Guaranteed Characterization of the Explored Space of a Mobile Robot by using Subpavings

Vincent Drevelle [*] **Luc Jaulin** [*] **Benoit Zerr** [*]

[*] *Lab-STICC - ENSTA Bretagne, Brest, France*
(e-mail: vincent.drevelle@ensta-bretagne.fr, luc.jaulin@ensta-bretagne.fr, benoit.zerr@ensta-bretagne.fr)

Abstract: This paper proposes a method to characterize the space explored by a mobile robot during a mission. Because of localization uncertainty, the area osculated by a sensor at a given time is uncertain too. The problem is modeled by using intervals to represent trajectory uncertainties and a "visibility function" to describe the area in view at a given time. A set-inversion method is then applied to compute a "guaranteed visible area" and a "possible visible area" with respect to positioning uncertainty. A bracketing of the actual explored area between a "guaranteed explored area" and a "possible explored area" for the whole mission is finally obtained by respectively taking the union of the guaranteed and possible areas. Results from a simulated underwater exploration mission are presented.

Keywords: interval analysis, mobile robots, exploration, sets, mapping, uncertainty

1. INTRODUCTION

Mobile robots are often the only safe way to perform measurements and observations in dangerous, hard to reach or extreme environments, like underwater or radioactive areas. The development of autonomous robots has also led to new applications in domestic environments. Some missions require a robot to fully cover a defined area with its embedded sensors or actuators. This is the case for mapping, search and recovery of lost items, but also lawn-mowing and vacuum-cleaning.

Estimation of the area that has been sensed (or touched by the robot's tool) enables to assess the completion of an area-covering mission. Since tools and sensors are attached to the body-frame of the robot, the area that has been covered during the mission is linked to the robot's trajectory. Estimation of the robot's explored area thus depends on trajectory accuracy. The uncertainty of position estimates needs to be taken into account when trying to characterize the explored area, which is the aim of this paper.

Set-membership methods have successfully been used to reliably deal with uncertainty, especially in non-linear problems. They have been employed for robot localization (see, *e.g*, [Bouron et al., 2001] for the localization of a moving car, Caiti et al. [2002] for an underwater positioning system, and Colle and Galerne [2013] for an application with heterogeneous measurements), but also simultaneous localization and mapping [Jaulin, 2011], fault detection and identification [Drevelle and Bonnifait, 2012] and computer vision [Telle et al., 2003]. Problems such as state estimation of non-linear systems have successfully been addressed [Raıssi et al., 2004] even in the presence of fleeting data [Le Bars et al., 2012], and interval based methods have been applied in control of uncertain dynamical systems [Rauh et al., 2008].

In this paper, an interval-based set-membership method is proposed to compute the area explored by a mobile robot and characterize its uncertainty. The explored area problem is introduced in Section 2. In Section 3, intervals and associated tools for solving set-inversion problems are presented. The explored area characterization with uncertain localization is then detailed and solved by using interval methods in Section 4. Finally, in Section 5, the method is applied to two simulated test cases of representative underwater exploration missions, and results are reported.

2. PROBLEM STATEMENT

Let us consider a mobile robot equipped with actuators and sensors. The robot is classically represented by the following state equations

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \\ \mathbf{y}(t) = \mathbf{g}\left(\mathbf{x}(t)\right), \end{cases} \tag{1}$$

where $\mathbf{x}$ denotes the robot's state vector (e.g. position, velocity...), $\mathbf{u}$ is the input vector and $\mathbf{y}$ is the observation vector. The robot's evolution is modeled by the function $\mathbf{f}$ and $\mathbf{g}$ is the observation function.

The robot uses a sensor for searching or mapping, that can cover a given area. The area auscultated by the robot's sensor depends on its technology, geometry and physics, but also on the robot's state (position, velocity...). Let $\mathbb{V}$ be the set-valued function that returns the area which is in the robot's field of view at each time. $\mathbb{V}$ is defined as

$$\mathbb{V}(t) = \left\{\mathbf{z} \in \mathbb{R}^2 : v\left(\mathbf{z}, \mathbf{x}(t)\right) \leq 0\right\}. \tag{2}$$

The set $\mathbb{V}(t)$ is called the *visible area* at time t, *i.e.* the points of the sensed space that are in the range of the robot's sensor (Fig. 1). The visible area is defined by the *visibility function* v, such that $v\left(\mathbf{z}, \mathbf{x}(t)\right)$ is negative if and only if the point $\mathbf{z}$ of the sensed space is in the range of the sensor for the given the robot state $\mathbf{x}(t)$.

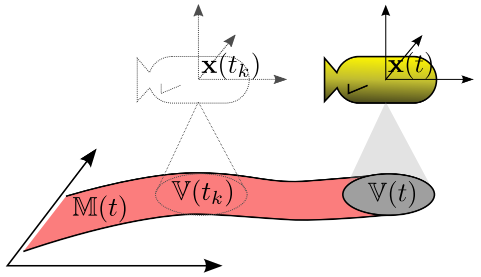


Fig. 1. Visible areas at times t and t_k, and current explored area $\mathbb{M}(t)$.

As the robot moves, its visible area changes accordingly, which enables to explore space. Let $\mathbb{M}(t)$ be the map of the *current explored area* at time t, *i.e* the set of all points in the sensed space that have been in the range of the robot's sensor at least once until time t. It is shown in red in Figure 1. $\mathbb{M}(t)$ is the union of all the visible areas over the whole trajectory, from the mission start t_0 to current time t:

$$\mathbb{M}(t) = \bigcup_{\tau \in [t_0, t]} \mathbb{V}(\tau). \tag{3}$$

The *mission's explored area* $\mathbb{M}$ is the area that has been covered by the robot's sensor at the end of the exploration mission. With t_f denoting the time at the end of mission, the mission's explored area is defined such as $\mathbb{M} = \mathbb{M}(t_f)$. In this paper, the terms *explored area* refer to the mission's explored area.

In practice, due to measurement noise and model inaccuracies, the robot trajectory is uncertain. This uncertainty can be represented by covariance in Bayesian filtering methods, or by a set of possible values. In this paper, we use intervals to model uncertainty. Next section presents intervals and some basic tools to manipulate them.

3. INTERVAL ANALYSIS AND SET-INVERSION

3.1 Interval analysis

Interval analysis (Moore [1966], see Jaulin et al. [2001] for algorithms and applications) is based upon computations with intervals and their multidimensional extension, *interval vectors* (or *boxes*). Intervals and boxes are easy to represent and manipulate with computer algorithms, contrarily to arbitrary sets. The set of real intervals is denoted $\mathbb{IR}$, and the set of n-dimensional boxes is $\mathbb{IR}^n$. In this paper, real intervals are written between brackets such as $[x] = [\underline{x}, \overline{x}]$, where $\underline{x}$ is the lower bound and $\overline{x}$ is the upper bound of $[x]$. Boxes are in bold: the vectors $\underline{\mathbf{x}}$ and $\overline{\mathbf{x}}$ are respectively the lower and upper bounds of the box $[\mathbf{x}] = [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$.

The width of an interval is defined as $\text{width}([x]) = \overline{x} - \underline{x}$. The width of a box is the width of its longest interval component: $\text{width}([\mathbf{x}]) = \max_i \text{width}([x_i])$.

Interval arithmetic allows performing computations on intervals and boxes thanks to the interval extension of classical real arithmetic operators $+, -, \cdot$ and $\div$.

$$[x] + [y] = [\underline{x} + \underline{y}, \overline{x} + \overline{y}],$$
$$[x] \cdot [y] = [\min\left(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\right), \max\left(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\right)].$$

In the same way, elementary functions such as *tan, sin* and *exp* can be extended to intervals. This is done by returning the smallest interval covering the range of the input through the function.

The image of a box by a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ is generally not itself a box, but an arbitrary set. This problem is solved using the so-called *inclusion functions*: The interval function $[\mathbf{f}]$ from $\mathbb{IR}^n$ to $\mathbb{IR}^m$ is an *inclusion function* for $\mathbf{f}$ if the image of $[\mathbf{x}]$ by $\mathbf{f}$ is included in the image of $[\mathbf{x}]$ by $[\mathbf{f}]$, *i.e.* $\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{x}])$. The n*atural inclusion function* is obtained by replacing each operator in the expression of the function by its interval counterpart.

3.2 Extensions of intervals

In addition to real intervals and boxes, closed intervals can be defined for different sets of mathematical objects, providing they form a complete lattice.

Tubes The set $\mathcal{F} = \{\mathbf{f} : \mathbb{R} \to \bar{\mathbb{R}}^n\}$ of all functions from $\mathbb{R}$ to $\bar{\mathbb{R}}^n$ is a complete lattice with the partial order $\mathbf{f} \leq \mathbf{g} \Leftrightarrow \forall t \in \mathbb{R}, \mathbf{f}(t) \leq \mathbf{g}(t)$. An interval of $\mathcal{F}$ is an interval of functions, called a *tube*.

Set intervals Since the set $2^{\bar{\mathbb{R}}^n}$ of all subsets of $\bar{\mathbb{R}}^n$ is a complete lattice with the partial order $\subset$, *set intervals* can also be defined. A set interval contains all sets that are subsets of its upper-bound and supersets of its lower-bound.

3.3 Set inversion and subpavings

The set inversion problem consists in determining the set $X = \mathbf{f}^{-1}([\mathbf{y}])$ such as $X = \{\mathbf{x} \mid \mathbf{f}(\mathbf{x}) \in [\mathbf{y}])\}$, where $[\mathbf{y}]$ is a known interval vector. To approximate compact sets in a guaranteed way, *subpavings* can be used. A subpaving of $\mathbb{R}^n$ is the union of non-empty and non-overlapping boxes of $\mathbb{R}^n$.

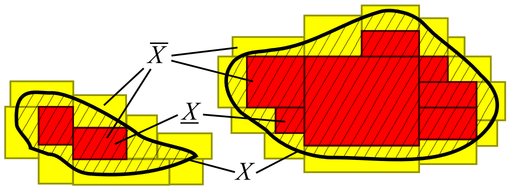


Fig. 2. Approximation of a set X between an inner subpaving $\underline{X}$ and an outer subpaving $\overline{X}$

The solution $X = \mathbf{f}^{-1}([\mathbf{y}])$ can be approximated in a guaranteed way by using interval analysis. The *Set Inversion Via Interval Analysis* algorithm (SIVIA) (Jaulin et al. [2001]) computes two subpavings $\underline{X}$ and $\overline{X}$ such that $\underline{X} \subseteq X \subseteq \overline{X}$, *i.e* a set interval $[\underline{X}, \overline{X}]$ that contains the solution (Fig. 2). The algorithm starts from an arbitrarily big initial box, and consists in recursively testing the inclusion of a box in the solution set. If the box cannot be proven to be inside or outside the solution set, then it is bisected into two boxes. SIVIA is described in Algorithm 1. It uses a list of boxes $\mathfrak{L}$ that is generally implemented as a stack, so that the algorithm employs a depth-first strategy. The parameter ϵ controls the sharpness of the approximation between the two subpavings $\underline{X}$ and $\overline{X}$. The algorithm stops when the width of boxes falls below ϵ.

Algorithm 1 SIVIA(in: $[\mathbf{x}_0], [\mathbf{f}], [\mathbf{y}], \epsilon$, out: $\underline{X}, \overline{X}$)

```
$\underline{X} \leftarrow \emptyset$; $\overline{X} \leftarrow \emptyset$      // inner and outer subpavings
push($[\mathbf{x}_0], \mathfrak{L}$)
while $\mathfrak{L}$ is not empty
    $[\mathbf{x}]$ ←grab the first box of $\mathfrak{L}$
    if $[\mathbf{f}]([\mathbf{x}]) \cap [\mathbf{y}] = \emptyset$ then
        continue
    else if $[\mathbf{f}]([\mathbf{x}]) \subset [\mathbf{y}]$ then
        $\underline{X} \leftarrow \underline{X} \cup [\mathbf{x}]$
    else if width($[\mathbf{x}]$) $< \epsilon$ then
        $\overline{X} \leftarrow \overline{X} \cup [\mathbf{x}]$
    else
        ($[\mathbf{x}_1], [\mathbf{x}_2]$) ← bisect($[\mathbf{x}]$)
        push $[\mathbf{x}_1]$ into $\mathfrak{L}$; push $[\mathbf{x}_2]$ into $\mathfrak{L}$
    endif
end
return $[\underline{X}, \overline{X}]$
```

4. EXPLORED AREA CHARACTERIZATION

4.1 Explored area with an uncertain trajectory

As previously stated, we aim to characterize the explored area, given an uncertain robot trajectory. Let us introduce the robot trajectory tube $[\mathbf{x}](t)$ which encloses the robot trajectory, i.e $\forall t \in [t], \mathbf{x}(t) \in [\mathbf{x}](t)$. This tube is a bounded estimate of the robot state at each time. The uncertain explored area mapping problem is now defined as follows:

$$\begin{cases} \mathbf{x}(t) & \in & [\mathbf{x}](t) \\ \mathbb{V}(t) & = & \left\{\mathbf{z} \in \mathbb{R}^2 : v(\mathbf{z}, \mathbf{x}(t)) \leq 0\right\} \\ \mathbb{M}(t) & = & \bigcup\limits_{\tau \in [t_0, t]} \mathbb{V}(\tau). \end{cases} \tag{4}$$

Uncertainty about the robot state leads to uncertainty about its visible area. We thus define the *guaranteed visible area* $\mathbb{V}^\forall$ as the set of points that have necessarily been observed, regardless of the robot's state uncertainty:

$$\mathbb{V}^\forall(t) = \left\{\mathbf{z} \in \mathbb{R}^2 : \forall \mathbf{x} \in [\mathbf{x}](t), v(\mathbf{z}, \mathbf{x}) \leq 0\right\} \tag{5}$$

The guaranteed visible area is the set of points in the sensed space for which the visibility function is negative for all states in the state uncertainty domain. Similarly, the *possible visible area* $\mathbb{V}^\exists$ is the set of all the points that may have been in the robot's field of view:

$$\mathbb{V}^\exists(t) = \left\{\mathbf{z} \in \mathbb{R}^2 : \exists \mathbf{x} \in [\mathbf{x}](t), v(\mathbf{z}, \mathbf{x}) \leq 0\right\} \tag{6}$$

A point of the sensed space belongs to the possible visible area if it exists a state in the state uncertainty domain for which the visibility function of this point is negative.

Proposition 1. $\mathbb{V}^\forall(t)$ and $\mathbb{V}^\exists(t)$ form a bracketing of the actual visible area $\mathbb{V}(t)$. They are the lower and upper bounds of a set interval that contains the visible area:

$$\forall t \in [t], \mathbb{V}^\forall(t) \subset \mathbb{V}(t) \subset \mathbb{V}^\exists(t) \tag{7}$$

Proof. Let us first prove that $\mathbb{V}^\forall(t) \subset \mathbb{V}(t)$:

$$\begin{aligned} \mathbf{p} \in \mathbb{V}^\forall(t) & \overset{(5)}{\Leftrightarrow} && \forall \mathbf{x} \in [\mathbf{x}](t), v(\mathbf{p}, \mathbf{x}) \leq 0 \\ & \overset{\mathbf{x}(t) \in [\mathbf{x}](t)}{\Rightarrow} && v(\mathbf{p}, \mathbf{x}(t)) \leq 0 \\ & \Rightarrow && \mathbf{p} \in \mathbb{V}(t) \end{aligned}$$

Now let us prove that $\mathbb{V}(t) \subset \mathbb{V}^\exists(t)$:

$$\begin{aligned} \mathbf{p} \in \mathbb{V}(t) & \Leftrightarrow && v(\mathbf{p}, \mathbf{x}(t)) \leq 0 \\ & \overset{\mathbf{x}(t) \in [\mathbf{x}](t)}{\Rightarrow} && \exists \mathbf{x} = \mathbf{x}(t) \in [\mathbf{x}](t), v(\mathbf{p}, \mathbf{x}) \leq 0 \\ & \overset{(6)}{\Rightarrow} && \mathbf{p} \in \mathbb{V}^\exists(t) \end{aligned}$$

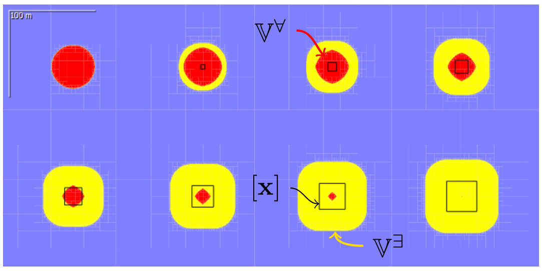


Fig. 3. $\mathbb{V}^\forall$ (in red) and $\mathbb{V}^\exists$ (union of yellow and red) for increasing position uncertainty (black squares).

Figure 3 shows the guaranteed and possible visible area of an omnidirectional sensor with $v(\mathbf{z}, \mathbf{x}) = \|\mathbf{z} - \mathbf{x}\|^2 - r^2$, where r is the maximum range of the sensor. $\mathbb{V}^\forall$ and $\mathbb{V}^\exists$ are plotted for state-uncertainty boxes of increasing width. It illustrates the erosion of the guaranteed visible area as state uncertainty increases, and simultaneously the expansion of the possible visible area with growing uncertainty. With large state uncertainty, the guaranteed visible area becomes empty (see lower-right of Fig. 3).

The notion of guaranteed and possible areas extends to the explored areas, by taking the union over time. The *guaranteed explored area* $\mathbb{M}^\forall$ is defined as the union of all the guaranteed visible areas

$$\mathbb{M}^\forall = \bigcup_{t \in [t_0, t_f]} \mathbb{V}^\forall(t), \tag{8}$$

and the *possible explored area* $\mathbb{M}^\exists$ is defined as the union of all the possible visible areas over time

$$\mathbb{M}^\exists = \bigcup_{t \in [t_0, t_f]} \mathbb{V}^\exists(t). \tag{9}$$

Thus, a bracketing of the actual explored area $\mathbb{M}$ is given by

$$\mathbb{M}^\forall \subset \mathbb{M} \subset \mathbb{M}^\exists. \tag{10}$$

4.2 Quantifier elimination

The $\forall$ and $\exists$ quantifiers appear in the expressions of the guaranteed and possible visible areas $\mathbb{V}^\forall(t)$ and $\mathbb{V}^\exists(t)$ given by equations (5) and (6). Solving set-inversion problems that involve quantifiers can be quite inefficient. To simplify the visible areas computations, quantifier-free expressions of $\mathbb{V}^\forall(t)$ and $\mathbb{V}^\exists(t)$ are derived in this sub-section.

We consider a sensor that sees inside a robot-centered disk of radius r. It can be described by the following visibility function:

$$v(\mathbf{z}, \mathbf{x}) = (z_1 - x_1)^2 + (z_2 - x_2)^2 - r^2. \tag{11}$$

Let us denote by $[v](\mathbf{z}, [\mathbf{x}])$ the natural inclusion function for v with respect to $\mathbf{x}$.

$$[v](\mathbf{z}, [\mathbf{x}]) = (z_1 - [x_1])^2 + (z_2 - [x_2])^2 - r^2. \tag{12}$$

Since v is continuous and the x_i's occur only once in the expression of v, we have (Moore [1966])

$$[v](\mathbf{z}, [\mathbf{x}]) = \{v(\mathbf{z}, \mathbf{x}), \mathbf{x} \in [\mathbf{x}]\}, \tag{13}$$

The guaranteed visible area condition $\forall \mathbf{x} \in [\mathbf{x}], v(\mathbf{z}, \mathbf{x}) \leq 0$ from formula (5) is equivalent to

$$[v](\mathbf{z}, [\mathbf{x}]) \subset [-\infty, 0]. \tag{14}$$

Inclusion (14) is thus true when the upper bound $\overline{v}$ of $[v](\mathbf{z}, [\mathbf{x}])$ is negative:

$$z \in \mathbb{V}^{\forall}(t) \Leftrightarrow \overline{v}(\mathbf{z}, [\mathbf{x}]) \leq 0. \tag{15}$$

Similarly, by using the natural inclusion function $[v]$, the possible visible area formula (6), $\exists \mathbf{x} \in [\mathbf{x}], v(\mathbf{z}, \mathbf{x}) \leq 0$ can be rewritten

$$[v](\mathbf{z}, [\mathbf{x}]) \cap [-\infty, 0] \neq \emptyset. \tag{16}$$

which can be also expressed as a negativity condition on the lower bound $\underline{v}$ of $[v]$:

$$z \in \mathbb{V}^{\exists}(t) \Leftrightarrow \underline{v}(\mathbf{z}, [\mathbf{x}]) \leq 0. \tag{17}$$

This way, quantifiers have been eliminated from the expressions $\mathbb{V}^{\forall}(t)$ and $\mathbb{V}^{\exists}(t)$. The visible areas are now defined by inequalities involving the bounds of $[v](\mathbf{z}, [\mathbf{x}])$. Expressions of the upper bound $\overline{v}$ and of the lower bound $\underline{v}$ can be derived by using symbolic interval arithmetic (Jaulin and Chabert [2010]). The lower bound is given by

$$\begin{aligned}\underline{v}(\mathbf{z}, [\mathbf{x}]) = {} & H\big((z_1-\overline{x_1})(z_1-\underline{x_1})\big) \min\big((z_1-\overline{x_1})^2, (z_1-\underline{x_1})^2\big) \\ & + H\big((z_2-\overline{x_2})(z_2-\underline{x_2})\big) \min\big((z_2-\overline{x_2})^2, (z_2-\underline{x_2})^2\big) - r^2\end{aligned} \tag{18}$$

where H denotes the Heaviside step function. The expression obtained for the lower bound is

$$\begin{aligned}\overline{v}(\mathbf{z}, [\mathbf{x}]) = {} & \max\big((z_1-\overline{x_1})^2, (z_1-\underline{x_1})^2\big) \\ & + \max\big((z_2-\overline{x_2})^2, (z_2-\underline{x_2})^2\big) - r^2\end{aligned} \tag{19}$$

4.3 Explored area mapping

It has been shown in Section 4.1 that the explored area $\mathbb{M}$ is bracketed between the guaranteed explored area $\mathbb{M}^{\forall}$ and the possible explored area $\mathbb{M}^{\exists}$ such that $\mathbb{M}^{\forall} \subset \mathbb{M} \subset \mathbb{M}^{\exists}$. Similarly, the visible area $\mathbb{V}(t)$ is bracketed between the guaranteed visible area $\mathbb{V}^{\forall}(t)$ and the possible visible area $\mathbb{V}^{\exists}(t)$ such that $\mathbb{V}^{\forall}(t) \subset \mathbb{V}(t) \subset \mathbb{V}^{\exists}(t)$.

Visible area Visible areas are arbitrary sets whose shape depends on sensor aperture, environment and robot state. Such arbitrary sets cannot be precisely described by a single box, but they can be approximated to any desired precision by subpavings.

At each time t, the SIVIA algorithm can be used to compute a bracketing of the guaranteed visible area between an inner subpaving $\underline{\mathbb{V}^{\forall}(t)}$ and an outer subpaving $\overline{\mathbb{V}^{\forall}(t)}$. Similarly, a guaranteed approximation of the possible visible area between the two subpavings $\underline{\mathbb{V}^{\exists}(t)}$ and $\overline{\mathbb{V}^{\exists}(t)}$ can be computed. We thus have

$$\underline{\mathbb{V}^{\forall}(t)} \subset \mathbb{V}^{\forall}(t) \subset \overline{\mathbb{V}^{\forall}(t)} \tag{20}$$

$$\underline{\mathbb{V}^{\exists}(t)} \subset \mathbb{V}^{\exists}(t) \subset \overline{\mathbb{V}^{\exists}(t)} \tag{21}$$

From (20) and (21), we obtain a bracketing of $\mathbb{V}(t)$ between the two subpavings $\underline{\mathbb{V}^{\forall}(t)}$ and $\overline{\mathbb{V}^{\exists}(t)}$ such that

$$\underline{\mathbb{V}^{\forall}(t)} \subset \mathbb{V}(t) \subset \overline{\mathbb{V}^{\exists}(t)}. \tag{22}$$

The visible area $\mathbb{V}(t)$ is thus known to be inside the set-interval $[\mathbb{V}(t)]$ made of a lower-bound $\underline{\mathbb{V}^{\forall}(t)}$ of the guaranteed visible area and an upper-bound $\overline{\mathbb{V}^{\exists}(t)}$ of the possible visible area. Figure (4) shows the fusion of the SIVIA computed set-intervals for $\mathbb{V}^{\forall}(t)$ and $\mathbb{V}^{\exists}(t)$ to obtain a bracketing of $\mathbb{V}(t)$.

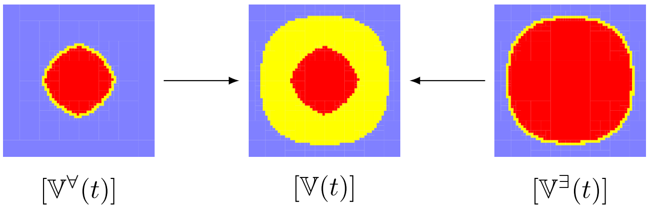


Fig. 4. The guaranteed visible area set-interval $[\mathbb{V}^{\forall}(t)]$ and the possible visible area set-interval $[\mathbb{V}^{\forall}(t)]$ are combined to obtain a bracketing of $\mathbb{V}(t)$.

Explored area As previously described in (8), the guaranteed explored area is the union of all guaranteed visible areas over the duration of the mission: $\mathbb{M}^{\forall} = \bigcup_{t \in [t]} \mathbb{V}^{\forall}(t)$. Let us define $\underline{\mathbb{M}^{\forall}} = \bigcup_{t \in [t]} \underline{\mathbb{V}^{\forall}(t)}$, the union of all the computed inner subpavings of $\mathbb{V}^{\forall}$. From the inclusion relation $\underline{\mathbb{V}^{\forall}(t)} \subset \mathbb{V}^{\forall}(t)$ of (20), by applying the union operation, we obtain $\underline{\mathbb{M}^{\forall}} \subset \mathbb{M}^{\forall}$. Similarly, let $\overline{\mathbb{M}^{\exists}} = \bigcup_{t \in [t]} \overline{\mathbb{V}^{\exists}(t)}$ be the union of the computed outer subpavings of the possible visible areas. Equations (9) and (21) lead to $\mathbb{M}^{\exists} \subset \overline{\mathbb{M}^{\exists}}$. Hence we have the inclusion relation

$$\underline{\mathbb{M}^{\forall}} \subset \mathbb{M}^{\forall} \subset \mathbb{M} \subset \mathbb{M}^{\exists} \subset \overline{\mathbb{M}^{\exists}}. \tag{23}$$

The computed solution $[\underline{\mathbb{M}^{\forall}}, \overline{\mathbb{M}^{\exists}}]$ thus form a set interval that contains the actual explored area $\mathbb{M}$.

5. APPLICATION

5.1 Underwater exploration simulation

To validate the proposed method, let us characterize the exploration of an area by an autonomous underwater vehicle (AUV). The mission of the AUV is to explore and fully cover a 500 m × 300 m rectangular target area. Exploration is done with a sensor whose range is a 40-meter radius disk.

An AUV simulator has been developed in C++ with Qt and OpenGL (Fig 5). The simulated AUV is equipped with a GPS receiver, an Inertial Measurement Unit (IMU) and a Doppler Velocity Log (DVL). The DVL provides the robot with speed measurements, either relative to water or to the seafloor.

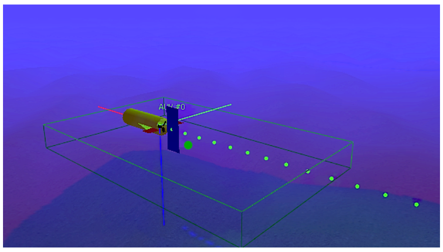


Fig. 5. 3D display of the AUV simulator. The green box is the real-time position uncertainty domain of the AUV.

A set-membership estimator-predictor provides real-time onboard positioning. This system computes a box in which the submarine is located. The center of the box is used as a point position estimate, and fed to the control system. The AUV is programmed to follow a "lawnmower pattern" that covers the target area. The spacing between the lanes in the reference trajectory is lower than the maximum spacing allowed by the sensor theoretical maximum range, in order to have some exploration overlap between successive lanes. This enables to cope with small navigation errors and also with environment effects on sensor performances.

All navigation data is saved and retrieved at the end of the mission for post-processing. Moreover, the true explored area is also computed by the simulator, which enables comparison with the computed solution sets.

Two test-cases have been simulated, both corresponding to a 30-minute mission:

- In the first test-case, the AUV starts from surface, where GPS is available, then dives to perform its exploration mission, and finally returns to surface when the mission is finished. Since GPS is not available underwater, there are only two position fixes: at the beginning and the end of the mission. When the AUV is underwater, positioning is only done by dead-reckoning.
- The second test-case is similar to the first, but an acoustic ranging system made of two GPS buoys has been deployed. Now, position fixes are also available underwater, which enables higher accuracy in positioning.

Figure 6 shows the trajectories actually followed by the AUV in the two test-cases, along with the reference "lawnmower pattern" trajectory. The actual explored area $\mathbb{M}$ is shown in green. Since the AUV is controlled from the center of its estimated position box, positioning errors induce errors in the followed trajectory. Without acoustic positioning during the mission, the AUV clearly drifts from the reference trajectory, which leads to uncovered zones in the bottom-left of the target area (Fig. 6a). In the second test-case, with acoustic positioning, the reference trajectory is more closely followed by the AUV, which provides a near perfect coverage of the target area (Fig. 6b).

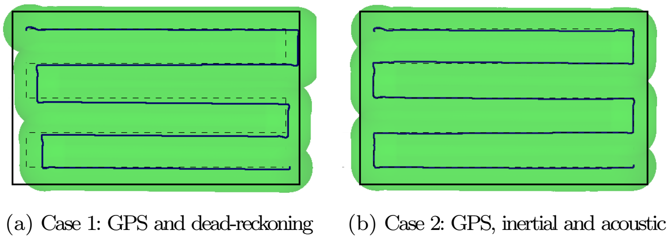
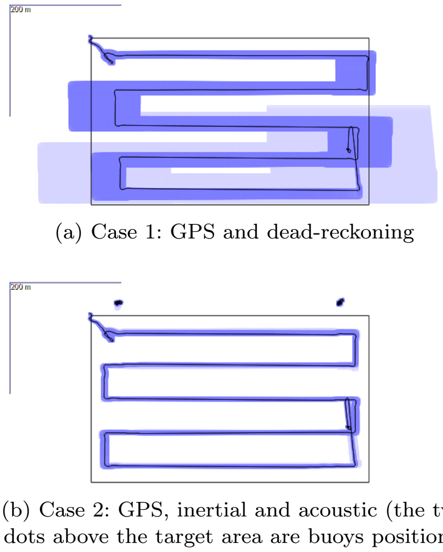

(a) Case 1: GPS and dead-reckoning (b) Case 2: GPS, inertial and acoustic

Fig. 6. Actual trajectory of the AUV (thick blue line), reference trajectory (dashed line) and actual explored area (light green). The rectangle is the target area.

5.2 Results

Figure 7 shows in light blue the real-time estimated position boxes of the AUV in the two test cases. The boxes are consistent with the actual robot position during the whole mission. In the first test case, position boxes width increases with time since the robot only relies on dead-reckoning for localization (Fig. 7a). Contrarily, positioning uncertainty is prevented from growing in test case 2 thanks to the use of an acoustic localization system (Fig. 7b).



(a) Case 1: GPS and dead-reckoning

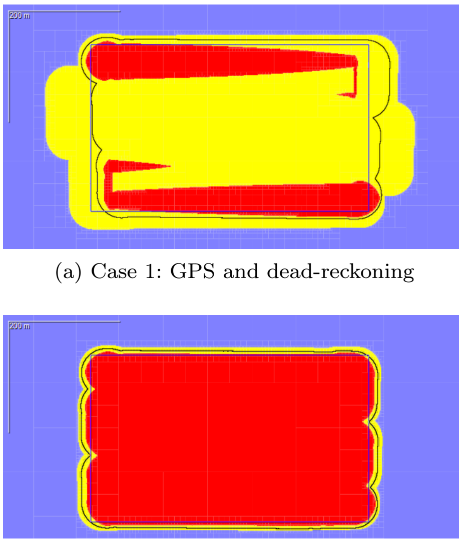
(b) Case 2: GPS, inertial and acoustic (the two dots above the target area are buoys positions)

Fig. 7. Real-time estimated position tubes (painted in light blue), post-processed position tubes (dark blue) and true trajectory (black line). The rectangle is the target area.

Before applying the explored area characterization algorithm, the AUV trajectory is post-processed with constraint propagation (Benhamou et al. [1999]). The AUV motion model is applied to contract the position uncertainty boxes in backward order, from the end of the mission to the start. This enables to retro-propagate the final GPS position fix to the whole trajectory. Moreover, in the second test-case, acoustic positioning corrections are also propagated back in time. The trajectory tubes obtained after post-processing are shown in dark blue in Figure 7. The second half of the first test case is greatly improved (Fig. 7a), the second test case is less improved since real-time positioning was already rather precise (Fig. 7b).

The guaranteed explored area $\mathbb{M}^{\forall}$ and the possible explored area $\mathbb{M}^{\exists}$ are computed from the post-processed trajectories of both test cases. Computation of the explored area for the 30-minute long simulated mission takes less than a minute on a standard computer. Since $\mathbb{M}^{\forall} \subset \mathbb{M}^{\exists}$, they can both be represented on the same figure. Figure 8 shows the guaranteed explored area in red; the possible explored area corresponds to the union of red and yellow. The outline of the actual explored area $\mathbb{M}$ is drawn in black. It can be seen of the figure that the computed $\mathbb{M}^{\forall}$ and $\mathbb{M}^{\exists}$ form a bracketing of $\mathbb{M}$, such that $\mathbb{M}^{\forall} \subset \mathbb{M} \subset \mathbb{M}^{\exists}$.

Comparison of Fig. 8a with Fig. 8b clearly shows the influence of positioning uncertainty (*i.e.* the width of the trajectory tube) on the possible and guaranteed explored areas. Areas of the computed guaranteed and possible explored areas in both test cases are reported in Table 1. In the second test case, thanks to the improved positioning accuracy, the guaranteed explored area $\mathbb{M}^{\forall}$ is more than three time larger than in the first test case. Moreover, the possible explored area is also tightened in the second test case. Indeed, while only 24.4% of the possible explored area has been guaranteed in the first test case, 88.9% of the computed $\mathbb{M}^{\exists}$ belongs to the guaranteed explored area in



(a) Case 1: GPS and dead-reckoning

(b) Case 2: GPS, inertial and acoustic

Fig. 8. Guaranteed explored area (red), Possible explored area (yellow and red) and actual explored area border (black).

the second test case. Reduction of positioning uncertainty has thus a great impact on explored area uncertainty.

Table 1.

	Test case 1	Test case 2
Area of $\mathbb{M}^{\forall}$	51,865 m^2	157,578 m^2
Area of $\mathbb{M}^{\exists}$	212,827 m^2	177,220 m^2
Part of $\mathbb{M}^{\exists}$ covered by $\mathbb{M}^{\forall}$	24.4%	88.9%
Part of target covered by $\mathbb{M}^{\exists}$	100%	100%
Part of target covered by $\mathbb{M}^{\forall}$	32.0%	99.7%

Two other interesting figures for mission assessment are presented in Table 1: the parts of the target area that have been possibly and guaranteedly explored. In both test cases, the whole target area belongs to the possible explored area. This means that, with the given positioning uncertainty, it is not possible to prove that a portion of the target area has not been covered. In the first test case, only 32% of the target area has been proven to have been explored. But in the second test case, 99.7% of the target is proven to have been explored. Moreover, contrarily to the first test case where the non-guaranteed explorer area is mostly located in the center of the mission zone, in the second test case, the 0.3% of the target area that have not been guaranteedly explored are located on the edge of the target rectangle.

With a sensor that provides images or than can detect landmarks, the remaining possible but not guaranteed explored area can be assessed by searching in the gathered data for overlap between two successive passes of the robot near the same area. Without analysis of the gathered data, the computed exploration map can also be used to plan a second mission to fill uncovered areas.

6. CONCLUSION

An interval based method has been presented to characterize the area explored by a robot. Due to localization uncertainty, the method computes a set interval that brackets the actual explored area between a guaranteed explored area and a possible explored area. As positioning uncertainty increases, the guaranteed explored area gets smaller and the possible explored area grows. Inversely, the computed explored area interval tightens as positioning uncertainty decreases.

Application on a simulated 30-minute underwater exploration mission shows the feasibility of the approach, yielding results in less than a minute. The computed set-interval of the explored area can be used after a search mission to ensure that space has been fully covered, to manually check possible but non-guaranteed explored areas, and to plan a complementary mission if necessary. Another use is for system design and mission planning, to ensure that positioning system accuracy and programmed navigation trajectory enable full exploration of a target area.

Future work will be focused on application to cooperative exploration with multiple robots and on improvement of the guaranteed explored area by the use of full trajectory information and additional constraints.

REFERENCES

F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget. Revising hull and box consistency. In *Int. Conf. on Logic Programming*, pages 230–244. MIT press, 1999.

P. Bouron, D. Meizel, and Ph. Bonnifait. Set-membership non-linear observers with application to vehicle localisation. In *6th European Control Conference*, pages 1255–1260, Porto, 2001.

A. Caiti, A. Garulli, F. Livide, and D. Prattichizzo. Set-membership acoustic tracking of autonomous underwater vehicles. *Acta Acustica united with Acustica*, 88(5):648–652, 2002.

E. Colle and S. Galerne. Mobile robot localization by multiangulation using set inversion. *Robotics and Autonomous Systems*, 61(1):39–48, January 2013.

V. Drevelle and P. Bonnifait. Interval-based fault detection and identification applied to global positioning. In *IFAC SYSID 2012*, volume 16, pages 1085–1090, Brussels, 2012.

L. Jaulin. Range-only SLAM with occupancy maps: A set-membership approach. *IEEE Transactions on Robotics*, 27(5):1004–1010, 2011.

L. Jaulin and G. Chabert. Resolution of nonlinear interval problems using symbolic interval arithmetic. *Eng. Appl. of Artif. Intel.*, 23(6):1035–1049, 2010.

L. Jaulin, M. Kieffer, O. Didrit, and É. Walter. *Applied Interval Analysis*. Springer-Verlag, 2001.

F. Le Bars, J. Sliwka, L. Jaulin, and O. Reynet. Set-membership state estimation with fleeting data. *Automatica*, 48(2):page 381–387, February 2012.

R.E. Moore. *Interval analysis*. Prentice Hall, 1966.

T. Raıssi, N. Ramdani, and Y. Candau. Set membership state and parameter estimation for systems described by nonlinear differential equations. *Automatica*, 40(10): 1771–1777, 2004.

A. Rauh, J. Minisini, E.P. Hofer, and H. Aschemann. Robust and optimal control of uncertain dynamical systems with state-dependent switchings using interval arithmetic. In *SCAN'2008*, page 116, El Paso, 2008.

B. Telle, M.-J. Aldon, and N. Ramdani. Camera calibration and 3D reconstruction using interval analysis. In *12th Int. Conf. on Image Analysis and Processing*, pages 374–379. IEEE, 2003.