Robot Motion Planning and the Single Cell Problem in Arrangements

Dan Halperin*

Abstract

Robot motion planning has become a central topic in robotics and has been studied using a variety of techniques. One approach, followed mainly in computational geometry, aims to develop combinatorial, non-heuristic solutions to motion-planning problems. This direction is strongly related to the study of arrangements of algebraic curves and surfaces in low-dimensional Euclidean space. More specifically, the motion-planning problem can be reduced to the problem of efficiently constructing a single cell in an arrangement of curves or surfaces. We present the basic terminology and the underlying ideas of the approach. We describe past work and then survey a series of recent results in exact motion planning with three degrees of freedom and the related issues of the complexity and construction of a single cell in certain arrangements of surface patches in three-dimensional space.

1 Introduction

Future generations of robots are expected to operate in remote and dangerous places like outer space, underseas, hazardous waste sites, and are therefore anticipated to be far more autonomous than today's existing robots. The ability of a robot to plan its own motion seems pivotal to its autonomy. For over more than a decade, robot motion planning has attracted much research in various fields and has become a central topic in robotics. Although today the title *robot motion planning* is considered to cover a wide variety of problems, we will use it for the problem of planning collision-free motion for a robot moving among obstacles.

The basic motion-planning problem, sometimes referred to as *the piano movers' problem*, is defined as follows:

> Let B be a robot system having k degrees of freedom and free to move within a two- or three-dimensional domain V which is bounded by various obstacles whose geometry is known to the system. The motion-planning problem for B is, given the initial and desired final placements of the system B, to determine whether there exists a continuous motion from the initial placement to the final one, during which B avoids collision with the known obstacles, and if so, to plan such a motion.

*Department of Computer Science, Tel-Aviv University. Author's current address: Robotics Laboratory, Department of Computer Science, Stanford University.

In this pure formulation of the problem, we are only interested in the geometric aspects of the motion. We ignore many issues, such as acceleration, speed, uncertainty or incompleteness in the geometric data, control strategies for executing the motion, etc. A comprehensive overview of problems and techniques in robot motion planning can be found in a recent book by Latombe [La]. Several surveys on the topic have also been published, e.g., [Sh89], [Ya].

A special approach to solving motion-planning problems, called *exact motion planning*, is followed by computer scientists, mainly in the field of computational geometry. The approach is exact (non-heuristic) in the sense that it aims to find a solution whenever one exists and otherwise report that no solution exists. In this framework, the quality of an algorithm that solves a motion-planning problem is measured by its asymptotic running time and storage requirements, which in turn are measured as functions of the combinatorial complexity of the problem input, e.g., number of obstacle features and robot features, the cost of the (algebraic) representation of each feature, etc. There are other criteria to evaluate the performance of motion-planning algorithms, for example, the clearance between the path produced for the robot motion and the obstacles (larger clearance meaning safer motion in the presence of uncertainty), or the length or other cost of the generated path. However, in this survey we will confine ourselves to the most prevailing criteria in theoretical computer science, namely the time and space requirements of the algorithm.

Much of the study of exact motion planning is carried out in the *configuration space* of the problem. The configuration space of a motion-planning problem with k degrees of freedom is k-dimensional and every point in it represents a possible placement of the robot in the physical space. We distinguish three types of points in the configuration space of a specific problem according to the placements of the robot that they represent: *free placements,* where the robot does not intersect any obstacle, *forbidden placements,* where the robot intersects the interior of an obstacle, and *semi-free placements,* where the robot is in contact with the boundary of an obstacle, but does not intersect the interior of any obstacle.

The collection of all the points in the configuration space that represent semi-free robot placements partitions the configuration space into *free regions* and *forbidden regions.* In a motion-planning problem with two degrees of freedom, for instance, the points representing semi-free placements of the robot lie on several curves in a two-dimensional configuration space. We will call these curves *constraint curves*. Note that each such curve is induced by the contact of a robot feature and an obstacle feature. We are therefore interested in studying the partitioning of 2D space by a collection of constraint curves (for motion-planning with two degrees of freedom), and similarly the partitioning of a three-dimensional configuration space by constraint surfaces (for motion-planning problems with three degrees of freedom). (The entire discussion applies to problems with more degrees of freedom as well, but this survey only deals with problems with two or three degrees of freedom.) This is the point where robot motion planning overlaps a basic study in computational geometry, namely, the study of the combinatorial structure of *arrangements* of algebraic curves or surfaces in low-dimensional Euclidean spaces.

The connection between robot motion planning and the study of arrangements has been noted by many authors. See, e.g., [Ca], [GS]. This connection, as well as most of the above terminology, are made concrete by examples and figures in Section 2 below. However, in order to present the most basic terminology on arrangements, let us consider the following special case of an arrangement induced by a collection of n lines in the plane. An arrangement of lines partitions the plane into vertices, edges and faces. A vertex is an intersection point of two lines, an edge is a maximal connected portion of a line that does not meet any vertex, and a face is a maximal connected region of the plane not meeting any edge or vertex. If we assume that the lines are in general position, namely, no two are parallel and no more than two meet at a single point, then it can be shown that the number of vertices (as well as the number of edges or faces) in the arrangement is $\Theta(n^2)$. We will be concerned in particular with the complexity of a single face in an arrangement, namely, with the number of vertices and edges along the boundary of one face in the arrangement. In an arrangement of n lines, for example, the maximum complexity of one face is $O(n)$ because any face in the arrangement is convex and a line cannot contribute more than one edge to a face. Similarly, in three dimensions, we may consider an arrangement of n planes, which partitions space into vertices, edges, faces and cells. Here too the complexity of a single cell is only $O(n)$, since such a cell is convex and bounded by at most n planes. (See [Ed] for detailed discussions on arrangements of lines and of hyperplanes in higher dimensional spaces.)

Our interest in a single face, or in a single three-dimensional cell in arrangements of surfaces in 3-space, comes from the following simple observation. If we are given a motion-planning problem with say, two degrees of freedom and we already know the constraint curves that it induces in the configuration space, and we are given an initial free placement, z_0, of the robot, then it suffices to know the one face f_0 of the arrangement that contains z_0 in order to solve the motion-planning problem. This is true, because the boundary of f_0 consists of constraint curves and getting out of the face, that is, crossing a constraint curve, would mean to go from a free placement to a forbidden placement, which is impossible. It will be shown throughout the paper, that a single cell in an arrangement is usually significantly less complex than the entire arrangement. This in turn, will open the way to obtaining efficient solutions to certain motion-planning problems.

Note that it might be the case that even a single cell is more than is needed in order to solve a motion-planning problem efficiently. However, the author is not aware of an exact solution to a non-trivial motion-planning problem that computes less than a single cell of the configuration space.

The heart of this survey is a report on a series of recent results in motion planning with three degrees of freedom. The corresponding arrangements of constraint surfaces may have complexity $\Theta(n^3)$ in the worst case, where n measures the combinatorial complexity of the input data; in fact, just the free cells of these arrangements may have $\Theta(n^3)$ complexity in the worst case. The goal in these studies is to show that the complexity of a single cell is substantially subcubic and ideally near-quadratic (in most cases, a lower bound $\Omega(n^2)$ is known). We are only interested in the free portions of the configuration space, those representing free placements of the robot. There are favorable motion-planning problems with three degrees of freedom where the entire free space is known to have near-quadratic complexity (see, e.g., [LS], [KS]). We will not discuss this type of problems and restrict ourselves to problems where the entire free space can have cubic complexity in the worst case.

The study of arrangements related to motion-planning problems has the following characteristics. The objects defining the partitioning of space are not unbounded, e.g., in Section 2 we will consider an arrangement of line segments rather than an arrangement of full lines. In 3-space we will be concerned with *surface patches* (2-manifolds with boundary) rather than with surfaces. Furthermore, in motion-planning problems that involve rotation, the induced constraint surfaces are non-flat. This proves to be a major difficulty as there is a large gap between the current fairly vast knowledge in computational geometry on the combinatorial structure of arrangements of flat objects (planes, hyperplanes, triangles) and the somewhat restricted knowledge on arrangements of non-flat objects, especially in three and higher dimensions.

In the study of motion planning it is common to distinguish between two types of robots: An *anchored* robot and a *free-flying* robot. An anchored robot, like most existing industrial robot manipulators, is a collection of links hinged together, with one link fixed in space (e.g., the base link of a robot arm is fixed to the shop floor). A free-flying robot has no fixed point and has the potential to be anywhere in the workspace (this type is sometimes called a *mobile robot*). We will observe below that for robots with three degrees of freedom in the plane, although there is similarity between the arrangements of constraint surfaces that both types induce, the arrangements related to the motion of anchored robot arms lend themselves more easily to the type of analysis that we carry out than the arrangements related to the motion of free-flying robots.

From the algorithmic point of view, we distinguish between two types of motion-planning problems: The *reachability problem,* which is to decide whether a continuous collision-free path from the initial placement to the final placement exists, and the *find-path problem,* which is to actually compute the path if it exists.

Finally, in the study of arrangements there are two topics closely related to the single cell problem—the *zone* problem and the *lower envelope* problem. In the next section, we provide definitions for the two-dimensional instances of these concepts. However, we concentrate on the single cell problem which has the tightest link to motion planning. For zones consider, e.g., [Ed], [APS], [EGPPSS], [dBvKS]. For lower envelopes consider, e.g., [Sh88], [PaS], [WS], [He]. (Both topics are addressed in many other papers.)

The paper is organized as follows. In Section 2 we discuss motion-planning problems with two degrees of freedom and the single face problem. Section 3 is devoted to intermediate results on motion planning with three degrees of freedom, presenting subcubic result on the complexity of a single cell in certain arrangements of surfaces in 3-space that are still substantially super quadratic. In Section 4 we survey a series of recent results obtaining near-quadratic bounds and algorithms with near-quadratic running time for the single cell problem in certain arrangements of surfaces and the corresponding motion-planning problems. We conclude with a summary and presentation of open problems in Section 5.

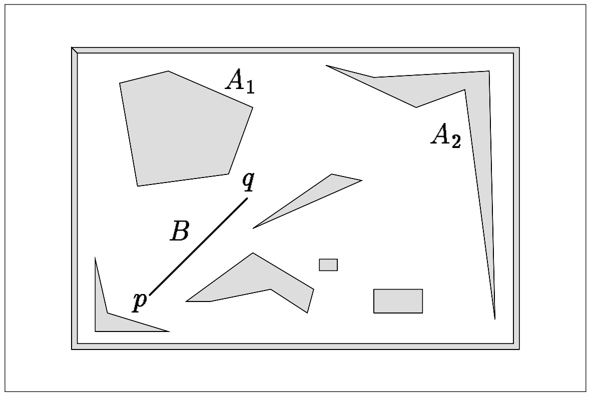

Figure 1: A rod translating among polygonal obstacles

2 Motion Planning with Two Degrees of Freedom and the Single Face Problem

We start our discussion of the simpler, two-dimensional, case with a concrete example of a motion-planning problem with two degrees of freedom and the corresponding arrangement of constraint curves. The moving body, the robot, is a line segment (a "rod") translating but not rotating in the plane among polygonal obstacles inside a rectangular room. See Figure 1. A rigid body translating in the plane has two degrees of freedom: Every possible placement of the robot can be designated by the coordinates of the placement in the plane of a fixed reference point on the robot. We denote the moving rod by B and its two endpoints by p and q. The rod is moving among polygons having a total of n edges and vertices.

We choose p to be the reference point on B and every possible placement of B is designated by the x and y coordinates of p. The configuration space E is two-dimensional and every point in it represents a placement of the endpoint p of B (and thus of B itself). Note that in this special case, both the physical space and the configuration space have the same coordinates—this will not be the case in general, as we will show in the following sections. Figure 2 shows the constraint curve (the dashed line segment) induced by the contact of the endpoint q of B and one obstacle edge.

Our first goal is to partition the configuration space into regions of *free placements* and of *forbidden placements*, where a free placement is a point of E that represents a placement of p such that the robot B does not touch any obstacle. A forbidden placement is a placement of p such that the robot intersects the interior of an obstacle. Recall also that we refer to placements where B is in contact with the boundary of an obstacle (but does not otherwise penetrate into an obstacle) as *semi-free* placements. In the configuration space, the loci of the semi-free placements due to the contact of a feature of the robot and a feature of an obstacle is called a *constraint curve*. If we "draw" in the configuration plane the constraint curves induced by all possible contacts between

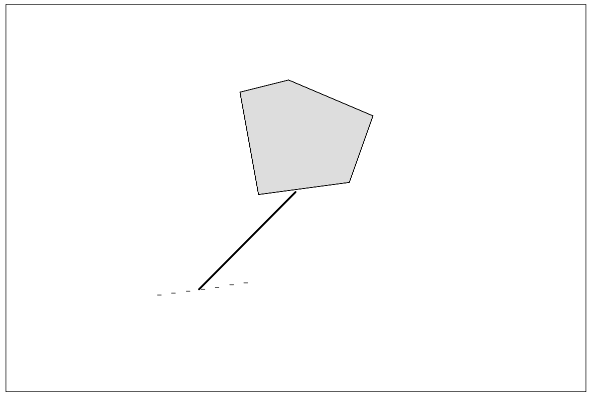

Figure 2: A constraint curve

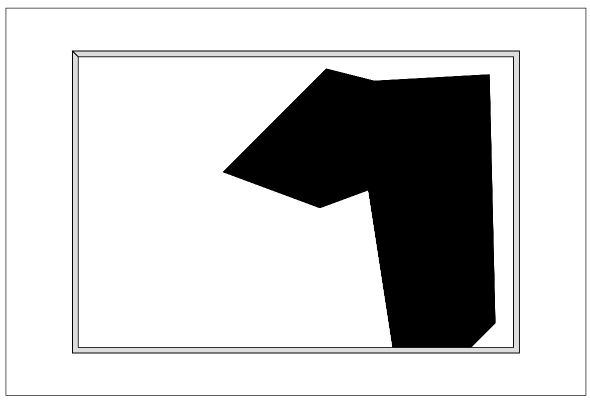

Figure 3: Forbidden placements in the configuration space

any robot feature and any obstacle feature, we obtain a decomposition of the plane into faces. The boundary of each face consists of portions of constraint curves. Inside a face we can move from one point to another without crossing a curve, therefore the interior of each face consists either entirely of free placements or entirely of forbidden placements, because to get from a free placement to a forbidden placement we must cross a constraint curve. Figure 3 shows the forbidden region of the configuration space for the rod B, induced by the obstacle A_2 of Figure 1.

If we are given an initial placement z_0 of B and we consider the face f_0 of the partitioned configuration plane that contains z_0, then it is evident that the robot cannot get out of this face. Our approach to exact motion planning is based on this observation, namely, for motion-planning purposes it suffices to know the single face in the arrangement of constraint curves, which contains the initial placement of the robot. Thus, we wish to have a discrete representation of a single face in a planar arrangement of curves and we concern ourselves with the following issues:

- How complex can a single face in an arrangement of constraint curves be.
- How efficiently can such a face be computed in a manner that will enable us to answer a motion-planning query efficiently.

Any reasonable motion-planning problem with two degrees of freedom can be described in terms of a two-dimensional arrangement of "well-behaving" constraint curves. These curves need not be straight line segments as in the example above but they must have a reasonably simple shape in order to obtain good bounds for the combinatorial and algorithmic problems listed above. For simplicity of exposition, we assume that the motion-planning problems we are discussing induce constraint curves that are all algebraic of bounded degree. This is a common assumption in motion planning and is not too restrictive. Clearly, any pair of such curves do not intersect more than s times, for a constant s depending on the maximum degree of the curves. Suppose then that we are given an arrangement of n curves in the plane, where any pair of curves do not intersect more than some constant number of times. The complexity of a planar arrangement of curves is predominated by the number of its vertices, and therefore by the number of intersections of pairs of curves. There are $\binom{n}{2}$ pairs, each contributing at most a constant number of intersection points, so the complexity of the entire arrangement is at most $O(n^2)$. Since, in general, any pair of curves may intersect, this bound can be achieved and hence the actual complexity of the entire arrangement can be $\Theta(n^2)$.

The major result that we review in this section, and a few variants of it, say that although the entire arrangement may have quadratic complexity, the complexity of a single face is only near-linear and it can be computed by an algorithm with running time that is also close to linear.

The main tool in obtaining these results is *Davenport-Schinzel sequences*, a combinatorial tool that has found many applications in computational geometry. It is beyond the scope of this introduction to give a detailed review of Davenport-Schinzel sequences. We briefly present the basic definitions related to the topic and refer the reader to the survey paper [Sh88]; many of the other papers cited in our survey also make use of Davenport-Schinzel sequences. Let n and s be positive integers. A sequence $U = (u_1, u_2, \ldots, u_m)$ composed of n distinct symbols is a *Davenport-Schinzel* sequence of order s (an (n, s)-DS sequence for short), if it satisfies the following two conditions:

(i) $\forall i < m, u_i \neq u_{i+1}$.

(ii) There do not exist $s + 2$ indices $1 \leq i_1 < i_2 < \ldots < i_{s+2} \leq m$ such that

$$\begin{array}{ccccccc} u_{i_1} & = & u_{i_3} & = & \ldots & = & a \\ u_{i_2} & = & u_{i_4} & = & \ldots & = & b \end{array}$$

and $a \neq b$.

Let $\lambda_s(n)$ denote the maximum length of an (n, s)-DS sequence. It is easy to see that $\lambda_1(n) = n$ and that $\lambda_2(n) = 2n - 1$. Hart and Sharir [HarS] have shown that $\lambda_3(n) = \Theta(n\alpha(n))$. Here $\alpha(n)$ is the extremely slowly growing functional inverse of Ackermann's function, which for all practical purposes can be regarded as a constant. Still, from a theoretical point of view, $\lambda_3(n)$ grows faster than any linear function of n. The best bounds known to date for $\lambda_s(n)$ with $s > 3$ were obtained by Agarwal et al. [ASS] and they are all almost linear in n, as well. In short, for any fixed s, $\lambda_s(n)$ is an almost linear function of n.

Before proceeding to quote the single face result, we define two entities that are closely related to the notion of a single face: lower envelopes and zones.

Definition 2.1 [Lower envelope] Given a collection Γ of n x-monotone curves $\gamma_1, \gamma_2, \ldots, \gamma_n$ where each curve γ_i can be viewed as the graph of a partially-defined function $y = \gamma_i(x)$, the *lower envelope* Ψ of Γ is the graph of the pointwise minimum of these functions, that is, of the function

$$\Psi(x) = \min_i \gamma_i(x).$$

The *combinatorial complexity of the lower envelope* Ψ is defined as the number of maximal connected pieces of the curves in Γ that appear along Ψ.

Definition 2.2 [Zone] Given a collection Γ of n curves defining an arrangement $\mathcal{A}(\Gamma)$ in the plane, the *zone* of an additional curve δ in $\mathcal{A}(\Gamma)$ is the collection of faces of $\mathcal{A}(\Gamma)$ intersected by δ. The *combinatorial complexity of the zone* of δ in $\mathcal{A}(\Gamma)$ is defined as the sum of the complexities of the faces in the zone.

Guibas et al. [GSS] have obtained the following result:

Theorem 2.3 [GSS] *The complexity of a single face in an arrangement of* n *Jordan arcs, having the property that no two of them intersect more than* s *times, is* $O(\lambda_{s+2}(n))$ *and it can be calculated in time* $O(\lambda_{s+2}(n) \log^2 n)$.

This result have settled, to a fairly general extent, the motion-planning problem with two degrees of freedom. This result, as well as most of the algorithmic results reported in this survey, assume the standard model of computation in *computational geometry,* namely, the algorithms use infinite-precision real numbers and the cost of each arithmetic operation (or any other simple operation) is fixed. See, e.g., [PrS, Section 1.4].

We mention several variants of this result, some of which constitute slight improvements over the bounds stated above. This result was preceded by a result by Pollack at al. [PSS] for the special case of line segments. The algorithm for line segments is also discussed in [EGS]. Schwartz and Sharir [SS] have shown that if the given Jordan arcs are unbounded or closed then the single cell complexity reduces to $O(\lambda_s(n))$. From the algorithmic point of view, Chazelle et al. [CEGSS] have devised a randomized algorithm with running time $O(\lambda_{s+2}(n) \log n)$ to compute a single cell in an arrangement of the type mentioned above, i.e., using randomization they are able to shave a log factor off the time bound of the deterministic algorithm; a similar result has also been obtained by Clarkson [Cl]. Alevizos et al. [ABP] have studied the case of a single face in an arrangement of n rays—they give a linear bound on its complexity and an optimal algorithm with running time $O(n \log n)$ for computing a single face in such an arrangement.

The proofs of all the above combinatorial results follow a similar line: Every connected component of the boundary of the face is handled separately. For a fixed boundary component, one chooses a start point on the boundary, walks along the boundary and records the labels of the curves met along the walk. Certain precaution measures should be taken in labeling the arcs, e.g., each side of an arc is given a distinct label and also the cyclicity around the boundary should be resolved. Once these precautions are taken, the resulting sequence is shown to be a certain Davenport-Schinzel sequence with the alphabet of the sequence corresponding to the extended set of curve labels. Another important ingredient in the proofs is a certain *consistency lemma* that relates the order of contributions of one curve to the boundary of a face with their order along the curve itself.

The lower envelope problem in two-dimensional arrangements of curves is, in fact, a special (rather simple) case of the single face problem in such arrangements. The connection between lower envelopes and Davenport-Schinzel sequences is immediate, and the results for the lower envelope problem predated those for the single face problem. The maximum complexity of the lower envelope of n x-monotone arcs, each pair intersecting in at most s points, is $O(\lambda_{s+2}(n))$ [HarS], and the envelope can be computed in time $O(\lambda_{s+1}(n) \log n)$ [He]. For a survey of results for lower envelopes (or more generally, Davenport-Schinzel sequences) and their applications, see [Sh88].

A linear bound on the zone complexity of a line in an arrangement of n lines has been obtained in [EOS], [CGL] (see also [Ed, Chapter 5]). Later, Edelsbrunner at al. [EGPPSS] have shown that the *zone* problem in a two-dimensional arrangement of curves, can be transformed into the *single face* problem in another arrangement which is a slightly modified version of the original arrangement. This result, which relies on the result of [GSS], implies that the complexity of the zone of a "well behaving" curve in an arrangement of n other curves, each pair of which intersect in at most s points, is also $O(\lambda_{s+2}(n))$, and thus settles, in a fairly general fashion, the zone problem in two dimensions.

So far for the two-dimensional case. The following sections deal with motion planning with three degrees of freedom and arrangements of surfaces in three-dimensional space.

3 An Intermediate 3D Cell Complex: The "Interesting" Cells

This paper is mainly concerned with motion planning with three degrees of freedom. Consider, for example, the motion-planning problem for a rod B presented in the previous section. If we let B translate and rotate, then this becomes a motion-planning problem with three degrees of freedom. Now every placement of B can be determined, for example, by the coordinates of the endpoint p and the angle between the segment $\overline{pq}$ and the positive x direction. We will denote by θ the value of this latter rotational degree of freedom of a rigid body moving in the plane.

For motion-planning problems with three degrees of freedom the configuration space is three dimensional, and the contact between a robot feature and an obstacle feature induces a (two-dimensional) *constraint surface* rather than a constraint curve. More precisely, such a contact usually induces a constraint *surface patch*, a 2-manifold with boundary. If we assume that the robot complexity is constant, i.e., that the number of robot features is bounded by some constant, and that the total number of obstacle features is n, then we get a collection of $O(n)$ surface patches in 3-space. By the same argument that we have employed in the two-dimensional case, if we are given a point z_0 in the configuration space, designating the initial placement of the robot, then we are only interested in the one cell of the arrangement of constraint surfaces that contains z_0. Hence, we are interested in the complexity and efficient construction of a single cell in an arrangement of surface patches in 3-space.

For two-dimensional instances of the single cell problem we have used Davenport-Schinzel sequences. For three-dimensional instances it would be desirable to have a generalization of that theory, that will enable to reduce the original combinatorial geometry problem into a purely combinatorial one. Unfortunately, such a generalization has not been established yet.

It is well known that the maximum complexity of the entire arrangement of n low-degree algebraic surface patches in 3-space is in $\Theta(n^3)$. It has long been conjectured that, in analogy with the 2D single face problem, the maximum complexity of a single cell in an arrangement of surface patches should be an order of magnitude smaller than that of the entire arrangement, that is, it should be roughly quadratic. Even for the simpler case of the lower envelope of such surface patches, no general subcubic bounds are known for its complexity, and the same conjecture, that the complexity of such an envelope should be near-quadratic, is also open in general. Originally, the challenge was to obtain any subcubic bound on the complexity of a single cell. A breakthrough in this direction was obtained by Aronov and Sharir [ArS90] who have identified a cell complex of a certain three-dimensional arrangement, whose complexity is significantly smaller than the complexity of the entire arrangement, and at the same time serves as an upper bound on the complexity of a single cell in that arrangement. Before we present this arrangement we introduce a certain terminology central to this section:

Definition 3.1 Let S be a collection of n surface patches in 3-space and let $\mathcal{A}(S)$ be the arrangement that they induce. We distinguish two types of 3D cells in $\mathcal{A}(S)$: (i) a cell whose boundary contains a portion of the 1D boundary of a surface patch, which we will refer to as an **interesting** cell; and (ii) any other cell, that is, any cell whose boundary does not contain a portion of the 1D boundary of any surface patch, which we will refer to as a **dull** cell. [1]

This choice of terms is clarified in the coming subsection.

3.1 Arrangements of Triangles in Space

The arrangement studied in [ArS90] is an arrangement of triangles in space. Note that an arrangement of triangles faithfully represents the following motion-planning problem: A three-dimensional

[1] This terminology has first appeared in [ArS90].

polyhedral robot translating (without rotating) in space among polyhedral obstacles. In this problem, each constraint surface is induced by a robot-obstacle contact of either a robot vertex against an obstacle face, or a robot edge against an obstacle edge, or a robot face against an obstacle vertex. In either case the resulting surface is a flat polygon in space, and this polygon can be decomposed into triangles.

Let S be a collection of n triangles in space. Let us interpret Definition 3.1 for the special arrangement $\mathcal{A}(S)$: A cell whose boundary contains a portion of a triangle edge is necessarily non-convex.[2] Similarly, a cell whose boundary does not contain a portion of a triangle edge is necessarily convex. A triangle cannot contribute more than one face to the boundary of a convex cell, therefore the maximum complexity of a single convex cell is at most $O(n)$, which is an explanation for the term *dull*. We will not concern ourselves with dull cells as we have a linear bound on their complexity; we are only interested in the complexity of interesting cells. The complexity of one cell in an arrangement of triangles may be $\Omega(n^2\alpha(n))$ in the worst case (this is a rather straightforward extension to three dimensions of a lower bound for the complexity of the lower envelope of segments in the plane by Wiernik and Sharir [WS]; we omit the details here). Aronov and Sharir ([ArS90], see also [AA]) have shown that the overall complexity of all the interesting cells in an arrangement of triangles is $O(n^{7/3})$ and that this bound is tight in the worst case. Evidently, this bound serves as an upper bound on the complexity of any single cell.

In [ArS90] the authors also devise a randomized algorithm to compute a single cell in an arrangement of triangles. We will return to this algorithm in Subsection 4.3.

When one or more degrees of freedom of a robot is rotational, the corresponding constraint surface patches are no longer flat. The corresponding arrangements become rather convolute and exhibit a collection of new problems. In the next subsection we start exploring the behavior of a special arrangement of non-flat surface patches.

3.2 A Reachability Algorithm for an L-Shaped Object

Consider the following motion-planning problem: Let L be an L-shaped moving object, translating and rotating in the plane among a set of n point obstacles. We are given an initial free placement z_0 and a final placement z_1 of the robot L and we are asked whether there is a collision-free continuous path for L to move from z_0 to z_1. We call such a problem a *reachability problem* to distinguish from a *find-path* problem which is to compute the path if it exists. The corresponding arrangement of constraint surfaces in the configuration space may have complexity $\Theta(n^3)$ in the worst case and a single cell may have complexity $\Omega(n^2)$ in the worst case; see [HS92a] for details. This time the induced surfaces are no longer flat. The configuration space where we present the surfaces has coordinates x, y and θ where x and y are the coordinates of the internal vertex q of L and θ is the angle formed by the horizontal bar $\overline{qp}$ of L and the positive x direction. See Figure 4.

If the moving body were convex then the problem would become "favorable" in the sense that

[2]The triangles are assumed to be in "general position" (see, e.g., [ArS90],[Ha92]). This implies that a pair of edges belonging to distinct triangles do not overlap.

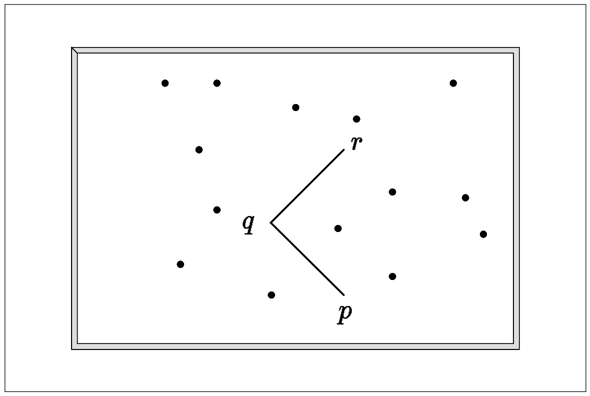

Figure 4: An L-shaped object

the entire free space would have complexity that is roughly quadratic. The L-shaped object is perhaps the simplest rigid moving body for which the motion-planning problem induces a free space with complexity that can be cubic in the worst case.

In [HOS], we have obtained an algorithm with running time $O(n^2 \log^2 n)$ to solve the reachability problem for an L-shaped object moving among n point obstacles in the plane. This has been the first near-quadratic time algorithm for a motion-planning problem whose entire arrangement of constraint surfaces may have cubic complexity. The basic observation in [HOS] is that there may indeed be $\Theta(n^3)$ features in the arrangement induced by the L motion-planning problem (L-arrangement for short) but they are organized in at most $O(n^2)$ clusters, or "events". Consequently, for example, all the vertices of the arrangement (whose number may be $\Theta(n^3)$) share at most $O(n^2)$ distinct θ values.

This observation has opened the way to devising an algorithm that builds a succinct representation of the free space, skipping details but still keeping sufficient information to answer reachability queries. This algorithm employs a battery of data structures that allows to encode the essential information about one event in polylogarithmic time even though the event may involve up to $\Omega(n)$ vertices of the arrangement.

Rephrased purely in terms of arrangements, our reachability algorithm is a decision procedure that accepts a certain collection of surface patches and two points in three-dimensional space and decides whether the two points lie in the same connected component of the partitioning of space determined by these surface patches. The connection between this algorithm and interesting cells is that the algorithm ignores most of the dull cells as it sweeps the arrangement and considers more carefully only the interesting cells, while recording partial information about them.

In [HOS] we remark that in a certain pragmatic setting, where the robot is equipped with tactile sensors, our algorithm records enough information to plan a path for the robot from a source placement to a destination placement. However, in the standard, "classical" definition of the piano movers' problem, it still falls short of producing the path when one exists. A first step in overcoming this failing is shown in the next subsection, where we obtain a subcubic bound on the complexity of all the interesting cells in two types of motion-planning arrangements, one of which is the L-arrangement, and devise a find-path algorithm for an L-shaped object with subcubic running time.

3.3 Improved Combinatorial Bounds and a Find-Path Algorithm for an L-Shaped Object

Whereas in [HOS] we have concentrated on the algorithmic aspect of the L-shaped object motion-planning problem, in [HS92a] we have established improved bounds on the complexity of a single cell in the L-arrangement and in another arrangement related to the motion of a certain robot arm with three degrees of freedom. Here again, our bounds and algorithms relate to the complex of all the interesting cells.

The main result in [HS92a] is an upper bound $O(n^{5/2})$ on the complexity of all the interesting cells in an L-arrangement. It is interesting to note that although a dull cell in an L-arrangement is no longer convex, its maximum complexity is nevertheless $O(n)$, hence the bound on the complexity of all the interesting cells serves as an upper bound on the complexity of a single cell. In [HS92a] we suggest a certain reduction of the original three-dimensional problem into a two-dimensional problem. The two-dimensional problem being that of bounding the joint combinatorial complexity of m *concave chains* in an arrangement of n pseudo lines. We refer the reader to [HS91], [HS92a] for a definition and various results on concave chains in arrangements of pseudo lines.

We then apply the same analysis to another motion-planning problem, to that of moving a so-called telescopic arm among polygonal obstacles in the plane. The telescopic arm (TA, for short) consists of two links, $\overline{op}$ and $\overline{pq}$; see Figure 5. o is an anchor point. The first link $\overline{op}$ is a telescopic link which can rotate around o, and extend or shrink along its length. The second link $\overline{pq}$ has a fixed length d, and can rotate around p. This system was previously studied by Aronov and Ó'Dúnlaing [AO] who showed that the configuration space of this arm moving among polygonal obstacles has $\Theta(n^3)$ connected components in the worst case, and obtained an $O(n^3 \log n)$-time and $O(n^3)$-space algorithm to compute it.

Applying the analysis of an L-arrangement to a TA-arrangement we obtain the same bound, $O(n^{5/2})$, on the complexity of all the interesting cells in a TA-arrangement. But then we show that a better bound can be obtained in the latter case. We obtain a bound on the complexity of many faces in certain arrangements of pseudo segments in the plane and with this bound we establish an upper bound of $O(n^{7/3})$ on the complexity of all the interesting cells in a TA-arrangement. The ability to obtain an improved bound for the case of a telescopic arm has drawn our attention to the fact that anchored robot arms with three degrees of freedom lend themselves more easily to combinatorial analysis of the type that we pursue than "free-flying" non-convex robots with three degrees of freedom. This phenomenon is further explained in the next section where more results that exploit it are presented.

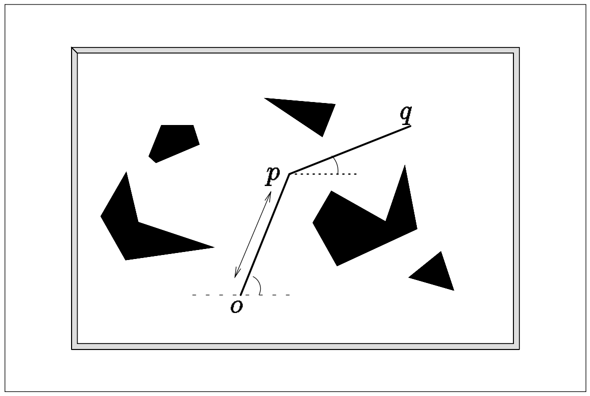

Figure 5: The telescopic arm

Using the combinatorial bounds on the complexity of the interesting cells in an L-arrangement and a TA-arrangement, we are able to augment the reachability algorithm of Subsection 3.2 to a find-path algorithm. The running time of the algorithm is $O(n^{5/2}\log^2 n)$ for an L-shaped object and $O(n^{7/3})$ for a telescopic arm moving among n point obstacles in the plane.

We conclude this section with the following open problem: For an arrangement of n triangles in space it is known that the maximum complexity of all the interesting cells is $\Theta(n^{7/3})$. The only lower bound known on the complexity of all the interesting cells in an L-arrangement or a TA-arrangement is $\Omega(n^2)$. The problem is to tighten the gap between the lower and upper bounds in each of these arrangements. Although we further explore these arrangements in the next section and obtain near-tight bounds on the complexity of a single cell in them, the actual bounds for the interesting cells remain an open problem.

4 Motion Planning with Three Degrees of Freedom and the Complexity of a Single 3D Cell

This section is devoted to near-tight bounds on the complexity of a single cell in various arrangements of surface patches related to motion-planning problems. We start with a bound of $(n^2\alpha(n))$ for a TA-arrangement. Then we present a somewhat more general result for a family of arrangements, and mention two applications of this result, one of which is to the motion-planning problem of a robot arm with three rotating joints. Afterwards, we describe a result for arrangements of triangles, and conclude with the L-arrangement.

Before we proceed to describe near-tight bounds on the complexity of a single cell in arrangements of surface *patches* we remark that tight bounds on the complexity of a single cell are known for certain arrangements of surfaces: A tight bound $\Theta(n)$ is obvious for arrangements of n planes in space. For arrangements of n spheres in space a tight bound $\Theta(n^2)$ was obtained, employing two different methods, in [Au] and in [KLPS].

4.1 A Sharp Bound on the Complexity of a Single Cell in a TA-Arrangement

Consider again the telescopic arm presented in Subsection 3.3 and in Figure 5. We obtain a bound $O(n^2\alpha(n))$ on the maximum complexity of one cell in the arrangement related to its motion among point obstacles. We distinguish two types of constraint surfaces in the arrangement that this problem induces: A *blue* surface is the result of a contact between the first link $\overline{op}$ of the arm and an obstacle point and a *red* surface which is the result of a contact between the second link $\overline{pq}$ and an obstacle point.

Observe that if we consider only the red surfaces, then the overall complexity of the free portions of the entire arrangement defined by them is at most $O(n^2)$—this follows from the result by Leven and Sharir [LS] for the motion-planning problem for a line segment moving among polygonal obstacles in the plane. If we consider the blue surfaces only, then we get an arrangement with linear complexity, since if the obstacles are in general position, then the blue surfaces are disjoint. Furthermore, since $\overline{op}$ is the anchored link, at every θ-cross-section of the configuration space, the relative displacement of the cross-sections of blue surfaces is the same. We interpret the facts about the blue surfaces as if there is a certain "depth order" among the blue surfaces in 3-space. Putting these observations together, we can confine ourselves to counting the number of vertices on the boundary of one cell that appear on blue surfaces and are the meeting point of two red surfaces and one blue surface. We are then able to reduce the 3D single cell problem into a collection of $O(n)$ two-dimensional single face problems in the arrangements defined naturally on the blue surfaces.

As mentioned earlier, the bound obtained is $O(n^2\alpha(n))$ which is within an $\alpha(\cdot)$ factor off the lower bound for this quantity. Further details can be found in [Ha92]. Although the bound applies to one, very special arrangement, we regard it as interesting for the following reasons: First, in all the bounds we will present below, certain recursion schemes are used and as a result a polylogarithmic factor shows up in each bound; our bound for the TA-arrangement suggests that the polylogarithmic factor in these bounds might be an artifact of the proof technique rather than describing the inherent complexity of a single cell in the arrangements for which the bounds are obtained. Second, it takes advantage of the fact that the moving body is an anchored robot arm and thus more explicitly shows what makes anchored arms easier to analyze then "free-flying" robots, using the type of analysis we are pursuing.

A slightly more general approach is described in the next subsection.

4.2 Moving a Three-Link Arm and a Special Arrangement of Triangles

In [Ha91] we have obtained the following result:

Theorem 4.1 *Let R be a set of n red surfaces*[3] *and B be a set of m blue surfaces in space, forming a three-dimensional arrangement $\mathcal{A}(R \cup B)$ such that:*

[3]The theorem applies to *surface patches*; we use *surfaces* for short.

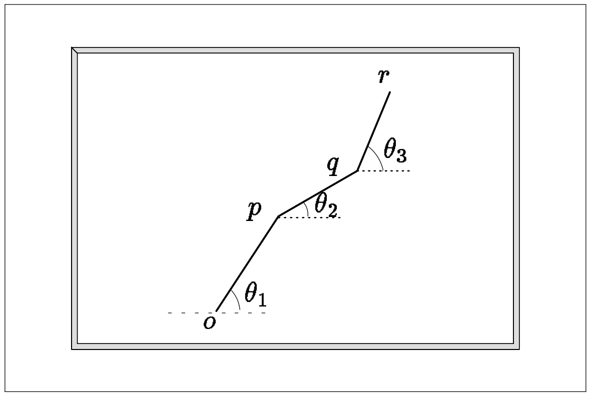

Figure 6: The three-link arm

(i) The blue surfaces are pairwise disjoint.

(ii) The intersection between a pair of red surfaces, or between one red and one blue surface, consists of at most one connected, simple, open curve.

(iii) No three red surfaces meet at a common point.

(iv) (We omit the forth property which would require some more definitions and is easily proven to exist in the applications that we describe below. See [Ha91] for details.)

Then the maximum combinatorial complexity of any single cell in $\mathcal{A}$ is at most $O(mn \log n + n^2)$.

Let R_i be a red surface in the collection of surfaces mentioned in the theorem and let A_i be the arrangement of curves on R_i obtained by intersecting all the other $m + n - 1$ (red or blue) surfaces with R_i. A special property of arrangements that comply with Conditions (i)-(iii) of the theorem is that the maximum complexity of any t distinct faces on the two-dimensional arrangements $A_i, i = 1, 2, \ldots, n$ is $O(mn + n^2 + t)$, or, in other words, the complexity of any t distinct faces in these arrangements is proportional to their number, up to an additive factor of $O(mn + n^2)$. This fact is the crux of the proof of Theorem 4.1. Note that this is not true for arrangements of surfaces in general, e.g., one can construct an arrangement of n triangles in space, where the complexity of n^2 distinct faces on the triangles may have $\Omega(n^{7/3})$ complexity.

We mention two applications of Theorem 4.1: The motion-planning problem for a three-link arm and a special arrangement of triangles in space. The standard three-link anchored arm in the plane has three rotational degrees of freedom: θ_1, θ_3 and θ_3 (see Figure 6). This arm is a prevailing kinematic substructure of existing robot manipulators and is therefore a natural problem to study in the framework of algorithmic motion planning. The arrangement induced by the motion-planning problem for a three-link arm consists, among other surfaces, of surfaces that are the result of a rotational sweep in space of conchoidal curves and therefore rather difficult to analyze by applying to its geometric structure. Since Theorem 4.1 is mainly phrased in topological terms we can apply it in this case to obtain an upper bound $O(n^2\alpha(n)\log n)$ on the complexity of a single cell in the arrangement. For details, see [Ha91].

We also apply the theorem to obtain the following result: The maximum complexity of a single cell in an arrangement of m pairwise disjoint triangles and n vertical (possibly intersecting) triangles is at most $O(mn\log n + n^2)$. This bound shows that the technique of [Ha91], unlike previous approaches to the single cell problem, distinguishes a single cell from all the "interesting" (non-convex, for triangles) cells. Indeed, if we take $\frac{n}{2}$ triangles of each type, the bound for a single cell is $O(n^2 \log n)$ whereas the complexity of all the interesting cells in such an arrangement can be shown to be as large as $\Omega(n^{7/3})$.

4.3 Arbitrary Arrangements of Triangles in Space

The proof of Theorem 4.1 as well as the proof of the combinatorial result of Subsection 3.1 are based on a technique that has proven useful in obtaining combinatorial geometry results—the so-called *combination lemma*. Recently, a new scheme has been proposed and applied to several combinatorial-geometry problems in [ESS], [AMS] and [ArS92]. In [ArS92], Aronov and Sharir combine it with a new geometric observation to obtain a bound $O(n^2 \log n)$ on the complexity of a single cell in arbitrary arrangements of triangles in space. Evidently, their technique distinguishes a single cell from all the interesting cells.

In [ArS92], the authors count the number of *popular faces* of any dimension on the boundary of a single cell of an arrangement of n triangles. A popular 2-face in a cell, for example, is a face both sides of which belong to the boundary of the cell; a popular 1-face (or edge) is an edge bounding the cell such that all its four sides appear on the cell boundary, etc. Using a previous result of theirs [ArS90] that says that a single cell in an arrangement of triangles can be decomposed into $O(n^2)$ interior-disjoint convex subcells, they prove that the maximum number of popular faces of dimension $0, 1$ or 2 bounding a single cell in such an arrangement is $O(n^2)$. Using this result they are able to establish the bound mentioned above.

As mentioned earlier, even when only the bound on all the interesting cells was known, Aronov and Sharir devised a randomized algorithm for computing a single cell in an arrangement of triangles. Its running time is dependent on the worst case complexity of a single cell in such an arrangement. Now that the bound on this quantity is $O(n^2 \log n)$, the expected running time of the algorithm is $O(n^{2+\varepsilon})$ for any $\varepsilon > 0$ and where the constant of proportionality depends on ε. The algorithm is based on random sampling (see, e.g., [CS]).

4.4 A Single Cell in an L-Arrangement

One of the problems encountered when trying to apply the technique of [ArS92] to an L-arrangement is that the proof of [ArS92] relies on the convex decomposition of a single cell. In arrangements relating to motion planning with rotation, even the simplest, dull cell is not necessarily convex. In [Ha92], we show that a weaker condition is sufficient to obtain a bound on the number of popular faces in a cell of an L-arrangement, and that this condition applies to a larger family of arrangements induced by motion-planning problems. Using this observation we derive a bound $O(n^2 \log^2 n)$ on the complexity of a single cell in an L-arrangement.

Another condition that is necessary for the proof of [ArS92] is that any three surfaces (in their case triangles) meet in at most one point. This also holds in the case of an L-shaped object if we restrict our analysis to a rotation range of π; this can be done without loss of generality. However, this condition narrows the range of application of the current technique to motion-planning problems. We have been able to identify another instance of a motion-planning problem where this condition holds, namely, moving a rigid "spider"—a collection of line segments all glued together in one of their endpoints, and therefore we could obtain a bound $O(n^2 \log^2 n)$ on the complexity of a single cell in the corresponding arrangement as well. We are currently studying more general motion-planning problems with three degrees of freedom, where the condition that any three constraint surfaces meet in at most one point does not hold [HS92b].

As for an algorithm, we are able to further augment the algorithm described in Subsection 3.2 to a deterministic algorithm for computing a single cell in an L-arrangement with near-quadratic running time [Ha92].

5 Conclusion

In this paper we have discussed an approach to robot motion planning that aims to develop exact (non-heuristic) solutions. We have mentioned past work in motion planning with two degrees of freedom and concentrated on recent results in motion planning with three degrees of freedom. The results consist of improved combinatorial bounds on the complexity of a single cell in arrangements of constraint surfaces that are induced by motion-planning problems, and of efficient techniques for constructing a single cell in these arrangements.

The paper raises several open problems. The main open problem is to extend the results, both combinatorial and algorithmic, to more general arrangements of surfaces and thus to improve the solution to the motion-planning problem for a wider family of robot systems with three degrees of freedom. Another desirable extension is to arrangements of hypersurfaces in higher dimensional spaces, leading to efficient motion planning for robot systems with more than three degrees of freedom.

An intriguing open problem is that of devising exact solutions to motion-planning problems without computing even a single cell. That is, can one solve the motion-planning problem with resources that are substantially smaller than the resources required to construct the one cell in the configuration space that contains the point corresponding to the initial placement of the robot?

References

[AA] P.K. Agarwal and B. Aronov, Counting facets and incidences, *Discrete and Computational Geometry* **7** (1992), pp. 359–369.

[ASS] P.K. Agarwal, M. Sharir and P.W. Shor, Sharp upper and lower bounds on the length of general Davenport-Schinzel sequences, *J. Combinatorial Theory, Series A* **52** (1989), pp. 228–274.

[ABP] P. Alevizos, J.-D. Boissonnat and F. Preparata, An optimal algorithm for the boundary of a cell in a union of rays, *Algorithmica* **5** (1990), pp. 573–590.

[AMS] B. Aronov, J. Matoušek and M. Sharir, On the sum of squares of cell complexities in hyperplane arrangements, *Proc. 7th Symposium on Computational Geometry*, 1991, pp. 307-313.

[AO] B. Aronov and C. Ó'Dúnlaing, Analysis of the motion-planning problem for a simple two-link planar arm, *Technical Report 314, Courant Institute of Mathematical Sciences*, 1987.

[APS] B. Aronov, M. Pellegrini and M. Sharir, On the zone of a surface in a hyperplane arrangement, to appear in *Discrete and Computational Geometry.*

[ArS90] B. Aronov and M. Sharir, Triangles in space, or building (and analyzing) castles in the air, *Combinatorica* **10** (1990), pp. 137–173.

[ArS92] B. Aronov and M. Sharir, Castles in the air revisited, *Proc. 8th Symposium on Computational Geometry*, 1992, pp. 146–156.

[Au] F. Aurenhammer, Power diagrams: Properties, algorithms and applications, *SIAM J. on Computing*, **16** (1987), pp. 78–96.

[Ca] J. Canny, The complexity of robot motion planning, Ph.D. Dissertation, Computer Science Department, M.I.T., May 1987.

[CEGSS] B. Chazelle, H. Edelsbrunner, L.J. Guibas, M. Sharir and J. Snoeyink, Computing a single face in an arrangement of line segments, *Proc. 2nd SIAM-ACM Symp. on Discrete Algorithms*, 1991, pp. 441–448.

[CGL] B. Chazelle, L.J. Guibas and D.T. Lee, The power of geometric duality, *BIT* **25**, 1985, pp. 76–90.

[Cl] K.L. Clarkson, Computing a single face in arrangements of segments, *manuscript*, 1990.

[CS] K.L. Clarkson and P.W. Shor, Applications of random sampling in computational geometry, II, *Discrete and Computational Geometry* **4** (1989), pp. 387–421.

[dBvKS] M. de Berg, M. van Kreveld and J. Snoeyink, Point location in zones of k-flats in arrangements, *Proc. 3rd Canadian Conference on Computational Geometry*, Vancouver, 1991, pp. 228–232.

[Ed] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer Verlag, Heidelberg, 1987.

[EGPPSS] H. Edelsbrunner, L.J. Guibas, J. Pach, R. Pollack, R. Seidel and M. Sharir, Arrangements of curves in the plane—topology, combinatorics, and algorithms, *Theoretical Computer Science* **92** (1992), pp. 319–336.

[EGS] H. Edelsbrunner, L.J. Guibas and M. Sharir, The complexity and construction of many faces in arrangements of lines and of segments, *Discrete and Computational Geometry* **5** (1990), pp. 161–196.

[EOS] H. Edelsbrunner, J. O'Rourke and R. Seidel, Constructing arrangements of lines and hyperplanes with applications, *SIAM J. on Computing* **15** (1986), pp. 341–363.

[ESS] H. Edelsbrunner, R. Seidel and M. Sharir, On the zone theorem for hyperplane arrangements, to appear in *SIAM J. on Computing.*

[GS] L. Guibas and M. Sharir, Combinatorics and algorithms of arrangements, to appear in *New Trends in Discrete and Computational Geometry*, (J. Pach, Ed.).

[GSS] L. Guibas, M. Sharir and S. Sifrony, On the general motion planning problem with two degrees of freedom, *Discrete and Computational Geometry* **4** (1989), 491–521.

[Ha91] D. Halperin, On the complexity of a single cell in certain arrangements of surfaces in 3-space, *Proc. 7th Symposium on Computational Geometry*, 1991, pp. 314–323; to appear in *Discrete and Computational Geometry.*

[Ha92] D. Halperin, *Algorithmic Motion Planning via Arrangements of Curves and of Surfaces*, Ph.D. Thesis, Tel-Aviv University, July 1992.

[HOS] D. Halperin, M.H. Overmars and M. Sharir, Efficient motion planning for an L-shaped object, *SIAM J. on Computing*, **21** (1992), pp. 1–23.

[HS91] D. Halperin and M. Sharir, On disjoint concave chains in arrangements of (pseudo) lines, *Information Processing Letters* **40** (1991), pp. 189–192.

[HS92a] D. Halperin and M. Sharir, Improved combinatorial bounds and efficient techniques for certain motion planning problems with three degrees of freedom, *Computational Geometry: Theory and Applications*, **1** (1992), pp. 269–303.

[HS92b] D. Halperin and M. Sharir, Improved combinatorial bounds for the motion planning problem for a polygon in a polygonal environment, *in preparation.*

[HarS] S. Hart and M. Sharir, Nonlinearity of Davenport-Scinzel sequences and of generalized path-compression schemes, *Combinatorica* **2**, (1986), pp.151–177.

[He] J. Hershberger, Finding the upper envelope of n line segments in $O(n \log n)$ time, *Information Processing Letters* **33** (1989), pp. 169–174.

[KLPS] K. Kedem, R. Livne, J. Pach and M. Sharir, On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles, *Discrete and Computational Geometry* **1** (1986), pp. 59–71.

[KS] K. Kedem and M. Sharir, An efficient motion planning algorithm for a convex polygonal object in two-dimensional polygonal space, *Discrete and Computational Geometry* **5** (1990), pp. 43–75.

[La] J.-C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, 1991.

[LS] D. Leven and M. Sharir, An efficient and simple motion planning algorithm for a ladder moving in two-dimensional space amidst polygonal barriers, *J. of Algorithms* **8** (1987), pp. 192–215.

[PaS] J. Pach and M. Sharir, The upper envelope of piecewise linear functions and the boundary of a region enclosed by convex plates: Combinatorial analysis, *Discrete and Computational Geometry* **4** (1989), pp. 291–309.

[PSS] R. Pollack, M. Sharir and S. Sifrony, Separating two simple polygons by a sequence of translations, *Discrete and Computational Geometry* **3** (1988), pp. 123–136.

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

[SS] J.T. Schwartz and M. Sharir, On the two-dimensional Davenport-Schinzel problem, *J. of Symbolic Computation* **10** (1990), pp. 371–393.

[Sh88] M. Sharir, Davenport-Schinzel sequences and their geometric applications, *Theoretical Foundations of Computer Graphics and CAD*, (R.A. Earnshaw, Ed.), NATO ASI Series, F40, Springer Verlag, Berlin, 1988, pp. 253–278.

[Sh89] M. Sharir, Algorithmic motion planning in robotics, *Computer* **22**, March 1989, pp. 9–20.

[WS] A. Wiernik and M. Sharir, Planar realizations of nonlinear Davenport-Schinzel sequences by segments, *Discrete and Computational Geometry* **3** (1988), pp. 15–47.

[Ya] C.-K. Yap, Algorithmic motion planning, *Advances in Robotics, Vol. I: Algorithmic and Geometric Aspects of Robotics*, (J. T. Schwartz and C.-K. Yap, Eds.), Lawrence Erlbaum Associates, Inc., New Jersey, 1987, pp. 95–143.