BEARING-ONLY SAM USING A MINIMAL INVERSE DEPTH PARAMETRIZATION
Application to Omnidirectional SLAM

Cyril Joly and Patrick Rives
INRIA Sophia Antipolis Méditerranée, 2004 route des Lucioles, Sophia Antipolis Cedex, France

Keywords: Simultaneous localization and mapping (SLAM), Smoothing and mapping (SAM), Extended Kalman filter (EKF), Bearing-only, Inverse depth representation.

Abstract: Safe and efficient navigation in large-scale unknown environments remains a key problem which has to be solved to improve the autonomy of mobile robots. SLAM methods can bring the map of the world and the trajectory of the robot. Monucular visual SLAM is a difficult problem. Currently, it is solved with an Extended Kalman Filter (EKF) using the inverse depth parametrization. However, it is now well known that the EKF-SLAM become inconsistent when dealing with large scale environments. Moreover, the classical inverse depth parametrization is over-parametrized, which can also be a cause of inconsistency. In this paper, we propose to adapt the inverse depth representation to the more robust context of smoothing and mapping (SAM). We show that our algorithm is not over-parameterized and that it gives very accurate results on real data.

1 INTRODUCTION

Simultaneous localization and mapping (SLAM) is a fundamental and complex problem in mobile robotics research which has mobilized many researchers since its initial formulation by Smith and Cheeseman ((Smith and Cheeseman, 1987)). The robot moves from an unknown location in an unknown environment and proceeds to incrementally build up a navigation map of the environment, while simultaneously using this map to update its estimated position. Traditional methods are based on the extended Kalman filter (EKF). It is now well known that EKF based methods yield inconsistencies ((Julier and Uhlmann, 2001; Bailey et al., 2006a)). This is essentially due to linearization errors. Furthermore, these algorithms suffer from computational complexity ($O(N^2)$ where N is the number of landmarks in the map).

The FastSLAM method, introduced by Thrun and Montemerlo ((Montemerlo et al., 2003)), is based on the particle filter. Each position is approximated by a set of M random particles and one map is associated to each particle. It can be shown that $M \log N$ complexity is achievable. However, the algorithm is sensitive to the number of particles chosen. Furthermore, the issue of particles diversity can make the FastSLAM inconsistent ((Bailey et al., 2006b)).

In the above-mentioned methods, SLAM is solved as a filtering problem: the state-space contains only the state of the robot at the current time and the state of the map. SLAM can also be stated as a smoothing problem: the whole trajectory is taken into account (Thrun and Montemerlo, 2006; Dellaert and Kaess, 2006). This approach is referred as *simultaneous smoothing and mapping* (SAM). Although it is based on the linearization of the equations, the approach gives better results than EKF because all the linearization points are adjusted during the optimization.

A very interesting problem is the monocular visual SLAM which is also referred as Bearing-Only SLAM. Since camera does not measure directly the distance between the robot and the landmark, there is an observability issue. In standard approach, landmarks are initialized with a delay. Recent approaches try to avoid this and propose undelayed formulation. The method presented in (Civera et al., 2008) consists in changing the parametrization of the landmarks. This parametrization includes the location of the first position where the landmark was observed and the inverse depth between the landmark and this position.

Also the latter algorithm produces good results in real situations, it has some drawbacks. First, the parametrization of the landmarks is not minimal. Then, this parametrization is often associated with the EKF (although it is inconsistent).

In this paper, we propose a new way to implement the inverse depth parametrization by using a SAM approach. First, the underlying hypothesis of the SLAM and the general equations are given in section 2. Then, section 3 presents the SAM algorithm and the particularities of the inverse depth parametrization with SAM. Section 4 gives some elements of comparison between the EKF and the SAM approach. Finally, results are presented and discussed on real data acquired by an omnidirectional camera on board of a mobile robot.

2 BEARING-ONLY SLAM

2.1 Notation and Hypotheses

In the following, we denote:

- $\mathbf{x}_t$ the robot state at time t.
- $\mathbf{u}_t$ the control inputs of the model at time t.
- $\mathbf{m}_{(i)}$ the i^{th}-landmark state. Concatenation of all landmarks state is $\mathbf{m}$
- $\mathbf{z}_{t(i)}$ the measurements of landmark i at t. Concatenation for all the landmark is $\mathbf{z}_t$

In this paper, we propose to compare two methods for computing the robot and landmarks states: a new implementation of the SAM algorithm and the classical EKF. Both methods use the same hypotheses:

- the robot state $\mathbf{x}_t$ is modeled as a Markov chain,
- conditioned to all the trajectory and the map, measurements at time t depend only on $\mathbf{x}_t$ and $\mathbf{m}$ and are independent of measurements at other time.

Finally, the densities functions for the prediction of the state of the robot and the observations are assumed to be Gaussian:

$$\left\{ \begin{array}{rcl} p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t) & = & \mathcal{N}\left(\mathbf{f}(\mathbf{x}_{t-1},\mathbf{u}_t),\mathbf{Q}_t\right) \\ p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) & = & \mathcal{N}\left(\mathbf{h}(\mathbf{x}_t,\mathbf{m}),\mathbf{R}_t\right) \end{array} \right. \tag{1}$$

where $\mathbf{f}$ and $\mathbf{h}$ are the prediction and observation functions, $\mathbf{Q}_t$ and $\mathbf{R}_t$ the covariance matrices of the model and the measurements.

2.2 Camera Motion

Let us consider a standard 6DOFs BO-SLAM configuration, i.e. the robot moves in the 3D space and observes 3D-landmarks with a camera. No more sensor is considered (no odometry and no IMU). The robot position at time t is given by its Euclidian coordinates ($\mathbf{r}_t = [x_t, y_t, z_t]^T$) and its orientation by a quaternion ($\mathbf{q}_t$).

Since we do not measure the linear and angular velocities, they are estimated in the state. Linear velocity is defined in the global frame as $\mathbf{v}_t = [v_{xt}, v_{y_t}, v_{zt}]^T$ and angular velocity is defined in the camera frame as $\Omega_t = [\omega_{xt}, \omega_{y_t}, \omega_{zt}]^T$.

Furthermore, we assume that constant linear and angular accelerations act as an input on the system between t and $t+1$. It produces increments $\mathbf{V}_k$ and Ω_k. They are modeled as Gaussian white noise with zero mean. However, the velocity is not exactly constant between two time steps, so that the position and rotation are not exactly given by the integration of the velocities defined in the state. We take that into account by adding two error terms (ν_v and ν_ω) in the model (they are taken as Gaussian white noise).

Finally, the function $\mathbf{f}$ which stands for the camera motion is given by:

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{r}_{k+1} \\ \mathbf{q}_{k+1} \\ \mathbf{v}_{k+1} \\ \Omega_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_k + (\mathbf{v}_k + \mathbf{V}_k + \nu_v)\Delta t \\ \mathbf{q}_k \times \mathbf{q}\left((\Omega_k + \Omega_k + \nu_\omega)\Delta t\right) \\ \mathbf{v}_k + \mathbf{V}_k \\ \Omega_k + \Omega_k \end{bmatrix} \tag{2}$$

where $\mathbf{q}(\mathbf{u})$ is the quaternion defined by the rotation vector $\mathbf{u}$.

2.3 Inverse Depth Representation

It is well known that several observations are necessary to estimate a landmark. Indeed, a camera gives a measurement of the direction of the landmarks, but not their distance. So, landmarks initialisation is delayed when using the classical euclidian representation (a criterion based on the parallax is generally used). However, landmarks can bring bearing information when the depth is unknown. An interesting solution was given by *Civera et al* ((Civera et al., 2008)). It consists in using a new landmark representation based on the *inverse depth*. In this scheme, a 3D point (i) can be defined by a 6D vector:[1]

$$\mathbf{y}_{(i)} = \begin{bmatrix} x_{(i)} & y_{(i)} & z_{(i)} & \theta_{(i)} & \phi_{(i)} & \rho_{(i)} \end{bmatrix}^T \tag{3}$$

where $x_{(i)}$, $y_{(i)}$, $z_{(i)}$ are the euclidian coordinates of the camera position from which the landmark was observed for the first time; $\theta_{(i)}$, $\phi_{(i)}$ the azimuth and elevation (expressed in the world frame) definining unit directional vector $\mathbf{d}(\theta_{(i)}, \phi_{(i)})$ and $\rho_{(i)}$ encodes the inverse of the distance between the first camera position and the landmark. So, we can write:

$$\mathbf{m}_{(i)}^{Euclidian\ state} = \begin{bmatrix} x_{(i)} \\ y_{(i)} \\ z_{(i)} \end{bmatrix} + \frac{1}{\rho_{(i)}} \mathbf{d}(\theta_{(i)}\phi_{(i)}) \tag{4}$$

$$\mathbf{d} = \begin{bmatrix} \cos\phi_{(i)}\cos\theta_{(i)} & \cos\phi_{(i)}\sin\theta_{(i)} & \sin\phi_{(i)} \end{bmatrix}^T$$

[1]in the following, (i) stands for the landmark number. Parenthesis are used for landmarks in order to avoid confusion between the time index and landmarks index.

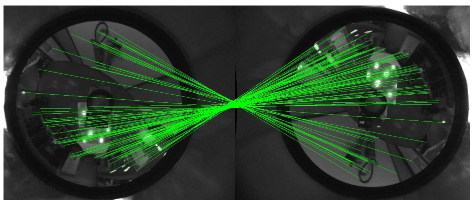


Figure 1: Data association for a loop closure. The angle between the two views is close to π. This case shows the advantage of the omnidirectional camera. Such loop closure would have been impossible with a perspective camera looking forward.

The main advantage of this representation is that it satisfies linearity condition for measurement equation even at very low parallax. According to (Civera et al., 2008),landmarks can be initialized with only one measurement.

2.4 Measurement Equation

2.4.1 Specificity of the Omnidirectional Camera Model

3D points are observed by an omnidirectional camera, which is the combination of a classical perspective camera and a mirror. For the low-level image processing, we use algorithms specifically designed to deal with omnidirectional images (see (Hadj-Abdelkader et al., 2008) for more details). An example of data association in omnidirectional images is given on Fig 1. First, Harris points are extracted from the current image. Data association with the points extracted from the previous image is performed thanks to a Sift descriptor. Finally, we obtain a set of matched points of interest in pixels coordinates.

The points in omnidirectional images are linked to the landmarks coordinates by the unified projection model (Barreto, 2003; Mei, 2007). It is done in the four steps which are recalled below. Let $\mathcal{X}$ be the euclidian coordinates of a point in the camera frame:

1. First, $\mathcal{X}$ is projected onto the unit sphere: $\mathcal{X}_s = \frac{\mathcal{X}}{\|\mathcal{X}\|} = [X_s\ Y_s\ Z_s]$
2. The third coordinate of $\mathcal{X}_s$ is shifted by ξ (mirror parameter): $\tilde{\mathcal{X}_s} = [X_s\ Y_s\ Z_s + \xi]$
3. Then, the result is projected onto the unit plane: $\mathbf{u} = [\frac{X_s}{Z_s+\xi}\ \frac{Y_s}{Z_s+\xi}\ 1]$
4. Finaly, the result is projected onto the image by $\mathbf{p} = \mathbf{K}\mathbf{u}$ where $\mathbf{K}$ is the calibration matrix of the system.

In the following, let us write $\mathbf{h}_1$ the function that returns $\mathbf{p}$ from $\mathcal{X}$:

$$\mathbf{p} = \mathbf{h}_1(\mathcal{X}) \tag{5}$$

2.4.2 Final Measurement Equation

It is shown in (Civera et al., 2008) that the Euclidian coordinates of landmark (i) in camera frame at time t are proportional to:

$$\mathbf{h}_2(\mathbf{y}_{(i)}, \mathbf{r}_t, \mathbf{q}_t) = \mathbf{R}_t^{CW} \times \\ \times \left(\rho_{(i)} \left(\begin{bmatrix} x_{(i)} \\ y_{(i)} \\ z_{(i)} \end{bmatrix} - \mathbf{r}_k \right) + \mathbf{d}(\theta_{(i)}, \phi_{(i)}) \right)$$

where $\mathbf{R}_t^{CW}$ is the rotation matrix relative to camera frame and world frame. It is the transpose of the rotation matrix associated to $\mathbf{q}_t$.

Finally, the measurement equation $\mathbf{h}$ is given by:

$$\mathbf{h} = \mathbf{h}_1 \circ \mathbf{h}_2 \tag{6}$$

3 SMOOTHING AND MAPPING

3.1 SAM Strategy

In this paper, we propose a SAM method for computing the states of the robot and the landmarks. So, we are trying to obtain $p(\mathbf{x}_{0:t}, \mathbf{m} | \mathbf{z}_{0:t}, \mathbf{u}_{1:t})$, which is the joint probability of the trajectory and the map when all the measurements and inputs are known. So, the state contains all the trajectory of the robot and the landmarks. However, SAM methods can be implemented in real-time due to the block structure of the information matrix and the use of re-ordering rules ((Dellaert and Kaess, 2006)).

3.2 SAM Equations

With the previous hypotheses, and assuming that no prior knowledge is available for the landmarks, the *pdf* of the trajectory and landmarks can be formulated as:

$$p(\mathbf{x}_{0:t}, \mathbf{m} | \mathbf{z}_{0:t}, \mathbf{u}_{1:t}) \propto p(\mathbf{x}_0)\, p(\mathbf{z}_0 | \mathbf{x}_0, \mathbf{m}) \times \\ \prod_{k=1}^{t} \left[p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)\, p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) \right] \tag{7}$$

Taking the logarithm of (7), and assuming that the functions $\mathbf{f}$ and $\mathbf{h}$ can be linearized around the approximated trajectory defined by $\mu^T = [\mu_{0:t}^{\mathbf{x}}{}^T, \mu^{\mathbf{m}T}]^T$, $\log p(\mathbf{x}_{0:t}, \mathbf{m} | \mathbf{z}_{0:t}, \mathbf{u}_{1:t})$ yields:

$$\begin{aligned}
&\log p\left(\mathbf{x}_{0:t},\mathbf{m}|\mathbf{z}_{0:t},\mathbf{u}_{1:t}\right) = \mathrm{cst} - \tfrac{1}{2}\mathbf{x}_0^T\, \mathcal{I}_0 \mathbf{x}_0 + \mathbf{x}_0^T \xi_0 \\
&+\Sigma_{k=1}^t \left\{ -\tfrac{1}{2} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{bmatrix}^T \begin{bmatrix} \mathbf{I} \\ -\mathbf{G}_k^T \end{bmatrix} \mathbf{Q}_k^{-1} \begin{bmatrix} \mathbf{I} & -\mathbf{G}_k \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{bmatrix} \right. \\
&\quad \left. + \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{bmatrix}^T \begin{bmatrix} \mathbf{I} \\ -\mathbf{G}_k^T \end{bmatrix} \mathbf{Q}_k^{-1} \left(\mathbf{f}_k\left(\mu^{\mathbf{x}}_{k-1},\mathbf{u}_k\right) + \mathbf{G}_k \mu^{\mathbf{x}}_{k-1}\right) \right\} \\
&+\Sigma_{k=0}^t \left\{ -\tfrac{1}{2} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{m} \end{bmatrix}^T \begin{bmatrix} (\mathbf{H}_k^{\mathbf{x}})^T \\ (\mathbf{H}_k^{\mathbf{m}})^T \end{bmatrix} \mathbf{R}_k^{-1} \begin{bmatrix} \mathbf{H}_k^{\mathbf{x}} & \mathbf{H}_k^{\mathbf{m}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{m} \end{bmatrix} \right. \\
&\quad \left. + \begin{bmatrix} \mathbf{x}_k \\ \mathbf{m} \end{bmatrix}^T \begin{bmatrix} (\mathbf{H}_k^{\mathbf{x}})^T \\ (\mathbf{H}_k^{\mathbf{m}})^T \end{bmatrix} \mathbf{R}_k^{-1} \left(\begin{array}{c} \mathbf{z}_k - \mathbf{h}\left(\mu^{\mathbf{x}}_k,\mu^{\mathbf{m}}\right) + \\ \begin{bmatrix} \mathbf{H}_k^{\mathbf{x}} & \mathbf{H}_k^{\mathbf{m}} \end{bmatrix} \begin{bmatrix} \mu^{\mathbf{x}}_k \\ \mu^{\mathbf{m}} \end{bmatrix} \end{array} \right) \right\}
\end{aligned} \tag{8}$$

where $\mathbf{G}_k$ is the Jacobian of $\mathbf{f}$ with respect to $\mathbf{x}_k$, $\mathbf{H}_k^{\mathbf{x}}$ (resp. $\mathbf{H}_k^{\mathbf{m}}$) is the Jacobian of $\mathbf{h}$ with respect to $\mathbf{x}_t$ (resp. $\mathbf{m}$) (the Jacobian are computed at μ). $\mathcal{I}_0$ and ξ_0 are the information matrix and vector associated to $\mathbf{x}_0$. If no prior information is available on $\mathbf{x}_0$, we set these variables to zero. Equation (8) shows that $\log p\left(\mathbf{x}_{0:t},\mathbf{m}|\mathbf{z}_{0:t},\mathbf{u}_{1:t}\right)$ is quadratic in $\left[\mathbf{x}_{0:t}^T\ ,\ \mathbf{m}^T\right]^T$. Therefore, $p\left(\mathbf{x}_{0:t},\mathbf{m}|\mathbf{z}_{0:t},\mathbf{u}_{1:t}\right)$ is Gaussian, and the computation of the information parameters are provided by (8). This yields to a sparse matrix $\mathcal{I}$[2] which simplifies the recovering of the mean and covariance.

3.3 SAM Implementation

SAM is a filtering algorithm which uses all the data to compute all the trajectory of the robot and the map. It needs a prediction of all the trajectory and the positions of the landmarks. To simulate real time conditions, we decide to use it incrementally (instead of using once all the measurements). Let us assume that we have already an estimation of $\mu^{\mathbf{x}}_{1:t-1}$ and $\mu^{\mathbf{m}}$. When the inputs and measurements are available, we update our estimation in two steps:

1. First, we compute a prediction of $\mu^{\mathbf{x}}_t$ by applying $\mathbf{f}$
2. Then, we compute $\mathcal{I}$ and ξ with Eq. (8)
3. μ is given by $\mathcal{I}^{-1}\xi$

Usually, the computation of $\mathcal{I}$ and ξ is done several times up to convergence. However, we add only one pose at each step. Thus, one iteration is enough in practice with our implementation.

[2]Sparseness results from the classical assumption that measurement of landmark (i) is independent of measurement of landmark $(j) \neq (i)$ ($\mathbf{R}_k$ matrix is block diagonal).

3.4 Minimal Landmark Parametrization

3.4.1 Definition of the Parametrization

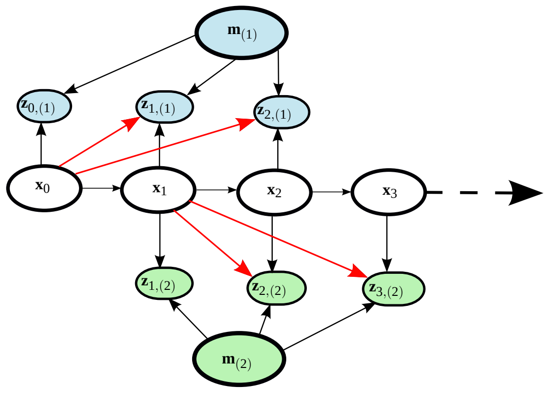
Let us discuss about the expression of the vector $\mathbf{m}$ that is estimated in the SAM algorithm. We use an inverse depth representation in order to initialize $\mathbf{m}$ at the first iteration. However, the inverse depth representation is not minimal since it uses a 6D vector to encode a 3D point. We can arbitrarily fix the three first components and compute the three others. Thus, the 3 last components are enough to represent a landmark in the state if we fix the first ones. Now, let us assume that 3 first components of $\mathbf{y}_{(i)}$ are *fixed* to $\tilde{x}_{(i)}$, $\tilde{y}_{(i)}$ and $\tilde{z}(i)$, so that we have $\mathbf{m}_{(i)} = \left[\theta_{(i)}\ \phi_{(i)}\ \rho_{(i)}\right]^T$.

However, we can not choose an arbitrary value for $\tilde{x}_{(i)}, \tilde{y}_{(i)}$ and $\tilde{z}(i)$ if the goal is to insert $\mathbf{m}_{(i)}$ in the filter at the first iteration. Indeed, the linearity constraints has to be respected, which implies that the approximated trajectory has to be close enough to the true one. To do that, $\tilde{x}_{(i)}$, $\tilde{y}_{(i)}$ and $\tilde{z}_{(i)}$ are fixed to $\mu^{\mathbf{x}}_k$ (assuming that first observation of landmark (i) is at time k). Choosing any other value would lead to delay the initialization since we can not guarantee that the angles will be close enough.

Finally, we can notice that the values chosen for $\tilde{x}_{(i)}$, $\tilde{y}_{(i)}$ and $\tilde{z}(i)$ are valid for one iteration of the SAM algoritnm. Indeed, imagine that a part of the estimated trajectory has changed significantly after several time steps (because of a loop closure for example). The values of the fixed coordinates have to be update for the next step in the same way than μ^x. Reader should note that this is not an “ estimation ” (in a filter point of view) of $\tilde{x}_{(i)}, \tilde{y}_{(i)}$ and $\tilde{z}_{(i)}$ but a shift of equilibrium point in order to optimize the validity of the linearization. We do not compute the jacobian of $\mathbf{h}$ with respect to these fixed components.

3.4.2 Discussion

The parametrization presented in the previous paragraph takes advantage of the inverse depth representation but remains minimal since only 3 parameters are estimated. Using a minimal representation reduce the risk of “ falling in local minima ”. Furthermore, we can show that keeping the six variables in the estimation would lead to severe inconsistencies in the case of SAM. Assume for example that landmark (i) was first observed at time k and that we use the six variables in $\mathbf{m}_{(i)}$. At time k, the three first variables of $\mathbf{m}_{(i)}$ are fully correlated with $\mathbf{r}_k$. Let us examine what happens for each new observation of $\mathbf{m}_{(i)}$ for $t > k$:



(a)

Figure 2: Bayesian network in the case of 2 landmarks. Black links indicates normal links of the network. The red links indicates spurious links due to over-parametrization for the EKF case (they are not present with the SAM).

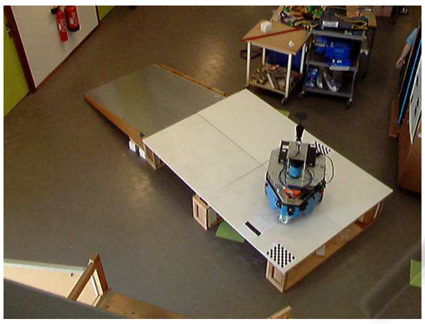


Figure 3: Photo of the experiment.

1. First, the observation will give information about the current variable $\mathbf{x}_k$ and $\mathbf{m}_{(i)}$. This gain of information implicitly improves the estimation of all the other variables of the filter by inference, including $\mathbf{r}_{(k)}$
2. Then, we should remember that the first 3 components of $\mathbf{m}_{(i)}$ are fully correlated with $\mathbf{r}_k$ which implies that the information added on $x_{(i)}, y_{(i)}$ and $z(i)$ are directly propagated up to $\mathbf{r}_k$.

The second source of information gathered by $\mathbf{r}_k$ is spurious information. In fact, the full inverse depth parametrization adds shorcuts in the Bayesian Graph associated to $\mathbf{x}$ and $\mathbf{m}$ (Figure 2). With this model, the measurement of landmark (i) depends directly on the current robot pose, the landmark and the first position of observation. **The second hypothesis given in paragraph 2.1 is not satisfied**. In practice, such implementation of the SAM algorithm will lead to divergence.

Finally, an intuitive way to understand the theoretical differences between the EKF formulation and the minimal SAM representation is to examine the basic properties of the measurement function:

1. in the case of EKF, $\mathbf{h}$ is a function of $\mathbf{r}_k$, $\mathbf{x}_t$ and $\mathbf{m}_{(i)}$. So, we have $\mathbf{z}_t = \mathbf{h}(\mathbf{r}_k, \mathbf{x}_t, \mathbf{m}_{(i)})$,
2. in the case of minimal SAM representation, $\mathbf{h}$ is a function of $\mathbf{x}_t$ and $\mathbf{m}_{(i)}$ parameterized by $\mu_k^{\mathbf{x}}$. So, we have $\mathbf{z}_t = \mathbf{h}_{\mu_k^{\mathbf{x}}}(\mathbf{x}_t, \mathbf{m}_{(i)})$.

4 COMPARISON WITH THE EKF

In this section, we propose to compare some aspects of the EKF-SLAM with the SAM. First, it is now well known that the EKF algorithm is inconsistent. It was shown in simulations by Tim Bailey (Bailey et al., 2006a). Moreover, Simon Julier made a proof of the inconsistency in the case of a static Robot (Julier and Uhlmann, 2001).

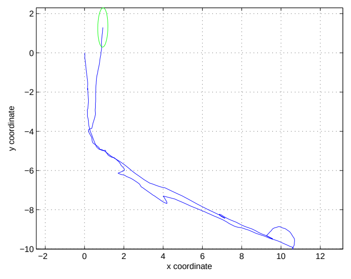
More recently Huang shown the general inconsistency in the case of 3 degrees of freedom SLAM by an analysis of observability ((Huang et al., 2008)). Indeed, an observability of the 3DOF-SLAM shows that the system is unobservable and that the observability matrix has a nullspace of dimension 3. Intuitively, it corresponds to a rigid motion of the whole system (2 translations and one rotation). However, the authors of (Huang et al., 2008) shown that the observability matrix associated to the system linearized by the EKF has a nullspace of dimension 2. This corresponds to spurious information gain in rotation.

They also present an other filter derived from the EKF (the *The First-Estimates Jacobian EKF SLAM (FEJ-EKF)*) which *does not update the linearization points of the landmarks*. This new filter respects the observabilty nullspace dimension. So, Huang claims that the root of the EKF inconsistency is the updates of equilibrium points. **We can notice that the formulation of the SAM algorithm guarantees that the linearization points for the landmarks are the same for all the observations**. In consequence, we can guess that the good properties of the FEJ-EKF are kept by the SAM algorithm.

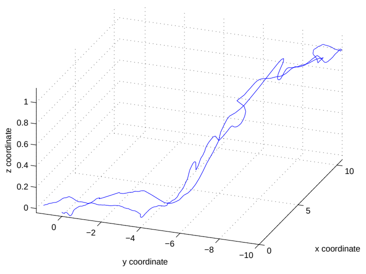
5 RESULTS

5.1 Experimental Testbed

Both algorithms were tested with true data acquired by a mobile robot in an indoor environment. The trajectory length is about 15m. **All the results were obtained in the case of pure bearing-only SLAM (no odometry).**



(a)

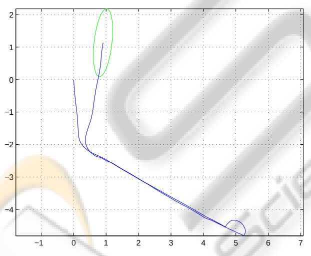


(b)

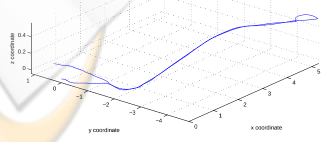
Figure 4: Trajectory provided by the EKF-SLAM.

In order to test the capability of the algorithms to estimate the 6 DOF of the robot, we made a plateform at 35cm high from the ground (Figure 3). To access to this plateform, the robot is driven on a 180cm long ramp. This gives us a ground truth concerning the angle of the trajectory of the robot during its “ climb ”:

$$\gamma = \arcsin(35/180) \approx 11.2deg \tag{9}$$



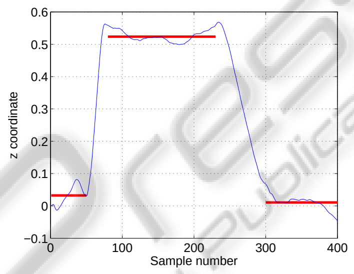
(a)



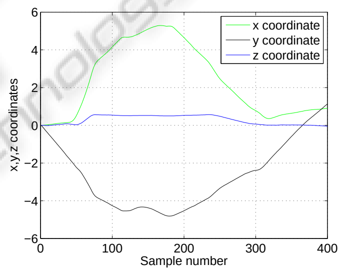
(b)

Figure 5: Trajectory provided by the SAM algorithm.

Furthermore, the robot is submitted to high changes in rotation velocity. Particulary, the robot makes a turn just before climbing on the plateform. Then, the robot rotates on itself (without translation) once it is on the plateform. Some pictures of the experiment are proposed on Figure 3. As the camera is not mounted on the center of rotation of the robot, we expect to see a semicircle in the trajectory. The end of the trajectory consists in going down to the ground and moving in the plane. We also used the fact that the robot came back close to the first position to force a loop closure (the loop closure correspond to the data association shown on Fig 1). Both algorithms are tested with exactly the same data (the image processing is done off-line). Finally, even if we do not



(a)

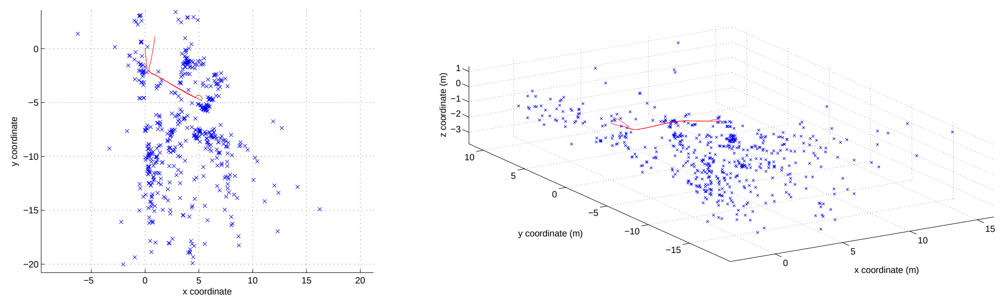


(b)

Figure 6: SAM solution — (a) z coordinate – (b) x, y and z coordinates: we see the relevance of z with respect to the main scale (x and y).

have a ground truth on all the trajectory, we have 4 parameters which can be tested:

1. The robot is on the ground during the first part of the trajectory. So, we expect the z coordinate of the trajectory to be zero,
2. The angle of the ramp is known. We can use it as a ground truth,
3. After the climb, the robot movement is parallel to the ground. So, we expect the z coordinate to be a constant,
4. The robot is on the ground during the last part of the trajectory.



(a) (b)

Figure 7: Map provided by the SAM algorithm. The trajectory is also plotted in red.

5.2 Results on the EKF

The trajectory given by the EKF is given on Figure 4. The trajectory starts from the origin. The green ellipsis on Figure 4a shows the uncertainty of the last robot pose. We can see that there are many discontinuities in the trajectory at the end due to the loop closure. This is due to the fact that when the robot observe an old landmark, the drift accumulated force the filter to make a discontinuity in the solution. Finally, even if the trajectory looks fine (we see the different heights and the semicircle), results of the EKF are not much impresive.

5.3 Results on the SAM

The results provided by the SAM algorithm are given on Figure 5 to 7. They look like much interesting than the EKF-SLAM results. Indeed, the estimation of the trajectory is very smooth comparing to the one provided by the EKF. Moreover, we can see that the uncertainty ellipsis is bigger in the case of the SAM algorithm. This is certainly a manifestation of the EKF inconsistency: the EKF is over-confident.

Moreover, we were able to compute an approximation of the angle γ with the trajectory by using the coordinates at the beginning and at the end of the climb. We got 10.1deg (the ground truth is 11.2deg). We have just an **error of 1.1deg**. This confirms the excellent results provided by the SAM algorithm.

We also checked the relevance of the z coordinate (Figure 6). It appears that the mean value during the first part of the trajectory is 0.03m with a standard deviation of 0.03m (first red bar on Figure 6a). During the second part of the trajectory (after the climb), the mean value of z is 0.52m (standard deviation: 0.015m). Finally, the mean value of z during the last part of the trajectory is 0.01 (standard deviation: 0.02m). These results are good: we get the two parts were z is close from 0 and see a near constant z during the second part of the trajectory. Then, z values are scaled with the values of x and y on Figure 6b: we can see that displacements in z are much smaller than displacement in x,y. However, the shape of the z is well estimated, which shows the good precision of the algorithm. We got similar results with the EKF algorithm (but with a higher level of noise).

Finally, the map provided by the SAM algorithm is given on Figure 7. It is quite difficult to interpret since we do not have any ground truth (the map is made by interest points selected during the video sequence). Nevertheless, we can see on the left part a wall (close to the zero vertical axis) which seems to be coherent.

6 CONCLUSIONS

In this paper, we proposed an accurate method to solve the monocular SLAM problem. To do it, we used the inverse depth representation for landmarks and the SAM algorithm for the filtering part. However, we do not use the whole inverse depth parametrization as proposed by the original authors. Indeed, we decide to fix the three first components of the representation by using the linearization points provided by the SAM algorithm. Thus, we solved the over-parametrization problem introduced by the EKF-version of the algorithm. In consequence, two origins of inconsitencies are avoided: the one due to the inherent inconsistency of EKF, and the one due to the over-parametrization. This makes our algorithm more robust than the EKF proposed by *Civera et al*.

We shown that we had a very good estimation without any odometry. First, we recovered the angle of the ramp if good precision. Then the z coordinate estimation was quite relevant in spite of the small vertical displacements comparing to the x and y coordinates.

Our future work will focus on optimizing the algorithm and implementing a strategy to switch from inverse depth coordinates to classical euclidian coordinates. Indeed, inverse depth parametrization is very relevant when parallax is small. Then, classical coordinates can be used. The authors of (Civera et al., 2007) propose a test to switch from inverse depth coordinates to global ones. We will focus on adapting this strategy to our algorithm.

REFERENCES

Bailey, T., Nieto, J., Guivant, J., Stevens, M., and Nebot, E. (2006a). Consistency of the ekf-slam algorithm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Bailey, T., Nieto, J., and Nebot, E. (2006b). Consistency of the fastslam algorithm. In *IEEE International Conference on Robotics and Automation*.

Barreto, J. P. (2003). *General Central Projection Systems, modeling, calibration and visual servoing*. PhD thesis, Department of electrical and computer engineering.

Civera, J., Davion, A. J., and Montiel, J. M. (2008). Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions On Robotics*, 24(5).

Civera, J., Davison, A. J., and Montiel, J. (2007). Inverse depth to depth conversion for monocular slam. In *International Conference on Robotics and Automation (ICRA)*.

Dellaert, F. and Kaess, M. (2006). Square Root SAM: Simultaneous Location and Mapping via Square Root Information Smoothing. *Journal of Robotics Research*.

Hadj-Abdelkader, H., Malis, E., and Rives, P. (2008). Spherical Image Processing for Accurate Odometry with Omnidirectional Cameras. In *The Eighth Workshop on Omnidirectional Vision*.

Huang, G., Mourikis, A., , and Roumeliotis, S. (2008). Analysis and improvement of the consistency of extended kalman filter based slam. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA)*.

Julier, S. J. and Uhlmann, J. K. (2001). A counter example to the theory of simultaneous localization and map building. In *International Conference on Robotics and Automation (ICRA)*.

Mei, C. (2007). *Laser-Augmented Omnidirectional Vision for 3D Localisation and Mapping*. PhD thesis, Ecole des Mines de Paris, INRIA Sophia Antipolis.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2003). FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. In *International Joint Conference on Artificial Intelligence*, pages 1151–1156.

Smith, R. and Cheeseman, P. (1987). On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68.

Thrun, S. and Montemerlo, M. (2006). The GraphSLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *The International Journal of Robotics Research*, 25(5-6):403–429.