SIAM J. COMPUT
Vol. 12, No. 1, February 1983

0097-5397/83/1201-0002 $01.25/0

OPTIMAL SEARCH IN PLANAR SUBDIVISIONS*

DAVID KIRKPATRICK†

Abstract. A planar subdivision is any partition of the plane into (possibly unbounded) polygonal regions. The subdivision search problem is the following: given a subdivision S with n line segments and a query point P, determine which region of S contains P. We present a practical algorithm for subdivision search that achieves the same (optimal) worst case complexity bounds as the significantly more complex algorithm of Lipton and Tarjan, namely $O(\log n)$ search time with $O(n)$ storage. Our subdivision search structure can be constructed in linear time from the subdivision representation used in many applications.

Key words. computational geometry, analysis of algorithms, point location, planar graphs, hierarchical search

1. Introduction. Any finite collection of finite, semi-infinite or infinite line segments induces a partition of the plane into polygonal regions. We will restrict our attention, for the present, to collections of line segments whose pairwise intersections are restricted to segment endpoints. We call such a collection (or the finite set of polygonal regions induced by the collection) a (*planar*) *subdivision*.

We define the *subdivision search problem* to be the following: Given a subdivision S with n line segments and an arbitrary query point P, determine which region of S contains P. Our subdivision search problem is equivalent to the "region-searching" problem of Dobkin and Lipton [6]. It is a slight (but, as we shall see, inconsequential) generalization of both the "point-location" problem studied by Lee and Preparata [14] and the "triangle" problem of Lipton and Tarjan [18]. The "point in polygon" problem [1], [3], [24] (given a simple polygon, does it contain a specified query point?), the "rectangle searching" problem [27] (given a set of nonoverlapping rectangles, which, if any, contains a specified query point?), and the "line searching" problem [6] (given a set of lines in the plane, which, if any, contains a specified query point?) can all be formulated as instances of our subdivision search problem.

Dobkin and Lipton [6] were the first to cast Knuth's [12] "post-office" problem (given a set of points in the plane, which is closest to a specified query point?) as a subdivision search problem. Shamos [25] (and independently Dewdney [4]) refined this formulation by introducing the Voronoi diagram of a point set, a planar subdivision of remarkable utility in connection with nearest neighbor and other related problems.

In many applications, a planar subdivision is the object of numerous location queries. For this reason, algorithms for point location are generally characterized by three attributes: i) *preprocessing time*—the time required to construct a search structure from a standard representation of S; ii) *space*—the storage used in the construction and representation of the search structures; and iii) *search time*—the time required to locate a specified query point, given the search structure. We restrict our attention here to the worst-case behaviour of these attributes.

Dobkin and Lipton [6] employ a projective technique to reduce subdivision search to linear search. The resulting algorithm is asymptotically optimal (among comparison-based algorithms) in terms of search time but may be quite expensive in terms of both preprocessing time and storage. Specifically, Dobkin and Lipton provide an $O(\lg n)$[1]

* Received by the editors November 24, 1981. This work was supported in part by the National Sciences and Engineering Research Council of Canada, grant A3593.

† Department of Computer Science, University of British Columbia, Vancouver, British Columbia, Canada V6T 1W5.

[1] lg denotes $\log_2$.

search-time, $O(n^2)$ space, and $O(n^2 \lg n)$ preprocessing-time algorithm for subdivision searching. Dobkin and Lipton were also the first to raise the question: Can subdivision searching be done with $O(\lg n)$ search-time and $O(n)$ (or even $O(n \lg n)$) space?

Shamos [25] introduces an $O((\lg n)^2)$ search-time, $O(n)$ space, and $O(n \lg n)$ preprocessing-time algorithm suitable for searching a class of subdivisions that includes Voronoi diagrams. Employing an $O(n \lg n)$ algorithm for constructing a Voronoi diagram on n points [28], this leads to an $O((\lg n)^2)$ search-time, $O(n)$ space, and $O(n \lg n)$ preprocessing-time solution of the "post-office" problem. Shamos' algorithm is generalized by Lee and Preparata [15] to an $O((\lg n)^2)$ search-time, $O(n)$ space, and $O(n \lg n)$ preprocessing-time algorithm for the location in arbitrary subdivisions. Lee and Preparata's approach is divide-and-conquer; each reduction of the subdivision is achieved by discrimination of the query point with respect to a monotone chain of edges that splits the subdivision (at a cost of $O(\lg n)$ comparisons, in the worst case).

The first affirmative answer to the question of Dobkin and Lipton was provided by Lipton and Tarjan [18]. Lipton and Tarjan's $O(\lg n)$ search time, $O(n)$ space, and $O(n \lg n)$ preprocessing time algorithm for search in arbitrary triangular subdivisions (each interior region of the subdivision is bounded by exactly three line segments) is one of many important applications of their planar separator theorem [18], [19]. That general subdivision search can be efficiently reduced to triangular subdivison search follows from the $O(n \lg n)$ polygon triangulation algorithm of Garey et al. [8]; the details of this reduction are discussed in § 4. Unfortunately, Lipton and Tarjan's algorithm is of primarily theoretical interest; to quote Lipton and Tarjan [19], "We do not advocate this algorithm as a practical one, but its existence suggests that there may be a practical algorithm with an $O(\lg n)$ time bound and $O(n)$ space bound".

A recent result of Preparata [21] claims to come "very close to providing a complete substantiation" of Lipton and Tarjan's conjecture. Preparata's algorithm, which he describes as an evolution of the approach of Dobkin and Lipton [6], uses $O(\lg n)$ search time, $O(n \lg n)$ space, and $O(n \lg n)$ preprocessing time.

The purpose of this paper is to affirm Lipton and Tarjan's conjecture; we present a new subdivision search algorithm with exactly the same asymptotic bounds as Lipton and Tarjan's algorithm. The simplicity of our approach (and the existence of an implementation) suggests that it may also deserve to be called practical. A discussion of the implementation and more detailed evaluation of our algorithm will be presented elsewhere.

In the next section we present some preliminary definitions and comments on the data structures used by our algorithm. Sections 3, 4 and 5 describe our algorithm and a number of its applications. Section 6 concludes the paper with a discussion of some related open problems.

2. Definitions and preliminaries. A *finite planar subdivision* is a planar subdivision each of whose line segments is finite. Such a subdivision is indistinguishable from a straight-line embedding of a planar graph. Thus we can refer without confusion, not only to the vertices, edges and regions (or faces) of a finite planar subdivision, but also to such graph-theoretic notions as degree, incidence and independence [10]. It is an immediate consequence of Euler's formula (cf. [10]) that the numbers of vertices and edges of a finite planar subdivision are linearly related, and hence either number serves to characterize the size of such a subdivision. Hereafter, $|S|$ will denote the number of vertices of the finite subdivision S.

Let S be a finite planar subdivision. We take as a starting point for our algorithm what we call an *edge-ordered representation* of S. Specifically:

(a) if x is a line segment joining vertex v to vertex w, then x is represented by the pair of directed edges $\{(v, w), (w, v)\}$;

(b) each vertex v has associated with it not only its coordinates but also a list, in counterclockwise order, of all directed edges whose source is v; and

(c) each directed edge (v, w) has associated with it a pointer to the edge (w, v) as well as the name of the region lying immediately to the right of (v, w).

An edge-ordered representation is provided either implicitly or explicitly by the representations taken as standard in a number of earlier papers [20], [23]. It differs from the basic (unordered) list of adjacencies chosen by Lee and Preparata [15] as their initial representation. However, it should be clear that:

i) it occupies $O(|S|)$ space;

ii) it can be constructed in $O(|S| \lg |S|)$ time from a list of adjacencies or other standard representations of planar graphs;

iii) it can be constructed in $O(|S|)$ time if the underlying planar graph has bounded degree; and

iv) it can be constructed in $O(|S|)$ time from the natural graph representation provided in certain applications (cf. § 5).

Thus our choice of representation for subdivisions is intended to allow a realistic estimate of actual preprocessing costs.

The obvious redundancy in an edge-ordered representation can be neatly exploited in the development of our hierarchical search structure. A detailed description of the data structures used in one efficient implementation or our algorithm will be presented elsewhere.

A finite planar subdivision S has exactly one unbounded region, called the *external region* of S. Its complement is called the *interior* of S. The edges bounding the external region define what we call the *boundary* of S.

A *convex subdivision* is any finite planar subdivision whose interior is convex and whose interior regions are all convex. A *triangular subdivision* is a special case of a convex subdivision in which each region (including the exterior region) is bounded by three line segments. It is easily confirmed that a triangular subdivision on $n \geqq 3$ vertices has exactly $3n-6$ edges and $2n-4$ regions (including the external region).

In § 3, we give a new constructive proof of the following:

THEOREM 3.1. *There is an $O(\lg n)$ search time, $O(n)$ space and $O(n)$ preprocessing time algorithm for the triangular subdivison search problem.*

This result is extended to arbitrary planar subdivisions in § 4.

3. Fast search in triangular subdivisions. Let S be an arbitrary triangular subdivision with n vertices. A *subdivision hierarchy* associated with S is a sequence $S_1, S_2, \cdots, S_{h(n)}$ of triangular subdivisions, where $S_1 = S$ and each region R of S_{i+1} is linked to each region R' of S_i for which $R' \cap R \neq \emptyset$ (the so-called *parents* of R in S_i), for $1 \leqq i < h(n)$. We call $h(n)$ the *height* of the subdivision hierarchy. Obviously the space required for a subdivision hierarchy is just the space required for the individual subdivisions $(O(\sum_{i=1}^{h(n)} |S_i|))$ plus the space used by the intersubdivision links.

Our basic point location algorithm involves a single pass through the subdivision hierarchy, locating the test point at each level. Let p denote an arbitrary test point.

ALGORITHM HIERARCHICAL SUBDIVISION SEARCH
CANDIDATES$_{h(n)} \leftarrow$ regions of $S_{h(n)}$
$R \leftarrow$ region in CANDIDATES$_{h(n)}$ containing p
$i \leftarrow h(n) - 1$
while $i > 0$ **do**
  CANDIDATES$_i \leftarrow$ parents(R)
  $R \leftarrow$ region in CANDIDATES$_i$ containing p
  $i \leftarrow i - 1$
report (region R)

Since membership in any triangular region can be tested in constant time, the complexity of this search procedure is $O(\sum_{i=1}^{h(n)}|\text{CANDIDATES}_i|)$. Obviously we are motivated to construct subdivision hierarchies in which both the height and the size of all CANDIDATE sets are minimized.

We start by constructing a subdivision hierarchy of height two.

LEMMA 3.1. *There exist positive constants c and d such that for any triangular subdivision S with $|S|>3$, a triangular subdivision T can be constructed in $O(|S|)$ time, satisfying*:

i) $|T|\leqq(1-1/c)|S|$, *and*

ii) *each region of T has at most d parents in S.*

Proof. Let v be any internal (nonboundary) vertex of S, and let $\deg(v)$ denote its degree. Then, exactly $\deg(v)$ regions of S are incident with v. The union of these regions, which we call the *neighborhood* of v, forms a star-shaped polygonal region with $\deg(v)$ bounding edges. Now, if v and its $\deg(v)$ incident edges are removed from S and the neighborhood of v is retriangulated (introducing $\deg(v)-3$ new edges) what results is a new triangular subdivision with $|S|-1$ vertices. It should be clear that, regardless of how the neighborhood of v is retriangulated, each new region intersects at most $\deg(v)$ regions of S. Of course, the simplification achieved by this vertex removal and retriangulation is minimal. However, if w is any vertex which is independent of (i.e. nonadjacent to) v in S, then the neighborhoods of v and w do not intersect except possibly along one or more edges of S. Hence, such a pair of vertices can be removed in parallel and the triangular subdivision that is created by retriangulating their vacated neighborhoods has the property that each of its regions intersects at most $\max\{\deg(v), \deg(w)\}$ regions of S. By identical reasoning, if $v_1, \cdots, v_t$ form an independent set of vertices in S, then the $|S|-t$ vertex triangular subdivision T formed by removing $v_1, \cdots, v_t$ and retriangulating all t vacated neighborhoods has the property that each of its regions intersects at most $\max\{\deg(v_j), 1\leqq j\leqq t\}$ regions of S. To complete the proof it suffices to show that if c and d are sufficiently large then an independent set $v_1, \cdots, v_t$ with $\deg(v_i)\leqq d$, $1\leqq i\leqq t$, and $t\geqq|S|/c$ can always be identified in $O(|S|)$ time. This is an immediate consequence of the following lemma. □

LEMMA 3.2. *There exist positive constants c and d such that every planar graph on n vertices has at least n/c independent vertices of degree at most d. Furthermore, at least n/c of these can be identified in $O(n)$ time.*

Proof. We make no attempt to optimize c and d here. (Their optimal values influence the asymptotic constants for each of space, preprocessing time, and search time, and some tradeoffs can be expected.) We have already noted that an n-vertex planar graph has at most $3n-6$ edges. Hence the average vertex degree is less than 6, and so less than half of the vertices have degree exceeding 11. Starting with the set V of vertices of degree at most 11 (which can be identified easily in linear time), a straightforward elimination procedure identifies an independent subset containing at least $|V|/12\geqq n/24$ vertices. □

Of course, a subdivision hierarchy of height two provides no significant simplification over the original subdivision. However, if Lemma 3.1 is applied iteratively, we are led to a subdivision hierarchy in which asymptotically improved (in fact, optimal) search is possible.

LEMMA 3.3. *There exist positive constants c and d such that, for any triangular subidivision S with n vertices, an associated subdivision hierarchy $S_1, \cdots, S_{h(n)}$ can be constructed in $O(n)$ time, satisfying*:

i) $|S_{h(n)}|=3$;

ii) $|S_{i+1}|\leqq(1-1/c)|S_i|$; *and*

iii) *each region of S_{i+1} has at most d parents in S_i.*

Proof. Immediate from Lemma 3.1. □

COROLLARY 3.1. *The subdivision hierarchy above has height $h(n) = O(\lg n)$ and uses $O(n)$ space in total.*

Proof. It suffices to note that the sequence $|S_1|, |S_2|, \cdots, |S_{h(n)}|$ forms a decreasing geometric progression. □

We now restate and prove our basic result.

THEOREM 3.1. *There is an $O(\lg n)$ search time, $O(n)$ space, and $O(n)$ preprocessing-time algorithm for the triangular subdivision search problem.*

Proof. We use the hierarchical subdivision search algorithm in conjunction with the subdivision hierarchy constructed in Lemma 3.3. By Lemma 3.3, the preprocessing time is $O(n)$. By Corollary 3.1, the total space is $O(n)$. By our earlier observations, the complexity of search is $O(|S_{h(n)}| + \sum_{i=1}^{h(n)-1} p_i)$ where $p_i = \max_{R \in S_{i-1}} (|\text{parents}(R)|)$. But, by Lemma 3.1 and Corollary 3.1, $S_{h(n)}$ and p_i are $O(1)$ and $h(n)$ is $O(\lg n)$, so the search time is $O(\lg n)$. □

4. Fast search in general subdivisions. In this section we consider the reduction of general subdivision searching problems to triangular subdivision search. Let S be an arbitrary planar subdivision. We can reduce the question of searching in S to searching in a finite planar subdivision by intersecting S with a sufficiently large triangle chosen to contain all intersections of line segments of S. The interior of this triangle is clearly a finite planar subdivision. The exterior can be searched using a straightforward generalization of binary search, exploiting the fact that none of the semi-infinite line segments intersect in this region. This reduction adds a factor of only $O(|S|)$ to both the preprocessing time and space and $O(\lg |S|)$ to the search time used in the resulting finite subdivision search problem. Hence the asymptotic complexities of general and finite subdivision searching are equivalent.

It remains to reduce finite subdivision searching to triangular subdivision searching. Let S be a finite planar subdivision. We can assume from the preceding reduction that the boundary of S is triangular. Let T be the subdivision formed from S by triangulating each interior region of S. The size of T remains proportional to the size of S and, since T is a refinement of S, the location of points in T immediately implies their location in S. In the general case T can be formed from S in time $O(|S| \lg |S|)$, using the general polygon triangulation algorithm of Garey et al. [8]. However, if the regions of S are all convex, or even star-shaped, a straightforward linear algorithm exists for constructing T. Thus, we have demonstrated the following:

THEOREM 4.1. *There is an $O(\lg n)$ search time, $O(n)$ space and $O(n \lg n)$ preprocessing time algorithm for the general subdivision search problem.*

THEOREM 4.2. *There is an $O(\lg n)$ search time, $O(n)$ space and $O(n)$ preprocessing time algorithm for the convex subdivision search problem.*

5. Applications. Earlier papers on subdivision search, notably [15], [21], have mentioned a number of applications. We recall and expand on a few of these here.

5.1. Point in polygon problem. A planar polygon is a special case of a finite planar subdivision. Theorem 4.1 gives an immediate $O(\lg n)$ search time, $O(n \lg n)$ preprocessing time and $O(n)$ space algorithm for testing the inclusion of an arbitrary point in an n vertex planar polygon. For convex or star-shaped polygons, or any other family of polygons that can be triangulated in $O(n)$ time, the preprocessing time is linear.

5.2. Point in convex polyhedron problem. Lee and Preparata [15] note that the problem of testing the inclusion of an arbitrary point in an n-vertex convex polyhedron can be reduced to convex subdivision search with $O(n)$ preprocessing. It follows, by

Theorem 4.2, that an $O(\lg n)$ search time, $O(n)$ preprocessing time and $O(n)$ space algorithm exists for the point in convex polyhedron problem.

By dualization, an algorithm with identical attributes can be formulated for the problem of testing for the intersection of an arbitrary plane and a polyhedron in 3-space [5].

5.3. Locating a set of points in a planar subdivision. Preparata [22] shows that a set of k points can be located on an n-vertex planar subdivision in $O(k \lg k + n + k \lg n)$ time, given $O(n \lg n)$ preprocessing time. This result is an immediate consequence of Theorem 4.1. Furthermore, by Theorem 4.2, the preprocessing time can be reduced to $O(n)$ for convex planar subdivisions (which arise in a principal application [20] of Preparata's batched point location algorithm).

5.4. Closest point problems. The problem of determining which of a set of data points is closest to a given test point has been extensively studied. Shamos [25] (and independently Dewdney [4]) show how this problem can be reduced to point location in a particular family of planar subdivisions known as Voronoi diagrams. Voronoi diagrams (in any L_p metric) can be constructed in $O(n \lg n)$ time [13]. While Voronoi diagrams in arbitrary L_p metrics may involve curved edges, every region is star-shaped and hence Voronoi point location can be solved using subdivision search followed by at most one test against a curved edge. Furthermore, only linear preprocessing is required following the construction of the Voronoi diagram. This fact can be exploited in the dynamic maintenance of Voronoi diagrams and dynamic solution of closest point problems [9].

By replacing Voronoi diagrams by what are called generalized Voronoi diagrams [11], [14] it is possible to use an analogous approach to solve the closest line problem (which of a set of lines or line segments is closest to a given test point?).

6. Open problems and conclusions. It is tempting to extend the approach of this paper to the location of points in higher-dimensional subdivisions. Such an extension is by no means obvious. The number of vertices, edges, faces and regions of three-dimensional subdivisions are not necessarily linearly related, and the analogue of triangulation (tetrahedralization) is not a straightforward process. A more detailed discussion of subdivision search in higher dimensions will be taken up elsewhere.

Our algorithm seems to depend on the fact that the given subdivision is formed out of straight line segments. While the algorithm can be adapted to certain other situations (for example, when all internal regions are star shaped), the general problem of optimal search in subdivisions formed from arbitrary curve segments may require a totally new approach. As a concrete example of such a subdivision, consider those subdivisions which arise in the so-called locus approach to the fixed-radius nearest neighbor search problem [2]. Such subdivisions are formed by the intersection of fixed-radius circles, and in general do not seem to admit a simple refinement using straight edges. Thus the fixed-radius nearest neighbor search problem still awaits an $O(\lg n)$ search time, $O(n^2)$ space and $O(n^2 \lg n)$ preprocessing time solution. A solution using $O(\log n)$ search time, $O(n^2 \log n)$ space and $O(n^2 \log n)$ preprocessing time is a byproduct of Preparata's subdivision search algorithm [21]. Edelsbrunner and Maurer [7] present search algorithms for subdivisions formed by segments other than straight lines.

We have described a new subdivision search algorithm which, as pointed out by Lipton and Tarjan [18], is optimal for both search time and space, assuming only binary decisions are possible. Our algorithm is based on the hierarchical decomposition of an arbitrary subdivision. It is conjectured that this technique will find a number of other applications in computational geometry and elsewhere. On this point we should acknowledge the fact that this technique does not originate with this paper: Lipton and Miller [17] use a very similar idea in developing a fast algorithm for coloring planar graphs.

REFERENCES

[1] J. L. BENTLEY AND W. CARRUTHERS, *Algorithms for testing the inclusion of points in polygons*, in Proc. 18th Annual Allerton Conference on Communication, Control and Computing, 1980, pp. 11–19.

[2] J. L. BENTLEY AND H. A. MAURER, *A note on Euclidean near neighbour searching in the plane*, Inform. Process. Lett., 8 (1979), pp. 133–136.

[3] W. BURTON, *Representation of many-sided polygons and polygonal lines for rapid processing*, Comm. ACM, 16 (1973), pp. 230–236.

[4] A. K. DEWDNEY, *Complexity of nearest neighbour searching in three and higher dimensions*, Rep. 28, Dept. Computer Science, Univ. of Western Ontario, London, Ontario, 1977.

[5] D. P. DOBKIN AND D. G. KIRKPATRICK, *Fast detection of polyhedral intersections*, Proc. 9th International Colloquium on Automata, Languages and Programming, Aarhus, Denmark, 1982, to appear.

[6] D. P. DOBKIN AND R. J. LIPTON, *Multidimensional searching problems*, this Journal, 5 (1976), pp. 181–186.

[7] H. EDELSBRUNNER AND H. A. MAURER, *On region location in the plane*, Rep. 52, Institut für Informationsverarbeitung, Technical Univ. Graz, Graz, Austria, 1980.

[8] M. R. GAREY, D. S. JOHNSON, F. P. PREPARATA AND R. E. TARJAN, *Triangulating a simple polygon*, Inform. Process. Lett., 7 (1978), pp. 175–179.

[9] I. G. GOWDA, D. G. KIRKPATRICK, D. T. LEE AND A. NAAMAD, *Dynamic Voronoi diagrams*, IEEE Trans. Inform. Theory, to appear.

[10] F. HARARY, *Graph Theory*, Addison-Wesley, Reading, MA, 1969.

[11] D. G. KIRKPATRICK, *Efficient computation of continuous skeletons*, in Proc. 20th Annual IEEE Symposium on Foundations of Computer Science, 1979, pp. 18–27.

[12] D. E. KNUTH, *The Art of Computer Programming, Vol. 3, Sorting and Searching*, Addison-Wesley, Reading, MA, 1973.

[13] D. T. LEE, *Two-dimensional Voronoi diagrams in the L_p-metric*, J. Assoc. Comput. Mach., 27 (1980), pp. 604–618.

[14] D. T. LEE AND R. L. DRYSDALE III, *Generalization of Voronoi diagrams in the plane*, this Journal, 10 (1981), pp. 73–87.

[15] D. T. LEE AND F. P. PREPARATA, *Location of a point in a planar subdivision and its applications*, this Journal, 6 (1977), pp. 594–606.

[16] D. T. LEE AND C. C. YANG, *Location of multiple points in a planar subdivision*, Inform. Process. Lett. 9 (1979), pp. 190–193.

[17] R. J. LIPTON AND R. E. MILLER, *A batching method for coloring planar graphs*, Inform. Process. Lett., 7 (1978), pp. 185–188.

[18] R. J. LIPTON AND R. E. TARJAN, *Applications of a planar separator theorem*, in Proc. 18th Annual IEEE Symposium on Foundations of Computer Science, 1977, pp. 162–170.

[19] ———, *A separator theorem for planar graphs*, SIAM J. Appl. Math., 36 (1979), pp. 177–189.

[20] D. E. MULLER AND F. P. PREPARATA, *Finding the intersection of two convex polyhedra*, Theoret. Comput. Sci., 7 (1978), pp. 217–236.

[21] F. P. PREPARATA, *A new approach to planar point location*, this Journal, 10 (1981), pp. 473–482.

[22] ———, *A note on locating a set of points in a planar subdivision*, this Journal, 8 (1979), pp. 542–545.

[23] F. P. PREPARATA AND S. J. HONG, *Convex hulls of finite sets of points in two and three dimensions*, Comm. ACM, 20 (1977), pp. 87–93.

[24] K. B. SALOMON, *An efficient point-in-polygon algorithm*, Comput. Geosci., 4 (1978), pp. 173–178.

[25] M. I. SHAMOS, *Geometric complexity*, in Proc. 7th Annual ACM Symposium on Theory of Computing, 1975, pp. 224–233.

[26] ———, *Computational geometry*, Ph.D. Thesis, Yale Univ., New Haven, CT, 1978.

[27] M. I. SHAMOS AND J. L. BENTLEY, *Optimal algorithms for structuring geographic data*, in Proc. Symposium on Topological Data Structures for Geographic Information Systems, Harvard Univ., Cambridge, MA, 1977, pp. 43–51.

[28] M. I. SHAMOS AND D. HOEY, *Geometric intersection problems*, in Proc. 17th Annual IEEE Symposium on Foundations of Computer Science, 1976, pp. 208–215.