DIMACS Series in Discrete Mathematics
and Theoretical Computer Science
Volume 6, 1991

Lines in Space—A Collection of Results

HERBERT EDELSBRUNNER

ABSTRACT. Many computational geometry problems are exceedingly more difficult if the setting is the (three-dimensional real) space $\mathbf{R}^3$ rather than the plane $\mathbf{R}^2$. Most often the reason for this striking increase in complexity is the appearance of new geometric phenomena caused by one-dimensional objects in space. The intention of recent studies on problems for lines in space is to shed light on these new phenomena and their complexities. This paper reviews some of the most important results and shows how they are related to problems in dimensions 2 and 5.

1. Introduction

In computational geometry it is a truism that problems in three-dimensional space, $\mathbf{R}^3$, are significantly more difficult and complex than the corresponding problems in the two-dimensional plane, $\mathbf{R}^2$. Of course, reality is not that simplistic, but it seems this way often enough to be a generally accepted view. As an example consider the problem of finding the *width* of a convex polygon/polytope with n vertices, that is, the smallest distance between two parallel lines/planes that bracket the polygon/polytope. In $\mathbf{R}^2$ this distance is realized by a vertex and an edge, and it is fairly easy to see how to enumerate and test $O(n)$ pairs in time $O(n)$ in order to find the width. In $\mathbf{R}^3$ the problem is more complicated because the width can either be realized by a vertex-facet pair or an edge-edge pair, and it is not clear how to limit the number of edge pairs that have to be considered. Indeed, no algorithm is known that computes the width of a convex polytope in time anywhere close to linear in n.

1991 *Mathematics Subject Classification*. Primary 05C35, 52A37, 68C05.

Key words and phrases. Computational and discrete geometry, three dimensions, lines, line segments, weavings, quadrics, cycles, hidden line elimination, Plücker coordinates, segment trees, polyhedral terrains.

Research of the author was supported by the National Science Foundation under grant CCR-8921421.

©1991 American Mathematical Society
1052-1798/91 $1.00 + $.25 per page

In a nut-shell, the kind of interaction between the edges as it appears in the width problem is a geometric phenomenon which cannot be observed in $\mathbf{R}^2$ and which significantly complicates matters in $\mathbf{R}^3$. More generally, the intricate interplay between one-dimensional objects in space is what makes computational geometry more difficult and interesting in $\mathbf{R}^3$ than in $\mathbf{R}^2$. Fairly recent research in the area specifically concentrated on this aspect of spatial geometry. This survey is based on this research, more specifically on the papers [**4**, **5**, **6**, **10**], and summarizes some of the more important results in a fairly leisurely manner.

The first part of this survey, §§2–7, is combinatorial in nature and focuses on the notion of cycles defined by lines or line segments in space. Section 2 formally introduces the notion of a cycle and presents some motivation for why cycles seem a worthwhile object of study. The second part of this survey, §§8–15, has a more algorithmic flavor than the first part. The objects under investigation are sets of line segments and polyhedral terrains.

2. Lines and cycles

Given a finite collection of lines in space and a viewpoint, we can define a relation "$\prec$" that reflects when one line obstructs the view of another line, see Figure 2.1 for an example.
The main topic of the first part of this survey paper are combinatorial considerations related to "$\prec$". The objects under consideration are (infinite) lines and line segments (also called *rods*) in space. Typical questions asked are:

(i) How many cycles can n lines define?
(ii) How many cuts are necessary to eliminate all cycles?
(iii) How fast can a cycle be detected, if there is one?

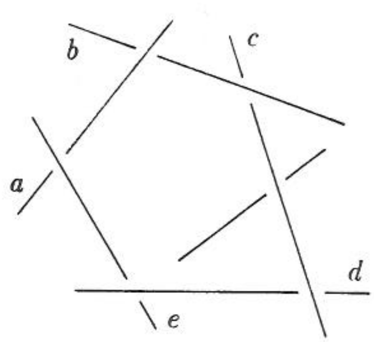

FIGURE 2.1. $a \prec b$, $b \prec c$, $c \prec d$, $d \prec e$, and $e \prec a$, therefore $a \prec b \prec c \prec d \prec e \prec a$ is a cycle.

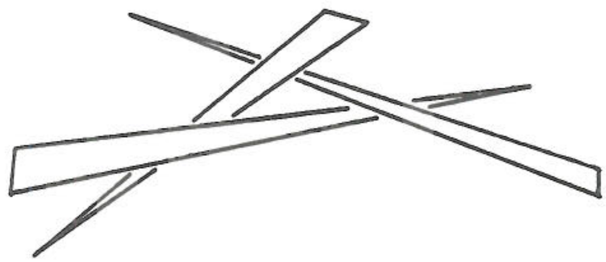

FIGURE 2.2. A cycle in "$\prec$" makes Step 1 of the Painter's algorithm impossible. A way out is to cut the objects so that all cycles are removed.

While these questions are of independent interest, there *are* problems in computational geometry that motivate the study of cycles. The *hidden surface removal problem* for a set of opaque polygonal objects, e.g. triangles in space, is the problem of drawing the view of the objects as seen from a given viewpoint. One of the common algorithm for this problem is the *Painter's algorithm*:

1. Sort the triangles consistently with "$\prec$."
2. Paint the objects according to this ordering from back to front.

Overmars and Sharir **[9]** have algorithms that eliminate hidden lines and surfaces by constructing the view from the front to the back. The complexity of their algorithms is sensitive to the output. A common drawback of the Painter's algorithm and the algorithms in **[9]** is that they assume acyclicity of "$\prec$". As illustrated in Figure 2.2, "$\prec$" can be cyclic for as few as three objects. So-called binary space partition trees are a common method for removing all cycles by effectively cutting the objects into smaller pieces; see **[8, 11]**. This method is related to the results presented in §§6 and 7.

The *point location problem* in space seeks to store a cell complex so that for a given query point the cell that contains it can be found as fast as possible. While in the plane there are solutions that take storage $O(n)$ and query time $O(\log n)$, where n is the total number of faces (regions, edges, and vertices) of the complex, the problem in space seems much harder. A data structure that takes storage $O(n)$ and query time $O(\log^2 n)$ can be found in **[3]**; it assumes, however, that the cells of the complex are acyclic with respect to the viewpoint at $(0, 0, \infty)$. Only recently, Preparata and Tamassia **[15]** gave an algorithm that is reasonable efficient in the general case. It takes storage $O(n \log^2 n)$ and query time $O(\log^2 n)$.

3. Weavings and perfect weavings

A *weaving* is a simple arrangement of lines (or line segments) in the plane, together with a binary "over-under" relation $\prec$ defined for them. A weaving is *perfect* if along each line the lines intersecting it alternate between "over" and "under." A weaving is *realizable* if there are lines (or line segments) in

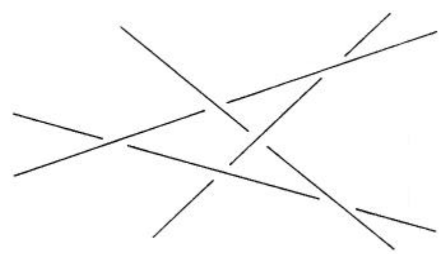

FIGURE 3.1. A perfect 4-weaving.

space that project (vertically) onto the weaving. A *weaving pattern* is a class of weavings that are combinatorially the same and it is *realizable* if at least one representative in the class is realizable. The following result is due to Pach, Pollack, and Welzl [**10**].

THEOREM 3.1. *Any perfect weaving of* $n \geq 4$ *lines is unrealizable.*

The case $n = 4$ can proved simply by rotating two of the four lines in space (see Figure 3.1) so that their projections remain the same, three of the four lines pairwise intersect, and the fourth line still has the strict alternation between "over" and "under" with respect to the three lines. This, of course, is impossible because the first three lines lie in a common plane and the fourth line intersects this plane in a single point. The proof for $n \geq 5$ is more difficult and can be found in [**10**].

4. Bipartite weavings

The roots of a quadratic form in x, y, z define a quadratic surface in space (also called *quadric*). Some members of the family of quadrics, in

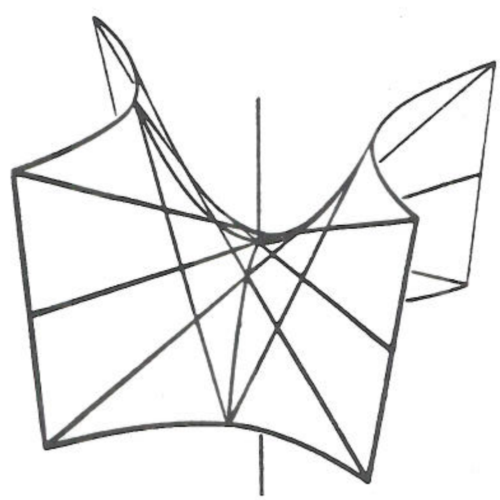

FIGURE 4.1. A hyperbolic paraboloid.

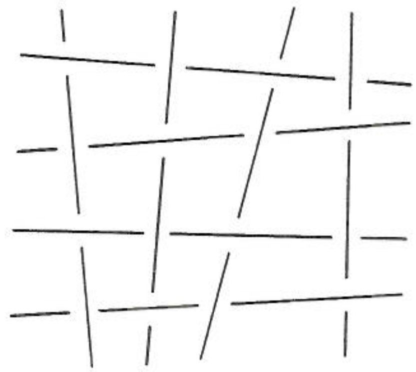

FIGURE 4.2. A perfect 4-by-4 weaving is not realizable.

particular the hyperboloid of one sheet and the hyperbolic paraboloid (see Figure 4.1), turn out to be useful in the study of weavings. Both surfaces have a number of interesting properties.

1. A line not contained in the surface meets it in at most two points.
2. There are two infinite families of lines, A and B, so that the surface is equal to $\bigcup A = \bigcup B$, the lines of each family are pairwise skew, and any two lines of different families either intersect or are parallel.
3. The surface divides space into two connected components.
4. Any three pairwise skew lines define a unique such surface that contains all three lines.

These properties imply fairly straightforwardly that for four lines in general position in space there are either two or zero other lines that meet all four.

A weaving of $m + n$ line segments is *bipartite* if the set of line segments can be split into sets $H = \{h_1, h_2, \dots, h_m\}$ and $V = \{v_1, v_2, \dots, v_n\}$ so that no two line segments of the same set intersect, and all line segments in H (V) intersect all line segments in V (H) in the same order. It is *perfect* if along each line segment the other line segments strictly alternate between "over" and "under," see Figure 4.2. Note that for fixed m and n there is only one perfect bipartite weaving *pattern*. The following result is taken from **[10]**.

THEOREM 4.1. *A perfect bipartite m-by-n weaving is realizable iff* $\min\{m, n\} \le 3$.

We will not repeat the proof of this result which is more complicated than one would like. It heavily uses properties of the two ruled surfaces mentioned above. Using either surface it is easy to construct a perfect bipartite 3-by-n weaving that is realizable. Just take three lines of one ruling family and n lines of the other family. By slightly rotating a line of the second family we can generate either an over-under-over or an under-over-under sequence, as desired. Since arbitrarily large perfect bipartite weavings are nonrealizable it seems natural to ask how contaminated with cycles $\prec$ can get. This leads to the question of counting or bounding the number of cycles.

5. Counting cycles in bipartite weavings

The question considered in this section is the maximum number of cycles, where the maximum is taken over all *realizable* m-by-n weavings. Assume for simplicity that $m = n$ and that either $h_i \prec v_j$ or $v_j \prec h_i$ for every $h_i \in H$ and $v_j \in V$.

Note that every cycle has even length and contains a subcycle of length 4. Furthermore, the maximum number of 4-cycles is $\Theta(n^4)$ (see Figure 5.1). The following lemma turns out to be rather useful in our combinatorial analysis. It can be found in Bollobás [2].

LEMMA 5.1. *If a bipartite graph on m and n nodes in each class contains no $K_{s,t}$, s of the m and t of the n nodes, then it has only* $O(t^{1/s}mn^{1-1/s} + sn)$ *arcs.*

5.1. An upper bound for elementary cycles. A *elementary cycle* is a 4-cycle defined by two adjacent line segments in H and two adjacent line segments in

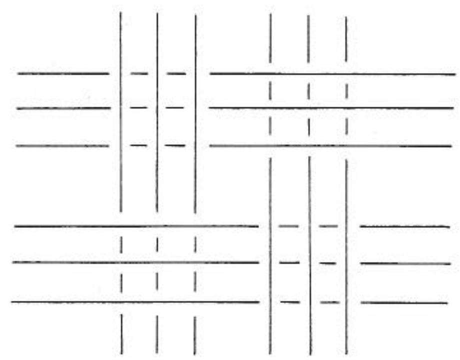

FIGURE 5.1. There are $4k$ line segments and k^4 4-cycles.

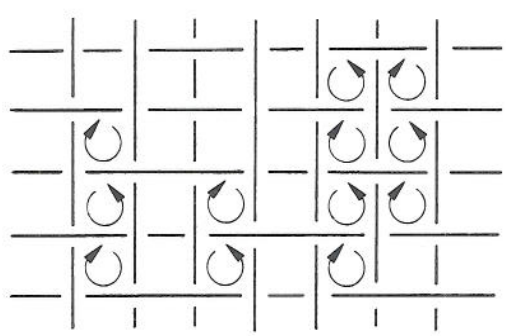

FIGURE 5.2. Cw and ccw elementary cycles in a 5-by-7 weaving.

V; it is either *cw* or *ccw* (see Figure 5.2). Every two adjacent line segments from H and from V in a perfect bipartite weaving define an elementary cycle. The following result, taken from **[6]**, shows that in the realizable case the number of elementary cycles is far less than $(n-1)^2$, the maximum number in the nonrealizable case.

Upper Bound. *A realizable n-by-n weaving has at most* $O(n^{3/2})$ *elementary cycles.*

PROOF. For reasons of symmetry we count only ccw elementary cycles. Define a graph $(\bar{H} \cup \bar{V}, A)$ with $\bar{H} = \{(h_i, h_{i+1}) \mid 1 \le i \le n-1\}$, $\bar{V} = \{(v_j, v_{j+1}) \mid 1 \le j \le n-1\}$, and $\{(h_i, h_{i+1}), (v_j, v_{j+1})\} \in A$ if the four line segments define a ccw cycle. A $K_{2,2}$ in this graph corresponds to a perfect 4-by-4 weaving. Since such a weaving is nonrealizable, the graph cannot have a $K_{2,2}$ and has therefore at most $cn^{3/2}$ arcs. □

5.2. A lower bound for elementary cycles. A lower bound of $\Omega(n^{4/3})$ on the maximum number of cycles in a realizable n-by-n bipartite weaving can be found in **[6]**. The example is based on a construction of n points and n lines in the plane that define $\Theta(n^{4/3})$ point-line incidences (see **[16, 7]**). The points in this construction are arranged in a grid-like fashion, and the line set contains the n lines that contain the most points.

Here is a rough outline (illustrated in Figure 5.3) how this two-dimensional construction with point set P and line set L can be lifted to three-dimensions.

1. Embed the point-line construction in the yz-plane in space.
2. From each point $p \in P$ erect a line v_p normal to the yz-plane that intersects it in p.

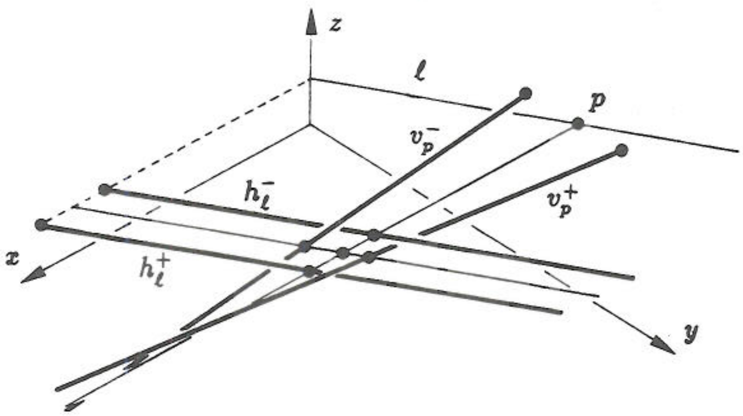

FIGURE 5.3. A point-line incidence in the yz-plane is lifted to an elementary cycle.

3. For each line $\ell \in L$ construct a line h_ℓ parallel to ℓ so that the orthogonal projection of h_ℓ onto the yz-plane is ℓ. The distance from the yz-plane is equal to the slope of ℓ. Notice that v_p and h_ℓ intersect iff $p \in \ell$.
4. Replace h_ℓ by two lines parallel to h_ℓ, one slightly in front and the other slightly behind h_ℓ, as viewed from the positive x-direction.
5. Replace v_p by two lines, one slightly to the left of v_p and with slightly negative slope, and the other slightly to the right of v_p and with slightly positive slope.

The construction can be done so that each point-line incidence becomes an elementary cycle, in the orthogonal projection of the lines onto the xy-plane.

6. Cutting cycles

As argued in §§5.1 and 5.2, a realizable n-by-n weaving can have $\Omega(n^{4/3})$ elementary cycles. This implies that sometimes $\Omega(n^{4/3})$ cuts of lines or line segments are necessary to eliminate all cycles. To get a subquadratic upper bound we first consider a topological property of cycles.

For a cycle we can look at its *polygon* and its *edges*, which are the polygon edges (see Figure 6.1).

LEMMA 6.1. *Any cycle contains a subcycle of length* 4 *whose "horizontal" edges lie on edges of the original cycle.*

To prove this lemma one can use two operations to simplify the polygon of the cycle. If the polygon has a self-intersection then a shortcut can be taken at this point leaving a shorter cycle. If the polygon is simple but not yet a 4-cycle there is a reflex vertex and the "vertical" edge at this vertex can be extended until it hits another "horizontal" edge. There is a shorter cycle either to the left or the right of this extended edge.

Another property of cycles in a bipartite weaving is that if the polygons of two ccw 4-cycles defined by common "horizontal" line segments overlap,

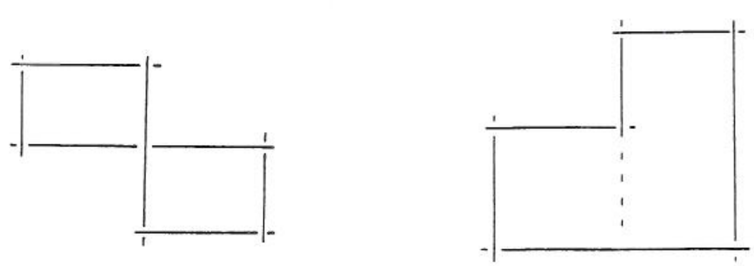

FIGURE 6.1. The left illustrates the polygon of a cycle. The right shows how an edge can be extended so as to get a shorter cycle.

then they overlap in the polygon of another ccw 4-cycle. Both results suggest that we cut 4-cycles at their "horizontal" edges and thus eliminate all cycles.

THEOREM 6.2. $O(n^{9/5})$ *cuts are sufficient to eliminate all cycles in a realizable* n*-by-*n *weaving.*

PROOF. First, set $m = n^{1/5}$ and cut along every $(m+1)$st rod of H and of V. This requires $O(n^{9/5})$ cuts and localizes each cycle to within an m-by-m bipartite weaving. Second, to eliminate the remaining cycles cut each 4-cycle at its two "horizontal" rods, unless its quadrilateral contains the quadrilateral of another 4-cycle and shares at least one of the "horizontal" rods with it.

To count we define a bipartite graph with sets of nodes A and B and prepare a $K_{2,2}$ argument. A is the set of pairs (v_i, v_j) with $1 \le j - i \le m$ and B is the set of pairs (h_i, h_j) with $1 \le j - i \le m$. So A and B are of size $O(mn)$ each. Connect a node $a \in A$ with a node $b \in B$ if the four lines define a ccw 4-cycle and the quadrilateral does not contain the quadrilateral of another ccw 4-cycle with which it shares at least one "horizontal" rod. Partition B into about $\binom{m+2}{2}$ subsets so that for each subset $j - i$ is invariant and any two nodes define nonoverlapping index intervals. The subgraph induced by A and such a subset of B has no $K_{2,2}$. If n_i denotes the size of the ith subset of B then the number of arcs/cuts in the entire graph is bounded by

$$\sum_{i=1}^{\binom{m+2}{2}} (n_i\sqrt{mn} + mn) = O(m^{3/2}n^{3/2} + m^3 n).$$

The assertion follows because $m = n^{1/5}$. □

7. A point-line incidence problem

For a weaving that is not bipartite it is still possible to associate a cycle with its polygon. In this more general case, a cycle is called *elementary* if its polygon is a face of the line arrangement. Clearly, there are at most $O(n^2)$ elementary cycles because there are at most that many faces. Currently no better upper bound on the number of elementary cycles is known. To shed some light on this problem we modify it somewhat and study lines in space and their intersection patterns.

Given a set of n lines in space, a point is called a *joint* if it lies on (at least) three noncoplanar lines of the set. It is fairly easy to see that $\Omega(n^{3/2})$ is a lower bound for the maximum number of joints defined by n lines. Take k planes in general position in space. They define $n = \binom{k}{2}$ lines and $\binom{k}{3} = \Omega(n^{3/2})$ vertices, each a joint.

The connection between this point-line incidence extremum problem and counting cycles is that a slight perturbation of the lines can make a joint an elementary 3-cycle. The following upper bound on the number of joints is taken from [6].

THEOREM 7.1. *A set of n lines in space defines at most* $O(n^{7/4})$ *joints.*

PROOF. Let L be the set of n lines, define $A = \{\{a, b\} \in \binom{L}{2} \mid a \cap b \neq \emptyset\}$, let $G = (L, A)$ be the intersection graph of L, and set $k = n^{1/4}$. Note that each joint is "witnessed" by at least one arc in A. The proof modifies G in three steps and finally presents the counting argument.

STEP 1. If a joint is incident to at least k lines then remove the $\binom{k}{2}$ arcs witnessing the joint. This creates at most $\frac{n(n-1)}{k(k-1)} = O(n^{3/2})$ *orphans*, that is, joints not witnessed by any arc.

STEP 2. If a plane contains at least k lines then remove the corresponding nodes, together with incident arcs. A plane contains fewer than n joints, which now possibly become orphans. This creates at most $\frac{n^2}{k} = O(n^{7/4})$ orphans.

STEP 3. If a quadric contains at least k lines then remove the corresponding nodes with incident arcs. A quadric contains fewer than $2n$ joints which now possibly become orphans. This creates at most $\frac{2n^2}{k} = O(n^{7/4})$ orphans.

The remaining graph contains no $K_{3,2k}$. Because if there are lines a_1, a_2, a_3 and $b_1, b_2, \ldots, b_{2k}$ that define a $K_{3,2k}$ then some k of the lines b_i either go through a common vertex or lie in a common plane or quadric, a contradiction. Using the lemma of §5 we get

$$|A| = O(k^{1/3}n^{5/3}) = O(n^{7/4}). \quad \square$$

8. Bichromatic problems

As mentioned in §1, for many algorithmic problems in space it is important to efficiently test the relative position of lines or rods. One example is the *ray-shooting problem*: given a collection of polytopes and a ray, determine which polytope intersects the ray closest to its starting point. We refer to Agarwal and Sharir [1] and Pellegrini [12] who use Plücker coordinates, explained in §12, for solutions to this problem. Other examples of such problems deal with so-called *(polyhedral) terrains*, that are continuous and piecewise linear maps from $\mathbf{R}^2$ to $\mathbf{R}$.

Given two terrains, one might want to determine whether or not they intersect, or compute their intersection, or compute their pointwise maximum; see Figure 8.1. These problems will be addressed in §§13–15. The underlying data structures, algorithms, and geometric techniques will be discussed in §§9–12. A particularly important data structure in this context is the segment tree and some of its variants as described in §§9 and 11. This structure facilitates the reduction of problems for polyhedral terrains and rods to problems

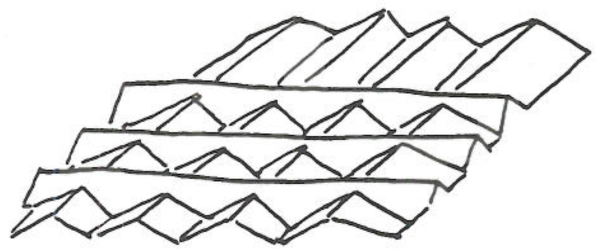

FIGURE 8.1. The pointwise maximum of two terrains.

for lines. The latter will be treated with algorithmic techniques based on the use of Plücker coordinates to represent lines.

9. The segment tree

The segment tree is a (one-dimensional) data structure storing a set S of n intervals (see e.g. [**5**, **14**]). The n intervals have $k \leq 2n$ endpoints, and the k endpoints decompose $\mathbf{R}^1$ into $k+1$ *atomic segments*. The *segment tree* is a minimum height, ordered binary tree with $k+1$ leaves corresponding to the atomic segments, from left to right. The *segment* $\sigma(\kappa)$ of a node κ is the corresponding atomic segment if κ is a leaf, and $\sigma(\mu) \cup \sigma(\nu)$ if μ and ν are the children of κ. Finally, for every node μ with parent κ (if it exists) we define $L_\mu = \{I \in S \mid \sigma(\mu) \subseteq I \text{ and } \sigma(\kappa) \not\subseteq I\}$, the *list* of μ, see Figure 9.1. Here is a short list of basic properties of the segment tree of S.

1. An interval I belongs to the list of at most two nodes per level, thus it belongs to at most $2\log_2 n + O(1)$ lists or nodes.
2. The segments $\sigma(\mu)$ of the nodes μ with $I \in L_\mu$ define a partition of I.

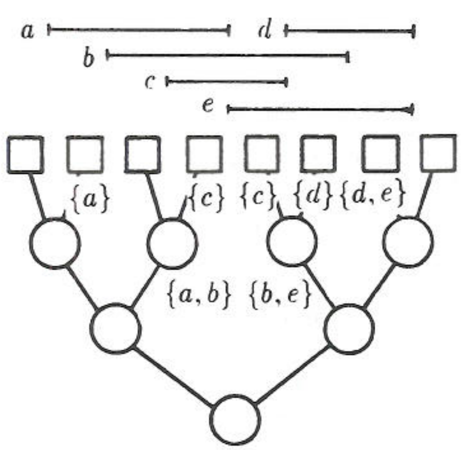

FIGURE 9.1. The segment tree defined by five intervals with a total of seven endpoints.

3. For a point p let $\rho = \mu_0, \mu_1, \ldots, \mu_h = \lambda$ be the path starting at the root ρ and ending at the leaf λ with $p \in \sigma(\lambda)$. Then $\{I \in S \mid p \in I\} = \bigcup_{i=0}^{h} L_{\mu_i}$.

Note that the lists of the nodes on the path in property 3 are disjoint, by definition. Define $L^*_\mu = \{I \in S \mid \sigma(\mu) \cap I \neq \emptyset \text{ and } \sigma(\mu) \not\subseteq I\}$.

4. $I \in L^*_\mu$ iff μ is ancestor of a node ν with $I \in L_\nu$.
5. An interval I belongs to lists L^* of at most two nodes per level of the segment tree.

Properties 1 and 5 imply that the segment tree, with lists L_μ and L^*_μ stored at nodes μ, takes storage $O(n \log n)$ to represent a set of n intervals.

10. A bichromatic intersection lemma

Let B be a set of m blue and R be a set of n red line segments in $\mathbf{R}^2$, with the property that no two line segments of the same color intersect. We use a single segment tree to store B and R. The segment tree is defined for the vertical projections of the line segments onto the x-axis, assuming no line segment is vertical. However, instead of intervals, the corresponding *line segments* are stored in the lists L and L^*. Accordingly, we extend the definition of $\sigma(\mu)$ from a one-dimensional interval to the vertical slab that intersects the x-axis in this interval. The blue and red line segments are stored in separate lists, so for each node ν we have lists B_ν, B^*_ν, R_ν, R^*_ν, defined just as L_ν and L^*_ν. The following fairly straightforward but very useful result is taken from [5].

LEMMA 10.1. *For every pair $b \cap r \neq \emptyset$, $b \in B$ and $r \in R$, there is a unique node ν with*

(i) *$b \cap r \in \sigma(\nu)$, and*
(ii) *$(b \in B_\nu$ and $r \in R_\nu)$ or $(b \in B_\nu$ and $r \in R^*_\nu)$ or $(b \in B^*_\nu$ and $r \in R_\nu)$.*

PROOF. For $p = b \cap r$ consider the path $\rho = \nu_0, \nu_1, \ldots, \nu_h = \lambda$ with $p \in \sigma(\lambda)$. Define i so that $b \in B_{\nu_i}$ and j so that $r \in R_{\nu_j}$. If $i = j$ then $\nu = \nu_i = \nu_j$, if $i < j$ then $\nu = \nu_i$, and if $i > j$ then $\nu = \nu_j$. □

The three cases in (ii) are disjoint which will be crucial in the upcoming applications of the lemma. For example, consider the problems of reporting and counting all pairs $(b, r) \in B \times R$ so that $b \cap r \neq \emptyset$. The solution sketched below is based on the segment tree for $B \cup R$ (with lists B_ν, B^*_ν, R_ν, R^*_ν per node ν). The above lemma admits a reduction to reporting/counting, for each node ν, the pairs $(b, r) \in (B_\nu \times R_\nu) \cup (B_\nu \times R^*_\nu) \cup (B^*_\nu \times R_\nu)$ that intersect. The intersecting pairs in $B_\nu \times R^*_\nu$ can be computed as follows; the other two cases are similar.

STEP 1. Sort the line segments in B_ν from top to bottom. STEP 2. Use binary search to locate the endpoints of the (pieces of the) line segments in R^*_ν amids the line segments in B_ν.

The amount of storage needed for the segment tree is $O(n \log n)$, and the time is $O(n \log^2 n)$ because there are endpoints of $O(n \log n)$ (pieces of) red line segments to be located. Also, sorting the lists B_ν and R_ν costs time $O(n \log^2 n)$. Using refined algorithmic techniques, the storage can be improved to $O(n)$ (by traversing the segment tree rather than storing it), and the time can be brought down to $O(n \log n)$ (using merging and fractional cascading). Details can be found in [5].

11. Another layer

The above bichromatic line segment intersection problem can also be solved by adding another layer to the segment tree. In effect, this defines coverings $B = \bigcup_{i=1}^{k} B_i$ and $R = \bigcup_{i=1}^{k} R_i$ with the following properties.

1. $b \cap r \neq \emptyset$ iff there is a unique index $1 \leq i \leq k$ with $b \in B_i$ and $r \in R_i$.
2. All $b \in B_i$ meet all $r \in R_i$ in the same sequence, and vice versa.
3. $k = O(N \log N)$ with $N = m + n$.
4. $\sum_{i=1}^{k} |B_i| + |R_i| = O(N \log^2 N)$.

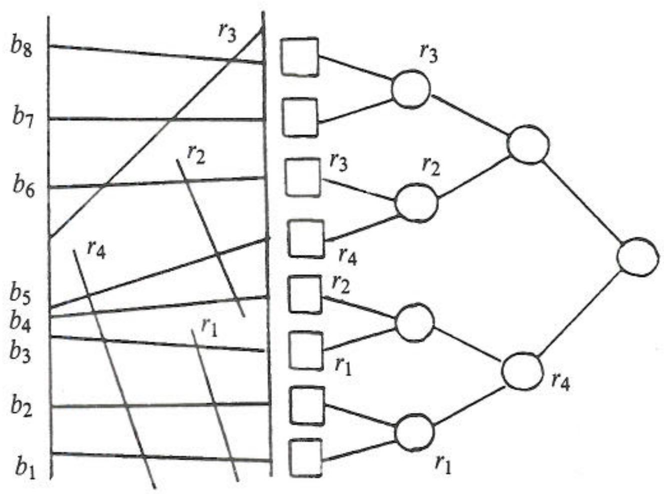

FIGURE 11.1. A slab $\sigma(\nu)$ with sets B_ν and R^*_ν is shown. The line segments in B_ν can be ordered from bottom to top, each corresponding to a leaf of a secondary segment tree. A blue line segment b is stored at each ancestor of its leaf. A line segment in R^*_ν intersects a contiguous subsequence of the line segments in B_ν and is stored in the nodes that cover this subsequence, as for ordinary segment trees.

It should be clear that after constructing the two coverings with the above properties it is easy to report or count intersecting pairs. Figure 11.1 illustrates the definition of the coverings. The significance of the coverings of B and R is that they can be used to reduce a bichromatic line segment problem into "few" and "not too big" subproblems which have the additional structure expressed in property 2 above.

12. Plücker coordinates

A (directed) line in space can be given by the homogeneous coordinates of two of its points:

$$\ell = \begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_0 & \beta_1 & \beta_2 & \beta_3 \end{pmatrix}, \quad \text{with } \alpha_0, \beta_0 > 0,$$

$$\ell' = \begin{pmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \gamma_3 \\ \delta_0 & \delta_1 & \delta_2 & \delta_3 \end{pmatrix}, \quad \text{with } \gamma_0, \delta_0 > 0.$$

The eight coordinates of a line can be arranged in a 2-by-4 matrix, as above, and by taking the determinants of the six 2-by-2 submatrices we obtain the *Plücker coordinates* of the line:

$$p(\ell) = (\pi_{01}, \pi_{02}, \pi_{03}, \pi_{12}, \pi_{13}, \pi_{23}), \quad \text{with } \pi_{ij} = \det \begin{pmatrix} \alpha_i & \alpha_j \\ \beta_i & \beta_j \end{pmatrix}.$$

We call $p(\ell)$ the *Plücker point* of ℓ, given by homogeneous coordinates in projective five-dimensional space, P^5. Similarly, the π_{ij} can be interpreted as coefficients of a hyperplane:

$$v(\ell) = (\pi_{23}, -\pi_{13}, \pi_{12}, \pi_{03}, -\pi_{02}, \pi_{01}),$$

$$h(\ell) = \{p \mid \langle v(\ell), p\rangle = 0\} \quad \text{and} \quad h^+(\ell) = \{p \mid \langle v(\ell), p\rangle \geq 0\}.$$

Observe that ℓ and ℓ' intersect (or are parallel which we interpret as an intersection at infinity) iff the four defining points are coplanar, which is the case iff the determinant of the 4-by-4 matrix defined by their sixteen coordinates is zero. This determinant is obtained by plugging $p(\ell)$ into the equation of $h(\ell')$, or $p(\ell')$ into the equation of $h(\ell)$. This implies the following basic properties; see **[4]**.

LEMMA 12.1. (1) *$\ell \cap \ell' \neq \emptyset$ iff $p(\ell) \in h(\ell')$ (or, equivalently, $p(\ell') \in h(\ell)$).*

(2) *If ℓ and ℓ' are oriented from left to right and ℓ is cw to ℓ' (in the projection onto the xy-plane), then ℓ lies above ℓ' iff $p(\ell) \in h^+(\ell')$.*

13. The relative position of lines

The segment tree reduction outlined in §§10 and 11 in combination with Plücker coordinates as introduced in §12 turn out to be powerful tools in attacking problems for rods in space. So let B and R be two sets of lines in $\mathbf{R}^3$, each line oriented consistently with the x-axis, so that in the orthogonal projection onto the xy-plane each line in R is cw to each line in B. Define $m = |B|$, $n = |R|$, and $N = m + n$. The following facts are taken from [4] and are useful when the above tools are put to work.

1. A line $r \in R$ lies above all lines in B iff $p(r) \in \mathscr{P}(B) = \bigcap_{b\in B} h^+(b)$.
2. $\mathscr{P}(B)$ is a convex cone in $\mathbf{R}^6$ (a polytope in P^5) with $O(m^2)$ faces.
3. There is a data structure (which we call the *envelope structure*) that stores B in storage $O(m^{2+\epsilon})$ so that for a given r we can decide in $O(\log m)$ time whether r lies above all $b \in B$.

Sets R and B are said to have the *towering property* if all $r \in R$ lie above all $b \in B$.

4. There is a randomized algorithm that tests in expected time $O(N^{4/3+\epsilon})$ whether or not R and B have the towering property.

14. Detecting the intersection of terrains

Here is a problem that can be solved by application of the segment tree in combination with Plücker coordinates. A *terrain* is a continuous function from $\mathbf{R}^2$ to $\mathbf{R}$ and it is *polyhedral* if it is piecewise linear. For two given polyhedral terrains, Σ_1 and Σ_2, with m and n edges in space, the problem is to determine whether or not they intersect.

Σ_1 is above Σ_2 iff first, every vertex of Σ_1 lies above Σ_2, second, every vertex of Σ_2 lies below Σ_1, and third, if $b \in \Sigma_1$ and $r \in \Sigma_2$ are two edges whose projections onto the xy-plane intersect then b lies above r. This characterization suggests the following high-level algorithm. Primes are used to denote orthogonal projections onto the xy-plane.

STEP 1. Locate every vertex of Σ_1' in Σ_2', and vice versa, and test points versus facets.

STEP 2. Use the two-layered segment tree for the line segment sets of Σ_1' and Σ_2' (see §11) and test the towering property for each pair $(B_i, R_i)_{i=1,2,\ldots,k}$ using the algorithm suggested in §12.

Step 1 takes time $O(N \log N)$ using any one of the optimal point location algorithms available in the literature. Step 2 takes expected time $O(N^{4/3+\epsilon})$.

15. The pointwise maximum of terrains

In the worst case, the *pointwise maximum* (or the *upper envelope*) Σ of Σ_1 and Σ_2 has $\Theta(mn)$ edges (see Figure 8.1), and a trivial algorithm can construct it in this time. For a given instance, let k be the number of edges of Σ.

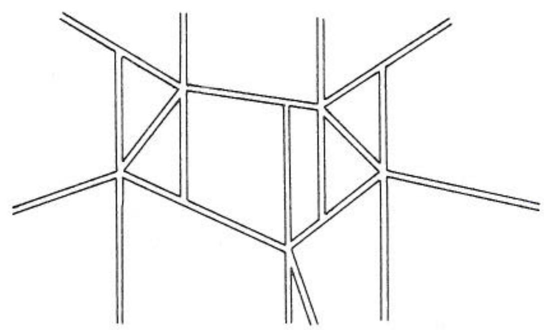

FIGURE 15.1. Projecting, decomposing into trapezoids, and shrinking.

THEOREM 15.1. *There is a randomized algorithm that constructs* Σ *in expected time* $O(N^{3/2+\epsilon} + k \log^2 N)$ *and storage* $O(N)$.

Σ can be constructed by reduction to edge-facet intersections. For one thing, Σ consists of pieces of Σ_1 and pieces of Σ_2 pasted together along curves of $\Sigma_1 \cap \Sigma_2$. Each vertex of such a curve is the intersection between an edge of Σ_1 and a facet of Σ_2, or vice versa. After one such vertex is found, the entire curve can be completed by tracing the intersection, edge-by-edge. The following representation of a polyhedral terrain Σ_1 is used to obtain the above result. Project Σ_1 onto the xy-plane, decompose the regions into trapezoids, and shrink the regions symbolically a tiny amount, as shown in Figure 15.1. Now store the nonvertical sides of the trapezoids as a collection of blue pairwise nonintersecting line segments in a two-layered segment tree with lists B_μ only. Observe that the upper and lower sides of a trapezoid are stored in exactly the same lists B_μ, and that B_μ can be viewed as an ordered list of line segments spanning $\sigma(\mu)$ or, alternatively, as an ordered list of trapezoids. Finally, store a list B_μ as an envelope structure if $|B_\mu| \le N^{(1-\epsilon)/2}$. A list takes $O(N)$ randomized time and storage, and altogether this adds up to $O(N^{3/2+\epsilon})$. We find intersections between red edges and blue facets as follows.

1. Distribute each line segment $r \in R$ (the projection of Σ_2) to the appropriate nodes μ.
2. Query B_μ, that is,
   - 2.1. if μ is a leaf then test r against the only trapezoid; report if there is an intersection;
   - 2.2. if B_μ is not stored as an envelope structure then recurse for both children;
   - 2.3. if B_μ is stored as an envelope structure then
     Case 1. r is above all $b \in B_\mu$; return;
     Case 2. r is not above all $b \in B_\mu$; recurse for both children.

The overhead per $r \in R$ is the time spent at lists not stored as envelope structures, which is $O(N^{(1+\epsilon)/2})$. For each intersection we follow a path with queries, so we spend another $O(\log^2 n)$ time per intersection, as claimed in the theorem. Details can be found in **[5]**.

REFERENCES

1. P. K. Agarwal and M. Sharir, *Applications of a new space partitioning technique*, Proc. Workshop on Algorithms and Data Structures, 1991 (to appear).
2. B. Bollobás, *Extremal graph theory*, Academic Press, London, 1978.
3. B. Chazelle, *How to search in history*, Inform. and Control **64** (1985), 77–99.
4. B. Chazelle, H. Edelsbrunner, L. J. Guibas, M. Sharir, and J. Stolfi, *Lines in space: combinatorics and algorithms*, Rept. UIUCDCS-R-90-1569, Dept. Comput. Sci., Univ. Illinois, Urbana, IL, 1990.
5. B. Chazelle, H. Edelsbrunner, L. J. Guibas, and M. Sharir, *Algorithms for bichromatic line segment problems and polyhedral terrains*, Rept. UIUCDCS-R-90-1578, Dept. Comput. Sci., Univ. Illinois, Urbana, IL, 1990.
6. B. Chazelle, H. Edelsbrunner, L. J. Guibas, R. Pollack, R. Seidel, M. Sharir, and J. Snoeyink, *Counting and cutting cycles of lines and rods in space*, Rept. UIUCDCS-R-90-1608, Dept. Comput. Sci., Univ. Illinois, Urbana, IL, 1990.
7. H. Edelsbrunner, *Algorithms in combinatorial geometry*, Springer-Verlag, Heidelberg, 1987.
8. H. Fuchs, Z. Kedem, and B. Naylor, *On visible surface generation by a priori tree structures*, Computer Graphics (SIGGRAPH'80 Conference Proceedings), pp. 124–133.
9. M. H. Overmars and M. Sharir, *Output-sensitive hidden surface removal*, Proc. 30th Ann. IEEE Sympos. Found. Comput. Sci. 1989, pp. 598–603.
10. J. Pach, R. Pollack, and E. Welzl, *Weaving patterns of lines and line segments in space*, Lecture Notes in Comput. Sci., vol. 450, Springer-Verlag, 1990, pp. 439–446.
11. M. S. Paterson and F. F. Yao, *Binary partitions with applications to hidden-surface removal and solid modelling*, Proc. 5th Ann. Sympos. Comput. Geom. 1989, pp. 23–32.
12. M. Pellegrini, *Ray shooting and isotopy classes of lines in 3-dimensional space*, Proc. Workshop on Algorithms and Data Structures, 1991 (to appear).
13. J. Plücker, *On a new geometry of space*, Philos. Trans. Royal Soc. London **155** (1865), 725–791.
14. F. P. Preparata and M. I. Shamos, *Computational geometry—an introduction*, Springer-Verlag, New York, 1985.
15. F. P. Preparata and R. Tamassia, *Efficient point location in a convex spatial cell complex*, Rept. CS-89-47, Dept. Comput. Sci., Brown Univ., Providence, RI, 1989.
16. E. Szemerédi and W. T. Trotter, *Extremal problems in discrete geometry*, Combinatorica 3 (1983), 381–392.

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN, URBANA, ILLINOIS 61801