Monocular-vision based SLAM using Line Segments

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

Abstract— **This paper presents a method to incorporate 3D line segments in vision based SLAM. A landmark initialization method that relies on the Plücker coordinates to represent a 3D line is introduced: a Gaussian sum approximates the feature initial state and is updated as new observations are gathered by the camera. Once initialized, the landmarks state is estimated along an EKF-based SLAM approach: constraints associated with the Plücker representation are considered during the update step of the Kalman filter. The whole SLAM algorithm is validated in simulation runs and results obtained with real data are presented.**

I. Introduction

The use of vision sensors is becoming more and more popular in the SLAM community. Indeed, the cameras can easily be embedded on a robot, they gather a lot of useful data, and allow the development of 3D SLAM approaches. But they do not provide the depth information of the perceived features: this raises the need to develop a specific feature initialization algorithm for the commonly used extended Kalman filter framework. Vision based SLAM is often referred to as "*bearings-only* SLAM" (boSLAM).

Several contributions propose different solutions for *delayed* initial state estimation in bearings-only SLAM. In [1], 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 [2], 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 [3]. However its application in large environments is not straightforward, as the number of particles would become very large. In [4] the initial PDF of a feature is approximated by a sum of Gaussians, bad members are pruned until only a single Gaussian remains, which is then simply added to the Kalman stochastic map.

In the *Undelayed* methods proposed in [5], [6], the feature initial state is approximated with a sum of Gaussians and is explicitly added to the state of the Kalman filter. Very recent works in [7], [8] make use of an inverse parametrization of the depth: a landmark is initialized considering only a single Gaussian hypothesis on the inverse depth.

In this paper, we propose to use 3D lines as landmarks, derived from line segments perceived in images. There are some advantages to use 3D lines as landmarks: first, such primitives are very numerous in structured environments (indoor or urban outdoor), second, contrary to sparse map of points which are only useful for localization purposes, a map of segments gives relevant information on the structure of the environment: it is a sound basis to extract planes for instance. Finally, edge matching can be achieved even when important viewpoint changes occur, like in loop closing, or when matching aerial and ground data.

* Thomas Lemaire is currently working at INRIA Rhônes-Alpes, France.

Work by Folkesson *et al.* published in [9] addresses the problem of vision based SLAM with segments. In this work, an innovative estimation process based on a Kalman filter is developed, it takes advantage of partially uninitialized landmarks. An illustrative example is given in which horizontal lines are extracted from an image acquired by a camera looking up to the ceiling. The lines are supposed horizontal, so the first observation gives an estimate of their direction, but the depth is left unknown until it can be computed by triangulation. This work does not address the estimation of plain 3D lines.

The algorithms presented in [7], [8] for points were extended in [10], [11] to incorporate any 3D segments in the map. These two methods require to work at a high frame-rate (30Hz) to verify the linearization validity of the observation function with respect to the inverse depth parameter. This is not easily achieved on a robot which embeds other sensors and runs several algorithms simultaneously.

This paper focuses on the estimation problem for SLAM using image segments, and is organized as follows: section II justifies the choice of the Plücker coordinates to represent 3D lines. Section III describes the process that initializes a 3D line landmark from various image segment observations, and section IV presents how the satisfaction of the Plücker constraint can be considered within a classic EKF-based SLAM approach. Section V then presents the validation of the algorithm conducted in simulation, and finally section VI shows results obtained with real data.

II. 3D Segments for SLAM

A. 3D line representation

Several sets of parameters can be used to represent a 3D line L in euclidean space. The minimal representation consists of 4 scalars: such a minimal representation is (P_1, P_2) where $P_1 = (x_1, y_1, 0)^t$ is the intersection of L with the plane $\Pi_1(z = 0)$ and $P_2 = (x_2, y_2, 1)^t$ is the intersection of L with the plane $\Pi_2(z = 1)$. Several conventions for (Π_1, Π_2) coexist so as to represent all possible lines with a satisfactory numerical precision.

A more intuitive but non minimal representation of L is $(A, \underline{u})$, where A is any point of L, and $\underline{u}$ is a direction vector of L[1]. In this representation, the choice of A is arbitrary and A is not observable since it cannot be distinguished on the line.

An other representation often used in the vision community is the Plücker coordinates, because it is well adapted to the projection through a pinhole camera (an in-depth presentation of the Plücker coordinates can be found in [12]). In this work, we use the so called Euclidean Plücker coordinates, this is the following 6-vector:

$$L_{(6\times1)} = \begin{pmatrix} n = h.\underline{n} \\ \underline{u} \end{pmatrix} \qquad (1)$$

[1]Notation: $\underline{x}$ means that vector x is a unit vector.

$\underline{n}$ is the normal to the plane containing the line and the origin O of the reference frame, h is the distance between O and the line and $\underline{u}$ is a unit vector which represents the direction of the line. The Plücker constraint has to be satisfied[2]:

$$n \cdot \underline{u} = 0$$

This ensures that the representation is geometrically consistent. Any point P on the line satisfies the relation:

$$P \wedge \underline{u} = n \qquad (2)$$

It is also interesting to note that the closest point to the origin is given by:

$$P_O = \underline{u} \wedge n$$

The state vector of a line feature being defined by its Euclidean Plücker coordinates, the observation function and the reference frame transformation need to be defined so as to tackle the SLAM problem.

Observation function. The projection of a 3D line L in an image is a 2D line l which is defined by the intersection of the image plane and the plane defined by n: the canonical representation of l ($ax + by + c = 0$) is exactly n expressed in image coordinate:

$$l = P_{l(3\times3)} \begin{bmatrix} 1_{(3\times3)} & 0_{(3\times3)} \end{bmatrix} L \qquad l = (a, b, c)^t \qquad (3)$$

P_l is the camera projection matrix for a Plücker line, it is defined on the basis of the camera intrinsic calibration parameters. This representation is not unique and more work has to be done to compute a correct innovation (see section IV-A). A common normalized parametrization for 2D lines is (ρ, θ). It is trivially related to the canonical representation $l = (a, b, c)^t$, by the following non-linear expression:

$$(a, b, c)^t / \sqrt{a^2 + b^2} = (\cos\theta, \sin\theta, -\rho)^t$$

Frame transformation. Given a reference frame transformation $(R, t)_{1 \to 2}$, the Plücker coordinates of L in the two frames are related by:

$$L_1 = \begin{bmatrix} R & [t]_\wedge R \\ 0_{(3\times3)} & R \end{bmatrix} L_2 \qquad (4)$$

where $[x]_\wedge$ denotes the (3×3) anti-symmetric matrix that corresponds to the cross product: $\forall y, x \wedge y = [x]_\wedge . y$.

B. About segment extremities

Only the representation of the supporting line of a segment has been presented so far. The extremities of a segment cannot be used in the stochastic map: their extraction from images is not stable and depends on the viewpoint because of occlusions. Nevertheless extremities of a segment are very useful: they can be used to find visible segments so that only relevant features are tested during the matching process, and segments are more informative than infinity lines on the structure of the environment. The two extremities P_1, P_2 can be stored as two abscissas s_1, s_2 on the 3D line, the frame of the line being defined by the closest point to the origin and its direction:

$$P_O = \underline{u} \wedge n \qquad P_{\{1,2\}} = P_O + s_{\{1,2\}} \underline{u}$$

We chose to update s_1, s_2 each time their new values increase the length of the segment. That way, the estimated segment is not sensitive to occlusions or false detection of the segment extremities.

[2]Notation: $x \cdot y$ is the dot product between x and y

III. INITIAL STATE APPROXIMATION

Several recent approaches to solve the landmark initialization problem in boSLAM are based on the same idea: the non-Gaussian landmark state estimate is firstly approximated with a sum of Gaussians (or particles, which are degenerated Gaussians), then the following observations update this PDF which will converge to a single Gaussian. In *undelayed* approaches, all the Gaussians are added to the stochastic map after the first observation. But the state of a 3D line is defined in a space of higher dimension than a 3D point, and therefore requires many more members to be approximated by a Gaussian sum: for this reason, we choose a *delayed* algorithm. The algorithm presented in [4] offers a delayed method well adapted to a huge number of Gaussians. In this method, the initial Gaussian sum is expressed in the robot frame, so that it is not correlated with the current map. When a single Gaussian member is selected, it is added to the map in a consistent way, and then the past observations can be used to update the map.

Let's consider a line observation $z = (\rho, \theta)^t$ extracted from an image. This observation of the 3D line constraints L to lie on the plane Π supported by the focal point of the camera and the (ρ, θ) line of the image plane. This observation is not perfect and is modeled using a centered Gaussian noise. The problem is to find an analytical approximation of the PDF $p(L_c)$, where $L_c = (h.\underline{n}, \underline{u})$ is the 6-vector Plücker coordinates of L expressed in the camera frame. Because of the projection function of the camera, $p(L_c)$ cannot be approximated using a single Gaussian.

Nevertheless, the unit vector $\underline{n}$ which is the normal vector of Π is readily estimated and its PDF is approximated by a single Gaussian. The projection equation of the Plücker coordinates (3) gives:

$$\underline{n} = \frac{l'}{\|l'\|} \quad ; \quad l' = P_l^{-1} \begin{pmatrix} \cos\theta \\ \sin\theta \\ -\rho \end{pmatrix}$$

The result of the retro-projection needs to be normalized to recover $\underline{n}$. Using usual uncertainties propagation calculus, the Gaussian PDF $p(\underline{n})$ is obtained.

There are two quantities which prevent $p(L_c)$ from being a Gaussian: depth and direction of the line in the plane Π. In other words, depth and orientation cannot be approximated with a single Gaussian, and are approximated with a Sum of Gaussians.

In order to sample $p(L_c)$, in a relevant way, we proceed as follows:

- a "generate" vector $\underline{g}$ is defined,
- the depth d of the line is defined along $\underline{g}$, $p(d)$ is sampled,
- the direction ϕ of the line is defined with respect to $\underline{g}$, $p(\phi)$ is sampled.

Depth. A relevant "depth" of line L is not trivial to define. The natural depth is the distance of the line to the origin O of the camera frame, this distance is represented by h (1). But depending on the direction of the line, the distance to the segment (the visible part of the line) can be very different from h. As a consequence, sampling directly the parameter h is not relevant to represent the hypotheses on the depth of the object from which the segment is extracted: we rather consider as the "depth" the distance d from the camera to the real object, in the direction defined by $\underline{g}$.

To define the generate vector $\underline{g}$, an arbitrary point M is chosen on the segment in the image, for example the middle point of the visible part of the segment. M is partially stochastic: it lies on the stochastic line (ρ, θ) going through the segment, at the abscissa s_M (figure 1). s_M is an arbitrary value, so it is not stochastic. $\underline{g}$ is
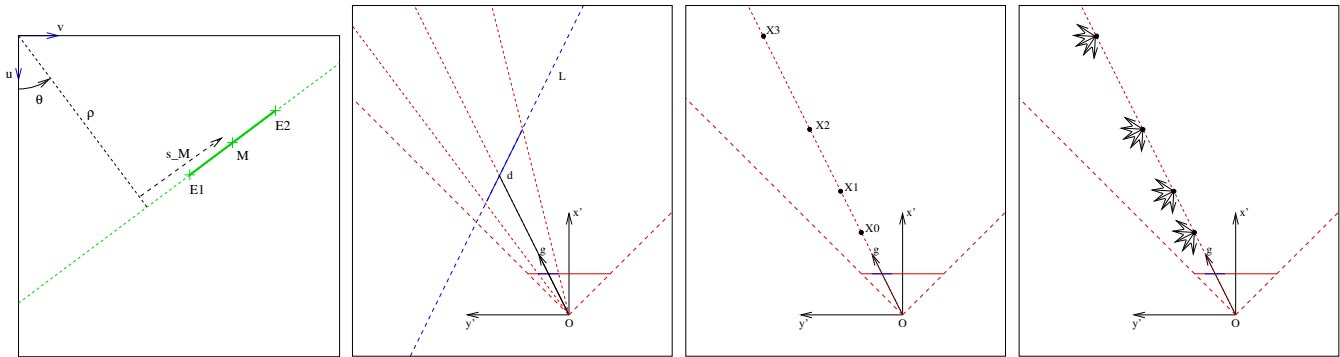
Fig. 1. From left to right: *(a)* in the image plane, definition of the point M with $(\rho, \theta, s_M)^t$. (Ox'y') corresponds to the plane Π, in red the image plane projection in Π and the camera aperture. *(b)* Projection of line L and definition of vector $\underline{g}$. *(c)* The set of points X_i. *(d)* The set of lines $L^{i,j}$.

the stochastic unit vector pointing to M, it is approximated by the Gaussian $\Gamma(\underline{g}, P_{\underline{g}})$.

$p(d)$ is supported by $\underline{g}$ and can now be expressed. $p(d)$ is a uniform distribution, its range is limited to $[d_{min}, d_{max}]$ by an *a priori* knowledge of the environment. Moreover d is a scale invariant parameter: as suggested in [4], [6], a Gaussian sum defined by a geometric series respects this property and is a good approximation of $p(d)$. Let's call β_d the rate of the geometric series, and α_d the constant ratio for the variances, $p(d)$ is approximated by:

$$p(d) \approx \sum_{0 \leq i < k_1} w_i \Gamma(d_i, \sigma_{d_i})$$

$$d_0 = d_{min}/(1 - \alpha_d) \quad d_i = \beta_d^i . d_0 \quad \sigma_{d_i} = \alpha_d . d_i \quad w_i = 1/k_d$$

The sum is composed of k_d Gaussians:

$$d_{k_d-2} < d_{max}/(1 - \alpha_d) \quad d_{k_d-1} \geq d_{max}/(1 - \alpha_d)$$

Figure 2-left illustrates this sum.

The above sampling defines a set of points X_i which is the basis for the sampling of $p(L_c)$:

$$X_i = d_i . \underline{g}$$

Direction. For each point X_i, the set of possible lines going through this point is sampled considering the direction ϕ of the line L. Again, $\underline{g}$ is used as a reference. Any possible direction u_ϕ for the line L can be obtained with $\underline{g}$ by a simple rotation of ϕ radians around $\underline{n}$:

$$\underline{u}_\phi = Rot(-\phi . \underline{n}) . \underline{g}$$

where $Rot(x)$ is the (3×3) rotation matrix associated to the rotation vector x. Any direction is equally likely, which means that $p(\phi)$ is a uniform PDF. Lets chose $\phi \in [0, \pi[$, the following Gaussian
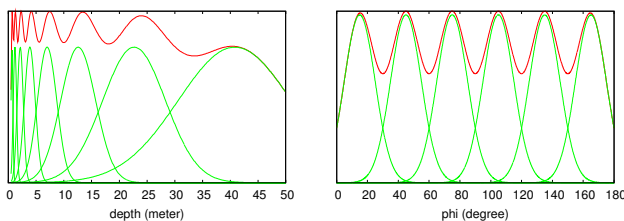

Fig. 2. Left: geometric sum of Gaussian in the range $[0.5, 50]$ with $\alpha_d = 0.25$ and $\beta_d = 1.8$. Right: uniform sum of Gaussians in the range $[0, 180]°$ with $\sigma_\phi = 10°$ and $k_{\sigma_\phi} = 1.5°$

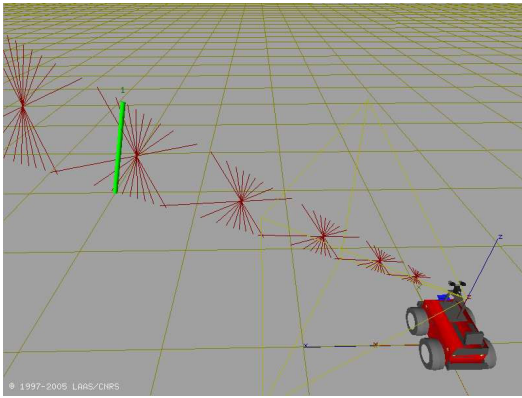sum is proposed to approximate $p(\phi)$: each Gaussian has a standard-deviation σ_ϕ and meets its two neighbors at $-k_{\sigma_\phi} . \sigma_\phi$ and $+k_{\sigma_\phi} . \sigma_\phi$ (0 and π are also considered as neighbors).

$$p(\phi) \approx \sum_{0 \leq j < k_\phi} w_j \Gamma(\phi_j, \sigma_\phi)$$

$$\phi_j = (1 + 2j) . k_{\sigma_\phi} . \sigma_\phi \qquad w_j = 1/k_\phi$$

The sum is composed of k_ϕ Gaussians:

$$k_\phi = \pi/(2 . k_{\sigma_\phi} . \sigma_\phi)$$

Approximation of $p(L_c)$. The Plücker coordinates of a sample $L_c^{i,j}$ are now clarified: the lower part $\underline{u}^{i,j}$ is trivially obtained from $\underline{u}_{\phi_j}$. The upper part is obtained with equation (2):

$$n^{i,j} = h^{i,j} . \underline{n} = (d_i . \underline{g}) \wedge \underline{u}_{\phi_j}$$

The following relation is true by definition of $\underline{g}$:

$$\underline{n} = \underline{g} \wedge \underline{u}_{\phi_j}$$

So we are left with the following equation derived from the cross product:

$$h^{i,j} = d_i \sin \phi_j$$

Which gives for $L_c^{i,j}$:

$$L_c^{i,j} = \begin{pmatrix} (d_i \sin \phi_j) . \underline{n} \\ Rot(-\phi_j . \underline{n}) . \underline{g} \end{pmatrix}$$

The original stochastic variables are the observation $(\rho, \theta)^t$ and the sampling variables over depth and direction. The following function can be formulated:

$$L_c^{i,j} = pluckerInit(\rho, \theta, d_i, \phi_j)$$

Its Jacobian is also computed so that the usual uncertainties propagation equation can be applied to compute the covariance matrix $P_{L_c^{i,j}}$

Finally, $p(L_c)$ is approximated with a Gaussian sum of $k_d \times k_\phi$ members.

$$p(L_c) \approx \sum_{\substack{0 \leq i < k_d \\ 0 \leq j < k_\phi}} w_{i,j} \Gamma(L_c^{i,j}, P_{L_c^{i,j}}) \qquad w_{i,j} = 1/k_d k_\phi$$

The step by step computation of this sum is illustrated in figure 1 and a three dimensions result is presented in figure 3.

Once the set of Gaussian hypotheses is defined, the likelihoods computation and the selection process is similar to the one presented

Fig. 3. 3D view: in green the real segment, in red the set of hypothesis.

parameter	description	value
β_d	rate of the geometric series	1.3
α_d	ratio between mean and standard-deviation	0.2
σ_ϕ	standard-deviation of each hypothesis	4°
k_{σ_ϕ}	where 2 consecutive Gaussians meet in a fraction of σ_ϕ	1.3
τ	threshold to prune bad hypothesis	10^{-2}
α_c	initial constraint noise factor	0.1
th_c	threshold on relative strength to trigger constraint application	100

Tab. 1. Summary of the parameters of the algorithm.

in [4]. The best hypothesis is converted to the map frame (equation (4)) and appended to the stochastic map. Past observations are now used to update the map.

IV. ESTIMATION PROCESS

In this work, the Kalman filter is used to estimate the 3D stochastic map. For an in-depth presentation one can refer to [13]. This section focuses on two additional problems which have to be solved:

- computation of the innovation in the (ρ, θ) space,
- consideration of the Plücker representation constraints.

A. Innovation

It is a key value in the Kalman update process. It is given by: $y = z - \hat{z}$. In the case of 2D lines observations (ρ, θ), a problem arises when the observed line and the predicted line are on both sides of the origin of the image frame. In that case, the observation is artificially modified and expressed with a negative ρ value so that the innovation reflects the correct error:

$$y = z^* - \hat{z} \qquad z^* = (-\rho, \theta \pm \pi)^t \qquad (\theta \pm \pi) \in [-\pi, \pi]$$

B. Constraints

A 3D line represented with the 6-dimension Plücker vector $L = (n, \underline{u})^t$ must respect two constraints so that it represents a valid line (section II-A):

$$\begin{cases} ||u|| = 1 & \text{(normalization)} \\ n \cdot \underline{u} = 0 & \text{(Plücker constraint)} \end{cases}$$

All the hypotheses obtained with the procedure of section III do meet these two constraints by construction. But once an hypothesis has been chosen and added to the stochastic map, there is no guarantee that the Kalman updates will not break the constraints.

An interesting work on that topic can be found in [14], it is called the "Plücker correction". Given a 6-dimension vector which does not satisfies the Plücker constraint, the algorithm finds the vector which fulfills this constraint and minimizes a special correction criterion. This method is quite complex and is not adapted to the stochastic representation.

More simply, u can be normalized at each step of the filter, or after each update of the line. This method has already been successfully applied for orientation quaternion in [3]. But it is impossible to enforce the Plücker constraint that way.

In the case of a linear Kalman filter, with a linear constraint C applying to the state vector x ($C.x = c$), the solution is trivial. The constraint is exactly enforced when updating the filter once

with observation c, null observation noise and observation matrix C. The state gets correlated so that subsequent updates do not break the constraint C.

In the case of an extended Kalman filter, with non-linear constraints, the problem is much more tedious. Badly linearized constraints can add a large base-point error and lead to divergence of the filter. The work by De Geeter *et al.* in [15] presents a *smoothly constrained Kalman filter*. This work covers strong and weak nonlinear constraints. The Plücker constraint is a strong constraint since it has to be enforced *exactly*. The constraint is *smoothly* applied to the state vector: instead of applying the constraint once with null noise (in the case of a strong constraint), the constraint is applied several times with an added *weakening* noise.

When the line L is added to the stochastic map, its estimate $\hat{L}$ respects the Plücker constraint. Then, past observations are used to update the map. After these updates, it is likely that $\hat{L}$ breaks the Plücker constraint. At this point, L is considered to be sufficiently updated so as to start the algorithm (this is our *start criteria* as defined in [15]). The initial weakening value ξ_0^w is given by:

$$\xi_0^w = \alpha_c.C\hat{P}_L C^t$$

Where C is the (1×6) Jacobian of the Plücker constraint computed at $\hat{L}$, and $\hat{P}_L$ is the covariance matrix of $\hat{L}$. The value of α_c is empirically chosen according to simulation tests.

The constraint updates are interlaced with the usual observation updates. In the case of a strong constraint, constraint updates are triggered by the test $s^c < th_c$ where th_c is a threshold and s^c is the *relative strength* defined in [15]. s^c measures how well $\hat{P}_L$ respects the correlations induced by the Plücker constraint.

$$s^c = \frac{\max_i C_{0i}.\hat{P}_{L\,ii}.C_{0i}}{C\hat{P}_L C^t}$$

The weakening values of the consecutive updates are obtained with the following formula, as suggested in [15]:

$$\xi_i^w = \xi_0^w \exp^{-nc}$$

Where nc is the number of times the constraint has already been applied.

V. SIMULATION TESTS

We ran the algorithm in simulation in order to tune the parameters, evaluate the constraints application and check the filter consistency. The simulation environment contains eight segments and the robot moves along a circle with a diameter of 10 meters. A Gaussian noise is added to the odometry $(d_s, d_\theta)^t$ with σ_{d_s} = 2.5 cm.m^{-1} and σ_{d_θ} = 1°.m^{-1}. A Gaussian noise with σ = 0.5 pixel is also added to the observations. This noise is added to the detected extremities $(u_1, v_1)^t$ and $(u_2, v_2)^t$ of the segment and then propagated to the line observation $(\rho, \theta)^t$. The algorithms are
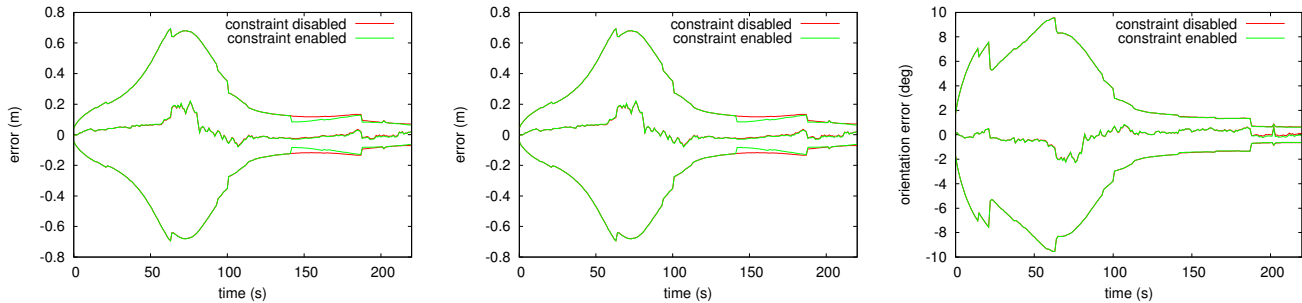
Fig. 4. Errors and 3-σ bounds of the robot pose (x, y, θ) in a simulation run, comparison with constraint disabled (red) and enabled (green).
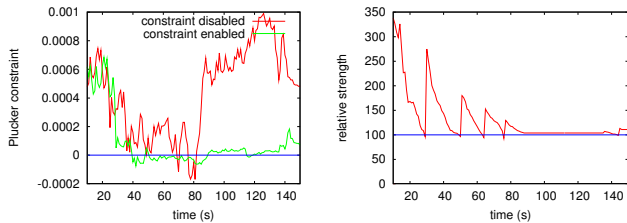

Fig. 5. Landmark number 4. Left: evaluation of $n.\underline{u}$, comparison with constraint disabled (red) and enabled (green). Right: relative strength of the constraint.

implemented in full 3D, but in the simulation the robot is moving on a plane.

A. Parameters definition

The parameters have all an intuitive meaning, but their effect is not really independent. (β_d, α_d) and $(\sigma_\phi, k_{\sigma_\phi})$ define respectively the Gaussian sums over depth d and direction ϕ of the approximation of the initial PDF of the 3D line:

- α_d and σ_ϕ define the size of each Gaussian: the subsequent linearization of the observation function must be valid around each member,
- β_d and k_{σ_ϕ} 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.

α_c is the ratio which sets the initial value of the weakening variance for the strong nonlinear Plücker constraint. This ratio is adjusted so that constraint application has no strong effect on overall consistency of the Kalman filter. th_c is set to 100, we found this value is enough, as advised in [15].

The set of parameters is summarized in table 1, their value have been empirically set in simulation.

B. Consistency check

Figure 4 presents the errors and the 3-σ bounds on the robot pose during a simulation run. This estimate is consistent all along the loop and also when the loop is closed, which is the main source of consistency violation in SLAM. Using the same random seed, in order to obtain the same sequence of noise values, the simulation was run with constraints disabled and enabled. The three plots of figure 4 shows that no significant difference appears: the application of the Plücker constraint does not affect the filter consistency.

Figure 5 gives details on the constraint efficiency and application for the feature number 4. The plot of the dot product of the Plücker constraint (left hand-side) shows that it is closer to zero when soft constraint update is applied. The plot of the relative strength (right hand-side) exhibits when the constraint is applied. Just after the landmark is initialized, the relative strength of the Plücker constraint is quite high: this is due to our initialization method which properly propagates the correlations.

VI. EXPERIMENTS WITH REAL IMAGES

A. Image segments matching

The segments are extracted in the images according to a classical procedure: first a gradient filter is applied, then the gradient is thresholded and the resulting binary image is structured into *contours*, that links neighboring high gradient pixels. A line fitting process is then applied, yielding an images of line segments (figure 6).

As can be seen on figure 6, the image noise strongly influences the segment extraction process: even for images acquired from very close positions, some segments are not repeated, and some are extracted with very different extremities – not to mention long segments that are split in two shorter segments. As a consequence, segments can hardly be matched on the basis of the coordinates of their extremities. To ensure robust and reliable segment matches, we rely on the Harris interest points matching algorithm presented in [16]: to each segment are associated the closest matched interest points, with a distance threshold very easy to specify. Segment matches are then established according to a hypotheses generation and confirmation paradigm. This simple process has proved to yield outlier-free matches (figure 6), even for large viewpoint changes, which is very helpful to associate landmarks when closing loops.

B. Results

We present results on an image sequence acquired with our ATRV rover. The robot odometry is used to feed the prediction step of SLAM. The robot moves along a circular trajectory with a diameter of 5 meters. The camera is looking sidewards to the center of the circle where two boxes have been put. In order to reconstruct the boxes edges, only segments within the blue rectangular are
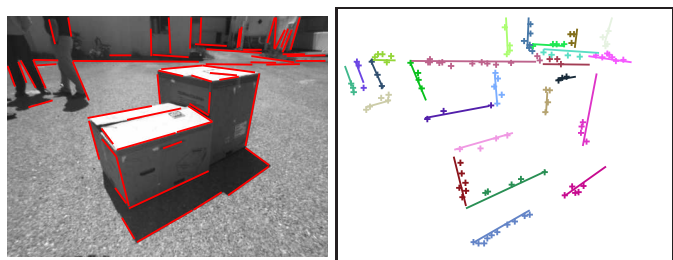

Fig. 6. Left: segments extracted in an image. Right: segments matched based on interest points matches.

id	n_x	n_y	n_z	u_x	u_y	u_z
1	1.597	-0.266	-0.083	0.051	-0.004	0.999
2	1.790	-0.117	0.059	-0.031	0.034	0.999
3	2.189	-0.563	-0.020	0.014	0.019	0.999
4	-0.268	-0.228	1.890	-0.662	0.749	-0.004
4'	-0.226	-0.246	1.877	-0.659	0.752	0.019

Tab. 2.   Plücker coordinates of the supporting line of the five landmarks
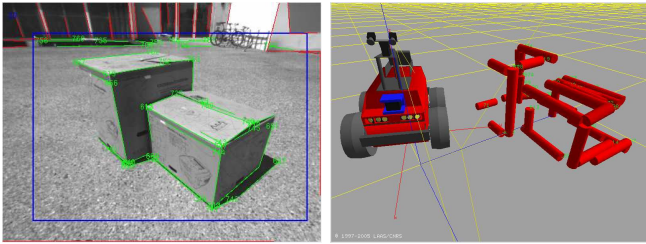

Fig. 7.   Left: an image of the sequence. Right: the 3D model from approximately the same view point.

considered (figure 7-left). Segment 1,2 and 3 are vertical edges, and segments 4 and 4' actually model the same horizontal edge. It can be verified on table 2 that segments 1,2 and 3 are close to vertical ($u_z \approx 1.0$), also segments 4 and 4' are close to horizontal ($u_z \approx 0.0$), and that they have nearly the same supporting line since there Plücker coordinates are very close. Moreover, the angle between segment 3 and 4 which forms a corner of the right box is $88.61°$ with standard deviation of $1.75°$, which is consistent with the expected value of $90°$.

An other 3D model is shown in the figure 7, this sequence was taken closer to the boxes, and with bigger boxes. This model was built along a full loop around the boxes.

VII. Discussion and future work

Depending on the trajectory of the robot, some landmarks are never initialized. When using point features, these are the points which are in front of the robot, this is why usually the camera is configured to look sidewards. When using segment features, the space of uninitialized landmarks is larger. When the camera is moving within a plane, which is mostly the case in our simulations and experiments, all the 3D segments lying in that plane cannot be initialized. This is not a drawback of our algorithm but a geometric fact.

In the light of simulations, it appears that the application of the Plücker constraint is not absolutely necessary. Numerical values of figure 5 shows that the constraint is reasonably verified. This is certainly due to the correlations which are computed in the initialization procedure. Nevertheless, this has to be verified in long terms experiments. Also, the use of the SP-Map representation [17] for the estimation process should be investigated as linearizations of the observation function do not depend on the distance of the segment to the origin of the reference frame.

Future efforts certainly include more work on the perception side, especially the stability of the segment detection and extraction, and the matching algorithm. Segments features are more invariant than points when the viewpoint changes. This property can be very useful for loop closing and also for multi-robots cooperative localization and mapping.

References

[1] 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

[2] 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

[3] 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

[4] T. Lemaire, S. Lacroix, and J. Solà, "A practical 3D bearing only SLAM algorithm," in *IEEE International Conference on Intelligent Robots and Systems*, august 2005. [Online]. Available: http://www.laas.fr/~tlemaire/publications/lemaireIROS2005.pdf

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

[6] 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: http://www.laas.fr/~tlemaire/publications/solaIROS2005.pdf

[7] E. Eade and T. Drummond, "Scalable Monocular SLAM," in *CVPR*, 2006.

[8] J. M. M. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular SLAM," in *RSS 2006*, 2006. [Online]. Available: http://www.doc.ic.ac.uk/~ajd/Publications/montiel_etal_rss2006.pdf

[9] J. Folkesson, P. Jensfelt, and H. Christensen, "Vision slam in the measurement subspace," in *Intl Conf. on Robotics and Automation*, Barcelona, Spain, April 2005. [Online]. Available: http://www.cas.kth.se/~hic/publications.html

[10] E. Eade and T. Drummond, "Edge Landmarks in Monocular SLAM," in *BMVC*, 2006. [Online]. Available: http://www.macs.hw.ac.uk/bmvc2006/papers/412.pdf

[11] P. Smith, I. Reid, and A. Davison, "Real-time monocular SLAM with straight lines," in *BMVC*, 2006. [Online]. Available: http://www.macs.hw.ac.uk/bmvc2006/papers/162.pdf

[12] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed.   Cambridge University Press, ISBN: 0521540518, 2004.

[13] Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles, Techniques, and Software*.   Artech House, 1993.

[14] A. Bartoli, "Reconstruction et alignement en vision 3d : points, droites, plans et caméras," Ph.D. dissertation, GRAVIR, 2003. [Online]. Available: http://www.lasmea.univ-bpclermont.fr/ftp/pub/bartoli/Thesis.pdf

[15] J. D. Geeter, H. V. Brussel, J. D. Schutter, and M. Decreton, "A smoothly constrained kalman filter," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 19, pp. 1171–1177, October 1997. [Online]. Available: http://people.mech.kuleuven.be/~jdgeeter

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

[17] J. Castellanos, J. Montiel, J. Neira, and J. Tardós, "The SPmap: A Probabilistic Framework for Simultaneous Localization and Map Building," *IEEE Trans. Robotics and Automation,*, vol. 15, pp. 948–953, 1999. [Online]. Available: http://webdiis.unizar.es/~jdtardos/papers/Castellanos_TRA_1999.pdf