# A practical 3D Bearing-Only SLAM algorithm

Thomas Lemaire, Simon Lacroix and Joan Solà
*LAAS-CNRS*
*Toulouse, France*
{*thomas.lemaire, simon.lacroix, joan.sola*}@*laas.fr*

***Abstract*— This article presents a bearing only 3D SLAM algorithm which has the same complexity and optimality as the usual extended kalman filter used in classical SLAM. We especially focus on the landmark initialization process, which relies on visual point features tracked in the sequence of acquired images: a probabilistic approach to estimate their parameters is presented. This induces a particular structure of the filter architecture, in which are memorized a set of past robot poses. Simulations are made to compare the influence of some parameters required by our approach, and results with an indoor robot and an airship are presented.**

***Index Terms*— SLAM 3D, bearing only SLAM, sum of gaussians, vision**

## I. Introduction

The Simultaneous Localization And Mapping (SLAM) problem is one of the fundamental problem of robotics. It fuses proprioceptive data (odometry IMU…) with perception data (laser range finders, cameras,…) to incrementally build a consistant map of the environment, using landmarks or raw data. This kind of information is required for a variety of functionalities to endow a robot with autonomy.

Using a video camera to tackle a SLAM problem has many assets over the commonly used laser range finder:

- It is a sensor from which 3D information can be extracted. Even for indoor robots which pose can be represented in 2D $[x, y, \theta]$, the ability to gather 3D information on the environment is essential.
- Cameras are more and more used in robotics: they are well adapted for embedded systems, they are cheap, quite light, and power-saving,
- Many algorithms are available from the vision community, that can be used to extract high level primitives from an image, and match them with primitives memorized in the map – thus allowing reliable data associations, which is one of the essential problem to solve in SLAM.

On the other hand, depth information cannot be retrieved using a single image. Classic EKF SLAM algorithms can therefore not be directly applied in this case, as the addition of a feature to the stochastic map requires a full gaussian estimation of its state. As a consequence, a special landmark initialization process must be used, which combines at least two observations of the same feature from sufficiently separated robot poses.

The scale of the observed world cannot be retrieved with only vision. An arbitrary scale can be selected letting the first observed feature lay at, for exemple, one meter. But in many cases scale is necessary. In [1] a known pattern is introduced in the environemt (a white A4 sheet) which delivers this information. In robotic this metric information is available through odometry or inertial sensors.

Also, many authors use the Mahalanobis distance and the $\chi^2$ test to match observed features with landmarks in the map: this is not appropriate in the bearing-only case, because a single observation can be matched with all the landmarks that are near the corresponding line in the map. But this difficulty is nicely dealt with thanks to vision-based data association.

In this article, we propose a bearing-only SLAM algorithm based on the EKF. A description of the feature initialization process is given in section III. To assess the influence of the parameters that drive the process, simulations are presented in section IV: they show that standard values for the parameters can easily be found. Section V presents some results obtained with data gathered by two kinds of robots. Various issues that must be addressed so that the approach can be applied in any context without modification or parameter tuning are also discussed.

## II. Related work

The bearing-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 [2]: 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 bearing-only SLAM. In [3], 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. A method based on a particle filter to represent the initial depth of a feature is proposed in [1], [4]. However its application in large environments is not straightforward, as the number of particles would not scale nicely. In [5], 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. All these works propose a delayed initialization of a feature in the map, with either a solution less optimal than the kalman filter, or at the price of a higher complexity.

To our knowledge, only the work presented in [6] proposes an *undelayed* feature initialization. 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 [7] using Gaussian Sum Filter. In parallel to the work presented in our article, an undelayed algorithm based on *federated* kalman filtering is also proposed in [8].

Bearing 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 can run in a time consuming batch process. Links between non linear optimization algorithms and standard kalman filter for SLAM and bearing-only SLAM are studied in [9].

Note that other contributions use vision sensors to feed a SLAM algorithm. In [10], rectangular patches are used as landmarks, the distance from the camera to the landmarks being computed knowing the real size of the patches. In [11], landmarks are interest points, whose depth is computed thanks to a stereovision bench. Although based on vision, these approaches are however not related to a bearing-only SLAM problem.

## III. Bearing-only SLAM algorithm

The algorithm presented here is in the *delayed* category. Figure 1 depicts our overall approach. When a new feature is observed, a full gaussian estimate of its state cannot be computed from the measure, since the bearing-only observation function cannot be inverted. We initialize the representation of this feature with a sum of Gaussians (section III-B). Then, a process updates this initial state representation, until the feature can be declared as a landmark whose full state is estimated (section III-C). Once estimated, the landmark is introduced in the stochastic map, which is managed by the usual EKF (the properties of this well known approach to SLAM are presented in [12]).

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 decorrelated 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 consistant way, the pose where the robot was when the feature was first seen has to be estimated in the filter. Since then, all observations of the feature can also be memorized, and corresponding robot poses estimated, so that all available information is added to the filter at initialization.

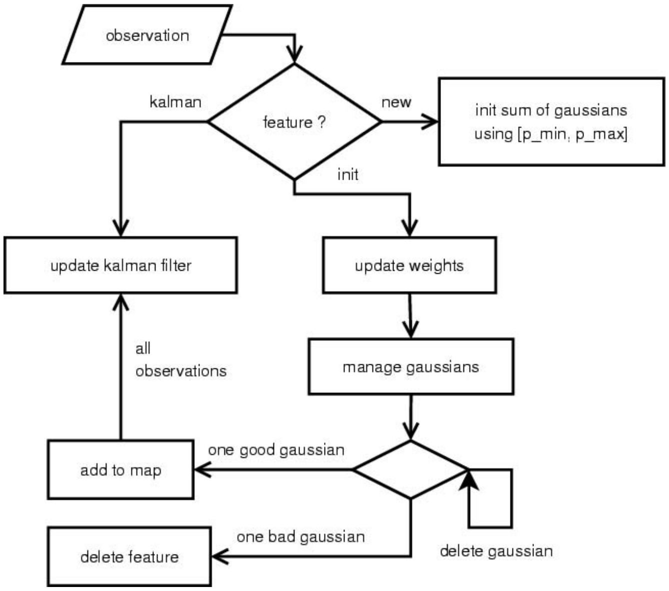

Fig. 1. Our approach to the bearing-only SLAM problem

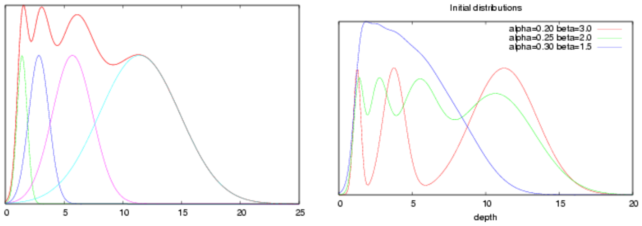

Fig. 2. Left: gaussian sum approximating the initial distribution over depth. Right: different initial distributions

### A. 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. In our implementation, the orientation of the robot is represented using Euler parameters, so each pose is a 6 dimensions vector. 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 the associated covariance matrix $P$, $P_{X_i}$ refers to covariances of substate $X_i$ and $P_{X_i,X_j}$ refers to cross covariance of substates $X_i$ and $X_j$.

The update steps of the filter are not detailed here, as they correspond to the usual kalman update equations. 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: $X_r = \left[X_r^0, X_r^2, \cdots, X_r^k, X_r^{k+1}\right]$. 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 $X_r^1$ (this is implemented using a *ring buffer* mechanism):

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

### B. Feature initialization

From now we consider 3D point features represented by their cartesian coordinates $X_f = (x, y, z)$ and the associated bearing-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 a corner point (see section V).

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 $\mathbf{to}()$ and $\mathbf{from}()$ functions: $\mathbf{to}(f, v)$ computes vector $v$ in frame $f$, and $\mathbf{from}(f, v)$ 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 angle, quaternion,...). This also makes the implementation more modular, and observation models easier to implement.

In the sensor polar coordinate system $(\rho, \theta, \phi)$, the density probability of the feature state is already jointly gaussian on $(\theta, \phi)$, 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 $\rho$ an *a priori* uniform distribution in the range $[\rho_{min}, \rho_{max}]$.

Gaussian PDFs offer nice properties, so we choose to approximate this *a priori* knowledge on the depth with a sum of gaussians:

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

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:

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

Figure 2 shows a plot of this distribution for typical values of $\alpha$ and $\beta$. We use a constant ratio $\alpha$ between the mean and the variance (the same ratio can also be found in [1]). The rate $\beta$ of geometric series defines the density of gaussians to fill in the depth range. $\alpha$ and $\beta$ are choosen so as to meet the following contraints:

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

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 $X_r^{t_f}$ (figure 3):

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

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.

### C. 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 figure 3.

Subsequent observations are used to compute the likelihood of each gaussian $i$ given observation $z_t$ at time $t$. The likelihood of $\Gamma_i$ to be an estimation of the observed feature is:

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

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 $\Gamma_i$:

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

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:

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

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

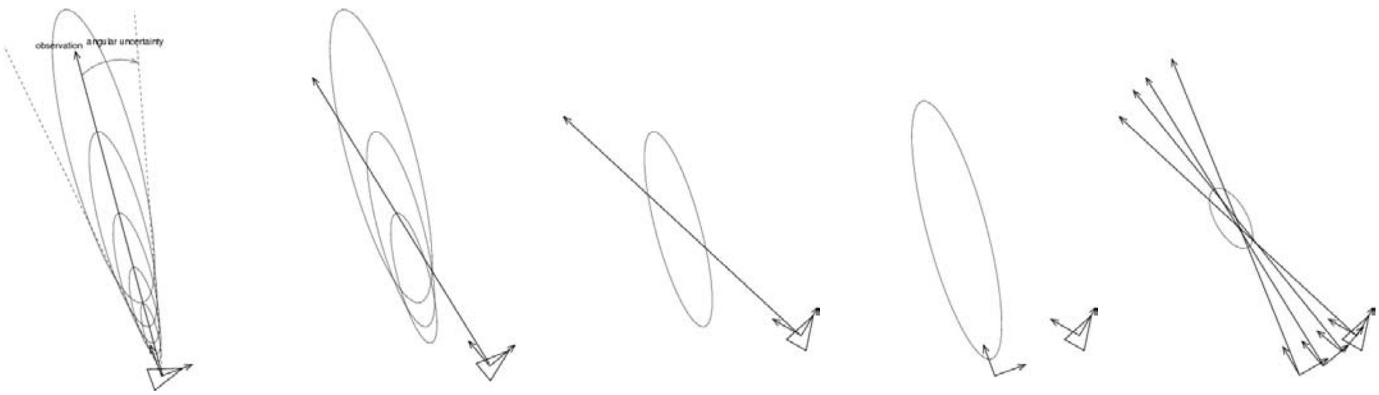

Fig. 3. From an observed feature in the images to a landmark in the map. From left to right: the sum of gaussians is initialized in the robot frame; some gaussians are pruned based on their likelihood after additional observations of the feature; when a single hypothesis remains, the feature is declared as a landmark and it is projected into the map frame; and finally past observations are used to update the landmark estimate.

Then we can select the bad hypotheses and prune the associated Gaussian. Bad hypotheses are those whose likelihood $\Lambda_i$ is low. In [6], the statistically well-founded *sequential ratio test* is used to prune bad hypotheses. This test does not seem appropriate in our case since it considers pairwise hypotheses comparisons: with 10 Gaussians, it would lead to $2 \times 45$ tests. When observing the evolution of the likelihoods $\Lambda_i$ computed with simulated or with real data, we see that the likelihood of an 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 an hypothesis if its likelihood is under a certain threshold $\tau/n$. The value of $\tau$ is discussed in section IV.

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 $\chi^2$ test. Such a convergence is plotted step by step in figure 4. 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.

### D. 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:

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

$$\begin{aligned} X_f^j &= \mathbf{from}(\hat{X}_r^{t_f^j}, \mu_i^c) \\ P_{X_f^j} &= F_1 P_{X_r^{t_f^j}} F_1^T + F_2 \Sigma_i^c F_2^T \\ P_{X_f^j, X^-} &= F_2 P^- \end{aligned}$$

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. Our algorithm tries to be as optimal as possible in the sense that all available information in the initial step is added to the stochastic map just after the feature is added as a landmark. and a robot pose, the whole correlated map is updated.

## IV. Evaluations

We made some simulations to study the influence of the different parameters of the algorithm: $(\alpha, \beta)$ which define the initial PDF, and $\tau$ which defines the threshold used to prune the hypotheses. In order to evaluate how much information we loose with our bearing-only SLAM algorithm, we compared it in simulation to a standard fully observable SLAM. We simulated a sensor which senses sizeless points with a field of view of 90 degrees and noise standard-deviation of 0.2 degree, which is realistic for camera. For the fully observable SLAM we add range information with varying noise. All tests were done in the simulated environment presented in figure 4, the robot moving along a circle.

In all our tests we track the *quality* of the estimated map and robot pose: such information is represented by the covariance matrix. Different relevant measures can be defined on a covariance matrix. We can compute the eigen values and the higher one can be selected, or the determinant of the matrix (which is equal to the product of the eigen values), or simply the trace of the matrix (which is equal to the sum of the eigen values): we use this latter criteria to analyze the results.

### A. Influence of $\tau$ and $(\alpha, \beta)$

The parameter $\tau$ directly influences the number of observations required before a feature is declared as a landmark and inserted in the map. When $\tau$ gets smaller, fewer features can be added. Figure 4 shows that values of $\tau$ from $10^{-1}$ to $10^{-5}$ do not notably reduce the number of initialized features.

Parameters $(\alpha, \beta)$ define the initial probability distribution. This distribution approximates a constant distribution but

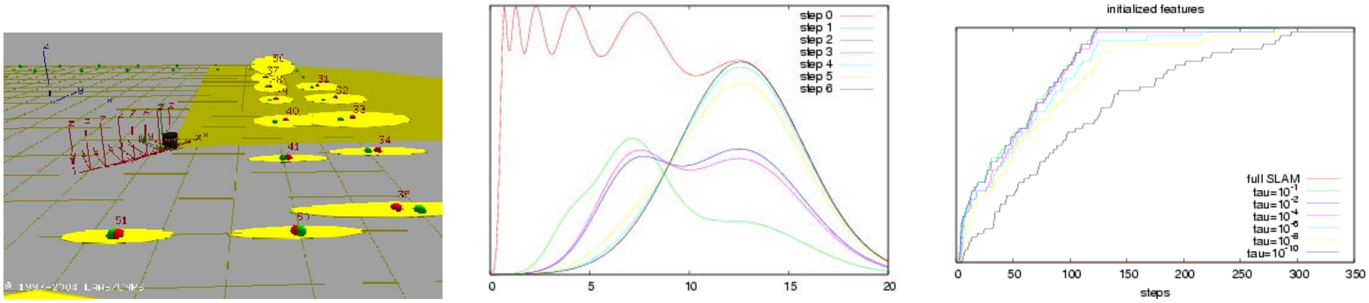

Fig. 4. Left: Simulated environment in 2D. Center: evolution of the weighted sum of gaussians through initialization steps. Right: influence of $\tau$ on the speed of feature initialization.

presents some waves. Figure 2 shows different distributions obtained with 3 sets of values. At first sight, one could try to reduce the waves by accordingly choosing $(\alpha, \beta)$. This has two bad consequences: it augments the number of gaussians, and moreover the gaussians tend to have larger overlaps. If the overlap is too important, successive measures will not discriminate a single gaussian and with our algorithm the feature will rarely be added to the map. Figure 5 shows the number of initialized features for different values of $(\alpha, \beta)$. It appears that for $\beta = 1.5$, features take significantly more time to be initialized. As one can expect, plot of map uncertainty (figure 5 plots two first laps) shows that larger value of $\alpha$ leads to larger map uncertainty since the initial gaussian added to the filter is larger.

In conclusion, the values of $\tau$ and $(\alpha, \beta)$ does not have a crucial influence, as variations of these parameters in reasonable range do not influence the algorithm result. From our simulations, we can choose $\tau = 10^{-4}$, $\alpha = 0.2$, $\beta = 1.8$.

### B. Comparison with fully observable SLAM

In order to quantify the loss of information induced by the *bearing-only* SLAM, full SLAM was run in simulation with increasing noise on range data. Figure 5 shows the evolution of the uncertainty of the pose of the robot during three laps. One can see that this uncertainty in the bearing-only case is twice the one of full observable case with $\sigma_\rho = 0.3 * \rho$.

## V. Experiments

For our experiments, we used the matching algorithm described in [13], that reliably matches Harris points from two images and can run in real-time. The camera is calibrated and the usual pin-hole model is used to compute metric observations $(\theta, \phi)$ from pixel coordinate $(u, v)$. Observation covariance $R$ is computed considering the uncertainty on $(u, v)$ is a fraction of a pixel - a typical value is 0.5 pixel.

Figure 6 gives an example of extracted and matched points. As one can see, there are a lot of points extracted and matched between two pictures, so we need to select some of them. One of the problem is to select points that will be tracked over several frames, but we do not have any *a priori* information to help us choosing such points. In order to end with enough landmarks in the map, we initialize more features than needed. This can be done at low cost with our approach, and a later selection process can be applied.

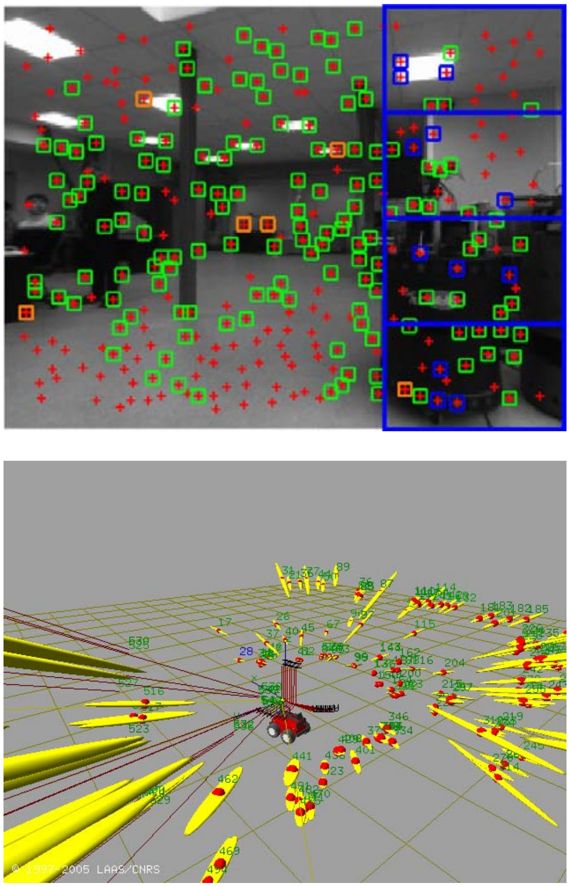

Fig. 6. Top: Camera looking on the left side of the robot, interest points (red crosses) are matched between two successive images (green), current tentative features (blue) and map features (orange). Bottom: Reconstructed map of 3D points (3-sigma ellipsoid are drawn)

As the robot moves, the part of the environment seen by the camera changes. Depending on the speed of the robot and the position of the camera on the robot, the scene in the image can move from right to left, top to bottom…or any other directions. The proposed strategy watches at the more recent part of the video image (the left part for example if the scene moves from left to right) and tries to initialize

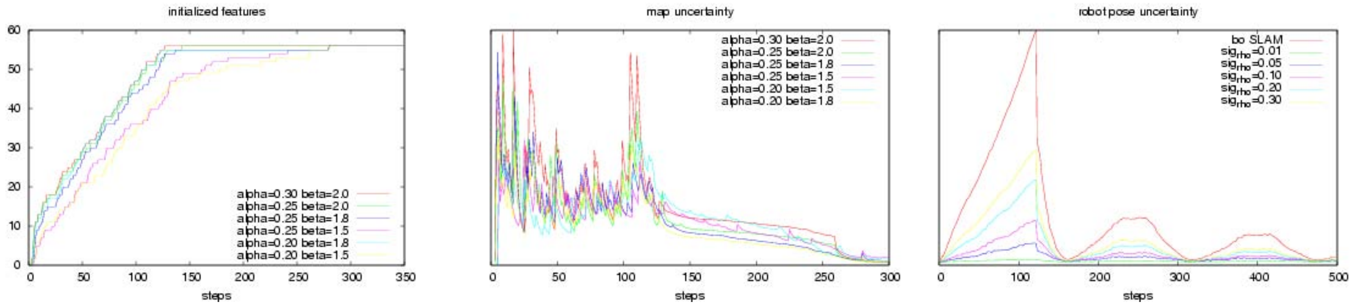

Fig. 5. Left: Influence of $\alpha, \beta$ on the speed of feature initialization. Center: influence of $\alpha, \beta$ on map uncertainty. Right: comparison of bearing-only and full SLAM

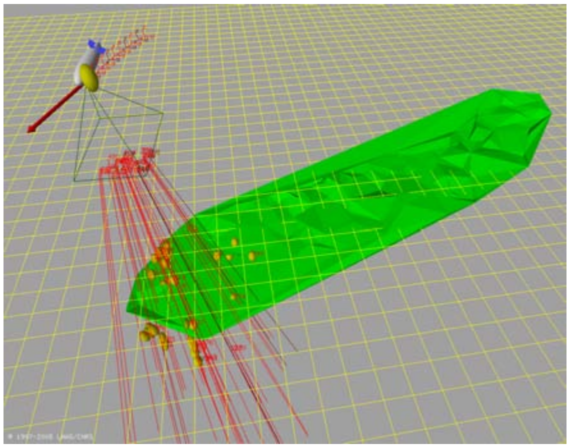

Fig. 7. Sparse elevation map of a flat area reconstructed using low altitude aerial images - the gridstep is 5 meters.

features with the interest points extracted in this particular region. We maintain a constant number of features in the "landmark nursery", and try to choose them uniformally. A simple way to achieve this is to define some subparts in the zone of interest and maintain a constant number of features in each of them, as illustrated on figure 6.

Some of the tentative features will be lost. Features whose state converges to a single Gaussian are added to the map, if the landmark density in the current map is not high enough.

We successfully ran our algorithm using images and odometry acquired from a rover in indoor environment, camera looking sideward (figure 6), the robot is driving along a circular path with diameter of about 3 meters. Also images acquired from a blimp looking downward were processed, along with GPS and attitude central data, to build a sparse elevation map. A simple Delaunay triangulation is constructed from the landmarks. Figure 7 illustrates results obtained with 100 images taken along a 120 meters straight trajectory at a low altitude (40 meters).

## VI. Discussion and future work

The algorithm proposed in this paper is efficient, offers an interesting optimality and is well adapted to a real-time implementation. The estimation part has been explained using 3D point features, but more complex primitives can be used such as lines. This is an ongoing research work, corresponding perceptions algorithms need to be finalized. Also this algorithm can be extended with large-scale map approaches so as to build large outdoor 3D maps using vision.

## References

[1] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. International Conference on Computer Vision, Nice*, October 2003. [Online]. Available: http://www.robots.ox.ac.uk/ActiveVision/Papers/davison_iccv2003/davison_iccv2003.pdf

[2] P. N. J. Leonard, R. Rikoski and M. Bosse, "Mapping partially observable features from multiple uncertain vantage points," *International Journal of Robotics Research*, jan 2002. [Online]. Available: http://oe.mit.edu/∼jleonard/pubs/leonard_etal_ijrr_w_ransac_result.pdf

[3] T. Bailey, "Constrained initialisation for bearing-only slam," in *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003. [Online]. Available: http://www.acfr.usyd.edu.au/publications/downloads/2003/Bailey206/bearing_only_constrained.pdf

[4] A. J. Davison, Y. G. Cid, and N. Kita, "Real-time 3d slam with wide-angle vision," in *Proc. IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, july 2004. [Online]. Available: http://www.robots.ox.ac.uk/ActiveVision/Papers/davison_etal_iav2004/davison_etal_iav2004.pdf

[5] M. Deans and M. Hebert, "Experimental comparison of techniques for localization and mapping using a bearings only sensor," in *Proc. of the ISER '00 Seventh International Symposium on Experimental Robotics*, December 2000. [Online]. Available: http://www.ri.cmu.edu/pubs/pub_3453_text.html

[6] N. M. Kwok and G. Dissanayake, "An efficient multiple hypothesis filter for bearing-only slam," in *IROS 2004*, 2004.

[7] N. M. Kwok, G. Dissanayake, and Q. P. Ha, "Bearing-only slam using a sprt based gaussian sum filter," in *ICRA 2005*, 2005.

[8] J. Solà, M. Devy, A. Monin, and T. Lemaire, "Undelayed initialization in bearing only slam," in *IEEE International Conference on Intelligent Robots and Systems*, august 2005. [Online]. Available: www.laas.fr/∼tlemaire/publications/solaIROS2005.pdf

[9] K. Konolige, "Constraint maps: A general least squares method for slam," 2005, submited for publication.

[10] J. H. Kim and S. Sukkarieh, "Airborne simultaneous localisation and map building," in *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003. [Online]. Available: http://www.acfr.usyd.edu.au/publications/downloads/2003/Kim194/ICRA2003SLAM

[11] I.-K. Jung and S. Lacroix, "Simultaneous localization and mapping with stereovision," in *International Symposium on Robotics Research, Siena (Italy))*, Oct 2003.

[12] Dissanayake, Newman, Durrant-Whyte, Clark, and Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on Robotic and Automation*, vol. 17, no. 3, pp. 229–241, 2001. [Online]. Available: http://oceanai.mit.edu/pnewman/papers/SLAM_TransRandA.pdf

[13] I.-K. Jung and S. Lacroix, "A robust interest point matching algorithm," in *International Conference on Computer Vision*, Vancouver (Canada), July 2001. [Online]. Available: http://www.laas.fr/∼simon/publis/JUNG-ICCV-2001.pdf