2014 13th International Conference on Control, Automation, Robotics & Vision
Marina Bay Sands, Singapore, 10-12th December 2014 (ICARCV 2014)

Guaranteed Simultaneous Localization and Mapping Algorithm using Interval Analysis

Bastien Vincke, Alain Lambert and Abdelhafid Elouardi
Institut D'Electronique Fondamentale
Université Paris-Sud, 91405, Orsay, FRANCE
Institut Fondamentale d'Electronique
Centre National de la Recherche Scientifique(CNRS) 8622, 91405, Orsay, FRANCE
Email: {bastien.vincke, alain.lambert, abdelhafid.elouardi}@u-psud.fr

***Abstract*—To increase the autonomy of robots, it is necessary to have a precise and guaranteed localization. We propose a SLAM algorithm based on interval analysis and constraints propagation. The problem is casted into a constraint satisfaction which is solved in a guaranteed way via Interval Analysis. We define a new bounded landmark parameterization and an initialization method for monocular camera. Finally, we introduce the post-localization process which improves the localization accuracy using the future observations. Both simulations and experiments show guaranteed and consistent results of CP-SLAM.**

I. INTRODUCTION

Most SLAM approaches use probabilistic estimation algorithms like EKF-SLAM [4] or FastSLAM [14]. The good quality of the obtained results is incontestable. Nevertheless these probabilistic algorithms have a common drawback: they present some consistency problems that have not been solved yet. This drawback can be overcame using Interval Analysis (IA) methods rather than the probabilistic one. Indeed IA methods does not suffer from linearization errors (like Kalman based methods) or quantification errors (like particle based methods). Furthermore, IA guarantees, by its design, the calculations and IA methods does not suffer from biased measurements. Thus IA provide guaranteed and consistent results. Such advantages have been highlighted in localization issues [12], [18] as well as in SLAM issues [7], [9].

The first SLAM algorithm based on IA was introduced by Di Marco and al. [3] in 2001. The robot localization was modeled by a single three-dimensional box (x, y and θ) whose volume evolved according to the proprioceptive data (odometers) or exteroceptive data (range and bearing sensors). Landmarks were modeled by parallelotopes (generalized parallelograms). This work was extended in 2004 by adding a matching step [2]. Both 2001 and 2004 papers shows consistent experimental results [3], [2].

In 2003, Drocourt and al. [5] proposed a new SLAM algorithm using IA and based on SIVIA (Set Inversion Via Interval Analysis) [10]. It uses odometers and a stereoscopic omnidirectional vision sensor which provides a 360° view of the scene surrounding the robot. The robot pose was represented by a three-dimensional subpaving. The robot's localization is realized thanks to the SIVIA algorithm. The landmark's localization updating task is realized by intersecting the previous landmark's subpaving with the new one. Whereas, in 2005, Porta [17] proposed to update a landmark localization by using the constraint linking the robot localization and the landmark (the constraint is a 3 axis translation and a 3 axis rotation).

In 2006, Jaulin [7], [8] has presented the first SLAM algorithm based on constraint propagation (CP). Experiment was realized with an underwater robot embedding a GPS, a sonar, a loch Doppler, a gyrocompass and a barometer. The robot localization was represented by 6-D boxes (x,y,z,φ,θ,ψ) (rather than 3-D boxes as in Porta's work). The initial robot's localization was achieved using GPS measurements. Next, the prediction step was processed using a bounded displacement model with the gyrocompass and the loch Doppler. A human operator pre-processed the sonar data used in the estimation step. Both the displacement and the observation model defined a set of constraints that have been solved using constraint propagation. Experimental results have shown a consistent localization.

We follow the approach presented in [7], [11], [8] but using different sensors: our robot embedded two odometers and a monocular front camera (rather than a sonar [7], [8]). Jointly with the use of a monocular camera, we propose a new bounded landmark parametrization. Furthermore, our approach does not require data pre-processing by a human operator as in [7], [8]. Finally, our CP-SLAM algorithm is validated with both simulations and experimental results while similar works were only tested in simulated environment.

The paper is structured as follows: Section II presents the basics of the Interval Analysis and Constraint Propagation for the SLAM algorithm. Section III details the CP-SLAM. Simulations are presented in Section IV followed by experimental results in section V.

II. BASICS OF INTERVAL ANALYSIS AND CONSTRAINT PROPAGATION FOR SLAM

Interval analysis [16] was introduced in the sixties to solve the problem of approximations made during calculations. The key idea of interval analysis is to represent numbers by intervals which include the real values. An interval is represented using [ ]. For example: $[x] = [\underline{x}, \overline{x}] = \{x \in \mathbb{R} | \underline{x} \leq x \leq \overline{x}\}$ is the interval which is guaranteed to contained the real value x. A set of rules have been defined to perform all the usual mathematical operations on intervals.

978-1-4799-5199-4/14/$31.00 ©2014 IEEE

A. Interval Analysis

1) Overview: IA proposes to represent a solution of a problem by an interval in which the real solution is guaranteed to be contained. IA provides a set of rules to calculate with intervals $[x] = [\underline{x}, \overline{x}] \subset \mathbb{R}$ where $\underline{x}$ and $\overline{x}$ are respectively the minimum and the maximum of $[x]$. The width of an interval is $w[x] = \overline{x} - \underline{x}$. Arithmetical operations ($+$, $-$, * and $/$) and standard mathematical functions readily extend to intervals. For example,

$$\begin{aligned} [1,2] + [3,4] &= [4,6] \\ \ln([1,e]) &= [0,1]. \end{aligned} \tag{1}$$

2) Inclusion function: The notion of *inclusion function* is one of the most important provided by IA [10]. For any function $f : \mathcal{D} \subset \mathbb{R} \rightarrow \mathbb{R}$ defined as a combination of arithmetical operators and elementary functions, interval analysis makes it possible to build inclusion functions $f_{[]}$ satisfying:

$$\forall [x] \subset \mathcal{D}, f([x]) \subset f_{[]}([x]), \tag{2}$$

where $f([x])$ denotes the set of all values taken by f over $[x]$.

The simpliest way to obtain an inclusion function is to replace all real variables by interval ones and all real-valued operators or elementary functions by their interval counterparts. The *natural inclusion function* is then obtained. For example, the natural inclusion function for:

$$f(x) = x^2 - x + 1 \tag{3}$$

is:

$$f_{[]}([x]) = [x]^2 - [x] + 1. \tag{4}$$

It is then possible to enclose the set of all values taken by a function over a given interval into a computable image interval.

B. Constraints propagation

Constraint propagation techniques were proposed in the seventies by [19] and mainly used in artificial intelligence [13]. Constraints propagation was merged with IA in the late 80's [1].

1) Constraints: A constraint C_i between the intervals $[z_1], ..., [z_n]$ is defined as:

$$C_i([z_1], ..., [z_n]) = 0, \quad i = 1, ..., m \tag{5}$$

2) Constraints Satisfaction Problem: The goal of a Constraints Satisfaction Problem (CSP) [10] is to obtain the smallest box $[\mathbf{z}]$ verifying a set of constraints. The solution of a CSP will be the smallest box satisfying all the constraints.

A CSP is solved by making successive contractions of the initial box which ensures the less possible pessimistic result. The solution box contains every points satisfying the set of constraint. Such contractions can be realized by a Forward Backward Propagation (FBP) technique [10], [6].
Constraints are written under the form $f(x) = y$ where x and y can be measured. Then, the intervals are propagated from x to y in a first step (Forward propagation) and, in a second step, the intervals are propagated from y to x (Backward propagation).

For example, we add a second constraints to our previous example:

$$\begin{cases} z = x + \ln(y) \\ y = z^2 \end{cases} \tag{6}$$

Then, two phases can contract the intervals:

1) Forward propagation which reduces the left terms of Eq. (6)

$$[z] \in [z] \bigcap ([x] + ln([y])) \tag{7}$$

$$[y] \in [y] \bigcap [z]^2 \tag{8}$$

2) Backward propagation which reduces the right terms of Eq. (6)

$$[x] \in [x] \bigcap ([z] - ln([y])) \tag{9}$$

$$[y] \in [y] \bigcap exp([z] - [x]) \tag{10}$$

$$[z] \in [z] \bigcap \sqrt{[y]} \tag{11}$$

For complex CSP, it can be necessary to make successive forward/backward steps until there is no more interval contraction.

III. The CP-SLAM

This section presents our Constraints Propagation SLAM algorithm designed for a wheeled robot embedding two odometers (located on each rear wheel) and a front camera (see Fig 1). Our robot embbeds a Kinect which can give RGB-D images but we focus on the use of the RGB image only.

A. Robot Pose Representation

The imprecise pose of the robot is represented by a box $[\mathbf{x}] = ([x][z][\theta])^{\mathrm{T}}$ with x, z the robot position in the global frame (define by the initial robot pose) and θ the rotation only around y-axis. Initialy, the robot is oriented along the z axis. We suppose that the robot is moving on a plane $(\overrightarrow{x}, \overrightarrow{z})$.

B. Landmark Representation

In probabilistic SLAM algorithms, landmarks are often represented using the inverse depth parametrization [15]. Effectively, this parametrization can cope with features over a huge range of depths and produces measurement equations with a higher degree of linearity than simple xyz parametrization. Nevertheless, a Gaussian is not an efficient way to represent the linear depth uncertainty of a camera. Ideally, the landmark uncertainty should be represented by an infinite cone including the observed landmark. Contrarily to probabilistic SLAM algorithms, interval analysis can cope with this model in an easy and efficient way with an interval vector:

$$[\mathbf{y}_i] = ([x_i], [z_i], [\varphi_i], [\theta_i], [d_i]) \tag{12}$$

with:

• $[x_i], [z_i]$: the intervals including the camera pose during the first observation of the landmark. We suppose that the robot is moving on a plane ($y_i = 0$).

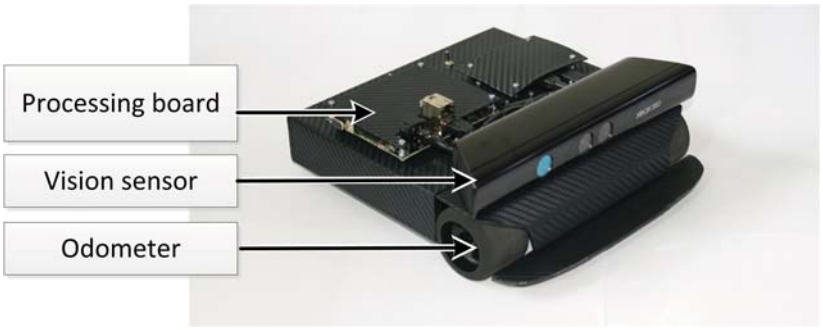

Figure 1. MiniB, our experimental platform

• $[\varphi_i], [\theta_i]$: the intervals including the elevation and the azimuth of the vector pointing to the landmark $\overrightarrow{m}(\varphi_i, \theta_i)$.

• $[d_i]$: the interval representing the distance of the landmark along the direction vector.

C. Landmark Initialization

The landmark parameters are initialized as follows:

• $[x_i] = [x_{camera}]$ and $[z_i] = [z_{camera}]$: both intervals are defined by the camera pose.

• $[\theta_i] = [\theta] + [u_{angle}]$ and $[\varphi_i] = [v_{angle}]$: θ is the robot angle which is also the camera angle. Both the orientation angle $[u_{angle}]$ and the elevation angle $[v_{angle}]$ of the landmark in the robot frame are deduced from the pinhole model:

$$[u_{angle}] = -\arctan_{[]}\left(\frac{[c_u] - [u_{obs}]}{[f][k_u]}\right) \quad (13)$$

$$[v_{angle}] = -\arctan_{[]}\left(\frac{[c_v] - [v_{obs}]}{[f][k_v] cos_{[]}([u_{angle}])}\right) \quad (14)$$

where u_{obs}, v_{obs} are the landmark's pixel on the camera plane, f is the focal length, (k_u, k_v) are the pixel size and (c_u, c_v) is the principal point. All these parameters are obtained thanks to a classical calibration process. $[c_u], [f], [k_u], [c_v], [k_v]$ are defined from the calibrated value with added imprecision.

• $[d_i] =]0; +\infty[$: the unknown distance to the landmark.

D. Features Matching

The feature extraction is realized thanks to the Speeded Up Robust Features (SURF) algorithm. SURF is a performant scale and rotation invariant interest point detector and descriptor. The features are matched by:

- projecting the landmark cone on the image plane (probabilistic algorithms proceed in the same way by projecting an ellipsoïd). This projection define a searching area ($[u_{search}]$,$[v_{search}]$):

$$\begin{pmatrix} [u_{search}] \\ [v_{search}] \end{pmatrix} = \begin{pmatrix} [c_u] - [f][k_u]\frac{[\mathbf{L}_{i,x}]}{[\mathbf{L}_{i,z}]} \\ [c_v] - [f][k_v]\frac{[\mathbf{L}_{i,y}]}{[\mathbf{L}_{i,z}]} \end{pmatrix} \quad (15)$$

With $[\mathbf{L}_{i,x}]$, $[\mathbf{L}_{i,y}]$, $[\mathbf{L}_{i,z}]$ the landmark localization coordinates in the camera frame (defined in Eq. 19) and $[c_u]$, $[f]$, $[k_u]$, $[c_v]$, $[k_v]$ the camera parameters (defined in Eq. 14).

- selecting, in the projected area, an interest point ($[u_{i,obs}]$, $[v_{i,obs}]$) which has the closest SURF-descriptor to the landmark descriptor using the euclidean distance.

E. State Estimation

The estimation uses two sets of constraints deduced from the displacement model and the observation model. It leads to prediction and correction steps.

1) Prediction Step: The prediction process consists in moving the localization box [12]:

$$[\mathbf{x}_k] = \mathbf{f}_{[]}([\mathbf{x}_{k-1}], [\mathbf{u}_k]) = \begin{cases} [x_{k-1}] + [\delta s_k]\cos([\theta_{k-1}] + \frac{[\delta\theta_k]}{2}) \\ [z_{k-1}] + [\delta s_k]\sin([\theta_{k-1}] + \frac{[\delta\theta_k]}{2}) \\ [\theta_{k-1}] + [\delta\theta_k] \end{cases} \quad (16)$$

where [ ] are intervals including the real values. $[\mathbf{u}_k] = ([\delta s_k]\ [\delta\theta_k])^{\mathrm{T}}$ is the input vector. δs_k represents the elementary displacement of the vehicle and $\delta\theta_k$ represents the elementary rotation. Both δs_k and $\delta\theta_k$ are measured between the instants $k-1$ and k. δs_k and $\delta\theta_k$ are computed using:

$$\begin{matrix} [\delta s_k] = \frac{[\pi]([w_l][\delta p_l] + [w_r][\delta p_r])}{P} \\ [\delta\theta_k] = \frac{[\pi]([w_l]([\delta p_l]) - [w_r]([\delta p_r]))}{2\cdot[e]\cdot P} \end{matrix} \quad (17)$$

where:

- w_r and w_l: the radius of the right and left wheels.
- δp_i: the number of steps measured by the odometers between times k and $k-1$, with $i \in \{r, l\}$ (r=right, l=left).
- P: the odometer resolution,
- e: the half length of the rear axle.

The maximum error of an odometer is one step (one step more or one step less). Consequently the real value of the displacement of a non sliding wheel can be bounded by $[\delta p] = [\delta p - 1, \delta p + 1]$ with δp the number of measured steps. When considering sliding, the movement of a sliding wheel can be deduced from a non-sliding one by adding a sliding noise $[\varepsilon_p] : [\delta p] = [\delta p - 1 - \varepsilon_p, \delta p + 1 + \varepsilon_p]$.

The displacement model (Eq. 16 and 17) defines a set of 5 constraints between the intervals $[x_k]$, $[x_{k-1}]$, $[z_k]$, $[z_{k-1}]$, $[\theta_{k-1}]$, $[\theta_k]$, $[\delta s_k]$, $[\delta\theta_k]$.

2) Correction Step: The bounded landmark pose of the i landmark in the global frame ($[\mathbf{L}_i]$) is:

$$[\mathbf{L}_i] = \begin{pmatrix} [x_i] \\ 0 \\ [z_i] \end{pmatrix} + [d_i]\mathbf{m}_{[]}([\theta_i], [\varphi_i]) \quad (18)$$

with $\mathbf{m}_{[]}([\theta_i],[\varphi_i])$ the vector pointing from the camera to the landmark $\mathbf{L}_i$.

Thus, $[d_i]\mathbf{m}_{[]}([\theta_i],[\varphi_i])$ is a box encompassing the truncated cone introduced in Sec. III-B.

The bounded landmark pose in the camera frame is:

$$[\mathbf{L}_i^c] \quad = \quad R_{[],rob}([\mathbf{L}_i] - T_{[],rob}) \tag{19}$$

where $R_{[],rob}$ is the rotation matrix between the coordinates of the global and the robot frame and $T_{[],rob} = (\ [x] \quad 0 \quad [z]\ )^T$ is the robot position.

The camera observe the projection $[\mathbf{L}_i^c]$ of the i^{th} landmark on the image plane:

$$[\mathbf{h}_i] = \begin{pmatrix} [u_{i,obs}] \\ [v_{i,obs}] \end{pmatrix} = \begin{pmatrix} [c_u] - [f][k_u]\frac{[L_{i,x}^c]}{[L_{i,z}^c]} \\ [c_v] - [f][k_v]\frac{[L_{i,y}^c]}{[L_{i,z}^c]} \end{pmatrix} \tag{20}$$

where c_u, f, k_u, c_v, k_v are the camera parameters (defined in Eq. 14).

A set of 25 constraints, for each observed landmark, at time k/k is obtained from Eq. (20) between the intervals $[x]$, $[z]$, $[\theta]$, $[x_i]$, $[z_i]$, $[\theta_i]$, $[\varphi_i]$, $[d_i]$, $[u_{i,obs}]$,$[v_{i,obs}]$. In the same way than with the prediction step, this set of constraints lead to forward/backward equations.

3) Post Localization: CP-SLAM uses a set of constraints which is build along the experimentation. At each step, it added new constraints into the Constraints Satifactions Problem. It include constraints based on the prediction model and constraint based on each observation. Using the Forward/Backward algorithms, all the intervals will be contracted. In fact, the localization boxes, corresponding to each step, can be improves using new observation. The localization improvement due to the new observation will be propagated to the past localization. Consequently, we define two differents localization: the real time localization which correspond to the current localization at each step and the post localization which correspond to the best localization using the entire experimentation.

IV. Simulations

We have used simulations to thoroughly test our CP-SLAM algorithm with an exact reference. The simulated environment is a $15\times15m^2$ area including 300 landmarks. Simulation are made using known data associations between observations and previous landmarks observations. For each simulation, we have verified the algorithm consistency thanks to the interval error [12] and we have studied the volume of the localization box ($V = (\overline{x}-\underline{x})(\overline{z}-\underline{z})(\overline{\theta}-\underline{\theta})$). The interval error of an estimated state a is defined by $[\underline{a} - a_{ref}, \overline{a} - a_{ref}]$. Consequently, a filter exhibits precise and consistent results if its corridor is thin and if it always includes the zero value (it means that the filter imprecision embraces the reference).

A. Stability Tests

If the robot made the same path several times, the localization results must be more accurate in the second lap than those of the first lap. In fact, in the second lap the robot have an already build map in which it could localize itself. Moreover, during all the experimentation, the algorithm must give consistent results. In this simulation, the robot made 3 circles (a diameter of 6m) in the defined test-environment.
For this simulation, we add only white gaussian noises (3σ=0.25mm) on the wheel radius ($r = 4cm$) and white gaussian noises (3σ=0.5mm) on the length of the rear axle ($e = 20cm$). Observations noises are set to 3σ=3 pixels. We bounded each noises at 3σ because CP-SLAM can't deal with unbounded noises. At each simulation step, real parameters are randomly computed in these intervals following white gaussian noise. At each simulation step, corresponding to a displacement of 6 cm, the true robot's localization (a_{ref}) is known.

Fig. 2 represents the obtained results of CP-SLAM in the case of gaussian noises for the real-time and post localization. We can notice that the process is stabilized: the imprecision on the robot's localization remains stable despite the number of laps. The post localization provide a similar localization during the 3 laps, it uses the same full updated map. CP-SLAM (real time and post localization) is consistent during the entire simulation: it always embrase the zero value. Fig. 3 represents the volume V of the localization box, we can note the convergence of the localization process: the volume of the real-time localization box is smaller in the second lap than those of the first lap. The third lap has approximately the same localization results as the second lap: the reobservation of the landmarks does not give a new information because the observation has already been done in the past (the robot follows exactly the same path during the three laps). The algorithm cannot improve the map anymore. To improve the map, the robot needs to observe the landmarks from different view points. For the post localization, the localization are similar for the three laps.

B. Convergence Tests

Fig. 4 represents the obtained interval errors in the case of a random path in our test envrionment. CPSLAM is always consistent during the experiment. Moreover, globally, the volume of the CP-SLAM real-time localization box decreases (Fig. 5): its volume is never over 1.5 m*m*rad (which is the volume at 460th step just before the first loop closure). However, the volume still depends on the robot path. The post localization grealty improve the localization results. Moreover, the results are still consistent.

Fig. 6 represents the map created by the CP-SLAM. The CP-SLAM map is consistent: the real landmarks localization (+) are included in the landmark uncertainty. Real localization landmarks (+) are not included in the small ellipse. CP-SLAM produce a consistant map. CP-SLAM provides less accurate localization results than EKF-SLAM but the results are guaranteed.

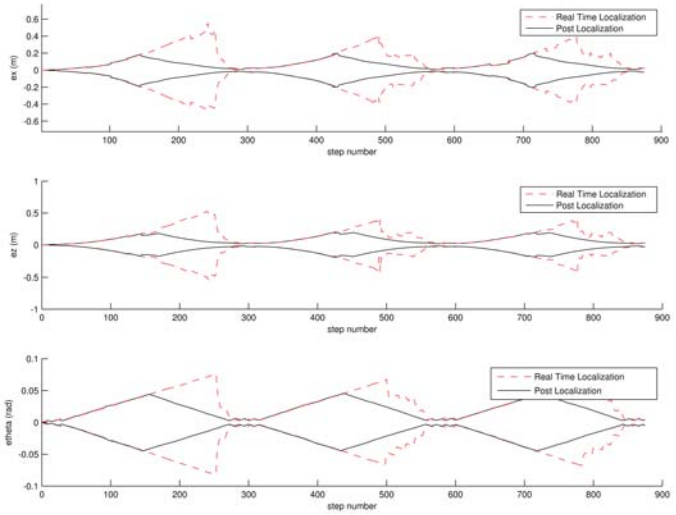

Figure 2. CP-SLAM interval error with gaussian noises

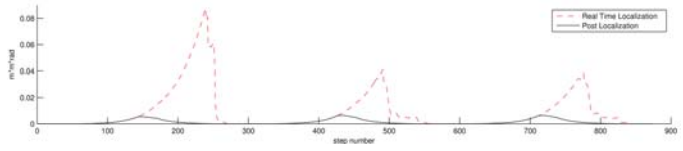

Figure 3. Volume of the CPSLAM box

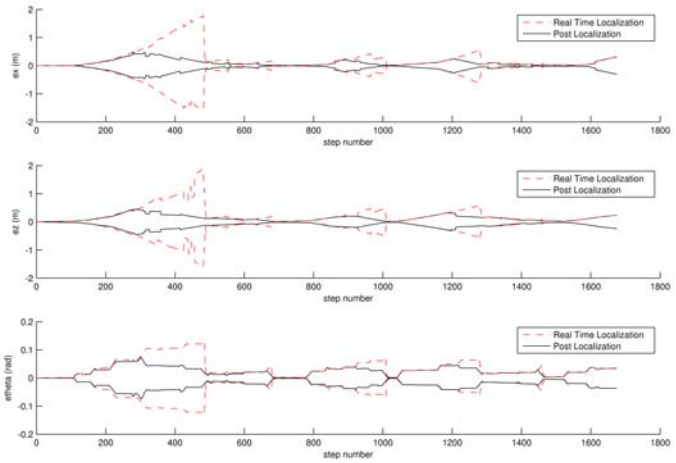

Figure 4. CPSLAM interval error with gaussian noises

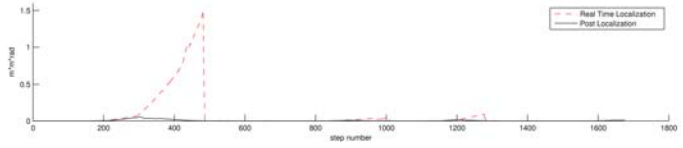

Figure 5. Volume of the CPSLAM box

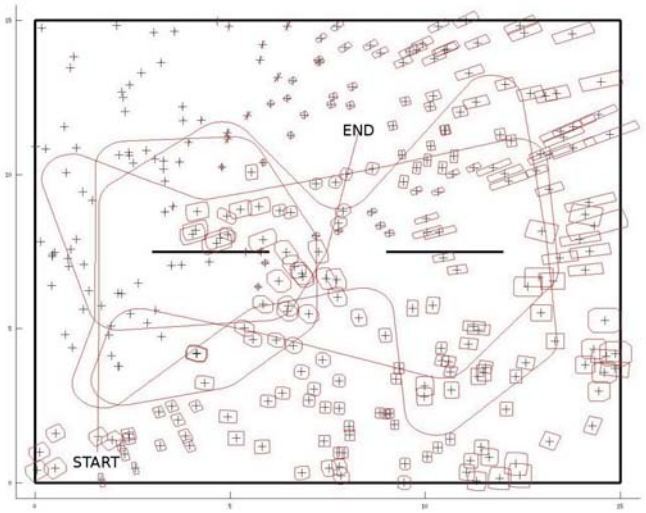

Figure 6. CP-SLAM map results

V. Experimentation

We have realized an experiment in our laboratory with the same noises than those defined for the simulation. Consistency tests are performed thanks to reference points drawn by the robot on the ground.

The figure 7 shows the first trajectory lap reconstructed by the robot using CP-SLAM (each part of the path is about 6 meters). We notice that all boxes include the reference points (noted "R" on figure 7) and thus visually verified the consistency of CP-SLAM.

Results of the interval error of the CP-SLAM, during 5 laps, are represented on Figure 7. CP-SLAM is consistent during the whole experiment: the corridor always includes the zero value.

A loop closure is performed by CP-SLAM at the end of each lap (each eight reference points, see Fig. 7). Once the loop closure is realized, the size of the localization box is about 10 cm long and the orientation uncertainty is 0.1 rad.

Unlike the simulation, the localization of the robot during the second lap is a bit more imprecise than those during the first lap: this is due to the difficulties to match known landmarks. Nevertheless, the localization uncertainty is quickly stabilized: the uncertainty of the third lap is less important than the uncertainty of the second one. We obtained similar results in simulation (see Fig. 7). The growing of the robot uncertainty, between two laps, is mainly due to the heading imprecision. At the experimentation start, the robot heading incertainty was 0.01 rad. After the loop closure, the heading uncertainty was 0.1 rad. We do not show the interval error on the heading because we have not experimentaly measured the reference heading.

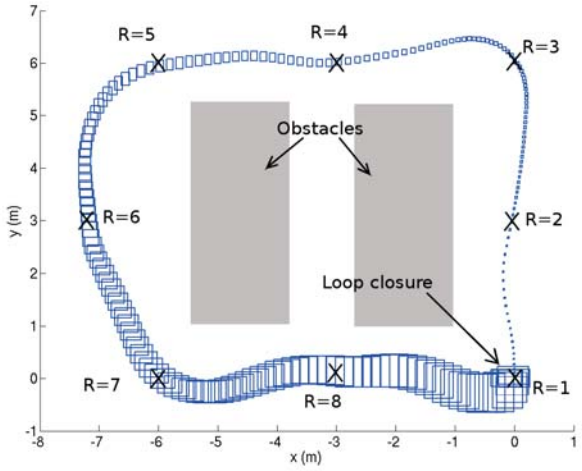

Figure 7. CP-SLAM experimental results

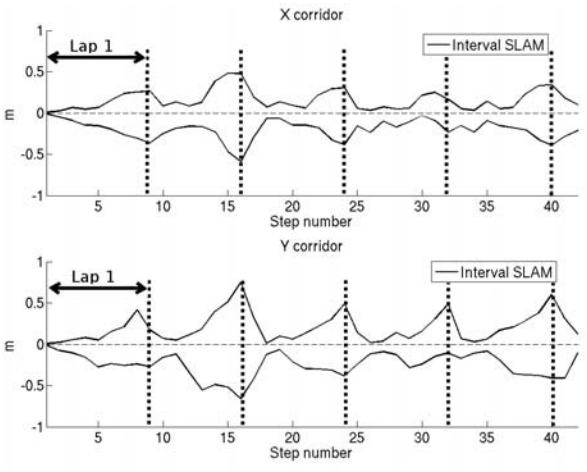

Figure 8. CP-SLAM interval error obtained in experimentation

VI. CONCLUSION

We proposed a Constraints Propagation SLAM algorithm based on Interval Analysis. The algorithm is casted into two steps. The prediction step uses a bounded displacement model and the correction step uses a monocular camera. Landmark parameterization uses interval analysis which is well suited to represent the depth uncertainty of the camera. The algorithm was evaluated using both simulations and experiments. CP-SLAM provided consistent, stable and convergent results. Our algorithm exhibits guaranteed and consistent experimental results. We will then perform a real-time implementation on an embedded system. Indeed, methods based on the interval analysis are time-consuming and require high processing performances. To achieve a real-time implementation, it will be necessary to achieve a compromise between processing time and localization quality.

REFERENCES

[1] J. Cleary. Logical arithmetic. *Future Computing Systems*, 2(2):125–149, 1987.

[2] M. Di Marco, A. Garulli, A. Giannitrapani, and A. Vicino. A set theoretic approach to dynamic robot localization and mapping. *Autonomous robots*, 16(1):23–47, 2004.

[3] M. Di Marco, A. Garulli, S. Lacroix, and A. Vicino. Set membership localization and mapping for autonomous navigation. *International Journal of robust and nonlinear control*, 11(7):709–734, 2001.

[4] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17:229–241, 2001.

[5] C. Drocourt, L. Delahoche, E. Brassart, B. Marhic, and A. Clerentin. Incremental construction of the robot's environmental map using interval analysis. In *Global optimization and constraint satisfaction: second international workshop*, pages 127–141, 2003.

[6] A. Gning and P. Bonnifait. Constraints propagation techniques on real intervals for guaranteed fusion of redundant data. application to the localization of car-like vehicles. *Automatica*, pages 1167–1175, 2006.

[7] L. Jaulin. Localization of an underwater robot using interval constraints propagation. In *International Conference on Principles and Practice of Constraint Programming*, 2006.

[8] L. Jaulin. A nonlinear set-membership approach for the localization and map building of an underwater robot using interval constraint propagation. *IEEE Transaction on Robotics*, 25:88–98, 2009.

[9] L. Jaulin, F. Dabe, A. Bertholom, and M. Legris. A set approach to the simultaneous localization and map building - application to underwater robots. In *International Conference on Informatics in Control, Automation and Robotics*, pages 65–69, 2007.

[10] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag, 2001.

[11] Cyril Joly and Patrick Rives. Bearing-only SLAM: comparison between probabilistic and deterministic methods. Research Report RR-6602, INRIA, 2008.

[12] A. Lambert, D. Gruyer, B. Vincke, and E. Seignez. Experimental vehicle localization by bounded-error state estimation using interval analysis. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1211–1216, 2009.

[13] A.K. Mackworth. Consistency in networks of relations. *Artificial intelligence*, 8(1):99–118, 1977.

[14] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. 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, 2003.

[15] J.M.M. Montiel, J. Civera, and A.J. Davison. Unified inverse depth parametrization for monocular SLAM. In *International Conference on Robotics: Science and Systems*, pages 16–19, 2006.

[16] R.E. Moore. *Methods and Applications of Interval Analysis*. Studies in Applied Mathematics, 1979.

[17] J.M. Porta. CuikSlam: A kinematics-based approach to SLAM. In *International Conference on Robotics and Automation*, pages 2425–2431, 2005.

[18] B. Vincke and A. Lambert. Experimental comparison of bounded-error state estimation and constraints propagation. In *IEEE International Conference on Robotics and Automation*, pages 4724–4729, 2011.

[19] D. Waltz. *Understanding line drawings of scenes with shadows*. 1975.