New ways of generating better configurations using geometric features

Antonio Benitez Daniel Vallejo

CENTIA
Univeridad de las Américas Puebla
Sta. Catirina Martir S/N
Cholula, Puebla 72820
México
sc098381,dvallejo@mail.udlap.mx

Abstract

This work describes a new approach to capture the C-space connectivity and propose new algorithms to improve the connectivity of the configuration space using Probabilistic Roadmap Methods (PRMs). The main idea is to use some geometric features into the work-space, and to obtain free configurations close to the obstacles. In order to reach a better performance of these planners, we use the *"straightness"* feature and propose a new heuristic which let us to solve the narrow corridor problems. We apply this technique to solve the motion planning problems for a specific kind of robots, the "*free flying objects*" with six degrees of freedom. We show results that allow us to think that some geometric features on the work-space can be used to build heuristics that can be used to solve the narrow corridor problems improving the connectivity of the configuration space.

1 Introduction

Automatic motion planning has been an expanding area of research over the past several years in computer science. In its simplest form, motion planning deals with finding a collision-free path to move an object between a start and goal position [Latombe, 1991].

Recently, a new class of randomized path planning methods has gained much attention (see, e.g., [Ahuactzin and Gupta, 1997; Amato and Wu, 1996; Kavraki and Latombe, 1994; Kavraki et al., 1996; Overmars 1992; Overmars and Svestka, 1994; Simeón et al]). This methods know as Probabilistic Roadmap Methods (PRMs) have shown great potential for solving complicated high-dimensional problems. PRMs use randomization (usually during the preprocessing) to construct a graph of representative paths in C-space (a roadmap) whose vertices correspond to collision free configurations of the robot and in which two vertices are connected by an edge if a path between the two corresponding configuration can be found by a local planning method.

We chose to deal with the obstacles and robot geometry on the workspace because in general it is not feasible to work with the geometric of the configuration space. We condense the description of the obstacles and some one of their geometric features into a simpler structure which can be utilized more directly in finding good configurations.

We select to use geometric features of the workspace as a guide for choosing configurations. We describe methods for computing free configurations near to the obstacles given some geometric features about the robot and the obstacles.

2 Related Work

Planning for robots with many degrees of freedom has been extensively treated in the literature [Ahuactzin et al., 1992; Amato et al, 1998; Gupta and Poibil, 1998; Halperin et al., 1997; Kondo, 1991; Latombe, 1991]. The probabilistic roadmap approach to planning (PRM) [Kavraki and Latombe, 1994; Kavraki and Latombe, 1998; Kavraki et al., 1996; La Valle et al., 1999; Overmars and Svestka, 1995; Vallejo et al., 2001] has gained wide acceptance because the method is easy to implement and use and provides good performance results.

An important issue in PRM planners is the method for choosing the random configurations for the construction of the roadmaps. Recent works have considered many alternatives to a uniform random distribution of configurations as means for dealing with the narrow passage problem. A resampling step, creating additional nodes in the vicinity of nodes that are connected with few others, is shown in [Kavraki and Latombe, 1998]. Nodes close to the surface of the obstacles are added in [Amato et al., 1998], and in [Boor et al., 1999] a probabilistic method for choosing configurations close to the obstacles is presented. A dilation of the configuration space has been suggested in [Hsu et al., 1998], as well as an in depth analysis of the narrow passage problem. In [Wilmart et al., 1999] a procedure for retracting configurations onto the free space medial axis is presented. In this paper, we treat rigid objects and we attempt to generate configurations close to the obstacles into the workspace. They give algorithms that perform this retraction while avoiding explicit computation of the medial axis.

Holleman and Kavraki [Holleman et al., 1998], have proposed another solution to reduce the number of configurations to calculate for building the configuration space. They claim that by constructing a skeleton-like subset of the free configurations of the workspace. They examine the medial axis as a skeleton.

Independently, Kavraki and Latombe in [Kavraki and Latombe, 1994] and Overmars and Svestka [Overmars, 1992, Overmars and Svestka, 1994], proposed similar path planning methods which use randomization during preprocessing to construct a graph in *C-space* (often called a roadmap [Latombe, 1991]). Planning can be done very fast in many situations. However, long, narrow passages between C-obstacles might be difficult to find.

3 Notation

Let $B = B_0,\ldots,B_n$ be a set of obstacles in the workspace W. Let r_i be the radius associate to the sphere which involve each body in the workspace (including the robot which will be denoted by R.

Let d_i be the distance between two configurations, one of them associate to the robot and another one associate to the obstacle B_i.

We will call c(B_i) the random configuration calculated by the heuristic with respect the B_i obstacle.

We will call v to each c(B_i) added to the roadmap and this will be denoted by V.

Let v_i be the vector which define the direction of the *"straightness"* feature for each B_i in B, and vr_i will define the same feature on the robot. This feature indicates the direction which the body presents its long side.

The symbol $||$ will be used to show that two configurations c(B_i) and c(B_j) are in parallel, that means that, the bodies associated to each configuration keep in parallel their direction vectors.

4 Algorithm Description

The description of the algorithm is divided into three parts: first approximation of the configuration space, improving the connectivity using geometric features and planning.

The following sections describe how the algorithm works and we give some details about its implementation. Some Figures are included to show the main idea in this new technique.

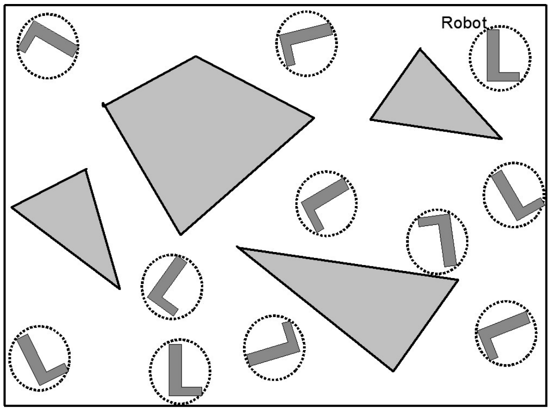

Figure 1. First approximation using spheres, which involving the robot in Workspace.

4.1 Preliminars

The moving objects (robot) considered in this paper are rigid objects in three space. We present configurations using six-tuples ($x,y,z,\alpha,\beta,\gamma$), where the three first coordinates define the position and the last three define the orientation. The orientation coordinates are represented in radians.

The configuration space (*C-space*), is the parameter space of the robot. A robot in *W-space* is represented by a point in C-sapce, and any point in *C-space* corresponds to some placement of an actual robot in *W-sapce* [Lozano-Pérez, 1983]. The underlying idea of C-space, is to represent the robot as a point called the reference point of the robot snd map the obstacles into this space by growing their size of the robot.

In adition to collision detection, all PRMs make heavy use of so-called *local planners* and *distance computation*. Local are simple, fast, deterministic methods used to make connections between roadmap nodes when building the roadmap, and to connect the start and goal to the roadmap during queries. Distance metric are used to determine which pairs of nodes one should try to connect.

The distance metric we use is an Euclidean distance in Work-space.

The local planners currently implemented in Geometric Based Obstacles PRM are the common straight-line in work-space.

4.2 First Aproximation of C-Space

This approximation is generated using spheres which involve the robot. Using this idea the robot has the advantage to rotate in any direction (α,β,γ).

In this stage we can obtain a first sampling of configuration space. The Figure 1 shows the view of the first sampling. In this stage we have the advantage to reduce the cost of collision detection, because the routine is reduced to detect when two spheres are in collision (the sphere associated to each obstacle and the sphere associated to the robot). In this way we only have to compute a reduced number of configurations. The next algorithm present how to obtain this representation.

Sampling_with_spheres()

```
for k = 1 to CTE_NODES
    c = new Configuration()
    if ( c is_not in_collision_sphere)
    add_configuration  c  to the roadmap
endfor
```

4.3 Improving the Connectivity

After the first approximation has been computed, the algorith has a primary roadmap. In this secound step the technique will try to improve the conectivitity of the roadmap. To reach that goal, a new process is run.

This technique attempt to take advantage of some geometric features of the robot and the obstacles to obtain information that allows guide the search for useful configurations, and reduce the account of non-valid configurations to be calculated. In order to reach a better performance of these planners we will obtain a better representation of the connectivity of the configuration space.

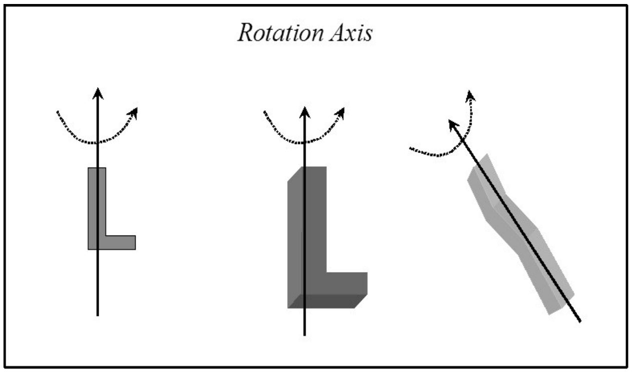

Figure 2. Rotation axis defined with *straightness* feature.

The first feature used is *"Straightness"*. This feature is given by a vector, which we have used as the direction of the rotation axis. We begin by calculating a random configuration c(B_i) near to each obstacle B_i. If the c(B_i) is in collision, then the algorithm attempts to move this configuration to turn it in a free configuration, which will be close to some obstacle (Amato and Wu in [Amato and Wu, 1996], propose an algorithm which compute configurations on *C - Obstacle surfaces)*.

The Figure 2. shows the way which the rotation axis is represented on the robot and the obstacles. We can see that the body will sweep a minor area as result of defining the rotation axis in that way.

4.4 Node Generation

First, the algorithm search a collision configuration c(B_i) close to the B_i obstacle, such c(B_i) is calculated uniformly distributed around the sphere which the obstacle is involved. Before the c(B_i) is found the process obtains some free configurations which are added to the roadmap. The next algorithm describes how these nodes are calculated near to the obstacles.

Obtaining a Collision Configuration to i-Obstacle

```
$r_i$ = radius of i-obstacle
$c_i$_init = position of the i-obstacle
while ( c($B_i$) is not in collision ) do
    c($B_i$)=get_configuration_in_vicinity($c_i$_init, $r_i$)
    if c($B_i$) is free configuration
        add_configuration  c($B_i$)  to the roadmap
    endif
endwhile
return c($B_i$)
```

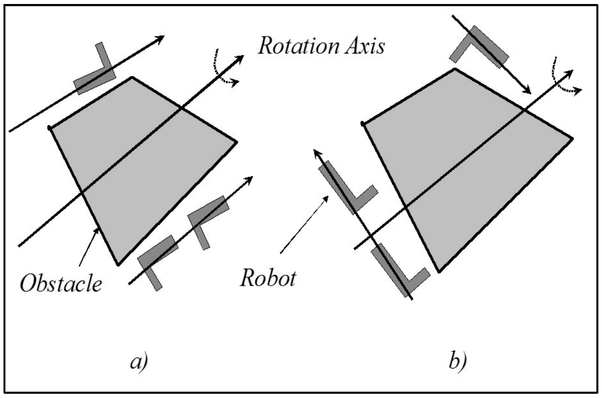

Figure 3. a) Parallel configurations to the obstacle
b) Perpendicular configurations to the obstacle

4.5 Parallel and Perpendicular Configurations

Once the collision configuration c(B_i) has been found, the first strategy is to rotate it until the rotation axis of the robot vr_i will be parallel to the rotation axis v_i of the B_i obstacle, that means that, $vr_i \parallel v_i$. The configuration obtained can be seen in the Figure 3a, where we can see different parallel configurations. If the new c(B_i) is not in collision then is added to the roadmap, if it is in collision a process called elastic band is applied on c(B_i).

The second strategy is to get a perpendicular configuration to the obstacle, this option can be see in Figure 3b. In the same way that parallel configuration, the perpendicular configuration is added to the roadmap if and only if it is not in collision, otherwise a elastic band process is applied on it.

4.6 The Elastic Band Technique

This process works as following, first it calculates the distance vector d_i between the obstacle position and the c(B_i) configuration, and attempt to approach and moving away the robot with respect to the obstacle. While the band is working (scaling the d_i vector to compute the next position where the c(B_i) is going to be placed) the robot is rotated on its rotation axis, searching a free configuration.

4.7 Applying the Heuristic

The first criteria tries to move and rotate the configuration in smooth way, searching to keep the rotation axis of the robot parallel to the rotation axis of the obstacle where there was collision. The technique works like a band which was described in the previous section. While the elastic band process is computed, *R* is rotated on vr_i, this heuristic imagine that if the *"straightness"* feature was defined as in notation section then, when we try to rotate *R* about vr_i the swept area will be the lowest. The Figure 4. shows how the algorithm works, searching for calculating configuration around the obstacle and close to it. The next algorithm presents the elastic band process.

Elastic Band Heuristic

```
$r_i$ = radius of i-obstacle
$c_i$_init = position of the i-obstacle
escalar = 0
angle = 0
k=0
$d_i$ = distance vector between i-obstacle and robot
do
    scale = random between (0.5 and 1.5)
    scale ($d_i$, escalar)
    angle = random between $(0,2\pi)$
    rotate_robot_on_rotation_axis(angle)
    c($B_i$)=get_configuration_with_position ($d_i$)
while ( c($B_i$) is not free configuration and k < CTE )
add_configuration c($B_i$) to the roadmap
```

4.8 Connecting Roadmap Candidates

We now consider how to connect the candidate nodes to create the roadmap. The basic idea is to use a simple, fast, local planner to connect pairs of roadmap candidate nodes. To save space, the paths found in this stage will not be recorded since they can be generated quickly. After the connections are made, the connected components in the roadmap are identified, e.g., by depth first search.

Ideally, the roadmap will include paths trough all corridors in work-space. The degree to which this goal can be met depends upon a number of factors: the number and distribution of the candidate nodes, the effectiveness of the simple planner, and the number of connections attempted for each candidate node. Thus, a trade-off exists between the quality of the resulting roadmap and the resources (computation and space) one is willing to invest in building it.

Many different connection strategies could be used in path planning applications. For example, the method used in 8 into try to connect each node $v \in V$ to its *k* nearest neighbors in *V*.

5 Planning

Planning is carried out as in any roadmap method: we attempt to connect the nodes x_1 and x_2, representing the start and goal configurations, respectively, to the same connected component of the roadmap, and then find a path in the roadmap between these two connection points. The following approach, proposed in [Kavraki and Latombe, 1994], is well suited for our roadmap.

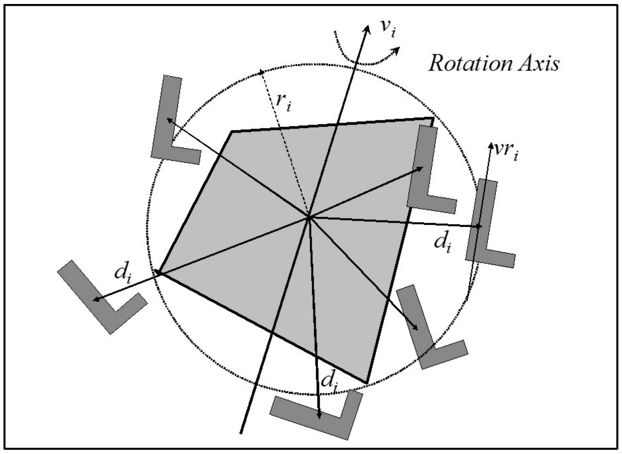

Figure 4. Configurations calculated around the obstacle using the *elastic band process*.

First, the simple planner is used to try to connect the start a goal nodes to the roadmap; connections are attempted between x_1 and the *k* - closets roadmap nodes. If we can not connect both nodes to the some connected component of the roadmap, then we declare failure. After both connections are made, we find a path in the roadmap between the two connection points using depth-first-search. Recall that we must regenerate the path between adjacent roadmap nodes since they are not stored with the roadmap.

The dominant operation in the creation of the roadmap is collision detection: it is heavily used both in the node generation and in the roadmap connection phases. Thus, its efficiency is of vital importance to the overall efficiency of the method. So, collision detection was performed with the *Rapid* library [Lin et al., 1997] .

6 Experimental Results

We implemented a path planner for a particular free flying object robot in a three-dimensional space. Our implementation was written in C++, and all test cases were run on PC Pentium processor.

In the following, we analyze the performance of the method (in effectively way) a few environments. In all cases we have used a free-flying object robot whit six degree of freedom. The various environments, and some representative configurations of the robot, are shown in Figures 5, 6, 7, 8 and 9. Note that the roadmap size is influenced by the number of obstacles in the workspace since a set of roadmap nodes is generated for each obstacle, i.e., the size of the network is related to the complexity of the environment. We used the heuristic interconnection strategy of [Kavraki and Latombe, 1994] of picking $k = 10$ candidates from each set V_i for every roadmap node.

The three samples shown are presented in two stages; in the first stage the Figure a) shows how the algorithm sample the configuration space using spheres technique (Figure 1.), and in the second stage the Figure b) shows how the algorithm improve the connectivity of the roadmap using the "straightness" geometric feature and the elastic band method.

E1: This environment contains five obstacles and the robot is represented by a tetrahedron. The roadmap for this environment is simple, because there is no corridor narrows, but we can see that for this example the path generated on the roadmap is very good. See figure 5.

E2: This environment has a passage between four obstacles. For this example the robot is represented by a tetrahedron, We can see in figure 6 the more important configurations computed for the algorithm. In figure 7, a sequence of configurations is presented to show the path between two given configurations, and the query phase could solve the narrow corridor problem.

E3: This environment contains five obstacles and the robot is represented by a large tetrahedron. This roadmap is not easy to calculate, because even if there is not a narrow corridor the size of the robot is bigger than some obstacles, and the configuration around the obstacles become difficult to calculate. We can see in figure 8 some representative configurations computed for the method. In figure 9, a sequence of configurations is presented to show the path between the init and goal configurations.

7 Conclusions

We have described a new randomized roadmap method for motion planning for collision free path planning. To test the concept, we implemented the method for path planning for free flying object robots in a three-dimensional workspace. The method was shown to perform well.

Currently, we are working in the problem and we are adding the "*flatness*" features on the heuristic. The main idea is to take advantage of the geometric features on the workspace that we can obtain a representative connection of configuration space.

8 Aknowledgments

We would like to tank the Center of Research in Information and Automation Technologies (CENTIA) at the Universidad de las Américas – Puebla.

This research is supported by the project: Access to High Quality Digital Services and Information for Large Communities of Users, CONACYT.

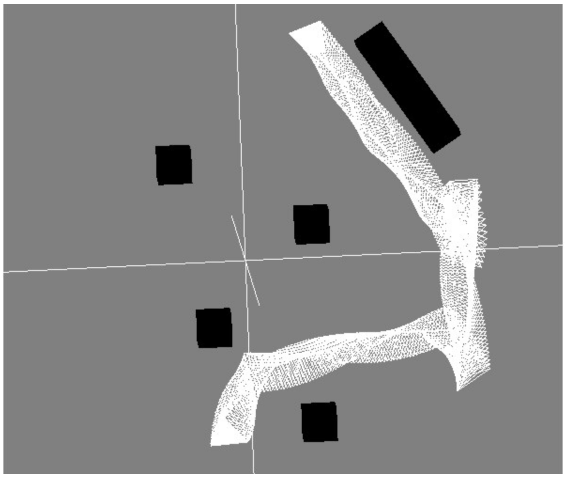

Figure 5. E1. The robot is represented as a tetrahedron and here is shown the path found by the algorithm.

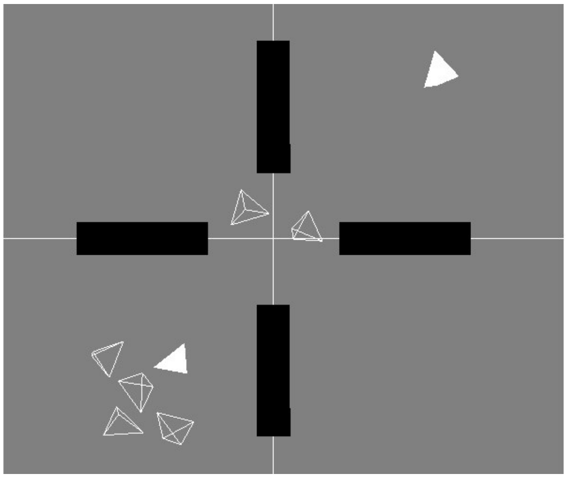

Figure 6. E2. The robot is represented as a tetrahedron and there are some configurations in the corridor.

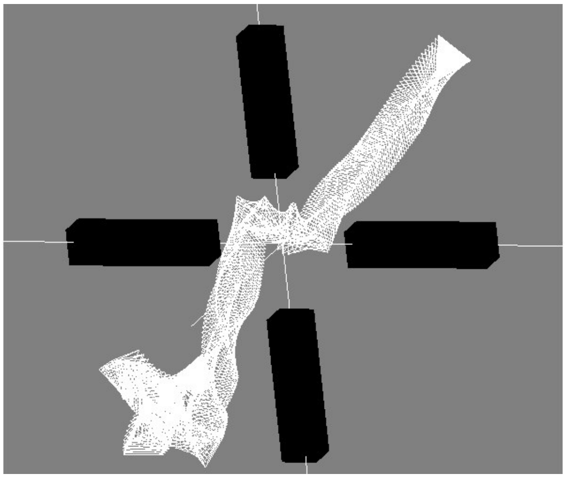

Figure 7. E2. The path between the init and goal configurations is shown.

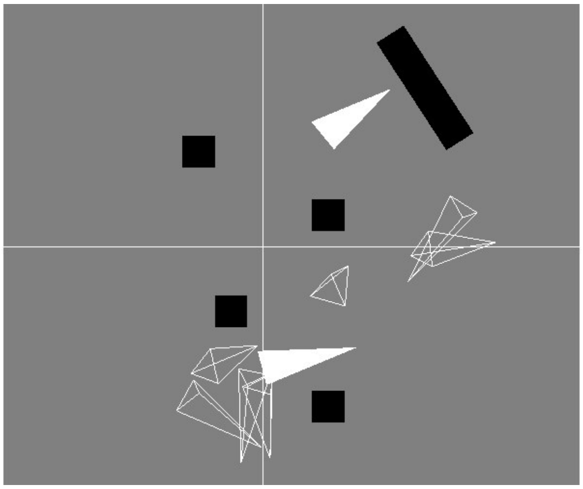

Figure 8. E3. The robot is represented as a large tetrahedron.

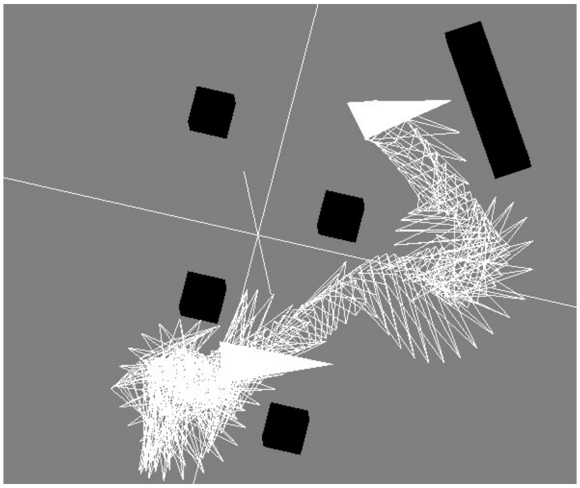

Figure 9. E3. The path computed by the algorithm can be see in this figure.

References

[Ahuactzin et al., 1992] J. M. Ahuactzin, E.-G. Talbi, P. Bessière, and E. Mazer. Using genetic algorithms for robot motion planning. In *10th Europ. Conf. Artific.* Intelligence, pages 671-675, London, England, 1992.

[Ahuactzin and Gupta, 1997] J. M. Ahuactzin and K. Gupta. A motion planning based approach for inverse kinematics of redundant robots: The kinematic roadmap. In *Proc. IEEE Int.Conf. on Rob. and Autom.,* pages 3609-3614,1997.

[Amato and Wu, 1996] N. M. Amato and Y. Wu. A randomized roadmap method for path and manipulation planning. *In Proc. IEEE Internat. Conf. Robot. Autoum,.* Pages 113-120, Mineapolis, MN, April 1996.

[Amato et al., 1998] N. Amato, B. Bayazit, L. Dale, C. Jones, and D. Vallejo. Obprm: An obstacle-based prm for 3D workspaces. In P.K. Agarwal, L. Kavraki, and M. Mason, editors, *Robotics: The Algorithm Perspective.* AK Peters, 1998.

[Boor et al., 1999] V. Boor, M. Overmars, and F. van der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int.Conf. on Rob. and Autom.,* pages 1018-1023,1999.

[Chang and Li, 1995] H.Chang and T.Li. Assembly maintainability study with motion planning. In *Proc. IEEE Int.Conf. on Rob. and Autom.,* pages 1012-1019,1995.

[Gupta and Poibil] K. Gupta and A. P. de Poibil. *Practical Motion Planning in Robotics.* John Wiley, West Sussex, England, 1998.

[Halperin et al., 1997] D. Halperin, L. E. Kavraki, and J.-C. Latombe. Robotics. In J. Goodman and J. O'Rourke, editors, *Discrete and Computational Geometry*, pages 755-778. CRC Press, NY, 1997.

[Holleman et al., 1998] C. Holleman, L. E. Kavraki, and J. Warren. Planning paths for a flexible surface patch. In *Proc. IEEE Int.Conf. on Rob. and Autom.,* pages 1012-1019,1998.

[Hsu et al., 1998] D. Hsu, L. E. Kavraki, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In P.K. Agarwal, L. Kavraki, and M. Mason, editors, *Robotics: The Algirithm Perspective*, pages 141-154. AK Peters, Natick, MA, 1998.

[Kavraki and Latombe, 1994] L. Kavraki and J.-C. Latombe. Randomized preprocessing of configuration space for fast path planning. In *Proc. IEEE Internat. Conf. Robotics and Automation*, pages 2138-2145, San Diego, CA, 1994.

[Kavraki and Latombe, 1998] L. Kavraki and J.-C. Latombe. Probabilistic roadmaps for robot path planning. In K. G. and A. P. del Pobil, editor, *Practical Motion Planning in Robotics: Current Approaches and Future Challenges,* pages 33-53. John Wiley, West Sussex, England, 1998.

[Kavraki et al., 1996] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. In *Proc. IEEE Int. Conf. Robot. and Autom.,* 12(4): 566-580, August 1996.

[Kondo, 1991] K. Kondo. Motion planning with six degrees of freedom by multi strategy bidirectional heuristic free-space enumeration. IEEE Tr. on Robotics and Automation, 7(3):267-277, 1991.

[Latombe, 1991] J-C. Latombe. *Robot Motion Planning*. Kluwer, Boston, MA, 1991.

[La Valle et al., 1999] S. M. La Valle, J. H. Yakey, and L. E. Kavraki. A probabilistic roadmap approach for systems with closed kinematic chains. In *Proc. IEEE Int. Conf. Robot. and Autom.*,1999.

[Lin et al., 1997] M. Lin, D. Manocha, J. Cohem and S. Gottschalk. Collision detection: Algorithms and applications. *In Algorithms for Robotic Motion and Manipulation (WAFR9)*, JP. Laumond and M. Overmars (Eds), AK Peters 1997.

[Lozano-Pérez, 1983] T. Lozano-Pérez. Spatial planning: a configuration space approach. *IEEE Tr On Computers*, 32:108-120, 1983.

[Overmars 1994] M. Overmars. A random approach to path planning. Technical Report RUU-CS-92-32, Computer Science, Utrecht University, The Netherlands, 1992.

[Overmars and Svestka, 1995] M. Overmars and P. Svestka. A probabilistic learning approach to motion planning. In *Proc. Workshop on Algorithmic*

Foundations of Robotics, pages 19-37. A K Peters, Ltd., 1995.

[Siméon et al] T.Siméon, J-P. Laumond, C. Nissoux. Visibility-based Probabilistic Roadmaps for Motion Planning. LAAS-CNRS 7, avenue du Colonel-Roche 311077 Toulouse Cedex-France. Submitted to the Advanced Robotics journal.

[Vallejo et al., 2001] D. Vallejo, I. Remmler, N. Amato. An Adaptive Framework for "Single Shot" Motion planning: A Selft-Tuning System for Rigid and Articulated Robots. In *Proc. IEEE Int. Conf. Robot. and Autom.*, (ICRA), 2001.

[Wilmart et al., 1999] S. A. Wilmarth, N. M. Amato, and P. F. Stiller. Maprm: A probabilistic roadmap planner with sampling on the medial axis of the freespace. In *Proc. IEEE Int. Conf. Robot. and Autom*., Detroit, MI, 1999.