Capturing crossings: Convex hulls of segment and plane intersections

Esther M. Arkin
AMS SUNY Stony Brook

Joseph S. B. Mitchell
AMS SUNY Stony Brook

Jack Snoeyink
CS UNC Chapel Hill

Abstract

We give a simple $O(n \log n)$ algorithm to compute the convex hull of the (possibly $\Theta(n^2)$) intersection points in an arrangement of n line segments in the plane. We also show an arrangement of dn planes in d-dimensions whose arrangement has $\Theta(n^{d-1})$ intersection points on the convex hull.

1 Introduction

Over 20 years ago, Ching and Lee [4] showed that the diameter of the finite vertices of an arrangement of n lines could be computed in $\Theta(n \log n)$ time, thanks to their observation that the extreme vertices are intersections of lines that are adjacent in slope order. (This is despite the fact that there are potentially $\Theta(n^2)$ vertices in an arrangement.) Atallah [2] separately used it to compute the convex hull in the same time. This observation does not apply to arrangements of line segments, however; we know of no equivalent algorithm published for the convex hull of the crossings in an arrangement of n line segments.

It is easy to observe that only crossings that are extreme on all defining segments can be hull vertices, and that there are at most n candidate crossings. The challenge is to efficiently find the extreme crossings, or a small superset. We employ a "clipping sweep" in two ways: first to reduce to intersecting two families of disjoint segments, then to solve this special case. Thus, we establish:

Theorem 1 *Given n line segments in the plane, one can find the convex hull of their intersection points in $\Theta(n \log n)$ time.*

The lower bounds for the problem, and for the subsequent subproblems, follow easily from element uniqueness, in the same manner as the lower bound of Ching and Lee [4].

One way to find the hull vertices in the plane using existing machinery is to build the outer face in an arrangement of segments, which has $\Theta(n\alpha(n))$ complexity and can be constructed in $O(n\alpha(n) \log n)$ expected time [7], or in worst-case $O(n\alpha^2(n) \log n)$ time [1]. Both algorithms are of substantially greater programming complexity than ours, which is a simple modification of the classic Bentley-Ottmann sweep algorithm for intersecting segments [3].

Although curiosity was our primary motivation, researchers from pattern recognition looking for concise descriptors of lines have studied hulls of random line arrangements [6, 9], and "envelopes of lines," which are the union of all finite cells in line arrangements [8, 10, 12].

We also consider the problem of planes in higher dimensions, and show a lower bound:

Theorem 2 *The convex hull of the finite vertices in an arrangement of n planes in d dimensions, where d is a fixed constant, can have $\Theta(n^{d-1})$ vertices.*

2 Preliminaries

Our input is a finite set of segments, whose endpoints can be specified to be included or not. (Segments may be relatively closed, open, or half-open.) We will not assume distinct endpoints or distinct slopes, but will define an *intersection point* as the intersection of segments, at least two of which have different slopes. Thus, the set of all intersection points is a finite set of distinct points, which may include segment endpoints, but need not. In particular, segments on the same line do not contribute intersection points, except as segments of other slopes intersect them.

We may assume, by a change of coordinates if necessary, that no line segment is vertical. This allows us to order intersection points on a segment from left to right, and to speak of the left and right extreme intersection points. An intersection is *doubly extreme* if it is extreme on at least two segments.

Lemma 3 *Any intersection point that is a vertex of the convex hull is extreme on all of its segments.*

The *left tail* of a segment is the portion from the left endpoint to the left extreme intersection point, or to the right endpoint if the segment has no intersection point. Note that the left tail degenerates to a single point if the left endpoint is also an intersection point. Similarly, the *right tail* is defined from the right endpoint to right extreme (or left endpoint). The only intersection between tails on different lines can be at their extreme ends, which are then doubly extreme. It is enough to find a small superset of the intersections of tails, since we can compute the convex hull of n points in $\Theta(n \log n)$ time.

3 Clipping sweep to find and intersect tails

Our basic tool is a clipping version of Bentley-Ottmann's plane sweep [3] for finding segment intersections. Whenever we encounter an intersection as we sweep from left to right, we clip all, or all but one, segments at that intersection and discard the unswept portion(s) to the right. (If we encounter overlapping collinear segments during the sweep, we keep only the segment extending furthest to the right, and clip the rest.)

Lemma 4 *A left-to-right clipping sweep on $O(n)$ segments runs in $O(n \log n)$ time. It returns $O(n)$ intersection points, and a set of non-crossing segments that has all left tails as a subset.*

Proof. We assume familiarity with the analysis of the Bentley-Ottmann sweep, which can be found in several textbooks [5, 11].

Since each intersection ends a segment, there are at most $O(n)$ intersection points. Moreover, since at most one segment continues beyond the intersection point, there are no crossings in the output. The sweep will thus process $O(n)$ intersection point and end point events. Each event involves $O(1)$ operations in a priority queue that holds $O(n)$ elements, establishing the running time bound. Finally, since each segment enters the sweep at its left endpoint and continues until some intersection (possibly after the leftmost) or a containing collinear segment or the right endpoint is found, a segment is returned that includes the left tail. □

Let us first use this tool to solve a simpler, red/blue subproblem:

Corollary 5 *Given n red and n blue line segments, with no red/red or blue/blue crossings, we can find the at most n intersection points that are left extreme on red segments in $\Theta(n \log n)$ time.*

Proof. Perform a clipping sweep from left to right, clipping red segments at each intersection detected, and leaving blue segments unclipped. □

Finding the intersections of all tails can be reduced to an instance of the red/blue subproblem.

Lemma 6 *In $\Theta(n \log n)$ time we can find a set of $O(n)$ intersection points that includes the intersection points of all tails.*

Proof. Perform two clipping sweeps, one from the left and one from the right, that clip all segments at an intersection. By Lemma 4, the left-to-right sweep returns a set of non-crossing segments that we'll color red – these segments include the left tails. The intersection points found, therefore, include all intersections of two left tails – all left/left extreme points.

Similarly, the right-to-left sweep returns a blue set of non-crossing segments that contains the right tails, and reports intersections that include all right/right extreme points.

Applying Corollary 5 to these red and blue segments produces a set containing all left/right extreme points. Since each sweep produces a linear number of points, the lemma is established. □

This also completes the proof of Theorem 1.

4 Hulls of arrangements in d dimensions

For planes in d dimensions, we define an intersection point to be the intersection of d planes that do not contain a common line. Because any $d-1$ planes in general position intersect in a common line, any line contributes at most the two extreme intersections to the convex hull, and any hull vertex is extreme on at least d lines, if we have n planes, then we have at most $\frac{2}{d}\binom{n}{d-1}$ vertices on the convex hull.

Since we usually think of d as a constant and $n \geq d$ growing asymptotically, we will assume that n is even and define a family of dn planes with $\Omega(n^{d-1})$ intersections on the convex hull. First, let us sketch the idea in three dimensions, where it is easier to visualize. Imagine on the $z = n$ plane an integer grid $n/2 \leq x, y < 3n/2$, which can be formed as the intersection of two families of n planes, one containing the x axis and one containing the y axis. A third family goes from (0,0,1/6) to cut the middle n diagonals of this grid; this places a quadratic number of intersections on the $z = n$ plane, and the slopes of intersection lines are such that all other intersections are below the $z = n/2$ plane. Finally, to make the grid lie on a convex surface instead of a plane, the slopes of planes containing the axes are perturbed slightly.

The figure on the next page illustrates the construction for 3 families of $n = 30$ planes each, marking intersections with faint "+" signs and convex hull vertices with darker circles. The full paper gives coordinates and details of the proof; we include only the coordinates here, for the benefit of those who would like to construct this example themselves.

In dimension d, we can represent planes as $(d+1)$-tuples in homogeneous coordinates so that that plane $\pi = (\pi_0, \pi_1, \ldots, \pi_d)$ contains the Cartesian points p that satisfy $\pi_0 + (\pi_1, \ldots, \pi_d) \cdot p = 0$. Scalar multiples of a tuple denote the same plane, so although we use rational numbers, we could clear fractions and use integers of $O(\log n)$ bits instead.

For $i = 1 \ldots d-1$, we define family π^i with planes indexed by $j = n/2 \ldots 3n/2$ having nonzero coordinates $\pi_i^{i,j} = dn$ and $\pi_d^{i,j} = -j + 1/(2dj)$. For example, the last plane of the first family is $\pi^{1,3n/2} = \big(0, n, 0, \ldots, 0, -3n/2 + 1/(3dn)\big)$. We define the final family for $k = (d-3/2)n \ldots (d-1/2)n$ as $\rho^k = \big(-k, 1-$

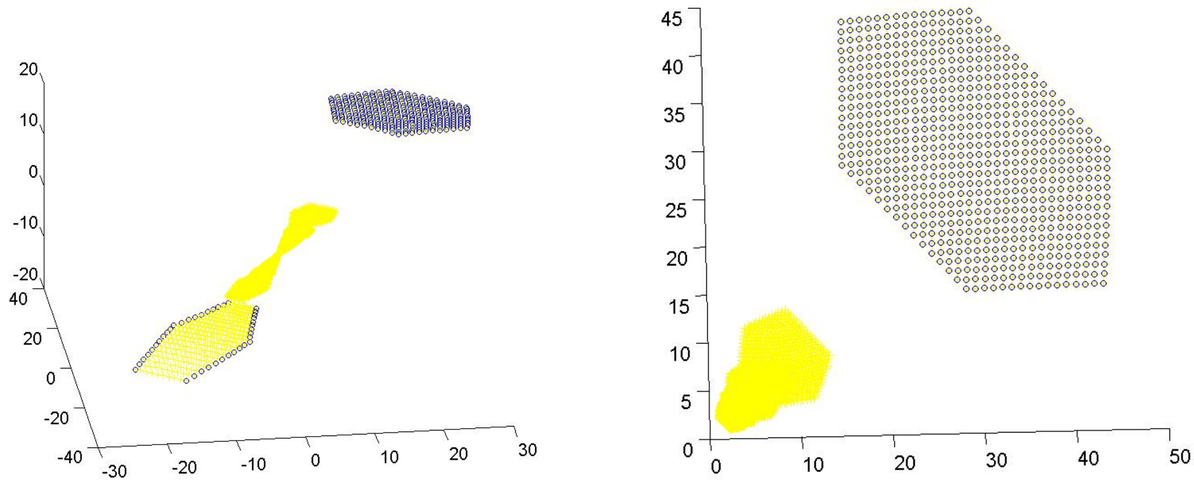

Figure 1: Two views of all intersections between 3 families of 30 planes each; circles indicate convex hull vertices.

$2dn, 1-2dn, \ldots, 1-2dn, 2dk\big)$. As mentioned above, the full paper includes details that establish Theorem 2.

Acknowledgments

This research has been partially supported by grants from the National Science Foundation (CCF-0431030, CCF-0528209, CCA-0429901), Metron Aviation (NASA Ames subcontract), and NGA/Darpa (HM1582-05-2-0003).

References

[1] N. M. Amato, M. T. Goodrich, and E. A. Ramos. Computing faces in segment and simplex arrangements. In *Proc. 27th Annu. ACM Sympos. Theory Comput.*, pages 672–682, 1995.

[2] M. J. Atallah. Computing the convex hull of line intersections. *J. Algorithms*, 7:285–288, 1986.

[3] J. L. Bentley and T. A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, C-28(9):643–647, September 1979.

[4] Y. T. Ching and D. T. Lee. Finding the diameter of a set of lines. *Pattern Recogn. Lett.*, 18(3–4):249–255, 1985.

[5] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, Germany, 2nd edition, 2000.

[6] Luc Devroye and Godfried Toussaint. Convex hulls for random lines. *J. Algorithms*, 14(3):381–394, 1993.

[7] H. Edelsbrunner, Leonidas J. Guibas, and Micha Sharir. The complexity and construction of many faces in arrangements of lines and of segments. *Discrete Comput. Geom.*, 5:161–196, 1990.

[8] D. Eu, E. Guévremont, and G. T. Toussaint. On envelopes of arrangements of lines. *J. Algorithms*, 21:111–148, 1996.

[9] Mordecai Golin, Stefan Langerman, and William Steiger. The convex hull for random lines in the plane. In *Discrete and Computational Geometry (JCDCG'02)*, volume 2866 of *Lecture Notes in Computer Science*, pages 172–175. Springer-Verlag, 2002.

[10] M. Keil. A simple algorithm for determining the envelope of a set of lines. *Inform. Process. Lett.*, 39:121–124, 1991.

[11] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 3rd edition, October 1990.

[12] Subhash Suri. Computing the envelope of a set of lines. Unpublished manuscript. Elect. Engrg. Comput. Sci., Johns Hopkins Univ., Baltimore, MD, 1985.