Metrics for Comparing C-Space Roadmaps

Marco A. Morales A., Roger Pearce, Aimee Vargas E., Nancy M. Amato
{marcom,rap2317,aimee,amato}@cs.tamu.edu

Technical Report TR05-012
Parasol Lab
Department of Computer Science
Texas A&M University
College Station, TX 77843-3112

September 21, 2005

Abstract

There are many sampling-based motion planning methods that model the connectivity of a robot's configuration space (C-space) with a graph (or roadmap) whose nodes are valid configurations and whose edges represent sequences of valid configurations for moving between nodes. Many planning strategies have been proposed to construct roadmaps. Although their authors have discussed planner strengths and have presented experimental evaluations, the emphasis is usually put on performance traits. There is a need for metrics to strengthen qualitative analysis of planners. Roadmap comparison can be applied to study the evolution of a roadmap as it is built and to compare roadmaps made with different planners. We propose several metrics to compare roadmaps as a basis for analyzing the roadmap generation process and for comparing the relative strengths and weaknesses of different planning strategies.

1 Introduction

In the motion planning problem, a movable object has to move in an environment while not violating motion constraints (such as reaching invalid configurations). Since the motion planning problem is considered intractable [5, 18, 19], research on heuristic approaches has flourished [1–4, 9, 11, 12, 14, 16, 17, 20].

One of these approaches, the Probabilistic Roadmap Method (PRM) [11, 16, 17], has successfully solved complex problems with many degrees of freedom (DOFs) that other approaches cannot handle. PRMs address the motion planning problem by producing a graph (or roadmap) that represents the connectivity of the space of valid configurations for the robot (also called free C-space or C-free).

Traditionally, PRM roadmaps are made in two main steps: node generation and node connection. During node generation, robot configurations are randomly sampled from the C-space and tested for validity—in typical robotics applications through a collision detection test. Valid samples are kept as roadmap nodes. Then, during node connection, pairs of configurations are selected to attempt connections with deterministic local planners. The pairs that can be connected are stored as edges in the roadmap. The same roadmap can be applied to solve multiple queries requiring the robot to move between a pair of configurations.

Researchers have produced many strategies for node generation and node connection to try to produce better roadmaps at a lower cost. The analysis of these strategies usually involves performance evaluations (in terms of time or the number of validity tests, which is considered the primitive operation in PRMs) and basic qualitative evaluations (in terms of whether the roadmap produced can solve predefined queries). Using the existing comparison techniques, it is very difficult to answer the basic question of "Is roadmap A better than B?"

In this paper, we propose metrics to compare the relative quality of roadmaps in order to better study planning strategies and their parameters. We have categorized these evaluation metrics into four categories: Coverage, Connectivity, Efficiency, and Similarity. Each category captures important features in PRM roadmap building. Coverage metrics are concerned with how well the C-free of an environment is sampled. Connectivity metrics evaluate the connectivity of the C-free space. Efficiency metrics measure whether the samples in the map effectively expand discovered areas of the C-free. Similarity metrics show if two maps cover and connect the free space in a similar manner. The importance of each metric will vary depending on the application; for example, an analysis of a single shot planner will vary greatly from that of a multi-query planner.

Our main concern is identifying important roadmap attributes that we can use to derive conclusions about the relative strengths and weaknesses of the techniques used to construct them.

In this paper we propose four categories of metrics to compare roadmaps at several levels of interest. These metrics enable analysis as a means to better understand the different planning strategies so that we can better select them and tune their performance in different situations. We apply these metrics to study the evolution of a PRM roadmap during the exploration phase; this gives insight into how much sampling a PRM needs to achieve a satisfactory representation of the connectivity of the C-space. We also use the metrics to compare a variety of node generation methods, allowing us to study their relative strengths and weaknesses.

2 Related Work

Researchers in the motion planning community have used some metrics to analyze roadmap generation with an emphasis on performance. Among the metrics that have been studied we have:

- *Time:* computation time needed to generate a roadmap.
- *Validity checks:* the number of times a configuration is tested for validity (in many applications this is done through collision detection).
- *Nodes:* the number of nodes needed to generate a roadmap that is able to solve a query.

These metrics give useful information, and they have indeed been used in some studies to compare planners [6, 13]. However, these metrics leave out information that should be used in qualitative analysis of roadmaps.

2.1 Features of an Ideal Roadmap

In an ideal roadmap, every free configuration in the C-space should be connectable to a node in the roadmap and the roadmap should model the connectivity of the C-free. However, there are MP instances where such a roadmap is not likely to be obtained because the density, complexity, and distribution of C-obstacles reduces the probability of sampling valid configurations or making simple connections in some regions of C-space. This is the case in the narrow passage problem where relatively open areas are connected by a small-volume passage that runs through a dense and complex area. [8]

In order to model the connectivity of C-free, an ideal roadmap should have the same number of connected components as C-free. But PRMs limit the number of connection attempts for a node to a few selected nodes to reduce the number of validity tests that the local planners need to perform. This way, it is possible that some connections are missed.

One metric that has been used to make qualitative analysis of MP instances is ϵ-goodness [10]. A roadmap is ϵ-good if all its nodes can be connected to a volume that is at least ϵ times the volume of the C-free. Another qualitative test consists of querying the roadmap for selected paths of interest.

In [15] several metrics are introduced to characterize C-space. These metrics include the number of connected components, the percentage of successful connections, the percentage of valid samples and some analysis of roadmap subgraphs. These metrics can also be used in qualitative roadmap analysis.

3 Metrics to compare roadmaps

Metrics to compare roadmaps give a mechanism to better understand different sampling and connection strategies at different stages. With these metrics we can gain insight about the best use of each strategy. We are interested in metrics that can be extracted from multiple roadmaps. We can apply these metrics to analyze the progress of roadmap construction by taking "snapshots" at different steps of the process. We can also compare roadmaps made with different strategies to evaluate their relative ability to map different areas of the free space.

3.1 Definitions

- *Node Visibility:* Two nodes are *visible* if they can be connected using a selected deterministic local planner.

- *Usable Component:* A connected component in a roadmap *rdmp* is *usable* if it is larger than x% of the largest connected component in *rdmp*. This x is used to define a resolution for C-Free. The definition of component size is independent of the metrics introduced here.

- *Revealing Node:* Given a pair of roadmaps $(rdmp_a, rdmp_b)$, a *revealing node* is a node from a usable component in $rdmp_a$ that is not visible to any usable component in $rdmp_b$.

- *Trapped Node:* Given a pair of roadmaps $(rdmp_a, rdmp_b)$, a *trapped node* is a node from $rdmp_a$ that is not visible to any usable component in both roadmaps $(rdmp_a, rdmp_b)$.

- *Witness Nodes:* An independently obtained set of configurations in C-free.

- *Component Visibility:* Two connected components are *visible* if any two nodes in their respective components are visible.

- *Spanning Component:* Given a pair of roadmaps $(rdmp_a, rdmp_b)$, a *spanning component* is a component in $rdmp_a$ that is visible to two components in $rdmp_b$.

- *Node Similarity:* Given a pair of roadmaps $(rdmp_a, rdmp_b)$, a node in $rdmp_a$ is *similar* to a node in $rdmp_b$ if they are visible to each other.

- *Edge Similarity:* Given a pair of roadmaps $(rdmp_a, rdmp_b)$, an edge, $edge_i$, in $rdmp_a$ is *similar* to a sequence of one or more edges in $rdmp_b$, if both the start and end nodes of $edge_i$ are visible to the same component in $rdmp_b$. Note: this similarity does not consider path length, only that there exists a path represented by $edge_i$ in $rdmp_b$.

- *Path Similarity:* Given a pair of roadmaps $(rdmp_a, rdmp_b)$, a path between $node_a$ and $node_b$, represented some sequence of one or more edges in $rdmp_a$ is *similar* to a path in $rdmp_b$ if both $node_a$ and $node_b$ are visible to the same usable connected component in $rdmp_b$.

- *Roadmap Similarity:* For a pair of roadmaps $(rdmp_a, rdmp_b)$, the two roadmaps are *similar* if every node and path in $rdmp_a$ have a similar counterpart in $rdmp_b$, and vice versa.

3.2 Coverage Metrics

For a given roadmap *rdmp*, let us define $Coverage_{rdmp}$ as the ratio of *Witness Nodes* that are visible to some *usable* connected component in *rdmp* to the total number of *Witness Nodes*. This metric captures the coverage of *rdmp* in terms of independent C-free configurations; in a roadmap with ideal coverage, this will approach 1.

For a pair of roadmaps $(rdmp_a, rdmp_b)$, let us define $Coverage_{a\to b}$ as number of *Revealing Nodes* in

$rdmp_a$. A high value of $Coverage_{a \to b}$ means that $rdmp_a$ has explored (in terms of number of samples) a larger portion of C-free.

3.3 Connectivity Metrics

For a given roadmap *rdmp*, let us define $Connectivity_{rdmp}$ as the ratio of successful pairwise queries between *Witness Nodes* to the total number of potential pairwise queries. Unlike $Coverage_{rdmp}$, the $Connectivity_{rdmp}$ may not always approach 1; $Connectivity_{rdmp}$ is bound to the $Connectivity_{C-free}$.

For a pair of roadmaps $(rdmp_a, rdmp_b)$, let us define $Connectivity_{a \to b}$ as the number of *Spanning Components* in $rdmp_a$. This metric encompasses both Coverage and Connectivity traits; if $rdmp_a$ has high $Connectivity_{a \to b}$ relative to $Connectivity_{b \to a}$ then it has successfully explored and connected more C-free than $rdmp_b$.

3.4 Efficiency Metrics

For a given roadmap *rdmp*, let us define $Efficiency_{rdmp}$ as the ratio of the total number of nodes in *Usable Components* to the total number of nodes in *rdmp*. This metric penalizes exploration methods for generating nodes that are not visible or *trapped*. It measures how efficiently *rdmp* was created; low $Efficiency_{rdmp}$ may mean that in inappropriate exploration method was used for the given C-Space.

For a pair of roadmaps $(rdmp_a, rdmp_b)$, let us define $Efficiency_{a \to b}$ as the ratio of $Efficiency_a$ to the $Efficiency_b$.

3.5 Similarity Evaluation

For a pair of roadmaps $(rdmp_a, rdmp_b)$, let us define $Similarity_{rdmp_{a \to b}}$ as true if every node and path in $rdmp_a$ is similar to a node and path in $rdmp_b$, and vice versa.

4 Application: Analyzing Node Sampling Evolution

Roadmap comparison metrics can be used to study the evolution of roadmap construction strategies over time. To demonstrate the use of roadmap comparison metrics to effectively analyze roadmap evolution we analyze several Node Generation methods as they progress. At chosen time-steps, or after a fixed number of nodes have been generated, we study the progress made using a particular node generation method toward *convergence*. We define the *convergence* of a node generation strategy as the point where successive sampling fails to improve results. We show the evolution of some of the metrics defined at each time-step.

4.1 Selection of PRM Parameters

In order to study the evolution of a node sampling strategy we used the following PRM methods and parameters for all methods studied.

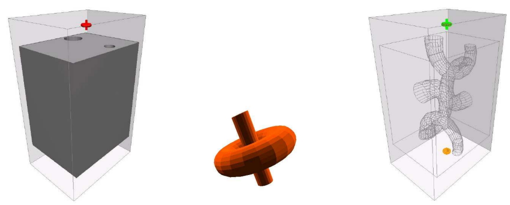
- *Node Visibility (local planner):* Two nodes are *visible* if they can be connected by a straight-line planner.
- *Roadmap Connection Method:* Connections were attempted between all pairs of roadmap nodes.
- *Usable Component:* A connected component in a roadmap *rdmp* is *usable* if it is larger than 1% of the largest connected component in *rdmp*.
- *Component Visibility:* Two connected components are *visible* if any two nodes in their respective components are visible.
- *Witness Nodes:* Randomly sampled with uniform distribution.

These parameters were set in an attempt to be treat all node sampling methods tested equally. To enable the comparison to concentrate on the nodes sampled, and to avoid biasing the results by the distance metric or connection strategy used, straight-line local planning was used in all cases and connections were attempted between all pairs of nodes at each iteration. To provide a time-line of evolution, we iteratively generate nodes until sampling convergence is achieved over some fixed iteration size. Unfortunately sampling convergence is difficult to predict given the high variance of some sampling strategies, thus finding the exact time of convergence is expensive to accurately find. However, this was not a concern as our main objective in this paper is to study the evolution of the roadmaps using different node generation strategies in different environments.

4.2 Experiment Setup

We applied different node generation methods to three different environments.

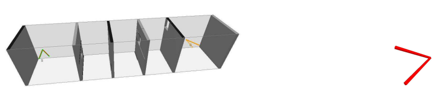
The *maze* environment is composed of a series of tunnels, some of them are dead ends. Figure 1 shows two perspectives of the same environment. The robot is a rigid body (shown in Figure 1(b)) that has to go from the top to the bottom part of the maze.



(a) (b) (c)

Figure 1: *maze* environment (a) solid view, (b) 6-DOF robot, (c) wire view

The *serial walls* environment is composed of five chambers divided by walls with small holes on them. The robot is an articulated object. The objective of the robot is to go through each chamber passing through each of the narrow passages. Figure 2(a) shows this environment. The robot is a 2 link manipulator with 7-DOF shown in Figure 2.



(a) *walls* environment (b) Robot with 7 DOF

Figure 2: *walls* environment

The *hook* environment is composed of two walls with narrow holes. The 6-DOF rigid body robot must pass through both narrow passages using translational and rotational motion. Figure 3 shows this environment.

We applied four sampling methods to each environment: BasicPRM [11], Bridge-Test [7], GaussPRM [4], OBPRM [1]. These methods were chosen to compare biased vs. unbiased sampling methods as well as the effects of using local C-Space information to guide sampling.

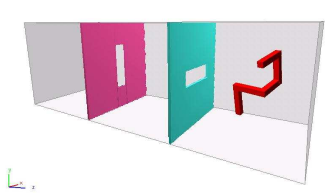


Figure 3: *hook* environment

Table 1: Node Generation Step Size

Method	Hook	Maze	Serial
BasicPRM	300	200	400
OBPRM	50	50	200
GaussPRM	200	50	100
Bridge-Test	150	50	50

For each method, nodes were sampled in an iterative fashion in ten independent runs using different seeds for the random number generator. In each run the step sizes shown in Table 1 were used. The step sizes were chosen to show ten steps before average convergence.

After each node generation step, we constructed a roadmap using the previously mentioned all pairs connection strategy with the straight-line local planner. We then computed a number of metrics for each roadmap. We analyzed the coverage of the current roadmap by trying to connect it to the *Witness Nodes* and its connectivity by trying to solve all possible queries defined by all pairs of *Witness Nodes*. Also we compared the connectivity of the current roadmap to the one from the previous step by measuring the number of *Revealing Nodes* of the roadmap in relation to the previous one. In a similar comparison, we measured the number of *Trapped Nodes*.

4.3 Analysis of Results

Figures 4, 5, 6 show plots of metrics computed in the hook, maze and serial environments, respectively.

Hook Environment. All the methods applied to the *hook* environment (Fig. 4) find more revealing nodes during the initial iterations compared to later iterations. The BridgeTest identifies many revealing nodes initially. OBPRM has a similar trend, but because it has a smaller step size than that of BridgeTest, it finds fewer revealing nodes at each time step. GaussPRM and BasicPRM are unable to find many revealing nodes until later iterations when they may find interesting areas to connect.

When analyzing the number of trapped nodes, we should recall that a node is considered trapped if it is not visible from a usable component, which is defined in these experiments as a component containing at least 1% of the total nodes. Since previously usable components can become unusable as the more nodes are added to the roadmap, and vice versa, we note that the trapped/untrapped characterization of a node can also change over time. The BridgeTest produces many trapped nodes after only three iterations, and then the number of trapped nodes fluctuates. Overall all iterations, OBPRM generated few trapped nodes, while both GaussPRM and BasicPRM increased the number of trapped nodes in their final iterations.

When considering the *Witness Queries* metric, both OBPRM and the BridgeTest show a steady and rapid increase in the number of queries that they can solve. GaussPRM is much slower in covering the space and BasicPRM keeps its coverage almost constant until the final iterations. For both these strategies, their success in solving queries is limited until their roadmaps connectivity spans the relevant regions of the planning space.

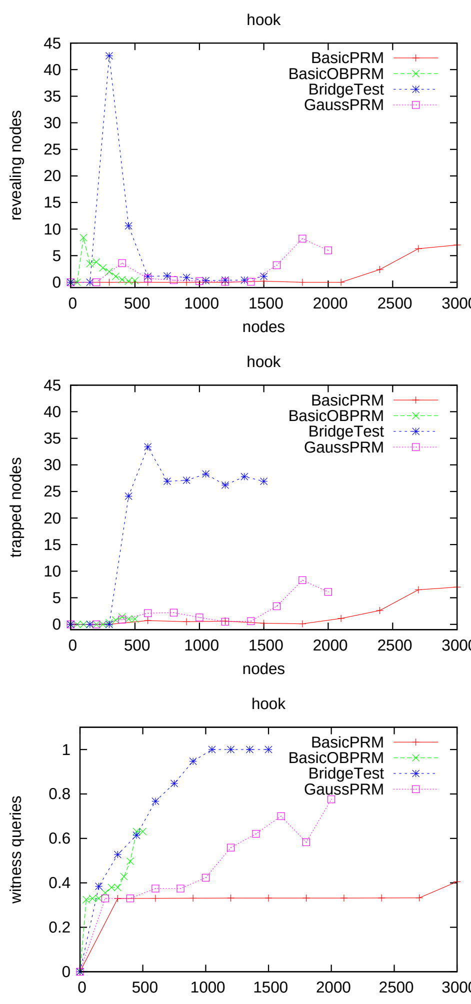


Figure 4: *hook* environment analysis

Maze Environment. The maze environment shows a smaller range of revealing nodes for all methods (see Fig. 5). OBPRM is able to generate more revealing nodes than the BridgeTest. GaussPRM finds fewer revealing nodes, but it has a trend similar to OBPRM and BridgeTest. BasicPRM is unable to find revealing nodes after the first two iterations.

GaussPRM finds few trapped nodes; OBPRM and BasicPRM find very few trapped nodes until later iterations whereas BridgeTest finds many. BridgeTest, GaussPRM, and OBPRM have similar patterns in solving witness queries for this environment, while BasicPRM's success is fairly constant because it has not succeeded in building an adequate roadmap.

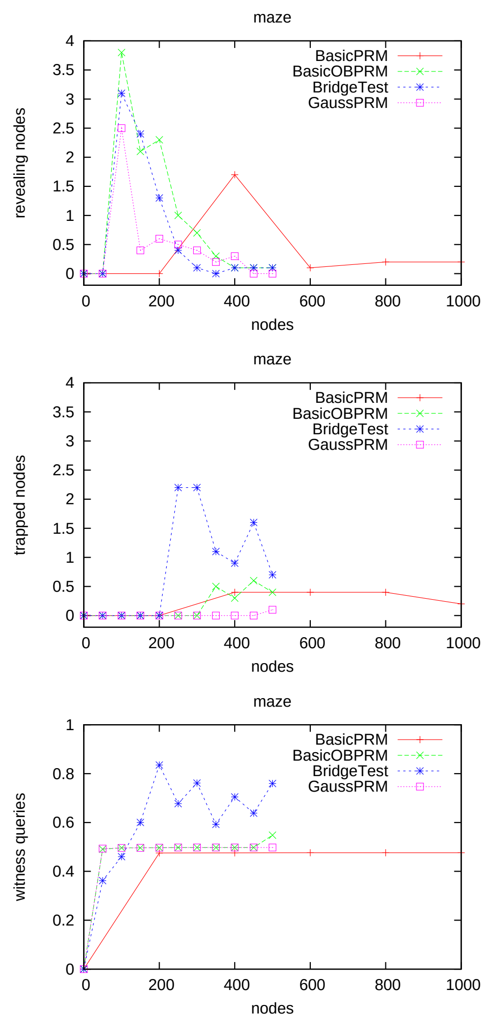
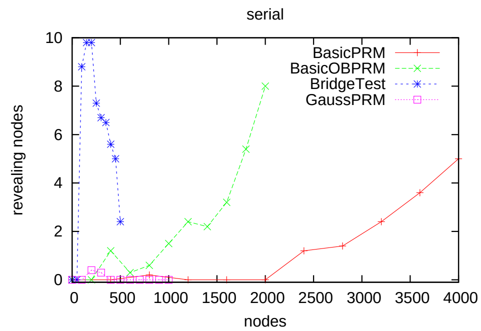


Figure 5: *maze* environment analysis

Serial Environment. In the serial environment, BridgeTest quickly finds it hard to increase its coverage although it continues to solve more witness queries in successive iterations. (see Fig. 6). On the other hand, we can see how OBPRM increases its ability to solve more witness queries as it increases its production of revealing nodes. In this experiment, due to time constraints, only BasicPRM was allowed to run for long enough to be able to solve all witness queries.



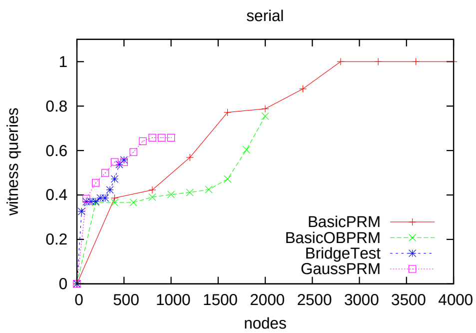


Figure 6: *serial* environment analysis

5 Application: Comparing Node Generation Methods

We use the proposed roadmap metrics to compare different node generation strategies. Using the same visibility and usable component setup as in Section 4.2 we compare OBPRM, MAPRM [20], Bridge-Test, GaussPRM and BasicPRM at two levels of map building. The *Witness Nodes* for these comparisons are 25 samples from each method for a total of 125 witness samples. First, we compare the node generation methods at 100 nodes per map (Table 2.) We then compare the maps at completion (Table 3).

5.1 Analysis of Results

Table 2 shows the comparison results for 100 samples from each of the methods applied to the Hook environment. While 100 samples may be too few to adequately evaluate and compare maps of different methods, they can give us insight into how well covered the C-free is in terms of *revealing nodes*. The first portion of the table is a general evaluation of each map. At 100 nodes, no planner has connected the three chambers of the Hook environment, this can be seen through the maximum value of 31% witness queries. The second portion of Table 2 shows the pairwise comparisons of each planner. At this small number of nodes, $Connectivity_{a \to b}$ gives us little insight into the maps because the sparse graphs are highly disconnected. At 100 nodes, we can see that MAPRM, BridgeTest and OBPRM have larger *Coverage* than BasicPRM, GaussPRM. We can also see that OBPRM's samples have covered the C-free more than BridgeTest, GaussPRM, BasicPRM and MAPRM. Lastly, we see that BridgeTest covers more than GaussPRM, MAPRM and BasicPRM.

Table 3 show the comparison results for each method applied to the Hook environment at either a converged map or a maximum sample of 5000 nodes. At this stage in the map development, we can see how the connectivity differs between the various planners. All methods except BasicPRM have converged, this leads to every planner finding *Spanning Components* in the BasicPRM roadmap as shown in the *Connectivity* metric. In this comparison we can see similar *Coverage* results as in the 100 sample version, although GaussPRM has now found more *Revealing Nodes* than BasicPRM. We can also see the high number of *Trapped* nodes created by Bride-Test, also observable in the results from Section 4.3.

In general, we see that in the problems studied here OBPRM achieved better coverage than other methods, and BridgeTest made many trapped nodes and unusable components.

6 Conclusion

In this paper, we propose using four types of metrics to evaluate qualitative traits in PRM roadmaps: Coverage, Connectivity, Efficiency, and Similarity. Using these metrics, we have studied the evolution of various PRM node generation strategies, and compared the strategies at different stages of map development. This allowed us to analyze traits that have not been studied previously. For example, we plotted *revealing node* production for different sampling strategies and problems. Also, in one of the environments we showed coverage metrics that are part of the similarity evaluation proposed.

Many of the Roadmap metrics proposed here can be used in an on-line fashion to test for convergence. In particular, the witness coverage and connectivity tests may be used given a small set of witness samples. Used in an iterative fashion, these metrics may

Table 2: Environment: Hook. Robot: 6 DOF rigid body

Method		Nodes	#CD NodeGen	#CCs	Usable #CCs	Witness connectable	Witness Queries
BasicPRM		100	154	5	5	95%	31%
OBPRM		100	5230	34	34	96%	31%
GaussPRM		100	1200	6	6	92%	29%
Bridge-Test		100	6230	31	31	91%	28%
MAPRM		100	7124	11	11	93%	29%
Node Generation Comparisons							
A	B	$Connectivity_{a \to b}$	$Connectivity_{b \to a}$	$Coverage_{a \to b}$	$Trapped_{a \to b}$	$Coverage_{b \to a}$*	$Trapped_{b \to a}$
Bridge-Test	GaussPRM	1	2	23	0	7	0
Bridge-Test	MAPRM	3	3	27	0	7	0
GaussPRM	MAPRM	1	1	1	0	6	0
OBPRM	Bridge-Test	4	6	31	0	20	0
OBPRM	GaussPRM	3	5	25	0	0	0
OBPRM	BasicPRM	1	3	25	0	0	0
OBPRM	MAPRM	5	6	26	0	4	0
BasicPRM	Bridge-Test	2	2	1	0	29	0
BasicPRM	GaussPRM	2	0	1	0	0	0
BasicPRM	MAPRM	1	0	2	0	7	0

give a good indication when adequate sampling and connection has occurred. We are exploring these issues in ongoing work.

References

[1] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Robotics: The Algorithmic Perspective*, pages 155–168, Natick, MA, 1998. A.K. Peters. Proceedings of the Third Workshop on the Algorithmic Foundations of Robotics (WAFR), Houston, TX, 1998.

[2] J. Barraquand and J. C. Latombe. Robot motion planning: A distributed representation approach. *Int. J. Robot. Res.*, 10(6):628–649, 1991.

[3] P. Bessiere, J. M. Ahuactzin, E. G. Talbi, and E. Mazer. The Ariadne's clew algorithm: Global planning with local methods. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, volume 2, pages 1373–1380, 1993.

[4] V. Boor, M. H. Overmars, and A. F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, volume 2, pages 1018–1023, 1999.

[5] J. F. Canny. *The Complexity of Robot Motion Planning.* MIT Press, Cambridge, MA, 1988.

[6] R. Geraerts and M. H. Overmars. A comparative study of probabilistic roadmap planners. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, December 2002.

[7] D. Hsu, T. Jiang, J.H. Reif, and Z. Sun. Bridge test for sampling narrow passages with proabilistic roadmap planners. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 4420–4426, 2003.

[8] D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 141–153, 1998.

[9] D. Hsu, R. Kindel, J. C. Latombe, and S. Rock. Randomized kinodynamic motion planning with moving obstacles. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages SA1–SA18, 2000.

[10] L. E. Kavraki, J.-C. Latombe, R. Motwani, and P. Raghavan. Randomized query processing in robot path planning. In *Proc. ACM Symp. Theory of Computing (STOC)*, pages 353–362, May 1995.

[11] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, 12(4):566–580, August 1996.

[12] J.-P. Laumond and T. Siméon. Notes on visibility roadmaps and path planning. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2000.

[13] S. M. LaValle and M. S. Branicky. On the relationship between classical grid search and probabilistic roadmaps. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, December 2002.

Table 3: Environment: Hook. Robot: 6 DOF rigid body

Method	Nodes	#CD NodeGen	#CCs	Usable #CCs	Witness connectable	Witness Queries
BasicPRM	5000	7774	11	3	99%	33%
OBPRM	575	30349	18	1	98%	96%
GaussPRM	1300	16072	19	1	96%	93%
Bridge-Test	1150	76360	238	2	98%	96%
MAPRM	2050	119389	15	1	98%	96%

Node Generation Comparisons

A	B	$Connectivity_{a \to b}$	$Connectivity_{b \to a}$	$Coverage_{a \to b}$	$Trapped_{a \to b}$	$Coverage_{b \to a}$	$Trapped_{b \to a}$
Bridge-Test	GaussPRM	0	0	8	185	13	8
Bridge-Test	MAPRM	0	0	5	184	26	8
GaussPRM	MAPRM	0	0	1	2	11	2
OBPRM	Bridge-Test	0	0	26	7	12	193
OBPRM	GaussPRM	0	0	9	2	2	1
OBPRM	BasicPRM	1	0	14	2	1	0
OBPRM	MAPRM	0	0	15	2	1	0
BasicPRM	Bridge-Test	0	1	5	2	9	172
BasicPRM	GaussPRM	0	1	2	0	4	3
BasicPRM	MAPRM	0	1	0	2	9	4

[14] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 473–479, 1999.

[15] M. Morales, L. Tapia, R. Pearce, S. Rodriguez, and N. M. Amato. A machine learning approach for feature-sensitive motion planning. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Utrecht/Zeist, The Netherlands, July 2004.

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

[17] M. Overmars and P. Svestka. A probabilistic learning approach to motion planning. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 19–37, 1994.

[18] J. H. Reif. Complexity of the mover's problem and generalizations. In *Proc. IEEE Symp. Foundations of Computer Science (FOCS)*, pages 421–427, San Juan, Puerto Rico, October 1979.

[19] J. T. Schwartz and M. Sharir. On the "piano movers" problem II: General techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math.*, 4:298–351, 1983.

[20] S. A. Wilmarth, N. M. Amato, and P. F. Stiller. MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, volume 2, pages 1024–1031, 1999.