51 ALGORITHMIC MOTION PLANNING

Dan Halperin, Oren Salzman and Micha Sharir

INTRODUCTION

Motion planning is a fundamental problem in robotics. It comes in a variety of forms, but the simplest version is as follows. We are given a robot system B, which may consist of several rigid objects attached to each other through various joints, hinges, and links, or moving independently, and a 2D or 3D environment V cluttered with obstacles. We assume that the shape and location of the obstacles and the shape of B are known to the planning system. Given an initial placement Z_1 and a final placement Z_2 of B, we wish to determine whether there exists a collision-avoiding motion of B from Z_1 to Z_2, and, if so, to plan such a motion. In this simplified and purely geometric setup, we ignore issues such as incomplete information, nonholonomic constraints, control issues related to inaccuracies in sensing and motion, nonstationary obstacles, optimality of the planned motion, and so on.

Since the early 1980s, motion planning has been an intensive area of study in robotics and computational geometry. In this chapter we will focus on *algorithmic motion planning*, emphasizing theoretical algorithmic analysis of the problem and seeking worst-case asymptotic bounds, and only mention briefly practical heuristic approaches to the problem. The majority of this chapter is devoted to the simplified version of motion planning, as stated above. Section 51.1 presents general techniques and lower bounds. Section 51.2 considers efficient solutions to a variety of specific moving systems with a small number of degrees of freedom. These efficient solutions exploit various sophisticated methods in computational and combinatorial geometry related to arrangements of curves and surfaces (Chapter 30). Section 51.3 then briefly discusses various extensions of the motion planning problem such as computing *optimal* paths with respect to various quality measures, computing the path of a tethered robot, incorporating uncertainty, moving obstacles, and more.

51.1 GENERAL TECHNIQUES AND LOWER BOUNDS

GLOSSARY

Some of the terms defined here are also defined in Chapter 52.

Robot B: A mechanical system consisting of one or more rigid bodies, possibly connected by various joints and hinges.

Physical space (workspace): The 2D or 3D environment in which the robot moves.

Placement: The portion of physical space occupied by the robot at some instant.

Degrees of freedom k: The number of real parameters that determine the robot B's placements. Each placement can be represented as a point in $\mathbb{R}^k$.

Free placement: A placement at which the robot is disjoint from the obstacles.

Semifree placement: A placement at which the robot does not meet the interior of any obstacle (but may be in contact with some obstacles).

Configuration space $\mathcal{C}$: A portion of k-space (where k is the number of degrees of freedom of B) that represents all possible robot placements; the coordinates of any point in this space specify the corresponding placement.

Expanded obstacle / C-obstacle / forbidden region: For an obstacle O, this is the portion O^* of configuration space consisting of placements at which the robot intersects (collides with) O.

Free configuration space $\mathcal{F}$: The subset of configuration space consisting of free placements of the robot: $\mathcal{F} = \mathcal{C} \setminus \bigcup_O O^*$. (In the literature, this usually also includes semifree placements. In that case, $\mathcal{F}$ is the complement of the union of the *interiors* of the expanded obstacles.)

Contact surface: For an obstacle feature a (corner, edge, face, etc.) and for a feature b of the robot, this is the locus in $\mathcal{C}$ of placements at which a and b are in contact with each other. In most applications, these surfaces are semialgebraic sets of constant description complexity (see definitions below).

Collision-free motion of B: A path contained in $\mathcal{F}$. Any two placements of B that can be reached from each other via a collision-free path must lie in the same (arcwise-)connected component of $\mathcal{F}$.

Arrangement $\mathcal{A}(\Sigma)$: The decomposition of k-space into cells of various dimensions, induced by a collection Σ of surfaces in $\mathbb{R}^k$. Each cell is a maximal connected portion of the intersection of some fixed subcollection of surfaces that does not meet any other surface. See Chapter 30. Since a collision-free motion should not cross any contact surface, $\mathcal{F}$ is the union of some of the cells of $\mathcal{A}(\Sigma)$, where Σ is the collection of contact surfaces.

Semialgebraic set: A subset of $\mathbb{R}^k$ defined by a Boolean combination of polynomial equalities and inequalities in the k coordinates. See Chapter 38.

Constant description complexity: Said of a semialgebraic set if it is defined by a constant number of polynomial equalities and inequalities of constant maximum degree (where the number of variables is also assumed to be constant).

Example. Let B be a rigid polygon with k edges, moving in a planar polygonal environment V with n edges. The system has three degrees of freedom, (x, y, θ), where (x, y) are the coordinates of some reference point on B, and θ is the orientation of B. Each contact surface is the locus of placements where some vertex of B touches some edge of V, or some edge of B touches some vertex of V. There are $2kn$ contact surfaces, and if we replace θ by $\tan \frac{\theta}{2}$, then each contact surface becomes a portion of some algebraic surface of degree at most 4, bounded by a constant number of algebraic arcs, each of degree at most 2.

51.1.1 GENERAL SOLUTIONS

GLOSSARY

Cylindrical algebraic decomposition of $\mathcal{F}$ **:** A recursive decomposition of $\mathcal{C}$ into cylindrical-like cells originally proposed by Collins [Col75]. Over each cell of the decomposition, each of the polynomials involved in the definition of $\mathcal{F}$ has a fixed sign (positive, negative, or zero), implying that $\mathcal{F}$ is the union of some of the cells of this decomposition. See Chapter 38 for further details.

Connectivity graph: A graph whose nodes are the (free) cells of a decomposition of $\mathcal{F}$ and whose arcs connect pairs of adjacent cells.

Roadmap $\mathcal{R}$**:** A network of one-dimensional curves within $\mathcal{F}$, having the properties that (i) it *preserves the connectivity* of $\mathcal{F}$, in the sense that the portion of $\mathcal{R}$ within each connected component of $\mathcal{F}$ is (nonempty and) connected; and (ii) it is ***reachable***, in the sense that there is a simple procedure to move from any free placement of the robot to a placement on $\mathcal{R}$; we denote the mapping resulting from this procedure by $\phi_{\mathcal{R}}$.

Retraction of $\mathcal{F}$ ***onto*** $\mathcal{R}$**:** A continuous mapping of $\mathcal{F}$ onto $\mathcal{R}$ that is the identity on $\mathcal{R}$. The roadmap mapping $\phi_{\mathcal{R}}$ is usually a retraction. When this is the case, we note that for any path ψ within $\mathcal{F}$, represented as a continuous mapping $\psi : [0,1] \mapsto \mathcal{F}$, $\phi_{\mathcal{R}} \circ \psi$ is a path within $\mathcal{R}$, and, concatenating to it the motions from $\psi(0)$ and $\psi(1)$ to $\mathcal{R}$, we see that there is a collision-free motion of B between two placements Z_1, Z_2 iff there is a path within $\mathcal{R}$ between $\phi_{\mathcal{R}}(Z_1)$ and $\phi_{\mathcal{R}}(Z_2)$.

Silhouette: The set of critical points of a mapping; see Chapter 38.

CELL DECOMPOSITION

$\mathcal{F}$ is a semialgebraic set in $\mathbb{R}^k$. Applying Collins's cylindrical algebraic decomposition results in a collection of cells whose total complexity is $O((nd)^{3^k})$, where d is the maximum algebraic degree of the polynomials defining the contact surfaces; the decomposition can be constructed within a similar time bound. If the coordinate axes are generic, then we can also compute all pairs of cells of $\mathcal{F}$ that are ***adjacent*** to each other (i.e., cells whose closures (within $\mathcal{F}$) overlap), and store this information in the form of a connectivity graph. It is then easy to search for a collision-free path through this graph, if one exists, between the (cell containing the) initial robot placement and the (cell containing the) final placement. This leads to a doubly-exponential general solution for the motion planning problem:

THEOREM 51.1.1 *Cylindrical Cell Decomposition* [SS83]

Any motion planning problem, with k degrees of freedom, for which the contact surfaces are defined by a total of n polynomials of maximum degree d, can be solved by Collins's cylindrical algebraic decomposition, in randomized expected time $O((nd)^{3^k})$.

Remakrs. (1) The randomization is needed only to choose a generic direction for the coordinate axes. (2) Here and throughout the chapter we adhere to the *real RAM* model of computation, which is standard in computational geometry [PS85].

ROADMAPS

An improved solution is given in [Can87, BPR00] based on the notion of a ***roadmap*** $\mathcal{R}$, a network of one-dimensional curves within (the closure of) $\mathcal{F}$, having properties defined in the glossary above. Once such a roadmap $\mathcal{R}$ has been constructed, any motion planning instance reduces to path searching within $\mathcal{R}$, which is easy to do. $\mathcal{R}$ is constructed recursively, as follows. One projects $\mathcal{F}$ onto some generic 2-plane, and computes the silhouette of $\mathcal{F}$ under this projection. Next, the critical values of the projection of the silhouette on some line are found, and a roadmap is constructed recursively within each slice of $\mathcal{F}$ at each of these critical values. The resulting "sub-roadmaps" are then merged with the silhouette, to obtain the desired $\mathcal{R}$.

The original algorithm of Canny [Can87] relies heavily on the polynomials defining $\mathcal{F}$ being in general position, and on the availability of a generic plane of projection. This algorithm runs in $n^k(\log n)d^{O(k^4)}$ deterministic time, and in $n^k(\log n)d^{O(k^2)}$ expected randomized time. Later work [BPR00] addresses and overcomes the general position issue, and produces a roadmap for any semialgebraic set; the running time of this solution is $n^{k+1}d^{O(k^2)}$.

If we ignore the dependence on the degree d, the algorithm of Canny is close to optimal in the worst case, assuming that some representation of the entire $\mathcal{F}$ has to be output, since there are easy examples where the free configuration space consists of $\Omega(n^k)$ connected components.

THEOREM 51.1.2 *Roadmap Algorithm* [Can87, BPR00]

Any motion planning problem, as in the preceding theorem, in general position can be solved by the roadmap technique in $n^k(\log n)d^{O(k^4)}$ *deterministic time, and in* $n^k(\log n)d^{O(k^2)}$ *expected randomized time.*

51.1.2 LOWER BOUNDS

The upper bounds for both general solutions are (at least) exponential in k (but are polynomial in the other parameters when k is fixed). This raises the issue of calibrating the complexity of the problem when k can be arbitrarily large.

THEOREM 51.1.3 *Lower Bounds*

The motion planning problem, with arbitrarily many degrees of freedom, is PSPACE-hard for the instances of: (a) coordinated motion of many rectangular boxes along a rectangular floor [HSS84]; the problem remains PSPACE-hard even if only two types of rectangles are used [HD05] or if only unit squares are used [SH15] (b) motion planning of a planar mechanical linkage with many links [HJW84]; and (c) motion planning for a multi-arm robot in a 3-dimensional polyhedral environment [Rei87].

All early results can also be found in the collection [HSS87]. There are NP-hardness results for other systems; see, e.g., [HJW85] and [SY84]. Hearn and Demaine [HD09] introduced a general tool called the *nondeterministic constraint logic* (NCL) model of computation that facilitates derivation of hardness results (in particular, the results in [HD05] and [SH15]). Using the NCL model, Hearn and Demaine also derive the PSPACE-hardness of a variety of motion-related puzzle-like problems that consist of sliding game pieces. In particular, they applied their technique to the SOKOBAN puzzle, where multiple "crates" need to be pushed to target locations, and the Rush Hour game, where a parking attendant has to evacuate a car from a parking lot, by clearing a route blocked by other cars.

Facing the aforementioned hardness results, we consider specific problems with small values of k, with the goal of obtaining solutions better than those yielded by the general techniques. Alternatively, we can approach the general problem with heuristic or approximate schemes. We will mostly survey here the former approach, which allows for efficient computation for a restricted set of problems. However, as the general motion-planning problem is of practical interest, considerable research effort has been devoted to practical solutions to this problem. Noteworthy is the *sampling-based* approach in which $\mathcal{F}$ is approximated via a roadmap constructed by randomly sampling the configuration space. We will review this practical approach (as well as some alternative approaches) briefly and refer the reader to Chapter 52 for an in-depth discussion of sampling-based algorithms.

51.2 MOTION PLANNING WITH A SMALL NUMBER OF DEGREES OF FREEDOM

In this main section of the chapter, we review solutions to a variety of specific motion-planning problems, most of which have 2 or 3 degrees of freedom. Exploiting the special structure of these problems leads to solutions that are more efficient than the general methods described above.

GLOSSARY

Jordan arc/curve: The image of the closed unit interval under a continuous bijective mapping into the plane. A closed Jordan curve is the image of the unit circle under a similar mapping, and an unbounded Jordan curve is an image of the open unit interval (or of the entire real line) that separates the plane.

Randomized algorithm: An algorithm that applies internal randomization ("coin-flips"). We consider here "Las Vegas" algorithms that always terminate, and produce the correct output, but whose running time is a random variable that depends on the internal coin-flips. We will state upper bounds on the expectation of the running time (the ***randomized expected time***) of such an algorithm, which hold for any input. See Chapter 45.

General position: The input to a geometric problem is said to be in general position if no nontrivial algebraic identity with integer coefficients holds among the parameters that specify the input (assuming the input is not overspecified). For example: no three input points should be collinear, no four points cocircular, no three lines concurrent, etc. (In general, this requirement is too restrictive and many instances explicitly specify which identities are not supposed to hold.)

Minkowski sum: For two planar (or spatial) sets A and B, their Minkowski sum, or pointwise vector addition, is the set $A \oplus B = \{x + y \mid x \in A, y \in B\}$.

Convex distance function: A convex region B that contains the origin in its interior induces a convex distance function d_B defined by

$$d_B(p, q) = \min \{\lambda \mid q \in p \oplus \lambda B\}.$$

If B is centrally symmetric with respect to the origin then d_B is a metric whose unit ball is B.

***B*-Voronoi diagram:** For a set S of sites, and a convex region B as above, the B-Voronoi diagram $\mathrm{Vor}_B(S)$ of S is a decomposition of space into Voronoi cells $V(s)$, for $s \in S$, such that

$$V(s) = \{p \mid d_B(p,s) \le d_B(p,s') \text{ for all } s' \in \mathcal{S}\}.$$

Here $d_B(p,s) = \min_{q\in s} d_B(p,q)$.

***α(n)*:** The extremely slowly-growing inverse Ackermann function; see Chapter 30.

Contact segment: The locus of (not necessarily free) placements of a polygon B translating in a planar polygonal workspace, at each of which either some specific vertex of B touches some specific obstacle edge, or some specific edge of B touches some specific obstacle vertex.

Contact curve: A generalization of "contact segment" to the locus of (not necessarily free) placements of a more general robot system B, assuming that B has only two degrees of freedom, where some specific feature of B makes contact with some specific obstacle feature.

51.2.1 TWO DEGREES OF FREEDOM

A TRANSLATING POLYGON IN 2D

This is a system with two degrees of freedom (translations in the x and y directions).

A CONVEX POLYGON

Suppose first the translating polygon B is a *convex* k-gon, and there are m convex polygonal obstacles, $A_1, \ldots, A_m$, with pairwise disjoint interiors, having a total of n edges. The region of configuration space where B collides with A_i is the *Minkowski sum*

$$K_i = A_i \oplus (-B) = \{x - y \mid x \in A_i,\ y \in B\}.$$

The free configuration space is the complement of $\bigcup_{i=1}^{m} K_i$. Assuming general position, one can show:

THEOREM 51.2.1 [KLPS86]

(a) Each K_i is a convex polygon, with $n_i + k$ edges, where n_i is the number of edges of A_i.

(b) For each $i \ne j$, the boundaries of K_i and K_j intersect in at most two points. (This also holds when the A_i's and B are not polygons but are still convex.)

(c) Given a collection of planar regions $K_1, \ldots, K_m$, each enclosed by a closed Jordan curve, such that any pair of the bounding curves intersects at most twice, then the boundary of the union $\bigcup_{i=1}^{m} K_i$ consists of at most $6m - 12$ maximal connected portions of the boundaries of the K_i's, provided $m \ge 3$, and this bound is tight in the worst case.

Such a collection $K_1, \ldots, K_m$ is called a collection of *pseudo-disks*. Now, these properties, combined with several algorithmic techniques [KLPS86, MMP$^+$91, BDS95], imply:

THEOREM 51.2.2

(a) *The free configuration space for a translating convex polygon, as above, is a polygonal region with at most* $6m-12$ *convex vertices and* $N = \sum_{i=1}^{m}(n_i+k) = n + km$ *nonconvex vertices.*

(b) $\mathcal{F}$ *can be computed in deterministic time* $O(N \log n \log m)$ *or in randomized expected time* $O(N \log n)$.

If the robot is translating in a convex room with n walls, then the complexity of the free space is $O(n)$ and it can be computed in $O(n+k)$ time.

AN ARBITRARY POLYGON

Suppose next that B is an arbitrary polygonal region with k edges. Let A be the union of all obstacles, which is another polygonal region with n edges. As above, the free configuration space is the complement of the Minkowski sum

$$K = A \oplus (-B) = \{x - y \mid x \in A,\ y \in B\}.$$

K is again a polygonal region, but, in this case, its maximum possible complexity is $\Theta(k^2n^2)$ (see, e.g., [AFH02]), so computing it might be considerably more expensive than in the convex case. Efficient practical algorithms for the exact computation of Minkowski sums in this case (together with their implementation) are described in [AFH02, Wei06, BFH$^+$15].

A single face suffices. If the initial placement Z of B is given, then we do not have to compute the entire (complement of) K; it suffices to compute the connected component f of the complement of K that contains Z, because no other placement is reachable from Z via a collision-free motion.

Let Σ be the collection of all contact segments; there are $2kn$ such segments. The desired component f is the face of $\mathcal{A}(\Sigma)$ that contains Z. Using the theory of *Davenport-Schinzel sequences* (Chapter 30), one can show that the maximum possible combinatorial complexity of a single face in a two-dimensional arrangement of N segments is $\Theta(N\alpha(N))$. A more careful analysis [HCA$^+$95], combined with the algorithmic techniques of [CEG$^+$93, GSS89], shows:

THEOREM 51.2.3

(a) *The maximum combinatorial complexity of a single face in the arrangement of contact segments for the case of an arbitrary translating polygon is* $\Theta(kn\alpha(k))$ *(this improvement is significant only when* $k \ll n$*).*

(b) *Such a face can be computed in deterministic time* $O(kn\alpha(k)\log^2 n)$ *[GSS89], or in randomized expected time* $O(kn\alpha(k)\log n)$ *[CEG$^+$93].*

VORONOI DIAGRAMS

Another approach to motion planning for a translating *convex* object B is via generalized *Voronoi diagrams* (see Chapter 29), based on the convex distance function

$d_B(p, q)$. This function effectively places B centered at p and expands it until it hits q. The scaling factor at this moment is the d_B-distance from p to q (if B is a unit disk, d_B is the Euclidean distance). d_B satisfies the triangle inequality, and is thus "almost" a metric, except that it is not symmetric in general; it is symmetric iff B is centrally symmetric with respect to the point of reference.

Using this distance function d_B, a *B-Voronoi diagram* $\mathrm{Vor}_B(\mathcal{S})$ of $\mathcal{S}$ may be defined for a set $\mathcal{S}$ of m pairwise disjoint obstacles. See [LS87a, Yap87a].

THEOREM 51.2.4

Assuming that each of B and the obstacles in $\mathcal{S}$ has constant-description complexity, and that they are in general position, the B-Voronoi diagram has $O(m)$ complexity, and can be computed in $O(m \log m)$ time. If B and the obstacles are convex polygons, as above, then the complexity of $\mathrm{Vor}_B(\mathcal{S})$ is $O(N)$ and it can be computed in time $O(N \log m)$, where $N = n + km$.

One can show that if Z_1 and Z_2 are two free placements of B, then there exists a collision-free motion from Z_1 to Z_2 if and only if there exists a collision-free motion of B where its center moves only along the edges of $\mathrm{Vor}_B(\mathcal{S})$, between two corresponding placements W_1, W_2, where W_i, for $i = 1, 2$, is the placement obtained by pushing B from the placement Z_i away from its d_B-nearest obstacle, until it becomes equally nearest to two or more obstacles (so that its center lies on an edge of $\mathrm{Vor}_B(\mathcal{S})$).

Thus motion planning of B reduces to path-searching in the one-dimensional network of edges of $\mathrm{Vor}_B(\mathcal{S})$. This technique is called the ***retraction technique***, and can be regarded as a special case of the general roadmap algorithm. The resulting motions have "high clearance," and so are safer than arbitrary motions, because they stay equally nearest to at least two obstacles.

THEOREM 51.2.5

The motion-planning problem for a convex object B translating amidst m convex and pairwise disjoint obstacles can be solved in $O(m \log m)$ time, by constructing and searching in the B-Voronoi diagram of the obstacles, assuming that B and the obstacles have constant description complexity each. If B and the obstacles are convex polygons, then the same technique yields an $O(N \log m)$ solution, where $N = n + km$ is as above.

THE GENERAL MOTION-PLANNING PROBLEM WITH TWO DEGREES OF FREEDOM

If B is any system with two degrees of freedom, its configuration space is 2D, and, for simplicity, let us think of it as the plane (spaces that are topologically more complex can be decomposed into a constant number of "planar" patches). We construct a collection Σ of contact curves, which, under reasonable assumptions concerning B and the obstacles, are each an algebraic Jordan arc or curve of some fixed maximum degree b. In particular, each pair of contact curves will intersect in at most some constant number, $s \leq b^2$, of points.

As above, it suffices to compute the single face of $\mathcal{A}(\Sigma)$ that contains the initial placement of B. The theory of Davenport-Schinzel sequences implies that the complexity of such a face is $O(\lambda_{s+2}(n))$, where $\lambda_{s+2}(n)$ is the maximum length of an $(n, s+2)$-Davenport-Schinzel sequence (Chapter 30), which is slightly superlinear in n when s is fixed.

The face in question can be computed in deterministic time $O(\lambda_{s+2}(n)\log^2 n)$, using a fairly involved divide-and-conquer technique based on line-sweeping; see [GSS89] and Chapter 30. (Some slight improvements in the running time have subsequently been obtained.) Using randomized incremental (or divide-and-conquer) techniques, the face can be computed in randomized expected $O(\lambda_{s+2}(n)\log n)$ time [CEG$^+$93, SA95].

THEOREM 51.2.6 [GSS89, CEG$^+$93, BDS95]

Under the above assumptions, the general motion-planning problem for systems with two degrees of freedom can be solved in deterministic time $O(\lambda_{s+2}(n)\log^2 n)$, or in $O(\lambda_{s+2}(n)\log n)$ randomized expected time.

51.2.2 THREE DEGREES OF FREEDOM

A ROD IN A PLANAR POLYGONAL ENVIRONMENT

We next pass to systems with three degrees of freedom. Perhaps the simplest instance of such a system is the case of a line segment B ("rod," "ladder," "pipe") moving (translating and rotating) in a planar polygonal environment with n edges. The maximum combinatorial complexity of the free configuration space $\mathcal{F}$ of B is $\Theta(n^2)$ (recall that the naive bound for systems with three degrees of freedom is $O(n^3)$). A cell-decomposition representation of $\mathcal{F}$ can be constructed in (deterministic) $O(n^2 \log n)$ time [LS87b]. Several alternative near-quadratic algorithms have also been developed, including one based on constructing a Voronoi diagram in $\mathcal{F}$ [OSY87]. A worst-case optimal algorithm, with running time $O(n^2)$, has been given in [Veg90].

An $\Omega(n^2)$ lower bound for this problem has been established in [KO88]. It exhibits a polygonal environment with n edges and two free placements of B that are reachable from each other. However, any free motion between them requires $\Omega(n^2)$ "elementary moves," that is, the specification of any such motion requires $\Omega(n^2)$ complexity. This is a fairly strong lower bound, since it does not rely on lower bounding the complexity of the free configuration space (or of a single connected component thereof); after all, it is not clear why a motion planning algorithm should have to produce a full description of the whole free space (or of a single component).

THEOREM 51.2.7

Motion planning for a rod moving in a polygonal environment bounded by n edges can be performed in $O(n^2)$ time. There are instances where any collision-free motion of the rod between two specified placements requires $\Omega(n^2)$ "elementary moves."

A CONVEX POLYGON IN A PLANAR POLYGONAL ENVIRONMENT

Here B is a convex k-gon, free to move (translate and rotate) in an arbitrary polygonal environment bounded by n edges. The free configuration space is 3D, and there are at most $2kn$ contact surfaces, of maximum degree 4. The naive bound on the complexity of $\mathcal{F}$ is $O((kn)^3)$ (attained if B is nonconvex), but, using Davenport-Schinzel sequences, one can show that the complexity of $\mathcal{F}$ is only $O(kn\lambda_6(kn))$. Geometrically, a vertex of $\mathcal{F}$ is a semifree placement of B at which it makes simultaneously three obstacle contacts. The above bound implies that the number of such *critical placements* is only slightly super-quadratic (and not cubic) in kn.

Computing $\mathcal{F}$ in time close to this bound has proven more difficult, and only in the late 1990's has a complete solution, running in $O(kn\lambda_6(kn)\log kn)$ time and constructing the entire $\mathcal{F}$, been attained [AAS99]. Previous solutions, that were either incomplete with the same time bound, or complete and somewhat more expensive, are given in [KS90, HS96, KST97].

Another approach was given in [CK93]. It computes the Delaunay triangulation of the obstacles under the distance function d_B, when the orientation of B is fixed, and then traces the discrete combinatorial changes in the diagram as the orientation varies. The number of changes was shown to be $O(k^4 n\lambda_3(n))$. Using this structure, the algorithm of [CK93] produces a high-clearance motion of B between any two specified placements, in time $O(k^4 n\lambda_3(n)\log n)$.

Since all these algorithms are fairly complicated, one might consider in practice an alternative approximate scheme, proposed in [AFK$^+$92]. This scheme, originally formulated for a rectangle, discretizes the orientation of B, solves the translational motion planning for B at each of the discrete orientations, and finds those placements of B at which it can rotate (without translating) between two successive orientations. This scheme works very well in practice.

THEOREM 51.2.8

Motion planning for a k-sided convex polygon, translating and rotating in a planar polygonal environment bounded by n edges, can be performed in $O(kn\lambda_6(kn)\log kn)$ or $O(k^4 n\lambda_3(n)\log n)$ time.

EXTREMAL PLACEMENTS

A related problem is to find the free placement of the largest scaled copy of B in the given polygonal environment. This has applications in manufacturing, where one wants to cut out copies of B that are as large as possible from a sheet of some material.

If only translations are allowed, the B-Voronoi diagram can be used to find the largest free homothetic copy of B. If general rigid motions are allowed, the technique of [CK93] computes the largest free similar copy of B in time $O(k^4 n\lambda_3(n)\log n)$. An alternative technique is given in [AAS98], with randomized expected running time $O(kn\lambda_6(kn)\log^4 kn)$. Both bounds are nearly quadratic in n. See also earlier work on this problem in [ST94].

Finally, we mention the special case where the polygonal environment is the interior of a convex n-gon. This is simpler to analyze. The number of free critical placements of (similar copies of) B, at which B makes simultaneously four obstacle contacts, is $O(kn^2)$ [AAS98], and they can all be computed in $O(kn^2\log n)$ time. If only translations are allowed, this problem can easily be expressed as a linear program, and can be solved in $O(n+k)$ time [ST94].

THEOREM 51.2.9

The largest similar placement of a k-sided convex polygon in a planar polygonal environment bounded by n edges can be computed in randomized expected time $O(kn\lambda_6(kn)\log^4 kn)$ or in deterministic time $O(k^4 n\lambda_3(n)\log n)$. When the environment is the interior of an n-sided convex polygon, the running time improves to $O(kn^2 \log n)$, and to $O(n+k)$ if only translations are allowed.

A NONCONVEX POLYGON

Next we consider the case where B is an arbitrary polygonal region (not necessarily connected), translating and rotating in a polygonal environment bounded by n edges, as above. Here one can show that the maximum complexity of $\mathcal{F}$ is $\Theta((kn)^3)$. Using standard techniques, $\mathcal{F}$ can be constructed in $\Theta((kn)^3 \log kn)$ time, and algorithms with this running time bound have been implemented; see, e.g., [ABF89]. However, as in the purely translational case, it usually suffices to construct the connected component of $\mathcal{F}$ containing the initial placement of B. The general result, stated below, for systems with three degrees of freedom, implies that the complexity of such a component is only near-quadratic in kn. A special-purpose algorithm that computes the component in time $O((kn)^{2+\epsilon})$ is given in [HS96], where the constant of proportionality depends on ϵ. A more general algorithm with a similar running time bound is reported below. An earlier work considered the case where B is an L-shaped object moving amid n point obstacles [HOS92]. Motion planning can be performed in this case in time $O(n^2 \log^2 n)$.

THEOREM 51.2.10

Motion planning for an arbitrary k-sided polygon, translating and rotating in a planar polygonal environment bounded by n edges, can be performed in time $O((kn)^{2+\epsilon})$, for any $\epsilon > 0$. If the polygon is L-shaped and the obstacles are points, the running time improves to $O(n^2 \log^2 n)$.

A TRANSLATING POLYTOPE IN A 3D POLYHEDRAL ENVIRONMENT

Another interesting motion planning problem with three degrees of freedom involves a polytope B, with a total of k vertices, edges, and facets, translating amidst polyhedral obstacles in $\mathbb{R}^3$, with a total of n vertices, edges, and faces. The contact surfaces in this case are planar polygons, composed of a total of $O(kn)$ triangles in 3-space. Without additional assumptions, the complexity of $\mathcal{F}$ can be $\Theta((kn)^3)$ in the worst case. However, the complexity of a single component is only $O((kn)^2 \log kn)$. Such a component can be constructed in $O((kn)^{2+\epsilon})$ time, for any $\epsilon > 0$ [AS94].

If B is a convex polytope, and the obstacles consist of m convex polyhedra, with pairwise disjoint interiors and with a total of n faces, the complexity of the entire $\mathcal{F}$ is $O(kmn \log m)$ and it can be constructed in $O(kmn \log^2 m)$ time [AS97]. (Note that, in analogy with the two-dimensional case, $\mathcal{F}$ is the complement of the union of the Minkowski sums $A_i \oplus (-B)$, where A_i are the given obstacles. The above-cited bound is about the complexity and construction of such a union.) An earlier study [HY98] considered the case where B is a box, and obtained an $O(n^2\alpha(n))$ bound for the complexity of $\mathcal{F}$. Efficient practical algorithms for the exact computation of Minkowski sums for convex polyhedra are described in [FH07], and for general polyhedra in [Hac09].

THEOREM 51.2.11

Translational motion planning for an arbitrary polytope with k facets, in an arbitrary 3D polyhedral environment bounded by n facets, can be performed in time $O((kn)^{2+\epsilon})$, for any $\epsilon > 0$. If B is a convex polytope, and there are m convex pairwise disjoint obstacles with a total of n facets, then the motion planning can be performed in $O(kmn \log^2 m)$ time.

A BALL IN A 3D POLYHEDRAL ENVIRONMENT

Let B be a ball moving in 3D amidst polyhedral obstacles with a total of n vertices, edges, and faces. The complexity of the entire $\mathcal{F}$ is $O(n^{2+\epsilon})$, for any $\epsilon > 0$ [AS00a]. Note that, for the special case of line obstacles, the expanded obstacles are congruent (infinite) cylinders, and $\mathcal{F}$ is the complement of their union.

THEOREM 51.2.12

Motion planning for a ball in an arbitrary 3D polyhedral environment bounded by n facets can be performed in time $O(n^{2+\epsilon})$, for any $\epsilon > 0$.

It is worth mentioning that the combinatorial complexity of the union of n infinite cylinders in $\mathbb{R}^3$, having arbitrary radii, is $O(n^2 + \epsilon)$, for any $\epsilon > 0$ where the bound is almost tight in the worst case [Ezr11].

3D B-VORONOI DIAGRAMS

A more powerful approach to translational motion planning in three dimensions is via B-Voronoi diagrams, defined in three dimensions in full analogy to the two-dimensional case mentioned above. The goal is to establish a near-quadratic bound for the complexity of such a diagram. This would yield near-quadratic algorithms for planning the motion of the moving body B, for planning a high-clearance motion, and for finding largest homothetic free placements of B. The analysis of B-Voronoi diagrams is considerably more difficult in 3-space, and there are only a few instances where a near-quadratic complexity bound is known. One instance is for the case where B is a translating convex polytope with $O(1)$ facets in a 3D polyhedral environment [KS04]; the complexity of the diagram in this case is $O(n^{2+\epsilon})$. If the obstacles are lines or line segments, the complexity is $O(n^2\alpha(n)\log n)$ [CKS$^+$98, KS04].

The case where B is a ball appears to be more challenging. Even for the special case where the obstacles are lines, no near-quadratic bounds are known. However, if the obstacles are n lines with a constant number of orientations, the B-diagram has complexity $O(n^{2+\epsilon})$ [KS03].

THE GENERAL MOTION PLANNING PROBLEM WITH THREE DEGREES OF FREEDOM

The last several instances were special cases of the general motion planning problem with three degrees of freedom. In abstract terms, we have a collection Σ of N contact surfaces in $\mathbb{R}^3$, where these surfaces are assumed to be (semi-algebraic patches of) algebraic surfaces of constant maximum degree. The free configuration space consists of some cells of the arrangement $\mathcal{A}(\Sigma)$, and a single connected component of $\mathcal{F}$ is just a single cell in that arrangement.

Inspecting the preceding cases, a unifying observation is that while the maximum complexity of the entire $\mathcal{F}$ can be $\Theta(N^3)$, the complexity of a single component is invariably only near-quadratic in N. This was shown in [HS95a] to hold in general: the combinatorial complexity of a single cell of $\mathcal{A}(\Sigma)$ is $O(N^{2+\epsilon})$, for any $\epsilon > 0$, where the constant of proportionality depends on ϵ and on the maximum degree of the surfaces; see Chapter 30.

A general-purpose algorithm for computing a single cell in such an arrangement was given in [SS97]. It runs in randomized expected time $O(N^{2+\epsilon})$, for any $\epsilon > 0$, and is based on *vertical decompositions* in such arrangements (see Chapter 30).

THEOREM 51.2.13

An arbitrary motion planning problem with three degrees of freedom, involving N contact surface patches, each of constant description complexity, can be solved in time $O(N^{2+\epsilon})$, for any $\epsilon > 0$.

51.2.3 OTHER PROBLEMS WITH FEW DEGREES OF FREEDOM

MORE DEGREES OF FREEDOM

The general motion planning problem for systems with d degrees of freedom, for $d \geq 4$, calls for estimating the complexity of a single cell in the d-dimensional arrangement of the appropriate contact surfaces, and for efficient algorithms for constructing such a cell. Basu [Bas03] shows that the complexity of such a cell in a d-dimensional arrangement of n surfaces of constant description complexity is $O(n^{d-1+\epsilon})$, for any $\epsilon > 0$, where the constant of proportionality depends on d, ϵ, and the maximum degree of the polynomials defining the surfaces.

In contrast, computing such a cell within a comparable time bound remains an open problem.

COORDINATED MOTION PLANNING

Another class of motion-planning problems involves coordinated motion planning of several independently moving systems. Conceptually, this situation can be handled as just another special case of the general problem: Consider all the moving objects as a single system, with $k = \sum_{i=1}^{t} k_i$ degrees of freedom, where t is the number of moving objects, and k_i is the number of degrees of freedom of the ith object. However, k will generally be too large, and the problem then will be more difficult to tackle (see, e.g., Section 51.1.2).

A better approach is as follows [SS91]. Let $B_1, \ldots, B_t$ be the given independent objects. For each $i = 1, \ldots, t$, construct the free configuration space $\mathcal{F}^{(i)}$ for B_i alone (ignoring the presence of all other moving objects). The actual free configuration space $\mathcal{F}$ is a subset of $\prod_{i=1}^{t} \mathcal{F}^{(i)}$. Suppose we have managed to decompose each $\mathcal{F}^{(i)}$ into subcells of constant description complexity. Then $\mathcal{F}$ is a subset of the union of Cartesian products of the form $c_1 \times c_2 \times \cdots \times c_t$, where c_i is a subcell of $\mathcal{F}^{(i)}$.

We next compute the portion of $\mathcal{F}$ within each such product. Each such subproblem can be intuitively interpreted as the coordinated motion planning of our objects, where each moves within a small portion of space, amidst only a constant number of nearby obstacles; so these subproblems are much easier to solve. Moreover, in typical cases, for most products $P = c_1 \times c_2 \times \cdots \times c_t$ the problem is trivial, because P represents situations where the moving objects are far from one another, and so cannot interact at all, meaning that $\mathcal{F} \cap P = P$. The number of subproblems that really need to be solved will be relatively small.

The connectivity graph that represents $\mathcal{F}$ is also relatively easy to construct. Its nodes are the connected components of the intersections of $\mathcal{F}$ with each of the above cell products P, and two nodes are connected to each other if they are adjacent in the overall $\mathcal{F}$. In many typical cases, determining this adjacency is easy.

As an example, one can apply this technique to the coordinated motion planning of k disks moving in a planar polygonal environment bounded by n edges, to get a solution with $O(n^k)$ running time [SS91]. Since this problem has $2k$ degrees of freedom, this is a significant improvement over the bound $O(n^{2k} \log n)$ yielded by Canny's general algorithm.

See [ABS$^+$99] for another treatment of coordinated motion planning, for two or three general independently moving robots, where algorithms that are also faster than Canny's general technique are developed.

UNLABELED MOTION PLANNING

A recent variant of the coordinated motion-planning problem considers a collection of m identical and indistinguishable moving objects. Given a set of m initial and m final placements for the objects, the goal is to find a collision-free path for the objects. In contrast to the standard (labeled) formulation of the problem, here we are not interested in a specific assignment between robots and final placements as long as each final placement is occupied by some robot at the end of the motion.

In general, the unlabeled problem is PSPACE-hard [SH15] (as well as several simplified variants of the problem) for the case of unit squares. In contrast, the setting with unit disks can be solved efficiently, if one makes simplifying assumptions regarding the separation among the initial and final placements of the robots, and sometimes also between these placements and the obstacles. Adler et al. [AB$^+$15] give an $O(m^2 + mn)$-time algorithm that solves the problem for the case of simple polygonal environments. Turpin et al. [TuMK14] present an $O(m^4 + m^2n^2)$-time algorithm (with some additional poly-logarithmic factors), which also guarantees to return a solution that minimizes the length of the longest path of an individual robot. More recently, Solovey et al. [SYZH15] devised an algorithm with similar running time, which minimizes the sum of lengths of the individual paths. In particular, their algorithm returns a near-optimal solution whose additive approximation factor is $O(m)$.

A ROD IN A 3D POLYHEDRAL ENVIRONMENT

This problem has five degrees of freedom (three of translation and two of rotation). The complexity of $\mathcal{F}$ can be $\Omega(n^4)$ [KO87] in such a setting. This bound has almost been matched by Koltun [Kol05] who gave an $O(n^{4+\epsilon})$-time algorithm to solve this problem.

MOTION PLANNING AND ARRANGEMENTS

As can be seen from the preceding subsections, motion planning is closely related to the study of arrangements of surfaces in higher dimensions. Motion planning has motivated many problems in arrangements, such as the problem of bounding the complexity of, and designing efficient algorithms for, computing a single cell in an arrangement of n low-degree algebraic surface patches in d dimensions, the problem of computing the union of geometric objects (the expanded obstacles), and the problem of decomposing higher-dimensional arrangements into subcells of constant description complexity. These problems are only partially solved and present major challenges in the study of arrangements. See Chapter 30 and [SA95] for further details.

IMPLEMENTATION OF COMPLETE SOLUTIONS

Previously, complete solutions have rarely been implemented, mainly due to lack of the nontrivial infrastructure that is needed for such tasks. With the recent advancement in the laying out of such infrastructure, and in particular with tools now available in the software libraries LEDA [MN99] and CGAL [CGAL] (cf. Chapter 70), implementing complete solutions to motion planing has become feasible. A summary of progress and prospects in this domain can be found in [Hal02, FHW12].

SUMMARY

Some of the above results are summarized in Table 51.2.1. For each specific system, only one or two algorithms are listed.

TABLE 51.2.1 Summary of motion planning algorithms.

SYSTEM	MOTION	ENVIRONMENT	df	RUNNING TIME
Convex k-gon	translation	planar polygonal	2	$O(N \log m)$
Arbitrary k-gon	translation	planar polygonal	2	$O(kn \log^2 n)$
General			2	$O(\lambda_{s+2}(n) \log^2 n)$
Line segment	trans & rot	planar polygonal	3	$O(n^2 \log n)$
Convex k-gon	trans & rot	planar polygonal	3	$O(k^4 n \lambda_3(n) \log n)$
				$O(kn\lambda_6(kn) \log n)$
Arbitrary k-gon	trans & rot	planar polygonal	3	$O((kn)^{2+\epsilon})$
Convex polytope	translation	3D polyhedral	3	$O(kmn \log^2 m)$
Arbitrary polytope	translation	3D polyhedral	3	$O((kn)^{2+\epsilon})$
Ball		3D polyhedral	3	$O(n^{2+\epsilon})$
General 3 D.O.F.			3	$O(N^{2+\epsilon})$

51.2.4 PRACTICAL APPROACHES TO MOTION PLANNING

When the number of degrees of freedom is even moderately large, exact and complete solutions of the motion planning problem are very inefficient in practice, so one seeks heuristic or other incomplete but practical solutions. Several such techniques have been developed.

Potential field. The first heuristic regards the robot as moving in a potential field [Kha86] induced by the obstacles and by the target placement, where the obstacles act as repulsive barriers, and the target as a strongly attracting source. By letting the robot follow the gradient of such a potential field, we obtain a motion that avoids the obstacles and that can be expected to reach the goal. An attractive feature of this technique is that planning and executing the desired motion are done in a single stage. Another important feature is the generality of the approach; it can easily be applied to systems with many degrees of freedom.

This technique, however, may lead to a motion where the robot gets stuck at a local minimum of the potential field, leaving no guarantee that the goal will be reached (see [KB91] and references within). To overcome this problem, several solutions have been proposed. One is to try to escape from such a "potential well" by making a few small random moves, in the hope that one of them will put the robot in a position from which the field leads it away from this well. Another approach is to use the potential field only for subproblems where the initial and final placements are close to each other, so the chance to get stuck at a local minimum is small.

Probabilistic roadmaps. Over the past two decades, this method has picked up momentum, and has become the method of choice in many practical motion-planning systems [BKL+97, KSLO96, Lat91, CBH+05, Lav06, KLa08, MLL08].

The general approach is to generate many random free placements throughout the workspace, and to apply any "local" simple-minded planner to plan a motion between pairs of these placements; one may use for this purpose the potential field approach, or simply attempt to connect the two placements by a straight line segment in configuration space. If the configuration space is sufficiently densely sampled, enough local free paths will be generated, and they will form a roadmap, in the sense of Section 51.1.1, which can then be used to perform motion planning between any pair of input placements. These algorithms are categorized as *single-query* algorithms such as the Rapidly Exploring Random Tree (RRT) [LK99], or *multi-query* algorithms such as the Probabilistic Roadmap Method [KSLO96]. Many of these algorithms are *probabilistically complete*. Namely, the probability that they will return a solution (if one exists) approaches one as the number of samples tends to infinity. By now, both *asymptotically optimal* and *asymptotically near-optimal* variants were devised (see e.g., [KF11, JSCP15, SH14, DB14] for a partial list). We say that an algorithm is asymptotically (near)-optimal if the cost of the solution returned by the algorithm (nearly) approaches the cost of the optimal solution as the number of samples tends to infinity.

Interestingly, sampling-based motion-planning algorithms have been applied to molecular simulations. Specifically, these algorithms have been used for the analysis of conformational transitions, protein folding and unfolding, and protein-ligand interactions [ASC12].

A significant problem that arises is how to sample well the free configuration space; informally, the goal is to detect all "tight" passages within $\mathcal{F}$, which will be missed unless some placements are generated near them. See [ABD+98, BKL+97, HLM99, KSLO96, KL01] and Chapter 52 for more details concerning this technique, its extensions and variants.

The geometric methods for exact and complete analysis of low-dimensional configuration spaces as described so far in this chapter are combined in [SHRH13, SHH15] with the practical, considerably simpler sampling-based approaches that are appropriate for higher dimensions. This is done by taking samples that are entire low-dimensional manifolds of the configuration space and that capture the connectivity of the configuration space much better than isolated point samples. To do so, on each low-dimensional manifold an arrangement is computed, which subdivides the manifold into free and forbidden regions. Subsequently, geometric algorithms for analysis of low-dimensional manifolds provide powerful primitive operations to construct a roadmap-like data structure. Experiments show that although this hybrid approach uses heavy machinery of exact algebraic computing, it significantly outperforms the sampling-based algorithms in tight settings, where the robots need to move in densely cluttered environments.

Fat obstacles. Another technique exploits the fact that, in typical layouts, the obstacles can be expected to be "fat" (this has several definitions; intuitively, they do not have long and skinny parts). Also, the obstacles tend not to be too clustered, in the sense that each placement of the robot can interact with only a constant number of obstacles. These facts tend to make the problem easier to solve in such so-called *realistic* input scenes. See [SHO93] for the case of fat obstacles, [SOBV98] for the case of environments with low obstacle density, and [BKO^{+}02] for two other models of realistic input scenes.

51.3 VARIANTS OF THE MOTION PLANNING PROBLEM

We now briefly review several variants of the basic motion planning problem, in which additional constraints are imposed on the problem. Further material on many of these problems can be found in Chapter 52.

OPTIMAL MOTION PLANNING

The preceding section described techniques for determining the existence of a collision-free motion between two given placements of some moving system. It paid no attention to the optimality of the motion, which is an important consideration in practice. There are several problems involved in optimal motion planning. First, optimality is a notion that can be defined in many ways, each of which leads to different algorithmic considerations. Second, optimal motion planning is usually much harder than motion planning per se.

SHORTEST PATHS

The simplest case is when the moving system B is a single point. In this case the cost of the motion is simply the length of the path traversed by the point (normally, we use the Euclidean distance, but other metrics have been considered as well). We thus face the problem of computing ***shortest paths*** amidst obstacles in a 2D or 3D environment.

The planar case. Let V be a closed planar polygonal environment bounded by n edges, and let s (the "source") be a point in V. For any other point $t \in V$, let $\pi(s,t)$ denote the (Euclidean) shortest path from s to t within V. Finding $\pi(s,t)$ for any t is facilitated by construction of the *shortest path map* $SPM(s,V)$ from s

in V, a decomposition of V into regions detailed in Chapter 33. Computing the map can be done in optimal $O(n \log n)$ time [HS99].

The same problem may be considered in other metrics. For example, it is easier to give an $O(n \log n)$ algorithm for the shortest path problem under the L_1 or L_∞ metric. See Chapter 33. Another issue that arrises in this context is the *clearance* of the path (namely, the minimal distance to the closest obstacle). The Euclidean shortest path may touch obstacle boundaries and therefore its clearance at certain points may be zero. Conversely, if maximizing the distance from the obstacles is the main optimization criterion, then the path can be computed by constructing a minimum spanning tree in the Voronoi diagram of the obstacles [OY85] in $O(n \log n)$ time. Wein et al. [WBH07] considered the problem of computing shortest paths that have a minimal given clearance. Specifically, they precompute a data structure called the *visibility Voronoi complex* in time $O(n^2 \log n)$ which allows to compute shortest paths that have clearance at least δ, for any specified parameter δ. An alternative measure to quantify the tradeoff between the length and the clearance was suggested by Wein et al. [WBH08] where the optimization criterion is minimizing the reciprocal of the clearance, integrated over the length of the path. While it is still not known whether the problem of computing the optimal path in this measure is NP-hard, only recently, the first polynomial-time approximation algorithm for this problem was proposed [AFS16]; it produces a $(1+\epsilon)$-approximation in time $O(\frac{n^2}{\epsilon^2} \log \frac{n}{\epsilon})$.

The three-dimensional case. Let V be a closed polyhedral environment bounded by a total of n faces, edges, and vertices. Again, given two points $s, t \in V$, we wish to compute the shortest path $\pi(s,t)$ within V from s to t. Here $\pi(s,t)$ is a polygonal path, bending at *edges* (sometimes also at vertices) of V. To compute $\pi(s,t)$, we need to solve two subproblems: to find the sequence of edges (and vertices) of V visited by $\pi(s,t)$ (the *shortest-path sequence* from s to t), and to compute the actual points of contact of $\pi(s,t)$ with these edges. These points obey the rule that the incoming angle of $\pi(s,t)$ with an edge is equal to the outgoing angle. Hence, given the shortest-path sequence of length m, we need to solve a system of m polynomial equations in m variables in order to find the contact points; each equation turns out to be quadratic. This can be solved either approximately, using an iterative scheme, or exactly, using techniques of computational real algebraic geometry; the latter method requires exponential time. Even the first, more "combinatorial," problem of computing the shortest-path sequence is NP-hard [CR87], so the general shortest-path problem is certainly much harder in three dimensions.

Many special cases of this problem, with more efficient solutions, have been studied. The simplest instance is the problem of computing shortest paths on a convex polytope. Schreiber and Sharir [SS08] present an optimal $O(n \log n)$ algorithm, following earlier near-quadratic solutions [MMP87, CH96] and a simple linear-time approximation [AHSV97]. A related problem involves shortest paths on a polyhedral terrain, where near-quadratic exact algorithms, as well as more efficient approximation algorithms are known [MMP87, VA01, LMS97]. Approximation algorithms were also developed for computing shortest paths in weighted polyhedral surfaces [ADG$^+$10] and in polyhedral domains [ADMS13]. See also Chapter 33.

VARIOUS OPTIMAL MOTION PLANNING PROBLEMS

Suppose next that the moving system B is a rigid body free only to translate in two or three dimensions. Then the notion of optimality is still well defined—it is the total distance traveled by (any reference point attached to) B. One can then apply the same techniques as above, after replacing the obstacles by their expanded versions. For example, if B is a convex polygon in the plane, and the obstacles are m pairwise openly-disjoint convex polygons $A_1, \ldots, A_m$, then we form the Minkowski sums $K_i = A_i \oplus (-B)$, for $i = 1, \ldots, m$, and compute a shortest path in the complement of their union. Since the K_i's may overlap, we first need to compute the complement of their union, as above. A similar approach can be used for planning shortest motion of a polyhedron translating amidst polyhedra in 3-space, etc.

If B admits more complex motions, then the notion of optimality begins to be fuzzy. For example, consider the case of a line segment ("rod") translating and rotating in a planar polygonal environment. One could measure the cost of a motion by the total distance traveled by a designated endpoint (or the centerpoint) of B, or by a weighted average between such a distance and the total turning angle of B, etc. A version of this problem has been shown to be NP-hard [AKY96]. See Chapter 33.

The notion of optimality gets even more complicated when one introduces kinematic constraints on the motion of B (for example, bounds on the radius of the curvature of the path [AW01, ABL$^+$02]). It is then often challenging even without obstacles; see Chapter 52.

MOTION PLANNING FOR A TETHERED ROBOT

An interesting family of motion-planning problems occurs when the robot is anchored to a fixed base point by a finite tether. In the basic form of this problem, the tether is not an obstacle (the robot may drive over it), and the additional constraint is to ensure that the robot does not get too far from the base. The objective is to compute the shortest path of the robot between any two given points while satisfying the constraint that the distance between the base and the robot following the tether does not exceed the tether's length. Xu et al. [XBV15] study the problem in the plane and give an algorithm that runs in time $O(kn^2 \log n)$ where n is the number of obstacle vertices and k is the number of segments in the polyline defining the initial tether configuration. Salzman and Halperin [SH15] considered the problem of preprocessing a planar workspace to efficiently answer multiple queries. Their work relies on an extension of the *visibility graph* which encodes for each vertex of the graph, all homotopy classes that can be used to reach that vertex using a tether of predefined length.

A different variant was considered by Hert and Lumelsky [HL99] who study the motion of multiple tethered point robots in a workspace with no obstacles. This problem focuses on allowing each robot to reach a goal without undue tangling. They devised algorithms which take start and goal configurations for the robots and produce an ordering of the robots. In the planar case [HL96], the tethers are tangled, but in a prespecified way, and the problem is to find an ordering for the robots' motions such that the goal is reached with the specified tether locations. This is done in $O(n^3 \log n)$ time where n is the number of robots. In the spatial case [HL99] (applicable to multiple underwater vehicles), the tethers remain untangled, and the problem is to find an ordering in which to move the robots such that tangling does not occur. This algorithm runs in $O(n^4)$ time.

EXPLORATORY MOTION PLANNING

If the environment in which the robot moves is not known to the system a priori, but the system is equipped with sensory devices, motion planning assumes a more "exploratory" character. If only tactile (or proximity) sensing is available, then a plausible strategy might be to move along a straight line (in physical or configuration space) directly to the target position, and when an obstacle is reached, to follow its boundary until the original straight line of motion is reached again. This technique has been developed and refined for arbitrary systems with two degrees of freedom (see, e.g., [LS87]). It can be shown that this strategy provably reaches the goal, if at all possible, with a reasonable bound on the length of the motion. This technique has been implemented on several real and simulated systems, and has applications to maze-searching problems.

One attempt to extend this technique to a system with three degrees of freedom is given in [CY91]. This technique computes within $\mathcal{F}$ a certain one-dimensional skeleton (roadmap) $\mathcal{R}$ which captures the connectivity of $\mathcal{F}$. The twist here is that $\mathcal{F}$ is not known in advance, so the construction of $\mathcal{R}$ has to be done in an incremental, exploratory manner. This exploration can be implemented in a controlled manner that does not require too many "probing" steps, and which enables the system to recognize when the construction of $\mathcal{R}$ has been completed (if the goal has not been reached beforehand).

If vision is also available, then other possibilities need to be considered, e.g., the system can obtain partial information about its environment by viewing it from the present placement, and then "explore" it to gain progressively more information until the desired motion can be fully planned. Results that involve such *model-building* tasks can be found in [GMR97, ZF96]. Online algorithms for mobile robots that use vision for searching a target and for exploring a region in the plane are surveyed in [GK10]. Variants of this basic problem have been introduced which include the use of only a discrete number of visibility queries [FS10, FMS12] or minimizing the number of turns that the robot performs while exploring [DFG06].

This problem is closely related to the problem of *coverage* in which the robot is equipped with the task of determining a path that passes over all points of an area or volume of interest while avoiding obstacles. For surveys on the subject see [Cho01, GC13].

This problem becomes substantially harder when errors in localization and in mapping exist. However, by combining localization and mapping into one process, the error converges. This paradigm, called *Simultaneous Localization and Mapping*, or SLAM, is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of the robot's location within it. Pioneering work in this field include the work by Smith et al. [SC86] on representing and estimating spatial uncertainty. This has become a thriving area of research; see, e.g., the surveys [DB06a, DB06b] and Chapter 35.

TIME-VARYING ENVIRONMENTS

Interesting generalizations of the motion planning problem arise when some of the obstacles in the robot's environment are assumed to be moving along known trajectories. In this case the robot's goal will be to "dodge" the moving obstacles while moving to its target placement. In this "kinetic" motion planning problem, it is reasonable to assume some limit on the robot's velocity and/or acceleration. Two studies of this problem are [SM88, RS94]. They show that the problem of avoiding moving obstacles is substantially harder than the corresponding static problem. By using time-related configuration changes to encode Turing machine states, they show that the problem is PSPACE-hard even for systems with a small and fixed number of degrees of freedom. However, polynomial-time algorithms are available in a few particularly simple special cases. Another variant of this problem involves movable obstacles, which the robot B can, say, push aside to clear its passage. Again, it can be shown that the general problem of this kind is PSPACE-hard, some special instances are NP-hard, and polynomial-time algorithms are available in certain other special cases [Wil91, DZ99]. There exist sampling-based planners (see., e.g., [NSO06]) that solve this problem successfully using heuristics to provide efficient solutions in time-varying environments encountered in practical situations.

COMPLIANT MOTION PLANNING

In realistic situations, the moving system has only approximate knowledge of the geometry of the obstacles and/or of its current position and velocity, and it has an inherent amount of error in controlling its motion. The objective is to devise a strategy that will guarantee that the system reaches its goal, where such a strategy usually proceeds through a sequence of free motions (until an obstacle is hit) intermixed with *compliant motions* (sliding along surfaces of contacted obstacles) until it can be ascertained that the goal has been reached.

A standard approach to this problem is through the construction of pre-images (or back projections) [LPMT84]. Specific algorithms that solve various special cases of the problem can be found in [Bri89, Don90, FHS96]. See Chapter 52.

NONHOLONOMIC MOTION PLANNING

Another realistic constraint on the possible motions of a given system is kinematic (or *kinodynamic*). For example, the moving object B might be constrained not to exceed certain velocity or acceleration thresholds, or has only limited steering capability. Even without any obstacles, such problems are usually quite hard, and the presence of (stationary or moving) obstacles makes them extremely complicated to solve. These so-called *nonholonomic motion planning* problems are usually handled using tools from control theory. A relatively simple special case is that of a car-like robot in a planar workspace, with a bound on the radius of curvature of its motion. Issues like reachability between two given placements (even in the absence of obstacles) raise interesting geometric considerations, where one of the goals is to identify canonical motions that always suffice to get to any reachable placement. See [Lat91, LC92, Lau98] for several books that cover this topic, and Chapter 52. Kinodynamic motion planning is treated in [CDRX88, CRR91]. The problem of finding a shortest curvature-constrained path in a polygonal domain with holes was recently shown to be NP-Hard [KKP11]. Simplified cases of this problem as well as approximation algorithms are treated in [AW01, RW98, ABL$^+$02, ACMV12, BK05, BB07, KC13].

GENERAL TASK AND ASSEMBLY PLANNING

In task planning problems, the system is given a complex task to perform, such as assembling a part from several components or restructuring its workcell into a new layout, but the precise sequence of substeps needed to attain the final goal is not specified and must be inferred by the system.

Suppose we want to manufacture a product consisting of several parts. Let S be the set of parts in their final assembled form. The first question is whether the product can be disassembled by translating in some fixed direction one part after the other, so that no collision occurs. An order of the parts that satisfies this property is called a ***depth order***. It need not always exist, but when it does, the product can be assembled by translating the constituent parts one after another, in the reverse of the depth order, to their target positions. Products that can be assembled in this manner are called ***stack products*** [WL94]. The simplicity of the assembly process makes stack products attractive to manufacture. Computing a depth order in a given direction (or deciding that no such order exists) can be done in $O(m^{4/3+\epsilon})$ time, for any $\epsilon > 0$, for a set of polygons in 3-space with m vertices in total [BOS94]. Faster algorithms are known for the special cases of axis-parallel polygons, c-oriented polygons, and "fat" objects.

Many products, however, are not stack products, that is, a single direction in which the parts must be moved is not sufficient to (dis)assemble the product. One solution is to search for an assembly sequence that allows a subcollection of parts to be moved as a rigid body in *some* direction. This can be accomplished in polynomial time, though the running time is rather high in the worst case: it may require $\Omega(m^4)$ time for a collection of m tetrahedra in 3-space [WL94]. A more modest, but considerably more efficient, solution allows each disassembly step to proceed in one of a few given directions [ABHS96]. It has running time $O(m^{4/3+\epsilon})$, for any $\epsilon > 0$.

A general approach to assembly planning, based on the concept of a *nondirectional blocking graph* [WL94], is proposed in [HLW00]. It is called the *motion space approach*, where the motion space plays a role parallel to configuration space in motion planning. Every point in the motion space represents a possible (dis)assembly sequence motion, all having the same number of degrees of freedom. The motion space is decomposed into an arrangement of cells where in each cell the blocking relations among the parts are invariant, namely, for a every pair of parts P, Q, P will either hit Q for all the possible motions of a cell, or avoid it. It thus suffices to check one specific motion sequence from each cell, leading to a finite complete solution.

Often we restrict ourselves to two-handed partitioning steps, meaning that we partition the given assembly into two complementing subsets each treated as a rigid body. Even for two-handed partitioning, if we allow arbitrary translational motions the problem becomes NP-hard [KK95]. When we restrict ourselves to infinite translations, efficient algorithms together with exact implementations [FH13] exist. For a recent survey on assembly planning, see [Jim13]. Sampling-based motion-planning algorithms (see Section 51.2.4) have been used to (heuristically) overcome the hardness of assembly planning (see, e.g., [CJS08, SHH15]).

See Chapter 52 and [HML91] for further details on assembly sequencing, and Chapter 59 for related problems.

ON-LINE MOTION PLANNING

Consider the problem of a point robot moving through a planar environment filled with polygonal obstacles, where the robot has no a priori information about the obstacles that lie ahead. One models this situation by assuming that the robot knows the location of the target position and of its own absolute position, but that it only acquires knowledge about the obstacles as it contacts them. The goal is to minimize the distance that the robot travels. See also the discussion on exploratory motion planning above.

Because the robot must make decisions without knowing what lies ahead, it is natural to use the ***competitive ratio*** to evaluate the performance of a strategy. In particular, one would like to minimize the ratio between the distance traveled by the robot and the length of the shortest start-to-target path in that scene. The competitive ratio is the worst-case ratio achieved over all scenes having a given source-target distance. A special case of interest is when all obstacles are axis-parallel rectangles of width at least 1 located in Euclidean plane. Natural greedy strategies yield a competitive ratio of $\Theta(n)$, where n is the Euclidean source-target distance. More sophisticated algorithms obtain competitive ratios of $\Theta(\sqrt{n})$ [BRS97]. Randomized algorithms can do much better [BBF$^+$96]. Through the use of randomization, one can transform the case of arbitrary convex obstacles [BRS97] to rectilinearly-aligned rectangles, at the cost of some increase in the competitive ratio. If the scene is not on an infinite plane but rather within some finite rectangular "warehouse," and the start location is one of the warehouse corners, then the competitive ratio drops to $\log n$ [BBFY94].

COLLISION DETECTION

Although not a motion planning problem per se, collision detection is a closely related problem in robotics [LG98]. It arises, for example, when one tries to use some heuristic approach to motion planning, where the planned path is not guaranteed apriori to be collision-free. In such cases, one wishes to test whether collisions occur during the proposed motion. Collision detection is also used as a primitive operation in sampling-based algorithms (see Section 51.2.4 and Chapter 52). Several methods have been developed, including: (a) Keeping track of the closest pair of features between two objects, at least one of which is moving, and updating the closest pair, either at discrete time steps, or using *kinetic data structures* (Chapter 54). (b) Using a hierarchical representation of more complex moving systems, by means of bounding boxes or spheres, and testing for collision recursively through the hierarchical representation (see, e.g., [LGLM00, TaMK14] and references therein). See Chapter 40 for more details.

51.4 SOURCES AND RELATED MATERIAL

SURVEYS

The results not given an explicit reference above, and additional material on motion planning and related problems may be traced in these surveys:

[Lat91, CBH+05, Lav06, KLa08, MLL08]: Books or chapters of books devoted to robot motion planning.

[HSS87]: A collection of early papers on motion planning.

[SA95]: A book on Davenport-Schinzel sequences and their geometric applications; contains a section on motion planning.

[HS95b]: A review on arrangements and their applications to motion planning.

[SS88, SS90, Sha89, Sha95, AY90]: Several survey papers on algorithmic motion planning.

[AS00b, AS00c]: Surveys on Davenport-Schinzel sequences and on higher-dimensional arrangements.

RELATED CHAPTERS

Chapter 29: Voronoi diagrams and Delaunay triangulations
Chapter 30: Arrangements
Chapter 33: Shortest paths and networks
Chapter 38: Computational and quantitative real algebraic geometry
Chapter 40: Collision and proximity queries
Chapter 52: Robotics
Chapter 54: Modeling motion
Chapter 70: Two computation geometry libraries: LEDA and CGAL

REFERENCES

[AB+15] A. Adler, M. de Berg, D. Halperin, and K. Solovey. Efficient multi-robot motion planning for unlabeled discs in simple polygons. *IEEE Trans. Autom. Sci. Eng.*, 12:1309–1317, 2015.

[AAS98] P.K. Agarwal, N. Amenta, and M. Sharir. Largest placement of one convex polygon inside another. *Discrete Comput. Geom.*, 19:95–104, 1998.

[AAS99] P.K. Agarwal, B. Aronov, and M. Sharir. Motion planning for a convex polygon in a polygonal environment. *Discrete Comput. Geom.*, 22:201–221, 1999,

[ABHS96] P.K. Agarwal, M. de Berg, D. Halperin, and M. Sharir. Efficient generation of k-directional assembly sequences. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 122–131, 1996.

[ABL+02] P. K. Agarwal, T. C. Biedl, S. Lazard, S. Robbins, S. Suri, and S. Whitesides. Curvature-constrained shortest paths in a convex polygon. *SIAM J. Comput.*, 31:1814–1851, 2002.

[AFH02] P.K. Agarwal, E. Flato, and D. Halperin. Polygon decomposition for efficient construction of Minkowski sums. *Comput. Geom. Theory Appl.*, 21:39–61, 2002.

[AFS16] P.K. Agarwal, K. Fox, and O. Salzman. An efficient algorithm for computing high-quality paths amid polygonal obstacles. In *Proc. 27th ACM-SIAM Sympos. Discrete Algorithms*, 2016.

[AHSV97] P.K. Agarwal, S. Har-Peled, M. Sharir, and K.R. Varadarajan. Approximate shortest paths on a convex polytope in three dimensions. *J. ACM*, 44:567–584, 1997.

[AS00a] P.K. Agarwal and M. Sharir. Pipes, cigars, and kreplach: The union of Minkowski sums in three dimensions. *Discrete Comput. Geom.*, 24:645–685, 2000.

[AS00b] P.K. Agarwal and M. Sharir. Davenport-Schinzel sequences and their geometric applications. In J.R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 1–47, Elsevier North-Holland, Amsterdam, 2000.

[AS00c] P.K. Agarwal and M. Sharir. Arrangements of surfaces in higher dimensions. in J.R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 49–119, Elsevier North-Holland, Amsterdam, 2000.

[AW01] P.K. Agarwal and H. Wang. Approximation algorithms for shortest paths with bounded curvature. *SIAM J. Comput.*, 30:1739–1772, 2001.

[ACMV12] H.-K. Ahn, O. Cheong, J. Matoušek, and A. Vigneron. Reachability by paths of bounded curvature in a convex polygon. *Comput. Geom.*, 45(1-2): 21–32, 2012.

[ASC12] I. Al-Bluwi, T. Siméon, and J. Cortés Motion planning algorithms for molecular simulations: A survey. *Computer Science Review*, 6(4): 125–143 (2012)

[ADG+10] L. Aleksandrov, H. Djidjev, H. Guo, A. Maheshwari, D. Nussbaum, and J.-R. Sack. Algorithms for approximate shortest path queries on weighted polyhedral surfaces. *Discrete Comput. Geom.*, 44:762–801, 2010.

[ADMS13] L. Aleksandrov, H. Djidjev, A. Maheshwari, and J.-R. Sack. An approximation algorithm for computing shortest paths in weighted 3-d domains. *Discrete Comput. Geom.*, 50:124–184, 2013.

[AFK+92] H. Alt, R. Fleischer, M. Kaufmann, K. Mehlhorn, S. Näher, S. Schirra, and C. Uhrig. Approximate motion planning and the complexity of the boundary of the union of simple geometric figures. *Algorithmica*, 8:391–406, 1992.

[AY90] H. Alt and C.K. Yap. Algorithmic aspects of motion planning: A tutorial, Parts 1 and 2. *Algorithms Rev.*, 1:43–60 and 61–77, 1990.

[ABD+98] N.M. Amato, B. Bayazit, L. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Robotics: The Algorithmic Perspective (WAFR'98)*, pages 155–168, A.K. Peters, Wellesley, 1998.

[ABS+99] B. Aronov, M. de Berg, A.F. van der Stappen, P. Švestka, and J. Vleugels. Motion planning for multiple robots. *Discrete Comput. Geom.*, 22:505–525, 1999.

[AS94] B. Aronov and M. Sharir. Castles in the air revisited. *Discrete Comput. Geom.*, 12:119–150, 1994.

[AS97] B. Aronov and M. Sharir. On translational motion planning of a convex polyhedron in 3-space. *SIAM J. Comput.*, 26:1785–1803, 1997.

[AKY96] Te. Asano, D.G. Kirkpatrick, and C.K. Yap. d_1-optimal motion for a rod. In *Proc. 12th Sympos. Comput. Geom.*, pages 252–263, ACM Press, 1996.

[ABF89] F. Avnaim, J.-D. Boissonnat, and B. Faverjon. A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. In *Geometry and Robotics*, volume 391 of *Lecture Notes Comp. Sci.*, pages 67–86, Springer, Berlin, 1989.

[BB07] J. Backer and D. G. Kirkpatrick. Finding curvature-constrained paths that avoid polygonal obstacles. In *Proc. 23rd Sympos. Comput. Geom.*, pages 66–73, ACM Press, 2007.

[BBFY94] E. Bar-Eli, P. Berman, A. Fiat, and P. Yan. On-line navigation in a room. *J. Algorithms*, 17:319–341, 1994.

[BFH+15] A. Baram, E. Fogel, D. Halperin, M. Hemmer, and S. Morr. Exact Minkowski sums of polygons with holes. In *Proc. European Sympos. Algorithms*, volume 9294 of *Lecture Notes Comp. Sci.*, pages 71–82, Springer, Berlin, 2015.

[BKL+97] J. Barraquand, L.E. Kavraki, J.-C. Latombe, T.-Y. Li, R. Motwani, and P. Raghavan. A random sampling framework for path planning in large-dimensional configuration spaces. *Internat. J. Robot. Res.*, 16:759–774, 1997.

[Bas03] S. Basu. On the combinatorial and topological complexity of a single cell. *Discrete Comput. Geom.*, 29:41–59, 2003.

[BPR00] S. Basu, R. Pollack, and M.-F. Roy. Computing roadmaps of semi-algebraic sets on a variety. *J. AMS*, 13:55–82, 2000.

[BK05] S. Bereg and D. G. Kirkpatrick. Curvature-bounded traversals of narrow corridors. In *Proc. 21st Sympos. Comput. Geom.*, pages 278–287, ACM Press, 2005.

[BDS95] M. de Berg, K. Dobrindt, and O. Schwarzkopf. On lazy randomized incremental construction. *Discrete Comput. Geom.*, 14:261–286, 1995.

[BKO+02] M. de Berg, M.J. Katz, M.H. Overmars, A.F. van der Stappen, and J. Vleugels. Models and motion planning. *Comput. Geom. Theory Appl.*, 23:53–68, 2002.

[BOS94] M. de Berg, M.H. Overmars, and O. Schwarzkopf. Computing and verifying depth orders. *SIAM J. Comput.*, 23:437–446, 1994.

[BBF+96] P. Berman, A. Blum, A. Fiat, H. Karloff, A. Rosen, and M. Saks. Randomized robot navigation algorithms. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 75–84, 1996.

[BRS97] A. Blum, P. Raghavan, and B. Schieber. Navigating in unfamiliar geometric terrain. *SIAM J. Comput.*, 26:110–137, 1997.

[Bri89] A.J. Briggs. An efficient algorithm for one-step planar compliant motion planning with uncertainty. In *Proc. 5th Sympos. Comput. Geom.*, pages 187–196, ACM Press, 1989.

[CGAL] CGAL, The Computational Geometry Algorithms Library. `http://www.cgal.org`.

[CJS08] J. Cortés, L. Jaillet, and T. Siméon Disassembly path planning for complex articulated objects. *IEEE Trans. Robot.*, 24:475–481, 2008

[Can87] J.F. Canny. *The complexity of robot motion planning.* MIT Press, Cambridge, 1987. See also: Computing roadmaps in general semi-algebraic sets. *Comput. J.*, 36:504–514, 1993.

[CDRX88] J.F. Canny, B.R. Donald, J.H. Reif, and P. Xavier. On the complexity of kinodynamic planning. In *Proc. 29th IEEE Sympos. Found. Comput. Sci.*, pages 306–316, 1988.

[CRR91] J.F. Canny, A. Rege, and J.H. Reif. An exact algorithm for kinodynamic planning in the plane. *Discrete Comput. Geom.*, 6:461–484, 1991.

[CR87] J.F. Canny and J.H. Reif. New lower bound techniques for robot motion planning problems. In *Proc. 28th IEEE Sympos. Found. Comput. Sci.*, pages 49–60, 1987.

[CEG+93] B. Chazelle, H. Edelsbrunner, L.J. Guibas, M. Sharir, and J. Snoeyink. Computing a face in an arrangement of line segments and related problems. *SIAM J. Comput.*, 22:1286–1302, 1993.

[CH96] J. Chen and Y. Han. Shortest paths on a polyhedron. *Internat. J. Comput. Geom. Appl.*, 6:127–144, 1996.

[CK93] L.P. Chew and K. Kedem. A convex polygon among polygonal obstacles: placement and high-clearance motion. *Comput. Geom. Theory Appl.*, 3:59–89, 1993.

[CKS+98] L.P. Chew, K. Kedem, M. Sharir, B. Tagansky, and E. Welzl. Voronoi diagrams of lines in three dimensions under polyhedral convex distance functions. *J. Algorithms*, 29:238–255, 1998.

[Cho01] H. Choset. Coverage for robotics - A survey of recent results. *Ann. Math. Artif. Intell.*, 31(1-4):113–126, 2001.

[CBH+05] H. Choset, W. Burgard, S. Hutchinson, G. Kantor, L. E. Kavraki, K. Lynch, and S. Thrun. *Principles of robot motion: Theory, algorithms, and implementation.* MIT Press, 2005.

[Col75] G.E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Proc. 2nd GI Conf. Automata Theory Formal Languages*, volume 33 of *Lecture Notes Comp. Sci.*, pages 134–183, Springer-Verlag, Berlin, 1975.

[CY91] J. Cox and C.K. Yap. On-line motion planning: Case of a planar rod. *Ann. Math. Artif. Intell.*, 3:1–20, 1991.

[DFG06] E. D. Demaine, S. Fekete, and S. Gal. Online searching with turn cost. *Theor. Comput. Sci.*, 361:342–355, 2006.

[DB14] A. Dobson and K. Bekris. Sparse roadmap spanners for asymptotically near-optimal motion planning. *Internat. J. Robot. Res.*, 33:18–47, 2014.

[Don90] B.R. Donald. The complexity of planar compliant motion planning under uncertainty. *Algorithmica*, 5:353–382, 1990.

[DZ99] D. Dor and U. Zwick. SOKOBAN and other motion planning problems. *Comput. Geom. Theory Appl.*, 13:215–228, 1999.

[DB06b] H.F. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part II. *IEEE Robot. Automat. Mag.*, 13:108–117, 2006.

[DB06a] H.F. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robot. Automat. Mag.*, 13:99–110, 2006.

[Ezr11] E. Ezra. On the union of cylinders in three dimensions. *Discrete Comput. Geom.*, 45:45–46, 2011.

[FH07] E. Fogel and D. Halperin. Exact and efficient construction of Minkowski sums of convex polyhedra with applications. *Computer-Aided Design*, 39:929–940, 2007.

[FH13] E. Fogel and D. Halperin. Polyhedral assembly partitioning with infinite translations or the importance of being exact. *IEEE T. Aut. Sci. and Eng.*, 10:227–247, 2013.

[FHW12] E. Fogel, D. Halperin, and R. Wein. *CGAL arrangements and their applications: A step-by-step guide.* volume 7 of *Geometry and Computing*, Springer-Verlag, Berlin, 2012.

[FHS96] J. Friedman, J. Hershberger, and J. Snoeyink. Efficiently planning compliant motion in the plane. *SIAM J. Comput.*, 25:562–599, 1996.

[FMS12] S. Fekete, J.S.B. Mitchell, and C. Schmidt. Minimum covering with travel cost. *J. Comb. Optim.*, 24:32–51, 2012.

[FS10] S. Fekete and C. Schmidt. Polygon exploration with time-discrete vision. *Comput. Geom.*, 43:148–168, 2010.

[GC13] E. Galceran and M. Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61:1258–1276, 2013.

[GK10] S. K. Ghosh and R. Klein. Online algorithms for searching and exploration in the plane. *Computer Science Review*, 4(:189–201, 2010.

[GMR97] L.J. Guibas, R. Motwani, and P. Raghavan. The robot localization problem. *SIAM J. Comput.*, 26:1120–1138, 1997.

[GSS89] L.J. Guibas, M. Sharir, and S. Sifrony. On the general motion planning problem with two degrees of freedom. *Discrete Comput. Geom.*, 4:491–521, 1989.

[Hac09] P. Hachenberger. Exact Minkowksi sums of polyhedra and exact and efficient decomposition of polyhedra into convex pieces. *Algorithmica*, 55:329–345, 2009.

[Hal02] D. Halperin. Robust geometric computing in motion. *Internat. J. Robot. Res.*, 21:219–232, 2002.

[HY98] D. Halperin and C.K. Yap. Combinatorial complexity of translating a box in polyhedral 3-space. *Comput. Geom. Theory Appl.*, 9:181–196, 1998.

[HLW00] D. Halperin, J.-C. Latombe, and R.H. Wilson. A general framework for assembly planning: The motion space approach. *Algorithmica*, 26:577–601, 2000.

[HOS92] D. Halperin, M.H. Overmars, and M. Sharir. Efficient motion planning for an L-shaped object in the plane. *SIAM J. Comput.* 21:1–23, 1992.

[HS95a] D. Halperin and M. Sharir. Almost tight upper bounds for the single cell and zone problems in three dimensions. *Discrete Comput. Geom.*, 14:385–410, 1995.

[HS95b] D. Halperin and M. Sharir. Arrangements and their applications in robotics: Recent developments. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, editors, *The Algorithmic Foundations of Robotics*, pages 495–511, A.K. Peters, Boston, 1995.

[HS96] D. Halperin and M. Sharir. A near-quadratic algorithm for planning the motion of a polygon in a polygonal environment. *Discrete Comput. Geom.*, 16:121–134, 1996.

[HCA+95] S. Har-Peled, T.M. Chan, B. Aronov, D. Halperin, and J. Snoeyink. The complexity of a single face of a Minkowski sum. In *Proc. 7th Canad. Conf. Comput. Geom.*, Québec City, pages 91–96, 1995.

[HD05] R.A. Hearn and E.D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theor. Comput. Sci.*, 343:72–96, 2005.

[HD09] R.A. Hearn and E.D. Demaine. *Games, puzzles and computation.* A. K. Peters, Boston, 2009.

[HS99] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Comput.*, 28:2215–2256, 1999.

[HL96] S. Hert and V.J. Lumelsky. The ties that bind: Motion planning for multiple tethered robots. *Robotics and Autonomous Systems*, 17:187–215, 1996.

[HL99] S. Hert and V.J. Lumelsky. Motion planning in $\mathbb{R}^3$ for multiple tethered robots. *IEEE T. Rob. and Aut.*, 15:623–639, 1999.

[HML91] L.S. Homem de Mello and S. Lee, editors. *Computer-aided Mechanical Assembly Planning.* Kluwer Academic Publishers, Boston, 1991.

[HJW84] J.E. Hopcroft, D.A. Joseph, and S.H. Whitesides. Movement problems for 2-dimensional linkages. *SIAM J. Comput.*, 13:610–629, 1984.

[HJW85] J.E. Hopcroft, D.A. Joseph, and S.H. Whitesides. On the movement of robot arms in 2-dimensional bounded regions. *SIAM J. Comput.* 14:315–333, 1985.

[HSS84] J.E. Hopcroft, J.T. Schwartz, and M. Sharir. On the complexity of motion planning for multiple independent objects: P-space hardness of the “Warehouseman’s Problem.” *Internat. J. Robot. Res.*, 3:76–88, 1984.

[HSS87] J.E. Hopcroft, J.T. Schwartz, and M. Sharir, editors. *Planning, Geometry, and Complexity of Robot Motion.* Ablex, Norwood, 1987.

[HLM99] D. Hsu, J.-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *Internat. J. Comput. Geom. Appl.*, 9:495–512, 1999.

[JSCP15] L. Janson, E. Schmerling, A.A. Clark, and M. Pavone. Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *Internat. J. Robot. Res.*, 34:883–921, 2015.

[Jim13] P. Jiménez. Survey on assembly sequencing: a combinatorial and geometrical perspective *J. Intelligent Manufacturing*, 24:235–250, 2013.

[KF11] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *Internat. J. Robot. Res.*, 30:846–894, 2011.

[KB91] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proc. IEEE Internat. Conf. Robotics Autom.*, pages 1398–1404, 1991.

[Kha86] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots *Internat. J. Robot. Res.*, 5:90–98, 1986.

[KK95] L. Kavraki and M. Kolountzakis. Partitioning a planar assembly into two connected parts is NP-complete. *Inform. Process. Lett.*, 55:159–165, 1995.

[KLa08] L.E. Kavraki and S.M. LaValle. *Motion planning.* In *Springer Handbook of Robotics*, pages 109–131, 2008.

[KSLO96] L.E. Kavraki, P. Švestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for fast path planning in high dimensional configuration spaces. *IEEE Trans. Robot. Autom.*, 12:566–580, 1996.

[KO87] Y. Ke and J. O'Rourke. Moving a ladder in three dimensions: upper and lower bounds. In *Proc. 3rd Sympos. Comput. Geom.*, pages 136–145, ACM Press, 1987.

[KO88] Y. Ke and J. O'Rourke. Lower bounds on moving a ladder in two and three dimensions. *Discrete Comput. Geom.*, 3:197–217, 1988.

[KLPS86] K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–71, 1986.

[KS90] K. Kedem and M. Sharir. An efficient motion planning algorithm for a convex rigid polygonal object in 2-dimensional polygonal space. *Discrete Comput. Geom.*, 5:43–75, 1990.

[KST97] K. Kedem, M. Sharir, and S. Toledo. On critical orientations in the Kedem-Sharir motion planning algorithm. *Discrete Comput. Geom.*, 17:227–239, 1997.

[KC13] H.-S. Kim and O. Cheong. The cost of bounded curvature. *Comput. Geom.*, 46:648–672, 2013.

[KKP11] D. G. Kirkpatrick, I. Kostitsyna, and V. Polishchuk. Hardness results for two-dimensional curvature-constrained motion planning. In *Proc. 23rd Canadian Conf. Comput. Geom.*, pages 27–32, 2011.

[Kol05] V. Koltun. Pianos are not flat: rigid motion planning in three dimensions. In *Proc. 16th ACM-SIAM Sympos. Discrete Algorithms*, pages 505–514, 2005.

[KS03] V. Koltun and M. Sharir. Three-dimensional Euclidean Voronoi diagrams of lines with a fixed number of orientations. *SIAM J. Comput.*, 32:616–642, 2003.

[KS04] V. Koltun and M. Sharir. Polyhedral Voronoi diagrams of polyhedral sites in three dimensions. *Discrete Comput. Geom.*, 31:93–124, 2004.

[KL01] J.J. Kuffner and S.M. LaValle. Rapidly exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics: New Dimensions (WAFR'00)*, pages 293–308, A.K. Peters, Wellesley, 2001.

[Lav06] S.M. LaValle. *Planning Algorithms.* Cambridge University Press, 2006.

[LK99] S.M. LaValle and J.J. Kuffner. Randomized kinodynamic planning. In *Proc. IEEE Internat. Conf. Robotics Autom.*, pages 473–479, 1999.

[LMS97] M. Lanthier, A. Maheshwari, and J.-R. Sack. Approximating weighted shortest paths on polyhedral surfaces. In *Proc. 13th Sympos. Comput. Geom.*, pages 274–283, ACM Press, 1997.

[LGLM00] E. Larsen, S. Gottschalk, M.C. Lin, and D. Manocha. Fast distance queries using rectangular swept sphere volume. In *Proc. IEEE Internat. Conf. Robotics Autom.*, pages 3719–3726, 2000.

[Lat91] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic, Boston, 1991.

[Lau98] J.-P. Laumond, editor. *Robot Motion Planning and Control.* Volume 229 of *Lectures Notes Control Inform. Sci.*, Springer-Verlag, Berlin, 1998.

[LS87b] D. Leven and M. Sharir. An efficient and simple motion planning algorithm for a ladder moving in 2-dimensional space amidst polygonal barriers. *J. Algorithms*, 8:192–215, 1987.

[LS87a] D. Leven and M. Sharir. Planning a purely translational motion for a convex object in two-dimensional space using generalized Voronoi diagrams. *Discrete Comput. Geom.*, 2:9–31, 1987.

[LC92] Z. Li and J.F. Canny, editors. *Nonholonomic Motion Planning.* Kluwer Academic, Norwell, 1992.

[LG98] M.C. Lin and S. Gottschalk. Collision detection between geometric models: A survey. In *Proc. IMA Conf. Math. Surfaces*, pages 37–56, 1998.

[LPMT84] T. Lozano-Pérez, M.T. Mason, and R.H. Taylor. Automatic synthesis of fine-motion strategies for robots. *Internat. J. Robot. Res.*, 3:3–24, 1984.

[LS87] V.J. Lumelsky and A.A. Stepanov. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2:403–430, 1987.

[MMP+91] J. Matoušek, N. Miller, J. Pach, M. Sharir, S. Sifrony, and E. Welzl. Fat triangles determine linearly many holes. In *Proc. 32nd IEEE Sympos. Found. Comput. Sci.*, pages 49–58, 1991.

[MN99] K. Mehlhorn and S. Näher. *The LEDA Platform of Combinatorial and Geometric Computing*, Cambridge University Press, 1999.

[MLL08] J. Minguez, F. Lamiraux, and J.-P. Laumond. *Motion planning and obstacle avoidance.* In *Springer Handbook of Robotics*, pages 827–852, 2008.

[MMP87] J.S.B. Mitchell, D.M. Mount, and C.H. Padadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16:647–668, 1987.

[NSO06] D. Nieuwenhuisen, A.F. van der Stappen, and M.H. Overmars. An effective framework for path planning amidst movable obstacles. In *Algorithmic Foundation of Robotics VII (WAFR'06)*, volume 47 of *Springer Tracts in Advanced Robotics*, pages 87–102, Springer, 2006.

[OSY87] C. Ó'Dúnlaing, M. Sharir, and C.K. Yap. Generalized Voronoi diagrams for a ladder: II. Efficient construction of the diagram. *Algorithmica*, 2:27–59, 1987.

[OY85] C. Ó'Dúnlaing, and C.K. Yap. A "retraction" method for planning the motion of a disc. *J. Algorithms*, 6:104–111, 1985.

[PS85] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, 1985

[Rei87] J.H. Reif. Complexity of the generalized mover's problem. In J.E. Hopcroft, J.T. Schwartz, and M. Sharir, editors, *Planning, Geometry, and Complexity of Robot Motion*, pages 267–281, Ablex, Norwood, 1987.

[RW98] J.H. Reif and H. Wang. The complexity of the two-dimensional curvature-constrained shortest-path problem. In *Proc. 3rd Workshop the Algo. Found. Robotics (WAFR'98)*, pages 49–58, A.K. Peters, Natick, 1998.

[RS94] J.H. Reif and M. Sharir. Motion planning in the presence of moving obstacles. *J. ACM*, 41:764–790, 1994.

[SH15] O. Salzman and D. Halperin. Optimal motion planning for a tethered robot: Efficient preprocessing for fast shortest paths queries. In *Proc. IEEE Internat. Conf. Robotics Autom.*, pages 4161–4166, 2015.

[SH14] O. Salzman and D. Halperin. Asymptotically near-optimal RRT for fast, high-quality, motion planning. In *Proc. IEEE Internat. Conf. Robotics Autom.*, pages 4680–4685, 2014.

[SHH15] O. Salzman, M. Hemmer, and D. Halperin. On the power of manifold samples in exploring configuration spaces and the dimensionality of narrow passages. *IEEE Trans. Autom. Sci. Eng.*, 12:529–538, 2015.

[SHRH13] O. Salzman, M. Hemmer, B. Raveh, and D. Halperin. Motion planning via manifold samples. *Algorithmica*, 67:547–565, 2013.

[SS83] J.T. Schwartz and M. Sharir. On the piano movers' problem: II. General techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math.*, 4:298–351, 1983.

[SS90] J.T. Schwartz and M. Sharir. Algorithmic motion planning in robotics. In J. van Leeuwen, editor, *Handbook of Theoret. Comput. Sci., Volume A: Algorithms and Complexity*, pages 391–430. Elsevier, Amsterdam, 1990.

[SS88] J.T. Schwartz and M. Sharir. A survey of motion planning and related geometric algorithms. *Artif. Intell.*, 37:157–169, 1988. Also in D. Kapur and J. Mundy, editors, *Geometric Reasoning*, pages 157–169. MIT Press, Cambridge, 1989. And in S.S. Iyengar and A. Elfes, editors, *Autonomous Mobile Robots*, volume I, pages 365–374. IEEE Computer Society Press, Los Alamitos, 1991.

[SS08] Y. Schreiber and M. Sharir. An Optimal-Time Algorithm for Shortest Paths on a Convex Polytope in Three Dimensions. *Discrete Comput. Geom.*, 39:500–579, 2008.

[SS97] O. Schwarzkopf and M. Sharir. Vertical decomposition of a single cell in a 3-dimensional arrangement of surfaces. *Discrete Comput. Geom.*, 18:269–288, 1997.

[Sha89] M. Sharir. Algorithmic motion planning in robotics. *Computer*, 22:9–20, 1989.

[Sha95] M. Sharir. Robot motion planning. *Comm. Pure Appl. Math.*, 48:1173–1186, 1995. Also in E. Schonberg, editor, *The Houses That Jack Built.* Courant Institute, New York, 1995, 287–300.

[SA95] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, 1995.

[SS91] M. Sharir and S. Sifrony. Coordinated motion planning for two independent robots. *Ann. Math. Artif. Intell.*, 3:107–130, 1991.

[ST94] M. Sharir and S. Toledo. Extremal polygon containment problems. *Comput. Geom. Theory Appl.*, 4:99–118, 1994.

[SS87] S. Sifrony and M. Sharir. A new efficient motion planning algorithm for a rod in two-dimensional polygonal space. *Algorithmica*, 2:367–402, 1987.

[SC86] R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *Internat. J. Robot. Res.*, 5:56–68, 1986.

[SH15] K. Solovey and D. Halperin. On the hardness of unlabeled multi-robot motion planning. In *Robotics: Science and Systems*, 2015.

[SYZH15] K. Solovey, J. Yu, O. Zamir, and D. Halperin. Motion planning for unlabeled discs with optimality guarantees. In *Robotics: Science and Systems*, 2015.

[SY84] P. G. Spirakis and C.-K. Yap. Strong NP-Hardness of moving many discs. *Inform. Process. Lett.*, 19:55–59, 1984.

[SHO93] A.F. van der Stappen, D. Halperin, and M.H. Overmars. The complexity of the free space for a robot moving amidst fat obstacles. *Comput. Geom. Theory Appl.*, 3:353–373, 1993.

[SOBV98] A.F. van der Stappen, M.H. Overmars, M. de Berg, and J. Vleugels. Motion planning in environments with low obstacle density. *Discrete Comput. Geom.*, 20:561–587, 1998.

[SM88] K. Sutner and W. Maass. Motion planning among time-dependent obstacles. *Acta Inform.*, 26:93–122, 1988.

[TaMK14] M. Tang, D. Manocha, and Y. J. Kim. Hierarchical and controlled advancement for continuous collision detection of rigid and articulated models. *IEEE Trans. Vis. Comput. Graph*, 20:755–766, 2014.

[TuMK14] M. Turpin, N. Michael, and V. Kumar. Capt: Concurrent assignment and planning of trajectories for multiple robots. *Internat. J. Robot. Res.*, 33:98–112, 2014.

[VA01] K.R. Varadarajan and P.K. Agarwal. Approximate shortest paths on a nonconvex polyhedron. *SIAM J. Comput.*, 30:1321–1340, 2001.

[Veg90] G. Vegter. The visibility diagram: A data structure for visibility problems and motion planning. In *Proc. 2nd Scand. Workshop Algorithm Theory*, volume 447 of *Lecture Notes Comput. Sci.*, pages 97–110, Springer-Verlag, Berlin, 1990.

[Wei06] R. Wein. Exact and efficient construction of planar Minkowski sums using the convolution method. In *Proc. European Sympos. Algorithms*, volume 4168 of *Lecture Notes Comp. Sci.*, pages 829–840, Springer, 2006.

[WBH07] R. Wein, J.P. van den Berg, and D. Halperin. The visibility-Voronoi complex and its applications. *Comput. Geom.*, 36:66–87, 2007.

[WBH08] R. Wein, J. P. van den Berg, and D. Halperin. Planning high-quality paths and corridors amidst obstacles. *Internat. J. Robot. Res.*, 27:1213–1231, 2008.

[Wil91] G. Wilfong. Motion planning in the presence of movable obstacles. *Ann. Math. Artif. Intell.*, 3:131–150, 1991.

[WL94] R.H. Wilson and J.-C. Latombe. Geometric reasoning about mechanical assembly. *Artif. Intell.*, 71:371–396, 1994.

[XBV15] N. Xu, P. Brass, and I. Vigan. An improved algorithm in shortest path planning for a tethered robot. *Comput. Geom.*, 48:732–742, 2015.

[Yap87a] C.K. Yap. An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments. *Discrete Comput. Geom.*, 2:365–393, 1987.

[ZF96] Z. Zhang and O. Faugeras. A 3D world model builder with a mobile robot. *Internat. J. Robot. Res.*, 11:269–285, 1996.