A Dynamic Motion Planning Problem Using Alarms

Rajeev Sharma
Center for Automation Research, University of Maryland
College Park, MD 20742
Anup Basu
Department of Computing Science, University of Alberta
Edmonton, Alberta, CANADA T6G 2H1

Abstract

We consider the problem of motion planning for a point robot using a probabilistic model for a partially known dynamic environment. The static known part of the environment consists of point shelters distributed in a planar terrain, and the dynamic, unknown part is abstracted in the form of alarms *that cause the robot to leave its current (pre-planned) path and divert to the nearest shelter. We give a probabilistic analysis of the expected times for the dynamic paths generated when the alarms follow a Poisson distribution with parameter λ. Two strategies are presented suitable for low and high values respectively of the alarm rate λ. We also discuss ways of generalizing the approach and possible applications.*

1 Introduction

The ability to navigate successfully in a dynamically changing environment would be an important feature of the behavior of an autonomous mobile robot. In the absence of complete *a priori* knowledge of the environment, a robot would have to rely on its sensory input to detect moving (e.g. [9]) and stationary obstacles in its vicinity and adapt its path according to its current set of goals. A large portion of the motion planning literature (see [11] for a good survey) is concerned with planning efficient paths using models that either assume a complete knowledge of the environment or that simply ignore its dynamic nature. We are concerned here with analyzing and comparing alternate motion planning strategies in a framework that captures both the dynamic *and* uncertain elements of the environment.

For motion planning in a dynamic environment the requirement that the motion of the obstacles be known at the time of planning is fairly restrictive since it does not represent a typical situation. Moreover, it also turns out that with complete information the problem of motion planning in a dynamic environment is computationally hard as demonstrated by various lower complexity bounds reported in the literature [7]. These intractability results have encouraged heuristic approaches, especially as applied to more limited geometric models — for example, in [3]. However, these models are inadequate for dealing with a dynamic environment that is known only partially.

On the other hand, considerable research has been done in path planning with incomplete information but where the environment is static. Typically, a point robot is assumed to have positional information only about the starting point and the goal point, and is assumed to possess a sensory device, the input from which is then used for path planning. For example, a primitive operation of visual input or "scan" is used in [6] to build structures based on the visibility graph or the Voronoi diagram of the two-dimensional scene populated with non-overlapping polygonal obstacles. In [4] the two-dimensional obstacles are allowed to be of arbitrary shape, and it is then shown that for a point robot tactile sensory input is sufficient to guarantee reachability of the goal. Upper bounds on the length of paths generated by such algorithms are established in terms of the perimeters of the obstacles falling within a region. An interesting lower bound result in terms of path length is established in [4], applicable to any algorithm in the given framework. No systematic work, however, has been reported on how these approaches may be generalized to include a changing environment. Probabilistic methods presented in the literature largely deal with the uncertainty involved with the sensors [2, 1].

A model for the uncertain dynamic environment was presented in [8] that captures the dynamic variations in terms of *alarms* that obey a probabilistic distribution. It then becomes feasible to study the effects of the dynamic events on the robot's path, defining some rules that guarantee safe traversals with appropriate responses to the alarms. The criterion used

---

The support of the Defense Advanced Research Projects Agency (ARPA Order No. 6989), U.S. Army Engineer Topographic Laboratories (Contract DACA76-89-C-0019),IBM Graduate Fellowship and NSERC (under grant OGP0105739) is gratefully acknowledged.

to compare alternate paths is their *expected* dynamic costs for the given probabilistic model. Using this model an optimal solution to a *local* version of the navigation problem was obtained in [8]. Here we use the same dynamic model for the alarms — but consider a terrain with a set of *shelters* at known locations to which the robot can head for safety whenever an alarm occurs. Two strategies are presented and compared on the basis of their *expected* performance using the probabilistic model of the environment. Using this analysis it becomes possible to establish the superiority of each of the path planning strategies for a particular range of the probabilistic parameter.

2 Preliminaries

We consider the following model. The mobile automaton (MA) is a point and moves in two modes. In the *normal* mode MA moves with a constant velocity v and in the *emergency* mode it moves with a velocity V ($V >> v$). MA switches to an emergency mode whenever it detects certain events in its environment (e.g. through its visual sensors) called *alarms*. The planar terrain has a set of special locations called *shelters*. Immediately following an alarm, MA moves straight to the nearest shelter with a velocity V. Here we consider the case when the plane is populated with n point shelters with two special shelters — the starting point (S) and the goal point (G). The spatial locations of the shelters are assumed to be known while the times when the alarms occur are not known *a priori* but detected dynamically by MA. In order to analyze the expected dynamic behavior of MA for a particular path planning strategy, we use the Poisson process to model the occurrence of the dynamic events or alarms. (See [8] for a justication for using this model).

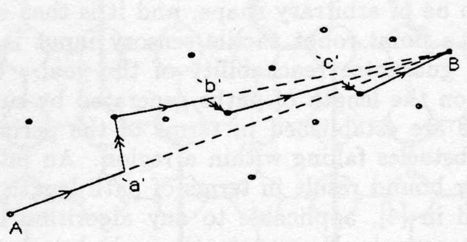

Figure 1.

We further define the following:

static path: The pre-planned path, preferably one that gives efficient performance for the model chosen. This is the path computed before MA starts moving, on the basis of a particular strategy (e.g. the dashed lines in Fig. 1).

dynamic path: The actual path taken by MA — starting out on a given static path and with the occurence of each alarm moving to the nearest shelter and continuing from there toward the goal (e.g. the solid, arrowed lines in Fig. 1).

The *expected* time to reach the goal following a particular strategy is used as a measure of performance. The problem is then to find strategies which generate the static paths from S to G with minimum expected time. We use the term $T_r[P]$ to denote the expected time to travel from a point P to the goal using a static path that either passes through the point r (when it r is a point shelter) or uses a segment r (when r is a line segment). The subscripts are dropped when there is no ambiguity in the context.

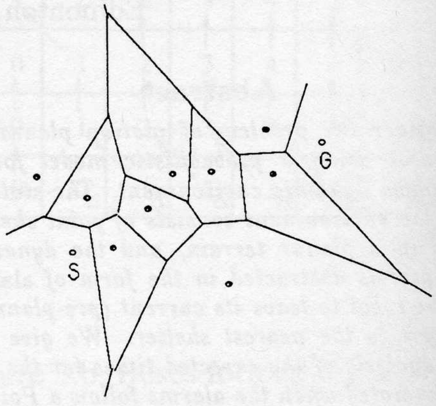

Figure 2.

3 Probabilistic Analysis

For the dynamic model used — it is useful to consider the Voronoi diagram (see [5] for definition and Fig. 2 of an illustration) for the n points on the plane that constitute the shelters. The Voronoi diagram incorporates all the proximity information needed to determine efficiently the nearest shelter from any point on the terrain. With each shelter P we can thus associate a convex polygon (possibly unbounded) called the Voronoi polygon V(P) having at most $n-1$ sides. If the automaton MA is anywhere in V(P) when the alarm occurs, then the dynamic rule requires it to move directly to P before restarting toward the goal.

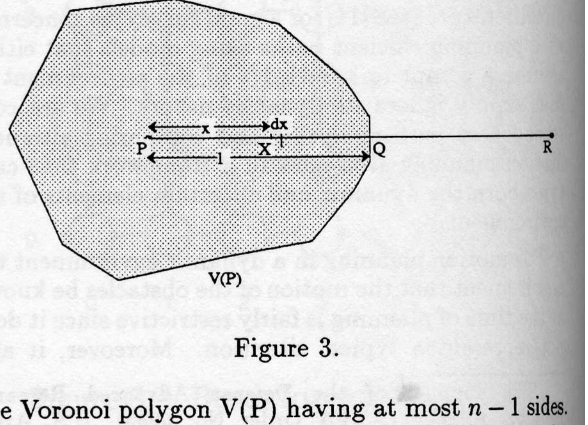

Figure 3.

We now state some general properties of the dynamic paths in the form of four lemmas, the proofs of which are given in [10].

Lemma 1 The expected time for the automaton MA to move out (or 'escape') from the Voronoi polygon V(P) of a shelter P (Fig. 3) along a given direction PQ is given by

$$T_{esc}[PQ] = \frac{1}{\lambda}(e^{\frac{\lambda l}{v}} - 1) \tag{1}$$

where $|PQ| = l$ and Q lies on the border of V(P).

Lemma 2 The expected time for the automaton MA to move (or 'home') to a shelter P from a point Q (Fig. 4) within the Voronoi polygon V(P) is given by

$$T_{home}[QP] = \frac{1}{\lambda}(1 - e^{-\frac{\lambda l}{v}}) \tag{2}$$

where $|QP| = l$.

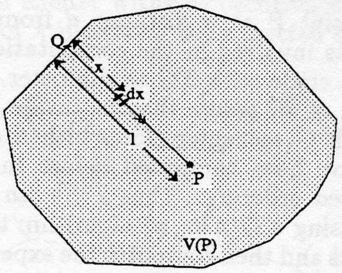

Figure 4.

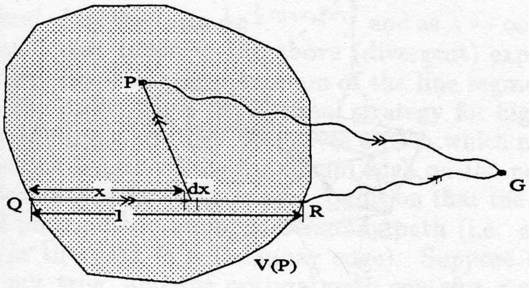

Figure 5.

Lemma 3 If the static path of MA from s to g includes a straight line segment QR that cuts through the Voronoi polygon of a shelter P (Fig. 5), the expected time for MA to travel from Q to the goal is given by

$$T_{QR}[Q] = (\frac{1}{\lambda} + T[P])(1 - e^{-\frac{\lambda l}{v}}) + T[R] \tag{3}$$

where T[P] and T[R] are the expected times for a path to the goal from P and R respectively and $|QR| = l$.

Lemma 4 Let AB be an arbitrary line segment ('edge') between two shelters A and B, lying on a static path from A to the goal (Fig. 6). If AB cuts across the Voronoi polygons of $k \geq 0$ intermediate shelters $P_1, P_2, \ldots P_k$, such that it can be divided into $k+2$ consecutive segments of length $a, l_1, l_2, \ldots l_k, b$ in the corresponding Voronoi region, then the expected time from A to the goal via AB is given by

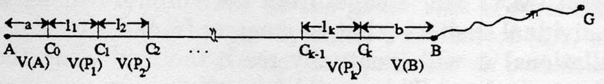

Figure 6.

$$\begin{aligned} T_{AB}[A] = \quad & \textstyle\sum_{i=1}^{k}[(\frac{1}{\lambda} + T[P_i])(1 - e^{-\frac{\lambda l_i}{v}})] + T[B] \\ & + \tfrac{1}{\lambda}(e^{\frac{\lambda a}{v}} - e^{-\frac{\lambda b}{v}}) \end{aligned} \tag{4}$$

where $T[P_i](i = 1 \ldots k)$ and $T[B]$ are the expected times for a path from $P_i(i = 1 \ldots k)$ and B respectively to the goal.

4 Strategy for Low Alarm Rate

The first strategy that we consider is the *direct strategy* of moving in a straight line to the goal. If and when the alarm forces MA to move to the nearest shelter, it again continues to move on a straight line to the goal (G).

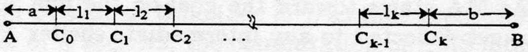

Figure 7.

Optimality at low λ. The superiority of this strategy for low values of λ is established as follows. We first show that as $\lambda \longrightarrow 0$, the expected dynamic cost of any arbitrary edge AB between two shelters (Fig. 7) becomes independent of all intermediate shelters, and varies linearly with the length AB. For low values of λ, we can use the approximation $e^{\frac{\lambda l}{v}} \approx 1 + \frac{\lambda l}{v}$, where the length of the path segment l and the velocity v are both finite. Thus substituting in the expression of the expected cost in lemma 4, we get

$$T_{AB}[A] \approx \frac{(a+b)}{v} + T[B] + \lambda . \sum_{i=1}^{k} \frac{l_i T[P_i]}{v} \tag{5}$$

In the limit when $\lambda \rightarrow 0$,

$$T_{AB}[A] = \frac{AB}{v} + T[B]$$

The expected dynamic cost thus approaches the static cost (time). Thus the optimal path would also be the shortest one and hence the direct strategy (which gives the shortest static path in this simple model without any 'obstacles') becomes optimal as $\lambda \rightarrow 0$. This also holds good for very low values of λ with the the probability of the direct strategy being the optimal one increasing with decreasing λ. This corresponds to the simple intuition that as the alarm rate approaches 0, the dynamic case becomes closer to the static case.

Convergence. We first show that once the mobile agent (MA) can 'escape' from the Voronoi regions of individual shelters (that is, emerges from the local oscillations) it will eventually reach the goal using the *direct strategy*. This would be true for any sequence of alarms and for any distribution of the shelters such that the distance D from start (S) to the goal (G) is finite. We then show that for low λ the expected time to escape the Voronoi region of any shelter for a finite value of D and v is finite thus substantiating the practical convergence of the algorithm.

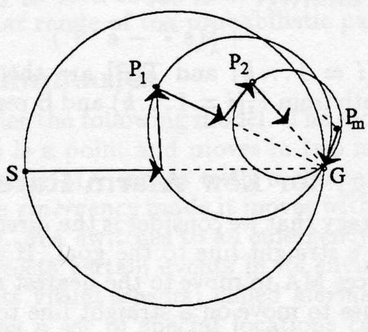

Figure 8.

As MA starts toward the goal on the line SG, for it to get deflected to any intermediate shelter P_1, the Voronoi region of P_1 must intersect SG, a condition that is true when P_1 lies inside the circle with SG as the diameter (see Fig. 8). According to the Direct Strategy, MA restarts from P_1 along a segment P_1G of length $d_1 < D$. The argument is applied recursively as follows. After the i^{th} alarm, the remaining distance d_i is less than d_{i-1}, the distance to the goal after the $(i-1)^{th}$ alarm, with $d_0 = D$. The condition that must be true for convergence is that MA should never revisit any shelter once it escapes its Voronoi region, so that there can be no cycles in the dynamic path, other than the local oscillations. Since the distances (Fig. 8) to the goal for the successive shelters form a monotonically decreasing sequence, it is clear that no shelter can be revisited once the path escapes the associated Voronoi region.

The expected time to escape from the Voronoi region of any shelter P along the direction to the goal G is given by Lemma 1 as

$$T_{esc}[PG] = \frac{1}{\lambda}(e^{\frac{\lambda l}{v}} - 1)$$

Or, using the earlier approximation for low values of λ,

$$T_{esc}[PG] \approx \frac{1}{\lambda}(1 + \frac{\lambda l}{v} - 1) = \frac{l}{v}$$

with the approximation becoming more and more accurate as $\lambda \rightarrow 0$. Now as shown earlier the distance remaining from any intermediate shelter is less than D, so that for all such points $l < D/2$ and for a finite D and v, the time $\frac{l}{v}$ to escape a shelter is finite. Thus it can be shown that with probability tending to 1, MA will escape each shelter that it visits.

Evaluation of expected cost. The exact value for the expected travel time can be computed as follows. We do not use the approximation for low λ because even though the direct strategy is proven to be optimal for low values of λ it may actually be used in a situation when that condition may not be true. Using Lemma 4 (and Fig. 7, for $b = g$) we get the expected time,

$$T[A] = T_{AG}[S] = \sum_{i=1}^{k}[(\tfrac{1}{\lambda} + T[P_i])(1 - e^{-\frac{\lambda l_i}{v}})] + \tfrac{1}{\lambda}(e^{\frac{\lambda a}{v}} - e^{-\frac{\lambda g}{v}}) \quad (6)$$

since T[G] = 0. Thus $T[S]$ depends on the computation of $T[P_i]$ for all the intermediate points P_i. The expected costs are computed using Eqn. 6 for the points in the order of increasing distance from the goal G. For any point P at a distance d from G, the only other points involved in the computation of $T[P]$ all lie inside a circle with TG as diameter. This in turn is a subset of all points that lie inside a circle of radius d with G as center, for which the costs would already have been computed before that of P using the proposed ordering. Thus $T[S]$ can be computed correctly using a $O(n \log n)$ algorithm that first sorts the n points and then computes the expected distances using equation 6.

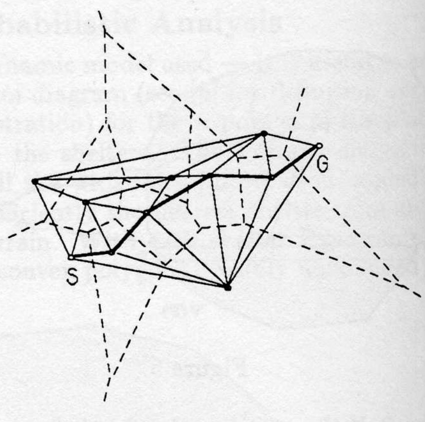

Figure 9.

5 Strategy for High Alarm Rate

We consider the situation when the alarm rate (λ) is high, and in particular the limiting situation when $\lambda \rightarrow \infty$. Under this condition, the expected cost (time) of crossing a path is dominated by the cost of crossing the largest edge in the path, thus motivating a navigation strategy that minimizes the maximum edge in a Delaunay path from any point to the goal point — a Delaunay path being defined as a sequence of edges from the Delaunay triangulation of the plane for the set of point shelters. See [5] for a definition of the Delaunay triangulation of the plane and Fig. 9 for an example of a Delaunay triangulation and a Delaunay path (the dark line). We study the properties of this MiniMax Delaunay Path (MDP), which can be given by an $O(n \log n)$ algorithm and is shown to be suitable for high λ.

Optimality at high λ. The general form of the expected time for a dynamic path across an arbitrary edge (from Lemma 4) as $\lambda \rightarrow \infty$ reduces to

$$\left(\frac{1}{\lambda}\sum_{i=1}^{k} e^{\frac{\lambda}{v}x_i}\right),$$

by recursive substitution in equation 4 and using the limiting conditions, $e^{-\frac{\lambda}{v}x_i} \rightarrow 0$ and $\frac{1}{\lambda} \rightarrow 0$, and where $k \leq n$ is finite and x_i are the various line segments for which MA can oscillate within the Voronoi region of a particular shelter with the occurrence of alarms. As the value of λ increases, the expected cost is thus dominated by the term $\frac{1}{\lambda}e^{\frac{\lambda}{v}\max_i\{x_i\}}$ and as $\lambda \rightarrow \infty$ the expected cost tends to the above (divergent) expression that involves the maximum of the line segments x_i in the path. Thus the optimal strategy for high λ is the Minimax strategy that gives a path which minimizes the length of the maximum edge on the path. We next argue for the stronger condition that the optimal path is the minimax Delaunay path (i.e. each edge in the path is a Delaunay edge). Suppose this were not true, and the optimal path contains a non-Delaunay edge ND. Consider an alternate path where ND is replaced by a Delaunay path. Now each of the edges that appear in this Delaunay path would have to be traversed with a high probability (with the probability $\rightarrow 1$ as $\lambda \rightarrow \infty$) in the original path. Thus the expected cost of the new path is at least as high as that of the original path. However, by choosing from all possible Delaunay paths one that has the smallest maximum edge to replace ND, the expected cost of the new path can be decreased. Thus the minimax Delaunay path (MDP) strategy that returns the minimax Delaunay path is optimal for high λ.

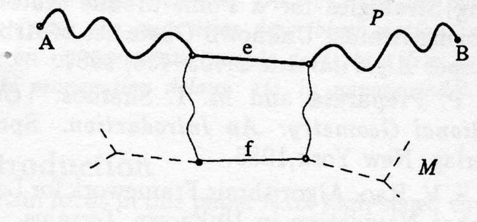

Figure 10.

EMST gives the Minimax Delaunay Path. An algorithm for MDP results by showing that a Euclidean Minimum Spanning Tree (EMST) (see [5] for a defintion) actually gives the MDP between any two points. An EMST would thus contain the optimal paths from any shelter to the goal. This is especially important since for the dynamic behavior of MA, we would like to store the optimal path for *all n* shelters since MA can restart toward the goal from any intermediate shelter, beside the starting point. We observe that an EMST is a subset of the Delaunay graph ([5]). Next we prove that a path on an EMST between two shelters is a Minimax path between them. Consider a minimax path P between two points A and B. Suppose it has an edge e that does not belong to the EMST. Consider the EMST M and add the edge e to it. This will introduce a cycle in the graph (Fig. 10). Let the tree M' be formed by removing the largest element f in the resulting cycle. Let P' be the path between A and B formed by replacing the edge e in P with the remaining part of the cycle. There are two possiblities. If e was not a largest edge in the cycle, then M' will be a spanning tree with a lower cost, contradicting the fact that M was the EMST. If e was a largest element then the new path P' is also a Minimax path. Thus a path confined to an EMST will give a Minimax path.

An EMST can be constructed in $O(n)$ time when a Delaunay triangulation of the plane for the n points is available. The Delaunay triangulation itself can be done in $O(n \log n)$ time (See [5]), so that given the locations of n shelters, the computation of EMST takes overall $O(n \log n)$ time. Thus for the MDS strategy the static path from any shelter to the goal is simply the unique path on the EMST from that shelter to the goal.

Evaluation of expected cost. The exact value for the expected travel time can be computed as follows. Again we do not use the approximation for high λ because the MDP strategy may be used for lower values of λ though it may not be the optimal for that range of λ. The computation of the expected cost for each edge becomes simple, using Lemmas 3 and 4. If the minimax Delaunay path has $k < n$ edges with lengths $a_1, a_2, \ldots a_k$ respectively, the expected cost of the dynamic path from S to G is given by

$$T[S] = \frac{1}{\lambda}\sum_{i=1}^{k}(e^{\frac{\lambda a_i}{2v}} - e^{-\frac{\lambda a_i}{2v}}).$$

6 Generalization

The model used in the previous case can be modified to allow for known polygonal obstacles in the environment, thus formulating a more general motion planning problem. The dynamic response of the robot to an alarm at a point is to move to a shelter in the minimum possible time. This would correspond to moving toward a shelter to which the shortest path from that point, while avoiding the obstacles, is the minimum among the n shelters. In Fig. 11, the dark line represents a dynamic path generated for a single alarm and for the static path shown with broken lines.

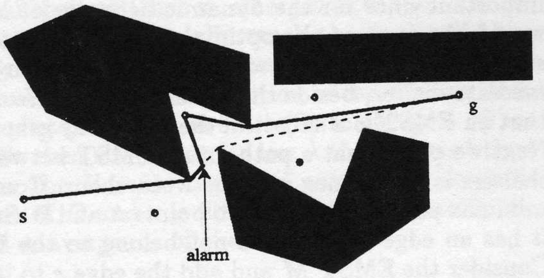

Figure 11.

Strategies can be given analogous to those for the case with no obstacles [10]. However, both the computation and the proofs become more involved. A new structure is being considered that generalizes the Voronoi diagram for points, to include the effect of the polygonal obstacles on the proximity information. Note that this is different from other generalizations of the Voronoi diagram that expand its definition to include lines and polygons since we treat the polygons (obstacles) differently from the point (shelters). An application of this model would be a service robot in a factory that may be called upon to report to the nearest of many centers at random times (e.g. for recharging its batteries, when the consumption is not predictable), while it attempts to minimize the time to reach a goal point.

The strategies presented in this paper are heuristics in the sense that that the paths are confined to be piecewise linear, with each linear segment being the line connecting two shelters. When we do not impose this restriction we must seek the optimal solution as a function of the location of the n shelters and the parameter $\frac{\lambda}{v}$. The problem appears to be hard for the dynamic model chosen and proofs of convergence of the different strategies appear to be difficult. On the other hand, restricting ourselves to a subset of the solution space yields analytical results that should serve to give some insight into the overall behavior of the dynamic path planning problem. In particular, the resulting strategies for limiting values of λ (for a fixed v) should serve as a good heuristics for the optimal solution.

7 Conclusion

We have presented a model for a hazardous environment and demonstrated its use in evaluating alternate strategies for navigation using a rule for evading danger by re-routing to the nearest shelter. While the model is simple, it serves to provide insights into the influence of the probabilistic parameters of the dynamic environment on path planning. It may be considered as a step toward treating more general and realistic environments in which all static components (e.g. the obstacles and the shelters) are known while the dynamic components that also influence the path are unknown and can only be detected online.

References

[1] A. Basu. *A Framework for Motion Planning in the Presence of Moving Obstacles.* Technical Report CAR-TR-481, Center for Automation Research, University of Maryland, December 1989.

[2] P. Burlina, D. DeMenthon, and L. S. Davis. *Navigation with Uncertainty: II. Avoiding High Collision Risk Regions.* Technical Report CAR-TR-568, Center for Automation Research, University of Maryland, July 1991.

[3] K. Fujimura and H. Samet. Motion Planning in a Dynamic Domain. In *Proc. IEEE International Conference on Robotics and Automation*, pages 324–330, 1990.

[4] V. J. Lumelsky and A. A. Stepanov. Path Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape. *Algorithmica*, 2:403–430, 1987.

[5] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York,1985.

[6] N. S. V. Rao. Algorithmic Framework for Learned Robot Navigation in Unknown Terrains. *Computer*, pages 37–43, June 1989.

[7] J. Reif and M. Sharir. Motion Planning in the Presence of Moving Obstacles. In *Proc. IEEE Conference on Foundations of Computer Science*, pages 144–154, 1985.

[8] R. Sharma. Locally Efficient Path Planning in an Uncertain, Dynamic Environment using a Probabilistic Model. *IEEE Transactions on Robotics and Automation*, 8(1), pages 105–110, 1992.

[9] R. Sharma and Y. Aloimonos. Dynamic Control of Motion Detection by an Active, Mobile Observer. In *Proc. 1991 IEEE Symposium on Intelligent Control*, August 1991.

[10] R. Sharma, D. M. Mount, and Y. Aloimonos. *Probabilistic analysis of Some Navigation Strategies in a Dynamic Environment.* Technical Report CAR-TR-569, Center for Automation Research, University of Maryland, College Park, July 1991.

[11] C. K. Yap. Algorithmic Motion Planning. In J. T. Schwartz and C. K. Yap, editors, *Algorithmic and Geometric Aspects of Robotics*, pages 95–143, Erlbaum, Hillsdale, NJ, 1987.