

A Framework for Using the Workspace Medial Axis in PRM Planners

Christopher Holleman Lydia E. Kavraki
 Department of Computer Science
 Rice University
 6100 Main Street, Houston, TX 77005
 {rusty,kavraki}@cs.rice.edu

Abstract

Probabilistic roadmap planners have been very successful in path planning for a wide variety of problems, especially applications involving robots with many degrees of freedom. These planners randomly sample the configuration space, building up a roadmap that connects the samples. A major problem is finding valid configurations in tight areas, and many methods have been proposed to more effectively sample these regions. By constructing a skeleton-like subset of the free regions of the workspace, these heuristics can be strengthened. The skeleton provides a concise description of the workspace topology and an efficient means of finding points with maximal clearance from the obstacles. We examine the medial axis as a skeleton, including a method to compute an approximation to it. The medial axis is a two-equidistant surface in the workspace. We form a heuristic for finding difficult configurations using the medial axis, and demonstrate its effectiveness in a planner for rigid objects in a three dimensional workspace.

1 Introduction

Robot motion planning is an important tool in a wide variety of settings ranging from computational pharmacology to computer aided industrial design [7, 9, 11, 19]. We present a framework which can be applied to a range of variations on the general path planning problem stated in [21]. The problem is to find a path for a robot from an initial configuration to a final configuration without colliding with the environment. In the most common case, the robot is a single rigid object for which motions are defined by rigid-body transformations. Since our method processes workspace geometry, it can be applied to problems where configuration space methods are no longer feasible due to the increased dimension of the configuration space (such as flexible and articulated robots).

The probabilistic roadmap planner (PRM) [16, 17, 18, 22, 23] is a common method for planning in potentially high dimensional configuration spaces. PRM planners begin by generating a large number of configurations randomly throughout the free part of the configuration space (the *freespace*) and then making local connections in an attempt to create a con-

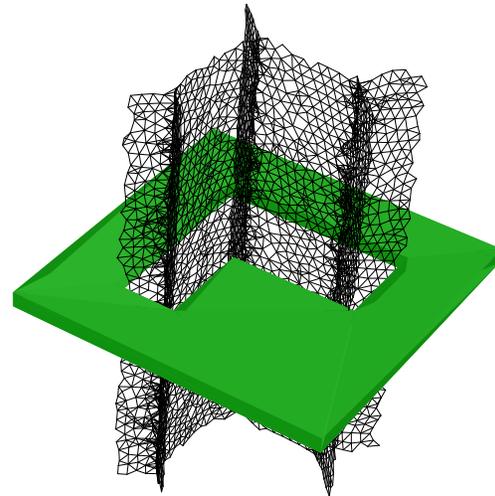


Figure 1: The medial axis approximation of a sample workspace

nected graph which spans the freespace. Unfortunately, PRM planners tend to fail when crucial configurations lie in very narrow regions of the configuration space, an issue sometimes referred to as the narrow passage problem [14]. The probability of randomly guessing such a configuration is prohibitively small, and the planner may populate the rest of the freespace with many superfluous configurations before finding the necessary difficult configurations. The goal of our planner is to address the narrow passage problem for a certain class of robots with a sampling heuristic based on the workspace geometry. Additionally, we aim to find configurations with maximal clearance from the surrounding obstacles. This increases the success rate of the local planner and ultimately generates a more fault-tolerant path.

We choose to deal with workspace geometry because in general it is not feasible to work with the geometry of the configuration space. (Algorithms have been proposed which approximate configurations on the medial axis of the freespace [24], but these approaches do not scale beyond single rigid body robots and even in this case rotations are difficult to han-

dle.) The workspace geometry in and of itself is too complicated to use directly. We condense the description of the obstacles into a simpler geometric structure which can be utilized more directly in finding good configurations. We want to find configurations of maximal clearance, which suggests using a geometric structure which encodes points of maximal clearance. We choose to use the medial axis (also referred to as the *generalized voronoi diagram*) of the workspace as a guide for choosing configurations. The medial axis is the set of points in the free regions of the workspace for which at least two points on the boundary of the obstacles are equally close, and all other points on the boundary of the obstacles are farther away. We describe methods for computing a polygonal approximation to the medial axis given the workspace geometry and heuristics for using this approximation to generate favorable configurations.

2 Related Work

Planning for robots with many dof has been extensively treated in recent literature ([1, 2, 11, 12, 20, 21]). The probabilistic roadmap approach to planning (PRM) [16, 17, 18, 22, 23] has gained wide acceptance because the method is easy to implement and use and provides good performance results. We provide references both to other approaches for dealing with the narrow passage problem and to geometric work related to our construction of the medial axis.

An important issue in PRM planners is the method for choosing the random configurations for the construction of the roadmaps. Recent works have considered many alternatives to a uniform random distribution of configurations as means for dealing with the narrow passage problem. A re-sampling step, creating additional nodes in the vicinity of nodes that are connected with few others, is shown in [17, 18]. Nodes close to the surface of obstacles are added in [2], and in [6] a probabilistic method for choosing configurations close to the obstacles is presented. A dilation of the configuration space has been suggested in [14], as well as an in depth analysis of the narrow passage problem. In [24] a procedure for retracting configurations onto the freespace medial axis is presented. In this paper, we treat rigid objects and we attempt to generate configurations close to the medial axis of the workspace. A similar method, from which this paper has evolved, is presented in [10].

The generalized voronoi graph, which is related to the medial axis, is addressed in detail in [8]. While the medial axis is the set of 2-equidistant points, the generalized voronoi graph is the set of n -equidistant points in \mathbb{R}^n . The GVG is useful in constructing a roadmap of a space. We draw heavily from [8], extending their methods from the generalized voronoi graph to the medial axis (*aka* the generalized voronoi diagram). [4] presents a method for computing an optimal reconstruction of a surface from a set of points. A modification to this method, briefly mentioned in [4] and implemented for two dimensions

ACCESSMA(PPOINT p_0 , TOLERANCE ϵ)

1. Find $c(p_0)$
2. Define $m(t) = p_0 + t * (p_0 - c(p_0))$. Find the minimal value of α such that, given $u = m(\alpha - \epsilon)$ and $v = m(\alpha + \epsilon)$, $\|c(u) - c(v)\| \gg \|u - v\|$.
3. Define $m'(t) = p_0 + t * (c(v) - c(u))$. Find the minimal value of α' such that, given $u' = m'(\alpha' - \epsilon)$ and $v' = m'(\alpha' + \epsilon)$, $\|c(u') - c(v')\| \gg \|u' - v'\|$.
4. Return $m'(\alpha')$

Figure 2: Algorithm for obtaining a point on the medial axis

in [10], provides for the computation of the medial axis of a surface implied by a set of points. This method is best suited for well sampled curved surfaces, but has been difficult to implement for typical CAD models with sharp edges. Amenta ([3]) recently has shown promising results in this direction.

3 Background and Notation

Let $B = \{B_0, \dots, B_n\}$ be a set of obstacles in the workspace W . Let ∂B_i be the boundary of the obstacle B_i , and $\partial B = \bigcup_i \partial B_i$. $d(p)$ is the distance from a point p to the closest point on the boundary of the obstacles, ∂B . $c(p)$ denotes the point in ∂B closest to p , *i.e.* $d(p) = \|p - c(p)\|$.

The medial axis is then defined as $MA(W) = \{p : \exists a, b \in \partial B, \|p - a\| = \|p - b\| = d(p)\}$. In this definition the plane of the medial axis at p is characterized by the normal vector $a - b$.

4 Main Algorithm Description

The description of the algorithm is divided into three parts: computation of a polygonal approximation to the medial axis, generation of configurations and planning.

4.1 Computation of the Medial Axis

The basic step of the algorithm is a method for moving a point in the workspace to a nearby point on the medial axis. This method is a simplified version of the method for finding points on the GVG in [8] — for the GVG one must find an n -equidistant point for an n -dimensional workspace while for the medial axis we must only find a 2-equidistant point. It is sufficient merely to approximate the medial axis since we employ it only in rough heuristics.

Obtaining a point on the medial axis The algorithm described in Figure 2 takes a point p_0 in the workspace and attempts to find a nearby point which is on the medial axis. After finding the closest point to p_0 on the boundary of the obstacle (step 1), we move directly away from the obstacle;

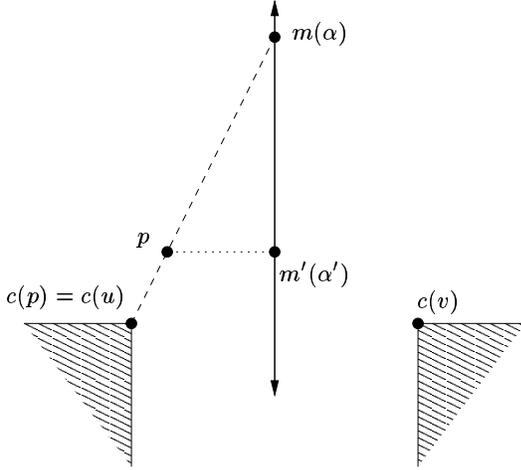


Figure 3: Illustration of the access method

this path is denoted by the parametric function $m(t)$. α denotes the value of t for which $c(m(t))$ is discontinuous. At this discontinuity $m(\alpha)$ is equidistant from two points on the obstacles, and is thus on the medial axis. $m(\alpha)$ is on the medial axis but is not the closest point to p_0 on the medial axis. $m(\alpha)$ may be quite far from the closest point to p_0 on the medial axis. Figure 3 shows a simple scenario in which $m(\alpha)$ drifts up along the medial axis. $m(t)$ is the dashed line, and the medial axis is the solid vertical line. This stems from the fact that we limit our search for medial axis points to the line $m(t)$ which is not perpendicular to the medial axis. We define a new search direction along a line $m'(t)$ normal to the medial axis (as approximated by the normal of the medial axis at $m(\alpha)$). In Figure 3 $m'(t)$ is shown as a dotted line. We then search for a value of α' such that $c(m'(t))$ is discontinuous at $t = \alpha'$. The resulting point, $m'(\alpha')$, is on the medial axis and approximately the closest point to p_0 on the medial axis. The accuracy and even success of this method depend on the medial axis being approximately planar locally. In practice this is not a problem since, with the exception of branches, which are handled specially, the curvature of the medial axis is bounded by the inverse of the clearance. The algorithm may fail to find a point on the medial axis in a region of minimal clearance, but such points are of little use anyway.

Exploring the medial axis Our goal is to compute a polygonal approximation to the medial axis, which we achieve by a simple exploration technique that employs the method described in Figure 2. This method, presented in Figure 4, is similar to the method employed by [8] to trace the GVG. Since the GVG is a set of curves and the medial axis (in \mathbb{R}^3) is a set of surfaces, we extend the method of [8] to explore in a plane rather than along a line. We first obtain a point on the medial axis by attempting the ACCESSMA method on points randomly chosen from the workspace, repeating until it succeeds. We keep a queue of points which corresponds

1. Repeatedly sample a point p randomly from the workspace until $x = \text{ACCESSMA}(p)$ succeeds.
2. Initialize a queue Q with x , and add x to the graph M .
3. Repeat until $\text{empty}[Q]$:
 - (a) $x = \text{dequeue}(Q)$.
 - (b) Generate a small set of points $P = p_0, \dots, p_n$ evenly about x , in the plane of the medial axis at x , but not close to existing points in Q or M .
 - (c) Add an edge in M between x and any point y in Q or M that is close to x .
If the normal of the medial axis at x is significantly different than at y , add additional points to P in the plane perpendicular to the medial axis at both x and y and in the opposite direction of the common closest point of x and y .
 - (d) For each $p_i \in P$ if $x_i = \text{ACCESSMA}(p_i)$ succeeds, $\text{queue}(Q, x_i)$.
4. For all cycles in M of length 3, add a triangle to the medial axis surface approximation.

Figure 4: Algorithm for exploring the medial axis

to the border of the already explored region of the medial axis. To extend the explored region near a point x we generate points in the plane of the medial axis (step 2), taking care not to generate points in previously explored areas. In step 3 we add edges to connect the border point x to newly discovered nearby points on the medial axis. When adding edges, it is necessary to compare the normal of the medial axis at both endpoints. If these normals differ substantially, the edge crosses a “branch” in the medial axis. Figure 5 shows an edge that crosses a branch in the medial axis. Branches in the medial axis come in the form of three planes intersecting in a line. In this case we generate points in the vicinity of the third branch and add them to the set of points generated in step 2. Without these points the algorithm will only explore in one direction when crossing branches. Each of the points generated in step 2 and 3 are then moved onto the medial axis and added as new border points (discarding points for which ACCESSMA fails).

In addition to the geometry of the medial axis, we also have a fast evaluation of the clearance at any point on the medial axis.

4.2 Generating Configurations with the Medial Axis

General Framework The idea behind the framework for generating configurations is to condense the robot’s geometry to a small set of “handle” points and try to position these handle points close to the medial axis. We first choose a configuration randomly from the entire configuration space, and then

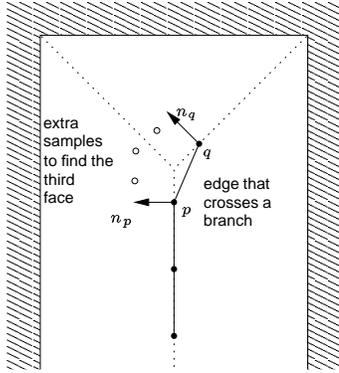


Figure 5: Illustration of exploration method at a branch in the medial axis

Given handles h_0, \dots, h_n and anchors a_0, \dots, a_i :

$$\begin{aligned}
 &CM = \frac{1}{n} \sum_i h_i \\
 &\text{do} \\
 &\quad \tau_0 = \sum_i (a_i - h_i) \times (CM - h_i) \\
 &\quad \text{rotate around } CM \text{ on the axis defined by } \tau_0 \text{ to minimize} \\
 &\quad \quad \tau \cdot \tau_0 \\
 &\text{while } \tau > \epsilon
 \end{aligned}$$

Figure 6: Method for minimizing anchor-handle distance for rigid objects

adjust the configuration according to a minimization procedure. The minimization attempts to bring handle points (and in more complex objects, only a subset of the handle points) close to the medial axis. The choice of handle points is discussed below, but first we present the minimization method for rigid objects.

Minimization for Rigid Robots For a rigid robot, we can model the minimization of distances between handles and their respective anchors as the solution to a physical model. In our model a spring in extension exists between each handle and anchor pair. The forces exerted by this collection of springs can be decomposed into linear forces and rotational forces. The linear force will be zero when the center of mass of handles is coincident with the center of mass of anchors. We find a zero-torque configuration with a gradient descent method.

Choosing Handle Points The choice of handles depends largely upon the type of robot. Their exact number and location are not critical, but here we provide some intuitive heuristics for choosing handle points. For three dimensional robots with significant bulk the handles correspond to the centers of spheres of maximal radius, completely contained by the boundary of the robot. There are two criteria which provide an intuitive guide for choosing the handle points. Handle points should be chosen such that

- the radius of the sphere around each handle is maxi-

mized, and

- the distances between handles are maximized.

By finding configurations for which these spheres are close to the medial axis, and such that each sphere’s radius is no greater than the medial axis clearance at the nearest point on the medial axis (referred to as the handle’s *anchor*), we generate useful and difficult configurations. If a handle point is on the medial axis, and the radius of the robot around it is less than the clearance of the medial axis at that point, no part of the robot inside the sphere will be in collision with the workspace. We aim to cover as much volume of the robot as possible with the least number of handles, so maximizing the radii around each handle is important. The distance between handles should be maximized in order to avoid overlapping spheres and so that each handle captures as much geometry as possible. Additionally, the probability of the extremities of the robot being in collision is higher than the probability of interior parts of the robot being in collision. As such, we place handles closer to the extremities of the robot. More discussion follows in Section 6 in choosing handles, but here we present specific examples. It should be noted that the choice of handles is strengthened by redundancy, and we can get useful results without a precise method of choosing these points.

4.3 Planning

Once we have generated free configurations, we attempt to plan local paths between nearby configurations. In regions where configurations from different connected components are near each other, we attempt to join components with a random walk. This enhancement step is particularly useful for connecting configurations which are in narrow passages, but relies on the sampling step to create configurations that are in or near narrow passages. The random walk also provides additional random configurations which supplement those found by minimization. A more complete description of the architecture of the planner is given in [13]. We currently have only developed medial-axis based heuristics for the sampling step. See section 6 for comments on using the medial axis during the planning step.

5 Experimental Results

Our implementation is in C++, and all test cases were run on a 175MHz SGI O2. We rely on the RAPID library for collision detection as well as computation of $c(p)$ and $d(p)$.

Example 1 This example, shown in Figure 7, comes from a medical application involving a robotic hip operation [15]. The problem is to maneuver a boot-like robot into an oval shaped cavity. The problem is complicated not only by the small clearances involved but also by the twist in the cavity, which is captured by the medial axis. In Figure 8 the medial

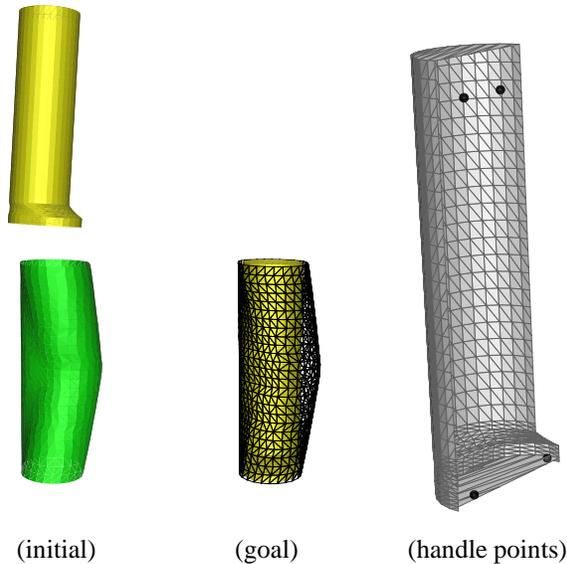


Figure 7: Example 1

axis is pictured with half of the obstacle cut away. The ribbon-like medial axis has been trimmed such that only points with sufficient clearance are shown. The handles for the robot are shown in Figure 7. We spend 4 minutes computing the medial axis and approximately 5 minutes to plan. By comparison, the same planner fails to complete the problem (after 4 hours) when configurations are chosen at random.

Example 2 For this example, shown in Figure 9, the medial axis allows us to align necessary subparts of the robot in the plane such that it can slip through the slot. Computing the medial axis in this case takes about 50 seconds, and the planning takes about 8 minutes. With random generation of configurations, the planner takes approximately 32 minutes. In 10 the handle points are shown with their radii doubled (in order to make them visible).

6 Discussion

We presented a basic method for computing the medial axis explicitly as well as a general framework for utilizing this information in the context of a probabilistic roadmap planner. We modified and extended the methods presented in [8] to generate a polygonal approximation to the medial axis of the workspace. Given this surface, we presented a method for generating useful and difficult configurations.

The selection of handle points in our current implementation requires manual interaction from the user before the planner may begin. The meet points of the generalized voronoi graph [8] of the robot are good candidates for handle points. The meet points of the GVG are points which have four (or more) closest points on the surface of the robot. This property makes for good handle points because the radius around such

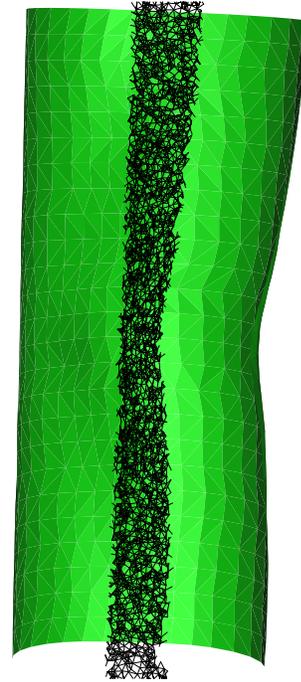


Figure 8: Medial axis for the rigid test case showing the twist of the obstacle (the pinch in the middle is actually a twist)

a point is locally maximized, and roughly speaking each convex feature of the robot (*i.e.* an extremity of the robot) creates one more meet point.

We only use a small portion of the information provided by the medial axis. A semi-local planner could use connectivity information in the medial axis. During enhancement, we can compare the distance between two configurations with the distance between them following the medial axis, and give up if there is too large a discrepancy. Random walks might be weighted in the direction suggested by following the medial axis.

We showed positive results for a difficult problem with a rigid robot in three dimensions. There is a natural extension to articulated and flexible robots. For flexible objects, as in [5], we can integrate the energy function for deformations with an energy term based on the spring model of the rigid case. For articulated robots, handle points can be assigned for each link and a similar minimization procedure moves the robot onto the medial axis. The key to these methods is encoding configuration information into an energy function which can be evaluated in the context of the workspace. In the future we hope to test this framework with a wider variety of planning problems and to investigate additional uses for the geometric information contained in the medial axis. Additional work is also planned for choosing handle points in a more automatic fashion. Also, our methods for computing the medial axis, while effective, could be enhanced in terms of robustness and efficiency.

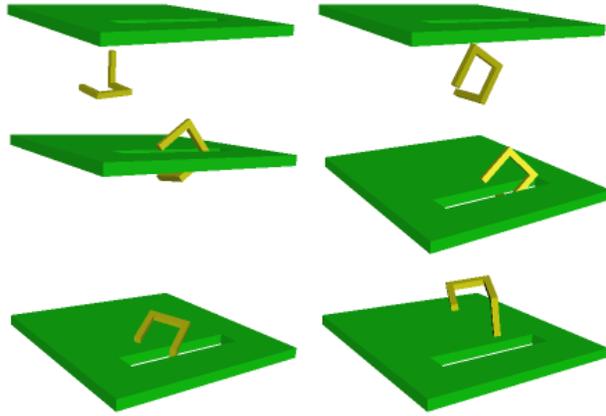


Figure 9: Path for example 2

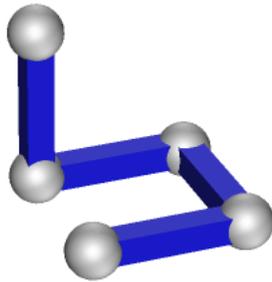


Figure 10: Handle points for example 2

Acknowledgments

Work on this paper by Lydia Kavraki and Rusty Holleman has been supported in part by NSF CAREER Award IRI-970228 and NSF CISE SA1728-21122N. The authors would like to thank Leo Guibas, Tim Danner, Ron Goldman, Florent Lamiroux, Jean-Claude Latombe, Nina Amenta, Elliot Anshelevich and Scott Owens for their comments. We would also like to thank Leo Joskowitz for the example data.

References

- [1] J. M. Ahuactzin, E.-G. Talbi, P. Bessière, and E. Mazer. Using genetic algorithms for robot motion planning. In *10th Europ. Conf. Artif. Intelligence*, pages 671–675, London, England, 1992.
- [2] N. Amato, B. Bayazit, L. 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, *Robotics: The Algorithmic Perspective*. AK Peters, 1998.
- [3] N. Amenta. Personal Communication.
- [4] N. Amenta, M. Bern, and M. Kamvysellis. A new voronoi-based reconstruction algorithm. In *SIGGRAPH*, Orlando, FL, 1998.
- [5] E. Anshelevich, S. Owens, F. Lamiroux, and L. Kavraki. Deformable volumes in path planning applications. To appear in *ICRA '00*, 2000.
- [6] V. Boor, M. Overmars, and F. van der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1018–1023, 1999.
- [7] H. Chang and T. Li. Assembly maintainability study with motion planning. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, pages 1012–1019, 1995.
- [8] H. Choset. Incremental construction of the generalized voronoi diagram, the generalized voronoi graph, and the hierarchical generalized voronoi graph. In *Proc. of 1st CGC Workshop on Comp. Geom.*, Baltimore, MD, 1997.
- [9] P. W. Finn and L. E. Kavraki. Computational approaches to drug design. To appear in *Algorithmica*.
- [10] L. Guibas, C. Holleman, and L. Kavraki. A probabilistic roadmap planner for flexible objects with a workspace medial-axis-based sampling approach. In *Proc. IEEE Int. Conf. on Intelligent Robots*, 1999.
- [11] K. Gupta and A. P. del Pobil. *Practical Motion Planning in Robotics*. John Wiley, West Sussex, England, 1998.
- [12] D. Halperin, L. E. Kavraki, and J.-C. Latombe. Robotics. In J. Goodman and J. O'Rourke, editors, *Discrete and Computational Geometry*, pages 755–778. CRC Press, NY, 1997.
- [13] C. Holleman, L. E. Kavraki, and J. Warren. Planning paths for a flexible surface patch. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 1998.
- [14] D. Hsu, L. E. Kavraki, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In P. Agarwal, L. E. Kavraki, and M. Mason, editors, *Robotics: The Algorithmic Perspective*, pages 141–154. AK Peters, Natick, MA, 1998.
- [15] L. Joskowitz and R. H. Taylor. Interference-free insertion of a solid body into a cavity: An algorithm and a medical application. *The International Journal of Robotics Research*, 1995.
- [16] L. Kavraki and J.-C. Latombe. Randomized preprocessing of configuration space for fast path planning. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2138–2145, San Diego, CA, 1994.
- [17] L. E. Kavraki and J.-C. Latombe. Probabilistic roadmaps for robot path planning. In K. G. nd A. P. del Pobil, editor, *Practical Motion Planning in Robotics: Current Approaches and Future Challenges*, pages 33–53. John Wiley, West Sussex, England, 1998.
- [18] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. Probabilistic roadmaps for fast path planning in high dimensional configuration spaces. *IEEE Tr. on Rob. and Autom.*, 12:566–580, 1996.
- [19] Y. Koga, K. Kondo, J. Kuffner, and J.-C. Latombe. Planning motions with intentions. *Computer Graphics (SIGGRAPH'94)*, pages 395–408, 1994.
- [20] K. Kondo. Motion planning with six degrees of freedom by multistrategic bidirectional heuristic free-space enumeration. *IEEE Tr. on Robotics and Automation*, 7(3):267–277, 1991.
- [21] J.-C. Latombe. *Robot Motion Planning*. Kluwer, Boston, MA, 1991.
- [22] S. M. LaValle, J. H. Yakey, and L. E. Kavraki. A probabilistic roadmap approach for systems with closed kinematic chains. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, 1999.
- [23] M. Overmars and P. Svestka. A probabilistic learning approach to motion planning. pages 19–37. A K Peters, Ltd., 1995.
- [24] S. A. Wilmarth, N. M. Amato, and P. F. Stiller. Maprm: A probabilistic roadmap planner with sampling on the medial axis of the freespace. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, Detroit, MI, 1999.