Landmark-Based Robot Navigation

Anthony Lazanas
lazanas@flamingo.stanford.edu

Jean-Claude Latombe
latombe@cs.stanford.edu

Robotics Laboratory
Department of Computer Science, Stanford University
Stanford, CA 94305, USA

Abstract

To operate in the real world robots must deal with errors in control and sensing. Achieving goals despite these errors requires complex motion planning and plan monitoring. We present a reduced version of the general problem and a complete planner that solves it in polynomial time. The basic concept underlying this planner is that of a landmark. Within the field of influence of a landmark, robot control and sensing are perfect. Outside any such field control is imperfect and sensing is null. In order to make sure that the above assumptions hold, we may have to specifically engineer the robot workspace. Thus, for the first time, workspace engineering is seen as a means to make planning problems tractable. The planner was implemented and experimental results are presented. An interesting feature of the planner is that it always returns a universal plan in the form of a collection of reaction rules. This plan can be used even when the input problem has no guaranteed solution, or when unexpected events occur during plan execution.

Introduction

To operate in the real world robots must deal with errors in control and sensing. Achieving goals despite these errors requires complex motion planning and plan monitoring (Latombe 1991). This problem has attracted a lot of interest recently, but many of the proposed approaches are based on unclear assumptions and/or are incomplete. The most rigorous approach so far is the LMT preimage backchaining approach (Lozano-Pérez et al. 1984). Several effective planning methods based on this approach have been proposed, but most of them require exponential time in the size of the input problem or its solution (Donald 1988a; Canny 1989), or they are incomplete with respect to the class of problems they attack (Latombe et al. 1991). Motion planning algorithms will not be applicable to real-world problems, as long as they remain exponential or unreliable. Since the general problem seems to be intrinsically hard (Canny & Reif 1987), a promising line of research is to identify a more restricted, but still interesting subclass of problems that can be solved in polynomial time. This subclass can be obtained, for example, by engineering the robot's workspace. *Thus, for the first time, workspace engineering is formally seen as a means to make planning problems tractable.* Of course, workspace engineering has its own cost, and we should be careful not to specialize the class of problems too much.

In this paper we consider a class of planning problems in the context of the navigation of a mobile robot. We assume that *landmarks* are scattered across the robot's two-dimensional workspace. Each landmark is a physical feature of the workspace, or a combination of features, that the robot can sense and identify, if it is located in some appropriate subset of the workspace. This subset is the *field of influence* of the landmark. A landmark may be a natural feature of the workspace (e.g., the corner made by two walls) or an artificial one specifically provided to help robot navigation (e.g., a radio beacon or a magnetic device buried in the ground). Robot control and sensing are assumed to be perfect within the fields of influence of the landmarks; control is imperfect and sensing is null outside any such field. Given an initial region in the workspace, where the robot is known to be, and a goal region, where we would like the robot to go, the planning problem is to generate motion commands whose execution guarantees that the robot will move into the goal and stop there, provided that our assumptions are satisfied.

We propose a planning method based on the LMT approach to solve the above problem. The method iteratively *backchains non-directional preimages* of the goal, until one encloses the set of possible initial positions of the robot. Each non-directional preimage is computed as a set of directional preimages for critical directions of motion. At every iteration, the intersection of the current non-directional preimage with the fields of influence of the landmarks define the intermediate goal from which to backchain. The overall algorithm takes polynomial time in the total number of landmarks. It is complete with respect to the problems it attacks, that is, it produces a *guaranteed* plan (for input control uncertainty bounds), whenever one such plan exists, and returns failure, otherwise. (A guaranteed plan is one whose execution is guaranteed to succeed if the actual errors are within the uncertainty bounds.) The polynomiality and completeness of the algorithm essentially derive from the combination of the two notions of a landmark and a non-directional preimage.

Another interesting aspect of the method is that, whether it returns success or failure, it always constructs a plan in the form of a non-ordered collection of *reaction rules* described as motion commands associated with regions of the workspace from which the goal can be reliably achieved. This is important in two ways. First, if the input problem has no solution, the robot may nevertheless try to enter one of the regions where a rule is available by performing an initial random motion. Second, if an unexpected event occurs at execution time, the robot may attempt to reconnect to the plan in the same way. When the mean duration of a random motion before it enters one of the regions where reaction rules are available is small enough, i.e. when the total area of these regions is large relative to the workspace area, the idea of inserting random motions is a very attractive one.

In the following we present the planning problem we attack, our planning method and its use for navigation, experimental results and possible extensions of the approach. A more detailed presentation of the method, along with several extensions can be found in (Lazanas & Latombe 1992).

Related Work

The planning method we propose is an instance of the LMT preimage backchaining approach introduced in (Lozano-Pérez et al. 1984; Mason 1984; Erdmann 1984). The complexity of the general problem addressed by the LMT approach is shown to be NEXPTIME-hard in three dimensions (Canny & Reif 1987), which strongly suggests that planning can take double exponential time in some measure of the size of the input problem. To our best knowledge no lower-bound time complexity result has been established for the two-dimensional problem, but there are several upper-bound results applying to this case. A rather general planning procedure based on algebraic decision techniques is described in (Canny 1989), which takes double exponential time in the number of steps of the motion plan. A less general algorithm obtained by restricting sensory feedback is given in (Donald 1988a), which is simply exponential in the number of steps. A perhaps more practical, but incomplete algorithm is presented in (Latombe et al. 1991).

Part of the complexity of LMT, and, more generally, of motion planning under uncertainty comes from the interaction between goal reachability and goal recognizability. We not only want the robot to reach the goal despite uncertainty in control; we also want it to recognize goal achievement despite uncertainty in sensing. Erdmann suggested to simplify planning by assuming partial independence between these two notions (Erdmann 1984). This consists of extracting a subset of the goal that can be unambiguously recognized by the sensors independently of the way it has been achieved. This notion is central to the methods described in (Donald 1988a; Latombe et al. 1991). It is also related to the notion of landmarks used in (Levitt et al. 1987), and under different names in (Buckley 1986), (Christiansen et al. 1990) and (Donald & Jennings 1991). See also (Hutchinson 1991).

Our planning algorithm iteratively computes non-directional preimages of the goal. The notion of a non-directional preimage was already present in the original LMT, but its exact computation was first described in (Donald 1988a). Although our algorithm applies to a different setting, it uses several of the ideas introduced in (Donald 1988a; Briggs 1989), namely the fact that when the direction of motion varies continuously, the preimage of a goal remains qualitatively (i.e. topologically) the same, except at critical directions where it changes suddenly. Furthermore, the amount of changes over all critical directions is bounded. These ideas, combined with a strong landmark notion, are the basis of our polynomial-time planning algorithm. A previous instantiation of LMT into a polynomial-time planner is described in (Friedman 1991).

LMT assumes bounded errors and produces guaranteed plans, that is, plans whose success is guaranteed as long as the actual errors during execution stay within these bounds. The concept of a plan that may fail recognizably is introduced in (Donald 1988b). The concept of a probabilistically guaranteed plan (a plan whose probability of success converges toward one when time grows to infinity) is developed in (Erdmann 1989). Both concepts are related to the notion of a universal plan produced by our planner, when we add random motions. The notion of a universal plan was first proposed in (Schoppers 1989).

Planning Problem

The robot is a point moving in a plane called the *workspace*. There are no obstacles in the workspace. The robot can move in either one of two control modes, the *perfect* and the *imperfect* modes.

The perfect control mode is feasible only in some circular areas of the workspace called the *landmark disks*. When the robot is in a landmark disk, it recognizes the landmark without ambiguity, it knows its position exactly, and it has perfect control over its motions. In total, there are ℓ landmark disks scattered across the workspace. Some disks may intersect, creating larger areas through which the robot can move in the perfect control mode.

The imperfect control mode can be used everywhere. In that mode the robot is requested to execute some motion command $(d, \mathcal{L})$, where $d \in S^1$ is a direction in the plane and $\mathcal{L}$ is a subset of landmark disks, called the *termination set* of the command. When it executes the command $(d, \mathcal{L})$, the robot follows a path whose tangent at any point makes an angle with the direction d that is no greater than some angle θ called the *directional uncertainty*. The cone of angle 2θ whose axis points along d is the *control uncertainty cone*. The robot stops as soon as it enters a landmark disk in $\mathcal{L}$.

The robot has no sense of time, which means that the modulus of its velocity is irrelevant to the planning problem.

The robot is known to be anywhere in a specified *initial region* $\mathcal{I}$ that consists of one or several disks. The *goal region* $\mathcal{G}$ is any subset (connected or not) of the workspace whose intersection with the landmark disks is easily computable. The problem is to generate a *motion plan* (i.e., a sequence of motion commands in the perfect and imperfect control modes) that is guaranteed to attain $\mathcal{G}$, if such a plan exists, and to return that no such plan exists otherwise. As we will see, our planner delivers more than that. The current planner assumes that there are no obstacles in the workspace and that influence fields are circular disks. However, both these assumptions can be relaxed, with the computational complexity of the method remaining polynomial. See (Lazanas & Latombe 1992).

Planning Method

Preimage of a Goal

Consider the goal region $\mathcal{G}$. We define the *kernel* of $\mathcal{G}$ as the largest set of landmark disks such that, if the robot is in one of them, it can attain the goal by moving in the perfect control mode only. Let a *landmark area* be any maximal connected subset of landmark disks. The kernel of $\mathcal{G}$ is constructed as the union of all the landmark areas having non-zero intersection with $\mathcal{G}$. We let $K(\mathcal{G})$ denote the kernel of $\mathcal{G}$.

The *preimage* of $\mathcal{G}$, for any given direction d, is the *largest* subset of the workspace such that, if the robot executes a motion command $(d, K(\mathcal{G}))$ in the imperfect control mode, starting anywhere in this subset, then it is guaranteed to reach $K(\mathcal{G})$. (With our hypotheses, there is no more powerful condition to stop the motion than to recognize the entry into $K(\mathcal{G})$.) From the entry point in the kernel, the robot can attain $\mathcal{G}$ in the perfect control mode. We let $P(\mathcal{G}, d)$ denote the preimage of $\mathcal{G}$ for d.

A preimage $P(\mathcal{G}, d)$ consists of one or several connected subsets. Each connected subset is bounded by straight and circular edges. The circular edges are portions of the boundary of the landmark disks in the kernel of the goal. The straight edges are supported by lines parallel to the two sides of the control uncertainty cone and tangent to some landmark disks. All the straight edges in the boundary of a connected subset of the preimage end in circular edges except two that end at the same vertex, called a *spike* of the preimage. Fig. 1 shows an example of a preimage with four connected subsets.

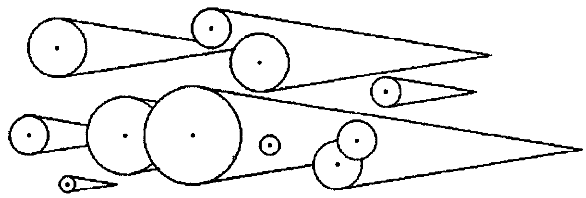

Figure 1: A preimage of a set of disks for $d = \pi$

Each edge on the boundary of a preimage can be labeled with the name of a landmark disk, the straight edges with the name of the disk from which they start, and the arcs with the name of the disk to which they belong. If ℓ is the number of landmark disks, the total number of symbols is also ℓ. Note, that if we describe the boundary of the preimage with a sequence of symbols, no two symbols can alternate more than twice. It is a well-known result that the size of such a sequence is linear in ℓ (Guibas & Stolfi 1989). Therefore the total size of a preimage is $O(\ell)$.

The construction of a preimage is rather straightforward and is not described here. Using a line-sweep algorithm our planner computes a preimage in $O(\ell \log \ell)$ time, after an initial (and performed only once) divide-and-conquer precomputation of the landmark areas that takes $O(\ell \log^2 \ell)$ time.

Preimage Backchaining

If we select d such that $P(\mathcal{G}, d)$ contains the initial region $\mathcal{I}$, then we have succeeded in producing a motion plan to achieve the goal. Indeed, from its initial position in $\mathcal{I}$, the robot can attain the kernel $K(\mathcal{G})$ by executing the motion command $(d, K(\mathcal{G}))$. Then, by switching to the perfect control mode, it can reach the goal without leaving $K(\mathcal{G})$.

However, in general, such a one-step motion plan[1] does not exist. Of course, if $K(\mathcal{G})$ is empty, so is $P(\mathcal{G}, d)$ for any $d \in S^1$, and the planner returns failure. If $K(\mathcal{G})$ is not empty and $P(\mathcal{G}, d)$, for the selected direction d, does not contain $\mathcal{I}$, we treat $P(\mathcal{G}, d)$ as an intermediate goal $\mathcal{G}_1$ and try to build a one-step motion plan to achieve it from $\mathcal{I}$. If we select a new direction, say d_1, such that $P(\mathcal{G}_1, d_1)$ contains $\mathcal{I}$, we have a two-step motion plan to achieve $\mathcal{G}$; otherwise, we consider $P(\mathcal{G}_1, d_1)$ as the new intermediate goal, and so on. The whole process is called *preimage backchaining*.

There is still a major issue to address: How to choose a direction at every iteration of the preimage backchaining process? We could arbitrarily discretize

[1] We measure the number of steps as the number of motion commands in the imperfect control mode.

the continuous set S^1 into a finite set and try all possible combinations of directions in this finite set. But the planner would not be complete, even if we used a very fine discretization (which by the way would also lead to searching a huge graph). We solve this problem by using the notion of a non-directional preimage and slightly modifying the preimage backchaining process.

Non-Directional Preimage

Let us consider the preimage $P(\mathcal{G}, d)$. It turns out that when d moves around the circle S^1, the answers to the questions "Does $P(\mathcal{G}, d)$ include $\mathcal{I}$?" and "What landmark disks does $P(\mathcal{G}, d)$ intersect?" change at a finite number of *critical orientations*. In order to detect these changes we must track the variation of $P(\mathcal{G}, d)$. Fortunately, $P(\mathcal{G}, d)$ varies continuously with the same topology, except at a finite number of other critical orientations where the topology of its boundary changes (e.g., new edges appear or old edges disappear). The open angular slice between any two consecutive critical orientations is called a *regular interval.*

Let $d_1, \ldots, d_p$ denote the critical orientations in counterclockwise order and $I_1, \ldots, I_p$ be the regular intervals between them (the endpoints of I_i are d_i and $d_{i+1 (\mathrm{mod}\, p)}$). For any interval I_i, let d'_i be any orientation in I_i. In order to characterize the preimages of $\mathcal{G}$ and their relations with $\mathcal{I}$ and the landmark disks, it suffices to compute $P(\mathcal{G}, d)$ for all $d \in \{d_1, d'_1, d_2, \ldots, d'_p\}$. The set $P(\mathcal{G})$ of all these preimages is called the *non-directional preimage* of $\mathcal{G}$. Each of the $2p$ preimages in $P(\mathcal{G})$ will now be called a *directional preimage.*

The events that give rise to critical orientations are caused by the motion of the straight edges of the preimage, as their positioning with respect to landmark or initial region disks changes. Fig. 2 shows the subset of events where the topology of the directional preimage changes. The events of Fig. 2 occur when a straight segment enters or exits a disk by becoming tangent to it (a,c,f,h), when segments appear or disappear (b,g), when segments cross the intersection of disks (d,i), and when a spike enters or exits a disk (e,j). The total number of potential events of each type is $O(\ell)$ or $O(\ell^2)$, except for Spike (e) and Hidden Spike (j) events (and similar events not shown in Fig. 2) whose number is $O(\ell^3)$. All events, except two, cause local changes in the preimage which can be computed in constant or logarithmic time. Only events (a) and (h) may cause $O(\ell)$ changes in the topology of the preimage (catastrophic events). After these events we recompute the preimage from scratch. Therefore, the complexity of our algorithm is $O(\ell^3 \log \ell)$.

Non-Directional Preimage Backchaining

Our planner first computes the non-directional preimage $P(\mathcal{G}_0)$, with $\mathcal{G} = \mathcal{G}_0$. (Again, if the kernel $K(\mathcal{G}_0)$ is empty, the planner terminates with failure.) If $P(\mathcal{G}_0)$ contains a directional preimage $P(\mathcal{G}_0, d)$ that includes

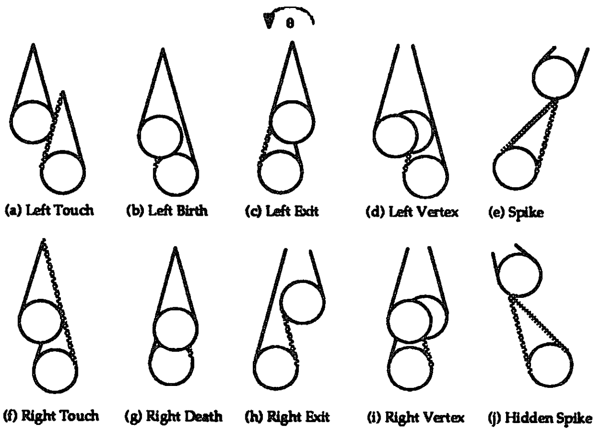

Figure 2: Events responsible for critical orientations

$\mathcal{I}$, then d determines a one-step motion plan to achieve $\mathcal{G}$ as we already described, and the planner terminates with success. Otherwise, the planner considers the union of the directional preimages in $P(\mathcal{G})$ as an intermediate goal $\mathcal{G}_1$.

The kernel $K(\mathcal{G}_1)$ consists of all the landmark areas that have a non-zero intersection with at least one of the directional preimages in $P(\mathcal{G}_0)$. By construction, the set of landmark disks in $K(\mathcal{G}_1)$ is a superset of, or equal to the set of landmark disks in $K(\mathcal{G}_0)$. If the two sets are equal, the planner terminates with failure because it cannot compute a larger preimage than $P(\mathcal{G}_0)$. Instead, if $K(\mathcal{G}_1) \backslash K(\mathcal{G}_0) \neq \emptyset$, the non-directional preimage of $\mathcal{G}_1$ is computed. If it contains a directional preimage that includes $\mathcal{I}$, the planner terminates with success; otherwise it proceeds as above by treating the union of the directional preimages in $P(\mathcal{G}_1)$ as the new intermediate goal $\mathcal{G}_2$, and so on.

During this process, the set of landmark disks in the kernels of the successive goals increases monotonically. At every iteration, either there is a new landmark disk in the kernel, and the planner proceeds, or there is no new disk, and the planner terminates with failure. The planner terminates with success whenever it has constructed a non-directional preimage $P(\mathcal{G}_N)$ that includes $\mathcal{I}$, for some $N \geq 0$. If $s \in O(\ell)$ denotes the total number of landmark areas in the workspace, the number of iterations is bounded by s. Hence, the total time complexity of the planner is $O(sl^3 \log l)$.

In both cases (success and failure), the planner returns the sequence of non-directional preimages it has constructed.

Robot Navigation

Case where the planner returns success

Let $P(\mathcal{G}_N)$ be the non-directional preimage that contains $\mathcal{I}$. The plan actually built by the planner is a set of *reaction rules.* The reaction rules are associated with the initial region $\mathcal{I}$ and the landmark areas. The reaction rule associated with $\mathcal{I}$ prescribes to execute the command $(d_N, \mathcal{L}_N)$ (in the imperfect control mode), where d_N is such that $P(\mathcal{G}_N, d_N)$ is in $P(\mathcal{G}_N)$ and contains $\mathcal{I}$, and $\mathcal{L}_N$ is the set of landmark disks in the kernel $K(\mathcal{G}_N)$. Executing this command guarantees the robot to move from its initial position to a landmark disk in $\mathcal{L}_N$.

The plan also provides one or several reaction rules per landmark area A in $\mathcal{L}_N$. Each such rule is constructed during the backchaining process when a landmark disk L in A intersects a constructed preimage for the first time. Let $\mathcal{G}_k$ be the goal whose preimage was being constructed when this happens. If $k = 0$, then $\mathcal{G}_k$ is the original goal $\mathcal{G}$, and the reaction rule is simply to move to $\mathcal{G}$ in the perfect control mode. If $k > 0$, $\mathcal{G}_k$ is the union of the directional preimages in $P(\mathcal{G}_{k-1})$. Then the reaction rule is defined by three parameters denoted by d_{k-1}, E_{k-1}, and $\mathcal{L}_{k-1}$. The direction d_{k-1} is any direction[2] such that $P(\mathcal{G}_{k-1}, d_{k-1}) \in P(\mathcal{G}_{k-1})$ and $P(\mathcal{G}_{k-1}, d_{k-1})$ intersects L. E_k is equal to $L \cap P(\mathcal{G}_{k-1}, d_{k-1})$ and is called the *exit region* of L. $\mathcal{L}_{k-1}$ is the set of landmark disks in $K(\mathcal{G}_{k-1})$. The reaction rule says: from the point of entrance in A, move in perfect control mode to the exit region E_{k-1} of L; then switch to the imperfect control mode and execute the motion command $(d_{k-1}, \mathcal{L}_{k-1})$. Since there may be several rules attached to a landmark area A (there are at most as many rules as there are disks in A), the navigation system must choose one. It chooses the one that is the closest to the point of entry in A to avoid an unnecessary long motion in the perfect control mode. By construction, no landmark disk in A is in the termination set $\mathcal{L}_{k-1}$ and $\mathcal{L}_0 \subset \mathcal{L}_1 \subset \ldots\mathcal{L}_N$. Hence, the plan cannot loop. The robot is guaranteed to reach $\mathcal{G}$ in a finite number of steps that is at most $N + 1$ (but this number can be smaller).

Case where the planner returns failure

It nevertheless provides a set of reaction rules. Every landmark area in the last goal kernel has one such rule, at least, associated with it. The set of rules can be regarded as a ***universal plan*** in the sense that it provides an appropriate motion command for all recognizable subsets (landmark areas) in the workspace from which a guaranteed plan to the goal exists. This universal plan can be used as follows: Let us assume that the workspace is bounded and the robot can detect that it attains the workspace boundary. Let us also define a third control mode for the robot, the ***random mode***, which consists of executing a Brownian motion with reflection on the workspace boundary. The robot first executes a motion in the random mode until it reaches one of the recognizable subsets. Then it switches to using reaction rules in the universal plan. The probability that a Brownian motion will attain a recognizable subset converges toward 1 when time grows to infinity. The expected duration of the Brownian motion depends on the size of the landmark areas that are equipped with reaction rules. Obviously, this method is attractive only if this area is big enough, so that the expected duration of the motion is small.

Unexpected event

In any of the previous two cases, imagine that an unexpected event occurs at execution time (e.g., the robot slipped and the error in control has been exceptionally large, or a landmark has been accidently "turned off"). If this event leads the robot to miss all the landmark areas it was expected to attain, it will ultimately reach the workspace boundary. Then it may switch to a motion in the random mode, attain a landmark area with a reaction rule, and resume executing the planner's plan.

Experimental Results

We implemented the above planner along with navigation techniques and a robot simulator in C on a DECstation 5000. Below we present some examples of produced plans and their simulated execution. In the figures, white disks are landmark disks that intersect the last non-directional preimage computed by the planner. Grey disks are landmark disks that have not been touched by any non-directional preimage; thus no plan exists for them. There is a single initial region disk marked $\mathcal{I}$ and a single goal region disk marked $\mathcal{G}$. Every time the robot enters a new landmark area with white disks, it executes an appropriate reaction rule, namely it moves into an exit region using the perfect control mode, and from there it executes the specified motion command in the imperfect control mode, until it reaches one of the landmark disks in the termination set of the rule. Each reaction rule is represented by the commanded direction of motion. The outline of exit regions is also shown, unless the exit region covers the entire disk. The termination sets of the rules are not represented. The directional uncertainty θ (see Section 3) is measured in radians.

We ran the algorithm on an example with 51 landmarks, one initial region disk and one goal disk. With $\theta = 0.1$, a plan was returned in less than two seconds, after two backchaining steps. As we let uncertainty grow, the planner returned more and more complicated plans, leading the robot through many landmark disks in order to reduce uncertainty. We ran the algorithm on this same example with $\theta = 0.15, 0.2, 0.25, 0.29, 0.3$, and 0.35, obtaining successful plans after 3, 4, 5, 6, 7, and 8 iterations, respectively. Each example was completed in less than one minute. Fig. 3 displays the case $\theta = 0.15$. It took 3 iterations of the planner before the initial region got included in a preimage. In the process

[2] Actually, our implemented planner selects the median direction in the largest interval of directions that allow the robot to attain $\mathcal{G}_k$ from L. The intuition for this choice is that it is more robust to unmodelled control errors.

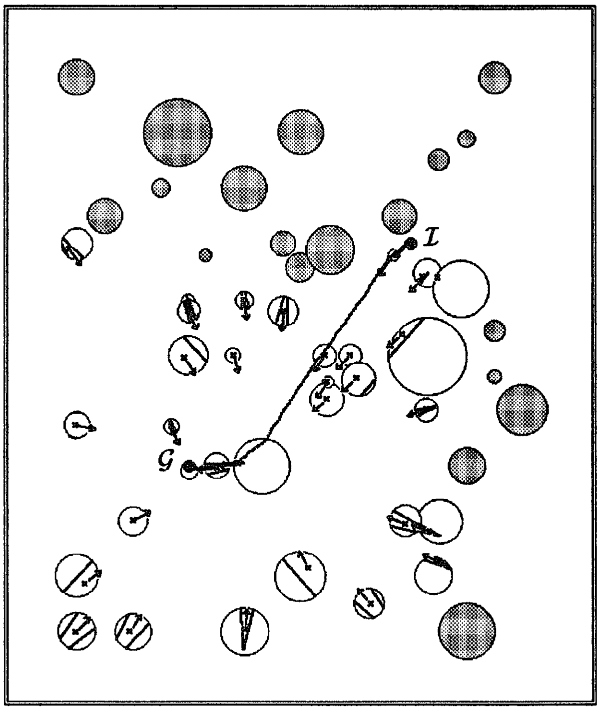

Figure 3: Successful planning and execution ($\theta = 0.15$)

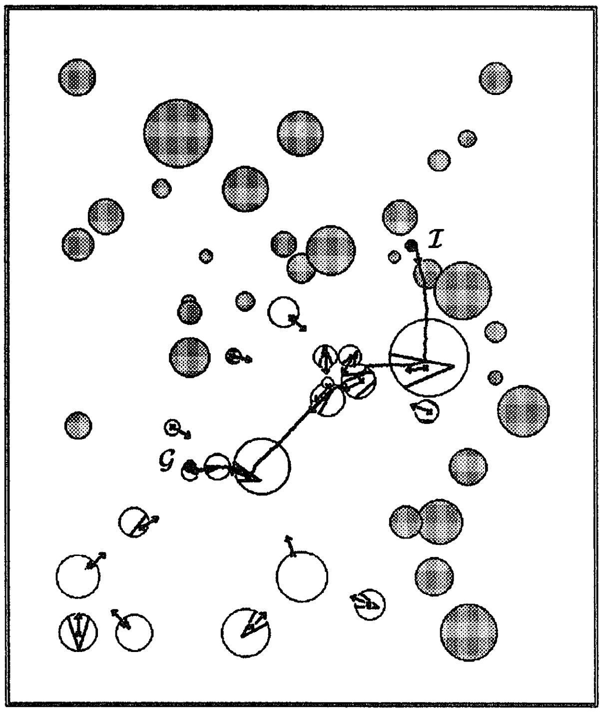

Figure 4: Successful planning and execution ($\theta = 0.3$)

the planner discovered guaranteed plans for many more landmark disks. A simulated plan (with three steps as well) is shown in the figure. Fig. 4 displays the case $\theta = 0.3$. The resulting plan is clearly more complicated, as is evident from its simulated execution. Reaction rules are markedly different, as they now point towards neighboring rather than remote disks.

Fig. 5 illustrates the use of the output of the planner in a simpler workspace (6 landmarks), when no guaranteed plan exists. With θ set to 0.35, even the preimage of all six landmark disks in this example fails to include the initial region disk. The universal plan produced by the planner is represented as a reaction rule attached to each landmark disk not intersecting the goal. The robot first executes a Brownian motion and it is lucky enough to enter the upper-left landmark disk in a relatively short amount of time. From there it reaches the goal safely.

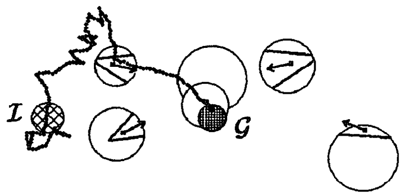

Figure 5: Success using a Brownian motion

Conclusion

We have described a complete polynomial planning algorithm for landmark-based robot navigation. The algorithm addresses a rather simple class of problems, but it is fast and provides robust plans. In addition, the class of problems is strongly related to many real-life problems with mobile robots operating in open areas.

The algorithm can be extended to deal with more complicated problems. Two relatively straightforward extensions are the use of more general landmarks with polygonal and/or circular fields of influence and the introduction of polygonal/circular obstacles in the workspace (Lazanas & Latombe 1992). Dealing with polygonal rather than circular landmark fields is actually easier. We just need to adapt our critical events to the polygonal structure of the areas. Also, obstacles can be handled by the preimage computation algorithms, by shading out the subsets of the preimage containing the points that have a chance to lead the robot to collide with an obstacle. The adaptive selection of a value for θ, as well as precomputing plans for faster planning in a stable workspace are also discussed in (Lazanas & Latombe 1992).

So far, most algorithms to plan motion strategies under uncertainty were either exponential in the size of the input problem or its solution, or incomplete, or both. Such algorithms may be interesting from a theoretical point of view, but their computational complexity or lack of reliability prevent them from being applied to real-world problems. Our work shows that it is possible to identify a restricted, but still realistic, subclass of planning problems that can be solved in polynomial time. This subclass is obtained through assumptions whose satisfaction may require prior engineering of the workspace and/or the robot. In our case, this implies the creation of adequate landmarks, either by taking advantage of the natural features of the workspace, or by introducing artificial beacons. We call this type of simplification *engineering the workspace for planning tractability.*

Engineering the workspace has its own cost and we would like to minimize it. This will lead us to investigate further extensions of our planner. For example, the task of creating landmarks would be considerably simplified if the planner could deal with small uncertainty in sensing and control within the landmark fields of influence, and/or in the location and the size of these fields, perhaps allowing influence fields with soft, rather than sharp boundaries. Our goal will be to find a more general class of problems than the one solved by the current planner, but requiring less workspace/robot engineering and still solvable in polynomial time. A related issue will be to investigate the following "inverse" problem: Given our planning method and the description of a family of tasks (e.g., the set of all possible initial and goal regions), how to minimally engineer the workspace, that is, what is the minimal number of landmarks that we should place and where should we place them so that every possible problem admits a guaranteed solution.

References

Briggs, A.J. 1989. An efficient Algorithm for One-Step Planar Compliant Motion Planning with Uncertainty. In Proceedings of the Fifth ACM Symposium on Comp. Geometry, 187-196. Saarbrücken, Germany.

Buckley, S.J. 1986. Planning and teaching Compliant Motion Strategies. Ph.D. diss., Dept. of Electrical Engineering and Computer Science, MIT.

Canny, J.F., and Reif, J. 1987. New Lower Bound Techniques for Robot Motion Planning Problems. In Proceedings of the 28th IEEE Symposium on Foundations of Computer Science, 49-60. Los Angeles, CA.

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

Canny, J.F.1989. On Computability of Fine Motion Plans. In Proceedings of the IEEE International Conference on Robotics and Automation, 177-182. Scottsdale, AZ.

Christiansen, A.; Mason, M.; and Mitchell, T.M. 1990. Learning Reliable Manipulation Strategies without Initial Physical Models. In Proceedings of the IEEE International Conference on Robotics and Automation, 1224-1230. Cincinnati, OH.

Donald, B.R. 1988. The Complexity of Planar Compliant Motion Planning Under Uncertainty. In Proceedings of the Fourth ACM Symposium on Computational Geometry, 309-318. Champaign, IL.

Donald, B.R. 1988. A Geometric Approach to Error Detection and Recovery for Robot Motion Planning with Uncertainty. *Artificial Intelligence Journal* 37(1-3):223-271.

Donald, B.R., and Jennings, J. 1991. Sensor Interpretation and Task-Directed Planning Using Perceptual Equivalence Classes. In Proceedings of the IEEE International Conference on Robotics and Automation, 190-197. Sacramento, CA.

Erdmann, M. 1984. On Motion Planning with Uncertainty, Technical Report, 810. AI Laboratory, MIT.

Erdmann, M. 1989, On Probabilistic Strategies for Robot Tasks. Ph.D. diss., Dept. of Electrical Engineering and Computer Science, MIT.

Friedman, J. 1991. Computational aspects of Compliant Motion Planning. Ph.D. diss., Dept. of Computer Science, Stanford Univ.

Guibas, L.J., and Stolfi, J. 1989, Ruler, Compass, and Computer: The Design and Analysis of Geometric Algorithms, Technical Report No. 37. Systems Research Center, Digital, Palo Alto.

Hutchinson, S. 1991. Exploiting Visual Constraints in Robot Motion Planning, In Proceedings of the IEEE International Conference of Robotics and Automation, 1722-1727. Sacramento, CA.

Latombe, J.C. 1991. *Robot Motion Planning.* Boston, MA: Kluwer Academic Publishers.

Latombe, J.C.; Lazanas, A.; and Shekhar, S. 1991. Robot Motion Planning with Uncertainty in Control and Sensing, *Artificial Intelligence Journal* 52(1):1-47.

Lazanas, A., and Latombe, J.C. 1992. Landmark-Based Robot Navigation, Technical Report, Dept. of Computer Science, Stanford Univ. Forthcoming.

Levitt, T.S.; Lawton, D.T.; Chelberg, D.M.; and Nelson, P.C. 1987. Qualitative Navigation. In Proceedings of the Image Understanding Workshop, 447-465. Los Angeles, CA.

Lozano-Pérez, T.; Mason, M.T.; and Taylor, R.H. 1984. Automatic Synthesis of Fine-Motion Strategies for Robots. *International Journal of Robotics Research* 3(1):3-24.

Mason, M.T. 1984. Automatic Planning of Fine Motions: Correctness and Completeness. In Proceedings of the IEEE International Conference on Robotics and Automation, 492-503. Atlanta, GA.

Schoppers, M.J. 1989. Representation and Automatic Synthesis of Reaction Plans. Ph.D. diss., Dept. of Computer Science, University of Illinois at Urbana-Champaign.