

# UMAPRM: Uniformly Sampling the Medial Axis

Hsin-Yi (Cindy) Yeh, Jory Denny, Aaron Lindsey, Shawna Thomas, and Nancy M. Amato

**Abstract**—Maintaining clearance, or distance from obstacles, is a vital component of successful motion planning algorithms. Maintaining high clearance often creates safer paths for robots. Contemporary sampling-based planning algorithms that utilize the medial axis, or the set of all points equidistant to two or more obstacles, produce higher clearance paths. However, they are biased heavily toward certain portions of the medial axis, sometimes ignoring parts critical to planning, e.g., specific types of narrow passages. We introduce Uniform Medial Axis Probabilistic RoadMap (UMAPRM), a novel planning variant that generates samples uniformly on the medial axis of the free portion of  $C_{space}$ . We theoretically analyze the distribution generated by UMAPRM and show its uniformity. Our results show that UMAPRM’s distribution of samples along the medial axis is not only uniform but also preferable to other medial axis samplers in certain planning problems. We demonstrate that UMAPRM has negligible computational overhead over other sampling techniques and can solve problems the others could not, e.g., a bug trap. Finally, we demonstrate UMAPRM successfully generates higher clearance paths in the examples.

## I. INTRODUCTION

Planning valid (e.g., collision-free) motions for a movable object (robot) is challenging. Motion planning has been extensively studied and has applications not only in robotics but also in augmented reality [11], computer-aided design [2], and bioinformatics [15].

One method of computing motions is sampling-based planning [8]. A major breakthrough, sampling-based planners solved many previously intractable problems, especially for high-dimensional robotic systems. While many of these methods are *probabilistically complete*, i.e., they guarantee finding a solution, if one exists, as they sample ad infinitum. They may compute low clearance paths, making the extracted paths high-risk and undesirable for certain applications. Generally, most methods aim to simply generate feasible paths instead of paths with certain characteristics, e.g., high clearance.

One method of computing high clearance paths, Medial Axis Probabilistic RoadMap (MAPRM) [17], generates milestones along the medial axis of the planning space, or the set of points equidistant to two or more obstacles. Lien et. al. extended this to high dimensional spaces by allowing

for approximate clearance and penetration computations [12]. This approach computes higher clearance paths and biases samples to segments of the medial axis with large Voronoi regions. Moreover, in MAPRM the probability of sampling a configuration in a narrow passage depends on both its volume and its surrounding obstacle volume. Therefore, MAPRM can efficiently solve problems with narrow passages that have large surrounding obstacle volume. However, MAPRM does not provide any guarantee regarding the distribution of samples along the medial axis.

In this paper, we propose a new planning variant, Uniform Medial Axis PRM (UMAPRM), which uniformly samples the medial axis of the planning space. Briefly, UMAPRM samples fixed length line segments from the planning space and analyzes adjacent configurations along the segment with a simple test determining if the medial axis has been crossed. The test essentially analyzes changes in the nearest obstacles of successive configurations. It retains samples of maximal clearance when medial axis crossings are detected. Our specific contributions are as follows:

- We present a novel sampling technique, UMAPRM, that *uniformly* samples the medial axis of the planning space.
- We theoretically prove that UMAPRM samples are uniformly distributed along the medial axis.
- We experimentally show how UMAPRM uniformly samples the medial axis and how this uniformity can yield efficiency improvements in planning compared to MAPRM and uniform random sampling.

UMAPRM is inspired by both MAPRM and the recently proposed UOBPRM (Uniform Obstacle-Based PRM) [19]. UOBPRM guarantees uniformly distributed samples along the boundary of obstacle space; it works by analyzing random line segments for changes in validity and retains intersections between line segments and obstacles.

## II. PRELIMINARIES AND RELATED WORK

In this section, we discuss motion planning preliminaries and the basics of sampling-based planning. We discuss methods focused on planning in the presence of narrow passages and methods which generate solutions biased to the medial axis of the space.

### A. Motion Planning

A robot is a movable object whose position and orientation can be described by  $n$  parameters, or *degrees of freedom* (DOFs), each corresponding to an object component (e.g., object positions, object orientations, link angles, and/or link displacements). Hence, a robot’s placement, or configuration, can be uniquely described by a point  $\langle x_1, x_2, \dots, x_n \rangle$  in

This research supported in part by NSF awards CNS-0551685, CCF-0833199, CCF-0830753, IIS-0916053, IIS-0917266, EFRI-1240483, RI-1217991, by NIH NCI R25 CA090301-11, by Chevron, IBM, Intel, Oracle/Sun and by Award KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST). J. Denny supported in part by an NSF Graduate Research Fellowship.

Hsin-Yi (Cindy) Yeh, Jory Denny, Aaron Lindsey, Shawna Thomas, and Nancy M. Amato are with the Parasol Lab, Department of Computer Science and Engineering, Texas A&M University, College Station, TX, USA, {hyeh, jdenny, alindsey, sthomas, amato}@cse.tamu.edu.

an  $n$ -dimensional space (where  $x_i$  is the  $i$ th DOF), called the *configuration space* ( $\mathcal{C}_{space}$ ) [13]. The subset of all feasible configurations is the *free space* ( $\mathcal{C}_{free}$ ), while the union of all infeasible configurations is the *obstacle space* ( $\mathcal{C}_{obst}$ ). Thus, the motion planning problem becomes that of finding a continuous trajectory in  $\mathcal{C}_{free}$  from a given start configuration to a goal configuration. In general, it is infeasible to compute explicit  $\mathcal{C}_{obst}$  boundaries [14], but we can often determine the validity of a configuration quite efficiently, e.g., by performing a collision detection (CD) test in the *workspace*, the robot's natural space.

Sampling-based methods [8], [10] are quite successful at solving motion planning problems. Specifically, Probabilistic RoadMaps (PRMs) [8] construct a map of  $\mathcal{C}_{free}$  by first randomly sampling valid configurations. Then, nearby samples are connected by validating simple paths which form edges of the roadmap. Finally, start and goal configurations are connected to the roadmap and a graph search, e.g.,  $A^*$ , extracts a solution path. Despite their successes, PRMs are inefficient at mapping narrow passages [7].

To improve the mapping of narrow passages, some PRM variants attempt to map  $\mathcal{C}_{obst}$  surfaces. Obstacle-Based PRM (OBPRM) [1] and Uniform OBPRM [19] sample configurations near  $\mathcal{C}_{obst}$  surfaces either by pushing configurations to the  $\mathcal{C}_{obst}$  boundary or by finding surface intersections of randomly placed line segments, respectively. In particular, UOBPRM generates fixed length line segments and checks the validity of the intermediate configurations along the segments at a fixed resolution. If there is any validity change between neighboring configurations, the valid one is retained. UOBPRM has been theoretically and experimentally shown to generate configurations uniformly distributed on  $\mathcal{C}_{obst}$  surfaces. Gaussian PRM [3] and Bridge Test PRM [6] filter samples with inexpensive tests to find samples near  $\mathcal{C}_{obst}$  boundaries or directly in narrow passages, respectively. However, both Gaussian PRM and Bridge Test PRM suffer from parameter tuning of their tests which can greatly affect the performance and quality of the mappings produced.

### B. Medial Axis Planning

The *medial axis* is the set of all points equidistant to two or more obstacles. It has dimension  $n - 1$  where  $n$  is the dimension of the overall space. The medial axis is a *strong deformation retraction* meaning that every point in the space has a unique corresponding point on the medial axis. The medial axis has been used in planning in various ways.

Medial Axis PRM (MAPRM) [12], [17] pushes randomly sampled configurations to the medial axis. Depending on the initial validity of the sampled configuration, MAPRM pushes samples toward (if initially invalid) or away from (if initially valid) their closest point on the boundary of  $\mathcal{C}_{obst}$  until this witness point changes. In higher dimensions, approximate clearance and penetration computations are used to determine this point [12]. MAPRM provably improves the probability of sampling in narrow passages over uniform random sampling. In MAPRM, sampling a narrow passage is dependent not only on the volume of the narrow passage

but on the volume of the obstacles surrounding it. However, it is computationally intensive, even with approximations.

The workspace medial axis can also be used to bias planning [5], [18]. These methods use either an explicit or an approximated calculation of the workspace medial axis to bias sampling distributions to more effectively plan in the presence of narrow passages. This approach has been extended to plan for flexible objects [4]. However, these approaches do not maximize clearance in  $\mathcal{C}_{space}$ .

## III. UNIFORM MEDIAL-AXIS PRM

In UMAPRM, like in PRM, samples are generated and then connected to form a roadmap. Once this roadmap is constructed, a path can be extracted to solve queries, as in [8]. UMAPRM is unique in how it generates samples, specifically along the medial axis of  $\mathcal{C}_{free}$ . UMAPRM combines techniques from MAPRM [17] and UOBPRM [19] in a novel sampling approach discussed in Section III-A. Essentially, UMAPRM randomly samples line segments in  $\mathcal{C}_{space}$  and analyzes these segments to find where they cross the medial axis of  $\mathcal{C}_{free}$ . We show in Section III-B that UMAPRM guarantees a uniform distribution of samples along the medial axis of  $\mathcal{C}_{free}$ .

### A. Algorithm

Algorithm 1 describes how UMAPRM generates samples. First, the algorithm generates a random line segment of length  $l$  that at least partially lies in  $\mathcal{C}_{space}$ . To do this, a configuration  $c$  and a random directional unit vector  $\vec{d}$  are sampled uniformly from  $\mathcal{C}_{space}$ . The endpoints of the segment are  $c$  and  $c + l\vec{d}$ . Without loss of generality, we assume we generate line segments of length  $l$  whose endpoints are not bound to the  $\mathcal{C}_{space}$ , but at least some portion of the line is within the  $\mathcal{C}_{space}$ . Next, we check every intermediate configuration on the line segment at some fixed resolution  $t$  for medial axis crossings. The medial axis is found when the closest obstacle changes between two consecutive configurations. When a crossing is detected, a binary search finds and retains the configuration on the medial axis.

1) *Refine the bounding box if needed:* Often, the boundary of the planning space is defined as a bounding box. If there is not enough space for UMAPRM to generate line segments because the medial axis is within distance  $l$  of the boundary, it will invalidate the uniformity guarantee along the medial axis since fewer segments would be generated near the boundary. Hence, as in UOBPRM [19], we temporarily adjust the bounding box during line segment generation to ensure a uniform distribution of random line segments of length  $l$  within  $\mathcal{C}_{space}$ . However, the planning problem is unchanged because we analyze segments with respect to the original bounding box.

### B. Uniformity Analysis

Here, we prove that the configurations generated by UMAPRM are uniformly distributed on the medial axis. This proof follows similar reasoning as in UOBPRM [19].

---

**Algorithm 1** UMAPRM Configuration Generation( $l, t$ )

---

**Input:** A line segment length  $l$  and a step size  $t$

**Output:** A set of uniformly distributed configurations  $V$  that lie on the medial axis of the free space

```
1:  $V = \emptyset$ 
2: Expand the bounding box by  $l$ 
3: Uniformly sample a configuration  $c$ 
4: Generate a random direction  $\vec{d}$ 
5: Extend a segment  $s$  from  $c$  to  $c + l\vec{d}$ 
6: for  $i = 1 \rightarrow (l/t)$  do
7:   Let  $c_i$  and  $c_{i+1}$  be the  $i$ -th and  $(i+1)$ -th configurations along  $s$ .
8:   if The medial axis is between  $c_i$  and  $c_{i+1}$  then
9:      $c_{ma} = \text{BinarySearch}(c_i, c_{i+1})$ 
10:     $V = V \cup \{c_{ma}\}$ 
11: return  $V$ 
```

---

**Proposition 1:** Given a  $\mathcal{C}_{space}$ , the probability of finding an intersection point  $p$  on the medial axis of  $\mathcal{C}_{free}$  from a line segment of length  $l$  chosen uniformly at random is constant throughout  $\mathcal{C}_{space}$ .

*Proof:* Let  $\mathcal{C}'_{space}$  be the enlarged space where line segments are sampled,  $p$  be a point on the medial axis of  $\mathcal{C}_{free}$ ,  $S$  be the sphere centered at  $p$  with radius  $l$ , and  $(c, l\vec{d})$  be a line segment where  $c$  is a random configuration and  $\vec{d}$  is a random direction.  $MA_{p,\epsilon}$  is the portion of the medial axis that is contained in a ball of radius  $\epsilon$  centered at  $p$ , i.e.,  $MA_{p,\epsilon} = \text{medial axis} \cap B_{p,\epsilon}$ , where  $\epsilon > 0$ .  $p$  is on  $(c, l\vec{d})$  if and only if  $S$  contains  $c$  and  $\vec{d}$  intersects  $MA_{p,\epsilon}$ , as shown in Figure 1. Therefore, the probability  $P_{MA}$  that the line segment intersects  $MA_{p,\epsilon}$  is equivalent to the probability that  $c$  resides in  $S$  and  $\vec{d}$  intersects  $MA_{p,\epsilon}$ , i.e.,  $P_{MA} = P((c \in S) \wedge (\vec{d} \cap MA_{p,\epsilon}))$ . Given the conditions that  $c$  and  $\vec{d}$  are both selected uniformly at random,  $P_{MA}$  is uniform along the medial axis of  $\mathcal{C}_{free}$ . ■

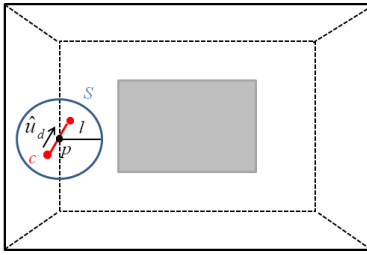


Fig. 1. The probability to find a line segment  $(c, l\vec{d})$  which crosses the medial axis (dashed line) at point  $p$  is uniform throughout the environment.

**Corollary 1:** For  $n$  randomly generated line segments of fixed length  $l$ , the probability of finding intersection points with the medial axis of  $\mathcal{C}_{free}$  is constant throughout  $\mathcal{C}_{space}$ . Since the probability of occurrence is the same, the distribution of the intersection points found by UMAPRM is uniform along the medial axis of  $\mathcal{C}_{free}$ .

UMAPRM has dual benefits for planning. First, samples lie on the medial axis, thus node clearance is increased.

Paths extracted from such roadmaps will have overall higher clearance, which leads to safer paths for robotic systems. Second, UMAPRM generates samples uniformly on the medial axis. Uniformly sampling the medial axis allows UMAPRM to map narrow passages with small surrounding obstacle volume efficiently. The probability instead depends on the surface area of the medial axis itself in the narrow passage. In other words, UMAPRM can map certain types of passages more efficiently than MAPRM [12].

#### IV. EXPERIMENTS

In this section, we analyze both the uniform distribution of samples on the medial axis with UMAPRM and the benefit of this uniformity for motion planning. We show that UMAPRM is able to generate more uniformly distributed configurations compared to MAPRM. Also, unlike MAPRM, we see the performance of UMAPRM does not decrease when the surrounding obstacle width decreases. For the problems studied here, the cost for UMAPRM to solve the problem is comparable to or even less than MAPRM and it generates paths of comparable quality to MAPRM.

##### A. Experimental Setup

UMAPRM, MAPRM, and basic PRM (uniform sampling) were implemented in a C++ motion planning library developed in the Parasol Lab at Texas A&M University. This library uses a distributed graph data structure from the Standard Template Adaptive Parallel Library (STAPL) [16], a C++ library designed for parallel computing. All methods use PQP [9] for collision detection and the Euclidean distance metric for distance calculation. All robots are considered point robots. We ran the experiments on a Rocks Cluster running CentOS 5.1 with Intel XEON CPU 2.4 GHz processors with the GNU gcc compiler version 4.1.

##### B. Finding the medial axis for point robots

The robots in our experiments are all point robots, so the medial axis of  $\mathcal{C}_{space}$  is the same as of workspace. First, we compute the closest witness point on the boundary of free space and identify which workspace obstacle component it belongs to. Obstacles are modeled as polyhedra composed of triangles, so the witness obstacle component is a vertex or a triangle. Then, we consider  $c_i$  and  $c_{i+1}$  on the opposite sides of the medial axis if one of the following occurs:

- Witnesses belong to different obstacles (Figure 3(a)).
- Witnesses belong to the same obstacle and are not adjacent model components (Figure 3(b)).
- Witnesses belong to the same obstacle, are adjacent, and are on opposite sides of a concavity in the model, e.g., two neighboring concave triangles (Figure 3(c)).

##### C. Uniformity

In this section, we provide two experiments showing comparative distributions of medial axis samples between UMAPRM and MAPRM. In the first experiment, we show how certain environments cause MAPRM samples to be non-uniformly distributed, while UMAPRM samples are

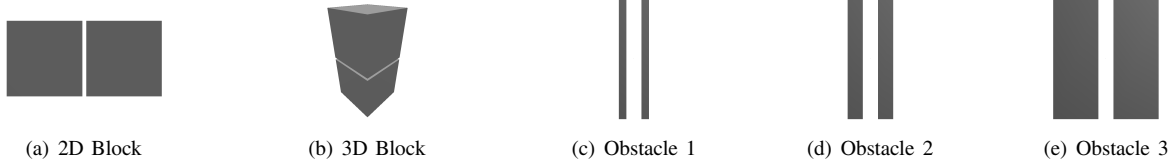


Fig. 2. (a, b) Two environments used to compare the distributions of UMAPRM and MAPRM. (c, d, e) Narrow passages of varying surrounding obstacle volume to compare sampling densities of UMAPRM and MAPRM.

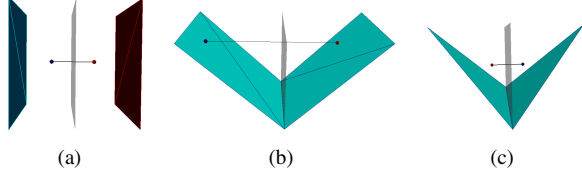


Fig. 3. Three examples showing how UMAPRM finds configurations on the medial axis by checking changes in closest triangles on obstacles. The grey face is the medial axis. The medial axis is crossed when (a) closest triangles are on different obstacles, (b) closest triangles are on the same obstacle but not adjacent to each other, or (c) neighboring concave triangles are on the same obstacle.

uniformly distributed along the medial axis. In the second experiment, we compare the methods' abilities to sample in various narrow passages. While MAPRM's probability of sampling in a narrow passage is tied to the surrounding obstacle volume, UMAPRM's probability is related to the surface area of the medial axis through the narrow passage.

1) *Distributions*: We compare UMAPRM's and MAPRM's distributions and levels of uniformity in simple 2D and 3D environments with a narrow passage created by two unit blocks (shown in Figure 2 (a, b)). In each trial, we generated 1000 samples with each method along the segment of the medial axis between the blocks (we ignore the portions of the medial axis related to the boundary). Because the medial axis is simple, i.e., a line or plane, we compare with uniformly distributed points along this structure. As a measure of uniformity, we computed the standard deviation of the distances between each node and its closest neighbor. We averaged this over 40 trials.

Figure 4 shows that UMAPRM has a lower average standard deviation in both environments, implying a more uniform distribution. Additionally, UMAPRM has roughly the same average as uniformly random distributed points along the medial axis plane.

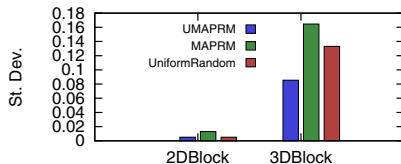


Fig. 4. The average of standard deviations of distances between each node and its closest neighbor for roadmaps of 1000 samples between UMAPRM (green), MAPRM (blue), and uniform random sampling on the medial axis (red).

Figures 5 and 6 show the sample distributions from representative maps generated from 1,000 samples for UMAPRM,

MAPRM, and uniform random sampling on the medial axis. MAPRM is highly biased towards the area between the blocks in the experiment, while UMAPRM is uniformly distributed across the entire medial axis plane for both 2D and 3D. MAPRM is biased because it pushes samples away from its closest point on the boundary of  $C_{obst}$ . From this, we can clearly see that only samples whose the closest point is one of the corners of the block will be pushed towards the portion of the medial axis not covered by the block. The probability of this happening is much lower than for other portions of the medial axis. UMAPRM on the other hand, uniformly samples and analyzes line segments which do not have biases based on  $C_{obst}$  boundaries.

2) *Varying surrounding obstacle width*: MAPRM has a known bias towards narrow passages because its probability of sampling in a narrow passage is proportional to the volume of the narrow passage and its surrounding obstacles. UMAPRM however, is unaffected by changes in surrounding obstacle volume. Here, we generate 1000 samples along the medial axis of the whole space (considering the boundary) and determine how many lie inside the narrow passage. We compare this for Obstacle 1, 2, and 3 shown in Figure 2(c-e). Obstacle 1 has the smallest obstacle volume, while Obstacle 3 has the largest. We averaged results over 40 trials and report the ratio of nodes inside the narrow passage and the time it takes to generate the samples in Figure 7 and Figure 8, respectively.

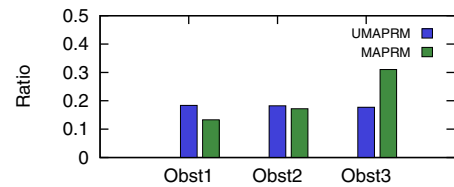


Fig. 7. The number of nodes in the narrow passage out of 1000 samples for UMAPRM and MAPRM in Obstacle 1, 2, and 3.

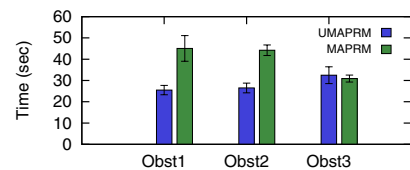


Fig. 8. The time to generate 1000 samples for UMAPRM and MAPRM in Obstacle 1, 2, and 3, error bars show standard deviation.

As the plots show, UMAPRM consistently generates around 18% of nodes in each narrow passage, as the surface

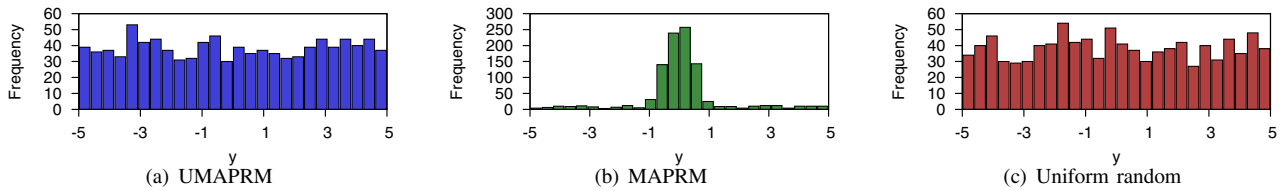


Fig. 5. Distributions of 1,000 samples generated by UMAPRM, MAPRM, and uniform random sampling in the 2D Block environment.

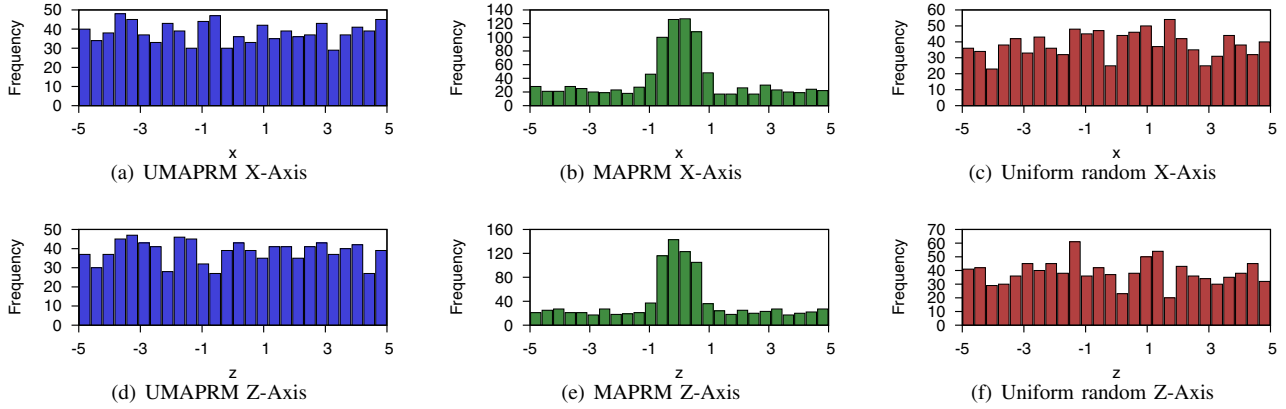


Fig. 6. Distributions of 1,000 samples generated by MAPRM, UMAPRM, and uniform random sampling in the 3D Block environment.

area of the medial axis in the narrow passage is constant. MAPRM performance is not consistent. It has difficulty generating samples in the narrow passage in the smallest case, generating only about 13% in Obstacle 1. These consistency trends are evidence that UMAPRM generates a more uniform distribution of samples compared to MAPRM.

UMAPRM is also more consistent in the time it takes to generate successful samples across the three narrow passage examples, whereas MAPRM's efficiency is related to the distance each node must be pushed to reach the medial axis. As the obstacle width increases, each node must traverse a smaller distance to the medial axis on average. UMAPRM's efficiency is only slightly affected in the third case. Although the change in obstacle volume does not affect the probability of sampling the narrow passage for UMAPRM, the total surface area of the medial axis has changed, thus causing the average time to generate a sample to increase.

As we have shown in these experiments, UMAPRM has a uniformly random distribution of samples along the medial axis of the space. Conversely, the distribution of samples in MAPRM can be highly non-uniform. In the following section, we will see how this affects planning in general.

#### D. Motion Planning

In this section, we compare PRM, MAPRM, and UMAPRM for planning between start and goal configurations in the environments shown in Figure 9. In 2DMaze (Figure 9(a)), the path traverses a long and windy narrow passage. In STunnel (Figure 9(b)), thin obstacles surround a narrow passage. In 2DHeterogeneous (Figure 9(c)), a path winds through varying narrow passages of differing types. In Bug Trap (Figure 9(d)), the robot must escape the trap

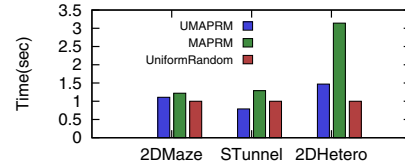


Fig. 10. The time to solve the problem for UMAPRM, MAPRM, and PRM in different environments.

by traversing a small opening. All results are averaged over 40 trials. Recall that while both MAPRM and UMAPRM sample configurations along the medial axis, MAPRM's distribution is not guaranteed to be uniform and performance is dependent on the surrounding obstacle volume while UMAPRM guarantees a uniform distribution and its performance is unaffected by obstacle volume.

1) *Time*: We first study the efficiency for each sampler to solve the query. The time is normalized to PRM. The results in Figure 10 show that UMAPRM takes less time than MAPRM, but is slower than PRM since PRM performs fewer collision detection calls in 2DMaze and 2DHeterogeneous environment. In the STunnel, UMAPRM outperforms both methods. PRM struggles because the volume of the narrow passage is small compared with the rest of the planning space. MAPRM is hampered because the volume of the surrounding obstacles is also small compared with the rest of the planning space. In Bug Trap, UMAPRM finds the solution path in 2 hours, while neither PRM nor MAPRM is able to solve the problem after running for 10 hours.

2) *Clearance*: In addition to the time, we are also interested in the path quality by calculating the path clearance for each sampler. The clearance is again normalized to





Fig. 9. Motion planning environments studied. (a) 2DMaze. The start and the goal reside at the two ends in the free space. (b) STunnel. The start and the goal are in the top left and the bottom right corners. (c) 2DHeterogeneous. The start is in the top free space and the goal is placed in the bottom cluttered region. (d) Bug Trap. The objective is to get out of the trap through the narrow passage.

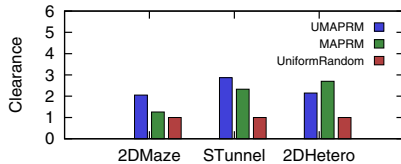


Fig. 11. The average clearance of the path for UMAPRM, MAPRM, and PRM in different environments.

PRM. Figure 11 compares the average path clearance for different samplers in various environments. The average path clearance is the average of the edge clearances for the solution path. UMAPRM generates higher quality paths than MAPRM in both 2DMaze and STunnel. In 2DHeterogeneous, the average path clearance between UMAPRM and MAPRM is comparable. Only UMAPRM solves the Bug Trap, so there is no normalized average path clearance in this environment.

## V. CONCLUSION

We introduce UMAPRM as a novel planning variant that uniformly samples the medial axis of the planning space. We prove that the distribution of samples is indeed uniform, and demonstrate how this uniformity can benefit planning in certain scenarios, e.g., solving hard problems such as the Bug Trap efficiently. In the future, we would like to explore this method considering rigid body robots and higher dimensional robots instead of merely point robots, possibly with approximation techniques. Also, we believe that when used in combination with other planners, UMAPRM can yield high clearance paths efficiently for many problems.

## REFERENCES

- [1] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo. OBPRM: an obstacle-based PRM for 3d workspaces. In *Proceedings of the third workshop on the algorithmic foundations of robotics on Robotics : the algorithmic perspective: the algorithmic perspective*, WAFR '98, pages 155–168, Natick, MA, USA, 1998. A. K. Peters, Ltd.
- [2] O. B. Bayazit, G. Song, and N. M. Amato. Enhancing randomized motion planners: Exploring with haptic hints. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 529–536, 2000.
- [3] 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, May 1999.
- [4] L. Guibas, C. Holleman, and L. Kavraki. A probabilistic roadmap planner for flexible objects with a workspace medial-axis-based sampling approach. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, volume 1, pages 254–259, 1999.
- [5] C. Holleman and L. E. Kavraki. A framework for using the workspace medial axis in prm planners. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, volume 2, pages 1408–1413, San Francisco, CA, 2000.
- [6] D. Hsu, T. Jiang, J. Reif, and Z. Sun. Bridge test for sampling narrow passages with probabilistic roadmap planners. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 4420–4426, 2003.
- [7] D. Hsu, J.-C. Latombe, and H. Kurniawati. On the probabilistic foundations of probabilistic roadmap planning. *Int. J. Robot. Res.*, 25:627–643, July 2006.
- [8] L. E. Kavraki, P. Švestka, 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.
- [9] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha. Distance queries with rectangular swept sphere volumes. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, volume 4, pages 3719–3726 vol.4, 2000.
- [10] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 473–479, 1999.
- [11] J.-M. Lien, O. B. Bayazit, R.-T. Sowell, S. Rodriguez, and N. M. Amato. Shepherding behaviors. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 4159–4164, April 2004.
- [12] J.-M. Lien, S. Thomas, and N. Amato. A general framework for sampling on the medial axis of the free space. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 3, pages 4439 – 4444, sept. 2003.
- [13] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, October 1979.
- [14] 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.
- [15] A. P. Singh, J.-C. Latombe, and D. L. Brutlag. A motion planning approach to flexible ligand binding. In *Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, pages 252–261, 1999.
- [16] G. Tanase, A. Buss, A. Fidel, Harshvardhan, I. Papadopoulos, O. Pearce, T. Smith, N. Thomas, X. Xu, N. Mourad, J. Vu, M. Bianco, N. M. Amato, and L. Rauchwerger. The STAPL Parallel Container Framework. In *Proc. ACM SIGPLAN Symp. Prin. Prac. Par. Prog. (PPoPP)*, pages 235–246, San Antonio, Texas, USA, 2011.
- [17] 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.
- [18] Y. Yang and O. Brock. Adapting the sampling distribution in prm planners based on an approximated medial axis. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, volume 5, pages 4405–4410, 2004.
- [19] H.-Y. C. Yeh, S. Thomas, D. Eppstein, and N. M. Amato. UOBPRM: A uniformly distributed obstacle-based PRM. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, pages 2655–2662, Vilamoura, Algarve, Portugal, 2012.