



**HAL**  
open science

## ”Localization Space”: a framework for localization and planning, for systems using a Sensor/Landmarks module

Cédric Pradalier, Sepanta Sekhavat

### ► To cite this version:

Cédric Pradalier, Sepanta Sekhavat. ”Localization Space”: a framework for localization and planning, for systems using a Sensor/Landmarks module. [Research Report] RR-4385, INRIA. 2002. inria-00072203

**HAL Id: inria-00072203**

**<https://inria.hal.science/inria-00072203v1>**

Submitted on 23 May 2006

**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.

***“Localization Space” : a framework for localization  
and planning, for systems using a Sensor/Landmarks  
module***

Cédric Pradalier — Sepanta Sekhavat

**N° 4385**

Février 2002

THÈME 3



***Rapport  
de recherche***



## “Localization Space” : a framework for localization and planning, for systems using a Sensor/Landmarks module

Cédric Pradalier , Sepanta Sekhavat \*

Thème 3 — Interaction homme-machine,  
images, données, connaissances  
Projet Sharp

Rapport de recherche n° 4385 — Février 2002 — 18 pages

**Abstract:** One of the common ways of localization in robotics is the triangulation using a system composed of a sensor and some landmarks (which can be artificial or natural). This paper presents a framework, namely the Localization Space, in order to deal with problems such as the landmark placement and motion planning including the localization constraint. Based on this framework, we present general approaches to the optimal distribution of the landmarks or to the computation of reliable trajectories. The case of a mobile robot equipped with an orientable sensor (such as a pan vision system) is the recurrent example of the paper, meant to illustrate the formal concepts and to also show the practical relevance of the proposed tools.

**Key-words:** robotic localization, sensor oriented path planning, landmarks placement.

\* {pradalie,sekhavat}@inrialpes.fr

## **“Espace de localisation” : un outils de travail générique pour la localisation et la planification, pour un système capteur/balises.**

**Résumé :** Une des techniques classiques de localisation en robotique est la triangulation utilisant un système composé d'un capteur et d'un ensemble d'amers (artificiels ou naturels). Cet article présente la notion d'espace de localisation, notion qui permet de traiter des problèmes tels que le placement optimal des amers ou la planification sous contraintes de localisation.

Grâce à cet outils, nous présentons une approche générale pour la répartition optimale des amers et le calcul de trajectoire fiables. Le cas d'un robot mobile équipé d'un capteur orientable servira d'exemple tout au long de cet article, de façon à illustrer les concepts formels et à montrer l'intérêt pratique de la méthode.

**Mots-clés :** Localisation, robotique, planification sous contraintes de localisation, placement d'amers

## 1 Introduction

When moving a robot, the classical way to deal with the modeling errors and the execution errors is to equip the robot with some localization ability so it can compute its error with respect to the nominal path and correct it through a closed loop. A common way of localization in robotics is the triangulation using a system composed of a sensor and some landmarks or beacons (which can be artificial or natural). Depending on the sensor type and accuracy and the robot dimension, different number of landmarks may be needed to have a reliable localization. In general, the subspace of the configuration space in which the localization is possible depends on the placement of the landmarks. To our knowledge, very few work deals with the problem of landmarks placement. In other respects, guaranteeing the maximum localization ability along a trajectory in order to have a safe execution is an important additional constraint for the motion planner (a.k.a. the sensor-oriented motion planning). In this paper we present a framework which leads to some general approaches to the two pre-cited problems : namely, the optimal distribution of the beacons and the computation of the reliable trajectories.

However, let us first have an overview of some of the existing works in the literature on the problems related to landmarks and the associated planning. Concerning the sensor-oriented planning and the safe execution of the paths, some works such as [7] and [8] suggest methods to determine which of the visible landmarks should be used to improve the localization results. In [1], the authors consider a mobile robot following a path and they compute the velocity profile and the sensor motion in order to optimize the path following while guaranteeing the visibility of the landmarks. Finally some more general works such as [6] or [4] consider the problem of uncertainty propagation along a path and suggest different ways of keeping this uncertainty bounded by leading the path through some areas where relocalization is possible. In our work, we aim at the practical computation of these areas for a Sensor/Landmarks system. We are also interested in applications where the safe zone is big enough so that most of the problems can be solved without leaving the safe zone. We aim at giving a general scheme for safe motion planning in such cases.

Concerning the optimal placement of the landmarks there has been so far very few works in the literature. [10] presents a placement method maximizing the localization accuracy. However, this work concerns only the sensors where one beacon is enough for localization. The estimation of the accuracy in [10] is also very specific to the type of considered beacon. Our goal is more to optimize the size of the safe area, than the localization accuracy. Also, we consider landmark optimal placement for localization systems requiring any number  $l$  of landmarks. This problem is closer to the illumination problem presented in [3]. However, [3] considers only quadrilateral obstacles and can be applied only to sensors with an unlimited range and angular opening. In this paper, we consider a general sensor with a limited range and angular opening.

Our work is based on the notion of *Localization Space* which will be defined and build in section 2. We will show in section 3 how to compute a discretized version of this continuous set which would be conservative for localization. Then, in section 4, we present a practical procedure of optimal landmark placement using the localization space. Finally,

section 5 deals with the relevance of the localization space for path planning and includes some simulation results of a planner for mobile robots computing reliable paths.

## 2 Localization Space : definition and construction

We consider the case of a generic robot with  $n$  degrees of freedom, moving in a polygonal environment with some obstacles. We assume that the robot is equipped with a sensor characterized by a range  $p$  and an angular opening  $\alpha$ . To the physical sensor, we associate a virtual (punctual) sensor  $(x, y, \theta)$ , where  $(x, y)$  is the absolute position of the sensor reference point and  $\theta$  its absolute (main) orientation. The physical aspects of the sensor (such as the relative motion to the robot, collision avoidance, etc) will be treated by considering the physical sensor solid, as a part of the articulated robot.

In other respects, we assume that the environment includes some landmarks such that if a sufficient number of them is within the sensor field of perception, the localization module is able to compute the pose  $(x, y, \theta)$  of the sensor. This last assumption supposes that the problem of detecting and identifying the landmarks have already been solved and that the positions of the landmarks are precisely known.

### 2.1 Symbols and notation

Let us note :

- $\mathcal{W}$  : Polygonal workspace of the robot (free space).
- $(x, y, \theta)$  : Sensor pose.
- $(x, y)$  : Sensor position in  $\mathcal{W}$ .
- $\theta$  : Sensor orientation.
- $\mathcal{P}$  : Space of the free poses of the sensor = space of poses such that  $(x, y)$  is outside the obstacles.
- $\mathcal{CS}_{free}$  : Space of the free configurations of the robot.
- $q$  : robot configuration in  $\mathcal{CS}_{free}$ .

In the following, we first define the **Sensor Localization Space**  $\mathcal{L}$  as the subset of  $\mathcal{P}$  where the sensor pose can be computed. For instance, with a Sick laser range-finder, we need to identify at least two landmarks in order to compute the robot pose, thus  $\mathcal{L}$  will be the subset of  $\mathcal{P}$  where at least two landmarks are visible by the sensor.

Then we define the **Localization Space**  $\mathcal{L}$  as the set of configurations  $q$  in  $\mathcal{CS}_{free}$  for which the resulting sensor pose is in  $\mathcal{L}$ . Note that, due to the collision of the physical robot with the obstacles, it may exist sensor pose in  $\mathcal{L}$  which do not correspond to any configuration in  $\mathcal{L}$ . Let us now describe the different steps to obtain  $\mathcal{L}$ .

## 2.2 Sensor field of view $\mathcal{S}_\theta^{p,\alpha}$

Given  $\theta$ ,  $p$  and  $\alpha$ , we define the orientation vector :

$$\vec{d}_\theta = (\cos(\theta), \sin(\theta))^T$$

and the field of view :

$$\mathcal{S}_\theta^{p,\alpha} = \{\vec{v} / \|\vec{v}\| \leq p \text{ and } |(\vec{d}_\theta, \vec{v})| \leq \alpha/2\} \quad (1)$$

In order to simplify the notation, we will use  $\mathcal{S}_\theta$  for  $\mathcal{S}_\theta^{p,\alpha}$ . Then, for a given position  $R(x, y)$  and given orientation  $\theta$  of the sensor, the set  $E \subset \mathcal{W}$  of the points in the sensitive field of the sensor is :

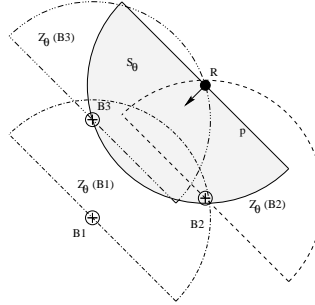
$$E = R \oplus \mathcal{S}_\theta = \{R + \vec{v} / \vec{v} \in \mathcal{S}_\theta\} \quad (2)$$

## 2.3 Visibility area $Z_\theta(B)$

For a given sensor orientation  $\theta$ , we define the visibility area  $Z_\theta(B)$  of a landmark  $B$ , as the set of the sensor positions where  $B$  is in the sensor field of view (see Fig. 1). We have :

$$Z_\theta(B) = B \ominus \mathcal{S}_\theta = \{B - \vec{v} / \vec{v} \in \mathcal{S}_\theta\} \quad (3)$$

Figure 1: Sensor field of view (in gray) and visibility area of three landmarks



## 2.4 Visibility polygon $P(B)$

The visibility area does not take into account the environment and its obstacles. To deal with this problem, we use the **visibility polygon** notion : the visibility polygon  $P(B)$  of a landmark  $B$  is the set of points  $P \in \mathcal{W}$  such that  $[PB] \subset \mathcal{W}$ . Finding the visibility polygon of a point is a complex but feasible problem (see [2] and [5]).



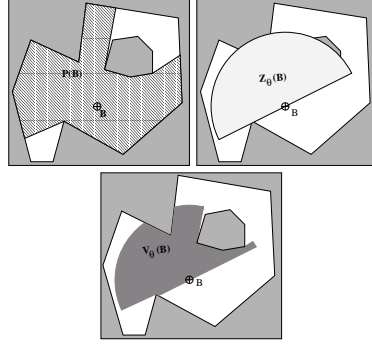
## 2.5 Visibility $\mathcal{V}_\theta(B)$

Using the visibility area (for the constraints imposed by the sensor) and the visibility polygon (for the constraints imposed by the environment) we build the visibility  $\mathcal{V}_\theta(B)$  of a landmark  $B$ , for a sensor orientation  $\theta$ .  $\mathcal{V}_\theta(B)$  is the set of positions where the landmark  $B$  can be perceived despite the presence of obstacles :

$$\mathcal{V}_\theta(B) = Z_\theta(B) \cap P(B) \quad (4)$$

Fig.2 illustrates this construction.

Figure 2: Construction of the visibility of a landmark



## 2.6 Localization layer $L_\theta$

Let us assume that  $n$  landmarks  $\{B_i\}_{i=1..n}$  are distributed in  $\mathcal{W}$  and that we need to identify at least  $l$  landmarks to be able to localize the sensor. Then, given a sensor orientation  $\theta$ , we define the localization layer  $L_\theta$  as the subset of  $\mathcal{W}$  where  $l$  landmark(s) are in the sensitive field of the sensor. Formally :

$$L_\theta = \{(x, y) \in \mathcal{W} / \text{card}(\{i / (x, y) \in \mathcal{V}_\theta(B_i)\}) \geq l\} \quad (5)$$

We will use the notation  $\mathcal{L}_\theta$  to define the corresponding subset of  $\mathcal{P}$  :

$$\mathcal{L}_\theta = L_\theta \times \{\theta\}$$

## 2.7 Sensor localization space $\mathcal{L}$

Now, we can easily express the whole localization space, i.e. the subset of  $\mathcal{P}$  where sufficient landmarks are visible to compute the pose of the sensor.

$$\mathcal{L} = \bigcup_{\theta=0}^{2\pi} \mathcal{L}_\theta \quad (6)$$

## 2.8 Localization space $\mathbb{L}$

For each pose  $p$  in  $\mathcal{L}$ , there exists a set of possible configurations in  $\mathcal{CS}_{free}$  for which the resulting sensor pose is  $p$ . Calling  $K$  the function mapping a configuration  $q$  to the resulting sensor pose, we have :

$$\mathbb{L} = \{q \in \mathcal{CS}_{free} / K(q) \in \mathcal{L}\} \quad (7)$$

Note that due to the collisions, there may exist poses in  $\mathcal{L}$  which are not the image by  $K$  of a configuration in  $\mathcal{CS}_{free}$ .

## 3 The practical computation of $\mathcal{L}$

The computation of  $\mathcal{L}$  requires some approximations that will be presented and justified in this section.

### 3.1 Approximations

- First, the localization space is defined as a union of layers over the continuous set  $\Theta = [0..2\pi]$ . We thus have to chose a discretization step  $\theta_{step}$  and replace  $\Theta$  by

$$\tilde{\Theta} = \{k \theta_{step}\}_{k \in \{0,1,\dots,[2\pi/\theta_{step}]\}}$$

Then, we assume that the localization layer is constant for

$$\theta \in [\theta - \theta_{step}/2 \dots \theta + \theta_{step}/2]$$

and we get

$$\mathcal{L} \approx \tilde{\mathcal{L}} = \bigcup_{\theta \in \tilde{\Theta}} L_{\theta} \times [\theta - \frac{\theta_{step}}{2}, \theta + \frac{\theta_{step}}{2}] \quad (8)$$

- Second, for each  $\theta \in \tilde{\Theta}$ , we build the localization layer  $L_{\theta}$  thanks to the partitioning of  $\mathcal{W}$  according to the visible landmarks. Fig. 3 represents the algorithm that achieves this partitioning .The result is a set of region  $R_i^{\theta}$  of  $\mathcal{W}$  associated with the set  $V_i^{\theta}$  of landmarks visible from  $R_i^{\theta}$  by a sensor with the orientation  $\theta$ .

In order to simplify the computation of the intersections and unions, we choose to approximate the bordering arcs of the visibility area, with regular polygons inscribed in the visibility area. Obviously, the more edges for these polygons, the closer the polygons to the real arcs. The main consequence of this approximation is a reduction of the localization space : for each landmark and for each orientation, the approximated visibility is inside the real visibility, thus, it exists some (small) subset of  $\mathcal{P}$  where localization is said to be impossible though in reality, it is possible.

Note that the algorithm of Fig.3 leads not only to a construction of  $\mathcal{L}$  but also to a partitionning of  $\mathcal{L}$  with respect to the list of the visible landmarks.

Figure 3: Partitioning  $\mathcal{W}$ 

```

Given  $n$  landmarks, we note
 $\mathcal{P}_k^\theta = \{(R_{k,i}^\theta, V_{k,i})\}_{i=1\dots n_k}$  the partition
state at step  $k \in [1..n]$ . At start, no
landmarks are visible in  $\mathcal{W}$ .

1  $\mathcal{P}_0^\theta \leftarrow \{(\mathcal{W}, \emptyset)\}$ ,  $n_0 = 0$ 
2 for each landmark  $k \in 1, \dots, n$ ,
   /* Refining of the partition */
3  $\mathcal{P}_k^\theta \leftarrow$ 
4    $\{(\mathcal{R}_{k-1,i}^\theta \setminus \mathcal{V}_\theta(B_k), V_{k-1,i})/i = 1 \dots n_{k-1}\}$ 
5    $\cup$ 
6    $\{(\mathcal{R}_{k-1,i}^\theta \cap \mathcal{V}_\theta(B_k), V_{k-1,i} \cup \{B_k\})/i = 1 \dots n_{k-1}\}$ 
   /* Removing the empty regions */
7  $\mathcal{P}_k^\theta \leftarrow \mathcal{P}_k^\theta \setminus \{(R, V)/R = \emptyset\}$ ,  $n_k = \text{card}(\mathcal{P}_k^\theta)$ 

```

### 3.2 Validity of the approximation

We will prove here that  $\tilde{\mathcal{L}}^{\alpha-\theta_{step}, p} \subset \mathcal{L}^{\alpha, p}$ . Let's choose

$$\theta \in \tilde{\Theta}$$

$$\text{and } t \in [-\theta_{step}/2, +\theta_{step}/2]$$

- For  $\vec{v} \in \mathcal{S}_\theta^{\alpha-\theta_{step}, p}$ , we have

$$\|\vec{v}\| \leq p \text{ and } |(\vec{v}, \vec{d}_\theta)| \leq \frac{\alpha - \theta_{step}}{2}$$

By using the complex notation, we get  $v = r e^{i(\theta+\beta)}$ , with

$$-\frac{\alpha - \theta_{step}}{2} \leq \beta \leq +\frac{\alpha - \theta_{step}}{2}$$

$$-\frac{\alpha - \theta_{step}}{2} - t \leq \beta - t \leq +\frac{\alpha - \theta_{step}}{2} - t$$

$$-\frac{\alpha - \theta_{step} + \theta_{step}}{2} \leq \beta - t \leq +\frac{\alpha - \theta_{step} + \theta_{step}}{2}$$

Therefore  $v = r e^{i((\theta+t)+(\beta-t))}$  also belongs to  $\mathcal{S}_{\theta+t}^{\alpha, p}$ .

- For each  $\theta$  and  $t$ ,

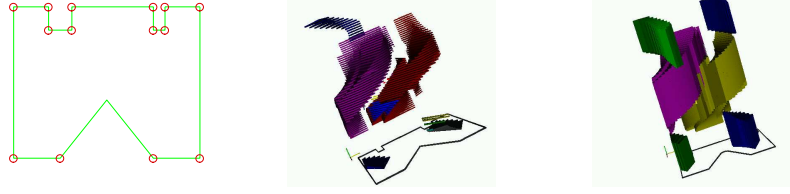
$$\mathcal{S}_\theta^{\alpha-\theta_{step}, p} \subset \mathcal{S}_{\theta+t}^{\alpha, p}$$

Thus, for each landmark  $B$ ,

$$Z_\theta^{\alpha-\theta_{step}, p}(B) \subset Z_{\theta+t}^{\alpha, p}(B)$$

And, as  $\mathcal{W}$  does not depend on  $\theta$ ,

$$\mathcal{V}_\theta^{\alpha-\theta_{step}, p}(B) \subset \mathcal{V}_{\theta+t}^{\alpha, p}(B)$$

Figure 4:  $\tilde{\mathcal{L}}$  (left),  $\tilde{\mathcal{L}}$  3D projection (right) and the corresponding workspace

From this, we deduce

$$L_{\theta}^{\alpha - \theta_{step}, p} \subset L_{\theta+t}^{\alpha, p}$$

And finally, since the inclusion remains over  $t \in [-\theta_{step}/2, +\theta_{step}/2]$ ,

$$\tilde{\mathcal{L}}^{\alpha - \theta_{step}, p} \subset \mathcal{L}^{\alpha, p} \quad (9)$$

This result can be expressed as the following proposition :

**Proposition 1** *With a sensor of angular opening  $\alpha$ , the approximated localization space should be built for an angular opening of  $\alpha - \theta_{step}$  in order to have a conservative approximation for localization.*

In other words, by this way, we will be certain that the approximated localization space we get is included in the real one.

### 3.3 Example

Fig.4 shows an example of a representation of a given localization space. We used :  $l = 2$ ,  $\theta_{step} = 5^\circ$ ,  $\alpha = 60^\circ$  and a range approximatively equal to 75% of the workspace width.

- In the workspace  $\mathcal{W}$ , the small circles represent the landmarks.
- In the localization space  $\tilde{\mathcal{L}}$ , the vertical axis represents  $\theta$  (discretization is done orthogonally to this direction). The coloring is made according to the connectivity of  $\tilde{\mathcal{L}}$  (see section 4.1).

In general, the practical computation of  $\mathcal{L}$  based on the approximated  $\tilde{\mathcal{L}}$  requires some more works which may be more or less complex, according to the robot geometry and dimension. Let us here consider the case of a circular robot  $(x_r, y_r, \theta_r)$  equipped with a mobile sensor with an opening  $\alpha$  able to rotate with respect to the robot by a relative angle  $|\beta| < \beta_{lim}$ . The robot/sensor system  $(x_r, y_r, \theta_r, \beta)$  is of dimension 4. If  $(x, y) = (x_r, y_r)$  then  $\mathcal{L}$  for this system is the one of Fig.4 and the projection of  $\mathcal{L}$  on  $(x_r, y_r, \theta_r)$  can be easily deduced from  $\tilde{\mathcal{L}}$  by :

- deducing the  $(x_r, y_r)$  free space  $\mathcal{W}_r$  from  $\mathcal{W}$  by computing the Minkowski sum of the robot circle and  $\mathcal{W}$ ,
- substituting each layer of  $\mathcal{L}$  by its intersection with  $\mathcal{W}_r \times [\theta - \theta_{step}/2, \theta + \theta_{step}/2]$ ,
- vertically expanding each layer of  $\mathcal{L}$  with  $\beta_{lim}$  at the top and the bottom,

Notice that the first 2 steps are not equivalent with computing  $\mathcal{L}$  with  $\mathcal{W}_r$  instead of  $\mathcal{W}$ . Indeed such a substitution will cause an unjustified reduction of the localization space due to the fact that when computing the visibility polygon at one point, the other positions of the robot along  $\mathcal{W}$  border will be considered as obstacles preventing the landmark visibility (which is of course false). Fig.4 represents the 3D projection of the approximated  $\mathcal{L}$  for our mobile robot. Notice that  $\mathcal{L}$  and  $\mathcal{L}$  may not have the same connectivity (we will see its influence on motion planning in section 5).

## 4 Optimization of the localization space

Let us recall that we consider the case of a localization module using landmarks. The performances of this localization obviously depends on the “good” placement of the landmarks in the environment. As far as we know, the problem of finding an optimal placement for a minimal number of landmarks is an open problem in the general case. We believe that  $\mathcal{L}$  is an interesting tool to evaluate a given landmark placement. A better space would be  $\mathcal{L}$  but the construction of  $\mathcal{L}$  may be complex. We believe that  $\mathcal{L}$  gives us a first estimation, easier to compute.

In this section, we discuss about different optimization criteria (expressed with respect to  $\mathcal{L}$ ) for the computation of an optimal placement of the landmarks. However, we first introduce some relevant measures of  $\mathcal{L}$  that will be later used in our cost function. The results of our numerical optimization are presented at the end of the section.

### 4.1 Measures

*Volume* : The proportion of  $\mathcal{P}$  which is occupied by the localization space is obviously a good indication on the quality of the landmark distribution. We use the following classical formula to compute  $\mathcal{L}$  volume :

$$\text{Volume}(\mathcal{L}) \approx \sum_{\theta \in \tilde{\Theta}} \text{Area}(L_\theta) \cdot \theta_{step} \quad (10)$$

*Connectivity* : The connectivity of the localization space is very important for motion planning. Indeed, as long as a trajectory remains in the same connected component of the localization space, it is theoretically always possible to relocalize the system and therefore correct the execution errors and prevent the growth of the uncertainty on the position. The best scenario is a single connected component for  $\mathcal{L}$  covering  $\mathcal{W}$ .

Practically, for a given distribution of the landmarks, we study the connectivity of  $\mathcal{L}$  by building a graph  $\mathcal{G}(\mathcal{L})$  that we will later use for motion planning. The nodes of this graph are the regions  $R_i^\theta \times [\theta - \theta_{step}/2, \theta + \theta_{step}/2]$  for which  $card(V_i^\theta) > \ell$  where  $\ell$  is the minimum number of landmark required for localization (see section 3.1). We have seen that  $\mathcal{L}$  is the union of such regions. Then, we consider that two nodes are connected if they are adjacent. At this point, a simple exploration algorithm gives the separation of the different connected parts of the graph. One can proof that :

**Proposition 2** *for a sensor with an angular opening  $\alpha$ , if the approximated localization space is built with an angular opening less than  $\alpha - 2\theta_{step}$ , all the edges of the graph correspond to a path that is completely included in  $\mathcal{L}$ , despite the approximation.*

$$\mathcal{G}(\tilde{\mathcal{L}}^{\alpha - 2\theta_{step}, p}) \subset \mathcal{G}(\mathcal{L}^{\alpha, p}) \quad (11)$$

## 4.2 Optimization of the landmark placement using $\mathcal{L}$

The following list of criteria using  $\mathcal{L}$  gives some quantities which can be used in a cost function for the landmark placement optimization :

- The connectivity : a connected localization space will lead to an easier planning and to a more reliable execution of the resulting trajectory.
- The volume : if a big proportion of  $\mathcal{P}$  is covered by  $\mathcal{L}$ , the robot will have more opportunities to localize.
- The secure area : i.e. the subset of  $\mathcal{W}$  where localization is possible for any orientation. This is particularly important when dealing with a nonholonomic system. For instance, the classical mobile robots are small-time locally controllable which implies that any path in the localization space can be approximated within this space, by a feasible path for the nonholonomic system. However, (specially with a fixed sensor), the complexity of this approximation is strongly influenced by the space available for the change of orientation...
- The projected area : i.e. the subset of  $\mathcal{W}$  where localization is possible for at least one orientation. Typically, when a  $360^\circ$ -orientable sensor is used, it is sufficient to maximize this area. Another use of this space is the case of a holonomic mobile robot applied to problems where the final orientation is not important.
- The localization accuracy : it may be important to place the landmarks such that an upper bound of the localization error could be guaranteed for a real sensor. For instance, in the case of triangulation, the influence of measurement errors on the localization is minimal when the angle between the sensor and the two visible landmarks is close to 90 degrees.
- ...

Moreover, it will be necessary to decide whether the landmarks are accepted inside  $\mathcal{W}$  or only on its border. In the first case, we may have to consider them as obstacles during the navigation. Nevertheless, as the landmarks will be visible from any direction, each landmark will allow more localization possibilities.

### 4.3 Example

Let us get back to our mobile robot with an on-board orientable sensor. In this framework, we try to use as few artificial landmarks as possible.

#### 4.3.1 Cost function

For a given number of landmarks, we want to be able to localize in a large portion of the space and to have the least possible number of separate localization subspace. We decided therefore to maximize the average volume of the separate localization subspace. In other respects, in order to improve the localization accuracy, we impose a minimum distance between two landmarks. In this way, we increase the average angle between the sensor and a pair of landmarks used for localization. Indeed, with a small angle, measurement errors leads to larger localization errors. These constraints lead to the following cost function  $\mathcal{C}$  :

$$\begin{aligned} d_{\min} &= \text{smallest dist. between 2 landmarks} \\ \mathcal{K} &= \begin{cases} (d_{\min}/d_{\text{lim}})^2 & \text{if } d_{\min} \leq d_{\text{lim}} \\ 1 & \text{otherwise} \end{cases} \\ \mathcal{C} &= \mathcal{K} \frac{\text{Volume}(\mathcal{L})}{\text{Number of connected parts in } \mathcal{L}} \end{aligned}$$

#### 4.3.2 Results

We use this cost function with a library implementing genetic algorithms. In the both following examples, we have  $\alpha = 90^\circ$  and the robot needs at least two landmarks to compute its configuration. We ran the optimization process for 2 to 20 landmarks placed either inside the environment, or on its border only and we compared the obtained occupied space (see Fig.6 and Fig.7).

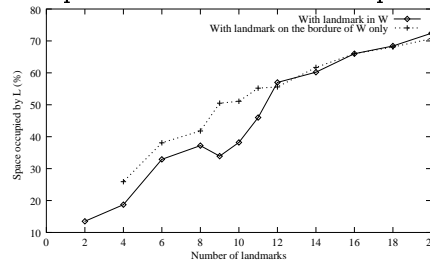
- In the first example, the workspace has a U-shape (see Fig.5) and the sensor has an infinite range compared to the environment size. The results on Fig.6 show that

Figure 5: U-shape workspace and placement of 11 landmarks.



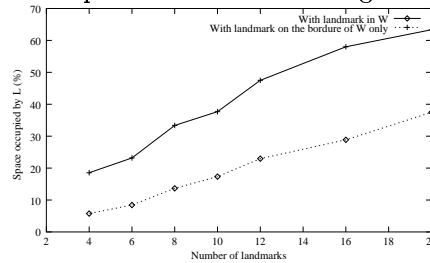
when the sensor range is large enough, there is no need to place landmarks inside the environment, except if we have very few of them. Indeed, landmarks inside the environment have maximum visibility. However, if the sensor range is large enough, an edge of the environment is visible from any configuration. Fig.5 shows the placement obtained for 11 landmarks with and without forbidding inside the space.

Figure 6: Optimization results : U-shape workspace



- In the second example, we ran the optimization process in a large hexagonal environment, with a small sensor range. In this case, we can observe that landmarks should be allowed inside the environment (see Fig.7) since those on the border can only be used locally.

Figure 7: Optimization results : large workspace



## 5 Planning in the localization space

Now that we know how to optimize the landmark placement, let us see how to use the resulting localization space for motion planning. Indeed, planning in the  $\mathcal{L} \subset \mathcal{CS}_{free}$  is more interesting than planning in  $\mathcal{CS}_{free}$ . Along a path contained in  $\mathcal{L}$  the robot can measure the execution errors by relocalizing and therefore correct them. Such a path is thus more reliable. Moreover, at each point, we know the exact subset of visible landmarks and their positions relative to the robot. This will ease the data association process.



## 5.1 Definitions

For a robot with  $n$  degrees of freedom, a configuration  $q$  is defined by  $n$  parameters  $q = (x_1, \dots, x_n)$  which we divide in two (possibly empty) classes :

**Extern parameters** : parameters for which exteroceptive sensors are required in order to estimate their values with a bounded uncertainty. For instance, the position parameters of a mobile robot.

**Intern parameters** : parameters whose estimation with proprioceptive sensors is accurate enough for our application. For instance, the orientations of a pan & tilt camera. Note that when intern parameters exists, their movement do not depend on the others (otherwise, they would need exteroceptive localization). Consequently, it is possible to move them while maintaining the extern parameters fixed.

**Reliable trajectory** : a trajectory is reliable if the robot can estimate all its parameters at each point of it. Typically, if the trajectory is totally contained in  $\mathcal{L}$ , it is reliable. This is a sufficient condition but not a necessary one. Actually, as intern parameters can be estimated without exteroceptive sensing, a trajectory is reliable if it is admissible ( $\subset \mathcal{CS}_{free}$  and kinematically feasible) and if the extern parameters are fixed for the portions of the path outside  $\mathcal{L}$ .

## 5.2 Algorithm

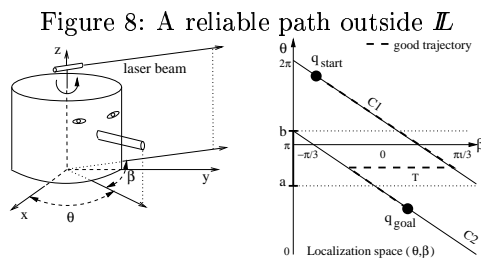
Before all, we have to define two projections :  $P_{ex}(x_1, \dots, x_n) = (x_1, \dots, x_p)$ , on the extern parameters and  $P_{in}(x_1, \dots, x_n) = (x_{p+1}, \dots, x_p)$ , on the intern parameters. Let us consider the planning of a reliable trajectory between  $q_{start}$  and  $q_{goal}$ .

- If  $q_{start}$  and  $q_{goal}$  are in the same connected part of  $\mathcal{L}$ , and if the kinematic constraints of the robot allow it (which is for instance always the case for holonomic and small-time controllable nonholonomic robots), there exists a reliable trajectory between  $q_{start}$  and  $q_{goal}$ , we “just” have to find it.
- Otherwise, if the robot does not have intern parameters, no reliable trajectory can be found.
- If the robot has intern parameters, we will see how to attempt reliable trajectory planning between the disconnected parts of  $\mathcal{L}$ .

*Linking disconnected parts* : Let us consider  $C_1$  and  $C_2$ , two connected parts of  $\mathcal{L}$  and let us try to find a reliable trajectory  $\mathcal{T}$  going from  $C_1$  to  $C_2$ . We have seen that a reliable trajectory must keep extern parameters fixed outside  $\mathcal{L}$ , thus  $P_{ex}(\mathcal{T})$  must be a point inside  $P_{ex}(C_1) \cap P_{ex}(C_2)$ . Typically if this intersection is empty, there is no way of finding a reliable path. Now assume that there is  $q \in P_{ex}(C_1) \cap P_{ex}(C_2)$  with  $q = P_{ex}(q_1) = P_{ex}(q_2)$  such that there is feasible path  $\mathcal{T} \subset q^\perp$  going from  $q_1$  to  $q_2$ . Then  $\mathcal{T}$  is a reliable path. For any

$q_{start} \in C_1$  and  $q_{goal} \in C_2$  there is a reliable path going from  $q_{start}$  to  $q_1$  within  $C_1$ , then following  $\mathcal{T}$  from  $q_1$  to  $q_2$  outside  $\mathcal{L}$  and finally going from  $q_2$  to  $q_{goal}$  within  $C_2$ .

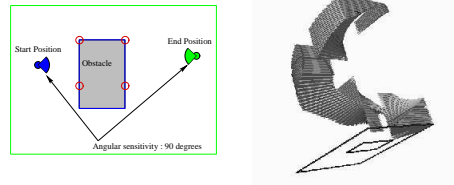
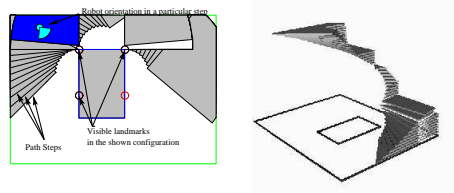
Let us illustrate the above construction through an example with our experimental platform. For sake of simplicity consider that the position of our robot is fixed and that the sensor is an orientable laser pointer, see Fig.8. Thence we consider only two system parameters : the robot orientation  $\theta$  (considered as an extern parameter) and the relative orientation of the laser :  $\beta \in [-\pi/4, \pi/4]$  (accurate enough to be considered as an intern parameter). We assume that the robot is between two reflexive landmarks  $L_1$  and  $L_2$  such that the angle formed by the robot and them is between  $\frac{\pi}{4}$  and  $\frac{\pi}{2}$ . Therefore, there is an interval  $\theta \in [a, b]$  where the laser can aim at  $L_1$  and  $L_2$ . Now if we consider that  $q_{start}$  is facing  $L_1$  and  $q_{goal}$  is facing  $L_2$ , these configurations are obviously in two separate parts of  $\mathcal{L}$ . Indeed in order to face  $L_2$  the robot has to turn for more than  $\frac{\pi}{4}$ , therefore the laser has to leave  $L_1$  at some point and aim later at  $L_2$ . During this phase, the robot cannot localize. However, there is one reliable way of going from  $q_{start}$  to  $q_{goal}$  (see Fig.8) : we can turn the robot towards  $L_2$  until  $\theta \in [a, b]$  with the laser always pointing to  $L_1$ . Then stop the robot and reorient the laser from  $L_1$  to  $L_2$  (this corresponds to the portion of the reliable path outside  $\mathcal{L}$  and is indeed orthogonal to the extern parameters). Finally, rotate the robot until facing  $L_2$  while the laser keeps aiming at  $L_2$ .



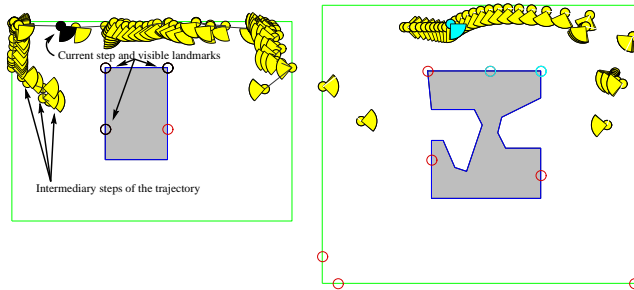
### 5.3 Simulation results

This section is meant to illustrate how the tools presented in this paper (such as the localization space or the corresponding planning algorithm) can be practically used in sensor-oriented planning. We consider once again our circular mobile robot equipped with a seto- vision system. For sake of simplicity in the space representations, we assume that the robot is holonomic and that the sensor is fixed with respect to the robot. Let us assume also that  $\alpha = 90^\circ$ ,  $p$  is half the environment width and that the robot needs two landmarks to compute its localization. We consider the environment (including 4 landmarks) represented on Fig.9 for which we compute the approximated localization space as explained in section 3.

Now, let us solve the planning problem of Fig. 9. First, we find a shortest path in the graph  $\mathcal{G}(\mathcal{L})$  (see section 4), between the two nodes containing the extremal configurations (this may not lead to the shortest ground path...). Fig.10 illustrate such a path.

Figure 9: Simulation environment, landmarks and  $\mathcal{L}$ .Figure 10: Path from *start* to *end*Path steps in  $\mathcal{W}$ Path steps in  $\mathcal{L}$ 

From the resulting path, we extract a complete trajectory (see Fig.11, left). In each node the robot follows a linear trajectory (black lines) and at some edges it may reorient its cameras. The resulting path is a *reliable* path along which the robot is always able to localize itself. Notice that the goal here is not to obtain optimal paths but just to show how

Figure 11: Trajectories from *start* to *end*

the perception constraints can be taken into account during planning and how they may affect the result. Typically the examples of Fig.11, show how the robot has to first move in the opposite direction of the goal or graze the wall in the corridor (corresponding to the narrow neck of Fig.10) while reorienting itself in order to guarantee that at each point at

least 2 landmarks are visible.

**Remark :** for nonholonomic robots, the planning scheme above is still complete as long as the robot is small-time locally controllable. The basic maneuvers of straight line and turning obviously need to be substituted by some feasible set of maneuvers remaining in the corresponding  $\mathcal{L}$  layer. Another approach would be to apply in  $\mathcal{L}$ , nonholonomic planning methods based on the approximation of a geometric path or even applying probabilistic schemes in  $\mathcal{L}$ , using an adequate steering method (e.g. [9] for details).

## 6 Conclusion

This paper deals with the problems associated with sensor-oriented planning for systems using landmarks for localization. We presented a framework, based on the notion of *Localization Space* that we mainly used for two applications : i) the first optimization method (in the sense of space coverage), concerning the placement of the landmarks for localization and ii) planning a reliable path along which, localization remains always possible. We also presented practical computations of near optimal landmarks placement and reliable trajectories for classical mobile robots with an on-board sensor, showing the practical relevance of our formalization.

As for future works, we would like to consider the computation of the best path when there is no reliable solution to the problem (and to integrate the existing work in the literature on the propagation of the uncertainty). In other words, where to cross safely : where and how to move outside  $\mathcal{L}$  to guarantee a bound on the uncertainty. Some secondary interesting topic may be : building the localization space for a robot with a general shape, improving the path planner to get “nice” reliable paths and working more specifically on the nonholonomic case.

## References

- [1] J.M. Armingol, L. Moreno, A. de la Escalera, and M.A. Salichs. Landmark perception planning for a mobile robot localization. In *Proc. of the 1998 IEEE Int. Conf. on Robotics and Automation*, 1998.
- [2] Aronov, Guibas, Teichmann, and Zhang. Visibility queries in simple polygons and applications. In *ISAAC: 9th International Symposium on Algorithms and Computation*, 1998.
- [3] G. Blanco, H. Everett, J. García-Lopez, and G.T. Toussaint. Illuminating the free space between quadrilaterals with point light sources. In M. Gigante and T. L. Kunii, editors, *Proc. Computer Graphics Int.*, World Scientific, 27 –1 1994.
- [4] Th. Fraichard and A. Lambert. Planning safe paths for nonholonomic car-like robots navigating through computed landmarks. In *Int. Conf. on Intelligent Autonomous Systems*, pages 447–454, Venice (IT), July 2000.
- [5] Guibas, Motwani, and Raghavan. The robot localization problem. In Goldberg, Halperin, Latombe, and Wilson, editors, *Algorithmic Foundations of Robotics, The 1994 Workshop on the Algorithmic Foundations of Robotics, A. K. Peters*, 1995.

- 
- [6] A. Lazanas and J.-C. Latombe. Landmark-based robot navigation. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, pages 816–822, San Jose, California, 1992. AAAI Press.
  - [7] C. Madsen and C. Andersen. Optimal landmark selection for triangulation of robot position. *Journal of Robotics and Autonomous Systems*, 13(4):pp277–292, 1998.
  - [8] Russell, Greiner, and Isukapalli. Learning to select useful landmarks. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 26(3), 1996.
  - [9] S. Sekhavat and J.-P. Laumond. Topological property for collision-free nonholonomic motion planning: the case of sinusoidal inputs for chained form systems. *IEEE Trans. Robotics and Automation*, 14(5):671–680, October 1998.
  - [10] K. Tashiro, J. Ota, Y.C. Lin, and T. Arai. Design of the optimal arrangement of artificial landmarks. In *Proc. of the 1995 IEEE Int. Conf. on Robotics and Automation*, 1995.



---

Unité de recherche INRIA Rhône-Alpes  
655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique  
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

---

Éditeur  
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-6399