21 ARRANGEMENTS

Dan Halperin

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 21.2 we describe substructures in arrangements and their combinatorial complexity. Section 21.3 deals with data structures for representing arrangements and with special refinements of arrangements. The following two sections deal with algorithms: algorithms for constructing full arrangements are described in Section 21.4, and algorithms for constructing substructures in Section 21.5. Situations where arrangements have lower complexity than the general worst-case bounds are presented in Section 21.6. In Section 21.7 we discuss the relation between arrangements and other structures. Several applications of arrangements are reviewed in Section 21.8.

21.1 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.

21.1.1 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 $\mathcal{H}$; a k-dimensional cell in

$\mathcal{A}(\mathcal{H})$, for $0 \le k \le d-1$, is a maximal connected region 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.

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

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

EXAMPLE: AN ARRANGEMENT OF LINES

Let $\mathcal{L}$ be a finite set of lines in the plane, and let $\mathcal{A}(\mathcal{L})$ be a simple arrangement induced by $\mathcal{L}$. 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 $\mathcal{L}$.

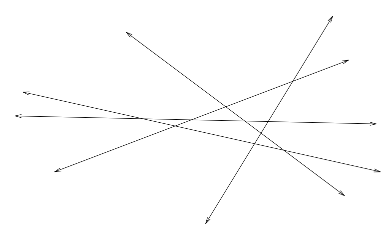


FIGURE 21.1.1
A simple arrangement of 5 lines.
It has 10 vertices, 25 edges (10 of which are unbounded), and 16 faces (10 of which are unbounded).

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 21.1.1

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.

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-Oh notation. Theorem 21.1.1 implies the following

COROLLARY 21.1.2

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*.*

21.1.2 ARRANGEMENTS OF CURVES AND SURFACES

We now introduce more general arrangements, allowing for objects that are non-linear and/or bounded. We distinguish between planar arrangements and arrangements in three or higher dimensions. For planar arrangements we only require 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 (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.

In both cases we 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. 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 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.

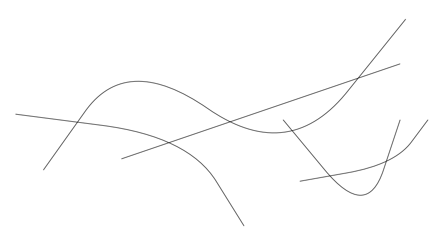


FIGURE 21.1.2
A simple arrangement of 5 bounded arcs, where $s = 2$*. It has 17 vertices (10 of which are arc endpoints), 19 edges, and 4 faces (one of which is unbounded).*

We assume that the arcs in $\mathcal{C}$ are in ***general position***, namely, 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 21.1.3

Let $\mathcal{C}$ be 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.

THREE AND HIGHER DIMENSIONS

We denote the coordinate axes of $\mathbb{R}^d$ by $x_1, x_2, \ldots, x_d$. For a collection $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ 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.
3. Every d-tuple of 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:

- 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 Chapters 29 and 47). 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 arrangement. 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.
- Assumption (5) 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

[Sei94].

THEOREM 21.1.4

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 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 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 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 intersection 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, [PA95, Chapter 11] and [Szé]. Another noteworthy topic that for lack of space we will not cover here is ***combinatorial equivalence*** of arrangements [BLW$^+$93].

21.2 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, together with combinatorial complexity bounds for these substructures. 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 21.1.

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 in 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 only discuss 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 attain the minimum.

Zone: For an additional curve γ, the collection of faces of the arrangement $\mathcal{A}(\mathcal{C})$ intersected by γ. See Figure 21.4.1. The zone is sometimes called the ***horizon***.

Single cell: In this section, a d-cell 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. Similarly a vertex in a simple arrangement of lines has four sides, and it is a ***0-border*** of four faces. The definition extends 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 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 ***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. 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 to k-sets in point configurations; see Chapter 1.

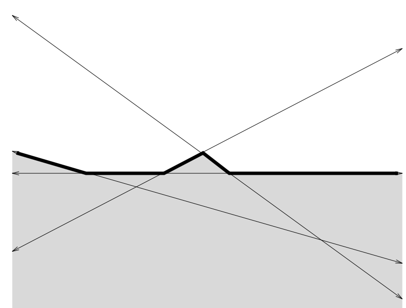


FIGURE 21.2.1
The bold line is the 2-level of the arrangement of four lines. The shaded region is the $\leq$2-level of the arrangement.

Union boundary: If each surface s in an arrangement in $\mathbb{R}^d$ is the boundary of a d-dimensional object then the boundary of the union of the objects is another interesting substructure. The study of union boundary has largely been motivated by robot motion planning problems; for details see Chapter 40.

$\alpha(n)$: The extremely slowly-growing functional inverse of Ackermann's function. See Section 40.4.

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. 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 fixed (low) dimensional space, 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 the 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 21.2.1, 21.2.2, 21.2.3, and 21.2.4 respectively. Bounds in the tables using ε should read "for any $\varepsilon > 0$."

TABLE 21.2.1 Substructures in arrangements of n 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	$O(n\sqrt{k+1}/\log^*(k+1))$	[PSS92]
	$\Omega(n\log(k+1))$	[EW85]
$(\leq k)$-level	$\Theta(n(k+1))$	[AG86]

TABLE 21.2.2 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]
zone of a hyperplane	$\Theta(n^{d-1})$	[ESS93]
single cell	$\Theta(n^{\lfloor \frac{d}{2} \rfloor})$	upper bound theorem [McM70]
m cells	$O(m^{1/2} n^{d/2} \log^{(\lfloor \frac{d}{2} \rfloor -1)/2} n)$	this bound is almost tight [AMS94b]; for bounds on the number of facets see [AA92]
$(\leq k)$-level	$\Theta(n^{\lfloor d/2 \rfloor} (k+1)^{\lceil d/2 \rceil})$	[CS89]

Aronov et al. [APS93] showed that the maximum complexity of the zone of a p-dimensional algebraic surface of constant maximum degree in an arrangement of hyperplanes in $\mathbb{R}^d$ is $O(n^{\lfloor (d+p)/2 \rfloor} \log^{\gamma} n)$, where $\gamma = (d+p) (\bmod 2)$. This bound is almost tight in the worst case.

CURVES

For a collection $\mathcal{C}$ of n well-behaved curves as defined in Section 21.1, 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 symbol, and it is almost linear in n for any fixed s. Davenport-Schinzel sequences play a central role in the analysis of substructures of arrangements of curves and surfaces. See Section 40.4.

THEOREM 21.2.1

For a set $\mathcal{C}$ of n x-monotone Jordan arcs such that each pair intersect 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 zone in arrangements of surfaces [HS95a].

The results in Table 21.2.3 are for Jordan arcs (bounded curves). There are slightly better bounds in the case of unbounded curves.

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.

TABLE 21.2.3 Substructures in arrangements of n Jordan arcs.

SUBSTRUCTURE	BOUND	NOTES
envelope	$\Theta(\lambda_{s+2}(n))$	
single face, zone	$\Theta(\lambda_{s+2}(n))$	[GSS89]
m cells	$O(m^{1/2}\lambda_{s+2}(n))$	[EGP+92]
	$\Omega(m^{2/3}n^{2/3})$	lower bound for lines
$(\leq k)$-level	$\Theta((k+1)^2\lambda_{s+2}(\lfloor\frac{n}{k+1}\rfloor))$	[Sha91]

TABLE 21.2.4 Substructures in arrangements of n surfaces.

OBJECTS	SUBSTRUCTURE	BOUND	NOTES
surfaces in $\mathbb{R}^d$	lower envelope	$O(n^{d-1+\varepsilon})$	[HS94],[Sha94]
surfaces in $\mathbb{R}^3$	single cell, zone	$O(n^{2+\varepsilon})$	[HS95a]
$(d-1)$-simplices in $\mathbb{R}^d$	lower envelope	$\Theta(n^{d-1}\alpha(n))$	[Ede89]
	single cell, zone	$O(n^{d-1}\log n)$	[Tag95]
$(d-1)$-spheres in $\mathbb{R}^d$	lower envelope, single cell	$\Theta(n^{\lceil\frac{d}{2}\rceil})$	linearization

Sum of Squares. 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. [AMS94b] 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. They use it to obtain bounds on the complexity of m cells in 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 super-positioning the minimization diagram of A with that of B. Given two sets $\mathcal{C}_1$ and $\mathcal{C}_2$ of n x-monotone Jordan arcs each, such that no pair of (the collection of $2n$) arcs intersect 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 of n well-behaved surfaces each the complexity of the overlay is $O(n^{2+\epsilon})$ [ASS95]. 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).

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 21.2.1. This is a long-standing open problem in combinatorial geometry.

2. What is the complexity of m faces in an arrangement of well-behaved Jordan arcs? For lines a tight bound is known, whereas for curves a considerable gap still exists—see Table 21.2.3.

3. What is the complexity of the union boundary of n infinite unit cylinders in 3-space (namely, each cylinder is the Minkowski sum of a line in 3-space and a unit ball)? Only trivial bounds are known to-date: the known upper bound is $O(n^3)$ (Theorem 21.1.4), and a lower bound of $\Omega(n^2)$ is easy to construct.

21.3 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 29 and 40.

GLOSSARY

Let $\mathcal{S}$ be a collection of surfaces in $\mathbb{R}^d$ (or curves in $\mathbb{R}^2$) as defined in Section 21.1, and $\mathcal{A}(\mathcal{S})$ be the arrangement induced by $\mathcal{S}$. Let c_1 be a k_1-dimensional cell of $\mathcal{A}(\mathcal{S})$ and c_2 be 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:*** Most 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***.

21.3.1 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 $\mathcal{A}(\mathcal{S})$ $-1 \le k \le d{+}1$, and an arc between two nodes if the corresponding cells are incident to one another. For an arrangement of n surfaces in $\mathbb{R}^d$ the number of nodes in V is

$O(n^d)$ by Theorem 21.1.4. This is also a bound on the number of arcs in E: every cell (beside 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 their connectivity, it misses on *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 ***quad-edge structure*** of Guibas and Stolfi [GS85] and the three-dimensional ***facet-edge structure*** of Dobkin and Laszlo [DL89]. The cell-tuple structure gives a simple and uniform representation of the incidence 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 to represent 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 algebraic geometry representations. There however the skeleton, or ***roadmap***, is far more complicated (indeed it represents more general arrangements); see Chapter 40.

21.3.2 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 such 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 open problem below.)

BOTTOM VERTEX DECOMPOSITION OF 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). 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 21.3.1. Repeat for all faces of $\mathcal{A}(\mathcal{L})$. The original arrangement together with the added edges constitute 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.

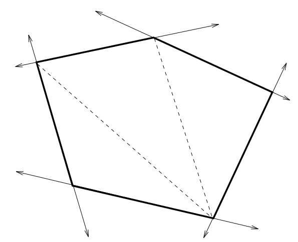


FIGURE 21.3.1
The bottom vertex decomposition of a face in an arrangement of lines.

VERTICAL DECOMPOSITION

The bottom vertex decomposition does not in general extend to arrangements of non-linear objects. 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 21.3.2. 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-dimensions. In higher dimensions it is still the general decomposition method that is known to have the best (lowest) complexity.

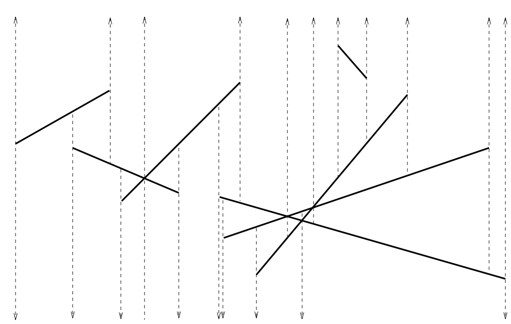


FIGURE 21.3.2
The vertical decomposition of an arrangement of segments: a vertical line segment is extended upward and downward from each vertex of the arrangement until it either hits another segment or extends to infinity.

The extension to higher dimensions is defined recursively. For details of the extension to three dimensions, see [CEG+90] for the case of spheres, and [dBGH96] for the case of triangles. The extension to higher dimensions is given in [CEGS91]. Table 21.3.1 summarizes the bounds on the maximum combinatorial complexity of the vertical decomposition for several types of arrangements and substructures. Certain assumptions that curves and surfaces are "well-behaved" are not detailed.

TABLE 21.3.1 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 $\mathbb{R}^d$	$O(n^{2d-3}\beta(n))$	[CEGS91], [SA95]
		$\beta(n) = \lambda_q(n)/n$, q a constant
triangles in $\mathbb{R}^3$	$\Theta(n^3)$	[dBGH96]
triangles in $\mathbb{R}^3$	$O(n^2\alpha(n)\log n + K)$	K is the complexity of $\mathcal{A}$ [Tag95]
surfaces in $\mathbb{R}^3$, single cell	$O(n^{2+\epsilon})$	[SS96]
surfaces in $\mathbb{R}^3$, $(\leq k)$-level	$O(n^{2+\epsilon}k)$	for this and more refined bounds see [AES95]
hyperplanes in $\mathbb{R}^4$	$O(n^4 \log n)$	[GHMS95]

THE SLICING AND CHOPPING THEOREMS

Aronov and Sharir devised alternative decomposition methods for arrangements of simplices [AS90], [AS94]. These are more involved than the decompositions described above and we omit their description here. These method were instrumental in obtaining improved combinatorial bounds and efficient algorithms for arrangements of simplices.

STRUCTURES FOR POINT LOCATION AND RAY SHOOTING

To access certain cells of an arrangement without traversing the entire arrangement, we need more elaborate structures than those described above. See Chapters 31 and 32 for details.

OPEN PROBLEMS

1. Obtain an improved combinatorial bound on the complexity of the vertical decomposition of arrangements of surfaces in four and higher dimensions. Such a result would have a wide-range effect on other combinatorial bounds, algorithms and a variety of applications of arrangements.

2. Consider the decomposition in Figure 21.3.2. Original edges of the arrangement are in fact *two-sided*. They have different sets of endpoints of vertical extensions on either side. This raises the question how to maintain these sets such that moving from one side of an edge to the other will be carried out efficiently (while traversing the arrangement). In the two-dimensional case there is an elegant solution where some of the vertical extensions are further extended so that along an edge of one face there are at most two neighboring faces on the other side [Cha86]. However, this idea does not seem to extend directly to higher dimensions.

21.4 ALGORITHMS FOR ARRANGEMENTS

This section covers constructing an arrangement: producing a representation of an arrangement in one of the forms described in the previous section (or similar). 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 each take unit time. 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; see Chapter 29.

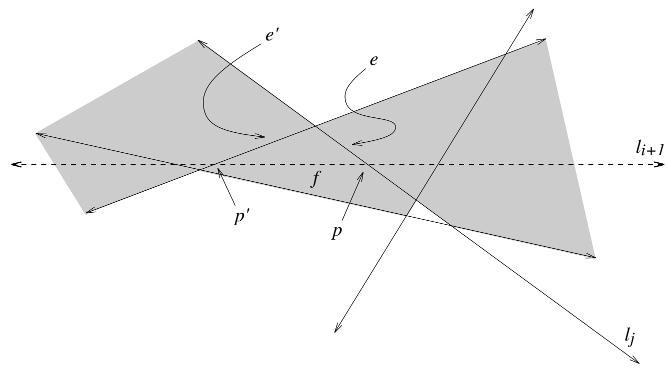
21.4.1 DETERMINISTIC ALGORITHMS

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 21.2). We describe it next 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 quad-edge representation for $\mathcal{A}(\mathcal{L}_i)$, so in addition to the incidence information we also have the order of edges along the boundary of each face readily available. 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

FIGURE 21.4.1
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.



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 $\mathcal{A}(\mathcal{L}_{i+1})$.

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 on 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 $\mathcal{A}(\mathcal{L}_i)$, and some of its edges. The amount of time spent is proportional to the number of edges we visit, and hence bounded by the complexity of the zone. The space required for the algorithm is the space to maintain the quad-edge structure. The same approach extends to higher dimensions. For details see [Ede87, Chapter 7] (note that the algorithm as described in [Ede87] uses the incidence graph for maintaining the arrangement).

THEOREM 21.4.1

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*.

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

THEOREM 21.4.2

Let $\mathcal{C}$ be a set of n Jordan arcs as defined in Section 21.1. 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 21.4.3

Let $\mathcal{C}$ be a set of n Jordan arcs as defined in Section 21.1. 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 $O(n\lambda_q(n)\log n + V\log n)$, where V is the combinatorial complexity of the vertical decomposition ($\lambda_q(n)$ is a near-linear function related to Davenport-Schinzel sequences), and q is a constant that depends on the degree of the surface patches and their boundaries. For details see [dBGH96].

Topological Sweep

Edelsbrunner and Guibas [EG89] devised an algorithm for constructing an arrangement of lines that requires only linear working storage and still 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.'

21.4.2 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 objects' distribution in space. See also Chapter 34.

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 21.3. At the end of this section we remark on the *derandomization* techniques used to turn these algorithms into deterministic algorithms.

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 21.4.4

Let C be a set of n Jordan arcs as defined in Section 21.1. The arrangement $\mathcal{A}(C)$ 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 21.3), 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.

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

THEOREM 21.4.5

Given a collection S of n algebraic surfaces in $\mathbb{R}^d$ as defined in Section 21.1, then for any $\varepsilon > 0$, a data structure of size $O(n^{2d-3+\varepsilon})$ for the arrangement $\mathcal{A}(S)$ can be constructed in $O(n^{2d-3+\varepsilon})$ time, 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, probably close to $O(n^d)$ for both. See [Can93],[BPR96] for algebraic geometry methods.

Derandomization

Techniques have been proposed to derandomize many of the randomized geometric algorithms, often without increase in their asymptotic running time; see Chapter 34. However, in most cases the randomized versions are conceptually much simpler and hence may be better candidates for efficient implementation.

21.4.3 OTHER ALGORITHMIC ISSUES

For algebraic geometry tools, see Chapter 29. See Chapter 35 for a discussion of precision and degeneracies. Parallel algorithms are discussed in Chapter 36.

21.5 CONSTRUCTING SUBSTRUCTURES

ENVELOPE AND SINGLE CELL IN ARRANGEMENTS OF HYPERPLANES

Any single cell in an arrangement of hyperplanes is a convex polytope. Also, the lower envelope of a collection of hyperplanes is the boundary of the cell that lies below all the hyperplanes. Therefore computing a single cell or an envelope in such an arrangement is the same as computing the polytope which is the intersection of the appropriate halfspaces. See Chapter 19.

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 $\mathbb{R}^{d+1}$.

LOWER ENVELOPE

The lower envelope of a collection of well-behaved curves can be computed 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 3-space, Agarwal et al. [ASS95] showed that a simple divide-and-conquer scheme can be used to compute the envelope of n surfaces in time $O(n^{2+\varepsilon})$. This is an application of the bound on the complexity of the overlay of envelopes cited in Section 21.2. 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 $\mathbb{R}^d$, and an efficient data structure for point location in the minimization diagram of surfaces in $\mathbb{R}^4$.

SINGLE CELL AND ZONE

All the results cited below for a single cell hold for the zone problem as well (see remark in Section 21.2 on the connection between the problems).

Computing a single face in an arrangement of n Jordan arcs as defined in Section 21.1 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]. There are also algorithms for special cases of planar arrangements such as arrangements of rays.

In three dimensions, Schwarzkopf and Sharir [SS96] give an algorithm with running time $O(n^{2+\varepsilon})$ for any $\varepsilon > 0$ to compute a single cell in an arrangement of n well-behaved surfaces. Algorithms with improved running time to compute a single cell in three-dimensional arrangements are known for arrangements of surfaces induced by certain motion planning problems [Hal92], [Hal94], and for arrangements of triangles [dBDS95].

LEVELS

In an arrangement of n lines in the plane the k-level can be computed in $O(n \log f + f \log^2 n)$ time [EW86],[Cha95], where f is the combinatorial complexity of the k-level, and the $(\leq k)$-level in worst-case optimal time $O(n \log n + kn)$ [ERvK93]. Algorithms for computing the $(\leq k)$-level in an arrangement of Jordan arcs and the $(\leq k)$-level in an arrangement of planes in $\mathbb{R}^3$ are described in [AdBMS94]. For computing the k-level in an arrangement of hyperplanes in $\mathbb{R}^d$ see [Cha95].

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 [AMS94a]. Algorithms for arrangements of planes are described in [EGS90], and for arrangements of triangles in 3-space in [AS90].

OPEN PROBLEMS

Devise efficient algorithms for computing:

1. The lower envelope of an arrangement of surfaces in five and higher dimensions; for an algorithm in four dimensions see [AAS94].

2. A single cell in an arrangement of surfaces in four and higher dimensions; for an algorithm in three dimensions see [SS96].

21.6 SPARSE ARRANGEMENTS

So far we have discussed arrangements of n objects in $\mathbb{R}^d$ where each object has constant descriptive complexity and the total complexity of the entire arrangement can be $\Omega(n^d)$ in the worst case. In many situations arrangements do not achieve this worst-case complexity, or there are additional parameters that control the complexity of the arrangement. In this section we survey several such situations.

Let $\mathcal{C}$ be a collection of n Jordan arcs, where each pair of arcs in $\mathcal{C}$ intersect at most a constant number of times, and with the additional condition that any vertical line intersects at most k of the curves in $\mathcal{C}$. In this case the maximum combinatorial complexity of the arrangement $\mathcal{A}(\mathcal{C})$ is $\Theta(nk)$. For an application of this result and for more results on arrangements with low ***vertical stabbing number*** (the number of objects stabbed by any vertical line) see [dBHOvK].

A general way to take advantage of reduced complexity of an arrangement is to construct the arrangement using an output-sensitive algorithm. However, by understanding the source of the reduced complexity it may be possible to devise algorithms that perform better than general purpose output-sensitive algorithms. In several cases this has indeed been achieved. The collection of atom spheres in the geometric model of molecules exhibits sparseness properties that have led to improved combinatorial bounds and relatively simple algorithms. These algorithms have been implemented and perform well in practice [HO94], [FHK+96]. Another area where results of this nature have been obtained is robot motion planning among "fat obstacles"; see Chapter 40.

ARRANGEMENTS OF CONVEX POLYTOPES

Consider the subdivision of 3-space induced by k convex polytopes with a total of n vertices. To bound the complexity of this arrangement we can regard this as an arrangement of $O(n)$ triangles in 3-space, implying an upper bound $O(n^3)$. However, Aronov et al. [ABE91] showed that the complexity of such an arrangement is $O(nk^2)$. More generally they showed that the complexity of an arrangement of k convex polytopes in $\mathbb{R}^d$ with a total of n facets is $\Theta(n^{\lfloor \frac{d}{2} \rfloor} k^{\lceil \frac{d}{2} \rceil})$.

A useful substructure in an arrangement of convex polytopes is the collection of ***maximally covered cells***, namely cells of the arrangement that are covered by more polytopes than any other cell in their immediate neighborhood [GHH+95]. The ability to access these cells efficiently has led to an efficient and practical algorithm to test whether an object consisting of polyhedral parts is interlocked (i.e., cannot be taken apart with two hands).

21.7 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'R94]. Edelsbrunner [Ede87, Chapter 12] describes a collection of problems stated for point configurations and solved by applying to their corresponding arrangements. See also Chapter 1.

Plücker coordinates are a tool that enables to treat k-flats in $\mathbb{R}^d$ as points or hyperplanes in a possibly different dimensional space. This has been taken advantage of in the study of families of lines in 3-space—see Chapter 32.

Lower envelopes (or more generally the k-level in arrangements) relate to Voronoi diagrams—see Chapter 20.

For the connection of arrangements to polytopes and zonotopes see [Ede87].

21.8 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 21.7 above and Chapters 1, 20, and 32.

Another strand of applications consists of the '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 is to solving the following problem: Given a convex polyhedron in 3-space, determine how many combinatorially distinct orthographic and perspective views it induces. 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].

Arrangements have been applied to solving problems in robot motion planning (Chapter 40) and several of its variants (Chapter 41). For example, the most efficient algorithm known for computing a collision free path for a polygonal robot translating and rotating among polygonal obstacles in the plane is based on computing a single connected component in an arrangement of surfaces in 3-space.

As mentioned earlier, arrangements on spheres are prevalent in applications. Beside in vision applications, they also occur in: computer-assisted radio-surgery [SAL93], molecular modeling, assembly planning (Section 41.3), and more.

Arrangements have been used to solve problems in many other areas. More applications can be found in the sources cited below and in several other chapters in this book.

21.9 SOURCES AND RELATED MATERIAL

FURTHER READING

The study of arrangements until the early 1970's 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 recent results for arrangements of curves and surfaces.

Chapters dedicated to arrangements of hyperplanes in recently published books: Mulmuley emphasizes randomized algorithms [Mul94], O'Rourke discusses basic combinatorics, relation to other structures and applications [O'R94], and Pach and Agarwal [PA95] discuss problems involving arrangements in discrete geometry.

RELATED CHAPTERS

Chapter 1: Finite point configurations
Chapter 5: Pseudoline arrangements
Chapter 6: Oriented matroids
Chapter 13: Basic properties of convex polytopes
Chapter 19: Convex hull computations
Chapter 20: Voronoi diagrams and Delaunay triangulations
Chapter 29: Computational real algebraic geometry
Chapter 30: Point location
Chapter 32: Ray shooting and lines in space
Chapter 34: Randomized algorithms
Chapter 36: Parallel algorithms in geometry
Chapter 40: Motion planning
Chapter 41: Robotics

REFERENCES

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

[AAS94] P.K. Agarwal, B. Aronov, and M. Sharir. Computing envelopes in four dimensions with applications. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 348–358, 1994.

[ABE91] B. Aronov, M. Bern, and D. Eppstein. Arrangements of polytopes, 1991. Manuscript.

[AdBMS94] P.K. Agarwal, M. de Berg, J. Matoušek, and O. Schwarzkopf. Constructing levels in arrangements and higher order Voronoi diagrams. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 67–75, 1994.

[AES95] P.K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 39–50, 1995.

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

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

[AMS94a] P.K. Agarwal, J. Matoušek, and O. Schwarzkopf. Computing many faces in arrangements of lines and segments. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 76–84, 1994.

[AMS94b] 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.

[APS93] B. Aronov, M. Pellegrini, and M. Sharir. On the zone of a surface in a hyperplane arrangement. *Discrete Comput. Geom.*, 9(2):177–186, 1993.

[AS90] B. Aronov and M. Sharir. Triangles in space or building (and analyzing) castles in the air. *Combinatorica*, 10(2):137–173, 1990.

[AS94] B. Aronov and M. Sharir. Castles in the air revisited. *Discrete Comput. Geom.*, 12(2):119–150, September 1994.

[ASS95] P.K. Agarwal, O. Schwarzkopf, and M. Sharir. The overlay of lower envelopes in three dimensions and its applications. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 182–189, 1995.

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

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

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

[BPR96] S. Basu, R. Pollack, and M.-F. Roy. Computing roadmaps of semi-algebraic sets. In *Proc. ACM Symp. Theory of Computing*, pages 168–173. 1996.

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

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

[CEG+90] K. Clarkson, H. Edelsbrunner, L. 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. 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 version: Proc. ICALP 1989, Lecture Notes in Computer Science 372, Springer-Verlag.

[Cha86] B. Chazelle. Filtering search: a new approach to query-answering. *SIAM J. Comput.*, 15:703–724, 1986.

[Cha95] T.M. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 10–19, 1995.

[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.

[dBDS95] M. de Berg, K. Dobrindt, and O. Schwarzkopf. On lazy randomized incremental construction. *Discrete Comput. Geom.*, 14:261–286, 1995.

[dBGH96] M. de Berg, L.J. Guibas, and D. Halperin. Vertical decompositions for triangles in 3-space. *Discrete Comput. Geom.*, 15:35–61, 1996.

[dBHOvK] M. de Berg, D. Halperin, M.H. Overmars, and M. van Kreveld. Sparse arrangements and the number of views of polyhedral scenes. *Internat. J. Comput. Geom. Appl.* To appear.

[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-Verlag, 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 (1991), 249–251.

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

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

[ERvK93] H. Everett, J.-M. Robert, and M. van Kreveld. An optimal algorithm for the ($\leq k$)-levels, with applications to separation and transversal problems. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 38–46, 1993.

[ESS93] H. Edelsbrunner, R. Seidel, and M. Sharir. On the zone theorem for hyperplane arrangements. *SIAM J. Comput.*, 22(2):418–429, 1993.

[EW85] H. Edelsbrunner and E. Welzl. On the number of line separations of a finite set in the plane. *J. Combin. Theory Ser. A*, pages 15–29, 1985.

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

[FHK+96] P.W. Finn, D. Halperin, L. Kavraki, J.-C. Latombe, R. Motwani, C. Shelton, and S. Venkatsubramanian. Geometric manipulation of flexible ligands. In *Proc. ACM Workshop on Applied Computational Geometry*, 1996. To appear.

[GHH+95] L.J. Guibas, D. Halperin, H. Hirukawa, J.-C. Latombe, and R.H. Wilson. A simple and efficient procedure for assembly partitioning under infinitesimal motions. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 2553–2560, 1995.

[GHMS95] L.J. Guibas, D. Halperin, J. Matoušek, and M. Sharir. On vertical decomposition of arrangements of hyperplanes in four dimensions. *Discrete Comput. Geom.*, 14:113–122, 1995.

[Grü67] B. Grünbaum. *Convex Polytopes.* Wiley, New York, NY, 1967.

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

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

[GS85] L.J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the 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.

[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. *Inform. Process. Lett.*, 33:169–174, 1989.

[HO94] D. Halperin and M.H. Overmars. Spheres, molecules, and hidden surface removal. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 113–122, 1994.

[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. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, editors, *Algorithmic Foundations of Robotics.* A. K. Peters, Boston, MA, 1995.

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

[Mul94] K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms.* Prentice Hall, Englewood Cliffs, NJ, 1994.

[O'R94] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1994.

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

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

[PSS92] J. Pach, W. Steiger, and E. Szemerédi. An upper bound on the number of planar k-sets. *Discrete Comput. Geom.*, 7:109–123, 1992.

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

[Sei94] R. Seidel. The nature and meaning of perturbations in geometric computing. In *Proc. STACS*, volume 775 of *Lecture Notes in Computer Science*, pages 3–17, Berlin, 1994. Springer-Verlag.

[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.

[SS96] O. Schwarzkopf and M. Sharir. Vertical decomposition of a single cell in a three-dimensional arrangement of surfaces and its applications. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 20–29, 1996.

[Szé] L.A. Székely. Crossing numbers and hard Erdös problems in discrete geometry. *Combinatorics, Probability and Computing.* To appear.

[Tag95] B. Tagansky. A new technique for analyzing substructures in arrangements. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 200–210, 1995.

[Zas75] T. Zaslavsky. *Facing up to Arrangements: Face-Count Formulas for Partitions of Space by Hyperplanes*, volume 1:154 of *Memoirs Amer. Math. Soc.* Amer. Math. Soc., Providence, RI, 1975.