Algorithmica (1996) 15: 428–447

Algorithmica

Lines in Space: Combinatorics and Algorithms[1]

B. Chazelle,[2] H. Edelsbrunner,[3] L. J. Guibas,[4] M. Sharir,[5] and J. Stolfi[6]

Abstract. Questions about lines in space arise frequently as subproblems in three-dimensional computational geometry. In this paper we study a number of fundamental combinatorial and algorithmic problems involving arrangements of n lines in three-dimensional space. Our main results include:

1. A tight $\Theta(n^2)$ bound on the maximum combinatorial description complexity of the set of all oriented lines that have specified orientations relative to the n given lines.
2. A similar bound of $\Theta(n^3)$ for the complexity of the set of all lines passing above the n given lines.
3. A preprocessing procedure using $O(n^{2+\varepsilon})$ time and storage, for any $\varepsilon > 0$, that builds a structure supporting $O(\log n)$-time queries for testing if a line lies above all the given lines.
4. An algorithm that tests the "towering property" in $O(n^{4/3+\varepsilon})$ time, for any $\varepsilon > 0$: do n given red lines lie all above n given blue lines?

The tools used to obtain these and other results include Plücker coordinates for lines in space and ε-nets for various geometric range spaces.

Key Words. Computational geometry, Lines in space, Plücker coordinates, ε-Nets.

1. Introduction.

In this paper we address certain combinatorial and algorithmic problems about lines in three-dimensional Euclidean space. Algorithmic questions about lines in three dimensions arise in numerous applications, including the hidden surface removal and ray tracing problems in computer graphics, motion planning, placement and assembly problems in robotics, object recognition using three-dimensional range data in computer vision, interaction of solids and of surfaces in solid modeling and CAD, and terrain analysis and reconstruction in geography.

Though the geometry of lines in the plane is a well-studied part of computational geometry, the corrresponding investigation of lines in three-dimensional space is a relatively new area. Progress on three-dimensional problems is still relatively slow, mainly because these problems are often harder than their planar counterparts. Typically, the combinatorial structure of the geometric space of interest is more complicated and has larger complexity in the spatial case than in the planar case. Thus, efficient algorithms

[1] Work on this paper by Bernard Chazelle has been supported by NSF Grant CCR-87-00917. Work on this paper by Herbert Edelsbrunner has been supported by NSF Grant CCR-87-14565. Work on this paper by Leonidas Guibas has been supported by grants from the Mitsubishi and Toshiba Corporations. Work on this paper by Micha Sharir has been supported by ONR Grant N00014-87-K-0129, by NSF Grants DCR-83-20085 and CCR-89-01484, and by grants from the U.S.–Israeli Binational Science Foundation, the NCRD—the Israeli National Council for Research and Development, and the EMET Fund of the Israeli Academy of Sciences.
[2] Department of Computer Science, Princeton University, Princeton, NJ 08544, USA.
[3] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA.
[4] Stanford University, Stanford, CA 94305, USA and DEC Systems Research Center, Palo Alto, CA 94301, USA.
[5] Tel Aviv University, Ramat Aviv 69 978, Israel and New York University, New York, NY 10012, USA.
[6] DEC Systems Research Center, Palo Alto, CA 94301, USA.

Received November 27, 1993; revised August 17, 1994. Communicated by D. P. Dobkin.

for processing it are more difficult to obtain. Straight lines, one of the simplest types of objects encountered in spatial problems, already present many of these difficulties. In fact, as we will see below, lines in space are modeled best by nonlinear objects. For a classical treatment of the geometry of lines in 3-space see the book by Sommerville [So], or various kinematics texts [BR], [Hu].

In this paper we make contributions to three-dimensional computational geometry by studying several combinatorial problems involving *arrangements of lines in space*. By the arrangement of a given set of lines we mean the partitioning of the space of all lines introduced by the given lines. We first provide a combinatorial and algorithmic analysis of what we call an *orientation class* of a collection of lines in space, i.e., the topological boundary of the space of lines having a given orientation with all the given lines. We show how to express the "above/below" relationship of lines in space by means of the orientation relationship and use this reduction to analyze various problems concerning the vertical relationship of lines in space. Even though (as is discussed below) the "natural complexity" of an arrangement of n lines in space is $\Theta(n^4)$ (see [MO]), we are able to solve each of the problems that we consider in nearly quadratic space, or better. In a companion paper [CEGS1] we then apply these results to several important practical problems involving *polyhedral terrains* (i.e., images of piecewise-linear continuous bivariate real functions) and obtain reasonably efficient solutions.

In order to introduce and summarize our results in more detail here, we review some basic geometric properties of lines in 3-space. A line requires four real parameters to specify it, so it is natural to study arrangements of lines in space within an appropriate parametric 4-space. Unfortunately, any reasonable such representation introduces nonlinear surfaces. For example, in many standard parametrizations the space of all lines intersecting a given line is a quadric surface in 4-space. To obtain a combinatorial representation of an arrangement of n lines in space it is therefore necessary to construct an arrangement of n quadric surfaces in 4-space (in fact, the recent paper [MO] does provide an implicit construction of such an arrangement; see also [Mc]). This arrangement has complexity $O(n^4)$ (as follows, e.g., from a theorem of Milnor and Thom [Mi]) which is usually unacceptable for practical applications; moreover, even if we were to construct the arrangement, performing point-location (that is, "line-location") in it is difficult. These observations indicate why many of the recent works on visibility problems involving arbitrary collections of lines (or segments, or polyhedra) in space produce bounds like $O(n^4)$ or worse (see [PD], [GCS], [MO], and [Mc]).

Fortunately, there are two lucky breaks that we are able to exploit in this work, which lead to improved solutions in many applications. The first is that there is an alternative way to represent lines, using *Plücker coordinates* (see, e.g., [St], [BR], and [Hu]; the original reference is [Pl]). These coordinates transform (oriented) lines into either points or hyperplanes in homogeneous 6-space (more precisely, in oriented projective 5-space) in such a way that the property of one line l_1 intersecting another line l_2 is transformed into the property of the Plücker point of l_1 lying on the Plücker hyperplane of l_2 (or vice versa). Thus, at the cost of passing to five dimensions, we can linearize the incidence relationship between lines. This Plücker machinery is developed in Section 2. For completeness, we mention that another representation of lines in 3-space by two points in the plane, based on the *parallel coordinates* introduced by Inselberg [I], has also been found useful in practice.

In studying arrangements of lines in space, it is more important to analyze the *relative orientation* of two lines rather than only the incidence between them, as the latter is a degenerate case of the former. We develop this concept of relative orientation in Section 3 and show how to determine efficiently if a query line is of a particular orientation class with respect to n given lines. We give a method that takes preprocessing and storage of $O(n^{2+\varepsilon})$ and allows a query time of $O(\log n)$. In the process we show that the total combinatorial description complexity of any particular orientation class is in the worst case $\Theta(n^2)$. We get this bound by mapping our lines to hyperplanes in oriented projective 5-space using Plücker coordinates. Our orientation class then corresponds to a convex polyhedron defined by the intersection of n half-spaces based on these hyperplanes. Our second lucky break now comes from the Upper Bound Theorem (see, e.g., [Ed]), stating that the complexity of such a polyhedron is only $O(n^{\lfloor 5/2 \rfloor}) = O(n^2)$ (the same asymptotic order as in 4-space, which means that passing to five dimensions did not really cost us anything extra in terms or complexity).

For many applications, however, we need to analyze the property of one line lying above or below another. In Section 4 we show how, by adding certain auxiliary lines, we can express "above/below" relationships by means of orientation relationships. Using this reduction we provide an efficient method for testing if a query line lies above the n given lines. Specifically, we give an algorithm for preprocessing a collection $\mathcal{L}$ of n lines in space in $O(n^{2+\varepsilon})$ preprocessing time and storage. Our algorithm builds a data structure that supports $O(\log n)$ queries of the form: given a line l, does it lie above all the lines of $\mathcal{L}$? If so, which line of $\mathcal{L}$ lies "immediately below" l, i.e., what is the first line of $\mathcal{L}$ to be hit as l is translated downward? We also demonstrate in Section 5 that the worst-case combinatorial complexity of the "upper envelope" of n lines is $\Theta(n^3)$—the main observation being that such an envelope can be expressed as the union of n orientation classes of the kind discussed in the previous paragraph.

We also provide in Section 6 a *batched version* of the algorithm for testing the "above/below" relationship: Given m blue lines and n red lines, determine whether all blue lines lie above all red lines (we call this the "towering property"), and, if so, find for each blue line the red line lying immediately below it (in the above sense). We achieve this by an algorithm with running time $O((m+n)^{4/3+\varepsilon})$, for any $\varepsilon > 0$.

The algorithms just mentioned, as well as many others developed in this paper, are based on the recently introduced technique of ε-nets in computational geometry by Haussler and Welzl, and Clarkson (see [HW], [Cl1], and [CF]). Originally ε-nets were obtained by random sampling. The randomizations employed by the algorithms draw a small random sample of the data objects, and use it to partition the problem into smaller subproblems in a uniform manner. Even more recently, various efficient deterministic techniques for constructing ε-nets were obtained by Matoušek and others [Ma1]–[Ma3], [CF]; see also the survey paper by Agarwal [AG]. Using these methods all the algorithmic bounds given in this paper can be made to be worst-case bounds. The corresponding randomized versions would be easier to implement and probably preferable in any practical application.

We close the paper with a discussion of line separability by translation in Section 7, and by describing several open problems about lines in space in Section 8. We hope that this paper will stimulate further combinatorial and algorithmic work in three-dimensional line geometry.

2. Geometric Preliminaries. The main geometric object studied in this paper is a line in 3-space. Such a line l can be specified by four real parameters in many ways. For example, we can take two fixed parallel planes (e.g., $z = 0$ and $z = 1$) and specify l by its two intersections with these planes. We can therefore represent all lines in 3-space, except those parallel to the two given planes, as points in four dimensions. However, as already noted, even simple relationships between lines, such as incidence between a pair of lines, become nonlinear in 4-space. More specifically, a collection $\mathcal{L} = \{l_1, \ldots, l_n\}$ of n lines induces a corresponding collection of hypersurfaces $\mathcal{S} = \{s^1, \ldots, s^n\}$ in 4-space, where s^i represents the locus of all lines that intersect, or are parallel to, l_i (it is easily checked that each s^i is a quadratic hypersurface). The *arrangement* $\mathcal{A} = \mathcal{A}(\mathcal{S})$ induced by these hypersurfaces represents the arrangement of the lines in $\mathcal{L}$, in the sense that each (four-dimensional) cell of $\mathcal{A}$ represents an isotopy class of lines in 3-space (i.e., any such line in the class can be moved continuously to any other line in the same class without crossing, or becoming parallel to, any line in $\mathcal{L}$).

This arrangement can be understood in three dimensions as follows. Given three lines in general position[7] in 3-space, they define a quadratic ruled surface, called a *regulus*, which is the locus of all lines incident with the given three lines. A fourth line will in general cut this surface in two, or zero, points. Thus four lines in general position will have either two or zero lines incident with all four of them. These quadruplets of lines with a common stabber correspond to vertices of the arrangement $\mathcal{A}$. (In other words, a line moving within an isotopy class comes to rest when it is in contact with four of the given arrangement lines—each of them removing one of the four degrees of freedom that the moving line has. Note that some of these contacts can be at infinity, corresponding to the moving line becoming parallel to one of the given lines.) Similarly, the edges of $\mathcal{A}$ correspond to the motion of a line while incident with three given lines, in the regulus fashion described earlier. In the general case each vertex of $\mathcal{A}$ has eight incident edges. Higher-dimensional faces of $\mathcal{A}$ can be obtained similarly, by letting the common stabber move away from two, three, or all four of the lines defining a vertex. This shows that the number of these higher-dimensional faces of $\mathcal{A}$ is, in each case, related by at most a constant factor to the number of vertices of $\mathcal{A}$. This last statement remains valid even if the given lines are not in general position, as follows from a standard perturbation argument.

By the discussion in the preceding paragraph (or by invoking the theorem of Milnor and Thom [Mi], as mentioned in the Introduction) we can conclude that the combinatorial complexity of the arrangement of n quadratic surfaces in 4-space is $O(n^4)$ and this bound is attainable (see [MO]). In particular, $\mathcal{A}$ has $O(n^4)$ vertices, where each such vertex represents a line that meets four of the lines in $\mathcal{L}$. Unfortunately, it is difficult to handle such an arrangement of nonlinear hypersurfaces explicitly; tasks such as efficient calculation and representation of $\mathcal{A}$, processing it for fast point location,[8] and obtaining sharp complexity bounds for certain portions of it become quite difficult.

---

[7] We take this to mean that the lines are pairwise nonintersecting and nonparallel. For more lines we add the condition that no five of our lines can be simultaneously incident with another line (not necessarily of our collection).

[8] An efficient technique for point location among algebraic manifolds was recently given in [CEGS2]. However, that method requires $\Omega(n^5)$ space.

We therefore exploit another representation of lines, using *Plücker coordinates and coefficients* (see [St], [BR], and [Hu] for a review of these concepts). Let l be an oriented line, and let a, b be two points on l such that the line is oriented from a to b. Let $[a_0, a_1, a_2, a_3]$ and $[b_0, b_1, b_2, b_3]$ be the homogeneous coordinates of a and b, with $a_0, b_0 > 0$ being the homogenizing weights. (By this we mean that the Cartesian coordinates of a are $(a_1/a_0, a_2/a_0, a_3/a_0)$). By definition, the Plücker coordinates of l are the six real numbers

$$\pi(l) = [\pi_{01}, \pi_{02}, \pi_{12}, \pi_{03}, \pi_{13}, \pi_{23}],$$

where $\pi_{ij} = a_i b_j - a_j b_i$ for $0 \le i < j \le 3$. Similarly, the Plücker coefficients of l are

$$\varpi(l) = \langle \pi_{23}, -\pi_{13}, \pi_{03}, \pi_{12}, -\pi_{02}, \pi_{01} \rangle,$$

i.e., the Plücker coordinates listed in reverse order with two signs flipped. The most important property of Plücker coordinates and coefficients is that incidence between lines is a bilinear predicate. Specifically, l^1 is incident to l^2 if and only if their Plücker coordinates π^1, π^2 satisfy the relationship

$$\pi^1_{01}\pi^2_{23} - \pi^1_{02}\pi^2_{13} + \pi^1_{12}\pi^2_{03} + \pi^1_{03}\pi^2_{12} - \pi^1_{13}\pi^2_{02} + \pi^1_{23}\pi^2_{01} = 0. \tag{1}$$

This formula follows from expanding the four-by-four determinant whose rows are the coordinates of four distinct points a, b, c, d, with a, b on l^1 and c, d on l^2. This determinant is equal to 0 if and only if the two lines are incident (or parallel). In general, the absolute value of the quantity in (1) is six times the volume of the tetrahedron $abcd$,[9] and its sign gives the orientation of the tetrahedron $abcd$. As long as l^1 is oriented from a to b and l^2 from c to d, this sign is independent of the choice of the four points, and defines the *relative orientation* of the pair l^1, l^2, which we denote by $l^1 \diamond l^2$ [St].

It is easily checked that any positive scalar multiple of $\pi(l)$ is also a valid set of Plücker coordinates for the same oriented line l, corresponding to a different choice of the defining points a and b, or to a positive scaling of their homogeneous coordinates. Also, any negative multiple of $\pi(l)$ is a representation of l with the opposite orientation. Therefore, we can regard the Plücker coordinates $\pi(l)$ as the homogeneous coordinates of a point projective oriented 5-space $\mathcal{P}^5$, which is a double covering of ordinary projective 5-space.[10] Dually, we can regard the Plücker coefficients $\varpi(l)$ as the homogeneous coefficients of an oriented hyperplane of $\mathcal{P}^5$. Equation (1) merely states that line l^1 is incident to line l^2 if and only if the Plücker point $\pi(l^1)$ lies on the Plücker hyperplane $\varpi(l^2)$. In fact, the relative orientation $l^1 \diamond l^2$ of the two lines is $+1$ if $\pi(l^1)$ lies on the positive side of the hyperplane $\varpi(l^2)$, and -1 if it lies on the negative side.

We observe that not every point of $\mathcal{P}^5$ is the Plücker image of some line. It is well known that the real six-tuple (π_{ij}) is such an image if and only if it satisfies the

---

[9] It also equals the product $\overline{ab}\,\overline{cd}\,D \sin\alpha$, where D is the distance between l^1 and l^2, and α is the angle between the two lines.

[10] The points of $\mathcal{P}^5$ can be viewed as the oriented lines through the origin of $\Re^6$, with the geometric structure induced by the linear subspaces of $\Re^6$; or, equivalently, as the points of the five-dimensional sphere S^5, with the geometric structure induced by its great circles. See [St] for more details on the theory of oriented projective spaces.

quadratic equation

$$\pi_{01}\pi_{23} - \pi_{02}\pi_{13} + \pi_{12}\pi_{03} = 0, \tag{2}$$

which states that every line is incident to itself. Thus among the six Plücker coordinates *two* are redundant. Equation (2) defines a four-dimensional subset of $\mathcal{P}^5$, called the *Plücker hypersurface* Π. Notice that the relative orientation of a line relative to a sixtuplet of numbers that does not correspond to a point on the Plücker hypersurface still makes perfect sense—simply plug the appropriate numbers into (1). It turns out that such "imaginary" lines do have a natural geometric interpretation in 3-space. They are known as *linear complexes* and their properties are studied in [FT] and [J].

3. The Orientation of a Line Relative to n Given Lines. We wish to analyze the set $\overline{\mathcal{C}}(\mathcal{L},\sigma)$ consisting of all lines l in 3-space that have specified orientations $\sigma = (\sigma^1, \sigma^2, \ldots, \sigma^n)$ relative to n given lines in $\mathcal{L} = (l^1, l^2, \ldots, l^n)$. (We call this set the *orientation class* σ relative to $\mathcal{L}$.) Translated to Plücker space, the definition says that point $\pi(l)$ has to lie on side σ^i of every hyperplane $\varpi(l^i)$, and therefore inside the convex polytope $C(\mathcal{L},\sigma)$ in $\mathcal{P}^5$ that is the intersection of those n half-spaces. The orientation class $\overline{C}(\mathcal{L},\sigma)$ is thus the intersection of the polytope $C(\mathcal{L},\sigma)$ and the Plücker hypersurface Π. Note that since Π is of degree 2, it can interact in at most "a constant fashion" with each feature of the polytope $C(\mathcal{L},\sigma)$.

For the purpose of this paper, we consider the polytope $C(\mathcal{L},\sigma)$ to be an adequate description of the orientation class $\overline{\mathcal{C}}(\mathcal{L},\sigma)$. Computing the class then means computing all the features of this polytope, i.e., all its faces (of any dimension). The number of such features is the *combinatorial complexity* of the class—intersecting with Π can only increase this number by a constant factor. By the Upper Bound Theorem (see, e.g., [Ed]), this complexity is only $O(n^{\lfloor 5/2 \rfloor}) = O(n^2)$. It is not difficult to find configurations of lines $\mathcal{L}$ that attain this bound. Consider the regulus (actually hyperbolic paraboloid) $z = xy$ and two families of $n/2$ lines each of the regulus. One family consists of lines from one of the two rulings of the regulus, and the other of lines from the other ruling. By perturbing the lines of one family to be slightly off the regulus, we can make this a nondegenerate arrangement. It is simple to check that in every elementary square defined by two successive lines from one ruling and two successive lines from the other ruling there corresponds a line incident to all four of the lines defining the square and passing above all the rest. A more detailed construction of this kind is given in Section 5.

A possible data structure for representing the polytope $C(\mathcal{L},\sigma)$ is its face-incidence lattice, as described in [Ed]. Seidel's output-sensitive convex hull algorithm [Se] constructs this representation in $O(\log n)$ amortized time per face. The more recent optimal but complex convex hull algorithm of Chazelle [Ch] can also be used to obtain the polytope C in $O(n^2)$ time. As it turns out, in the algorithms to follow we only need to compute orientation classes for collections $\mathcal{L}$ whose size is bounded by a constant, so the representation issue does not arise in a significant way.

THEOREM 1. *The set of all lines in* 3*-space that have specified orientations to n given lines has combinatorial complexity* $\Theta(n^2)$ *in the worst case, and can be calculated in time* $O(n^2)$.

It was shown by Neil White (see [MO]) that the intersection of the convex polytope $C(\mathcal{L}, \sigma)$ and the Plücker hypersurface Π may consist of many connected components. In other words, an orientation class relative to the fixed lines $\mathcal{L}$ may contain multiple distinct isotopy classes. We note that the vertices of those isotopy classes are intersections of the Plücker hypersurface Π with the edges of the polytope $C(\mathcal{L}, \sigma)$. Since Π is a quadratic hypersurface, there are at most two such intersections per edge, and therefore the total number of vertices in all those isotopy classes is only $O(n^2)$. In other words, there are at most $O(n^2)$ lines that touch four of the lines of $\mathcal{L}$ and have specified orientations with all the others. A slightly more complicated argument shows that there are at most $O(n^2)$ isotopy classes in one orientation class. We do not know if this bound can be attained.

We now give an efficient algorithm for deciding whether a given query line l in 3-space lies in a particular orientation class σ relative to a set $\mathcal{L}$ of n fixed lines. We begin by preprocessing the fixed lines into a tree-like data structure $\Sigma(\mathcal{L}, \sigma)$, using a net-based partitioning technique that somewhat resembles those of [Cl2] and [CF]. For simplicity, we describe the construction for the class $\sigma = (+, +, \ldots, +)$; the same construction can be applied to other classes by reversing the orientation of the appropriate lines of $\mathcal{L}$. Consider the n Plücker hyperplanes that correspond to the given lines $\mathcal{L}$. We choose an ε-net $\mathcal{R}$ for simplex range queries among these hyperplanes, with some fixed size $r > 0$. We compute the open five-dimensional polytope $C(\mathcal{R}) = C(\mathcal{R}, +^r)$ that is the intersection of their positive half-spaces. Then we decompose $C(\mathcal{R})$ into a collection $\mathcal{K}(\mathcal{R})$ of open k-dimensional simplices, for $k \leq 5$, by picking a vertex v of $C(\mathcal{R})$, recursively triangulating all the faces of $C(\mathcal{R})$ that are not incident to v, and then taking the convex hull of the point v and each of these simplices. By the Upper Bound Theorem, $C(\mathcal{R})$ has only $O(r^2)$ faces, and $\mathcal{K}(\mathcal{R})$ contains only $O(r^2)$ simplices. The time required for these steps is dominated by $O(nr^4/\log^4 r)$, the cost of selecting the net deterministically [Ma3]. Since none of these simplices meets any of the hyperplanes of $\mathcal{R}$, it follows from the net property (see [HW] and [Cl1]) that each simplex in $\mathcal{K}(\mathcal{R})$ will meet at most $c(n/r)\log r$ of the n original hyperplanes, for some constant $c > 0$ independent of r and n.

We then proceed to discard any simplex of $\mathcal{K}(\mathcal{R})$ that lies entirely on the negative side of some of the n hyperplanes. Each surviving simplex s becomes a child of the root of our data structure; the subtree rooted at s consists of the $O((n/r)\log r)$ hyperplanes that intersect s, recursively preprocessed as described above. If all simplices get discarded, or if the polytope $C(\mathcal{R})$ was empty to begin with, then the orientation class is empty, and the problem is trivial: *no* query line can be positively oriented with respect to all lines in $\mathcal{L}$. (Note that the converse is not necessarily true.)

The storage and preprocessing time required in this technique obey the recursion

$$T(n) = O(r^2) \cdot T\left(C\frac{n}{r}\log r\right) + O\left(\frac{nr^4}{\log^4 r}\right)$$

for some constant C. It is not hard to prove that this solves to $T(n) = O(n^{2+\varepsilon})$, for some positive number ε that tends to 0 as r increases. (Note however that increasing r also increases the constant of proportionality.)

Testing a query line l proceeds as follows. We first test whether any of the $O(r^2)$ simplices at the top level of the tree contains the Plücker point $\pi(l)$. If so, we search recursively in the subtree rooted at that simplex. If not, then we know that l is not positively oriented relative to $\mathcal{L}$. There are only $O(\log n)$ levels to recurse in, so the worst-case query time is $O(\log n)$. Again, the constant of proportionality depends on r (or, alternatively, on ε).

THEOREM 2. *Given n lines in space and an orientation class σ, we can preprocess these lines by a procedure whose running time and storage is $O(n^{2+\varepsilon})$, for any $\varepsilon > 0$, so that, given any query line l, we can determine, in $O(\log n)$ time, whether l lies in the orientation class σ with respect to the given lines.*

We note that a simple modification of this data structure allows us to *compute* in $O(\log n)$ time the orientation class of a line l relative to n fixed ones, rather than merely test whether l is in a predetermined class. The modification consists in computing (and triangulating) the whole zone of the Plücker hypersurface Π in the arrangement of the net hyperplanes $\mathcal{R}$, rather than just the cell $C(\mathcal{R}, +^r)$. The complexity of this zone is $O(r^4 \log r)$ in the worst case, by a recent result [APS]. By an analysis similar to that given above, it follows that there is a structure of size $O(n^{4+\varepsilon})$ that can be used to compute the orientation class of a given line within the above time bound.

4. Testing Whether a Line Lies Above *n* Given Lines. We now consider a particular case of the general problem discussed in the previous section, which turns out to have significant applications on its own. We are concerned with the property of one line lying above or below another. Formally, l^1 lies above l^2 if there is a vertical line that meets both lines, and its intersection with l^1 is higher than its intersection with l^2. We assume that neither l^1 nor l^2 is vertical, and the two lines are not parallel. Our previous nondegeneracy assumptions already exclude concurrent or parallel lines; whenever we discuss the "above/below" relation, we also exclude vertical lines from consideration.

We can express this notion in terms of the relative orientation of these lines, as follows. Assume the lines l^1 and l^2 have been oriented in an arbitrary way, and consider their (oriented) perpendicular projections $1'$, $l^{2'}$ onto the xy-plane, seen from above. Observe that l^1 is above l^2 if and only if

$$\text{the direction of } l^{1'} \text{ is clockwise to that of } l^{2'} \text{ and } l^1 \diamond l^2 = +1,$$

or

$$\text{the direction of } l^{1'} \text{ is counterclockwise to that of } l^{2'} \text{ and } l^1 \diamond l^2 = -1.$$

Now we introduce the line at infinity λ^2 that is parallel to l^2 and passes through zenith point $z_\infty = (0, 0, 0, 1)$, the point at positive infinity on the z-axis. We orient the line λ^2 so that its projection on the xy-plane has the same direction as the projection of l^2. It is easy to check that the direction of $l^{1'}$ is clockwise of $l^{2'}$ if and only if $l^1 \diamond \lambda^2 = -1$. Therefore, we conclude that l^1 is above l^2 if and only if

$$l^1 \diamond l^2 = -l^1 \diamond \lambda^2. \tag{3}$$

Intuitively, l^1 passes above l^2 if and only if l^1 passes "between" the lines l^2 and λ^2.

Thus, to express the fact that one line lies above another we need to check consistency between two linear inequalities. This fact complicates the analysis of the above/below relationship, in particular when many lines are involved.

Now let $\mathcal{L}$ be a collection of n lines in 3-space, and consider the set $\mathcal{U}(\mathcal{L})$, *the upper envelope of* $\mathcal{L}$, consisting of all lines l that pass above every line of $\mathcal{L}$. We introduce the auxiliary lines at infinity $\Lambda = \{\lambda^1, \lambda^2, \ldots, \lambda^n\}$, with each λ^i parallel to the corresponding l^i and passing through the point z_∞. Then, according to (3), a line l is above all lines in $\mathcal{L}$ if and only if $l \diamond l^i = -l \diamond \lambda^i$; that is, if the orientation class of l relative to the set $\mathcal{L}$ is exactly opposite to its orientation with respect to the set Λ.

Therefore, the set $\mathcal{U}(\mathcal{L})$ is the union of all orientation classes $\overline{\mathcal{C}}(\mathcal{L} \cup \Lambda, \sigma \cdot \bar{\sigma})$ where $\sigma \cdot \bar{\sigma}$ is a sign sequence of the form $(\sigma^1, \sigma^2, \ldots, \sigma^n, -\sigma^1, -\sigma^2, \ldots, -\sigma^n)$. Luckily for us, only n of these classes are nonempty. To see why, we assume that the x and y coordinate axes have been rotated and the lines oriented so that the projection of l^1 coincides with the negative y-axis, and all other lines (including the query line l) point toward increasing x. We assume also that the lines $l^2, \ldots, l^n$ are sorted in order of increasing xy-slope. It is easy to see that if the xy slope of l lies between those of l^k and l^{k+1}, then its orientation class relative to the set Λ is $(-^k+^{n-k})$. Therefore, we conclude that there are only n orientation classes relative to Λ.

This observation leads to a fast algorithm for deciding whether a query line l passes above n fixed lines $\mathcal{L}$. For each of the n valid orientation classes $\sigma_k = (-^k+^{n-k})$, we build a data structure $\Sigma_k(\mathcal{L}) = \Sigma(\mathcal{L} \cup \Lambda, \sigma_k \cdot \bar{\sigma}_k)$, as described in Section 3. Then to test a given query line l we first use binary search to locate its xy-slope among the slopes of the n given lines. This information determines the orientation class σ_k of l relative to the lines in Λ. Once this has been found, we use the data structure $\Sigma_k(\mathcal{L})$ to test whether l has the opposite orientation class $\bar{\sigma}_k$ relative to the lines in $\mathcal{L}$.

This straightforward algorithm uses space approximately cubic in n. To reduce the amount of space, we merge all the n data structures $\Sigma_k(\mathcal{L})$ into a single data structure $\Sigma^*(\mathcal{L})$ as follows. Assume all lines in $\mathcal{L}$ have been sorted by xy slope and oriented as described above. Let m be a parameter, to be chosen later. Partition $\mathcal{L}$ into m subsets $\mathcal{L}_1, \ldots, \mathcal{L}_m$, each subset consisting of approximately n/m consecutive lines in slope order. Prepare the data structures $\Sigma_j^p(\mathcal{L}) = \Sigma(\mathcal{L}_j^p, (+\,+\cdots+))$ and $\Sigma_j^s(\mathcal{L}) = \Sigma(\mathcal{L}_j^s, (-\,-\cdots-))$ for each prefix set $\mathcal{L}_j^p = \bigcup_{1\le k<j} \mathcal{L}_k$ $(1 \le j \le m)$ and each suffix set $\mathcal{L}_j^s = \bigcup_{j<k\le m} \mathcal{L}_k$ $(2 \le j \le m)$. The storage and preprocessing time for these steps amount to $O(mn^{2+\varepsilon})$, for any $\varepsilon > 0$. Then recursively build the data structure $\Sigma^*(\mathcal{L}_j)$ for each subset $\mathcal{L}_j$ (using the same choice of the parameter m). Therefore $\Sigma^*(\mathcal{L})$ is a data structure tree whose degree is m and whose depth will be $O(\log n/\log m)$. Testing a query line l now proceeds as follows. As before, we use binary search to locate the xy slope of l between the slopes of two lines l^k and l^{k+1} of $\mathcal{L}$. (This step has to be performed only once.) Let $\mathcal{L}_j$ be the subset containing the line l^k. By construction, the xy slope of l is greater than the slopes of all lines in $\mathcal{L}_j^p$ and less than the slopes of all lines in $\mathcal{L}_j^s$. Then we can test, in $O(\log n)$ time, whether l lies above all lines in these two subsets, using the data structures Σ_j^p and Σ_j^s. If l does not lie above all these lines we stop immediately; otherwise we recursively test l against $\mathcal{L}_j$ using the data structure $\Sigma^*(\mathcal{L}_j)$. If we set $m = \lceil n^\nu \rceil$, for some fixed and very small $\nu > 0$, the entire procedure takes time $O((\log n)^2/\log m)) = O(\log n)$. The storage and preprocessing time amount to $O(mn^{2+\varepsilon}(\log n)/\log m)$, which can also be written as $O(n^{2+\varepsilon})$, for a different yet still arbitrarily small value of $\varepsilon > 0$.

We can also provide a modified version of this procedure, having the same complexity bounds, that can determine, for each query line l lying above all lines of $\mathcal{L}$, which is the first line of $\mathcal{L}$ that l will hit when translated vertically downward. The key observation is that translation of l downward corresponds to motion of $\pi(l)$ along a straight line, say $\rho(l)$, on the Plücker hypersurface Π: the coordinates $\pi(l)$ change linearly with the altitude of l, as follows:

$$[\pi_{01}, \pi_{02}, \pi_{12}, \pi_{03}, \pi_{13} - t\pi_{01}, \pi_{23} - t\pi_{02}],$$

where t is a parameter denoting the altitude. As l moves vertically, it will become incident with another line l' exactly when $\rho(l)$ crosses the plane $\varpi(l')$, and the crossing point can be computed in constant time. Moreover, the crossing point determines the line $\rho(l)$ uniquely, since it corresponds to a unique line in 3-space and the inverse of the downward translation is a unique upward translation. Note that as t tends to infinity (which corresponds to lifting the line up by an infinite amount), its Plücker image tends to a point of the form $[0, 0, 0, 0, -\pi_{01}, -\pi_{02}]$. These limit points constitute a line τ in $\mathcal{P}^5$, and correspond to lines at infinity of 3-space passing through the zenith point.

Recall that at each step in the construction of the data structure $\Sigma(\mathcal{L})$ we take a net $\mathcal{R}$ of the hyperplanes $\varpi(l^i)$ and construct the convex polytope $C(\mathcal{R})$. Instead of decomposing $C(\mathcal{R})$ into simplices, we divide its interior by a set of hypersurfaces with the property that no line $\rho(l)$ crosses one of these hypersurfaces, and the resulting cells still have constant complexity. Specifically, take a decomposition of the boundary of $C(\mathcal{R})$ into simplices, and back-project from each such simplex s along the lines ρ that terminate at points on s. The collection of these back-projections yields a decomposition of $C(\mathcal{R})$ into $O(r^2)$ cells. We argue that the combinatorial complexity of each cell is a constant independent of r. Indeed, the base of each cell is a four-dimensional simplex, the walls of the cell are a lifting of the boundary of this simplex along the lines $\rho(l)$, and the roof of the cell is some interval on the line τ. Because of the way these cells are constructed, to each cell c there corresponds a unique line $l(c)$ of $\mathcal{R}$ that is first hit as we translate downward any line whose Plücker point lies in c. Again, the ε-net theory tells us that we can find a subset of $O((n/r)\log r)$ lines of $\mathcal{L}$ such that the downward translation of any line l, with $\pi(l)$ in c, will not meet any other line of $\mathcal{L}$ until it reaches λ.

Therefore, if we use this modified cell decomposition of $C(\mathcal{R})$ when constructing the data structure $\Sigma^*(\mathcal{L})$, then we test a line l for being above the n given lines, we can at the same time locate the nearest line below l.

THEOREM 3. *Given n lines in space, we can preprocess them by a procedure whose running time and storage is $O(n^{2+\varepsilon})$, for any $\varepsilon > 0$, so that, given any query line l, we can determine, in $O(\log n)$ time, whether l lies above all the given lines, and, if so, which is the first line of $\mathcal{L}$ that l will hit when translated downward.*

5. The Complexity of the Upper Envelope of n Lines. In the previous section we saw that the upper envelope $\mathcal{U}(\mathcal{L})$ of a set of n lines in 3-space is the union of n orientation classes relative to the set $\mathcal{L} \cup \Lambda$. Each of these classes can be described as polytope of

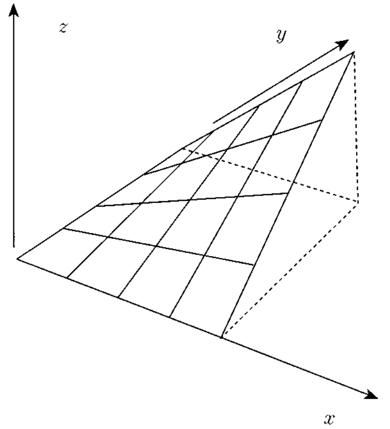

Fig. 1. The hyperbolic paraboloid $z = xy$ and its two families of generating lines.

$\mathcal{P}^5$ with at most $O(n^2)$ features. Therefore, the combinatorial complexity of $\mathcal{U}(\mathcal{L})$ is at most $O(n^3)$.

Notice that each of these n selected orientation classes relative to $\mathcal{L} \cup \Lambda$ defines a single isotopy class. This is so because any two lines ρ^1, ρ^2 in this class point in the same sector defined by the lines of $\mathcal{L}$ down in the xy-plane. Thus we can always continuously move ρ^1 to ρ^2 by first lifting it up high enough, then rotating it to align with ρ^2, and then dropping it down onto ρ^2. In particular, this implies that in each of these n orientation classes, there are at most $O(n^2)$ lines that touch four lines of $\mathcal{L}$, and lie above all the remaining ones. (Each such line is the intersection of the Plücker hypersurface Π with an edge of polytope $C(\mathcal{L}, \sigma)$; since Π is a quadric, there are at most two such intersections per edge.)

We now exhibit a set of n lines that attains this cubic bound. The example consists of three collections of lines, A, B, and C, of roughly equal size. The lines in sets A and B are parallel to the xz-plane and to the yz-plane, respectively, and form a grid of orthogonal generating lines of hyperbolic paraboloid $z = xy$. See Figure 1. The lines in set C pass well below the paraboloid and have a steep z slope; their xy projections form a narrow pencil near the line $x + y = 0$—see Figure 2. The lines of C are arranged so that as we walk along their "upper envelope" we visit each of them in a sufficiently long interval along which we can obtain "tangential views" of the entire portion of the hyperbolic paraboloid covered by A and B. Thus for every triplet of lines a, b, c, one line from each collection, we can find a line lying so that it connects the intersection of a and b with an appropriate point on c, and lies above all other lines. The bound $\Omega(n^3)$ then follows. The technical details of this construction are given below.

To start, we let $m = \lfloor n/3 \rfloor$, $A = \{a^1, a^2, \ldots, a^m\}$, $B = \{b^1, b^2, \ldots, b^m\}$, and $C = \{c^1, c^2, \ldots, c^{n-2m}\}$, where

$$a^i = \{(x, y, z)\colon\ (y = i) \text{ and } (z = ix)\},$$
$$b^j = \{(x, y, z)\colon\ (x = j) \text{ and } (z = jy)\},$$

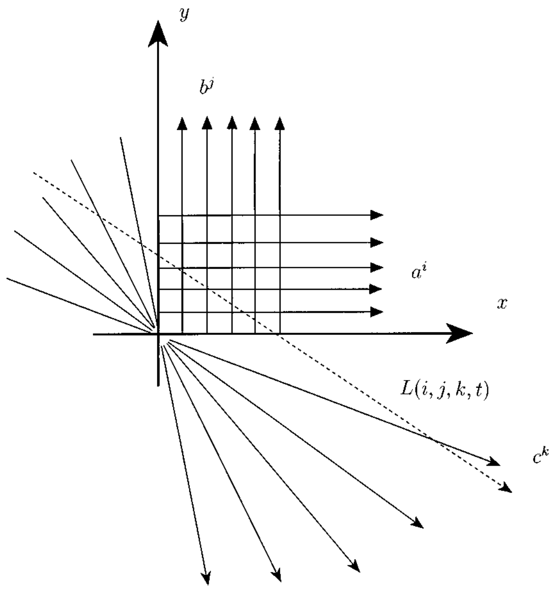

Fig. 2. The oriented line groups A, B, C (solid) and a representative of the oriented lines $L(i, j, k, t)$ (dashed), viewed from above.

and

$$c^k = \left\{(x, y, z): \left(y = \left(\frac{k}{n^2} - 1\right) x\right) \text{ and } (z = nx - n^5)\right\}.$$

Note that the lines in group C all lie on the plane $z = nx - n^5$ and are concurrent to the point $(0, 0, -n^5)$ on the z-axis.

We now choose the Plücker coordinates for each of the lines defined above. We pick the points $[1, 0, i, 0]$ and $[1, 1, i, i]$ on line a^i, which gives the Plücker coordinates $[1, 0, -i, i, 0, i^2]$. For line b^j, we take the points $[1, j, 0, 0]$ and $[1, j, 1, j]$, which yields $[0, 1, j, j, j^2, 0]$. Finally, for line c^k we choose the points $[1, 0, 0, -n^5]$ and $[1, 1, k/n^2 - 1, n - n^5]$, which gives the Plücker coordinates $[1, k/n^2 - 1, 0, n, n^5, kn^3 - n^5]$.

Now we introduce the line $L(i, j, k, t)$ which passes through the points $[1, j, i, ij]$ and $[1, t, (k/n^2 - 1)t, nt - n^5]$. Note that $L(i, j, k, t)$ intersects lines a^i, b^j, and c^k. Its Plücker coordinates are

$$\left[t - j, \left(\frac{k}{n^2} - 1\right) t - i, \left(\frac{jk}{n^2} - i - j\right) t, nt - n^5 - ij,\right.$$
$$\left. j((n - i)t - n^5), \left(n - \frac{jk}{n^2} + j\right) it - in^5\right].$$

In order to show that for proper choices of t the line $L(i, j, k, t)$ lies above all lines in $A \cup B \cup C$ (except for those it intersects), we compute

$$L(i, j, k, t) \diamond a^r = (i - r)\left(n - r + j - \frac{jk}{n^2}\right)t + (i - r)(jr - n^5),$$
$$L(i, j, k, t) \diamond b^r = (j - r)\left(i - n - r + \frac{kr}{n^2}\right)t + (j - r)(n^5 - ri),$$

and

$$L(i, j, k, t) \diamond c^r = (k - r)\left(\frac{j}{n} - \frac{ij}{n^2} - n^3\right)t.$$

Now set $t = (n^5 - ij)/(n - i + j)$. For large enough n the projection on the xy-plane of any $L(i, j, k, t)$ is clockwise of the projection of any line in A or B. The slope of the projection of c^r is $\alpha = -1 + r/n^2$ and that of $L(i, j, k, t)$ is $\beta = (i - (-1 + k/n^2)t)/(j - t)$. After performing some algebraic computations we find that if n is large enough, then $\alpha > \beta$ if and only if $k \leq r$. In other words, $L(i, j, k, t)$ is clockwise to all c^r with $k \leq r$ and it is counterclockwise to the others. Thus, line $L(i, j, k, t)$ lies above (or intersects) all lines in $A \cup B \cup C$ if and only if the four inequalities below are satisfied:

(4) $$L(i, j, k, t) \diamond a^r \geq 0 \quad \text{for } 1 \leq r \leq m,$$

(5) $$L(i, j, k, t) \diamond b^r \geq 0 \quad \text{for } 1 \leq r \leq m,$$

(6) $$L(i, j, k, t) \diamond c^r \geq 0 \quad \text{for } 1 \leq r < k,$$

and

(7) $$L(i, j, k, t) \diamond c^r \geq 0 \quad \text{for } k \leq r \leq n - 2m.$$

It is easy to check that (6) and (7) are always satisfied if n is sufficiently large. To see that the same is true for (4) and (5) we rewrite the inequalities. Inequality (4) becomes

$$-\left(j - \frac{n^5 - ij}{n - i + j}\right)(r - i)^2 + \left(\frac{n^5 - ij}{n - i + j}\right)\frac{jk}{n^2}(r - i) \geq 0,$$

while, for inequality (5), we get

$$+\left(i + \frac{n^5 - ij}{n - i + j}\right)(r - j)^2 - \left(\frac{n^5 - ij}{n - i + j}\right)\frac{kr}{n^2}(r - j) \geq 0.$$

In both cases the first term dominates the other if n is sufficiently large. Therefore each inequality is satisfied and $L(i, j, k, t)$ lies above each of the n lines. Note that the n lines can be made mutually disjoint by perturbing them a little without making the combinatorial complexity of the upper envelope any smaller. This completes the detailed construction.

THEOREM 4. *The maximum combinatorial complexity of the entire upper envelope of n lines in space is* $\Theta(n^3)$.

6. Testing the Towering Property. In this section we exhibit a reasonably efficient deterministic algorithm for testing whether n blue lines $b_1, \ldots, b_n$ in 3-space lie above m other red lines $r_1, \ldots, r_m$; this is what we call the "towering property." Our method runs in time $O((m+n)^{4/3+\varepsilon})$, for any $\varepsilon > 0$, a substantial improvement over the obvious $O(mn)$ method.

We first consider the case where the xy-slope of every red line is at least as large as that of any blue line. In that case if we map the blue lines (oriented so as to have xy-projections going from left to right) to points $\lambda_1, \ldots, \lambda_n$ in $\mathcal{P}^5$ via Plücker coordinates, and the red lines (similarly oriented) to hyperplanes $\rho_1, \ldots, \rho_m$ in $\mathcal{P}^5$ via Plücker coefficients, then the towering property is equivalent to asserting that all n blue points lie in the convex polyhedron $\overline{\mathcal{C}}$ obtained by intersecting the appropriate half-spaces bounded by the m red hyperplanes, as given by (1).

How do we test this latter property? We again use a net-based partitioning method, as in the previous section. However, first we dispose of some boundary cases. If $n \geq m^2$ (so there are relatively few red lines), then we compute the upper envelope of the red lines, as in the preceding section. That is, we compute the intersection $\overline{\mathcal{C}}$ of the appropriate half-spaces bounded by the red hyperplanes, and preprocess it for point location. Then we test whether the Plücker image of every blue line lies in $\overline{\mathcal{C}}$. All this can be done in time $O(m^{2+\varepsilon} + n\log m)$. Dually, in the case $m \geq n^2$, we can solve our problem in time $O(n^{2+\varepsilon} + m\log n)$ (by mapping the blue lines to hyperplanes and the red lines to points).

Otherwise, we choose a net $\mathcal{R}$ of a constant number r of red half-spaces in $O(nr^4/\log^4 r)$ time, compute their intersection, denoted as above by $C_{\mathcal{R}}$, and obtain a simplicial cell decomposition of this convex polyhedron into $O(r^2)$ simplices. Just as in Section 4, each simplex σ of this decomposition will meet at most $c(m/r)\log r$ of the red hyperplanes, for some absolute constant $c > 0$. Again as before, it is possible to choose these simplices so that if a red hyperplane avoids a simplex σ, then σ is contained in the half-space of the hyperplane. We now locate the n blue points in these chosen simplices (by an exhaustive method, for example). If all the points do not lie in them, we have a negative answer to the towering question and we are done. If all goes according to plan, however, we end up with $O(r^2)$ separate towering subproblems, each involving some blue points together with $O((m/r)\log r)$ red half-spaces. Because a blue point can lie in only one simplex, the subsets of the blue points belonging to each simplex form a partition of the set of all blue points.

Let $D(m,n)$ denote the time complexity of testing the towering property for n blue points and m red hyperplanes in $\mathcal{P}^5$. The above divide-and-conquer method gives us the following recurrence for $D(m,n)$:

$$\begin{aligned} D(m,n) &= O(m^{2+\varepsilon} + n\log m), \qquad \text{if} \quad n \geq m^2, \\ D(m,n) &= O(n^{2+\varepsilon} + m\log n), \qquad \text{if} \quad m \geq n^2, \end{aligned}$$

and

$$D(m,n) = \sum_i D\left(c\left(\frac{m}{r}\right)\log r, n_i\right) + O\left(\frac{(m+n)r^4}{\log^4 r}\right) \qquad \text{otherwise,}$$

where c is some fixed constant, and the n_i's are $O(r^2)$ positive integers summing up to

n. We easily prove that the worst case occurs when the n_i's are roughly equal. This gives

$$D(m, n) \le br^2 D\left(c\left(\frac{m}{r}\right)\log r, \frac{dn}{r^2}\right) + O\left(\frac{(m+n)r^4}{\log^4 r}\right),$$

for some additional constants b and d. A similar recurrence is solved in [EGS1]. Using the techniques of that paper, we derive that for any fixed $\varepsilon > 0$ we can first choose the ε in the boundary cases $n \ge m^2$ or $m \ge n^2$ above small enough, and then the net size r large enough so that

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

We now return to the general towering problem and relax all assumptions on the slopes of the projections. Compute the median slope among the projections onto the xy-plane of all the red and blue lines together. This partitions the red lines into two sets, R_1 and R_2, and the blue lines into two sets, B_1 and B_2, such that each line in $R_2 \cup B_2$ projects onto the xy-plane into a line of slope at least as large as that of any projected line of $R_1 \cup B_1$; furthermore, the sizes of $R_1 \cup B_1$ and $R_2 \cup B_2$ are roughly equal. Now we solve the towering problem recursively with respect to R_1 versus B_1 and then for R_2 versus B_2. If no negative answer has been produced yet, then we may apply the previous algorithm to the pairs (R_1, B_2) and (R_2, B_1). The correctness of the procedure follows from the fact that all pairs of red and blue lines are (implicitly) checked.

If $T(m, n)$ is the expected time of this algorithm, then

$$T(m, n) = T(m_1, n_1) + T(m_2, n_2) + O(m^{2/3+\varepsilon} n^{2/3+\varepsilon} + (m+n)\log(m+n)),$$

with $m_1 + m_2 = m$, $n_1 + n_2 = n$, and $m_1 + n_1 = m_2 + n_2$. The solution to this recurrence relation is maximized if $m_1 = m_2, = m/2$ and $n_1 = n_2 = n/2$. In this case we get

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

An additional computation similar to that detailed at the end of the previous section allows us to determine, within the same bounds, the red line immediately below each blue line, and thus also, if the towering property holds, the smallest vertical distance between the two groups of lines. So we conclude with our theorem:

THEOREM 5. *Given n blue lines and m red lines in space, we can test that all the blue lines pass above all the red lines (the towering property) in time and space $O(m^{2/3+\varepsilon} n^{2/3+\varepsilon} + (m+n)\log^2(m+n))$, for any $\varepsilon > 0$. If so, within the same time bound, we can actually find the first red line below each blue line.*

This is upper-bounded by $O((m+n)^{4/3+\varepsilon})$, a simpler expression to remember (but where the coefficient of proportionality depends on ε).

7. Separating Lines by Translation. We address here the question of whether it is always possible to "take apart" a set of lines in 3-space by moving a proper subset of

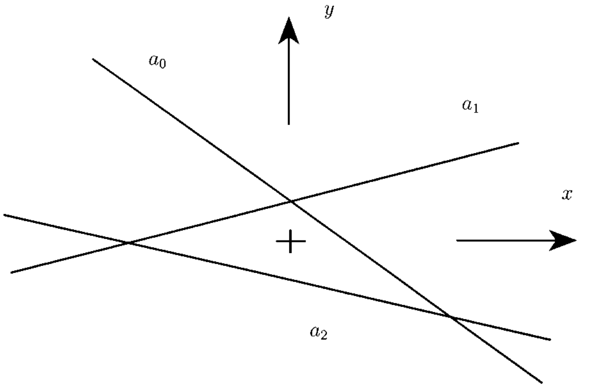

Fig. 3. The group A of lines a_1, a_2, and a_3.

them to infinity, through a continuous sequence of translations, without ever causing lines to cross.

More precisely, let $\mathcal{L}$ be a set of pairwise-disjoint lines in 3-space, and let $X + v$ denote the result of translating a set of lines X by a vector v. We ask whether there are always a proper partition of $\mathcal{L}$ in two subsets F (fixed) and M (moving), and a continuous function $v(t)$ from $\Re$ to $\Re^3$ such that $v(0) = \vec{0}$, no line in $M + v(t)$ meets or is parallel to a line in F for all $t \geq 0$, and all lines in $M + v(t)$ get infinitely far from the origin as $t \to \infty$.

The answer is "no." Here is a counterexample with nine lines, consisting of three groups A, B, C of three lines each. Group A consists of the lines a_0 through a_2 joining the following pairs of points, given in Cartesian coordinates:

$$\begin{array}{llll} a_0 & \text{through} & (4, -2, +\varepsilon) & \text{and} \quad (0, 1, -\varepsilon) \\ a_1 & \text{through} & (0, 1, +\varepsilon) & \text{and} \quad (-4, 0, -\varepsilon) \\ a_2 & \text{through} & (-4, 0, +\varepsilon) & \text{and} \quad (4, -2, -\varepsilon) \end{array}$$

where ε is a small number, say 10^{-100}. See Figure 3. The other two groups are obtained from A through $\pm 120^\circ$ rotation around the $(1, 1, 1)$ axis. See Figure 4. Note that A "surrounds" one of the other two groups, B, in the sense that all the lines of B pass through the triangle defined by projecting a_0 through a_2 onto the xy-plane. In the same way, group B surrounds the third group C, and C surrounds A.

Now suppose the partition leaves one group—say A—entirely in F and suppose $b_i \in M$. Then the displacement vectors $v(t)$ are confined to a bi-infinite triangular prism whose axis is parallel to b_i and whose faces are parallel to a_0 through a_2. Since these displacements never take b_i very far from the origin, the line b_i must be in F. However, if all lines of B are fixed, the same argument shows that C is entirely fixed, and $M = \emptyset$, a contradiction. We conclude that no group can be entirely in F; and since we can swap M and F by negating all displacements $v(t)$, the same argument shows that no group can be entirely in M.

So, the M, F partition must split all three groups. We consider group A for a moment. Note that the z-slope of the lines a_0 through a_2 is less than ε, and that a_{i+1} passes only

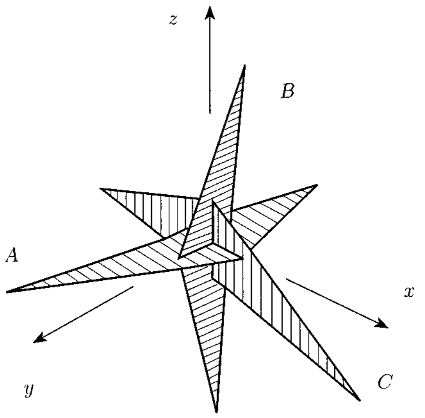

Fig. 4. All three groups A, B, and C together—the full counterexample.

2ε above a_i (where indices are computed modulo 3). Therefore, if a_{i+1} is fixed and a_i is moving, the displacement vectors $v(t)$ must lie below a plane of slope close to ε that passes 2ε above the origin of $\Re^3$. Conversely, if a_j is fixed and a_{j+1} is moving, the displacements $v(t)$ must lie *above* a similar plane that passes 2ε below the origin. Combining these two arguments, we conclude that if the partition M, F splits group A, then the displacements $v(t)$ are confined to a narrow wedge whose faces are very close to the xy-plane.

Applying the same argument to the other two groups, we conclude that the displacements $v(t)$ lie in the intersection of three narrow wedges, each close to the corresponding coordinate plane. However, this intersection is bounded (its diameter is at most a few times ε), which means M cannot be moved to infinity, a contradiction. We have thus proved:

THEOREM 6. *There is a set of nine mutually disjoint lines in* 3*-space that cannot be taken apart by continuously translating a proper subset off to infinity.*

8. Open Problems. Although the manifold of all nonoriented lines in 3-space has been well studied [HP], less seems to be known about the manifold of oriented lines that we have used in this paper, and which seems to be computationally of significant advantage. It is known that this manifold is topologically equivalent to the oriented Grassmann manifold $\tilde{M}_{4,2}(\Re)$, which happens to be the same as $S^2 \times S^2$.[11]

In general, it appears that most questions about lines in space are still open. Below we list some of the most natural ones.

[11] A geometric proof can be given by associating to every pair of unit vectors u, v (placed at the origin of 3-space) the oriented line l that passes through the tip of the vector $(u \times v)/(1 + u \cdot v)$ and has the direction of the vector $u + v$. When $u = -v$ the line l is by definition the line at infinity on the plane normal to v and oriented relative to v according to the right-hand rule. It is easy to check that this mapping is continuous, one-to-one, and generates all oriented lines of 3-space.

A. *Isotopy Classes*. We already mentioned several open problems about isotopy classes. What is the maximum number of isotopy classes than can be associated with a single orientation class? We conjecture that this number is $\Theta(n^2)$.

B. *Many Cells in Line Arrangements*. We saw in Section 3 that any single "line" cell in an arrangement of n lines has combinatorial complexity $O(n^2)$. On the other hand, we know that all $O(n^4)$ cells have total combinatorial complexity of only $O(n^4)$. So what about the total combinatorial complexity of m distinct cells? A particularly interesting case of this would be to determine the total complexity of the "unbounded component" of the arrangement, that is, of those cells containing lines that can be pulled away to infinity. Such problems have been extensively studied for arrangements of lines and segments in the plane, and for arrangements of planes and hyperplanes in three and higher dimensions [EGS1], [CEGSW], [EGS2]. If we blindly use the result for m cells in an arrangement of n hyperplanes in five dimensions from [EGS2] we get a very weak estimate for our problem. The reason is that only those features of the m cells lying on the Plücker hypersurface matter for us. It would be interesting to develop such a "many-faces" theory for arrangements of lines in space. A curious subproblem here is to find a geometrically intuitive way to partition a line cell into natural subcells, each of constant description complexity (i.e., to triangulate the cell), so that the number of such subcells is proportional to the feature complexity of the original cell.

Related to many-faces problems are questions about incidences. We have been able to obtain an $O(n^{7/4})$ upper bound on the number of triple intersections of noncoplanar lines among n given lines in space [CEG$^+$], and very recently this was improved to $O(n^{23/14} \log^{31/14} n)$ [Sh2]. A lower bound of $\Omega(n^{3/2})$ is easy to construct and a natural open problem is to close this gap.

C. *The Complexity of a Surface Upper Envelope*. Given a collection of n lines in 3-space, we can consider a surface $\varphi(x, y)$ defined as follows. For each point (x, y) in the plane the value of the surface $\varphi(x, y)$ is the smallest z with the property that there is a line through the point (x, y, z) which passes above the n given lines. We know that this surface consists of a bunch of patches of different reguli joined together. What is the combinatorial complexity of this surface?

D. *k-Sets and Related Concepts for Line Arrangements*. We saw that the upper envelope of n lines in space has $O(n^2)$ vertices in the consistent orientation case. This means that there are only $O(n^2)$ other lines stabbing four of the given lines and passing above all the rest. How many lines are there stabbing four of the given lines and passing above at most k of the given lines? A preliminary calculation using the techniques of [CS] and [Sh1] suggests that the right answer is $O(n^2k^2)$. Many more questions about standard k-sets [Ed] have analogs in this line setting and deserve further study.

E. *Order Statistics and Centerlines*. Since lines in space can form cycles, they can have strange order statistics. For example, if all lines form a cycle in the above/below relation, then each line could be above half of the other lines and below the other half. We can associate with a line arrangement these counts of how many lines lie above and below each line; it would be nice to characterize the valid count sequences. We may also think of analogs in the line case of the notion of centerpoints for collections of points (in the plane or space). For example, given a collection of lines in space, does another line (the "centerline") always exist such that in all planes passing through the centerline the intersections of the given lines with that plane are roughly evenly distributed (a constant fraction lying) in each of the half-planes defined by the centerline? In a somewhat related vein, Paterson [P] was able to show recently that for any set of n lines in space three mutually orthogonal planes always exist so that each orthant thus defined is cut by only $n/2$ of the lines.

F. *Cycles in Line Arrangements*. In [CEG+] the following result is presented. Let $\mathcal{L}$ be a given collection of n nonvertical lines in 3-space. Define a directed graph G whose vertices are the lines of $\mathcal{L}$ and whose directed edges are of the form $\overrightarrow{l_1 l_2}$ if l_1 lies above l_2. Then we can test, in randomized expected time $O(n^{4/3+\varepsilon})$, for any $\varepsilon > 0$, whether G is acyclic. Many related open questions remain. For example, how fast can we compute the strong components of this graph G? If there is a cycle present, then what is the minimum number of cuts we need to break up our lines so that the resulting collection of segments is acyclic?

G. *Taking Lines Apart*. We could try to extend Theorem 6 in a number of ways. For instance, we conjecture that a configuration of lines in 3-space exists that cannot be taken apart even if we allow the moving subset to go through arbitrary rigid (Euclidean) motions, or arbitrary affine maps, instead of just translations. We may also study what happens if we are allowed to partition the lines into three or more independently moving subsets.

Acknowledgments. The support of the Digital Systems Research Center and the Digital Paris Research Laboratory, where much of this research was carried out, is gratefully acknowledged. Various suggestions by the referees helped improve the presentation of this paper.

References

[AG] P. Agarwal, Geometric partitioning and its applications, in *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, American Mathematical Society, Providence, RI, 1991.

[APS] B. Aronov, M. Pellegrini, and M. Sharir, On the zone of a surface in a hyperplane arrangement, In preparation. See also *Proc. 2nd WADS Workshop*, 1991, Lecture Notes in Computer Science, Vol. 519, Springer-Verlag, Berlin, pp. 13–19, for an early version by two of the authors.

[BR] O. Bottema and B. Roth, *Theoretical Kinematics*, Dover, New York, 1990.

[CEG+] B. Chazelle, H. Edelsbrunner, L. Guibas, R. Pollack, R. Seidel, M. Sharir, and J. Snoeyink, Counting and cutting cycles of lines and line segments in space, *Proc. 31st IEEE Symp. on Foundations of Computer Science*, 1990, pp. 242–251.

[CEGS1] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir, Algorithms for bichromatic line segment problems and polyhedral terrains, In preparation. An earlier version of this paper, combined with the current paper, has appeared in *Proc. 21st ACM Symp. on Theory of Computing*, 1989, pp. 382–393.

[CEGS2] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir, A singly-exponential stratification scheme for real semi-algebraic varieties and its applications, *Proc. 16th ICALP Colloq.*, Lectures Notes in Computer Science, Vol. 372, Springer-Verlag, Berlin, 1989, pp. 179–193.

[CEGSW] 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.

[CF] B. Chazelle and J. Friedman, A deterministic view of random sampling and its use in geometry, *Combinatorica*, **10** (1990), 229–249.

[Ch] B. Chazelle, An optimal convex hull algorithm and new results on cuttings, *Proc. 32nd Symp. on Foundations of Computer Science*, 1991, pp. 29–38.

[Cl1] K. Clarkson, New applications of random sampling in computational geometry, *Discrete Comput. Geom.*, **2** (1987), 195–222.

[Cl2] K. Clarkson, A probabilistic algorithm for the post office problem, *Proc. 17th ACM Symp. on Theory of Computing*, 1985, pp. 175–184.

[CS] K. Clarkson, Applications of random sampling in computational geometry, *Discrete Comput. Geom.*, **4** (1989), 387–421.

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

[EGS1] H. Edelsbrunner, L. Guibas, and M. Sharir, The complexity of many faces in arrangements of lines and of segments, *Proc. 4th ACM Symp. on Computational Geometry*, 1988, pp. 44–55.

[EGS2] H. Edelsbrunner, L. Guibas, and M. Sharir, The complexity of many faces in arrangements of planes and related problems, *Discrete Comput. Geom.*, **5** (1990), 197–216.

[FT] G. Fano and A. Terracini, *Lezioni di Geometria Analitica e Proiettiva*, Paravia, 1939.

[GCS] Z. Gigus, J. Canny, and R. Seidel, Efficiently computing and representing aspect graphs of polyhedral objects, Technical Report UCB/CSD 88/432, Univesrity of California at Berkeley, August 1988.

[HP] W. Hodge and D. Pedoe, *Methods of Algebraic Geometry*, Cambridge University Press, Cambridge, 1952.

[Hu] K. Hunt, *Kinematic Geometry of Mechanisms*, Oxford, 1990.

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

[I] A. Inselberg, N-dimensional graphics. Part I—lines and hyperplanes. Scientific Center Report G320-2711, IBM, Los Angeles, CA, 1981.

[J] C. M. Jessop, *A Treatise on the Line Complex*, Chelsea, New York, 1969.

[Ma1] J. Matoušek, Construction of ε-nets, *Discrete Comput. Geom.*, **5** (1990), 427–448.

[Ma2] J. Matoušek, Cutting hyperplane arrangements, *Discrete Comput. Geom.*, **6** (1991), 385–406.

[Ma3] J. Matoušek, Approximations and optimal geometric divide-and-conquer, *Proc. 23rd Annual ACM Symp. on Theory of Computing*, 1991, pp. 506–511.

[Mc] M. McKenna, Arrangements of lines in 3-space: a data structure with applications, Ph.D. Dissertation, Johns Hopkins University, 1989.

[Mi] J. Milnor. On the Betti numbers of real varieties, *Proc. Amer. Math. Soc.*, **15** (1964), 275–280.

[MO] M. McKenna and J. O'Rourke, Arrangements of lines in 3-space: a data structure with applications, *Proc. 4th ACM Symp. on Computational Geometry*, 1988, pp. 371–380.

[P] M. Paterson, Personal communication.

[PD] W. H. Plantinga and C. R. Dyer, An algorithm for constructing the aspect graph, *Proc. 27th IEEE Symp. on Foundations of Computer Science*, 1986, pp. 123–131.

[Pl] J. Plücker, On a new geometry of space, *Philos. Trans. Roy. Soc. London*, **155** (1865), 725–791.

[PPW] J. Pach, R. Pollack, and E. Welzl, Weaving patterns of lines and line segments, *Proc. 1st SIGAL Symp.*, Lecture Notes in Computer Science, Vol. 450, Springer-Verlag, Berlin, 1990, pp. 437–446.

[Se] R. Seidel, Constructing higher dimensional convex hulls at logarithmic cost per face, *Proc. 18th ACM Symp. on Theory of Computing*, 1986, pp. 404–413.

[Sh1] M. Sharir, On k-sets in arrangements of curves and surfaces, *Discrete Comput. Geom.*, **6** (1991), 593–613.

[Sh2] M. Sharir, On joints in arrangements of lines in space and related problems, In preparation.

[So] D. M. Y. Sommerville, *Analytical Geometry of Three Dimensions*, Springer-Verlag, Heidelberg, 1987.

[St] J. Stolfi, *Oriented Projective Geometry*, Academic Press, New York, 1991.