Discrete Comput Geom 5:197–216 (1990)

Discrete & Computational
Geometry

The Complexity of Many Cells in Arrangements of Planes and Related Problems*

Herbert Edelsbrunner,[1] Leonidas Guibas,[2,3] and Micha Sharir[4,5]

[1] Computer Science Department, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

[2] Computer Science Department, Stanford University, Stanford, CA 94305, USA

[3] DEC Systems Research Center, 130 Lytton Avenue, Palo Alto, CA 94301, USA

[4] Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, USA

[5] School of Mathematical Sciences, Tel Aviv University, 69978 Tel Aviv, Israel

Abstract. We consider several problems involving points and planes in three dimensions. Our main results are: (i) The maximum number of faces bounding m distinct cells in an arrangement of n planes is $O(m^{2/3}n \log n + n^2)$; we can calculate m such cells specified by a point in each, in worst-case time $O(m^{2/3}n \log^3 n + n^2 \log n)$. (ii) The maximum number of incidences between n planes and m vertices of their arrangement is $O(m^{2/3}n \log n + n^2)$, but this number is only $O(m^{3/5-\delta}n^{4/5+2\delta} + m + n \log m)$, for any $\delta > 0$, for any collection of points no three of which are collinear. (iii) For an arbitrary collection of m points, we can calculate the number of incidences between them and n planes by a randomized algorithm whose expected time complexity is $O((m^{3/4-\delta}n^{3/4+3\delta} + m) \log^2 n + n \log n \log m)$ for any $\delta > 0$. (iv) Given m points and n planes, we can find the plane lying immediately below each point in randomized expected time $O([m^{3/4-\delta}n^{3/4+3\delta} + m] \log^2 n + n \log n \log m)$ for any $\delta > 0$. (v) The maximum number of facets (i.e., $(d-1)$-dimensional faces) bounding m distinct cells in an arrangement of n hyperplanes in d dimensions, $d > 3$, is $O(m^{2/3}n^{d/3} \log n + n^{d-1})$. This is also an upper bound for the number of incidences between n hyperplanes in d dimensions and m vertices of their arrangement. The combinatorial bounds in (i) and (v) and the general bound in (ii) are almost tight.

* Work on this paper by the first author has been supported by Amoco Fnd. Fac. Dev. Comput. Sci. 1-6-44862 and by NSF Grant CCR-8714565. Work by the third author has been supported by Office of Naval Research Grant N00014-87-K-0129, by National Science Foundation Grant DCR-82-20085, by grants from the Digital Equipment Corporation, and the IBM Corporation, and by a research grant from the NCRD—the Israeli National Council for Research and Development. An abstract of this paper has appeared in the *Proceedings of the 13th International Mathematical Programming Symposium*, Tokyo, 1988, p. 147.

1. Introduction

In this paper we extend the techniques developed in a companion paper [EGS] to a variety of problems involving arrangements of planes in three dimensions or hyperplanes in higher dimensions, as listed in the abstract. Problems of this kind, in particular those involving incidences between points and lines, or hyperplanes, or various other types of curves or surfaces, were considered for a long time as key problems in combinatorial geometry (see [PE] for a survey). A major breakthrough was achieved by Szemerédi and Trotter [ST] who showed that the maximum number of incidences between m points and n lines in the plane is $\Theta(m^{2/3}n^{2/3} + m + n)$. Edelsbrunner and Welzl [EW] extended the lower bound to apply also to the complexity of m distinct cells in an arrangement of n lines in the plane (see also [Ca]), and also obtained a weaker upper bound for that complexity. These lower bound and weak upper bound were later extended to three and higher dimensions by Edelsbrunner and Haussler [EH] and by Edelsbrunner [E1], who showed that the maximum complexity of m distinct cells in an arrangement of n hyperplanes in $d \geq 3$ dimensions is $\Omega(m^{2/3}n^{d/3} + n^{d-1})$ and $O(m^{1/2}n^{d/2} + n^{d-1})$; here the complexity of a cell is the number of facets (i.e., $(d-1)$-dimensional faces) bounding it. The lower bound is derived by a fairly easy "lifting" of the two-dimensional construction of [ST] and [EW] to higher dimensions.

Recent progress was achieved by the authors in two companion papers [CEG*], [EGS]. These papers introduce a new technique, whose aim is to decompose problems, such as the above, into several subproblems of smaller size; such a decomposition is achieved by drawing a random sample either of the given points or of the given lines or curves, and then by splitting the collection of points and curves into subcollections according to their interaction with the sample objects. This decomposition leads to a recurrence formula for the desired quantity which is then solved to yield improved upper bounds. This approach led to a tight bound $\Theta(m^{2/3}n^{2/3} + n)$ on the complexity of m distinct cells in an arrangement of n lines in the plane, and to several other improved upper bounds for the maximum number of incidences between points and curves, as well as for the maximum complexity of many cells in arrangements of curves, where the curves are either line segments, or circles, or pseudolines.

These techniques were also adapted to yield comparably efficient randomized algorithms for calculating m cells in a given arrangement or for finding all incidences between given points and curves (see, e.g., [EGH*]). Very recently, improved deterministic algorithms of this kind were obtained by Agarwal [A].

In this paper we extend these recent techniques to three and higher dimensions, to obtain the results stated in the abstract. The bounds that we obtain (for cases (i), (v), and the general bound in (ii)) are almost tight, apart for a logarithmic factor, as follows from [EH] and [E1]. Our techniques rely heavily on the bounds obtained for the corresponding two-dimensional problems. In certain cases (notably in cases (i) and (v)) our decomposition strategy is considerably simpler than the one used in the two-dimensional cases, mainly because the bounds we want to establish are larger; in the other cases, we use a more complex decomposition strategy which is a generalization of the strategies used in the two-dimensional problems.

The paper is organized as follows. In Section 2 we analyze the complexity of many cells in arrangements of planes in three dimensions. Section 3 presents an efficent worst-case algorithm for their calculation. In Section 4 we consider the "simpler" problems (ii)–(iv) for three-dimensional arrangements of planes, and in Section 5 we generalize the analysis of Section 2 to higher dimensions.

2. The Complexity of Many Cells in Arrangements of Planes

Let $\Pi = \{\pi_1, \ldots, \pi_n\}$ be a collection of n planes in three dimensions, and let $c_1, \ldots, c_m$ be m distinct cells of the arrangement $A = A(\Pi)$ of these planes. (We assume familiarity of the reader with the notion of arrangements; see [E1] for basic information concerning arrangements.) The complexity of a cell is the number of (two-dimensional) faces bounding the cell (by Euler's formula, this complexity also bounds, up to a constant factor, the number of vertices and edges of the cell). Our goal is to obtain a sharp upper bound on the sum $K(m, n)$ of the complexity of m such cells. A lower bound $K(m, n) = \Omega(m^{2/3}n + n^2)$ has been established in [EH] (see also [E1]). Here we prove an almost matching upper bound of the form $K(m, n) = O(m^{2/3}n \log n + n^2)$.

It is convenient to reformulate the problem as the following one. Let $p_1, \ldots, p_m$ be m given points. Our task is to analyze the complexity of the cells of $A(\Pi)$ that contain these points, where any such cell, even if it contains more than one point, is to be counted just once. We will however assume that no point lies on any plane, and that initially no two of these points lie in the same cell. (The problem is formulated in a more general way, because our analysis will break the problem recursively into subproblems where cells may contain more than one point.)

The analysis uses the following straightforward divide and conquer approach. We first partition Π into two subsets Π_1, Π_2, each containing roughly $n/2$ planes; we refer to the planes in these subsets as the red planes and the blue planes, respectively. We next calculate recursively the cells containing the points p_i in each of the subarrangements $A(\Pi_1)$, $A(\Pi_2)$, and then intersect these cells to obtain the desired cells of $A(\Pi)$. Note that if a point p lies in a red cell R and in a blue cell B, then the final cell E containing p is simply $R \cap B$, and, since all these cells are convex polyhedra, the complexity of E is bounded by the sum of the complexities of R and of B. However, R (or B) might contain more than one point p_i, in which case we would not want to use the entire complexity of R in bounding the complexity of each of the final cells containing these points, as this duplication might result in an unacceptably high bound. This major technical difficulty is dealt with in the following subsection, and its solution (the "combination lemma") then enables us to complete our analysis of $K(m, n)$.

2.1. *The Combination Lemma for Planes in Three-Dimensional Space*

Let $p_1, \ldots, p_m$ be m points. We are given two arrangements of planes, the red arrangement, consisting of n_r red planes, and the blue arrangement, consisting of n_b

blue planes. Let $n = n_r + n_b$. Let $R_1, \ldots, R_s$ be the cells in the red arrangement containing the points p_i (where a cell can contain more than one point), and let $B_1, \ldots, B_t$ be the blue cells containing these points. For each p_i let $E_i = R_{s_i} \cap B_{t_i}$ be the "purple" cell containing p_i (where R_{s_i}, B_{t_i} are the red and blue cells containing that point). By considering only a subset of the points p_i, if necessary, we can assume, without loss of generality, that all the cells E_i are distinct, and thus have pairwise disjoint interiors. (While this assumption explicitly holds at the first level of recursion, further levels will generally have more than one point in the cells E_i; to enforce the above property, we simply choose one representative point out of each cell E_i and (temporarily) ignore the others.) Let ρ, β denote the total number of faces on the boundaries of the red cells and the blue cells, respectively. Our goal is to obtain a sharp upper bound, in terms of ρ and β, on the total number of faces on the boundaries of the purple cells E_i.

Similar to the two-dimensional case [EGS], and as noted above, it is easily verified that if each red cell and each blue cell contains just a single point, then the desired purple complexity is at most $\rho + \beta$. The main step in bounding the purple complexity will thus aim to partition each red cell that contains more than one point into subcells, each containing only a single point, using (portions of) the blue planes, and to obtain a symmetric partitioning of the blue cells.

We consider the blue planes in a fixed sequence, and add them to the red arrangement one at a time. We use each of them to decompose further as many red cells as necessary into subcells, called "red-blue" cells, as follows. Let $R = R_i$ be a fixed red cell which contains $m_i > 1$ points. Originally, all these points lie in a single red-blue cell, namely R itself. As some blue planes are added, they partition R into subcells; the subcells that still contain at least one of those points are the current red-blue subcells of R. This decomposition is applied simultaneously to all red cells. After adding all blue planes, the final red-blue cells coincide with the desired purple cells E_i.

What we want to estimate is the "red complexity" of the red-blue cells, namely the number of red (sub)faces bounding these cells (so the blue faces along their boundaries are ignored); more precisely, we want to estimate the number of *additional* red subfaces generated by the intersections of each blue plane with the current set of red-blue cells.

Consider the cell decomposition caused by the ith blue plane π_i. Let R_j be a red cell that has already been decomposed into a number of red-blue subcells, and assume that π_i passes through some of them. For each red-blue subcell c crossed by π_i one of the following two cases arises: either π_i splits c into two red-blue subcells, because it splits the set of points contained in c into two nonempty subsets, or it cuts off a "useless" portion of c that contains none of the points, so there results only one new red-blue cell. Note that in the latter case, the red complexity of the new red-blue cell cannot be larger than the red complexity of c, so this case does not cause an increase in the red complexity of red-blue cells, and we can ignore it. Let k_i denote the number of red-blue subcells that were split into two red-blue subcells by π_i. Since this split must be accompanied by a corresponding split of a subset of the given points into two nonempty subsets, we must have $\sum_i k_i \leq m$.

Let c be red-blue subcell that has been split by π_i (in the above sense). The

increase in the red complexity of c caused by this split is equal to the number of red edges bounding the convex polygon $c \cap \pi_i$. But this polygon is a face in the planar arrangement A_i, formed in the plane π_i by its intersection with all red planes and with all previously added blue planes. It follows that the overall increase in the red complexity of red-blue cells caused by π_i is bounded by the maximum complexity of k_i faces in the arrangement A_i. Since A_i is formed by at most n lines, the results of [CEG*] imply that this increase is at most $O(k_i^{2/3}n^{2/3} + n)$. Summing this over all blue planes π_i, we obtain

$$\sum_{i=1}^{n_b} O(k_i^{2/3}n^{2/3} + n) = O\left(\left(\sum_{i=1}^{n_b} k_i^{2/3}\right) \cdot n^{2/3} + nn_b\right).$$

But

$$\sum_{i=1}^{n_b} k_i^{2/3} \leq \left(\sum_{i=1}^{n_b} k_i\right)^{2/3} \cdot n_b^{1/3} \leq m^{2/3}n_b^{1/3},$$

because $\sum k_i \leq m$, as noted above. Hence the total increase in the red complexity is at most

$$O(m^{2/3}n + n^2).$$

We now obtain in a completely symmetric manner a decomposition of the blue cells into a collection of "blue-red" subcells, each containing just a single point, and derive a symmetric bound on their overall "blue" complexity. As observed, if a point p_i lies in a red-blue subcell c_i and in a blue-red subcell $\bar{c}_i$, then these two cells coincide with the final purple cell E_i containing p_i, so the number of faces of E_i is obviously equal to the red complexity of c_i plus the blue complexity of $\bar{c}_i$.

Putting everything together, we finally obtain

Lemma 2.1 (Combination Lemma for Arrangements of Planes). *Given an arrangement of n_r red planes, another arrangement of n_b blue planes, and m points not lying on any of these planes, the total number of faces bounding the "purple" cells containing these points, in the arrangement formed by all $n = n_r + n_b$ planes, is at most*

$$\rho + \beta + O(m^{2/3}n + n^2),$$

where ρ (resp. β) is the total number of faces bounding the cells of the red (resp. blue) arrangement containing these points.

2.2. *The Complexity of Many Cells*

We now return to our analysis of $K(m, n)$. Applying the combination lemma to the two subarrangements $A(\Pi_1)$, $A(\Pi_2)$, we can readily obtain a recurrence of the form

$$K(m, n) = 2K(m, n/2) + O(m^{2/3}n + n^2).$$

However, we do not continue this recursive process all the way down. When n first becomes smaller than or equal to $m^{1/3}$ we use the trivial bound $K(m, n) = O(n^3) = O(m)$. Since when this happens we have $m \leq (2n)^3$, it follows that $m = m^{2/3}m^{1/3} = O(m^{2/3}n)$. It is now easy to verify that the solution to the above recurrence is

$$K(m, n) = O(m^{2/3}n \log n + n^2).$$

(Note that the factor $\log n$ enters the formula for $K(m, n)$ because in the recurrence relation the term $O(m^{2/3}n)$ is linear in n.)

We have thus shown

Theorem 2.2. *The complexity of m distinct cells in an arrangement of n planes in three dimensions is $O(m^{2/3}n \log n + n^2)$.*

Corollary 2.3. *The maximum number of incidences between n planes in three dimensions and m vertices of their arrangement is $O(m^{2/3}n \log n + n^2)$.*

Proof. We use the following well-known argument (see [EW] and [E1]). Replace each plane by two parallel planes translated a small distance away in both directions. If all these translation distances are equal and sufficiently small, each vertex p of the arrangement of the original planes will lie in a distinct cell of the new arrangement, whose complexity is $2d$, where d is the number of original planes incident to p. Thus the maximum number of incidences between m points and n planes as above is bounded by $\frac{1}{2}K(m, 2n)$, which completes the proof of the corollary. □

Remarks. (1) The proof of Theorem 2.2 is significantly simpler than the elaborate proof of the corresponding bound in two dimensions (see [CEG*] and [EGS]), and of the simpler variants of the three-dimensional problem analyzed in the following section. This is because (i) we have already made strong use of the results of the two-dimensional problem in the proof of the combination lemma, and (ii) the weaker bound in three dimensions allows a straightforward divide-and-conquer strategy, which cannot be used to derive the smaller bound in two dimensions (or the smaller bounds for the variant problems given below).

(2) The bound given in Corollary 2.3 almost matches the lower bound $\Omega(m^{2/3}n + n^2)$ given in [EH] and [E1]. It is interesting to note that the lower bound construction in [EH] is a fairly trivial extension of the two-dimensional lower bound for incidences between points and lines. Thus the main tools used in the derivations of both the lower and upper bounds for the case of planes are principally based on results on two-dimensional arrangements.

(3) A direct proof of Corollary 2.3 could also be given, without reducing incidences to the complexity of many cells. One such direct proof is an easy modification of the proof given above, including a modified combination lemma for "merging incidences" between a set of points and two collections of red and of

blue planes. As a hint to the interested reader, who may wish to work out the details of this modified proof, we note that the main technical difficulty that arises is that the given points are assumed to be vertices of the overall arrangement, but they need not be vertices of the blue or of the red subarrangements. A combination lemma is then needed to handle points of this sort (which generally lie on an edge of one arrangement and on a plane of the other).

(4) An obvious open problem is whether the upper bounds in Theorem 2.2 or in Corollary 2.3 can be tightened to match the corresponding lower bounds, i.e., whether the $\log n$ factor can be removed. This factor is just an artifact of our simple-minded divide-and-conquer strategy, and we would hope that better strategies might get rid of this factor.

(5) The technique presented in this section has recently been generalized (and somewhat simplified) in [AS] to obtain sharp upper bounds on the complexity of cells in an arrangement of n (possibly intersecting) triangles in 3-space. For example, it is shown there that the total complexity of all nonconvex cells in such an arrangement is $O(n^{7/3}\alpha(n)^{2/3} \log^{4/3} n)$, which is almost tight in the worst case.

(6) It would be interesting to extend these results to arrangements of other surfaces (e.g., spheres) in three dimensions. Since the analysis given above strongly uses the straightness of planes and convexity, new techniques will be required to handle curved surfaces.

3. Calculating Many Cells

Next we sketch an algorithm for efficient calculation of m cells in an arrangement of n planes in time $O(m^{2/3}n \log^3 n + n^2 \log n)$.

The algorithm relies on the following operation. Suppose we are given two intersecting convex polyhedra P and Q in three dimensions, of size p and q, respectively. We are told that P and Q intersect and we are furthermore given a point x in their intersection. Our goal is to construct $R = P \cap Q$ in time proportional to r, the size of R, times a logarithmic factor in p, q, and r. (We think of p as representing the number of faces of P, which is linearly related to the number of edges and the number of vertices of P; similarly for q and r.)

In order to do this we first develop a subroutine that intersects a planar convex polygon with a convex polyhedron in three dimensions within an analogous time bound. So let C be a convex polygon of size c, and let Q and q be as above. Again, we assume that we are given a point z in the intersection S of C and Q, where S is a planar convex polygon of size s.

It is actually instructive to go down in dimension one more time and consider how to intersect two convex polygons in output-sensitive time. A straightforward technique for doing so is described in [EGS] and is based on "shooting." The idea is that we find a point on the boundary of the intersection and then walk along this boundary by shooting from the current edge of the inner polygon toward the outer one, in order to find the next vertex of the intersection. There are two cases. It may happen that the inner polygon edge finishes before it exits the outer polygon, in

which case we move to the next edge of the inner polygon and repeat the process. Or we may encounter the outer polygon before the inner polygon edge finishes, in which case we have a new vertex of the intersection polygon and the roles of the inner and outer polygons are swapped. Since shooting a ray in a convex polygon takes logarithmic time, the whole intersection procedure can be done in time proportional to the size of the intersection times a logarithmic factor. Note that logarithmic shooting within a convex polygon can be done without any preprocessing, provided we store the polygon vertices in an array or a binary tree in their order along the polygon.

We now discuss how to "lift" this argument to three dimensions and solve our subproblem. Note first that ray shooting in a convex polyhedron can also be done in logarithmic time, for example by using a Dobkin–Kirkpatrick style [DK] hierarchical representation (see also Section 9.5 of [E1]). So we proceed as follows. From our given point z in the intersection of C and Q shoot any ray in the plane of C and find its nearest intersection point v with C or Q. We now implement a "walk-around" procedure similar to the one outlined above. If v is on C, then we shoot along the current edge of C to see if we exit Q. As long as we have not, we just keep walking around C. If we do exit Q, then we have a new vertex w of S, lying in some face f of Q. Now things are more interesting, because we need to shoot from w in both C and the face f of Q in order to find the next vertex of S. It is still true, however, that both of these shots can be accomplished in logarithmic time, and this remains so as we go around along the surface of Q. So the polygon–polyhedron case is only slightly more complex than the polygon–polygon case. The intersection S can again be computed in time proportional to s times a logarithmic factor.

We now go back to the original polyhedron–polyhedron problem. From the common point x of P and Q let us shoot an arbitrary ray and let the nearest face hit be C, say it is a face of P. (Since our polyhedra could be unbounded, it is advisable to choose this initial ray to connect x to a point on some face of P or of Q; otherwise we might not hit any face at all.) Then we can apply the above procedure to intersect C with Q and obtain S, a face of the desired intersection output R. Now let e be an edge of S. We claim that our subprocedure above can be used to compare the face T of R adjacent to S along e. There are three cases. If e is a subedge of an edge of P, then we simply take the other face of P along that edge and intersect it with Q to get T. If e is a subedge of an edge of Q we do the symmetric operation. Finally if e is an edge arising out of the intersection of a face (in this case C) of P with a face D of Q, then we simply switch the roles of P and Q (inner with outer) and now intersect polygon D with polyhedron P. If we repeat this process we will eventually discover all the faces of the intersection polyhedron R. It is clear that the cost of the computation is proportional to the number of edges of R, times a factor which is the cost of shooting, paid a constant number of times per edge. Overall this gives us the desired output-sensitive algorithm for polyhedral intersection:

Lemma 3.1. *Given two convex polyhedra P and Q in three dimensions, of size p and q, respectively, which have already been preprocessed for logarithmic-time*

ray shooting queries, as in [DK], we can compute their intersection R in time $O(r \log(p+q))$, where r is the size of R.

We now apply this lemma to compute the m desired cells in an n plane arrangement. The cells are assumed to be specified by m points $p_1, \ldots, p_m$, one point lying in the interior of each cell. We apply a divide-and-conquer technique as in the proceding proof. That is, we divide the collection of planes into two subcollections ("blue" and "red"), each containing roughly $n/2$ planes, and recursively compute the cells containing the given points in the blue and in the red subarrangements. We next preprocess each of these convex polyhedral cells, as in [DK], to prepare for logarithmic-time ray shooting inside them. The total cost of preprocessing is bounded by $O(K(m, n/2))$. Then, for each point p_j, we apply the above polyhedron–polyhedron intersection routine. However, since two of these points could lie in the same blue and the same red cell (at least when we are deep down the recursive process), we have to exercise some care to make sure that no intersection polyhedron is computed more than once. However, two points lie in the same intersection cell if and only if they lie in the same red cell and in the same blue cell. We can therefore easily partition the points by their red and blue cell membership, detect multiply-occurring intersection cells, and choose a single point to represent each such cell (this is similar to the technique used in [EGS] for a variant of this problem in two dimensions). With this precaution, the output-sensitive behavior of the above routine implies that all desired cells can be computed in time $O((K(m, n) + m) \log n)$ (including the preprocessing overhead as well). Moreover, when $m \geq n^3$, we can simply calculate the entire arrangement in time $O(n^3) = O(m)$ (as in [EOS]), and pick out the desired cells. To accomplish this last step, we apply the point-location algorithm in [EOS] (or that in [Ch]) to locate each of the given m points in the computed arrangement, in total time $O(m \log^2 n)$. All these considerations lead to the following recurrence:

$$T(m, n) = \begin{cases} 2T(m, n/2) + O(m^{2/3} n \log^2 n + n^2 \log n), & m < n^3, \\ O(m \log^2 n), & m \geq n^3, \end{cases}$$

for the maximum time $T(m, n)$ required by the algorithm; this recurrence is easily seen to solve to

$$T(m, n) = O(m^{2/3} n \log^3 n + n^2 \log n + m \log^2 n).$$

Finally, if we wish to calculate m distinct cells in an arrangement of planes, and the cells are given by a single point in each, then necessarily $m < n^3$, so that we can drop the term $O(m \log^2 n)$ because it is then dominated by the first term of this bound. (However, when we are not guaranteed that each desired cell contains a unique designating point, the last term may become dominant.) We thus conclude (the bound on the space complexity of the algorithm is straightforward to verify):

Theorem 3.2. *We can calculate m distinct cells in an arrangement of n planes in 3-space, when the cells are designated by a unique point in each, in time $O(m^{2/3} n \log^3 n + n^2 \log n)$ and space $O(m^{2/3} n \log n + n^2)$.*

4. Simpler Problems Involving Points and Planes

Let $\Pi = \{\pi_1, \ldots, \pi_n\}$ be a collection of n planes in three-dimensional space, and let $p_1, \ldots, p_m$ be m points. In this section we consider several variants of the many-cells problem studied above. Specifically, we consider the following problems:

(a) Determine for each point p_i the plane π_j lying immediately below it.
(b) Count the number of incidences between the given points and planes.
(c) Obtain an upper bound on the number of incidences between these points and planes, assuming that no three points are collinear (or, symmetrically, that no three planes pass through a common line).

Note that problem (a), and, with some extra care, problem (b) as well, can be solved by the algorithm of Section 2.3. However, as shown below, these variants (as well as problem (c)) admit solutions with bounds that are significantly lower than those obtained in the previous section.

We present in detail a solution to the first problem, comment briefly on the modifications necessary to obtain a similar solution to the second problem, and then analyze the third problem, which happens to have a better bound than the others.

We use an approach similar to that used in [EGS] for problems involving two-dimensional arrangements; this approach involves a divide-and-conquer strategy based on random sampling. Specifically, we pass to the dual space (using standard duality that preserves above/below relationship; see [E1]), so that each plane π_i is transformed to a point π_i^*, and each point p_j becomes a plane p_j^*.

We choose a random sample of r dual planes p_j^* (for some large but fixed integer r), form their arrangement, and decompose its cells into tetrahedra with pairwise disjoint interiors; the number of tetrahedra is $O(r^3)$. For each resulting tetrahedron τ we create a recursive subproblem involving the subset Π_τ of all planes in Π whose dual points lie in τ and the subset P_τ of all points of P whose dual planes cut τ. (If some dual point π_i^* lies on a tetrahedron boundary, we can assign its corresponding plane to any of the adjacent tetrahedra without affecting the behavior of the algorithm.) The subproblem associated with τ is to determine, for each p in P_τ, the plane in Π_τ lying immediately below p. (For problem (b), the subproblem is to count the number of incidences between these points and planes.)

The ε-net theory of [HW] or the similar probabilistic analysis of Clarkson [Cl2], implies that, with high probability, every such tetrahedron will be cut by at most $(cm/r) \log r$ dual planes, for some positive constant c.

We stop the recursion if $m_\tau \geq n_\tau^3$, where $m_\tau = |P_\tau|$ and $n_\tau = |\Pi_\tau|$. In this case the subproblem at τ is easily solved by constructing the full arrangement of the set Π_τ and by locating in it each of the m_τ points of P_τ, using the point-location algorithm of [EOS] or [Ch]. This can be accomplished in time $O(m_\tau \log^2 n_\tau)$.

Otherwise, when $m_\tau < n_\tau^3$, let us denote by B_τ the set of all points in P whose dual planes do not cut τ. Let p_i be a point in B_τ. Then the plane p_i^* passes either above all the points π_j^* for $\pi_j \in \Pi_\tau$ or below all of them. By duality, the point p_i lies either above all the planes $\pi_j \in \Pi_\tau$ or below all of them. In the latter case, no plane in Π_τ can be the plane lying directly below p_i, so these points need not be further processed at τ. Points p_i of the first type lie in the upper unbounded cell c of the arrangement $A(\Pi_\tau)$, and the plane of Π_τ lying directly below p_i is the one containing the face of c lying directly below p_i. We thus calculate the cell c in $O(n_\tau \log n_\tau)$ time (as in [PM]), and then project the boundary of c, as well as all points $p_i \in B_\tau$ of the first type, onto the xy-plane. We then locate each of these projected points in the planar map obtained by vertically projecting the boundary of c, thereby obtaining the plane of Π_τ lying directly below p_i. Thus this step can be accomplished in $O((b_\tau + n_\tau) \log n_\tau)$ time, where b_τ is the size of B_τ.

After solving all subproblems recursively, and performing the above step for each τ, we complete the procedure by taking, for each point p_i in P, the highest plane among those obtained at each of the tetrahedra τ, where at each τ we obtain such a candidate plane either from the recursive processing at τ, if $p_i \in P_\tau$, or from the above processing of B_τ, if p_i belongs to that set. This final step can clearly be performed in $O(m)$ time.

Let $T(m, n)$ denote the maximum time required by the algorithm for an input of m points and n planes, under the assumption that each recursive random sampling step does indeed produce a good partition of the dual space (see below for a discussion of this issue). Taking also into account the overhead needed for the partitioning (and bearing in mind that r is a constant), we thus obtain the following recurrence formula for $T(m, n)$:

$$T(m, n) \leq \begin{cases} am \log^2 n & \text{if } m \geq n^3, \\ \sum_{i=1}^{M} T(m_i, n_i) + (bm + b'n) \log n & \text{if } m < n^3, \end{cases}$$

for some constants $a, b, b' > 0$ (where b, b' depend on r), so that, by the preceding arguments, the m_i, n_i, and M can be assumed to satisfy the following three conditions:

(i) $M = O(r^3)$;
(ii) $\sum_{i=1}^{M} n_i \leq n$;
(iii) $m_i \leq (cm/r) \log r$ for each i, for some constant $c > 0$ (independent of r).

Before stating our results, we need to discuss the probabilistic aspects of the above algorithm. Even though there is high probability of choosing a good random sample at any single recursive step, the accumulated probability of successful drawings at all recursive steps can become small, unless the sample size increases from level to level, eventually becoming nonconstant near the bottom of recursion and thereby increasing the overhead of processing a single node. An alternative approach, which is the one we follow, is to verify, at each step of the algorithm, that the currently sampled subset of dual planes p_i^* is indeed a good sample, namely that

no cell in its triangulated arrangement is cut by more than $O((m/r) \log r)$ dual planes. This can be easily verified in linear time, assuming r to be a constant. Thus, with no increase in the asymptotic complexity, we can verify that the sample is good. If the sample is bad, we simply discard it and choose another. Again, the results of [Cl2] and [HW] imply that the expected number of draws until a good sample is obtained is constant. Employing this approach, we obtain a randomized algorithm, whose expected time complexity is proportional to $T(m, n)$ defined above, and which has the property that at each recursive step the resulting sample and corresponding space partitioning are good.

Returning to the analysis, we have

Theorem 4.1. *We can determine which of n given planes lies directly below each of m given points, using a randomized algorithm whose expected running time is bounded by $[Dm^{3/4-\delta}n^{3/4+3\delta} + Am] \log^2 n + Bn \log n \log m$, for any $\delta > 0$, where the coefficients A, B, D depend on δ. (Here the expectation is over the randomizations performed by the algorithm, and does not depend on the input.)*

Proof. If we follow the recurrence given above for $T(m, n)$ through the resulting recursion tree, the contribution of the rightmost term $b'n_\tau \log n_\tau$, over all tetrahedra τ at the same recursive level, is at most $b'n \log n$ (see the analysis in [EGS] for a similar argument). Since the tree has only $\log m$ levels of recursion, it follows that the overall contribution of these terms is $O(n \log n \log m)$. It is therefore sufficient to drop this term from the recurrence defining $T(m, n)$, and prove that the solution T of the modified recurrence satisfies $T(m, n) \leq [Dm^{3/4-\delta}n^{3/4+3\delta} + Am] \log^2 n$ for any $\delta > 0$.

Fix $0 < \delta < \frac{1}{12}$ and choose $r = r(\delta)$ to be sufficiently large (how large will be apparent from subsequent analysis).

If $m \geq n^3$, then $T(m, n) \leq am \log^2 n$ plainly satisfies the required inequality, assuming $A \geq a$. So assume $m < n^3$. In this case

$$m = m^{3/4-\delta} \cdot m^{1/4+\delta} \leq m^{3/4-\delta}n^{3/4+3\delta}. \qquad (*)$$

By induction hypothesis we then have

$$T(m, n) \leq \sum_{i=1}^{M} [Dm_i^{3/4-\delta}n_i^{3/4+3\delta} + Am_i] \log^2 n_i + bm \log n.$$

But because of (iii) we have $\sum_1^M m_i \log^2 n_i \leq ((Mc/r) \log r) \cdot m \log^2 n$, thus

$$T(m, n) \leq \left[D \sum_1^M m_i^{3/4-\delta}n_i^{3/4+3\delta} + \left(A \frac{Mc}{r} \log r + b \right) m \right] \log^2 n.$$

Thus, using (*) and putting $d = A(Mc/r)\log r + b$, we obtain

$$T(m, n) \le \left[D \sum_1^M m_i^{3/4-\delta} n_i^{3/4+3\delta} + dm^{3/4-\delta} n^{3/4+3\delta} \right] \log^2 n$$

$$\le \left[D \frac{c^{3/4-\delta}}{r^{3/4-\delta}} \log^{3/4-\delta} r \cdot m^{3/4-\delta} \cdot \sum_1^M n_i^{3/4+3\delta} + dm^{3/4-\delta} n^{3/4+3\delta} \right] \log^2 n.$$

But since Hölder's inequality implies

$$\sum_1^M n_i^{3/4+3\delta} \le \left(\sum_1^M n_i \right)^{3/4+3\delta} \cdot M^{1/4-3\delta} \le M^{1/4-3\delta} \cdot n^{3/4+3\delta}$$

and since $M \le c'r^3$ for some constant c', we obtain, for an appropriate constant c'',

$$T(m, n) \le \left[Dc'' \frac{\log^{3/4-\delta} r}{r^{8\delta}} + d \right] \cdot m^{3/4-\delta} n^{3/4+3\delta} \log^2 n.$$

Thus, if we choose r sufficiently large so as to make the expression in the brackets less than $D/2 + d$, and choose $D = 2d$, we obtain

$$T(m, n) \le Dm^{3/4-\delta} n^{3/4+3\delta} \log^2 n$$

thus establishing the asserted inequality. □

Remark. The preceding theorem implies that $T(m, n) = O(m^{3/4-\delta} n^{3/4+3\delta})$ for any $\delta > 0$, provided neither m nor n is too small. This is significant because, as shown in [EH] and noted in Section 2, the worst-case total complexity of the m cells containing the given points can be $\Omega(m^{2/3}n)$, which is greater than $O(m^{3/4}n^{3/4})$ for any $m = o(n^3)$. In other words, we have shown that finding just the plane lying immediately below each of m given points in an arrangement of n planes can be accomplished faster (albeit in a randomized manner) than calculating the entire cell around each point. While this result seems intuitively plausible, it is noteworthy that in the two-dimensional case no algorithm is known as yet for this seemingly simpler task, which is asymptotically faster than the algorithm for calculating the entire cells (see [EGS] and [EGH*]).

We next consider the other problems listed at the beginning of the section. The solutions to these problems follow closely that of problem (a) given above, and differ from it only in three key substeps: processing the data at the bottom of recursion, processing the sets B_τ, and combining output from recursive subproblems to obtain the desired solution for the whole problem.

Consider problem (b), in which we want to count the number of incidences between m given points and n given planes. For a tetrahedron τ at the bottom of recursion, we calculate the entire arrangement of the planes in Π_τ, assign to each

(zero-, one-, or two-dimensional) face of the arrangement the number of planes containing that face (this number is always 1 for two-dimensional faces), and then locate in the arrangement the m_τ points of P_τ. If a point happens to lie on a low-dimensional face f, we add to the total incidence count the number of planes containing f. As above, all this can be accomplished in time $O(m_\tau \log^2 n_\tau)$.

When processing a set B_τ associated with some tetrahedron τ, we calculate the upper and lower envelopes of the n_τ planes in Π_τ, and project them onto the xy-plane to obtain two planar maps M^+, M^-. We also maintain, for each vertex or edge of these maps, the number of planes in Π_τ passing through that vertex or edge. We then project each point p in B_τ onto the xy-plane and locate its projection p^* in, say, M^+. We can then easily determine whether p lies on the corresponding upper envelope U or not. If not, p (coupled with the planes in Π_τ) contributes nothing to the total incidence count. Otherwise, if p lies on a two-dimensional face of U, it contributes 1 to the total count; if it lies on an edge or coincides with a vertex of U, we add to the total count the number of planes in Π_τ passing through that edge or vertex. We repeat this procedure for M^-. It is clear that this processing can be accomplished in time $O((b_\tau + n_\tau) \log n_\tau)$.

Finally, combining the output from the recursive subproblems is trivial. We simply add up the incidence counts obtained recursively for each subproblem, as well as the incidence counts for points in the corresponding set B_τ. This takes only constant time.

It can now be easily checked that the time complexity of the resulting algorithm obeys the same recurrence as the algorithm for problem (a). Thus problem (b) can be solved within the same randomized expected time complexity given in Theorem 4.1. That is,

Theorem 4.2. *We can count the total number of incidences between m given points and n given planes in three-dimensional space, in randomized expected time $O([m^{3/4-\delta}n^{3/4+3\delta} + m] \log^2 n + n \log n \log m)$ for any $\delta > 0$.*

Problem (c), namely that of obtaining an upper bound on the number of incidences between points and planes, is handled in a way similar to that of problem (b), except that now we are not concerned with the time complexity of the algorithm, but rather with estimating the incidence count. In addition, the assumption that no three points are collinear turns out to be quite strong, and yields better bounds than those obtained for the previous problems. (However, some assumption like this has to be made to avoid a trivial bound; otherwise we could have all of our points lie on a common line and all given planes passing through that line, yielding an incidence count of mn, matching the trivial upper bound. Note that, in contrast, the counting algorithm presented above will handle this case efficiently.)

As above, we follow a recursive partitioning scheme for the dual points and planes, with the difference that the recursion stops this time at tetrahedra τ for which $m_\tau \geq n_\tau^2$. At each such τ consider the arrangement $A = A(\Pi_\tau)$ of the n_τ planes in Π_τ. If a point $p \in P_\tau$ does not lie on any of these planes it contributes nothing to the total incidence count. If it lies in (the relative interior of) a two-dimensional

face of A, then it contributes just 1 to that count. Finally, there are $O(n_\tau^2)$ intersection lines containing all vertices and edges of A, and none of them contains more than two of the given points. If such a line l lies in k of the planes of Π_τ, it contributes at most $2k$ to the incidence count, but then we can regard l as $\binom{k}{2}$ distinct lines, each formed by the intersection of two of these planes. This (rather standard) argument clearly implies that the total incidence count at τ is at most $O(n_\tau^2 + m_\tau) = O(m_\tau)$.

In processing the set B_τ of points whose dual planes have missed τ, we consider, as above, the upper and lower envelopes of the n_τ planes in Π_τ, and project them onto the xy-plane to obtain two planar maps M^+, M^-. Each point p in B_τ lies either on or above the upper envelope M^+, or on or below the lower envelope M^-. Clearly it suffices to consider only points lying on these envelopes. If such a point p lies in (the relative interior of) some face of, say M^+, it contributes just 1 to the total incidence count. On the other hand, these envelopes have $O(n_\tau)$ edges (even when each edge is counted with multiplicity equal to the number of planes containing it), and each can contain at most two points. Thus the overall contribution of B_τ to the incidence count is $O(b_\tau + n_\tau)$, where b_τ is the cardinality of B_τ.

Finally, bounding the overall incidence count is trivially done by summing up the incidence counts computed recursively, and adding the contributions to this count by points in the sets B_τ, for the corresponding tetrahedra τ. Hence if $I(m, n)$ denotes the maximum number of incidences between m points and n planes, with no three points collinear, we obtain the following recurrence for $I(m, n)$:

$$I(m, n) = \begin{cases} O(m) & \text{if } m \geq n^2, \\ \sum_{i=1}^{M} I(m_i, n_i) + O(m + n) & \text{if } m < n^2, \end{cases}$$

where we can assume that the m_i, n_i, and M satisfy conditions (i)–(iii) given above. By modifying the proof of Theorem 4.1, we easily obtain

Theorem 4.3. *The maximum number of incidences between m points and n planes, provided no three points are collinear, is $O(m^{3/5-\delta}n^{4/5+2\delta} + m + n \log m)$ for any $\delta > 0$.*

It is interesting to relate the above bound to the $O(m^{3/5}n^{4/5} + m + n)$ bound for the number of incidences between m points and n circles in the plane [CEG*]. We can map points in two dimensions into points in three dimensions, by lifting them to the paraboloid $z = x^2 + y^2$, and circles in two dimensions to planes in three dimensions, by lifting each circle to the paraboloid and taking the plane passing through the lifted circle. This gives an incidence problem for m points (no three collinear) and n planes, which shows that the bound in Theorem 4.3 also applies to incidences between points and circles in two dimensions. As can be seen, this bound is, however, slightly weaker than the bound of [CEG*] but then it is for a more general problem (note that in the above transformation to three dimensions, all resulting points lie on the paraboloid $z = x^2 + y^2$).

Remark. If, instead of assuming that no three points are collinear, we require that no three planes pass through a common line, we can use duality to transform the problem to that just studied, resulting in the following:

Corollary 4.4. *The maximum number of incidences between m points and n planes, provided no three planes pass through a common line, is* $O(m^{4/5+2\delta}n^{3/5-\delta} + m \log n + n)$ *for any* $\delta > 0$.

5. The Complexity of Many Cells in an Arrangement of Hyperplanes

In this section we extend the results of Section 2 to arrangements of hyperplanes in d dimensions, for any $d > 3$, obtaining upper bounds on the complexity of m distinct cells in arrangements of n hyperplanes. Here the complexity of a cell is defined as the number of facets (i.e., $(d-1)$-dimensional faces) bounding the cell. The bounds that we obtain again almost match the lower bounds given in [E1]. Specifically, we show

Theorem 5.1. *The complexity of m distinct cells in an arrangement of n hyperplanes in* $d \geq 3$ *dimensions is* $O(m^{2/3}n^{d/3} \log n + n^{d-1})$.

Proof. The proof proceeds by induction on the dimension d, but is otherwise quite similar to the analysis in Section 2. The basis case $d = 3$ has been established in Section 2. Assume that $d > 3$ and that the claim has been established for all $d' < d$. Let $\Pi = \{\pi_1, \ldots, \pi_n\}$ be a collection of n hyperplanes in d dimensions, and let $p_1, \ldots, p_m$ be m points lying in m distinct cells of the arrangement $A = A(\Pi)$ of these hyperplanes. As before, our task is to bound the complexity of the cells of $A(\Pi)$ that contain these points.

Again, we partition Π into two subsets Π_1, Π_2, each containing roughly $n/2$ hyperplanes, and refer to the hyperplanes in these subsets as the red hyperplanes and the blue hyperplanes, respectively. We obtain recursively the cells containing the points p_i in each of the subarrangements $A(\Pi_1)$, $A(\Pi_2)$, and then intersect these cells to obtain the desired cells of $A(\Pi)$. Again, if a point p lies in a red cell R and in a blue cell B, then the final cell E containing p is simply $R \cap B$, and since all these cells are convex polyhedra, the complexity of E is bounded by the sum of the complexities of R and of B (in $d > 3$ dimensions, this argument holds only for the number of $(d-1)$-dimensional faces, which is our complexity measure). Since R (or B) might contain more than one point p_i, we again need to establish a combination lemma that controls the complexity of the "merged" cells.

5.1. *The Combination Lemma for Hyperplanes*

Lemma 5.2 (Combination Lemma for Arrangements of Hyperplanes). *Given an arrangement of* n_r *red hyperplanes in d-dimensional space,* $d \geq 3$, *another arrangement of* n_b *blue hyperplanes, and m points not lying on any of these hyperplanes, the total number of facets bounding the "purple" cells containing these points in the*

arrangement formed by all $n = n_r + n_b$ hyperplanes, is at most

$$\rho + \beta + O(m^{2/3}n^{d/3} \log n + n^{d-1}),$$

where ρ (resp. β) is the total number of facets bounding the cells of the red (resp. blue) arrangement containing these points.

Proof. Again we proceed by induction on d, with the basis case $d = 3$ established in Section 2. Let $p_1, \ldots, p_m$ be m points in d-dimensional space. We are given two arrangements of hyperplanes, the red arrangement, consisting of n_r red hyperplanes, and the blue arrangement, consisting of n_b blue hyperplanes. Let $R_1, \ldots, R_s$ be the cells in the red arrangement containing the points p_i (where a cell can contain more than one point), and let $B_1, \ldots, B_t$ be the blue cells containing these points. For each p_i let $E_i = R_{s_i} \cap B_{t_i}$ be the "purple" cell containing p_i (where R_{s_i}, B_{t_i} are the red and blue cells containing the point p_i). By considering only a subset of the points p_i, if necessary, we can assume, without loss of generality, that all the cells E_i are distinct (and thus have pairwise disjoint interiors).

In complete analogy with the analysis in Section 2, we obtain the purple cells E_i by starting with the red cells R_j, and by adding the blue hyperplanes in a fixed order, thereby decomposing each red cell into "red-blue" subcells that contain the given points; when all blue hyperplanes are added, the final red-blue cells are the desired cells E_i. We apply a symmetric decomposition process starting with the blue cells B_j and adding the red hyperplanes, to obtain "blue-red" subcells, which again, when all red hyperplanes are added, coincide with the purple cells. The complexity of the purple cell is bounded by the "red complexity" of the red-blue cells plus the "blue complexity" of the blue-red cells. Again, our task is to bound the increase in the red complexity caused by adding the blue hyperplanes, and the symmetric increase in blue complexity of blue-red cells.

Consider the decomposition caused by the ith blue hyperplane π_i. Let R_j be a red cell that has already been split into a number of red-blue subcells, and assume that π_i splits k of these subcells (in the sense of splitting the sets of points they contain). If π_i intersects a red or red-blue cell but does not split the set of points in this cell, then, as argued in Section 2, this intersection cannot cause an increase in the red complexity. Since the red-blue subcells are convex polyhedra by construction, the intersection of π_i with each of them is a convex $(d-1)$-polyhedron q (where these polyhedra have pairwise disjoint interiors), contained in the intersection $c = c_j = \pi_i \cap R_j$ which itself is a convex $(d-1)$-polyhedron. To each facet f of a red-blue subcell which is split into two subfacets by π_i, there corresponds a $(d-2)$-subfacet of some facet of c (which is the intersection of f with π_i). Thus it follows easily that the total increase I in the red complexity of the red-blue subcells within R_j as caused by π_i is equal to the number of "red" facets of the red-blue intersection polyhedra q, namely facets that lie along the boundary of c. Let k_i denote the number of red-blue cell splitting caused by π_i; again we have $\sum_i k_i \leq m$.

Let A_i be the arrangement of the collection of the n_r $(d-2)$-flats formed by intersecting π_i with each of the n_r red hyperplanes, and of the $i-1$ $(d-2)$-flats formed by intersecting π_i with the preceding $i-1$ blue hyperplanes. Let I_i denote

the increase in the red complexity of all k_i red-blue cells split by π_i. Then the preceding argument implies that I_i is equal to the "red complexity" of k_i cells in A_i, where these cells are the intersections of π_i with the red-blue cells it splits. But the red complexity of these cells is bounded by their total complexity. Since this arrangement has at most n $(d-2)$-flats, it follows, by induction hypothesis on Theorem 5.1 in $d-1$ dimensions, that the complexity in question is at most

$$O(k_i^{2/3} n^{(d-1)/3} \log n + n^{d-2}).$$

Summing these quantities over all n_b blue hyperplanes, the total increase in the red complexity is therefore

$$O\left(\left(\sum_{i=1}^{n_b} k_i^{2/3}\right) \cdot n^{(d-1)/3} \log n + n^{d-1}\right)$$

which, as in Section 2, is

$$O(m^{2/3} n^{d/3} \log n + n^{d-1}).$$

We now obtain in a completely symmetric manner a partitioning of the blue cells into a collection of "blue-red" subcells, each containing just a single point, and derive a symmetric bound on their overall "blue" complexity. Arguing as in Section 2, we obtain the assertion of the combination lemma. □

5.2. *The Complexity of Many Cells*

We now continue the proof of Theorem 5.1. Let $K^{(d)}(m, n)$ denote the maximum complexity of m distinct cells in an arrangement of n hyperplanes in d dimensions. Applying the combination lemma to the two subarrangements $A(\Pi_1)$, $A(\Pi_2)$, we readily obtain a recurrence of the form

$$K^{(d)}(m, n) = 2K^{(d)}(m, n/2) + O(m^{2/3} n^{d/3} \log n + n^{d-1}).$$

Again, we do not continue this recursive process all the way down. When n first becomes smaller than $m^{1/d}$ we use the trivial bound $K^{(d)}(m, n) = O(n^d) = O(m)$. When this happens we have $m \leq (2n)^d$; it follows that $m = m^{2/3} m^{1/3} = O(m^{2/3} n^{d/3})$. It is now easy to verify that, since $d/3 > 1$, the solution to the above recurrence is

$$K^{(d)}(m, n) = O(m^{2/3} n^{d/3} \log n + n^{d-1}).$$

This completes the inductive proof of Theorem 5.1. □

Corollary 5.3. *The maximum number of incidences between m points and n hyperplanes in d dimensions, where it is assumed that each point is a vertex of the arrangement of these hyperplanes, is $O(m^{2/3} n^{d/3} \log n + n^{d-1})$.*

Proof. Identical to that of Corollary 2.3. □

Remarks. (1) The bound given in Corollary 5.3 almost matches the lower bound $\Omega(m^{2/3}n^{d/3} + n^{d-1})$ given in [E1]. It would be nice to get rid of the log n factor in our upper bound; we note that if suffices to do so in three dimensions, since in higher dimensions the recurrence for $K^{(d)}(m, n)$ will not introduce this factor.

(2) An obvious open problem is whether the bound in Theorem 5.1 also applies to the number of faces of any dimension bounding the given m cells. Recent results in [E2] lend hope to an affirmative solution to this problem.

(3) Again, can this analysis be modified to yield an algorithm which actually calculates the facets bounding m cells in an arrangement of n hyperplanes in $d > 3$ dimensions? This seems to be a considerably more difficult problem than the corresponding problem for three dimensions, because it calls, among other things, for a procedure which produces the facets bounding the intersection of two convex polyhedra in time proportional to the number of output facets, and no such procedure seems to be available.

(4) Note that our divide-and-conquer approach can be made into an efficient algorithm for calculating all incidences between m points and n hyperplanes. The general recursive step is trivial, since all we need to do is to sum up the incidences calculated in the two subproblems. At the bottom of the recursion, when $m \geq n^d$, we need a procedure that finds the incidences between m points and n hyperplanes in time close to $O(n^d)$. This can be accomplished using Clarkson's randomized point-location algorithm in arrangements of hyperplanes (see [Cl1]).

Acknowledgments

The authors would like to thank Boris Aronov for helpful discussions, and the referees for their detailed and critical comments that helped to improve the presentation of this paper. Part of the work on this paper has been carried out at DEC Systems Research Center, and the authors would like to express their gratitude for its hospitality.

References

[A] P. K. Agarwal, Intersection and decomposition algorithms for arrangements of curves in the plane, Ph.D. Dissertation, Computer Science Department, New York University, June 1989.

[AS] B. Aronov and M. Sharir, Triangles in space, or: Building and analyzing castles in the air, *Proc. 4th ACM Symp. on Computational Geometry*, 1988, pp. 381–391.

[Ca] R. J. Canham, A theorem on arrangements of lines in the plane, *Israel J. Math.* **7** (1969), 393–397.

[Ch] B. Chazelle, How to search in history, *Inform. and Control* **64** (1985), 77–99.

[Cl1] K. Clarkson, A probabilistic algorithm for the post office problem, *Proc. 17th ACM Symp. on Theory of Computing*, 1985, pp. 75–84.

[Cl2] K. Clarkson, New applications of random sampling in computational geometry, *Discrete Comput. Geom.* **2** (1987) 195–222.

[CEG*] K. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir, and E. Welzl, Combinatorial complexity bounds for arrangements of curves and surfaces, *Proc. 29th IEEE Symp. on Foundations of Computer Science*, 1988, pp. 568–579. (Also, *Discrete Comput. Geom.*, this issue, 99–160.)

[DK] D. Dobkin and D. Kirkpatrick, Fast detection of polyhedral intersections, *Theoret. Comput. Sci.* **27** (1983), 241–253.

[E1] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer-Verlag, Heidelberg, 1987.

[E2] H. Edelsbrunner, The upper envelope of piecewise linear functions: Tight bounds on the number of faces, *Discrete Comput. Geom.* **4** (1989), 337–343.

[EGH*] H. Edelsbrunner, L. Guibas, J. Hershberger, R. Seidel, M. Sharir, J. Snoeyink, and E. Welzl, Implicitly representing arrangements of lines or segments, *Proc. 4th ACM Symp. on Computational Geometry*, 1988, pp. 56–69. (Also, *Discrete Comput. Geom.* **4**, (1989), 433–466.)

[EGS] H. Edelsbrunner, L. Guibas, and M. Sharir, The complexity of many faces in arrangements of lines or segments, *Proc. 4th ACM Symp. on Computational Geometry*, 1988, pp. 44–55. (Also, *Discrete Comput. Geom.*, this issue, 161–196.)

[EH] H. Edelsbrunner and D. Haussler, The complexity of cells in three-dimensional arrangements, *Discrete Math.* **60** (1986), 139–146.

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

[EW] H. Edelsbrunner and E. Welzl, On the maximal number of edges of many faces in an arrangement, *J. Combin. Theory Ser. A* **41** (1986), 159–166.

[HW] D. Haussler and E. Welzl, Epsilon-nets and simplex range queries, *Discrete Comput. Geom.* **2** (1987), 127–151.

[PM] F. P. Preparata and D. E. Muller, Finding the intersection of n half space in time $O(n \log n)$, *Theoret. Comput. Sci.* **8** (1979), 44–55.

[PE] G. Purdy and P. Erdős, Some extremal problems in combinatorial geomery, manuscript, 1987.

[ST] E. Szemerédi and W. T. Trotter, Extremal problems in discrete geometry, *Combinatorica* **3** (1983), 381–392.

Received December 1, 1988, and in revised form July 10, 1989.