SIAM J. COMPUT.
Vol. 15, No. 2, May 1986

CONSTRUCTING ARRANGEMENTS OF LINES AND HYPERPLANES WITH APPLICATIONS*

H. EDELSBRUNNER†, J. O'ROURKE‡, AND R. SEIDEL§

Abstract. A finite set of lines partitions the Euclidean plane into a cell complex. Similarly, a finite set of $(d-1)$-dimensional hyperplanes partitions d-dimensional Euclidean space. An algorithm is presented that constructs a representation for the cell complex defined by n hyperplanes in optimal $O(n^d)$ time in d dimensions. It relies on a combinatorial result that is of interest in its own right. The algorithm is shown to lead to new methods for computing λ-matrices, constructing all higher-order Voronoi diagrams, halfspatial range estimation, degeneracy testing, and finding minimum measure simplices. In all five applications, the new algorithms are asymptotically faster than previous results, and in several cases are the only known methods that generalize to arbitrary dimensions. The algorithm also implies an upper bound of 2^{cn^d}, c a positive constant, for the number of combinatorially distinct arrangements of n hyperplanes in E^d.

Key words. arrangements, configurations, geometric transformation, combinatorial geometry, computational geometry, optimal algorithm

1. Introduction. Let H denote a finite set of lines in the Euclidean plane E^2. H determines a partition of the plane called the *arrangement* $A(H)$ *of* H or the *cell complex induced by H*. $A(H)$ consists of *vertices* (intersections of lines), *edges* (maximal connected components of the lines containing no vertex), and *regions* (maximal connected components of E^2 containing no edge or vertex). All geometric entities of this paper will be defined in Euclidean space which should make clear that our discussion does not take the projective view. However, no essential use is made of the concept of distance which implies that all results, but the ones on minimum measure simplices in § 4.5, also hold in real affine space.

Arrangements of lines have been studied from various mathematical points of view since Steiner [St] in 1826. The first attempt to provide a systematic exposition of the subject was made in 1967 in Grünbaum [G1], and to a more exhaustive extent five years later in Grünbaum [G2]. In spite of the extensive literature on arrangements of lines, there is a host of easily formulated but unsolved questions in this area. The interested reader is referred to [G2] where many open conjectures are stated. Recent advances on questions posed in [G2] are reported e.g. in Goodman and Pollack [GP1] and Edelsbrunner and Welzl [EW3].

The notion of a two-dimensional arrangement is easily generalized to three and higher dimensions. There, H is a finite set of ($(d-1)$-dimensional) hyperplanes in the d-dimensional Euclidean space E^d. The arrangement $A(H)$ consists of open convex d-dimensional polyhedra and various relatively open convex k-dimensional polyhedra bounding them, for $0 \leqq k \leqq d-1$.

Arrangements in E^d, for $d \geqq 3$, have received considerably less attention in the mathematical literature than arrangements in E^2. We refer to Grünbaum [G1, Chap. 18], and Grünbaum [G3] for surveys of d-dimensional arrangements. A reason for the lack

* Received by the editors September 15, 1983, and in revised form November 5, 1984.

† Institutes for Information Processing, Technical University of Graz, A-8010 Graz, Austria. The work of this author was supported by the Austrian Fonds zur Förderung der wissenschaftlichen Forschung.

‡ Department of Electrical Engineering and Computer Science, The Johns Hopkins University, Baltimore, Maryland 21218. This research was conducted while this author visited the Technical University of Graz. The work of this author was supported in part by the National Science Foundation under grant MSC-8117424.

§ Computer Science Department, Cornell University, Ithaca, New York 14583. This research was conducted while this author visited the Technical University of Graz.

of attention is probably the difficulty of visualizing arrangements even in E^3. Furthermore, Goodman and Pollack [GP2] demonstrated that a tool that is useful in E^2 (the "Levi enlargement lemma," see e.g. [G2]) does not generalize to E^3.

Much of the significance of arrangements in E^d is due to a dual correspondence to configurations of points in E^d. Many problems for sets of points are more conveniently solved for the corresponding arrangement. Examples for this thesis are the computational geometry problems discussed in § 4. Additional significance stems from the correspondence of arrangements in E^d to a special type of polytopes, called zonotopes, in E^{d+1} that can be defined as the Minkowski sum of segments (see e.g. [G3]).

The purpose of this work is to describe an optimal algorithm for constructing arrangements in E^d, for $d \geqq 2$. The optimality of the algorithm follows from the fact that the time required to construct an arrangement does not exceed asymptotically the space needed to store it. To be more specific: An arrangement of n hyperplanes in E^d, for $d \geqq 2$, is constructed in $O(n^d)$ time, and the arrangement actually needs space proportional to n^d unless it is highly degenerate; see e.g. Grünbaum [G1], [G3], Zaslavsky [Z], and Alexanderson and Wetzel [AW]. (For $d = 1$, the arrangement is essentially a sorted set of points on a line and cannot be constructed faster than in $O(n \log n)$ time.)

The optimality of the algorithm relies heavily on a nontrivial combinatorial fact that appears to be new (Thms. 2.7 and 2.8). This fact and other geometric preliminaries are demonstrated in § 2. Section 3 outlines the algorithm for constructing arrangements. In § 4, applications of the algorithm to λ-matrices, halfspatial range estimation, Voronoi diagrams, degeneracy tests, and minimum measure simplices are demonstrated. Finally, § 5 reviews the main results and lists some open problems.

2. Geometric fundamentals. This section discusses properties of arrangements of hyperplanes and configurations of points. It falls into three parts. Section 2.1 is devoted to a geometric transform that realizes the duality between arrangements and configurations; that will be exploited in the applications of § 4. Section 2.2 lists rather straightforward properties of arrangements that are relevant for the algorithmic part of this paper, §§ 3 and 4. Finally, § 2.3 presents a combinatorial result that is the key to the optimality of the algorithm outlined in § 3.

2.1. Arrangements and configurations. Let h be a nonvertical hyperplane in E^d, for $d \geqq 2$, that is, h is a $(d-1)$-dimensional hyperplane that intersects the dth coordinate axis in a unique point. Then the points on h with coordinates $x_1, \cdots, x_d$ satisfy a relation of the form $x_d = h_1x_1 + \cdots + h_{d-1}x_{d-1} + h_d$. Let $p = (p_1, \cdots, p_d)$ be a point in E^d. We say that p is *above*, *on*, and *below* h if p_d is greater than, equal to, and smaller than $h_1p_1 + \cdots + h_{d-1}p_{d-1} + h_d$. Let T be the geometric transform that maps the hyperplane h into the point $T(h) = (h_1, \cdots h_d)$ and the point p into the hyperplane $T(p)$ whose points $(x_1, \cdots x_d)$ satisfy $x_d = -p_1x_1 - \cdots - p_{d-1}x_{d-1} + p_d$. Where convenient, we will also use the natural extension of T to sets of hyperplanes and sets of points. One of the significant properties of T is that it preserves the relative positions of h and p.

Observation 2.1. If p is above, on, or below h then $T(h)$ is below, on, or above $T(p)$ respectively.

This observation establishes that T leads to dual and order preserving arrangements of hyperplanes if applied to configurations of points and vice versa. This duality of T has found applications to computing intersections of halfspaces (Brown [B]) and other tasks [EMPRWW], [EW2], [O].

We continue the development with an implication of Observation 2.1. First, some definitions are introduced. A set in E^d is called a *subspace of dimension* k (or a k-*flat*), for $0 \leqq k \leqq d$, if there are $d-k$ hyperplanes (and no fewer) such that the set is the intersection of these hyperplanes. Thus, E^d is a d-flat, each hyperplane is a $(d-1)$-flat, and, for convenience, the empty set is said to be a (-1)-flat. (The terms "points," "lines," and "planes" are used to designate 0-flats, 1-flats, and 2-flats in E^2 or E^3.)

Observation 2.2. Let S be a set of points in E^d, and let H be the set of all hyperplanes containing S. Then $T(H)$ is the intersection of all hyperplanes $T(p)$, for p in S.

For the reader particularly interested in the transform T, we note that a rather extensive list of similar implications restricted to E^2 can be found in Goodman and Pollack [GP3].

2.2 Properties of arrangements. Let $H = \{h_1, \cdots, h_n\}$ denote a set of n nonvertical hyperplanes in E^d, for $d \geqq 2$; since they are nonvertical, none contains a 1-flat parallel to the dth coordinate axis. Let h_i^+ and h_i^- denote the open halfspaces above and below h_i, for $1 \leqq i \leqq n$. The arrangement $A(H)$ consists of *faces* f with

$$(*) \qquad f = \bigcap_{1 \leqq i \leqq n} s_f(h_i),$$

where $s_f(h_i)$ is either h_i, h_i^+, or h_i^-. Thus, each face f can be assigned its *intersection word* $w(f) = w_1 \cdots w_n$, with $w_i = 0, +,$ or $-$ depending on whether $s_f(h_i)$ is h_i, h_i^+, or h_i^-. f is called a k-*face*, for $0 \leqq k \leqq d$, if the affine hull of f is a k-flat. (The *affine hull of* a set X is the collection of points of the form $\sum_{i=1}^m a_i x_i$ with a_i real, $\sum_{i=1}^m a_i = 1$, and x_i in X, for $0 \leqq i \leqq m$.) The terms "vertices," "edges," and "regions" are synonymous with 0-faces, 1-faces, and 2-faces for arrangements in E^2 and E^3. If f is a k-face, then $w(f)$ contains $d-k$ 0's, for $k = d-1, d$, and at least $d-k$, for $0 \leqq k \leqq d-2$. A k-face g and a $(k-1)$-face f are said to be *incident* if f is contained in the closure of g, for $1 \leqq k \leqq d$. Thus, $w(f)$ matches $w(g)$ up to a positive number of letters which are 0 in $w(f)$. Also g is termed a *superface of* f and f is called a *subface of* g. (To avoid confusion, we say that f is a subface of g (or g a superface of f) only if the dimensions of f and g differ by one. A synonym for subface is facet.)

$A(H)$ is called *simple* if the intersection of any k hyperplanes is a $(d-k)$-flat, for $1 \leqq k \leqq d+1$. Observe that this condition excludes parallelism between any two subspaces unless one contains the other. If $A(H)$ is simple, then f is a k-face if and only if $w(f)$ contains exactly $d-k$ 0's for $0 \leqq k \leqq d$. Thus, a face f is subface of a face g if and only if $w(f)$ and $w(g)$ differ in exactly one letter which is 0 in $w(f)$.

It will be necessary to have a system of notation to describe the relationship between the faces of an arrangement and a new hyperplane not part of the arrangement. The reader who is less interested in the algorithm for constructing arrangements may skip the introduction of this notation as well as Lemmas 2.3 and 2.4. Let $H = \{h_1, \cdots, h_n\}$ denote a set of n nonvertical hyperplanes in E^d, and let h denote a nonvertical hyperplane not in H. We assign to each face f of $A(H)$ one of the colours white, red, black, and grey, depending on its relationship to h:

f is *black* if h contains f,
f is *red* if h intersects f but does not contain f,
f is *grey* if h does not intersect f but intersects the closure of f, and
f is *white* if h does not intersect the closure of f.

The nonwhite faces in $A(H)$ are exactly those that are involved in updates if h is to be added to the arrangement. Using the introduced notation, we present a few basic properties of faces in arrangements.

LEMMA 2.3. *Let $A(H)$ be an arrangement in E^d, f a subface of face g in $A(H)$, and h a nonvertical hyperplane not in H.*

(i) *Only the pairs of colours indicated by 1's in Table 2-1 can occur.*

(ii) *If g has two black subfaces, then g is black.*

(iii) *g is red if and only if it is a 1-face that intersects h, it has a red subface, or it has grey subfaces on both sides of h.*

(iv) *If g is a 1-face, then only the circled 1's in Table 2.1 hold.*

TABLE 2.1
Matching colours.

f \ g	white	grey	red	black
white	①	①	①	0
grey	0	1	1	0
red	0	0	1	0
black	0	①	0	①

The proof of Lemma 2.3 is omitted as straightforward arguments using (*) and the definitions imply the assertion. The same is true for the proof of the next lemma. Let $A(H)$ and h be as above. We call a face f in $A(H \cup \{h\})$ *blue* if it is contained in h but was not present in $A(H)$.

LEMMA 2.4. *Let $A(H)$ and h be as in Lemma 2.3 and let g be a red k-face in $A(H)$, for some k with $1 \leqq k \leqq d$.*

(i) *$g \cap h^+$ and $g \cap h^-$ are k-faces in $A(H \cup \{h\})$, and $g \cap h$ is a blue $(k-1)$-face in $A(H \cup \{h\})$.*

(ii) *A $(k-1)$-face f of $A(H \cup \{h\})$ is a subface of $g \cap h^+$ if and only if either* (1) *f is a white or grey subface of g above h,* (2) *$f = f' \cap h^+$, for a red subface f' of g, or* (3) *$f = g \cap h$. The symmetric statements hold for $g \cap h^-$.*

(iii) *A $(k-2)$-face f'' is incident with $g \cap h$ if and only if either* (1) *$f'' = f' \cap h$, for a red subface f' of g, or* (2) *f'' is a black face in $A(H)$ incident with a grey subface of g.*

For the analysis of the algorithms to follow in § 3, the cardinalities of several sets of faces and incidences are of interest. Let $C_k(H)$ denote the number of k-faces of $A(H)$, for $0 \leqq k \leqq d$, and let $I_k(H)$ be the number of incidences between k-faces and $(k+1)$-faces of $A(H)$, for $0 \leqq k \leqq d-1$. We prove below that both $C_k(H)$ and $I_k(H)$ are in $O(n^d)$, if n denotes the number of hyperplanes in H.

LEMMA 2.5. *Let H be a set of n hyperplanes in E^d. Then*

(i) *$C_k(H) \leqq \sum_{i=d-k}^{d} \binom{i}{d-k}\binom{n}{i}$, for $0 \leqq k \leqq d$, and*

(ii) *$I_k(H) \leqq 2(d-k) \sum_{i=d-k}^{d} \binom{i}{d-k}\binom{n}{i}$, for $0 \leqq k \leqq d-1$.*

Equality occurs if $A(H)$ is simple.

Proof. Part (i) of the assertion follows from the exact formula for simple arrangements, e.g. given in Zaslavsky [Z] or Alexanderson and Wetzel [AW], and the fact that the number of k-faces is maximized when $A(H)$ is simple.

Observe now that a k-face f that is contained in i hyperplanes (so $i \geqq d-k$) has at most $2\binom{i}{d-k-1}$ incident $(k+1)$-faces. This follows from the fact that the i hyperplanes define at most $\binom{i}{d-k-1}$ $(k+1)$-flats each containing two superfaces of f. But as f represents $\binom{i}{d-k}$ k-faces (the maximal number of k-faces created by i hyperplanes), there are at most $2(d-k)$ $(k+1)$-faces for each k-face that f represents. The maximum $2(d-k)$ is achieved if $i = d-k$, which implies that $I_k(H)$ is maximal if $A(H)$ is simple. This completes the argument.

Let now v be a 0-face in an arrangement $A(H)$. Then the number of 1-faces incident with v is called the *degree* deg (v) *of* v.

LEMMA 2.6. *Let H be a set of n hyperplanes in E^d and let V denote the set of 0-faces contained in a 1-flat of $A(H)$. Then $\sum_{v \in V} \deg(v) = O(n^{d-1})$.*

Proof. There are at most $\binom{n}{d-1}$ 1-flats in $A(H)$. One of these 1-flats can intersect each of the other 1-flats at most once. Hence, $\sum_{v \in V} \deg(v) < 2\binom{n}{d-1} = O(n^{d-1})$.

2.3. Combinatorial results. The combinatorial results demonstrated in this subsection are crucial for the algorithms in §§ 3 and 4. They appear to be new and are of independent interest. We start with the introduction of some notation.

Let $H = \{h_0, h_1, \cdots, h_n\}$ denote a set of $n+1$ nonvertical hyperplanes in E^d. For convenience, h_0 is assumed to coincide with the hyperplane spanned by the first $d-1$ coordinate axes. In § 3, h_0 will play the role of the new hyperplane added to the existing arrangement formed by $h_1, \cdots, h_n$. A d-face g in $A(H)$ is said to be *active* (*with respect to* h_0) if g is above h_0 and the closure of g intersects h_0. Note that g is active if and only if it is contained in a grey or red face (with respect to h_0) in $A(H - \{h_0\})$. Extending the notion of an incidence, a k-face f, for $0 \leqq k \leqq d-1$, is said to *bound* d-face g if f is contained in the closure of g. We call the pair (f, g) a k-*border* (*of* g); often the d-face g will not be explicitly mentioned when it is irrelevant or clear from the context. Where convenient, a flat is said to contain (f, g) if it contains f, and also f is said to contain (f, g). The intersection of all open halfspaces containing g that are defined by hyperplanes in H containing f is termed the *cone of* (f, g). In a two-dimensional arrangement, the cone of a vertex is a wedge with apex at the vertex, and the cone of an edge is a halfplane with the edge on its boundary. The k-*degree* $\deg_k(g)$ of g is now defined as the number of k-faces that bound g, for $0 \leqq k \leqq d-1$. The sum of the k-degrees of all active d-faces in $A(H)$ is denoted by $S_k^d(H, h_0)$. These definitions are illustrated in Fig. 2.1, which shows the regions active with respect to the horizontal line h_0. In the arrangement depicted, $S_0^2(H, h_0) = 17$ and $S_1^2(H, h_0) = 19$.

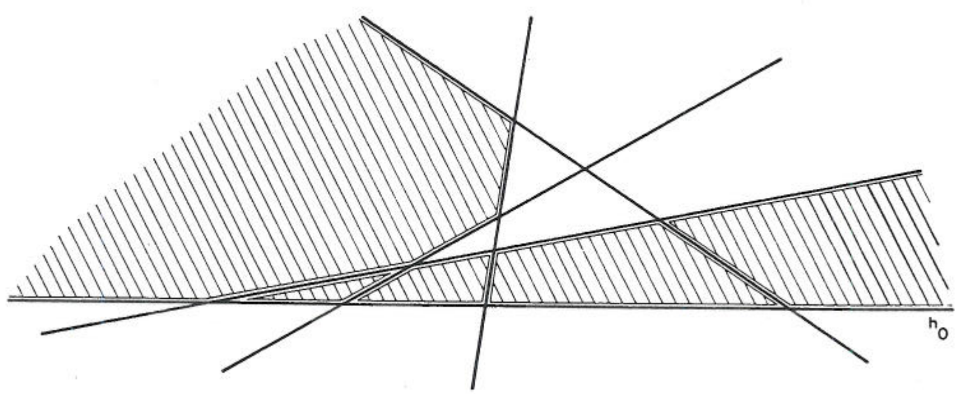


FIG. 2.1. *Regions active with respect to* h_0.

We call a k-border (f, g) *active* if g is active. So $S_k^d(H, h_0)$ counts the number of active k-borders (rather than the number of k-faces that bound active d-faces). For $0 \leqq k \leqq d-1$, define $S_k^d(n) = \max\{S_k^d(H, h_0)$: H a set of $n+1$ nonvertical hyperplanes in E^d and h_0 in $H\}$. It is easy to see that there are simple arrangements $A(H)$ of $n+1$ hyperplanes that achieve $S_k^d(n)$. Thus, for deriving upper bounds it suffices to examine simple arrangements.

We prove below that $S_k^d(n)$ is in $O(n^{d-1})$, which permits the insertion of h_0 into $A(H - \{h_0\})$ in $O(n^{d-1})$ time. The algorithm for performing the insertion will be described in § 3. Since the result is easiest to understand in E^2, this case is considered first and generalized to higher dimensions later. In both cases, the main technique is to sweep the arrangement with a unidirected hyperplane initially coincident with h_0. During the sweep, faces in the hyperplane are classified into three states that change over time. The rules obeyed by these changes are finally exploited to infer bounds on $S_k^d(n)$.

THEOREM 2.7. *$S_0^2(n)=5n-3$ and $S_1^2(n)=5n-1$.*

Proof. Let $H=\{h_0, \cdots, h_n\}$ denote a set of $n+1$ nonvertical lines in E^2 such that h_0 coincides with the x_1-axis and $A(H)$ is simple. We first show that $5n-1$ is an upper bound for $S_1^2(n)$ and demonstrate that it is tight. Then we argue that $S_0^2(n)=S_1^2(n)-2$.

Observe first that the number of active 1-borders contained in h_0 is $n+1$. It remains to show that $4n-2$ is the maximum number of active 1-borders that are not contained in h_0. To this end, we perform a continuous upwards sweep with a horizontal line h. Initially $h=h_0$, and at each point in time h intersects $A(H)$ in a one-dimensional arrangement $A_h(H)$. Let p_i denote the intersection of h with h_i, and let v_i^L and v_i^R denote the 0-borders on p_i in $A_h(H)$. The superscript L indicates that $v_i^L=(p_i, e_L)$, for e_L the segment in $A_h(H)$ to the left of p_i; the superscript R indicates the symmetric situation to the right. Consult Fig. 2.2 for an illustration.

At each point in time, a 0-border v in $A_h(H)$ is in one of three *states*. Let e denote the 1-border in $A(H)$ such that the cone of v in $A_h(H)$ is the intersection of h and the cone of e in $A(H)$. The cone of v in $A_h(H)$ is a horizontal ray within h with endpoint v; the cone of e in $A(H)$ is a halfplane with e on its boundary. Thus e must contain v for the intersection of h and the cone of e to be the cone of v. Define the state of v as follows:

v is *alive* or *live* if e is active.

v is *dead* if there are two lines h_i and h_j in $H-\{h_0\}$ such that the intersection of h_i and h_j is between h_0 and h, e is contained in h_i or h_j, and the cone of e contains the wedge between h_i and h_j that lies entirely above h_0. (Note that death is irreversible.)

Otherwise, v is *sleeping*.

Intuitively, v is sleeping when it traverses a "dead sector" of $A(H)$ and still has the chance to leave it and become alive. In Fig. 2.2, v_3^L and v_1^R are alive, v_3^R, v_2^L, v_2^R, v_4^R, and v_1^L are dead, and v_4^L is sleeping. In the argument below, we watch the states of 0-borders changing from live to dead which allows us to infer results on the number of active 1-borders in $A(H)$. During the sweep of h, the states of the 0-borders in

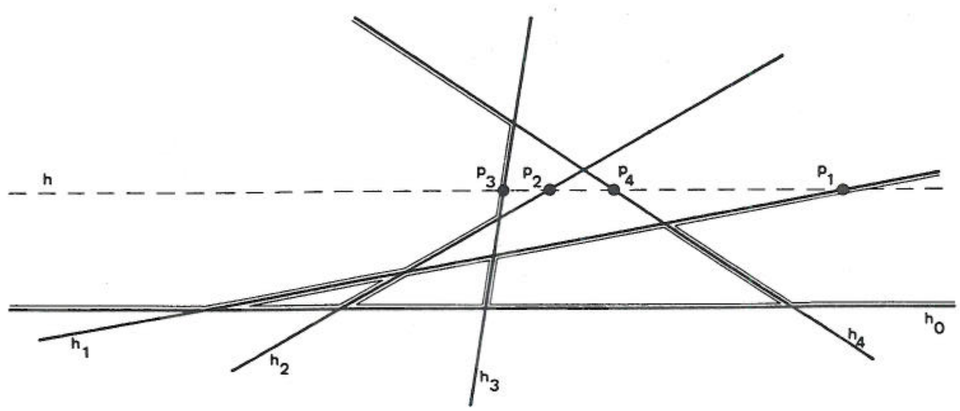


FIG. 2.2. *The bottom-up sweep.*

$A_h(H)$ change only when two points p_i and p_j switch. The following rules can be observed:

R1. All $2n$ 0-borders are alive in the beginning of the sweep.

R2. At least two 0-borders are alive when h has passed all vertices of $A(H)$.

Let now h pass the intersection of h_i and h_j such that p_i is to the left of p_j before they switch. The states of v_i^L and v_i^L after the switch depend only on their states before the switch, and similarly for v_i^R and v_j^R. As the rules for the states of v_i^R and v_j^R are strictly symmetric, we consider only those for v_i^L and v_j^L. The rules observed in each of five cases follow immediately from the definitions of the states "alive," "sleeping," and "dead." Table 2.2 indicates the possible states before and after the switch.

TABLE 2.2

Rule	Before v_i^L	Before v_i^L	After v_i^L	After v_i^L
R3	alive	alive	dead	alive
R4	dead	alive	dead	sleeping
R5	alive	sleeping	dead	alive
R6	dead	sleeping	dead	sleeping
	dead	dead	dead	dead
R7	sleeping	sleeping	dead	sleeping

The two cases where v_i^L is alive or sleeping and v_j^L is dead have not been enumerated as they cannot occur. Consult Fig. 2.3 for an illustration of rules R3–R7. Living 0-borders are indicated by solid lines, sleeping 0-borders by dashed lines, and dead 0-borders by dotted lines.

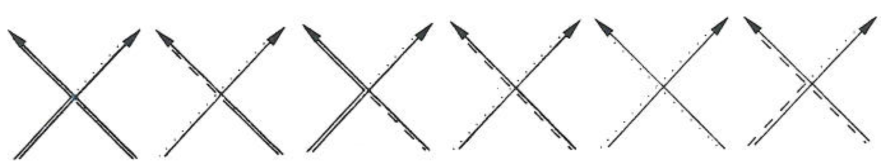


FIG. 2.3. *Illustration of rules* R3–R7.

These rules are exploited for deriving an upper bound on the number of active 1-borders in $A(H)$ that are not contained in h_0. To this end, four counters are used: ACT, to designate the current number of active 1-borders in $A(H)$ that are entirely below h and not contained in h_0, and A, S, and D, to designate the current number of 0-borders in $A_h(H)$ that are alive, sleeping, and dead, respectively. Initially, $ACT=0$, $A=2n$, $S=0$, and $D=0$ by R1; ultimately, $A \geqq 2$ by R2. Application of the rules R3–R7 causes the following changes to ACT, A, S, and D:

R3: $ACT=ACT+2,\ A=A-1,\ S=S,\ D=D+1.$

R4: $ACT=ACT+1,\ A=A-1,\ S=S+1,\ D=D.$

R5: $ACT=ACT+1,\ A=A,\ S=S-1,\ D=D+1.$

R6: $ACT=ACT,\ A=A,\ S=S,\ D=D.$

R7: $ACT=ACT,\ A=A,\ S=S-1,\ D=D+1.$

The transitions flow only from alive to dead (R3) or sleeping (R4), and from sleeping to dead (R5 and R7). Both the alive→dead and the alive→sleeping→dead paths give rise to at most two active 1-borders in $A(H)$. Since $A=2n$ initially, $4n$ active borders could be generated. But $A \geqq 2$ after the complete sweep, and each of the remaining live 0-borders is contained in an unbounded active 1-border of $A(H)$. Therefore, $4n-2$ is an upper bound on the number of active 1-borders in $A(H)$ that are not contained in h_0. This shows that $(4n-2)+(n+1)=5n-1$ is an upper bound on $S_1^2(n)$. The arrangement shown in Fig. 2.1 actually realizes equality for $n=4$ and can be generalized to arbitrary n in an obvious way. This shows $S_1^2(n)=5n-1$.

To establish $S_0^2(n) \leqq S_1^2(n)-2$, observe that $\deg_1(r)=\deg_0(r)$ for each bounded region r in $A(H)$, and that $\deg_1(r)=\deg_0(r)+1$ for each unbounded region r in $A(H)$. At least two of the active regions are unbounded, so $S_0^2(n) \leqq S_1^2(n)-2$. In fact, the arrangement (shown in Fig. 2.1) that realizes $S_1^2(n)$ has exactly two unbounded active regions, which implies $S_0^2(n)=S_1^2(n)-2=5n-3$. This completes the proof.

We recently learned that Theorem 2.7 was independently discovered by Chazelle, Guibas, and Lee [CGL]. The proof given in [CGL] is considerably simpler than ours; however, it does not seem to generalize to three and higher dimensions. In fact, the motivation for presenting the proof given above (out of a number of possible proofs) is its generalizability.

It is worth mentioning that the assertion of Theorem 2.7 also holds for families of pseudo-lines. (A *pseudo-line* is an unbounded and connected curve in E^2 such that any two in a given arrangement intersect in exactly one point and cross there.) Consult Grünbaum [G2] for an account of this natural generalization of lines. The proof of Theorem 2.7 for arrangements of pseudo-lines is the same as that for lines except that the sweep is performed with a pseudo-line.

Next, the analogue of Theorem 2.7 in $d \geqq 3$ dimensions will be established. The essential idea in the proof is the same as in the proof of Theorem 2.7: the halfspace above h_0 is swept by a hyperplane h parallel to h_0. The switches of pairs of points are now replaced by switches of d-tuples of $(d-2)$-flats in h, that is, the $(d-1)$-dimensional bounded simplex defined by d $(d-2)$-flats collapses and reappears in mirrored shape. Consult Fig. 2.4, which depicts a switch when h is a plane.

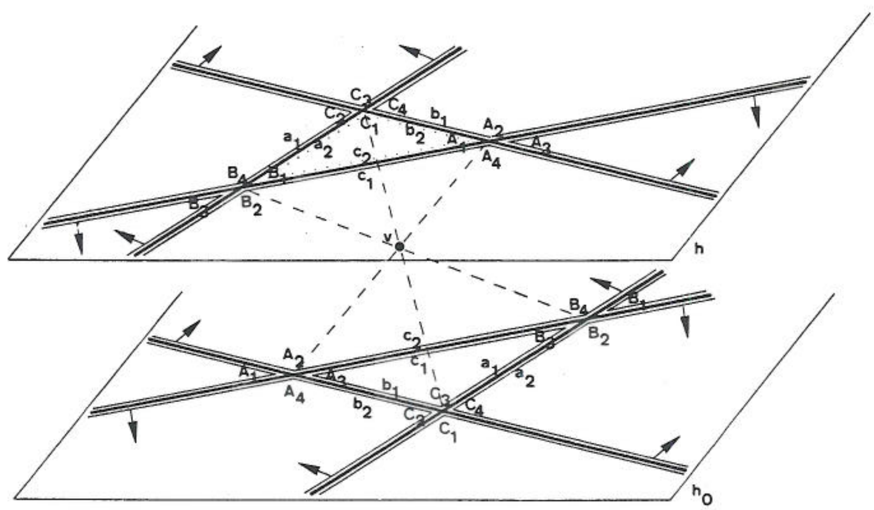


FIG. 2.4. *Switch in* E^3.

THEOREM 2.8. $S_k^d(n)=\theta(n^{d-1})$, *for* $d \geqq 3$ *and* $0 \leqq k \leqq d-1$.

Proof. Let $H=\{h_0, \cdots, h_n\}$ denote a set of $n+1$ nonvertical hyperplanes in E^d such that h_0 coincides with the hyperplane spanned by the first $d-1$ coordinate axes and such that $A(H)$ is simple. The intersection of $A(H-\{h_0\})$ and h_0 is isomorphic to a simple arrangement of n hyperplanes in E^{d-1}. There are $\theta(n^{d-1})$ k-faces ($0 \leqq k \leqq d-1$) in this arrangement (see Lemma 2.5) and these k-faces bound d-faces of $A(H)$ that are active with respect to h_0. Thus, $S_k^d(n) = \theta(n^{d-1})$, for $0 \leqq k \leqq d-1$, which establishes the asymptotic lower bound of the assertion.

For a proof of the upper bound, we perform an upwards sweep with a hyperplane h parallel to h_0, that is, h sweeps in the direction of the x_d-axis. Initially, $h = h_0$, and at each point in time, h intersects $A(H)$ in a $(d-1)$-dimensional arrangement $A_h(H)$. We define a relation R between the faces of $A(H)$ such that (g, g') in R, for $(k+1)$-faces g and g' and $0 \leqq k \leqq d-1$, if

(i) there is a $(k+1)$-flat p (defined by $d-k-1$ hyperplanes in H) that contains g and g',

(ii) g and g' share a bounding 0-face, and

(iii) there is no hyperplane parallel to h_0 that intersects both g and g'.

We call two faces g and g' *equivalent* if they are in the same equivalence class induced by the transitive closure of R. So all 1-faces of a 1-flat are equivalent, and h intersects exactly one face of each equivalence class unless it contains a 0-face of $A(H)$. The notion of "equivalence" can be extended to borders of $A(H)$ such that two $(k+1)$-borders c and c' are *equivalent* if

(i) the two $(k+1)$-faces g and g' that contain c and c' are equivalent, and

(ii) the cones of c and c' are the same.

Let now b and b' be two k-borders in $A_h(H)$, for $0 \leqq k \leqq d-2$, at different points in time. Let c and c' be the $(k+1)$-borders in $A(H)$ such that the cone of b (and b') is the intersection of h and the cone of c (and c'). We *identify* b and b' if c and c' are equivalent. Consult Fig. 2.4 for an illustration of this identification of borders which is natural when the sweep of h is considered as a process in time.

At each point in time, a k-border b ($0 \leqq k \leqq d-2$) in $A_h(H)$ is in one of three states. Let c denote the $(k+1)$-border in $A(H)$ such that the cone of b (in $A_h(H)$) is the intersection of h and the cone of c (in $A(H)$. Then the state of b is defined as follows:

b is *alive* if c is active.

b is *dead* if there are d hyperplanes in $H - \{h_0\}$ such that their common 0-face lies between h_0 and h, c is contained in the intersection of $d-k-1$ of these hyperplanes, and the cone of c contains the unique sector defined by the d hyperplanes that lies above h_0. (Death is thus irreversible.)

Otherwise, b is *sleeping*.

During the sweep of h, the states of the k-borders in $A_h(H)$ change only when d $(d-2)$-flats in $A_h(H)$ switch (see Fig. 2.4 for a switch of three lines (1-flats) in E^3). The rules for the changes of the states that are observed can be related and reduced to the rules described in the proof of Theorem 2.7: All $\Theta(n^{d-1})$ k-borders in $A_h(H)$, for $0 \leqq k \leqq d-2$, are alive in the beginning of the sweep. Let now h pass the common 0-face v of d hyperplanes $h_{i1}, \cdots, h_{id}$ in H (see Fig. 2.4). There are certain k-borders in $A_h(H)$, for $0 \leqq k \leqq d-2$, that collapse into v as h comes closer to v. We call such a k-border *collapsing*. A collapsing i-border B is *paired* with a collapsing $(d-i-2)$-border b, for $0 \leqq i \leqq d-2$, if the following holds:

(i) there is no hyperplane in H that contains B and b (before they collapse), and

(ii) there is a proper halfspace in h that contains the cone of B and the cone of b.

In the arrangement shown in Fig. 2.4, the following 0-borders and 1-borders are paired: (A_1, a_1), (A_3, a_2), (B_1, b_1), (B_3, b_2), (C_1, c_1), and (C_3, c_2). Note that there are collapsing i-borders in $A_h(H)$ that are not paired. However, each (collapsing) i-border, for $0 \leqq i \leqq d-2$, of the collapsing $(d-1)$-simplex s in $A_h(H)$ is paired with a $(d-i-2)$-border whose cone does not contain s. Let f be a $(d-1)$-face in $A_h(H)$ that shares a bounding i-face with s, with i maximal and $0 \leqq i \leqq d-2$. Then the i-border b of f contained in that common i-face is paired with some $(d-i-2)$-border B of s. Paired borders play the same role in this proof as v_i^L and v_j^L have done in the proof of Theorem 2.7. Let B and b be two paired borders such that the cone of B contains s and the cone of b contains f (before s collapses). After the collapse, B becomes a border of f, and b becomes a border of s. Hence, b dies in any case, and the new state of B depends on the old states of B and b. The changes of the states of B and b follow exactly the rules R3–R7 (see Table 2.2 and Fig. 2.3). For instance, if b was alive before the collapse then B stays or becomes alive as it belongs now to a $(d-1)$-face (in $A_h(H)$) that is the intersection of h and an active d-face in $A(H)$. If b was not alive before the collapse, then it cannot change the state of B unless B is alive in which case B falls asleep.

Let us now exploit this property for deriving an upper bound on the number of active k-borders in $A(H)$, for $0 \leqq k \leqq d-1$. Note first that an active k-border is created during the sweep of h (that is, the upper end of the k-face that contains the active k-border is passed by h) only when a switch occurs in $A_h(H)$ such that some of the collapsing borders are alive. Furthermore, each collapse creates at most a constant number of active k-borders.

Let g be the d-face in $A(H)$ such that the collapsing $(d-1)$-complex s in $A_h(H)$ is the intersection of g and h. We distinguish two cases: First assume that g is not active. Then there is another $(d-1)$-face f in $A_h(H)$ with the following properties:

(i) Let g' be the d-face in $A(H)$ such that $f = g' \cap h$. Then g' is active and f shares a bounding i-face, (i maximal and $0 \leqq i \leqq d-2$) with s.

(ii) There is a (living) i-border b of f contained in that i-face that is paired with a (nonliving) $(d-i-2)$-border of s.

By rule R3 or R5, b dies. However, this implies that this case can occur only $O(n^{d-1})$ times as each occurrence increases the number of dead borders in $A_h(H)$ by at least one. Second, assume that g is active. As h is passing the topmost point of g (since s is collapsing) and there are only $O(n^{d-1})$ active d-faces in $A(H)$, this case can also occur only $O(n^{d-1})$ times. This completes the proof.

In order to keep the proof of Theorem 2.8 short, we have refrained from deriving more accurate than only asymptotic upper bounds. Nevertheless, we conjecture that the applied proof technique is well suited for calculating more accurate bounds as well. It is worthwhile to note here that Theorem 2.8 also holds for arrangements of pseudo-hyperplanes appropriately defined (see e.g. [GP2]). In this more general setting, the proof of Theorem 2.8 can be adapted by performing a sweep with a pseudo-hyperplane.

There is an interesting consequence of Theorems 2.7 and 2.8:

COROLLARY 2.9. *Let H be a set of n hyperplanes in E^d, let f be a d-face in $A(H)$, and let* $\deg_k(f)$ *denote the number of k-faces* $(0 \leqq k \leqq d-1)$ *bounding f. Then the sum of the products* $\deg_{d-1}(f)\deg_k(f)$, *for all d-faces f in $A(H)$, is in $O(n^d)$.*

Proof. The sum of $S_k^d(H, h)$, for all h in H, is in $O(n^d)$ by Theorems 2.7 and 2.8. Turning $A(H)$ upside-down and repeating the evaluation of $S_k^d(H, h)$ gives again $O(n^d)$. But now, each k-face in $A(H)$ has been counted $\deg_{d-1}(f)$ times for each d-face f that is bounded by the k-face.

3. Constructing arrangements. This section describes algorithms for constructing arrangements in Euclidean spaces. For expository reasons, the algorithm working in E^2 is presented first and the general algorithm later. The next subsection presents the overall structure of the algorithm and the data structure used for representing arrangements.

3.1. The overall structure. The construction of an arrangement proceeds incrementally, that is, the arrangement is built by adding hyperplanes one at a time to an already existing arrangement. The order in which the hyperplanes are added is irrelevant. To avoid tedious special cases that occur for sets of hyperplanes whose normal-vectors do not span E^d, we start with a carefully chosen subcollection of the given set.

Let H denote a set of n hyperplanes $h_1, \cdots, h_n$ in E^d and define $H_i = \{h_1, \cdots, h_i\}$, for $1 \leqq i \leqq n$. $D(H)$ denotes the data structure to be described that represents the arrangement $A(H)$. We assume that the normal-vectors of the hyperplanes in H span E^d. Otherwise, each hyperplane is intersected with the k-dimensional subspace of E^d (with $k < d$) spanned by the normal-vectors, and the resulting k-dimensional arrangement, which captures the essential information of the d-dimensional arrangement, is constructed. We will see that this preprocessing phase only requires $O(n)$ time if d is considered a constant. Now, the overall structure of the algorithm can be described as follows:

Without loss of generality, assume the normal-vectors of $h_1, \cdots, h_d$ span E^d.
Construct $D(H_d)$.
For i running from $d+1$ to n, construct $D(H_i)$ from $D(H_{i-1})$ by insertion of h_i.
Finally, $D(H) = D(H_n)$.

Some comments are in order to clarify the preprocessing phase that computes the space spanned by the normal-vectors of the hyperplanes. It is readily seen that this action can be performed in $O(n)$ time by successively testing whether the normal-vector of the current hyperplane is contained in the subspace spanned so far. Let k denote the dimension of the spanned subspace. Then this strategy can also be used to identify k hyperplanes whose normal-vectors span the subspace. Without loss of generality let this subspace be spanned by the last k coordinate axes. Then each hyperplane is replaced by its intersection with this subspace and the arrangement of the resulting hyperplanes in k dimensions is constructed. The method for constructing $D(H_d)$ (or $D(H_k)$) is demonstrated in the next subsection.

3.2. The representation of arrangements. For storing an arrangement $A(H)$, we basically use the incidence lattice of $A(H)$ defined for polytopes in Gruenbaum [G1]. By convention, $A(H)$ is called a $(d+1)$-face and the empty set is called a (-1)-face of $A(H)$. Also $A(H)$ is said to be incident with all its d-faces, and the empty set is said to be incident with all 0-faces. The incidence lattice of $A(H)$ represents each k-face by a node, for $-1 \leqq k \leqq d+1$, and contains connections between nodes of incident faces. Where convenient in the subsequent discussion, no distinction will be made between a node and its corresponding face. See Fig. 3.1 for an arrangement of two lines in E^2 and its incidence lattice.

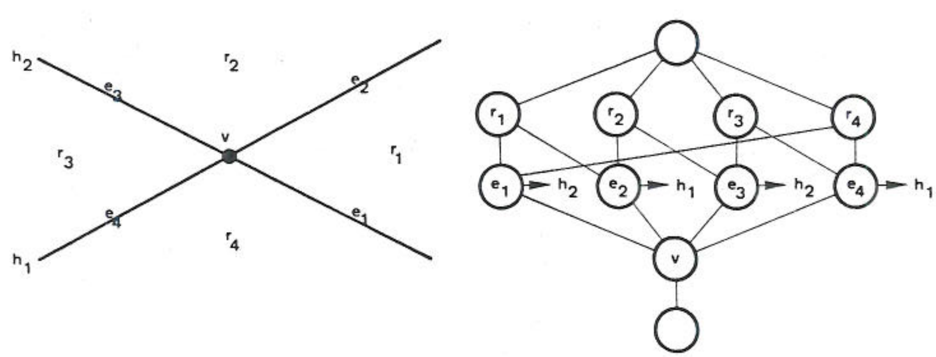


FIG. 3.1. *Arrangement and incidence lattice.*

A point $p(f)$ in f is also associated with each k-face f, for $0 \leqq k \leqq d$. If f is a 0-face, then $p(f) = f$. To be precise, we use the following definition for $p(f)$:

(1) If f is an unbounded 1-face then $p(f)$ is the unique point of f with distance 1 from the only incident vertex.

(2) If f is a bounded 1-face or f is a k-face with $k \geqq 2$, then $p(f) = (\sum_{i=1}^{m} p(f_i))/m$, for $f_1, \cdots, f_m$ the subfaces of f.

In addition, each $(d-1)$-face is associated with its supporting hyperplane. The data structure $D(H)$ is thus the incidence lattice of $A(H)$ augmented with some auxiliary information as described. Without confusion, we will use $A(H)$ and $D(H)$ interchangeably. The auxiliary information used above is not the only choice: it could easily be replaced by equivalent information, such as lists of supporting hyperplanes. The preferred structure depends on the particular application for which the arrangement is being used, and indeed we will further augment the structure when discussing specific problems in § 4.

Before proceeding to the algorithms for building $D(H)$, we discuss the construction of $D(H_d)$ as required in the initial step of the algorithm. We make use of the special structure of $D(H_d)$, which results from the assumption that the normal-vectors of H_d span E^d. Recall that $C_k(H)$ denotes the number of k-faces in $A(H)$, for $0 \leqq k \leqq d$. By definition $C_{-1}(H_d) = C_{d+1}(H_d) = 1$.

LEMMA 3.1. *$C_k(H_d) = 2^k \binom{d}{k}$, for $0 \leqq k \leqq d$.*

The assertion follows from the duality of $A(H_d)$ and the d-dimensional cube in conjunction with Theorem 4.4.2 in [G1]. By "duality" we mean that the incidence lattices of $A(H_d)$ and the cube are isomorphic. The assertion can also be verified directly from the observation that the subarrangement of $A(H_d)$ in one of the hyperplanes is isomorphic to an arrangement defined by $d-1$ hyperplanes in E^{d-1} whose normal-vectors span E^{d-1}. Both facts can be exploited to find the connections to be established between the nodes, thus determining the incidence lattice of $A(H_d)$. The following gives a simple and constructive description of the incidence lattice of $A(H_d)$. We will use the intersection words of the faces as defined in § 2.2.

$A(H_d)$ is a simple arrangement, so for each word w in $\{0, +, -\}^d$ there is a face f with $w(f) = w$. If there are $d-k$ 0's in w then f is a k-face. For a proof of this observe that there are exactly $2^k \binom{d}{k}$ words of length d which contain $d-k$ 0's. But Lemma 3.1 tells us that there are also exactly that many k-faces in $A(H_d)$.

Thus, the incidence lattice of $A(H_d)$ can be set up by creating a node for each word in $\{0, +, -\}^d$. In addition, two nodes representing $A(H_d)$ and the empty set are created. Figure 3.2 shows the incidence lattice of $A(H_2)$ with the intersection word of each node marked.

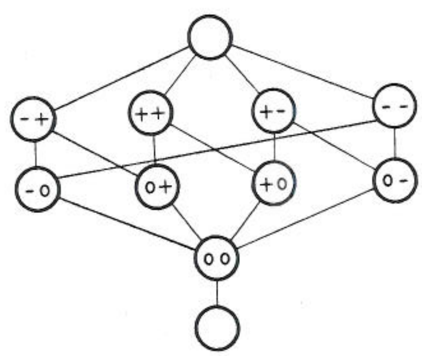


FIG. 3.2. *Incidence lattice of $A(H_2)$.*

Two nodes are connected if their words differ in only one letter, which is 0 in one word. In addition, all d-faces are connected to the $(d+1)$-face and the 0-face is connected to the (-1)-face. It is easy to then augment $D(H_d)$ with the necessary auxiliary information.

3.3. Constructing arrangements of lines. This section describes an algorithm that inserts a line h into an arrangement $A(H)$ of a set H of n lines in E^2. The assumptions on $H \cup \{h\}$ are that no line is vertical and that $A(H \cup \{h\})$ is simple. The strategy for inserting h into $A(H)$ is presented on a rather intuitive level. The full details, including the handling of degenerate cases, can be derived from the general strategy presented in § 3.4. The main purpose of this section is to provide intuition for the explanations in § 3.4.

The insertion of h into $A(H)$ is accomplished in three steps:

Step 1. An edge of $A(H)$ that intersects h is identified.
Step 2. All edges and regions that intersect h are marked red.
Step 3. The marked edges and regions are updated and the new vertices and edges contained in h are integrated.

The three steps are now explained in more detail.

Step 1. To identify an edge e_0 that intersects h, the edges on an arbitrary line of H are visited and tested. The process starts at an arbitrary edge e and proceeds edge by edge closer to h until e_0 is reached.

Step 2. Starting with e_0, all edges and regions that intersect h are marked and remembered in separate storage. To initialize the process, e_0 is marked red and remembered. In addition, the incident regions of e_0 are also marked red, remembered, and put into an empty queue Q. While Q is not empty, the first region r is deleted from Q, and its incident white edges are tested for intersection with h. Those that intersect h are marked red and remembered. Also, if they have an incident region that is not yet marked red then it is marked red and put into Q to await its computation.

Step 3. This step concentrates on splitting each red edge and each red region into two, establishing their new incidences, and integrating the new vertices and edges contained in h into the data structure.

Each red edge e is replaced by two new red edges e_a and e_b representing the parts of e above and below h. Next, the incidences of e_a and e_b are established in the appropriate way. That is: (1) Both are connected to the incident regions of e, and (2) e_a (e_b) is connected to the vertex of e above (below) h. In addition, a blue node v is created that represents $e \cap h$ and is thus connected to e_a, e_b, and the (-1)-face. See Fig. 3.3, where the red nodes are shaded and the blue node cross-hatched.

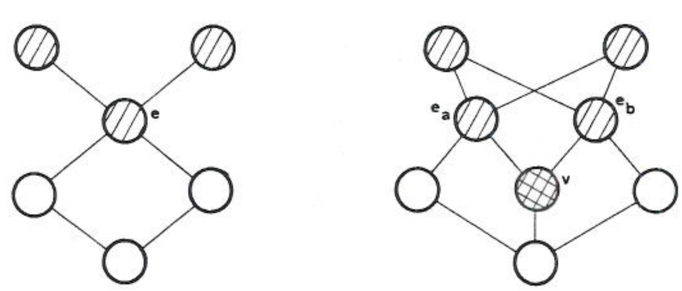


FIG. 3.3. *Updating a red edge e.*

Additional adjustment of incidences is carried out when the red regions are updated. Instead of discussing the update of a red region (which is similar to that of a red edge), we refer to Fig. 3.4, which depicts the actions to be taken in order to split a red region r. Finally, all marked vertices, edges, and regions are unmarked by coloring them white.

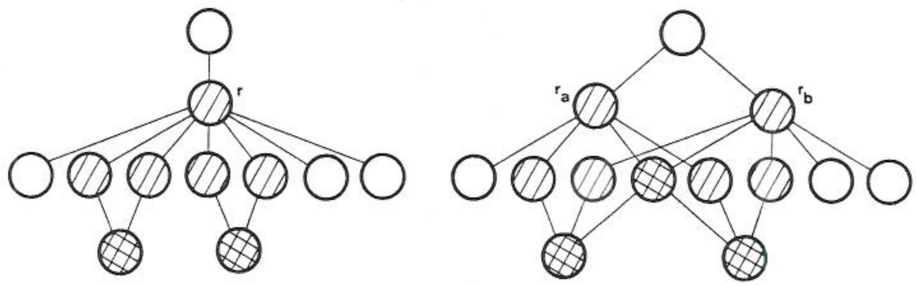


FIG. 3.4. *Updating a red region r.*

We only mention that $O(n)$ time suffices to insert h into $A(H)$; see also Lemma 3.2. Thus, $O(n^2)$ time suffices to set up the arrangement for n lines in E^2; see also Theorem 3.3.

3.4. Constructing arrangements in d dimensions. In this section, the insertion of a hyperplane in E^d into an arrangement $A(H)$ of a set H of n hyperplanes is discussed. It is assumed that the normal-vectors of the hyperplanes in H span E^d, as discussed in § 3.1. No restriction on the position of the hyperplanes is assumed except for the exclusion of multiple hyperplanes and of vertical hyperplanes; in particular, the arrangement is not assumed to be simple.

The algorithm that inserts h into $A(H)$ proceeds in three steps:

Step 1. A 1-face in $A(H)$ is identified whose closure intersects h.
Step 2. All faces in $A(H)$ whose closures intersect h are marked black, red, or grey.
Step 3. The marked faces are updated and the new faces contained in h are integrated.

The remainder of this section describes in detail the actions taken in each step and finally gives an analysis of the time and space requirements.

We apply the following strategy to determine a 1-face e_0 of $A(H)$ whose closure intersects h:

Step 1.1. Let w be an arbitrary 0-face and e an incident 1-face not parallel to h. If the closure of e intersects h, then e is the 1-face e_0 required. Otherwise, set v to w if e has only one incident 0-face, and set v to the incident 0-face nearer to h if there are two.

Step 1.2. Let e' (distinct from e) denote the superface of v collinear with e. If the closure of e' intersects h then e' is the 1-face e_0. Otherwise, let v' be the subface of e' distinct from v. Step 1.2 is now repeated with e and v replaced with e' and v'.

Starting with the 1-face e_0, all k-faces in $A(H)$, for $0 \leqq k \leqq d$, whose closures intersect h are marked and remembered in queue Q_k. In addition to the queues $Q_0, \cdots, Q_d$, we use a queue Q to store temporarily those 2-faces that are awaiting examination.

Step 2.1. The 1-face e_0 is marked red if it intersects h, and grey, otherwise. (Note that due to the method of choosing e_0, e_0 is not contained in h.) In addition, e_0 is put into Q_1. If e_0 is red then all incident 2-faces are marked red and put into Q and Q_2 (Table 2.1, third row and column). If e_0 is grey then its incident 0-face contained in h is marked black and put into Q_0. The incident 2-faces are marked grey, for the moment, and put into Q and Q_2 (Table 2.1, second row and column).

Step 2.2. While Q is not empty, the first 2-face r is deleted from Q. All incident white 1-faces e are tested for intersection with h.

Case 2.2.1. If e is contained in h, then e is marked black and put into Q_1. The white subfaces of e are marked black and put into Q_0. The white superfaces of e are marked grey, for the moment, and put into Q and Q_2 (Table 2.1, fourth row and column).

Case 2.2.2. If e intersects h but is not contained in h, then e is marked red and put into Q_1. All white superfaces are marked red and put into Q and Q_2 (Table 2.1, third row and column).

Case 2.2.3. If e does not intersect h but its closure does then e is marked grey and put into Q_1. The white subface contained in h (if it exists) is marked black and put into Q_0. The white superfaces are marked grey, for the moment, and put into Q and Q_2 (Table 2.1, second row and column).

Case 2.2.4. No action is taken if the closure of e does not intersect h (Table 2.1, first column).

Step 2.3. All grey 2-faces in Q_2 that have either a red subface or grey subfaces above and below h are marked red (Lemma 2.3(iii)). All grey 2-faces which have at least two black subfaces are marked black (Lemma 2.3(ii)).

Step 2.4. For k running from 3 to d and for all faces f in Q_{k-1}, the following actions are taken for the white superfaces of f:

Case 2.4.1. If f is red then they are marked red (Table 2.1, third row).

Case 2.4.2. If f is black, then they are marked black if they have at least two black subfaces, and grey otherwise (Lemma 2.3(ii)).

Case 2.4.3. If f is grey, then they are marked red if they have also red subfaces or grey subfaces above and below h, and grey otherwise (Lemma 2.3(iii)).

In any of the three cases, the examined superfaces of f are put into Q_k.

In Step 2, the nodes which are relevant for structural changes in $D(H)$ have been colored appropriately and stored in queues $Q_0, \cdots, Q_d$. Step 3 performs these changes by replacing each red node by two new ones, establishing their incidences, and integrating the blue faces of $A(H \cup \{h\})$. The steps for updating the auxiliary information in the incidence lattice are not discussed in detail. The only action that is not completely trivial is to provide a blue and unbounded 1-face e with $p(e)$. For this action note that e is contained in a red 2-face r in $A(H)$ that has at least two subfaces e_1 and e_2. The intersection of h with the two 1-flats through $p(e_1)$ and $p(e_2)$, and $2p(e_1)$ and $2p(e_2)$, gives two points on the 1-flat that contains e. $p(e)$ is easily derived from these two points.

Step 3.1. For k running from 1 to d and for each red face f in Q_k the following actions are taken:

Step 3.1.1. In $D(S)$ and in Q_k, f is replaced by two new red faces f_a and f_b representing the parts above and below h (Lemma 2.4(i)). f_a is called an above-node and f_b is called a below-node (see Fig. 3.5).

Step 3.1.2. Each superface of f is connected to both f_a and f_b, as in Fig. 3.5.

Step 3.1.3. Each white or grey subface of f is tested for lying above or below h. (It is convenient to use the auxiliary information for this test.) In the former case, it is connected to f_a, in the latter case to f_b. Each red subfacial above-node is connected to f_a, and each red subfacial below-node is connected to f_b (Lemma 2.4(ii)); again see Fig. 3.5.

Step 3.1.4. A new blue node f' is created and put into Q_{k-1}. f' represents the new $(k-1)$-face $f \cap h$, and thus, f' is connected to f_a and f_b (Lemma 2.4(ii.3)). In addition, it is connected to the blue subfaces of the red subfaces of f and to the black subfaces of the grey subfaces of f (Lemma 2.4(iii)); see Fig. 3.5. (If f' is a 0-face, then it is connected to the (-1)-face.)

Step 3.2. Finally, $Q_0, \cdots, Q_d$ are emptied and all black, red, grey, and blue nodes are unmarked by coloring them white.

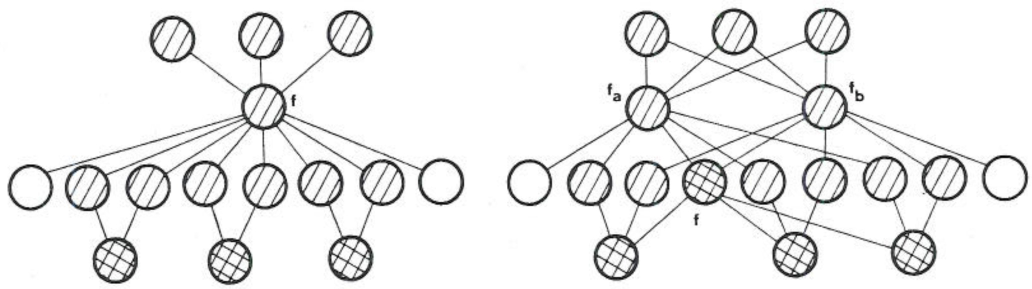


FIG. 3.5. *Updating a red face f.*

This completes the description of the algorithm. It is worthwhile to note that the same algorithm can also be used to construct cell complexes defined by a set of pseudo-hyperplanes. In this case, however, assumptions on the computability of intersections of the pseudo-hyperplanes must be made. We now turn to the analysis of the time requirements.

LEMMA 3.2. *Let H be a set of n hyperplanes in E^d and h be a hyperplane not in H. Then the above algorithm constructs $D(H \cup \{h\})$ in $O(n^{d-1})$ time from $D(H)$.*

Proof. It is trivial to implement Step 1 such that the time needed is proportional to the sum of deg (v), for all 0-faces v examined. This sum is in $O(n^{d-1})$ by Lemma 2.6.

A tedious look at Steps 2 and 3 reveals that the time required is proportional to the number of incidences of all faces in $A(H \cup \{h\})$ whose closures intersect h. Let inc be such an incidence between a k-face g and a $(k-1)$-face f, for some k with $0 \leqq k \leqq d$. Observe that the closure of g intersects h whereas the closure of f may not (Table 2.1, first row). Assume first that g is contained in h, that is, g and f are contained in h. As $h \cap A(H)$ is an arrangement in $d-1$ dimensions, there are at most $O(n^{d-1})$ incidences of this kind according to Lemma 2.5. Now assume that g is not contained in h and without loss of generality that g is above h. We will show that there are at most $O(n^{d-1})$ incidences of this kind.

For counting purposes, inc is attributed to the unique k-flat p that contains g. Define $H_p = \{h^*: h^* = h' \cap p$, for h' in H such that h' does not contain $p\}$ and define $h_p = h \cap p$. Then the faces in p whose closures intersect h are exactly the faces in $A(H_p \cup \{h_p\})$ that are active with respect to h_p. Thus, the number of incidences attributed to p is at most $S^k_{k-1}(n) = O(n^{k-1})$ by Theorems 2.7 and 2.8. However, there are at most $\binom{n}{d-k} = O(n^{d-k})$ k-flats defined by H, which implies that there are at most $O(n^{d-1})$ incidences attributed to all k-flats in $A(H)$. Summing up for k running from 0 to d gives again $O(n^{d-1})$, which completes the argument.

As shown in the beginning of § 3, the strategy to set up $A(H)$ for a set H of n hyperplanes in E^d is to successively insert the hyperplanes. Thus, we state the main result of this section, which follows directly from Lemma 3.2 and Lemma 2.5.

THEOREM 3.3. *Let H be a set of n hyperplanes in E^d, for $d \geqq 2$. Then the outlined algorithm constructs $A(H)$ in $O(n^d)$ time, and this is optimal.*

4. Applications. The problem of constructing an arrangement of hyperplanes is an underlying task for several applications, five of which are demonstrated in this section. The algorithm introduced in § 3 leads to optimal methods for computing λ-matrices and Voronoi diagrams. It also leads to methods for halfspatial range estimation, degeneracy testing, and finding minimum measure simplices that are faster than those previously known.

It turns out that the first three applications are closely related to the concept of "levels" in arrangements. It is for this reason that we introduce what we call the "ranked representation" of an arrangement, which is essentially the incidence lattice augmented with some additional information stored in the nodes.

Let H be a set of n nonvertical hyperplanes in E^d and let f be an arbitrary k-face in $A(H)$. The *ranks* $a(f)$, $o(f)$, and $b(f)$ *of* f denote the number of hyperplanes strictly above f, containing f, and strictly below f. Clearly, $a(f)+o(f)+b(f)=n$, $o(f)=d-k$, for $k=d-1, d$, and $o(f)\geqq d-k$, for $0\leqq k\leqq d-2$. $D(S)$ augmented with the ranks of its faces is called the *ranked representation of* $A(H)$. In what now follows, an algorithm is outlined that computes the ranks of each k-face, for $0\leqq k\leqq d$. The algorithm proceeds in three steps and uses a queue Q.

Step 1. The d-face f_{top} that has no hyperplane above it is identified. This can be done by testing, for each d-face f, whether there is an incident $(d-1)$-face whose supporting hyperplane is above $p(f)$.

Step 2. For each d-face f, the numbers $a(f)$, $o(f)$, and $b(f)$ are computed as follows:

Step 2.1. $a(f_{\text{top}})=o(f_{\text{top}})=0$ and $b(f_{\text{top}})=n$. f_{top} is marked and put into Q.

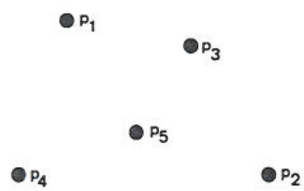
Step 2.2. If Q is not empty, then the first d-face f is removed from Q and the following actions are taken for each subface f^* of f: Let g denote the superface of f^* different from f. Unless g is already marked, the ranks of g are computed as follows: If f is above and g is below the hyperplane that supports f^*, then $a(g)=a(f)+1$, $o(g)=0$, and $b(g)=b(f)-1$. Otherwise, $a(g)=a(f)-1$, $o(g)=0$, and $b(g)=b(f)+1$. Finally, g is put into Q and Step 2.2 is repeated.

Step 3. For k running from $d-1$ to 0 and for each k-face f, the numbers $a(f)$, $o(f)$, and $b(f)$ are calculated as follows: $a(f)=\min\{a(g): g \text{ superface of } f\}$, and $b(f)=\min\{b(g): g \text{ superface of } f\}$. Finally, $o(f)=n-a(f)-b(f)$.

It is readily seen that this algorithm requires constant time per incidence, which implies that it is in $O(n^d)$ by Lemma 2.5.

4.1. The λ-matrix. Goodman and Pollack [GP4] introduced the λ-matrix of a finite set of points as a generalization of sorting to arbitrary dimensions. Among the applications, they suggest it can be used as a tool in pattern recognition, as it characterizes the set with respect to convexity properties.

Let $(p_1, \cdots, p_{d+1})$ denote a sequence of $d+1$ points in E^d, for $d\geqq 2$, with $p_i=(p_{i,1}, \cdots, p_{i,d})$, for $1\leqq i\leqq d+1$. The sequence is said to have *positive orientation* $((1, \cdots, d+1)>0)$ if $\det(p_{i,j})>0$, where $p_{i,d+1}=1$, for $1\leqq i, j\leqq d+1$. $(1, \cdots, d+1)=0$ and $(1, \cdots, d+1)<0$ are analogously defined. As noted in [GP4], $(1,2,3)>0$ if (p_1, p_2, p_3) is oriented counterclockwise, $(1,2,3)=0$ if the points lie on a common line, and $(1,2,3)<0$ if the sequence is oriented clockwise. Let now $S=\{p_1, \cdots, p_n\}$ denote a set of n points in E^d. Then $\lambda(i_1, \cdots, i_d)$ denotes the number of points p_j in S such that $(i_1, \cdots, i_d, j) > 0$. The *λ-matrix* $\lambda(S)$ *of S* is the d-dimensional matrix with $\lambda(i_1, \cdots, i_d)$ as the element with indices $i_1, \cdots, i_d$ provided the points with indices $i_1, \cdots, i_d$ determine a unique hyperplane containing them. Otherwise, the element is not defined. For $d=2$, the λ-matrix is a two-dimensional array with entry (i, j) filled with the number of points that fall to the left of the directed line from p_i to p_j. Figure 4.1 shows a set of points in E^2 and the corresponding λ-matrix. (The undefined elements are denoted w.) For $d=3$, the λ-matrix is a three-dimensional array with entry (i, j, k) filled with the number of points that fall to the "positive" side of the plane determined by p_i, p_j, and p_k.



$$\lambda(S) = \begin{pmatrix} w & 1 & 0 & 3 & 2 \\ 2 & w & 3 & 0 & 1 \\ 3 & 0 & w & 2 & 1 \\ 0 & 3 & 1 & w & 2 \\ 1 & 2 & 2 & 1 & w \end{pmatrix}$$

FIG. 4.1. *Point-set and λ-matrix.*

Let $H = T(S)$ using the geometric transform T defined in § 2.1. Furthermore, let $h_i = T(p_i)$, for $1 \leqq i \leqq n$. By Observation 2.2, the points with indices $i_1, \cdots, i_d$ define a unique hyperplane if and only if the intersection of the hyperplanes in H with the same indices is a 0-face v of $A(H)$. In addition, $\lambda(i_1, \cdots, i_d) = a(v)$ if the positive side defined by the points is above $T^{-1}(v)$, and $\lambda(i_1, \cdots, i_d) = b(v)$, otherwise. These explanations suggest that $\lambda(S)$ be computed as follows:

Step 1. Construct the ranked representation of $A(H)$.

Step 2. Associate with each 0-face in $A(H)$ the list of hyperplanes in H which contain it.

Step 3. Derive the elements of $\lambda(S)$ from the ranks of the vertices of $A(H)$.

By now, the details of this strategy should be obvious. Due to Theorem 3.3 and the fact that $\lambda(S)$ consists of n^d elements, we conclude:

THEOREM 4.1. *Let S denote a set of n points in E^d, for $d \geqq 2$. Then there exists an algorithm which computes $\lambda(S)$ in $O(n^d)$ time, which is optimal.*

This is an improvement over the $O(n^d \log n)$ time algorithm presented in [GP4].

4.2. Halfspatial range estimation. Let S denote a set of n points in E^3 and let h denote a nonvertical plane. Let $a(h)$ denote the number of points strictly above h. The *halfspatial range search problem* requires that S be stored in a data structure such that for any nonvertical plane h, $a(h)$ can be computed easily. This problem is a generalization of the halfplanar range search problem as considered, e.g., in Willard [W] and Edelsbrunner and Welzl [EW4]. Since there seems to be no solution for the problem (as well as for the one in E^2) that is efficient in both time and space, we consider the following simpler *halfspatial range estimation problem*: S is to be stored such that it is easy to decide for a plane h whether $a(h) < \lceil n/2 \rceil$, or $a(h) \geqq \lceil n/2 \rceil$. The solution to be described below is a generalization of a data structure in Edelsbrunner and Welzl [EW2].

By Observation 2.1, a point p in S is above h if and only if $T(h)$ is below $T(p)$. Let the *K-level of* $A(H)$ (with $H = T(S)$) be the collection of regions (2-faces) r in

$A(H)$, with $a(r)=K-1$ together with the bounding edges and vertices. Clearly, $a(h)<K$ if and only if $T(h)$ is above or contained in the K-level of $A(H)$. This suggests that the K-level L_K, for $K=\lceil n/2 \rceil$, be used as the basis for our data structure.

Note that L_K intersects each vertical line exactly once and that the projection L'_K of the edges and vertices of L_K onto the x_1x_2-plane gives a planar subdivision defined by straight line edges. Let m denote the number of edges in L'_K. Then there exists a data structure that requires $O(m)$ space and $O(m)$ time for construction from L_K such that $O(\log m)$ time suffices to determine a region of L'_K whose closure includes a query point (see Kirkpatrick [K] or Edelsbrunner, Guibas, and Stolfi [EGS]). Thus, to determine whether or not $T(h)$ is above L_K, we locate projection $T(h)'$ of $T(h)$ in L'_K and then test whether or not $T(h)$ is above the region of L_K that belongs to the located region. This implies:

THEOREM 4.2. *Let S denote a set of n points in E^3 and m the number of edges of L_K, for $K=\lceil n/2 \rceil$. Then there exists a data structure that requires $O(m)$ space such that $O(\log n)$ time suffices to answer a halfspatial range estimation query. $O(n^3)$ time and space is needed to construct the data structure.*

Unfortunately, no upper bound better than $O(n^3)$ is currently known for m. We refer to Erdös, Lovasz, Simmons, and Straus [ELSS] and Edelsbrunner and Welzl [EW1], who derived independently nontrivial bounds for the corresponding quantity in E^2.

Clearly, the notion of a "level" and thus the above method can be generalized beyond three dimensions. Using all K-levels for $1 \leqq K \leqq n$, and binary elimination to determine the one immediately below a query point, yields a solution for the halfspatial range search problem. The complexities are $Q(n^3)$ space and $O(\log^2 n)$ time which matches the best but more general structure by Chazelle [C].

4.3. Order-*K* Voronoi diagrams. Voronoi diagrams have received a great deal of attention in such diverse areas as geography, archeology, crystallography, physics, mathematics, and computer science. Let S denote a set of n points in E^d, for $d \geqq 2$. Then $V(p)=\{x \text{ in } E^d: d(x,p)<d(x,q), \text{ for } q \text{ in } S-\{p\}\}$ is called the *Voronoi polyhedron of p in S*. The cell complex consisting of the Voronoi polyhedra and the bounding lower dimensional polyhedra is called the *order-*1 *Voronoi diagram* 1-VOD (S) *of S*. Shamos and Hoey [SH] were the first to describe an optimal algorithm for constructing 1-VOD (S) if S is in E^2. They also introduced "higher-order" Voronoi diagrams: $V(S')=\{x \text{ in } E^d: d(x,p)<d(x,q), \text{ for } p \text{ in } S' \text{ and } q \text{ in } S-S'\}$ is called the *Voronoi polyhedron of S'*. Let K be an integer with $1 \leqq K \leqq n-1$. Then the cell complex consisting of the Voronoi polyhedra (plus lower dimensional bounding polyhedra) for the subsets S' of S with cardinality K is called the *order-K Voronoi diagram* K-VOS (S) *of S*.

For simplicity, we restrict our attention to E^2; generalizations to three and higher dimensions are straightforward. In a separate paper, [ES], a transformation P is described that relates Voronoi diagrams in E^2 with arrangements of planes in E^3. Each point $p=(p_1,p_2)$ in S is transformed into the plane $P(p)$ that is tangent to the paraboloid $x_3=x_1^2+x_2^2$ and touches it in the point $(p_1,p_2,p_1^2+p_2^2)$. Let L_K denote the K-level of $A(H)$ (with $H=P(S)$) as defined in § 4.2). The vertical projection of the intersection of L_K with L_{K+1} (that is, all 1-faces e with $a(e)=K-1$ and their endpoints) yield K-VOD (S). The generalization of these considerations implies:

THEOREM 4.3. *Let S denote a set of n points in E^d. Then $O(n^{d+1})$ time suffices to construct all order-K Voronoi diagrams for S, for $1 \leqq K \leqq n-1$.*

In E^2, K-VOD (S) can be exploited to determine the K closest points to a query point x in $O(\log n+K)$ time. To this end, a region of K-VOD (S) is determined whose closure includes x. The region r in K-VOD (S) found uniquely determines the K closest points which can, e.g., be stored with r. Thus, the data structures in [K] or [EGS] (this issue, pp. 317–340) yield the result.

Unfortunately, storing the lists of closest points with each region r in each K-VOD (S), for $1 \leqq K \leqq n-1$, increases the space required to $O(n^4)$. If the explicit neighbor lists are required, then $O(n^4)$ is optimal for constructing all higher-order Voronoi diagrams, as Dehne claimed [D]. However, the lists can be encoded into the diagrams as follows:

> Let r denote a region in K-VOD (S), for some $K \geqq 1$. Then r is equipped with a pointer into an arbitrary region r' in $(K-1)$-VOD (S) such that the list of $K-1$ closest points of r' is contained in the list of r. (0-VOD (S) is defined to be E^2 and has an empty list of closest points associated.) The pointer from r to r' is labelled with the one point in the list of r that is missing in the one of r'.

This strategy reduces the space required to $O(n^{d+1})$ and retains the query time of $O(\log n + K)$.

4.4. Degeneracy testing. A set S of $n \geqq d+1$ points in E^d is said to be in *general position* if any subset of $d+1$ points is affinely independent, that is, there is no hyperplane that contains $d+1$ points of S. Recently, van Leeuwen [vL] posed the question whether $O(n^2 \log n)$ time is the best possible time bound for an algorithm that decides whether or not n points in E^2 are in general position. Theorem 3.3 implies that the answer is negative:

THEOREM 4.4. *Let S denote a set of n points in E^d. Then there is an algorithm that decides in $O(n^d)$ time and $O(n^2)$ space whether or not S is in general position.*

Proof. S is in general position if and only if no $d+1$ hyperplanes in H (with $H = T(S)$) intersect in a common point or are normal to a common hyperplane. Construct all two-dimensional subarrangements and determine whether or not any one contains d 1-flats intersecting in a common point or normal to a common 1-flat. There are $O(n^{d-2})$ such subarrangements, and each requires $O(n^2)$ time and space to construct.

4.5. Minimum measure simplices. For simplicity, we confine the discussion in this section to E^2 and thus to minimum area triangles. Generalizations to higher dimensions are straightforward. Let S denote a set of n points in E^2. Any three points p_i, p_j, p_k of S define a triangle TR (i, j, k) with area $m(i, j, k)$. Then MAT (S) = TR (i_0, j_0, k_0) such that $m(i_0, j_0, k_0)$ assumes the minimum is called a *minimum area triangle of S*. We can restrict our attention to S in general position. Otherwise, there are three points on a line that define a triangle with area zero. However, this case can be checked in $O(n^2)$ time by Theorem 4.4.

The problem of finding a minimum area triangle was first considered by Dobkin and Munro [DM] who gave an $O(n^2 \log^2 n)$ time and space algorithm. Later, Edelsbrunner and Welzl [EW2] improved their result to $O(n^2 \log n)$ time and $O(n)$ space. Both approaches are based on:

Observation 4.5. Let MAT (S) = TR (i, j, k). Then p_k is the closest point among S-$\{p_i, p_j\}$ to the line through p_i and p_j.

The line through p_i and p_j corresponds to the intersection of $T(p_i)$ and $T(p_j)$ by Observation 2.1. Furthermore, p_k corresponds to the line $T(p_k)$ immediately above or below (vertically) the intersection. (As two parallel lines have no intersection, S is assumed to contain no two points on a vertical line. Otherwise, S is assumed to contain no two points on a vertical line. Otherwise, S is rotated by an appropriate angle, which takes $O(n^2)$ time.) The following strategy for computing MAT (S) is suggested by these observations:

Step 1. Construct $A(H)$, with $H = T(S)$.

Step 2. For each region r in $A(H)$ and for each vertex $v = T(p_i) \cap T(p_j)$ on the boundary of r the following actions are taken: Determine each line $T(p_k)$ that contains an edge of r and calculate the area $m(i, j, k)$, provided k is different from i and j. Record the triple (i, j, k) if $m(i, j, k)$ is less than the area of the smallest triangle determined so far.

Obviously, for each vertex $v = T(p_i) \cap T(p_j)$ in $A(H)$, the line $T(p_k)$ immediately above or below v is among the lines tested for v. Not counting the requirements for Step 1, the amount of time required for each region r in $A(H)$ is proportional to the product of $\deg_0(r)$ and $\deg_1(r)$. The sum of these products, over all regions r in $A(H)$, is in $O(n^2)$ by Corollary 2.9. Observing that the algorithm given above as well as all results used in this section generalize to three and higher dimensions, we conclude:

THEOREM 4.6. *Let S denote a set of $n \geqq d+1$ points in E^d, $d \geqq 2$. Then the minimum measure simplex determined by $d+1$ points in S can be found in $O(n^d)$ time and space.*

This result, and the presented algorithm, were independently discovered for $d = 2$ by Chazelle, Guibas, and Lee [CGL].

5. Discussion. We have presented an optimal method for constructing cell complexes defined by hyperplanes in E^d, basing our algorithm on a new combinatorial result (Theorems 2.7 and 2.8). The result also holds for arrangements of pseudo-hyperplanes. In fact, the algorithm applies to the problem of constructing such more general arrangements, provided that the pseudo-hyperplanes are, in some sense, computationally simple.

Bieri and Nef [BN] described the only existing algorithm known to the authors which computes the faces of an arrangement in E^d. The disadvantage of their algorithm is that it requires more time than ours and does not explicitly establish the incidences between the faces. The significance of the presented optimal method is that there is a host of applications leading to new and faster solutions for problems thought to be unrelated in the past. The five applications shown in § 4 are:

(1) An optimal algorithm for computing λ-matrices for finite sets of points in Euclidean spaces, improving the best result known to date [GP4].

(2) A new data structure and algorithm for halfspatial range estimation for which no sophisticated solution was yet known.

(3) An optimal algorithm for constructing all higher-order Voronoi diagrams. This improves the result of Dehne [D] in E^2 and appears to be the first algorithm known for higher dimensions.

(4) A faster algorithm for testing for degeneracies in a set of points, providing an improvement of existing algorithms and a partial answer to question P20 of [vL].

(5) A faster algorithm for computing minimum measure simplices defined by a set of points. This improves the results of [DM] and [EW2] in E^2 and appears to be the first nontrivial result beyond $d = 2$.

The algorithm also immediately leads to an upper bound on the number of arrangements. For n hyperplanes in E^d, the algorithm takes $O(n^d)$ binary decisions and so can construct only $2^{O(n^d)}$ combinatorially different arrangements. This is also an upper bound for the number of different combinatorial types since the algorithm is not restricted to any subclass of arrangements. This improves the upper bound given in [GP4].

Several open problems are suggested by the results in this paper. Three of the most important ones are as follows:

(1) Let H be a set of n planes in E^3 and let $|L_K(H)|$ denote the number of regions of the K-level $L_K(H)$ of $A(H)$. We define $b_K(n) = \max\{|L_K(H)|: H \text{ a set of } n \text{ planes in } E^3\}$. The authors can show $b_k(n) = \Omega(nk \log k)$, but no nontrivial upper bound is currently available. It is likely that the methods in [ELSS] or [EW1] can be extended in some nontrivial way to obtain an upper bound.

(2) Is $\Omega(n^2)$ a lower bound for deciding whether or not a set of n points in E^2 is in general position? Due to its simple appearance, this computational problem seems to be well suited for a lower bound analysis. Also, there are several problems to which degeneracy testing in E^2 can be reduced. Examples are the minimum area triangle problem of § 4.5, and several geometric problems posed in [LP], [EOW], and [EMPRWW].

(3) There are some geometric problems for which $O(n^2 \log n)$ time solutions are known that might be amenable to applications of Theorem 3.3 to reduce the time to $O(n^2)$. An example is the "shadow problem" of Lee and Preparata [LP]:

> Let S denote a set of n line segments in E^2. Compute a direction (if it exists) such that each line parallel to the direction intersects at most one line segment. If light shines parallel to this direction, none of the shadows overlap.

Acknowledgments. We thank Emmerich Welzl for discussions on Theorem 2.7. We also thank Friedrich Huber for implementing the construction of arrangements in arbitrary dimensions, and Gerd Stoeckl for implementing the algorithms presented in §§ 4.1 and 4.3. The third author wishes to thank Jack Edmonds for the many enlightening discussions.

REFERENCES

[AW] G. L. ALEXANDERSON AND J. E. WETZEL, *Simple partitions of space*, Math. Mag., 51 (1978), pp. 220-225.

[BN] H. BIERI AND W. NEF, *A recursive plane-sweep algorithm, determining all cells of a finite division of R^d*, Computing, 28 (1982), pp. 189-198.

[B] K. Q. BROWN, *Geometric transforms for fast geometric algorithms*, Ph.D. thesis, Rep. CMU-CS-80-101, Dept. Computer Science, Carnegie-Mellon Univ., Pittsburgh, PA, 1980.

[C] B. M. CHAZELLE, *How to search in history*, Proc. International Symposium on Fundamental Computer Theory, Springer-Verlag, Berlin, 1983.

[CGL] B. M. CHAZELLE, L. J. GUIBAS AND D. T. LEE, *The power of geometric duality*, Proc. 24th Annual IEEE Symposium of Foundations of Computer Science, 1983, pp. 217-225.

[D] F. DEHNE, *An optimal algorithm to construct all Voronoi diagrams for k nearest neighbor searching in the Euclidean plane*, Proc. 20th Annual Allerton Conference on Communication Control and Computing, 1982.

[DM] D. P. DOBKIN AND J. I. MUNRO, private communication.

[EGS] H. EDELSBRUNNER, L. GUIBAS AND J. STOLFI, *Optimal point location in a monotone subdivision*, this issue, pp. 317-340.

[EMPRWW] H. EDELSBRUNNER, H. A. MAURER, F. P. PREPARATA, A. L. ROSENBERG, E. WELZL AND D. WOOD, *Stabbing line segments*, BIT, 22 (1982), pp. 274-281.

[ES] H. EDELSBRUNNER AND R. SEIDEL, *Voronoi diagrams and arrangements*. Rep. 85-669, Dept. Comput. Sci., Cornell Univ., Ithaca, NY, 1985.

[EOW] H. EDELSBRUNNER, M. H. OVERMARS AND D. WOOD, *Graphics in Flatland: a case study*, in Advances in Computing Research, Vol. 1: Computational Geometry, F. Preparata, ed., JAI Press, 1983, pp. 53-59.

[EW1] H. EDELSBRUNNER AND E. WELZL, *On the number of line-separations of a finite set in the plane*, J. Combin. Theory Ser. A., to appear.

[EW2] ———, *Constructing belts in two-dimensional arrangements with applications*, this Journal, to appear.

[EW3] ——, *On the maximal number of edges of many faces in an arrangement*, Rep. F99, Inst. for Information Processing Technical Univ. Graz, Graz, Austria, 1982.

[EW4] ——, *Halfplanar range search in linear space and $O(n^{0.695})$ query time*, Rep. F111, Inst. for Information Processing Technical Univ. Graz, Graz, Austria, 1983.

[ELSS] P. ERDÖS, L. LOVASZ, A. SIMMONS AND E. G. STRAUS, *Dissection graphs of planar point sets*, in A Survey of Combinatorial Theory, J. N. Srivastava et al., eds., North-Holland, Amsterdam, 1973, pp. 139–149.

[GP1] J. E. GOODMAN AND R. POLLACK, *Proof of Grünbaum's conjecture on the stretchability of certain arrangements of pseudolines*, J. Combin. Theory Ser. A, 29 (1980), pp. 385–390.

[GP2] ——, *Three points do not determine a (pseudo-) plane*, J. Combin. Theory Ser. A, 30 (1981), pp. 215–218.

[GP3] ——, *A theory of ordered duality*, Geom. Dedicata, 12 (1982), pp. 63–74.

[GP4] ——, *Multidimensional sorting*, this Journal, 12 (1983), pp. 484–507.

[G1] B. GRÜNBAUM, *Convex Polytopes*, Interscience, London, 1967.

[G2] ——, *Arrangements and spreads*, CBMS Regional Conference Series in Applied Mathematics 10, American Mathematical Society, Providence, RI, 1972.

[G3] ——, *Arrangements and hyperplanes*, Congressum Numerantium III, Louisiana Conference on Combinatrics Graph Theory and Computing, 1971, pp. 41–106.

[K] D. G. KIRKPATRICK, *Optimal search in planar subdivisions*, this Journal, 12 (1983), pp. 28–35.

[LP] D. T. LEE AND F. P. PREPARATA, *Euclidean shortest paths in the presence of rectilinear barriers*, Proc. 7th Conference on Graphtheoretical Concepts in Computer Science, (WG 81), Carl Hanser, 1981, pp. 303–314.

[O] J. O'ROURKE, *On-line algorithms for fitting straight lines between data ranges*, Comm. ACM, 24 (1981), pp. 574–578.

[SH] M. I. SHAMOS AND D. HOEY, *Closest-point problems*, Proc. 16th Annual IEEE Symposium Foundations of Computer Science, 1975, pp. 151–162.

[St] J. STEINER, *Einige Gesetze über die Theilung der Ebene und des Raumes*, J. Reine Angew. Math., 1 (1826), pp. 349–364.

[vL] J. VAN LEEUWEN, *P*20, Bull. EATCS, 19 (1983), p. 150.

[W] D. E. WILLARD, *Polygon retrieval*, this Journal, 11 (1982), pp. 149–165.

[Z] TH. ZASLAVSKY, *Facing up to arrangements: face-count formulas for partitions of space by hyperplanes*, Memoirs Amer. Math. Soc. 154, American Mathematical Society, Providence, RI, 1975.