TO APPEAR IN
IEEE International Conference on Robotics & Automation, 2003

The Bridge Test for Sampling Narrow Passages with Probabilistic Roadmap Planners

David Hsu*,1 Tingting Jiang† John Reif† Zheng Sun†

*Department of Computer Science
National University of Singapore
Singapore, 117543, Singapore
dyhsu@comp.nus.edu.sg

†Department of Computer Science
Duke University
Durham, NC 27708, USA
{ruxu, reif, sunz}@cs.duke.edu

***Abstract*— Probabilistic roadmap (PRM) planners have been successful in path planning of robots with many degrees of freedom, but narrow passages in a robot's configuration space create significant difficulty for PRM planners. This paper presents a hybrid sampling strategy in the PRM framework for finding paths through narrow passages. A key ingredient of the new strategy is the *bridge test*, which boosts the sampling density inside narrow passages. The bridge test relies on simple tests of local geometry and can be implemented efficiently in high-dimensional configuration spaces. The strengths of the bridge test and uniform sampling complement each other naturally and are combined to generate the final hybrid sampling strategy. Our planner was tested on point robots and articulated robots in planar workspaces. Preliminary experiments show that the hybrid sampling strategy enables relatively small roadmaps to reliably capture the connectivity of configuration spaces with difficult narrow passages.**

I. INTRODUCTION

During the past decade, probabilistic roadmap (PRM) planners [ABD+98], [BK00], [BOvdS99], [HLM99], [KŠLO96], [NSL99], [LK01] have emerged as a powerful framework for path planning of robots with many degrees of freedom (dofs). The main idea of a classic PRM planner [KŠLO96] is to sample at random a robot's configuration space to construct a network, called a *roadmap*, that captures the connectivity of the free space. PRM planners are both simple to implement and efficient, and thus have found many applications, including robotics, virtual prototyping, computer animation, and computational biology (see, *e.g.*, [ABG+02], [ADS02], [HLM99], [KL00], [LK01], [SLvGC01], [SLB99]).

Despite the success of PRM planners, path planning for many-dof robots is difficult. Several instances of the problem have been proven to be PSPACE-hard [HJW84], [Rei79], [SS83]. It is unlikely that random sampling, the key idea behind PRM planners, can overcome such difficulty entirely. Indeed, narrow passages in a robot's configuration space pose significant difficulty for PRM planners. Intuitively a narrow passage is a small region critical to the connectivity of the free space. We can also give formal characterizations [BKL+97], [HLM99] using the notion of *visibility sets*. To capture the connectivity of the free space accurately, a PRM planner must sample configurations in the narrow passages. This is difficult, because narrow passages have small volumes, and the probability of drawing random samples from small sets is low.

[1] Part of the work was completed while the author was at the University of North Carolina at Chapel Hill.

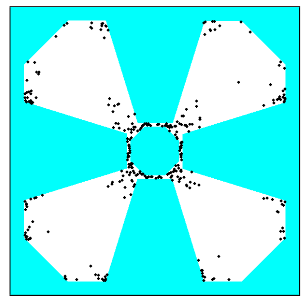


Fig. 1. An example of samples generated with the bridge test. In this and all later figures, black dots indicate sampled milestones, and shaded regions indicate obstacles.

In this paper, we propose a hybrid sampling strategy in the PRM framework in order to find paths through narrow passages efficiently. Key to our new strategy is the *bridge test*, which boosts the sampling density inside narrow passages and thus improves the connectivity of roadmaps. In a bridge test, we check for collision at three sampled configurations: the two endpoints and the midpoint of a short line segment s. We accept the midpoint as a new node in the roadmap graph being constructed, if the two endpoints are in collision and the midpoint is collision-free. We call this a bridge test, because the line segment s resembles a bridge: the endpoints of s, located inside obstacles, act as piers, and the midpoint hovers over the free space. Inside narrow passages, building *short* bridges is easy, due to the geometry of narrow passages; in wide-open free space, doing so is much more difficult. By favoring short bridges, we increase the chance of accepting configurations inside narrow passages (Fig. 1).

The bridge test uses only collision checking as a primitive operation and does not require complex geometric processing in the configuration space. It can be easily generalized to high-dimensional configuration spaces. It is also simple to implement and runs efficiently.

While being very effective in boosting the sampling density inside narrow passages, the bridge test severely reduces the sampling density in wide-open collision-free regions. This may be undesirable, because nodes in the roadmap need to cover the free space adequately [BKL+97]. Interestingly the difficulty encountered by the bridge test can be overcome by the uniform sampling strategy, which tends to place many samples in wide-open free space. The strengths of these two strategies complement each other naturally, and are combined with suitable weights to produce a hybrid sampling strategy to achieve better results. Our approach is related to the stratification methods for Monte Carlo integration [KW86].

The difficulty posed by narrow passages and its importance were noted in early work on PRM planners (see, *e.g.*, [KŠLO96]) and were later articulated in [HKL+98]. Several sophisticated sampling strategies can alleviate this difficulty, but a satisfactory answer remains elusive. One possibility is to sample more densely near obstacle boundaries [ABD+98], [BOvdS99], because configurations inside narrow passages lie close to obstacles. This approach admits a simple, efficient algorithm, the Gaussian sampler [BOvdS99]. However, many configurations near obstacle boundaries lie outside of narrow passages and do not help in improving the connectivity of roadmaps. So despite the improvement, sampling near obstacle boundaries may waste many samples in uninteresting regions. See Fig. 3 for a comparison with samples generated with the bridge test. In some special cases, the Gaussian sampler can be extended to reduce the number of wasted samples by paying a higher computational cost [BOvdS99]. Other approaches for sampling narrow passages include dilating the free space [HKL+98] and retracting to the medial axis of the free space [WAS99]. Both require complex geometric operations that are difficult to implement in high-dimensional configuration spaces. The visibility roadmap [NSL99] is related to the narrow passage problem. It tries to reduce the number of unnecessary samples by checking their visibility.

The rest of the paper is organized as follows. Section II gives an overview of our planner. Sections III and IV describe and analyze the bridge test, and show how to combine it with uniform sampling to produce the hybrid sampling strategy. Section V reports experimental results. Section VI discusses alternatives to some choices made in our current planner. Section VII summarizes the main results and points out direction for future research.

II. Overview of the planner

A classic multi-query PRM planner proceeds in two stages. In the first stage, it tries to construct a roadmap graph G that captures the connectivity of the free space $\mathcal{F}$. The nodes of G are randomly sampled points from $\mathcal{F}$, called *milestones*. There is an edge between two milestones if they can be connected via collision-free canonical paths, typically, straight-line segments. A good roadmap G has two properties. First, the set of milestones in G *covers* the free space well. In other words, for every point $p \in \mathcal{F}$, there is a collision-free straight-line segment between p and a milestone in G with high probability. Second, there is an edge in G between two milestones q and q', if and only if q and q' lie in the same connected component of $\mathcal{F}$. After constructing the roadmap, the planner searches it for a collision-free path between two given query configurations in the second stage. In this paper, we address only the first stage, roadmap construction. Methods for the second stage are well-known [KŠLO96], [ABD+98].

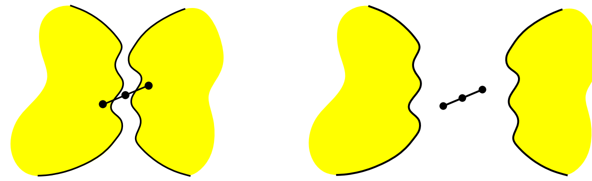

(*a*) (*b*)

Fig. 2. Building short bridges is much easier in narrow passages than in wide-open free space.

Our goal is to build a good roadmap by sampling a small number of well-placed milestones. To obtain milestones in narrow passages, we pay a higher cost per milestone than simpler strategies such as uniform sampling; however, our roadmap size is often much smaller, thus saving lots of time in checking whether collision-free straight-line paths exist between pairs of milestones. The trade-off is well worthwhile as shown by our experiments (see Section V).

The sampling distribution that we use is a weighted mixture of π_{B}, the distribution generated by the bridge test, and π_{U}, the uniform distribution. We describe how to construct π_{B} and combine the two distributions in the next two sections. After generating the milestones, for every pair of milestones close to each other, we check whether a collision-free straight-line segment exists between them. If so, we insert an edge between them into the roadmap.

III. The bridge test

Narrow passages in a free $\mathcal{F}$ are small regions critical in preserving the connectivity of a roadmap built in $\mathcal{F}$. It is difficult to sample in narrow passages because of their small volumes. Any sampling distribution based on the volumes is likely to fail. In particular, the uniform distribution does not work well. Furthermore, when dealing with many-dof robots, we do not have an explicit representation of configuration space $\mathcal{C}$ and cannot locate narrow passages directly by processing the global geometry of $\mathcal{C}$.

The bridge test is designed to boost the sampling density inside narrow passages using only simple tests of local geometry. It is based on the following observation. A narrow passage in an *n*-dimensional configuration space has at least one direction v, in which the robot's motion is very restricted. Small perturbation of the robot's configuration along v results in collision with obstacles. The robot is free to move only in those directions perpendicular to the restricted ones. Therefore, for a collision-free point p in a narrow passage, it is easy to sample at random a short line segment s through p such that the endpoints of s lie in obstacles in $\mathcal{C}$ (Fig. 2*a*). The line segment s is called a *bridge*, because it resembles a bridge across the narrow passage, with the endpoints of s acting as piers and the point p hovering over the free space. We say that a point $p \in \mathcal{F}$ passes the bridge test, if we succeed in building a bridge through p. Clearly building *short* bridges is much easier in narrow passages than in wide-open free space (Fig. 2). By favoring short bridges over longer ones, we increase the chance of accepting points in narrow passages.

Sampling milestones. To sample a new milestone using the bridge test, we pick a line segment s from $\mathcal{C}$ at random by choosing its endpoints and determine whether s passes the bridge test. If so, we insert the midpoint of s into the roadmap G as a new milestone. The details are shown in Algorithm 1, which is called Randomized Bridge Builder (RBB). RBB calls the function CLEARANCE to determine whether a point in $\mathcal{C}$ is collision-free.

Algorithm 1 Randomized Bridge Builder (RBB).

repeat
  Pick a point x from $\mathcal{C}$ uniformly at random.
  if CLEARANCE(x) returns FALSE **then**
    Pick a point x' in the neighborhood of x according to a suitable probability density λ_x.
    if CLEARANCE(x') returns FALSE **then**
      Set p to be the midpoint of line segment $\overline{xx'}$.
      if CLEARANCE(p) returns TRUE **then**
        Insert p into G as a new milestone.

To perform the bridge test, RBB uses only one geometric primitive, CLEARANCE, which can be implemented very efficiently using a collision detection algorithm (see, *e.g.*, [Qui94], [GLM96]). The bridge test is purely local and does not require processing the global geometry of $\mathcal{C}$.

RBB pays a higher cost to obtain a milestone than simpler strategies such as uniform sampling, because it accepts a milestone only if a sampled point passes the bridge test, which makes three calls to CLEARANCE each. However, RBB generates milestones in narrow passages critical in capturing the connectivity of the free space, resulting in much smaller roadmaps; lots of computation time is saved in checking whether collision-free paths exist between pairs of milestones, a much more expensive operation than the simple collision check CLEARANCE.

Choosing the probability density λ. The density function λ determines how frequently a particular bridge is chosen for a test. Short bridges are preferred over longer ones in order to increase the probability of sampling in narrow passages. We choose λ_x to be a Gaussian with its center at x and the same small standard deviation σ for each dimension of $\mathcal{C}$. If we have *a priori* information on the narrow passages, then there may be other distributions more suitable than this radially symmetric Gaussian. See Section VI for further discussion.

Analysis of the sampling distribution. One may wonder: what does π_{B}, the probability density created by RBB, look like? To calculate π_{B}, let us first define X and X' to be two random variables, representing respectively the two endpoints of a bridge. The first endpoint X is distributed uniformly over the set of configuration-space obstacles $\mathcal{B} = \mathcal{C} \backslash \mathcal{F}$. So the density $f_X(x)$ is non-zero if and only if x lies in $\mathcal{B}$. Assume, without loss of generality, that $\mathcal{B}$ has volume 1. Then $f_X(x)$ is 1 if $x \in \mathcal{B}$ and 0 otherwise. Given $X = x$, we choose the other endpoint X' according to the density λ_x. The point X' is accepted only if it lies in $\mathcal{B}$. Let I be a binary function such that for any point $p \in \mathcal{C}$, $I(p) = 1$ if $p \in \mathcal{B}$ and 0 otherwise. The conditional density of X' given X is given by

$$f_{X'|X}(x' \mid x) = \lambda_x(x') I(x') / Z_x,$$

where $Z_x = \int_{\mathcal{C}} \lambda_x(x') I(x')\, dx'$ is a normalizing constant. To calculate π_{B} at a point $p \in \mathcal{F}$, we condition on X:

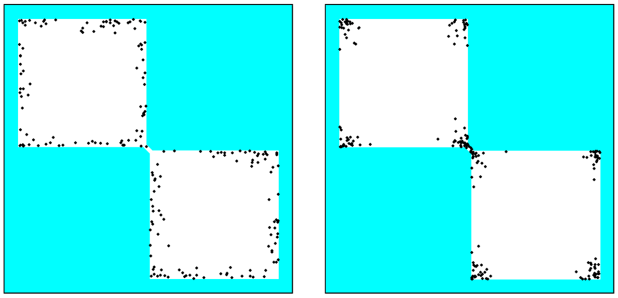
$$\pi_{\mathrm{B}}(p) = \int_{\mathcal{C}} f_{X'|X}(x' \mid x) f_X(x)\, dx. \tag{1}$$

Note that p is the midpoint of the line segment $\overline{xx'}$ and so $x' = 2p - x$. Substituting the expressions for f_X, $f_{X'|X}$, and x' into (1), we have

$$\pi_{\mathrm{B}}(p) = \int_{\mathcal{B}} \lambda_x(2p - x) I(2p - x) / Z_x\, dx. \tag{2}$$

Recall that λ_x is a Gaussian with its center at x and a small standard deviation. The density λ_x is large if $x' = 2p - x$ lies close to x. Furthermore, the integrand in (1) is non-zero only if $I(2p - x) = 1$, *i.e.*, $x' \in \mathcal{B}$. For a point p in a narrow passage, both conditions are more likely satisfied, resulting in a large value for π_{B} at p.

Comparison with sampling near obstacle boundaries. RBB is related to the Gaussian sampler [BOvdS99]. Both use one simple geometric primitive CLEARANCE to create favorable distributions. Their objectives, however, are quite different. RBB increases the sampling density inside narrow passages; the Gaussian sampler increases the sampling density near obstacle boundaries. RBB is slightly more expensive: it makes one more call to CLEARANCE per sample than the Gaussian sampler. However, by focusing on narrow passages, RBB gains efficiency by avoiding sampling uninteresting obstacle boundaries that do not contribute to improving the connectivity of roadmaps.



(a) (b)

Fig. 3. The samples generated by (a) the Gaussian sampler and (b) RBB.

See Fig. 3 for a comparison between these two sampling strategies. In some special cases, an extension of the Gaussian sampler can reduce the number of wasted samples by picking a triple instead of a pair of points and checking that two of the three picked points lie in *different* obstacles [BOvdS99]. It is unclear how well this extension works in general, *e.g.*, in a narrow passage formed by one single non-convex obstacle.

The idea of sampling near obstacle boundaries fails when the boundaries are uninteresting. The bridge test may fail, too, though less often. This happens when $\mathcal{F}$ contains sharp corners, because near the tip of a corner, it is easy to build short bridges. In Fig. 3(b), RBB generated a number of milestones in the six corners of $\mathcal{F}$. These samples are unhelpful. Nonetheless, our experiments suggest that the benefits gained by sampling in narrow passages outweigh the computation time wasted in sampling near sharp corners (see Section V).

IV. Combining complementary sampling distributions

We have seen that the bridge test is effective in boosting the sampling density in $\mathcal{P}$, the subset of $\mathcal{F}$ occupied by narrow passages. The density $\pi_{\rm B}$ is heavily biased towards $\mathcal{P}$. At the same time, $\pi_{\rm B}$ penalizes wide-open collision-free regions: few points are sampled in $\mathcal{F}\backslash\mathcal{P}$. This may be undesirable, because a good roadmap must cover the entire free space adequately.

Interestingly we can make up the deficiency of $\pi_{\rm B}$ with the uniform distribution $\pi_{\rm U}$, which samples $\mathcal{F}$ with probability proportional to the volumes of subsets in $\mathcal{F}$. For $\pi_{\rm U}$, most samples fall into $\mathcal{F}\backslash\mathcal{P}$. The two sampling distributions complement each other: $\pi_{\rm U}$ provides good coverage of $\mathcal{F}\backslash\mathcal{P}$, and $\pi_{\rm B}$ samples more densely in $\mathcal{P}$ and thus improves the connectivity of the roadmap. They are combined to produce a hybrid sampling distribution:

$$\pi = (1-w)\cdot\pi_{\rm B} + w\cdot\pi_{\rm U}, \tag{3}$$

where w is a weight, with $0 \le w \le 1$. The choice of w depends on the difficulty of sampling in narrow passages and the number of milestones needed to cover $\mathcal{F}$. The best choice depends on the specific problem. Currently we set w manually to favor $\pi_{\rm B}$, because we assume that $\mathcal{F}$ contains at least some difficult narrow passages. We intend to conduct more experiments to determine the range of w that works well for typical problems.

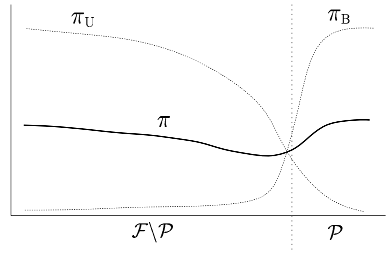


Fig. 4. The hybrid sampling distribution π. The distributions $\pi_{\rm B}$ and $\pi_{\rm U}$ perform well on $\mathcal{P}$ and $\mathcal{F}\backslash\mathcal{P}$, respectively. Combining them with suitable weights leads to good performance over the entire sampling domain.

One fruitful way of thinking about this hybrid distribution π is to divide the free space $\mathcal{F}$ into two subsets, the narrow passages $\mathcal{P}$ and its complement $\mathcal{F}\backslash\mathcal{P}$. We use a different sampling strategy tailored to each subset to achieve good performance over the entire sampling domain. See Fig. 4 for an illustration. This approach is related to the stratification methods for Monte Carlo integration [KW86] and the multiple-importance sampling for ray-tracing photo-realistic images [VG95].

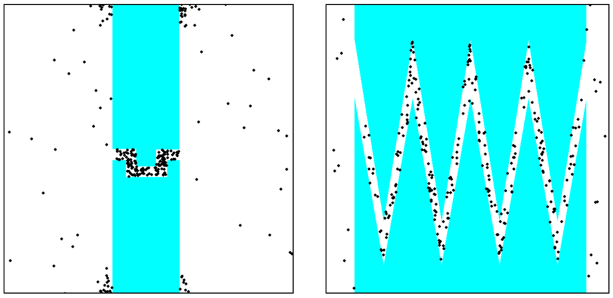
The significance of a hybrid distribution is not about putting together two distributions, but rather about identifying distributions complementary in their strengths and combining them so that their individual strengths are preserved. Our approach differs from the previous work (*e.g.*, [DA01]) in that the two sampling distributions $\pi_{\rm B}$ and $\pi_{\rm U}$ naturally complement each other. No computation is necessary to explicitly decompose the sampling domain.

To implement the hybrid distribution, we can certainly generate new random points from $\pi_{\rm U}$, but actually we can get at least some of these points "for free" by reusing the points rejected in line 3 of Algorithm 1.

V. Implementation and Experiments

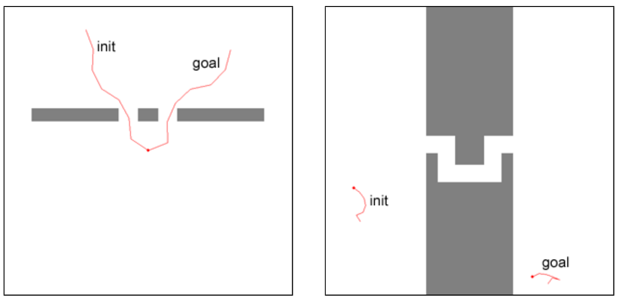
To test the hybrid sampling strategy, we applied it to both a point robot and articulated robots in planar environments. Preliminary experiments indicate that our planner is able to efficiently capture the connectivity of free spaces containing difficult narrow passages.

Implementation details. Two parameters need to be chosen for our hybrid sampling strategy. First, for RBB, we chose the density function λ to be an independent Gaussian for each dof of a robot, with a small standard deviation σ to bias towards sampling short bridges. In our experiments, we set σ to be roughly 10% of the smallest allowable range of motion among all dofs. Making σ too small may adversely impact the performance of the planner. The reason is that if a bridge is too short, the second endpoint



(*a*) (*b*)

Fig. 5. Environments used for testing our planner.



(*a*) (*b*)

Fig. 6. Experiments with (*a*) a 7-dof articulated robot with a fixed base and (*b*) an 8-dof articulated robot with a mobile base.

of the bridge would be unable to cross the narrow passage and fall in $\mathcal{F}$, causing many potentially useful points to fail the bridge test. The second parameter is the weight for combining $\pi_{\rm B}$ and $\pi_{\rm U}$. We use the ratio 5:1 in favor of $\pi_{\rm B}$, meaning that for every five milestones generated from $\pi_{\rm B}$, we pick one milestone from $\pi_{\rm U}$.

Our program was implemented in Java, and the results reported below were acquired from a PC with a 1.8GHz Pentium 4 processor.

Experimental results. We first tested our planner on a point robot in planar environments, containing different kinds of narrow passages. Environment A (Fig. 1) consists of four chambers connected by multiple narrow corridors. To go from one chamber to the diagonally opposite one, the point robot must pass through two narrow corridors. Environment B (Fig. 3) contains a very narrow and short corridor that connects two large square chambers. The corridor in environment C (Fig. 5*a*) is longer and has multiple turns. So each milestone in the corridor has low visibility and covers only a small portion of the free space. Environment D (Fig. 5*b*) contains a very long narrow corridor. It illustrates an interesting scenario in which RBB and the Gaussian sampler behave similarly, because almost every point in the narrow passages lies close to obstacle boundaries and *vice versa*.

We also performed preliminary experiments with planar articulated robots. Environment E contains a seven-dof

TABLE I

PERFORMANCE STATISTICS OF DIFFERENT SAMPLING STRATEGIES.

Env.	Sampler	$N_{\rm mil}$	$N_{\rm clear}$	$N_{\rm con}$	Time (sec.)
A	uniform	773	1859	3184	0.44
	RBB	46	4071	266	0.08
	hybrid	63	4543	384	0.09
B	uniform	675	1710	2685	0.81
	RBB	16	3040	46	0.03
	hybrid	22	3836	75	0.04
C	uniform	566	963	2556	0.19
	RBB	359	47372	1674	0.44
	hybrid	69	6680	515	0.06
D	uniform	111	315	839	0.03
	RBB	101	2255	873	0.04
	hybrid	98	1776	951	0.04
E	uniform	3200	4677	37270	325
	RBB	1092	735056	20206	148
	hybrid	1164	638421	19303	174
F	uniform	13699	24005	83170	1431
	RBB	1091	806759	23290	193
	hybrid	384	231305	3146	38

$N_{\rm mil}$: number of milestones in the resulting roadmap
$N_{\rm clear}$: number of calls to CLEARANCE
$N_{\rm con}$: number of calls to check collision-free connection between two milestones

articulated robot with a fixed base (Fig. 6*a*). At both the initial and goal configurations, the robot is trapped in narrow openings and must execute difficult maneuvers in order to find a path. Fig. 6*b* shows environment F. The workspace is similar to that in environment C, but the robot is an articulated robot with six links and a mobile base, eight dofs in total.

For each of the environments A–D, we generated 30 random queries that require the point robot to go through narrow passages. For environment E and F, we handpicked the queries. We then performed 10 independent runs for each query. We terminated the planner as soon as a path was found between query configurations, and recorded the running times and other statistics. For comparison, we performed the same experiments with both pure RBB (without mixing with uniform sampling) and hybrid sampling. We also used uniform sampling as a way to calibrate the relative difficulty of queries. The results for each environment and sampling strategy are averaged and reported in Table I. Note that the running times were acquired from a Java implementation. So the relative performance is more important than the absolute values of running times.

Table I shows consistent results from experiments on different robots and environments. Hybrid sampling is usually the best performer in terms of running times. Although it rejects more samples and makes more calls to CLEARANCE, it produces a roadmap with a smaller number of milestones and thus greatly reduces the time in checking that pairs of milestones can be connected via collision-free straight-line segments. In general, such a connection check is much more expensive than a call to CLEARANCE. So hybrid sampling was able to achieve good overall performance.

For some queries, pure RBB performs as well as hybrid sampling. However, when the initial and goal configurations lie in wide-open free space (environment C and F), pure RBB performs much worse, because it places almost all samples inside narrow passages and does not cover the free space well. In such cases, pure RBB may need to use a larger σ to cover the free space.

The only exception to our general observations is environment D. As expected, all three sampling strategies generated roadmaps of comparable sizes to answer the queries. However, since hybrid sampling and RBB made more calls to CLEARANCE, they took slightly longer.

VI. Discussion

In designing the bridge test, the choice of the density function λ is important. So far we have assumed that λ is a radially symmetric Gaussian, which works well if a robot's dofs are all symmetric, *e.g.*, a point robot. The symmetry breaks down on free-flying rigid-bodies and articulated robots, for which each dof must be scaled to reflect its influence on the global movement of the robot. To deal with the problem, we assign a different Gaussian standard deviation to each dof. This simple extension has already been implemented for planar articulated robots in our experiments.

Furthermore, one may question whether there are other density functions better than the Gaussian. In practice, we may have some estimates on the width of narrow passages. For instance, if a robot is stuck in a narrow corridor, it has a rough idea of how much room there is to maneuver based on its knowledge of the environment. Let us assume that the width of the narrow passage is roughly c. Then it is useless to sample bridges much shorter than c, because the bridge test is bound to fail, as explained in Section V. So λ should have a shape that peaks at roughly c and decreases quickly to 0 as the length of the bridge gets much shorter or longer.

VII. Conclusion and future work

We have presented a new sampling strategy in the PRM framework for finding paths through narrow passages. A key ingredient of the new strategy is the bridge test, which boosts the sampling density inside narrow passages. The bridge test makes use of only one geometric primitive, which checks whether a configuration is collision-free. The bridge test is purely local and can be implemented efficiently in high-dimensional configuration spaces. The strengths of the bridge test and the uniform sampling are complementary. We combine them to obtain a hybrid sampling strategy that generates small roadmaps that cover the free space well and have good connectivity. In our preliminary tests on a point robot and articulated robots with up to eight dofs, our planner was able to reliably capture the connectivity of free spaces with difficult narrow passages.

Several interesting issues regarding the bridge test and the hybrid sampling strategy require further exploration.

We are conducting additional experiments with high-dof planar articulated robots to better understand the strength and weakness of our planner. We are also implementing the planner for free-flying rigid bodies in 3-D workspaces. Based on our experience with PRM planners and that of other researchers, we are confident that the new planner will perform well in 3-D workspaces.

We would also like to further develop the hybrid sampling strategy. The important issues here are to identify sampling distributions that are naturally complementary and do not require explicitly decomposing the sampling domain, and to find a systematic weight assignment method that preserves the strengths of individual distributions.

VIII. Acknowledgements

This work is partially supported by DARPA/AFSOR contract F30602-01-2-0561 and NSF grants EIA-0218376, EIA-0218359, 11S-01-94604, EIA-0086015. Hsu was supported by the NSF ITR grant CCR-0086013.

IX. References

[ABD+98] N.M. Amato, O.B. Bayazit, L.K. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In P.K. Agarwal et al., editors, *Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics*, pages 155–168. A. K. Peters, Wellesley, MA, 1998.

[ABG+02] M.S. Apaydin, D.L. Bruglag, C. Guestrin, D. Hsu, and J.C. Latombe. Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion. In *Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, pages 12–21, 2002.

[ADS02] N.M. Amato, K.A. Dill, and G. Song. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. In *Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, pages 2–11, 2002.

[BK00] R. Bohlin and L.E. Kavraki. Path planning using lazy PRM. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 521–528, 2000.

[BKL+97] J. Barraquand, L.E. Kavraki, J.C. Latombe, T.-Y. Li, R. Motwani, and P. Raghavan. A random sampling scheme for path planning. *Int. J. Robotics Research*, 16(6):759–774, 1997.

[BOvdS99] V. Boor, M.H. Overmars, and F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 1018–1023, 1999.

[DA01] L.K. Dale and N.M. Amato. Probabilistic roadmaps—putting it all together. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 1940–1947, 2001.

[GLM96] S. Gottschalk, M. Lin, and D. Manocha. OBB-Tree: A hierarchical structure for rapid interference detection. In *SIGGRAPH 96 Conference Proceedings*, pages 171–180, 1996.

[HJW84] J.E. Hopcroft, D.A. Joseph, and S.H. Whitesides. Movement problems for 2-dimensional linkages. *SIAM J. on Computing*, 13(3):610–629, 1984.

[HKL+98] D. Hsu, L.E. Kavraki, J.C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In P.K. Agarwal et al., editors, *Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics*, pages 141–154. A. K. Peters, Wellesley, MA, 1998.

[HLM99] D. Hsu, J.C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *Int. J. Computational Geometry & Applications*, 9(4 & 5):495–512, 1999.

[KL00] J.J. Kuffner and J.C. Latombe. Interactive manipulation planning for animated characters. In *Proc. Pacific Graphics*, 2000.

[KŠLO96] L.E. Kavraki, P. Švestka, J.C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration space. *IEEE Trans. on Robotics & Automation*, 12(4):566–580, 1996.

[KW86] M.H. Kalos and P.A. Whitlock. *Monte Carlo Methods*, volume 1. John Wiley & Son, New York, 1986.

[LK01] S.M. LaValle and J.J. Kuffner. Randomized kinodynamic planning. *Int. J. Robotics Research*, 20(5):278–400, 2001.

[NSL99] C. Nissoux, T. Siméon, and J.-P. Laumond. Visibility based probabilistic roadmaps. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1316–1321, 1999.

[Qui94] S. Quinlan. Efficient distance computation between non-convex objects. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3324–3329, 1994.

[Rei79] J.H. Reif. Complexity of the mover's problem and generalizations. In *Proc. IEEE Symp. on Foundations of Computer Science*, pages 421–427, 1979.

[SLB99] A.P. Singh, J.C. Latombe, and D.L. Brutlag. A motion planning approach to flexible ligand binding. In *Proc. Int. Conf. on Intelligent Systems for Molecular Biology*, pages 252–261, 1999.

[SLvGC01] T. Siméon, J.P. Laumond, C. van Geem, and J. Cortes. Computer aided motion: Move3D within MOLOG. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 1494–1499, 2001.

[SS83] J.T. Schwartz and M. Sharir. On the 'piano movers' problem: II. General techniques for computing topological properties of real algebraic manifolds. *Advances in Applied Mathematics*, 4:298–351, 1983.

[VG95] E. Veach and L.J. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *SIGGRAPH 95 Conference Proceedings*, pages 419–428, 1995.

[WAS99] 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. on Robotics & Automation*, pages 1024–1031, 1999.