Capturing the Connectivity of High-Dimensional Geometric Spaces by Parallelizable Random Sampling Techniques

David Hsu[1], Lydia E. Kavraki[2], Jean-Claude Latombe[1], and Rajeev Motwani[1]

[1] Computer Science Department, Stanford University, Stanford, CA 94305, USA
[2] Computer Science Department, Rice University, Houston, TX 77005, USA

Abstract. Finding paths in high-dimensional gemetric spaces is a provably hard problem. Recently, a general randomized planning scheme has emerged as an effective approach to solve this problem. In this scheme, the planner samples the space at random and build a network of simple paths, called a probabilistic roadmap. This paper describes a basic probabilistic roadmap planner, which is easily parallelizable, and provides a formal analysis that explains its empirical success when the space satisfies two geometric properties called ϵ-goodness and expansiveness.

1 Introduction

The path planing problem can be stated as follows:

Given:
- A geometric and kinematic model of a rigid or articulated object, called the *robot*,
- A geometric model of the obstacles in the physical space where the robot operates,

Find a *path*, i.e., a continuous sequence of collision-free configurations of the robot, connecting two arbitrary input configurations, called *query* configurations, whenever such a path exists; otherwise indicate that no such path exists.

A classical way to look at this problem is to represent a configuration by n independent parameters, one for each of the robot's degrees of freedom (dof). Thus, each physical placement of the robot is represented by a point in an n-D parameter space, called the robot's *configuration space* (or *C-space*). The obstacles map into that space as forbidden regions, called *C-obstacles*. The complement of the C-obstacles in the C-space is referred to as the *free space*. A path is a continuous curve segment in the free space connecting the two query configurations. Such a path exists if and only if the query configurations lie in the same component of the free space.

Robot path planning is a provably hard problem [25]. There is strong evidence that it requires exponential time in the dimension n of the C-space. This result still holds for specific robots such as planar linkages consisting of links serially

connected by revolute joints [14] and sets of rectangles executing axis-parallel translations in a rectangular workspace [8, 9]. Though general and complete algorithms have been proposed [4, 27], their high complexity precludes any useful application.

The dimension of the C-space beyond which existing complete algorithms become practically useless is low, somewhere between 3 and 5. This means that they cannot be applied to rigid objects translating and rotating in 3-D workspace, nor to 6-dof manipulator arms, two important cases in practice. On the other hand, applications tend to involve more degrees of freedom than ever before. For example, manufacturing workcells use with more than 20 dofs are no longer exceptions. In computer graphics, animation of synthetic actors to produce digital movies or video games requires dealing with several dozen dofs. In molecular biology, motion planning can help compute plausible docking motions of molecules modeled as spatial linkages with many dofs.

Recently, a general randomized planning scheme has emerged as a viable approach to path planning in high-dimensional C-spaces. In this scheme, no explicit representation of the free space is computed. Instead, the free space is implicitly defined by a function $\mathtt{dist}(q)$ that computes the Euclidean distance between the robot at configuration q and the obstacles. Several reasonably efficient implementations of such a function have already been proposed (e.g., [6, 7, 13, 19, 20, 21, 24]). The planner samples the C-space at random. Using $\mathtt{dist}$, it retains the configurations in free space as *milestones* and, for every pair of milestones, it checks that a simple path (usually, the straight path in C-space) connecting them is collision-free. The result is a graph R called a *probabilistic roadmap*. Given a pair of query configurations, the planner tries to connect each of them to a node of R. It outputs a path if it connects the two configurations to two milestones in the same component of R. A number of planners based on this scheme have been proposed [1, 2, 3, 12, 10, 15, 17, 23, 26]. None of them is complete in the strongest sense, but most achieve some form of probabilistic completeness, i.e., if a path exists, the planner will find one with high probability. In fact, implemented probabilistic roadmap planners have been remarkably successful in solving difficult path-planning problems in high-dimensional C-spaces. Moreover, the computational scheme is easily parallelizable, though to our knowledge this feature has not yet been significantly exploited.

Section 2 describes a "basic" probabilistic roadmap planner that performs a uniform sampling of the C-space, and sketches more sophisticated sampling strategies proposed in the literature. A "good" probabilistic roadmap is one which provides adequate coverage of the free space, so that every query configuration can easily be connected to it, and whose connectivity conforms to that of the free space. In Sections 3 and 4 we will formally analyze how large a roadmap needs to be (i.e., how many milestones it should contain) in order to achieve adequate coverage and connectivity with high probability. This analysis shows that the probabilistic roadmap approach is efficient if the free space satisfies two geometric assumptions that we call ϵ-*goodness* [16] and *expansiveness* [12]. Under these assumptions, the free space does not contain "narrow passages". Dealing

with such passages is the main remaining issue for probabilistic roadmap planners. In Section 5 we will briefly present ongoing efforts addressing this issue.

This paper does not present any experimental result with implemented probabilistic roadmap planners. Such results have been presented in several previous papers, with robots having from 3 to several dozen dofs. Complex practical applications of these planners include maintenance planning for aircraft engines [5], design for manufacturing in the automotive industry [12], and graphic animation of human characters [18].

2 Basic Probabilistic Roadmap Planner

Let $\mathcal{C}$ denote the C-space of a robot and $F \subset \mathcal{C}$ its free space. For simplification, we assume that $\mathcal{C}$ is the Euclidean hyper-cube $[0,1]^n$. We say that two free configurations *see* each other if they can be connected by a straight-line path in F.

The basic probabilistic roadmap planner is a simplified version of the planner described in [17]. It consists of two algorithms: `roadmap`, which precomputes a probabilistic roadmap, and `query`, which uses this roadmap to answer pathplanning queries. Each query is defined by two configurations, q_b and q_e, in F.
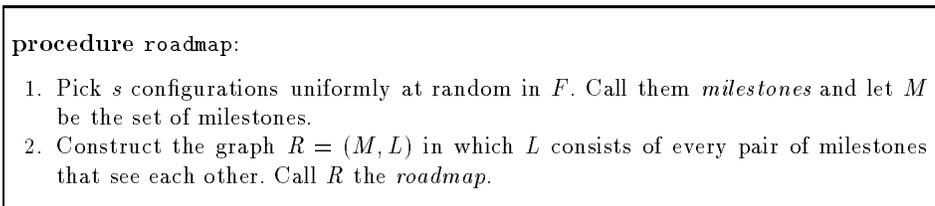
---

procedure `roadmap`:

1. Pick s configurations uniformly at random in F. Call them *milestones* and let M be the set of milestones.
2. Construct the graph $R = (M, L)$ in which L consists of every pair of milestones that see each other. Call R the *roadmap*.

Fig. 1. Roadmap-construction algorithm

2.1 Roadmap Construction

The procedure `roadmap` constructs a roadmap in two steps, as shown in Figure 1. The milestones are chosen at Step 1. The links between milestones are created at Step 2.

Recall that `dist`(q) is a procedure that computes the Euclidean distance between the robot placed at q and the obstacles. Step 1 generates each milestone by picking successive configurations q in $[0,1]^n$, until one satisfies `dist`$(q) > 0$. Every q is obtained by choosing each of its coordinates at random uniformly in $[0,1]$. Step 2 checks the straight path between every two milestones for collision, by recursively decomposing it into two half segments and invoking `dist` at each segment endpoint. One can show that if a segment is short enough relative to

the robot-obstacle distance computed at its two endpoints, the whole segment is guaranteed to be collision-free [2].

---

procedure `query`:

1. **for** $i = \{b, e\}$ do:
   (a) **if there exists** a milestone m that sees q_i **then** $m_i \leftarrow m$.
   (b) **else**
      i. **repeat** t **times:**
         Pick a configuration q in F uniformly at random in a neighborhood of q_i
         until q sees both q_i and a milestone m.
      ii. **if all** t trials failed **then return** FAILURE, **else** $m_i \leftarrow m$.
2. **if** m_b and m_e are in the same component of the roadmap, **then return** a path connecting them, **else return** NO-PATH.

Fig. 2. Query processing algorithm

2.2 Query Processing

The query-processing algorithm is shown in Figure 2. It tries to connect each of the query configurations to a milestone of the roadmap, either directly (Step 1(a)), or through an intermediate configuration chosen in a neighborhood of the query configuration (Step 1(b)). The implementation of Step 1 makes use of the function `dist`. Each free configuration q at Step 1(b)i is obtained by picking successive configurations at random in a hyper-cube centered at q_i until one is collision-free. The `query` procedure returns NO-PATH if it connects the query configurations to two distinct components of the roadmap. This answer is correct whenever no two components of the roadmap lie in the same component of F. Moreover, the procedure outputs FAILURE if it cannot connect a query configuration to some milestone of the roadmap. We would like the planner to rarely return either an incorrect answer or FAILURE.

2.3 Other Sampling Strategies

A variety of more sophisticated sampling strategies have been proposed. We review some below.

The roadmap planner in [15, 17] constructs an initial roadmap by uniformly sampling the C-space. Then it performs a resampling step for improving roadmap connectivity. In this second step, additional milestones are added in the neighborhood of milestones that see no or few other milestones. Experiments have shown that this resampling scheme makes it possible to adequately represent the free space connectivity with smaller roadmaps.

A number of roadmap planners use sampling strategies aimed towards generating a greater density of milestones near the boundary of the free space [1, 10, 22]. In [1], when a configuration q is generated outside F, a number of rays are shot from q along random directions uniformly distributed in $\mathcal{C}$. For each ray, a binary search is used to identify a point on the boundary of F. In [22] a single ray is shot from q along a random direction; the procedure then simulates a walk of the robot along this direction, in small incremental steps, until it is in free space. The technique described in [10] tries to connect roadmap components that could not be connected by straight paths, by creating milestones in the free space boundary, A random direction is chosen starting from a milestone in one component. Using a technique similar to [22]. a milestone is created where this ray encounters the free space boundary, and the ray is reflected in a random direction at this point to find another boundary point. Etc. All the references listed above observe that adding milestones near the free space boundary improves the planners' performance.

Precomputing a roadmap is advantageous when multiple path-planning queries are made in the same free space. To deal with single-query cases, some planners build a new roadmap for each new query. The planner in [12] generates clouds of milestones picked at random in small neighborhoods of previously generated milestones, expanding from the two query configurations, until two clouds meet (bi-directional search). The planner in [3] starts from one of the query configurations and builds an expanding set of milestones until one can be connected to the other query configuration (one-directional search); in this planner, milestone selection is strongly biased by a heuristic "potential field".

3 Roadmap Coverage

A first desirable property for a probabilistic roadmap is that it provides adequate coverage of the free space F. This means that the milestones should collectively see a large portion of F, so that any query configuration can easily be connected to one of them. Note that Step 1(b) of query allows for the case where a query configuration does not see any milestone. Indeed, it would be unrealistic to expect that a probabilistic roadmap provides complete coverage of F; in general, the probability of picking a new milestone that sees a portion of F not seen by previous milestones decreases and tends toward zero as the number of milestones grows.

In this section we establish that the milestones chosen by **roadmap** see a large portion of F with high probability if every point in F sees a significant portion of F (a property that we call ϵ-goodness). We also state that when the roadmap achieves adequate coverage of the free space, **query** efficiently connects query configurations to the roadmap.

For any subset $S \subseteq \mathcal{C}$, we let $\mu(S)$ denote its volume. For any $q \in F$, $\mathcal{V}(q)$ denotes the set of all free configurations seen by q; we call it the *visibility set* of q.

Definition 1. Let ϵ be a constant in $(0, 1]$. A free configuration q is ϵ-good if $\mu(\mathcal{V}(q)) \geq \epsilon\mu(F)$. The free space is ϵ-good if every $q \in F$ is ϵ-good.

Definition 2. A set of milestones provides *adequate coverage* of an ϵ-good free space F if the volume of the subset of F not visible from any of these milestones is at most $(\epsilon/2)\mu(F)$.

Note that, as ϵ increases, the coverage requirement grows weaker, i.e., the portion of F that has to be visible by at least one milestone gets smaller. This comes from the fact that a greater ϵ will make it easier for `query` to connect query configurations to the roadmap. Naturally, the number of milestones needed becomes smaller.

Theorem 3. *Assume that F is ϵ-good. Let ϕ be a constant in $(0, 1]$ and K be a positive real large enough that for any $x \in (0, 1]$, $(1 - x)^{(K/x)\log(2/x\phi)} \leq x\phi/2$. If s is chosen such that:*

$$s \geq \frac{K}{\epsilon}(\log\frac{1}{\epsilon} + \log\frac{2}{\phi}),$$

then `roadmap` generates a set of milestones that adequately covers F, with probability at least $1 - \phi$.

We refer the reader to [2, 16] for a proof of this theorem.

Theorem 3 does not allow us to compute s since we do not know the value of ϵ, except for simple spaces. Nevertheless, its significance is twofold. First, it tells us that although adequate coverage of the free space is not guaranteed, the probability that this happens decreases exponentially with the number of milestones. Second, the number of milestones needed increases moderately when ϵ decreases.

It now remains to establish that adequate coverage allows `query` to connect any query configuration to the roadmap, with high probability.

Theorem 4. *Let the maximum number of iterations t at Step 1(b)i of `query` be set to $\log(2/\psi)$, where ψ is a constant in $(0, 1]$. If the milestones provide adequate coverage of F, then the probability that `query` outputs FAILURE is at most ψ.*

In other words, the failure probability of `query` decreases exponentially with the number t of iterations at Step 1(b)i. See [16] for a proof of this theorem.

However, ϵ-goodness is too weak to guarantee that `roadmap` will construct a roadmap whose connectivity represents that of the free space. For example, the free space of Figure 3 is ϵ-good for $\epsilon \approx 0.5$. But a roadmap of moderate size constructed by `roadmap` will most likely consist of two connected components. In [16] we dealt with this issue by allowing `roadmap` to invoke a complete planner to try to connect the components of a roadmap. However, running such a planner can be totally impractical. In [12] we eliminated its need by introducing the notion of an expansive free space.
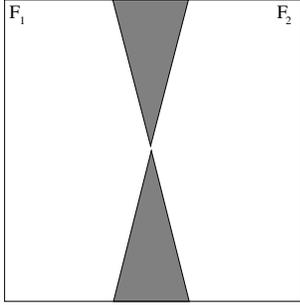
Fig. 3. A narrow passage in an ϵ-good space

4 Roadmap Connectedness

Let us now define precisely the kind of roadmap we would like **roadmap** to construct.

Definition 5. Let F be an ϵ-good free space. A roadmap R is an *adequate representation* of F if its milestones provide adequate coverage of F and no two components of R lie in the same component of F.

Let R be an adequate representation of F. Since F is ϵ-good, no component of F has volume less than $\epsilon\mu(F)$. Therefore, at least one milestone of R lies in every component of F. Since no two components of R lie in the same component of F, there is a one-to-one correspondence between the components of R and those of F.

The notion of an expansive free space is directly related to the difficulty that **roadmap** has to connect milestones through narrow passages. The reason why it would require considerable time for the roadmap-construction procedure to build a connected roadmap in the free space of Figure 3 is that a very small subset of points in F_1 (the half space on the left) see a large fraction of F_2 (the half space on the right); therefore, the probability that the planner picks a milestone in F_1 that sees a milestone in F_2 is small. By narrowing the passage between F_1 and F_2, one can make this probability arbitrarily small. Let us refer to the subset of points in a subset $S \subset F$ that can see a large portion of $F \backslash S$ as the lookout of S. The previous example suggests that we characterize narrow passages by the minimum volume of the lookout of the visibility set $\mathcal{V}(q)$ over all $q \in F$. If a set $\mathcal{V}(q)$ has a small lookout, **roadmap** will have difficulty computing a roadmap capturing the connectivity of the free space.

Definition 6. Let β be a constant in $(0, 1]$ and S be a subset of a component E of the free space F. The *β-lookout* of S is the set:

$$\beta\text{-}\textsc{lookout}(S) = \{q \in S \mid \mu(\mathcal{V}(q)\backslash S) \geq \beta \times \mu(E\backslash S)\}.$$

Definition 7. Let ϵ, α, and β be constants in $(0, 1]$. The free space F is $(\epsilon, \alpha, \beta)$-*expansive* if it is ϵ-good and for every $q \in F$ we have:

$$\mu(\beta\text{-LOOKOUT}(\mathcal{V}(q))) \geq \alpha \times \mu(\mathcal{V}(q)).$$

For simplification, we will abbreviate the term "$(\epsilon, \alpha, \beta)$-expansive" by "expansive".

Theorem 8. *Assume that F is $(\epsilon, \alpha, \beta)$-expansive. Let ξ be a constant in $(0, 1]$. If s is chosen such that:*

$$s \geq \frac{16}{\epsilon \alpha} \log \frac{8}{\epsilon \alpha \xi} + \frac{6}{\beta} + 4,$$

then `roadmap` *generates a roadmap that is an adequate representation of F with probability at least $1 - \xi$.*

Given a set M of milestones, let a *linking sequence* of a milestone m_0 be a sequence $m_1, m_2, ...$, such that for all $i > 1$, $m_i \in \text{LOOKOUT}(\mathcal{V}(m_{i-1}))$. Let the visibility set of a linking sequence be the union of the visibility sets $\mathcal{V}(m_i)$ of all the milestones m_i in the sequence. The proof of Theorem 8, given in [12], first establishes that in a properly sized roadmap, for every pair of milestones (m, m') in the same component of F, with high probability m admits a linking sequence whose visibility set intersects that of a linking sequence of m'; it then shows that with high probability there exists a milestone which lies in the intersection of the visibility sets of these two linking sequences, so that m and m' are in the same component of the roadmap.

Again, the significance of Theorem 8 is twofold. The probability that a roadmap does not adequately represent F decreases exponentially with the number of milestones, and the number of milestones needed increases moderately when ϵ, α, and β decrease.
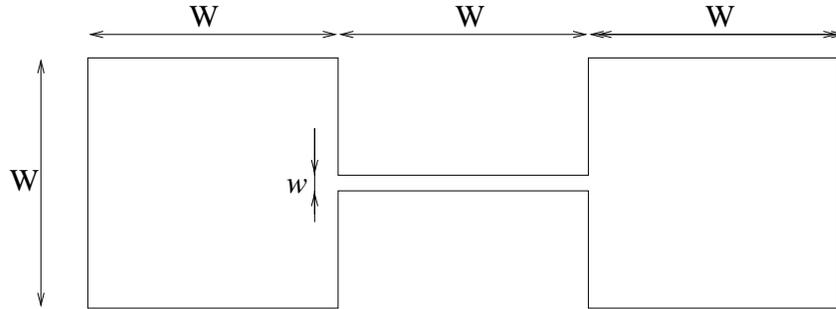

Fig. 4. An $(\epsilon, \alpha, \beta)$-expansive free space where $\epsilon, \alpha, \beta \sim w/W$

Note that none of Theorems 3, 4, or 8, explicitly mention the dimension n of the configuration space. This comes from the fact that both ϵ-goodness and

expansiveness are visibility properties whose definitions only refer to volumes of subsets of F. But the dependence on n may be hidden in the parameters ϵ, α, and β. To illustrate this point, consider the example of Figure 4. The free space consists of two cubes whose sides have length W; these cubes are connected by a rectangular narrow passage of length W and width w, where $w \ll W$. Up to a constant factor, each of the parameters ϵ, α, and β is on the order of w/W. Indeed, the volume of F is $W(2W + w) \approx 2W^2$. The points with the smallest ϵ-goodness are located in the narrow passage. Each such point sees only a subset of F of volume approximately $3wW$; hence, $\epsilon \approx 3w/2W \sim w/W$. A point near the top right corner of the left square sees this entire square; but only a subset of this square, of approximate volume wW, contains points that, each, see a set of volume $2wW$; hence, $\alpha \approx w/2W \sim w/W$ and $\beta \approx w/W$. In the n-D version of this example, two hyper-cubes, each having volume W^n, are connected by a hyper-parallelepipedic passage that has size w along k dimensions ($k \in [1, n-1]$) and size W along the $n - k$ other dimensions. Each of the parameters ϵ, α, and β is on the order of $(w/W)^k$. The worst case happens when $k = n - 1$, that is, when the passage is narrow along $n - 1$ dimensions.

5 Current and Future Work

During the last few years a number of path planning algorithms based on the construction of probabilistic roadmaps have been proposed and experimented with great success. This paper has described a basic probabilistic roadmap planner and has provided a formal analysis that explains its empirical success.

However, current probabilistic roadmap planners share the same relative inability to efficiently find paths through narrow channels. This inability has been experimentally observed, and it is formally explained by our results in expansive free space. Current research aims at dealing efficiently with free spaces that are poorly expansive.

Sampling strategies that generate a greater density of milestones near the boundary of the free space are intended to facilitate the discovery of narrow passages. Though they have been somewhat successful, they are not sufficient and this is not surprising. Recently, we have devised a new strategy that consists of first generating a roadmap in an expanded free space and then "pushing" this roadmap in the original free space [11]. Expanding the free space increases its expansiveness, with the narrow channels benefiting much more than the already large areas of the free space. Experimental results, though still preliminary, are very encouraging.

References

1. Amato, N., Wu, Y: A Randomized Roadmap Method for Path and Manipulation Planning. *Proc. IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN (1996) 113-120.

2. Barraquand, J., Kavraki, L.E., Latombe, J.C., Li, T.Y., Motwani, R., Raghavan, P.: A Random Sampling Scheme for Path Planning. *Int. J. of Robotics Research*, 16(6) (1997) 759–774.

3. Barraquand, J., Latombe, J.C.: Robot Motion Planning: A Distributed Representation Approach. *Int. J. of Robotics Research*, 10(6) (1991) 628-649.

4. Canny, J.F.: *The Complexity of Robot Motion Planning*, Cambridge:MIT Press (1988).

5. Chang, H., Li, T.Y.: Assembly Maintainability Study with Motion Planning, *Proc. IEEE Int. Conf. on Robotics and Automation*. Nagoya: IEEE (1995) 1012-1019.

6. Dobkin, D.O., Hershberger, J., Kirkpatrick, D.G., Suri, S.: Computing the Intersection Depth of Polyhedra. *Algorithmica*, 9 (1993) 518-533.

7. E.G. Gilbert, D.W.Johhson, and S.S. Keerthi. A Fast Procedure for Computing the Distance Between Complex Robots in Three-Dimensional Space. *IEEE Transactions on Robotics and Automation*, 4:193-203, 1988.

8. Hopcroft, J.E., Schwartz, J.T., and Sharir, M. 1984. On the Complexity of Motion Planning for Multiple Independent Objects: PSPACE-Hardness of the 'Warehouseman's Problem'. *Int. J. of Robotics Res.* 3(4):76-88.

9. Hopcroft, J.E. and Wilfong, G.T. 1986. Reducing Multiple Object Motion Planning to Graph Searching. *SIAM J. on Computing.* 15(3):768-785.

10. T. Horsch, F. Schwarz, and H. Tolle. Motion Planning for Many Degrees of Freedom - Random Reflections at C-Space Obstacles. *Proc. IEEE Int. Conf. on Robotics and Automation*, San Diego, CA April 1994, pp. 3318-3323.

11. D. Hsu, L.E. Kavraki, J.C. Latombe, R. Motwani, and S. Sorkin. On Finding Narrow Passages with Probabilistic Roadmap Planners. To appear in *Proc. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Houston, TX, April 1988.

12. D. Hsu, J.C. Latombe, and R. Motwani. Path Planning in Expansive Configuration Spaces. Proc. *IEEE Int. Conf. on Robotics and Automation*, Albuquerque, NM, 1997, pp. 2719-2726. An extended version of this paper will appear in *Int. J. of Computational Geometry and Applications*.

13. P. Jiménez, F. Thomas, and C. Torras. Collision Detection Algorithms for Motion Planning. *Robot Motion Planning and Control*, J.P. Laumond (ed.), Lecture Notes in Control and Information Sciences, 229, Springer, New York, NY, 1998, pp. 305-343.

14. Joseph, D.A. and Plantiga, W.H. 1985. On the Complexity of Reachability and Motion Planning Questions. *Proc. 1st ACM Symp. on Computational Geometry*, pp. 62-66.

15. L. Kavraki. *Random Networks in Configuration Space for Fast Path Planning.* Ph.D. Thesis, Rep. No. STAN-CS-TR-95-1535, Department of Computer Science, Stanford Univ., Stanford, CA, 1995.

16. L. Kavraki, J.C. Latombe, R. Motwani, and P. Raghavan. Randomized Query Processing in Robot Motion Planning. *Proc. ACM SIGACT Symposium on the Theory of Computing (STOC)*, Las Vegas, Nevada, 1995, pp. 353-362.

17. L. Kavraki, P. Švestka, J.C. Latombe, and M. Overmars Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566-580, August 1996.

18. Koga, Y., Kondo, K., Kuffner, J., and Latombe, J.C. 1994. Planning Motions with Intentions. *Proc. of SIGGRAPH'94*, ACM, pp. 395-408.

19. M. Lin and J.F. Canny. A Fast Algorithm for Incremental Distance Computation. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Sacramento, CA, 1994,

pp. 602-608.

20. M. Lin, D. Manocha, J. Cohen, and S. Gottschalk. Collision detection: Algorithms and applications. *Algorithmic Foundations of Robotics*, Goldberg et al. (Eds), A K Peters, Ltd., 1995, pp. 129-141.

21. C.J. Ong and E.G. Gilbert. Growth Distances: New Measures for Object Separation and Penetration. *IEEE Tr. on Robotics and Automation*, 12(6):888-903, 1996.

22. M. Overmars. A random Approach to Motion Planning. Technical Report, RUU-CS-92-32, Department of Computer Science, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands, 1992.

23. M. Overmars and P. Švestka. A Probabilistic Learning Approach to Motion Planning. *Algorithmic Foundations of Robotics*, K. Goldberg et al. (eds.), A.K. Peters, Wellesley, MA, 1995, pp. 19-37.

24. S. Quinlan. Efficient Distance Computation Between Non-Convex Objects. *Proc. IEEE Int. Conf. on Robotics and Automation*, San Diego, CA, 1994, pp. 3324-3330.

25. Reif, J. 1979. Complexity of the Mover's Problem and Generalizations. *Proc. IEEE Symp. on Foundations of Computer Science*. IEEE, pp. 421-4127.

26. P. Švestka and M. Overmars. Probabilistic Path Planning. *Robot Motion Planning and Control*, J.P. Laumond (ed.), Lecture Notes in Control and Information Sciences, 229, Springer, New York, NY, 1998, pp. 255-304.

27. Schwartz, J.T. and Sharir, M. 1983. On the 'Piano Movers' Problem: II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds. *Advances in Applied Mathematics*. 4:298-351.