1

Motion Planning in the Presence of Moving Obstacles

John Reif

Department of Computer Science

Duke University, Durham, NC 27706, USA

Micha Sharir

School of Mathematical Sciences

Tel-Aviv University,

Tel Aviv 69978, Israel

and

Courant Institute of Mathematical Sciences,

New York University,

New York, NY 10012, USA

*Work on this paper by the first author was supported in part by the Office of Naval Research Contract N00014–80–C–0647, Air Force Contract AFOSR–87–0386, DARPA/ISTO Contract N00014–

Abstract

This paper investigates the computational complexity of planning the motion of a body B in 2–D or 3–D space, so as to avoid collision with moving obstacles of known, easily computed, trajectories. Dynamic movement problems are of fundamental importance to robotics, but their computational complexity has not previously been investigated.

We provide evidence that the 3-D dynamic movement problem is intractable even if B has only a constant number of degrees of freedom of movement. In particular, we prove the problem is $PSPACE$-hard if B is given a velocity modulus bound on its movements and is NP hard even if B has no velocity modulus bound, where in both cases B has 6 degrees of freedom. To prove these results we use a unique method of simulation of a Turing machine which uses time to encode configurations (whereas previous lower bound proofs in robotic motion planning used the system position to encode configurations and so required unbounded number of degrees of freedom).

We also investigate a natural class of dynamic problems which we call *asteroid avoidance problems*: B, the object we wish to move, is a convex polyhedron

88–K–0458, DARPA/ARO Contract DAAL03–88–K–0195, and NASA/CESDIS Contract 550–63 NAS 5–30428 URSA. The work was completed while the first author was on sabbatical at the Laboratory for Computer Science, MIT, Cambridge, MA. Work by the second author was supported in part by a grant from the U.S.-Israeli Binational Science Foundation, by the Office of Naval Research Contracts N00014-82-K-0381, N00014-87-K-0129, and N00014-90-J-1284, and by National Science Foundation Grants DCR-83-20085 and CCR-89-01484. A preliminary version of the paper has appeared in the Proceedings of the 26th IEEE Symposium on Foundations of Computer Science, 1985.

which is free to move by translation with bounded velocity modulus, and the polyhedral obstacles have known translational trajectories but cannot rotate. This problem has many applications to robot, automobile, and aircraft collision avoidance. Our main positive results are polynomial time algorithms for the 2–D asteroid avoidance problem where B is a moving polygon and we assume a constant number of obstacles, as well as single exponential time or polynomial space algorithms for the 3–D asteroid avoidance problem where B is a convex polyhedron and there are arbitrarily many obstacles. Our techniques for solving these asteroid avoidance problems use "normal path" arguments which are an interesting generalization of techniques previously used to solve static shortest path problems.

We also give some additional positive results for various other dynamic movers problems, and in particular give polynomial time algorithms for the case in which B has no velocity bounds and the movements of obstacles are algebraic in space–time.

1 Introduction

1.1 Static Movers Problems

The *static movers problem* is to plan a collision–free motion of a body B in 2–D or 3–D space avoiding a set of obstacles stationary in space. For example, B may be a sofa which we wish to move through a room crowded with furniture, or B may be an articulated robot arm which we wish to move in a fixed workspace.

Reif [14] first showed that a generalized 3–D static movers problem is $PSPACE$-hard, where B consists of n linked polyhedra. Hopcroft, Joseph and Whitesides [6] and Hopcroft, Schwartz and Sharir [8] later proved $PSPACE$-lower bounds for 2–D static movers problems. If the number of degrees of freedom of motion is kept constant, then the problem has polynomial–time solutions, provided that the geometric constraints on the motion can be stated algebraically (Schwartz and Sharir [19]). More efficient polynomial time algorithms for various specific cases of static movers problems are given by Lozano-Perez and Wesley [11], Reif [14], Schwartz and Sharir [18, 20, 21], Hopcroft, Joseph and Whitesides [7], Ó'Dúnlaing, Sharir and Yap [12], Ó'Dúnlaing and Yap [13]. Some of these results are compiled in a recent book [9]. See also a more recent survey (Sharir [22]) that reviews these and later works on the topic.

1.2 Dynamic Movers Problems

In this paper, we consider the problem of planning a collision–free motion of a body B which is free to move within some 2–D or 3–D space S, containing several obstacles which move in S along known trajectories. We require that the obstacle trajectories be easily computable functions of time, and not be at all dependent on any movement of B. Some applications are:

(1) Robotic Collision Avoidance. B might be a robot arm which must be moved through a workspace such as an assembly line in which various machine parts make predictable movements.

(2) Automobile Collision Avoidance. B is an automobile with an automatic steering system which must avoid collision with other automobiles with known trajectories on a highway.

(3) Aircraft Collision Avoidance. B is an aircraft which we wish to automatic–pilot through an airspace containing a number of aircraft and other obstacles with known flight paths.

(4) Spacecraft Navigation. B might be a spacecraft which we wish to automatically maneuver among a field of moving obstacles, such as asteroids.

Although the dynamic movers problem is fundamental to robotics, there are only very few works which have considered the computational complexity of such problems, and they all appeared after the original version of this paper [15].

We can formally define a *dynamic movers* problem as follows. Let B be an arbitrary fixed system of moving bodies (each of which can translate and rotate, and some of which may be hinged), having overall d degrees of freedom. B is allowed to move within a space S which contains a collection of obstacles moving in an arbitrary (but known) manner. To cope with the time-varying environment we represent the time as an additional parameter of the configuration of B. More precisely, we define the *free configuration space* FP of B to consist of all pairs $[X, t] \in E^{(d+1)}$, where $X \in E^d$ represents a configuration of B, and such that if at time t the system B is at configuration X then B does not meet any obstacle at that time (here E^d denotes the d-dimensional Euclidean space). In this representation of FP a continuous motion of

B is represented by a continuous arc $[x_t, t] = p(t)$, which is *monotone* in t. Note that the slope of this arc (relative to the t-axis) represents the "velocity" (i.e., the rate of change of the parameters of the motion) of B. If we impose no restrictions on this velocity, any such t-monotone path corresponds to a possible motion of B. However, the dynamic version of the problem is usually further complicated by imposing certain constraints on the allowed motions of B. One such constraint is that the velocity modulus of B cannot exceed a given bound (the modulus is the Euclidean norm of the velocity vector); we refer to this as a 'bounded velocity modulus' constraint. Such a constraint of a "uniform" bound on the velocity of B is particularly appropriate if B is a single rigid body free only to translate; most of the versions of the problem (e.g., the asteroid avoidance problem) studied in this paper will be of this kind.

Using the above terminology, the problem that we wish to solve is: Given an initial free configuration $[X_0, 0]$ and a final free configuration $[X_1, T]$, plan a continuous motion of B (if one exists) between these configurations which will avoid collision with the obstacles, or else report that no such motion is possible. (Note that we also specify the time T at which we want to be at the final configuration X_1; as will be seen below, a variant of our techniques can be used to obtain minimal time movement of B.) In other words, we wish to find a monotone path in FP between the two configurations $[X_0, 0]$ and $[X_1, T]$, where the path satisfies the velocity modulus bound constraint (if imposed).

To avoid technical difficulties in the analysis in this paper, we relax the condition that the movement of B be collision-free, so as to allow B also to make contact with obstacles during its motion, but still forbid B from intersecting the interior of any obstacle. Such movement is usually called 'semi-free', but we will continue to refer also to this kind of movement as collision-free. We will make one exception to this convention in Section 4.2, where we will not allow B to make any contact with the obstacles.

The goal of this paper is to systematically investigate the complexity of various fundamental classes of dynamic movement planning problems.

1.3 Summary of Our Results

In summary, the *main results* of this paper are:

1. *PSPACE lower bounds* of 3-D dynamic movement planning of a single disc with bounded velocity and rotating obstacles.

2. *Decision algorithms* for 1-D, 2-D or 3-D dynamic movement planning of a translating polyhedron with bounded velocity and purely translating obstacles.

We also have additional results for some dynamic movement planning problems with unbounded velocity.

1.4 Our Lower Bound Results for Rotating Obstacles

In the case the obstacles rotate, they may generate non-algebraic trajectories in space-time which appear to make movement planning intractable. Our *main negative result*, given in Section 2, is a proof that 3–D dynamic movement planning with rotating obstacles is $PSPACE$-hard, even in the case the object to be moved is a disc with bounded velocity. (We also have a related NP-hardness result, described below, in the case B has no velocity bounds.)

Remark. All previously known lower bound results for movers problems utilize the *position* of B for encoding n bits, and thus require that B have $\Omega(n)$ degrees of freedom. We use substantially different techniques for our lower bound results. In particular, we use *time* to encode the configuration of a Turing machine that we wish to simulate (therefore we call our construction a "time–machine"). In our lower bound construction it suffices that B have only $O(1)$ degrees of freedom. (In contrast, static movement planning is polynomial time decidable in case B has only $O(1)$ degrees of freedom.) The key to our $PSPACE$-hardness proof is a "delay box" construction, which by use of rotating obstacles generates an exponential number of disconnected components in the free configuration space.

1.5 Efficient Algorithms for Asteroid Avoidance Problems

In Section 3 we investigate an interesting class of tractable dynamic movement problems where the obstacles do not rotate. An *asteroid avoidance problem* is the dynamic movement problem where each of the obstacles is a polyhedron with a fixed (possibly distinct) translational velocity, and B is a convex polyhedron which may make arbitrary translational movements but with a bounded velocity modulus. Neither B nor the obstacles may rotate. (This problem is named after the well known ASTEROID video game, where a spacecraft of limited velocity modulus must be maneuvered to avoid swiftly moving asteroids.) The problem is efficiently solved in the 1–D case by line scanning techniques but is quite difficult in the 2–D and 3–D cases.

The assumptions of the asteroid avoidance problem are applicable in many of the above mentioned practical problems, such as robot, automobile, airplane and spacecraft collision avoidance problems, where both B and the obstacles are approximated by convex polyhedra.

The *major positive results* of this paper are a polynomial time algorithm for the 2–D asteroid avoidance problem where the object B is a polygon and we assume a constant number of convex obstacles, as well as $2^{n^{O(1)}}$ time or polynomial space decision algorithms for the 3–D asteroid avoidance problem where B is a convex polyhedron and there are arbitrarily many obstacles.

The methods we develop such as "normal movement" decomposition of paths are an interesting extension of the much simpler normal path techniques previously used by shortest path algorithms in the static case.

These techniques are also extended to yield algorithms for the *minimum-time* asteroid avoidance problem, in which we wish to reach a desired final position in the shortest possible time.

We note that since the original appearance of this paper in [15] several other works addressed dynamic motion planning problems. Among those we mention the work by Sutner and Maass [23] where results similar to ours have been independently obtained. Sutner and Maass have studied the variant where minimum-time movement is being sought; this variant is also implicit in the earlier version of our paper [15]. See also Canny and Reif [3] for related results.

1.6 Dynamic Movers Problems with no Velocity Bound on B

In the final Section 4 of this paper, we consider the complexity of dynamic movement planning in the case where B, the object to be moved, has no velocity modulus bounds. We first show that the 3–D dynamic movement problem for a cylinder B with unrestricted velocity is NP-hard.

We then consider algorithms for dynamic movement planning in the case in which no velocity bounds are imposed on the motion of B, and the geometric constraints on the possible positions of B can be specified by algebraic equalities and inequalities (in the parameters describing the possible degrees of freedom of B and in *time*). We show that this problem is solvable in polynomial time for any *fixed* moving system B (which may consist of several independent hinged translating and rotating bodies in 2–D or 3–D).

2 A Time Machine Simulation of PSPACE

We show here that

Theorem 2.1 *Dynamic movement planning in the case of bounded velocity is PSPACE-hard, even in the case where the body B to be moved is a disc moving in 3–space.*

Proof. Let M be a deterministic Turing machine with space bound $S(n) = n^{O(1)}$. We can assume M has tape alphabet $\{0, 1\}$, state set $Q = \{0, \ldots, |Q|-1\}$ with initial state 0 and accepting state 1. A *configuration* of M consists of a tuple $C = (u, q, h)$ where $u \in \{0,1\}^{S(n)}$ is the current tape contents, $q \in Q$ is the current state, and $h \in \{0, \ldots, S(n) - 1\}$ is the position of the tape head. Let $next(C)$ be the configuration immediately succeeding C. Given input string $w \in \{0,1\}^n$ considered to be a binary number, the initial configuration is $C_0 = (w0^{S(n)-n}, 0, 0)$. We can assume $(0^{S(n)}, 1, 0)$ is the accepting configuration. We can also assume that if M accepts, then it does so in exactly $T = 2^{cS(n)}$ steps for some constant $c > 0$. Thus M accepts iff C_T is accepting, where $C_0, C_1, \ldots, C_T$ is the sequence of configurations of M satisfying $C_i = next(C_{i-1})$ for $i = 1, \ldots, T$.

To simulate the computation of M on input w, we will construct a 3–D instance of the dynamic movers problem where the body B to be moved is a disc of radius 1, and where we bound the velocity modulus of B by $v = 100|Q|S(n)$. The basic idea of our simulation is to use *time* to encode the current configuration of M. The dynamic movement problem we construct will be specified giving the exact size, velocity and initial position of each obstacle as well as the initial and final position (and maximum velocity modulus) of the object B to be moved. This specification will use a polynomial number of bits (specified by a binary encoding) and will be constuctible using an $O(\log n)$-space bounded deterministic Turing Machine.

Let

$$N = S(n) + \lceil \log |Q| \rceil + \lceil \log S(n) \rceil ,$$

so that 2^N is at least $2^{S(n)}|Q|S(n)$. Note that since $S(n)$ is polynomial, N is also polynomial in n. We shall encode each configuration $C = (u, q, h)$ as an N bit binary number

$$\#(C) = u + q2^{S(n)} + h2^{S(n)+\lceil \log |Q| \rceil}.$$

Note that $\#(C)$ is at most

$$2^{S(n)} - 1 + (|Q| - 1)2^{S(n)} + (S(n) - 1)2^{S(n)+\lceil \log |Q| \rceil}$$

which is at most $2^N - 1$. A *surface configuration* of M is a triple $< u_h, q, h >$ where $u_h \in \{0, 1\}$ is the value of the tape cell currently scanned, q is the current state and h is the head position. For each $q \in Q$, $h \in \{0, \ldots, S(n) - 1\}$, and $u_h \in \{0, 1\}$, we associate a distinguished position $P_{<u_h,q,h>}$ of B in 3-dimensional space corresponding to surface configuration $< u_h, q, h >$ of M. Note that since $S(n)$ and N are polynomial in n, there are only a polynomial number of surface configurations.

We will fix a distinguished initial position, HOME-POSITION, of B in 3-dimensional space (it has no time component). B is located at HOME-POSITION at the initial time $t_0 = w$. The dynamic movers problem will be to move B so that it is at position HOME-POSITION also at time $t_T = 2^{S(n)} + T2^N$. We will construct a collection of moving obstacles which will force B to move to position HOME-POSITION exactly at each time $t_i \geq w$ such that $\lfloor t_i \rfloor = \#C_i + i2^N$, and $t_i < \lfloor t_i \rfloor + \frac{2}{v}$. Thus we use $2N$ bits of t_i for the encoding, in particular the lower N bits of t_i encode the configuration C_i and the higher bits encode the step number. (Note that t_0 encodes the initial configuration, at step 0, and t_T encodes the final configuration at step T.) Since N is polynomial in n, the number of bits used in the encoding of t_0 and t_T is polynomial.

To simulate M, we need two kinds of devices, one to test that M is at a particular surface configuration, and the other to simulate one step of M at a specific surface configuration. The first kind of device is constructed as follows. Fix some ϵ between 0 and 0.5. The entries and exits of the devices to be described below will be connected to the rest of the construction by the use of cylindrical tubes (which will be called "connecting tubes") of diameter $1+\epsilon$. We now describe a 'test box' which is a device to test the value a given bit b_j in position j of t_i. The test box will be a cylinder of diameter 3, and will have a distinguished entry slot and two exit slots: $exit_0$ and $exit_1$ each of width $1+\epsilon$. Let Δ_j be the time required by B to reach the entry slot of the test box from HOME-POSITION (this can be easily determined from the construction given below of the tree of test boxes). We design a test box so that if B is placed at the entrance slot at time $t_i + \Delta_j$, then within time delay $6/v$, B is forced to move through $exit_{b_j}$. We now give the specification of this test box, which will be polynomial in n. First, we will force B from the connecting tube into and through the entry slot of the test box by use of a semidisk rotating once every $2/v$ time units. We will force B to exit the test box by use of a semidisk which sweeps out the cylindrical test box once every $1/v$ time units. By use of an additional semidisk rotating once every 2^j time units, we will open and close the exits so that $exit_1$ is open iff $exit_0$ is closed at time $\lfloor t_i + \Delta_j \rfloor$. Thus in delay at most $6/v$ the test box forces B to depart through $exit_{b_j}$.

Next we will describe a construction, which we call the test tree, which will force B to be moved from HOME-POSITION at time t_i to distinguished position $P_{<u_h,q,h>}$ at time between $\lfloor t_i \rfloor + 1$ and $\lfloor t_i \rfloor + 1 + 2/v$, where $< u_h, q, h >$ is the surface configuration associated with the configuration which is encoded as above by the $2N$ low order bits of $\lfloor t_i \rfloor$. To do this, we construct a balanced tree whose nodes are test boxes. From HOME-POSITION there is a connecting tube to the entry slot of the root. The exit slots of the leaves are connected by a connecting tube to distinguished positions of the form $P_{<u_h,q,h>}$, where $< u_h, q, h >$ is a surface configuration. All such distinguished positions will be arranged in a straight line at distance 10 units between each other. The exit slot of each test box in the interior of this test tree are be connected via a connecting tube to the entry slot of each of their interior children. The j-th level of the test tree is be used to test the j-th bit of the current surface configuration. Since the number of surface configurations is at most $2|Q|S(n)$, the depth of this tree will be $\log(\#\text{surface configurations}) \leq \lceil \log(2|Q|S(n)) \rceil$ bits.

The interior of each such connecting tube is swept by a sequence of semidisks rotating once every $2/v$ time units, so as to force B through the connecting tube from the previous exit to the next entry in time upper bounded by $2/v$ times the length of the connecting tube. Since the total length of the connecting tubes on any path from the root to a leaf is at most $20|Q|S(n)$, the delay through them is at most $40|Q|S(n)/v$, and furthermore the delay through each of the test boxes of the nodes on this path is at most $6/v$. Thus the total delay from HOME-POSITION to a leaf is at most $50|Q|S(n)/v \leq 1/2$, since $v = 100|Q|S(n)$, and the delay is clearly at least

$4/v$.

We now modify the above construction to make this total delay to at least 1 and at most $1 + 2/v$, by adding at the end of the connecting tube leading to each leaf a pair of semidisks, each rotating once per unit time step. The first semidisk will, for any number m, allow B to exit only at times between m and $m + 2/v$ and the second will sweep out the this area during the time interval between $m + 2/v$ and $m + 4/v$, thus forcing B to exit only at times between m and $m + 2/v$. The test tree thus has the property that if B is at HOME-POSITION at time t_i, then at time at least $\lfloor t_i \rfloor + 1$ and at most $\lfloor t_i \rfloor + 1 + 2/v$, B is forced to the distinguished position $P_{<u_h,q,h>}$, where $< u_h, q, h >$ is the current surface configuration encoded by t_i. The test tree has only a polynomial number of nodes, and each node and edge of the test tree requires only a polynomial size specification; thus the test tree requires only a specification of size polynomial in n.

Hence (by using a balanced tree of such test boxes plus some additional sweeping semidiscs), we can force B to be moved from HOME-POSITION to arrive in distinguished position $P_{<u_h,q,h>}$ in time at least $\lfloor t_i \rfloor + 1$ and less than $\lfloor t_i \rfloor + 1 + \frac{2}{v}$. Let $C_{i+1} = next(C_i) = (u', q', h')$ be the configuration of M immediately following C_i. Since $\#C_{i+1} - \#C_i$ depends only on $< u_h, q, h >$, there is a function $g(u_h, q, h)$ such that $\#C_{i+1} = \#C_i + g(u_h, q, h)$ and $|g(u_h, q, h)| < 2^N$. Hence we will require an additional gadget, a *delay box*, to be described below, to force B to move from position $P_{<u_h,q,h>}$ back again to position HOME-POSITION at time t_{i+1} such that

$$\lfloor t_{i+1} \rfloor = \#C_{i+1} + (i+1)2^N = \lfloor t_i \rfloor + g(u_h, q, h) + 2^N,$$

and $t_{i+1} \leq \lfloor t_{i+1} \rfloor + \frac{2}{v}$. The total time delay for this move through the delay box should be (in integral terms) $\Delta = g(u_h, q, h) + 2^N - 2$, to which we add about 1 time unit consumed by the move through the test tree, and another 1 time unit to force B back to HOME-POSITION from the exit of the delay box — see below.

Thus our key remaining construction still required is a "delay Δ box" (where Δ is an integer less than 2^{2N}). If B enters the delay box at any time $t \geq 0$ such that $t < \lfloor t \rfloor + \frac{2}{v}$, then B must be made to exit the delay box at a time at least $\lfloor t \rfloor + \Delta$, and at most $\lfloor t \rfloor + \Delta + \frac{2}{v}$. Note that we can assume Δ is greater than a constant, say 10, or else the construction is trivial. (Our construction is not trivial, however, in the general case where Δ is exponential in N since it is based on an explicit construction of an exponential number of disconnected components in the free configuration space, using only a small number of moving (essentially rotating) obstacles having polynomially describable velocities.)

Our delay box consists of a fixed torus-shaped obstacle, plus some additional moving obstacles (see Figure 1). We can precisely define this torus as the surface generated by the revolution of an (imaginary) circle of radius 3 around the x axis, so that its center is always at distance $\frac{\Delta v}{2\pi}$ from the x axis, and so that the circle is always coplanar with the x-axis. Let Θ be the angular position of a point with respect to rotation around the x-axis.

[Figure 1 goes here.]

The torus will have open *entrance* and *exit* slots at $\Theta = 0$ and $\Theta = \pi$, respectively, just sufficiently wide for entrance and exit of disc B from the torus. The idea of our delay box construction will be to create various disconnected "free spaces" within the torus in which B must be located. These free spaces will be constructed so that they move within the torus π radians of Θ (i.e., make 1/2 a revolution) in Δ time units. Once B enters the torus via the entrance slot, our construction will force B to be located in exactly one such free space, and revolve with it around the torus until B leaves the interior of the torus at the exit slot after the required delay of Δ time units.

We now show precisely how to create these moving "free spaces". A moving obstacle D moves through the interior of the torus with angular velocity (with respect to Θ) of $16 + \frac{1}{2\Delta}$ revolutions per time unit. D consists of three discs D_0, D_1, D_2 placed face to face so that their centers are nearly in contact and so that they are each coplanar with the x-axis. Discs D_0, D_1, D_2 are of radius almost 3. D_0 has a 1/4 section removed, D_1 has a 3/4 section removed, and D_2 has a 1/2 section removed. D_1 and D_2 each rotate around their center, but D_0 does not. Let ψ_i be the angular displacement of D_i as it rotates around its center, for $i = 1, 2$. We set the angular velocity of D_1 with respect to ψ_1 to be the same as the angular velocity of D_1 with respect to Θ. We set the angular velocity of D_2 with respect to ψ_2 to be 32Δ revolutions per time unit (see Figure 2).

[Figure 2 goes here.]

We assume that when D has angular displacement $\Theta = \frac{3\pi}{2}$, D_1 is positioned so that the remaining solid quarter section of D_1 completely overlaps the removed quarter section of D_0. This creates an immobile "dead space" at $\Theta = \frac{3\pi}{2}$ every roughly $\frac{1}{16}$ time units, which B cannot cross (because its velocity is too small), and will force B (if it is to avoid collision) to exit the torus via the exit slot at $\Theta = \pi$. However, while D has angular displacement Θ, for $0 \leq \Theta \leq \pi$, the removed $3/4$ section of D_1 completely overlaps the removed quarter section of D_0 which therefore remains completely unobscured (see Figure 3).

[Figure 3 goes here.]

Let a "free space" consist of the space-time region created during roughly every $1/2$ revolution of D_2 around its center, when the removed quarter section of D_0 and the removed half-section of D_2 sufficiently overlap to accommodate B between them. By construction, B can be located in this free space without ever contacting an obstacle. By contrast, a "dead space" is the space-time region where the removed sections of disks D_0, D_2 do not sufficiently overlap to accommodate B between them; therefore B cannot be located in this dead space longer than a revolution of D_1. Since D_2 rotates around its center at most 2Δ times every time interval in which D rotates through the torus, at most 2Δ such free spaces are created during one revolution of D around the torus (see Figure 4).

[Figure 4 goes here.]

Finally, we claim that B cannot move between any two distinct free spaces while in the interior of the torus. If this was possible, then B could move across a dead space without colliding with D. But D makes a revolution of Θ at least every $\frac{1}{16}$ time units. In this time, B (which has maximum velocity v) can move at most distance $\frac{v}{16}$, which is less than the minimum distance $\frac{1}{4\Delta} \cdot \frac{\Delta v}{2\pi} \cdot 2\pi = \frac{v}{4}$ between any two free spaces, a contradiction.

Since D makes an integral number plus $\frac{1}{2\Delta}$ revolutions of Θ every time unit, each free space moves $\frac{1}{2\Delta}$ revolutions of Θ every time unit, and thus each free space moves $1/2$ a revolution of Θ in Δ time units as required in our construction. Moreover we have chosen the size of the torus and $\Delta \geq 10$, so that it is easy to verify that the maximum velocity v of B is sufficient for B to enter the torus, to move along within a free space and to finally exit the torus. Note that although we have used exponential velocities in the construction of this delay box, the number of bits required for their specification is polynomial in n.

For each distinguished position $P_{<u_h,q,h>}$, where $< u_h, q, h >$ is a surface configuration, we will place a distinct delay box with entry slot at $P_{<u_h,q,h>}$, with delay $\Delta = g(u_h, q, h) + 2^N - 2$ as defined above. Let the exit of this delay box be denoted $P'_{<u_h,q,h>}$. Next we will describe a construction which will force B to be moved from any such position $P'_{<u_h,q,h>}$ at time $t_{i+1} - 1$ to HOME-POSITION at time between $\lfloor t_{i+1} \rfloor$ and $\lfloor t_{i+1} \rfloor + 2/v$.

To do this, we construct (in a manner rather similar to the tree of test boxes) a balanced tree, which we call the join tree, whose edges are connecting tubes and whose nodes consist of "join boxes". The specification of this join tree will easily be seen to be polynomial in n. A 'join box' is a simple device which has two entry slots and only one exit slot. The join box will have a single exit slot and two entry slots: $entry_0$ and $entry_1$, each of width $1+\epsilon$. The join box will be a cylinder of diameter 3. To force B into each of the entry slots, we use a separate semidisk rotating once every $2/v$ time steps. Following each of the entry slots $entry_0$, $entry_1$ there is a connecting tube J_0, J_1 respectively, which join at distance 2. The resulting joined connecting tube J_2 goes to the single exit (again all these tubes have diameter $1+\epsilon$). All of these tubes J_0, J_1, J_2, are each swept by a pair of rotating semidisks located distance 1 apart and rotating once every $1/v$ time steps; The direction of the sweep insures that B can not backtrack through the other entry. This construction shows that if B is placed at either of the entry slots, then within time delay $6/v$, B is forced to move through the exit slot.

The join tree will be defined similarly to the test tree, except that the direction of forced movement is from the leaves to the root and the nodes are join boxes rather than test boxes. The exit slot of the root has a connecting tube to HOME-POSITION. The entry slots of the leaves are connected by a connecting tube to distinguished positions of the form $P'_{<u_h,q,h>}$, where $<u_h,q,h>$ is a surface configuration. All such distinguished positions will be arranged in a straight line at distance 10 units between each other. The entry slot of each join box in the interior of this tree is connected via a connecting tube to the exit slot of each of its interior children.

The depth of this tree will be the same as the previously defined test box tree. The interior of each such connecting tube is again swept by a sequence of semidisks rotating once every $2/v$ time units, so as to force B through the connecting tube from the previous exit to the next entry in time upper bounded by $2/v$ times the length of the connecting tube. We can show (by an identical calculation as for the test tree) that the total delay from HOME-POSITION to leaf is now at least $4/v$ and at most $1/2$. The delay through this tree can again be increased by modifying the construction (this time by adding a pair of semidisks (rotating just as in the case of the test tree modification) at the exit slot of the connecting tube leading from the root to HOME-POSITION so as to delay between 1 and $1 + 2/v$) so the resulting "join tree" thus has the property that if B is at the distinguished position $P'_{<u_h,q,h>}$ at time $t_{i+1} - 1$ then B is forced back to HOME-POSITION at time between $\lfloor t_{i+1} \rfloor$ and $\lfloor t_{i+1} \rfloor + 2/v$.

Thus we conclude that if B is at HOME-POSITION at time t_i encoding configuration C_i, then B is forced back to HOME-POSITION at a time t_{i+1} encoding configuration C_{i+1}. A description of the above construction can easily be computed by an $O(\log n)$ space bounded deterministic Turing Machine. □

Remarks. (1) This "time-machine" construction can be simplified further, to the case involving dynamic movement planning in 2–D space in the presence of a single moving obstacle which is a single point. Giving this obstacle a rather irregular (but still polynomially describable) motion, we can simulate both testing devices and delay devices and any additional obstacles needed to force B to move from and back to the starting position HOME-POSITION. Nevertheless, we prefer the construction given here since it uses more natural and regular kinds of motion. (We are grateful to Jack Schwartz for making this observation.)

(2) Note that our construction has utilized velocities that are n-bit integers, i.e. their moduli can grow exponentially. It remains an open question whether the dynamic movement problem is still PSPACE-hard in the case where the velocities are specified as $O(\log n)$-bit integers.

3 Efficient Algorithms for the Asteroid Avoidance Problem

Our $PSPACE$-hardness result of the previous section indicates that it may be inherently difficult to solve dynamic movers problems where the obstacles rotate. Therefore we confine our attention to the following case, which we call the *asteroid avoidance problem.*

Assume that B is an arbitrary convex polyhedron in d-space which can move only by translating with maximum velocity modulus v but without rotating (so that its motion has only d translational degrees of freedom). We also assume that each of the obstacles is a convex polyhedron which moves (without rotating) from a known initial position at a fixed and known velocity (which may vary from one obstacle to another). The obstacles are initially assumed not to collide with each other; however, as we will see below, they may collide when we "grow" them to reduce the problem to one involving a moving point. We comment on this technical difficulty below. The free configuration space FP (including time as an extra degree of freedom, as above) is $(d+1)$-dimensional. While the case $d=1$ is easy to solve, the cases $d=2,3$ of the asteroid avoidance problem are quite challenging, and require some interesting algorithmic techniques.

We have efficient algorithms for various asteroid avoidance problems. These results utilize some basic facts described in the next two subsections, of which the most important is that normal movements suffice.

3.1 Reduction to the Movement of a Point

We begin with the following simple transformation (see Lozano-Perez and Wesley [11]) to reduce the problem to the case in which B is a single moving point. Let B_0 denote the set of points occupied by B at time $t=0$. Replace each moving obstacle C by the set $C-B_0$ (which consists of pointwise differences of points of C and points of B_0). Call the resulting set the "grown obstacle" corresponding to C. Suppose that we wish to plan an admissible motion of B from the initial position B_0 to a final position B_1, and let X_1 denote the relative displacement of B_1 from B_0. Then such a motion exists if and only if there exists an admissible motion of a single point from the origin to X_1 which avoids collision with the moving grown obstacles $C-B_0$ (each such body moving with the same velocity as the obstacle body C). Since the grown obstacles are also convex polyhedra we have reduced the problem to a similar one in which B can be assumed to be a single moving point. Note that the grown obstacles may intersect even if the original obstacles were assumed not to collide. However, if the individual grown obstacles have a total of n faces then their union in space–time (which is the space our moving point must not enter) has complexity at most $O(n^{d+1})$.

In the remainder of this section, we will therefore assume that B is a moving point. To simplify the foregoing analysis, we will assume there that even the grown obstacles do not collide. To handle the general case, where these obstacles may collide, we can take the union of the space–time trajectories of all obstacles and decompose it into a collection of pairwise openly disjoint convex polyhedra. Since the overall number of faces bounding those polyhedra is still polynomial in n, we can easily adapt the following analysis so that it can also handle the intersecting case. The only case which requires a more careful analysis is the 1–D case, where we aim to obtain an $O(n \log n)$ algorithm. To retain this efficiency, we use certain properties of 2–D grown obstacles, derived in [10], to argue that the overall complexity of the grown obstacles is still proportional to the complexity of the original obstacles — see below.

3.2 Normal Movements

We will require some special notation for various types of movement of a point B over a given time interval. In all the following types of movement of B, we allow B to touch an obstacle boundary, but do not allow it to move to the interior of any obstacle, and require that B not exceed a maximum velocity modulus v.

1. A *direct movement* is a movement of B with a constant velocity vector. During a direct movement, B may touch an obstacle only at the endpoints of that

movement. A special case of direct movement is *static movement* in which B does not move (i.e., has 0 velocity).

2. A *contact movement* is a movement of B in which B moves on the boundary of an obstacle C (i.e., the boundary of the region of FP induced by the movement of C). In the 2–D asteroid avoidance problem, we also require that any such (maximal) contact movement begin and end at (contact with) vertices of an obstacle (including possible points of intersection between edges of distinct colliding obstacles). In the 3–D asteroid avoidance problem, we require that each contact movement begin and end only at (contact with) edges or vertices of an obstacle (again including possible intersections between faces or a face and an edge of two distinct colliding obstacles). In addition, since obstacles can collide with each other, we require from a contact movement that it occurs along the boundary of the union of all space-time obstacles (thus excluding contact movements along artificial cuts between two adjacent convex pieces of space-time obstacles). We also allow degenerate contact movements, where the contact happens during a single instance of time (at an obstacle vertex in the 2-D case, or along an obstacle edge in the 3-D case).

3. A *fundamental movement* of B is a direct movement followed possibly by a contact movement.

4. A *normal movement* of B is a (possibly empty) sequence of fundamental movements of B where the movements must satisfy the following restrictions:

R1: Between any two distinct direct movements, there must be a contact movement, and

R2: If the space-time obstacles do not collide, no two distinct (maximal) contact movements are allowed to visit (the boundary of) the same obstacle. In general, no two distinct (maximal) contact movements are allowed to visit the same vertex (in the 2-D case) or the same edge (in the 3-D case) of an obstacle.

Note that R1 requires that a normal movement does not change its direction except while in contact with an obstacle. R2 ensures that a normal movement consists of $\leq k+1$ fundamental movements, where k is the number of obstacles.

Lemma 3.1 *B has a collision-free movement $p(t) = [X_t, t]$ from $[X_0, 0]$ to $[X_T, T]$ iff B has a finite sequence of fundamental movements from $[X_0, 0]$ to $[X_T, T]$ satisfying R1.*

Proof. If B has a sequence of fundamental movements from $[X_0, 0]$ to $[X_T, T]$, then this clearly constitutes a collision-free movement between its endpoints (in the weak sense defined in the Introduction).

For the converse part, consider the class K of all paths $p(t) = [X_t, t]$ in space-time from $[X_0, 0]$ to $[X_T, T]$, whose slope at any given time is of modulus at most v, and which avoids penetration into the interior of any obstacle. By assumption K is not empty. Let $\pi_0 \in K$ be the shortest path in K (where the length of a path in K is its Euclidean length in E^{d+1}).

Observe that if $\pi \in K$ and if $[X_1, t_1]$, $[X_2, t_2] \in \pi$, then the path π', obtained by replacing the portion of π between these two points by the straight segment joining them (in space-time), is such that its slope at any given time is $\leq v$. Since the space-time trajectory of each obstacle is a convex polyhedron, it follows, using standard shortest-path arguments, that π_0 must be a polygonal path, which consists of an alternating sequence of free straight segments and of polygonal subpaths in which B is in contact with an obstacle. Moreover, the vertices of π_0 must lie along $(d-1)$-dimensional faces of the space-time trajectories of the moving obstacles, so they correspond to contacts of B with $(d-2)$-dimensional faces of these obstacles. Thus π_0 is a sequence of fundamental movements satisfying R1. $\Box$

Lemma 3.2 *B has a collision-free movement from $[X_0, 0]$ to $[X_T, T]$ iff B has a normal movement from $[X_0, 0]$ to $[X_T, T]$.*

Proof. By Lemma 3.1, we can assume B has a movement $[X_t, t]$ defined for $0 \leq t \leq T$, consisting of a sequence of fundamental movements beginning at times $0 \leq t_1, t_2, \ldots, t_m \leq T$ and satisfying R1. Moreover, the proof of Lemma 3.1 is easily seen to imply that the weaker part of R2 is also satisfied. If the space-time obstacles do not collide and the stronger part of restriction R2 is violated, then there must be times t_i, t_j such that $[X_t, t]$ is in contact with the same obstacle C during times t_i and t_j. But since C is convex, its trajectory $C^\star$ in space-time is also convex. It is then easy to construct a single contact path $[X'_t, t]$ along $C^\star$ for $t_i \leq t \leq t_j$, such that $X'_{t_i} = X_{t_i}$ and $X'_{t_j} = X_{t_j}$, and such that the slope of this path at any given time is of modulus $\leq v$. (Intuitively, "pull taut" the path $[X_t, t]$ in space–time between t_i and

t_j in the presence of C^* alone.) Repeating this process as required, we get a normal movement satisfying both R1 and R2.

The other direction follows from Lemma 3.1. $\square$

Remark. In particular, the preceding lemma implies that a minimum time movement of B between two given spatial positions can always be realized by a normal movement.

3.3 The Asteroid Avoidance Problem with One Degree of Freedom of Movement

We will first consider (a slight generalization of) the case of a 1–D asteroid avoidance problem, where we assume B is constrained to move along a fixed line, in the presence of 2–D convex polygonal obstacles which can pass through that line. The problem is not difficult in this case since B has only one degree of freedom movement. (Nevertheless, a brief investigation of this case will aid the reader to understand better the techniques which we use for the more difficult cases of $d = 2, 3$ degrees of freedom.) Let n be the total number of obstacle edges. Let k be the number of obstacles. By the reduction of Section 3.1, we can assume B is a single point.

Theorem 3.3 *The asteroid avoidance problem can be solved in time $O(n \log n)$ If B is constrained to move only along a 1–dimensional line L.*

Proof. The key observation is that the (space-time) configuration space FP in this case is a 2–dimensional space bounded by polygonal barriers generated (in a manner detailed below) by the uniform motions along L of the intersections of obstacle edges with L. We explicitly construct FP using a scan-line technique. We first sort in time $O(n \log n)$ all obstacle edges and vertices in the order of times in which they first intersect L. Let this sorted sequence of times be $t_1, \ldots, t_m$ (where $m = O(n)$). As we sweep the scan line across time, we maintain for each time t the set FP_t of all accessible free configurations at time t, and also a sorted list Q of the intersections of obstacle edges with L at time t. Suppose $[X_0, 0]$ is the initial configuration of B. Initially FP_0 consists of the single point $[X_0, 0]$ in space-time, and the initial value of Q is easily calculated in time $O(n \log n)$. Inductively, suppose for some $t_i \geq 0$ we have constructed FP_{t_i}. We represent FP_{t_i} as an ordered, finite sequence of disjoint intervals $I_1, \ldots, I_{s_i}$ of L, whose union is the set of all points X such that there is a collision-free movement of B, whose velocity modulus never exceeds v, from $[X_0, 0]$ to $[X, t_i]$. Let t_{i+1} be the next time following t_i that an obstacle vertex intersects L. Between the times t_i and t_{i+1}, each endpoint of an interval I_j moves at a uniform velocity in one of two possible ways:

(a) If this endpoint is not incident to an obstacle, or is initially incident to an obstacle that moves away at speed larger than v, then it moves at velocity v so as to expand the interval I_j.

(b) Otherwise, the endpoint moves so as to remain incident to the obstacle edge it is initially at.

Thus as t varies from t_i to t_{i+1}, L can change combinatorially when two adjacent expanding intervals meet and merge into one interval or when an interval shrinks and disappears (when one of its endpoints is at an obstacle edge that moves too fast towards the other endpoint). In either case the number of intervals in L can only get smaller. New intervals are added to L only when an obstacle first meets the line L and this happens only $k \leq n$ times. In addition, when an obstacle vertex crosses L, the velocity of an interval endpoint can change.

These considerations easily imply that the total number of updating steps that are needed to maintain FP_t is only $O(n)$, and each step is easy to carry out in $O(\log n)$ time, using an appropriate balanced tree structure for Q and FP_t and an additional priority queue to record all critical times at which the combinatorial structure of FP_t changes. The total time of the algorithm is therefore $O(n \log n)$. $\Box$

Remark. As per our convention, we have assumed above that the expanded obstacles do not collide in space–time. If this is not the case, we can still apply the above analysis by splitting the union of the expanded obstacles in space–time into pairwise openly disjoint convex polygons. Fortunately, the results of [10] imply that the total complexity of the union of the expanded obstacles is only $O(n)$, so the modified algorithm still runs in $O(n \log n)$.

3.4 A Polynomial Time Algorithm for the 2–D Asteroid Avoidance Problem for a Bounded Number of Obstacles

In this subsection, we consider the 2–D asteroid avoidance problem. The configuration space FP in this case is 3–dimensional. We can assume, by the reduction of Section

3.1, that B is a single point. We wish to move B from $[X_0, 0]$ to $[X_T, T]$. The obstacles $C_1, \ldots, C_k$ are k (expanded) convex polygons. To simplify the analysis, we assume again that the obstacles C_j are pairwise disjoint, as would be the case if B is originally a moving point. Let n, the *size* of the problem, be the total number of vertices and edges of the obstacles (in the general case, it would be the number of vertices, edges, and faces of the decomposed space-time obstacles, which is still only polynomial in the number of original obstacle edges and vertices). We will show that if k is a constant, then we can solve the problem in $n^{O(1)}$ time.

Our basic technique will be to first consider the problem of computing the time intervals in which single direct and contact movements between obstacle vertices can be made, and then use a recursive method to determine the time intervals in which it is possible to do normal movements.

For technical reasons, we consider the initial and final positions of B to be additional immobile "obstacles" $C_0 = X_0$, $C_{k+1} = X_T$, each consisting of a single vertex. Let $V(C_j)$ be the set of vertices of obstacle C_j for $j = 1, \ldots, k$ and let $V(C_0) = \{X_0\}$ and $V(C_{k+1}) = \{X_T\}$. Let $V = \cup_{j=0}^{k+1} V(C_j)$ be the set of all vertices. Note that for each $j = 0, \ldots, k+1$, all vertices $a \in V(C_j)$ undergo a translational motion with the same fixed velocity vector.

We will use I to denote the set of times a certain event will occur. Let $|I|$ denote the minimum number of disjoint intervals into which the points of I can be partitioned. Clearly, I can be written using $O(|I|)$ inequalities. We will store the intervals of I in sorted order using a balanced binary tree of size $O(|I|)$, in which we can do insertions and deletions in time $O(\log |I|)$.

For each $a, a' \in V(C_j)$, let $CM_{a,a'}(I)$ be the set of all times $t' \geq 0$ at which vertex a' can be reached by a contact movement of B on the boundary of C_j starting at vertex a at some time $t \in I$.

Lemma 3.4 *$CM_{a,a'}(I)$ can be computed in time $O(|I| + |V(C_j)|)$ and furthermore $|CM_{a,a'}(I)| \leq |I|$.*

Proof. There are fixed reals $0 \leq \Delta_1 \leq \Delta_2$ (both of which are possibly infinite) such that vertex a' can be reached from vertex a by a contact movement within minimum delay Δ_1 and maximum delay Δ_2. These delay parameters Δ_1, Δ_2 can be easily computed (by computing the sum of the delay bounds required for near-contact movement of each of the edges of C_j from a to a') in time $O(|V(C_j)|)$.

(We note that for this property to hold we need to assume that the given velocities of the obstacles are "well-behaved", in the sense that they do not require too many bits to write down, so that operations on these velocities can be accomplished in constant time (or at worst within some time bound that is polynomial in n).)

Since trivially

$$CM_{a,a'}(I) = \{t' | \Delta_1 + t \leq t' \leq \Delta_2 + t, t \in I\},$$

we have $|CM_{a,a'}(I)| \leq |I|$, and it can be computed (under the assumption just made) within time $O(|I| + |V(C_j)|)$. $\square$

For each $a, a' \in V$, let $DM_{a,a'}(I)$ be the set of all times $t' \geq 0$ such that vertex

a' can be reached at time t' by a single direct movement of B starting at vertex a at some time $t \in I$.

To calculate $DM_{a,a'}(I)$, we consider the following subproblem: Find the set $F_{a,a'}$ of all pairs of times t, t' such that the position $a'(t')$ of a' at time t' can be reached from the position $a(t)$ of a at time t by a single direct movement.

Fix a time t and let $A(t)$ denote the set of all times t' such that the slope of the motion from $[a(t), t]$ to $[a'(t'), t']$ has modulus $\leq v$. Plainly $A(t)$ is a closed interval $[t_1, t_2]$. Consider the triangle Δ whose corners are $w = [a(t), t]$, $w_1 = [a'(t_1), t_1]$, $w_2 = [a'(t_2), t_2]$. For each obstacle C_j, its space-time trajectory C_j^* intersects Δ at a convex set Δ_j. The two tangents from w to Δ_j cut an interval $A_j(t)$ off the segment $w_1 w_2$. $A_j(t)$ is exactly the set of positions $[a'(t'), t']$ of a' which are not reachable from $[a(t), t]$ by a single direct movement, due to the interference of C_j. Let $I_j(t)$ denote the projection of $A_j(t)$ onto the t–axis. The set $F_{a,a'}$ is then

$$\{(t, t') : t' \notin \cup_{j=1}^{k} I_j(t)\}.$$

Suppose $F_{a,a'}$ has been calculated. Then

$$DM_{a,a'} = \{t' : \exists t \in I, (t, t') \in F_{a,a'}\}.$$

To calculate the two-dimensional set $F_{a,a'}$ we can use a standard technique of sweeping a line $t = const$ across the (t, t')-plane. Note that for each t and j, each endpoint of $I_j(t)$ is determined by a specific vertex of C_j, and that, given such a vertex v, the corresponding endpoint $e_v(t)$ of $I_j(t)$ is an algebraic function in t of constant degree. Hence the structure of $F_{a,a'} \cap \{t = const\}$ can change during the sweeping only at points t where two functions $e_v(t)$, $e_{v'}(t)$ intersect, or where one such function has a vertical tangent, i.e. at $O(n^2)$ points at most. This readily implies

Lemma 3.5 *$F_{a,a'}$ can be calculated in time $O(n^2 \log n)$, and stored in $O(n^2)$ space. Furthermore, for each I, $|DM_{a,a'}(I)| \leq (|I| + n^2)k$, and $DM_{a,a'}(I)$ can be calculated in time $O((|I| + n^2)k)$.*

Proof. The first part follows by the sweeping technique mentioned above. The second part follows from the fact that, as a result of the sweeping, the t–axis is split into $O(n^2)$ intervals, over each one of which the combinatorial structure of $F_{a,a'}$ remains constant, and consists of at most $k+1$ disjoint intervals. Hence, merging these intervals with the intervals of I, we can calculate $DM_{a,a'}(I)$ in a straightforward manner within the asserted time bound, and also obtain the required bound on the complexity of that set. $\Box$

Theorem 3.6 *The 2–D asteroid avoidance problem can be solved in time $O(n^{2(k+2)}k)$, and hence in time $n^{O(1)}$ in the case of $k = O(1)$ obstacles.*

Proof. Initially let $I_{X_0}^{(0)} = \{t | 0 \leq t \leq T\}$ and let $I_a^{(0)} = \emptyset$ for each $a \in V - \{X_0\}$. Inductively for some $i \geq 0$, suppose for each $a \in V$, $I_a^{(i)}$ is the set of times t that vertex a is reachable from $[X_0, 0]$ by a (collision-free) normal movement of B consisting of $\leq i$ fundamental movements in time $\leq T$. Then for each $a' \in V$,

$$J_{a'}^{(i)} = \bigcup_{a \in V} DM_{a,a'}(I_a^{(i)})$$

is the set of times that vertex a' is reachable from $[X_0, 0]$ by a movement of B consisting of $\leq i$ fundamental movements followed by a direct movement and no other kinds of movement. Hence if $a'' \in V(C_j)$ then

$$I_{a''}^{(i+1)} = \bigcup_{a' \in V(C_j)} CM_{a',a''}(J_{a'}^{(i)})$$

is the set of times that vertex a'' is reachable from $[X_0, 0]$ by a normal movement of B consisting of $\leq i+1$ fundamental movements. Thus for each $a \in V$, $I_a^{(k+1)}$ is the set of times vertex a is reachable by a normal movement of B from $[X_0, 0]$. By Lemma 3.2 such a normal movement suffices. So $T \in I_{X_T}^{(k+1)}$ iff there exists a collision-free movement of B from $[X_0, 0]$ to $[X_T, T]$.

Lemmas 3.4 and 3.5 imply $|I_a^{(i)}| \leq O(n^{2i+2}k)$ and so the $i-th$ step takes time $O(n^2(n^{2i+2}k + k\log(n^{2i+3}k)))$. Therefore, the total time is $O(n^{2(k+2)}k)$. $\square$

3.5 A Decision Algorithm for the 3-D Asteroid Avoidance Problem with an Unbounded Number of Obstacles

We next consider the 3–D asteroid avoidance problem. The configuration space FP is in this case 4–dimensional. By the results of Section 3.1, we can assume we wish to move a point B from $[X_0, 0]$ to $[X_T, T]$, avoiding k (possibly intersecting) convex polyhedral obstacles $C_1, \ldots, C_k$. In this case the *size* n of the problem is the total number of edges of the polyhedra (or, in case they intersect, the total number of features on the boundary of the union of their space-time trajectories). Again, we will present the analysis under the assumption that the obstacles do not intersect, but an appropriate modification of the analysis will also apply in the general case. We will show that the problem is decidable.

Recall that each contact movement is required to begin and end at an obstacle edge or vertex. We will consider each obstacle edge $e = (u, v)$ to be directed from u to v. If e has length L we will let $e(y)$, for $0 \leq y \leq 1$, denote the point on e at distance yL from vertex u, so $e(0) = u$ and $e(1) = v$. Let $E = \{e_1, \ldots, e_n\}$ be the set of all obstacle edges. Let $E(C_j) \subset E$ be the set of (directed) edges of obstacle C_j for $j = 1, \ldots, k$.

For technical reasons we again consider the initial and final positions of B to be immobile obstacles $C_0 = X_0$ and $C_{k+1} = X_T$. We consider $E(C_0)$ to contain a single edge of length 0 at point X_0 and $E(C_{k+1})$ to contain a single edge of length 0 at point X_T.

An *open formula* $F(y_1, \ldots, y_r)$ in the theory of real closed fields consists of a logical expression containing conjunctions, disjunctions, and negations of atomic formulas, where each atomic formula is an equality or inequality involving rational polynomials in the variables $y_1, \ldots, y_r$. A (partially quantified) formula in this theory is a formula of the form $Q_1 y_1 \cdots Q_a y_a F(y_1, \ldots, y_r)$ where $a \leq r$, and where each Q_i is an existential or a universal quantifier. Such a formula will be called an *algebraic predicate*; its *degree* is the maximum degree of any polynomial within the formula, and its *size* is the number of atomic formulas it contains. We will use the following results:

Lemma 3.7 [Collins, 75]. *A given formula of the theory of real closed fields of size n, constant degree, with r variables can be decided in deterministic time $n^{2^{O(r)}}$.*

Lemma 3.8 [Canny, 88, and Renegar, 92]. *A given formula of the* existential *theory of real closed fields of size n, constant degree and r variables can be decided in space polynomial in n and r.*

We will first show that we can describe by algebraic predicates the time intervals for which fundamental movements can be made, and then use the existential theory of real closed fields to decide the feasibility of movements consisting of finite sequences (of length at most n) of these fundamental movements. Below we fix a pair of edges e_i, $e_{i'}$ lying on any common obstacle C_j and $0 \leq y, y' \leq 1$. Let $cm(i, i', y, y', \Delta)$ be the predicate which holds just if B has a contact movement along a single face of C_j from $e_i(y)$ to $e_{i'}(y')$ with delay Δ (i.e. the motion takes Δ time units); note that in this notation C_j is implicitly defined by the indices i, i', and that for cm to be true it is necessary that e_i, $e_{i'}$ bound the same face of some obstacle.

Lemma 3.9 *$cm(i, i', y, y', \Delta)$ can be constructed in polynomial time as a predicate of size $n^{O(1)}$ with no quantified variables, which is algebraic, of constant degree, in y, y', and Δ.*

Proof. Let $face(i, i')$ be the predicate that holds iff e_i and $e_{i'}$ are both on the same face of an obstacle. Let (w_x, w_y, w_z) be the velocity vector of obstacle C_j containing e_i and $e_{i'}$. Let (u_x, u_y, u_z) be the distance vector from $e_i(y)$ to $e_{i'}(y')$, where u_x, u_y, u_z are all linear functions of y, y'. B will move in contact with C_j with velocity vector (v_x, v_y, v_z) with modulus

$$\sqrt{v_x^2 + v_y^2 + v_z^2} \leq v.$$

If B moves from $e_i(y)$ to $e_{i'}(y')$ with delay Δ then we must have $v_x\Delta = w_x\Delta + u_x$, $v_y\Delta = w_y\Delta + u_y$, and $v_z\Delta = w_z\Delta + u_z$.

Solving for v_x, v_y, v_z and substituting into $v_x^2 + v_y^2 + v_z^2$, we derive the formula

$$cm(i, i', y, y', \Delta) \equiv$$

$$\left(\left[\left(\frac{w_x\Delta + u_x}{\Delta}\right)^2 + \left(\frac{w_y\Delta + u_y}{\Delta}\right)^2 + \left(\frac{w_z\Delta + u_z}{\Delta}\right)^2\right] \le v^2\right) \wedge face(i, i').$$

□

Let $dm(i, i', y, y', t, t')$ be the predicate which holds just if B has a (collision-free) direct movement from $e_i(y)$ at time t to $e_{i'}(y')$ at time t'. The following is proved using arguments similar to those used in Lemma 3.5.

Lemma 3.10 *$dm(i, i', y, y', t, t')$ can be constructed in polynomial time as an algebraic predicate of size and degree $n^{O(1)}$ with no quantified variables.*

Proof. For a given set of obstacles $\mathcal{C} \subset \{C_1, \ldots, C_k\}$ let $dm_{\mathcal{C}}(i, i', y, y', t, t')$ be defined as above, except that we allow possible collisions of B with obstacles in $\{C_1, \ldots, C_k\} - \mathcal{C}$. Then $dm_{\emptyset}(i, i', y, y', t, t')$ can easily be given as an algebraic predicate of size $n^{O(1)}$ bounding the time t' to a single (possibly empty) interval, whose bounds vary algebraically with t.

Inductively we can write $dm_{\{C_1,\ldots,C_j\}}(i, i', y, y', t, t')$ as the conjunction

$$dm_{\{C_1,\ldots,C_{j-1}\}}(i, i', y, y', t, t') \wedge p\ ,$$

where p is an algebraic predicate of size $n^{O(1)}$, restricting t' outside a single (possibly empty) interval of time. Thus $dm(i, i', y, y', t, t') = dm_{\{C_1,\ldots,C_k\}}(i, i', y, y', t, t')$ is an algebraic predicate of size $n^{O(1)}$. $\square$

Let $fm(i, i', y, y', t, t')$ hold iff there is a fundamental movement of B from $e_i(y)$ at time t to $e_{i'}(y')$ at time t'. Lemmas 3.9 and 3.10 imply

$$fm(i, i', y, y', t, t') \equiv dm(i, i', y, y', t, t') \vee cm(i, i', y, y', t' - t)$$

$$\vee \exists i'', y'', \Delta \;\; dm(i, i'', y, y'', t, t' - \Delta) \wedge cm(i'', i', y'', y, \Delta) .$$

(Recall that cm can be true only if the two indices appearing in it denote edges lying on the same obstacle.)

Lemma 3.11 *$fm(i, i', y, y', t, t')$ can be constructed in polynomial time as an algebraic predicate of size $n^{O(1)}$, constant degree and $O(1)$ quantified variables.*

Let $m(i, i', y, y', t, t')$ be the predicate that holds iff B has a collision-free (normal) movement from $e_i(y)$ at time t to $e_{i'}(y')$ at time t'. Note that the formula for $m(i, i', y, y', t, t')$ requires $\Theta(n)$ existentially quantified variables. We thus have

Theorem 3.12 *The 3–D asteroid avoidance problem can be solved in time $2^{n^{O(1)}}$, or alternatively in polynomial space.*

Proof. We assume immobile obstacle edges e_1, e_2 such that $e_1(0) = X_0$ and $e_2(0) = X_T$. By definition, B has a collision-free movement from $[X_0, 0]$ to $[X_T, T]$ if and only if $m(1, 2, 0, 0, 0, T)$ holds.

Since $m(1, 2, 0, 0, 0, T)$ has $n^{O(1)}$ size and $O(n)$ existentially quantified variables, we can test satisfiability of $m(1, 2, 0, 0, 0, T)$ by Lemma 3.8 in polynomial space, and thus also in time $2^{n^{O(1)}}$. $\square$

The fact that the 3–D asteroid avoidance problem can be solved in singly exponential time can also be derived from the older result of Collins (Lemma 3.7). We include a description of this technique because we will need to use this variant in our analysis of minimum-time movements, to be given in the following subsection. In addition, we think this technique is interesting in its own right and may have other applications as well. Specifically, we first need

Lemma 3.13 *$m(i, i', y, y', t, t')$ can be constructed in polynomial time as an algebraic predicate of size $n^{O(1)}$ with constant degree using $O(\log n)$ quantified variables.*

Proof. For each $\ell = 0, 1, \ldots, \log n$, we define $m^{(\ell)}(i, i', y, y', t, t')$ to be the predicate that holds iff B has a movement from $e_i(y)$ at time t to $e_{i'}(y')$ at time t' consisting of a sequence of $\leq 2^\ell$ fundamental movements. Clearly, $m^{(0)}(i, i', y, y', t, t') = fm(i, i', y, y', t, t')$. We can then define

$$m^{(\ell+1)}(i, i', y, y', t, t') \equiv \exists i'', y'', t''$$

$$m^{(\ell)}(i, i'', y, y'', t, t'') \wedge m^{(\ell)}(i'', i', y'', y', t'', t').$$

However, this definition, when applied recursively yields a formula of size $\geq 2^n$. A more compact definition is gotten by

$$m^{(\ell+1)}(i, i', y, y', t, t') \equiv \exists i'', y'', t'' \; \forall a_1, a_2, a_3, a_4, a_5, a_6$$

$$[(a_1 = i \wedge a_2 = i'' \wedge a_3 = y \wedge a_4 = y'' \wedge a_5 = t \wedge a_6 = t'')\vee$$

$$(a_1 = i'' \wedge a_2 = i' \wedge a_3 = y'' \wedge a_4 = y' \wedge a_5 = t'' \wedge a_6 = t')]$$

$$\supset m^{(\ell)}(a_1, a_2, a_3, a_4, a_5, a_6).$$

The formula $m^{(\lceil \log n \rceil)}(i, t', y, y', t, t')$ is of size $n^{O(1)} \log n \leq n^{O(1)}$ and requires only $O(\log n)$ quantified variables. By Lemmas 3.1 and 3.2 we have $m(i, i', y, y', t, t') \equiv m^{(\lceil \log n \rceil)}(i, t', y, y', t, t')$. $\square$

Now Lemma 3.7 implies that we can test satisfiability of $m(1, 2, 0, 0, 0, T)$ in time $2^{n^{O(1)}}$, as asserted in the preceding theorem.

3.6 Minimum-time Asteroid Avoidance Problems

We conclude Section 3 by observing that the techniques developed in Sections 3.3, 3.4, and 3.5 can be easily modified to solve minimum-time asteroid avoidance problems, which ask for planning collision-free movement of the body B, which will take it from the initial configuration $[X_0, 0]$ to a final position X_1 in the shortest possible time. (Here B is allowed to contact, but not penetrate into any of the obstacles.)

To solve problems of this kind, we consider the 1–D, the 2–D, and the 3–D cases separately. In the 1–D case, the algorithm given in Section 3.3 calculates (the closure of) FP explicitly. Given the desired final position X_1 of B, we just need to find the point of intersection of the line $x = X_1$ with (the closure of) FP which has the smallest t-value. This task is easily accomplished in $O(n)$ time.

In the 2–D case, the algorithm given in Section 3.4 produces for each vertex $a \in V$

a set of times $I_a^{(k+1)}$ in which a can be reached by a normal movement from $[X_0, 0]$. Here all we have to do is to consider the destination X_1 as an additional stationary obstacle. Then $T = min I_{X_1}^{(k+1)}$ is the shortest time in which such a normal movement can reach X_1. That normal movement itself can also be easily calculated.

Finally, in the 3–D case, using the notations of Section 3.5, we consider the predicate

$$m^{\star} \equiv \ \exists T \ \ m(1, 2, 0, 0, 0, T) \ .$$

The technique quoted in Lemma 3.7 is based on decomposition of E^r into a collection of finitely many connected cells having relatively simple structure, such that within each such cell c the Boolean value of each atomic subformula in $m^{\star}$ has a constant value.

Furthermore, by tuning the algorithm which calculates this decomposition, we can obtain a partitioning of the T-axis into finitely many disjoint intervals such that each cell in the decomposition projects onto such an interval or onto an endpoint of such an interval. Hence, by scanning these T-intervals in increasing order, it is easy to find the smallest T for which $m^{\star}$ is true. This technique is similar to that described in Section 4.2 below, and the reader is referred to this section for more details.

Note that the arguments just given for the 2–D and 3–D cases show how to find minimum-time *normal* movement. However, by Lemma 3.1 and the remark following it, the minimum time achieved by a normal movement is the same as that achieved by any admissible collision-free movement.

Summing up all these observations, we conclude

Theorem 3.14 *The minimum-time asteroid avoidance problem can be solved in time $O(n \log n)$ in the 1–D case, in time $O(n^{2(k+2)}k)$ in the 2–D case, and in time $2^{n^{O(1)}}$ in the 3–D case.*

Remark. Although the original version of this paper [15] did not mention minimum-time movements explicitly, the ability to calculate minimum-time movement was implicit in the techniques presented there. The paper by Sutner and Maass [23] also considers this problem.

4 Dynamic Movement Problems with Unrestricted Velocity

Throughout the last two sections we have assumed that B had a given velocity modulus bound. Here we will allow B to have unrestricted motion; and in particular we will impose no velocity bounds.

This case appears still intractable, as we show that the 3–D dynamic movement problem for the case where B is a cylinder with unrestricted motion, is NP-hard. Again this proof requires that B has only $O(1)$ degrees of freedom and we make critical use of the presence of rotating obstacles to encode time.

We will next show, in contrast with what has just been stated, that the problem is in polynomial time if all the obstacle motions are algebraic (of bounded degree) in space-time; that is, the movement of B is constrained by algebraic inequalities of bounded degree, and there is no bound on the velocity modulus of B.

4.1 The Case of Unrestricted Motions in the Presence of Rotating Obstacles is NP-hard

We will reduce the 3–satisfiability problem to that of planning the motion of a cylindrical body B in 3–space in the presence of several rotating obstacles. Suppose that we are given an instance of 3–satisfiability involving n Boolean variables $x_1, \ldots, x_n$. With each variable x_i we associate several *semidiscs* $D_{i,k}$ of radius 1, where a semidisc is a disc with half its interior removed so that it is bounded by a semicircle and a line segment. Each semidisc $D_{i,k}$ rotates in some plane lying parallel to the $x - y$ plane at some height $h_{i,k}$ with its center at some point $w_{i,k}$. For each $i = 1, \ldots, n$, all the semidiscs $D_{i,k}$ rotate with the same angular velocity $v_i = \frac{\pi}{2^{i-1}}$. Thus the first set of semidiscs complete half a revolution in 1 time unit, the second set in 2 time units, and so forth. The idea behind this mechanism is that it can be used to encode the binary digits of time. Specifically, if U is a sufficiently small disc contained in the interior of some unit disc on the $x - y$ plane and lying near its perimeter, then we can position some of our semidiscs $D_{1,k_1}, \ldots, D_{n,k_n}$ above U in such a way that after t whole time units each semidisc D_{i,k_i} will cover the set $U + w_{i,k_i}$ if and only if the i-th binary digit of t has some designated value ϵ_i. We assume that the horizontal cross section of B has an area smaller than that of U and that B is sufficiently long, so that after $t < 2^n$ whole time units B can stand vertically with its base on U without colliding with any of these semidiscs if and only if $t = \epsilon_1\epsilon_2 \cdots \epsilon_n$ in binary. This useful feature will be crucial in the following construction.

Suppose that the given instance of 3–satisfiability involves p clauses, where the m-th clause has the form $z_{m_1} \vee z_{m_2} \vee z_{m_3}$, where each z_t is either x_t or the negation of x_t. We represent this clause by three semidiscs $D_{m_1,m}, D_{m_2,m}, D_{m_3,m}$, all placed on a plane at some height h_m (without touching or intersecting each other), such that their centers all lie on the y axis of this plane, and such that the empty half of $D_{m_i,m}$ is placed initially to the right of the y-axis if $z_{m_i} = x_{m_i}$; otherwise the semidisc is placed initially with its empty half to the left of the y-axis. We then construct three narrow tunnels, all connecting some point C_m lying between the $(m-1)$-th plane and the m-th plane just introduced, to a point C_{m+1} lying above the new plane. Each tunnel is circular, and its intersection with the plane is a sufficiently small disc lying within the right half of the corresponding disc D near its highest (in y) point. This construction implies that at time t the body B that we wish to move can quickly go from C_m to C_{m+1} iff the assignment of the i-th binary digit of t to the variable x_i, for each $i = 1, \ldots, n$, satisfies the m-th clause. It follows that we can move B from an initial position C_1 to a final C_{m+1} iff there exists a time t for which the above assignment satisfies the given instance of the 3–satisfiability problem. (It is easy to add more rotating discs that would enforce B to traverse the whole system of tunnels in a very short time that begins at an integral number of time units.) This proves that

Theorem 4.1 *In the presence of rotating obstacles, dynamic motion planning of a body B with no velocity modulus bound is NP-hard, even in the case where the body*

B is a rigid cylinder in 3–space.

Remark. As in the case of the time–machine construction in Section 2, this construction can also be simplified to a two-dimensional dynamic movement planning with a single moving point obstacle, at the cost of using an irregular and more complex motion of that obstacle.

4.2 The Case of Unrestricted Algebraic Motions

Let B be an arbitrary fixed system of moving bodies with a total of d degrees of freedom. Let S be a space bounded by an arbitrary collection of moving obstacles. Let the (space-time) free configuration space FP of B be defined as in Section 1. We will assume that the problem is algebraic in the sense that the geometric constraints on the possible free configurations of B (i.e., the constraints defining FP) can be expressed as algebraic (over the rationals) equalities and inequalities in the $d+1$ parameters $[X, t]$. For technical reasons, and unlike the convention used so far in the paper, we allow here only movements of B in which it really avoids any contact with (and, of course, penetration into) any obstacle.

Remark. Some of the motions used in the preceding lower bound proofs are not algebraic in the above sense. The simplest such motion is rotation of a two-dimensional body about a fixed center. Indeed, suppose for simplicity that the rotating body is a single point at distance r from the center of rotation (which we assume to be the origin). Then the curve in space-time traced by the rotating point is a helix, parametrized as $(x, y, t) = (r \cos \omega t,\ r \sin \omega t,\ t)$, which is certainly not algebraic.

To obtain a polynomial-time solution to this problem, we decompose E^{d+1} into a *cylindrical algebraic decomposition* as proposed by Collins [4] (or Collins' decomposition in short; cf. Cooke and Finney [5] for a basic description of cell complexes) relative to the set P of polynomials appearing in the definition of FP. (We have already cited Collins' technique in Lemma 3.7.) Roughly speaking, this technique partitions E^{d+1} into finitely many connected cells, such that on each of these cells each polynomial of P has a constant sign (zero, positive, or negative). Thus FP is the union of a subset of these cells, and it is a simple matter to identify those cells which are contained in FP (we refer to such cells as *free* Collins cells). Moreover, by using the modified decomposition technique presented in Schwartz and Sharir [19], one can also compute the *adjacency* relationships between Collins cells (i.e., find pairs $[c_1, c_2]$ of Collins cells such that one of these cells is contained in the boundary of the other). Thus any continuous path in FP can be mapped to the sequence of free Collins cells through which it passes, and conversely, for any such sequence of free adjacent Collins cells we can construct a continuous path in FP passing through these cells in order. This observation has been used by Schwartz and Sharir [19] to reduce the continuous (static) motion planning problem to the discrete problem of searching for an appropriate path in an associated *connectivity graph* whose nodes are the free Collins cells, and whose edges connect pairs of adjacent such cells.

We would like to apply the same ideas to the dynamic problem that we wish to solve, but we face here the additional problem that we are allowed to consider only t-monotone paths in FP. To overcome this difficulty, we note that the Collins decomposition procedure is recursive, proceeding through one dimension at a time. When it comes to decompose the subspace E^{i+1}, it has already decomposed E^i into "base" cells, and the decomposition of E^{i+1} will be such that for each base cell b of E^i there will be constructed several "layered" cells of E^{i+1} all projecting into b. Hence if we apply the Collins decomposition technique in such a way that the time axis t is decomposed in the innermost recursive step, it follows that each final cell c (in E^{d+1}) consists of points $[X, t]$ whose t either lies between two boundary times $t_0(c) < t_1(c)$ or is constant. Moreover, if c is a Collins cell of the first type, then it is easy to show, using induction on the dimension, that for any point $[X_0, t_0(c)]$ lying on the "lower" boundary of c, and for any point $[X_1, t_1(c)]$ on its "upper" boundary, there exists a continuous t–monotone path through c connecting these two points. In fact, the preceding property also holds if one or both points are interior to c.

These observations suggest the following procedure.

1. Apply the Collins decomposition technique to E^{d+1} relative to the set of polynomials defining FP, so that t is the innermost dimension to be processed. Also find the adjacency relationship between the Collins cells, using the technique described in Schwartz and Sharir [19].

2. Construct a *connectivity graph* CG, which is a *directed* graph defined as follows. The nodes of CG are the free Collins cells. A directed edge $[c, c']$ connects two free cells c and c' provided that (a) c and c' are adjacent; (b) either c and c' both project onto the same base segment on the t axis, or c projects onto an open t segment $(t_0(c), t_1(c))$ and c' projects onto its upper endpoint $t_1(c)$, or

c' projects onto an open t segment $(t_0(c'), t_1(c'))$ and c projects onto its lower endpoint $t_0(c')$. Intuitively, each edge of CG represents a crossing between two adjacent cells which is either stationary in time (crossing in a direction orthogonal to t) or else progresses forward in time.

3. Find the cells c_0, c_1 containing respectively the initial and final configurations $[X_0, 0]$, $[X_1, T]$. Then search for a directed path in CG from c_0 to c_1. If there exists such a path then there also exists a motion in FP between the two given configurations (and the latter motion can be effectively constructed from the path in CG); otherwise no such motion exists.

To see that the procedure just described is correct, note first that if p is a continuous motion through FP between the initial and final configurations (which we assume to cross between Collins cells only finitely many times), then it is easily seen that the sequence of free cells through which p passes constitutes a directed path in CG. Conversely, if p' is a directed path in CG between c_0 and c_1, then p' can be transformed into a continuous (t–monotone) motion through FP as follows: First choose for each free Collins cell c a representative interior point $[X_c, t_c]$, such that the representative points of all the cells that project onto the same base segment on the t-axis have the same t value. Then transform each edge $[c, c']$ of p' into a monotone path in FP as follows. If $t_c = t_{c'}$ (i.e. if the crossing from c to c' is orthogonal to the time axis), then connect $[X_c, t_c]$ to $[X_{c'}, t_{c'}]$ by any path which is contained in the union $c \cup c'$ and on which t is held constant; the existence of such a path is guaranteed by the property of Collins cells noted above. If $t_c < t_{c'}$ then connect $[X_c, t_c]$

to $[X_{c'}, t_{c'}]$ by a t–monotone path contained in $c \cup c'$; again the existence of such a path is guaranteed by the structure of Collins cells. The resulting path p is plainly continuous, is contained in FP and is weakly monotone in t. Note that the crossings of the first type in which t remains constant represent extreme situations where the velocity of B is infinite. However, since p is continuous and FP is open, one can easily modify p slightly so as to make it strictly monotone in time, provided, of course, that the starting time of p is strictly smaller than its ending time, a condition that can be checked and excluded ahead of time. This establishes the correctness of our procedure.

Since the Collins decomposition is of size polynomial in the number of given polynomials and in their maximum degree (albeit doubly exponential in the number of degrees of freedom d), and can be computed within time of similar polynomial complexity, it follows that:

Theorem 4.2 *The dynamic unrestricted version of the movers problem for a fixed moving B can be solved in the general (space–time) algebraic case in time polynomial in the number of obstacle features, the number of parts of B, and their maximum algebraic degree.*

Acknowledgements

We would like to that Michael Ben-Or, John Cock, John Hopcroft and Jack Schwartz for insightful comments on dynamic motion planning. Thanks for K. Sutner for suggesting the inclusion of minimum–time dynamic movement problems in Section

3. Also thanks to Christoph Freytag and S. Rajasekaran, as well as the anonymous referees, for a careful reading of the earlier version of this manuscript.

References

[1] Ben-Or, M., Kozen, D., and Reif, J.H., The Complexity of Elementary Algebra and Geometry, *J. Computer and Systems Sciences* 32 (1986), pp. 251–264.

[2] Canny, J., Some algebraic and geometric computations in PSPACE, *Proc. 20th ACM Symp. on Theory of Computing,* 1988, pp. 460–467.

[3] Canny, J., and Reif, J., New lower bound techniques for robot motion planning problems, *Proc. 28th IEEE Symp. on Foundations of Computer Science,* 1987, pp. 49–60.

[4] Collins, G., Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition, in: *Second GI Conference on Automata Theory and Formal Languages,* Lecture Notes in Computer Science, 33, Springer Verlag, Berlin, 1975, pp. 134–183.

[5] Cooke, G.E. and Finney, R.R.L., *Homology of Cell Complexes,* Mathematical Notes, Princeton University Press, Princeton, NJ, 1967.

[6] Hopcroft, J., Joseph, D., and Whitesides, S., Movement Problems for 2-dimensional Linkages, *SIAM J. Computing* 13(1984) pp. 610–629.

[7] Hopcroft, J., Joseph, D., and Whitesides, S., On the Movement of Robot Arm in 2-dimensional Bounded Regions, *SIAM J. Computing* 14(1985), pp. 315–333.

[8] Hopcroft, J.E., Schwartz, J.T., and Sharir, M., On the Complexity of Motion Planning for Multiple Independent Objects; PSPACE-Hardness of the 'Warehouseman's Problem', *Int. J. Robstics Research* 3 (4) (1984) pp. 76–88.

[9] Hopcroft, J.E., Schwartz, J.T., and Sharir, M. (Eds.), *Planning, Geometry, and Complexity of Robot Motion,* Ablex Pub. Co., Norwood, NJ, 1987.

[10] Kedem, K., Livne, R., Pach, j. and Sharir, M., On the union of Jordan regions and collision–free translational motion amidst polygonal obstacles, *Discrete Comput. Geom.* 1 (1986), pp. 59–71.

[11] Lozano-Perez, T. and Wesley, M.A., An algorithm for Planning Collision-free Paths Among Polyhedral Obstacles, *Comm. ACM* 22(1979), pp. 560–570.

[12] Ó'Dúnlaing, C., Sharir, M., and Yap, C.K., Retraction: A New Approach to Motion-planning, *Proc. 15th ACM Symp. on Theory of Computing,* Boston, MA, 1983, pp. 207–220.

[13] Ó'Dúnlaing, C. and Yap, C.K., A 'Retraction' Method for Planning the Motion of a Disc, *J. Algorithms* 6 (1985) pp. 104–111.

[14] Reif, J., Complexity of the Mover's Problem and Generalizations, *Proc. 20th IEEE Symp. on Foundations of Computer Science,* Puerto Rico, October 1979, pp. 421-427. (Also in *Planning, Geometry, and Complexity of Robot Motion,* J.

Hopcroft, J. Schwartz, M. Sharir eds., Ablex Pub. Co., Norwood, NJ, 1987, pp. 267–281.

[15] Reif, J.H., and Sharir, M., Motion Planning in the Presence of Moving Obstacles, *Proc. 26th IEEE Symp. on Foundations of Computer Science,* 1985, pp. 144–154.

[16] Reif, J.H., and Storer, J.A., Shortest Paths in Euclidean Space with Polyhedral Obstacles, TR-05-85, Aiken Computation Lab, Harvard University, Cambridge, MA, April 1985.

[17] Renegar, J., On the computational complexity and geometry of the first order theory of the reals, Part I, *J. Symbolic Computation* 13 (1992), pp. 255–299.

[18] Schwartz, J.T. and Sharir, M., On the 'Piano Movers' Problem. I. The Case of a Two-dimensional rigid Polygonal Body Moving Amidst Polygonal Barriers, *Comm. Pure Appl. Math.* 36 (1983), pp. 345–398.

[19] Schwartz, J.T. and Sharir, M., On the 'Piano Movers' Problem. II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds, *Adv. Applied Math.* 4 (1983), pp. 298–351.

[20] Schwartz, J.T. and Sharir, M., On the 'Piano Movers' Problem. III. Coordinating the Motion of Several Independent Bodies: The Special Case of Circular Bodies Moving Amidst Polygonal Barriers, *Int. J. Robotics Research* 2 (3) (1983), pp. 46–75.

[21] Schwartz, J.T. and Sharir, M., On the 'Piano Movers' Problem. V. The Case of a Rod Moving in 3-D Space Amidst Polyhedral Obstacles, *Comm. Pure Appl. Math.* 37 (1984), pp. 817–846.

[22] Sharir, M., Algorithmic motion planning in robotics, *Computer* 22 (1989), pp. 9–20.

[23] Sutner, K. and Maass, W., Motion planning among time–dependent obstacles, *Acta Informatica* 26 (1988), pp. 93–122.