sigma$ bound.

<table>
<thead>
<tr>
<th></th>
<th>VME 10→1430</th>
<th>$\sigma$</th>
<th>SLAM 10→1430</th>
<th>$\sigma$</th>
<th>SLAM error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>-0.033</td>
<td>0.005</td>
<td>-0.015</td>
<td>0.016</td>
<td>0.018</td>
</tr>
<tr>
<td>$y$</td>
<td>0.126</td>
<td>0.003</td>
<td>-0.016</td>
<td>0.016</td>
<td>0.142</td>
</tr>
<tr>
<td>$z$</td>
<td>-0.011</td>
<td>0.004</td>
<td>0.050</td>
<td>0.011</td>
<td>0.061</td>
</tr>
<tr>
<td>$\theta$</td>
<td>5.1</td>
<td>0.1</td>
<td>4.8</td>
<td>0.1</td>
<td>0.3</td>
</tr>
<tr>
<td>$\phi$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.3</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>$\psi$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.6</td>
<td>0.1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 1: Position estimated by the SLAM process between the first and last positions of figure 14 (frames 10 and 1430), compared to the “ground truth” provided by VME applied between the corresponding stereo-vision images.

### 6 Discussion

We have presented a 3D bearings-only SLAM approach that proves to give precise robot position estimates over long trajectories, using an approach to detect loop closures totally independent of the estimation process.

Previous work presented in (Deans, 2002) has demonstrated 2D vision based SLAM using panoramic images on a short data sequence. Our work is to our knowledge the first effective 3D SLAM approach that exploits panoramic images and natural features. The benefits of such a sensor for SLAM are obvious:

- It allows the development of efficient loop-closing detection processes based on well mastered image indexing techniques, that are independent of the position estimates. This is of crucial importance in long range field applications,
- Furthermore, the panoramic field of view allows us to sustain the landmark observations over long distances and to re-observe the mapped landmarks from significantly different positions. This conditions very well the estimation process.
We also paid a lot of attention to practical issues, in particular to the management of the map. This latter point is also favored by the use of a panoramic camera, the landmarks that are “well observed” are naturally selected by our bearings-only initialization procedure.

Nevertheless, loops of a few hundreds meters would challenge the extended Kalman filter, as the linearizations errors would eventually prevent the filter from giving consistent estimates of the robot pose and the map. Good solutions exist in the literature to avoid this well known drawback of the EKF: we are planning to implement the “Hierarchical SLAM” approach presented in (Estrada et al., 2005), which would allow the achievement of robust large scale applications.

The 3D map obtained during the trajectory of figure 14 contains some far away landmarks (our maximum range is set to 50m). Although the distance information is here nearly useless because of very high uncertainty, the observation of such features gives a lot of information, especially on the robot bearing. Our bearings-only algorithm could easily be extended to detect such landmarks and tag them as being at infinity. Such an extension would well fit within a hierarchical SLAM approach, this infinity assumption holding within some given sub-maps. The issue of exploiting such landmarks to manage the relations between the various sub-maps seems an interesting one.

Finally, if a 3D points map adjoined with a panoramic images database allows to estimate precise robot positions, such a representation does not provide useful information for the other processes required by autonomous long range navigation. An important issue in SLAM is now to be able to build and manage more descriptive models of environment, either exploiting higher level geometric features such as segments (Lemaire and Lacroix, 2006), small planar facets or larger planar polygons, or using representations that exhibit semantic information.

Acknowledgements

Many thanks to Christopher Mei who made his calibration toolbox publicly available\(^5\). Results could be obtained with the rover Dala thanks to the help of Jerôme Manhes and Matthieu Herrb.

References


\(^5\)http://www-sop.inria.fr/icare/personnel/Christopher.Mei/Toolbox.html, and gently answered our questions on the panoramic camera calibration process and the Barreto model.


Kwok, N. M. and Dissanayake, G. (2004). An efficient multiple hypothesis filter for bearing-only SLAM. In IROS.


