In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2004

Workspace Importance Sampling for Probabilistic Roadmap Planning

Hanna Kurniawati David Hsu

Department of Computer Science
National University of Singapore
Singapore, Republic of Singapore
{hannakur, dyhsu}@comp.nus.edu.sg

***Abstract*— Probabilistic Roadmap (PRM) planners have been successful in path planning of robots with many degrees of freedom, but they behave poorly when a robot's configuration space contains narrow passages. This paper presents *workspace importance sampling* (WIS), a new sampling strategy for PRM planning. Our main idea is to use geometric information from a robot's workspace as "importance" values to guide sampling in the corresponding configuration space. By doing so, WIS increases the sampling density in narrow passages and decreases the sampling density in wide-open regions. We tested the new planner on rigid-body and articulated robots in 2-D and 3-D environments. Experimental results show that WIS improves the planner's performance for path planning problems with narrow passages.**

I. Introduction

In recent years, probabilistic roadmap (PRM) planning has emerged as one of the most successful approaches for path planning of robots with many degrees of freedom (dofs) [1], [6], [7], [10], [12], [16], [18], [19]. A classic multi-query PRM planner [16] samples points uniformly at random in a robot's configuration space, and connects these points with simple collision-free paths to construct a roadmap graph that approximates the connectivity of a robot's configuration space. The planner then answers path-planning queries by searching the roadmap for a path between query configurations. Due to their efficiency and simplicity, PRM planners have found applications in many areas, such as robotics, computer-aided design, computer graphics, and computational biology (see, *e.g.*, [2], [4], [8], [17]).

Despite their successes, PRM planners behave poorly when a robot's configuration space contains narrow passages [12], [16]. A narrow passage is a small region whose removal changes the connectivity of the configuration space. The probability of sampling points at random in narrow passages is low, because narrow passages have small volumes. This makes it difficult for PRM planners to capture the connectivity of the configuration space well, when narrow passages are present.

In this paper, we propose a new sampling strategy to address the narrow passage problem in PRM planning. The intuition is that narrow passages in a robot's configuration space are often caused by narrow passages in the workspace. We decompose the workspace via tetrahedralization to locate workspace narrow passages and use the geometric information from the workspace as heuristic importance values to guide sampling in the corresponding configuration space. Our goal is to increase the probability of sampling free configurations in narrow passages, which is critical for capturing the connectivity of the free space, and to decrease the probability of sampling free configurations in wide-open collision-free regions. We call this new sampling strategy *workspace importance sampling* (WIS), because it is related to importance sampling used in Monte Carlo integration [15].

In the rest of the paper, Section II briefly reviews related work. Section III gives an overview of our planner. Section IV describes WIS in detail. Section V reports experimental results. Section VI summarizes our main results and gives direction for future research.

II. Related Work

Several PRM planners have been proposed to deal with the narrow passage problem, including dilated free space [13], OBPRM [1], and MAPRM [24]. Unfortunately, these planners require geometric computation that is expensive to implement in high dimensional configuration spaces. The Gaussian sampler [7] and the hybrid bridge test [14], which rely heavily on rejection sampling, are much simpler to implement in high dimensional configuration spaces. However, their performance degrades when the narrow passages have extremely small volumes, which lead to high rejection ratio.

Instead of performing geometric computation on the configuration space, some earlier work has explored the idea of using workspace information. For example, to compute the motion for a rigid-body robot, one may construct the medial axis of the workspace and then samples configurations that place the robot close to the medial axis [11]. Another possibility is to use the medial axis to find a path for a reference point on the rigid-body robot and then "refine" the path to obtain the full translational and rotational motion of the robot [10]. Of course, the refinement may not always succeed. Inspired by the watershed segmentation method from image processing, the recent work of van den Berg and Oversmars tries to identify large collision-free regions in the workspace and label the space connecting such regions as narrow passages [23]. It then adjusts the sampling distribution accordingly. This method is effective when the narrow passages are short and are located between two large collision-free regions.

Our planner uses workspace information, too. It computes a tetrahedralization of the workspace and uses the tetrahedralization to locate narrow passages and guide sampling in the configuration space.

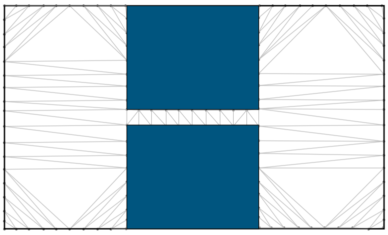


Fig. 1. A Delaunay triangulation of $\mathcal{W}'$ in a 2-D workspace.

III. Preliminaries on PRM Planning

The *configuration* of a robot is a set of parameters that uniquely determines the position of every point on the robot. The set of all configurations forms the *configuration space* $\mathcal{C}$, and a configuration is represented as a point in $\mathcal{C}$. A configuration q is *collision-free* or *free* if the robot placed at q does not collide with obstacles in the workspace or with itself. The free configurations form a subset $\mathcal{F}$ of $\mathcal{C}$.

Like most multi-query PRM variants, our planner consists of two phases, a pre-computation phase and a query phase. In the pre-computation phase, the planner constructs a roadmap graph G to capture the connectivity of $\mathcal{F}$. It incrementally samples a set of collision-free configurations, called *milestones*, from $\mathcal{C}$ using WIS. For every milestone q obtained, it inserts q into G as a new node and then checks whether q can be connected with nearby existing milestones via collision-free straight-line paths. If such a path exists between two milestones, the planner adds an edge between them in G. In the query phase, the planner is given an initial and a goal configuration. It tries to connect the two query configurations to two corresponding milestones in G and then searches for a path in G between these two milestones, using standard graph search algorithms. See [16] for more details on PRM planning.

IV. Workspace Importance Sampling

Narrow passages in $\mathcal{C}$ are small regions critical for preserving the connectivity of $\mathcal{F}$. Removing these small regions may change the way different parts of $\mathcal{F}$ are connected or even the number of connected components. To capture the connectivity of $\mathcal{F}$ well, it is important for PRM planners to sample free configurations in narrow passages. This is difficult, because narrow passages have small volumes. Any volume-based sampling strategy is unlikely to perform well. Furthermore, in general, we do not have an explicit representation of $\mathcal{C}$ and cannot process its geometry easily to locate the narrow passages.

However, narrow passages in the configuration space $\mathcal{C}$ are often caused by narrow passages in the workspace $\mathcal{W}$. The main idea of WIS is to locate narrow passages in $\mathcal{W}$ and use this information to guide sampling in the corresponding regions of $\mathcal{C}$. Locating narrow passages in $\mathcal{W}$ is simpler because we have an explicit geometric representation of the 2-D or 3-D workspace.

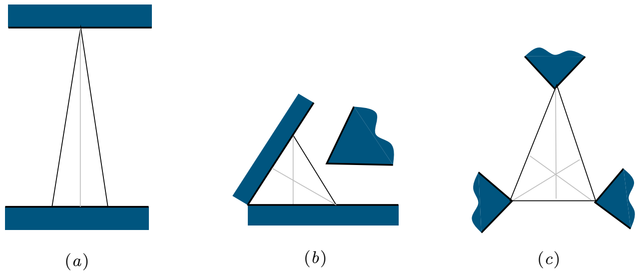


Fig. 2. Defining the importance value $h(t)$ of a triangle t in a 2-D workspace. There are three cases: (a) one edge, (b) two edges, and (c) no edge on the boundary of $\mathcal{W}'$.

A. Locating Workspace Narrow Passages

Let $\mathcal{W}'$ be the subset of $\mathcal{W}$ that is not occupied by obstacles. To locate narrow passages in $\mathcal{W}$, we would like to compute a tetrahedralization of $\mathcal{W}'$ by treating $\mathcal{W}'$ as a polyhedron. However, it is known that not all polyhedra can be tetrahedralized [22]. To avoid this difficulty, we sample points at a fixed resolution on the boundary of $\mathcal{W}'$, using an algorithm similar to scan conversion in computer graphics [9], and compute a Delaunay tetrahedralization $\mathcal{T}$ over the set of sampled points. See Fig. 1 for an example in a 2-D workspace. If the sampling resolution is sufficiently high, then under reasonable geometric assumptions, $\mathcal{T}$ is conformal in the sense that every face on the boundary of $\mathcal{W}'$ is a union of faces in $\mathcal{T}$ [3], and hence every tetrahedron in $\mathcal{T}$ is in either $\mathcal{W}'$ or its complement. Let $\mathcal{T}'$ denote the subset of all tetrahedra in $\mathcal{W}'$.

Next we assign an importance value $h(t)$ to every tetrahedron t in $\mathcal{T}'$. Fig. 1 seems to indicate that tetrahedra in narrow passages have small sizes, but because of bad tetrahedra (*e.g.*, slivers and skinny tetrahedra), neither volume nor edge length is a good measure of size here. Instead, we use the average height of t to define the importance $h(t)$. A tetrahedron t has four heights h_i for $i = 1, 2, 3, 4$, each corresponding to a face f_i of t. Only those heights that give an estimate of the local "width" of $\mathcal{F}$ are relevant. We thus define the importance $h(t)$ as follows:

- If t has one or more faces lying on the boundary of $\mathcal{W}'$, then
$$h(t) = \frac{\sum_{i=1}^{4} \beta_i h_i}{\sum_{i=1}^{4} \beta_i},$$
where β_i is 1 if f_i lies on the boundary of $\mathcal{W}'$ and 0 otherwise.
- If t has none of its faces lying on the boundary of $\mathcal{W}'$, then $h(t) = \sum_{i=1}^{4} h_i/4$.

See Fig. 2 for illustrations of the corresponding definition in 2-D. A small value of $h(t)$ indicates that t likely lies in a narrow passage, and more effort is needed to sample the corresponding region in $\mathcal{C}$. According to this definition, a skinny tetrahedron t, like the one shown in Fig. 2a, has a large value of $h(t)$. This indicates that t is not inside a narrow passage.

We now use the importance values to sample the configuration space of two common types of robots, rigid-body robots and articulated robots in 3-D workspaces.

B. Rigid-Body Robots in 3-D Workspaces

The configuration of a rigid-body robot consists of a positional component q_τ, which specifies the position of a reference point on the robot, and an orientational component q_θ, which specifies the orientation of the robot.

To sample a new milestone, we pick a tetrahedron t uniformly at random from $\mathcal{T}'$, and then sample a maximum of n_t times in the region of configuration space that corresponds to t, until we obtain a free configuration. The positional component q_τ and the rotational component q_θ are sampled independently. For q_τ, we pick a point uniformly at random from t. For q_θ, we sample an orientation in 3-D uniformly at random using quaternions [21].

The maximum number n_t of trials for sampling a free configuration from a tetrahedron t depends on $h(t)$, the importance of t. Assume that the probability p of getting a milestone, *i.e.*, a free configuration, for t is proportional to $h(t)$. The probability α of getting a milestone after n_t trials is given by

$$\alpha = 1 - (1-p)^{n_t}. \tag{1}$$

Now we normalize $h(t)$ and set $p = h(t)/h_{\text{total}}$, where $h_{\text{total}} = \sum_t h(t)$. It follows from (1) that

$$n_t = \frac{\ln(1-\alpha)}{\ln(1 - h(t)/h_{\text{total}})}. \tag{2}$$

The parameter α reflects our eagerness in obtaining one milestone for each tetrahedron, or the confidence level that there is at least one milestone that can be obtained for each tetrahedron.

A complete description of the algorithm is given in Algorithm 1. This algorithm calls a function IsFree(q) to determine whether a configuration q is collision-free. IsFree is implemented using a hierarchical collision detection algorithm [20].

The goal of WIS is to bias the sampling distribution towards narrow passages. It tries to achieve this by sampling the tetrahedra uniformly at random. Tetrahedra inside narrow passages tend to have small volumes. Thus, there are more tetrahedra per unit volume in narrow passages than in other regions, thereby increasing the sampling density inside narrow passages. However, this is not enough. Depending on the robot's size and shape, it may be more difficult to obtain a free configuration for a tetrahedron in a narrow passage. Therefore, we sample n_t times for each tetrahedron t chosen, with n_t depending on the importance value $h(t)$.

Algorithm 1 WIS for rigid-body robots.

Let $\mathcal{W}'$ be the subset of $\mathcal{W}$ that is not occupied by obstacles. Sample with sufficient density a set of points S on the boundary of $\mathcal{W}'$. Compute a tetrahedralization $\mathcal{T}'$ of S.
For each tetrahedron t in $\mathcal{T}'$, calculate n_t based on the importance $h(t)$ of t.
loop
  Pick a tetrahedron t uniformly at random from $\mathcal{T}'$.
  MaxNumTrials $\leftarrow n_t$.
  NumTrials $\leftarrow$ 0.
  repeat
    Pick the positional component q_τ by sampling a point uniformly at random from t.
    Pick the orientational component q_θ by sampling an orientation in 3-D uniformly at random.
    $q \leftarrow (q_\tau, q_\theta)$.
    NumTrials $\leftarrow$ NumTrials + 1.
  until IsFree(q) or NumTrials $\geq$ MaxNumTrials.
  if a collision-free configuration q is obtained **then**
    Add q to the roadmap G as a new milestone.
    for every milestone q' in G within a distance r from q, where r is a fixed constant **do**
      Check whether q and q' can be connected via a collision-free straight-line path. If so, add an edge between q and q' in G.

C. Articulated Robots in 3D Workspaces

For common articulated robots, such as PUMA and FANUC arms, WIS works similarly as it does for rigid-body robots, but the way it samples the robot's configuration space is slightly different. For an articulated robot, the positional component q_τ specifies the joint angles that determine the position of the robot's wrist point, and the orientational component q_θ specifies the orientation of the wrist. A point sampled from a tetrahedron t only specifies the position of the robot's wrist point and not q_τ. Hence, after sampling a position x of the robot's wrist point uniformly at random from a tetrahedron t, we need to solve the robot's inverse kinematics equations to get q_τ that places the robot's wrist point at x. If the number of inverse kinematics solutions is large or infinite, WIS samples a few at random. It then samples q_θ by picking uniformly at random the joint angles that control the orientation of the wrist.

D. Running Time

The total running time of WIS consists of three parts: the time T_t for tetrahedralizing $\mathcal{W}'$, the time T_s for sampling milestones, and the time T_c for checking collision-free connections between milestones in the roadmap. Compared with other PRM variants, WIS pays the additional cost of sampling the boundary of $\mathcal{W}'$ and computing the tetrahedralization. Suppose that there are n sampled points on the boundary of $\mathcal{W}'$. The cost of tetrahedralization is $O(n^2)$ in the worst case, but in practice, we can expect $O(n \lg n)$ [3]. In addition, it is well-known that T_c usually dominates the total running time of PRM planners. By paying a relatively small cost of tetrahedralization, we reduce the number of milestones needed and hence the time T_c for connecting the milestones, thereby reducing the total running time.

E. Limitations

WIS uses only workspace information to locate narrow passages and ignores the geometry of the robot. This sometimes leads to false positive results. Some regions of the workspace with high importance values are so narrow that the robot simply cannot pass through. Sampling densely in these regions is clearly counter-productive. A possible improvement is to try to roughly estimate the smallest passage that the robot can pass through.

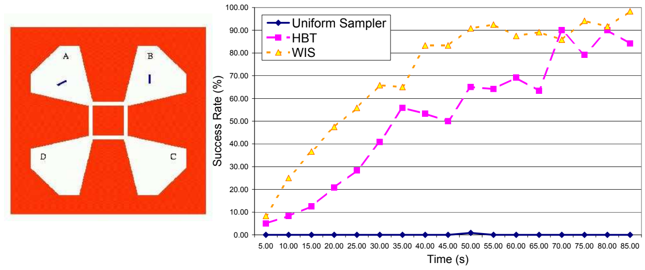
Furthermore, WIS currently uses workspace information for sampling the positional component of the configuration space only. It uses uniform sampling for the orientational component. When the robot has a complex shape and must reorient frequently in order to wiggle through narrow passages, the performance of WIS suffers, because more milestones are needed to take care of orientation changes. To deal with this problem, one possibility is to sample more milestones for tetrahedra with high importance values. Another possibility is to analyze the geometry of the robot and combine this information with workspace information to bias sampling in the orientational component as well.

V. Implementation and Experiments

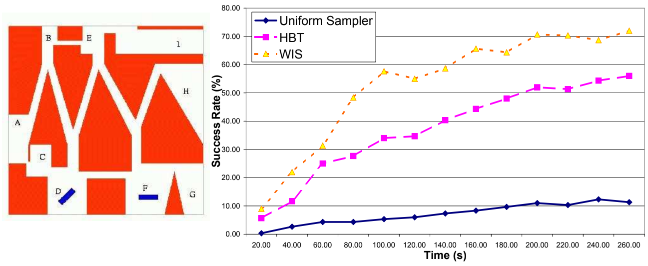
We implemented WIS and tested it on both rigid-body and articulated robots. For each test environment, we must determine the resolution for sampling the boundary of $\mathcal{W}'$ in order to compute a tetrahedralization of $\mathcal{W}'$. We set the resolution to be high enough so that under reasonable geometric assumptions, the resulting tetrahedralization is conformal. The tetrahedralization is computed using the Quickhull algorithm [5]. During the roadmap construction, we try to connect each new milestone q to other existing milestones whose normalized distance to q is less than 0.15.

For comparison, we also implemented two PRM planners using the uniform sampler [16] and the hybrid bridge test (HBT) [14]. The uniform sampler was used as a reference to measure the performance improvement of WIS and HBT. HBT was chosen because it has shown good performance for path planning problems with narrow passages. For HBT, we used $1 : 1$ as the relative weight for mixing two component samplers—the uniform sampler and the bridge test. In other words, the component samplers were equally weighted in HBT. We also performed trial runs to set the parameters of the bridge test so that HBT had good performance.

We ran the three planners on a number of different test environments. For each environment, we manually specified one or more queries such that solving these queries indicates that a roadmap captures the connectivity of the free space well. Each test run was repeated 30 times independently, and the results were averaged. For each test environment below, we show a graph that plots the average percentage of queries that a planner can answer correctly as the running time increases. The measured running time includes all the time needed for preprocessing, such as sampling $\mathcal{W}'$ and building a hierarchical representation for collision detection. Our program was implemented in C++. The tests were performed on an Intel P4 1.6GHz PC with 256MB RAM.

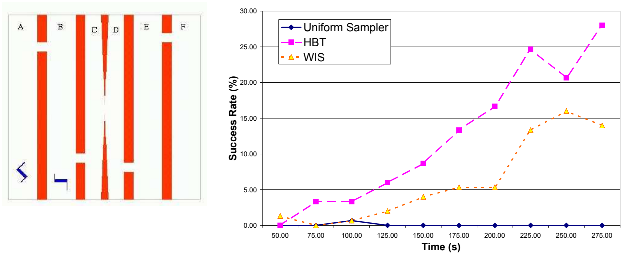


Test 1: This planar workspace consists of four chambers connected with narrow passages. The robot is a rigid body that translates and rotates freely with three dofs in total. There are four specified queries: (A, B), (B, C), (C, D), and (D, A), where A, B, C, and D are the corresponding workspace regions of the query configurations (see the above figure). WIS outperforms HBT in this test, because the workspace information helps WIS to locate narrow passages more efficiently.



Test 2: This workspace consists of many regions connected with narrow passages. The robot is again a rigid body with three dofs. There are ten specified queries: (A, B), (A, C), (B, D), (D, C), (D, E), (D, F), (E, F), (E, I), (F, G), and (F, H). To capture the correct connectivity of the free space, a planner must sample from all regions of the free space. WIS performs well here, because the tetrahedralization helps to distribute the samples over all the regions and cover the free space well.

Although there are false positive narrow passages between B and E and between A and C, their effects here are not significant. The reason is that the width of these passages is similar to that of the true narrow passages. As a result, the weights of the triangles inside the false positive regions are similar to those inside the true narrow passages, and the false positive regions are not significantly over-sampled. Therefore, they do not affect the overall performance significantly in this case. Of course, in other workspaces, the effects of false positives could be much more significant.



Test 3: Here the workspace consists of six vertical chambers connected with narrow openings. The robot is a rigid body with three dofs in total. There are five specified

queries: (A, B), (B, C), (C, D), (D, E), and (E, F).

HBT performs better than WIS in this case, because the robot has a relatively complex shape and must execute several maneuvers to reorient and wiggle through the narrow openings. As we have mentioned in Section IV-E, WIS samples orientations uniformly, and is less effective when complex reorientations are required. See the circled regions in Fig. 3 for a comparison of the three planners.

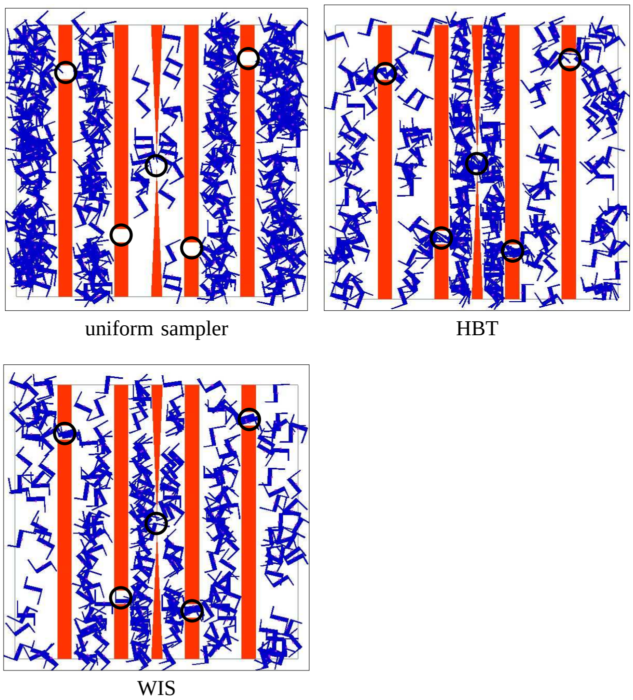
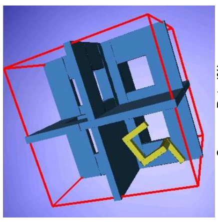
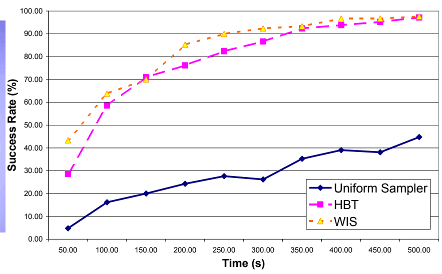


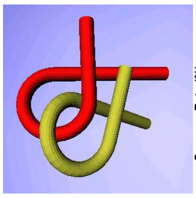
Fig. 3. Comparing the sampling distributions of three planners.

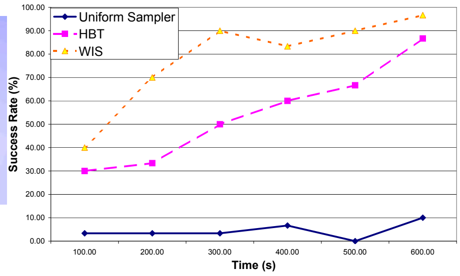




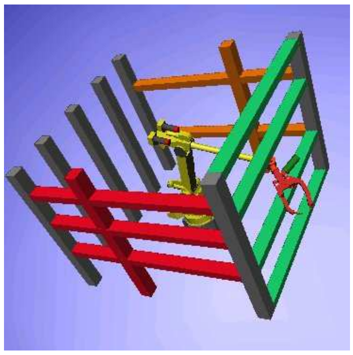
Test 4: This is a classic example for testing path planning algorithms. The 3-D workspace consists of eight chambers and seven narrow openings connecting adjacent chambers. The robot is a rigid body translating and rotating freely with six dofs in total. We specified seven queries, one for each opening.

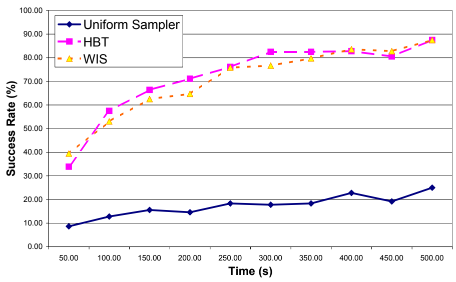
Although this test can be seen as a 3-D version of Test 3, the performance of WIS is comparable to HBT. The reason is that compared to Test 3, the narrow openings here are still relatively large with respect to the size of the robot. Thus, WIS can sample the orientations of the robot uniformly at random and still obtain enough collision-free configurations in the narrow openings.



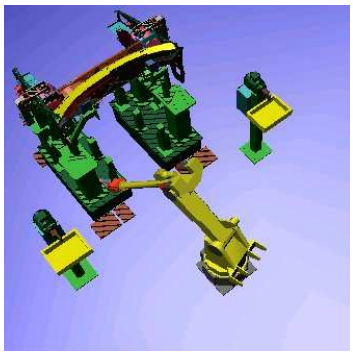


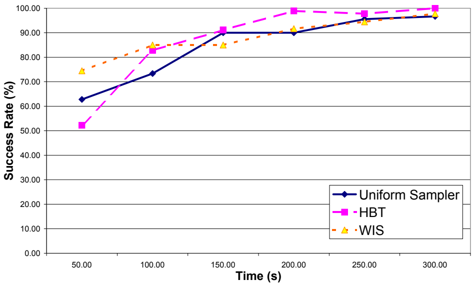
Test 5: We also considered the modified alpha puzzle environment used in [1]. As in [1], the puzzle was simplified by slightly increasing the size of the narrow opening on the alpha-shaped tubes. WIS performs very well, because the workspace information helps WIS to locate narrow passages efficiently.





Test 6: This 3-D workspace consists of horizontal and vertical bars to obstruct the motion of the robot. The robot is a six-dofs FANUC arm. We specified 12 queries. Each query requires the robot to move its end-effector from one opening between the bars to another. To answer a query, the robot must pull its end-effector out of a narrow opening, move in relatively open free space, and reinsert the end-effector into another narrow opening.





Test 7: This workspace has complex geometry, but does not contain any narrow passages. One concern about WIS is the preprocessing time for computing the tetrahedralization of $\mathcal{W}'$. This test shows that even in an environment with complex geometry (224,403 triangles), the preprocessing time needed for tetrahedralization does not have a significant impact on the total running time.

Overall WIS performs well when the environment contains very narrow passages, but does not require difficult reorientation of the robot to pass through the narrow passages. Since we set n_t to be dependent on the importance value $h(t)$ (see IV-B), the probability of sampling a free configuration corresponding to any tetrahedron $t \in \mathcal{T}'$ is roughly the same, regardless of t's size. Hence, the sampling distribution constructed by WIS is roughly proportional to the number of tetrahedra per unit volume, rather than the volume itself. This effectively increases the sampling density inside narrow passages, because narrow passages in the workspace usually contain many tetrahedra with small volumes.

VI. Conclusion and Future Work

We have presented workspace importance sampling, a new sampling strategy for PRM planning. WIS tetrahedralizes the workspace and uses the workspace information to locate narrow passages and guide sampling in the configuration space. By doing so, WIS increases the sampling density in narrow passages and decreases that in wide-open regions. We conducted experiments with WIS on rigid-body and articulated robots in 2-D and 3-D workspaces. Our results show that WIS improves the performance of PRM planning when narrow passages are present.

There are several aspects of WIS that need improvement, both in algorithm development and in implementation. Shape analysis of the robot may help to improve the sampling of orientations and to identify workspace narrow passages too small for the robot to pass through, as we have discussed in Section IV-E. It may also be beneficial to combine WIS with other sampling strategies to construct a hybrid sampling strategy.

Acknowledgment

We thank Alan Cheng for many helpful discussions, Mitul Saha for the data set used in Test 7, and Saurabh Garg for reading the first draft of this paper. We also thank the anonymous reviewers for their detailed comments, which greatly helped to improve the paper.

References

[1] Amato, N.M., O.B. Bayazit, L.K. Dale, C. Jones, 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*, pp.155-168, A. K. Peters, Wellesley, MA, 1998.

[2] Amato, N.M., K.A. Dill, G.Song. Using Motion Planning to Map Protein Folding Landscapes and Analyze Folding Kinetics of Known Native Structures. In *Proc. ACM Int. Conf. Research in Computational Biology (RECOMB)*, pp. 2-11, 2002.

[3] Amenta, N., S. Choi, R.K. Kolluri. The Power Crust. In *Proc. ACM Symp. on Solid Modeling*, 2001.

[4] Apaydin, M.S., D.L. Bruglag, C. Guestrin, D. Hsu, J.C. Latombe. Stochastic Roadmap Simulation: An Efficient Representation and Algorithm for Analyzing Molecular Motions. In *Proc. ACM Int. Conf. Research in Computational Biology (RECOMB)*, pp.12-21, 2002.

[5] Barber, C.B., D.P. Dobkin, H.T. Huhdanpaa. The Quickhull Algorithm for Convex Hull. In *ACM Trans. on Mathematical Software*, no.22, pp.469-483, 1996.

[6] Bohlin, R., L.E. Kavraki. Path Planning Using Lazy PRM. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pp.521-528, 2000.

[7] Boor, V., M.H. Overmars, A.F. Stappen. The Gaussian Sampling Strategy for Probabilistic Roadmap Planners. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pp.1018-1023, 1999.

[8] Chang, H., T.-Y. Li. Assembly Maintainability Study with Motion Planning. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pp.1012-1019, 1995.

[9] Foley, J.D., A. van Dam, S.K. Feiner, J.F. Hughes. *Computer Graphics: Principles and Practice in C.* Addison-Wesley, 1995.

[10] Foskey, M., M. Garber, M.C. Lin, D. Manocha. A Voronoi-Based Hybrid Motion Planner. In *Proc IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2001.

[11] Holleman, C., L.E.Kavraki. A Framework for Using the Workspace Medial Axis in PRM Planners. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pp.1408-1413, 2000.

[12] Hsu, D., J.C. Latombe, R. Motwani. Path Planning in Expansive Configuration Spaces. In *Int. J. of Computational Geometry & Applications*, v.9 n.4&5, pp.495-512, 1999.

[13] Hsu, D., L.E. Kavraki, J.C. Latombe, R. Motwani, 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*, pp.141-154, A. K. Peters, Wellesley, MA, 1998.

[14] Hsu, D., T. Jiang, J. Reif, Z. Sun. The Bridge Test for Sampling Narrow Passages with Probabilistic Roadmap Planners. In *IEEE Int. Conf. on Robotics & Automation*, 2003.

[15] Kalos, M.H., P.A. Whitlock. *Monte Carlo Methods.* John Wiley & Sons, New York, 1986.

[16] Kavraki, L.E., P. Švetska, J.C. Latombe, M.H. Overmars. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. In *IEEE Trans. on Robotics & Automation*, v.12 n.4, pp.566-580, 1996.

[17] Koga, Y., K. Kondo, J. Kuffner, J.C. Latombe. Planning Motions with Intentions. In *SIGGRAPH'94 Conference Proceedings*, pp.395-408, 1994.

[18] LaValle, S.M., J.J. Kuffner. Randomized Kinodynamic Planning. In *Int. J. Robotics Research*, v.20 n.5, pp.378-400, 2001.

[19] Nissoux, C., T. Siméon, J.-P. Laumond. Visibility Based Probabilistic Roadmaps. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pp.1316-1321, 1999.

[20] Quinlan, S. Efficient Distance Computation between Non-Convex Objects. In *Proc. Int. Conf. on Robotics & Automation*, pp.3324-3329, 1994.

[21] Shoemake, K. Animating Rotation with Quaternion Curves. In *ACM SIGGRAPH*, v.19, no.3, pp.245-254, 1985.

[22] Toussaint, G.T., C. Verbrugge, C. Wang, B. Zhu. Tetrahedralization of Simple and Non-Simple Polyhedra. In *Proc. 5th Canadian Conf. Computational Geometry*.

[23] van den Berg, J.P., M. Overmars. Using Workspace Information as a Guide to Non-Uniform Sampling in Probabilistic Roadmap Planners. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pp.453-460, 2004.

[24] Wilmarth, S.A., N. Amato, 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*, pp. 1024-1031, 1999.