Adapting the Sampling Distribution in PRM Planners Based on an Approximated Medial Axis

Yuandong Yang Oliver Brock
Laboratory for Perceptual Robotics
Department of Computer Science
University of Massachusetts Amherst
Email: {yuandong, oli}@cs.umass.edu

Abstract— **Probabilistic roadmap planners have proven to be effective in solving complex path planning problems. These planners sample the configuration space to compute a representation of its free space connectivity. One of the major difficulties for this approach is the planning of a path through narrow configuration space passages, since samples are placed inside narrow passages only with small probability. To address this problem, approaches have been devised that rely on the medial axis of the workspace to bias sampling in configuration space such that the probability of generating samples inside narrow passages is increased. This paper introduces a novel algorithm for computing an approximation to the medial axis, which can be computed more efficiently than the exact or discretized medial axis. We demonstrate that, compared to the true medial axis, this approximation is equally well suited to bias the sampling in probabilistic roadmap planners. Furthermore, we present a novel sampling strategy based on the approximated medial axis. This strategy results in high sampling density in narrow passages, while sampling open spaces sparsely. Experiments demonstrate the effectiveness of the medial axis approximation and its application to motion planning based on the proposed sampling scheme.**

I. Introduction

Path planning is a provably difficult problem [16]. Sampling-based approaches, and most prominently probabilistic roadmap planners [11], have proven to be very effective techniques for solving path planning problems. These approaches compute a roadmap that represents the connectivity of the configuration free space by collision-free edges. The roadmap is obtained by sampling the configuration space and subsequently connecting nearby samples lying in free space. For probabilistic roadmap (PRM) planners, sampling is performed uniformly at random. If the configuration space contains a narrow free space passage, the probability of placing a collision-free sample in it is very low. This is commonly referred to as the narrow passage problem.

To address the narrow passage problem, the density of samples can be increased. As a consequence, the sampling density for the entire configuration space will be dictated by its most narrow passage, resulting in an excessive number of samples in open areas of configuration space. To bias the sampling towards narrow passages it has been proposed to place samples in close spatial proximity to the medial axis of the workspace and thus far away from obstacle boundaries. Biasing samples towards the center of narrow passages increases the probability of generating a free configuration inside the passage, improving the performance of PRM methods [7], [9], [10], [14].

The computation of the medial axis, however, can itself be a computationally expensive operation. While various algorithms can compute the medial axis of simple polyhedra composed of a few hundreds of triangles, it is non-trivial to scale them to more complicated models. This is a consequence of the instability and the algebraic complexity of the medial axis. In order to address these problems, various algorithms to approximate the medial axis have been proposed (see Section II). Most of these methods focus on the accuracy of the approximation and consequently still require extensive computations. For the application in PRM planners the accuracy of the approximation is less important than the computational efficiency. Furthermore, in large open areas only a coarse approximation of the medial axis is needed, as it is easy to determine collision-free samples in those areas.

In this paper, we propose a novel algorithm to approximate the medial axis. The proposed algorithm has several desirable properties. First, the algorithm adapts the density of points used to represent the approximated medial axis based on local properties of the solid. This is desirable for probabilistic motion planning. Second, the algorithm is computationally very efficient. As shown in Section V, it can compute the approximated medial axis much more efficiently than previous methods. Third, the algorithm allows the user to trade off accuracy for speed of computation explicitly. This is particularly interesting for the application of medial axis approximation in probabilistic roadmap planning, as the accuracy of medial axis computation (and consequently also its cost) can be adjusted to the difficulty of the motion planning problem.

The approximated medial axis computed by the proposed method is represented by samples in close proximity to the surface of the true medial axis. As a consequence of the approximation algorithm, these samples are dense in narrow spaces and sparse in open spaces. Based on this observation, we propose a new sampling scheme for PRM methods. This scheme is heavily biased towards placing samples in narrow passages. The experimental results in Section V show that this novel sampling method achieves significantly better performance than previous methods. This performance gain can be observed for the computation of the medial axis as well as for the computation of a collision-free path, as much fewer samples have to be taken to construct a successful roadmap.

II. Related Work

The medial axis of a solid D [17] is the locus of points inside D, which lie at the centers of all closed discs or balls which are maximal in D and have at least two contact points with the solid. The medial axis has found application in robot motion planning because it captures points at a maximum distance from obstacles and can be used as a heuristic to find configurations of the robot that lie in free space.

Early approaches to motion planning relied on the Voronoi diagram, which is closely related to the medial axis. The generalized Voronoi graph (GVG), which has been used in sensor-based motion planning [4], represents a subset of the medial axis in three dimensions. In [4], the algorithm to compute the GVG uses local distance information obtained from the sensors, as the robot explores its environment. Starting from the initial position, the robot moves to the closest medial axis point, follows the medial axis and gets off the medial axis when it is close to the goal position.

To increase the computational efficiency of PRM planners, hierarchical PRM (HPRM) planners [5] sample the configuration space adaptively. Sampling is biased towards narrow passages by using the information contained in an initial sparse sampling to estimate the local amount of free space. To avoid examination of the entire configuration space required for HPRM planners, other approaches attempt to restrict sampling to a subset of the configuration space. The medial axis PRM introduced in [18] and subsequently extended in [13], approximates the distance to the closest configuration space obstacle to obtain sample points on the medial axis of the *configuration space*. The algorithms in [7], [9], [10], [14], on the other hand, use the medial axis of the *workspace*. The information captured by the medial axis can be used to locate narrow passages [7], [14] or to attain collision-free sample points in [9], [10]. In the remainder of the paper we will be concerned with PRM planners that sample based on the medial axis of the workspace, rather than the configuration space. We refer to these methods as workspace medial axis PRM planners (MA-PRM).

Methods of medial axis computation can broadly be divided into three categories. The algorithms most commonly used in the robotics literature represent the category of tracing approaches [4], [10]. These methods start from a point on the medial axis and perform a local exploration to determine a nearby medial axis point. This procedure is repeated recursively until all parts of the medial axis have been traced. Computational considerations limit the applicability of these approaches to polyhedra composed of only a few thousand faces.

Voxel-based methods [8], [15] represent free space as voxels. Ragnemalm [15] assigns to each voxel the Euclidean distance to the nearest voxel on the boundary of the free space and computes the local directional maxima to determine the approximated medial axis. Foskey [8] uses hardware to compute distances while adaptively and recursively dividing the voxels if they contain a portion of the medial axis. This method computes a simplified medial axis [8], which is a subset of the actual medial axis. The resulting data structure is more stable than the true medial axis but it does not necessarily maintain the connectivity of the medial axis, thereby limiting its applicability.

Voronoi-based methods [1], [6] divide the free space into Voronoi regions based on sample points on the surface of the solid. The resulting Voronoi regions are small because a dense sampling of the surface is required. These algorithms can be applied to complex models, if an appropriate set of sample points can be easily determined.

III. Sampling the Approximated Medial Axis

The main idea of our algorithm [19] is to generate a small set of partially overlapping maximal spheres to cover almost the entire free space within the environment. These spheres are constructed to intersect features of the medial axis. By sampling points on the surface of the spheres and determining their closest features in the environment, a set of points in proximity to the medial axis can be identified. These points serve as an approximation to the medial axis. The union of these points comprises the approximated medial axis (aMA). Because large open areas can be covered by large spheres, the aMA consists of few points in wide open areas and is sampled more densely in geometrically complex regions. The precision with which the aMA approximates the medial axis can be specified as a parameter of the algorithm. This allows users to consciously trade accuracy for computational efficiency.

A. Description of the Algorithm

Each point inside the solid has at least one closest feature on the surface of the solid. The *direction vector* $\vec{v}$ of a point p in the solid D is the unit vector pointing from point p to the closest feature on the surface of D. The *distance* $\delta(p)$ associated with a point p in the solid D is the distance from point p to the closest feature on the surface of D. Note that points on the medial axis must have at least two direction vectors.

The description of the algorithm relies on two primitive operations. The first identifies an initial point m and associated distance $\delta(m)$, such that the resulting sphere of radius $\delta(m)$ around m intersects the medial axis and does not intersect the solid. The second primitive, given a solid D, a set of points P in the interior of D, and their direction vectors, indentifies those points $p_i \in P$ which are closest to the medial axis of D and based on these computes the points on the approximated medial axis. These primitives will be described after we have detailed the algorithm of computing the aMA.

Assume point m lies on the medial axis and is distance $\delta(m)$ away from the closest obstacle. This point is determined using aforementioned primitive. A priority queue Q is initialized to contain the sphere described by point m and radius $\delta(m)$. The set B of spheres describing the free space inside the solid D is initialized to be the empty set.

The largest sphere s is extracted from Q and a set U of uniformly distributed samples on its surface is generated.

Points in U that are contained in one of the spheres in B are discarded. The second aforementioned primitive is used to determine those points in U that lie closest to the medial axis. These points p are added into the aMA and, along with their distance $\delta(p)$, into the priority queue Q. The sphere s is added to B. These steps are repeated until the maximum size of the spheres in Q is smaller than a threshold, which we call the *expansion threshold* K_e. We can control computation time and the number of aMA points by changing K_e (compare with Figure 1).

```
1) Find sphere $s$ with center $c$ inside $D$ such that $\delta(c) > K_e$
   and the medial axis intersects $s$ $m$
2) $B := \emptyset$ ; $M := \emptyset$ ; $Q := \{(c, \delta(c))\}$
3) Extract sphere $s = (p, \delta(p))$ from $Q$
4) While $\delta(p) > K_e$
    a) Generate $n$ uniformly distributed samples
       $U = \{u_1, \cdots, u_n\}$ on the surface of $s$
    b) Discard $u_i \in U$, if $\exists b_j \in B$ such that $u_i \in b_j$
    c) Using $U$, determine approximated medial axis points
       $A = \{a_1, \cdots, a_k\}$
    d) $Q := Q \cup \{(a_1, \delta(a_1)), \cdots, (a_k, \delta(a_k))\}$
    e) $M := M \cup A$
    f) $B := B \cup \{(p, \delta(p))\}$
    g) Extract sphere $s = (p, \delta(p))$ from $Q$
5) Connect points in $M$ to generate the aMA
```

Fig. 1. The pseudo code of the algorithm. B is the set of spheres describing the interior of the solid D. M is the set of points describing the approximated medial axis. Q is the priority queue of spheres, ordered by radius.

Figure 2 illustrates our method in the two-dimensional case. Assume o_1 is the first element in Q. A maximal circle centered at o_1 is generated and n samples $p_1, p_2, ..., p_n$ are generated on its circumference (not all samples are shown in the figure). The point pairs (p_1, p_2), (p_3, p_4) and (p_5, p_6) have different direction vectors and the midpoints of these pairs of points, q_1, q_2 and q_3, are considered to be on the medial axis; they are added to the aMA and to the queue. Since q_3 has the largest radius it is expanded next and the procedure repeats.

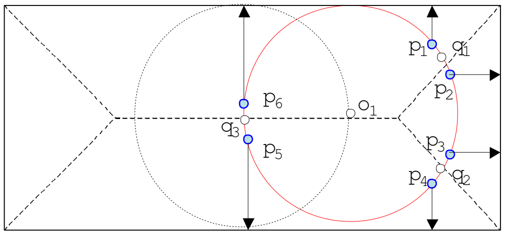

Fig. 2. An illustration of the algorithm. The dashed lines represent the medial axis of the rectangle.

We now discuss the two primitives used in the description of the algorithm.

B. Identifying the Initial Approximated Medial Axis Point

In the description of the algorithm it was assumed that the queue Q is initialized with a point m on the medial axis of the solid D. We use a similar expansion algorithm as the one described above to find m. Starting from a random point p inside D, we generate the maximal sphere s with the center at p. If we cannot find a medial axis point of the surface of the sphere (how medial axis points are identified is described in Section III-C), we sample the surface of s and determine the sample p' with the biggest distance $\delta(p')$ associated with it. The point p' serves as the center of the next sphere. This process is repeated until the sphere intersects the medial axis. Since this procedure converges towards a sphere of locally maximum radius, its center converges towards a point on the medial axis and the surrounding sphere thus must intersect the medial axis.

C. Identifying Approximated Medial Axis Points

Given a set of uniformly distributed sample points U on the surface of a sphere, we apply the *separation angle criteria* [1], [2], [3], [8] to determine the set containing points of the aMA. If the direction vectors of two adjacent sample points span an angle larger than a threshold θ_t, we take the samples' midpoint as the aMA point. In Section III-D we will bound the error of this approximation.

If a sphere only intersects one facet of the aMA, there will be two sets of sample points, each set with direction vectors pointing towards different parts of the solid. These sets are distinguished using the angle criteria. In this case we simply insert the center of the sphere into the aMA. The samples on the surface are superfluous. If a sphere intersects more than two facets of the medial axis, however, these facets may meet inside the sphere. We identify sets of adjacent samples with distinct direction vectors, based on the angle criteria. We then find intermediate points between those sets and add them to the aMA. These points are called *critical points*; they designate an edge or a vertex between multiple facets of the aMA. Critical points can be used to approximate the hierarchical generalized Voronoi graph[4].

D. Approximation Error

In this section we bound the error made by the proposed method of approximating the medial axis of a solid. We differentiate quantitative errors that result from the finite sampling density of our algorithm, and qualitative errors. The latter are a consequence of the sphere expansion algorithm to explore the free space inside the solid.

1) Quantitative Errors: Let M be the set of approximated medial axis points for a given solid D attained by our algorithm. For each point p_i in M, there exists a medial axis point t_i which is closest to p_i. The absolute and relative errors for the sample points M of the approximated medial axis, relative to the true medial axis T are

- Absolute error ϵ_a: $\max_{p_i \in M}\{|p_i - t_i|\}$
- Relative error ϵ_r: $\frac{\epsilon_a}{\delta(t_i)}$

The absolute error can be bounded as follows. Given a set of uniform samples on a sphere, we define two points as neighbors of each other if the distance between them is smaller than the *neighbor threshold* d_n. If the closest features of two neighboring points p_1 and p_2 are different, there is a point between them which is closest to both of those features and thus lies on the medial axis. Since we chose the midpoint between p_1 and p_2 to represent the approximated medial axis, the maximum distance between points of the approximated medial axis and the real medial axis cannot exceed $\epsilon_a = \frac{d_n}{2}$. This equation holds in both 2D for circles and 3D for spheres. We define the $\frac{d_n}{2}$ of the smallest sphere generated by the algorithm described in Section III as the *resolution* of the algorithm.

There is an obvious relation between the number of samples placed on a sphere and the quality of the approximated medial axis. Given a uniform sampling of the sphere's surface, let d_n be the maximum distance between any sample and all samples in a set of its closest neighbors on the sphere. For a sphere with radius r then, the absolute error is given by $\epsilon_a = r\ \frac{d_n}{2}$. If the midpoint between more than two samples with different direction vectors is selected to be part of the aMA, we use the triangle inequality to argue that $\epsilon_a < r\ d_n$. These arguments hold in 2D and 3D.

We can now specify the number of samples on a sphere required to achieve a desired absolute or relative error. In a 2D environment $N = \frac{\pi}{\arcsin \frac{\epsilon_a}{r}}$ samples are required to achieve an absolute error ϵ_a, where r is the radius of the sphere. Suppose q_1 is a sample point on the circle. For a given relative error ϵ_r, the number of samples needed is given by $N = \frac{\pi}{\arcsin \epsilon_r}$ when $r \leq \delta(q_1)$ and by $\frac{\pi}{\arcsin \frac{\epsilon_r r}{\delta(q_1)}}$ when $r > \delta(q_1)$. The argument in a 3D environment is similar.

2) Qualitative Errors: The aMA points we obtain only represent a subset of the simplified medial axis [8]. The proposed algorithm misses part of the simplified medial axis because it does not expand spheres into the entire free space and because it only considers a small set of points on the surfaces of spheres. Given an absolute error ϵ_a, the algorithm will stop if the maximum $\delta(p)$ of all sample points p is smaller than ϵ_a. For a relative error ϵ_r, the algorithm will terminate if the radius of all spheres is smaller than the threshold K_e. Obviously, the algorithm can not reach a space with a 'gate' smaller than ϵ_a or the threshold K_e. The spheres stop spreading when they meet that 'gate' and the free space behind the gate will be missed, including the associated features of the medial axis.

In the case of relative error, the sample points on large spheres are sparser. Consequently, the size of the largest gate which is missed depends on the amount of local free space. This is illustrated in Figure 3: p_1 and p_2 are adjacent sample points on the bigger circle and θ_b is the angle between the direction vectors of p_1 and p_2; q_1 and q_2 are the adjacent sample points on the smaller circle and θ_s is the angle between the direction vectors of q_1 and q_2. It is obvious that $|p_1p_2| > |q_1q_2|$ and $\theta_b > \theta_s$. According to the separation angle criteria, the algorithm can find the aMA point on the smaller circle but not on the bigger circle, if $\theta_b > \theta_t > \theta_s$.

E. Discussion

In Section II we introduced two tracing approaches which compute the medial axis using local information [4], [10]. The algorithm proposed here can also be regarded as a tracing approach. It traces the medial axis by sampling on the spheres during expansion. However, the algorithm differs importantly from previous approaches: it explores all possible tracing directions to pursue the near-optimal step size, which reflects the local geometrical properties of the free space. This is the main reason for the computational efficiency of the proposed approach.

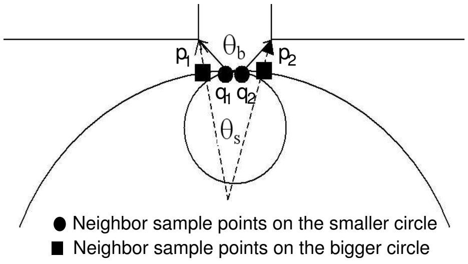
p1
θb
p2
q1
q2
θs
● Neighbor sample points on the smaller circle
■ Neighbor sample points on the bigger circle

Fig. 3. Illustration of the influence of relative error on missed features.

The most costly operation performed is the distance computation, which generally is assumed to be $O(\log n)$ in practice, where n is the number of faces of the model. The number of distance computations, as well as the number of aMA points, is bounded by the number of samples on each sphere times the number of spheres generated in our algorithm. If we regard distance computation as a constant time operation, the proposed algorithm is output-sensitive, i.e., its computational cost only depends on the size of the output. This is a very desirable property in a sense that the computational cost is related to the geometric complexity of the solid itself and not its model. For example, it takes the same amount of time to compute the medial axis of a box, no matter it is composed of 12 triangles or 12 million triangles. The algorithm has successfully been applied to a number of benchmark models, consisting of up to one million triangles [19].

IV. Adaptive Sampling in Configuration Space

We obtain a set of aMA points using the algorithm described in Section III. This set reflects the local geometrical complexity of the environment in its spatial density. We now describe a probabilistic roadmap algorithm based on the approximated medial axis (aMA-PRM). The main idea of aMA-PRM is to sample configuration space in the proximity of the points that represent the approximated medial axis. The resulting sampling density is thus given by the density of the aMA points. Consequently sampling will be biased towards narrow passages.

The sampling technique based on the approximated medial axis is shown in Figure 4. It generates k random configurations in the proximity of each aMA point. Here, proximity is defined in $\mathbb{R}^3$. These configurations are subsequently modified iteratively to improve general proximity to the medial axis, using the method described in [10]. Several so-called handle points H on the robot are chosen in order to capture the main spatial features of the robot. The closest aMA points to these handle points are identified and a potential field-based method is applied to minimize the distance between the handle points and their corresponding aMA points. This increases the probability that the configuration lies in free space.

```
For each aMA point $m_i$ in $M$ repeat $k$ times
  1) generate random configuration $q_i$ in proximity to $m_i$
  2) for each $h_i \in H$ identify the closest $m_j \in M$
  3) obtain $q_i'$ by iteratively reducing the distances between all $h_i$
     and the corresponding $m_j \in M$
  4) if it lies in free space, add $q_i'$ to the roadmap $R$
```

Fig. 4. The pseudo code for the sampling algorithm used in the aMA-PRM planner. M represents the set of aMA points; H is the set of handle point on the robot; R denotes the probabilistic roadmap built by the planner.

The proposed sampling method differs from previous approaches [10] in two important aspects. By using the approximated medial axis presented in this paper, the sampling density is varied based on the geometric properties of the workspace, with high densities in narrow passages. This is illustrated in Figure 5 for the environment shown in Figure 6. The figure shows the medial axis points generated by a regular tracing approach (left) and by the proposed sphere expansion (right). Furthermore, the sampling methods attempts to sample k times for each point on the aMA, providing a convenient way of varying the adaptive sampling density in a global fashion. By placing fewer samples, the size of the roadmap can be reduced significantly and computational efficiency is increased.

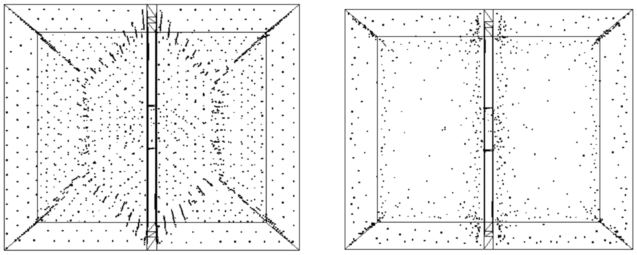

Fig. 5. The medial axis points obtained by the tracing method [10] (left) and the proposed sphere expansion method (right).

V. Experimental Results

The experiments reported in Table I were performed on a dual-PentiumIII PC with 1024 MB SRAM and a 32MB DDR NVIDIA GeForce2 GTS graphics card. The proposed planner was developed based on the MSL [12]. Our experiments are restricted to rigid body robots and were averaged over ten independent runs. The minimum number of samples for the various planners as well as the parameter k for the aMA-PRM approach which were required to solve a given problem were determined by incrementally building and testing the roadmaps. If a planning problem could not be solved, the values given in Table I describe the expended resources at the point at which the experiment was aborted.

The simulation environment (see Figure 6) is a box which is divided into two parts by a wall with a square opening, representing the only path from one side of the box to the other. Figure 7 shows snapshots of an L-shaped robot passing through the hole. We compare the results for planning a path for two L-shaped robots of different sizes from one side of the box to the other using four PRM variants: the basic PRM [11], the PRM using sampling based on the medial axis (MA-PRM) computed by a regular tracing approach [10], a variant of the former relying on the aMA instead (MA-PRM with sphere expansion), and finally the proposed aMA-PRM using sphere expansion to compute the aMA and the sampling method described in Section IV.

The experimental results are summarized in Table I. We compare MA-PRM with the tracing and sphere expansion for medial axis computation to show that no penalty is incurred for using an approximation to the medial axis. Surprisingly, we observe that for the same MA-PRM method a speed-up results from using the aMA rather than the MA. This can be explained by the fact that due to the sparser sampling density in open spaces, more randomly generated configurations will be iteratively modified to lie closer to medial axis points in the narrow passage, thus resulting in a higher sampling density in narrow passages. Due to the long computation time for the medial axis, however, the MA-PRM planner with tracing is outperformed by basic PRM for the easier scenario with the small robot. Note that the clearance between the large robot and the opening in the wall is too small for any part of the robot to be able to rotate freely while traversing the opening. For this very difficult scenario only the aMA-PRM method was able to compute a path within reasonable amounts of computation.

From the experimental results it can be seen that the aMA-PRM planner proposed in this paper outperforms both the regular PRM and the MA-PRM [10] by orders of magnitude for many of the shown metrics and results in a substantial decrease in overall computation time. In particular, as a result of the biased sampling, the size of a successful roadmap is reduced substantially.

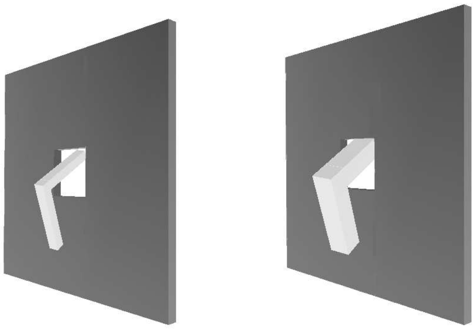

Fig. 6. The environment is a box ($120 \times 100 \times 100$, not shown in the figure), divided into two equal parts by a wall with a hole ($4 \times 20 \times 20$). The robots are composed of two blocks (small robot: $30 \times 4 \times 4$, large robot: $34 \times 8 \times 8$).

Robot	Medial Axis Computation			Path Planning						Total
	Method	Vertices	$t_m(s)$	PRM Method	Vertices	Edges	Checks	$t_p(s)$	Solved	$t(s)$
Small	N/A	N/A	N/A	Basic	2,375	72,008	4,469,599	861	Yes	861
	Tracing	1,722	808	MA	1,583	43,356	2,954,210	476	Yes	1284
	Sphere Expansion	832	33	MA	1,328	31,612	2,160,183	351	Yes	384
				aMA (k=7)	289	1,203	91,793	65	Yes	98
Large	N/A	N/A	N/A	Basic	12,931	446,068	300,804,571	13,146	No	N/A
	Tracing	1,722	808	MA	14,867	519,246	35,539,629	15,076	No	N/A
	Sphere Expansion	832	33	MA	9,663	323,882	22,251,921	10,145	No	N/A
				aMA (k=36)	640	8,112	589,995	340	Yes	373

TABLE I. Comparison of different PRM methods generating the motion of an L-shaped robots through a narrow passage. The first set of columns reports results on the computation of the medial axis, the second on path planning, and the last column reports total planning time. For the medial axis, the method of computation, the number of vertices in the resulting medial axis representation, and the time t_m to compute the medial axis are reported. For path planning, the specific PRM method, the number of vertices (milestones) and edges of the resulting roadmap, the total number of collision checks, the time for path planning, and whether or not the problem was solved are given. The total time $t = t_m + t_p$ specifies the overall execution time. All times are averaged over ten independent runs and are given in seconds.

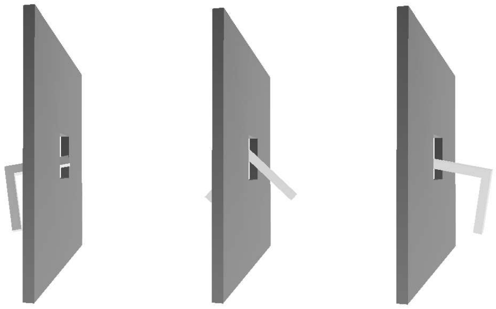

Fig. 7. Snapshots of an L-shaped robot passing through the hole.

VI. Conclusion

Motivated by the goal of biasing sampling in the PRM framework towards narrow passages, a novel algorithm for computing an approximation of the medial axis was presented. This algorithm distinguishes itself from previous work by its computational efficiency. Experiments show that simply using the approximated medial axis instead of a evenly discretized medial axis to bias sampling based on a previous PRM sampling technique accelerates planning. This is caused by the fact that the proposed medial axis approximation is represented by sparse samples in open areas and by dense samples in narrow passages. This property is exploited by a novel sampling strategy based on the approximated medial axis presented in this paper. Applying this approach to sampling yields an additional, very significant increase in computational efficiency, as demonstrated by experimental results.

References

[1] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry*, 19(2-3):127–153, 2001.

[2] D. Attali and A. Montanvert. Computing and simplifying 2D and 3D continuous skeletons. *Computer Vision and Image Understanding*, 67(3):261–273, 1997.

[3] J. D. Boissonnat and F. Cazals. Smooth surface reconstruction via natural neighbor interpolation of distance function. In *16th Proc. Annu. ACM Symposium on Computer Geometry*, pages 185–203, 2000.

[4] Howie Choset. Incremental construction of the generalized Voronoi diagram, the generalized Voronoi graph, and the hierarchical generalized Voronoi graph. In *1st CGC Workshop on Computation Geometry*, 1997.

[5] Anne D. Collins, Pankaj K. Agarwal, and John L. Harer. HPRM: A hierarchical PRM. In *Proceedings of teh IEEE International Conference on Robotics and Automation*, pages 4433–4438, 2003.

[6] T. K. Dey and W. Zhao. Approximate medial axis as a Voronoi subcomplex. In *Proc. 7th ACM Sympos. Solid Modeling Applications*, pages 356–366, 2002.

[7] M. Foskey, M. Garber, M. C. Lin, and D. Manocha. A Voronoi-based hybrid planner. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 55–60, 2001.

[8] Mark Foskey, Ming Lin, and Dinesh Manocha. Efficient computation of a simplified medial axis. In *Proc. ACM Symposium on Solid Modeling and Applications*, 2003.

[9] Maxim Garber and Ming C. Lin. Constraint-based motion planning using Voronoi diagrams. In *Proc. Fifth International Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2002.

[10] C. Holleman and L. E. Kavraki. A framework for using the workspace medial axis in PRM planners. In *Proceedings of the International Conference on Robotics and Automation*, pages 1408–1413, 2000.

[11] Lydia E. Kavraki, Peter Švestka, Jean-Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[12] Steven M. LaValle. http://msl.cs.uiuc.edu/msl.

[13] Jyh-Ming Lien, Shawna L. Thomas, and Nancy M. Amato. A general framework for sampling on the medial axis of the free space. In *Proceedings of the International Conference on Robotics and Automation*, pages 4439–4444, 2003.

[14] C. Pisula, K. Ho, M. Lin, and D. Manocha. Randomized path planning for a rigid body based on hardware accelerated Voronoi sampling. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, 2000.

[15] I. Ragnemalm. Pattern recognition letters. *Pattern Recognition Letters*, 14(11):883–888, Nov. 1993.

[16] J. H. Reif. Complexity of the mover's problem and generations. In *Proceedings of the Symposium on Foundations of Computer Science*, pages 421–427, 1979.

[17] Evan C. Sherbrooke, Nicholas M. Patrikalakis, and Erik Brisson. Computation of the medial axis transform of 3-D polyhedral. In *Symposium on Solid Modeling and Applications*, pages 187–200, 1995.

[18] S. Wilmarth, N. Amato, and P. Stiller. Maprm: A probabilistic roadmap planner with sampling on the medial axis of the free space. Technical Report TR98-022, Department of Computer Science, Texas A&M University, 1998.

[19] Yuandong Yang, Oliver Brock, and Robert N. Moll. Efficient and robust computation of an approximated medial axis. In *Proceedings of the ACM Symposium on Solid Modeling and Applications*, Genoa, Italy, 2004.