2020

Fast Medial Axis Sampling for Use in Motion Planning

Hanglin Zhou
University of Richmond

Fast Medial Axis Sampling for Use in Motion Planning

Hanglin Zhou
Honor Thesis[1]
Department of Mathematics and Computer Science
University of Richmond
May 2, 2019

[1]Under the direction of Jory Denny

This paper is part of the requirements for the honors program in computer science. The signatures below, by the advisor, a departmental reader, and a representative of the depart-mental honors committee, demonstrate that Hanglin Zhou has met all the requirements needed to receive honors in computer science.

Advisor – Jory Denny

Reader – Lewis Barnett

Honors Committee Representative – Lewis Barnett

ABSTRACT

Motion planning is a difficult but important problem in robotics. Research has tended toward approximations and randomized algorithms, like sampling-based planning. Probabilistic RoadMaps (PRMs) are one common sampling-based planning approach, but they lack safety guarantees. One main approach, Medial Axis PRM (MAPRM) addressed this deficiency by generating robot configurations as far away from the obstacles as possible, but it introduced an extensive computational burden.

We present two techniques, Medial Axis Bridge and Medial Axis Spherical Step, to reduce the computational cost of sampling in MAPRM and additionally propose recycling previously computed clearance information to reduce the cost of connection in MAPRM. We provide experimental results that demonstrate the effectiveness of our proposed methods by: (1) showing that Medial Axis Bridge and Medial Axis Spherical Step both reduce the sampling time of MAPRM by nearly 50% while guaranteeing the same degree of safety, and (2) showing a nearly 50% decrease in connection time in MAPRM.

DEDICATION

To my mentors and friends.

ACKNOWLEDGEMENTS

First, I am extremely grateful to my research advisor, Dr. Jory Denny. Without your help, feedback and guidance, this work would not have been accomplished. Thank you for your support and patience since I took Introduction to Computer Science in Spring 2017. Thanks to your constant encouragement and constructive advice, I will always choose to do the things that I enjoy the most even if I am afraid to do so. Last, thanks to the years working with you, I have almost overcome procrastination.

Special thanks to my family and Kaiwen Chen, for all the love and support.

I must thank Dr. Kelly Shaw for being a great mentor to me. Your passion towards what you do and your perseverance empower me to focus on what I want to achieve the most. Thank you for your belief in my work and in me. I am really appreciative of your insightful suggestions.

Thank you to my dear friends, Maxine Xin, Angela Wang, and Bonnie Qiu. You have always supported me and encouraged me to pursue what I am passionate about. You have always believed in me.

Many thanks to David Qin, my research partner. We had great discussions. You helped me with this work and never complained of me occupying the lab computer.

Lastly, thank all the great professors in University of Richmond, especially the professors in the Department of Math and Computer Science, for your passion, wisdom, support, and patience.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

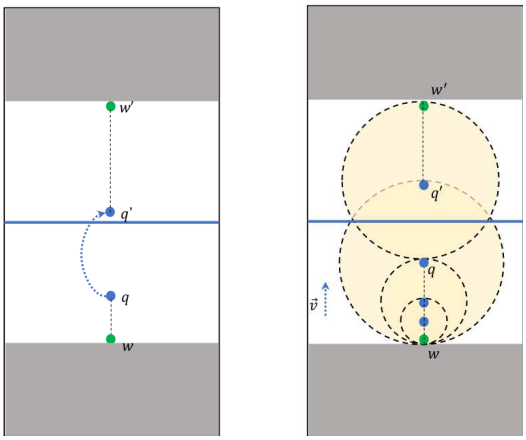
1. INTRODUCTION

Motion planning is an important problem in many domains, such as robotics, bioinformatics, virtual prototyping, and graphic animation [17]. The motion planning problem is the search for a contiguous sequence of valid (e.g., collision-free) states that begins at an initial position and ends at a desired goal region. In most cases, it is computationally infeasible to find a path deterministically. As such, research has focused on probabilistic methods that achieve efficiency and applicability at the cost of completeness.

Sampling-based approaches [13][19] perform well in solving a number of difficult motion planning problems. They generally compute solutions to a motion planning problem by constructing a random graph that represents the planning space and finding solution paths within that graph. Probabilistic RoadMap (PRM) is one such approach. However, PRM performs poorly in situations that require paths to pass through narrow passages. Additionally, PRM generates very jagged paths and/or paths that are close to obstacles, which can be dangerous for a robot.

PRM variants have been developed to overcome the above issues. Some methods [1][5][11][27] find samples near the boundaries of obstacles and in narrow passages to improve sampling. Other methods [20][26][28] sample away from obstacles and thus provide safety.

In all of these variants, collision checking is considered to be the computational bottleneck early on in their executions [18]. Various techniques have been proposed to make the planning process more efficient by decreasing the time spent on collision checking. Some methods [4][23] use laziness to avoid unnecessary collision checking. Other research [3][25] uses the idea of utilizing clearance information to define a



(a) Medial Axis Bridge

(b) Medial Axis Spherical Step

Figure 1.1: Proposed sampling methods: (a) Medial Axis Bridge and (b) Medial Axis Spherical Step.

safety certificate in the form of a valid hypersphere to eliminate unnecessary collision checks.

This work applies biased sampling with exploitation of valid hyperspheres to improve the efficiency of planning on the medial axis of the space, i.e., set of all points equidistant to two or more obstacles. Specifically, we improve Medial Axis PRM(MAPRM) [26]. As such, our methods accelerate the sampling and total planning processes used within MAPRM.

1.1 Research Contribution

This work proposes two approaches to make MAPRM sampling more efficient. The first approach, Medial Axis Bridge (Figure 1.1(a)), uses filtering to perform inexpensive checks to quickly rule out initial samples that typically cause MAPRM to perform poorly. The second approach, Medial Axis Spherical Step (Figure 1.1(b)), employs clearance from past collision detection to exploit valid hyperspheres in generating samples to prevent unnecessary collision checks. We provide additional en-

hancement to MAPRM by improving edge generation using valid hyperspheres. This work provides empirical evidence that shows the effectiveness of our techniques in reducing computation cost in a variety of environments.

1.2 Outline

Chapter 2 provides an overview of the motion planning problem and reviews related approaches. Chapter 3 describes our Medial Axis Bridge and Medial Axis Spherical Step approach. Chapter 4 provides empirical results of Medial Axis Bridge, Medial Axis Spherical Step approach, and medial axis sampling with valid hyperspheres and compares them to related approaches. Chapter 5 summarizes the work and discusses future research directions.

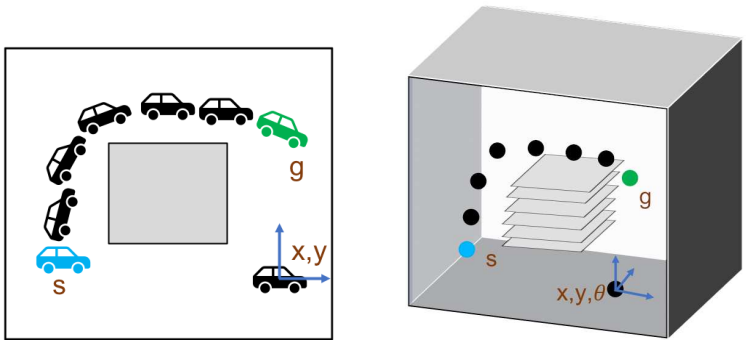
2. PRELIMINARIES AND RELATED WORK

First, we review the basics of motion planning, and then we describe related approaches to our proposed methods. Specifically, we focus our discussion on planning variants that provide safety guarantees or improve the efficiency of planning.

2.1 Motion Planning Problem

A robot is a movable object with *d Degrees of Freedom* (DOFs). DOFs parameterize a unique placement of the robot (e.g., joint angles or center of mass position). A *configuration* $q = \langle x_1, x_2, ..., x_d \rangle$ is a specification of the values for the DOFs, where x_i is the ith DOF. The set of all possible configurations is the *configuration space*, denoted as $\mathcal{C}_{space}$ [21]. The subset of all feasible configurations is the *free space*, $\mathcal{C}_{free}$, and the set of all infeasible configurations is the *obstacle space*, $\mathcal{C}_{obst}$. The comparison between *workspace*, i.e., the robot's natural two- or three-dimensional world, and $\mathcal{C}_{space}$ is shown in Figure 2.1. A car-like robot in a two-dimensional *workspace* is a point in a three-dimensional $\mathcal{C}_{space}$. A path in the *workspace* is a swept volume, whereas in $\mathcal{C}_{space}$ it is a one-dimensional trajectory.

With this notion, the motion planning problem becomes that of finding a continuous trajectory in $\mathcal{C}_{free}$ between a start and goal configuration q_s, $q_g \in \mathcal{C}_{free}$. It is intractable to solve the motion planning problem in general [24]. However, we can quickly perform a *collision detection* test (e.g., [6][22]) in the *workspace* to determine if a *configuration* is feasible or not — the basic operation of efficient randomized planning algorithms. Randomness helps overcome the intractability of motion planning by sacrificing a complete solution for a probabilistically complete one — if a solution path exists, the probability of finding a path approaches one as the algorithm continues to run.



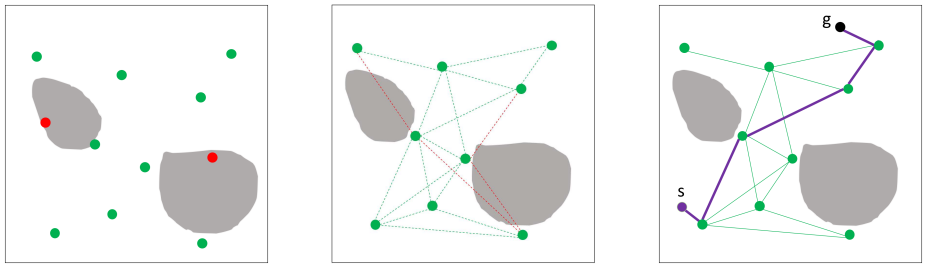
(a) Workspace (b) $\mathcal{C}_{space}$

Figure 2.1: Example of (a) workspace and (b) $\mathcal{C}_{space}$ for a car like robot with 3 DOFs.

2.2 Probabilistic RoadMaps

Sampling-based planners, like Probabilistic RoadMaps (PRMs) [13] and Rapidly-exploring Random Trees (RRTs) [19], solve the motion planning problem by generating random roadmaps, i.e., undirected graphs, that represent $\mathcal{C}_{free}$. One such approach PRM (Algorithm 1 and Figure 2.2) divides planning into a learning phase and a query phase. In the learning phase, PRM constructs a roadmap in $\mathcal{C}_{free}$ by generating random valid (i.e. collision-free) configurations and connecting neighboring samples that have collision-free transitions (e.g. straight-lines) between them. In the query phase, user defined start and goal configurations are connected to the roadmap and a path from the start to the goal configuration will be extracted using a graph search algorithm (Figure 2.2(c)). The learning phase and the query phase will be repeated until a path is found.

In practice, PRM can solve high-dimensional motion planning problems quickly. However, in many cases, as PRM uniformly generates random samples, the less volume a corridor in $\mathcal{C}_{free}$ takes up, then the smaller probability it will have any samples in it [12]. With this drawback, PRM performs poorly in the scenarios that



(a) Sample (b) Connect (c) Query

Figure 2.2: Example execution of PRM construction: (a) sampling, (b) connecting, and (c) querying.

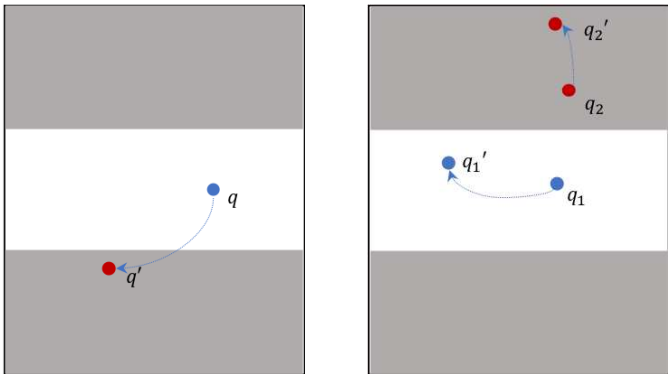
Algorithm 1 PRM [13]

Input: Start configuration q_s, goal configuration q_g

1: *Roadmap* $R = (V, E) \leftarrow (\emptyset, \emptyset)$
2: **while** $\neg$*done* **do**
3: $V \leftarrow$SAMPLE()
4: $E \leftarrow$CONNECT(V)
5: R.FINDPATH(q_s, q_g)

require solution paths to pass through narrow passages. In order to have enough samples in narrow passages, PRM needs to sample and connect more configurations, which is inefficient. Meanwhile, PRM does not provide any safety guarantees — it can generate jagged paths and/or paths that are close to the obstacles, which are dangerous for a robot.

In order to improve planning in narrow passages, PRM variants have been developed with techniques that bias or filter sampling towards the boundaries of $\mathcal{C}_{obst}$. Obstacle-Based PRM (OBPRM) [1][27] pushes samples near the surface of the obstacles. Gaussian PRM [5] and Bridge Test PRM [11] use filtering technique that performs inexpensive tests to find samples near the boundaries of $\mathcal{C}_{obst}$ or in narrow passages respectively. As shown in Figure 2.3(a) and 2.3(b), Gaussian PRM uni-



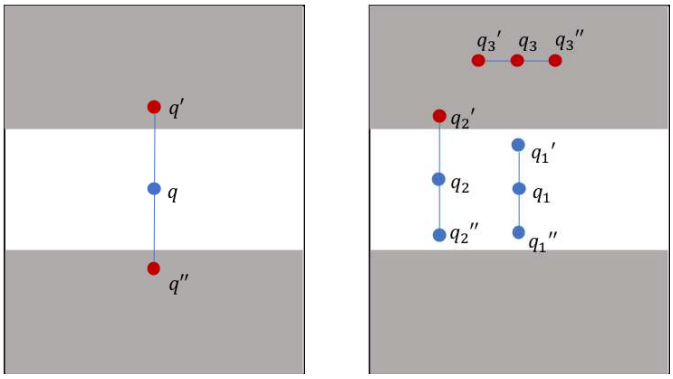
(a) Successful sample (b) Unsuccessful sample

Figure 2.3: (a) Successful and (b) unsuccessful Gaussian PRM sampling attempts.

formly generates a random sample and a second sample at a random distance away from the first sample based on a Gaussian distribution. A sample is added to the roadmap if and only if one is valid and the other is invalid. Bridge Test [11] uniformly samples a random configuration q' and finds q'' at a random distance away from q' based on a Gaussian distribution. If q' and q'' are both invalid, it finds the middle point q of these two samples. The middle point q is added to the roadmap if and only if it is valid, as shown in Figure 2.4(a), otherwise, samples q, q' and q'' are discarded (Figure 2.4(b)). The downside of these approaches is that it may fail many times before it successfully finds the sample that bridges the gap between the narrow passage. This class of PRM variants samples close to obstacles, thus, generating dangerous paths for a robot.

2.3 Efficient Collision Checking Variants

Collision detection is considered to be one of the main computational bottlenecks in sampling-based planning in practice [18]. Many variants that reduce the number of collision detection tests to accelerate the planning process exist. Lazy PRM [4] minimizes the number of collision detection calls during planning by delay-

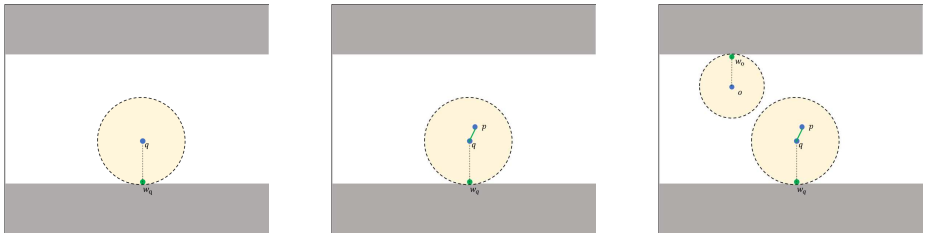


(a) Successful sample (b) Unsuccessful sample

Figure 2.4: (a) Successful and (b) unsuccessful Bridge Test PRM sampling examples.

ing them until needed. It initially assumes that all nodes and edges in the roadmap are collision-free and only perform collision checks if a solution path is found. Fuzzy PRM [23] uses the idea of laziness to avoid unnecessary collision detection calls to solve manipulation planning problems. It uses a fuzzy roadmap, which is an edge probability annotated roadmap. In the fuzzy roadmap, edges are not verified by local planners but are assigned a number which represents the probability of its feasibility and later verified if they are a part of a potential solution path.

The idea of using clearance information to define a safety certificate, i.e., a hypersphere in both $\mathcal{C}_{free}$ and $\mathcal{C}_{obst}$ has been considered [3] [25]. As shown in Figure 2.5, when a *configuration* is collision checked, the planner stores the *clearance* and defines a region, that will have the same state (i.e., in $\mathcal{C}_{free}$ or in $\mathcal{C}_{obst}$) as the center of the hypersphere (the *configuration* that was collision checked). The use of hyperspheres improves the efficiency of PRM by reducing unnecessary collision checks. In both of the sampling and connection processes, the nearest neighboring hyperspheres of the *configuration* are searched. If the *configuration* is located inside of a hypersphere, it can forgo the collision detection since its state of validity is known. The benefits



(a) An existing hypersphere

(b) Sampling within a hypersphere

(c) Sampling outside any hyperspheres causing a validity test

Figure 2.5: PRM sampling using hyperspheres: (a) an existing hypersphere, (b) sampling within a hypersphere, and (c) sampling outside any hyperspheres causing a validity test.

of using hypersphere increases quickly in spaces that are relatively free of obstacles. However, in a high dimensional space with many obstacles, it takes more time before the benefit of using hypersphere becomes substantial [3]. Also, the trade-off between fewer collision detection calls and increased neighbor finding calls is unclear.

2.4 Medial Axis PRM

Another solution to solve the narrow passage problem efficiently is to sample inside narrow passages but as far away from the obstacles as possible [26]. The medial axis or generalized Voronoi diagram has this appealing property [2][8][9][10][14][16]. The medial axis of a polyhedron is the set of all points equidistant to two or more obstacle boundaries [7]. The medial axis has one lower dimension than $\mathcal{C}_{space}$, but it is still a complete representation of a motion planning problem (e.g., the medial axis in a two dimensional problem is a one dimensional graph-like structure, as shown in Figure 2.6). The medial axis provides high *clearance*, and thus safe paths for a robot to travel. Unfortunately, it is inefficient to compute the exact medial axis.

Medial Axis PRM (MAPRM) [26] generates samples on or near the medial axis of free space. Based on the clearance of a configuration q, MAPRM retracts

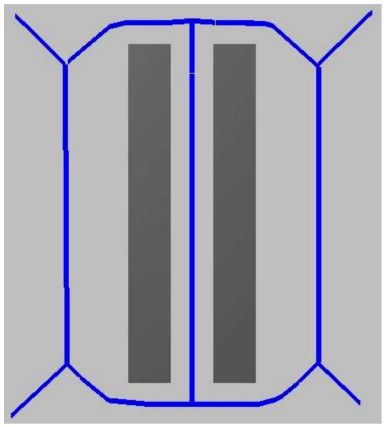


Figure 2.6: MAPRM roadmap for a 2D point robot composed of 1000 samples.

q sampled from $\mathcal{C}_{obst}$ or $\mathcal{C}_{free}$ onto the medial axis of the free space without the explicit computation of the medial axis. Shown in Algorithm 3, MAPRM begins by uniformly generating a random configuration q in $\mathcal{C}_{space}$ (Figure 2.7(a)). Using the clearance, the *witness* configuration w, i.e., the closest configuration on the boundary of $\mathcal{C}_{obst}$ to q, can be computed. If q is is in $\mathcal{C}_{free}$, the retraction direction is set to $\vec{wq}$ and the start configuration q does not change. If q is in $\mathcal{C}_{obst}$, the retraction direction is set to $q\vec{w}$ and the start configuration q is set to w. Starting at q, MAPRM moves the sample in the direction of $\vec{v}$ with step size *dist* until a configuration q' with a different witness point than w is found (Figure 2.7(b)). Then, a binary search is performed between q and q' with a resolution parameter δ to find a configuration m, which is at most δ distance away from the actual medial axis, as shown in Figure 2.7(c). After all random samples are retracted onto the medial axis, MAPRM tries to connect valid configurations and answer queries. Since MAPRM samples in both $\mathcal{C}_{obst}$ and $\mathcal{C}_{free}$, the number of samples found in the narrow passages is less dependent of the volume of the passage. Uniform MAPRM (UMAPRM) improves on MAPRM so that it generate samples uniformly on the medial axis of $\mathcal{C}_{free}$ [28]. However, it does

Algorithm 2 Initialize

Output: Configuration q, w, direction $\vec{v}$

1: $q \leftarrow$ RANDOMCFG()
2: $w \leftarrow$ WITNESS(q)
3: **if** $q \in \mathcal{C}_{free}$ **then**
4: $\quad \vec{v} \leftarrow \vec{wq}$
5: **else**
6: $\quad \vec{v} \leftarrow \vec{qw}$, $q \leftarrow w$
7: **return** $(q, w, \vec{v})$

Algorithm 3 Sample MA [26]

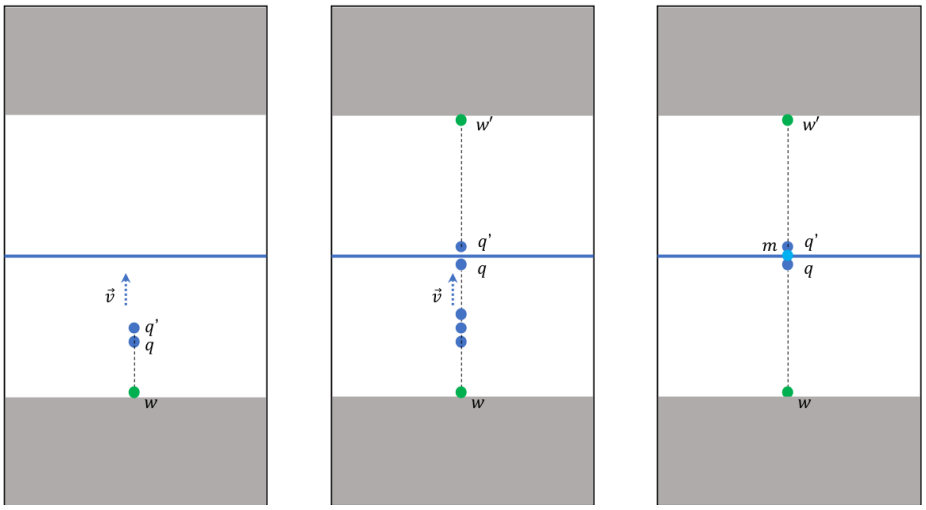
Output: Configuration m on the medial axis

1: $(q, w, \vec{v}) \leftarrow$INITIALIZE()
2: $q' \leftarrow q$
3: **while** WITNESS(q') $= w$ **do**
4: $\quad q \leftarrow q'$
5: $\quad q'$ moves in direction $\vec{v}$ with step size $dist$
6: BINARYSEARCH(q, q', δ) for a configuration m which has two nearest witness points
7: **return** m

not make MAPRM significantly more efficient.

MAPRM greatly increases the probability of sampling in a narrow corridor. Moreover, MAPRM works well in two or three dimensions. However, the bottleneck of MAPRM is the retraction process as it needs to invoke collision detection checks often and compute clearance of configurations to validate q on each step of the retraction. Due to the difficulty in computing witnesses, MAPRM cannot easily or efficiently be applied to high dimension spaces.

A witness approximation method that can be applied to high dimensional problems was introduced in [20]. It casts rays out in multiple random directions and moves configurations along each ray by a given step size until the collision state (i.e.



(a) Initialize (b) Retract (c) Binary search

Figure 2.7: Process of generating samples on the medial axis: (a) initializing the sample and retraction direction, (b) retracting a sample to the medial axis, and (c) binary search for a configuration that is ϵ-close to the medial axis.

valid and invalid) changes on a specific ray. Then, a binary search is used to find a more accurate approximation of the witness point of q. As the number of random stepping directions is increased, the approximate witness point approaches the true witness point. With the approximated witness point, MAPRM can be applied to solve motion planning problems with high DOFs. However, this approach is extremely inefficient because it invokes a *collision detection* routine very frequently.

As seen, no PRM variant simultaneously offers safety and efficient planning.

3. IMPROVING THE EFFICIENCY OF MEDIAL AXIS SAMPLING

In this section, we present and describe our two approaches that efficiently sample on the medial axis. The first approach, Medial Axis Bridge, uses randomness and filtering, similar to the Gaussian PRM and Bridge Test PRM. The second approach, Medial Axis Spherical Step, uses valid hyperspheres to avoid unnecessary collision detection calls.

3.1 Medial Axis Bridge

In Medial Axis Bridge (Algorithm 4 and Figure 3.1), the algorithm first finds the witness point w of a start configuration q, updates q, and computes the retraction direction $\vec{v}$ using the same approach as MAPRM, as shown in Figure 3.1(a). Then, the algorithm finds a configuration q' at a random distance away in the direction of $\vec{v}$ based on a Gaussian distribution, shown in Figure 3.1(b). Then, the witness w' of q' is found. If w and w' are not the same, $\overline{qq}'$ crosses the medial axis. Thus, the algorithm performs a binary search between q and q' with a resolution δ until a configuration m with 2 witness points is found, as shown in Figure 3.1(c). If the witness point w' does not change, q' is discarded and a new q' is attempted, as shown in Figure 3.1(d). After all random samples are retracted onto the medial axis, the algorithm starts to connect nearby samples as in MAPRM.

Recall that MAPRM steps in $\vec{v}$ direction (Algorithm 3) until a configuration which has two witness points is found. Because at each step of the retraction, a costly collision checking routine is invoked, MAPRM is inefficient in the retraction process. Instead, Medial Axis Bridge bypasses collision detection by jumping instead of stepping to the medial axis. In the cases that we could not find the configuration q' whose witness point w' is different from w, the algorithm invokes only two collision

Algorithm 4 Sample MA with Bridge

Output: Configuration m on the medial axis

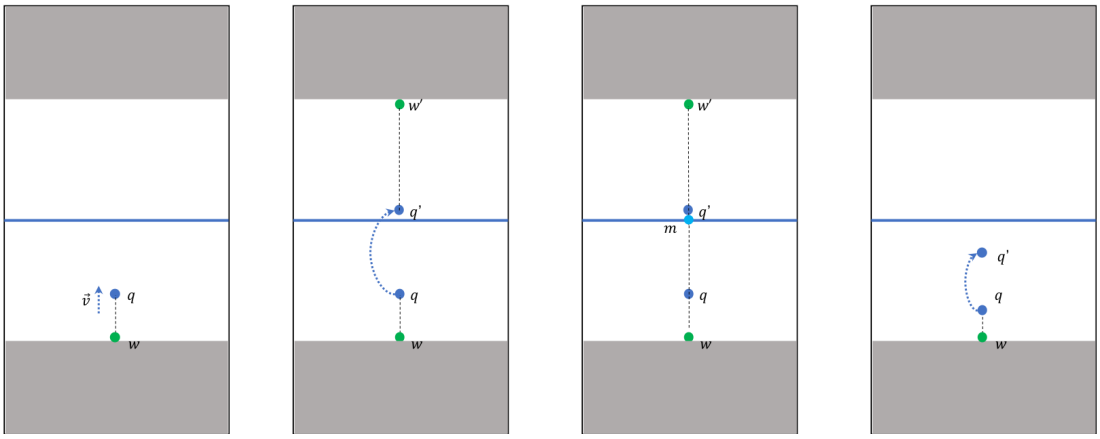
1: $(q, w, \vec{v}) \leftarrow$INITIALIZE()
2: **for** $i \leftarrow \emptyset$ to $numTries$ **do**
3: $q' \leftarrow$ RANDOMJUMP$(\mu, \vec{v}, q)$
4: $w' \leftarrow$WITNESS(q')
5: **if** $w \neq w'$ **then**
6: BINARYSEARCH(q, q', δ) for a configuration m which has two nearest witness points
7: **return** m
8: **return** null or failure

detection calls, which is a comparatively inexpensive test, and it starts the next iteration immediately.

Depending on the Gaussian distribution that we choose, the performance of Medial Axis Bridge varies. Since this distribution is environment dependent, environments may each have a unique Gaussian distribution by which Medial Axis Bridge performs most efficiently. For example, in an environment with sparse obstacles and a relatively small-sized robot, a Gaussian distribution with a large mean parameter μ would provide better performance than a Gaussian distribution with a small μ value because the environment has a greater distance between obstacles.

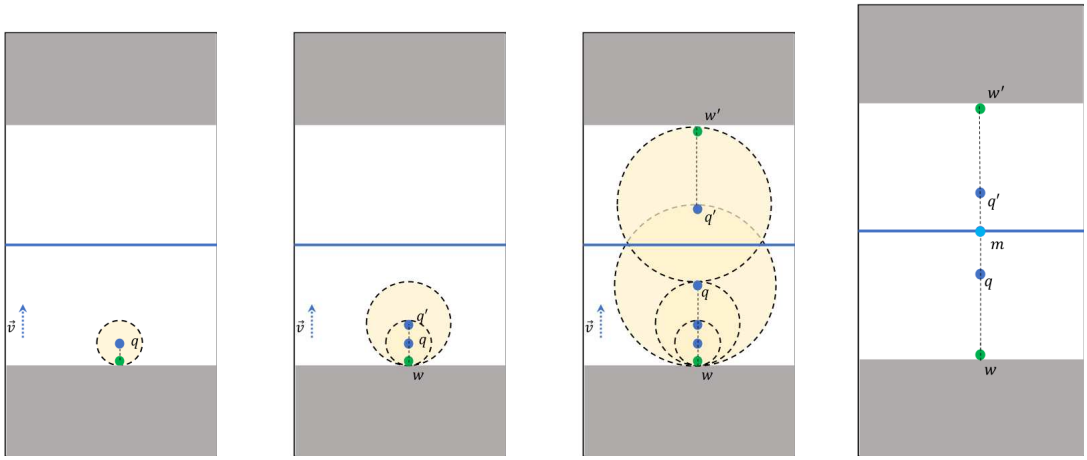
3.2 Medial Axis Spherical Step

In Medial Axis Spherical Step (Algorithm 5 and Figure 3.2), the algorithm first finds the witness point w of the start configuration q and initializes the direction $\vec{v}$, as shown in Figure 3.2(a). It iteratively exploits the valid hypersphere centered at q with radius of the clearance of q to update q' as the furthest point in the hypersphere in direction $\vec{v}$. A collision detection test is performed on each q' to find its witness point w' until w and w' are different (Figure 3.2(b) and Figure 3.2(c)). Then, we perform a binary search between resulting q and q' until a configuration m with two



(a) Initialize (b) Jump (c) Binary search (d) Unsuccessful sample

Figure 3.1: Process of Medial Axis Bridge samples on the medial axis: (a) initializing the sample and retraction direction, (b) retracting the sample to the medial axis, and (c) binary search for a configuration that is ϵ-close to the medial axis. (d) Shows an unsuccessful sampling attempt.



(a) Initialize (b) Stepping 1 (c) Stepping j (d) Binary search

Figure 3.2: Process of Medial Axis Spherical Step sampling on the medial axis: (a) initializing the sample and retraction direction, (b-c) stepping to find the medial axis, and (d) binary search for a configuration that is ϵ-close to the medial axis.

Algorithm 5 Sample MA with Spherical Step

Output: Configuration m on the medial axis

1: $(q, w, \vec{v}) \leftarrow$INITIALIZE()
2: $q' \leftarrow q$
3: **while** WITNESS(q') =WITNESS(q) **do**
4: $\quad q \leftarrow q'$
5: $\quad q' \leftarrow q + \vec{v} \cdot$ CLEARANCE(q)
6: BINARYSEARCH(q, q', δ) for a configuration m which has two nearest witness points
7: **return** m

witness points is found, shown in Figure 3.2(d).

Medial Axis Spherical Step combines the idea of using valid hyperspheres and MAPRM to make the retraction step of MAPRM more efficient by eliminating unnecessary collision checks. MAPRM requires collision checking on each configuration along the retraction. On the other hand, Medial Axis Spherical Step reasons about the computed clearance and witness point intelligently by exploiting valid hyperspheres to eliminate collision checks within the coverage of the hypersphere. By efficiently stepping towards the medial axis, Medial Axis Spherical Step decreases the total number of collision checks and can accelerate the sampling process.

4. EXPERIMENTS

In this chapter, we present our experiment results and analysis. We describe our experimental setup, and relay our comparison of our methods to PRM and MAPRM in different environments.

4.1 Experimental Setup

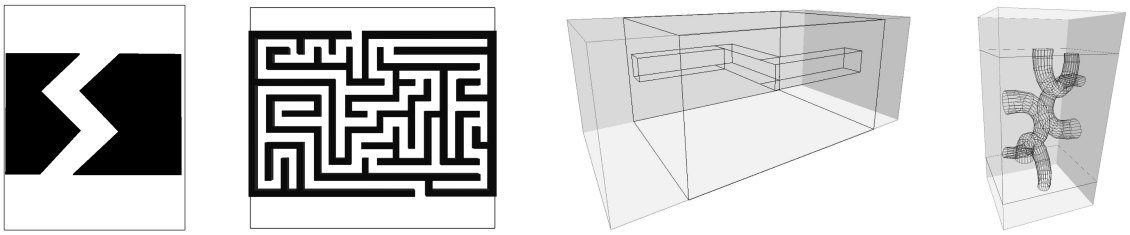
PRM, MAPRM, Medial Axis Bridge, and Medial Axis Spherical Step are all implemented in a C++ motion planning library. The Proximity Query Package (PQP) is used for collision detection computations [15].

In our experiments, the stopping condition is to generate a roadmap with 100 samples. Connections are attempted between a configuration and its k-nearest neighbors where k = 10 according to Euclidean distance in $\mathcal{C}_{space}$. We used a straight line local planner with bisection evaluation.

We conducted our experiments in 2-dimensional zigzag (Figure 4.1(a)), 2-dimensional maze (Figure 4.1(b)), 3-dimensional S tunnel (Figure 4.1(c)), and 3-dimensional maze tunnel (Figure 4.1(d)) since they all are challenging problems with narrow passages. We ran trials with different μ value for Gaussian distribution and found a relatively good value for μ in each environment. We set μ to be 1.0, 1.45, 0.75, 1.0 for 2-dimensional Zigzag, 3-dimensional Maze, 3-dimensional S-Tunnel, and 3-dimensional Maze Tunnel respectively.

4.2 Experiment Discussion

We compare sampling time, connection time, and planning time. As shown in Figure 4.2(a), Medial Axis Bridge and Medial Axis Spherical Step reduce the sampling time of MAPRM by around 50%.



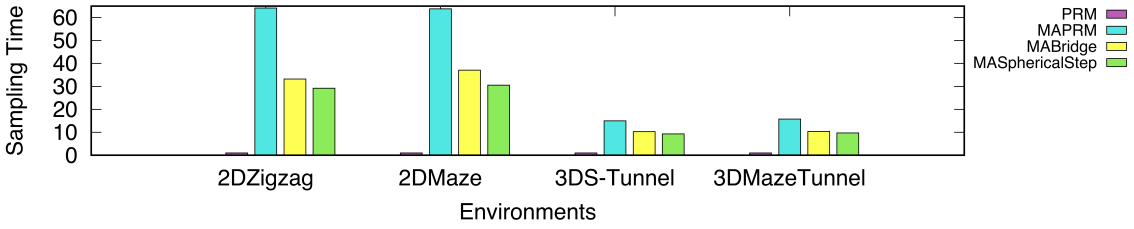
(a) 2D Zigzag (b) 2D Maze (c) 3D S-Tunnel (d) 3D Maze Tunnel

Figure 4.1: Environments used in experimental analysis.

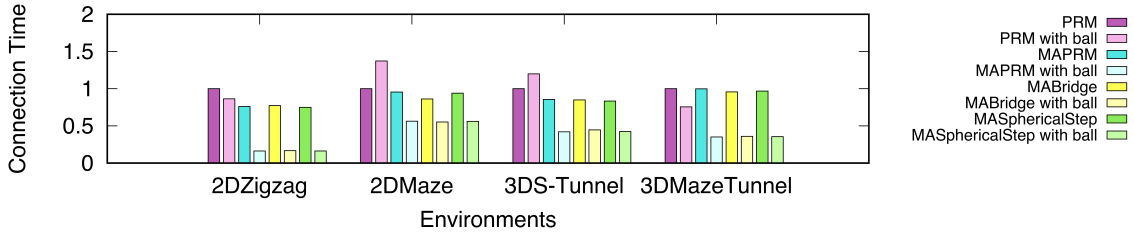
We combine medial axis methods with hyperspheres, which are labeled with "with ball" in Figure 4.2 [3][25]. From Figure 4.2(b), we see the improvement in time by using hyperspheres during the connection phase across all medial axis methods. In all environments, the connection time decreased by approximately 50%. On the other hand, there is little-to-no decrease of connection time in PRM when hyperspheres are used because it does not have clearance already computed — PRM needs to compute the clearance information to exploit hyperspheres.

In Figure 4.2(c), we see the total planning time of all Medial Axis approaches with hyperspheres is less than that of PRM with hyperspheres. Meanwhile, Medial Axis Bridge and Medial Axis Spherical Step are more efficient than MAPRM.

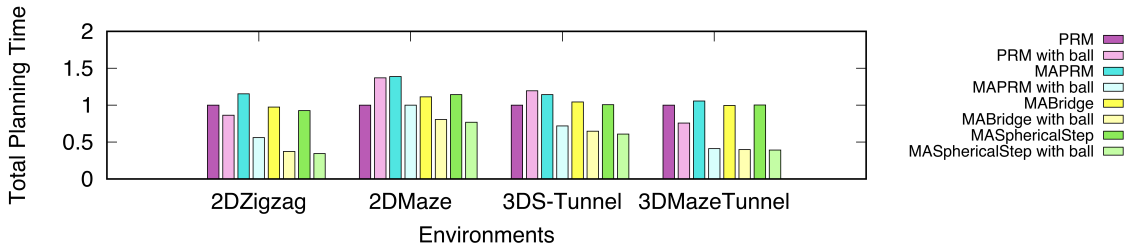
Medial Axis Spherical Step is possibly a better approach than Medial Axis Bridge, because it makes the most use of the clearance information and requires no additional parameters. It guarantees the sample with a step size that is furthest from the current position of a sample but within the safe range, whereas Medial Axis Bridge might require many failed attempts before successfully generating a sample. Nevertheless, both Medial Axis Bridge and Medial Axis Spherical Step improve MAPRM's efficiency while keeping its safety property as they sample on the medial axis.



(a) Sampling time



(b) Connection time



(c) Total planning time

Figure 4.2: Experiment results: (a) sampling time, (b) connection time, and (c) total planning time.

5. CONCLUSION

In this work, we presented and analyzed two medial axis sampling approaches that improve the efficiency of MAPRM while retaining its appealing properties, i.e., high clearance. We also improve the efficiency of connection in MAPRM by utilizing previously computed clearance information. We provide empirical evidence that demonstrates these improvements.

We have begun experimenting on solving queries with our approaches and currently find the same conclusions. However, some anomalies appeared and require further investigation that we leave to future work. Additionally, in the future, we will investigate the impact of high dimensionality on our approaches and their combination with approximate methods [20]. Further, we want to apply our approaches to kinematic systems.

REFERENCES

[1] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for $3D$ workspaces. *Proceedings of the Third Workshop on the Algorithmic Foundations of Robotics on Robotics : The Algorithmic Perspective: The Algorithmic Perspective*, pages 155–168, 1998.

[2] F. Aurenhammer. Voronoi diagrams–a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, September 1991.

[3] J. Bialkowski, M. Otte, S. Karaman, and E. Frazzoli. Efficient collision checking in sampling-based motion planning via safety certificates. *The International Journal of Robotics Research*, 35(7):767–796, 2016.

[4] R. Bohlin and L. E. Kavraki. Path planning using lazy PRM. *IEEE Trans. on Robotics and Automation*, 1:521–528, April 2000.

[5] V. Boor, M. H. Overmars, and A. F. van der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. *IEEE Trans. on Robotics and Automation*, 2:1018 – 1023, May 1999.

[6] J. Canny. Collision detection for moving polyhedra. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(2):200–209, February 1986.

[7] J. Denny, E. Greco, S. Thomas, and N. M. Amato. $MARRT$: Medial axis biased rapidly-exploring random trees. *IEEE Trans. on Robotics and Automation*, pages 90–97, May 2014.

[8] M. Foskey, M. Garber, M. C. Lin, and D. Manocha. A voronoi-based hybrid motion planner. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001.

[9] K. Hoff, T. Culver, J. Keyser, M. C. Lin, and D. Manocha. Interactive motion planning using hardware-accelerated computation of generalized voronoi diagrams. *IEEE Trans. on Robotics and Automation*, 3:2931–2937, April 2000.

[10] C. Holleman and L. E. Kavraki. A framework for using the workspace medial axis in PRM planners. *IEEE Trans. on Robotics and Automation*, 2:1408–1413, April 2000.

[11] D. Hsu, T. Jiang, J. Reif, and Z. Sun. The bridge test for sampling narrow passages with probabilistic roadmap planners. *IEEE Trans. on Robotics and Automation*, 3:4420–4426, September 2003.

[12] D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. *Proceedings of the Third Workshop on the Algorithmic Foundations of Robotics on Robotics : The Algorithmic Perspective: The Algorithmic Perspective*, pages 141–153, 1998.

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

[14] C. P. Kenneth, K. H. Iii, M. C. Lin, and D. Manocha. Randomized path planning for a rigid body based on hardware accelerated voronoi sampling. *In Proc. Workshop on Algorithmic Foundation of Robotics*, 2000.

[15] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha. Fast distance queries with rectangular swept sphere volumes. *IEEE Trans. on Robotics and Automation*, 4:3719–3726, April 2000.

[16] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[17] J. C. Latombe. Motion planning: A journey of robots, molecules, digital actors, and other artifacts. *International Journal of Robotics Research*, 18, November 1999.

[18] S. M. LaValle. *Planning Algorithms*, pages 153–205. Cambridge University Press, 2006.

[19] S. M. LaValle and J. James J. Kuffner. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.

[20] J.-M. Lien, S. L. Thomas, and N. M. Amato. A general framework for sampling on the medial axis of the free space. *IEEE Trans. on Robotics and Automation*, 3:4439–4444, January 2003.

[21] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22(10):560–570, October 1979.

[22] B. Mirtich. V-clip: Fast and robust polyhedral collision detection. *ACM Trans. Graph.*, 17:177–208, 1998.

[23] C. L. Nielsen and L. E. Kavraki. A two level fuzzy PRM for manipulation planning. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3:1716–1721, October 2000.

[24] J. H. Reif. Complexity of the movers problem and generalizations. *IEEE Symp. Foundations of Computer Science (FOCS)*, page 421427, October 1979.

[25] A. C. Shkolnik and R. Tedrake. Sample-based planning with volumes in configuration space. *CoRR*, abs/1109.3145, 2011.

[26] 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. *IEEE Trans. on Robotics and Automation*, 2:1024–1031, May 1999.

[27] H. Yeh, S. Thomas, D. Eppstein, and N. M. Amato. UOBPRM: A uniformly distributed obstacle-based PRM. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2655–2662, October 2012.

[28] H. C. Yeh, J. Denny, A. Lindsey, S. Thomas, and N. M. Amato. UMAPRM: Uniformly sampling the medial axis. *IEEE Trans. on Robotics and Automation*, pages 5798–5803, May 2014.