Computing Morphological Properties of Arrangements of Lines

Godfried Toussaint

School of Computer Science
McGill University
Montreal

Published in: *Proceedings of the 2nd Japan International Symposium of the Society for the Advancement of Materials and Process Engineering*, (***SAMPE***)Chiba, Japan, December 11-14, 1991, pp. 1406-1411.

ABSTRACT

An arrangement of n lines in the plane is a partition of the plane into $O(n^2)$ faces, edges, and vertices (intersection points). Such *line processes* play a fundamental role in modeling spatial patterns and studying a variety of problems such as traffic flow. We briefly survey recent results on the complexity of computing morphological properties of such arrangements.

1. Introduction

The analysis of *line patterns* (sets of lines in the plane here referred to as *arrangements*) are of considerable interest to geographers [GB78], [Ba67], [Da67], [Ba75], nuclear physicists [Ho65], urban planners [Ga71], [SW72] and the pulp-and-paper industry [Mi64a], [Mi64b] among others. In the "spatial analysis literature" cited above the properties of arrangements that have traditionally been used to measure the similarities between arrangements, and thus the processes generating these arrangements, have been restricted to simple geometric properties such as the lengths of the line segments or the areas of the polygons induced by the arrangement [Mi64a], [SY72]. On the other hand, in the computational geometry literature there has been a flurry of activity in the area of computing more complex morphological properties of arrangements. In this paper we survey some of these recent results which will it is hoped pump new ideas and existing algorithms into the field of *spatial analysis.*

Let $\mathbf{L} = \{L_1, L_2,...,L_n\}$ be a finite set of lines in the plane where each line L_i is specified by an equation $Y=a_iX + b_i$ for some real numbers a_i, b_i, i=1,2,...,n. **L** induces a partition of the plane, known as the arrangement $A(\mathbf{L})$, into $O(n^2)$ faces, edges, and vertices. The vertices are the points where the lines in L intersect. Let p_{ij} denote the intersection point of L_i and L_j. The set $\mathbf{I} = \{p_{ij} \mid 1 \le i,j \le n\}$ denotes the set of all $O(n^2)$ such intersection points. The edges are the connected components of the lines that remain when the vertices are deleted. The faces are the connected components of the complement of the union of the lines $L_1, L_2,...,L_n$. For a detailed fundamental treatment of the combinatorial properties of arrangements the reader is referred to [Gr67] and [Ed87]

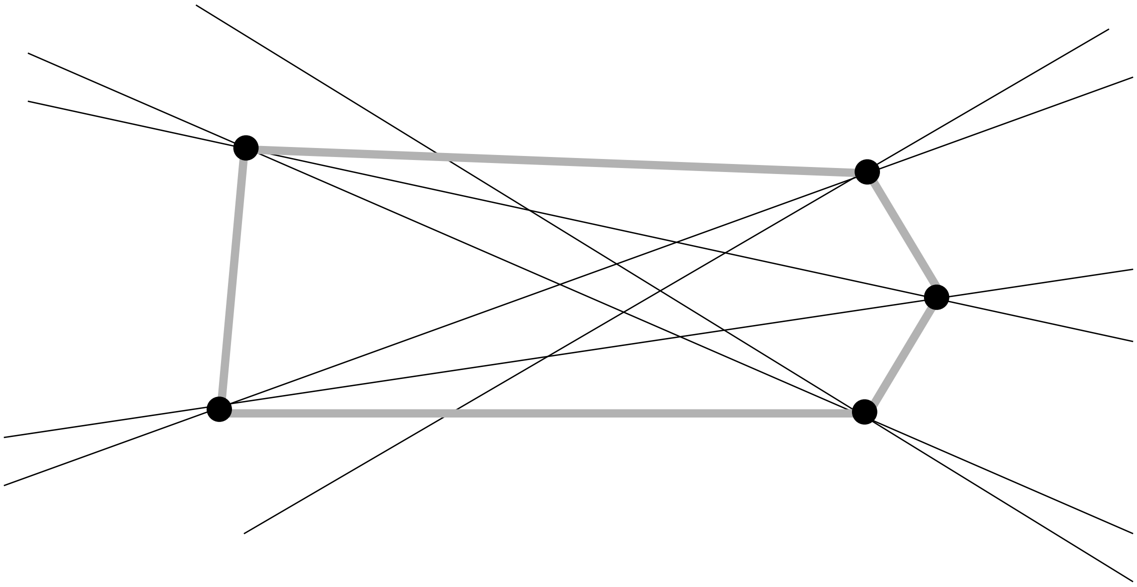

Fig. 1 An arrangement of six lines and their convex hull.

whereas a survey of recent research results can be found in [EGS90].

2. Convex Hulls of Arrangements

Definition: The intersection point p_{ij} for $i \neq j$ and $1 \leq i,j \leq n$ is said to be *extreme* with respect to line L_i if all other intersection points on L_i lie to one side of p_{ij}.

Definition: The intersection point p_{ij} for $i \neq j$ and $1 \leq i,j \leq n$ is said to be *critical* if and only if p_{ij} is extreme with respect to both lines L_i and L_j.

Definition: The *diameter of an arrangement*, denoted by D(*A*), is the maximum distance realized by a pair of points in **I**.

Lemma: (*Ching & Lee, 1985*) The diameter of an arrangement is determined by a pair of critical intersection points of **I**.

Definition: The *convex hull of an arrangement*, denoted by *CH*(*A*), is the convex hull of **I**.

Figure 1 illustrates an arrangement of six lines and the convex hull of **I**.

It is well known that the convex hull of n points in the plane can be computed in O(n log n) worst-case and linear expected time [DT81] under certain assumptions on the distributions of the points. Straight forward application of such algorithms to all the points in **I** thus leads to algorithms with $O(n^2 \log n)$ worst-case time, $O(n^2)$ expected time as well as $O(n^2)$ space under such assumptions. Surprisingly, Atallah [At86] and Ching & Lee [CL85] independently present an O(n log n) worst-case time algorithm with O(n) space for this problem. In [At86] and [CL85] it is shown that the vertices of the convex hull of **I** are a subset of the *critical* points determined by pairs of lines which are adjacent on a list in which they are sorted by slope. Therefore they first sort the lines by slope to obtain O(n) critical points and subsequently find their convex hull with any O(n log n) convex hull algorithm. Devroye and Toussaint [DT90] show that if we chose our convex hull algorithm carefully we can obtain an algorithm which will also exhibit O(n) expected complexity under a natural definition of a random line and almost any radially symmetric distribution on its parameters as well as under a model of computation that allows us to compute *floor* and *ceiling* functions in constant time.

Consider n i.i.d. random lines in the plane defined by their slope and distance from the origin. The slope is uniformly distributed on $[0,2\pi]$ and independent of the distance R from the origin. Let N_{ch} and N_{ol} be the number of points on the convex hull and outer layer (maximal vectors) of **I**, respectively. We will show below that there exist arrangements of lines in which N_{ol} = n(n-1)/2 = $O(n^2)$. It is shown in [DT90] that nevertheless N_{ol}, and therefore N_{ch}, have expected values O(1), and they give bounds that are uniform over all distributions of R with $0 < \mathbf{E}R < \infty$.

Therefore, if we first sort the lines using distributive partitioning [DK80] in O(n log n) worst-case, O(n) expected time and O(n) space, and subsequently find the convex hull of the adjacent critical points of **I** using the output-size sensitive algorithm of Kirkpatrick & Seidel [KS86] which has complexity O(k log h) where k is the number of input points and h is the cardinality of the convex hull, we obtain an algorithm for computing the convex hull of **I** in O(n log n) worst-case time, O(n) expected time and O(n) space. Furthermore, since once the convex hull has been obtained, the diameter can be found in O(n) time [To83], the above complexity results apply also to the problem of computing the diameter of an arrangement of n lines.

There has also been some work done on the *dynamic* convex hull problem for arrangements of lines. An algorithm was recently published [Bo90] for maintaining dynamically the convex hull of **I**. In other words, given the convex hull of **I** it is desired to *update* the convex hull when a new line is added to **L** (the *insertion* problem) and it is desired to update the convex hull when an old line is deleted from **L** (the *deletion* problem). No proofs of correctness are given in [Bo90]. In fact it is shown in [BET91] that the deletion algorithm proposed in [Bo90] is incorrect.

3. Maximal Vectors of Arrangements

Definition: We say that a point A in **I** *dominates* a point B in **I** if A is greater than B in both of its coordinates

Definition: A *maximal vector of an arrangement* A(**L**) is a point in **I** that is not dominated by any other point in **I**.

The above definition (stated in a *first-quadrant* version) can be modified to include maximal vectors in all four quadrants.

For an arbitrary set of n points in the plane it is well known that computing the maximal vectors has the same complexity as computing the convex hull. Surprisingly, for the case of arrangements computing maximal vectors is more difficult. The standard method [KLP75] clearly yields

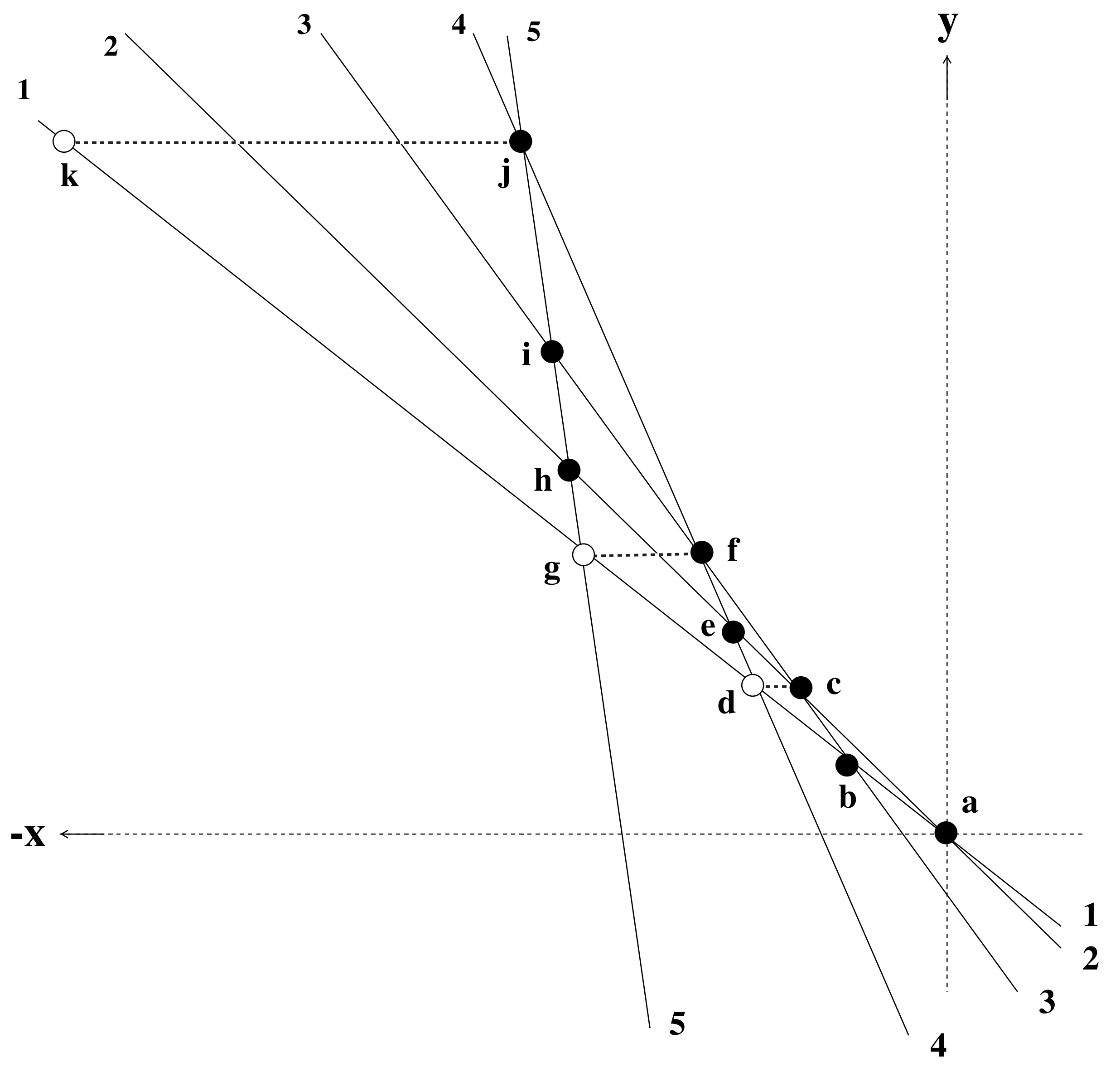

Fig. 2 Illustrating the construction that demonstrates that an arrangement of lines may have all its intersection points as maximal vectors.

an algorithm with $O(n^2 \log n)$ time and $O(n^2)$ space. However, it is difficult to improve on this for the following reason.

Lemma: An arrangement of lines $\mathbf{L} = \{L_1, L_2, \ldots, L_n\}$ may contain as many as $O(n^2)$ maximal vectors.

Proof: We provide a construction illustrated in Fig. 2 that has all points in **I** as maximal vectors.

Let the first point on **I** be point **a** located at the origin. We accomplish this by defining line **1** to have the equation y=-x, and line **2** to have the equation y=-(1+ε)x where ε is a small positive constant. Let θ denote the angle that the line with maximum slope makes (over all lines drawn so far) with the y axis measured in a counter clockwise manner. The third line has equation y=-(1+2ε)x - δ, where δ is a small positive constant. This consists of the three-step initialization phase of the construction and yields a triangle [**a**,**b**,**c**] such that all three vertices are maximal vectors and **c** has maximum y coordinate. We now show how to add, at step r, line **r** and create r-1 new maximal vectors. Let **d** denote the leftmost intersection point of a horizontal line collinear with **c**. Line **4** is then constructed to pass through point **d** and make an angle θ/2 with the y axis. This procedure is repeated until all n lines have been used up. For example, after line **4** has been inserted **f** is the intersection point with maximum y coordinate and **g** is the point on line **1** with y coordinate equal to that of **f**. Therefore line **5** passes through **g** and makes an angle θ/2 with the y axis, where θ is the angle made by line **4** with the y axis. Since at each step the angle with the y axis is decreased by half of the remaining angle all the lines in the arrangement have negative slope and this ensures that when we add a new line no *new* intersection point is dominated by any other *new* intersection point. By making each line pass through the point on line **1** that has the same y coordinate of the highest intersection point created thus far we ensure that no *new* intersection point is dominated by any other *old* intersection point. Therefore at each step all the new intersection points introduced are maximal vectors. Therefore all n(n-1)/2 intersection points are maximal vectors. Q.E.D.

It follows that even if we use the output-size sensitive algorithm of Kirkpatrick & Seidel [KS86] with O(k log v) complexity, where k is the number of input points and v is the number of maximal vectors found, our results imply an adaptive algorithm with O(n^2 log v) actual running time, O(n^2) expected time and O(n^2) space.

4. The Envelope of an Arrangement

Definition: The e*nvelope of an arrangement*, denoted by *E*(*A*), is the simple polygon whose boundary consists of the bounded edges of all the unbounded faces of *A*.

Fig. 3 illustrates an arrangement of six lines and their envelope. A naive approach to computing *E*(*A*) would first construct the arrangement *A*(**L**) (i.e., the data structure containing all the incidence relations between faces, edges and vertices) from the given n lines using the algorithm of Edelsbrunner, O'Rourke & Seidel [EOS83] and subsequently search the arrangement to find all the bounded edges of all the unbounded faces of *A*. This approach unfortunately leads to an algorithm requiring O(n^2) time and space. Suri [Su85] first established that the cardinality of *E*(*A*) is only O(n) and presented an O(n log n) time divide-&-conquer algorithm for computing *E*(*A*). Furthermore the Ω(n log n) lower bound for determination of the convex hull of **I** [CL85] holds also for computing *E*(*A*) and therefore Suri's algorithm is optimal. Unfortunately Suri's algorithm is complicated. However, since then at least two other O(n log n) algorithms have been discovered [Ve87], [Ke91] and the one due to Mark Keil [Ke91] is quite elegant and simple.

One can also define for an *arrangement* *A*(**L**) other traditional morphological properties (previously defined for sets of points or a polygon) such as the *geodesic* diameter and the *geodesic center* as the geodesic diameter of E(*A*) and the geodesic center of *E*(*A*), respectively. Since *E*(*A*) is a

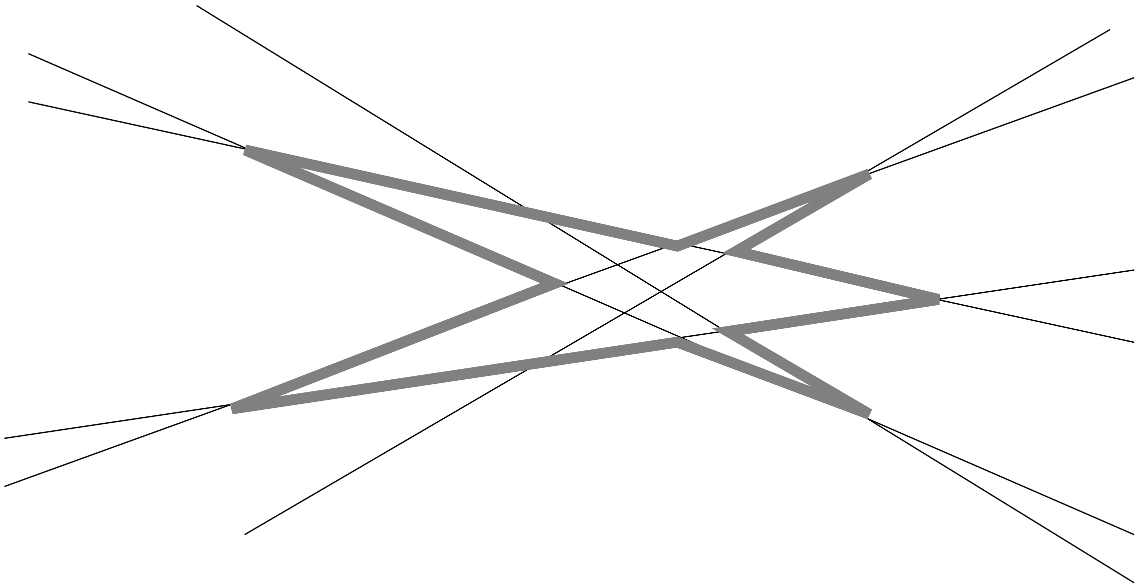

Fig. 3 An arrangement of six lines and their envelope.

simple polygon we can combine Keil's envelope algorithm with the O(n log n) time algorithms in [Su87] and [PSR89] to obtain O(n log n) time algorithms for both of these problems as well. On the other hand, $E(A)$ is not an arbitrary simple polygon and it is an open problem to determine if its structure will yield linear-time algorithms if $E(A)$ is given.

5. The Girth of an Arrangement

There are several ways in which one might measure the *girth* of an arrangement of lines. One obvious possibility is to use the shortest line segment that intersects every line of **L** (such a line segment is called a shortest *transversal*). Another is the smallest *disc* that intersects every line in **L**. Bhattacharya and Toussaint [BT90] were the first to explore this problem for the case of line segment transversals. In [BT90] they present an O(n $\log^2$ n) time and O(n) space algorithm for computing the shortest transversal of a set of n given lines or line segments in the plane.In the case of line segments that do not intersect the algorithm can be trimmed to run in O(n log n) time. Furthermore, in conjunction with convex hull and linear programming components the algorithm will also find the shortest line segment that intersects a set of n isothetic rectangles in O(n log k) time, where k is the combinatorial complexity of the space of transversals and $k \leq 4n$. These results find application in: (1) line-fitting between a set of n data ranges where it is desired to obtain the shortest *line-of-fit*, (2) finding the shortest line segment from which a convex n-vertex polygon is weakly externally visible, and (3) determining the shortest *line-of-sight* between two edges of a simple n-vertex polygon, for which O(n) time algorithms are also given All the algorithms are based on the solution to a fundamental geometric minimization problem that is of independent interest and promises to find application in several different contexts. Some extensions of these results can be found in [BCETSU91]. If the girth is measured as the smallest disc intersecting **L** then it is shown in [BJMR91] that it can be computed in O(n) time even for the case of line segments.

6. Other Properties of Arrangements

The above results imply that we can expect similar gains in the complexity of algorithms for computing other morphological or geometric structures that are inherently dependent on convex hulls. For example, define the *minimal enclosing rectangle of an arrangement*, denoted by *MER(A)*, as the rectangle of smallest area that encloses **I**. Straight forward application of the algorithm given in Toussaint [To83] to all the points in **I** leads to an O((n^2 log n) time algorithm. However, since the minimal enclosing rectangle of a set is equivalent to the minimal enclosing rectangle of the convex hull of the set it follows from the above results that *MER*(*A*) can be computed in O(n log n) time and O(n) space. For a second example define the maximum distance between two arrangements $A_1(\mathbf{L})$ and $A_2(\mathbf{L})$, denoted by $d_{\max}(A_1, A_2)$, as the maximum distance determined by an element of the set of intersection points of $A_1(\mathbf{L})$ and an element of the set of intersection points of $A_2(\mathbf{L})$. Straight forward application of the algorithms given in Bhattacharya & Toussaint [BT83] and Toussaint & McAlear [TM82] to all the points in I leads to algorithms that run in O(n^2 log n) time. However, since the maximum distance between two sets is equivalent to the maximum distance determined by a pair of points such that one is on the convex hull of one set and the other is on the convex hull of the other set, it follows from the above results that $d_{\max}(A_1, A_2)$ can be computed in O(n log n) time and O(n) space.

7. References

[At86] Atallah, M. J., "Computing the convex hull of line intersections," *Journal of Algorithms*, vol. 7, 1986, pp. 285-288.

[Ba67] Bartlett, M. S., "The spectral analysis of line processes," *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability,* vol. III, 1967, pp. 135-153.

[Ba75] Bartlett, M. S., *The Statistical Analysis of Spatial Pattern*, Chapman and Hall, London, 1975.

[BCETSU91] Bhattacharya, B. K., Czyzowics, J., Egyed, P., Toussaint, G. T., Stojmenovic, I. and Urrutia, J., "Computing shortest transversals of sets," *Proc. 7th ACM Symposium on Computational Geometry*, North Conway, New Hampshire, June 10-12, 1991.

[BET91] Bhattacharya, B. K., Everett, H. and Toussaint, G. T., "A counterexample to a dynamic algorithm for convex hulls of line arrangements," *Pattern Recognition Letters,* vol. 12, March 1991, pp. 145-147.

[BJMR91] Bhattacharya, B. K., Jadhav, S., Mukhopadhyay, A. and Robert, J. M., "Optimal algorithms for some smallest intersection radius problems," *Proc. 7th ACM Symposium on Computational Geometry*, North Conway, New Hampshire, June 10-12, 1991.

[BT83] Bhattacharya, B. K. and Toussaint, G. T., "Efficient algorithms for computing the maximum distance between two finite planar sets," *Journal of Algorithms*, vol. 4, 1983, pp. 121-136.

[BT90] Bhattacharya, B. K. and Toussaint, G. T., "Computing shortest transversals," Tech.

Rept. SOCS-90.6, McGill University, April 1990, also to appear in *Computing*, 1991.

[Bo90] Boreddy, J., "An incremental computation of the convex hull of planar line intersections," *Pattern Recognition Letters,* vol. 11, August 1990, pp. 541-543.

[CL85] Ching, Y. T. and Lee, D. T., "Finding the diameter of a set of lines," *Pattern Recognition*, vol. 18, 1985, pp. 249-255.

[Da67] Dacey, M. F., "Description of line patterns," *Northwestern Studies in Geography*, vol. 13, 1967, pp. 277-288.

[DK80] Devroye, L. and Klincsek, T., "Average time behavior of distributive sorting algorithms," *Computing*, vol. 26, 1980, pp. 1-7.

[DT81] Devroye, L. and Toussaint, G. T., "A note on linear expected time algorithms for finding convex hulls," *Computing,* vol. 26, 1981, pp. 361-366.

[DT91] Devroye, L. and Toussaint, G. T., "Convex hulls for random lines," Tech. Rept. SOCS-90.11, McGill University, May 1990, also to appear in the *Journal of Algorithms.*

[Ed87] Edelsbrunner, H., *Algorithms in Combinatorial Geometry*, Springer-Verlag, Heidelberg, 1987.

[EGS90] Edelsbrunner, H., Guibas, L., & Sharir, M., "The complexity and construction of many faces in arrangements of lines and of segments," *Discrete & Computational Geometry,* vol. 5, 1990, pp. 161-196.

[EOS83] Edelsbrunner, H., O'Rourke, J., & Seidel, R., "Constructing arrangements of lines and planes with applications," *24th FOCS,* 1983.

[Ga71] Garbrecht, D., "Pedestrian paths through a uniform environment," *Town Planning Review,* vol. 42, 1971, pp. 71-84.

[GB78] Getis, A. and Boots, B., *Models of Spatial Processes: An Approach to the Study of Point, Line and Area Patterns*, Cambridge University Press, Cambridge, 1978.

[Gr67] Grunbaum, B., *Convex Polytopes*, Wiley, London, 1967.

[Ho65] Horowitz, M., "Probability of random paths across elementary geometrical shapes," *Journal of Applied Probability,* vol. 2, 1965, pp. 169-177.

[Ke91] Keil, M., "A simple algorithm for determining the envelope of a set of lines," Tech. Rept. 91-1, University of Saskatchewan, 1991.

[KLP75] Kung, H. T., Luccio, F. and Preparata, F. P., "On finding the maxima of a set of vectors," *Journal of the ACM, vol. 22, October 1975, pp. 469-476.*

[KS86] Kirkpatrick, D. G. and Seidel, R., "The ultimate planar convex hull algorithm?" *SIAM Journal on Computing*, vol. 15, No. 1, February 1986, pp. 287-299.

[KS85] Kirkpatrick, D. G. and Seidel, R., "Output-size sensitive algorithms for finding maximal vectors," *Proc. Symposium on Computational Geometry*, Baltimore, June 5-7 1985, pp. 89-96.

[Mi64a] Miles, R. E., "Random polygons determined by random lines in a plane - I," *Proceedings of the National Academy of Sciences,* vol. 52, 1964, pp. 901-907.

[Mi64b] Miles, R. E., "Random polygons determined by random lines in a plane - I I," *Proceedings of the National Academy of Sciences,* vol. 52, 1964, pp. 1157-1160.

[PS85] Preparata, F. P. and Shamos, M. I., *Computational Geometry*, Springer-Verlag, 1985.

[PSR89] Pollack, R., Sharir, M., & Rote, G., "Computing the geodesic center of a simple polygon," *Journal of Discrete & Computational Geometry,* 1989.

[Su85] Suri, S., "Computing the envelope of a set of lines," manuscript, 1985.

[Su87] Suri, S., "The all-geodesic-furthest neighbors problem for simple polygons," *Proc. Third Annual ACM Symposium on Computational Geometry,* University of Waterloo, June 1987, pp. 64-75.

[SW72] Solomon, H. and Wang, P. C. C., "Non-homogeneous Poisson fields of random lines with application to traffic flow," *Proc. 6th Berkeley Symposium on Mathematical Statistics and Probability,* vol. III, 19672, pp. 383-400.

[SY72] Santalo, L. A. and Yanez, I., "Averages for polygons formed by random lines in Euclidean and hyperbolic planes," *Journal of Applied Probability,* vol. 9, 1972, pp. 140-157.

[TM82] Toussaint, G. T. and McCalear, J. A., "A simple O(n log n) algorithm for finding the maximum distance between two finite planar sets," *Pattern Recognition Letters,* vol. 1, October 1982, pp. 21-24.

[To83] Toussaint, G. T., "Solving geometric problems with the rotating calipers," *Proc. MELECON'83*, Athens, Greece, 1983.

[To85] Toussaint, G. T., "A historical note on convex hull finding algorithms," *Pattern Recognition Letters*, vol. 3, January 1985, pp. 21-28.

[Ve87] Vegter, G., "Computing the bounded region determined by finitely many lines in the plane," Tech. Rept. CS 8703, University of Groningen, 1987.