

Constraint Propagation based Approach for Simultaneous Localization And Mapping of Mobile Robots*

Bastien Vincke

SATIE, Univ Paris-Sud, Université Paris-Saclay

`bastien.vincke@u-psud.fr`

Zhan Wang

Institut Supérieur d'électronique de Paris

`zhan.wang@isep.fr`

Alain Lambert

LRI, Univ Paris-Sud, Université Paris-Saclay

`alain.lambert@u-psud.fr`

Abdelhafid Elouardi

SATIE, Univ Paris-Sud, Université Paris-Saclay

`abdelhafid.elouardi@u-psud.fr`

Abstract

Simultaneous Localization and Mapping (SLAM) problems have been classically solved by probabilistic methods (EKF-SLAM/Fast-SLAM) or Optimization method (Graph-SLAM). A common issue confronted by these methods is the consistency problem. This paper proposes a new method which theoretically relies on interval analysis to solve the SLAM problem. The aim is to achieve consistent solutions. Different from other methods which require probability distribution assumptions on uncertainties, only a soft prerequisite is needed (the errors are bounded). With proposed bounded-error parametric models, we cast the SLAM problem into an Interval Constraint Satisfaction Problem (ICSP), constraint propagation techniques are then applied to search for all feasible solutions, providing a guaranteed rather than a probabilistically defined result. Our method has been validated on both simulations and experimentations. Results demonstrate the capability of the our method in obtaining consistent solutions.

Keywords: consistent localization, interval analysis, constraint propagation, SLAM, bounded-error

*Submitted: January 14, 2019; Revised: July 23, 2019; Accepted: July 23, 2020

AMS subject classifications: 65-00

1 Introduction

Simultaneous Localization and Mapping (SLAM) has been an active topic in robotics literature over the past decades, and there has been a plethora of publications that proposed different solutions [17, 27, 34, 38]. Most of these approaches are classified as probabilistic method as they rely on studying the propagation of probabilistic distributions of the sensor noise and the unknown parameters (e.g., robot pose and landmark position). Some others, on the other hand, do not require any assumptions about the probability distribution of uncertainties, except that they are bounded by real intervals. These approaches are regarded as interval analysis based methods.

It has been well noticed by the research community that probabilistic methods suffer the inconsistency problem [12, 39, 40]. For instance, EKF-SLAM suffers from the linearization of the non-linear models which introduces spurious updates of the landmark estimate [1]. As the landmarks state uncertainty could be optimistic, the consistency is not guaranteed anymore. FastSLAM encounters different problems than those of the EKF-SLAM. Bailey shows that FastSLAM cannot be consistent due to the quantification errors (FastSLAM must have a finite number of particles) and the resampling process (resampling is necessary to focus the algorithm in the highest probability regions) [2]. Moreover, each deleted particle entails a depletion in the historical information that may generate consistency problems. FastSLAM degenerates over time, regardless of the number of particles used or the density of landmarks within the environment, and will always provide optimistic estimates of uncertainty in long-term.

Graph-SLAM is an important step toward more consistent results: they avoid linearization inconsistency by re-linearizing, at each iteration, all observations around the current state of the system. A graph-based SLAM algorithm represents the SLAM problem by a graph and uses a graph optimization method to solve it. The concept of graph-based SLAM has been first introduced by Lu and Milios [26]. It has been then used and developed later in [10], [35]. Although graph-based SLAM algorithms presents more consistent results than EKF-SLAM or FastSLAM, they have the same probabilistic foundation: all these algorithms assume zero mean Gaussian noises.

Constraints propagation algorithm involving interval analysis [30] is alternative and less known method which allows us to solve nonlinear problems in a guaranteed way. Instead of making hypothesis on the probability distribution of sensor noises, interval methods take a soft assumption that all the noise are bounded within known limits. This seems to be a very realistic representation as most sensor manufacturers provide the maximum and minimum possible measurement errors under suitable working conditions. These extreme values of error can then be regarded as the error bounds. Interval Constraint Propagation (ICP) algorithms can be used to recursively propagate such bounded errors by using consistency techniques and systematic searching methods. Contrary to probability methods, interval based methods provides guaranteed sets which enclose the real state, without losing any feasible value.

Interval analysis was introduced in the mobile robotic area decades ago. It was mainly used in outdoor vehicle localization field [8, 22, 23, 37] and under-water robot localization [14] with nonlinear motion and sensor models. The main advantage of interval based localization over Kalman filtering or Bayesian methods is that it guarantees that the position of the robot is in a region. Kalman filtering or Bayesian

methods can only associate a probability to such a region, safety cannot be guaranteed.

The first interval based SLAM algorithm was introduced by Di Marco et al. in 2001 [5], it defined the SLAM problem into four main phrases. This work was extended in 2004 by adding a matching step [6]. A number of follow-up papers [7, 17, 18, 19, 32], offer improved performance on SLAM problem. Drocourt et al. [7] proposed their IA based SLAM using SIVIA (Set Inversion Via Interval Analysis). It combined odometers and a stereoscopic omnidirectional vision sensor of 360° view. The robot pose and landmark's position are represented by subpavings. Simulation results show that the evolution of the subpavings is robust and coherent. Porta [32] applied CUIK (Complete Universal Inverse Kinematics) algorithm to solve the SLAM algorithm. Indoor experimental results with a sonar have shown the soundness of the approach. Jaulin [17] presented his first SLAM algorithm based on CP (Constraint Propagation) algorithm. The principle of his approach is to cast the SLAM problem into a constraint satisfaction problem (CSP) for which interval propagation algorithms are particularly powerful. Experiments [18, 25] were conducted on a under-water robot platform equipped with a GPS, a sonar, a Doppler Velocity Log, a gyrocompass and a depth sensor. Results showed that the algorithm gives a consistent localization.

The aim of the paper is to present an extension of constraint propagation based SLAM algorithm (CP-SLAM): instead of hiring ultrasonic [21], sonar [18, 33] or GPS [13, 23] as exteroceptive sensors, we propose to use a monocular camera. A bounded landmark parametrization and undelayed initialization method for nature landmarks is put forward. The undelayed initialization allows us to directly use all available observations to estimate the robot position. Furthermore, our approach does not require data pre-processing by human operators as in [15, 16]. Finally, the properties of our CP-SLAM algorithm are validated with both simulations and experimental results, a comparison with EKF-SLAM in terms of consistency is also presented.

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

2 Basics of Interval Analysis and Constraint Propagation

Interval analysis [31] was introduced in the sixties to solve the problem of approximations made during numerical calculations. The key idea of interval analysis is to represent numbers by intervals which include the real values. An interval is defined represented by $[\]$. For example: $[x] = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} | \underline{x} \leq x \leq \bar{x}\}$ is the interval which is guaranteed to contain the real value x . A set of rules have been defined to extend all the elementary mathematical operations to intervals.

2.1 Interval Analysis

2.1.1 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 $[x] = [\underline{x}, \bar{x}] \subset \mathbb{R}$ where \underline{x} and \bar{x} are respectively the lower and upper bound of

$[x]$. The width of an interval is $w([x]) = \bar{x} - \underline{x}$. Arithmetical operations (+, −, × and /) and standard mathematical functions readily extend to intervals. For example,

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

2.1.2 Free libraries

The computation with IA is simplified by the use of CXSC, a C++ class library supporting the most commonly needed interval and real operations in a user friendly way. This library allows manipulating intervals as numbers. All basic mathematical functions are implemented in order to accept numbers as well as intervals.

2.1.3 Inclusion function

The definition of *inclusion function* is one of the most important notion provided by IA [31]. 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:

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

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

The simplest 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

$$f(x) = x^2 - x + 1 \quad (3)$$

is:

$$[f]([x]) = [x]^2 - [x] + 1. \quad (4)$$

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

2.2 Constraints propagation

Constraint propagation techniques were proposed in the seventies by [36] and mainly used in artificial intelligence [28]. Constraints propagation was merged with IA in the late 80's [4].

A constraint C_i between the intervals $[z_1], \dots, [z_n]$ is defined as:

$$C_i([z_1], \dots, [z_n]) = 0, \quad i = 1, \dots, m \quad (5)$$

A Constraints Satisfaction Problem (CSP) [20] is defined by a set of constraints. The solution of such a problem is the smallest box or a list of pavings verifying the set of constraints.

A CSP is solved by making successive contractions of the initial box, which guarantees the least possible pessimistic result. The solution box contains every point satisfying the set of constraints. Such contractions may be carried out by Forward/Backward Propagation techniques [9, 20].

Let us assume the following set of constraints:

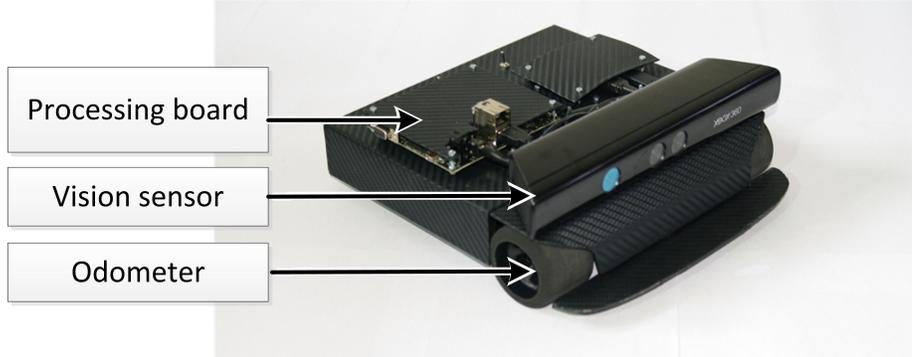


Figure 1: MiniB, our experimental platform

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

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

$$[z] \in [z] \cap ([x] + \ln([y])) \quad (7)$$

$$[y] \in [y] \cap [z]^2 \quad (8)$$

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

$$[x] \in [x] \cap ([z] - \ln([y])) \quad (9)$$

$$[y] \in [y] \cap \exp([z] - [x]) \quad (10)$$

$$[z] \in [z] \cap \sqrt{[y]} \quad (11)$$

Forward and Backward equations obtained from the constraints are computed one after the other (in this example we compute Eq. (7), (8), (9), (10) and finally (11)). When all the constraints have been computed, we restart at the beginning (Eq. (7)). We loop until the intervals are no more contracted or contracted less than a threshold.

3 CP-SLAM

This section presents our Constraints Propagation based SLAM algorithm designed for a wheeled robot embedding two odometers (located on each front wheel) and a front camera. We only use the RGB camera which is located in the middle of the vision sensor (see Fig 1).

3.1 Problem statement

The pose of the robot is represented by a characteristic point that is located midway between the front wheels (this point is also the camera's characteristic point). The imprecise pose of the robot is represented by one single box $[\mathbf{x}] = ([x], [z], [\theta])^T$ with

x , z the robot position in the global frame and θ the rotation around the y -axis (the vertical axis). We suppose that the robot is moving on a plane (\vec{x}, \vec{z}) and hat the global frame is defined by the initial robot pose on the ground.

The robot is localized by odometers and distinguishable landmarks which are features detected by the embedded camera. The feature extraction is performed using the Speeded Up Robust Features (SURF) algorithm [3].

3.1.1 Landmark Representation

Probabilistic SLAM algorithms often represent landmarks with an inverse depth parametrization [29]. This parametrization can cope with features over a wide range of depths and provides measurement equations with a high degree of linearity. Nevertheless, a Gaussian distribution 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 by using interval vectors:

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

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 ($y_i = 0$).
- $[\varphi_i], [\theta_i]$: the intervals including the elevation and the azimuth of the i^{th} vector $\mathbf{m}_i([\theta_i], [\varphi_i])$ pointing to the landmark.
- $[d_i]$: the interval representing the distance of the landmark along the direction vector.

We have designed such a representation as the core of our undelayed initialization. We claim that our undelayed initialization is more efficient than a delayed initialization using a pessimistic 3D landmark box. Indeed, a 3D box defined by the 3 axis of the space is bigger then $[\mathbf{y}_i]$ because it encompasses 3 parameters ($[\varphi_i], [\theta_i], [d_i]$) in a pessimistic way.

3.1.2 Landmark Initialization

The landmark parameters are initialized as follows:

- $[x_i] = [x]$ and $[z_i] = [z]$: both intervals are defined by the robot pose which is also the camera pose.
- $[d_i] = [0, +\infty]$: the unknown distance to the landmark.
- $[\theta_i] = [\theta] + [\theta_i^{RobotF}]$ and $[\varphi_i] = [\varphi_i^{RobotF}]$: θ is the heading which is also the camera angle. In the robot frame (which is also the camera frame), both the orientation angle $[\theta_i^{RobotF}]$ of the landmark and the elevation angle $[\varphi_i^{RobotF}]$ of the landmark are inferred using the classical pinhole model:

$$[\varphi_i^{RobotF}] = -\arctan\left(\frac{[c_u] - [u_{obs}]}{[k_u][L]}\right) \quad (13)$$

$$[\theta_i^{RobotF}] = -\arctan\left(\frac{[c_v] - [v_{obs}]}{[k_v][L] \cos([\varphi_i^{RobotF}])}\right) \quad (14)$$

where (u_{obs}, v_{obs}) are the landmark's pixel on the camera plane, L 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. $[c_u], [k_u], [c_v], [k_v], [L]$ are defined from the calibrated value with added imprecision.

3.2 State Estimation

The estimation of the robot pose uses two sets of constraints inferred from the displacement model and from the observation model. This leads to the prediction and correction steps.

3.2.1 Prediction Step

The prediction process consists in moving the localization box [24]:

$$\begin{pmatrix} [\mathbf{x}_k] \\ [x_k] \\ [z_k] \\ [\theta_k] \end{pmatrix} = \begin{pmatrix} \mathbf{f}([\mathbf{x}_{k-1}], [\mathbf{u}_k]) \\ [x_{k-1}] + [\delta s_k] \cos([\theta_{k-1}] + \frac{[\delta \theta_k]}{2}) \\ [z_{k-1}] + [\delta s_k] \sin([\theta_{k-1}] + \frac{[\delta \theta_k]}{2}) \\ [\theta_{k-1}] + [\delta \theta_k] \end{pmatrix} \quad (15)$$

where $[\]$ are intervals including the real values. $[\mathbf{u}_k] = ([\delta s_k], [\delta \theta_k])^T$ 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 instants $k-1$ and k . δs_k and $\delta \theta_k$ are computed using:

$$\begin{aligned} [\delta s_k] &= \frac{[\pi]([w_l][\delta p_l] + [w_r][\delta p_r])}{P} \\ [\delta \theta_k] &= \frac{[\pi]([w_l][\delta p_l] - [w_r][\delta p_r])}{2 \cdot [e] \cdot P} \end{aligned} \quad (16)$$

with:

- w_i : the radius of the wheel i , with $i \in \{r, l\}$ (r =right, l =left).
- δp_i : the number of steps measured by the odometers between instants $k-1$ and k .
- P : the odometer resolution,.
- e : the half length of the front 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 may 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 inferred from a non-sliding one by adding a sliding noise $[\varepsilon_p]$: $[\tilde{\delta p}] = [\delta p - 1 - \varepsilon_p, \delta p + 1 + \varepsilon_p]$.

The displacement model (Eq. 15) defines a set of 5 constraints among the intervals $[x_k], [x_{k-1}], [z_k], [z_{k-1}], [\theta_{k-1}], [\theta_k], [\delta s_k], [\delta \theta_k]$. This set of constraints lead to the Forward/Backward equations given in Table 1.

3.2.2 Correction Step

The bounded landmark pose of the i^{th} landmark in the global frame is:

$$\mathbf{L}_i = \begin{pmatrix} x_i \\ 0 \\ z_i \end{pmatrix} + d_i \cdot \mathbf{m}(\theta_i, \varphi_i) \quad (17)$$

Forward Propagation	
1.	$[x_k] = [x_k] \cap ([x_{k-1}] + [\delta s_k] \cos([\theta_{k-1}] + \frac{[\delta \theta_k]}{2}))$
2.	$[z_k] = [z_k] \cap ([z_{k-1}] + [\delta s_k] \sin([\theta_{k-1}] + \frac{[\delta \theta_k]}{2}))$
3.	$[\theta_k] = [\theta_k] \cap ([\theta_{k-1}] + [\delta \theta_k])$
Backward Propagation	
1.	$[x_{k-1}] = [x_{k-1}] \cap ([x_k] - [\delta s_k] \cos([\theta_{k-1}] + \frac{[\delta \theta_k]}{2}))$
2.	$[z_{k-1}] = [z_{k-1}] \cap ([z_k] - [\delta s_k] \sin([\theta_{k-1}] + \frac{[\delta \theta_k]}{2}))$
3.	$[\delta s_k] = [\delta s_k] \cap (([x_k] - [x_{k-1}]) / \cos([\theta_{k-1}] + \frac{[\delta \theta_k]}{2}))$
4.	$[\delta s_k] = [\delta s_k] \cap (([z_k] - [z_{k-1}]) / \sin([\theta_{k-1}] + \frac{[\delta \theta_k]}{2}))$
5.	$[\theta_{k-1}] = [\theta_{k-1}] \cap ([\theta_k] - [\delta \theta_k])$
6.	$[\theta_{k-1}] = [\theta_{k-1}] \cap (\arccos(([x_k] - [x_{k-1}]) / [\delta s_k]) - \frac{[\delta \theta_k]}{2})$
7.	$[\theta_{k-1}] = [\theta_{k-1}] \cap (\arcsin(([z_k] - [z_{k-1}]) / [\delta s_k]) - \frac{[\delta \theta_k]}{2})$
8.	$[\delta \theta_k] = [\delta \theta_k] \cap (\arccos(([x_k] - [x_{k-1}]) / [\delta s_k]) - [\theta_{k-1}])$
9.	$[\delta \theta_k] = [\delta \theta_k] \cap (\arcsin(([z_k] - [z_{k-1}]) / [\delta s_k]) - [\theta_{k-1}])$
10.	$[\delta \theta_k] = [\delta \theta_k] \cap ([\theta_k] - [\theta_{k-1}])$

Table 1: Forward/Backward Propagation

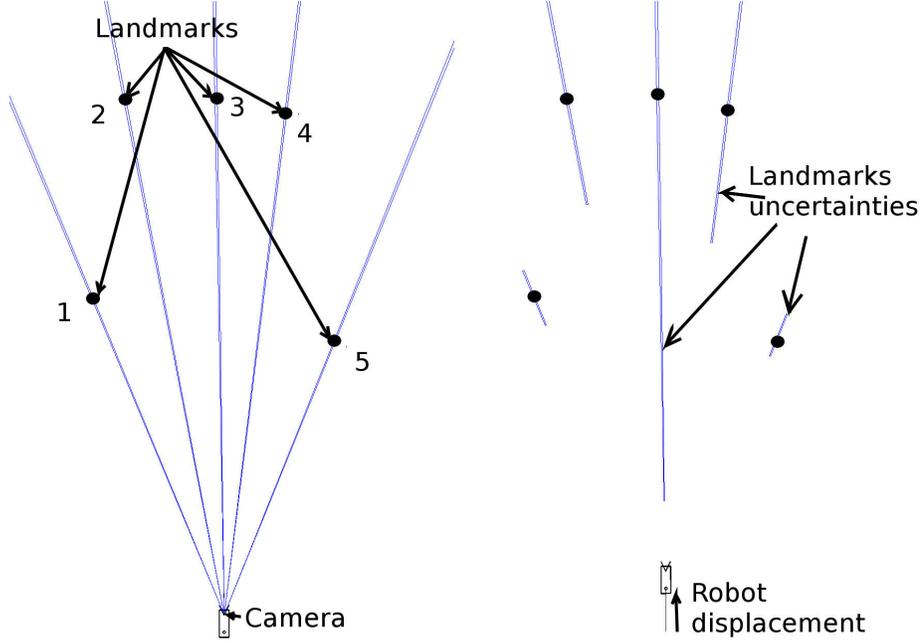


Figure 2: Landmarks map's evolution

where $y_i = 0$ because the robot only moves on the ground. $\mathbf{m}(\theta_i, \varphi_i)$ is a vector pointing from the camera to the landmark \mathbf{L}_i .

$$\mathbf{m}(\theta_i, \varphi_i) = \begin{pmatrix} -\sin(\theta_i) \cos(\varphi_i) \\ \sin(\varphi_i) \\ \cos(\theta_i) \cos(\varphi_i) \end{pmatrix} \quad (18)$$

The bounded landmark pose in the camera frame is:

$$\mathbf{L}_i^c = \mathbf{R}_{rob}(\mathbf{L}_i - \mathbf{T}_{rob}) \quad (19)$$

where \mathbf{R}_{rob} is the rotation matrix between the coordinates of the global and the robot frame and $\mathbf{T}_{rob} = (x, 0, z)^T = \mathbf{x}$ is the robot position.

The camera observes the projection of the i^{th} landmark on the image plane. Such an observation may be predicted by the measurement function which involves the robot pose through the landmark pose $[\mathbf{L}_i^c]$. It leads to the following constraints:

$$\mathbf{h}_i = \begin{pmatrix} u_{i,pred} \\ v_{i,pred} \end{pmatrix} = \begin{pmatrix} c_u - k_u L \frac{L_{i,x}^c}{L_{i,z}^c} \\ c_v - k_v L \frac{L_{i,y}^c}{L_{i,z}^c} \end{pmatrix} \quad (20)$$

where c_u , L , k_u , c_v , k_v are the camera parameters (defined beneath Eq. 14). Furthermore, by matching the predicted measurement with the observed one, we define two more constraints:

$$\begin{cases} [u_{i,pred}] = [u_{i,obs}] \\ [v_{i,pred}] = [v_{i,obs}] \end{cases} \quad (21)$$

The predicted and observed features are matched by:

- projecting the imprecise landmark on the image plane using Eq. 20 (probabilistic algorithms proceed in the same way by projecting an ellipsoid). This projection define a searching area $([u_{i,pred}], [v_{i,pred}])$ and links the robot pose with the landmark localization.
- selecting, in the projected area, an interest point $([u_{i,obs}], [v_{i,obs}])$ which has the closest SURF-descriptor to the landmark descriptor.

A Constraint Satisfaction Problem composed of a set of 4 constraints, for each observed landmark, is obtained from Eq. (20) and (21). In order to reduce the computational load of the processor, the problem may be simplified into two constraints which are the correction equations:

$$\begin{cases} u_{i,obs} = c_u - k_u L \frac{L_{i,x}^c}{L_{i,z}^c} \\ v_{i,obs} = c_v - k_v L \frac{L_{i,y}^c}{L_{i,z}^c} \end{cases} \quad (22)$$

Both the robot pose and the landmarks pose are updated from Equation (22) where $([L_{i,x}^c], [L_{i,y}^c], [L_{i,z}^c]) = [\mathbf{L}_i^c]$ is the landmark pose in the camera frame. Equation (19) links $[\mathbf{L}_i^c]$ to the robot pose and to the landmark pose in the global frame. Equation (22) leads, in the same way as during the prediction step, to a set of Forward/Backward equations (among the intervals $[x], [z], [\theta], [x_i], [z_i], [\theta_i], [\varphi_i], [d_i], [u_{i,obs}], [v_{i,obs}]$) which corrects both the robot pose and the landmark pose.

The pose of the robot and the landmarks are represented by boxes. There is no correlation in error as in probabilistic algorithms. Nevertheless there is a link between the poses which helps to jointly reduce the errors. This link is built by the whole set of equations at present time: all the landmarks are linked together via the robot pose. Furthermore there is another link between the variables of each pose: a Time Link (TL). This TL links each pose trough time and a modification on one parameter propagates in the past when it modifies another parameter (thanks to the prediction equations) and retropropagate in the present state.

3.3 First Simulation Results

Fig. 2 shows the evolution of the landmark uncertainties while the robot follows a straight path and uses our CP-SLAM algorithm. The leftmost image of Fig. 2 shows the initialization of the landmarks (see Sec. 3.1.2). Each landmark is initialized as an infinite cone: there is no information on the depth hypothesis. The following images show the evolution of the landmarks uncertainty after some prediction and correction steps. The more the parallax is important, the more the uncertainty correction is important. For instance, landmarks 1 and 5 are strongly corrected after a few observations. After a robot's displacement of 2 meters, the landmark depth uncertainty was reduced to less than 1m.

4 Simulations

We have used simulations to thoroughly test our CP-SLAM algorithm with an exact reference and compare the results to the classical EKF-SLAM.

4.1 Experimental setup

4.1.1 Environment

The simulated environment is a $15 \times 15 m^2$ area including 300 landmarks randomly positioned and two obstacles (see Fig. 8). Simulation are made using known data associations between observations and previous landmarks observations. We put a white Gaussian noise ($3\sigma_r=0.25mm$) on the wheel radius ($r = 4cm$) and a white Gaussian noise ($3\sigma_e=0.5mm$) on the length of the rear axle ($e = 20cm$). Observations noises are set to $3\sigma_o=3$ pixels. At each simulation step, corresponding to a displacement of 6 cm, the true robot's localization (a_{ref}) is known and compared to its estimate (a). In Section 4.2 the robot follows circles while in Section 4.3 he follows a random path. Whereas CP-SLAM use the 3σ bound noise, EKF-SLAM use a σ^2 variance. If we set CP-SLAM with a one standard deviation noise (σ), we would assume that about 68.26 percent of the data values are within our defined interval. It will be insufficient in most cases and leads to a great number of outliers. That's why, as others researchers, we have chosen the 3 standard deviations bound (99.73 percent of the data value are within our interval). Nevertheless, the interval error of the both algorithm are comparable as explained thereafter.

4.1.2 Consistency

For each simulation, we verify the algorithm consistency thanks to the interval error [37] and we study the volume of the localization box ($V = (\bar{x} - \underline{x})(\bar{z} - \underline{z})(\bar{\theta} - \underline{\theta})$). The interval error of an estimated state a is defined by $[\underline{a} - a_{ref}, \bar{a} - a_{ref}]$ where a_{ref} is the reference state (the true state). 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 includes the reference). The imprecise estimated state a is directly output by the CP-SLAM algorithm (whose input is a 3σ bounded noise) while the output standard deviation of the EKF-SLAM (whose input is a σ noise) is magnified 3 times. Consequently, the CP-SLAM imprecise state is defined by $a_{CP} = [\underline{a}_{CP}, \bar{a}_{CP}]$ whereas the EKF-SLAM state is $a_{EKF} = [\hat{a}_{EKF} - 3\sigma_{a_{EKF}}, \hat{a}_{EKF} + 3\sigma_{a_{EKF}}]$ (where \hat{a}_{EKF} is the estimated state).

4.1.3 Real time and post localization

CP-SLAM uses a set of constraints which is dynamically build along the experimentation. At each step, it adds new constraints into the Constraints Satisfaction Problem (constraints based on the prediction model and constraint based on each observation). Next, using a Forward/Backward algorithm, the intervals are contracted.

We define two different localization: the *real time localization* which corresponds to the current localization at each step, and the *post localization* which correspond to the best localization using all the data of the experimentation: the Forward/Backward algorithm uses future observations in order to correct past estimations. The localization improvement due to the new observation will be propagated to the past localization.

4.2 Stability of the robot pose

The robot follows 3 times the same circle (the diameter of the circle is 6 meters) in the test-environment described in Section 4.1.1. Fig. 3 presents the CP-SLAM results for the real-time and post localization. We can notice that the process is stabilized: the

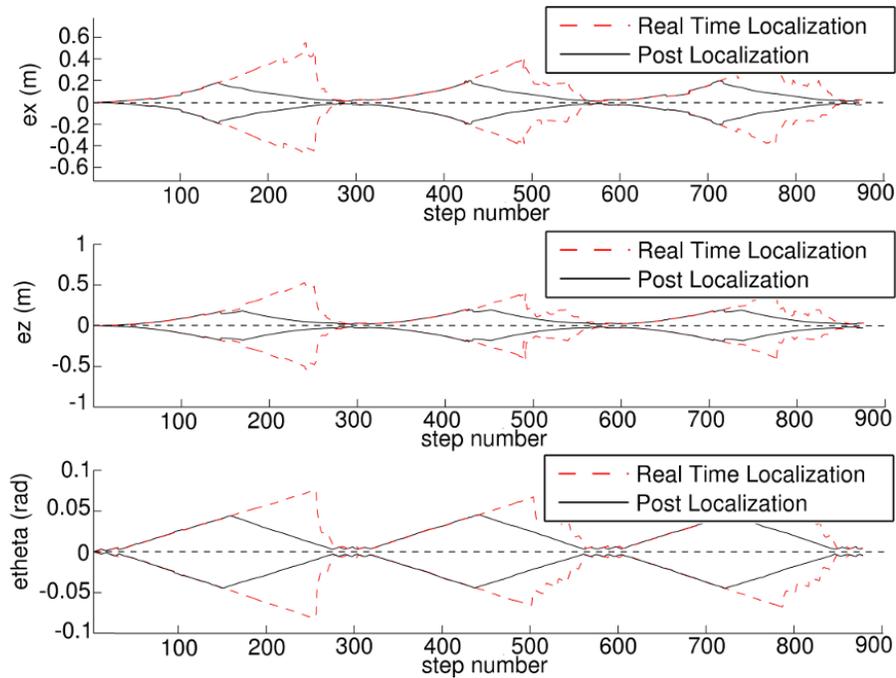


Figure 3: CP-SLAM interval error when looping on the exact same path

imprecision on the robot's localization remains stable despite the increasing number of laps. Theoretically, localization (of a major part of the followed path) could improve lap after lap due to the detection of previously detected landmarks with a different pose/uncertainty. Experimentally, the constraint satisfaction become very difficult after the second lap: the third lap is a copy of the second lap and so on. Further laps (which are a copy of the previous ones) are not shown. We have done another simulations with different landmarks locations and we get the same results. The post localization provides a similar performance during the 3 laps because it uses the same full updated map at each time step. Fig. 4 represents the EKF-SLAM results. Figures 3 (CP-SLAM) and 4 (EKF-SLAM) have not the same y-axis due to large differences between the interval errors of both algorithms. The EKF-SLAM localization is more accurate than in the case of the CP-SLAM. It has small consistency problems on the z axis (step 100, 400-500, 700). Both real time and post localization CP-SLAM are consistent during the entire simulation (their corridors always includes the zero value) but they have much larger uncertainties than the EKF-SLAM.

4.3 Convergence of the landmarks pose

To further improve the map, the robot should not follow the same path and should observe the landmarks from different view points. Fig. 5 and Fig. 6 (caution: the y-axis is not the same for both figures) represent the interval errors when the robot followed a random path in our test environment. We set the covariance of EKF-SLAM

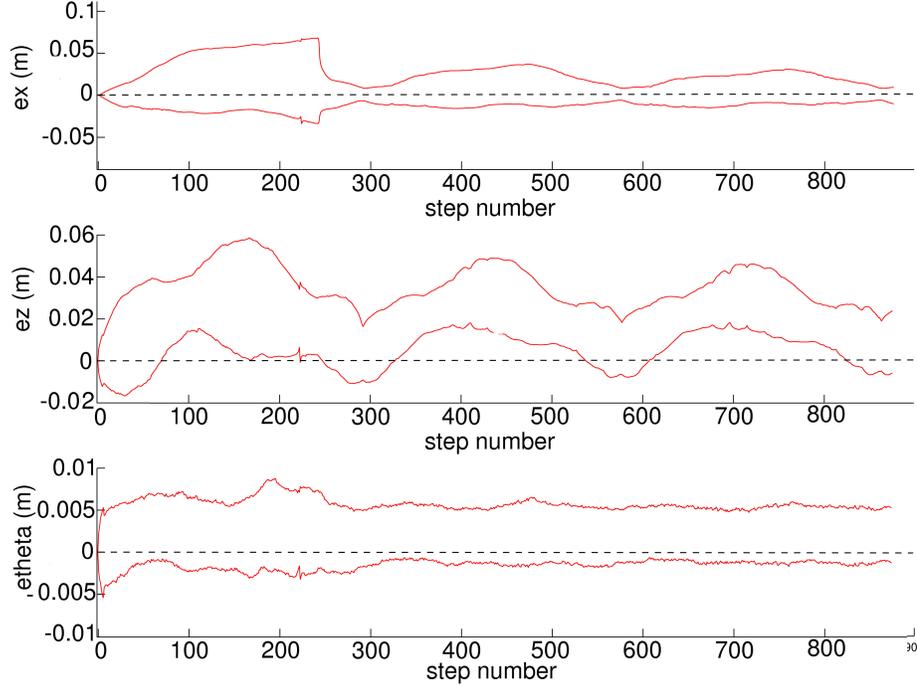


Figure 4: EKF-SLAM interval error when looping on the exact same path

by using the uncertainty models of sensors and the vehicle. EKF-SLAM is highly inconsistent (see Fig. 6) with these settings. We could get a consistent EKF-SLAM by tuning carefully (enlarging) the noises value for each simulation (we can do it at posteriori by using the reference). However such an EKF-SLAM tuning does not seem feasible for the SLAM problem in real world because we do not have such a reference. Such an inconsistency has already been described and studied in [11, 12, 39, 40]. CP-SLAM is consistent during the whole experiment. The post localization greatly improves the localization results, while at the same time keeping them consistent. Nevertheless, the imprecision of the CP-SLAM is ten time larger than the imprecision of the EKF-SLAM.

Figure 7 represents the final map created by the CP-SLAM and Fig. 8 shows the final EKF-SLAM map. The CP-SLAM map is consistent (the real landmarks positions “+” are included in the pink landmark uncertainty) whereas EKF-SLAM map is not consistent (real localization landmarks “+” are not included in the blue small ellipse). CP-SLAM provides a consistent map with a less accurate localization results due to a slower convergence than the EKF-SLAM.

5 Experimentation

We performed an experiment in a large classroom located in our laboratory building. Figure 9 shows the environment of the experiment. There are two pillars and two large

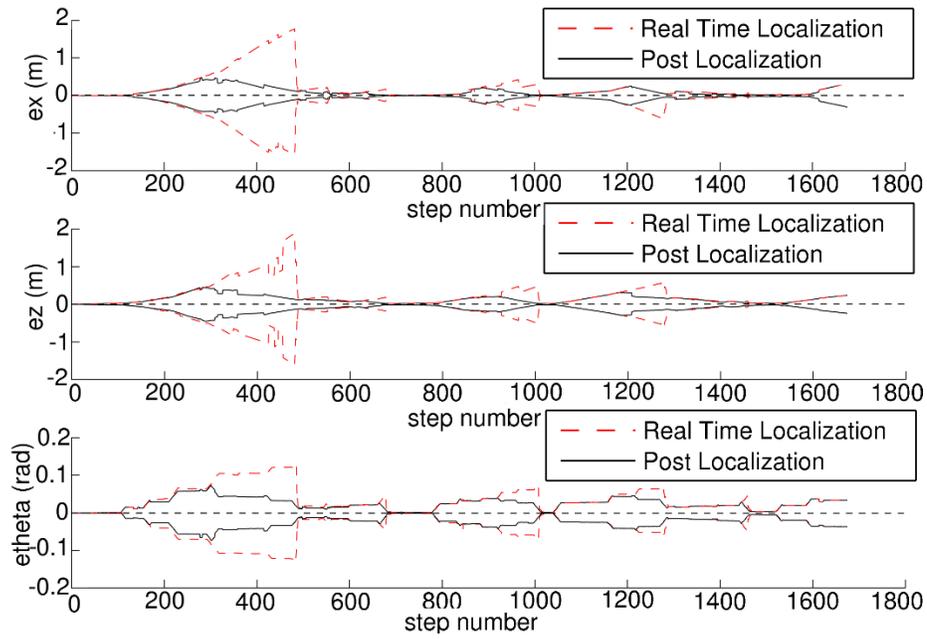


Figure 5: CP-SLAM interval error for a random path

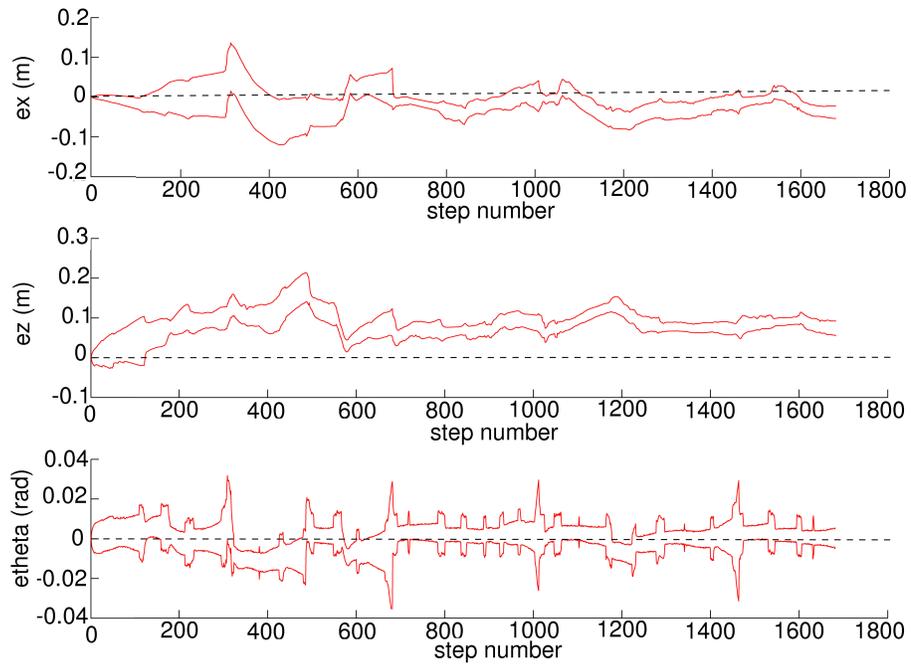


Figure 6: EKF-SLAM interval error for a random path

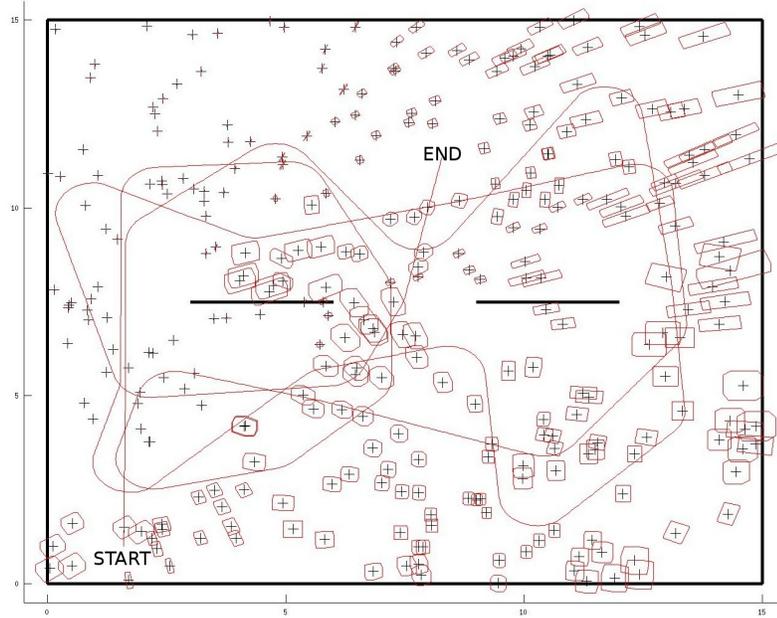


Figure 7: The random path and the associated CP-SLAM map results

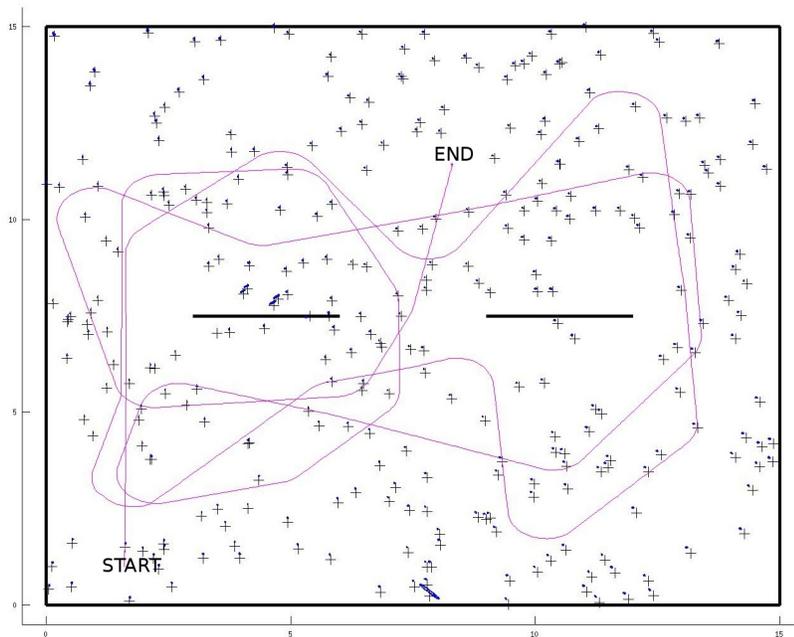


Figure 8: The random path and the associated EKF-SLAM map results



Figure 9: Environment of the experimentation

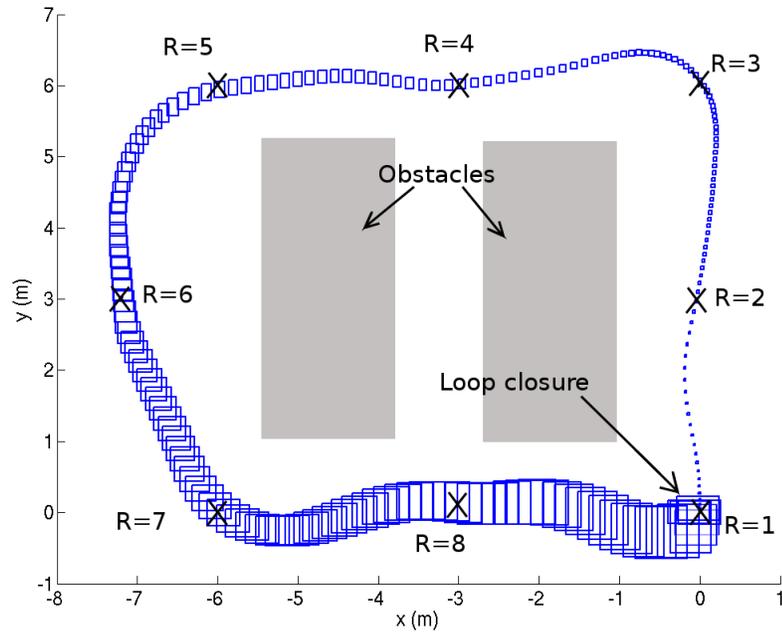


Figure 10: CP-SLAM experimental results

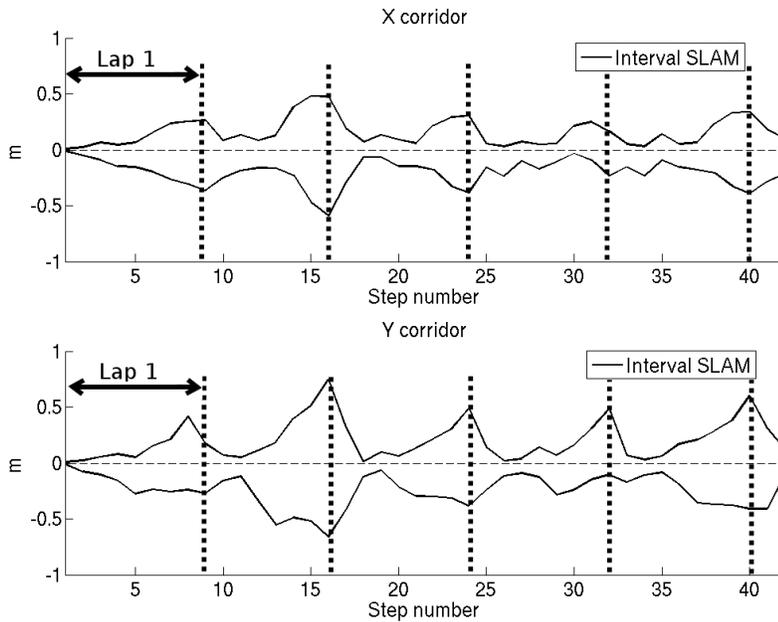


Figure 11: CP-SLAM interval error experimentally obtained

tables attached to each pillar. Computers are placed on each table and chairs are near the tables. We use the same noises as those defined for the simulation. Consistency tests were performed thanks to reference points drawn by the robot on the ground.

Figure 10 shows the first trajectory lap reconstructed by our robot using CP-SLAM (each horizontal or vertical part of the path is about 6 meters). The two largest obstacles of Fig. 10 are designed in order to see the occlusions created by the two pillars/tables. We notice that all boxes include the reference points (noted “R” on Figure 10) 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 11. 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 nine reference points, see Fig. 11). Once the loop closure is performed, the size of the localization box is about 10 cm long and the orientation uncertainty is 0.1 rad.

The localization of the robot during the second lap is a bit more imprecise than that computed during the first lap: this is due to the difficulties to match known landmarks. We do not show such a phenomenon during the simulation section because the matching was perfect. Nevertheless, the localization uncertainty is stabilized during the experiment: the uncertainty of the third lap is less important than the uncertainty of the second one. The increase in robot uncertainty, during a lap, is mainly due to the heading imprecision. 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 experimentally measured the reference heading.

6 Conclusion

To dealing with the consistency issue in SLAM literature, this paper presents a Constraints Propagation based SLAM algorithm. The proposed method works in two essential step using constraint propagation technique: the prediction step uses a bounded displacement model to integrate odometric data and the correction step process the vision observations by constructing ICSP. The bounded parametrization for nature landmark turns out to be effective and complete in monocular vision since no depth hypothesis or approximation are needed during the initialization. Our CP-SLAM algorithm is evaluated in both simulation and real data experiments. A comparison is made between CP-SLAM and the representative probabilistic method (EKF-SLAM). EKF-SLAM give more optimistic uncertainty estimation but consistency is not maintained whereas CP-SLAM gives stable and consistent result. Although CP-SLAM expresses a ten time larger uncertainties than those of the EKF-SLAM in the experiments, it worth the efforts to further study the CP-SLAM method since it is capable of providing guaranteed result. This is an important result for some robotic autonomous applications especially when safety is a crucial issue. Further work will dedicate to exploiting powerful ICP algorithm to reduce the uncertainties .

References

- [1] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot. Consistency of the EKF-SLAM algorithm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3562–3568, 2006.

- [2] T. Bailey, J. Nieto, and E. Nebot. Consistency of the FastSLAM algorithm. In *IEEE International Conference on Robotics and Automation*, pages 424–429, 2006.
- [3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110:346–359, 2008.
- [4] J. Cleary. Logical arithmetic. *Future Computing Systems*, 2(2):125–149, 1987.
- [5] 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.
- [6] M. Di Marco, A. Garulli, S. Lacroix, and A. Vicino. A set theoretic approach to dynamic robot localization and mapping. *Autonomous robots*, 16(1):23–47, 2004.
- [7] C. Drocourt, L. Delahoche, E. Brassart, B. Marhic, and A. Clémentin. Incremental construction of the robot environmental map using interval analysis. In *International Workshop on Global Optimization and Constraint Satisfaction*, pages 127–141. Springer, 2003.
- [8] A. Gning and P. Bonnifait. Guaranteed dynamic localization using constraints propagation techniques on real intervals. In *IEEE International Conference on Robotics and Automation*, pages 1499–1504, 2004.
- [9] 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.
- [10] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.
- [11] G.P. Huang, A.I. Mourikis, and S.I. Roumeliotis. Analysis and improvement of the consistency of extended kalman filter based slam. In *IEEE International Conference on Robotics and Automation*, pages 473–479, 2008.
- [12] S. Huang and G. Dissanayake. A critique of current developments in simultaneous localization and mapping. *International Journal of Advanced Robotic Systems*, 13(5):1–13, 2016.
- [13] P. Tarroux, I. K. Kueviakoe, A. Lambert. Vehicle localization based on interval constraint propagation algorithms. *IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, pages 179–184, 2013.
- [14] L. Jaulin. Localization of an underwater robot using interval constraint propagation. *International Conference on Principles and Practice of Constraint Programming*, pages 244–255, 2006.
- [15] L. Jaulin. Localization of an underwater robot using interval constraints propagation. In *Principles and Practice of Constraint Programming, Lecture Notes in Computer Science*, volume 4204, pages 244–255, 2006.
- [16] 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.

- [17] L. Jaulin. A nonlinear set membership approach for the localization and map building of underwater robots. *IEEE Transactions on Robotics*, 25(1):88–98, 2009.
- [18] L. Jaulin. Range-only slam with occupancy maps: A set-membership approach. *IEEE Transactions on Robotics*, 27(5):1004–1010, 2011.
- [19] L. Jaulin. Range-only slam with indistinguishable landmarks; a constraint programming approach. *Constraints*, 21(4):557–576, 2016.
- [20] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag, 2001.
- [21] L. Jaulin, M. Kieffer, E. Walter, and D. Meizel. Guaranteed robust nonlinear estimation with application to robot localization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 32(4):374–381, 2002.
- [22] I.K. Kueviakoe, Z. Wang, A. Lambert, E. Frenoux, and P. Tarroux. Localization of a vehicle: A dynamic interval constraint satisfaction problem-based approach. *Journal of Sensors*, 2018, 2018.
- [23] A. Lambert, D. Gruyer, B. Vincke, and E. Seignez. Consistent outdoor vehicle localization by bounded-error state estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1211–1216, 2009.
- [24] 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.
- [25] F. Le Bars, A. Bertholom, J. Sliwka, and L. Jaulin. Interval slam for underwater robots; a new experiment. In *NOLCOS 2010*, pages 132–139, 2010.
- [26] F. Lu and E. Miliotis. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.
- [27] J. Luo and S. Qin. A fast algorithm of slam based on combinatorial interval filters. *IEEE Access*, 6:28174–28192, 2018.
- [28] A.K. Mackworth. Consistency in networks of relations. *Artificial intelligence*, 8(1):99–118, 1977.
- [29] 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.
- [30] R.E. Moore. *Interval analysis*. Prentice-Hall Englewood Cliffs, NJ, 1966.
- [31] R.E. Moore. *Methods and Applications of Interval Analysis*. Studies in Applied Mathematics, 1979.
- [32] J.M. Porta. Cuikslam: A kinematics-based approach to slam. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2425–2431, 2005.

- [33] E. Seignez, M. Kieffer, A. Lambert, E. Walter, and T. Maurin. Real-time bounded-error state estimation for vehicle tracking. *The International Journal of Robotics Research*, 28(1):34–48, 2009.
- [34] S. Thrun and M. Montemerlo. The graph slam algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403–429, 2006.
- [35] S. Thrun and M. Montemerlo. The graph slam algorithm with applications to large-scale mapping of urban structures. *International Journal of Robotics Research*, 25(5-6):403–429, 2006.
- [36] D. Waltz. *Understanding line drawings of scenes with shadows*. 1975.
- [37] Z. Wang and A. Lambert. A low-cost consistent vehicle localization based on interval constraint propagation. *Journal of Advanced Transportation*, 2018, 2018.
- [38] X. Xie, Y. Yu, X. Lin, and C. Sun. An ekf slam algorithm for mobile robot with sensor bias estimation. In *32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 281–285, 2017.
- [39] T. Zhang, K. Wu, J. Song, S. Huang, and G. Dissanayake. Convergence and consistency analysis for a 3-d invariant-ekf slam. *IEEE Robotics and Automation Letters*, 2(2):733–740, 2017.
- [40] Y. Zhang, T. Zhang, and S. Huang. Comparison of ekf based slam and optimization based slam algorithms. In *IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2018.