30 ARRANGEMENTS

Dan Halperin and Micha Sharir

INTRODUCTION

Given a finite collection $\mathcal{S}$ of geometric objects such as hyperplanes or spheres in $\mathbb{R}^d$, the *arrangement* $\mathcal{A}(\mathcal{S})$ is the decomposition of $\mathbb{R}^d$ into connected open cells of dimensions $0, 1, \ldots, d$ induced by $\mathcal{S}$. Besides being interesting in their own right, arrangements of hyperplanes have served as a unifying structure for many problems in discrete and computational geometry. With the recent advances in the study of arrangements of curved (algebraic) surfaces, arrangements have emerged as the underlying structure of geometric problems in a variety of "physical world" application domains such as robot motion planning and computer vision. This chapter is devoted to arrangements of hyperplanes and of curved surfaces in low-dimensional Euclidean space, with an emphasis on combinatorics and algorithms.

In the first section we introduce basic terminology and combinatorics of arrangements. In Section 30.2 we describe substructures in arrangements and their combinatorial complexity. Section 30.3 deals with data structures for representing arrangements and with special refinements of arrangements. The following two sections focus on algorithms: algorithms for constructing full arrangements are described in Section 30.4, and algorithms for constructing substructures in Section 30.5. In Section 30.6 we discuss the relation between arrangements and other structures. Several applications of arrangements are reviewed in Section 30.7. Section 30.8 deals with robustness issues when implementing algorithms and data structures for arrangements and Section 30.9 surveys software implementations. We conclude in Section 30.10 with a brief review of Davenport-Schinzel sequences, a combinatorial structure that plays an important role in the analysis of arrangements.

30.1 BASICS

In this section we review basic terminology and combinatorics of arrangements, first for arrangements of hyperplanes and then for arrangements of curves and surfaces.

30.1.1 ARRANGEMENTS OF HYPERPLANES

GLOSSARY

Arrangement of hyperplanes: Let $\mathcal{H}$ be a finite set of hyperplanes in $\mathbb{R}^d$. The hyperplanes in $\mathcal{H}$ induce a decomposition of $\mathbb{R}^d$ (into connected open cells), the

arrangement $\mathcal{A}(\mathcal{H})$. A d-dimensional cell in $\mathcal{A}(\mathcal{H})$ is a maximal connected region of $\mathbb{R}^d$ not intersected by any hyperplane in $\mathcal{H}$; any k-dimensional cell in $\mathcal{A}(\mathcal{H})$, for $0 \le k \le d-1$, is a maximal connected region of dimension k in the intersection of a subset of the hyperplanes in $\mathcal{H}$ that is not intersected by any other hyperplane in $\mathcal{H}$. It follows that any cell in an arrangement of hyperplanes is convex.

Simple arrangement: An arrangement $\mathcal{A}(\mathcal{H})$ of a set $\mathcal{H}$ of n hyperplanes in $\mathbb{R}^d$, with $n \ge d$, is called simple if every d hyperplanes in H meet in a single point and if any $d+1$ hyperplanes have no point in common.

Vertex, edge, face, facet: $0, 1, 2$, and $(d-1)$-dimensional cell of the arrangement, respectively. (What we call *cells* here are in some texts referred to as *faces*.)

k-cell : A k-dimensional cell in the arrangement.

Combinatorial complexity of an arrangement: The overall number of cells of all dimensions in the arrangement.

EXAMPLE: AN ARRANGEMENT OF LINES

Let $\mathcal{L}$ be a finite set of lines in the plane, let $\mathcal{A}(\mathcal{L})$ be the arrangement induced by $\mathcal{L}$, and assume $\mathcal{A}(\mathcal{L})$ to be simple. A 0-dimensional cell (a vertex) is the intersection point of two lines in $\mathcal{L}$; a 1-dimensional cell (an edge) is a maximal connected portion of a line in $\mathcal{L}$ that is not intersected by any other line in $\mathcal{L}$; and a 2-dimensional cell (a face) is a maximal connected region of $\mathbb{R}^2$ not intersected by any line in $\mathcal{L}$. See Figure 30.1.1.

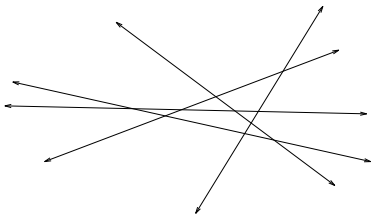


FIGURE 30.1.1
A simple arrangement of 5 *lines. It has* 10 *vertices,* 25 *edges (*10 *of which are unbounded), and* 16 *faces (*10 *of which are unbounded).*

COUNTING CELLS

A fundamental question in the study of arrangements is how complex a certain arrangement (or portion of it) can be. Answering this question is often a prerequisite to the analysis of algorithms on arrangements.

THEOREM 30.1.1

Let $\mathcal{H}$ be a set of hyperplanes in $\mathbb{R}^d$. The maximum number of k-dimensional cells in the arrangement $\mathcal{A}(\mathcal{H})$, for $0 \le k \le d$, is

$$\sum_{i=0}^{k} \binom{d-i}{k-i} \binom{n}{d-i}.$$

The maximum is attained exactly when $\mathcal{A}(\mathcal{H})$ is simple.

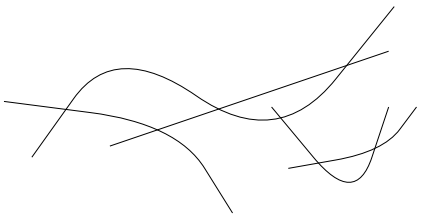


FIGURE 30.1.2
A simple arrangement of 5 x*-monotone bounded arcs, where* $s = 2$.
It has 17 *vertices (*10 *of which are arc endpoints),* 19 *edges, and* 4 *faces (one of which is unbounded).*

We assume henceforth that the dimension d is a (small) constant. With few exceptions, we will not discuss *exact* combinatorial complexity bounds, as in the theorem above, but rather use the big-O notation. Theorem 30.1.1 implies the following:

COROLLARY 30.1.2

The maximum combinatorial complexity of an arrangement of n *hyperplanes in* $\mathbb{R}^d$ *is* $O(n^d)$. *If the arrangement is simple its complexity is* $\Theta(n^d)$. *In these bounds the constant of proportionality depends on* d.

30.1.2 ARRANGEMENTS OF CURVES AND SURFACES

We now introduce more general arrangements, allowing for objects that are nonlinear and/or bounded. We distinguish between planar arrangements and arrangements in three or higher dimensions. For planar arrangements we require only that the objects defining the arrangement be x-monotone Jordan arcs with a constant maximum number of intersections per pair. For arrangements of surfaces in three or higher dimensions we require that the surfaces be algebraic of constant maximum degree, or suitable semi-algebraic portions ("patches") of such surfaces (a more precise definition is given below). This requirement simplifies the analysis and computation of such arrangements, and it does not seem to be too restrictive, as in most applications the arrangements that arise are of low-degree algebraic surfaces or surface patches.

In both cases we typically assume that the objects (curves or surfaces) are in general position. This is a generalization to the current setting of the simplicity assumption for hyperplanes made above. (This assumption is reconsidered in Section 30.8.) All the other definitions in the Glossary carry over to arrangements of curves and surfaces.

PLANAR ARRANGEMENTS

Let $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$ be a collection of (bounded or unbounded) Jordan arcs in the xy-plane, such that each arc is ***x-monotone*** (i.e., every line parallel to the y-axis intersects an arc in at most one point), and each pair of arcs in $\mathcal{C}$ intersect in at most s points for some fixed constant s. The arrangement $\mathcal{A}(\mathcal{C})$ is the decomposition of the plane into open cells of dimensions $0, 1$, and 2 induced by the arcs in $\mathcal{C}$. Here, a 0-dimensional cell (a vertex) is either an endpoint of one arc or an intersection point of two arcs. See Figure 30.1.2.

We assume that the arcs in $\mathcal{C}$ are in ***general position***; here this means that each intersection of a pair of arcs in $\mathcal{C}$ is either a common endpoint or a transversal

intersection at a point in the relative interior of both arcs, and that no three arcs intersect at a common point.

THEOREM 30.1.3

If $\mathcal{C}$ is a collection of n Jordan arcs as defined above, then the maximum combinatorial complexity of the arrangement $\mathcal{A}(\mathcal{C})$ is $O(n^2)$. There are such arrangements whose complexity is $\Theta(n^2)$. In these bounds the constant of proportionality depends linearly on s.

PSEUDO LINES, SEGMENTS, OR CIRCLES

Several special classes of curves have arrangements with favorable properties.

GLOSSARY

A collection of pseudo-lines: A set Γ of unbounded x-monotone connected curves (which can be regarded as graphs of totally-defined continuous functions), each pair of which intersect (transversally) at most once.

A collection of pseudo-segments: Same as above, but the curves of Γ are bounded (graphs of functions defined over bounded intervals).

A collection of pseudo-circles: A set C of simple closed curves, every pair of which intersect at most twice. If the curves are unbounded, we call C ***a collection of pseudo-parabolas***.

As it turns out, arrangements of such families of curves, defined in a purely topological manner, share many properties with arrangements of their standard counterparts—lines, segments, and circles (or parabolas); some of these properties will be noted later in this chapter. For example, a collection of n pseudo-circles has the useful property that the complexity of the union of n regions bounded by pseudo-circles or pseudo-parabolas is at most $6n-12$ [KLPS86]. In certain applications, it is desirable to cut the curves into subarcs, so that each pair of them intersect at most once (that is, cut the curves into pseudo-segments), and then solve a variety of combinatorial and algorithmic problems on the resulting pseudo-segments. An extensive work on this problem, starting with Tamaki and Tokuyama [TT98], has culminated in works by Agarwal et al. [ANP$^+$04] and by Marcus and Tardos [MT06], showing that n pseudo-circles can be cut into $O(n^{3/2}\log n)$ pseudo-segments. See [ANP$^+$04, AS05] for several combinatorial and algorithmic applications of this result.

THREE AND HIGHER DIMENSIONS

We denote the coordinates of $\mathbb{R}^d$ by $x_1, x_2, \ldots, x_d$. For a collection $\mathcal{S} = \{s_1, \ldots, s_n\}$ of (hyper)surface patches in $\mathbb{R}^d$ we make the following assumptions:

1. Each surface patch is contained in an algebraic surface of constant maximum degree.
2. The boundary of each surface patch is determined by at most some constant number of algebraic surface patches of constant maximum degree each. (Formally, each surface patch is a semialgebraic set of $\mathbb{R}^d$ defined by a Boolean combination of a constant number of d-variate polynomial equalities or inequalities of constant maximum degree each.)

3. Every d surface patches in $\mathcal{S}$ meet in at most s points.
4. Each surface patch is ***monotone*** in $x_1, \ldots, x_{d-1}$, namely every line parallel to the x_d-axis intersects the surface patch in at most one point.
5. The surface patches in $\mathcal{S}$ are in *general position*.

We use the simplified term ***arrangement of surfaces*** to refer to arrangements whose defining objects satisfy the assumptions above. A few remarks regarding these assumptions (see [AS00a, Section 2], [Mat02, Section 7.7], [Sha94], for detailed discussions of the required assumptions):

- Assumptions (1) and (2), together with the general position assumption (5), imply that every d-tuple of surfaces meet in at most some constant number of points. One can bound this number using Bézout's Theorem (see Chapter 38). The bound s on the number of d-tuple intersection points turns out to be a crucial parameter in the combinatorial analysis of substructures in arrangements. Often, one can get a better estimate for s than the bound implied by Bézout's theorem.
- Assumption (4) is used in results cited below. It can however be easily relaxed without affecting these results: If a surface patch does not satisfy this assumption, it can be decomposed into pieces that satisfy the assumption, and by assumptions (1) and (2) the number of these pieces will be bounded by a constant and their boundaries will satisfy assumption (2).
- Assumption (5), which is a generalization of the simplicity assumption for hyperplanes (and is discussed in detail in [AS00a, Section 2]), often does not affect the worst-case combinatorial bounds obtained for arrangements or their substructures, because it can be shown that the asymptotically highest complexity is obtained when the surfaces are in general position [Sha94]. For algorithms, this assumption is more problematic. There are general relaxation methods but these seem to introduce new difficulties [Sei98] (see also Section 30.8).

THEOREM 30.1.4

Given a collection $\mathcal{S}$ of n surfaces in $\mathbb{R}^d$, as defined above, the maximum combinatorial complexity of the arrangement $\mathcal{A}(\mathcal{S})$ is $O(n^d)$. There are such arrangements whose complexity is $\Theta(n^d)$. The constant of proportionality in these bounds depends on d and on the maximum algebraic degree of the surfaces and of the polynomials defining their boundaries.

ARRANGEMENTS ON CURVED SURFACES

Although we do not discuss such arrangements directly in this chapter, many of the combinatorial and algorithmic results that we survey carry over to arrangements on curved surfaces (which are assumed to be algebraic of constant degree) with only slight adjustments. Arrangements on spheres are especially prevalent in applications. The ability to analyze or construct arrangements on curved surfaces is implicitly assumed and exploited in the results for arrangements of surfaces in Euclidean space, since we often need to consider the lower-dimensional arrangement induced on a surface by its intersections with all the other surfaces that define the arrangement.

ADDITIONAL TOPICS

We focus in this chapter on simple arrangements. We note, however, that non-simple arrangements raise interesting questions; see, for example, [Szé97]. Another noteworthy topic that we will not cover here is ***combinatorial equivalence*** of arrangements; see Chapter 6 and [BLW+93].

30.2 SUBSTRUCTURES IN ARRANGEMENTS

A substructure in an arrangement (i.e., a portion of an arrangement), rather than the entire arrangement, may be sufficient to solve a problem at hand. Also, the analysis of several algorithms for constructing arrangements relies on combinatorial bounds for substructures. We survey substructures that are known in general to have significantly smaller complexity than that of the entire arrangement. For simplicity, some of the substructures are defined below only for the planar case.

GLOSSARY

Let $\mathcal{C}$ be a collection of n x-monotone Jordan arcs as defined in Section 30.1.

Lower (upper) envelope: For this definition we regard each curve c_i in $\mathcal{C}$ as the graph of a continuous univariate function $c_i(x)$ defined on an interval. The lower envelope Ψ of the collection $\mathcal{C}$ is the pointwise minimum of these functions: $\Psi(x) = \min c_i(x)$, where the minimum is taken over all functions defined at x. (The lower envelope is the 0-level of the arrangement $\mathcal{A}(\mathcal{C})$; see below.) Similarly, the upper envelope of the collection $\mathcal{C}$ is defined as the pointwise maximum of these functions. Lower and upper envelopes are completely symmetric structures, and from this point on we will discuss only lower envelopes.

Minimization diagram of $\mathcal{C}$: The subdivision of the x-axis into maximal intervals so that on each interval the same subset of functions attains the minimum.

In $\mathbb{R}^d$ we regard the surface patches in $\mathcal{S}$ as graphs of functions in the variables $x_1, \ldots, x_{d-1}$, the lower envelope is the pointwise minimum of these functions, and the minimization diagram is the subdivision of $\mathbb{R}^{d-1}$ into maximal connected relatively-open cells such that over each cell the lower envelope is attained by a fixed subset of $\mathcal{S}$.

Zone: For an additional curve γ, the collection of faces of the arrangement $\mathcal{A}(\mathcal{C})$ intersected by γ. See Figure 30.4.1. In earlier works, the zone is sometimes called the ***horizon***.

Single cell: In this section, a d-cell in an arrangement in $\mathbb{R}^d$.

Sandwich region: Given two sets of surfaces, this is the closure of the intersection of the cell below the lower envelope of one set and the cell above the upper envelope of the other set.

Many cells (m cells): Any m distinct d-cells in an arrangement in $\mathbb{R}^d$.

Sides and borders: Let e be an edge in an arrangement of lines, and let l be the line containing e. The line l divides the plane into two halfplanes h_1, h_2. We regard e as two-sided, and denote the two sides by (e, h_1) and (e, h_2). The edge e is on the boundary of two faces f_1 and f_2 in the arrangement. e is said to be a

1-*border* of either face, marked (e, f_1) and (e, f_2), respectively (more precisely, (e, f_1) corresponds to the side (e, h_1), in the sense that f_1 lies in h_1, near e, and (e, f_2) corresponds to (e, h_2)). Similarly a vertex in a simple arrangement of lines has four sides, and it is a **0-*border*** of four faces. The definition extends in an obvious way to arrangements of hyperplanes in higher dimensions and to arrangements of curved surfaces.

k*-level:** We assume here, for simplicity, that the curves are unbounded; the definition can be extended to the case of bounded curves. A point p in the plane is said to be at level k, if there are exactly k curves in $\mathcal{C}$ lying strictly below p (i.e., a relatively open ray emanating from p in the negative y direction intersects exactly k curves in $\mathcal{C}$). The level of an (open) edge e in $\mathcal{A}(\mathcal{C})$ is the level of any point of e; the level is not necessarily fixed on an edge when the arcs are bounded. The k-level of $\mathcal{A}(\mathcal{C})$ is the closure of the union of edges of $\mathcal{A}(\mathcal{C})$ that are at level k; see Figure 30.2.1. The ***at-most-k-level of $\mathcal{A}(\mathcal{C})$, denoted **($\leq k$)-*level***, is the union of points in the plane at level j, for $0 \leq j \leq k$. Different texts use slight variations of the above definitions. In particular, in some texts the ray is directed upwards thus counting the levels from top to bottom. k-levels in arrangements of hyperplanes are closely related (through *duality*, see Section 30.6) to k-sets in point configurations; see Chapter 1.

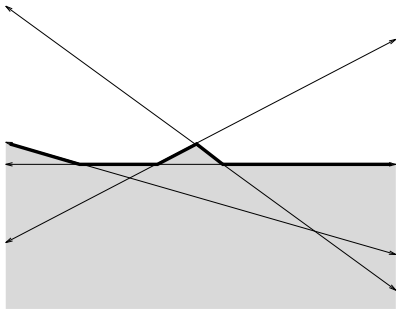


FIGURE 30.2.1
The bold polygonal line is the 2*-level of the arrangement of four lines.*
The shaded region is the (≤ 2)*-level of the arrangement.*

Union boundary: If each surface s in an arrangement in $\mathbb{R}^d$ is the boundary of a d-dimensional object (here we no longer assume monotonicity of the surfaces), then the boundary of the union of the objects is another interesting substructure. The study of the union boundary has largely been motivated by robot motion planning problems; for details see Chapter 51.

$\alpha(n)$: The extremely slowly growing functional inverse of Ackermann's function.

MEASURING THE COMPLEXITY OF A SUBSTRUCTURE

For an arrangement in $\mathbb{R}^d$, if a substructure consists of a collection C of d-cells, its combinatorial complexity is defined to be the overall number of cells of any dimension on the boundary of each of the d-cells in C. This means that we count certain cells of the arrangement with multiplicity (as *borders* of the corresponding d-cells). For example, for the zone of a line l in an arrangement of lines, each edge of the arrangement that intersects l will be counted twice. However, since we assume that our arrangements reside in a space of a fixed (low) dimension, this only implies a constant multiplicative factor in our count.

The complexity of the lower envelope of an arrangement is defined to be the complexity of its minimization diagram. In three or higher dimensions, this means that we count features that do not appear in the original arrangement. For example, in the lower envelope of a collection of triangles in 3-space, the projection of the edges of two distinct triangles may intersect in the minimization diagram although the two triangles are disjoint in 3-space.

The complexity of a k-level in an arrangement is defined in a similar way to the complexity of an envelope. The complexity of the $(\leq k)$-level is defined as the overall number of cells of the arrangement that lie in the region of space whose points are at level at-most-k.

COMBINATORIAL COMPLEXITY BOUNDS FOR SUBSTRUCTURES

In the rest of this section we list bounds on the maximum combinatorial complexity of substructures. For lines, hyperplanes, Jordan arcs, and surfaces, these are arranged in Tables 30.2.1, 30.2.2, 30.2.3, and 30.2.4, respectively. A bound of the form $(n^{k+\epsilon})$ means a bound $A_\epsilon n^{k+\epsilon}$ for every $\epsilon > 0$, where the coefficient A_ϵ depends on ϵ. In the bounds for k-levels and $(\leq k)$-levels we assume that $k \geq 1$ (otherwise one should use $k+1$ instead of k). For each substructure, many special cases of arrangements have been considered and the results are too numerous to cover here. For an extensive review of results for k-levels see [Mat02, Chapter 11], for other substructures see [AS00a], [Mat02, Chapter 7].

TABLE 30.2.1 Substructures in arrangements of n lines or pseudo-lines in the plane.

SUBSTRUCTURE	BOUND	NOTES
Envelope	n edges	
Single face	n edges	
Zone of a line	$\Theta(n)$	See [Ede87] for an exact bound on the number of 0- and 1-borders
m faces	$\Theta(m^{2/3}n^{2/3}+m+n)$	Upper bound [CEG$^+$90]; lower bound [Ede87]
k-level	$O(nk^{1/3})$	[Dey98]
	$n2^{\Omega(\sqrt{\log k})}$	[Tót01], [Niv08]
$(\leq k)$-level	$\Theta(nk)$	[AG86]

All the results in Table 30.2.1 also hold for arrangements of pseudo-lines; in the case of a zone, we assume that the curve defining the zone is another pseudo-line.

CURVES

For a collection $\mathcal{C}$ of n well-behaved curves as defined in Section 30.1, the complexity bounds for certain substructures involve functions related to *Davenport-Schinzel sequences*. The function $\lambda_s(n)$ is defined as the maximum length of a Davenport-Schinzel sequence of order s on n symbols, and it is almost linear in n for any

TABLE 30.2.2 Substructures in arrangements of n hyperplanes in $\mathbb{R}^d$.

SUBSTRUCTURE	BOUND	NOTES
Envelope	$\Theta(n^{\lfloor \frac{d}{2} \rfloor})$	Upper bound theorem [McM70]
Single cell	$\Theta(n^{\lfloor \frac{d}{2} \rfloor})$	Upper bound theorem [McM70]
Zone of a hyperplane	$\Theta(n^{d-1})$	[ESS93]
Zone of p-dimensional algebraic surface of constant degree	$O(n^{\lfloor (d+p)/2 \rfloor} \log^{\gamma} n)$	$\gamma = d + p (\mathrm{mod}\ 2)$ [APS93], the bound is almost tight in the worst case
m cells	$O(m^{\frac{1}{2}} n^{\frac{d}{2}} \log^{(\lfloor \frac{d}{2} \rfloor - 2)/2} n)$	Bound is almost tight [AS04], [AMS94]; see [AA92] for bounds on no. of facets
k-level, $d = 3$	$O(nk^{3/2})$	[SST01]
k-level, $d = 4$	$O(n^{4-1/18})$	[Sha11]
k-level, $d \geq 5$	$O(n^{\lfloor d/2 \rfloor} k^{\lceil d/2 \rceil - \epsilon_d})$	[AACS98], constant $\epsilon_d > 0$
$(\leq k)$-level	$\Theta(n^{\lfloor d/2 \rfloor} k^{\lceil d/2 \rceil})$	[CS89]

fixed s. Davenport-Schinzel sequences play a central role in the analysis of substructures of arrangements of curves and surfaces, and are reviewed in more detail in Section 30.10 below.

THEOREM 30.2.1

For a set $\mathcal{C}$ of n x-monotone Jordan arcs such that each pair intersects in at most s points, the maximum number of intervals in the minimization diagram of $\mathcal{C}$ is $\lambda_{s+2}(n)$. If the curves are unbounded, then the maximum number of intervals is $\lambda_s(n)$.

The connection between a zone and a single cell. As observed in [EGP+92], a bound on the complexity of a single cell in general arrangements of arcs implies the same asymptotic bound on the complexity of the zone of an additional well-behaved curve γ in the arrangement; "well-behaved" meaning that γ does not intersect any curve in $\mathcal{C}$ more than some constant number of times. This observation extends to higher dimensions and is exploited in the result for zones in arrangements of surfaces [HS95a].

The results in Table 30.2.3 are for Jordan arcs (bounded curves). There are slightly better bounds in the case of unbounded curves. For subquadratic bounds on k-levels in special arrangements of curves see [TT98], [Cha03], [ANP+04],[MT06]. Improved bounds on the complexity of m faces in special arrangements of curves are given in [AEGS92] for segments, [AAS03] for pseudo-segments and for circles, and [ANP+04],[MT06] for pseudo circles and some other types of curves.

Inner vs. outer zone. If γ is a Jordan curve, namely a simple closed curve, we distinguish between the portion of the zone in the interior region bounded by γ and the portion in the exterior region, which we call the ***inner zone*** and ***outer zone*** respectively. If γ is the boundary of a convex region then the complexity of the *outer zone* of γ in an arrangement of n lines is $\Theta(n)$ [AD11]. Similarly in higher dimensions, the complexity of the outer zone of the boundary of a convex shape in an arrangement of n hyperplanes in $\mathbb{R}^d$ is $\Theta(n^{d-1})$ [Raz15]. In either case the bound on the complexity of the corresponding inner zone is a tad larger (e.g., $O(n\alpha(n))$ in the planar case), and it is not known whether it is tight. See [Niv15] for recent progress on this problem.

TABLE 30.2.3 Substructures in arrangements of n Jordan arcs.

SUBSTRUCTURE	BOUND	NOTES
Envelope	$\Theta(\lambda_{s+2}(n))$	See Theorem 30.2.1
Single face, zone	$\Theta(\lambda_{s+2}(n))$	[GSS89]
m cells, general	$O(m^{1/2}\lambda_{s+2}(n))$	[EGP+92]
m cells, pseudo-segments	$O(m^{2/3}n^{2/3} + n\log^2 n)$	[AAS03]
m cells, circles	$O(m^{6/11+\epsilon}n^{9/11} + n\log n)$	[AAS03]
m cells	$\Omega(m^{2/3}n^{2/3})$	Lower bound for lines
$(\leq k)$-level	$\Theta(k^2\lambda_{s+2}(\lfloor\frac{n}{k}\rfloor))$	[Sha91]

Notice that the maximum number of edges in envelopes of n Jordan arcs as above is exactly $\lambda_{s+2}(n)$.

TABLE 30.2.4 Substructures in arrangements of n surfaces.

OBJECTS	SUBSTRUCTURE	BOUND	NOTES
Surfaces in $\mathbb{R}^d$	Lower envelope	$O(n^{d-1+\epsilon})$	[HS94],[Sha94]
	Single cell, zone	$O(n^{d-1+\epsilon})$	[Bas03],[HS95a]
	$(\leq k)$-level	$O(n^{d-1+\epsilon}k^{1-\epsilon})$	Combining [CS89] and Lower envelopes bound
$(d-1)$-simplices in $\mathbb{R}^d$	Lower envelope	$\Theta(n^{d-1}\alpha(n))$	[PS89], [Ede89]
	Single cell, zone	$O(n^{d-1}\log n)$	[AS94]
$(d-1)$-spheres in $\mathbb{R}^d$	Lower envelope, single cell	$\Theta(n^{\lceil\frac{d}{2}\rceil})$	Linearization

UNION BOUNDARY

For a collection of n pseudo-disks (regions bounded by pseudo-circles), there are at most $6n-12$ intersection points (for $n \geq 3$) between curves on the union boundary [KLPS86]. This bound is tight in the worst case. For variants and extensions of this result see [EGH+89], [PS99], [AEHS01].

Many of the interesting results in this area are for Minkowski sums where one of the operands is convex, motivated primarily by motion planning problems. In the plane this reduces to the union of pseudo-disks; see [KLPS86]. These results are reviewed in Chapter 51. We mention one exemplary result in three dimensions that (almost) settles a long-standing open problem: the complexity of the union boundary of n congruent infinite cylinders (namely, each cylinder is the Minkowski sum of a line in 3-space and a unit ball) is $O(n^{2+\epsilon})$ [AS00b]. 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].

Another family of results is for so-called *fat* objects. For example, a triangle is considered fat if all its angles are at least some fixed constant $\delta > 0$. For such triangles it is shown [MPS+94] that they determine at most a linear number of *holes* (namely connected components of the complement of the union) and that their union boundary has near-linear complexity (see below). Several works establish near-linear bounds for other classes of objects in $\mathbb{R}^2$; they are summarized in

[APS08]. Perhaps the most comprehensive result is due to de Berg et al. [ABES14] who show that the complexity of the union of n *locally* γ*-fat* objects of constant descriptive complexity is $\frac{n}{\gamma^4}2^{O(\log^* n)}$, where an object K is locally γ-fat if, for any disk D whose center lies in K and that does not fully contain K, we have $\text{area}(D \sqcap K) \geq \gamma \cdot \text{area}(D)$, where $D \sqcap K$ is the connected component of $D \cap K$ that contains the center of D. This is the most general class of fat objects. For γ-fat traingles (a special case), the union complexity improves to $O(n \log^8 n + \frac{n}{\gamma} \log^2 \frac{1}{\gamma})$. Typically (but not always) fatness precludes constructions with high union complexity, such as grid-like patterns with complexity $\Omega(n^d)$ in $\mathbb{R}^d$. See [ABES14] for references to many previous results, with slightly inferior bounds, for the union complexity of other classes of fat objects.

In three and higher dimensions, the following results are known: (i) The complexity of the union of n balls in $\mathbb{R}^d$ is $O(n^{\lceil d/2 \rceil})$ (easily established by lifting the balls into halfspaces in $\mathbb{R}^{d+1}$). (ii) The complexity of the union of n axis-parallel cubes in $\mathbb{R}^d$ is also $O(n^{\lceil d/2 \rceil})$, and it drops (in odd dimensions) to $O(n^{\lfloor d/2 \rfloor})$ [BSTY98]. (iii) In three dimensions, the complexity of the union of k convex polytopes with a total of n facets is $O(k^3 + nk \log k)$, and can be $\Omega(k^3 + nk\alpha(k))$ in the worst case [AST97]. (iv) The preceding bound improves to $O(nk \log k)$ (and the lower bound is $\Omega(nk\alpha(k))$) where the polytopes are Minkowski sums of a fixed convex polytope with a collection of k pairwise disjoint convex polytopes [AS97]. If the fixed polytope in the sum is a box, the bound further improves to $O((n^2\alpha(n))$ [HY98]. (v) The complexity of the union of n arbitrary fat tetrahedra in $\mathbb{R}^3$ is $O(n^{2+\epsilon})$, for any $\epsilon > 0$ [ES09]. In particular, the complexity of the union of n arbitrarily aligned cubes in 3-space is $O(n^{2+\epsilon})$, for any $\epsilon > 0$ (see also [PSS03] for a different proof for the case of nearly congruent cubes). (vi) The complexity of the union of n κ-round (not necessarily convex) objects in $\mathbb{R}^3$ (resp., in $\mathbb{R}^4$) of constant descriptive complexity is $O(n^{2+\epsilon})$ (resp., $O(n^{3+\epsilon})$), for any $\epsilon > 0$; an object c is called κ-round if for every $p \in \partial c$ there exists a ball that contains p, is contained in c, and has radius $\kappa \cdot \text{diam}(c)$ [AEKS06]. (vii) The maximum number of holes in the union of n translates of a convex set in $\mathbb{R}^3$ is $\Theta(n^3)$ [ACDG15].

ADDITIONAL COMBINATORIAL BOUNDS

The following bounds, while not bounds on the complexity of substructures, are useful in the analysis of algorithms for computing substructures and in obtaining other combinatorial bounds on arrangements.

Sum of squares of cell complexities. Let $\mathcal{H}$ be a collection of n hyperplanes in $\mathbb{R}^d$. For each d-cell c of the arrangement $\mathcal{A}(\mathcal{H})$, let $f(c)$ denote the number of cells of any dimension on the boundary of c. Aronov et al. [AMS94] show that $\sum_c f^2(c) = O(n^d \log^{\lfloor \frac{d}{2} \rfloor - 1} n)$, where the sum extends over all d-cells of the arrangement; see also [AS04] for a simpler proof. They use it to obtain bounds on the complexity of m cells in the arrangement. An application of the zone theorem [ESS93] implies a related bound: If we denote the number of hyperplanes appearing on the boundary of the cell c by $f_{d-1}(c)$ (this is equal to the number of facets of ∂c), then $\sum_c f(c) f_{d-1}(c) = O(n^d)$, where the sum extends over all d-cells of the arrangement.

Overlay of envelopes. For two sets A and B of objects in $\mathbb{R}^d$, the complexity of the *overlay of envelopes* is defined as the complexity of the subdivision of $\mathbb{R}^{d-1}$ induced by superposing the minimization diagram of A on that of B. Given two sets

$\mathcal{C}_1$ and $\mathcal{C}_2$, each of n x-monotone Jordan arcs, such that no pair of (the collection of $2n$) arcs intersects more than s times, the complexity of the overlay is easily seen to be $\Theta(\lambda_{s+2}(n))$. In 3-space, given two sets each of n well-behaved surfaces, the complexity of the overlay is $O(n^{2+\epsilon})$ [ASS96a] (a simpler proof of the bound appears in [KS03]). The bound is applied to obtain a simple divide-and-conquer algorithm for computing the envelope in 3-space, and for obtaining bounds on the complexity of *transversals* (see Chapter 4). The bound in $\mathbb{R}^4$ is $O(n^{3+\epsilon})$ [KS03], but analogously sharp bounds are not known in higher dimensions; see [KS09] for some progress in this direction. An interesting variant of this theme is presented by Kaplan et al. [KRS11], who show that the expected complexity of the overlay of all the minimization diagrams obtained during a randomized incremental construction of the lower envelope of n planes in $\mathbb{R}^3$ is $O(n \log n)$.

Sandwich region. As an immediate application of the bound on the overlay of envelopes, one can derive the same asymptotic bound for the complexity of the sandwich region for two families of n surfaces in total. That is, the complexity of the sandwich region is $O(n^{2+\epsilon})$ in $\mathbb{R}^3$ [ASS96b] and $O(n^{3+\epsilon})$ in $\mathbb{R}^4$ [KS03]

OPEN PROBLEMS

1. What is the complexity of the k-level in an arrangement of lines in the plane? For the gap between the known lower and upper bounds see Table 30.2.1. This is a long-standing open problem in combinatorial geometry. The analogous questions for arrangements of planes or hyperplanes in higher dimensions are equally challenging with wider gaps between the known bounds.
2. What is the complexity of m faces in an arrangement of well-behaved Jordan arcs? For lines and pseudo-lines a tight bound is known, as well as an almost tight bound for segments and a sharp bound for circles, whereas for more general curves a considerable gap still exists—see Table 30.2.3.

30.3 REPRESENTATIONS AND DECOMPOSITIONS

Before describing algorithms for arrangements in the next sections, we discuss how to represent an arrangement. The appropriate data structure for representing an arrangement depends on its intended use. Two typical ways of using arrangements are: (i) traversing the entire arrangement cell by cell; and (ii) directly accessing certain cells of the arrangement. We will present three structures, each providing a method for traversing the entire arrangement: the *incidence graph*, the *cell-tuple structure*, and the *complete skeleton*. We will then discuss refined representations that further subdivide an arrangement into subcells. These refinements are essential to allow for efficient access to cells of the arrangement. For algebraic geometry-oriented representations and decompositions see Chapters 7, 38 and 51.

GLOSSARY

Let $\mathcal{S}$ be a collection of surfaces in $\mathbb{R}^d$ (or curves in $\mathbb{R}^2$) as defined in Section 30.1,

and $\mathcal{A}(\mathcal{S})$ the arrangement induced by $\mathcal{S}$. Let c_1 be a k_1-dimensional cell of $\mathcal{A}(\mathcal{S})$ and c_2 a k_2-dimensional cell of $\mathcal{A}(\mathcal{S})$.

Subcell, supercell: If $k_2 = k_1 + 1$ and c_1 is on the boundary of c_2, then c_1 is a subcell of c_2, and c_2 is a supercell of c_1.

(−1)-dimensional cell, (d+1)-dimensional cell: Some representations assume the existence of two additional cells in an arrangement. The unique (-1)-dimensional cell is a subcell of every vertex (0-dimensional cell) in the arrangement, and the unique $(d+1)$-dimensional cell is a supercell of all the d-dimensional cells in the arrangement.

Incidence: If c_1 is a subcell of c_2, then c_1 and c_2 are *incident* to one another. We say that c_1 and c_2 define an ***incidence***.

30.3.1 REPRESENTATIONS

INCIDENCE GRAPH

The incidence graph (sometimes called the ***facial lattice***) of the arrangement $\mathcal{A}(\mathcal{S})$ is a graph $G = (V, E)$ where there is a node in V for every k-cell of $\mathcal{A}(\mathcal{S})$, $-1 \leq k \leq d + 1$, and an edge between two nodes if the corresponding cells are incident to one another (cf. Figure 16.1.3). For an arrangement of n surfaces in $\mathbb{R}^d$ the number of nodes in V is $O(n^d)$ by Theorem 30.1.4. This is also a bound on the number of edges in E: every cell (besides the (-1)-dimensional cell) in an arrangement $\mathcal{A}(\mathcal{S})$ in general position has at most a constant number of supercells. For an exact bound in the case of hyperplanes, see [Ede87, Section 1.2].

CELL-TUPLE STRUCTURE

While the incidence graph captures all the cells in an arrangement and (as its name implies) their incidence relation, it misses *order* information between cells. For example, there is a natural order among the edges that appear along the boundary of a face in a planar arrangement. This leads to the ***cell-tuple structure*** [Bri93] which is a generalization to any dimension of the two-dimensional doubly-connected-edge-list (DCEL) [BCKO08] or the similar ***quad-edge structure*** [GS85] and the 3D ***facet-edge structure*** [DL89]. The cell-tuple structure gives a simple and uniform representation of the adjacency and ordering information in the arrangement.

SKELETON

Let $\mathcal{H}$ be a finite set of hyperplanes in $\mathbb{R}^d$. A ***skeleton*** in the arrangement $\mathcal{A}(\mathcal{H})$ is a connected subset of edges and vertices of the arrangement. The ***complete skeleton*** is the union of all the edges and vertices of the arrangement. Edelsbrunner [Ede87] proposes a representation of the skeleton as a digraph, which allows for a systematic traversal of the entire arrangement (in the case of a complete skeleton) or a substructure of the arrangement. Using a one-dimensional skeleton to represent an arrangement in an arbitrary-dimensional space is a notion that appears also

in algebro-geometric representations. There, however, the skeleton, or ***roadmap***, is far more complicated (indeed it represents more general arrangements); see [BPR06] and Chapter 51.

30.3.2 DECOMPOSITIONS

A raw arrangement may still be an unwieldy structure as cells may have complicated shapes and many bounding subcells. It is often desirable to decompose the cells of the arrangement into subcomponents so that each subcomponent has a constant descriptive complexity and is homeomorphic to a ball. Besides the obvious convenience that such a decomposition offers (just like a triangulation of a simple polygon), it turns out to be crucial to the design and analysis of randomized algorithms for arrangements, as well as to combinatorial analysis of arrangements.

For a decomposition to be useful, we aim to add as few extra features as possible. The three decompositions described in this section have the property that the complexity of the decomposed arrangement is asymptotically close to (sometimes the same as) that of the original arrangement. (This is still not known for the *vertical decomposition* in higher dimensions—see the open problem below.)

BOTTOM VERTEX DECOMPOSITION OF HYPERPLANE ARRANGEMENTS

Consider an arrangement of lines $\mathcal{A}(\mathcal{L})$ in the plane. For a face f let $v_b = v_b(f)$ be the bottommost vertex of f (the vertex with lowest y coordinate, ties can be broken by the lexicographic ordering of the coordinate vectors of the vertices). Extend an edge from v_b to each vertex on the boundary of f that is not incident to an edge incident to v_b; see Figure 30.3.1. Repeat for all faces of $\mathcal{A}(\mathcal{L})$ (unbounded faces require special care). The original arrangement, together with the added edges, constitutes the ***bottom vertex decomposition*** of $\mathcal{A}(\mathcal{L})$, which is a decomposition of $\mathcal{A}(\mathcal{L})$ into triangles. The notion extends to arrangements of hyperplanes in higher dimensions, and it is carried out recursively [Cla88]. The combinatorial complexity of the decomposition is asymptotically the same as that of the original arrangement.

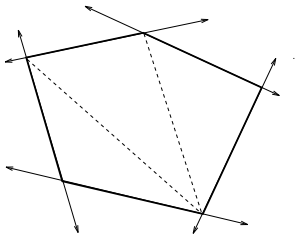


FIGURE 30.3.1
The bottom vertex decomposition of a face in an arrangement of lines.

VERTICAL DECOMPOSITION

The bottom vertex decomposition does not in general extend to arrangements of nonlinear objects, or even of line segments. Fortunately there is an alternative, rather simple, decomposition method that applies to almost any reasonable arrangement. This is the ***vertical decomposition*** or ***trapezoidal decomposition***.

See Figure 30.3.2. It is optimal for two-dimensional arrangements, namely its complexity is asymptotically the same as that of the underlying arrangement. It is near-optimal in three and four dimensions. In higher dimensions it is still the general decomposition method that is known to have the best (lowest) complexity.

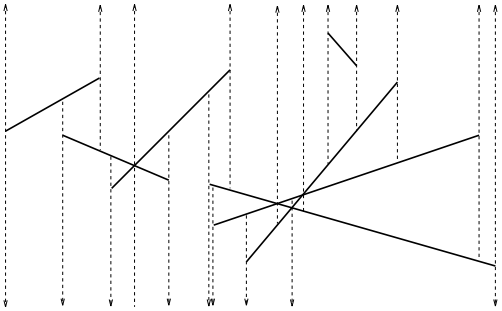


FIGURE 30.3.2
The vertical decomposition of an arrangement of segments: a vertical line segment is extended upwards and downwards from each vertex of the arrangement until it either hits another segment or extends to infinity. These segments decompose the arrangement into trapezoids, triangles, and degenerate variants thereof.

The extension to higher dimensions is defined recursively and is presented in full generality in [CEGS91]. For details of the extension to three dimensions, see [CEG+90] for the case of spheres, and [BGH96] for the case of triangles. The four-dimensional case is studied in [Kol04a], [Kol04b]. Table 30.3.1 summarizes the bounds on the maximum combinatorial complexity of the vertical decomposition for several types of arrangements and substructures. Certain assumptions that the input curves and surfaces are "well-behaved" are not detailed.

TABLE 30.3.1 Combinatorial bounds on the maximum complexity of the vertical decomposition of n objects.

OBJECTS	BOUND	NOTES
Curves in $\mathbb{R}^2$	$\Theta(K)$	K is the complexity of $\mathcal{A}$
Surfaces in $\mathbb{R}^3$	$O(n^2\lambda_t(n))$	[CEGS91], t depends on the algebraic complexity
Surfaces in $\mathbb{R}^d, d \geq 4$	$O(n^{2d-4+\epsilon})$	[CEGS91], [Kol04a]
Triangles in $\mathbb{R}^3$	$\Theta(n^3)$	[BGH96]
Triangles in $\mathbb{R}^3$	$O(n^2\alpha(n)\log n + K)$	K is the complexity of $\mathcal{A}$ [Tag96]
Surfaces in $\mathbb{R}^3$, single cell	$O(n^{2+\epsilon})$	[SS97]
Surfaces in $\mathbb{R}^3$, $(\leq k)$-level	$O(n^{2+\epsilon}k)$	See [AES99] for refined bounds
Hyperplanes in $\mathbb{R}^4$	$\Theta(n^4)$	[Kol04b]
Simplices in $\mathbb{R}^4$	$O(n^4\alpha(n)\log n)$	[Kol04b]

CUTTINGS

All the decompositions described so far have the property that each cell of the decomposition lies fully in a single cell of the arrangement. In various applications this property is not required and other decomposition schemes may be applied, such as *cuttings* (Chapter 45). Cuttings are the basis of efficient divide-and-conquer algorithms for numerous geometric problems on arrangements and otherwise.

POLYNOMIAL PARTITIONING

A novel approach to decomposing arrangements is due to Guth and Katz [GK15]. See also Chapter 7. Using algebraic techniques, combined with the polynomial Ham-Sandwich theorem of Stone and Tukey, they have obtained the following result.

THEOREM 30.3.1

Let P be a set of n points in $\mathbb{R}^d$, and let $r < n$ be a given parameter. There exists a real d-variate polynomial f, of degree $O(r^{1/d})$, such that each of the $O(r)$ connected components of $\mathbb{R}^d \setminus Z(f)$ (where $Z(f)$ denotes the zero set of f) contains at most n/r points of P.

Note several features of this *polynomial partitioning* technique. First, it offers no guarantee about the size of $P \cap Z(f)$. In principle, all the points of P could lie on $Z(f)$. Second, this is a technique for partitioning a set of points and not an arrangement of surfaces. Nevertheless, using standard techniques from real algebraic geometry (for which see, e.g., [BPR06]), any algebraic surface of constant degree and of dimension k intersects only $O(r^{k/d})$ cells of the partition (i.e., components of $\mathbb{R}^d \setminus Z(f)$). Hence, given a collection of n k-dimensional algebraic surfaces of constant descriptive complexity, each cell of the partition is crossed, on average, by $O(n/r^{(d-k)/d})$ surfaces. A more recent construction of Guth [Gut14] provides an alternative similar construction, where each cell of the partition is guaranteed to be crossed by at most $O(n/r^{(d-k)/d})$ of the surfaces.

This new approach strengthens considerably the earlier decomposition techniques mentioned above: (i) It applies to surfaces of any dimension (e.g., it provides a decomposition scheme for a set of lines in 3-space), which the older techniques could not do (for an exception, see Koltun and Sharir [KS05]). (ii) It provides sharp bounds for the size of the subproblems within the partition cells, which the older schemes (based on vertical decomposition) failed so far to do in dimension greater than 4.

A weak aspect of the new technique is that it does not provide a general scheme for handling the points of P that lie on the zero set $Z(f)$; as mentioned, there might be many such points. Many of the recent applications of polynomial partitioning had to provide ad-hoc solutions for this part of the problem, and a significant portion of the current research aims to provide general-purpose techniques for further partitioning $P \cap Z(f)$.

Yet another handicap is that the technique does not offer an efficient procedure for constructing the partition, mainly because there are no known efficient algorithms for constructing polynomial Ham-Sandwich cuts in higher dimensions. See Agarwal et al. [AMS13] for an efficient scheme for constructing approximate polynomial partitionings, with algorithmic applications.

In spite of these weaknesses, polynomial partitioning had a tremendous impact on combinatorial geometry, and has led to many new results on incidences between points and curves or surfaces in higher dimensions, distinct distances between points in a given set, repeated distances and other repeated patterns, efficient range searching with semi-algebraic sets, and more. (The most dramatic achievement of polynomial partitioning, in Guth and Katz's original paper [GK15], was to obtain the almost tight lower bound $\Omega(n/\log n)$ on the number of distinct distances determined by any set of n points in the plane, a classical problem posed by Erdős in 1946; see also Chapter 1.

OPEN PROBLEMS

We summarize parts of the preceding discussion into two major open problems:

1. How fast can a partitioning polynomial be constructed? In contrast with cuttings and other earlier decomposition techniques, where optimal or near-optimal algorithms are known for many cases, here it is not known whether a partitioning polynomial can be constructed in polynomial time. This is because a key step in the construction is the polynomial Ham Sandwich cut of Stone and Tukey, whose original proof of existence is nonconstructive, and the only know constructive proof, for discrete sets of points, takes exponential time. A partial solution of this problem is given in Agarwal et al. [AMS13].
2. Another challenge concerning partitioning polynomials has to do with the issue that, while the zero set $Z(f)$ of the partitioning polynomial f distributes the given points evenly among the cells of $\mathbb{R}^d \setminus Z(f)$, it may leave an uncontrolled number of points on $Z(f)$ itself. In many problems, handling these points becomes a nontrivial issue. Several methods have been proposed, such as the construction of a second polynomial g, which partitions evenly the points on $Z(f)$ among the cells of $Z(f) \setminus Z(g)$ (see [KMSS12, Zah11]), and even a third partitioning polynomial (see [BS16]), but the general problem, especially the variants that involve higher-dimensional surfaces that interact with the given points in higher dimensions, is still open.

30.4 ALGORITHMS FOR ARRANGEMENTS

This section covers the algorithmic problem of constructing an arrangement: producing a representation of an arrangement in one of the forms described in the previous section (or in a similar form). We distinguish between algorithms for the construction of the entire arrangement (surveyed in this section), and algorithms for constructing substructures of an arrangement (in the next section). We start with deterministic algorithms and then describe randomized ones.

MODEL OF COMPUTATION

We assume the standard model in computational geometry: infinite precision real arithmetic [PS85]. For algorithms computing arrangements of curves or surfaces, we further assume that certain operations on a small number of curves or surfaces take unit time each. For algebraic curves or surfaces, the unit cost assumption for these operations is theoretically justified by results on the solution of sets of polynomial equations, or more generally, on construction and manipulation of semi-algebraic sets; see Chapter 38. When implementing algorithms for arrangements some of these assumptions need to be reconsidered from the practical point of view; see Sections 30.8 and 30.9.

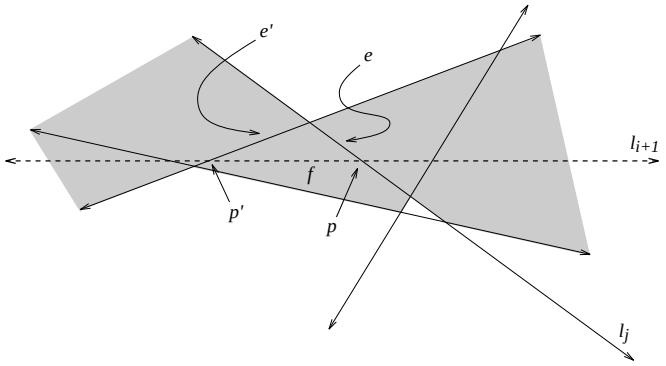
30.4.1 DETERMINISTIC ALGORITHMS

Incremental construction. The incremental algorithm proceeds by adding one object after the other to the arrangement while maintaining (a representation of)

the arrangement of the objects added so far. This approach yields an optimal-time algorithm for arrangements of hyperplanes. The analysis of the running time is based on the zone result [ESS93] (Section 30.2).

We present the algorithm for a collection $\mathcal{L} = \{l_1, \ldots, l_n\}$ of n lines in the plane, assuming that the arrangement $\mathcal{A}(\mathcal{L})$ is simple. Let $\mathcal{L}_i$ denote the set $\{l_1, \ldots, l_i\}$. At stage $i+1$ we add l_{i+1} to the arrangement $\mathcal{A}(\mathcal{L}_i)$. We maintain the DCEL representation [BCKO08] for $\mathcal{A}(\mathcal{L}_i)$, so that in addition to the incidence information, we also have the order of edges along the boundary of each face. The addition of l_{i+1} is carried out in two steps: (i) we find a point p of intersection between l_{i+1} and an edge of $\mathcal{A}(\mathcal{L}_i)$ and split that edge into two, and (ii) we walk along l_{i+1} from p to the left (assuming l_{i+1} is not vertical) updating $\mathcal{A}(\mathcal{L}_i)$ as we go; we then walk along l_{i+1} from p to the right completing the construction of $\mathcal{A}(\mathcal{L}_{i+1})$. See Figure 30.4.1.

FIGURE 30.4.1
Adding the line l_{i+1} to the arrangement $\mathcal{A}(\mathcal{L}_i)$.
The shaded region is the zone of l_{i+1} in the arrangement of the other four lines.



Finding an edge of $\mathcal{A}(\mathcal{L}_i)$ that l_{i+1} intersects can be done in $O(i)$ time by choosing one line l_j from $\mathcal{L}_i$ and checking all the edges of $\mathcal{A}(\mathcal{L}_i)$ that lie on l_j for intersection with l_{i+1}. This intersection point p lies on an edge e that borders two faces of $\mathcal{A}(\mathcal{L}_i)$. We split e into two edges at p. Next, consider the face f intersected by the part of l_{i+1} to the left of p. Using the order information, we walk along the edges of f away from p and we check for another intersection p' of l_{i+1} with an edge e' on the boundary of f. At the intersection we split e' into two edges, we add an edge to the arrangement for the portion $\overline{pp'}$ of l_{i+1}, and we move to the face on the other (left) side of e'. Once we are done with the faces of $\mathcal{A}(\mathcal{L}_i)$ crossed by l_{i+1} to the left of p, we go back to p and walk to the other side. This way we visit all the faces of the zone of l_{i+1} in $\mathcal{A}(\mathcal{L}_i)$, as well as some of its edges. Updating the DCEL structure due to the splitting or addition of edges is straightforward. The amount of time spent is proportional to the number of edges we visit, and hence bounded by the complexity of the zone of l_{i+1} in $\mathcal{A}(\mathcal{L}_i)$, which is $O(i)$. The total time, over all insertions steps, is thus $O(n^2)$. The space required for the algorithm is the space to maintain the DCEL structure. The same approach extends to higher dimensions; for details see [Ede87, Chapter 7].

THEOREM 30.4.1

If $\mathcal{H}$ is a set of n hyperplanes in $\mathbb{R}^d$ such that $\mathcal{A}(\mathcal{H})$ is a simple arrangement, then $\mathcal{A}(\mathcal{H})$ can be constructed in $\Theta(n^d)$ time and space.

The time and space required by the algorithm are clearly optimal. However, it turns out that for arrangements of lines one can do better in terms of *working space*. This is explained below in the subsection *topological sweep*. See [Goo93], [HJW90] for parallel algorithms for arrangements of hyperplanes.

The incremental approach can be applied to constructing planar arrangements of curves, using the vertical decomposition of the arrangement [EGP+92]:

THEOREM 30.4.2

Let $\mathcal{C}$ be a set of n Jordan arcs as defined in Section 30.1*. The arrangement $\mathcal{A}(\mathcal{C})$ can be constructed in $O(n\lambda_{s+2}(n))$ time using $O(n^2)$ space.*

Sweeping over the arrangement. The sweep paradigm, a fundamental paradigm in computational geometry, is also applicable to constructing arrangements. For planar arrangements, its worst-case running time is slightly inferior to that of the incremental construction described above. It is, however, output sensitive.

THEOREM 30.4.3

Let $\mathcal{C}$ be a set of n Jordan arcs as defined in Section 30.1*. The arrangement $\mathcal{A}(\mathcal{C})$ can be constructed in $O((n+k)\log n)$ time and $O(n+k)$ space, where k is the number of intersection points in the arrangement.*

One can similarly sweep a plane over an arrangement of surfaces in $\mathbb{R}^3$. There is an output-sensitive algorithm for constructing the vertical decomposition of an arrangement of n surfaces that runs in time $O(n\log^2 n + V\log n)$, where V is the combinatorial complexity of the vertical decomposition. For details see [SH02].

Topological sweep. Edelsbrunner and Guibas [EG89] devised an algorithm for constructing an arrangement of lines that requires only linear working storage and runs in optimal $O(n^2)$ time. Instead of sweeping the arrangement with a straight line, they sweep it with a pseudoline that serves as a "topological wavefront."

The most efficient deterministic algorithm for computing the intersections in a collection of well-behaved curves is due to Balaban [Bal95]. It runs in $O(n\log n+k)$ time and requires $O(n)$ working storage.

30.4.2 RANDOMIZED ALGORITHMS

Most randomized algorithms for arrangements follow one of two paradigms: (i) incremental construction or (ii) divide-and-conquer using random sampling. The randomization in these algorithms is in choices made by the algorithm; for example, the order in which the objects are handled in an incremental construction. In the expected performance bounds, the expectation is with respect to the random choices made by the algorithm. We do not make any assumptions about the distribution of the objects in space. See also Chapter 45.

In constructing a full arrangement, these two paradigms are rather straightforward to apply. Most of these algorithms use an efficient decomposition as discussed in Section 30.3.

Incremental construction. Here the randomization is in the order that the objects defining the arrangement are inserted. For the construction of an arrangement of curves, the algorithm is similar to the deterministic construction mentioned above.

THEOREM 30.4.4 [Mul93]

Let $\mathcal{C}$ be a set of n Jordan arcs as defined in Section 30.1*. The arrangement $\mathcal{A}(\mathcal{C})$ can be constructed by a randomized incremental algorithm in $O(n \log n + k)$ expected time and $O(n + k)$ expected space, where k is the number of intersection points in the arrangement.*

Divide-and-conquer by random sampling. For a set $\mathcal{V}$ of n objects in $\mathbb{R}^d$ the paradigm is: choose a subset $\mathcal{R}$ of the objects at random, construct the arrangement $\mathcal{A}(\mathcal{R})$, decompose it further into constant complexity components (using, for example, one of the methods described in Section 30.3), and recursively construct the portion of the arrangement in each of the resulting components. Then glue all the substructures together into the full arrangement. The theory of random sampling is then used to show that with high probability the size of each subproblem is considerably smaller than that of the original problem, and thus efficient resource bounds can be proved. See Chapters 48 and 49 for further application.

The divide-and-conquer counterpart of Theorem 30.4.4 is due to Amato et al. [AGR00]. It has the same running time, and uses slightly more space (or exactly the same space for the case of segments).

The result stated in the following theorem is obtained by applying this paradigm to arrangements of algebraic surfaces and it is based on the vertical decomposition of the arrangement.

THEOREM 30.4.5 [CEGS91], [Kol04a]

Given a collection $\mathcal{S}$ of n algebraic surfaces in $\mathbb{R}^d$ as defined in Section 30.1, a data structure of size $O(n^{2d-4+\epsilon})$ for the arrangement $\mathcal{A}(\mathcal{S})$ can be constructed in $O(n^{2d-4+\epsilon})$ time, for any $\epsilon > 0$, so that a point-location query can be answered in $O(\log n)$ time. In these bounds the constant of proportionality depends on ϵ, the dimension d, and the maximum algebraic degree of the surfaces and their boundaries.

If only traversal of the entire arrangement is needed, it is plausible that a simpler structure such as the incidence graph could be constructed using less time and storage space, close to $O(n^d)$ for both. See [Can93], [BPR06] for algebro-geometric methods.

Derandomization. Techniques have been proposed to derandomize many randomized geometric algorithms, often without increase in their asymptotic running time; see Chapter 45. However, in most cases the randomized versions are conceptually much simpler and hence may be better candidates for efficient implementation.

30.4.3 OTHER ALGORITHMIC ISSUES

For algebro-geometric tools, see Chapter 33. See Chapter 41 (and Section 30.8) for a discussion of precision and degeneracies. Parallel algorithms are discussed in Chapter 47.

30.5 CONSTRUCTING SUBSTRUCTURES

ENVELOPE AND SINGLE CELL IN ARRANGEMENTS OF HYPERPLANES

Computing a single cell or an envelope in an arrangement of hyperplanes is equivalent (through duality) to computing the convex hull of a set of points in $\mathbb{R}^d$ (Chapter 28). For the case of a single cell, one also needs to find a point inside the cell, to facilitate the duality, which can be done by linear programming (Chapter 50).

Using linearization [AM94], we can solve these problems for arrangements of spheres in $\mathbb{R}^d$. We first transform the spheres into hyperplanes in $\mathbb{R}^{d+1}$, and then solve the corresponding problems in $\mathbb{R}^{d+1}$. (transporting the solution back to $\mathbb{R}^d$ requires some care, as the single cell in $\mathbb{R}^{d+1}$ might be split into several subcells in $\mathbb{R}^d$.

LOWER ENVELOPE

The lower envelope of a collection of n well-behaved curves (where each pair intersect in at most s points) can be computed, in a suitable model of computation, by a simple divide-and-conquer algorithm that runs in time $O(\lambda_{s+2}(n)\log n)$ and requires $O(\lambda_{s+2}(n))$ storage. Hershberger [Her89] devised an improved algorithm that runs in time $O(\lambda_{s+1}(n)\log n)$; in particular, for the case of line segments, it runs in optimal $O(n\log n)$ time. In 3-space, Agarwal et al. [ASS96a] showed that a simple divide-and-conquer scheme can be used to compute the envelope of n surfaces in time $O(n^{2+\epsilon})$. This is an application of the bound on the complexity of the overlay of envelopes cited in Section 30.2. Boissonnat and Dobrindt give a randomized incremental algorithm for computing the envelope [BD96]. There are efficient algorithms for computing the envelope of $(d-1)$-simplices in $\mathbb{R}^d$ (see [EGS89] for the algorithm in 3D which can be efficiently extended to higher dimensions), and an efficient data structure for point location in the minimization diagram of surfaces in $\mathbb{R}^4$ [AAS97]. Output-sensitive construction of the envelope of triangles in $\mathbb{R}^3$ has mainly been studied in relation to hidden-surface removal (see [Ber93]). Partial information of the minimization diagram (vertices, edges and 2-cells) can be computed efficiently for arrangements of surfaces in any fixed dimension [AAS97]. See also [KRS11, AES99].

SINGLE CELL AND ZONE

All the results cited below for a single cell in arrangements of bounded objects hold for the zone problem as well (see the remark in Section 30.2 on the connection between the problems).

Computing a single face in an arrangement of n Jordan arcs as defined in Section 30.1 can be accomplished in worst-case near-optimal time: deterministically in $O(\lambda_{s+2}(n)\log^2 n)$ time, and using randomization in $O(\lambda_{s+2}(n)\log n)$ time [SA95].

In three dimensions, Schwarzkopf and Sharir [SS97] give an algorithm with running time $O(n^{2+\epsilon})$ for any $\epsilon > 0$ to compute a single cell in an arrangement of n well-behaved surfaces. Algorithms with improved running time to compute

a single cell in 3D arrangements are known for arrangements of surfaces induced by certain motion planning problems [Hal92], [Hal94], and for arrangements of triangles [BDS95].

It is still not known how to compute a single cell in arrangements of surfaces in dimension $d > 4$ in time $O(n^{d-1+\epsilon})$, similar to the bound on the complexity of a cell. However, less efficient algorithms from real algebraic geometry are known (see Basu et al. [BPR06]).

LEVELS

In an arrangement of n lines in the plane, the k-level can be computed in $O((n + f)\log n)$ time, where f is the combinatorial complexity of the k-level— the bound is for the algorithm described in [EW86] while using the data structure in [BJ02] which in turn builds on ideas in [Cha01]. For computing the k-level in an arrangement of hyperplanes in $\mathbb{R}^d$ see [AM95], [Cha96].

The $(\leq k)$-level in arrangements of lines can be computed in worst-case optimal time $O(n \log n + kn)$ [ERK96]. Algorithms for computing the $(\leq k)$-level in arrangements of Jordan arcs are described in [ABMS98], the $(\leq k)$-level in arrangements of planes in $\mathbb{R}^3$ (in optimal $O(n \log n + k^2 n)$ expected time) in [Cha00], and in arrangements of surfaces in $\mathbb{R}^3$ in $O(n^{2+\epsilon})$ time [AES99].

The $(\leq k)$-level of an arrangement of n surfaces in $\mathbb{R}^d$ is closely related to the notion of *k-shallow $(1/r)$-cuttings*, where we wish to partition the portion of space lying at or below the k-level into a small number of cells of constant descriptive complexity, each crossed by at most n/r of the surfaces. The fact that the complexity of the $(\leq k)$-level is smaller than that of the whole arrangement leads to improved bounds on the size of shallow cuttings, and, subsequently, to a variety of applications, most notably halfspace range reporting in arrangements of hyperplanes [Mat92, Ram99, AC09, CT15, HKS16] and of more general surfaces [AES99].

UNION BOUNDARY

For a given family of planar regions bounded by well-behaved curves, let $f(m)$ be the maximum complexity of the union boundary of a collection of m objects of the family (the interesting case is when $f(m)$ is linear or close to linear in $\mathbb{R}^2$). Then the union of n such objects can be constructed deterministically in $O(f(n)\log^2 n)$ time or by a randomized incremental algorithm in expected $O(f(n)\log n)$ time [BDS95]. A slightly faster algorithm for the case of fat triangles is given in [MMP+91]. A practically efficient algorithm is described in [EHS04]. An efficient randomized algorithm for computing the union of convex polytopes in $\mathbb{R}^3$ is given in [AST97]. The case of the boundary of the union of Minkowski sums (all having one summand in common) is covered in detailed in Chapter 51.

MANY CELLS

There are efficient algorithms (deterministic and randomized) for computing a set of selected faces in arrangements of lines or segments in the plane. These algorithms are nearly worst-case optimal [AMS98]. Algorithms for arrangements of planes are described in [EGS90], and for arrangements of triangles in 3-space in [AS90].

The related issue of computing the incidences between a set of objects (lines, unit circles) and a set of points is dealt with in [Mat93], with results that extend to higher dimensions [AS00a]. Generally, the bounds for the running time are roughly the same as those for the number of incidences. For lower bounds for the related *Hopcroft's problem* see [Eri96], [BK03].

OPEN PROBLEMS

Devise efficient algorithms for computing:

1. The lower envelope of an arrangement of surfaces in five and higher dimensions; for an algorithm that computes partial information see [AAS97].
2. A single cell in an arrangement of surfaces in four and higher dimensions; for a worst-case near-optimal algorithm in three dimensions see [SS97].

30.6 RELATION TO OTHER STRUCTURES

Arrangements relate to a variety of additional structures. Since the machinery for analyzing and computing arrangements is rather well developed, problems on related structures are often solved by first constructing (or reasoning about) the corresponding arrangement.

Using *duality* one can transform a set (or *configuration*) of points in $\mathbb{R}^d$ (the primal space) into a set of hyperplanes in $\mathbb{R}^d$ (the dual space) and vice versa. Different duality transforms are advantageous in different situations [O'R98].

Edelsbrunner [Ede87, Chapter 12] describes a collection of problems stated for point configurations and solved by operating on their corresponding dual arrangements. An example is given in the next section. See also Chapter 1. Another example is computing incidences between m points and n lines in the plane, or constructing a set of m marked faces in an arrangement of n lines, or computing the number of intersections between n line segments. In these problems one first constructs a decomposition of the plane of the sorts mentioned in Section 30.3.2, obtains subproblems within the cells of the decomposition, and solves each subproblem by passing to the dual plane. See, e.g., [Aga90].

Since many properties of line arrangements extend to pseudo-line arrangements (i.e., unbounded x-monotone curves, each pair of which intersect at most (or exactly) once), it is desirable to apply duality in pseudo-line arrangements too (e.g., for solving variants of the aforementioned problems). Such an effective (albeit fairly involved) scheme is presented in Agarwal and Sharir [AS05].

Plücker coordinates are a tool that enables one to treat k-flats in $\mathbb{R}^d$ as points or hyperplanes in a possibly different higher-dimensional space. This has been taken advantage of in the study of families of lines in 3-space; see Chapter 42.

Lower envelopes (or more generally k-levels in arrangements) relate to Voronoi diagrams; see Chapter 29.

For the connection of arrangements to polytopes and zonotopes see [Ede87] and Section 18.5 of this Handbook. For the connection to oriented matroids see Chapter 6.

30.7 APPLICATIONS

A typical application of arrangements is for solving a problem on related structures. We first transform the original structure (e.g., a point configuration) into an arrangement and then solve the problem on the resulting arrangement. See Section 30.6 above and Chapters 1, 29, 42, and 51.

EXAMPLE: MINIMUM AREA TRIANGLE

Let P be a set of n points in the plane. We wish to find three points of P such that the triangle that they define has minimum area. We use the duality transform that maps a point $p := (a, b)$ to the line $p^* := (y = ax - b)$, and maps a line $l := (y = cx + d)$ to the point $l^* := (c, -d)$. One can show that if we fix two points $p_i, p_j \in P$, and the line p_k^* has the smallest vertical distance to the intersection point $p_i^* \cap p_j^*$ among all other lines in $P^* = \{p^* | p \in P\}$, then the point p_k defines the minimum area triangle with the fixed points p_i, p_j over all points in $P \setminus \{p_i, p_j\}$. Finding the triple of lines as above (an intersecting pair and the other line closest to the intersection) is easy after constructing the arrangement $\mathcal{A}(P^*)$ (Section 30.4), and can be done in $\Theta(n^2)$ time in total. As a special case, we can determine whether P contains three collinear points (a zero-area triangle) in $\Theta(n^2)$ time. This is the most efficient algorithm known for the minimum-area problem [GO95], which for now survives the recent successful attacks on the related 3-SUM problem and its relatives [JP14]. The minimum volume simplex defined by $d + 1$ points in a set of n points in $\mathbb{R}^d$ can be found using arrangements of hyperplanes in $\Theta(n^d)$ time.

OTHER APPLICATIONS

Another strand of applications consists of "robotic" or "physical world" applications [HS95b]. In these problems a continuous space is decomposed into a finite number of cells so that in each cell a certain invariant is maintained. Here, arrangements are used to discretize a continuous space without giving up the completeness or exactness of the solution. An example of an application of this kind solves the following problem: Given a convex polyhedron in 3-space, determine how many combinatorially distinct orthographic and perspective views it induces; see Table 25.6.3 The answer is given using an arrangement of circles on the sphere (for orthographic views) and an arrangement of planes in 3-space (for perspective views) [BD90].

Many developments in the study of arrangements of curves and surfaces have been primarily motivated by problems in robot motion planning (Chapter 51) and several of its variants (Chapter 52). For example, the most efficient algorithm known for computing a collision-free path for an arbitrary polygonal robot (not necessarily convex) moving by translation and rotation among polygonal obstacles in the plane is based on computing a single connected component in an arrangement of surfaces in 3-space. The problem of planning a collision-free motion for a robot among obstacles is typically studied in the *configuration space* where every point represents a possible configuration of the robot. The related arrangements are of surfaces that represent all the contact configurations between the boundary of the robot and the boundaries of obstacles and thus partition configuration space into

free cells (describing configurations where the robot does not intersect any obstacle) and forbidden cells. Given the initial (free placement) of the robot, we need only explore the cell that contains this initial configuration in the arrangement.

A concept similar to configuration space of motion planning has been applied in assembly planning (Section 52.3). The assembly planning problem is converted into a problem in *motion space* where every point represents an allowed path (motion) of a subcollection of the assembly relative the rest of the assembly [HLW00]. The motion space is partitioned by a collection of constraint surfaces such that for all possible motions inside a cell of the arrangement, the collection of movable subsets of the assembly is invariant.

As mentioned earlier, arrangements on spheres are prevalent in applications. Aside from vision applications, they also occur in: computer-assisted radio-surgery [SAL93], molecular modeling [HS98], assembly planning (Section 48.3), manufacturing [ABB+02], and more.

Arrangements have been used to solve problems in many other areas including geometric optimization [AS98], range searching (Chapter 41), statistical analysis (Chapter 60), and micro robotics [BDH99], to name a few. More applications can be found in the sources cited below and in several other chapters in this book.

30.8 ROBUSTNESS

Transforming the data structures and algorithms described above into effective computer programs is a difficult task. The typical assumptions of (i) the real RAM model of computation and (ii) general position, are not realistic in practice. This is not only a problem for implementing software for arrangements but rather a general problem in computational geometry (see Chapter 70). However, it is especially acute in the case of arrangements since here one needs to compute *intersection points* of curves and surfaces and use the computed values in further operations (to distinguish from say convex hull algorithms that only select a subset of the input points).

EXACT COMPUTING

A general paradigm to overcome robustness problems is to compute exactly. For arrangements of linear objects, namely, arrangements of hyperplanes or of simplices, there is a fairly straightforward solution: using arbitrary precision rational arithmetic. This is regularly done by keeping arbitrary long integers for the enumerator and denominator of each number. Of course the basic numerical operations now become costly, and methods were devised to reduce the cost of rational arithmetic predicates through the use of *floating point filters* (Chapter 46) which turn out to be very effective in practice, especially when the input is nondegenerate.

Matters are more complicated when the objects are not linear. First, there is the issue of representation. Consider the following simplest planar arrangement of the line $y = x$ and the circle $x^2 + y^2 = 1$ (both described by equations with integer coefficients). The upper vertex (intersection point) v_1 has coordinates $(\sqrt{2}/2, \sqrt{2}/2)$. This means that we cannot have a simple numerical representation of the vertices of the arrangement. An elegant solution to this problem is provided by special number types, so-called *algebraic number types*. The approach is transparent to

the user who just has to substitute the standard machine type (e.g., double) for the corresponding novel number type (which is a C++ class). Two software libraries support such number types (called *real* in both): LEDA [MN00] (Chapter 70) and Core [KLPY99] (Chapter 46). The ideas behind the solution proposed by both are similar and rely on separation bounds. In terms of arrangements the power that these number types provide is that we can determine the exact topology of the arrangement in all cases including degenerate cases.

While exact computing may seem to be the solution to all problems, the situation is far from being satisfactory for several reasons: (i) The existing number types considerably slow down the computation compared with standard machine arithmetic. (ii) It is difficult to implement the full fledged number types required for arrangements of curves and surfaces. Significant progress has been made for planar algebraic curves of arbitrary degree [EK08]; a detailed review of support for special types of curves appears in [FHW12, Section 5.7]. (iii) It still leaves open the question of handling degeneracies (see PERTURBATION below).

The high cost of exact predicates has led researchers to look for alternative algorithmic solutions (for problems where good solutions, in the standard measures of computational geometry, have been known), solutions that use less costly predicates; see, e.g., [BP00].

ROUNDING

In rounding we transform an arbitrary precision arrangement into fixed precision representation. The most intensively studied case is that of planar arrangements of segments. A solution proposed independently by Hobby [Hob99] and by Greene (improving on an earlier method in [GY86]), snaps vertices of the arrangement to centers of pixels in a prespecified grid. The method preserves several topological properties of the original arrangement and indeed expresses the vertices of the arrangement with limited precision numbers (say bounded bit-length integers). A dynamic algorithm is described in [GM98], and an improved algorithm for the case where there are many intersections within a pixel is given in [GGHT97]. Snap rounding has several drawbacks though: a line is substituted by a polyline possibly with many links (a "shortest-path" rounding scheme is proposed in [Mil00] that sometimes introduces fewer links than snap rounding), and a vertex of the arrangement can become very close to a nonincident edge. The latter problem has been overcome in an alternative scheme *iterated snap rounding* which guarantees a large separation between such features of the arrangement but pays in the quality of approximation [HP02, Pac08]. Several more efficient algorithms and variants have also been proposed (see, e.g. [BHO07, Her13]). Notice that in the snap-rounded arrangement the rounded versions of a pair of input segments may intersect an arbitrarily large number of times. Finally, the 3D version seems to produce a huge number of extra features [For99]: a polyhedral subdivision of complexity n turns into a snapped subdivision of complexity $O(n^4)$; in addition the rounding precision depends on the combinatorial complexity of the input.

Effective and consistent rounding of arrangements remains an important and largely open problem. The importance of rounding arrangements stems not only from its being a means to overcome robustness issues, but, not less significantly, from being a way to express the arrangement numerically with reasonable bit-size numbers.

APPROXIMATE ARITHMETIC IN PREDICATE EVALUATION

The behavior of fundamental algorithms for computing line arrangements (both sweep line and incremental) while using limited precision arithmetic is studied in [FM91]. It is shown that the two algorithms can be implemented such that for n lines the maximum error of the coordinates of vertices is $O(n\epsilon)$ where ϵ is the relative error of the approximate arithmetic used (e.g., floating point). An approximate algorithm for constructing curve arrangements is presented in [MS07].

PERTURBATION

An arrangement of lines is considered degenerate if it is not simple (Section 30.1). A degeneracy occurs for example when three lines meet at a common point. Intuitively this is a degeneracy since moving the lines slightly will result in a topologically different arrangement. Degeneracies in arrangements pose difficulties for two reasons. First and foremost they incredibly complicate programming. Although it has been proposed that handling degeneracies could be the solution in practice to relax the general position assumption [BMS94], in three and higher dimensions handling all degeneracies in arrangements seems an extremely difficult task. The second difficulty posed by degeneracies is that the numerical computation at or near degeneracies typically requires higher precision and will for example cause floating point filters to fail and resort to exact computing resulting in longer running time.

To overcome the first difficulty, symbolic perturbation schemes have been proposed. They enable a consistent perturbation of the input objects so that all degeneracies are removed. These schemes modify the objects only symbolically and a limiting process is used to define the perturbed objects (corresponding to infinitesimal perturbations) such that all predicates will have nonzero results. They require the usage of exact arithmetic, and a postprocessing stage to determine the structure of the output. The case of arrangements of hyperplanes can be approached by *simulation of simplicity* [EM90] via point-hyperplane duality. For a unifying view of these schemes and a discussion of their properties, see [Sei98].

An alternative approach is to *actually* perturb the objects from their original placement. One would like to perturb the input objects as little as possible so that precision problems are resolved. This approach is viable in situations where the exact placement of the input can be compromised, as is the case in many engineering and scientific applications where the input is inexact due to measurement or modeling errors. An efficient such scheme for arrangements of spheres that model molecules is described in [HS98]; it has been adapted and extended to handle arrangements of line segments [Pac11], circles [HL04], polyhedral surfaces [Raa99], as well as Delaunay triangulations [FKMS05]. It is referred to as *controlled* perturbation since it guarantees that the final arrangement is degeneracy free (and predicates can be safely computed with limited precision arithmetic), to distinguish from heuristic perturbation methods. A general analysis methodology for controlled-perturbation algorithms is presented in [MOS11]. A variant called controlled *linear* perturbation has been devised and used to robustly compute three-dimensional Minkowski sums [SMK11].

OPEN PROBLEM

Devise efficient and consistent rounding schemes for arrangements of curves in the plane and for arrangements in three and higher dimensions.

30.9 SOFTWARE

In spite of the numerous applications of arrangements, robust software for computing and manipulating arrangements has barely been available until about a decade ago. The situation has changed significantly over the past decade, with the increased understanding of the underlying difficulties, the research on overcoming these difficulties that has intensified during the last several years (Chapter 46), and the appearance of infrastructure for developing such software in the form of computational geometry libraries that emphasize robustness (Chapter 70).

30.9.1 2D ARRANGEMENTS

LEDA enables the construction of arrangements of segments via a sweep line algorithm. The resulting subdivision is represented as a LEDA graph. Point location based on persistent search trees is supported. The construction is robust through the use of arbitrary precision rationals.

Arrangements of general types of curves are supported by CGAL as we describe next, not limited to the planar case but rather supporting arrangement on surfaces.

2D ARRANGEMENTS IN CGAL

The most generic arrangement package at the time of the writing is the CGAL arrangements package [FHW12]. The genericity is obtained through the separation of the combinatorial part of the algorithms and the numerical part [FHH+00], [WFZH07]. (The overall design follows [Ket99].) The combinatorial algorithms are coded assuming that a small set of numerical/geometric operations (predicates and constructions) is supplied by the user for the desired type of curves. These operations are packed in a traits class (Chapter 70) that is passed as a parameter to the algorithms. The algorithms include the dynamic construction of the arrangement, represented as a doubly-connected-edge-list (DCEL), allowing for insertion and deletion of curves. Alternatively one can construct the arrangement using a sweep line algorithm. Several algorithms for point location are supported [HH08], most notably a complete implementation of random incremental construction of a trapeziodal-map based structure for arbitrary curves [FHH+00, HKH12]. All algorithms handle arbitrary input, namely they do not assume general position. Several traits classes are supplied with the package for: line segments, circular arcs, canonical parabolas, polylines, and planar algebraic curves of arbitrary degree [EK08]; for a list of supported types of curves, see [FHW12, Section 5.7].

Several tools were built on top of the CGAL arrangements package for computing: Envelopes of surfaces [Mey06], which in turn have paved the way to computing general Voronoi diagrams [SSH10], Boolean operations, and Minkowski sums.

The CGAL arrangement package has been used to compute Voronoi diagrams of

lines in space [HSH10], to implement motion planning algorithms, [HH02],[SHRH13], several versions of snap rounding [HP02], art gallery optimization [KBFS12], NMR analysis tools [MYB+11], and many more applications.

A major recent development is the extension of CGAL's arrangement package from planar arrangements to arrangements on parametric surfaces [BFH+10b, BFH+10a]. The extended framework can handle planes, cylinders, spheres, tori, and surfaces homeomorphic to them. This extension has already been applied to computing Voronoi diagrams on the sphere, Minkowski sums of convex polytopes [BFH+10a] and to plan disassembly with infinite translations [FH13], among others.

30.9.2 3D ARRANGEMENTS

Software to construct arrangements of triangles in 3-space exactly, assuming general position, is described in [SH02]. The implementation uses a space sweep algorithm and exact rational arithmetic. The arrangement is represented by its vertical decomposition or a sparser variant called the *partial vertical decomposition*. Arrangements of algebraic surfaces in 3-space pose a much bigger challenge. Steps in this direction, including the handling of degeneracies, based on an efficient variant of Collins decomposition are described in [BS08], [BKS10].

OPEN PROBLEMS

1. Devise a systematic method to directly handle degeneracies in arrangements in three and higher dimensions (that is, to compute and represent degeneracies without removing them).
2. Extend the full-fledged support for 2D arrangements of curves to 3D arrangements of surfaces.

30.10 DAVENPORT-SCHINZEL SEQUENCES

Davenport-Schinzel sequences are interesting and powerful combinatorial structures that arise in the analysis and calculation of the lower or upper envelope of collections of functions, and therefore have applications in many geometric problems, including numerous motion planning problems, which can be reduced to the calculation of such an envelope. A comprehensive survey of Davenport-Schinzel sequences and their geometric applications can be found in [SA95].

An (n, s) ***Davenport-Schinzel sequence***, where n and s are positive integers, is a sequence $U = (u_1, \ldots, u_m)$ composed of n symbols with the properties:

(i) No two adjacent elements of U are equal: $u_i \neq u_{i+1}$ for $i = 1, \ldots, m-1$.

(ii) U does not contain as a subsequence any alternation of length $s+2$ between two distinct symbols: there do not exist $s+2$ indices $i_1 < i_2 < \cdots < i_{s+2}$ so that $u_{i_1} = u_{i_3} = u_{i_5} = \cdots = a$ and $u_{i_2} = u_{i_4} = u_{i_6} = \cdots = b$, for two distinct symbols a and b.

Thus, for example, an $(n, 3)$ sequence is not allowed to contain any subsequence of the form $(a \cdots b \cdots a \cdots b \cdots a)$. Let $\lambda_s(n)$ denote the maximum possible length of an (n, s) Davenport-Schinzel sequence.

The importance of Davenport-Schinzel sequences lies in their relationship to the combinatorial structure of the lower (or upper) envelope of a collection of functions (Section 30.2). Specifically, for any collection of n real-valued continuous functions $f_1, \ldots, f_n$ defined on the real line, having the property that each pair of them intersect in at most s points, one can show that the sequence of function indices i in the order in which these functions attain their lower envelope (i.e., their pointwise minimum $f = \min_i f_i$) from left to right is an (n, s) Davenport-Schinzel sequence. Conversely, any (n, s) Davenport-Schinzel sequence can be realized in this way for an appropriate collection of n continuous univariate functions, each pair of which intersect in at most s points.

The crucial and surprising property of Davenport-Schinzel sequences is that, for any fixed s, the maximal length $\lambda_s(n)$ is nearly linear in n, although for $s \geq 3$ it is slightly super-linear.

The best bounds on $\lambda_s(n)$, for every s, are due to Pettie [Pet15], and they are all asymptotically tight, or nearly tight. They are

$$\lambda_s(n) = \begin{cases} n & s = 1 \\ 2n - 1 & s = 2 \\ 2n\alpha(n) + O(n) & s = 3 \\ \Theta\left(n2^{\alpha(n)}\right) & s = 4 \\ \Theta\left(n\alpha(n)2^{\alpha(n)}\right) & s = 5 \\ n \cdot 2^{(1+o(1))\alpha^t(n)/t!} & \text{for both even and odd } s \geq 6;\ t = \lfloor \frac{s-2}{2} \rfloor \end{cases}$$

where $\alpha(n)$ is the inverse of Ackermann's function. Ackermann's function $A(n)$ grows extremely quickly, with $A(4)$ equal to an exponential "tower" of 65636 2's. Thus $\alpha(n) \leq 4$ for all practical values of n. See [SA95].

If one considers the lower envelope of n continuous, but only partially defined, functions, then the complexity of the envelope is at most $\lambda_{s+2}(n)$, where s is the maximum number of intersections between any pair of functions [SA95]. Thus for a collection of n line segments (for which $s = 1$), the lower envelope consists of at most $O(n\alpha(n))$ subsegments. A surprising result is that this bound is tight in the worst case: there are collections of n segments, for arbitrarily large n, whose lower envelope does consist of $\Omega(n\alpha(n))$ subsegments. This is perhaps the most natural example of a combinatorial structure defined in terms of n simple objects, whose complexity involves the inverse Ackermann's function; see [SA95, WS88].

30.11 SOURCES AND RELATED MATERIAL

FURTHER READING

The study of arrangements through the early 1970s is covered by Grünbaum in [Grü67, Chapter 18], [Grü71], and [Grü72]. See also the monograph by Zaslavsky [Zas75].

In this chapter we have concentrated on more recent results. Details of many of these results can be found in the following books. The book by Edelsbrunner [Ede87] takes the view of "arrangements of hyperplanes" as a unifying theme for a large part of discrete and computational geometry until 1987. Sharir and Agarwal's book [SA95] is an extensive report on results for arrangements of curves and surfaces. See also the more recent survey [AS00a] and book [PS09]. Chapters dedicated to arrangements of hyperplanes in books: Mulmuley emphasizes randomized algorithms [Mul93], O'Rourke discusses basic combinatorics, relations to other structures and applications [O'R98], de Berg et al. discuss planar arrangements of lines with application to discrepancy [BCKO08], and Pach and Agarwal [PA95] discuss problems involving arrangements in discrete geometry. Boissonnat and Yvinec [BY98] discuss, in addition to arrangements of hyperplanes, arrangements of segments and of triangles. Arrangements of hyperplanes and of surfaces are also the topics of chapters in Matoušek's book [Mat02].

RELATED CHAPTERS

Chapter 1: Finite point configurations
Chapter 5: Pseudoline arrangements
Chapter 6: Oriented matroids
Chapter 7: Polynomial partitions
Chapter 16: Basic properties of convex polytopes
Chapter 28: Convex hull computations
Chapter 29: Voronoi diagrams and Delaunay triangulations
Chapter 33: Computational real algebraic geometry
Chapter 39: Point location
Chapter 41: Range searching
Chapter 42: Ray shooting and lines in space
Chapter 45: Randomization and derandomization
Chapter 46: Robust geometric computation
Chapter 47: Parallel algorithms in geometry
Chapter 51: Algorithmic motion planning
Chapter 52: Robotics
Chapter 70: Two computational geometry libraries: LEDA and CGAL

REFERENCES

[AA92] P.K. Agarwal and B. Aronov. Counting facets and incidences. *Discrete Comput. Geom.*, 7:359–369, 1992.

[AACS98] P.K. Agarwal, B. Aronov, T.M. Chan, and M. Sharir. On levels in arrangements of lines, segments, planes, and triangles. *Discrete Comput. Geom.*, 19:315–331, 1998.

[AAS97] P.K. Agarwal, B. Aronov, and M. Sharir. Computing envelopes in four dimensions with applications. *SIAM J. Comput.*, 26:1714–1732, 1997.

[AAS03] P. Agarwal, B. Aronov, and M. Sharir. On the complexity of many faces in arrangements of pseudo-segments and of circles. In B. Aronov et al., editors, *Discrete and Computational Geometry—The Goodman-Pollack Festschrift*, pages 1–23. Springer, Heidelberg, 2003.

[AC09] P. Afshani and T.M. Chan. Optimal halfspace range reporting in three dimensions. In *Proc. 20th ACM-SIAM Sympos. Discrete Algorithms*, pages 180–186, S2009.

[ACDG15] B. Aronov, O. Cheong, M.G. Dobbins, and X. Goaoc. The number of holes in the union of translates of a convex set in three dimensions. In *Proc. 32nd Sympos. Comput Geom.*, LIPIcs, Schloss Dagstuhl, 2016.

[AD11] B. Aronov and D. Drusvyatskiy. Complexity of a single face in an arrangement of s-intersecting curves. Manuscript, `arXiv:1108.4336`, 2011.

[ABB+02] H.-K. Ahn, M. de Berg, P. Bose, S.-W. Cheng, D. Halperin, J. Matoušek, and O. Schwarzkopf. Separating an object from its cast. *Computer-Aided Design*, 34:547–559, 2002.

[ABES14] B. Aronov, M. de Berg, E. Ezra, and M. Sharir. Improved bounds for the union of locally fat objects in the plane. *SIAM J. Comput.*, 43:543–572, 2014.

[ABMS98] P.K. Agarwal, M. de Berg, J. Matoušek, and O. Schwarzkopf. Constructing levels in arrangements and higher order Voronoi diagrams. *SIAM J. Comput.*, 27:654–667, 1998.

[AEGS92] B. Aronov, H. Edelsbrunner, L.J. Guibas, and M. Sharir. The number of edges of many faces in a line segment arrangement. *Combinatorica*, 12:261–274, 1992.

[AEHS01] B. Aronov, A. Efrat, D. Halperin, and M. Sharir. On the number of regular vertices of the union of Jordan regions. *Discrete Comput. Geom.*, 25:203–220, 2001.

[AEKS06] B. Aronov, A. Efrat, V. Koltun, and M. Sharir. On the union of κ-round objects in three and four dimensions. *Discrete Comput. Geom.*, 36:511–526, 2006.

[AES99] P.K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM J. Comput.*, 29:912–953, 1999.

[AG86] N. Alon and E. Győri. The number of small semispaces of a finite set of points in the plane. *J. Combin. Theory Ser. A*, 41:154–157, 1986.

[Aga90] P.K. Agarwal. Parititoning arrangements of lines II: applications. *Discrete Comput. Geom.*, 5:533–573, 1990.

[AGR00] N.M. Amato, M.T. Goodrich, and E.A. Ramos. Computing the arrangement of curve segments: divide-and-conquer algorithms via sampling. In *Proc. 11th ACM-SIAM Sympos. Discrete Algorithms*, pages 705–706, 2000.

[AM94] P.K. Agarwal and J. Matoušek. On range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11:393–418, 1994.

[AM95] P.K. Agarwal and J. Matoušek. Dynamic half-space range reporting and its applications. *Algorithmica*, 13:325–345, 1995.

[AMS94] B. Aronov, J. Matoušek, and M. Sharir. On the sum of squares of cell complexities in hyperplane arrangements. *J. Combin. Theory Ser. A*, 65:311–321, 1994.

[AMS98] P.K. Agarwal, J. Matoušek, and O. Schwarzkopf. Computing many faces in arrangements of lines and segments. *SIAM J. Comput.*, 27:491–505, 1998.

[AMS13] P.K. Agarwal, J. Matoušek, and M. Sharir. On range searching with semialgebraic sets, II. *SIAM J. Comput.*, 42:2039–2062, 2013.

[ANP+04] P.K. Agarwal, E. Nevo, J. Pach, R. Pinchasi, M. Sharir, and S. Smorodinsky. Lenses in arrangements of pseudo-circles and their applications. *J. ACM*, 51:139–186, 2004.

[APS93] B. Aronov, M. Pellegrini, and M. Sharir. On the zone of a surface in a hyperplane arrangement. *Discrete Comput. Geom.*, 9:177–186, 1993.

[APS08] P.K. Agarwal, J. Pach, and M. Sharir. State of the union (of geometric objects). In J.E. Goodman, J. Pach, and R. Pollack, editors, *Surveys on Discrete and Computational Geometry: Twenty Years Laterms*, vol. 453 of *Contemporary Mathematics*, pages 299–305. AMS, Rpovidence, 2008.

[AS90] B. Aronov and M. Sharir. Triangles in space or building (and analyzing) castles in the air. *Combinatorica*, 10:137–173, 1990.

[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.

[AS98] P.K. Agarwal and M. Sharir. Efficient algorithms for geometric optimization. *ACM Comput. Surv.*, 30:412–458, 1998.

[AS00a] P.K. Agarwal and M. Sharir. Arrangements and their applications. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 49–119. Elsevier, Amsterdam, 2000.

[AS00b] 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.

[AS04] B. Aronov and M. Sharir. Cell complexities in hyperplane arrangements. *Discrete Comput. Geom.*, 32:107–115, 2004.

[AS05] P.K. Agarwal and M. Sharir. Pseudo-line arrangements: Duality, algorithms, and applications. *SIAM J. Comput.*, 34:526–552, 2005.

[ASS96a] P.K. Agarwal, O. Schwarzkopf, and M. Sharir. The overlay of lower envelopes and its applications. *Discrete Comput. Geom.*, 15:1–13, 1996.

[ASS96b] P.K. Agarwal, O. Schwarzkopf, and M. Sharir. The overlay of lower envelopes and its applications. *Discrete Comput. Geom.*, 15:1–13, 1996.

[AST97] B. Aronov, M. Sharir, and B. Tagansky. The union of convex polyhedra in three dimensions. *SIAM J. Comput.*, 26:1670–1688, 1997.

[Bal95] I.J. Balaban. An optimal algorithm for finding segment intersections. In *Proc. 11th Sympos. Comut. Geom.*, pages 211–219, ACM Press, 1995.

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

[BCKO08] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, Berlin, 3rd edition, 2008.

[BD90] K.W. Bowyer and C.R. Dyer. Aspect graphs: An introduction and survey of recent results. *Internat. J. Imaging Sys. Technology*, 2:315–328, 1990.

[BD96] J.-D. Boissonnat and K. Dobrindt. On-line construction of the upper envelope of triangles and surface patches in three dimensions. *Comput. Geom.* 5:303–320, 1996.

[BDH99] K.-F. Böhringer, B.R. Donald, and D. Halperin. On the area bisectors of a polygon. *Discrete Comput. Geom.*, 22:269–285, 1999.

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

[Ber93] M. de Berg. *Ray Shooting, Depth Orders and Hidden Surface Removal*. Vol. 703 of *LNCS*, Springer, Berlin, 1993.

[BFH+10a] E. Berberich, E. Fogel, D. Halperin, M. Kerber, and O. Setter. Arrangements on parametric surfaces II: concretizations and applications. *Math. Comput. Sci.*, 4:67–91, 2010.

[BFH+10b] E. Berberich, E. Fogel, D. Halperin, K. Mehlhorn, and R Wein. Arrangements on parametric surfaces I: general framework and infrastructure. *Math. Comput. Sci.*, 4:45–66, 2010.

[BGH96] M. de Berg, L.J. Guibas, and D. Halperin. Vertical decompositions for triangles in 3-space. *Discrete Comput. Geom.*, 15:35–61, 1996.

[BHO07] M. de Berg, D. Halperin, and M.H. Overmars. An intersection-sensitive algorithm for snap rounding. *Comput. Geom.*, 36:159–165, 2007.

[BJ02] G. S. Brodal and R. Jacob. Dynamic planar convex hull. In *Proc. 43rd IEEE Sympos. Foundations Comp. Sci.*, pages 617–626, 2002.

[BK03] P. Braß and C. Knauer. On counting point-hyperplane incidences. *Comput. Geom.*, 25:13–20, 2003.

[BKS10] E. Berberich, M. Kerber, and M. Sagraloff. An efficient algorithm for the stratification and triangulation of an algebraic surface. *Comput. Geom.*, 43:257–278, 2010.

[BLW+93] A. Björner, M. Las Vergnas, N. White, B. Sturmfels, and G. Ziegler. *Oriented Matroids*, vol. 46 of *Encyclopedia of Mathematics*. Cambridge Univ. Press, 1993.

[BMS94] C. Burnikel, K. Mehlhorn, and S. Schirra. On degeneracy in geometric computations. In *Proc. ACM-SIAM Sympos. Discrete Algorithms*, pages 16–23, 1994.

[BP00] J.-D. Boissonnat and F.P. Preparata. Robust plane sweep for intersecting segments. *SIAM J. Comput.*, 29:1401–1421, 2000.

[BPR06] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics)*. Springer, New York, 2006.

[Bri93] E. Brisson. Representing geometric structures in d dimensions: Topology and order. *Discrete Comput. Geom.*, 9:387–426, 1993.

[BS08] E. Berberich and M. Sagraloff. A generic and flexible framework for the geometrical and topological analysis of (algebraic) surfaces. In *Proc. ACM Sympos. Solid and Physical Modeling*, pages 171–182, 2008.

[BS16] S. Basu and M. Sombra. Polynomial partitioning on varieties of codimension two and point-hypersurface incidences in four dimensions. *Discrete Comput. Geom.* 55:158–184, 2016.

[BSTY98] J.-D. Boissonnat, M. Sharir, B. Tagansky, and M. Yvinec. Voronoi diagrams in higher dimensions under certain polyhedral distance functions. *Discrete Comput. Geom.*, 19:485–519, 1998.

[BY98] J.-D. Boissonnat and M. Yvinec. *Algorithmic Geometry*. Cambridge University Press, 1998. Translated by H. Brönnimann.

[Can93] J. Canny. Computing roadmaps in general semialgebraic sets. *Comput. J.*, 36:504–514, 1993.

[CEG+90] K. Clarkson, H. Edelsbrunner, L.J. Guibas, M. Sharir, and E. Welzl. Combinatorial complexity bounds for arrangements of curves and spheres. *Discrete Comput. Geom.*, 5:99–160, 1990.

[CEGS91] B. Chazelle, H. Edelsbrunner, L.J. Guibas, and M. Sharir. A singly-exponential stratification scheme for real semi-algebraic varieties and its applications. *Theoret. Comput. Sci.*, 84:77–105, 1991. An improved bound appears in the proceedings verion: *Proc. ICALP*, LNCS 372, pages 179–193, Springer, Berlin, 1989.

[Cha96] T.M. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. *Discrete Comput. Geom.*, 16:369–387, 1996.

[Cha00] T.M. Chan. Random sampling, halfspace range reporting, and construction of $(\leq k)$-levels in three dimensions. *SIAM J. Comput.*, 30:561–575, 2000.

[Cha01] T.M. Chan. Dynamic planar convex hull operations in near-logarithmic amortized time. *J. ACM*, 48:1–12, 2001.

[Cha03] T.M. Chan. On levels in arrangements of curves. *Discrete Comput. Geom.*, 29:375–393, 2003.

[Cla88] K.L. Clarkson. A randomized algorithm for closest-point queries. *SIAM J. Comput.*, 17:830–847, 1988.

[CS89] K.L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.

[CT15] T.M. Chan and K. Tsakalidis. Optimal deterministic algorithms for 2-d and 3-d shallow cuttings. In *Proc. 31st Sympos. Comput. Geom.*, pages 719–732, vol. 34 of LIPIcs, Dagstuhl, 2015.

[Dey98] T.K. Dey. Improved bounds on planar k-sets and related problems. *Discrete Comput. Geom.*, 19:373–382, 1998.

[DL89] D.P. Dobkin and M.J. Laszlo. Primitives for the manipulation of three-dimensional subdivisions. *Algorithmica*, 4:3–32, 1989.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer, Heidelberg, 1987.

[Ede89] H. Edelsbrunner. The upper envelope of piecewise linear functions: Tight complexity bounds in higher dimensions. *Discrete Comput. Geom.*, 4:337–343, 1989.

[EG89] H. Edelsbrunner and L.J. Guibas. Topologically sweeping an arrangement. *J. Comput. Syst. Sci.*, 38:165–194, 1989. Corrigendum in 42:249–251, 1991.

[EGH+89] H. Edelsbrunner, L.J. Guibas, J. Hershberger, J. Pach, R. Pollack, R. Seidel, M. Sharir, and J. Snoeyink. Arrangements of Jordan arcs with three intersections per pair. *Discrete Comput. Geom.*, 4:523–539, 1989.

[EGP+92] H. Edelsbrunner, L.J. Guibas, J. Pach, R. Pollack, R. Seidel, and M. Sharir. Arrangements of curves in the plane: Topology, combinatorics, and algorithms. *Theor. Comp. Sci.*, 92:319–336, 1992.

[EGS89] H. Edelsbrunner, L.J. Guibas, and M. Sharir. The upper envelope of piecewise linear functions: algorithms and applications. *Discrete Comput. Geom.*, 4:311–336, 1989.

[EGS90] H. Edelsbrunner, L.J. Guibas, and M. Sharir. The complexity of many cells in arrangements of planes and related problems. *Discrete Comput. Geom.*, 5:197–216, 1990.

[EHS04] E. Ezra, D. Halperin, and M. Sharir. Speeding up the incremental construction of the union of geometric objects in practice. *Comput. Geom.*, 27:63–85, 2004.

[EK08] A. Eigenwillig and M. Kerber. Exact and efficient 2d-arrangements of arbitrary algebraic curves. In *Proc. 19th ACM-SIAM Sympos. Discrete Algorithms*, pages 122–131, 2008.

[EM90] H. Edelsbrunner and E.P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9:66–104, 1990.

[Eri96] J. Erickson. New lower bounds for Hopcroft's problem. *Discrete Comput. Geom.*, 16:389–418, 1996.

[ERK96] H. Everett, J.-M. Robert, and M. van Kreveld. An optimal algorithm for the ($\leq k$)-levels, with applications to separation and transversal problems. *Internat. J. Comput. Geom. Appl.*, 6:247–261, 1996.

[ES09] E. Ezra and M. Sharir. On the union of fat tetrahedra in three dimensions. *J. ACM*, 57(1), 2009.

[ESS93] H. Edelsbrunner, R. Seidel, and M. Sharir. On the zone theorem for hyperplane arrangements. *SIAM J. Comput.*, 22:418–429, 1993.

[EW86] H. Edelsbrunner and E. Welzl. Constructing belts in two-dimensional arrangements with applications. *SIAM J. Comput.*, 15:271–284, 1986.

[Ezr11] E. Ezra. On the union of cylinders in three dimensions. *Discrete Comput. Geom.*, 45:45–64, 2011.

[FH13] E Fogel and D. Halperin. Polyhedral assembly partitioning with infinite translations or the importance of being exact. *IEEE Trans. Autom. Sci. Eng.*, 10:227–241, 2013.

[FHH+00] E. Flato, D. Halperin, I. Hanniel, O. Nechushtan, and E. Ezra. The design and implementation of planar maps in CGAL. *ACM J. Exper. Algorithmics*, 5:13, 2000.

[FHW12] E. Fogel, D. Halperin, and R. Wein. *CGAL Arrangements and Their Applications—A Step-by-Step Guide*. Vol. 7 of *Geometry and computing*, Springer, 2012.

[FKMS05] S. Funke, C. Klein, K. Mehlhorn, and S. Schmitt. Controlled perturbation for Delaunay triangulations. In *Proc. 16th ACM-SIAM Sympos. Discrete Algorithms*, pages 1047–1056, 2005.

[FM91] S. Fortune and V. Milenkovic. Numerical stability of algorithms for line arrangements. In *Proc. 17th Sympos. Comput. Geom.*, pages 334–341, ACM Press, 1991.

[For99] S. Fortune. Vertex-rounding a three-dimensional polyhedral subdivision. *Discrete Comput. Geom.*, 22:593–618, 1999.

[GGHT97] M. Goodrich, L.J. Guibas, J. Hershberger, and P. Tanenbaum. Snap rounding line segments efficiently in two and three dimensions. *Proc. 13th Sympos. Comput. Geom.*, pages 284–293, ACM Press, 1997.

[GK15] L. Guth and N.H. Katz. On the Erdős distinct distances problem in the plane. *Annals Math.*, 181:155–190, 2015.

[GM98] L.J. Guibas and D. Marimont. Rounding arrangements dynamically. *Internat. J. Comput. Geom. Appl.*, 8:157–176, 1998.

[GO95] A. Gajentaan and M.H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom.*, 5:165–185, 1995.

[Goo93] M. T. Goodrich. Constructing arrangements optimally in parallel. *Discrete Comput. Geom.*, 9:371–385, 1993.

[Grü67] B. Grünbaum. *Convex Polytopes*. John Wiley & Sons, New York, 1967.

[Grü71] B. Grünbaum. Arrangements of hyperplanes. *Congr. Numer.*, 3:41–106, 1971.

[Grü72] B. Grünbaum. *Arrangements and Spreads*. Regional Conf. Ser. Math., AMS, Providence, 1972.

[GS85] L.J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and computation of Voronoi diagrams. *ACM Trans. Graph.*, 4:74–123, 1985.

[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.

[Gut14] L. Guth. Polynomial partitioning for a set of varieties. *Math. Proc. Cambridge Phil. Soc.*, 159:459–469, 2015.

[GY86] D.H. Greene and F.F. Yao. Finite-resolution computational geometry. *Proc. 27th IEEE Sympos. Found. Comp. Sci.*, pages 143–152, 1986.

[Hal92] D. Halperin. *Algorithmic Motion Planning via Arrangements of Curves and of Surfaces*. Ph.D. thesis, Computer Science Department, Tel-Aviv University, 1992.

[Hal94] D. Halperin. On the complexity of a single cell in certain arrangements of surfaces related to motion planning. *Discrete Comput. Geom.*, 11:1–34, 1994.

[Her89] J. Hershberger. Finding the upper envelope of n line segments in $O(n \log n)$ time. *Inf. Proc. Lett.*, 33:169–174, 1989.

[Her13] J. Hershberger. Stable snap rounding. *Comput. Geom.*, 46:403–416, 2013.

[HH02] S. Hirsch and D. Halperin. Hybrid motion planning: Coordinating two discs moving among polygonal obstacles in the plane. In *Algorithmic Foundations of Robotics V*. Vol. 7 of *Springer Tracts in Advanced Robotics*, pages 239–256, 2002.

[HH08] I. Haran and D. Halperin. An experimental study of point location in planar arrangements in CGAL. *ACM J. Exper. Algorithmics*, 13, 2008.

[HJW90] T. Hagerup, H. Jung, and E. Welzl. Efficient parallel computation of arrangements of hyperplanes in d dimensions. In *Proc. 2nd ACM Sympos. Parallel Algorithms Architect.*, pages 290–297, 1990.

[HKH12] M. Hemmer, M. Kleinbort, and D. Halperin. Improved implementation of point location in general two-dimensional subdivisions. In *Proc. 20th European Sympos. Algorithms*, pages 611–623. Vol. 7501 of *LNCS*, Springer, 2012.

[HKS16] S. Har-Peled, H. Kaplan, and M. Sharir. Approximating the k-level in three-dimensional plane arrangements. In *Proc. 27th ACM-SIAM Sympos. Discrete Algorithms*, pages 1193–1212, 2016.

[HL04] D. Halperin and E. Leiserowitz. Controlled perturbation for arrangements of circles. *Int. J. Comput. Geometry Appl.*, 14:277–310, 2004.

[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.

[Hob99] J.D. Hobby. Practical segment intersection with finite precision output. *Comput. Geom.*, 13:199–214, 1999.

[HP02] D. Halperin and E. Packer. Iterated snap rounding. *Comput. Geom.*, 23:209–225, 2002.

[HS94] D. Halperin and M. Sharir. New bounds for lower envelopes in three dimensions, with applications to visibility in terrains. *Discrete Comput. Geom.*, 12:313–326, 1994.

[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. Goldbergs et al., editors, *Algorithmic Foundations of Robotics*, pages 495–511. A.K. Peters, Boston, 1995.

[HS98] D. Halperin and C.R. Shelton. A perturbation scheme for spherical arrangements with application to molecular modeling. *Comput. Geom.*, 10:273–287, 1998.

[HSH10] M. Hemmer, O. Setter, and D. Halperin. Constructing the exact Voronoi diagram of arbitrary lines in three-dimensional space—with fast point-location. In *Proc. 18th European Sympos. Algorithms*, pages 398–409. Vol. 6346 of *LNCS*, Springer, 2010.

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

[JP14] A.G. Jørgensen and S. Pettie. Threesomes, degenerates, and love triangles. In *55th IEEE Sympos Found. Comp. Sci.*, pages 621–630, 2014.

[KBFS12] A. Kröller, T. Baumgartner, S.P. Fekete, and C. Schmidt. Exact solutions and bounds for general art gallery problems. *ACM J. Exper. Algorithmics*, 17, 2012.

[Ket99] L. Kettner. Using generic programming for designing a data structure for polyhedral surfaces. *Comput. Geom.*, 13:65–90, 1999.

[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.

[KLPY99] V. Karamcheti, C. Li, I. Pechtchanski, and C. Yap. *The CORE Library Project*, 1.2 edition, 1999. `http://www.cs.nyu.edu/exact/core/`.

[KMSS12] H. Kaplan, J. Matoušek, M. Sharir, and Z. Safernová. Unit distances in three dimensions. *Combinat. Probab. Comput.* 21:597–610, 2012.

[Kol04a] V. Koltun. Almost tight upper bounds for vertical decompositions in four dimensions. *J. ACM*, 51:699–730, 2004.

[Kol04b] V. Koltun. Sharp bounds for vertical decompositions of linear arrangements in four dimensions. *Discrete Comput. Geom.*, 31:435–460, 2004.

[KRS11] H. Kaplan, E. Ramos, and M. Sharir. The overlay of minimization diagrams in a randomized incremental construction. *Discrete Comput. Geom.*, 45:371–382, 2011.

[KS03] V. Koltun and M. Sharir. The partition technique for overlays of envelopes. *SIAM J. Comput.*, 32:841–863, 2003.

[KS05] V. Koltun and M. Sharir. Curve-sensitive cuttings. *SIAM J. Comput.*, 34:863–878, 2005.

[KS09] V. Koltun and M. Sharir. On overlays and minimization diagrams. *Discrete Computat. Geom.*, 41:385–397, 2009.

[Mat92] J. Matoušek. Reporting points in halfspaces. *Comput. Geom.*, 2:169–186, 1992.

[Mat93] J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10:157–182, 1993.

[Mat02] J. Matoušek. *Lectures on Disrete Geometry.* Vol. 212 of *Graduate Texts in Mathematics*, Springer, New York, 2002.

[McM70] P. McMullen. The maximal number of faces of a convex polytope. *Mathematika*, 17:179–184, 1970.

[Mey06] M. Meyerovitch. Robust, generic and efficient construction of envelopes of surfaces in three-dimensional spaces. In *proc. 14th European Sympos. Algorithms*, pages 792–803. Vol. 4168 of *LNCS*, Sprnger, 2006.

[Mil00] V.J. Milenkovic. Shortest path geometric rounding. *Algorithmica*, 27:57–86, 2000.

[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. Comp. Sci.* pages 49–58, 1991.

[MN00] K. Mehlhorn and S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing.* Cambridge University Press, 2000.

[MOS11] K. Mehlhorn, R. Osbild, and M. Sagraloff. A general approach to the analysis of controlled perturbation algorithms. *Comput. Geom.*, 44:507–528, 2011.

[MPS+94] J. Matoušek, J. Pach, M. Sharir, S. Sifrony, and E. Welzl. Fat triangles determine linearly many holes. *SIAM J. Comput.*, 23:154–169, 1994.

[MS07] V. Milenkovic and E. Sacks. An approximate arrangement algorithm for semi-algebraic curves. *Int. J. Comput. Geometry Appl.*, 17:175–198, 2007.

[MT06] A. Marcus and G. Tardos. Intersection reverse sequences and geometric applications. *J. Comb. Theory, Ser. A*, 113:675–691, 2006.

[Mul93] K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms*. Prentice Hall, Englewood Cliffs, 1993.

[MYB+11] J.W. Martin, A.K. Yan, C. Bailey-Kellogg, P. Zhou, and B.R. Donald. A geometric arrangement algorithm for structure determination of symmetric protein homo-oligomers from NOEs and RDCs. *J. Comput. Bio.*, 18:1507–1523, 2011.

[Niv08] G. Nivasch. An improved, simple construction of many halving edges. In J.E. Goodman and J. Pach and R. Pollack, editor, *Surveys on Discrete and Computational Geometry: Twenty Years Laterms*, vol. 453 of *Contemporary Mathematics*, pages 299–305. AMS, Providence, 2008.

[Niv15] G. Nivasch. On the zone of a circle in an arrangement of lines. *Elect. Notes Discrete Math.*, 49:221–231, 2015.

[O'R98] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 2nd edition, 1998.

[PA95] J. Pach and P.K. Agarwal. *Combinatorial Geometry*. John Wiley & Sons, New York, 1995.

[Pac08] E. Packer. Iterated snap rounding with bounded drift. *Comput. Geom.*, 40:231–251, 2008.

[Pac11] E. Packer. Controlled perturbation of sets of line segments in $\mathbb{R}^2$ with smart processing order. *Comput. Geom.*, 44:265–285, 2011.

[Pet15] Seth Pettie. Sharp bounds on davenport-schinzel sequences of every order. *J. ACM*, 62:36, 2015.

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

[PS89] J. Pach and M. Sharir. The upper envelope of piecewise linear functions and the boundary of a region enclosed by convex plates: Combinatorial analysis. *Discrete Comput. Geom.*, 4:291–309, 1989.

[PS99] J. Pach and M. Sharir. On the boundary of the union of planar convex sets. *Discrete Comput. Geom.*, 21:321–328, 1999.

[PS09] J. Pach and M. Sharir. *Combinatorial Geometry and its Algorithmic Applications: The Alcalá Lecturess*. AMS, Providence, 2009.

[PSS03] J. Pach, I. Safruti, and M. Sharir. The union of congruent cubes in three dimensions. *Discrete Comput. Geom.*, 30:133–160, 2003.

[Raa99] S. Raab. Controlled perturbation for arrangements of polyhedral surfaces with application to swept volumes. In *Proc. 15th Sympos Comput. Geom.*, pages 163–172, ACM Press, 1999.

[Ram99] E.A. Ramos. On range reporting, ray shooting and k-level construction. In *Proc. 15th Sympos. Comput. Geom.*, pages 390–399, ACM Press, 1999.

[Raz15] O.E. Raz. On the zone of the boundary of a convex body. *Comput. Geom.*, 48:333–341, 2015.

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

[SAL93] A. Schweikard, J.E. Adler, and J.-C. Latombe. Motion planning in stereotaxic radiosurgery. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 764–774, 1993.

[Sei98] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete Comput. Geom.*, 19:1–17, 1998.

[SH02] H. Shaul and D. Halperin. Improved construction of vertical decompositions of three-dimensional arrangements. In *Proc. 18th Sympos. Comput. Geom.*, pages 283–292, ACM Press, 2002.

[Sha91] M. Sharir. On k-sets in arrangements of curves and surfaces. *Discrete Comput. Geom.*, 6:593–613, 1991.

[Sha94] M. Sharir. Almost tight upper bounds for lower envelopes in higher dimensions. *Discrete Comput. Geom.*, 12:327–345, 1994.

[Sha11] M. Sharir. An improved bound for k-sets in four dimensions. *Combin. Prob. Comput.*, 20:119–129, 2011.

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

[SMK11] E. Sacks, V. Milenkovic, and M.-H. Kyung. Controlled linear perturbation. *Computer-Aided Design*, 43:1250–1257, 2011.

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

[SSH10] O. Setter, M. Sharir, and D. Halperin. Constructing two-dimensional Voronoi diagrams via divide-and-conquer of envelopes in space. In *Transactions on Computational Science IX*, pages 1–27. Vol. 6290 of *LNCS*, Springer, 2010.

[SST01] M. Sharir, S. Smorodinsky, and G. Tardos. An improved bound for k-sets in three dimensions. *Discrete Comput. Geom.*, 26:195–204, 2001.

[Szé97] L.A. Székely. Crossing numbers and hard Erdős problems in discrete geometry. *Combin. Prob. Comput.*, 6:353–358, 1997.

[Tag96] B. Tagansky. A new technique for analyzing substructures in arrangements of piecewise linear surfaces. *Discrete Comput. Geom.*, 16:455–479, 1996.

[Tót01] G. Tóth. Point sets with many k-sets. *Discrete Comput. Geom.*, 26:187–194, 2001.

[TT98] Hisao Tamaki and Takeshi Tokuyama. How to cut pseudoparabolas into segments. *Discrete Comput. Geom.*, 19:265–290, 1998.

[WFZH07] R. Wein, E. Fogel, B. Zukerman, and D. Halperin. Advanced programming techniques applied to CGAL's arrangement package. *Comput. Geom.*, 38:37–63, 2007.

[WS88] A. Wiernik and M. Sharir. Planar realizations of nonlinear Davenport-Schinzel sequences by segments. *Discrete Comput. Geom.*, 3:15–47, 1988.

[Zah11] J. Zahl. An improved bound on the number of point-surface incidences in three dimensions. *Contrib. Discrete Math.* 8:100–121, 2013.

[Zas75] T. Zaslavsky. *Facing up to Arrangements: Face-Count Formulas for Partitions of Space by Hyperplanes.* Vol. 154 of *Memoirs of the AMS*, AMS, Procidence, 1975.