HAL
open science

Mobile Robots Pose Tracking: a Set-Membership Approach Using a Visibility Information

Rémy Guyonneau, Sébastien Lagrange, Laurent Hardouin

To cite this version:

Rémy Guyonneau, Sébastien Lagrange, Laurent Hardouin. Mobile Robots Pose Tracking: a Set-Membership Approach Using a Visibility Information. 9th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Jul 2012, Rome, Italy. pp.x-x. hal-00846576

HAL Id: hal-00846576

https://hal.science/hal-00846576v1

Submitted on 19 Jul 2013

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.

Mobile Robots Pose Tracking: a Set-Membership Approach Using a Visibility Information

Rémy Guyonneau, Sébastien Lagrange and Laurent Hardouin
Laboratoire d'Ingénierie des Systèmes Automatisés (LISA), Université d'Angers, 62 avenue Notre Dame du Lac, Angers, France
{remy.guyonneau,sebastien.lagrange,laurent.hardouin}@univ-angers.fr

Keywords: Mobile Robots : Localization : Pose Tracking : Visibility : Interval Analysis.

Abstract: This paper proposes a set-membership method based on interval analysis to solve the pose tracking problem. The originality of this approach is to consider weak sensors data: the visibility between two robots. By using a team of robots and this boolean information (two robots see each other or not), the objective is to compensate the odometry errors and be able to localize, in a guaranteed way, the robots in an indoor environment. This environment is supposed to be defined by two sets, an inner and an outer characterizations. Simulated results allow to evaluate the efficiency and the limits of the proposed method.

1 INTRODUCTION

Robot localization is an important issue in mobile robotics (J. Borenstein, 1996; M.J. Segura, 2009; J. Zhou, 2011) since it is one of the most basic requirement for many autonomous tasks. The objective is to estimate the pose (position and orientation, Figure 1) of a mobile robot by using the knowledge of an environment (*e.g.* a map) and sensors data.

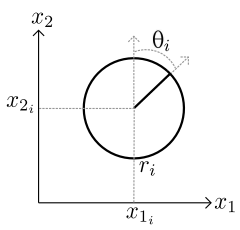

Figure 1: A robot r_i with a pose $\mathbf{q}_i = (\mathbf{x}_i, \theta_i)$. The vector $\mathbf{x}_i = (x_{1_i}, x_{2_i})^T$ represents its position and θ_i its orientation in the environment.

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[1] errors, external data are necessary. Contrary to most of the localisation approaches that use range sensors (P. Jensfelt, 2001; K. Lingemann, 2005; Abeles, 2011) this paper tends to prove that *weak* informations can lead to an efficient localization too. The chosen information is the visibility between two robots: two robots are visible if there is no obstacle between them, else there are not visible. This is a boolean information defined in Section 2.1.

Note that this information does not depend of the robots' orientation. That is the reason why θ_i is assumed to be given by a compass. The objective is then to estimate the position $\mathbf{x}_i = (x_{1_i}, x_{2_i})^T$ of a robot r_i.

A robot r_i is characterized by the following discrete time dynamic equation:

$$\mathbf{q}_i(k+1) = f(\mathbf{q}_i(k), \mathbf{u}_i(k)) \tag{1}$$

with k the discrete time, $\mathbf{q}_i(k) = (\mathbf{x}_i(k), \theta_i(k))$ the pose of the robot, $\mathbf{x}_i(k) = (x_{1_i}(k), x_{2_i}(k))^T$ its position, and $\mathbf{u}_i(k)$ the input vector (associated with the odometry and the compass). The function f characterizes the robot's dynamics. In order to exploit the visibility information a team of m robots $\mathcal{R} = \{r_1, \cdots, r_i, \cdots, r_m\}$ have to be considered.

In this paper the environment is assumed to be an indoor environment $\mathcal{E}$ composed by n obstacles $\mathcal{E}_j$, $j = 1, \cdots, n$. This environment is not known but is characterized by two known sets (see Figure 2): $\mathcal{E}^-$, an inner characterization, and $\mathcal{E}^+$ an outer characterization, such as $\mathcal{E}^- \subset \mathcal{E} \subset \mathcal{E}^+$.

To solve this problem a set-membership approach of the localization problem based on interval analysis is considered (E. Seignez, 2005; Jaulin, 2009). In this context the the LUVIA algorithm (**L**ocalisation **U**pdating with **V**isibility and **I**nterval **A**nalysis) has been developed to solve the pose tracking problem.

[1] Sensors that compute the moves of a robot.

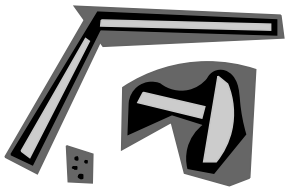

Figure 2: Example of an environment and its characterizations. The black shapes correspond to the environment $\mathcal{E}$, the dark grey shapes correspond to an outer characterization $\mathcal{E}^+$ and the light grey shapes correspond to an inner characterization $\mathcal{E}^-$. Note that $\mathcal{E}^-$ can be empty.

2 ALGEBRAIC TOOLS

This section introduces some algebraic tools needful in this paper. First the visibility is defined in Section 2.1. Then interval analysis is presented in Section 2.2 and the environment characterizations are presented in Section 2.3.

2.1 The visibility

The considered *weak* information to solve the pose tracking problem is the visibility between two robots. This corresponds to two binary relations: the visibility relation and the non-visibility relation (Definitions 1 and 4). Figure 3 shows an example of visibility informations.

Note that assuming that $\mathbf{x}_1$ and $\mathbf{x}_2$ are the positions of two robots, $Seg(\mathbf{x}_1,\mathbf{x}_2)$ denotes the segment from $\mathbf{x}_1$ to $\mathbf{x}_2$.

Definition 1. *The visibility between two robots r_1 and r_2 with their respective positions $\mathbf{x}_1$ and $\mathbf{x}_2$ in an environment $\mathcal{E}$ is a binary relation noted* V *such as*

$$(\mathbf{x}_1 \mathrm{V} \mathbf{x}_2)_{\mathcal{E}} \Leftrightarrow Seg(\mathbf{x}_1,\mathbf{x}_2) \cap \mathcal{E} = \emptyset. \quad (2)$$

Properties 2.

- V *is symmetric, i.e.,* $\forall \mathbf{x}_1, \forall \mathbf{x}_2, (\mathbf{x}_1 \mathrm{V} \mathbf{x}_2)_{\mathcal{E}} \Rightarrow (\mathbf{x}_2 \mathrm{V} \mathbf{x}_1)_{\mathcal{E}}$,
- V *is reflexive, i.e.,* $\forall \mathbf{x}, (\mathbf{x} \mathrm{V} \mathbf{x})_{\mathcal{E}}$.

Remark 3. V *is not transitive since* $(\mathbf{x}_1 \mathrm{V} \mathbf{x}_2)_{\mathcal{E}}$ *and* $(\mathbf{x}_2 \mathrm{V} \mathbf{x}_3)_{\mathcal{E}} \not\Rightarrow (\mathbf{x}_1 \mathrm{V} \mathbf{x}_3)_{\mathcal{E}}$.

Definition 4. *The non-visibility between two robots r_1 and r_2 with their respective positions $\mathbf{x}_1$ and $\mathbf{x}_2$ in an environment $\mathcal{E}$ is a binary relation noted* $\overline{\mathrm{V}}$ *such as*

$$(\mathbf{x}_1 \overline{\mathrm{V}} \mathbf{x}_2)_{\mathcal{E}} \Leftrightarrow Seg(\mathbf{x}_1,\mathbf{x}_2) \cap \mathcal{E} \neq \emptyset. \quad (3)$$

Properties 5.

- $\overline{\mathrm{V}}$ *is the complement of* V,
- $\overline{\mathrm{V}}$ *is symmetric, i.e.,* $\forall \mathbf{x}_1, \forall \mathbf{x}_2, (\mathbf{x}_1 \overline{\mathrm{V}} \mathbf{x}_2)_{\mathcal{E}} \Rightarrow (\mathbf{x}_2 \overline{\mathrm{V}} \mathbf{x}_1)_{\mathcal{E}}$.

Remark 6. $\overline{\mathrm{V}}$ *is not transitive since* $(\mathbf{x}_1 \overline{\mathrm{V}} \mathbf{x}_2)_{\mathcal{E}}$ *and* $(\mathbf{x}_2 \overline{\mathrm{V}} \mathbf{x}_3)_{\mathcal{E}} \not\Rightarrow (\mathbf{x}_1 \overline{\mathrm{V}} \mathbf{x}_3)_{\mathcal{E}}$.

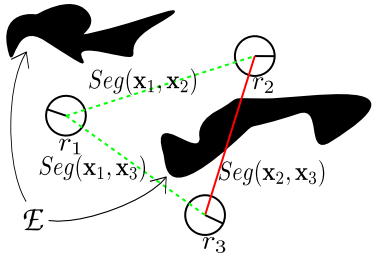

Figure 3: Example of visibility. The black shapes represent the environment $\mathcal{E}$ and r_1, r_2 and r_3 three robots. In this example: $(\mathbf{x}_1 \mathrm{V} \mathbf{x}_3)_{\mathcal{E}}$, $(\mathbf{x}_1 \mathrm{V} \mathbf{x}_2)_{\mathcal{E}}$, $(\mathbf{x}_2 \overline{\mathrm{V}} \mathbf{x}_3)_{\mathcal{E}}$.

2.2 Interval analysis

An *interval* is a closed subset of $\mathbb{R}$, noted $[x] = [\underline{x}, \overline{x}]$, with $\underline{x}$ its lower bound and $\overline{x}$ its upper bound. An *interval vector* (L. Jaulin, 2001; R. E. Moore, 2009), or a *box*, $[\mathbf{x}]$ is defined as a closed subset of $\mathbb{R}^n$:$[\mathbf{x}] = ([x_1],[x_2],\cdots) = ([\underline{x_1},\overline{x_1}],[\underline{x_2},\overline{x_2}],\cdots)$.

The size of a box is defined as

$$Size([\mathbf{x}]) = (\overline{x_1} - \underline{x_1}) \times (\overline{x_2} - \underline{x_2}) \times \cdots. \quad (4)$$

It can be noticed that any arithmetic operators such as $+,-,\times,\div$ and functions such as $exp, sin, sqr, sqrt, \ldots$ can be easily extended to intervals (Neumaier, 1991).

The hull of a set of intervals corresponds to the smallest connected interval that enclosed all the intervals. It can be easily extended to interval vectors.

The $Bisect()$ function divides an interval $[x]$ into two intervals $[x_1]$ and $[x_2]$ such as $[x_1] \cup [x_2] = [x]$, $[x_1] \cap [x_2] = \emptyset$ and $Size([x_1]) = Size([x_2])$. As for the hull, the bisection can be extended to interval vectors.

2.3 The environment characterization

As said in Section 1, the environment $\mathcal{E}$ is characterized by two sets $\mathcal{E}^-$ and $\mathcal{E}^+$. In this paper those sets are assumed to be interval segment sets. In this section interval segments are first defined then a final definition of $\mathcal{E}^-$ and $\mathcal{E}^+$ is given.

Definition 7. *Let $[\mathbf{x}_1]$ and $[\mathbf{x}_2]$ be two boxes, an interval segment (see Figure 4) is defined by*

$$Seg([\mathbf{x}_1],[\mathbf{x}_2]) = \{Seg(\mathbf{x}_1,\mathbf{x}_2) | \mathbf{x}_1 \in [\mathbf{x}_1], \mathbf{x}_2 \in [\mathbf{x}_2]\} \quad (5)$$

Note that it is possible to extend the segment intersection to interval segments.

Proposition 8. *Let $Seg([\mathbf{x}_1],[\mathbf{x}_2])$ and $Seg([\mathbf{x}_3],[\mathbf{x}_4])$ be two interval segments, the two following conditions hold*

$$Intersect(Seg([\mathbf{x}_1],[\mathbf{x}_2]), Seg([\mathbf{x}_3],[\mathbf{x}_4])) < 0 \quad (6)$$

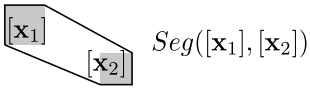

Figure 4: An interval segment. The light grey boxes represent the ended boxes of the interval segment and the black shape represents the interval segment.

$\Rightarrow$ *the two interval segments intersect,*

$$Intersect(Seg([\mathbf{x}_1],[\mathbf{x}_2]),Seg([\mathbf{x}_3],[\mathbf{x}_4])) > 0 \quad (7)$$

$\Rightarrow$ *the two interval segments do not intersect.*

Where the Intersect(.,.) function is defined in Appendix Definition 17.

Proof. for (6)
$Intersect(Seg([\mathbf{x}_1],[\mathbf{x}_2]),Seg([\mathbf{x}_3],[\mathbf{x}_4])) < 0$
$\Rightarrow \forall \mathbf{x}_i \in [\mathbf{x}_i], i = 1,\cdots,4,$
$Intersect(Seg(\mathbf{x}_1,\mathbf{x}_2),Seg(\mathbf{x}_3,\mathbf{x}_4)) < 0$
$\Rightarrow \forall \mathbf{x}_i \in [\mathbf{x}_i], i = 1,\cdots,4,$ $Seg(\mathbf{x}_1,\mathbf{x}_2)$ and $Seg(\mathbf{x}_3,\mathbf{x}_4))$ intersect
$\Rightarrow Seg([\mathbf{x}_1],[\mathbf{x}_2])$ and $Seg([\mathbf{x}_3],[\mathbf{x}_4]))$ intersect.
The same holds for (7). □

In the following $\mathcal{E}^-$ and $\mathcal{E}^+$ are two sets of interval segments defined by:

$$\mathcal{E}^- = \bigcup_{j_1,j_2} Seg([\mathbf{e}^-_{j_1}],[\mathbf{e}^-_{j_2}]), \quad (8)$$

$$\mathcal{E}^+ = \bigcup_{j_1,j_2} Seg([\mathbf{e}^+_{j_1}],[\mathbf{e}^+_{j_2}]). \quad (9)$$

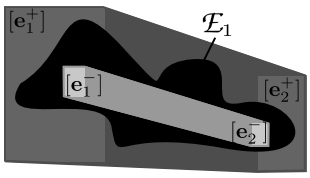

Figure 5: An obstacle $\mathcal{E}_1$ (black shape) known by an inner characterization $Seg([\mathbf{e}_1^-],[\mathbf{e}_2^-])$ (light grey) and an outer characterization $Seg([\mathbf{e}_1^+],[\mathbf{e}_2^+])$ (dark grey).

3 INTERVAL EXTENSION OF THE VISIBILITY

In order to solve the problem with a set-membership approach the visibility and non-visibility definitions have to be extended to the interval analysis context. Definition 9 is an extension of Definitions 1 and 4.

Definition 9. *Let* $[\mathbf{x_1}]$ *and* $[\mathbf{x_2}]$ *be two boxes, and an environment* $\mathcal{E}$

$$([\mathbf{x_1}]\mathrm{V}[\mathbf{x_2}])_{\mathcal{E}} \Leftrightarrow \forall \mathbf{x}_1 \in [\mathbf{x}_1], \forall \mathbf{x}_2 \in [\mathbf{x}_2], (\mathbf{x_1}\mathrm{V}\mathbf{x_2})_{\mathcal{E}}, \quad (10)$$

$$([\mathbf{x_1}]\overline{\mathrm{V}}[\mathbf{x_2}])_{\mathcal{E}} \Leftrightarrow \forall \mathbf{x}_1 \in [\mathbf{x}_1], \forall \mathbf{x}_2 \in [\mathbf{x}_2], (\mathbf{x_1}\overline{\mathrm{V}}\mathbf{x_2})_{\mathcal{E}}. \quad (11)$$

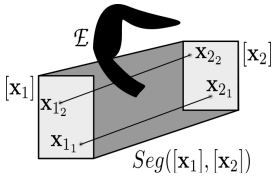

Figure 6: In this example $([\mathbf{x}_1]\overline{\mathrm{V}}[\mathbf{x}_2])_{\mathcal{E}}$ and $([\mathbf{x}_1]\mathrm{V}[\mathbf{x}_2])_{\mathcal{E}}$ are both false (since $(x_{1_1}\mathrm{V}x_{2_1})_{\mathcal{E}}$ and $(x_{1_2}\overline{\mathrm{V}}x_{2_2})_{\mathcal{E}}$).

Remark 10. *Note that* $([\mathbf{x}_1]\mathrm{V}[\mathbf{x}_2])_{\mathcal{E}}$ *and* $([\mathbf{x}_1]\overline{\mathrm{V}}[\mathbf{x}_2])_{\mathcal{E}}$ *can be both false as depicted in Figure 6.*

Lemma 11. *Let* r_1 *and* r_2 *be two robots with their respective positions* $\mathbf{x}_1 \in [\mathbf{x}_1]$ *and* $\mathbf{x}_2 \in [\mathbf{x}_2]$ *and an environment* $\mathcal{E}$ *with an inner approximation* $\mathcal{E}^-$*:*

$$(\mathbf{x}_1\mathrm{V}\mathbf{x}_2)_{\mathcal{E}} \Rightarrow ([\mathbf{x}_1]\overline{\mathrm{V}}[\mathbf{x}_2])_{\mathcal{E}^-} \textit{ is false}. \quad (12)$$

Proof.
$(\mathbf{x}_1\mathrm{V}\mathbf{x}_2)_{\mathcal{E}} \Rightarrow Seg(\mathbf{x}_1,\mathbf{x}_2) \cap \mathcal{E} = \emptyset,$
$\Rightarrow Seg(\mathbf{x}_1,\mathbf{x}_2) \cap \mathcal{E}^- = \emptyset$ since $\mathcal{E}^- \subset \mathcal{E},$
$\Rightarrow (\mathbf{x}_1\mathrm{V}\mathbf{x}_2)_{\mathcal{E}^-},$
$\Rightarrow \exists \mathbf{x}_{i_1} \in [\mathbf{x}_1], \exists \mathbf{x}_{i_2} \in [\mathbf{x}_2] \mid (\mathbf{x}_{i_1}\mathrm{V}\mathbf{x}_{i_2})_{\mathcal{E}^-},$
$\Rightarrow ([\mathbf{x}_1]\overline{\mathrm{V}}[\mathbf{x}_2])_{\mathcal{E}^-}$ is false.
□

Lemma 12. *Let* r_1 *and* r_2 *be two robots with their respective positions* $\mathbf{x}_1 \in [\mathbf{x}_1]$ *and* $\mathbf{x}_2 \in [\mathbf{x}_2]$ *and an environment* $\mathcal{E}$ *with an outer approximation* $\mathcal{E}^+$*:*

$$(\mathbf{x}_1\overline{\mathrm{V}}\mathbf{x}_2)_{\mathcal{E}} \Rightarrow ([\mathbf{x}_1]\mathrm{V}[\mathbf{x}_2])_{\mathcal{E}^+} \textit{ is false}. \quad (13)$$

Proof.
$(\mathbf{x}_1\overline{\mathrm{V}}\mathbf{x}_2)_{\mathcal{E}} \Rightarrow Seg(\mathbf{x}_1,\mathbf{x}_2) \cap \mathcal{E} \neq \emptyset,$
$\Rightarrow Seg(\mathbf{x}_1,\mathbf{x}_2) \cap \mathcal{E}^+ \neq \emptyset$ since $\mathcal{E} \subset \mathcal{E}^+,$
$\Rightarrow (\mathbf{x}_1\overline{\mathrm{V}}\mathbf{x}_2)_{\mathcal{E}^+},$
$\Rightarrow \exists \mathbf{x}_{i_1} \in [\mathbf{x}_1], \exists \mathbf{x}_{i_2} \in [\mathbf{x}_2] \mid (\mathbf{x}_{i_1}\overline{\mathrm{V}}\mathbf{x}_{i_2})_{\mathcal{E}^+},$
$\Rightarrow ([\mathbf{x}_1]\mathrm{V}[\mathbf{x}_2])_{\mathcal{E}^+}$ is false.
□

Lemma 13. *Let* r_1 *and* r_2 *be two robots with their respective positions* $\mathbf{x}_1 \in [\mathbf{x}_1]$ *and* $\mathbf{x}_2 \in [\mathbf{x}_2]$ *and an environment* $\mathcal{E}$ *with an outer approximation* $\mathcal{E}^+$*:*
$([\mathbf{x}_1]\mathrm{V}[\mathbf{x}_2])_{\mathcal{E}^+}$

$$\Rightarrow \forall j, Intersect(Seg([\mathbf{x}_1],[\mathbf{x}_2]),\mathcal{E}^+_j) > 0 \quad (14)$$

Proof.
$([\mathbf{x}_1]\mathrm{V}[\mathbf{x}_2])_{\mathcal{E}^+}$
$\Rightarrow \forall \mathbf{x}_1 \in [\mathbf{x}_1], \forall \mathbf{x}_2 \in [\mathbf{x}_2], (\mathbf{x}_1\mathrm{V}\mathbf{x}_2)_{\mathcal{E}^+},$
$\Rightarrow \forall \mathbf{x}_1 \in [\mathbf{x}_1], \forall \mathbf{x}_2 \in [\mathbf{x}_2], \forall j,$
$Intersect(Seg(\mathbf{x}_1,\mathbf{x}_2),\mathcal{E}^+_j) > 0,$
$\Rightarrow \forall j, Intersect(Seg([\mathbf{x}_1],[\mathbf{x}_2]),\mathcal{E}^+_j) > 0.$
□

Lemma 14. *Let* r_1 *and* r_2 *be two robots with their respective positions* $\mathbf{x}_1 \in [\mathbf{x}_1]$ *and* $\mathbf{x}_2 \in [\mathbf{x}_2]$ *and an environment* $\mathcal{E}$ *with an inner approximation* $\mathcal{E}^-$*:*
$([\mathbf{x}_1]\overline{\mathrm{V}}[\mathbf{x}_2])_{\mathcal{E}^-}$

$$\Rightarrow \exists j \mid Intersect(Seg([\mathbf{x}_1],[\mathbf{x}_2]),\mathcal{E}^-_j) < 0 \quad (15)$$

Proof.
$([\mathbf{x}_1]\overline{\mathrm{V}}[\mathbf{x}_2])_{\mathcal{E}^-}$
$\Rightarrow \forall \mathbf{x}_1 \in [\mathbf{x}_1], \forall \mathbf{x}_2 \in [\mathbf{x}_2], (\mathbf{x}_1 \overline{\mathrm{V}} \mathbf{x}_2)_{\mathcal{E}^-}$,
$\Rightarrow \exists j \mid \forall \mathbf{x}_1 \in [\mathbf{x}_1], \forall \mathbf{x}_2 \in [\mathbf{x}_2]$, $Intersect(Seg(\mathbf{x}_1, \mathbf{x}_2), \mathcal{E}_j^-) < 0$,
$\Rightarrow \exists j \mid Intersect(Seg([\mathbf{x}_1], [\mathbf{x}_2]), \mathcal{E}_j^-) < 0$.

□

From those lemmas it is possible to deduce the following propositions:

Proposition 15. *Let r_1 and r_2 be two robots with their respective positions $\mathbf{x}_1 \in [\mathbf{x}_1]$ and $\mathbf{x}_2 \in [\mathbf{x}_2]$ and an environment $\mathcal{E}$ with an inner approximation $\mathcal{E}^-$:*

$$(\mathbf{x}_1 \mathrm{V} \mathbf{x}_2)_{\mathcal{E}} \Rightarrow \forall j, Intersect(Seg([\mathbf{x}_1], [\mathbf{x}_2]), \mathcal{E}_j^-) \geq 0 \tag{16}$$

Proof.
$(\mathbf{x}_1 \mathrm{V} \mathbf{x}_2)_{\mathcal{E}}$
$\Rightarrow ([\mathbf{x}_1]\overline{\mathrm{V}}[\mathbf{x}_2])_{\mathcal{E}^-}$ is false (Lemma 11),
$\Rightarrow (\exists j \mid Intersect(Seg([\mathbf{x}_1], [\mathbf{x}_2]), \mathcal{E}_j^-) < 0)$ is false (Lemma 14),
$\Rightarrow \nexists j \mid Intersect(Seg([\mathbf{x}_1], [\mathbf{x}_2]), \mathcal{E}_j^-) < 0$.
$\Rightarrow \forall j, Intersect(Seg([\mathbf{x}_1], [\mathbf{x}_2]), \mathcal{E}_j^-) \geq 0$.

□

Proposition 16. *Let r_1 and r_2 be two robots with their respective positions $\mathbf{x}_1 \in [\mathbf{x}_1]$ and $\mathbf{x}_2 \in [\mathbf{x}_2]$ and an environment $\mathcal{E}$ with an outer approximation $\mathcal{E}^+$:*

$$(\mathbf{x}_1 \overline{\mathrm{V}} \mathbf{x}_2)_{\mathcal{E}} \Rightarrow \exists j \mid Intersect(Seg([\mathbf{x}_1], [\mathbf{x}_2]), \mathcal{E}_j^+) \leq 0 \tag{17}$$

Proof.
$(\mathbf{x}_1 \overline{\mathrm{V}} \mathbf{x}_2)_{\mathcal{E}}$
$\Rightarrow ([\mathbf{x}_1]\mathrm{V}[\mathbf{x}_2])_{\mathcal{E}^+}$ is false (Lemma 12),
$\Rightarrow (\forall j, Intersect(Seg([\mathbf{x}_1], [\mathbf{x}_2]), \mathcal{E}_j^+) > 0)$ is false (Lemma 13),
$\Rightarrow \exists j \mid Intersect(Seg([\mathbf{x}_1], [\mathbf{x}_2]), \mathcal{E}_j^+) \leq 0$.

□

4 THE LUVIA ALGORITHM

A set-membership approach considers a bounded error context: all the inputs and variables of the robots are supposed to be in intervals, *e.g.* $\mathbf{u}_i(k) \in [\mathbf{u}_i(k)]$ and $\mathbf{q}_i(k) \in [\mathbf{q}_i(k)]$ with $\mathbf{x}_i(k) \in [\mathbf{x}_i(k)]$.

Each robot $r_i \in \mathcal{R}$ does a measurement vector $\mathbf{y}_i(k)$ at time k.

$$\mathbf{y}_i(k) = (y_{i1}(k), \cdots, y_{ii'}(k), \cdots, y_{im}(k))^T \tag{18}$$

with $\mathbf{y}_{ii'}(k) \in \{\mathbf{true}, \mathbf{false}\}$ the visibility between the robots r_i and $r_{i'}$ at time k. Note that $\mathbf{y}_{ii'}(k) = \mathbf{true}$ means $(\mathbf{x}_i \mathrm{V} \mathbf{x}_{i'})_{\mathcal{E}}$ while $\mathbf{y}_{ii'}(k) = \mathbf{false}$ means $(\mathbf{x}_i \overline{\mathrm{V}} \mathbf{x}_{i'})_{\mathcal{E}}$.

Algorithm 1 computes the robots' pose in a bounded error context and for each robot the LUVIA algorithm (Algorithm 2) contracts the robot's estimated pose to all consistent values according to the environment approximations and the visibility measurements. Note that Algorithm 3 performs a visibility test using the Propositions 15 and 16. This algorithm has three possible return values:

- **true** if $([\mathbf{x}_1]\mathrm{V}[\mathbf{x}_2])_{\mathcal{E}^*}$,
- **false** if $([\mathbf{x}_1]\overline{\mathrm{V}}[\mathbf{x}_2])_{\mathcal{E}^*}$,
- **indeterminate** if no conclusion can be done.

Algorithm 1: The pose tracking algorithm

Data: $\mathcal{R}$, $\mathcal{E}^-$, $\mathcal{E}^+$
$\forall r_i \in \mathcal{R}$ initialize $[\mathbf{q}_i(0)]$;
for $k = 1$ **to end do**
  $\forall r_i \in \mathcal{R}, [\mathbf{q}_i(k)] = f([\mathbf{q}_i(k-1)], [\mathbf{u}_i(k-1)])$;
  $\forall r_i \in \mathcal{R}, \mathbf{y}_i(k) =$ new measurement set;
  $\forall r_i \in \mathcal{R}$, LUVIA$(r_i, \mathbf{y}_i(k), \mathcal{R}, \mathcal{E}^-, \mathcal{E}^+)$;
Result: $\mathcal{R}$.

5 RESULTS AND CONCLUSION

In order to test this pose tracking approach, a simulator has been developed. The simulated environment has a 10×10m size, see Figure 7. At each iteration a robot does a 10cm distance move with a bounded error of ± 0.1cm and a bounded compass error of ± 2.5deg. In the later, the results are obtained for 1500 iterations of the pose tracking algorithm. Note that $\forall r_i \in \mathcal{R}, Size([\mathbf{x}_i(0)]) = 1\mathrm{m}^2$ and $\mathbf{x}_i(0) \in [\mathbf{x}_i(0)]$, with $\mathbf{x}_i(0)$ the initial position of r_i.

The processor used for the simulations has the following characteristics:

Intel(R) Core(TM)2 CPU - 6420 @ 2.13GHz.

5.1 Influence of the number of robots

The objective it to evaluate the effect of the number of robots for the localization. The considered environment is $\mathcal{E} = \bigcup_{j=1}^{6} \mathcal{E}_j$. Figure 8 represents the obtained results.

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, $\mathbf{y}_i(k)$ carries few information. In this configuration, at least 7 robots are necessary to perform an efficient localization.

Algorithm 2: LUVIA algorithm

Data: r_i, $\mathbf{y}_i$, $\mathcal{R}$, $\mathcal{E}^-$, $\mathcal{E}^+$

$\mathcal{L} = \emptyset$, $\mathcal{L}_{OK} = \emptyset$;
$\mathcal{L}.Push_back([\mathbf{x}_i(k)])$;
while $\mathcal{L}.Size() > 0$ **do**
  $[\mathbf{x}] = \mathcal{L}.Pop_out()$;
  $bisect =$ **false**;
  for $r_{i'} \in \mathcal{R}$ *(with* $i \neq i'$*)* **do**
    if $y_{ii'} =$ **true then**
      $consistent = Visibility_Test([\mathbf{x}], [\mathbf{x}_{i'}(k)], \mathcal{E}^-)$;
    else
      $consistent = Visibility_Test([\mathbf{x}], [\mathbf{x}_{i'}(k)], \mathcal{E}^+)$;
      if $consistent =$ **true then**
        $consistent =$ **false**;
      else if $consistent =$ **false then**
        $consistent =$ **true**;
    if $consistent =$ **false then**
      break;
    else if $consistent =$ **indeterminate then**
      $bisect =$ **true**;
  if $consistent \neq$ **false then**
    if $bisect =$ **true and** $Size([\mathbf{x}]) > \varepsilon$ **then**
      $Bisect([\mathbf{x}], \mathcal{L})$;
    else
      $\mathcal{L}_{OK}.Push_back([\mathbf{x}])$;

Result: $Hull(\mathcal{L}_{OK})$.

Algorithm 3: Visibility_test

Data: $[\mathbf{x}_1]$, $[\mathbf{x}_2]$, $\mathcal{E}^*$

$visible =$ **true**;
for $\mathcal{E}^*_k \in \mathcal{E}^*$ **do**
  $inter = Intersect(Seg([\mathbf{x}_1], [\mathbf{x}_2]), \mathcal{E}^*_k)$;
  if $inter < 0$ **then**
    $visible =$ **false**;
    break;
  else if $inter \not> 0$ **then**
    $visible =$ **indeterminate**;

Result: $visible$.

On the other hand, too many robots do not improve significantly the localization but increase the computation time. This localization maximal precision is directly dependent of $\mathcal{E}^-$ and $\mathcal{E}^+$. In this example, over 9 robots the results are similar.

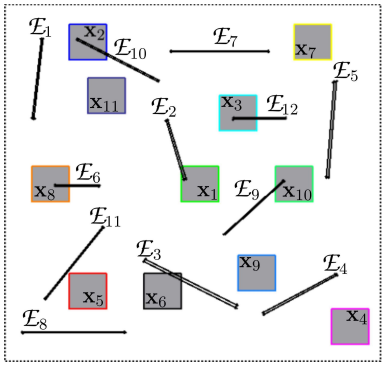

Figure 7: The simulated environment. The black shapes correspond to the environment $\mathcal{E} = \bigcup_{j=1}^{11} \mathcal{E}_j$ and the grey boxes correspond to the initial boxes $[\mathbf{x}_i(0)]$ of the robots such as $\mathbf{x}_i(0) \in [\mathbf{x}_i(0)]$, $\mathbf{x}_i$ the initial position of the robot $r_i \in \mathcal{R}$, $i = 1, \cdots, 11$. Note that for legibility reason $\mathcal{E}^+$ and $\mathcal{E}^-$ are not represented. The doted lines do not represent obstacles but robot's move limits.

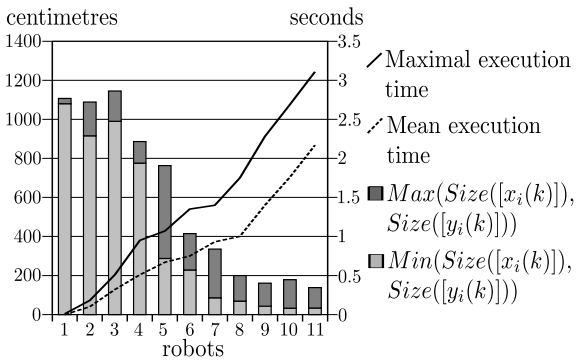

Figure 8: Results over 1500 iterations.

5.2 Influence of the number of obstacles

In this section the influence of the number of obstacles over the localisation results is evaluated. A team of 7 robots is considered $\mathcal{R} = \{r_i\}, i = 1, \cdots, 7$. Figure 9 represents the obtained results.

It appears that for a given number of robots it exists a minimal and a maximal number of obstacles that allow to perform an efficient localization. It can be explained by the fact that without any obstacle, the robots see each other all the time, so the visibility sensor returns always the same value and does not provide useful information. It is the same argument with too many obstacles.

In Figure 9 it is possible to see that under 4 obstacles and over 8 obstacles, the pose tracking does not lead to an efficient localization of the robots. It also appears that 7 obstacles do not lead to an efficient pose tracking. Hence the success of the pose tracking depends on the positions and the sizes of the

obstacles in the environment.

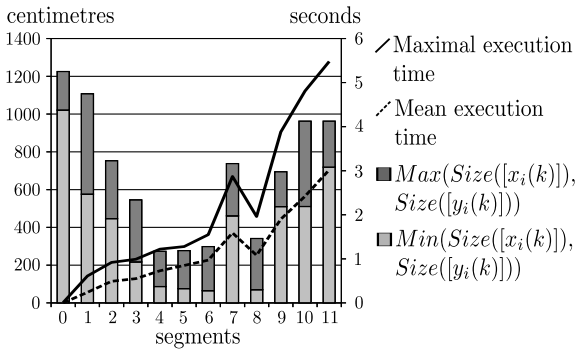

Figure 9: Results over 1500 iterations.

5.3 Conclusion

In this paper it is shown that using interval analysis it is possible to perform a pose tracking of mobile robots even assuming weak informations as the visibility between robots. The LUVIA algorithm is a guaranteed algorithm that exploits this boolean information.

It appears in Section 5.2 that characterizing the environments by counting the number of obstacles is not pertinent here. In a future work it could be interesting to characterize the environment by visibility zones allowing to calculate a minimal number of robots required to perform a pose tracking, according to the number and/or the size of the zones.

Finally it could be interesting to process an experimentation with actual robots.

REFERENCES

Abeles, P. (2011). Robust local localization for indoor environments with uneven floors and inaccurate maps. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 475 – 481.

E. Seignez, M. Kieffer, A. L. E. W. T. M. (2005). Experimental vehicle localization by bounded-error state estimation using interval analysis. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1084 – 1089.

J. Borenstein, H. R. Everett, L. F. (1996). *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd.

J. Zhou, L. H. (2011). Experimental study on sensor fusion to improve real time indoor localization of a mobile robot. In *Robotics, Automation and Mechatronics (RAM), 2011 IEEE Conference on*, pages 258 –263.

Jaulin, L. (2009). A nonlinear set membership approach for the localization and map building of underwater robots. *Robotics, IEEE Transactions on*, 25(1):88 – 98.

K. Lingemann, A. Nchter, J. H. H. S. (2005). High-speed laser localization for mobile robots. *Robotics and Autonomous Systems*, 51(4):275 – 296.

L. Jaulin, M. Kieffer, O. D. E. W. (2001). *Applied Interval Analysis*. Springer.

M.J. Segura, V.A. Mut, H. P. (2009). Mobile robot self-localization system using ir-uwb sensor in indoor environments. In *Robotic and Sensors Environments, 2009. ROSE 2009. IEEE International Workshop on*, pages 29 –34.

Neumaier, A. (1991). *Interval Methods for Systems of Equations (Encyclopaedia of Mathematics and its Applications)*. Cambridge University Press.

P. Jensfelt, H. C. (2001). Pose tracking using laser scanning and minimalistic environmental models. *Robotics and Automation, IEEE Transactions on*, 17(2):138 –147.

R. E. Moore, R. B. Kearfott, M. J. C. (2009). *Introduction to Interval Analysis*. SIAM.

APPENDIX

Segment Intersection

The function $Intersect()$, Definition 17, allows to test the intersection between two segments.

Definition 17. *Let $Seg(\mathbf{x}_1,\mathbf{x}_2)$ and $Seg(\mathbf{x}_3,\mathbf{x}_4)$ be two segments, the function*

$$Intersect(Seg(\mathbf{x}_1,\mathbf{x}_2),Seg(\mathbf{x}_3,\mathbf{x}_4)) \qquad (19)$$

is defined by

$Intersect(Seg(\mathbf{x}_1,\mathbf{x}_2),Seg(\mathbf{x}_3,\mathbf{x}_4)) = Max(Side(\mathbf{x}_1,Seg(\mathbf{x}_3,\mathbf{x}_4)) \cdot Side(\mathbf{x}_2,Seg(\mathbf{x}_3,\mathbf{x}_4)), Side(\mathbf{x}_3,Seg(\mathbf{x}_1,\mathbf{x}_2)) \cdot Side(\mathbf{x}_4,Seg(\mathbf{x}_1,\mathbf{x}_2)))$.

where $Side()$, Definition 18, allows to test the side of a point with a segment.

Definition 18. *Let $Seg(\mathbf{x}_1,\mathbf{x}_2)$ be a segment and $\mathbf{x}_3$ be a point, the function $Side(\mathbf{x}_3,Seg(\mathbf{x}_1,\mathbf{x}_2))$ is defined by*

$$Side(\mathbf{x}_3,Seg(\mathbf{x}_1,\mathbf{x}_2)) = \det(\mathbf{x}_3-\mathbf{x}_1\ \mathbf{x}_2-\mathbf{x}_1), \qquad (20)$$

with det *the determinant.*

Figure 10 represents three intersection tests.

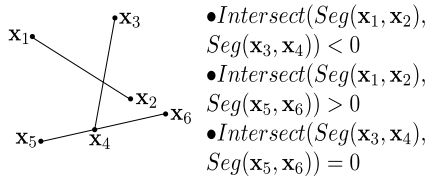

Figure 10: Three $Intersect()$ tests.