

Research Article

A Low-Cost Consistent Vehicle Localization Based on Interval Constraint Propagation

Zhan Wang and Alain Lambert 

Laboratoire de Recherche en Informatique (LRI), CNRS, Univ Paris-Sud, Université Paris-Saclay, 91403 Orsay, France

Correspondence should be addressed to Alain Lambert; alain.lambert@u-psud.fr

Received 10 March 2018; Revised 3 May 2018; Accepted 13 May 2018; Published 24 June 2018

Academic Editor: Krzysztof Okarma

Copyright © 2018 Zhan Wang and Alain Lambert. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Probabilistic techniques (such as Extended Kalman Filter and Particle Filter) have long been used to solve robotic localization and mapping problem. Despite their good performance in practical applications, they could suffer inconsistency problems. This paper proposes an interval analysis based method to estimate the vehicle pose (position and orientation) in a consistent way, by fusing low-cost sensors and map data. We cast the localization problem into an Interval Constraint Satisfaction Problem (ICSP), solved via Interval Constraint Propagation (ICP) techniques. An interval map is built when a vehicle embedding expensive sensors navigates around the environment. Then vehicles with low-cost sensors (dead reckoning and monocular camera) can use this map for ego-localization. Experimental results show the soundness of the proposed method in achieving consistent localization.

1. Introduction

Using sensory information to locate the robot in its environment is one of the most fundamental problem providing a mobile robot with autonomous capabilities. A vast number of works [1, 2] dedicated to such problems use both proprioceptive and exteroceptive sensors. Proprioceptive sensors such as odometers and gyrometers are used to calculate the elementary movements, which are used to estimate the robot pose (position and orientation). However, this generates cumulative errors as the robot moves [3, 4]. To cope with this problem, exteroceptive sensors (lasers, sonars, GPS, cameras, etc.) are used to gather information either from the surrounding environment or from a global reference, aiming at eliminating the accumulated errors and improving the estimation of the robot pose [5, 6].

Probabilistic techniques have long been used in solving robotic localization and mapping problem. The most commonly used methods are Kalman filtering and Particle filtering [7, 8]. These methods associate a probability to a position, but nothing ensures that the robot is indeed in the position with the highest probability. It has long been noticed by the research community that probabilistic methods suffer

the inconsistency problem [9]. It is due to the fact that they are based on the assumption that the sensor noises are described by probability density function. However, in real world, many sensors do not possess Gaussian noise model or their distributions are simply not available [10].

Interval analysis [11] is an 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 the assumption that all the noises are bounded within known limits. This seems to be a realistic representation as most sensor manufacturers provide the maximum and minimum measurement errors under suitable working conditions. These extreme values of error can then be regarded as the error bounds. Interval based algorithms can be used to recursively propagate such bounded errors by using consistency techniques and systematic search methods. Contrary to probability methods, interval methods provide guaranteed sets which enclose the real state, without losing any feasible value.

Interval methods have achieved promising results in parameter and state estimation tasks [12–14], and they found their way to the mobile robotic localization and mapping

area. Instead of considering Gaussian noise, all model and measurement noises are assumed to be bounded. The aim is to guarantee the configurations of the robot pose that are consistent with the given measurements and noise bounds. Kieffer [15] and Jaulin [16] provided simulation results for a robot equipped with a belt of on-board ultrasonic sensors. The feasibility of interval methods on localization applications had been shown. Followed-up researchers focused on real-time implementation of such localization technique. Kieffer [17] and Seignez [18] presented results for a mobile robot navigating in an indoor environment, with odometers mounted on each rear wheel to measure the motion and sonars located in different directions to perceive the environment. The ability of interval method to cope with erroneous data and to obtain accurate estimations of the robot pose was demonstrated. Lambert [19] extended such work by providing results for an outdoor vehicle equipped with odometers, gyro, and GPS receiver. Comparison was made with the Particle Filter localization, showing the better performance of interval method in terms of consistency. Gning [2] and Kueviakoe [20] dealt with the outdoor localization problem in the framework of Interval Constraint Satisfaction Problem (ICSP). Those works used constraint propagation techniques to fuse the redundant data of sensors. Drevelle [21] uses relaxed constraint propagation approach to deal with erroneous GPS measurements. Bonnifait [22] combined constraint propagation and set inversion techniques and presented a cooperative localization method with significant enhancement in terms of accuracy and confidence domains. Experimental results illustrate that constraint propagation techniques are well adapted to real-time implementation and provide consistent result for localization. Some other researchers proposed specific contractors [23] or separators [24] to solve the localization problem.

Indeed, when considering consistency issue, interval methods are advantageous in vehicle localization. However, they are prone to large output imprecision, which can make them unsuitable for high level task. We improve on it and present an extension of others' works dealing with the localization problem by using ICP. Instead of using ultrasonic sensors [16], sonar [18, 25], or GPS [19, 26] as exteroceptive sensors, we propose to use a monocular camera. The aim is to achieve both consistent and low imprecision localization result. We propose to drive a vehicle embedding precise but expensive sensors around the environment in order to build up a map represented in an interval way. Then the proposed method enables a set or a single vehicle to navigate consistently in such environment using this map and low-cost sensors. By constructing the problem as an ICSP, the proposed method is able to handle both mapping and localization in a unified form using Interval Constraint Propagation techniques.

The paper is organized as follows. Section 2 introduces the basics of interval analysis. Section 3 illustrates how the mapping and localization problem can be formulated as an Interval Constraint Satisfaction Problem. Sections 4 and 5 present the simulation and experimental results in terms of mapping and localization. Section 6 draws a conclusion of the paper.

2. Overview of Interval Analysis and Constraint Propagation

Interval analysis and constraint propagation are the theoretical tools used in interval based methods for robotic localization and mapping. In this section, we give a brief overview of these tools.

2.1. Interval Analysis. Interval analysis [11, 27] is a numerical method which allows us to solve nonlinear problems in a guaranteed way. One of the pioneers of interval analysis, Ramon E. Moore, proposed to represent a solution of a problem by an interval in which the real solution is guaranteed. An interval $[x]$ is a connected subset of \mathbb{R} , defined by its lower bound \underline{x} and upper bound \bar{x} :

$$[x] = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\} \quad (1)$$

The width of a nonempty interval $[x]$ is calculated by $w([x]) = \bar{x} - \underline{x}$ and the midpoint (or center) is $mid([x]) = (\bar{x} - \underline{x})/2$. If $[x]$ is an interval containing the x position of a vehicle, $mid([x])$ can be taken as an estimation of x and $w([x])/2$ can be regarded as the estimation error [28].

Let \mathbb{IR} be the set of the intervals of \mathbb{R} . A *box* (also called *interval vector*) $[\mathbf{x}]$ ($[\mathbf{x}] \in \mathbb{IR}^n$) is a generalization of the interval concept. It is a vector whose components are intervals:

$$[\mathbf{x}] = [x_1] \times [x_2] \times \cdots \times [x_n] \quad (2)$$

For instance, the configuration of a vehicle's pose usually contains 3 parameters: position x , position y , and heading angle θ . Consequently, for vehicle localization, the solution is a three-dimensional box: $[x] \times [y] \times [\theta]$.

The width of a nonempty box $[\mathbf{x}]$ can be computed by $w([\mathbf{x}]) = \max_{i=1,\dots,n} w([x_i])$. We define the volume of $[\mathbf{x}]$ as

$$vol([\mathbf{x}]) = \prod_{1 \leq i \leq n} w([x_i]) \quad (3)$$

The volume of the box is usually used to evaluate the estimation uncertainty of a state vector [28].

2.2. Operations of Interval Arithmetic. Rules have been defined to apply the basic arithmetical operations on intervals [27]. Let us consider two intervals $[x]$ and $[y]$ and a binary operator $\diamond \in \{+, -, \times, \div\}$; the smallest interval which contains all feasible values for $[x] \diamond [y]$ is defined as follows:

$$[x] \diamond [y] = \{x \diamond y \mid x \in [x], y \in [y]\} \quad (4)$$

Example 1. Addition: $[-4, 3] + [2, 5] = [-2, 8]$; subtraction: $[3, 5] - [1, 2] = [1, 4]$; multiplication: $[-1, 8] \times [-3, 1] = [-24, 8]$; division: $[-2, 8] \div [2, 4] = [-1, 4]$.

The set-theoretic operations can be applied to intervals. The *intersection* of two intervals is defined by

$$[x] \cap [y] = \{z \in \mathbb{R} \mid z \in [x] \text{ and } z \in [y]\} \quad (5)$$

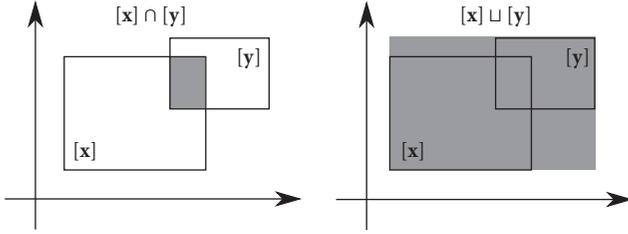
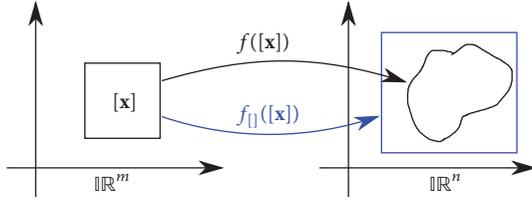


FIGURE 1: Intersection and interval hull of interval boxes.

FIGURE 2: Images of an interval box by function f and its inclusion function $f_{[]}$.

The result of *intersection* is always an interval, but it is not the case for their *union*

$$[x] \cup [y] = \{z \in \mathbb{R} \mid z \in [x] \text{ or } z \in [y]\} \quad (6)$$

Consequently, an *interval hull* is defined as the smallest interval that contains all the subsets of $[x] \cup [y]$, denoted by $[x] \sqcup [y]$:

$$[x] \sqcup [y] = [[x] \cup [y]] \quad (7)$$

For instance, the interval hull of $[2, 4] \cup [5, 8]$ is the interval $[2, 8]$. Figure 1 gives an example of *intersection* and *union* for 2 interval boxes; it is realized by applying the intersection and interval hull operator to each component of the box.

2.3. Inclusion Function. The image of a vector function $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$ (defined by arithmetical operators and elementary functions) over an interval box $[x]$ can be evaluated by its inclusion function $f_{[]}$, whose output contains all possible values taken by $f(\cdot)$ over $[x]$ (see Figure 2):

$$\forall [x] \in \mathbb{R}^m, f([x]) \subset f_{[]}([x]) \quad (8)$$

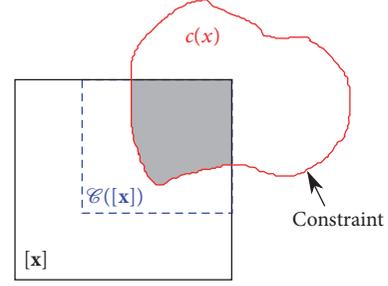
The simplest way to create an inclusion function is to replace all the variables and operators by their interval equivalents. The resulting interval function is called “natural inclusion function”. For instance,

$$\forall x \in \mathbb{R}, \hat{\mathbb{N}}\hat{\mathbb{L}}f(x) = 4x^4 - 7\ln(x) - 3 \quad (9)$$

has the natural inclusion function:

$$\forall [x] \in \mathbb{R}, f_{[]}([x]) = 4[x]^4 - 7\ln_{[]}([x]) - 3 \quad (10)$$

Inclusion function can also be used to represent equations defined by interval variables. Such equations, also called constraints, are the core of Interval Constraint Satisfaction Problems.

FIGURE 3: Contraction of a box by a contractor \mathcal{C} .

2.4. Contractor. The concept of contractor is directly inspired from the ubiquitous concept of filtering algorithms in constraint programming. Given a constraint c relating a set of variables x , an algorithm \mathcal{C} is called a contractor (or a filtering algorithm) if \mathcal{C} returns a subdomain of the input domain $[x]$ and the resulting subdomain $\mathcal{C}([x])$ contains all the feasible points (the gray domain in Figure 3) with respect to the constraint c :

$$\mathcal{C}([x]) \subseteq [x] \quad \text{and} \quad \forall x \in [x], c(x) \implies x \in \mathcal{C}([x]) \quad (11)$$

2.5. Interval Constraint Satisfaction Problem (ICSP). The concept of Interval Constraint Satisfaction Problem (ICSP) was introduced by Hyvönen [29] in 1992. It is typically defined as

- (i) a set V of variables $\{v_1, v_2, \dots, v_n\}$,
- (ii) a set D of domains $\{d_1, d_2, \dots, d_n\}$, such that for each variable v_i , a domain d_i with the possible values for that variable is given; d_i is an interval or union of intervals,
- (iii) a set C of p constraints $\{c_1, c_2, \dots, c_p\}$; each constraint c_i defines the relationships of a number of variables from V ; e.g., $c_1(v_1, v_2, v_3) = 0$ restricts the possible domains of v_1, v_2 , and v_3 .

An ICSP is a mathematical problem whose solution is the minimal domain of the variables satisfying all the constraints.

2.6. Interval Constraint Propagation (ICP). To solve an ICSP, an Interval Constraint Propagation (ICP) algorithm iterates domain reductions until no domain can be contracted. To satisfy the set of p constraints, it removes, from the domain of the variable, every value that is not compatible with the constraints and the other variables. ICP reduces the size of the domains in a consistent way by repeating this removal operation. This solving process is called *contraction* and the corresponding algorithm (e.g., Forward/Backward) used for contraction is regarded as a *contractor*.

Let us consider a simple ICSP instance which is defined by 3 variables and 2 constraints:

$$\begin{aligned} z &= x + \ln(y) \\ y &= z^2 \end{aligned} \quad (12)$$

The domains of x , y , and z are, respectively, $[x]$, $[y]$, and $[z]$. An inclusion function of (12) is

$$\begin{aligned} [z] &= [x] + \ln_{\square}([y]) \\ [y] &= [z]^2 \end{aligned} \quad (13)$$

The Forward/Backward Propagation (FBP), an ICP algorithm, can be applied to solve the ICSP. FBP executes the contraction process in two phases: firstly, the Forward propagation reduces the left terms of (13) via

$$[z] = [z] \cap ([x] + \ln_{\square}([y])) \quad (14a)$$

$$[y] = [y] \cap [z]^2 \quad (14b)$$

Secondly, Backward propagation reduces the right terms of (13) by

$$[x] = [x] \cap ([z] - \ln_{\square}([y])) \quad (15a)$$

$$[y] = [y] \cap \exp_{\square}([z] - [x]) \quad (15b)$$

$$[z] = [z] \cap \sqrt{\square}[y] \quad (15c)$$

Forward and Backward equations, obtained from the constraints, are computed one after the other (here we compute from (14a) to (15c)). When all the equations have been computed, the algorithm restarts from the beginning equation until the domains of $[x]$, $[y]$, and $[z]$ are no more contracted (or contracted less than a specified parameter).

3. Localization and Mapping

3.1. Problem Statement. Let us consider a vehicle navigating in a 2D environment. At time step k , the vehicle pose is represented by a three-dimensional vector $\mathcal{X}_k = (x_k, y_k, \theta_k)^T$ with x_k , y_k being the vehicle's position and θ_k the orientation. The n stationary landmarks are represented by $\mathcal{M} = (\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_n)^T$. \mathcal{L}_i is a state vector defining the landmark position (see Section 3.3). For the sake of simplicity, we assume that the pose evolution follows *Markov chain*, where the current pose \mathcal{X}_k depends only on the very last pose \mathcal{X}_{k-1} and the current control input vector \mathcal{U}_k . The input vector \mathcal{U}_k consists of linear and angular elementary movements, which were collected by proprioceptive sensors (e.g., odometer and gyro). If these odometric measurements are available, the state vector \mathcal{X}_k can be evaluated through the dead reckoning function (also called motion model) expressed as follows:

$$\mathcal{X}_k = f(\mathcal{X}_{k-1}, \mathcal{U}_k) + w_k \quad (16)$$

where w_k is the measurement noise affecting the odometric measurements.

The vehicle is equipped with exteroceptive sensors, recording information about the environment. At each time step k , the vehicle observes a number of landmarks which can be regarded as a subset of the entire map; the measurement \mathcal{X}_k depends only on the current vehicle pose \mathcal{X}_k and the map state \mathcal{M} :

$$\mathcal{X}_k = \mathcal{H}(\mathcal{X}_k, \mathcal{M}) + r_k \quad (17)$$

where $\mathcal{H}(\cdot)$ is generally a nonlinear function; it differs for different types of exteroceptive sensors. The output of $\mathcal{H}(\cdot)$ is a geometric parameterization of the landmark (e.g., distance, angle, or pixel coordinates), and r_k is the measurement noise of the exteroceptive sensors which introduce uncertainty to the landmark state.

Interval methods are based on the assumption that the support of sensor noises is unknown but bounded by real intervals. Mathematically, $\forall k > 0$, $w_k \in [\mathbf{w}]$ and $r_k \in [\mathbf{r}]$, where $[\mathbf{w}]$ and $[\mathbf{r}]$ are real intervals defined by lower and upper bounds. Then the localization problem can be regarded as an ICSP: the set of variables is defined by the vehicle pose and the landmark position; the motion and observation functions ((16) and (17)) define the constraints. ICP techniques allow us to propagate the interval uncertainties in a consistent way without any probability hypothesis or linearization process. As a result, all possible solutions are found and guaranteed.

3.2. Motion Model. The motion estimation process uses the measurements of proprioceptive sensors to predetermine the localization vector $[\mathcal{X}_k]$. EKF use a probabilistic motion model to fuse sensor data [30]. We make an extension of this model to obtain an interval based model defined by Seignez et al. [18]:

$$[\mathcal{X}_k] = f_{\square}([\mathcal{X}_{k-1}], [\mathcal{U}_k]) \quad (18)$$

$$\begin{aligned} &\begin{pmatrix} [x_k] \\ [y_k] \\ [\theta_k] \end{pmatrix} \\ &= \begin{pmatrix} [x_{k-1}] + [\delta s_k] \cos_{\square} \left([\theta_{k-1}] + \left[\frac{\delta \theta_k}{2} \right] \right) \\ [y_{k-1}] + [\delta s_k] \sin_{\square} \left([\theta_{k-1}] + \left[\frac{\delta \theta_k}{2} \right] \right) \\ [\theta_{k-1}] + [\delta \theta_k] \end{pmatrix} \end{aligned} \quad (19)$$

where $[\mathcal{U}_k] = ([\delta s_k], [\delta \theta_k])^T$ is the input vector. δs_k and $\delta \theta_k$ represent, respectively, the elementary displacement and rotation.

The elementary movements can be deduced from raw odometric data (odometer and gyro) using an interval based static fusion method:

$$\begin{aligned} [\delta s_k] &= \frac{[\pi]([\omega_l][\delta p_l] + [\omega_r][\delta p_r])}{[P]} \\ [\delta \theta_k] &= \frac{[\pi]([\omega_l][\delta p_l] - [\omega_r][\delta p_r])}{[e] \cdot [P]} \cap [\delta \theta_{gyro}] \end{aligned} \quad (20)$$

where δp_l and δp_r are the odometer measurements of the left and right wheels. ω_l and ω_r represent the radius of the wheels. P is the odometer resolution and e is the length of the rear axle. $\delta \theta_{gyro}$ is the gyro measurement. For further information about interval based odometric sensor integration, the reader might refer to [31]. Once we have constituted intervals from sensor data, the vehicle pose $[\mathcal{X}_k]$ can be estimated by applying FBP algorithm (Section 2.6) over constraints deduced from the motion model.

3.3. Landmark Parameterization. A landmark (or feature point) is a 3D point in the global world. Probabilistic methods generally represent landmark with an inverse depth parameterization [32]. They model the uncertainty of each parameter by Gaussian distributions. However, it turns out to be not so efficient to represent the linear depth uncertainty of monocular vision. Interval analysis provides an easy and efficient way to parameterize landmarks. Each landmark \mathcal{L}_i is defined as a six-dimensional state vector: $([x_o], [y_o], [z_o], [\alpha_i], [\varphi_i], [d_i])^T$, which models the estimated landmark position at

$$[\mathcal{L}_i] = ([x_o], [y_o], [z_o])^T + [d_i] \cdot [\mathbf{m}]([\alpha_i], [\varphi_i]) \quad (21)$$

where coordinates $[x_o]$, $[y_o]$, and $[z_o]$ represent the optical center of the camera when the landmark was seen for the first time, $[\alpha_i]$ and $[\varphi_i]$ represent azimuth and elevation angle for the ray which traces the landmark, and $[d_i]$ is the depth of the landmark. $[\mathbf{m}]([\alpha_i], [\varphi_i])$ is a unitary vector pointing from the camera to the landmark $[\mathcal{L}_i]$:

$$[\mathbf{m}]([\alpha_i], [\varphi_i]) = \begin{pmatrix} \cos_{\square}([\alpha_i]) \cos_{\square}([\varphi_i]) \\ -\sin_{\square}([\alpha_i]) \cos_{\square}([\varphi_i]) \\ \sin_{\square}([\varphi_i]) \end{pmatrix} \quad (22)$$

Since all the parameters are represented by intervals, and $[d_i]$ can be initialized as $[0, +\infty]$, the landmark's uncertainty is modeled as an infinite cone which combines the vehicle pose uncertainty and the observation uncertainty. It is a realistic representation for the monocular vision uncertainty. The major advantage is that the initialization of $[d_i]$ is undelayed, guaranteed, and efficient for landmarks over a wide range as it always includes all possible value without any prior information.

3.4. Observation Model. The observation of a landmark is the coordinates of the pixel where the landmark is projected on the image. In order to detect and extract landmark information from an image, we used Speed Up Robust Features (SURF) [33]. The work [34] recommended it after a comparison between several algorithms, showing that SURF provide results with a very low error rate. The projection of the i^{th} landmark to the image frame can be formulated as follows:

From (21), the landmark's position in the world coordinate \mathbf{O}_w can be expressed:

$$[\mathcal{L}_i^w] = ([x_o], [y_o], [z_o])^T + [d_i] \cdot [\mathbf{m}]([\alpha_i], [\varphi_i]) \quad (23)$$

In the camera coordinate \mathbf{O}_c , the landmark $[\mathcal{L}_i^w]$ will be seen in the position $[\mathcal{L}_i^c]$:

$$[\mathcal{L}_i^c] = [R^{wc}]([\mathcal{L}_i^w] - [R^{rw}][\mathcal{X}_k]) \quad (24)$$

where $[\mathcal{X}_k]$ is the current vehicle pose. R^{wc} and R^{rw} are, respectively, the transformational matrix between the world and the camera coordinate and the robot and world coordinate, respectively.

The camera does not observe $[\mathcal{L}_i^c]$ directly but its projection in the image frame $(u, v)^T$. According to the pin-hole

camera model [35], the observation \mathcal{Z}_k^i of $[\mathcal{L}_i^c]$ on the image frame can be predicted by

$$[\mathcal{Z}_k^i] = \mathcal{H}_{\square}([\mathcal{X}_k], [\mathcal{L}_i^c]) \quad (25)$$

$$\begin{pmatrix} [u_{pre}^i] \\ [v_{pre}^i] \end{pmatrix} = \begin{pmatrix} [c_u] + [f][k_u] \frac{[\mathcal{L}_i^c(x)]}{[\mathcal{L}_i^c(z)]} \\ [c_v] + [f][k_v] \frac{[\mathcal{L}_i^c(y)]}{[\mathcal{L}_i^c(z)]} \end{pmatrix} \quad (26)$$

where f , k_u , k_v , c_u , c_v are the camera intrinsic parameters. Equation (26) is also used as the projection function.

3.5. Data Association. Data association focuses on finding out the correspondence between the map and observations. Different from probabilistic method, which deals with uncertainty ellipsoid in projection procedure, interval method proceeds with bounding box. The matching process is carried out as follows:

- (i) Project the landmarks to the current image plane; discard those which lay out of the image bounds. Each landmark defines a rectangle searching area.
- (ii) In the searching area, search for the candidate feature that is matched to the landmark. We adopt Dan's method [36] which performs a two-stage matching process (considering both the Euclidean distance and dominant orientation information of feature descriptors) to provide a robust and accuracy matching result. For each pair of matched features, a Zero Normalized Cross Correlation (ZNCC) [37] score is computed in the mapping stage to help to evaluate the robustness of the landmarks.
- (iii) Filtering step is necessary to keep a desirable amount of matched points. The landmark uncertainty and ZNCC score can be used for the selection process. We make the matched results distributed uniformly over the image, providing enough parallax for the localization.

3.6. ICSP Formulation. The data association process generates a set of 3D to 2D correspondences, which we can use to build the ICSP constraints. Let us consider the i^{th} landmark $[\mathcal{L}_i]$ that has been matched. The observation of $[\mathcal{L}_i]$ is $([u_{obs}^i], [v_{obs}^i])^T$. The predicted position on the image plane, denoted by $([u_{pre}^i], [v_{pre}^i])^T$, is computed via (23), (24), and (26). The observation can be used to correct the system state through the predicted observation:

$$\begin{aligned} [u_{pre}^i] &= [u_{obs}^i] \\ [v_{pre}^i] &= [v_{obs}^i] \end{aligned} \quad (27)$$

Equation (27) creates a link between the prediction and observation data of the i^{th} landmark $[\mathcal{L}_i]$. This link (thanks to (23), (24), (26), and (27)) imposes constraints on $[\mathcal{X}_k]$ and $[\mathcal{L}_i]$, which can be used to update both the vehicle pose and landmark position.

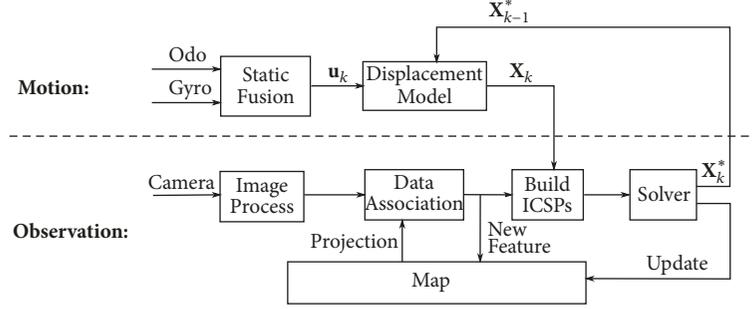


FIGURE 4: Mapping framework.

Based on the above derivation, an ICSP composed of 2 top level constraints and a set of 9 variables can be set up. For each matched landmark i , the $\text{ICSP}_k(V_k, D_k, C_k)$ is generated and defined as follows:

(i) A set of 9 variables:

$$V_k = \{x_k, y_k, \theta_k, x_o, y_o, z_o, \alpha_i, \varphi_i, d_i\} \quad (28)$$

(ii) A set of 9 domains:

$$D_k = \{[x_k], [y_k], [\theta_k], [x_o], [y_o], [z_o], [\alpha_i], [\varphi_i], [d_i]\} \quad (29)$$

(iii) A set of 2 top level constraints:

$$C_k = (C_k^1, C_k^2)^T \quad (30)$$

$$C_k^1 \iff [u_{pre}^i] = [u_{obs}^i] \quad (31)$$

$$C_k^2 \iff [v_{pre}^i] = [v_{obs}^i]$$

If the data association process found N matched landmarks, then we get a multiple ICSP to solve:

$$v\text{ICSP}_k = (\text{ICSP}_k^1, \text{ICSP}_k^2, \dots, \text{ICSP}_k^N) \quad (32)$$

Each ICSP contains constraints between the vehicle pose and a unique landmark position. The good point is that they are all linked together via the same variables x_k, y_k , and θ_k (the vehicle pose at current time k). This correlation makes ICP algorithms very powerful to jointly contract the domains of all variables.

3.7. System Overview. Figure 4 depicts the framework of the mapping stage. It is an interval SLAM process which is cast into two parts: motion and observation. In the motion part, the preliminary pose of the vehicle is predicted by fusing the measurements of the odometers and gyro with bounded error model; in the observation part, features are firstly extracted from the image, data association task aims at finding out the correspondences between the features and the map, new features will be initialized as new landmarks and added to the map, and old ones will be processed to generate new ICSPs; then these ICSPs are jointly solved by an ICP algorithm, which correct the vehicle pose and update the map.

```

(1) while Localization is required do
(2)   Input:  $[\mathcal{X}_{k-1}], [\mathcal{U}_k], [\mathcal{M}], I_k$ 
(3)   Output:  $[\mathcal{X}_k]$ 
(4)    $[\mathcal{X}_k] \leftarrow f_{[]}([\mathcal{X}_{k-1}], [\mathcal{U}_k])$ 
(5)    $S_k \leftarrow \text{ExtractFeature}(I_k)$ 
(6)    $[(u, v)_{obs}, \mathcal{L}]^N \leftarrow \text{DataAssociate}([\mathcal{X}_k], [\mathcal{M}], S_k)$ 
(7)   for  $j \leftarrow 1$  to  $N$  do
(8)      $\text{ICSP}_j \leftarrow \text{Build}([\mathcal{X}_k], [(u, v)_{obs}, \mathcal{L}]^j)$ 
(9)   end for
(10)   $v\text{ICSP}_k = \bigcup_{j=1}^N \text{ICSP}_j$ 
(11)   $[\mathcal{X}_k^*] \leftarrow \text{Solver}(v\text{ICSP}_k)$ 
(12) end while

```

ALGORITHM 1: Localization.

The localization stage is quite similar to the mapping stage. There are two main differences: high cost sensors are no longer used and no new feature is added to the map. Algorithm 1 gives the pseudocode of our localization method:

- (i) Line (4): motion estimation. The current vehicle pose is calculated based on the motion function $f_{[]}$ and input vector $[\mathcal{U}_k]$.
- (ii) Line (5): image processing. A set of SURF S_k are extracted from the new image I_k .
- (iii) Line (6): data association. The correspondence between the features and the map is found.
- (iv) Line (8): ICSP generating. New ICSP are generated based on the matched landmarks.
- (v) Line (10)-(11): Solving. $v\text{ICSP}_k$ is composed by the ICSPs and solved by an ICP algorithm.

4. Simulation Results

A simulation experiment is set up to evaluate the performance of our proposed ICSP based mapping and localization method. The vehicle starts from the origin (see Figure 5) and moves straight forward in the positive direction of x-axis with constant linear and angular velocity ($v = 0.1m \cdot s^{-1}$, $\omega = 0$). 5 landmarks are positioned in the surrounding environment without any prior knowledge of their positions. At each time

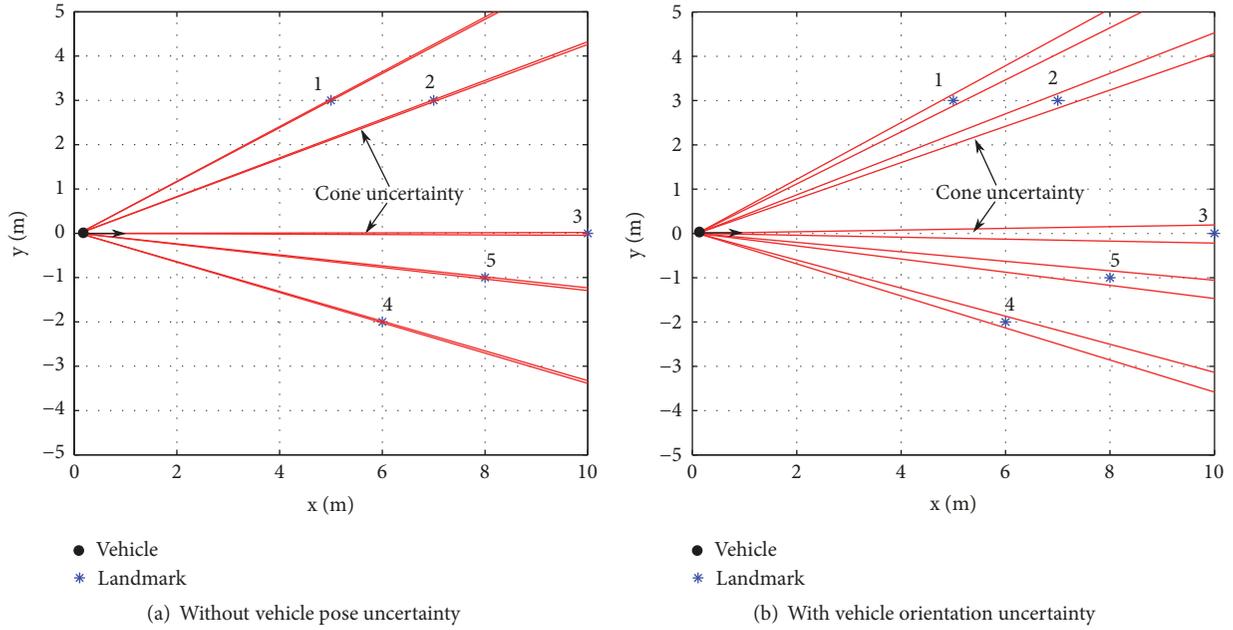


FIGURE 5: Interval based landmark initialization.

step, the vehicle detects the 5 landmarks with a simulated camera model $[f, k_u, k_v, c_u, c_v] = [40, 8, 8, 320, 240]$ and focuses on estimating their position. The experiment is first carried out without odometric sensor noise (the vehicle pose is perfectly known) to verify the feasibility of our method. This condition (no odometric noise) is then released in the experiment presented in Section 4.3.

4.1. Landmark Initialization. At time step $k = 1$, the 5 landmarks are first detected and the initialization procedure is executed. Figure 5(a) shows the top view of the initialization result of landmarks (see Section 3.3). Each landmark's uncertainty is initialized as an infinite cone because of the lacking information about the landmark depth. As there is no uncertainty on vehicle pose, the landmarks are well initialized and the opening angles which indicate the observation uncertainty are small (due to the bounded observation noise which is 1 pixel). If the vehicle has an uncertainty on its orientation (the heading angle is not perfectly defined: $[\theta_k] = [-\pi/180, \pi/180]$), then the cones of uncertainty are much larger than the previous ones (see Figure 5(b)). This is reasonable since the cone uncertainty is the fusion of observation uncertainty and vehicle pose uncertainty. Because our bounded error hypothesis and the use of interval analysis, the initialization result is guaranteed to contain the true landmark position.

4.2. Landmark Uncertainty Evolution. As the vehicle moves, the landmarks are repeatedly observed with different parallaxes. By building and solving the associated ICSP (see Section 3.6), each of the landmark's parameters can be estimated. Figure 7 shows, since initialization, the evolution of the poses uncertainties of landmarks 1, 2, and 5 in top view. Each row plots the estimation result at time steps 1, 5, 10, and

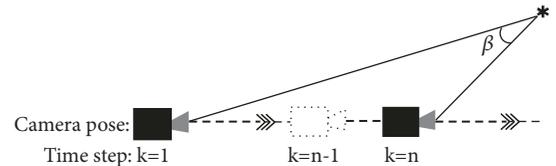


FIGURE 6: Parallax observations.

20. The parallax β (defined as the angle difference between the initial observation and current observation, see Figure 6) is displayed as the title of each subfigure. With the estimation proceeds, the landmark uncertainty estimate improves. As we can see, the bigger the parallax is, the better the contraction of the landmark uncertainty will be. When enough parallax is eventually available, the infinite interval uncertainty is significantly reduced. The uncertainty of landmark 3 keeps infinite because it locates on axis x and no parallax has been detected during the horizontal translation.

To visualize the evolution of a landmark state, it is necessary to convert the 6-dimensional representation to a global 3D point using (21): $(x_o, y_o, z_o, \alpha_i, \phi_i, d_i)^T \rightarrow (x_i, y_i, z_i)^T$. The interval uncertainty (IU) can be evaluated by computing the box volume [31]: $U(i) = w([x_i]) \times w([y_i]) \times w([z_i])$. As the depth of each landmark is initialized as $[0, +\infty]$, the x , y , and z coordinates have infinite value, and the uncertainty turns out to be infinite at the first hand. After multiple observations (at different times) from different parallaxes, the landmark's state is updated. Figure 8 depicts the evolution of the uncertainty associated with the landmarks 1, 2, 4, and 5. The 4 landmarks encounter different observation parallaxes, so the contraction of their uncertainties is not simultaneous. However, they exhibit the same performance:

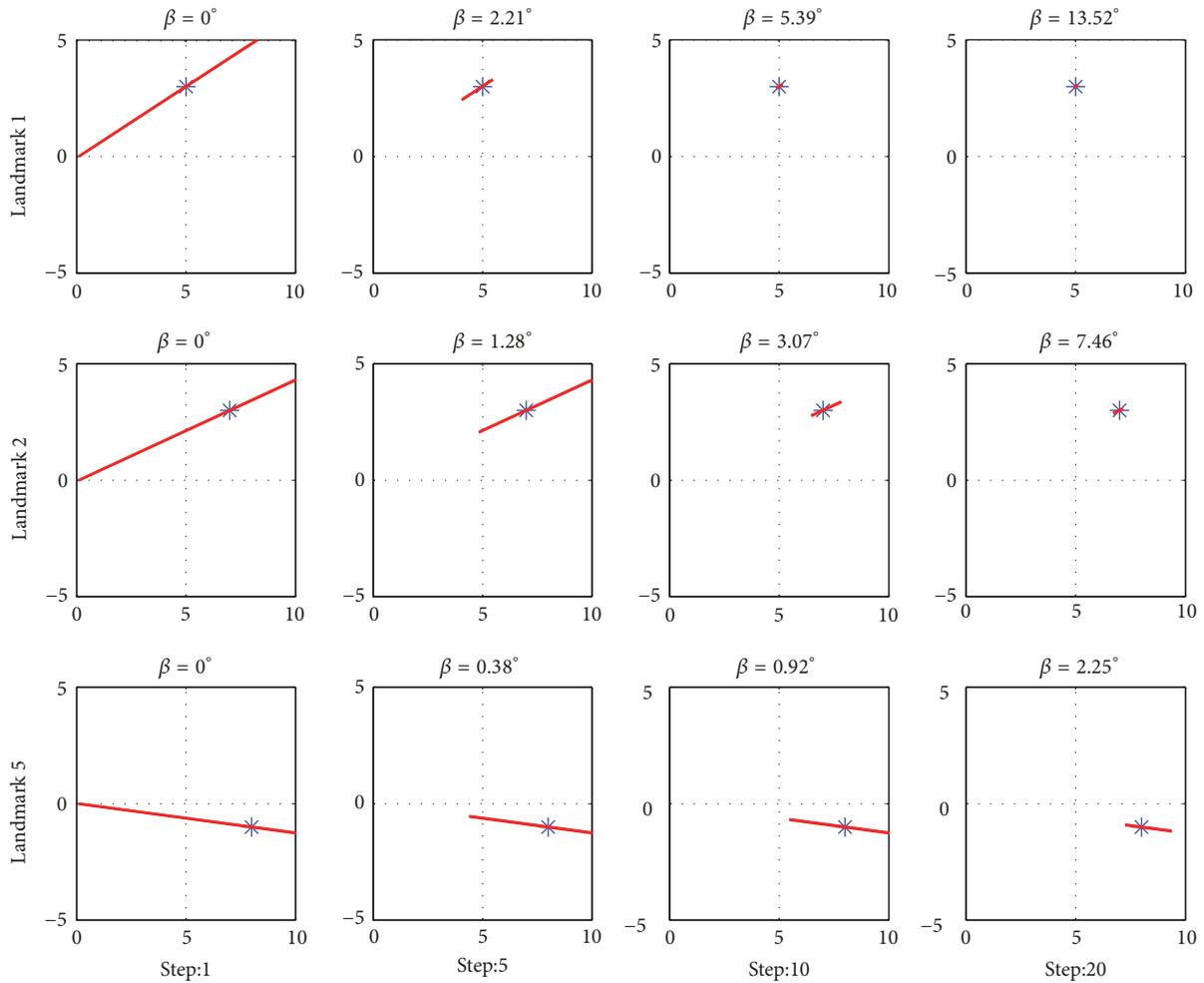


FIGURE 7: Interval uncertainty evolution.

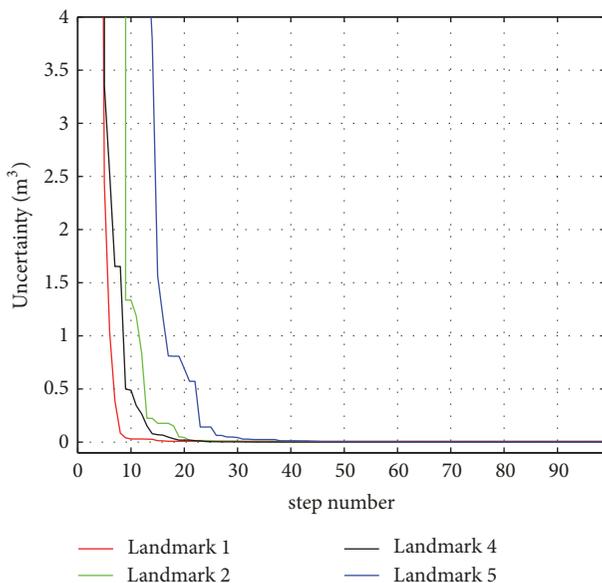


FIGURE 8: Landmark position uncertainty evolution.

the interval uncertainty of each landmark is well propagated and decreases exponentially with the vehicle navigating in the environment.

4.3. Estimation Accuracy. In real world, the vehicle pose is always accompanied with uncertainties due to the cumulative odometric noise. The vehicle pose uncertainty will affect the estimation accuracy of the landmarks (a smaller vehicle pose uncertainty means higher landmarks estimation accuracy). In our simulation, the error bound of the odometric measurement is supposed to be proportional to elementary displacement, such that $w([\mathcal{U}_k]) \leq \gamma \cdot \mathcal{U}_k + \Gamma$, $\Gamma = [0.0001m, 0.01^\circ]^T$, with \mathcal{U}_k being the real odometric measurement, γ denoting the ratio value (bigger γ value indicates a relative bigger uncertainty of odometric measurements), and Γ accounting for the steady system error. The average position uncertainty of the 4 landmarks has been computed at each time step. Figure 9 displays the results with different γ value ($\gamma = 0.01, 0.1$ and 0.2). As it is shown, the estimation accuracy decreases when γ value becomes bigger. This result supports the fact that the landmark estimation accuracy is strongly correlated with the vehicle pose uncertainty. Bigger

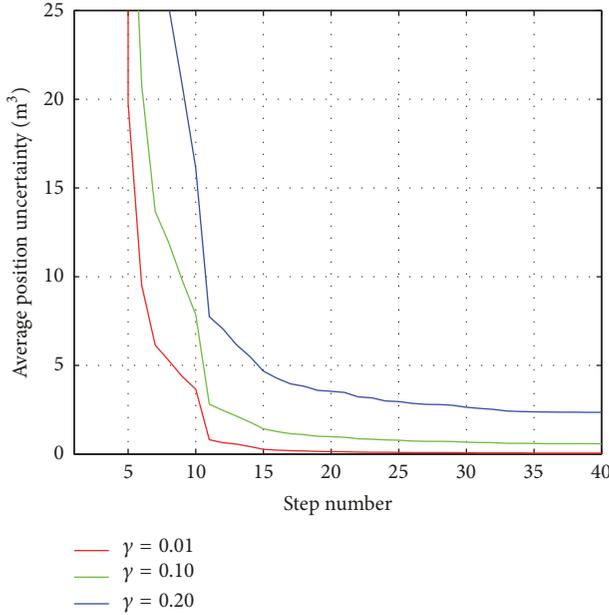


FIGURE 9: Average landmark uncertainty under different γ values.

odometric uncertainty will result in a worse estimation of vehicle pose; as a result, the landmark estimation accuracy decreases.

5. Real Data-Set Experiment

5.1. Experiment Set-Up. To evaluate the performance of our method in realistic circumstance, we used the experimental data set from Institute Pascal [38]. It contains multisensory timestamped data which can be used in a large variety of applications. It includes data from low-cost sensors, ground truth, and multiple sequences as well as data for calibration. Data were collected on the VIPALAB platform (see Figure 10(a)), driving on the PAVIN track (see Figure 11(a)), an experimental site located on the French campus of Blaise Pascal University. The environment is composed of scaled street, road markings, functional traffic lights, painted walls, buildings vegetation, and so on (see Figure 11(b)). The platform was equipped with multiple sensors (see Figure 10(b)). In our experiment, we need to use 4 sets of timestamped sensor data:

- (i) Odometers: each rear wheel is equipped with an odometer in the form of a ring gear providing 64 “top” per wheel revolution.
- (ii) Gyro (Melexis MLX90609-N2): it measures the rotational speed in an inertial reference system. Vibrating silicon structures, that use the Coriolis force, output the yaw rate from which we can deduce the elementary rotation.
- (iii) Camera (FOculus FO432B camera + Pentax C418DX lens): it provides information about the perceived environment. We use the camera which located in the left-front of the vehicle and looking ahead.

TABLE 1: Average bounding boxes size.

Parameter	x	y	z
Mean width (m)	0.207	0.221	0.013

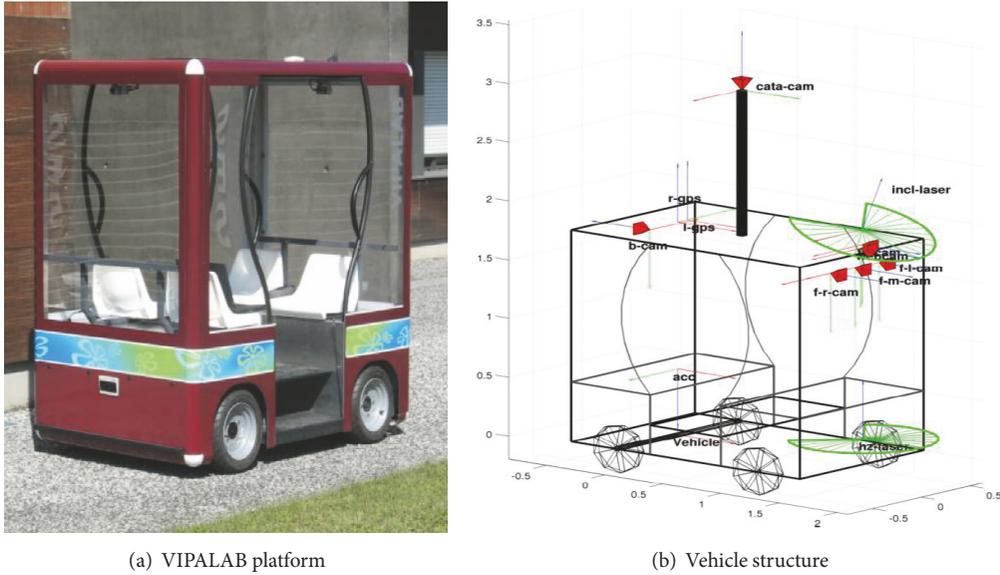
- (iv) RTK-GPS: the data acquired with an embedded ProFlex500 RTK-GPS receiver from NavtechGPS coupled with a Sagitta Magellan GPS base station are considered as the ground truth. This system provides an accurate absolute localization measurement to within 2 cm (± 1 cm).

5.2. Mapping Stage. A mapping stage is first conducted by our method in order to build a map of the environment. The vehicle follows 6 simple consecutive loops around the PAVIN environment. Figure 12(a) shows the reference trajectory that the vehicle passed by and Figure 12(b) details the 6 loops in order. As we have discussed in Section 4.3, the estimation of landmark is strongly correlated with the vehicle pose uncertainty. So it is necessary to maintain the pose uncertainty in a reasonable size during the mapping stage in order to pursue a precise map.

Figure 13 shows the vehicle position uncertainty (calculated by $(w([x])/2) \cdot (w([y])/2)$) with and without RTK-GPS correction during the mapping process. As we can see, without correction, the position uncertainty is cumulative and increases very quickly until loop closure is realized. This is because when navigating, the odometric noise is cumulative since there is no absolute measurement. Interval analysis is a pessimistic method maintaining all possible solutions, the cumulative errors will result in interval expansion, and the pose uncertainty thus becomes larger and larger. This is a hinder to pursue a good map. To overcome this problem, our method makes use of the RTK-GPS measurement, and the GPS error bounds are characterized as the maximum imprecision ($\leq 2cm$) of the GPS receiver. The cumulative errors can be eliminated and a precise estimation of the vehicle position is obtained (see Figure 13(b)). The result is assumed to be consistent since only ICP technique was used.

At the end of the loop, 3450 images are processed and a map of 1140 landmarks is created. Figure 14 shows part of the map projected on $x - y$ plane. The black bounding boxes denote the estimated landmark’s position. The average size of these bounding boxes is shown in Table 1, which can be regarded as a factor to evaluate the quality of the map. The z coordinate gets much higher precision than x and y ; this is because the estimation of z is only correlated with the observation elevation angle which is invariant w.r.t the vehicle’s heading angle, while x and y are affected by the heading uncertainty. A good way to improve the map is to get a better estimation of the vehicle’s orientation. Note that the map building process could be done offline such that computation intensive ICP algorithms and manually guidance could be used to obtain a better map.

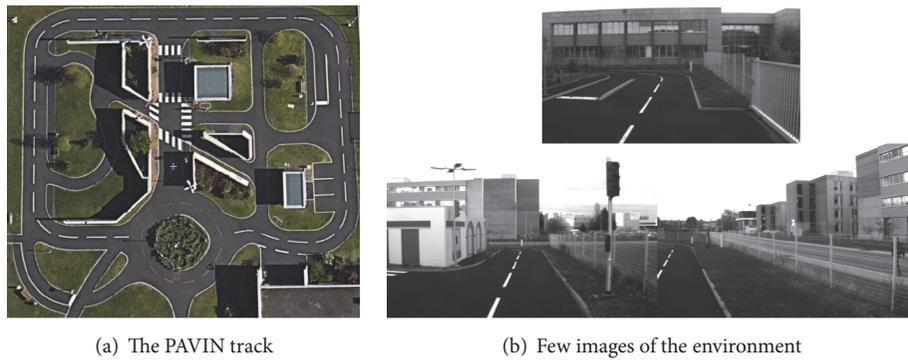
5.3. Localization Result. The localization stage benefits from the map data which offer position constraints. With the built map, the vehicle seeks consistent localization using the



(a) VIPALAB platform

(b) Vehicle structure

FIGURE 10: VIPALAB Experimental platform.



(a) The PAVIN track

(b) Few images of the environment

FIGURE 11: Institute Pascal Experimental Environment.

current image captured by the monocular camera. Three sequences of data from the whole data set are used to test our localization method. The reference path (RTK-GPS) of the vehicle when these data are collected is illustrated in Figures 15, 18(a), and 18(b).

Our proposed localization method is firstly tested on Sequence 1, and the vehicle follows a nearly 200m trajectory; see Figure 15. The blue path is the reference trajectory from RTK-GPS, the red path is the dead reckoning result, and the black squares are the localization boxes output by our proposed interval method. The dead reckoning result is obtained via a geometric evolution model using the odometer and steering angle data. It can be seen from the figure that the localization boxes followed the reference trajectory in a consistent way (intersect with the blue line). Table 2 (DR: dead reckoning; IM: interval method) presents the localization result of both methods at $t = 200$ and $t = 400$ and at the end of the track. By fusing the map, our interval

method gives a consistent estimation of the vehicle pose: the localization boxes well enclose the ground truth.

To compare the vehicle position estimated by our proposed method and the dead reckoning, the root of sum square error (RSSE) is computed at each time step. It is expressed by the following formula:

$$RSSE_k = \sqrt{E_k^2(x) + E_k^2(y)} \quad (33)$$

where $E_k(x)$ and $E_k(y)$ are the estimation error of $\text{mid}([x])$ position and $\text{mid}([y])$ position at time step k . RTK-GPS data is regarded as the ground truth.

Figure 16 shows the RSSE results of both methods. As it could be expected, the cumulative odometric error of dead reckoning is significant and caused an increasing discrepancy (at the end, the discrepancy reaches about 3 meters). Our method takes advantage of the map data and corrects the odometric error, providing a very small discrepancy.

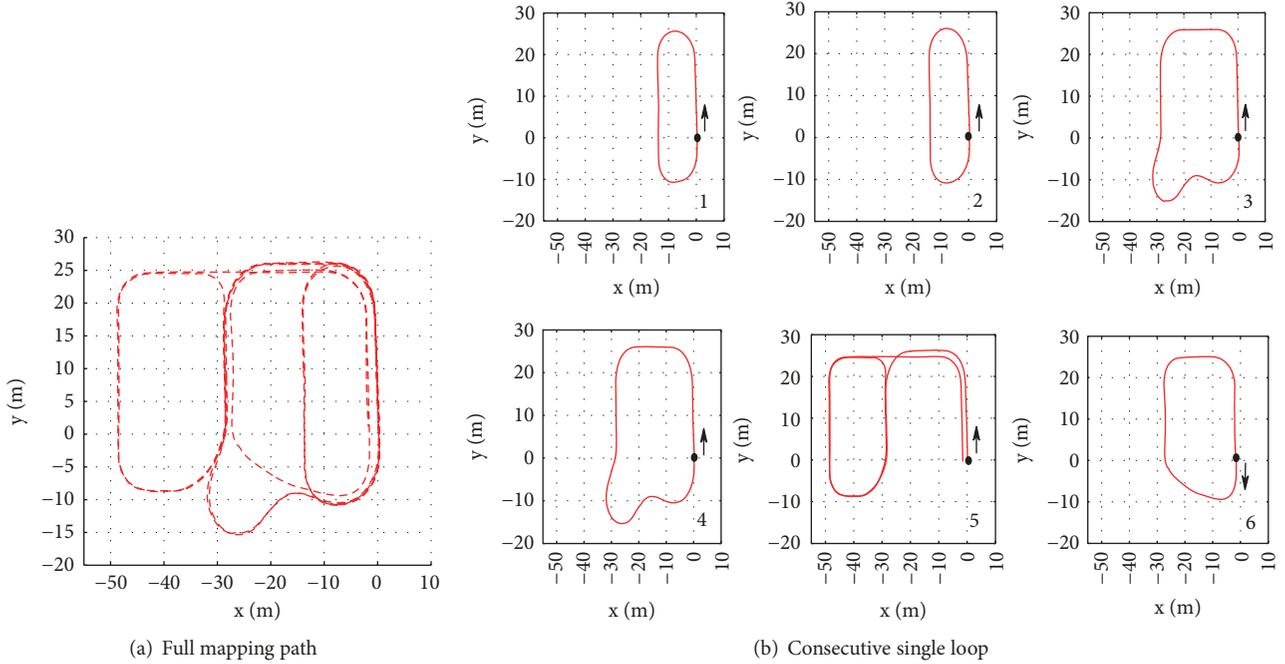


FIGURE 12: The reference path (RTK-GPS) of mapping sequence.

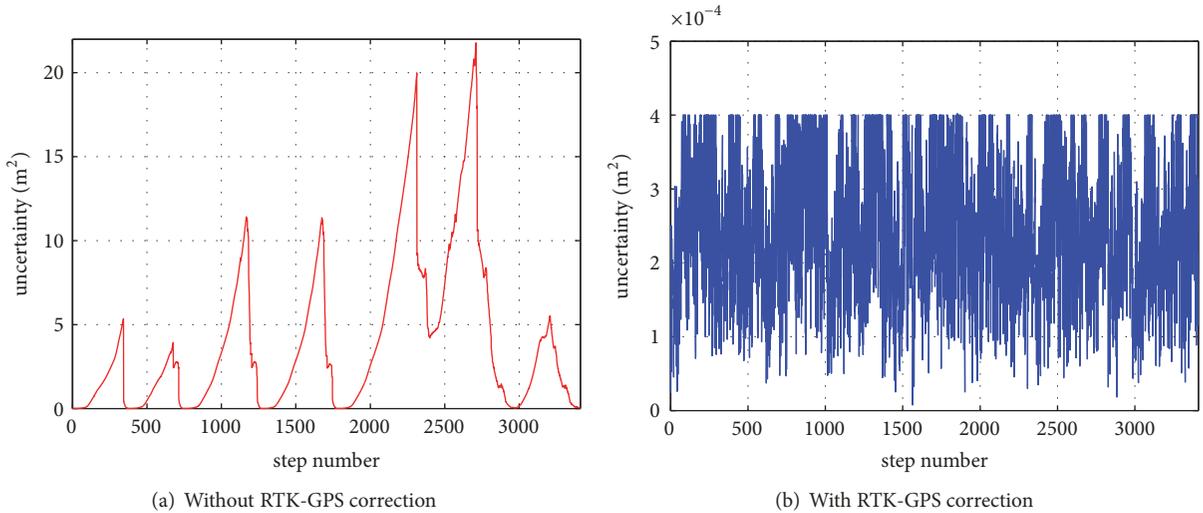


FIGURE 13: Vehicle position uncertainty.

In order to verify the consistency of the localization results, interval error is defined by the upper and lower bounds of the estimated state minus the corresponding reference (real value), mathematically,

$$[E_k(\alpha)] = [\underline{\alpha}_k - \alpha_k^*, \overline{\alpha}_k - \alpha_k^*] \quad (34)$$

where $[\underline{\alpha}_k, \overline{\alpha}_k]$ is the estimated state of variable α at time step k and α_k^* is the corresponding reference. The estimated state is said to be consistent if and only if $0 \in [E_k(\alpha)]$ is a true statement. Moreover, an estimation method is consistent and

precise if $\forall k \in \{1, 2, \dots, \infty\}, 0 \in [E_k(\alpha)]$ is always satisfied and the interval error width $w([E_k(\alpha)])$ is thin.

We compute the real-time interval error of our localization result. At each time instant k when the new captured image had been processed, $E_k(x)$ and $E_k(y)$ are calculated, respectively. Figure 17 depicts the interval corridors consisting of the upper and lower bounds of E_k . The zero line is well included by those corridors, proving that the localization results are consistent all along the track. This is a significant result in vehicle localization field where safety is a crucial issue.

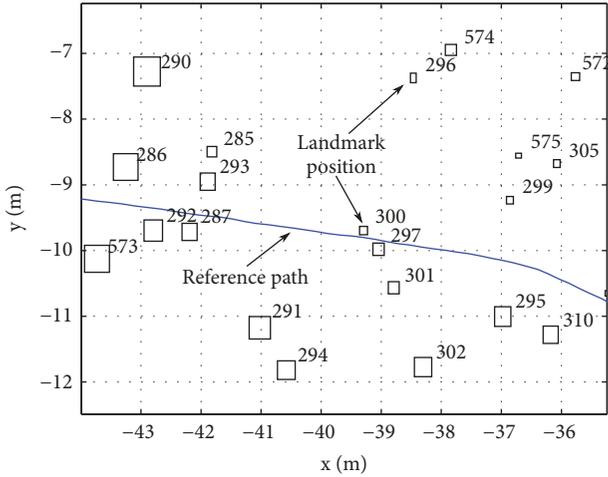


FIGURE 14: Map building result.

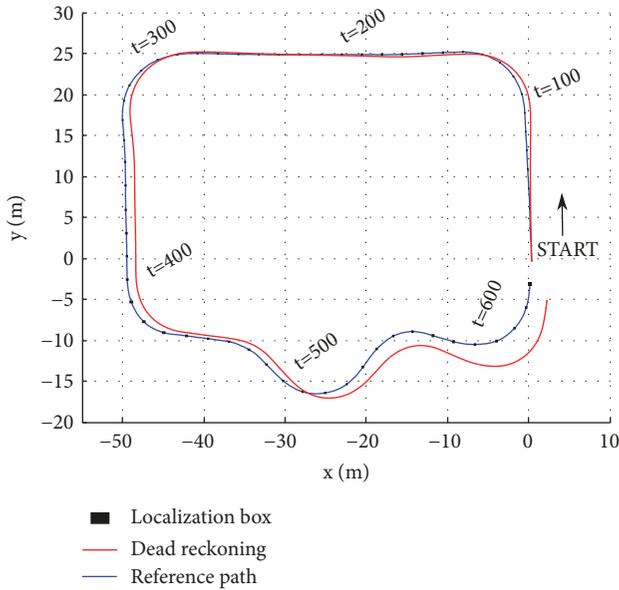


FIGURE 15: Localization trajectory of Sequence 1.

TABLE 2: Localization result.

Time	Position	RTK	DR	IM
t=200	x	-22.32	-22.72	[-22.35, -22.26]
	y	25.17	25.23	[25.14, 25.22]
t=400	x	-49.23	-48.13	[-49.29, -49.18]
	y	-2.51	-2.31	[-2.63, -2.45]
End	x	0.14	1.89	[0.10, 0.18]
	y	-3.26	-5.23	[-3.32, -3.21]

Similar results are obtained when performing the localization process on Sequences 2 and 3 with the same map; the output localization boxes by our method and the corresponding reference trajectory are displayed in Figure 18. It shows the robustness of our method providing consistent

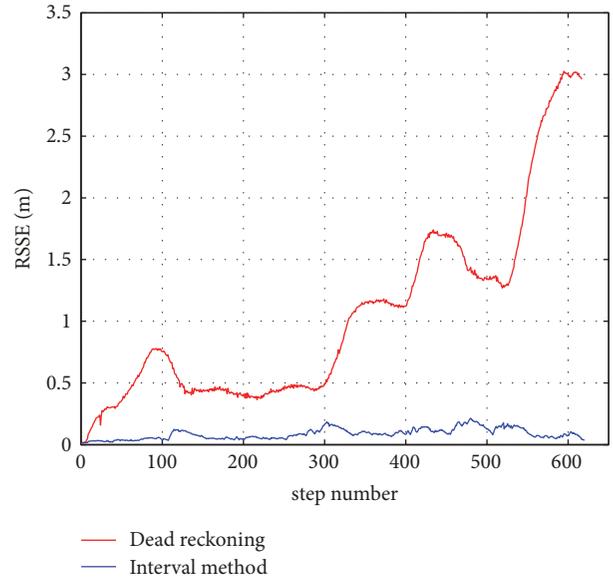


FIGURE 16: The error of localization results by DR and IA.

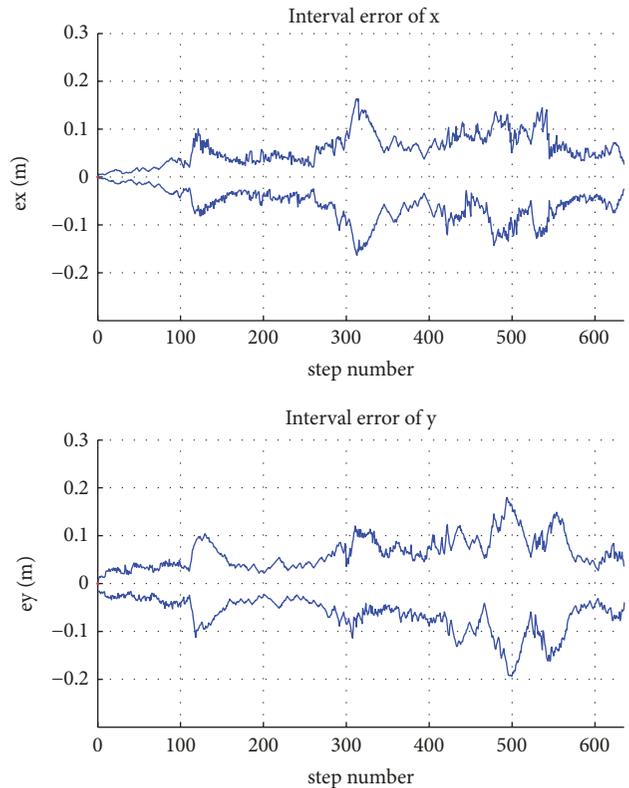


FIGURE 17: Interval error of real-time localization result.

localization result. The width of the interval error corridors (see Figure 17) with respect to the vehicle's x position and y position is displayed in Figure 19, reporting a maximum value of 0.6 m. The average interval error width and mean RSSE are computed to evaluate the localization accuracy. Table 3 details the results of the 3 experiments. Computing an average value for the 3 experiments gives 12.7 cm accuracy for x and

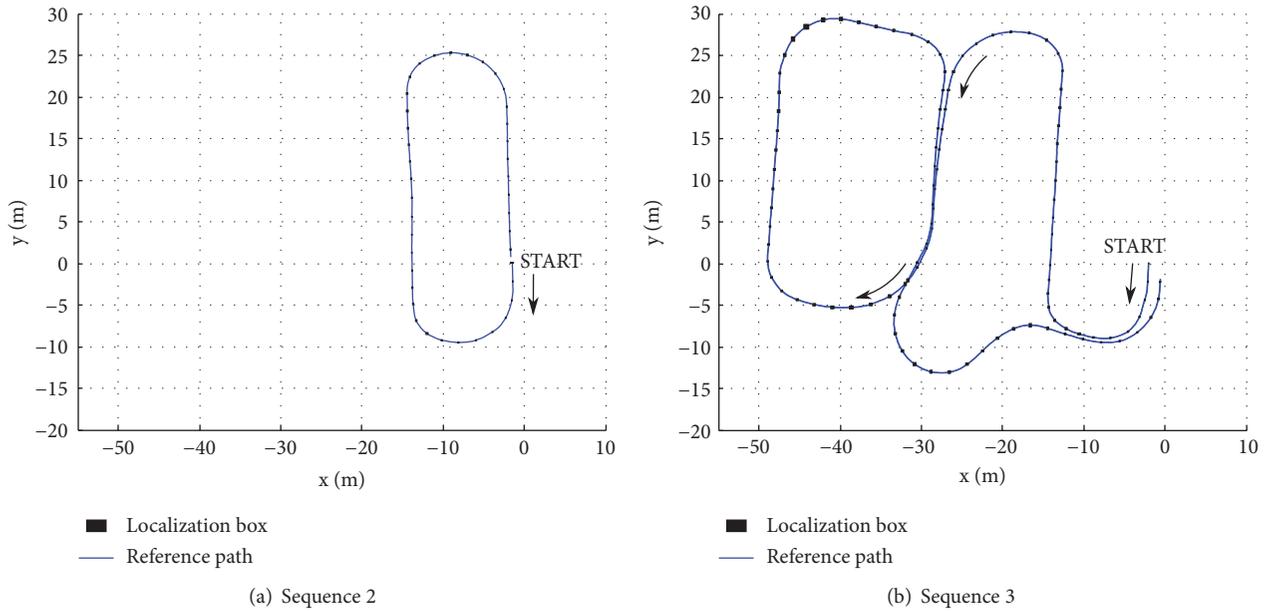


FIGURE 18: Localization boxes compare with reference path.

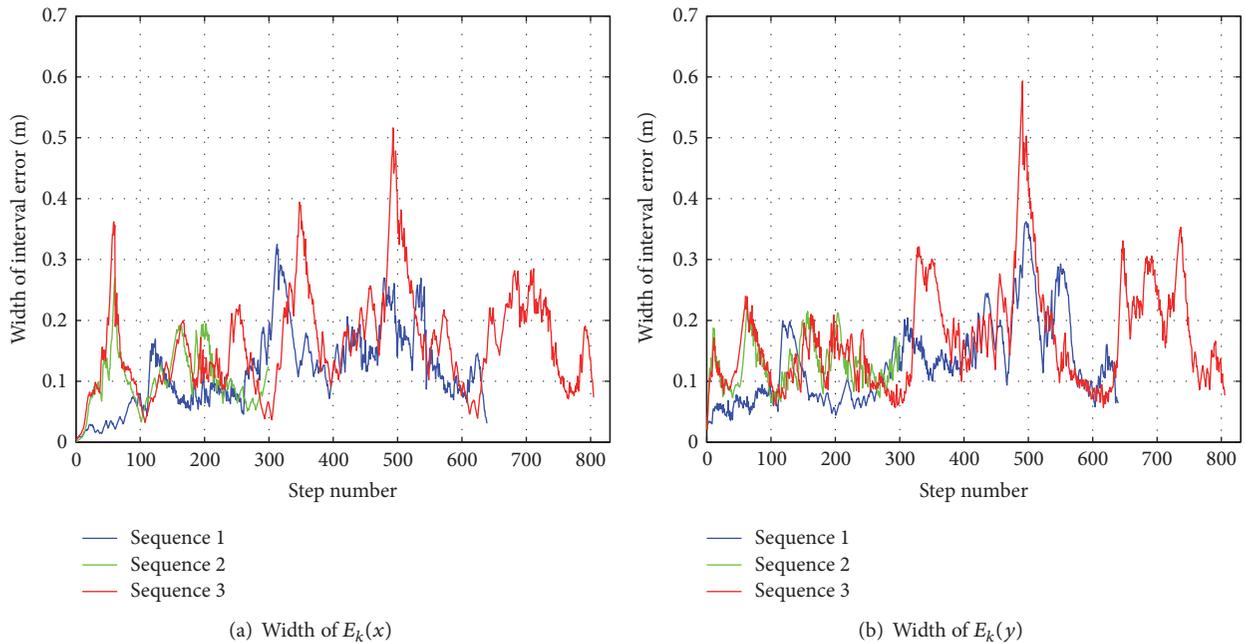


FIGURE 19: Interval error width of localization result.

14.4 cm for y . The average RSSE value is about 9.66 cm (less than 10 cm). Note that these localization results are consistent, and the localization boxes are guaranteed to enclose the real positions of the vehicle.

5.4. Discussion. In this paper, we present a complete methodology for localizing a vehicle within a prebuilt map. We bring forward a two-stage framework (mapping and localization) to solve the problem. Closer approaches have already been proposed by different researchers [39–41]. They all involve a

TABLE 3: Mean localization interval error and RSSE.

Experiment	x (cm)	y (cm)	RSSE (cm)
Sequence 1	11.7	13.4	8.90
Sequence 2	10.3	12.7	8.29
Sequence 3	16.1	17.2	11.8

visual learning step to reconstruct a map of the environment, and they use this map for localization and navigation task.

Eric Royer et al. [39] uses bundle adjustment in mapping step and the localization results are obtained via Newton iteration method. Hyon Lim et al. [41] uses a structure from motion (SFM) algorithm to reconstruct the 3D environment; then in the localization step, two discrete Kalman filters are employed to estimate the camera trajectory. Courbon's method [40] does not involve the 3D reconstruction procedure during the map building step; instead, it records some key views and the performed path as references and uses them for future navigation missions. Our method uses totally different theory and techniques from these approaches, we cast the two-stage problem into an ICSP, and deterministic techniques (ICP) are used in order to find the solution.

The localization accuracy of these works is close to ours. The paper [39] has an average localization error of 15 cm over 6 localization experiments on 3 outdoor sequences. The reported mean indoor localization errors of [41] are ranged between 5.6 cm and 10.8 cm among 6 experiments. The outdoor mean position error is less than 15 cm. The paper [40] presents an average lateral error of 25 cm on an urban vehicle navigating along a 750 m trajectory. Our method has been tested on 3 outdoor sequences; the position error (RSSE) ranges between 8.29 cm and 11.8cm. The strength of our method is that the localization boxes are guaranteed to contain the real position of the vehicle. Such a guarantee is due to the modeling of the noises as intervals and the computing of constraints using interval functions.

The formulas of our algorithm work whatever the camera manufacturer is. As for different cameras, the observation model always holds; only the camera intrinsic parameters (f, k_u, k_v, c_u, c_v) change. Furthermore, the feature points extracted from images of various cameras would be the same since the SURF algorithm is scale invariant, rotation invariant, and affine transformation invariant [33]. This property makes it possible to find correct correspondence between two images from different cameras. This feasibility has been shown by [39] in two sequences of images taken by a pair of cameras: one sequence was used in the map learning step, and the other one was used for vehicle localization; the result turns out to be robust.

6. Conclusion

In order to overcome the inconsistency problem of probabilistic methods, this paper proposed a consistent vehicle localization method based on interval analysis. We propose a 2-stage process: mapping and localization. The map is built by a leader vehicle equipped with a sensor or a set of sensors providing an accurate positioning. The built map can be beneficial for a fleet of vehicles intending to achieve a consistent localization in the environment. By fusing low-cost dead reckoning sensors and map data, the proposed localization method is able to locate the vehicle consistently in the environment using the current image taken by the camera. The cumulative odometric error is eliminated and the localization consistency is maintained. Our method is validated with an outdoor car like vehicle equipped with odometers, gyro, and monocular camera. The experiments highly illustrate the consistency of the localization results

provided by our method. Future work will deal with a formal proof of the obtained consistency.

Our methodology can be used to solve various problems, even problems with higher DOF. Indeed, the used ICP algorithms express a polynomial time with respect to the number of variables in the ICSP [42], so our ICP based method is capable of scaling up to higher DOF cases without high CPU load. In order to adapt to other applications, we have only to define a new motion model.

Data Availability

Data used in this manuscript are openly available at "http://ipds.univ-bpclermont.fr/"; see [38] for more details.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] M. Adams, S. Zhang, and L. Xie, "Particle filter based outdoor robot localization using natural features extracted from laser scanners," in *Proceedings of the Proceedings- 2004 IEEE International Conference on Robotics and Automation*, pp. 1493–1498, May 2004.
- [2] A. Gning and P. Bonnifait, "Guaranteed dynamic localization using constraints propagation techniques on real intervals," in *Proceedings of the Proceedings- 2004 IEEE International Conference on Robotics and Automation*, pp. 1499–1504, May 2004.
- [3] D. Gruyer, R. Belaroussi, and M. Revilloud, "Map-aided localization with lateral perception," in *Proceedings of the 25th IEEE Intelligent Vehicles Symposium, IV 2014*, pp. 674–680, June 2014.
- [4] A. Howard, M. Matarik, and G. Sukhatme, "Localization for mobile robot teams using maximum likelihood estimation," in *Proceedings of the IROS 2002: IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 434–439, Lausanne, Switzerland, 2002.
- [5] D. Gruyer, A. Lambert, M. Perrollaz, and D. Gingras, "Experimental comparison of bayesian positioning methods based on multi-sensor data fusion," *International Journal of Vehicle Autonomous Systems*, vol. 12, no. 1, pp. 24–43, 2014.
- [6] S. Thrun, W. Burgard, and D. Fox, "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots," *Autonomous Robots*, vol. 5, no. 3-4, pp. 253–271, 1998.
- [7] W. Burgard, D. Fox, D. Hennig, and T. Schmidt, "Estimating the absolute position of a mobile robot using position probability grids," in *Proceedings of the 1996 13th National Conference on Artificial Intelligence*, pp. 896–901, August 1996.
- [8] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001.
- [9] S. Huang and G. Dissanayake, "A critique of current developments in simultaneous localization and mapping," *International Journal of Advanced Robotic Systems*, vol. 13, no. 5, pp. 1–13, 2016.
- [10] M. Di Marco, A. Garulli, A. Giannitrapani, and A. Vicino, "A set theoretic approach to dynamic robot localization and mapping," *Autonomous Robots*, vol. 16, no. 1, pp. 23–47, 2004.
- [11] R. E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1966.

- [12] 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*, vol. 42, no. 7, pp. 1167–1175, 2006.
- [13] L. Jaulin, *Applied Interval Analysis: with Examples in Parameter And State Estimation, Robust Control And Robotics*, Springer, 2001.
- [14] T. Rassi, N. Ramdani, and Y. Candau, "Set membership state and parameter estimation for systems described by nonlinear differential equations," *Automatica*, vol. 40, no. 10, pp. 1771–1777 (2005), 2004.
- [15] M. Kieffer, L. Jaulin, É. Walter, and D. Meizel, "Robust autonomous robot localization using interval analysis," *Reliable Computing. An International Journal Devoted to Reliable Mathematical Computations Based on Finite Representations and Guaranteed Accuracy*, vol. 6, no. 3, pp. 337–362, 2000.
- [16] 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)*, vol. 32, no. 4, pp. 374–381, 2002.
- [17] M. Kieffer, E. Seignez, A. Lambert, E. Walter, and T. Maurin, "Guaranteed robust nonlinear state estimator with application to global vehicle tracking," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*, pp. 6424–6429, esp, December 2005.
- [18] E. Seignez, M. Kieffer, A. Lambert, E. Walter, and T. Maurin, "Real-time bounded-error state estimation for vehicle tracking," *International Journal of Robotics Research*, vol. 28, no. 1, pp. 34–48, 2009.
- [19] A. Lambert, D. Gruyer, B. Vincke, and E. Seignez, "Consistent outdoor vehicle localization by bounded-error state estimation," in *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 1211–1216, October 2009.
- [20] I. K. Kueviakoe, A. Lambert, and P. Tarroux, "A real-time interval constraint propagation method for vehicle localization," in *Proceedings of the 2013 16th International IEEE Conference on Intelligent Transportation Systems: Intelligent Transportation Systems for All Modes, ITSC 2013*, pp. 1707–1712, nld, October 2013.
- [21] V. Drevelle and P. Bonnifait, "Robust positioning using relaxed constraint-propagation," in *Proceedings of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010*, pp. 4843–4848, twm, October 2010.
- [22] K. Lassoued, P. Bonnifait, and I. Fantoni, "Cooperative Localization with Reliable Confidence Domains Between Vehicles Sharing GNSS Pseudorange Errors with No Base Station," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 22–34, 2017.
- [23] B. Desrochers and L. Jaulin, "A minimal contractor for the polar equation: Application to robot localization," *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 83–92, 2016.
- [24] L. Jaulin, A. Stancu, and B. Desrochers, "Inner and outer approximations of probabilistic sets," in *Proceedings of the American Society of Civil Engineers (ASCE) Second International Conference on Vulnerability and Risk Analysis and Management and Sixth International Symposium on Uncertainty Modelling and Analysis*, pp. 13–16.
- [25] L. Jaulin, "Range-only SLAM with occupancy maps: A set-membership approach," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 1004–1010, 2011.
- [26] P. T. I. K. Kueviakoe and A. Lambert, "Vehicle localization based on interval constraint propagation algorithms," in *Proceedings of the IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, pp. 179–184, May 2013.
- [27] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, Society for Industrial Mathematics, 2009.
- [28] F. Le Bars, E. Antonio, J. Cervantes, C. De La Cruz, and L. Jaulin, "Estimating the trajectory of low-cost autonomous robots using interval analysis: Application to the eurathlon competition," in *Marine Robotics and Applications*, pp. 51–68, Springer, 2018.
- [29] E. Hyvönen, "Constraint reasoning based on interval arithmetic: the tolerance propagation approach," *Artificial Intelligence*, vol. 58, no. 1-3, pp. 71–112, 1992.
- [30] A. Lambert and N. Le Fort-Piat, "Safe task planning integrating uncertainties and local maps federations," *International Journal of Robotics Research*, vol. 19, no. 6, pp. 597–611, 2000.
- [31] B. Vincke, A. Lambert, D. Gruyer, A. Elouardi, and E. Seignez, "Static and dynamic fusion for outdoor vehicle localization," in *Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010*, pp. 437–442, December 2010.
- [32] J. M. M. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular SLAM," in *Proceedings of the 2nd International Conference on Robotics Science and Systems (RSS '06)*, pp. 81–88, August 2006.
- [33] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [34] A. Schmidt, M. Kraft, and A. Kasiński, "An Evaluation of Image Feature Detectors and Descriptors for Robot Navigation," in *Computer Vision and Graphics*, pp. 251–259, 2010.
- [35] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd edition, 2003.
- [36] Z. Dan and H. Dong, "A robust object tracking algorithm based on surf," in *Proceedings of the IEEE International Conference on Wireless Communications & Signal Processing (WCSP)*, pp. 1–5, 2013.
- [37] J. P. Lewis, "Fast normalized cross-correlation," *Vision Interface*, vol. 10, pp. 120–123, 1995.
- [38] H. Korrapati, J. Courbon, S. Alizon, and F. Marmoiton, "The institut pascal data sets: un jeu de données en extérieur, multicapteurs et datées avec réalité terrain, données d'étalement et outils logiciels," in *Orasis, Congrès des jeunes chercheurs en vision par ordinateur*, 2013.
- [39] E. Royer, M. Lhuillier, M. Dhome, and T. Chateau, "Localization in urban environments: Monocular vision compared to a differential GPS sensor," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. 2, pp. 114–121, June 2005.
- [40] J. Courbon, Y. Mezouar, and P. Martinet, "Autonomous navigation of vehicles from a visual memory using a generic camera model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 392–402, 2009.
- [41] H. Lim, S. N. Sinha, M. F. Cohen, M. Uyttendaele, and H. J. Kim, "Real-time monocular image-based 6-DoF localization," *International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 476–492, 2015.
- [42] G. Chabert and L. Jaulin, "Contractor programming," *Artificial Intelligence*, vol. 173, no. 11, pp. 1079–1100, 2009.



Hindawi

Submit your manuscripts at
www.hindawi.com

