IAV2004 - PREPRINTS
5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles
Instituto Superior Técnico, Lisboa, Portugal
July 5-7, 2004

FAST APPROXIMATION TO GAUSSIAN OBSTACLE SAMPLING FOR RANDOMIZED MOTION PLANNING

N. Jouandeau and A. Ali Cherif

IARM, LIA, Paris8 University, Saint-Denis
{n,aa}@ai.univ-paris8.fr

Abstract:
We propose a new optimization of randomized motion planning via a local directed strategy. The basic motion planning problem is to find a collision free trajectory for a moving object (rigid, articulated or deformable) in a static or dynamic environment. We propose an improvement of the Rapidly-exploring Random Tree (*RRT*) method in associating the concepts of visibility and Gaussian sampling. This improvement focuses on the random sampling and its localization in free spaces. The new Gaussian sampling is described by a set of geometrical primitives and permits to define the random sampling behavior in the entire free space. In this paper, we first consider the existing alternatives for random sampling. Then we propose our localized random sampling that refines the environmental possibilities such as free space evaluation according to the mover's dynamic constraints. The environmental possibilities are identified during the *RRT* development. The experiments and results validate that our method improve the mover's trajectory in static environments.

Keywords: path and motion planning, computational geometry, static environment, random sampling

1. INTRODUCTION

Probabilistic planning methods have successfully shown their efficiency in resolving path planning problems (Amato and Wu, 1996; Hsu *et al.*, 1998; Kavraki and Latombe, 1998; LaValle, 1998; LaValle and Kuffner, 1999; LaValle and Kuffner, 2000). These methods have two advantages: dealing with a large number of problems and giving the possibility to use a robot with a high degree of freedom with probabilistic completeness. As Kavraki (Kavraki *et al.*, 1994) initially showed, various extensions of probabilistic methods are possible (Hsu *et al.*, 1998; Amato *et al.*, 1998; Boor *et al.*, 2001; Wilmarth *et al.*, 1999; LaValle, 1998; LaValle and Kuffner, 1999; Kuffner and LaValle, 2000; Branicky *et al.*, 2001; Caselli and Reggiani, 2000). They provide the same advantages and the same drawbacks due to a random sampling: planners are able to deal with the exponential complexity of the degree of freedom (*dof*) of the robot; planners' completeness seems to be guaranteed with a probability which tends asymptotically towards one when the time tends towards infinity. Then they are able to solve problems involving many *dof* like kinodynamic problems (Donald *et al.*, 1993). S. LaValle and J. Kuffner present a general method for kinodynamic planning in (LaValle and Kuffner, 1999).

In the following, we focus on the random sampling's localization. This reflexion is initiated by a knowledge of a well defined environment that allows us to define a fast approximation to Gaussian obstacle sampling. After a review of existing Rapidly-exploring Random Tree *(RRT)* methods, we describe our solution of bounding expansion with informations collected during the exploration. Results investigate the solution's performance and validate our motion planner with static environment.

1.1 Related Works

The *RRT* is a biased random walk in the state space. The *RRT* method is composed of two steps iterated until it provides a solution: the random step and the extend step. During the random step, a new state is randomly generated. This new state will guide the next extend step. During the extend step, the nearest neighbor of the new random state in the tree is selected. The random state provides the following direction which leads to the addition of a new state in the tree. Therefore the construction algorithm is associated to a collision detection function which determines if this new state is valid and then if it could be inserted in the tree. The *RRT* explores the free space with an uniform distribution which is assumed by random's linear congruence. The association of the *RRT* expansion with a Voronoï diagram shows that each step attempts to join unexplored regions. So *RRT* is naturally not working in the same way as a random walk. At each expansion step, every new state tries to break a new Voronoï cell. Largest cells have the highest probability to be broken. Based on this simple construction method, some problems occur iteratively : insuring the completeness; staying in the free space *Xfree* (which is equal to stay out of collision regions *(Xcol)* and imminent collision regions *(Xric)*); excluding non desired moves that the random walk could cause; managing the lack of simple metric able to select the nearest state in the tree; managing the convergence of the tree towards the goal.

In order to insure the completeness, P. Cheng and S. LaValle (Cheng and LaValle, 2002) suggest to compute the accessibility of the tree according to Lipschitz conditions. This is practical for an appropriate discretization resolution. The exploration function is controlled by a neighborhood analysis. This analysis maintains informations that exclude selected states. In this way, spin cycles are avoided and the exploration will certainly find a way to the goal. But this guarantee sacrifies the possible optimality of the path. If the tree grows up with a wrong starting sequence, the final path will be unexpectedly longer. Results show a successfully complete resolution involving kinematic and dynamic constraints with one steering input for a 9-dimensional nonlinear system.

The *Xfree* region is studied in (LaValle and Kuffner, 2001). S. LaValle and J. Kuffner show the differences between the configuration free space *C* and the state space X. After defining *Xobs* by assimilating it to *Cobs* in *X*, *Xric* is described as the *X*'s region of inevitable collisions, so the free state space *X* is reduced. Consequently the random possibility is also reduced, achieving at the same time an improvement of *RRT*. So undesired moves seem to be avoided by converging more rapidly through a solution trajectory.

The avoidance of undesired moves has found an expecting expansion in the bidirectional *RRT* (*bi-RRT*) (LaValle and Kuffner, 2001; LaValle and Kuffner, 1999; LaValle and Kuffner, 2000) (as a global *RRT* improvement, *bi-RRT* are currently mentioned *RRT*). In the context of differential constraints computing, it is difficult to define a solution trajectory between two states. Here the relationship between different states is defined and the possible attractiveness of *bi-RRT* is explained. *S1* and *S2* are two states. If a short sequence of control can be applied in *X* then it can reach *S2* starting from *S1*. *S1* is a fair state for *S2*. With a single *RRT*, it is important to provide fair states to reach the goal. If the neighbor space of the goal is narrowed, the single *RRT* may provide unfair states. In the worst case, it provides only unfair states, requiring then backward moves to return in a fair state. The *bi-RRT* is a way to avoid this by trying to achieve the vicinity of the starting position and the goal more efficiently. In *bi-RRT* planners, the expansion step is divided in two cases : the first case is the standard randomized expansion and the second one is a new tree connecting step. The first *RRT* construction veers off the second one, during which each one of the trees tries to grow into each other. Variations by using connection or extension step are achieved in *RRT-ConCon*, *RRT-ExtCon* and *RRT-ExtExt* (LaValle and Kuffner, 2000).

The *RRT* is very useful to determine if there is a feasible trajectory for a general constraints movers. But the trajectories then produced are often under optimal (with turnings or useless input's fluctuations). Cheng (Cheng *et al.*, 2000) reminds of two generalized solutions that solve such optimization problem: the first-order gradient descent and the perturbation introduction methods. In the first-order gradient method, starting from a given input sequence, a perturbation is iteratively introduced to converge towards a locally-optimal solution. This first solution is hardly applicable for models exceeding 3 *dof* and may be trapped in C_{Obs}. The second solution is based on the introduction of a disturbance during iterations; The problem then resides on the disturbance sources qualification and on these sources quantification along the trajectory. Tests are presented for a

Dubins car. Albeit the remaining problem is the definition of the disturbance sources, this method is good for its simplicity and its suitability with the *RRT*'s algorithm.

The efficiency of the distance metric is addressed at each extension step of *RRT*. Involving the growing behavior of the tree, the nearest state selection can improve the *RRT* connection. By rightly selecting the nearest neighbor state, this will be fairly reliable. A. Atramentov and S. LaValle have developed an approach based on *KdTree* (Atramentov and LaValle, 2002). Using a recursive subdividing, the nearest state search is associated to a shortest states list. It provides an algorithm for the nearest state search based on the comparison of distance between states and the Hausdorff distance. P. Cheng and S. LaValle reduce the metric sensitivity (Cheng and LaValle, 2001) with two solutions: introducing an exploring information and calculating the constraint violation probability (*CVP*). These two considerations are gathered during the expansion step. The efficiency of this *RRT* improvement depends on these exploration informations.

The reinforcement of converging toward the goal has a solution in *RRT-GoalBias* (LaValle and Kuffner, 2000). This is obtained by replacing the random with a probability biased toward the goal. The random distribution is no longer uniform. By introducing biases, the *RRT* will be trapped in some local minimum like in the randomized potential field methods (Latombe, 1991). An improvement called *RRT-GoalZoom* suggests to biase the random gradually around the goal. The convergence toward the goal is successfully done without considering the controllability influence in the *RRT* growing. To manage the convergence towards a solution trajectory, S. Carpin and E. Pagello (Carpin and Pagello, 2002) proposed a parallel formulation of motion planning to manage the convergence towards a solution trajectory. By increasing the number of processors, they increased at the same time the number of generated bi-trees starting from the same state and contributing to the same solution. They showed that a concurrent paradigm and a cooperative one have to be combined for more efficiency.

All investigations consider the first solution they found. It would be useful to improve the solution near the optimal, developing *RRTs* in the same space. When using *RRT* over a parallel computer, each processor is engaged in a distinct *RRT* with the same tuple start-end states. Therefore taking the best of all these iterations would improve the solution trajectory (Carpin and Pagello, 2001).

1.2 Our solution

This paper focuses on defining localized random sampling (*LR*). During its two previously defined basic steps (random and extend), the *RRT* explores free surrounding spaces. The traditional random explores uniformly free spaces. This property is fine in a space with uniform density. In a space made up of obstacles, the density is not uniform any more, and so this property becomes invalid. In the visibility based *PRM* method (Nissoux *et al.*, 1999), the visibility decreases the map size in charge and raises up the probability of narrow passages integration. We propose here a visibility based *RRT* to evaluate various open spaces. To increase the capture of narrow spaces, (Boor *et al.*, 2001) guides the *PRM* sampling with a Gaussian random. Therefore, we propose to use a visibility based localized random to improve the *RRT* algorithm.

For a localized random distribution G_{glob} around an obstacle, we define an uniform random sampling U_{loc} in a dimension D and a random sampling G_{loc} on the normal axis of each point of the same. In this G_{loc}, the length and the height are dynamically defined. The proportion between the height (with the length l) and the sum of all other heights L defines the random sampling G_{loc} on the obstacle axis. The height defines the ratio of the densities of various lengths. The dimension nature D can be for example the external perimeter or the whole surface of the random distribution. This distribution G_{loc} is carried outside the obstacles.

In a 2D space, the random sampling G_{glob} combines at the first dimension U_{loc} and at the second dimension G_{loc}. G_{glob} defines a surface. This surface is able to be divided to a set of identifiable surfaces by the points and the faces of each obstacle. Another random sampling U_{glob} assumes the equiprobability between all surfaces according to the parameter D. Therefore, the localized random distribution is defined. By applying a Gaussian distribution to G_{loc}, it evolves in a 2D space according to a function defined by:

$$f(x, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

with σ the standard deviation of the Gaussian one.

2. LOCALIZED RANDOM SAMPLING

In one of the simplest case, each obstacle is represented by a simple convex polygon and the random distribution is uniform. This distribution around a convex obstacle is a succession of right-angled trapezoids and curves. Each face is associated to a trapezoid and each vertex is associated to a curve. The parameters are identified with one vertex pair in the clockwise order. Therefore,

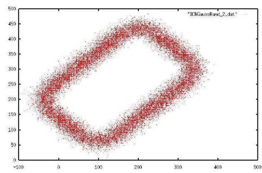

Fig. 1. Random sampling around a polygon.

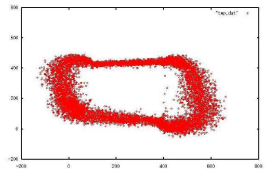

Fig. 2. Scaled random sampling around a polygon.

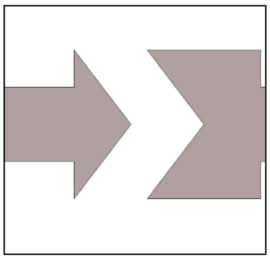

Fig. 3. A free space example.

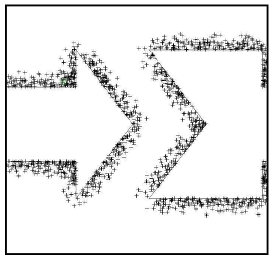

Fig. 4. Boor's random sampling.

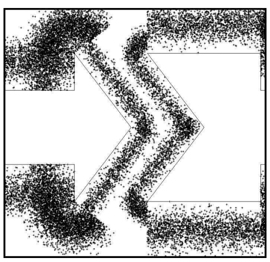

Fig. 5. Localized random sampling.

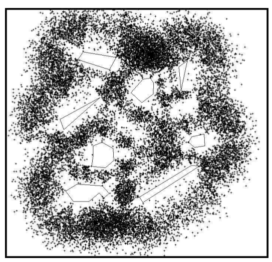

Fig. 6. Gaussian random sampling.

height values are computed to define the random shape. If two height values are associated to these two vertexes, we call this scaled sampling. If only one height value is associated to these two vertexes, this height value can be the minimum, the middle or the highest of the previously defined pair of height values. Instead of being uniform, this distribution can be Gaussian. Figures 1 and 5 show simple Gaussian distributions. Figure 2 shows a scaled Gaussian distribution. A Gaussian distribution associated with a face is defined by two values σ and two lengths of normals corresponding to this vertex pair. This distribution associated with a single vertex is defined by two values of σ, two lengths of normals and two angles. The Gaussian distribution G_{glob} is proportionally uniform for all these surfaces. In preoccupations with a continuity of the distribution G_{glob}, the values of the maximum associated with a face are fixed by the values of the parameters describing its face.

G_{glob} is also defined by two local parameters, p_i and p_f, which allow to position the band-width of the Gaussian one. p_i avoids the configurations in collision and p_f defines the external limit of the surface corresponding to the Gaussian distribution G_{glob}. In this way p_i draws aside random samplings of the vertex's obstacles. In order not to remove possible configurations of the free space, we choose to fix p_i to the half of the profile of the mobile for the non-holonomic robot. For holonomic robots, a good value for p_i is the smallest distance in each normal face's axis between each vertex and its gravity center. For a circular robot, if p_i is equal to the value of its ray, then any configuration in the random sampling is at the most cases apart from the collision's space.

In the basic G_{glob} distribution, open spaces are expressed with obstacles's faces. This is divided into a set of arcs and trapezoids in which G_{glob} distribution is applied. This decomposition can be partial or complete. It can be uniform or Gaussian, with or without p_i. Figure 5 shows a uniform sampling and figure 6 shows a Gaussian sampling. Figure 4 shows the Boor's Gaussian sampling as described in (Boor *et al.*, 2001). In these figures, we fixed the max height values for each shape. To avoid doubling the narrows spaces density, the height value is worth half of the distance between facing obstacles. This value has been fixed between minimum and maximum values.

We studied repercussions of the random sampling of the *RRT* growth, so that we can be able to compare various randomization's possibilities. During its growth, the *RRT* uses random configurations. The good configurations make the *RRT* evolving in the entire space; the bad ones lead to collision with obstacles. The classical random produces a significant number of collisions. The distribution G_{glob} produces much less collisions, due to its definition. The effective time for collision detection is much less significant by using G_{glob}.

Table 1. sampling times.

time	CR	CRCD	BR	LR
0.001	3 200	16	20	270
0.005	1 600	56	55	1 550
0.01	32 500	105	110	3 120
0.05	163 900	515	540	15 500
0.1	325 100	1 032	1 020	31 700
0.5	1 600 400	5 056	5 100	159 300
1.	3 250 700	10 169	10 200	318 600

Table 2. density influence for classical random sampling with collision detection.

time	CR1 in	CR1 out	CR2 in	CR2 out
0.001	10	6	21	3
0.01	66	41	168	23
0.1	640	392	1 631	226
1.	6 292	3 877	16 197	2 213

Table 1 presents the number of generated configurations according to time in seconds in the map presented in figure 3. *CR* is a classical random. *CRCD* is a classical random with collision detection. *BR* is a Boor's random. *LR* is a Localized random. For *LR*, being uniform or Gaussian, with or without p_i, with segments only or segments and arcs does not change significantly the resulting computational time. Obviously the classical random produces a great number of configurations, by including the collision detection the random G_{glob} produces even more. The Boor's random produces less configurations. Its number of generated configurations falls as its coverage is increased. Moreover the traditional random is dramatically sensitive to the space density. Table 2 shows the number of configurations generated in two cases. They show the average of two randomization cases: the first (*CR1in*) expresses the number of configurations in free space and *CR1out* expresses the number of configurations in collision. *CR1* is carried out on the figure 3. The second (*CR2*) is carried out on the figure 3 in broader plan. The number of generated configurations increases as well of the proportion of the good configurations.

3. VISIBILITY BASED *RRT*

To use the concept of visibility with the *RRT* method, two phases should be added: the first initializes the localized random to the initial configuration's visible free spaces; the second maintains the list of visible free spaces. During the first phase, it is necessary to create a visible segments list. The segments inserted to this list are marked so later inserts can be avoided. To maintain this list, we fix the Visibility Refresh Constant *VRC*. In a simple case, the segments are added during *RRT* growth. In the *RRT* method, starting from a new random configuration *Cfa* we select the closest configuration *Cfb* in the *RRT*. This configuration *Cfb* is then associated to a local control function. This function generates a new configuration *Cfc* which is the result of a mobile move starting from *Cfb* towards *Cfa*. Therefore the *RRT* is built by adding *Cfc*. To increase the *RRT* convergence towards unexplored spaces, the probability of progression is balanced by the number of *Cfc* configurations that it contains. Therefore every local shape defined in G_{glob} is not initially considered. Only visible trapezoids and curves are added to the sampling distribution. Other shapes are added only when they become visible.

For the GoalBias management, we propose to dynamically modify the policy of random sampling. Simple policies use a random function (*CR*, *BR* or *LR*) as previously defined. GoalBias policies use half a random function (*CR*, *BR* or *LR*) and half the goal. So half configurations are added towards the goal. If we want to use dynamically these two policies, we should know if the objective is visible following each new *Cfc* addition. An *RRT* is *GB* (GoalBias) if it uses permanently a deviation towards the goal as soon as it sees the goal. In the previous paragraph, the visibility add trapezoids and curves to the sampling distribution. The visibility is also used to bias the random sampling toward the goal.

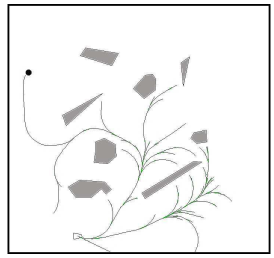

Fig. 7. RRT with visibility based Gaussian random sampling.

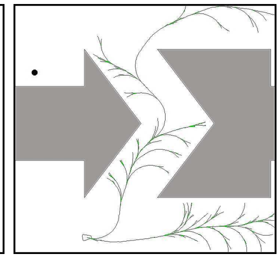

Fig. 8. unsuited growth with visibility based Gaussian sampling.

Table 3. random sampling results.

	success % (nb Cf)	failures % (nb Cf)
CR GB	0.04 (1163)	0.96 (1585)
CR VGB	0.21 (1713)	0.79 (2422)
BG GB	0.02 (600)	0.98 (485)
BG VGB	0.01 (628)	0.99 (332)
LR Uni GB	0.21 (1411)	0.79 (1650)
LR Gauss GB	0.29 (1141)	0.71 (1654)
LR Uni VGB	0.47 (1792)	0.53 (2500)
LR Gauss VGB	0.61 (1749)	0.39 (2514)

The table 3 shows results of different sampling policies in the map previously defined. figure 7, the mobile starts in the bottom left and must cross the narrow passage of the center to join the goal position on the top left. *CR* means Classical Random, *BR* Boor's random and *LR* Localized random, *GB* GoalBias, *VGB* Visible Goal Bias, *Uni* uniform and *Gauss* Gaussian. The left-hand column is the percentage of success and the column of right-hand side is the failure's one. For each column, the number of necessary configurations is noted between brackets. Without GoalBias, the percentage of success is quasi null. The resolving time is limited to 1,5 seconds. *VRC* has been fixed to 10. The model used is a 5-dimensional model presented in (Cheng and LaValle, 2001). The *RRT CR* believes uniformly in the space. In half cases it captures the narrow passage .Then it generate a great number of configurations leading to collisions in the narrow passage (so it justifies its percentage of 0.21 success). *RRT BR* not only generates few configurations but it also projects the *RRT* towards obstacles (so it leads to many collisions). The localization of the random allows better free spaces definition and so increases the *RRT*'s success chances. It maintains the *RRT* at a distance which is a function of free spaces evaluation. It also minimizes divergences in the *RRT*'s growth. Its disadvantage is the fact that it is based itself on the visible faces. Therefore it can involuntarily avoid the goal (as shown in figure 8).

4. CONCLUSION

This paper investigates the effect of localizing the random sampling with the performance of the *RRT* method. We have presented an improvement of the basic *RRT* method by using the concepts of visibility and Gaussian sampling. We proposed a new localized random sampling (*LR*) which is defined by a set of geometrical primitives. We define the random sampling behavior in the entire free space. Our localized random sampling allows better free spaces evaluation and increases the *RRT* success chances. It maintains the *RRT* at a distance from obstacles. This distance is a function of free spaces evaluation. It also minimizes divergences in the *RRT*'s growth according to free space evaluation and mover's dynamic constraints.

Acknowledgments: The authors would like thank anonymous reviewers from the International Program Committee of IAV2004 for their careful and considerate comments toward this work.

REFERENCES

Amato, N. M. and Y. Wu (1996). A randomized roadmap method for path and manipulation planning. In: *Int. Conf. on Robotics and Automation (ICRA96)*. pp. 113–120.

Amato, N., O. B. Bayazit, L. Dale, C. Jones and D. Vallejo (1998). Obprm: An obstacle-based prm for 3d workspaces. In: *Int. Workshop on Algorithmic Foundations of Robotics (WAFR98)*.

Atramentov, A. and S. M. LaValle (2002). Efficient nearest neighbor searching for motion planning. In: *Int. Conf. on Robotics and Automation (ICRA02)*.

Boor, V., M. H. Overmars and A. F. Van Der Stappen (2001). Gaussian sampling for probabilistic roadmap planners. In: *Int. Conf. on Robotics and Automation (ICRA01)*.

Branicky, M., S. Lavalle, K. Olson and L. Yang (2001). Quasirandomized path planning. In: *Int. Conf. on Robotics and Automation (ICRA01)*.

Carpin, S. and E. Pagello (2001). A distributed algorithm for multi-robot motion planning. In: *4th European workshop on advanced mobile robots (EUROBOT'01)*. pp. 207–214.

Carpin, S. and E. Pagello (2002). On parallel rrts for multi-robot systems. In: *8th Conf. of the Italian Association for Artificial Intelligence (AI*IA02)*. pp. 834–841.

Caselli, S. and M. Reggiani (2000). Erpp: An experience-based randomized path planner. In: *Int. Conf. on Robotics and Automation (ICRA00)*. pp. 1002–1008.

Cheng, P. and S. M. LaValle (2001). Reducing metric sensitivity in randomized trajectory design. In: *Int. Conf. on Intelligent Robots and Systems (IROS01)*.

Cheng, P. and S. M. LaValle (2002). Resolution complete rapidly-exploring random trees. In: *Int. Conf. on Robotics and Automation (ICRA02)*.

Cheng, P., Z. Shen and S. M. LaValle (2000). Using randomization to find and optimize feasible trajectories for nonlinear systems. In: *38th Annual Allerton Conf. on Communication, Control, and Computing*.

Donald, B. R., P. G. Xavier, J. F. Canny and J. H. Reif (1993). Kinodynamic motion planning. *Journal of the ACM* **40**(5), 1048–1066.

Hsu, D., L. Kavraki, J. Latombe, R. Motwani and S. Sorkin (1998). On finding narrow passages with probabilistic roadmap planners. In: *Int. Workshop on Algorithmic Foundations of Robotics (WAFR98)*.

Kavraki, L. and J. Latombe (1998). Probabilistic roadmaps for robot path planning. In: *Practical Motion Planning in Robotics: Current Approachesand Future Challenges*. pp. 33–53.

Kavraki, L., P. Svestka, J. C. Latombe and M. Overmars (1994). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. Technical Report CS-TR-94-1519.

Kuffner, J. and S. LaValle (2000). Rrt-connect: An efficient approach to single-query path planning. In: *Int. Conf. on Robotics and Automation (ICRA00)*.

Latombe, J.C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers.

LaValle, S. (1998). Rapidly-exploring random trees: A new tool for path planning. Technical Report 98-11. Dept. of Computer Science, Iowa State University.

LaValle, S. and J. Kuffner (1999). Randomized kinodynamic planning. In: *Int. Conf. on Robotics and Automation (ICRA99)*.

LaValle, S. and J. Kuffner (2000). Rapidly-exploring random trees: Progress and prospects. In: *Wokshop on the Algorithmic Foundations of Robotics (WAFR00)*.

LaValle, S. and J. Kuffner (2001). Randomized kinodynamic planning. Vol. 20. pp. 378–400.

Nissoux, C., T. Simeon and J. P. Laumond (1999). Visibility based probabilistic roadmaps. In: *Int. Conf. on Intelligent Robots and Systems (IROS99)*. pp. 1316–1321.

Wilmarth, S. A., N. M. Amato and P. F. Stiller (1999). A probabilistic method for rigid body motion planning using sampling from the medial axis of the free space. In: *15th Annual ACM Symp. on Comutational Geometry*. pp. 173–180.