Set-membership state estimation by solving data association

Simon Rohou[1], Benoît Desrochers[2], Luc Jaulin[1]

Abstract— **This paper deals with the localization problem of a robot in an environment made of indistinguishable landmarks, and assuming the initial position of the vehicle is unknown. This scenario is typically encountered in underwater applications for which landmarks such as rocks all look alike. Furthermore, the position of the robot may be lost during a diving phase, which obliges us to consider unknown initial position. We propose a deterministic approach to solve simultaneously the problems of *data association* and *state estimation*, without combinatorial explosion. The efficiency of the method is shown on an actual experiment involving an underwater robot and sonar data.**

I. INTRODUCTION

This paper deals with the localization of a mobile robot [11], [3] for which a map of the environment is available. It is assumed that

- the map is static and made of point landmarks;
- the landmarks are indistinguishable;
- the position of each landmark is known;
- the initial pose of the robot is not known.

This problem, already considered in the literature, often occurs when using sonars for localization [8], [21], [24], where it is difficult to distinguish a landmark from another [14]. The challenge in this type of scenario is the difficulty of *data associations*, *i.e.* to establish feature correspondences between perceptions of landmarks and the map of these landmarks, known beforehand. This is especially the case of challenging underwater missions [38], [27], [37], for which one can reasonably assume that the detected landmarks cannot be distinguished from the other: for instance, two different rocks may have the same aspect and dimension in sonar images, as illustrated by Figure I. In addition, the landmark detection process is sensitive to the point of view of the sensor and some landmarks of the map could not even be detected during a survey. Therefore, no reliable data associations based on landmarks shapes can be assumed. Lastly, we could benefit from a sonar tracking system to improve the localization when perceiving some unidentified landmark, but we consider this matching as non reliable and it will not be used for the localization.

Due to these difficulties, we consider that we cannot compute reliable data associations that stand on image processing of the observations. We will rather consider the data association problem together with state estimation. Indeed, any proprioceptive information of the vehicle, such as the distance travelled between two observations, may help to associate perceptions with nodes of the map.

[1]ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France
[2]DGA TN, Brest, France

Fig. 1. Different underwater rocks perceived with one side-scan sonar. These observations are view-point dependent and the sonar images are noisy, which makes it difficult to distinguish one rock from another only from image processing. The sonar images of this paper have been collected by the *Daurade* robot (DGA-TN, Shom, Brest) equipped with a Klein 5000 lateral sonar.

This approach has already been studied with methods combining a probabilistic filter with a branch-and-bound algorithm to match the landmarks between two different views [35], [20]. However, due to the branching operations, these methods involve a lot of computational resources. On top of that, they require knowledge about the initial position of the robot, that is often not available.

Our contribution is to cast the localization problem into a constraint network for which constraint propagation methods can be used [32]. The branchings are replaced by contraction operators (or *contractors*) which makes the resolution much more efficient by avoiding the combinatorial explosion. In practice, the contractors use interval computations to eliminate unfeasible parts of the search space. Interval analysis [17], [18], [25] has been used for more than thirty years in control [28], [33] or state estimation. It appears to be well suited to handle uncertainties such as measurement errors or unknown positions, without requiring linearizations, independence between variables or Gaussian distributions.

II. FORMALISM

The map is known beforehand and the robot obeys to state equations. The localization problem corresponds to the following state estimation:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) & \text{(evolution equation)} \\ \mathbf{y}^i = \mathbf{g}\big(\mathbf{x}(t_i)\big) & \text{(observation equation)} \end{cases} \quad (1)$$

where $\mathbf{x}$ is the unknown state vector, $\mathbf{u}$ is the input measurement vector and $\mathbf{y}^i$ is an output measurement vector made at time t_i. As $\mathbf{x}$ and $\mathbf{u}$ continuously evolve with time, we define them as trajectories denoted by $\mathbf{x}(\cdot)$ and $\mathbf{u}(\cdot)$, contrary to $\mathbf{y}^i$ that represents discrete data. In this paper, we will assume that all errors are bounded [36]. This assumption has often been made in the context of localization [6], [26], [23], [7], [10]. Therefore, it is assumed that $\mathbf{u}(t) \in [\mathbf{u}](t) \ \forall t$ and that for each measurement, $\mathbf{y}^i \in [\mathbf{y}^i]$.

Often, the observation equation also appears in the following implicit form:

$$\mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}^i\big) = \mathbf{0}. \tag{2}$$

Example. Assume that we have a robot located at $(x_1, x_2)^\mathsf{T}$ with a heading x_3. It perceives a landmark $\mathbf{m}$ which position is $(m_1, m_2)^\mathsf{T}$. The corresponding measurement vector $\mathbf{y}$ is made of the range y_1 and the bearing y_2, as illustrated by Figure 2.
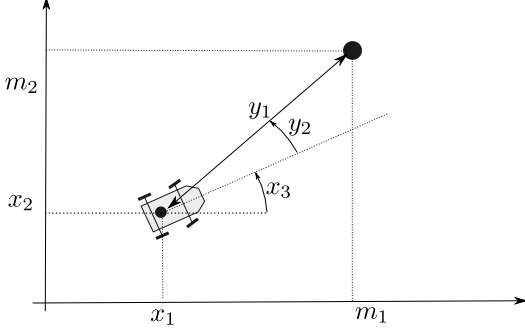

Fig. 2. The landmark $\mathbf{m} = (m_1, m_2)^\mathsf{T}$ is perceived by the robot at a distance y_1 and a bearing y_2. The pose of the robot is $(x_1, x_2, x_3)^\mathsf{T}$.

In this example, the observation equation is defined as:

$$\mathbf{g}(\mathbf{x}, \mathbf{y}) = \left(\begin{array}{c} x_1 + y_1 \cdot \cos(x_3 + y_2) - m_1 \\ x_2 + y_1 \cdot \sin(x_3 + y_2) - m_2 \end{array} \right).$$

Note that in a *range-only* case where the robot only measures the distance y to the landmark, we obtain the following relation:

$$g(\mathbf{x}, y) = (x_1 - m_1)^2 + (x_2 - m_2)^2 - y^2.$$

In the more general case, the observation function $\mathbf{g}$ is uncertain, or more precisely, it depends on some parameters that are not exactly known. The related observation equation can be written as:

$$\left\{ \begin{array}{l} \mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}^i, \mathbf{m}\big) = \mathbf{0}, \\ \mathbf{m} \in [\mathbf{m}]. \end{array} \right. \tag{3}$$

In a set-membership approach the parameter vector $\mathbf{m}$, expressing the uncertain position of the landmark, is known to be bounded within an interval-vector (a box): $[\mathbf{m}]$. We emphasize that uncertainties on $\mathbf{x}(t_i)$ and $\mathbf{y}^i$ are implicitly bounded by $[\mathbf{x}](t_i)$ and $[\mathbf{y}^i]$.

Data association. When several landmarks $\mathbf{m}_1, \ldots, \mathbf{m}_\ell$ exist, the observation data may not be *associated*: we do not know to which landmark the measurement $\mathbf{y}^i$ refers. Hence the observation constraint has the form:

$$\left\{ \begin{array}{l} \mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}^i, \mathbf{m}^i\big) = \mathbf{0}, \\ \big(\mathbf{m}^i \in [\mathbf{m}_1]\big) \vee \cdots \vee \big(\mathbf{m}^i \in [\mathbf{m}_\ell]\big), \end{array} \right. \tag{4}$$

with $\mathbf{m}^i$ the unidentified landmark perceived at time t_i. Equivalently, the system can be expressed as:

$$\left\{ \begin{array}{ll} \mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}^i, \mathbf{m}^i\big) = \mathbf{0}, & \text{(5a)} \\ \mathbf{m}^i \in \mathbb{M} = \{[\mathbf{m}_1], \ldots, [\mathbf{m}_\ell]\}, & \text{(5b)} \end{array} \right.$$

where $\mathbb{M}$ is the bounded map of the environment: the set of all landmarks represented by their bounded positions. In other words, we do not know the right parameter vector $\mathbf{m}_1, \ldots, \mathbf{m}_\ell$ associated with the observation function. Equation (5b) is a discrete constraint, since $\mathbb{M}$ is made of a finite number of isolated points, and corresponds to a problem of data association [35]. Figure 3 illustrates the difficulty to differentiate landmarks of underwater extents.
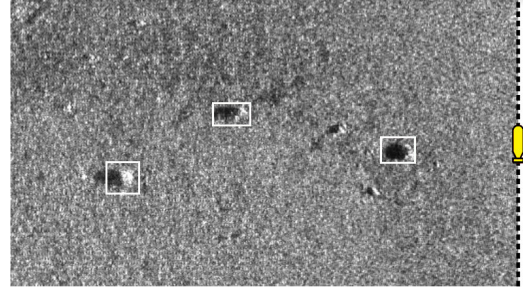

Fig. 3. The yellow robot, equipped with a side-scan sonar, perceives at port side some rocks $\mathbf{m}^i$ lying on the seabed. The rocks, that can be used as landmarks, are assumed to belong to small georeferenced boxes $[\mathbf{m}^i]$ enclosing uncertainties on their positions. The robot is currently not able to make any difference between the rocks, as it is typically the case in underwater extents when acoustic sensors are used to detect features [9].

In this paper, the data association problem is solved together with state estimation, without image processing. Indeed, any proprioceptive information such as inputs $[\mathbf{u}](t)$ may help to associate observations with nodes of the map. Valid associations are then useful to obtain accurate state estimation, which could allow further associations. This chicken-and-egg problem is formalized by the following equations:

$$\left\{ \begin{array}{ll} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) & \text{(evolution eq.)} \\ \mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}^i, \mathbf{m}^i\big) = \mathbf{0} & \text{(observation eq.)} \\ \mathbf{m}^i \in \mathbb{M} & \text{(data association)} \end{array} \right. \tag{6}$$

In order to solve the problem efficiently and in a simple manner, we propose an interval-based method coupled with constraint-programming [19], [6], [15], [16].

III. CONSTRAINT NETWORK

The problem will be described as a set of constraints depicting Equations (6). A solver based on these relations will then allow the resolution of both state estimation and data association.

Evolution equation. Without loss of generality, we will consider a robot moving on a plane among landmarks. The motion is described by the following state equation:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \\ &= \left(\begin{array}{c} \left(\begin{array}{cc} \cos(x_3(t)) & -\sin(x_3(t)) \\ \sin(x_3(t)) & \cos(x_3(t)) \end{array} \right) \cdot \mathbf{v}(t) \\ \omega(t) \end{array} \right). \end{aligned} \tag{7}$$

The state vector $\mathbf{x} \in \mathbb{R}^3$ represents the position of the robot and its heading. The input vector is $\mathbf{u} = (v_u, v_v, \omega)^\mathsf{T}$, where $\mathbf{v} = (v_u, v_v)^\mathsf{T}$ is the horizontal speed of the vehicle in its own reference frame and ω is its yaw-rate. In the underwater

case, $\mathbf{v}$ can be measured with a Doppler Velocity Log (DVL) sensor. In addition, gyroscopes are usually used to obtain the angular velocity ω.

Observation equation. For some times t_i, the robot collects the range-bearing vector $\mathbf{y}^i$ to a nearby landmark $\mathbf{m}^i$ that belongs to the map $\mathbb{M}$ made of a collection of georeferenced points. Hence, the observation function $\mathbf{g}$ is:

$$\mathbf{g}\big(\mathbf{x}(t_i), \mathbf{y}^i, \mathbf{m}^i\big) = \begin{pmatrix} x_1(t_i) \\ x_2(t_i) \end{pmatrix} + y_1^i \cdot \begin{pmatrix} \cos\big(x_3(t_i) + y_2^i\big) \\ \sin\big(x_3(t_i) + y_2^i\big) \end{pmatrix} - \begin{pmatrix} m_1^i \\ m_2^i \end{pmatrix}. \quad (8)$$

Decomposition. These equations can be broken down into the following set of elementary constraints:

$$\begin{cases} (i) & \mathbf{v}(\cdot) = \mathbf{f}\big(\mathbf{x}(\cdot), \mathbf{u}(\cdot)\big) \\ (ii) & \dot{\mathbf{x}}(\cdot) = \mathbf{v}(\cdot) \\ (iii) & \mathbf{p}^i = \mathbf{x}(t_i) \\ (iv) & \mathbf{d}^i = \mathbf{m}^i - \mathbf{p}_{1,2}^i \\ (v) & a^i = p_3^i + y_2^i \\ (vi) & \mathbf{d}^i = y_1^i \cdot \begin{pmatrix} \cos(a^i) \\ \sin(a^i) \end{pmatrix} \\ (vii) & \mathbf{m}^i \in \mathbb{M} \end{cases} \quad (9)$$

These constraints involve intermediate variables introduced for ease of decomposition: $a^i \in \mathbb{R}$, vectors $\mathbf{d}^i \in \mathbb{R}^2$, $\mathbf{p}^i \in \mathbb{R}^3$, and a trajectory $\mathbf{v}(\cdot) \in \mathbb{R} \to \mathbb{R}^3$. The notation $\mathbf{p}_{1,2}$ represents the subvector $(p_1, p_2)^\intercal$ of $\mathbf{p} \in \mathbb{R}^3$. This procedure of decomposition aims at revealing elementary constraints that have been already studied by the community in previous works. In the *contractor programming* framework [29], constraints are applied on sets by means of contractors: operators reducing interval domains according to the constraints without losing any solution. A contractor is denoted by $\mathcal{C}$.

Domains (sets). Domains have to be defined for each variable of Equations (9). They will be interval $[a^i]$, boxes $[\mathbf{p}^i]$, $[\mathbf{m}^i]$, $[\mathbf{d}^i]$, $[\mathbf{y}^i]$ and tubes as intervals of trajectories: $[\mathbf{x}](\cdot)$, $[\mathbf{v}](\cdot)$, $[\mathbf{u}](\cdot)$. Some domains are initialized according to measurements and their bounded uncertainties, *e.g.* $\mathbf{y}^i \in [\mathbf{y}^i]$. Other domains are initialized as sets of all real values, *e.g.* $[\mathbf{d}^i] = [-\infty, \infty]^2$.

The resolution of our problem consists in applying the necessary contractors on these domains in an iterative way. The resolution process stops when the contractors are no more efficient.

Catalog of contractors. Papers and software libraries already provide solutions for elementary contractors. The ones required for Equations (9) are listed hereafter.

- (iv), (v): difference and addition relations are respectively handled by simplest contractors $\mathcal{C}_-$ and $\mathcal{C}_+$. For purposes of understanding, we provide below the definition of $\mathcal{C}_+$ associated with the constraint $(a = x + y)$:

$$\begin{pmatrix} [a] \\ [x] \\ [y] \end{pmatrix} \xmapsto{\mathcal{C}_+} \begin{pmatrix} [a] \cap ([x] + [y]) \\ [x] \cap ([a] - [y]) \\ [y] \cap ([a] - [x]) \end{pmatrix}. \quad (10)$$

In our case, we would apply $\mathcal{C}\big([a^i], [p_3^i], [y_2^i]\big)$ in order to deal with relation (v), thus contracting the three domains of interest.

- (i): expressions that are more complex than addition, such as function $\mathbf{f}$ of Equation (7), can easily be dealt with by combining contractors. Algorithms such as those described in [2], [1] and software libraries like IBEX [4] are at hand to automatically combine these operators. For our problem, we will stand on this library to provide us with a complex contractor denoted $\mathcal{C}_\mathbf{f}$.
- (vi): the *polar equation* connects Cartesian coordinates to polar ones. The minimal contractor $\mathcal{C}_{\text{polar}}$, introduced in [5], can be used to contract the related domains: $\mathcal{C}_{\text{polar}}\big([d_1^i], [d_2^i], [y_1^i], [a^i]\big)$.
- (iii): $\mathbf{p}^i = \mathbf{x}(t_i)$ is a constraint that links the vector $\mathbf{p}^i$ to the evaluation of the trajectory $\mathbf{x}(\cdot)$ at time t_i. A dedicated contractor $\mathcal{C}_{\text{eval}}$ has been provided in [31]. It will allow the correction of the positions of the robot.
- (ii): $\dot{\mathbf{x}}(\cdot) = \mathbf{v}(\cdot)$ is the simplest differential constraint that binds a trajectory $\mathbf{x}(\cdot)$ to its derivative $\mathbf{v}(\cdot)$. The related contractor $\mathcal{C}_{\frac{d}{dt}}$ has been introduced in [30] and allows contractions on the tube $[\mathbf{x}](\cdot)$ to preserve only trajectories consistent with the derivatives enclosed in the tube $[\mathbf{v}](\cdot)$.

The last constraint to consider is (vii): $\mathbf{m}^i \in \mathbb{M}$, which corresponds to the data association part of our problem. To our knowledge, a contractor to deal with this relation does not exist yet in the literature. In order to expand the catalog of contractors, we provide in the following section material for the so-called *constellation contractor*.

IV. CONSTELLATION CONSTRAINT

This section proposes a new operator named *constellation contractor* denoted by $\mathcal{C}_{\text{constell}}$, that will allow us to solve the data association problem. Let us consider a constellation of ℓ points $\mathbb{M} = \{[\mathbf{m}_1], \dots, [\mathbf{m}_\ell]\}$ of $\mathbb{IR}^d$ and a box $[\mathbf{x}] \in \mathbb{IR}^d$. We want to compute the smallest box $\mathcal{C}_{\text{constell}}([\mathbf{x}])$ containing $\mathbb{M} \cap [\mathbf{x}]$, or equivalently:

$$\mathcal{C}_{\text{constell}}([\mathbf{x}]) = \bigsqcup_j \big([\mathbf{x}] \cap [\mathbf{m}_j]\big), \quad (11)$$

where $\bigsqcup$, called *squared union*, returns the smallest box enclosing the union of its arguments. Figure 4 illustrates this operator[1].

Since we will have many different boxes $[\mathbf{x}]$ to be processed based on a static constellation, a preprocessing step can be done to speed up the computations. This procedure is interesting for sets $\mathbb{M}$ made of a huge number of landmarks, as it can be the case in various applications: a preprocessing step will allow a logarithmic complexity of the contractor with respect to the number of points ℓ.

Preprocessing step. We build a balanced binary tree corresponding to an R-tree [12]. The R-tree has the following properties:

---

[1]Note that the map $\mathbb{M}$ could be represented by a subpaving or an image, for which a contractor such as the one used in [34] could be applied, but this would require a more complex data structure unnecessarily cumbersome.
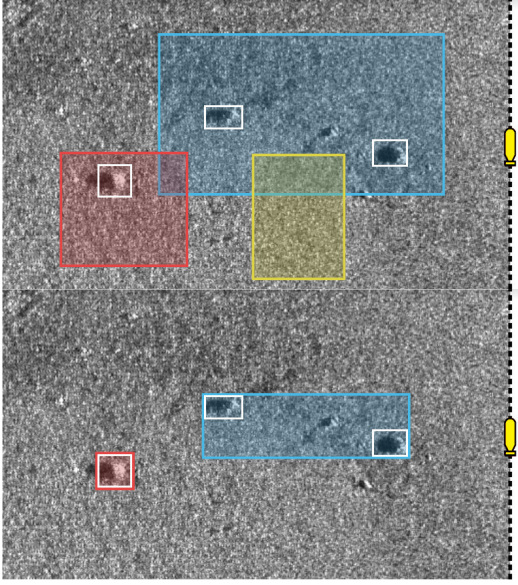
Fig. 4. Illustration of the constellation contractor, before and after the contraction step. This operator is reliable as it does not remove any significant rock. In this example, the red perception leads to a reliable association since the box contains only one item of $\mathbb{M}$.

Fig. 5. An R-tree is created beforehand in order to get an efficient constellation contractor.

Algorithm 1 $\mathcal{C}_{\text{constell}}\left([\mathbf{x}],\beta\right)$

---

1   if $\Box\beta \subset [\mathbf{x}]$, return $\Box\beta$
2   if $\Box\beta \cap [\mathbf{x}] = \varnothing$, return $\varnothing$
3   return $\mathcal{C}_{\text{constell}}\big([\mathbf{x}], \text{left\_child}(\beta)\big)$
       $\sqcup\ \mathcal{C}_{\text{constell}}\big([\mathbf{x}], \text{right\_child}(\beta)\big)$

- (i) To each node β of the tree is associated a box $\Box\beta$ containing at least one point of the constellation $\mathbb{M}$ and such that $\mathcal{C}(\Box\beta) = \Box\beta$. Notation $\Box\beta$ represents the minimal enveloping box of all points of $\mathbb{M}$ represented by β.
- (ii) If β_0 is the root, then $\mathbb{M} \subset \Box\beta_0$.
- (iii) If β_1, β_2 are children of β then $(\Box\beta_1 \cap \mathbb{M}) \cup (\Box\beta_2 \cap \mathbb{M}) = (\Box\beta \cap \mathbb{M})$.
- (iv) If β_1, β_2 are brothers then $\Box\beta_1 \cap \Box\beta_2 = \varnothing$. When this property is satisfied, the R-tree is called a R*-tree. This property is not restrictive in our context since we assume that the $\mathbf{m}_j$ are completely known beforehand.
- (v) The tree is balanced with respect to the space. This means that the bisection direction for the branching of the node β is decided with respect to the largest width of $\Box\beta$.
- (vi) The tree is balanced with respect to the constellation. It means that bisection position of β is decided with respect to the median in order to minimize the difference $|\text{card}(\Box\beta_1 \cap \mathbb{M}) - \text{card}(\Box\beta_2 \cap \mathbb{M})|$.

These properties are illustrated by Figure 5 in the case of a constellation of $\ell = 10$ points represented by blue tiny boxes. The root β_0 of the tree has two children β_1 and β_2. Note that $|\text{card}(\Box\beta_1 \cap \mathbb{M}) - \text{card}(\Box\beta_2 \cap \mathbb{M})| = 0$, which corresponds to the minimum that can be obtained. The corresponding bisection results from the median. The node β_2 has two children β_3 and β_4. The node β_5 is a leaf of the tree and $\Box\beta_5$ corresponds to a single point of the constellation $\mathbb{M}$.

Once constructed, the R-tree allows a logarithmic complexity with respect to ℓ. The corresponding constellation contractor $\mathcal{C}_{\text{constell}}$ is provided in Algorithm 1 and illustrated by Figure 6.
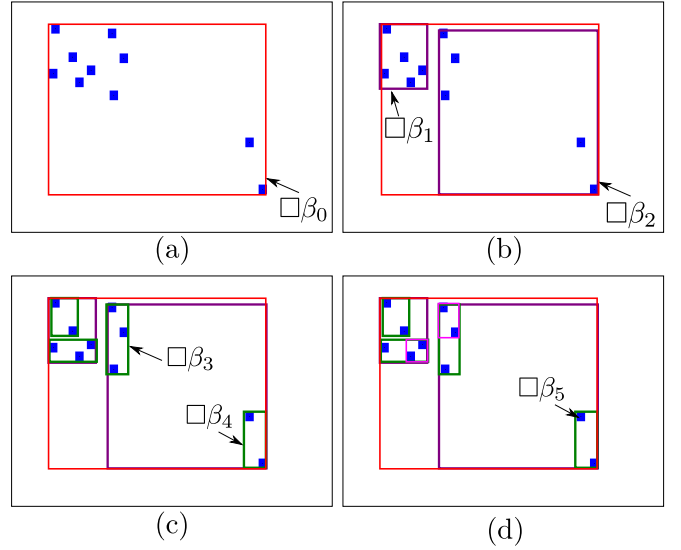
Example (Fig. 6). First, since $\Box\beta_0 \not\subset [\mathbf{x}]$, we explore the two children of β_0. Since $\Box\beta_1 \subset [\mathbf{x}]$ and $\Box\beta_2 \not\subset [\mathbf{x}]$, only the branch corresponding to β_2 is explored. Again, since $\Box\beta_3 \subset [\mathbf{x}]$ and $\Box\beta_4 \not\subset [\mathbf{x}]$, we only browse β_4.

Finally, the exploration of the tree corresponds to the following algebraic calculus:

$$
\begin{aligned}
\mathcal{C}_{\text{constell}}\left([\mathbf{x}],\beta\right) &= [\mathbb{M} \cap \Box\beta_0 \cap [\mathbf{x}]] \\
&= \Box\beta_1 \sqcup [\mathbb{M} \cap \Box\beta_2 \cap [\mathbf{x}]] \\
&= \Box\beta_1 \sqcup \Box\beta_3 \sqcup [\mathbb{M} \cap \Box\beta_4 \cap [\mathbf{x}]] \\
&= \Box\beta_1 \sqcup \Box\beta_3 \sqcup [\mathbb{M} \cap \Box\beta_5 \cap [\mathbf{x}]] \\
&= \Box\beta_1 \sqcup \Box\beta_3 \sqcup \Box\beta_5.
\end{aligned}
$$

V. APPLICATION

A. Solver

All the necessary contractors are at hand to build a solver for the problem. The domains of the variables are predefined as explained in Section III and the following contracting operations are applied in any order up to a fixed point:

$$
\begin{cases}
(i) & \mathcal{C}_{\mathbf{f}}\left([\mathbf{v}](\cdot),[\mathbf{x}](\cdot),[\mathbf{u}](\cdot)\right) \\
(ii) & \mathcal{C}_{\frac{d}{dt}}\left([\mathbf{x}](\cdot),[\mathbf{v}](\cdot)\right) \\
(iii) & \mathcal{C}_{\text{eval}}\left([t_i],[\mathbf{p}^i],[\mathbf{x}](\cdot)\right) \\
(iv) & \mathcal{C}_{-}\left([\mathbf{d}^i],[\mathbf{m}^i],[\mathbf{p}_{1,2}^i]\right) \\
(v) & \mathcal{C}_{+}\left([a^i],[p_3^i],[y_2^i]\right) \\
(vi) & \mathcal{C}_{\text{polar}}\left([d_1^i],[d_2^i],[y_1^i],[a^i]\right) \\
(vii) & \mathcal{C}_{\text{constell}}\left([\mathbf{m}^i]\right)
\end{cases}
\tag{12}
$$
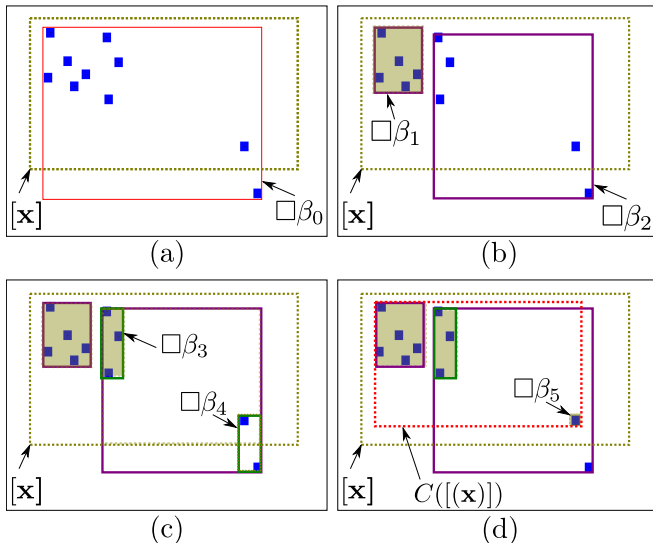
Fig. 6. Illustration of $\mathcal{C}_{\text{constell}}$ based on an R-tree.

Fig. 7. *Daurade* AUV managed by DGA Techniques Navales (Brest) and the Service Hydrographique et Océanographique de la Marine (Shom), during an experiment dedicated to this work, in the Rade de Brest, 2015.

The following test-case aims at providing a practical illustration of how the constraint propagation method can be used to:

1) estimate a set of feasible trajectories of the vehicle;
2) solve the data association problem without any combinatorial explosion.

Once these two issues are solved with contractors, any classical localization method, such as an EKF [13], can be added to obtain a more accurate estimate of the trajectory. The filter can be initialized with a starting point in the reliable set $[\mathbf{x}_{1,2}](t_0)$ and updated only with observations that are correctly associated.

B. Test-case

Our solver is illustrated on an actual dataset involving the Autonomous Underwater Vehicle (AUV) *Daurade*, see Figure 7. The robot evolved underwater during 45 minutes, 5 meters above the seabed. *Daurade* started its mission with a huge position uncertainty. This can happen during a dive in deep waters [22] or when, for discretion or security purposes, a long-range underwater transit phase is required to reach the working area. Therefore, we consider that $[\mathbf{x}_{1,2}](t_0) = [-\infty, \infty]^2$. We also assume that a part of the mission area has already been mapped during a previous survey. The corresponding map $\mathbb{M}$ describing this area is modeled as a set of 133 landmark points that are well georeferenced.

The robot performed some *boustrophedon* pattern, typically used for survey missions, as depicted in Figure 8. It sensed its environment using a side-scan sonar that gives lateral images of the seabed, with a scope of 75m on each side. During the mission, 54 objects have been extracted from the sonar images with the help of algorithms. Unfortunately, other objects are ignored and will not be used for the localization.

Only the 2-dimensional positions $\mathbf{x}_{1,2}(\cdot)$ of the robot need to be estimated since other state variables (roll, pitch, altitude

and depth) are directly measured by accurate sensors. The heading x_3 is assumed to be known with an accuracy of ± 5 degrees.
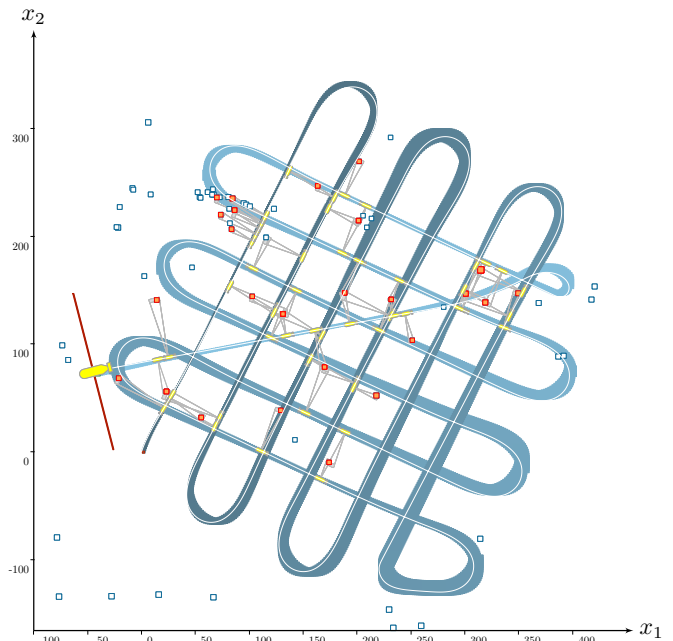

Fig. 8. Map of the mission with the results of the presented approach. The *actual* trajectory of the AUV is depicted by a white line, enclosed in the horizontal projection of the tube $[\mathbf{x}](\cdot)$, depicted in blue. The map is composed of 133 objects represented by boxes. Landmarks that have been perceived and recognized by the robot without ambiguity are depicted in red. The observations obtained from the sonar images are drawn with gray pies depicting both range and bearing uncertainties. The related measurement times t_i are represented by small robots along the white trajectory. The range of the side-scan sonar is represented by the red line on the last position.

C. Results

Table I shows bounds used to quantify errors on sensor readings. Figure 8 shows the final trajectory obtained after 9 iterations of the localization algorithm. The whole estimation is performed in less than 2.5 seconds on *i7-4700HQ CPU@2.40GHz*, up to a fixed point where no more contractions are obtained. At the end, 51 of the 54 objects
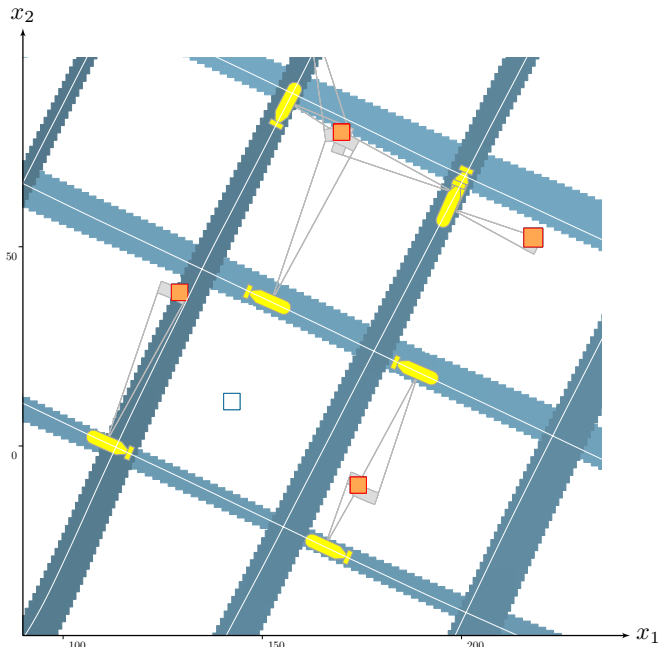
Fig. 9. Zoom on the mission map. One rock, depicted by the blue box, has never been seen by the robot. The four other landmarks have been perceived and identified.

Fig. 10. $d = w([\mathbf{x}_{1,2}])$ when reaching a contracting fixed point. The value d represents the diameter of each box $[x_1](t) \times [x_2](t)$, that is to say the localization error in the very worst case. The gray vertical lines are the 54 observation instants t_i.

TABLE II

NUMERICAL RESULTS OF THE ITERATIVE LOCALIZATION ALGORITHM.

#	time(s)	#min	#max	#ok
1	0.278	133	133	0
2	0.271	14	64	0
3	0.268	5	52	0
4	0.266	1	34	2
5	0.271	1	16	39
6	0.267	1	4	48
7	0.266	1	3	49
8	0.266	1	3	50
9	0.266	1	2	51

perceived in the sonar images have been correctly associated, which means that the remaining three boxes $[\mathbf{m}^i]$ contain more than one item of $\mathbb{M}$. The white trajectory depicted in Figures 8 and 9, filtered with an external USBL sensor that has not been used by the localization algorithm, is generally considered as the *truth* in our underwater experiments. However, USBL stands on acoustical measurements coming with strong errors. Hence, we can only consider it as a good support for verifying the quality of our set-membership estimation.

TABLE I

UNCERTAINTIES ON DATA USED FOR THE APPLICATION.

data	description	uncertainty	unit
$x_3(t)$	heading	$[-5, 5]$	deg
$\mathbf{v}(t)$	linear speed	$[-0.015, 0.015]^2$	m.s^{-1}
$\omega(t)$	angular velocity	$[-0.001, 0.001]$	deg.s^{-1}
$y_1(t)$	measur. range	$[-1.5, 1.5]$	m
$y_2(t)$	measur. bearing	$[-0.1, 0.1]$	deg

In Table II, the duration of each contraction step is given and appears to be constant. During the constraint propagation process, the thinner the trajectory, the smaller the number of landmarks contained in $[\mathbf{m}^i]$, and *vice versa*. As an indicator, column 3 (resp. 4) of Table II shows the minimal (resp. maximal) number of landmarks included in the $[\mathbf{m}^i]$ amongst all measurements. The last column corresponds to the number of correct associations, *i.e.* when $[\mathbf{m}^i]$ contains a single point of $\mathbb{M}$. Figure 10 shows the diameter of the tube $[\mathbf{x}_{1,2}](\cdot)$ with respect to time. We can note on this figure that the initial position of the robot is not known before the contractions, and is finally estimated with an error of 3.6m in the worst case.
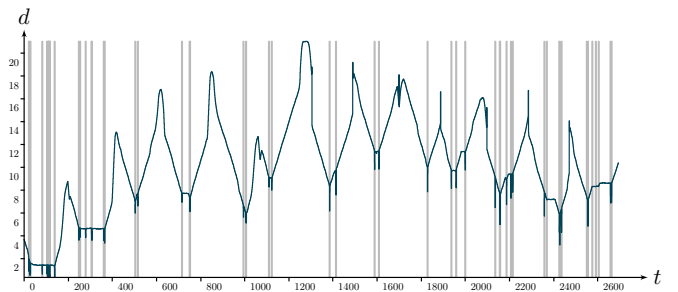
VI. CONCLUSION

This contraction method, enriched with the constellation contractor introduced in this paper, is shown to be powerful in situations involving a huge number of possible data associations. In comparison, other existing methods often run into difficulties when both the initial position and the data associations are unknown. Furthermore, the results of the algorithm are sets that are guaranteed to contain the actual but unknown values of our problem. This is particularly important in order to provide reliability for autonomous systems.

With the computation times in the application part of this work, we demonstrate that the proposed algorithm can also be considered for online localization. In addition, this approach provides a reliable way to gather different views of a same object. This is useful to build datasets, for mine hunting purposes among other things.

AVAILABLE LIBRARIES

The *Tubex* library implemented during this work and a video of the application are available on www.simon-rohou.fr/research/datasso/. Simple but controlled simulation experiments are also available for the reader wishing to appreciate the performances of the proposed method.

REFERENCES

[1] I. Araya, G. Trombettoni, and B. Neveu. A Contractor Based on Convex Interval Taylor. In N. Beldiceanu, N. Jussien, and É. Pinson, editors, *Integration of AI and OR Techniques in Contraint Programming for Combinatorial Optimzation Problems: 9th International Conference, CPAIOR 2012, Nantes, France, May 28 - June1, 2012. Proceedings*, pages 1–16. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[2] F. Benhamou, F. Goualard, L. Granvilliers, and J. F. Puget. Revising hull and box consistency. In *Proceedings of the International Conference on Logic Programming*, pages 230–244, Las Cruces, NM, 1999.

[3] A. Burgard, D. Derr, D. Fox, and A. B. Cremers. Integrating global position estimation and position tracking for mobile robots: the dynamic markov localization approach. In *IROS Conference*, pages 730–735, Victoria, Canada, 1998.

[4] G. Chabert. IBEX, a C++ library for constraint processing over real numbers. http://www.ibex-lib.org.

[5] B. Desrochers and L. Jaulin. A minimal contractor for the polar equation: Application to robot localization. *Engineering Applications of Artificial Intelligence*, 55(Supplement C):83–92, Oct. 2016.

[6] B. Desrochers, S. Lacroix, and L. Jaulin. Set-membership approach to the kidnapped robot problem. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3715–3720, Sep. 2015.

[7] V. Drevelle and P. Bonnifait. Localization confidence domains via set inversion on short-term trajectory. *IEEE Transactions on Robotics*, 29:1244–1256, 2013.

[8] M. Drumheller. Mobile robot localization using sonar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):325–332, 1987.

[9] M. F. Fallon, J. Folkesson, H. McClelland, and J. J. Leonard. Relocating underwater features autonomously using sonar-based SLAM. *IEEE Journal of Oceanic Engineering*, 38(3):500–513, 2013.

[10] A. Gning and P. Bonnifait. Constraints propagation techniques on intervals for a guaranteed localization using redundant data. *Automatica*, 42(7):1167–1175, July 2006.

[11] J. Gutmann, A. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *IROS conference*, pages 736–743, Victoria, Canada, 1998.

[12] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *International conference on management data*, pages 47–57. ACM, 1984.

[13] L. Jaulin. *Mobile Robotics*. ISTE editions, 2015.

[14] L. Jaulin. Pure range-only SLAM with indistinguishable landmarks; a constraint programming approach. *Constraints*, pages 1–20, Oct. 2015.

[15] L. Jaulin and B. Desrochers. Robust localisation using separators. In *COPROD 2014*, 2014.

[16] C. Jermann, G. Trombettoni, B. Neveu, and P. Mathis. Decomposition of geometric constraint systems: a survey. *International Journal of Computational Geometry & Applications*, 16(05n06):379–414, 2006.

[17] R. B. Kearfott and V. Kreinovich, editors. *Applications of Interval Computations*. Kluwer, Dordrecht, the Netherlands, 1996.

[18] V. Kreinovich, L. Longpré, P. Patangay, S. Ferson, and L. Ginzburg. Outlier detection under interval uncertaity: Algorithmic solvability and computational complexity. In I. Lirkov, S. Margenov, J. Wasniewski, and P. Yalamov, editors, *Large-Scale Scientific Computing*, Proceedings of the 4th International Conference LSSC'2003, 2003.

[19] S. I. Kumkov and Y. V. Mikushina. Interval approach to identification of catalytic process parameters. *Reliable Computing*, 19(2):197–214, 2013.

[20] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, June 1991.

[21] J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer, Boston, 1992.

[22] S. McPhail. Autosub6000: A Deep Diving Long Range AUV. *Journal of Bionic Engineering*, 6(1):55–62, mar 2009.

[23] D. Meizel, O. Lévêque, L. Jaulin, and E. Walter. Initial localization by set inversion. *IEEE transactions on robotics and Automation*, 18(6):966–971, 2002.

[24] D. Meizel, A. Preciado-Ruiz, and E. Halbwachs. Estimation of mobile robot localization: geometric approaches. In M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, editors, *Bounding Approaches to System Identification*, pages 463–489. Plenum Press, New York, NY, 1996.

[25] R. Moore. *Methods and Applications of Interval Analysis*. Studies in Applied and Numerical Mathematics. Society for Industrial and Applied Mathematics, 1979.

[26] R. Neuland, J. Nicola, R. Maffei, L. Jaulin, E. Prestes, and M. Kolberg. Hybridization of Monte Carlo and set-membership methods for the global localization of underwater robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 199–204, Sept. 2014.

[27] L. Paull, M. Seto, S. Saeedi, and J. J. Leonard. *Navigation for Underwater Vehicles*, pages 1–15. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018.

[28] A. Rauh, E. Auer, T. Dotschel, and H. Aschemann. Verified stability analysis of continuous-time control systems with bounded parameter uncertainties and stochastic disturbances. *Computing*, 92(2):345–356, 2012.

[29] O. Reynet, L. Jaulin, and G. Chabert. Robust TDOA passive location using interval analysis and contractor programming. In *Radar*, Bordeaux, France, 2009.

[30] S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, and S. M. Veres. Guaranteed computation of robot trajectories. *Robotics and Autonomous Systems*, 93:76–84, 2017.

[31] S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, and S. M. Veres. Reliable non-linear state estimation involving time uncertainties. *Automatica*, 93:379–388, July 2018.

[32] S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, and S. M. Veres. *Reliable robot localization: a constraint-programming approach over dynamical systems*. ISTE Ltd, London, 2019.

[33] S. I. Seddik. *Localization of a Swarm of Underwater Robots Using Set-Membership Methods*. PhD dissertation, Université de Bretagne Occidentale, Brest, France, Nov. 2015.

[34] J. Sliwka, F. Le Bars, O. Reynet, and L. Jaulin. Using interval methods in the context of robust localization of underwater robots. In *NAFIPS 2011*, El Paso, USA, 2011.

[35] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[36] É. Walter and H. Piet-Lahanier. Estimation of the parameter uncertainty resulting from bounded-error data. *Mathematical Biosciences*, 92(1):55–74, 1988.

[37] S. B. Williams, P. Newman, J. Rosenblatt, G. Dissanayake, and H. Durrant-Whyte. Autonomous underwater navigation and control. *Robotica*, 19(5):481496, 2001.

[38] Y. Wu, X. Ta, R. Xiao, Y. Wei, D. An, and D. Li. Survey of underwater robot positioning navigation. *Applied Ocean Research*, 90:101845, 2019.