A General Framework for Sampling on the Medial Axis of the Free Space *

Jyh-Ming Lien Shawna L. Thomas Nancy M. Amato
Department of Computer Science
Texas A&M University
{neilien,sthomas,amato}@cs.tamu.edu

Abstract. *We propose a general framework for sampling the configuration space in which randomly generated configurations, free or not, are retracted onto the medial axis of the free space. Generalizing our previous work, this framework provides a template encompassing all possible retraction approaches. It also removes the requirement of exactly computing distance metrics thereby enabling application to more realistic high dimensional problems. In particular, our framework supports methods that retract a given configuration* exactly *or* approximately *onto the medial axis. As in our previous work,* exact *methods provide fast and accurate retraction in low (2 or 3) dimensional space. We also propose new* approximate *methods that can be applied to high dimensional problems, such as many DOF articulated robots. Theoretical and experimental results show improved performance on problems requiring traversal of narrow passages. We also study tradeoffs between accuracy and efficiency for different levels of approximation, and how the level of approximation effects the quality of the resulting roadmap.*

1 Introduction

Due to the computational infeasibility of complete motion planning algorithms, recent attention has focused on probabilistic methods which sacrifice completeness for computational feasibility. In particular, several algorithms, known collectively as *probabilistic roadmap methods* (PRMs), have been shown to perform well in a number of practical situations, see, e.g., [9]. The idea behind these methods is to create a graph (or roadmap) of randomly generated collision-free configurations. Connections between these nodes are made by a simple and fast local planning method. Actual global planning is then carried out on the roadmap. These methods run quickly and are easy to implement. Unfortunately, simple situations exist in which they perform poorly, e.g., when paths are required to pass through narrow passages in configuration space.

The *medial axis*, or *generalized Voronoi diagram*, has a long history in motion planning, see [2, 4, 5, 6, 11, 15]. This is because the medial axis $\mathrm{MA}(C_{free})$ of the free space C_{free} (the set of all collision-free configurations) has lower dimension than C_{free} but is still a complete representation for motion planning purposes. Paths on the medial axis have appealing properties such as large clearance from obstacles. However, the medial axis is difficult and expensive to compute explicitly, particularly in higher dimensions. The Medial Axis PRM (MAPRM) [16, 17] combines these two approaches by generating random networks whose nodes lie on the medial axis of C_{free} which yields improved performance on problems requiring traversal of narrow passages.

Previous work developed MAPRM for two dimensional C-spaces [16] and rigid, convex bodies in three dimensional space [17]. In this paper, we present a general MAPRM framework. Our generalized framework extends MAPRM to arbitrary bodies and high DOF robots. The framework enables sampling on the medial axis in high (> 6) dimensional configuration space through the use of *approximate* methods for computing clearance and penetration depth.

2 Related Work

PRMs are easy to implement, run quickly, and are applicable to a wide variety of robots. Various sampling schemes and local planners have been used, see [8, 9, 14]. A shortcoming of these methods is their poor performance on problems requiring paths through narrow passages in the free space. This is a direct consequence of how the nodes are sampled from C_{free}. For example, uniform sampling over C_{free}, is unlikely to provide any samples in small volume corridors. Intuitively, such narrow corridors may be characterized by their large surface area to volume ratio. Several techniques have been proposed to increase the number of nodes sampled in such narrow corridors [1, 3, 7, **?**, 17].

A PRM variant, MAPRM, was proposed in [16, 17]. MAPRM generates random networks whose nodes lie on the medial axis of the free C-space. It is difficult and expensive to compute the medial axis explicitly, particularly in higher dimensions. As shown in [16, 17] for low dimensional C-space, it is possible, however, to efficiently retract any sampled configuration, free or not, onto the medial axis of the free space without having to compute the medial axis. Sampling and retracting in this way has been shown to give improved performance on problems requiring traversal of narrow passages for rigid bodies in two or three dimensions. Even for 6D C-space, MAPRM uses an inefficient brute force method to find penetrations between polyhedral objects. The requirement for exact computation of clearance and penetration depth in C-space is the primary rea-

[1]This research supported in part by NSF Grants ACI-9872126, EIA-9975018, EIA-0103742, EIA-9805823, ACR-0113971, CCR-0113974, EIA-9810937, EIA-0079874, and by the Texas Higher Education Coordinating Board grant ATP-000512-0261-2001. Thomas is supported in part by an NSF Graduate Research Fellowship.

son MAPRM can not be applied in arbitrary dimensions. In the following section, we introduce a general framework to cope with these difficulties while still maintaining the good properties of MAPRM.

3 Generalized MAPRM Framework

In this section we present a general framework for MAPRM. Let C be the C-space, C_{obst} be the C-obstacle, and C_{free} be $C \setminus C_{obst}$. We begin by sketching the MAPRM strategy. To retract any sampled configuration onto the medial axis of C_{free}, all that is needed is the closest point on ∂C_{obst}. If the sampled configuration $p \in C_{free}$, MAPRM computes the closest point, q, on ∂C_{obst} to p, and pushes p away from q until the nearest point on ∂C_{obst} is different. If $p \in C_{obst}$, MAPRM first pushes the configuration to C_{free} and then retracts it to the medial axis as before. Figure 1 shows the extended retraction map, $r(p)$, for 2D C-space.

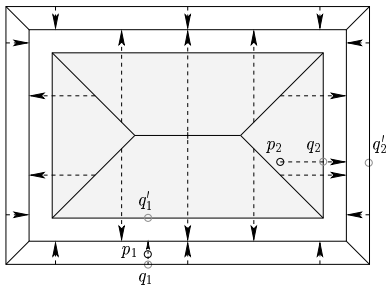


Figure 1: The extended retraction map, $r(p)$. $p_1 \in C_{free}$. q_1 is the witness for p_1's clearance, and q_1' is the second witness when p_1 reaches the medial axis. $p_2 \in C_{obst}$. q_2 is p_2's witness for penetration depth. When p_2 is pushed into free space, q_2 will be clearance witness of p_2 and q_2' is p_2's second clearance witness when p_2 reaches the medial axis.

Algorithm 3.1 Basic MAPRM Framework

Preprocessing:

Input. N, the number of nodes to generate.
Output. N nodes in C_{free} connected into a roadmap.

1: **repeat**
2: Sample a configuration p from C-space.
3: **if** $p \in C_{free}$ **then**
4: q = **NearestContactCfg_Clearance(**p**)**
5: Set retraction direction $\vec{v} = \overrightarrow{qp}$ and start point $s = p$
6: **else**
7: q = **NearestContactCfg_Penetration(**p**)**
8: Set retraction direction $\vec{v} = \overrightarrow{pq}$ and start point $s = q$
9: **end if**
10: Starting at configuration s, move robot in direction $\vec{v}$ until it has two nearest boundary points (i.e., is on medial axis).
11: **until** N nodes have been generated
12: Build connections between nodes using local planners.

The general MAPRM framework for roadmap construction is sketched in Algorithm 3.1. It involves uniform sampling in C, followed by application of the extended retraction map. If the initial sample is free, we compute a configuration q witnessing the minimum clearance using the *NearestContactCfg_Clearance* subroutine in Line 4. If the initial sample is in collision, a witness q to the minimum penetration depth is computed by the *NearestContactCfg_Penetration* subroutine in Line 7. Let $\ell_r(p)$ denote the reaction distance of $r(p)$. The cost of retracting p to the $\mathrm{MA}(C_{free})$ is $\lceil \frac{\ell_r(p)}{d} \rceil \times \mathrm{NC}(p)$, where d is the resolution of the workspace and $\mathrm{NC}(p)$ is the cost of finding the nearest contact configuration.

Note that Algorithm 3.1 makes no assumption about the dimension of C, distance metrics, the robot, or the environment (e.g., convexity, articulation). In fact, only the clearance and penetration depth computations depend on these properties. That is, the only steps of Algorithm 3.1 that depend on the problem instance are *NearestContactCfg_Clearance* and *NearestContactCfg_Penetration*.

For convex rigid bodies in two and three dimensions, there exist algorithms and libraries [10, 13] to efficiently compute clearance and penetration depth. Thus, in this case, the clearance and penetration depth, and configurations realizing them, can be computed exactly. Such methods were discussed in [16, 17].

Unfortunately, computing clearance and penetration depth for higher DOF robots is hard because it is difficult to define good distance metrics. For example, the previous approach [17] of defining the shortest distance between two configurations as purely translation is not appropriate for articulated robots. Indeed, to the best of our knowledge, no efficient algorithms exist for finding penetration depth for non-convex or articulated bodies. In this paper, we propose the use of approximate methods for computing clearance and penetration for high dimensional C.

The MAPRM algorithms proposed in this paper are classified in Table 1 according to their level of approximation. MAPRM [16, 17] uses exact computation for clearance and penetration depth. MAPRM$^{\sim}$ uses exact computation for clearance but an approximate approach for penetration depth and can thus be applied to environments containing non-convex rigid bodies[1]. MAPRM$^{\approx}$ uses an approximate method for both clearance and penetration depth and can be applied to arbitrarily high DOF robots.

Algorithm	Clearance Computation	Penetration Computation
MAPRM	exact	exact
MAPRM$^{\sim}$	exact	approximate
MAPRM$^{\approx}$	approximate	approximate

Table 1: MAPRM algorithms for various approximation levels.

4 MAPRM

MAPRM was first developed for a point robot in the plane

[1] MAPRM$^{\sim}$ uses the approximate method only if it is necessary. Thus in some situations, e.g., the robot is entirely contained in an obstacle, penetration depth can be calculated just like clearance.

[16] and then extended to convex rigid bodies in 3D [17].

4.1 MAPRM for a Point Robot in 2D

When the environment is composed of polygonal objects, clearance or penetration depth for a given configuration is just the shortest distance from this point to the boundary of the polygons. This algorithm plays the roles of both NearestContactCfg_Clearance and NearestContactCfg_Penetration in the framework (Algorithm 3.1). For 2D C-space, MAPRM has been shown, theoretically and empirically, to increase sampling in narrow corridors [16].

Algorithm 4.1 NearestContactCfg in 2D

Preprocessing:

Input. A configuration p.

Output. The nearest contact configuration q from p.

1: Find nearest configuration q on ∂C_{free} to p by computing the distance from p to ∂C_{free}

2: **return** q.

4.2 MAPRM for a Rigid Body in 3D

In MAPRM for convex rigid bodies in three dimensions [17], the objective is to compute the nearest contact configuration in ∂C_{obst} without explicitly computing C_{obst}.

Let p be a sampled configuration. If $p \in C_{free}$, then the nearest contact point is a witness realizing the clearance (Algorithm 4.2), which is provided by several algorithms and collision detection packages [10, 13].

Algorithm 4.2 NearestContactCfg_Clearance in 3D

Input. A collision-free configuration p.

Output. The nearest contact configuration q from p or **failure**.

1: Find a point, b, on B and a point, r, on Robot such that $\mathrm{dist}(b, r)$ is smallest.

2: **return** $p + \overrightarrow{r\,b}$.

If $p \in C_{obst}$, we compute the nearest contact point by computing its penetration depth (Algorithm 4.3). If all objects in the environment are convex, the penetration can be computed efficiently [12, 13]. Otherwise, a brute force method which tests all feature pairs is applied. Clearly, the brute force approach is not practical for large complex environments. MAPRM for a convex, rigid body in 3D has been shown to increase sampling in narrow corridors [17].

Algorithm 4.3 NearestContactCfg_Penetration in 3D

Input. An in-collision configuration p.

Output. The nearest contact configuration q from p or **failure**.

1: **if** Robot and Obstacles are all convex objects **then**

2: Use Lin-Canny closest features algorithm [12].

3: **else**

4: Use brute force method [16, 17].

5: **end if**

5 Approximate Variants of MAPRM

In this section, we present an approximate approach which enables us to apply the MAPRM philosophy in more general situations, such as high DOF robots. There are several complications with extending the strategy for 2D and 3D MAPRM to high DOF robots. For instance, finding the nearest contact configuration requires the ability to compute witnesses for clearance and penetration depth. Unfortunately, the exact closest contact configuration in C cannot be computed without computing the ∂C_{obst}, a computationally expensive process which PRM methods are designed to avoid.

We are able, however, to approximate the C-space clearance (or penetration) of a configuration without computing ∂C_{obst}. Let $Cl(p, \vec{v})$ be the C-space clearance of configuration p in direction $\vec{v}$. Then the approximate clearance is defined as: $Cl(p) = \min(\{Cl(p, \vec{v_i}) : i = 1 \ldots N\})$. To compute $Cl(p, \vec{v})$, we walk out from p towards $\vec{v}$ until the collision state changes. As we increase N, the approximate clearance approaches the actual clearance. Penetration is defined in the same way; i.e., $Pt(p) = \min(\{Pt(p, \vec{v_i}) : i = 1 \ldots N\})$. Algorithm 5.1 describes the process. It can be used to implement the *NearestContactCfg_Clearance* or *NearestContactCfg_Penetration* subroutines in Algorithm 3.1. The time complexity of Algorithm 5.1 is $O\left(N \times \lceil \frac{\ell}{d} \rceil\right) \times \mathrm{T}(CD)$, where ℓ is $Cl(p)$ or $Pt(p)$, d is the resolution of the workspace, and $\mathrm{T}(CD)$ is the cost for collision detection. The accuracy of the approximation depends on N and d. However, while N has a more profound effect on accuracy, $Cl(p)$ and $Pt(p)$ using large d can be refined by the bisection search.

MAPRM$^{\sim}$ approximates the penetration depth in the basic MAPRM framework (Algorithm 3.1) while clearances are computed explicitly. This allows the extension to non-convex rigid bodies.

MAPRM$^{\approx}$ approximates both clearance and penetration depth in the basic MAPRM framework. By approximating both clearance and penetration, we can apply the MAPRM sampling strategy to arbitrary robots with high DOF.

Algorithm 5.1 NearestContactCfg in for general C-space.

Input. A configuration c.

Output. An approximately closest contact configuration c'.

1: Let *CollStat()* be the collision detection function.

2: Let $c_i = c$, $i = 1$ to N.

3: Randomly create N normalized vectors, $\vec{v}_1$ to $\vec{v}_N$.

4: **while** true **do**

5: **for** $i = 1$ to N **do**

6: $c_i = c_i + \vec{v}_i$.

7: **if** *CollStat*$(c_i) \neq$ *CollStat*(c) **then**

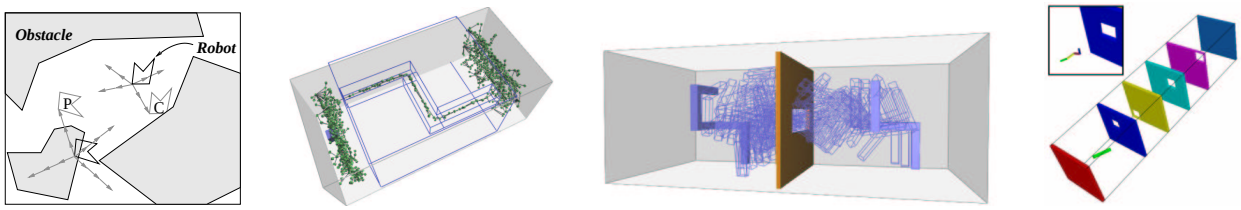
8: **if** $(c \in C_{obst})$ **then** $c_i = c_i - \vec{v}_i$.

9: **return** c_i.

10: **end if**

11: **end for**

12: **end while**



(a) (b) (c) (d)

Figure 2: (a) We approximate a configuration's C-space clearance (penetration) by finding its clearance (penetration) in several random directions. The configuration marked as "c" ("p") is an approximate contact configuration witnessing the clearance (penetration). (b) **S-Tunnel** is contained in a $6.5 \times 6.5 \times 14$ bounding box. The robot is a cube with side lengths of 0.6; the obstacle is a solid box of size $6 \times 6 \times 9$ with the indicated corridor cut through it. The corridor has a 1×1 cross section. The obstacle is in wireframe with the indicated corridor cut through it. An example of a roadmap generated by MAPRM$^{\sim}$ is shown. (c) **Hook** contains a hook-like rigid robot and an obstacle ($200 \times 150 \times 5$) with a hole ($100 \times 30 \times 5$) inside. The robot is composed of 5 blocks ($10 \times 10 \times 50$) with different orientations. The swept volume illustrates a solution path. (d) **Walls** contains six parallel walls ($4 \times 4 \times 0.25$). Each of the four walls in the middle has a (1×1) hole. The robot is between the two leftmost walls. The goal is placed between the two rightmost walls.

6 Experimental Results

In this section, we show how the MAPRM variants perform in practice. PQP [10] and V-Clip [13] are used to provide collision detection and exact closest pair calculations. Options for sampling and connection are controlled under the same condition unless stated otherwise. The execution of each method was terminated when the roadmap contained a component that reached from one mouth of the corridor to the other. Experiments were carried out on an Intel Pentium 4 processor at 1.80 GHz.

We tested examples in 3D environments with a variety of robots (Section 6.1). We also studied the tradeoffs between accuracy and efficiency of different approximate levels (Section 6.2).

6.1 3D Environments

We tested three environments: s-tunnel, hook with a hook-like rigid robot, and walls with both a stick robot and an articulated robot; see Figure 2 (b), (c), and (d). In Table 2, We present experimental results for MAPRM, MAPRM$^{\sim}$, and MAPRM$^{\approx}$, and compare them with results from uniformly sampled PRM. We averaged the results over 10 runs with different seeds.

6.1.1 S-Tunnel environment

In Figure 2(b), the obstacle is composed of 6 convex pieces which enables us to use V-Clip [13] to find exact contact configurations. The robot must pass through the tunnel to solve the query.

Uniform random sampling in Table 2 was unable to solve the query with a roadmap size of 64000 nodes and 11 hours of execution time. In contrast, all MAPRM variations were able to produce a valid solution path. MAPRM$^{\sim}$ takes slightly longer than MAPRM, because the approximation calculation in MAPRM$^{\sim}$ requires more time than the exact calculation. This can be seen in the longer average node generation time of 749.83ms versus 41.01ms for MAPRM. MAPRM$^{\approx}$ is the slowest of the MAPRM algorithms because both clearance and penetration calculations are approximated, requiring more time during node generation.

EXPERIMENTAL RESULTS FOR MAPRM ALGORITHMS

Env.	Method	Samp. time(s)	Connect time(s)	Total time(s)	Solved
S-Tunnel	Uniform	155	35570	39744	**N**
	MAPRM	14	9	22	Y
	MAPRM$^{\sim}$	19	5	26	Y
	MAPRM$^{\approx}$	45	53	213	Y
Hook	Uniform	686	97014	99869	**N**
	MAPRM$^{\sim}$	33	74	120	Y
	MAPRM$^{\approx}$	44	45	105	Y
Walls (Stick)	Uniform	2	161	162	Y
	MAPRM$^{\sim}$	11	30	40	Y
	MAPRM$^{\approx}$	25	35	61	Y
Walls (Articulated)	Uniform	2	119	121	Y
	MAPRM$^{\approx}$	47	51	99	Y

Table 2: For the S-Tunnel environment, MAPRM$^{\sim}$ uses $N = 4$ and MAPRM$^{\approx}$ uses $N = 100$ for penetration and $N = 4$ for clearance. For the Hook environment, MAPRM$^{\sim}$ uses $N = 20$ and MAPRM$^{\approx}$ uses $N = 4$ rays for penetration and 20 rays for clearance. For the Walls environment with the stick robot, MAPRM$^{\sim}$ uses $N = 4$ and MAPRM$^{\approx}$ uses $N = 20$ for clearance and penetration. For the Walls environment with an articulated robot, MAPRM$^{\approx}$ uses $N = 4$ for clearance and penetration.

6.1.2 Hook Environment

In the hook environment (Figure 2(c)), the robot starts from the left side of the environment and the unique solution path requires it to twist through the hole to reach the goal in the right side of the environment.

In Table 2, uniform random sampling was unable to solve the query with a roadmap size of 128000 nodes and 28 hours of execution time. MAPRM could not be used because the environment contains non-convex objects. In contrast, MAPRM$^{\sim}$ and MAPRM$^{\approx}$ were able to solve the query with only 2815 nodes in just a couple minutes. For environments

like this, it is critical that nodes are generated in the narrow corridor. The results demonstrate MAPRM's ability to increase sampling in narrow corridors, even when clearances and penetrations are only approximated.

6.1.3 Walls environment

In our third example (Figure 2(d)), the robot must pass through the holes in the walls to reach the goal at the other end of the environment. We tested both a rigid robot and a 4-link articulated robot.

In Table 2, MAPRM could not be used because the environment contains non-convex objects. Both MAPRM$^{\sim}$ and MAPRM$^{\approx}$ out-perform uniform sampling. Although MAPRM$^{\sim}$ finds a solution faster than MAPRM$^{\approx}$ the difference is not as pronounced as in the s-tunnel experiment. This is because constraints on rotation decrease the performance of MAPRM$^{\sim}$.

The articulated robot has high DOF, so only MAPRM$^{\approx}$ could be applied. MAPRM$^{\approx}$ again beat uniform random sampling by solving the query with a roadmap half the size. The speedup is not as great as the roadmap size reduction because MAPRM$^{\approx}$ requires more time to generate a node (30.69ms on average) than uniform random sampling (0.72ms on average). Note that the time MAPRM$^{\approx}$ lost during node generation was more than made up for during node connection. This shows MAPRM$^{\approx}$'s nodes to be of higher quality than those from uniform random sampling.

6.2 Approximation Study

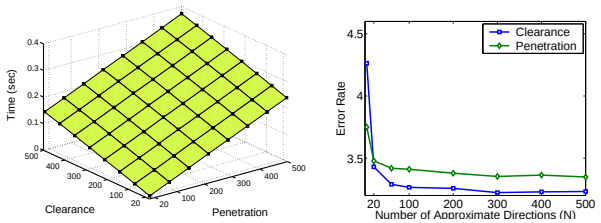
The efficiency of the approximate methods, MAPRM$^{\sim}$ and MAPRM$^{\approx}$ depends on the efficiency of computing the approximate clearance and penetration; e.g., $N \times \lceil \frac{Cl(p)}{d} \rceil$ for C-space clearance. In this section we study how N relates to the minimum time required to find a solution path and how this varies between environments. We are interested in questions such as what environment properties indicate a need for greater clearance accuracy and greater penetration accuracy. To investigate these issues, we varied the value of N for both clearance and penetration calculations.

6.2.1 Accuracy and Computation Time

First, we studied accuracy and computation time by varying N for both clearance and penetration depth. Accuracy is based on the normalized distance between the exact and approximate contact configurations. Larger distances indicate a less accurate result. The computation time is measured based on mean time to generate one contact configuration.

Figure 3(a) shows the computation time for different combinations of N for penetration and clearance. Since the s-tunnel environment has little free space, the computation for clearance is faster than that for penetration depth. It is clear that the computation time grows linearly with N.

Figure 3(b) shows the error rates introduced by approximation. Let error rates for clearance and penetration depth be $ErrorRate = \frac{\sum_{i=1}^{n} dist(cfg_i^e, cfg_i^a)}{\sum_{i=1}^{n} dist(cfg_i^e, cfg_i^o)}$. Then accuracy is defined as the reciprocal of the $ErrorRate$. Here cfg_i^o is the i-th randomly sampled configuration, cfg_i^e is the exact nearest contact configuration for cfg_i^o, and cfg_i^a is an estimated contact configuration. Since $dist(cfg^e, cfg^a)$ depends on properties of the environment, such as the size of the obstacles and the volume of the free space, we normalize the distance by dividing by the exact retraction distance, $dist(cfg^e, cfg^o)$. In the s-tunnel environment, large obstacles will produce a large mean distance between the exact nearest contact configuration and the configuration in collision, and small free space will have small mean exact clearance. From Figure 3(b), one can notice that the error rates drop significantly in the interval of 4 to 100, and little improvement is shown for > 100.



(a) (b)

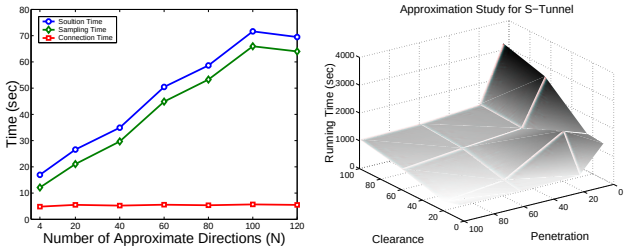
Figure 3: Approximation study using S-Tunnel environment. Average over $55,000$ samples. (a) The computation time for finding one nearest contact configuration for different N. (b) The error rate for finding a nearest contact configuration for different N.

6.2.2 S-Tunnel

Figure 4 shows the results of the approximation study for the s-tunnel environment (Figure 2). The larger N is, the more accurate the contact configuration will be and the more computation time it will take.

For MAPRM$^{\sim}$, the best value for N is 4. Due to the small volume of the tunnel relative to the obstacle, the accuracy of the penetration depth is important and larger values of N should solve the problem faster. However, the robot is so small that most of samples are contained inside the obstacle completely. This is a case in which penetration depth can be calculated exactly. Thus, the probability of using an approximation is relatively small. This is why $N = 4$ is enough to generate a good roadmap.

For MAPRM$^{\approx}$, the best combination was $N = 4$ for clearance and $N = 100$ for penetration. A close second was $N = 20$ for clearance and $N = 100$ for penetration. Because the corridor is small compared to the surrounding obstacle, accurate penetration calculations are critical to the algorithm's success. This is reflected in the fact that the top two clearance/penetration combinations had $N = 100$ for penetration. Planning in the space outside the corridor is trivial and the robot is small compared to the available space. This is reflected in the coarser clearance approximation ($N = 4$ or 20).



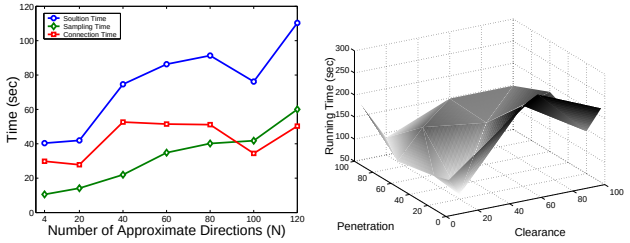
(MAPRM$^{\sim}$) (MAPRM$^{\approx}$)

Figure 4: S-Tunnel approximation study.

6.2.3 Walls

Figure 5 shows the results of the approximation study for the walls environment with a stick robot (Figure 2(a)).

For MAPRM$^{\sim}$, the best value for penetration was $N = 20$, and for MAPRM$^{\approx}$, the best clearance/penetration combination was $N = 20$ for clearance and $N = 20$ for penetration. In this environment, the walls are thinner than in the s-tunnel, so penetration calculations do not need to be approximated at as fine a detail. Also, the spaces between the walls are not cluttered. Planning here is easy to moderate, so only $N = 4$ or 20 is needed for clearance calculations.



(MAPRM$^{\sim}$) (MAPRM$^{\approx}$)

Figure 5: Walls with stick robot approximation study.

7 Conclusion

We have described a general framework for sampling configurations on the medial axis of the free C-space. It enables users to "plug in" appropriate retraction functions according to the properties of given problem. In particular, we propose using exact and approximate methods to implement the retraction functions. Exact methods, MAPRM, for convex rigid bodies in two and three dimensions, and approximate methods, MAPRM$^{\sim}$ and MAPRM$^{\approx}$, for arbitrary configuration space are proposed. Both MAPRM$^{\sim}$ and MAPRM$^{\approx}$ out performs uniform sampling in all studied environments. MAPRM$^{\approx}$out performs MAPRM$^{\sim}$ when rotation is significant for rigid robots. Only MAPRM$^{\approx}$can be used for high DOF robots. We observed the relationship between accuracy, i.e. number of approximate directions, and properties of environments, e.g. clearance and penetration depth.

Acknowledgment

Some of the models studied were provided by Jean-Paul Laumond's and Lydia Kavraki's groups.

References

[1] N. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In P. K. Agarwal, L. E. Kavraki, and M. Mason, editors, *Proc. Workshop Algorithmic Found. Robot.* A. K. Peters, Wellesley, MA, 1998.

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

[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)*, pages 1018–1023, 1999.

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

[5] K. Hoff, T. Culver, J. Keyser, M. Lin, and D. Manocha. Interactive motion planning using hardware-accelerated computation of generalized voronoi diagrams. In *Proceedings of the 2000 International Conference on Robotics and Automation (ICRA 2000)*, 2000.

[6] C. Holleman and L. E. Kavraki. A framework for using the workspace medial axis in prm planners. In *Proceedings of the 2000 International Conference on Robotics and Automation (ICRA 2000)*, pages 1408–1413, San Fransisco, CA, 2000. IEEE Press.

[7] D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In *Proc. 1998 Workshop Algorithmic Found. Robot.*, Wellesley, MA, 1998. A. K. Peters.

[8] L. Kavraki. *Random Networks in Configuration Space for Fast Path Planning*. PhD thesis, Stanford Univ., Stanford, CA, 1995.

[9] 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. Autom.*, 12:566–580, 1996.

[10] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha. Distance queries with rectangular swept sphere volumes. In *Proc. of IEEE Int. Conference on Robotics and Automation*, 2000.

[11] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[12] M. Lin and J. Canny. A fast algorithm for incremental distance calculation. In *Proceedings of International Conference on Robotics and Automation*, pages 1008–1014, 1991.

[13] B. Mirtich. V-Clip: Fast and robust polyhedral collision detection. Technical Report TR97-05, MERL, 201 Broadway, Cambridge, MA 02139, USA, July 1997.

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

[15] C. Pisula, K. Hoff, M. Lin, and D. Manocha. Randomized path planning for a rigid body based on hardware accelerated voronoi sampling. In *Proc. Workshop on Algorithmic Foundations of Robotics (WAFR 2000)*.

[16] 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)*, pages 1024–1031, 1999.

[17] S. A. Wilmarth, N. M. Amato, and P. F. Stiller. Motion planning for a rigid body using random networks on the medial axis of the free space. In *Proc. ACM Symp. on Computational Geometry (SoCG)*, pages 173–180, 1999.