A Visibility Information for Multi-Robot Localization
Rémy Guyonneau, Sébastien Lagrange, Laurent Hardouin

To cite this version:
Rémy Guyonneau, Sébastien Lagrange, Laurent Hardouin. A Visibility Information for Multi-Robot Localization. IEEE/RJS International Conference on Intelligent Robots and Systems (IROS), Nov 2013, Tokyo, Japan. hal-00874164

HAL Id: hal-00874164
https://hal.archives-ouvertes.fr/hal-00874164
Submitted on 19 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
A Visibility Information for Multi-Robot Localization

Rémy Guyonneau, Sébastien Lagrange and Laurent Hardouin

Abstract—This paper proposes a set-membership method based on interval analysis to solve the pose tracking problem for a team of robots. The originality of this approach is to consider only weak sensor data: the visibility between two robots. The paper demonstrates that with this poor information, without using bearing or range sensors, a localization is possible. By using this boolean information (two robots see each other or not), the objective is to compensate the odometry errors and be able to localize in an indoor environment all the robots of the team, in a guaranteed way. The environment is supposed to be defined by two sets, an inner and an outer characteristics. This paper mainly presents the visibility theory used to develop the method. Simulated results allow to evaluate the efficiency and the limits of the proposed algorithm.

I. INTRODUCTION

Robot localization is an important issue in mobile robotics [1], [2], [3] since it is one of the most basic requirement for many autonomous tasks. The objective is to estimate the pose (position and orientation) of a mobile robot by using the knowledge of an environment (e.g. a map) and sensor data.

In this paper the pose tracking problem is considered: the objective is to compute the current pose of a robot knowing its previous one and avoiding its drifting. To compensate the drifting, due to odometry errors, external data are necessary. Contrary to most of the localisation approaches that use range sensors [4], [5], [6] this paper tends to prove that only weak informations can lead to an efficient localization too. The information to be considered is the visibility between robots: two robots are visible if there is no obstacle between them, else there are not visible. It can be noticed that visibility sensors have already been considered for localization and mapping [7], [8], [9]. But those approaches associate the visibility information to bearing and/or range measurements.

In this paper the proposed visibility corresponds to a boolean information (true or false), illustrated in Figure 1 and presented in Section III. This information can be obtained using 360° camera for example.

Note that the presented visibility information does not depend of the robots’ orientations (it is assumed that the robots can see all around themselves). In order to simplify the localization problem it is assumed that each robots are equipped with a compass. Thus the objective is to estimate the position $x_i = (x_i, y_i)$ of a robot $r_i$.

A robot $r_i$ is characterized by the following discrete time dynamic equation: $q_i(k+1) = f(q_i(k), u_i(k))$, with $k$ the discrete time, $q_i(k) = (x_i(k), \theta_i(k))$ the pose of the robot, $x_i(k) = (x_i(k), y_i(k))$ its position, $\theta_i(k)$ its orientation (associated to the compass) and $u_i(k)$ the input vector (associated to the odometry). The function $f$ characterizes the robot’s dynamics. In order to exploit the visibility information a team of $n$ robots $\mathcal{R} = \{r_1, \ldots, r_i, \ldots, r_n\}$ is considered.

The environment is assumed to be an indoor environment $\mathcal{E}$ composed by $m$ obstacles $e_j, j = 1, \ldots, m$. This environment is not known perfectly but is characterized by two known sets: $\mathcal{E}^-$ an inner characterization, and $\mathcal{E}^+$ an outer characterization, presented in the Section II-B.

To solve this problem a set-membership approach of the localization problem based on interval analysis is considered as in [10], [11].

II. ALGEBRAIC TOOLS

This section introduces some algebraic needful tools.

A. Interval analysis

An interval vector [12], or a box $[x]$ is defined as a closed subset of $\mathbb{R}^n$: $[x] = ([x_1], [x_2], \ldots) = ([x_1, x_1], [x_2, x_2], \ldots)$.

The size of an interval $[x]$ is defined as $\mu([x]) = (x_1 - x_1)$.

It can be noticed that any arithmetic operators such as $+$, $-$, $\times$, $\div$ and functions such as $\exp, \sin, \sqrt{\cdot}, \sqrt[\cdot]{\cdot}$ can be easily extended to intervals, [13].

A Constraint Satisfaction Problem (CSP) is defined by three sets. A set of variables $V$, a set of domains $D$ for those variables and a set of constraints $C$ connecting the variables together. Example of CSP:

$$D = \{x_1 \in [7, +\infty], x_2 \in [-\infty, 2], x_3 \in [-\infty, 9]\}. \quad (1)$$

Solving a CSP consists into reducing the domains by removing the values that are not consistent with the constraints. It can be efficiently solved by considering interval arithmetic [14]. For the example (1):

$$x_1 = x_2 + x_3 \Rightarrow x_1 \in [x_1] \cap ([-\infty, 2] + [-\infty, 9]),$$
$$x_2 = x_1 - x_3 \Rightarrow x_2 \in [x_2] \cap ([7, 11] - [-\infty, 9]),$$
$$x_3 = x_1 - x_2 \Rightarrow x_3 \in [x_3] \cap ([7, 11] - [-2, +\infty]),$$
$$x_3 = x_1 - x_2 \Rightarrow x_3 \in [x_3] \cap ([7, 11] - [-2, +\infty]),$$
$$x_3 = x_1 - x_2 \Rightarrow x_3 \in [x_3] \cap ([7, 11] - [-2, +\infty]),$$
$$x_3 = x_1 - x_2 \Rightarrow x_3 \in [x_3] \cap ([7, 11] - [-2, +\infty]).$$

The solutions of that CSP are the following contracted domains $[x_1]^* = [7, 11], [x_2]^* = [-2, 2]$ and $[x_3]^* = [5, 13]$. In this example a backward/forward propagation method is used to contract the domains. The forward propagation refers to the contraction of $[x_1]$, then the earned information is propagated to the domains $[x_2]$ and $[x_3]$, which corresponds to the backward step. In the proposed localization method, the backward/forward propagation is used to contract the robots’ poses.
Let \( E \) illustrate an environment \( E \) (black shapes) and its characterizations \( \delta^- \) (light grey segments) and \( \delta^+ \) (dark grey segments). It can be noticed that an obstacle can have an empty inner characterization.

Examples of visibility and non-visibility relations are

1) According to an obstacle \( \epsilon_j \): The visibility relation between two points \( x_1, x_2 \) regards to an obstacle \( \epsilon_j \) is defined as \( (x_1 \backslash x_2)_{\epsilon_j} \Leftrightarrow \text{Seg}(x_1, x_2) \cap \epsilon_j = \emptyset \), with \( \text{Seg}(x_1, x_2) \) the segment defined by the points \( x_1 \) and \( x_2 \).

The visible space of a point \( x \) regards to an obstacle \( \epsilon_j \) with \( x \cap \epsilon_j = \emptyset \), is defined as \( E^v_j(x) = \{ x \mid (x \backslash x_1)_{\epsilon_j} \} \), and the non-visible space of \( x \) regards to \( \epsilon_j \) is defined as \( E^v_j(x) = E^v_j(x) \).

Examples of visible and non-visible spaces are presented in Figure 2.

2) According to an environment \( \delta \): As the robots are moving in an environment \( \delta \) composed by \( m \) obstacles, it is needed to extend the previous definitions to multiple obstacles:

\[
(x_1 \backslash x_3)_{\delta} \Leftrightarrow \text{Seg}(x_1, x_3) \cap \delta = \emptyset, \\
E^v_{\delta}(x) = \{ x \mid (x \backslash x_3)_{\delta} \} = \bigcap_{j=1}^{m} (x \backslash x_j)_{\delta}, \\
E^v_{\delta}(x) = E^v_{\delta}(x), \\
E^v_{\delta}(x) = E^v_{\delta}(x).
\]

It is possible to characterize the visibility over an environment by considering the visibility regards to the obstacles that composed this environment.

**Lemma 1:** Let \( x_1 \) and \( x_2 \) be two distinct points and \( \delta \) an environment, with \( x_1 \notin \delta \) and \( x_2 \notin \delta \). Then

\[
(x_1 \backslash x_3)_{\delta} \Leftrightarrow \bigcap_{j=1}^{m} (x_1 \backslash x_j)_{\delta}, \\
E^v_{\delta}(x) = \bigcup_{j=1}^{m} E^v_{\delta}(x).
\]

**Lemma 2:** Let \( x \) be a point and \( \delta \) an environment such as \( x \notin \delta \). Then

\[
E^v_{\delta}(x) = \bigcup_{j=1}^{m} E^v_{\delta}(x), \\
E^v_{\delta}(x) = \bigcap_{j=1}^{m} E^v_{\delta}(x).
\]

**Lemma 3:** Let \( x_1 \) and \( x_2 \) be two points, \( \delta \) an environment such as \( x \notin \delta \), and \( \delta^- \) and \( \delta^+ \) the inner and outer characterizations of the environment. Then

\[
(x_1 \backslash x_3)_{\delta} \Leftrightarrow \bigcap_{j=1}^{m} (x_1 \backslash x_j)_{\delta}, \\
(x_1 \backslash x_2)_{\delta} \Rightarrow (x_1 \backslash x_2)_{\delta^-}, \\
(x_1 \backslash x_2)_{\delta} \Rightarrow (x_1 \backslash x_2)_{\delta^+}.
\]
It is possible to extend those notions to an environment

\[ E_\ell(\mathcal{X}) = \{ x | \forall x \in \mathcal{X}, (x \lor x) \ell \} \]  
\[ E_\ell(\mathcal{X}) = \{ x | \forall x \in \mathcal{X}, (x \land x) \ell \} \]  

The visibility over an environment can be characterized by considering the visibility regards to the obstacles that composed this environment.

**Lemma 4:** Let \( \mathcal{X} \) be a connected and \( \ell \) an environment with \( \mathcal{X} \cap \ell = \emptyset \). Then,

\[ E_\ell(\mathcal{X}) = \bigcap_{j=1}^{m} E_{\ell_j}(\mathcal{X}) \]  
\[ E_\ell(\mathcal{X}) \supseteq \bigcup_{j=1}^{m} E_{\ell_j}(\mathcal{X}) \]

**Figure 5** illustrates the inclusion of Equation 18.

The following lemma provides a relation between the visibility according to the environment and the characterizations.

**Lemma 5:** Let \( \mathcal{X}_1 \) and \( \mathcal{X}_2 \) be two distinct points, with \( \mathcal{X}_1 \), \( \mathcal{X}_1 \) two connected sets such as \( \mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset \). Considering an environment \( \ell \) with its characterizations \( \ell^+ \) and \( \ell^- \)

\[(x_1 \lor x_2) \ell \rightarrow \begin{cases} x_1 \subseteq x_1 \cap (\bigcup_{j=1}^{m} E_{\ell_j}^-(x_2)) \\ x_2 \subseteq x_2 \cap (\bigcup_{j=1}^{m} E_{\ell_j}^-(x_1)) \end{cases} \]  
\[(x_1 \land x_2) \ell \rightarrow \begin{cases} x_1 \subseteq x_1 \cap (\bigcap_{j=1}^{m} E_{\ell_j}^+(x_2)) \\ x_2 \subseteq x_2 \cap (\bigcap_{j=1}^{m} E_{\ell_j}^+(x_1)) \end{cases} \]

This lemma is an extension of Lemma 3.

**IV. THE CONTRACTORS**

In this section the two contractors \( C_\ell(x_1, x_2, \ell_j^+) \) and \( C_\ell^-(x_1, x_2, \ell_j^-) \) are presented. A contractor is an operator that can remove the points of the domains \( [x_1] \) and \( [x_2] \) that are not consistent with a given constraint (visibility information). In our case the contractor \( C_\ell \) contracts over the visibility relation and \( C_\ell^- \) over the non-visibility relation. The Figure 5 presents an example of contraction according to the visibility and non-visibility. Those contractors are based on Equations 19 and 20. It can be noticed that the computation of the visible and non-visible spaces \( E_{\ell_j}^+([x_2]) \) and \( E_{\ell_j}^-([x_2]) \) are needed to contract the domains \( [x_1] \) and \( [x_2] \).

**Figure 4.** Let \( x_1 \in [x_1] \) and \( x_2 \in [x_2] \) be two points such that \( (x_1 \lor x_2) \ell_j \), then using the contractor \( C_\ell^+(x_1, x_2, \ell_j) \) it is possible to remove the hatched parts of the domains \( [x_1] \) and \( [x_2] \). **Right Figure:** With \( (x_1 \land x_2) \ell_j \), it is possible to contract the hatched parts.

Considering a segment \( \ell_j \) as an obstacle, the visible and non-visible spaces of a box \( [x] \) regards to the obstacle are delimited by lines. Those lines are passing throw the segment bounds and the box vertices (Figure 5). The objective is to identify the extremal lines that characterize the visible and non-visible spaces. It can be noticed that those lines correspond to the lines with the maximal and minimal slopes (Figure 5).

**Remark 2:** In order to avoid line singularities, the determinant is used to characterize the lines. Let \( a = (a_1, a_2) \), \( b = (b_1, b_2) \) and \( c = (c_1, c_2) \) be three points, the sign of \( \det(a - b|c - b) = (a_1 - b_1)(c_2 - b_2) - (a_2 - b_2)(c_1 - b_1) \) indicates the side of \( a \) regards to the vector \( bc \) (Figure 6).

**Fig. 6.** The sign of \( \det(a - b|c - b) \) depends of the side of \( a \) regards to \( bc \).

**1) Equation of the non-visible space of a box:** The non-visible space of a box \( [x] \) regards to an obstacle \( \ell_j \) = \( \text{Seg}(e_1, e_2) \) corresponds to the intersection of the non-visible spaces of the vertices of the box:

\[ E_{\ell_j}^-(x) = \bigcap_{z=1}^{4} E_{\ell_j}^-(x_z) \]

with \( x_1, x_2, x_3, x_4 \) the vertices of the box \( [x] \) (Figure 5).

**Remark 3:** The following equations correspond to the non-visible space of a point \( x \) regards to an obstacle \( \ell_j = \text{Seg}(e_1, e_2) \):
\[ \text{Seg}(e_{1j}, e_{2j}) \]
\[ \mathbb{E}_{e_j}(x_1) = \{ x_1 | \zeta_{x_1} \det(x_i - e_{1j}|e_{2j} - e_{1j}) \leq 0 \wedge \zeta_{x_1} \det(x_i - x_{1j}|e_{1j} - x_{1j}) \geq 0 \wedge \zeta_{x_1} \det(x_i - x_{1j}|e_{2j} - x_{1j}) \leq 0 \}, \] (22)

with
\[ \zeta_{x_1} = \begin{cases} 1 & \text{if } \det(x_i - e_{1j}|e_{2j} - e_{1j}) > 0, \\ -1 & \text{else.} \end{cases} \]

The Figure 7 presents an example of non-visibility characterization. In this example \( \zeta_{x_1} = -1 \) (Figure 6). \( \mathbb{E}_{e_j}(x_1) \) is then characterized by the points \( x_i \) such that
\[ \det(x_i - e_{1j}|e_{2j} - e_{1j}) \geq 0 \text{ and } \det(x_i - x_{1j}|e_{2j} - e_{1j}) \leq 0 \] (Equation 22). This corresponds to all the points above the line \((e_{1j}, e_{2j})\), under the line \((x_1, e_{1j})\) and above the line \((x_1, e_{2j})\) (Figure 6).

From the Equations 19 and 21 it can be deduced that
\[ (x_1 V x_2)e_j \Rightarrow [x_1]^* = [x_1] \cap \bigcup_{z=1}^{4} \mathbb{E}_{e_j}(x_2). \] (23)

with \( x_1 \in [x_1] \) and \( x_2 \in [x_2] \).

According to the equations 23 and 22 it is possible to build the visibility contractor \( C_v([x_1], [x_2], e_j) \), presented in the Algorithm 1. This contractor uses the backward/forward propagation presented in the Section II-A. It can be noticed that the equations lines 4 to 6 correspond to the complement of the Equation 22 (the \( \wedge \) become \( \vee \) and the signs change).

**Algorithm 1: \( C_v([x_1],[x_2],e_j) \)**

**Data:** \([x_1],[x_2],e_j = \text{Seg}(e_{1j},e_{2j})\)

1. \( \text{contraction of } [x_1] \);  
2. \( \text{for } z \text{ to 4 do} \)
3. \( \text{backward/forward propagation over} \)
4. \( \zeta_{x_1} \det([x_1] - e_{1j}|e_{2j} - e_{1j}) > 0 \vee \)
5. \( \zeta_{x_1} \det([x_1] - x_{1j}|e_{1j} - x_{1j}) < 0 \vee \)
6. \( \zeta_{x_1} \det([x_1] - x_{1j}|e_{2j} - x_{1j}) > 0 \);  
7. \( \text{The resulting box is noted } [x_1]^* \).  
8. \( [x_1]^* = \bigcup_{z=1}^{4} [x_1]^* \);  
9. \( \text{The same idea for the contraction of } [x_2] \);  
10. **Result:** \([x_1]^*, [x_2]^* \).  

2) Equation of the visible space of a box: Whereas the computation of the non-visible space of a box can be simplified to the computation of the non-visible spaces of its vertices (Equation 21), for the visible space it is needed to test all the possible lines. Let \([x] \) be a box with \( x_{z} = 1, \cdots, 4 \) its vertices and \( e_j \) an obstacle, the visible space of the box regards to the obstacle can be defined as

\[ \mathbb{E}_{e_j}([x]) = \bigcap_{z=1}^{4} \{ x_i | (\zeta_{x_i} \det(x_i - e_{1j}|x_i - e_{1j}) > 0 \wedge \zeta_{x_{i+1}} \det(x_i - e_{1j}|x_{i+1} - e_{1j}) > 0 \vee \zeta_{x_{i+1}} \det(x_i - e_{1j}|x_{i+1} - e_{1j}) < 0 \wedge \zeta_{x_{i+1}} \det(x_i - e_{1j}|x_{i+1} - e_{1j}) < 0 \vee \zeta_{x_{i+1}} \det(x_i - e_{1j}|x_{i+1} - e_{1j}) < 0 \wedge \zeta_{x_{i+1}} \det(x_i - e_{1j}|x_{i+1} - e_{1j}) < 0 \wedge \zeta_{x_{i+1}} \det(x_i - e_{1j}|x_{i+1} - e_{1j}) < 0 \} \). \] (24)

with
\[ x_5 = x_1, \]
\[ \zeta_{x_5} = \begin{cases} 1 & \text{if } \det(x_i - e_{1j}|e_{2j} - e_{1j}) > 0, \\ -1 & \text{else.} \end{cases} \]
\[ \zeta_{x_{i+1}} = \begin{cases} 1 & \text{if } \det(x_i - e_{1j}|e_{2j} - e_{1j}) > 0, \\ -1 & \text{else.} \end{cases} \]
\[ \zeta_{x_{i+1}} = \begin{cases} 1 & \text{if } \det(e_{1j} - x_i|e_{2j} - x_i) > 0, \\ -1 & \text{else.} \end{cases} \]
\[ \zeta_{x_{i+1}} = \begin{cases} 1 & \text{if } \det(e_{2j} - x_i|x_i - x_1) > 0, \\ -1 & \text{else.} \end{cases} \]

The first six relations of the Equation 24 determinate the lines with the maximal and minimal slopes. The last four equations deal with a singularity presented in the Figure 8. Without those four equations, the partial-visible space (medium grey) could be considered as included in the visible space.

Note that the non-visibility contractor \( C_v([x_1], [x_2], e_j) \) can be built as it is done for the visibility contractor presented in the Section IV-A.

V. THE POSE TRACKING ACCORDING TO THE VISIBILITY

A. The Pose Tracking Algorithm

As mentioned in the introduction we are interested in the pose tracking localization problem. Knowing the initial pose \( q_i(k_0) \) of a robot \( r_i \), the objective is to estimate the pose \( q_i(k) \) at each time \( k \). Using the dynamic equation of the system (Section I) it is possible to compute the pose of the robots at time \( k \) knowing the pose at time \( k - 1 \). To be able to
compute the new pose, the orientation $\theta_i(k)$ is measured by the compass and the input vector $u_i(k)$ is estimated by the odometry. In order to deal with the sensor imprecisions, we consider a bounded error context. Thus it is possible to define $[\theta_i(k)]$ and $[u_i(k)]$ according to the sensors’ measurements, such that $[\theta_i(k)] \in [\theta_i(k)]$ and $[u_i(k)] \in [u_i(k)]$, and $[q_i(k)]$ the initial robot’s pose estimation such that $[q_i(0)] \in [q_i(0)]$. In this context it is possible to compute the pose $[q_i(k+1)] = f([q_i(k)], [u_i(k)])$ using interval analysis principles.

In order to avoid the drifting of the robots (the increase of $[q_i(k)]$ size), the visibility information between the robots is considered. At each time $k$ each robot computes the visibility information regards to the other robots of the team. Let $r_i$ and $r_f$ be two different robots of $\mathcal{R}$.

\begin{align}
    r_i \text{ sees } r_f \Leftrightarrow (x_f, Vx_f)_{e_i}, \\
    r_i \text{ does not see } r_f \Leftrightarrow (x_f, Vx_f)_{e_f}.
\end{align}

It is also needed that at each time $k$, each robot $r_i$ communicates its current pose estimation $[q_i(k)]$ with the team $\mathcal{R}$.

Algorithm 2 presents the proposed pose tracking approach. First, Line 1, the initial poses of the robots are defined. Line 3, for each robots, the new pose is estimated regards to the knowledge of the previous one. Line 4, the robots share their pose estimations with the team. Finally, Lines 5 to 9 the visibility information is used to contract the robot’s pose estimations. Lines 7 and 9, two contractors are used: the visibility contractor $C_V$ and the non-visibility contractor $C_T$. The objective of those functions is to remove from the domains $[x_i]$ and $[x_f]$, the values that are not consistent with the visibility and non-visibility informations. They are detailed in the previous Section.

**Algorithm 2: The pose tracking algorithm**

Data: $\mathcal{R}$, $\mathcal{E}^-$, $\mathcal{E}^+$

1. $\forall r_i \in \mathcal{R}$, initialize $[q_i(0)]$
2. for $k = 1$ to end do
3.   $\forall r_i \in \mathcal{R}, [q_i(k)] = f([q_i(k-1)], [u_i(k-1)])$
4.   $\forall r_i \in \mathcal{R}$, share $[q_i(k)]$ with the team;
5.    forall the $r_i \in \mathcal{R}$, $r_f \in \mathcal{R}, r_i \neq r_f$ do
6.      if $r_i$ sees $r_f$ then
7.         $([q_i(k)]^+, [x_f(k)]^*) = \bigcap_{e_f \in \mathcal{E}^-} \{C_V([x_i(k)], [x_f(k)], e_f^-)\}$
8.      else
9.         $([q_i(k)]^+, [x_f(k)]^*) = \bigcup_{e_f \in \mathcal{E}^+} \{C_T([x_i(k)], [x_f(k)], e_f^+)\}$

**B. Experimental Results**

In order to test this approach, a simulator has been developed. The efficiency of the algorithm has been tested for three different environments $\mathcal{E}^1$, $\mathcal{E}^2$ and $\mathcal{E}^3$ (Figure 9). Each environment has a $10 \times 10$ m$^2$ size. It can be noticed that the simulated environments are polygonal. This has been done in order to simplify the computation of simulated data. The proposed algorithm manipulates only the inner and outer characterisations and would work as well in a non-polygonal environments (the characterisations considered for the presented experimentations are not perfect and could have been associated to non-polygonal shapes).

The following table presents the number of segments of the characterisations of each environment:

<table>
<thead>
<tr>
<th>Environment</th>
<th>$\mathcal{E}^1$</th>
<th>$\mathcal{E}^2$</th>
<th>$\mathcal{E}^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{E}^-$</td>
<td>19</td>
<td>59</td>
<td>89</td>
</tr>
<tr>
<td>$\mathcal{E}^+$</td>
<td>26</td>
<td>69</td>
<td>101</td>
</tr>
</tbody>
</table>

At each iteration (one moving and one contraction step) a robot does a 20cm distance move, with a bounded error of $\pm 1\%$, and has a bounded compass error of $\pm 1$ deg. Note that $\forall r_i \in \mathcal{R}, [x_i(k)] = [x_i(0)] - 50cm, \forall x_i(k) + 50cm$.

The processor used for the simulations has the following characteristics: Intel(R) Core(TM) CPU - 6420 @ 2.13GHz.

During those tests the simulated robots moved randomly in the environment, from $k_0 = 0$ to $k = 1500$. The results of the experimentations are presented in the Table I. Note that $\text{average } w([x_{1,2,3}])$ corresponds to the average size of $[x_{1,2,3}]$ during all experimentation and $\text{final } w([x_{1,2,3}])$ corresponds to the average of $[x_{1,2,3}]$ just for the final step.

As it can be noticed that the size of the initial boxes $w([x_{1,2,3}])$ are equal to 100cm (initial incertitude about the position). It can be concluded that the experimentations providing a final incertitude around 100cm (or smaller) lead to successful localizations (avoiding the drifting of the robots). In addition to that it is possible to classify as successful the experimentations that have: average $w([x_{1,2,3}]) \approx 100cm$ (the imprecision is maintained and do not increase).
experimentations is borderline as it only considers the weak boolean information. Note that the context of the presented algorithm is a guaranteed method that is able to exploit this same argument with too many obstacles. It can be explained by the fact that without any in the environment are important factors for an efficient localization process can be efficient in one environment but increase the computation time. The number of obstacles, their sizes and their dispositions environment is an important factor for the efficiency of the implemented localization. It appears in Section V-B that the topology of an environment is an important factor for the efficiency of the proposed localization. In a future work it could be interesting to characterize the environments, allowing to calculate for a given environment, a minimal number of robots required to perform a pose tracking.

Finally we are planning to consider a maximal range for the visibility information and to restrain the field of vision of the robots.

### EXPERIMENTAL RESULTS

<table>
<thead>
<tr>
<th>number of robots</th>
<th>4</th>
<th>8</th>
<th>12</th>
<th>16</th>
<th>20</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>average w((x_{i}))</td>
<td>439.0</td>
<td>316.0</td>
<td>196.0</td>
<td>131.0</td>
<td>127.0</td>
<td>115.0</td>
</tr>
<tr>
<td>average w((x_{i}))</td>
<td>460.0</td>
<td>334.0</td>
<td>176.0</td>
<td>147.0</td>
<td>121.0</td>
<td>117.0</td>
</tr>
<tr>
<td>final w((x_{i}))</td>
<td>897.0</td>
<td>668.0</td>
<td>172.0</td>
<td>147.0</td>
<td>121.0</td>
<td>117.0</td>
</tr>
<tr>
<td>final w((x_{i}))</td>
<td>919.0</td>
<td>674.0</td>
<td>193.0</td>
<td>167.0</td>
<td>159.0</td>
<td>133.0</td>
</tr>
<tr>
<td>iteration time</td>
<td>5.0</td>
<td>28.0</td>
<td>92.0</td>
<td>204.0</td>
<td>331.0</td>
<td>389.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>number of robots</th>
<th>4</th>
<th>8</th>
<th>12</th>
<th>16</th>
<th>20</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>average w((x_{i}))</td>
<td>545.0</td>
<td>355.0</td>
<td>183.0</td>
<td>102.0</td>
<td>93.0</td>
<td>84.0</td>
</tr>
<tr>
<td>average w((x_{i}))</td>
<td>552.0</td>
<td>375.0</td>
<td>185.0</td>
<td>115.0</td>
<td>108.0</td>
<td>97.0</td>
</tr>
<tr>
<td>final w((x_{i}))</td>
<td>1029.0</td>
<td>780.0</td>
<td>440.0</td>
<td>107.0</td>
<td>100.0</td>
<td>62.0</td>
</tr>
<tr>
<td>final w((x_{i}))</td>
<td>1038.0</td>
<td>811.0</td>
<td>435.0</td>
<td>110.0</td>
<td>97.0</td>
<td>95.0</td>
</tr>
<tr>
<td>iteration time</td>
<td>11.0</td>
<td>65.0</td>
<td>191.0</td>
<td>443.0</td>
<td>745.0</td>
<td>1160.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>number of robots</th>
<th>4</th>
<th>8</th>
<th>12</th>
<th>16</th>
<th>20</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>average w((x_{i}))</td>
<td>404.0</td>
<td>120.0</td>
<td>75.0</td>
<td>62.0</td>
<td>63.0</td>
<td>56.0</td>
</tr>
<tr>
<td>average w((x_{i}))</td>
<td>325.0</td>
<td>77.0</td>
<td>59.0</td>
<td>50.0</td>
<td>47.0</td>
<td>46.0</td>
</tr>
<tr>
<td>final w((x_{i}))</td>
<td>817.0</td>
<td>120.0</td>
<td>81.0</td>
<td>69.0</td>
<td>87.0</td>
<td>87.0</td>
</tr>
<tr>
<td>final w((x_{i}))</td>
<td>688.0</td>
<td>65.0</td>
<td>58.0</td>
<td>50.0</td>
<td>57.0</td>
<td>45.0</td>
</tr>
<tr>
<td>iteration time</td>
<td>15.0</td>
<td>116.0</td>
<td>313.0</td>
<td>646.0</td>
<td>1058.0</td>
<td>1727.0</td>
</tr>
</tbody>
</table>

**TABLE I.**

**EXPERIMENTAL RESULTS** (the values are in cm and ms).

Those successful experimentations are depicted in bold in Table I. **Iteration time** correspond to the average iteration time of the 1500 iteration step time (in ms).

Looking at those results it appears that two elements are important factors for the success of the localization: the topology of the environment and the number of robots.

It appears that for a given environment a minimal number of robots is necessary to perform an efficient pose tracking. It can be explained by the fact that with few robots, the visibility measurements carry few informations. In our experimentations, at least 8 robots are necessary to perform an efficient localization in the environment \(E_1\), 12 for the environment \(E_2\) and 16 for the environment \(E_3\). On the other hand, too many robots does not improve significantly the localization but increase the computation time.

It also appears that for a given number of robots the localization process can be efficient in one environment whereas it is not in the others. For instance with a team of 8 robots, the algorithm provides a good localization in the environment \(E_2\), but not in the environment \(E_1\) neither in \(E_3\). The number of obstacles, their sizes and their dispositions in the environment are important factors for an efficient localization. It can be explained by the fact that without any obstacle, or with too small obstacles, the robots see each other constantly, thus the visibility sensor will return always the same value and will not provide useful information. It is the same argument with too many obstacles.

**C. Conclusion**

In this paper it is shown that using interval analysis it is possible to localize a team of robots only assuming weak informations: the visibility between the robots. The proposed algorithm is a guaranteed method that is able to exploit this boolean information. Note that the context of the presented experimentations is borderline as it only considers the weak boolean visibility information. In practice this information can be added to classical localization methods, using range sensors for example, when a team of robots is considered, as in [15], [16].

References