



**HAL**  
open science

## Visibility Based Probabilistic Roadmaps

Carole Nissoux, Thierry Simeon, Jean-Paul Laumond

► **To cite this version:**

Carole Nissoux, Thierry Simeon, Jean-Paul Laumond. Visibility Based Probabilistic Roadmaps. IEEE/FSR International Conference on Intelligent Robots and Systems (IROS), Oct 1999, Kyongju, South Korea. hal-04292747

**HAL Id: hal-04292747**

**<https://laas.hal.science/hal-04292747>**

Submitted on 17 Nov 2023

**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.

# Visibility Based Probabilistic Roadmaps

C. Nissoux, T. Siméon and J-P. Laumond

LAAS-CNRS  
7, avenue du Colonel-Roche  
31077 Toulouse Cedex - France  
{nissoux,nic,jpl}@laas.fr

## Abstract

*This paper presents a variant of probabilistic roadmap algorithms that recently appeared as a promising approach to motion planning. We exploit a free-space structuring of the configuration space into visibility domains in order to produce small roadmaps. The algorithm has been implemented within a software platform allowing to address a large class of mechanical systems. Experiments show the efficiency of the approach in capturing narrow passages of collision-free configuration spaces.*

## 1 Introduction

Due to the continuous increasing power of the computers, probabilistic approaches to motion planning (e.g., [2, 6, 13, 8, 3]) allow today to solve practical problems which were not addressed few years ago. Apart some attempts aiming to provide formal models of complexity [10, 7, 4, 15], the success of such methods remains better noticed than well understood.

This paper proposes a variant of the Probabilistic RoadMap (PRM) algorithm introduced in [6] (and independently in [13] as the Probabilistic Path Planner). These algorithms generate collision-free configurations randomly and try to link them with a simple local path planning method. A roadmap is then generated, tending to capture the connectivity of the collision-free configuration space  $CS_{free}$ .

Our variant of these approaches takes advantage of the visibility notion. While usually each collision-free configuration generated by the PRM algorithm is integrated to the roadmap, our algorithm keeps only configurations which either connect two connected components of the roadmap, or are not “visible” by some so-called guard configurations.

This approach computes roadmaps with small number of nodes. It integrates a termination condition related to the volume of the free space covered by the roadmap. Experimental comparison shows good performances in terms of computation time, especially when applied to configuration spaces with narrow passages.

Section 2 introduces the notion of visibility roadmap. Section 3 describes a simple probabilistic algorithm that computes such roadmaps. In Section 4, we provide a comparison with the basic algorithm classically used in the PRM scheme. Finally, the last section presents several examples of problems solved by the planner that is integrated in the **Move3d** software platform dedicated to motion planning.

## 2 Visibility Roadmaps

**Definitions** Consider a robot  $\mathcal{R}$  moving in a workspace  $\mathcal{W}$ . Let  $CS$  be the configuration space of the robot. Let  $CS_{free}$  be the collision-free configuration space with respect to  $\mathcal{W}$ . Let  $\mathcal{L}$  be any local method that computes a path  $\mathcal{L}(q, q')$  (e.g., the straight line segment) between two given configurations  $q$  and  $q'$ .

We define the visibility domain of a configuration  $q$  for  $\mathcal{L}$  by:

$$Vis_{\mathcal{L}}(q) = \{q' \in CS_{free} \text{ such that } \mathcal{L}(q, q') \in CS_{free}\}$$

$q$  is said to be the *guard* of  $Vis_{\mathcal{L}}(q)$ . A configuration is said to be a *connection* if it lies within the visibility domains of at least two guards.

Consider a coverage of  $CS_{free}$  by  $s$  visibility domains  $Vis_{\mathcal{L}}(q_i)$ ,  $1 \leq i \leq s$ , such as  $\mathcal{L}(q_i, q_j) \notin CS_{free}$  for any pair of guards  $(q_i, q_j)$ . Then we build the following graph  $R$ . Guards are nodes of the graph. For any two intersecting visibility domains  $Vis_{\mathcal{L}}(q_i)$  and

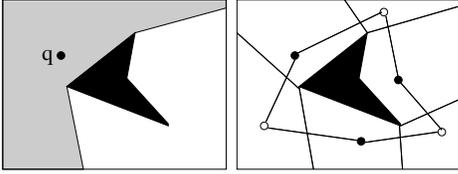


Figure 1: Visibility domain of a configuration and the roadmap defined by three guards nodes (black) and three connection nodes (white). Here paths  $\mathcal{L}(q, q')$  are the straight line segments  $[q, q']$ .

$Vis_{\mathcal{L}}(q_j)$ , we add a connection node  $q$  and two edges  $(q, q_i)$  and  $(q, q_j)$  (see Figure 1). The graph  $R$  clearly verifies the following property:

*Property:* Let us consider any two configurations  $q_{init}$  and  $q_{goal}$  such as there exists a sequence of collision-free paths of type  $\mathcal{L}$  between them. Then there is a guard node  $q_1$  and a guard node  $q_2$  in  $R$  such as:  $q_{init} \in Vis_{\mathcal{L}}(q_1)$ ,  $q_{goal} \in Vis_{\mathcal{L}}(q_2)$  with  $q_1$  and  $q_2$  lying in a same connected component of  $R$ .

This property means that  $R$  captures the connectivity of  $CS_{free}$ .  $R$  is said to be a *visibility roadmap*.

**Comments** The notion of visibility roadmap raises several comments. Since the definition of the visibility domains is related to a local method, it would have been better to use the term of “reachable domain”. Both notions are identical when the local method provides elementary straight line segments. We keep the word “visibility” because it is more intuitive. We consider implicitly that  $R$  is an undirected graph; that means that  $\mathcal{L}$  should be symmetric.

The existence of finite coverage by visibility domains depends both on the shape of  $CS_{free}$  and on  $\mathcal{L}$ . Such coverage may not exist<sup>1</sup>. Nevertheless these cases are difficult for any other alternative planners.

Finally the number of guards is not required to be optimal. Optimality refers to the well known and challenging art gallery problem [12].

### 3 A probabilistic algorithm

**Description** The algorithm iteratively processes two sets of nodes: *Guard* and *Connection*. The nodes of *Guard* belonging to a same connected component (i.e. connected by nodes of *Connection*) are gathered in subsets  $G_i$ . At each elementary iteration, the algorithm selects a collision-free configuration  $q$ .

The main loop of the algorithm processes all the current components  $G_i$  of *Guard*. It loops over the nodes  $g$  in  $G_i$ , until it finds a node visible from  $q$ . The

<sup>1</sup>A finite coverage by visibility domains does not exist for non  $\epsilon$ -good free spaces [5].

first time the algorithm succeeds in finding such a visible node  $g$ , it memorizes both  $g$  and its component  $G_i$  and switches to the next component  $G_{i+1}$ . When  $q$  “sees” another guard  $g'$  in another component  $G_j$ , the algorithm adds this configuration to the *Connection* set and the component  $G_j$  is merged with the memorized  $G_i$ .

The main loop fails to create a new node in two cases:

- either  $q$  is visible from only one component; in that case  $q$  is rejected.
- or  $q$  is not visible from any guard: in that case it becomes a guard node that creates a new component  $\{q\}$  inserted into set *Guard*.

Parameter *ntry* is the number of failures before the insertion of a new guard node.  $1/ntry$  gives an estimation of the volume not yet covered by visibility domains. It estimates the fraction between the non covered volume and the total volume of  $CS_{free}$ . This is a critical parameter which controls the end of the algorithm. Hence, the algorithm stops when *ntry* becomes greater than a user set value  $M$ , which means that the coverage of the free space by visibility domains becomes greater than  $(1 - \frac{1}{M})$ .

#### Algorithm

```

ntry ← 0
While (ntry < M)
  Select a random free configuration q
  guard ← ∅; G_vis ← ∅
  For all components G_i of Guard do
    found ← FALSE
    For all nodes g of G_i do
      If (q belongs to Vis(g)) then
        found ← TRUE
      If (¬guard) then guard ← g; G_vis ← G_i
      Else /* q is a connection node */
        Add q to Connection
        Create edges (q, g) and (q, guard)
        Merge components G_vis and G_i;
    until found = TRUE
  If (¬guard) then /* q is a guard node */
    Add {q} to Guard; ntry ← 0
  Else ntry ← ntry + 1

```

**End**

Figure 2 shows examples of visibility roadmaps computed by the algorithm.

**Termination criterion** Each new guard node inserted to the roadmap increases the coverage of  $CS_{free}$ . Therefore the probability of generating configurations in non covered regions keeps decreasing.

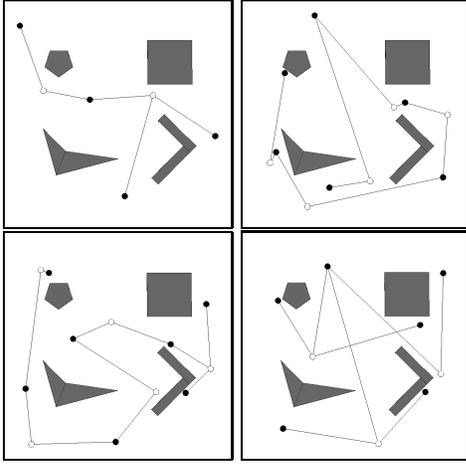


Figure 2: Four runs of the algorithm on a simple example with  $M = 50$ .

The algorithm is then guaranteed to terminate for any value of  $M$ . When it stops, a probabilistic estimation of the percentage of the free space not covered by the guards is  $\frac{1}{M}$ . Therefore, the probability for the local method to fail in connecting two configurations  $q_{init}$  or  $q_{goal}$  to the roadmap is  $(1 - \frac{1}{M})$ .

The solution path shown in Figure 3 was found from a visibility roadmap computed by the algorithm with  $M = 400$ .

**Analysis** Figure 4 shows the evolution of the free-space coverage for the 6-dof puzzle example (Figure 3) with respect to the number of guards generated by the algorithm. The solid curve plots  $(1 - \frac{1}{n_{try}})$ . The dashed curve plots the real percentages of the free space coverage (they were computed by testing the visibility of each set of guards with a huge number of free configurations). As shown by the figure, the estimation rapidly converges toward the actual coverage: the error becomes less than one percent with 100 guards. Also one can note that despite the rather complex shape of  $CS_{free}$  for this example, less than hundred guards are sufficient to see 98% of the free space.

**Side-effect and limitation** Two limitations of the approach are illustrated in Figures 5. In the left figure, two guard nodes have been generated. They fully cover  $CS_{free}$ . The intersection domain of both visibility domains is “unfortunately” small. The only way to complete the roadmap is to pick a connection node in the small triangle. Then the algorithm will fail if the parameter  $M$  is not sufficiently high. Nevertheless this case is only a side-effect of the algorithm. Indeed,

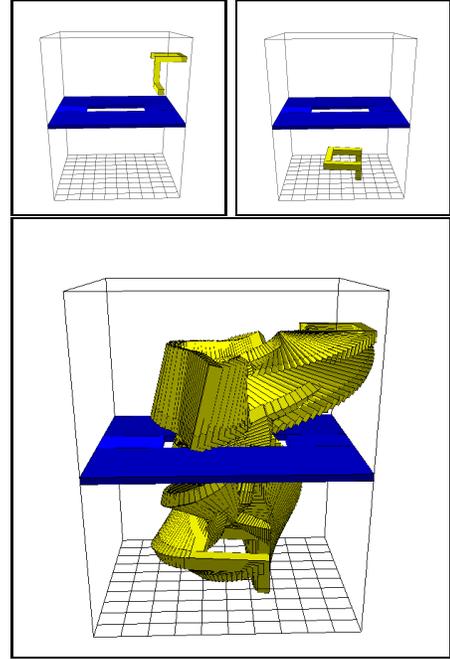


Figure 3: A solution path for the 6-dof puzzle problem presented in [9].

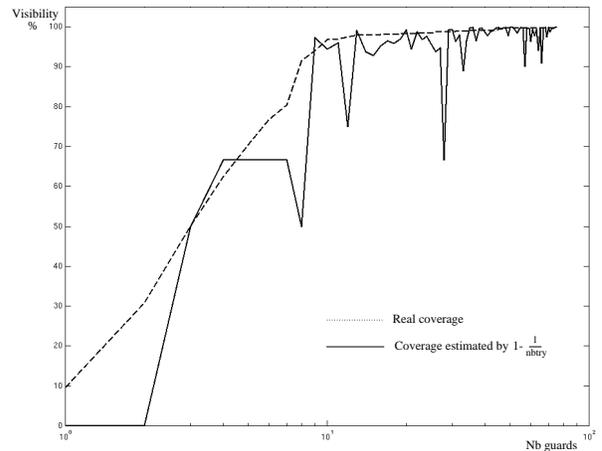


Figure 4: Evolution of the free space coverage w.r.t. the number of guards

in this example, the probability to select the first two guards with a small intersection domain is small. We may overcome this effect by running the algorithm twice.

The second figure (on the right) illustrates the difficult case to put guards in narrow passages with a probabilistic approach. In the next section we present experimental results comparing the performance of our approach with respect to other probabilistic ap-

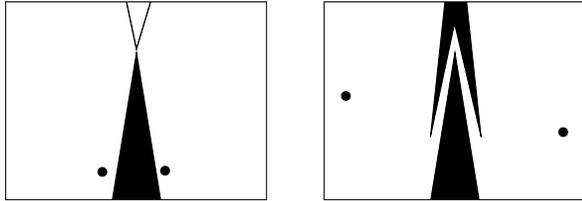


Figure 5: Two difficult situations: **a/** a pathologic case for our algorithm; both visibility domains have a very small intersection and **b/** existence of narrow passage difficult to capture for any random sampling algorithm.

proaches.

## 4 Visibility roadmap *versus* PRM

**The PRM scheme** Probabilistic roadmaps (PRM) have been introduced in [6, 13]. The algorithm consists of two phases. It first constructs a roadmap  $R$  that characterizes the connectivity of the free configuration space  $CS_{free}$  by randomly selecting a large number of collision-free *milestones*, and checking if each (or some) pair of milestones can be connected by a collision-free path generated by a simple (usually not complete) local method. When the preprocessing phase has produced a roadmap that is representative of the complexity of the free-space, the query phase can efficiently process motion planning problems by connecting the initial and final configurations to the same connected component of  $R$ . Several variants or extensions of this basic scheme have been proposed [5, 9, 1] to improve the efficiency of the preprocessing and the connectedness of the produced roadmap, in particular for difficult cases where the free space contains narrow passages.

**Comparison** With respect to these methods, the main advantage of the visibility roadmap is illustrated in Figure 6. In this example  $CS_{free}$  has two connected component. Our algorithm will quickly find the two guards that cover both connected components, and the algorithm will stop just after  $M$  elementary loops. At each of these loops the local method will be called only *twice*, while the basic PRM algorithm should call the local method a lot of times to check whether each new collision-free node can be connected to the roadmap.

With a same number  $n$  of random collision-free configurations, the visibility roadmap algorithm will call the local method  $O(n)$  times, while a basic PRM algorithm will call it  $O(n^2)$  times<sup>2</sup>.

<sup>2</sup>Limiting the calls to a given neighborhood (see [9] for instance) may allow to reach a linear complexity with a “well” chosen value of the threshold distance; however, a larger size  $n$

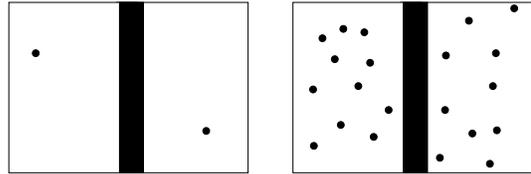
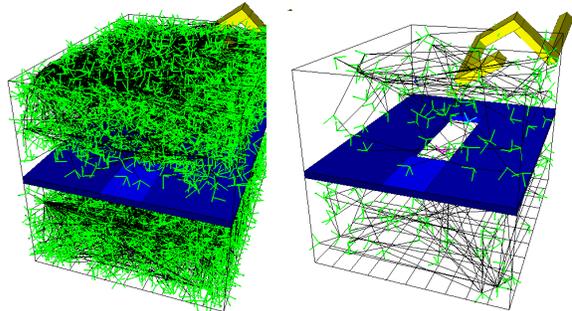


Figure 6: Visibility roadmap versus classical PRM



PRM	Basic	Visibility
Roadmap size	4723	103
$CS_{free}$ coverage	99.9%	99.7%
Random confs	14169	14369
Free confs	4723	4753
Local method #calls	700610	57622
Col. checker #calls	8.725985	1.121790
CPU time	3367 sec	281 sec

Figure 7: Comparison onto the 6-dof puzzle example.

Obviously, the analysis of this simple example is not a proof. We performed several experiments comparing the performances of **Basic-PRM** and **Visibility-PRM**. Figure 7 summarizes the results of one such experiment performed onto a 6-dof “puzzle” example [9]. The free-space of this example consists of two large regions connected by a narrow passage. As noted in [9], the difficulty of the problem depends on the relative width of the passage with respect to the smallest dimension of the moving object. It also depends on the total volume of the configuration space. The results summarized in the table were obtained for the following dimensions: the workspace has dimension  $200 * 150 * 150$  with a rectangular passage of width 30 and the moving object is made up five blocks of length 50 and cross-section 10. Each test was performed by running both algorithms with the *same set* of configurations (i.e. the random node generator was initialized with the same seed) until the roadmap captured the connection between the two large regions. All the values of the table were averaged over ten runs performed

may be required to maintain the roadmap’s connectedness.

with different seeds.

**Visibility-PRM** captured the free space connectivity with a roadmap much smaller than **Basic-PRM** while maintaining the same coverage of the free space. Moreover, the performance (number of calls to the local method or CPU time) was 12 times better. Notice that both required roughly the same total number of random configurations.

Similar experiments performed onto this example tend to show that the gain still increases for larger dimensions of the workspace (it drops to 18 when doubling the  $z$ -dimension) or a smaller width of the passage. On the other side, comparisons performed onto the less constrained examples of Section 5 showed a gain of performance ranging from one to a few times. For all cases, the produced visibility roadmaps are much smaller.

## 5 Other Results

Both algorithms were integrated into the **Move3d** software platform<sup>3</sup>. **Move3d** serves as a testbed for motion planners that we develop for polyhedral workspaces. It allows to model many kinds of mechanical systems: free-flying, manipulators, nonholonomic mobile vehicles, mobile manipulators... The library contains a set of functionalities that make easy the development of new motion planners. Collision detection is currently performed with the *I-Collide* library [11].

Figures 8, 9 and 10 present various problems solved with the same version of **visibility-PRM**. In this version, the algorithm includes starting and goal configurations as initial guards and stops as soon as it finds a solution path.

## 6 Conclusion

In this paper we have introduced the notion of visibility roadmaps as a way to capture the free-space connectivity with a small number of guards and also, to control the end of the roadmap's computation by estimating the quality of the free-space coverage within the algorithm.

The first results obtained with a preliminary implementation of the algorithm are encouraging, especially for the difficult cases of free-spaces containing narrow passages which usually require dense and expensive roadmaps with PRM schemes.

Although preliminary, these results raise several interesting issues that we expect to address. It may be interesting to investigate the idea of “movable

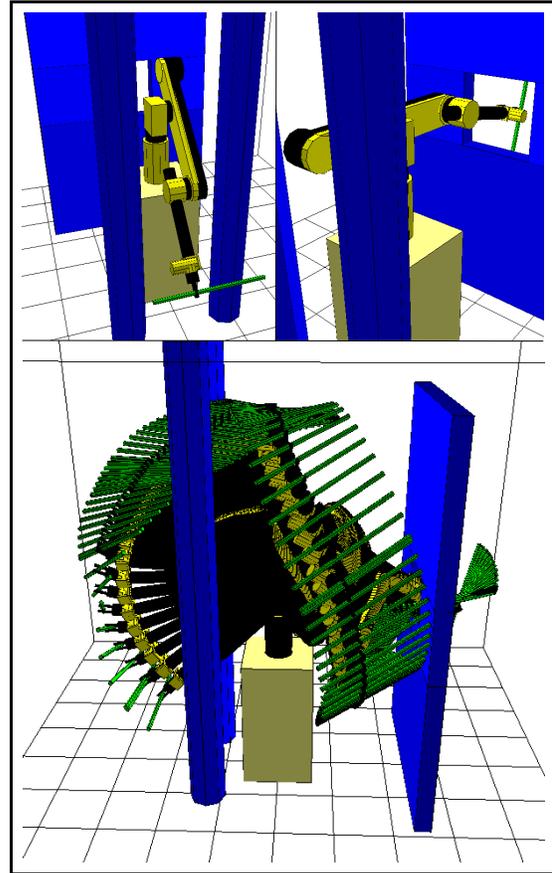


Figure 8: A path for a 6-dof manipulator found in 120 seconds.

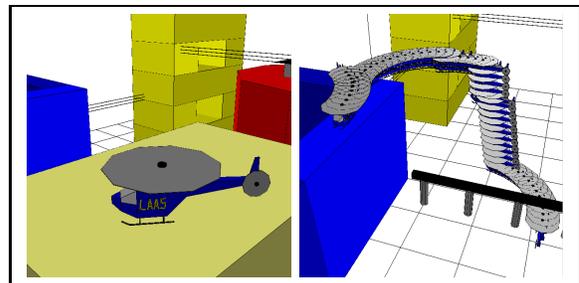


Figure 9: A simple problem with a 4-dof helicopter amidst buildings (< 1 second).

guards” in order to limit the risk of generating guards whose visibility domains have a very small intersection. Also, a formal analysis of the proposed method would be desirable to better characterize its performance from geometric properties of the free-space such as  $\epsilon$ -goodness or expansiveness introduced in [7].

<sup>3</sup>see <http://www.laas.fr/~nic/G3D/RIA-g3d.html>

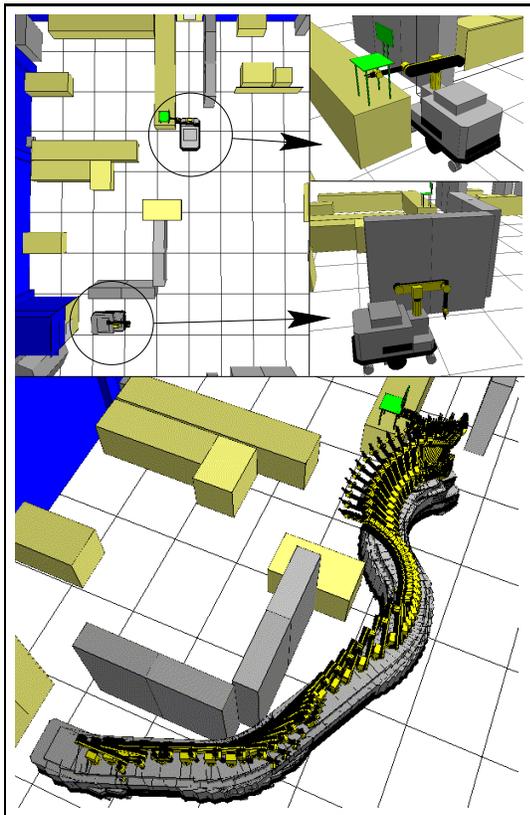


Figure 10: A coordinated motion of a 9-dof nonholonomic mobile manipulator (20 seconds). Here the local method addresses the nonholonomic constraint of the platform via the computation of Reeds and Shepp curves [14].

## References

- [1] N. Amato, O Bayazit, L. Dale, C. Jones and D. Vallejo. OBPRM: an obstacle-based PRM for 3D workspaces. In *Robotics: The Algorithmic Perspective (WAFR98)*, P. Agarwal and all (Eds), AK Peters 1998.
- [2] J. Barraquand and JC. Latombe. Robot motion planning: a distributed representation approach. In *International Journal of Robotics Research*, vol 10 n 6, 1991.
- [3] P. Bessiere, J. Ahuactzin, T. El-Ghazali and E. Mazer. The “Ariane’s clew” algorithm: Global planning with local methods. In *IEEE International Conference on Robots and Systems*, 1993.
- [4] D. Hsu, JC. Latombe, R. Motwani. Path planning in expansive configuration spaces. In *IEEE International Conference on Robotics and Automation*, Albuquerque (USA), 1997.
- [5] D. Hsu, L. Kavraki, JC. Latombe, R. Motwani and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In *Robotics: The Algorithmic Perspective (WAFR98)*, P. Agarwal et al (Eds), AK Peters, 1998.
- [6] L. Kavraki and JC Latombe. Randomized pre-processing of configuration space for fast path planning. In *IEEE International Conference on Robotics and Automation*, San Diego (USA), 1994.
- [7] L. Kavraki, L. Kolountzakis and JC. Latombe. Analysis of probabilistic roadmaps for path planning. In *IEEE International Conference on Robotics and Automation*, Minneapolis (USA), 1996.
- [8] L. Kavraki, P. Švestka, J.-C. Latombe and M.H. Overmars. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces, In *IEEE Transactions on Robotics and Automation vol 12, n4*, 1996.
- [9] L. Kavraki and JC. Latombe. Probabilistic roadmaps for robot path planning. In *Practical Motion Planning in Robotics*, K. Gupta and A. del Pobil (Eds), Wiley Press, 1998.
- [10] F. Lamiroux and JP. Laumond. On the expected complexity of random path planning. In *IEEE International Conference on Robotics and Automation*, Minneapolis (USA), 1996.
- [11] M. Lin, D. Manocha, J. Cohen and S. Gottschalk. Collision detection: Algorithms and applications. In *Algorithms for Robotic Motion and Manipulation (WAFR96)*, JP. Laumond and M. Overmars (Eds), AK Peters 1997.
- [12] J. Goodman and J. O’Rourke. *Handbook of Discrete and Computational Geometry*. CRC Press, 1997.
- [13] M. Overmars and P. Svestka. A Probabilistic learning approach to motion planning. In *Algorithmic Foundations of Robotics (WAFR94)*, K. Goldberg et al (Eds), AK Peters, 1995.
- [14] J. A. Reeds and R. A. Shepp. Optimal paths for a car that goes both forward and backwards. *Pacific Journal of Mathematics*, 145 (2), pp. 367–393, 1990.
- [15] L. Kavraki, M. Kolountzakis and J.-C. Latombe. Analysis of Probabilistic Roadmaps for Path Planning, In *IEEE Transactions on Robotics and Automation vol 14, n1*, 1998.