COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING **49**, 152–170 (1990)

# Stabbing Parallel Segments with a Convex Polygon

MICHAEL T. GOODRICH

*Department of Computer Science, The Johns Hopkins University, Baltimore, Maryland 21218*

AND

JACK SCOTT SNOEYINK*

*Department of Computer Science, Stanford University, Stanford, California 94305*

Received June 22, 1988; accepted April 14, 1989

We present an algorithm that, given a set of $n$ parallel line segments in the plane, finds a convex polygon whose boundary intersects each segment at least once or that determines that none exists. Our algorithm runs in $O(n \log n)$ steps and linear space, which is optimal. Our solution involves a reduction to a bipartite stabbing problem, using a "point-sweeping" or "chain-unwrapping" technique. We use geometric duality to solve bipartite stabbing. We also indicate how to extend our algorithm to find the convex polygon with minimum area or perimeter that intersects each segment. 

## 1. INTRODUCTION

Collections of parallel line segments appear in many facets of image processing. Objects displayed on a CRT are composed of segments of parallel scan lines. Bilevel images, such as characters in Pavlidis [1], can be stored and processed using run length encoding, which represents a line segment by storing its length and one endpoint. A robot forming an image of a part moving on a conveyor belt, such as the CONSIGHT system [2], can use structured lighting and a linear camera array to take pictures of parallel strips of the part as it passes by. In this paper we investigate the geometry of collections of parallel line segments. In particular, we look at when a straight line or convex polygon can be fitted to such a collection. Let us make this more precise before we motivate the problem.

Let $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ be a collection of parallel line segments in $\Re^2$. A straight line is said to *stab* $\mathcal{S}$ if it intersects every line segment in $\mathcal{S}$ [3]. We generalize this to convex polygons, saying that a convex polygon *stabs* $\mathcal{S}$ if its boundary intersects each segment in $\mathcal{S}$. This gives rise to the following problem: given a set $\mathcal{S}$ of segments in the plane, find a convex polygon $P$ that stabs $\mathcal{S}$, if such a polygon exists, and report failure otherwise. We call this the *convex stabbing* problem.

Stabbing segments with a straight line is an important subproblem in vectorizing scanned images, computing visibility for graphical display, and finding shortest paths for motion planning. It has been considered by mathematicians such as Grünbaum [4] and Katchalski, Lewis, and Liu [5], who were studying the existence of transversals. In 1982, Edelsbrunner and others [3] developed an algorithm to compute the line stabbing a set of line segments. Since then, algorithms have been developed to stab other geometric objects as well [6, 7].

The convex stabbing problem is a natural variation on stabbing geometric objects with a line. It was originally posed for arbitrary segments by Tamir at the Fourth

*Research supported by a National Foundation Graduate Fellowship.



0734-189X/90 $3.00

NYU Computational Geometry Day [8]. For the important special case of stabbing $n$ parallel segments, O'Rourke reports that Suri and Ke have developed an algorithm that runs in $O(n^2 \log^2 n)$ time [9].

Our algorithm solves the convex stabbing problem for $n$ parallel line segments in $O(n \log n)$ time and linear space. By reduction from sorting, any algorithm that outputs the stabbing polygon in clockwise order must take $\Omega(n \log n)$ steps to find a stabber of $n$ points on a circle. Thus, our algorithm is optimal.

The investigation of the geometry of parallel line segments also leads to an algorithm to find minimal perimeter or minimal area stabbing polygons in $O(n^2)$ time and linear space. Thus we can find the smallest convex object that could be manufactured to fit given parallel tolerances, or find the euclidean shortest convex vectorization of an image.

We will solve the convex stabbing problem for parallel line segments as follows. Based on a notion of *wrappers*, which we define in the next section, we distinguish two non-trivial cases for stabbing polygons. One case occurs when either the upper or lower chain of a canonical stabbing polygon is a single line segment, and the other occurs when the upper and lower chains each contain at least two segments. In both cases, we solve the convex stabbing problem by transforming it to an instance of a problem we call *bipartite stabbing*. Our reductions are based on what can be called a "point-sweeping" or a "chain-unwrapping" paradigm.

In the next section we present some geometric preliminaries that will be needed throughout this paper, and discuss stabbing with a straight line, which is the trivial case of the convex stabbing problem. In Section 3 we give a method for reducing the two non-trivial cases to bipartite stabbing. We show how to use duality to solve the bipartite stabbing problem in Section 4, and discuss the minimum area and perimeter problems in Section 5.

## 2. GEOMETRIC PRELIMINARIES

Given a point $p$ in $\Re^2$, let $x(p)$ and $y(p)$ denote the $x$ and $y$ coordinates of $p$, respectively. Thus, $p = (x(p), y(p))$. Denote the line segment from a point $p$ to a point $q$ by the pair $\langle p, q \rangle$. Sets of segments will be denoted by script letters.

Suppose we are given a set $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ of vertical line segments in $\Re^2$—that is, $s_i = \langle a_i, b_i \rangle$ with $x(a_i) = x(b_i)$ and $y(a_i) \geq y(b_i)$. We define $A(\mathcal{S}) = \{a_1, a_2, \ldots, a_n\}$ and $B(\mathcal{S}) = \{b_1, b_2, \ldots, b_n\}$ to be the set of upper segment endpoints and lower segment endpoints, respectively. If the set $\mathcal{S}$ of segments is clear from the context, then we simply use $A$ and $B$ to denote the sets of endpoints. We also distinguish the leftmost segment, $s_l$, and the rightmost, $s_r$. For purposes of explanation, assume that these segments are uniquely determined.

The boundary of any convex polygon can be decomposed into two vertically monotone chains, an *upper hull* and a *lower hull*. Algorithms to find the convex hull of a set of points frequently exploit this fact to find the upper and lower hulls independently [10]. Given a set of points $Q$ in $\Re^2$ we denote the upper hull of $Q$ by $\mathrm{UH}(Q)$ and the lower hull of $Q$ by $\mathrm{LH}(Q)$. We say that a polygonal chain $C$ is *upper convex* if $C = \mathrm{UH}(C)$ and *lower convex* if $C = \mathrm{LH}(C)$.

Given a set $\mathcal{S}$ of vertical line segments, we define the *upper wrapper* to be the upper hull of the lower endpoints, $\mathrm{UH}(B)$, and the *lower wrapper* to be the lower hull of the upper endpoints, $\mathrm{LH}(A)$. Any optimal convex hull algorithm can be used to construct the upper and lower wrappers of $\mathscr{S}$ in $O(n \log n)$ time [10]. The following lemma establishes an important fact about wrappers.

LEMMA 2.1. *Suppose $P$ is a convex polygon that stabs the set of segments $\mathscr{S}$. No point on the upper wrapper* UH($B$) *lies directly above a point on the upper chain of $P$.*

*Proof.* Since $P$ stabs $\mathscr{S}$, it contains points on $s_r$ and $s_l$. Thus any vertical line through UH($B$) intersects the upper chain of $P$. Suppose the lemma is false. Then there is a vertical line $l$ intersecting UH($B$) at a point $q$ and the upper chain of $P$ at a point $p$ such that $y(p) < y(q)$. Let $e$ be a non-vertical edge of UH($B$) containing $q$, and let $b_i$ and $b_j$ be the two endpoints of $e$, with $x(b_i) < x(b_j)$. Since $P$ stabs $\mathscr{S}$, it must stab $s_i$ and $s_j$. This in turn implies that $b_i$ and $b_j$ are on or below the upper chain of $P$. But if the endpoints of edge $e$ are below the upper chain of $P$, then, by upper convexity, $e$ lies below the chain; contradicting the fact that $y(p) < y(q)$. Thus UH($B$) is never above the upper chain of $P$. □

Similarly, each point on the lower wrapper LH($A$) must lie on or above the lower wrapper of a convex stabbing polygon. We define $W$, the *double wrapper* of $\mathscr{S}$, to be the union of the upper and lower wrappers of $\mathscr{S}$: $W = \mathrm{UH}(B) \cup \mathrm{LH}(A)$. If the upper wrapper and lower wrapper do not intersect in exactly two points the double wrapper is said to be *degenerate*. We study the degenerate cases in the following lemma.

LEMMA 2.2. *Suppose the double wrapper of the set of $n$ segments $\mathscr{S}$ is degenerate. Then the $\mathscr{S}$ can be stabbed by a degenerate convex polygon consisting of a single line segment. In addition, given the wrappers, the segment can be found in $O(n)$ time.*

*Proof.* There are a number of cases. We consider each one in turn.

*Case* 1. $s_l$ and $s_r$ have the same $x$ coordinate. Then all the segments in $\mathscr{S}$ have this same $x$ coordinate, and the line segment from UH($B$) (a single point) to LH($A$) (also a single point) stabs all the segments in $\mathscr{S}$.

*Case* 2. The upper wrapper and lower wrapper do not intersect. Since both endpoints of $s_l$ and $s_r$ appear in the double wrapper, if the upper and lower wrappers do not intersect, then LH($A$) is entirely above UH($B$). Thus, the wrappers are separable by a line $l$ that, using a method by Edelsbrunner [11], can be found in $O(\log n)$ time given UH($A$) and LH($B$). Since $l$ is below the points of $A$ and above the points of $B$, the segment of $l$ between $s_l$ and $s_r$ stabs each segment in $\mathscr{S}$.

*Case* 3. UH($B$) and LH($A$) intersect at only one vertex or along an edge. Then some line through the intersection will intersect every segment just as in Case 2. Since the intersection can be found in linear time (see [10 or 12], for example), this case can also be dealt with quickly. □

From now on we assume the double wrapper $W$ is non-degenerate; the wrappers intersect in two points as shown in Fig. 1 These intersection points define three polygons—two "tails," bounded by the segments $s_r$ and $s_l$, and a convex "body." If any segment in $\mathscr{S}$ lies strictly inside the body of $W$, then Lemma 2.1 proves no stabbing polygon exists. We will mention in Section 3 how our algorithm tests this condition implicitly; until that point we assume that there are no segments in $\mathscr{S}$ that are in the body of $W$.

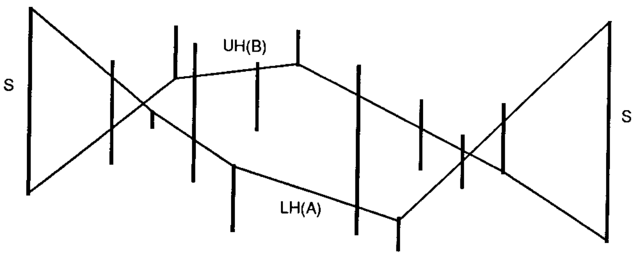


FIG. 1. A non-degenerate double wrapper.

Note that if $s_r$ and $s_l$ both degenerate into points, then $W$ is a convex polygon. By the construction of the wrappers, no segments of $\mathscr{S}$ lie strictly outside of $W$. If we assume there are no segments strictly inside $W$, then $W$ is a stabbing polygon. This suggests that if we can make the endpoints of the two wrappers coincide at well-chosen points on the left and right, then we will be done. We will prove this as Theorem 2.3. Intuitively, we can think of shrinking $s_l$ and $s_r$ in an attempt to eliminate the tails.

Given points $p \in s_l$ and $q \in s_r$, we define the double wrapper from $p$ to $q$, denoted $W(p, q)$, to be the double wrapper defined by shrinking $s_l$ to $p$ and $s_r$ to $q$. In other words, $W(p, q)$ is the double wrapper of the set $(S - \{s_l, s_r\}) \cup \{\langle p, p\rangle, \langle q, q\rangle\}$.

THEOREM 2.3. *A set of segments $\mathscr{S}$ can be stabbed by a convex polygon if and only if there exist points p on the leftmost segment, $s_l$, and p on the rightmost, $s_r$, such that $W(p, q)$ stabs $\mathscr{S}$.*

*Proof.* The *if* direction is immediate—$W(p, q)$ is a convex stabbing polygon. For the *only if* direction, suppose a convex polygon $P$ stabs $\mathscr{S}$. We need to show that $W(p, q)$ is a convex stabber for some $p \in s_l$ and $q \in s_r$. $P$ stabs $s_l$ and $s_r$, so $P \cap s_l$ and $P \cap s_r$ each contain at least one point. Choose $p \in P \cap s_l$ and $q \in P \cap s_r$ and let $\mathscr{S}' = (\mathscr{S} - \{s_l, s_r\}) \cup \{\langle p, p\rangle, \langle q, q\rangle\}$.

Since $P$ stabs $p$, $q$, and $\mathscr{S}$, $P$ stabs $\mathscr{S}'$. By Lemma 2.1, $W(p, q)$ does not intersect the exterior of $P$. Suppose there is a segment $s_i$ in $\mathscr{S}$ not stabbed by $W(p,q)$. $s_i$ is neither $s_l$ nor $s_r$, since $W(p, q)$ stabs the former at $p$ and the latter at $q$. Thus $s_i \in S'$ and is not stabbed by $W(p, q)$. Since $P$ stabs $s_i$, $s_i$ must be in the exterior of $W(p, q)$. But that means that either $a_i$ lies below $\mathrm{LH}(A(\mathscr{S}'))$ or $b_i$ lies above $\mathrm{UH}(B(\mathscr{S}'))$, both of which are contradictions. Therefore $\mathscr{S}$ can be stabbed by a convex polygon if and only if $W(p, q)$ stabs $\mathscr{S}$. □

We will use this theorem in the following section, in which we give a method to find a convex stabber for the non-degenerate case. But first, let us define the *bipartite stabbing problem* (BSP): We are given two sets $\mathbf{U} = \{\mathscr{U}_1, \mathscr{U}_2, \ldots, \mathscr{U}_l\}$ and $\mathbf{V} = \{\mathscr{V}_1, \mathscr{V}_2, \ldots, \mathscr{V}_m\}$. Each element $\mathscr{U}_i$ (resp. $\mathscr{V}_i$) is a collection of line segments in $\Re^2$ (not necessarily parallel). The $\mathscr{U}$'s and $\mathscr{V}$'s are pairwise independent in the following sense: for any $i$ and $j$, $i \neq j$, either no line stabs $\mathscr{U}_i \cup \mathscr{U}_j$ (resp. $\mathscr{V}_i \cup \mathscr{V}_j$) or all the lines that stab $\mathscr{U}_i \cup \mathscr{U}_j$ (resp. $\mathscr{V}_i \cup \mathscr{V}_j$) intersect in a common point. We want to report a line $l$ such that $l$ stabs the segments $\mathscr{U}_i \cup \mathscr{V}_j$, for some $i$ and $j$, or report that no such line exists.

Intuitively, we can consider the $\mathscr{U}_i$ and $\mathscr{V}_i$ as vertices of a graph $G$ that contains an edge if and only if a line simultaneously stabs the sets of segments associated with each endpoint of the edge. The independence condition means that $G$ is bipartite. The BSP asks if there exists an edge of $G$.

When $n$ is the total number of segments represented in the two groups, we will see in Section 4 how to solve this problem in $O(n \log n)$ time and $O(n)$ space using the concept of geometric duality [3, 12]. In the next section, we reduce both non-trivial cases of the convex stabbing problem to instances of BSP in $O(n \log n)$ time and $O(n)$ space, where $n$ is the number of segments in $\mathscr{S}$, thus solving the convex stabbing problem in the same bounds.

## 3. REDUCING CONVEX STABBING TO BIPARTITE STABBING

Let us review the problem that remains after the trivial cases are eliminated. We are given a collection $\mathscr{S}$ of vertical line segments in the plane such that the double wrapper $W$ is non-degenerate and no segment of $\mathscr{S}$ lies entirely in the body of $W$. We wish to find a convex stabber of $\mathscr{S}$ if one exists, otherwise we report, "none exists."

Our method employs the following paradigm, which can be regarded as either sweeping with a point or unwrapping a chain. Consider moving the lower endpoint of $s_1$ upwards, leaving all other points unchanged, in an attempt to shrink the left tail. We begin to unwrap the upper wrapper, UH($B$). Two kinds of events occur as we sweep upward: *drop* events and *unwrap* events. At a drop event we cause the upper wrapper UH($B$) to sweep past the upper end of a segment that previously intersected the upper wrapper. At an unwrap event we lose a vertex from the upper wrapper UH($B$) by passing the lower end of a segment. Both types of events will give us information about where the points $p$ and $q$ can be placed on the segments $s_1$ and $s_r$ such that $W(p, q)$ is a convex stabbing polygon.

As we unwrap the upper wrapper from the left, we can locate the drop and unwrap events on the vertical line containing $s_1$; denote the $y$ coordinate of the drop and unwrap events for segment $s_i$ by $d_i^{\mathrm{lu}}$ and $u_i^{\mathrm{lu}}$, respectively. Analogously, we can move the upper endpoint of $s_1$ downward to find the drop and unwrap events for unwrapping the lower wrapper from the left. We denote the $y$ coordinates for these events by $d_i^{\mathrm{ll}}$ and $u_i^{\mathrm{ll}}$. Not surprisingly, we must also locate events on the line containing $s_r$. Events $d_i^{\mathrm{ru}}$ and $u_i^{\mathrm{ru}}$ are caused by unwrapping the right end of the upper wrapper and $d_i^{\mathrm{rl}}$ and $u_i^{\mathrm{rl}}$ are caused by unwrapping the right end of the lower wrapper. When the segment and wrapper are clear from the context, the superscripts will be dropped. In the lemma that follows we show how to compute the events for all segments in $\mathscr{S}$.

LEMMA 3.1. *The drop and unwrap events for all $n$ segments in $\mathscr{S}$ can be determined in $O(n \log n)$ time and $O(n)$ space.*

*Proof.* We concentrate on determining all the events for sweeping the bottom endpoint of $s_1$ upward; similar methods find the other three kinds of events. Let $\{b_{i_1}, b_{i_2}, \ldots, b_{i_m}\}$ be the vertices of the upper wrapper UH($B$) in the left to right order. It is easy to determine where the unwrap event $u_{i_k}$ occurs as we sweep upward; simply extend the upper hull edge $\langle b_{i_k}, b_{i_{k+1}} \rangle$ until it crosses the line

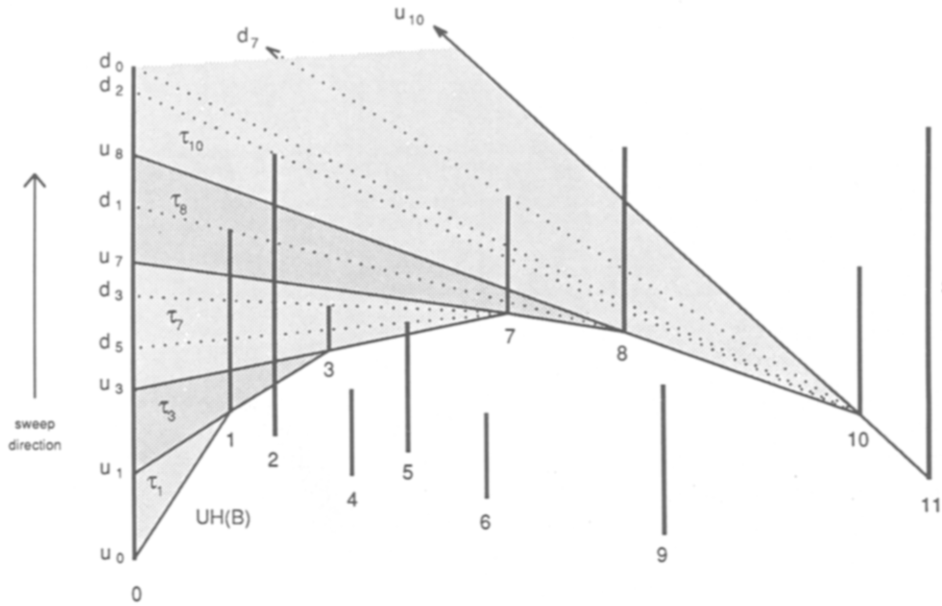


FIG. 2. Locating the events on $s_1$.

containing $s_1$. Added edges are called *extension edges* and are illustrated in Fig. 2. By simply "walking" along the upper wrapper from left to right, we obtain the ordered list of unwrap events along $x = x_1$ in $O(n)$ steps.

The extension edges partition the plane above the wrapper into triangles, where triangle $\tau_{i_k}$ has the point $b_{i_k}$ as its apex and the interval $[u_{i_{k-1}}, u_{i_k}]$ of the line containing $s_1$ as its base. As long as the sweeping point $p$ is in this interval, the upper wrapper will contain the vertices $\{p, b_{i_k}, \ldots, b_{i_m}\}$. As $p$ moves within the interval, only the edge $\langle p, b_{i_k} \rangle$ will change. We can say that the apex will be used as a pivot point for unwrapping the chain UH$(B)$. Since the location of the drop events depend on the pivot point of the wrapper, these triangles help us to locate the drop events.

If the upper endpoint $a_i$ of segment $s_i$ falls in triangle $\tau_j$, then there is a drop event $d_i$ located where the line from the apex $b_j$ through $a_i$ intersects the base of $\tau_j$. For if the lower endpoint of $s_1$ is at or below $d_i$, then the upper wrapper intersects segment $s_i$; above $d_i$ it does not. If $a_i$ is in no triangle, then it lies below the upper wrapper. In that case, the upper wrapper misses $s_i$ and $d_i$ is defined to be $-\infty$.

For each segment $s_i$ the triangle containing $a_i$ can easily be found in $O(\log n)$ steps by a binary search of the list of extension edges causing unwrap events. We then compute the $y$ coordinate for drop event $d_i$ in constant time. By repeating this procedure with each segment, we determine all the drop events in $O(n \log n)$ time. Once we have obtained them, we can sort them in the same time bound. We obtain ordered list of drop events for sweeping the left endpoint of the lower wrapper downward by a similar process—except if segment $s_i$ is entirely above the wrapper, we assign $d_i^{ll} = +\infty$. Determining the events $d_i^{ru}$ and $d_i^{rl}$ for the point sweeping on the right is analogous. □

Our method of reducing a convex stabbing problem to a bipartite stabbing problem depends on the following observations. Let $p$ be the left endpoint of both the upper and lower wrappers. By the definition of drop event $d_i^{\mathrm{lu}}$, the upper wrapper intersects the segment $s_i$ only if $y(p) \in (-\infty, d_i^{\mathrm{lu}}]$. Similarly, the lower wrapper intersects the segment $s_i$ only if $y(p) \in [d_i^{\mathrm{ll}}, +\infty)$. If we wish to form a convex stabbing polygon, any segment dropped by the upper wrapper must intersect the lower wrapper and vice versa. We can apply this insight, once we have located the events on $s_1$, to define regions in which the left endpoint $p$ of the upper and lower wrappers may lie. We shall utilize the regions from the left and right, together with some other constraints, in the reduction.

A convex polygon $P$ is *one-sided* if either the upper chain $\mathrm{UH}(P)$ or the lower chain $\mathrm{LH}(P)$ is a single segment. Otherwise $P$ is *two-sided*. We investigate one- and two-sided wrappers in Subsections 3.1 and 3.2. In case one we check if there exist points $p$ and $q$ on $s_1$ and $s_r$, respectively, such that $W(p, q)$ is a two-sided stabber. Failing that, we check in case two if there exist $p$ and $q$ such that $W(p, q)$ is a one-sided stabber. Theorem 2.3 permits us to restrict ourselves to these two cases.

As long as $W(p, q)$ is two-sided, moving point $p$ does not affect the portion of the wrapper in the vicinity of $q$. We cannot place points $p$ and $q$ independently, however; there remains a subtle interaction between the two endpoints that we will make precise below. Restricting $W(p, q)$ to be one-sided leaves a knottier problem —there are many degrees of freedom for placing the endpoints of the segment. Nevertheless, in both cases, we can either find $p$ and $q$ such that $W(p, q)$ is a stabbing polygon or determine that no solution exists by solving an instance of the bipartite stabbing problem.

### 3.1. *Case* 1—$W(p, q)$ *Is Two-Sided*

Let us restrict our search to that of finding a $p \in s_1$ and $q \in s_r$ such that $W(p, q)$ is a two-sided convex stabber. This condition implies that $W(p, q)$ intersects both the upper and lower wrappers of $\mathscr{S}$ at points other than $p$ and $q$. From the point of view of $p$ on $s_1$ it is irrelevant whether the rightmost segment $s_r$ has shrunk to $q$ or not. We can almost decouple the problems of finding $p$ and $q$.

To find $p$ and $q$, we must combine the information given by the drop and unwrap events. As before, we concentrate on the problem of combining events on $s_1$; the method for combining events on $s_r$ is similar. We summarize it here: First, utilize the events to divide the line containing $s_1$ into intervals in which $p$ may lie. Lemma 3.2 states that it is enough to consider these intervals. Second, eliminate certain infeasible intervals by combining information from drop events on the left. The eliminate all intervals not on $s_1$. The set $\mathscr{L}$ will consist of the feasible intervals of $s_1$ that remain and $\mathscr{R}$ will be the intervals of $s_r$. Finally, construct an instance of bipartite stabbing based on the events from the left and the right. The solution to this problem, if it exists, will stab $\mathscr{L}$ at $p$ and $\mathscr{R}$ at $q$ such that $W(p, q)$ is a convex stabbing polygon.

Let $I_1, I_2, \ldots, I_m$ be the closed intervals, ordered by increasing $y$ coordinate, between consecutive events on the line $x = x_1$ containing $s_1$. Since there are at most $n$ of each type of event, $m \leq 4n + 1$. This definition means that $I_i$ is a vertical segment. However, to avoid confusion with the segments of $\mathscr{S}$, we shall continue to call the $I_i$'s "intervals." Let $J_1, J_2, \ldots, J_{m'}$ be the intervals between events on

$x = x_r$. By the following lemma, it is enough to find a pair of intervals that can contain the extreme points of a convex stabber.

LEMMA 3.2. *Let $p \in I_i$ and $q \in J_j$ be points in intervals on the leftmost and rightmost segments, respectively. If $W(p, q)$ is a convex stabbing polygon then $W(p', q')$ is a convex stabber for all $p' \in I_i$ and $q' \in J_j$.*

*Proof.* Follows from the definition of drop and unwrap events. □

Some of these intervals are infeasible—they cannot contain the point $p$ because of the observation following the proof of Lemma 3.1: If $p$ is the left endpoint of both the upper and lower wrappers, then the upper wrapper intersects segment $s_i$ only if $y(p) \in (-\infty, d_i^{\mathrm{lu}}]$ and the lower only if $y(p) \in [d_i^{\mathrm{ll}}, +\infty)$. If $d_i^{\mathrm{lu}} < d_i^{\mathrm{ll}}$ for segment $s_i$ then $y(p)$ cannot lie in $(d_i^{\mathrm{lu}}, d_i^{\mathrm{ll}})$. Hence, all intervals that lie between $d_i^{\mathrm{lu}}$ and $d_i^{\mathrm{ll}}$ can be discarded. Notice that this means that if segment $s_i$ is missed by both wrappers, then $d_i^{\mathrm{lu}} = -\infty$ and $d_i^{\mathrm{ll}} = +\infty$ so $y(p) \notin (+\infty, -\infty)$. Thus, if a segment is missed by both wrappers, all of the intervals $I_1, I_2, \ldots, I_m$ on the left will be discarded and we can report that no stabber exists.

If each event knows which segment caused it and each segment knows its events, we can mark the intervals feasible or infeasible by scanning the list of intervals in order. The algorithm described in the following paragraphs does so by alternating between *feasible* and *infeasible* modes and tagging intervals with the mode. It looks at each interval once and thus runs in $O(n)$ time.

Begin in *feasible* mode and consider the intervals $I_1, I_2, \ldots, I_m$ in increasing $y$ coordinate order. Suppose first that the lower end of the interval under consideration is the event $d_i^{\mathrm{lu}}$ and $d_i^{\mathrm{lu}} < d_i^{\mathrm{ll}}$. If we are currently in *feasible* mode, switch to *infeasible* mode and remember that we switch back when we encounter $d_i^{\mathrm{ll}}$. If instead we are in *infeasible* mode and, because of an earlier encounter with $d_j^{\mathrm{lu}} < d_j^{\mathrm{ll}} < d_i^{\mathrm{l}}$, we are planning to switch to *feasible* mode at $d_j^{\mathrm{ll}}$, then we should switch back at $d_i^{\mathrm{ll}}$.

Otherwise, the lower end must be a drop event $d_i^{\mathrm{ll}}$ or an unwrap event. At a $d_i^{\mathrm{ll}}$ event we may need to switch from *infeasible* to *feasible* mode. No matter what the lower end of the interval is, we tag the interval under consideration with the current mode and proceed to the next interval.

After eliminating the intervals tagged *infeasible* by the algorithm, we also eliminate the intervals that do not intersect the segment $s_l$—since $p$ must lie on $s_l$ for $s_l$ to be stabbed by $W(p, q)$. Let $\mathscr{L} = \{L_1, L_2, \ldots, L_k\}$ (resp. $\mathscr{R} = \{R_1, R_2, \ldots, R_{k'}\}$) be the ordered set of intervals of $s_l$ (resp. $s_r$) that were tagged *feasible*.

Now we are looking for a pair of feasible intervals that can contain $p$ and $q$ such that $W(p, q)$ is a convex stabber. Unfortunately, there might be $\Omega(n^2)$ pairs of intervals to consider, so we cannot afford the luxury of testing every pair. Instead, we use them to help us construct an instance of the bipartite stabbing problem (BSP).

Recall that, in an instance of BSP, we are looking for a line $l$ that stabs the union of two collections of segments, $\mathscr{U}_i \cup \mathscr{V}_j$. We will describe how to reduce our problem to BSP in an incremental fashion. We define our first BSP instance to enforce the requirement that $p \in \mathscr{L}$ and $q \in \mathscr{R}$. We then add segments to enforce the property that $W(p, q)$ is two-sided. Finally, we add more segments to the problem instance to enforce the property that $W(p, q)$ is a stabbing polygon.

We begin by setting $\mathscr{U}_i = \{L_i\}$ and $\mathscr{V}_j = \{R_j\}$. If we ask that our solution line $l$ define the points $p$ and $q$—that is, $p = L_i \cap l$ and $q = R_j \cap l$—then we will force $p$ to lie in $\mathscr{L}$ and $q$ to lie in $\mathscr{R}$. In order for this to be a valid instance of BSP, the $\mathscr{U}$'s and $\mathscr{V}$'s must satisfy the pairwise independence property that, if $i \neq j$, all stabbers of $\mathscr{U}_i \cup \mathscr{U}_j$ intersect at a common point. However, since the intervals of $\mathscr{L}$ all lie on a common vertical line, if there is a non-vertical stabber of $\mathscr{U}_i \cup \mathscr{U}_j$ then $\mathscr{U}_i \cap \mathscr{U}_j$ is non-empty. But intervals $L_i$ and $L_j$ can only intersect at one endpoint—all stabbers of $\mathscr{U}_i \cup \mathscr{U}_j$ intersect this point and the condition is satisfied.

Next, we must ensure that $W(p, q)$ remains two-sided. Any point $p \in s_1$ defines a tangent line, also known as a supporting line, to the upper wrapper $\mathrm{UH}(B)$ and another to the lower wrapper $\mathrm{LH}(A)$. Let $t^{\mathrm{u}}(p)$ denote the point of tangency on the upper wrapper and $t^{\mathrm{l}}(p)$ denote the point of tangency on the lower wrapper. We have the following lemma:

LEMMA 3.3. *Let a collection of vertical segments $\mathscr{S}$ with a non-degenerate wrapper and points $p \in s_1$ and $q \in s_r$ be given. The double wrapper from $p$ to $q$, $W(p, q)$, is two-sided if and only if the segment $\langle p, q\rangle$ intersects the segment $\langle t^{\mathrm{u}}(p), t^{\mathrm{l}}(p)\rangle$.*

*Proof.* We shall first prove the *if* direction. Suppose $\langle p, q\rangle$ intersects $\langle t^{\mathrm{u}}(p), t^{\mathrm{l}}(p)\rangle$, then $t^{\mathrm{u}}(p)$ is above $\langle p, q\rangle$ and $t^{\mathrm{l}}(p)$ is below. This implies that neither the upper nor the lower chain of $W(p, q)$ can be the segment $\langle p, q\rangle$; both are non-trivial. Therefore $W(p, q)$ is two-sided.

For the converse, suppose $\langle p, q\rangle$ does not intersect $\langle t^{\mathrm{u}}(p), t^{\mathrm{l}}(p)\rangle$; without loss of generality, $t^{\mathrm{u}}(p)$ lies below $\langle p, q\rangle$. Then $W(p, q)$ contains segment $\langle p, q\rangle$ as an edge. Since $p$ and $q$ are the extremal points of $W(p, q)$, $W(p, q)$ is one-sided. □

Note that tangencies change only at unwrap events; since unwrap events never occur in the middle of an interval, we can uniquely associate upper and lower points of tangency with each interval. Define $t^{\mathrm{u}}(L_i) = t^{\mathrm{u}}(p)$ and $t^{\mathrm{l}}(L_i) = t^{\mathrm{l}}(p)$ for any point $p \in L_i$. We call these the *points of tangency* for $L_i$. If we include the segment $\langle t^{\mathrm{u}}(L_i), t^{\mathrm{l}}(L_i)\rangle$ with $L_i$ in our BSP instance we will ensure that $W(p, q)$ is two-sided.

The upper point of tangency $t^{\mathrm{u}}(L_i)$ is simply the apex of the triangle $\tau_j$ whose base contains the interval $L_i$. The points of tangency for all the intervals in $\mathscr{L}$ can be recovered in linear time from the triangles used to find unwrap events on $s_1$.

Thus, we have achieved the following by our construction of an instance of BSP: Restricting $p$ and $q$ to their respective feasible regions prevents us from dropping a segment from both wrappers on the left or on the right. In addition, Lemma 3.3 gives a method for enforcing the property that $W(p, q)$ is two-sided. This is not enough to solve the convex stabbing problem, however. Even if we satisfy the lemma and restrict $p$ and $q$, we still might not stab all the segments in $\mathscr{S}$. Figure 3 illustrates such a case. We need at least one more criterion to cleverly place $p$ on $s_1$ and $q$ on $s_r$.

The problem arises when a segment is dropped from the upper wrapper by the placement of the left endpoint, $p$, and from the lower wrapper by the placement of the right endpoint, $q$—or vice versa. We must use the segments dropped by the placement of $p$ to restrict the placement of $q$.

Let $\mathscr{D}(L_i)$ be the set of segments dropped by the upper wrapper when its left endpoint is in $L_i$, and the right end of the upper wrapper is left unchanged. In other

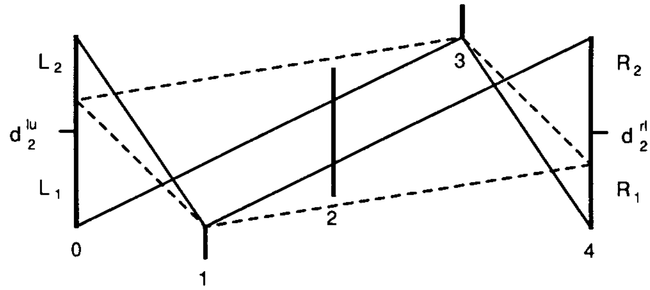


FIG. 3. Though $\mathscr{L} = s_1$ and $\mathscr{R} = s_r$, if we choose $p \in L_2$ and $q \in R_1$ then segment 2 is dropped.

words, $\mathscr{D}(L_i) = \{s_j : d_j^{\text{lu}} < y(p)\}$, where $p \in L_i$. Let $\mathscr{E}(L_i)$ be those dropped by the lower wrapper. We can prove the following technical lemma.

LEMMA 3.4. *Let points $p \in L_i$ and $q \in \mathscr{R}$ be given such that $W(p, q)$ is two-sided. Let the following segments exist: $s_d \in \mathscr{D}(L_i)$ and $s_e \in \mathscr{E}(L_i)$. $W(p, q)$ stabs segment $s_d$ if and only if $d_d^{\text{rl}} \leq y(q)$. $W(p, q)$ stabs segment $s_e$ if and only if $y(q) \leq d_e^{\text{ru}}$.*

*Proof.* Suppose $W(p, q)$ stabs $s_d$—since $s_d \in \mathscr{D}(L_i)$, it must do so with the lower wrapper. If $q$ is chosen such that $y(q)$ is less than the $y$ coordinate $d_d^{\text{rl}}$ of the drop event for losing segment $s_d$ from the right end of the lower wrapper, then segment $s_d$ will be dropped by $W(p, q)$. Therefore, $d_d^{\text{rl}} \leq y(q)$.

Conversely, suppose $W(p, q)$ does not stab $s_d$—it is dropped by both wrappers. Since $s_d \in \mathscr{D}(L_i)$, $s_d$ is dropped by the upper one by the placement of $p$. $L_i$ is feasible, however, so it is not dropped by the lower one because of the placement of $p$. Therefore, it is dropped because $y(q) < d_d^{\text{rl}}$.

The proof for segment $s_e$ is symmetric. □

Thus, we must guarantee that $q$ will be chosen with $y$ coordinate greater than or equal to the maximum $d^{\text{rl}}$ drop event of the segments in $\mathscr{D}(L_i)$ and less than the minimum $d^{\text{ru}}$ drop event of the segments in $\mathscr{E}(L_i)$. Define

$$d_{\max}^{\text{rl}}(L_i) = \max_{s_j \in \mathscr{D}(L_i)} \left\{ d_j^{\text{rl}} \right\},$$

$$d_{\min}^{\text{ru}}(L_i) = \min_{s_j \in \mathscr{E}(L_i)} \left\{ d_j^{\text{ru}} \right\}.$$

If $d_{\max}^{\text{rl}}(L_i) > d_{\min}^{\text{ru}}(L_i)$, then, by Lemma 3.4, no matter where we place $p$ in $L_i$ and $q$ in $\mathscr{R}$ there will be a segment that is not stabbed by $W(p, q)$. Thus, we remove from $\mathscr{L}$ all intervals $L_i$ such that $d_{\max}^{\text{rl}}(L_i) > d_{\min}^{\text{ru}}(L_i)$. For the intervals we do not remove, we will add the segment $\langle d_{\max}^{\text{rl}}(L_i), d_{\min}^{\text{ru}}(L_i) \rangle$ to our instance of BSP.

First, however, we will see how to calculate these quantities for all intervals. If we process the intervals in increasing $y$ coordinate order, we add segments to $\mathscr{D}(L_i)$ to form $\mathscr{D}(L_{i+1})$. Segments are never removed. Thus, we can find $d_{\max}^{\text{rl}}$ for each interval by maintaining the maximum $d^{\text{rl}}$ event as we visit the intervals in order. Total time required is $O(n)$. We find $d_{\min}^{\text{ru}}$ for each interval by processing intervals in decreasing $y$ coordinate order.

Now we can associate two segments with each segment $L_i$ that remains in $\mathscr{L}$; defining $\mathscr{U}_i$ and $\mathscr{V}_i$ as follows:

$$\mathscr{U}_i = \{L_i, \langle t^{\mathrm{u}}(L_i), t^{\mathrm{l}}(L_i)\rangle, \langle d^{\mathrm{rl}}_{\max}(L_i), d^{\mathrm{ru}}_{\min}(L_i)\rangle\},$$
$$\mathscr{V}_i = \{R_i\}.$$

Adding these two segments to each $\mathscr{U}_i$ does not cause any new lines to stab $\mathscr{U}_i$. Therefore the independence condition still holds and this is an instance of BSP. In the next theorem we show that this instance essentially solves the convex stabbing problem for the case of a two-sided convex stabber.

THEOREM 3.5. *Let $\mathscr{S}$ be a set of segments with leftmost $s_1$ and rightmost $s_{\mathrm{r}}$. Let the sets of segments $\mathscr{U}_1, \ldots, \mathscr{U}_k$ and $\mathscr{V}_1, \ldots, \mathscr{V}_{k'}$ be defined as above. There is a pair of points $(p, q)$, with $p \in s_1$ and $q \in s_{\mathrm{r}}$, such that $W(p, q)$ is a two-sided convex stabber of $\mathscr{S}$ if and only if the line through $p$ and $q$ stabs $\mathscr{U}_i \cup \mathscr{V}_j$ for some $i$ and $j$.*

*Proof.* Use Lemmas 3.3 and 3.4. Assume there is a pair $(p, q)$ such that $W(p, q)$ is a two-sided stabber. We will see that the line through $p$ and $q$ stabs $\mathscr{U}_i \cup \mathscr{V}_j$ for some $i$ and $j$.

Let the points $p \in s_1$ and $q \in s_{\mathrm{r}}$ lie in intervals $L_i$ and $R_j$, respectively. We observed that $p$ and $q$ must lie in feasible intervals for $W(p, q)$ to be a stabbing polygon, so $L_i \in \mathscr{L}$ and $R_j \in \mathscr{R}$. As a consequence of Lemma 3.4, we saw that $d^{\mathrm{rl}}_{\max}(L_i)$ must be greater than or equal to $d^{\mathrm{ru}}_{\min}(L_i)$ for $W(p, q)$ to be a stabber. Therefore there exist sets $\mathscr{U}_i$ and $\mathscr{V}_j$ containing the intervals $L_i$ and $R_j$, respectively.

By Lemma 3.3, the line through $p$ and $q$ intersects $\langle t^{\mathrm{u}}(L_i), t^{\mathrm{l}}(L_i)\rangle$. By Lemma 3.4, the point $q$ satisfies $d^{\mathrm{rl}}_{\max}(L_i) \leq y(q) \leq d^{\mathrm{ru}}_{\min}(L_i)$; meaning $q$ lies on $\langle d^{\mathrm{rl}}_{\max}(L_i), d^{\mathrm{ru}}_{\min}(L_i)\rangle$. But the union $\mathscr{U}_i \cup \mathscr{V}_j$ contains these two segments, $L_i$, $R_j$ and no other segments. Therefore the line through $p$ and $q$ stabs the union.

To prove the converse, assume the line $l$ stabs $\mathscr{U}_i \cup \mathscr{V}_j$. Let $p = l \cap s_1$ and $q = l \cap s_{\mathrm{r}}$. We will see that $W(p, q)$ is a two-sided convex stabber of $\mathscr{S}$.

Notice that $p \in L_i$ and $q \in R_j$. The segment $\langle p, q\rangle$ stabs $\langle t^{\mathrm{u}}(L_i), t^{\mathrm{l}}(L_i)\rangle$; thus Lemma 3.3 proves $W(p, q)$ is two-sided. Without loss of generality, suppose segment $s_k$ is dropped from the upper wrapper. By Lemma 3.4, $W(p, q)$ stabs $s_k$ if $d^{\mathrm{rl}}_k \leq y(q)$. But since $l$ stabs the segment $\langle d^{\mathrm{rl}}_{\max}(L_i), d^{\mathrm{ru}}_{\min}(L_i)\rangle$, we have $d^{\mathrm{rl}}_k \leq d^{\mathrm{rl}}_{\max}(L_i) \leq y(q)$. Therefore $W(p, q)$ stabs $\mathscr{S}$. □

We have created an instance of BSP in $O(n \log n)$ steps; if it has a solution, we can obtain the convex stabber $W(p, q)$ by the above theorem. Furthermore, the instance of BSP uses at most three segments for each feasible interval—using $O(n)$ segments in total. In Section 4 we show how to solve an instance of the bipartite stabbing problem in $O(n \log n)$ steps. So, in $O(n \log n)$ time and $O(n)$ space we can find a two-sided stabber $W(p, q)$ if one exists. If none is found, then we move to the second case and look for a one-sided stabbing polygon.

### 3.2. *Case 2—$W(p, q)$ Is One-Sided*

Figure 4 shows a collection of segments that admits a one-sided convex stabber but no two-sided stabber. Thus, if the method of Section 3.1 fails, we use the method of this section to see if there is a stabbing polygon that misses one of the

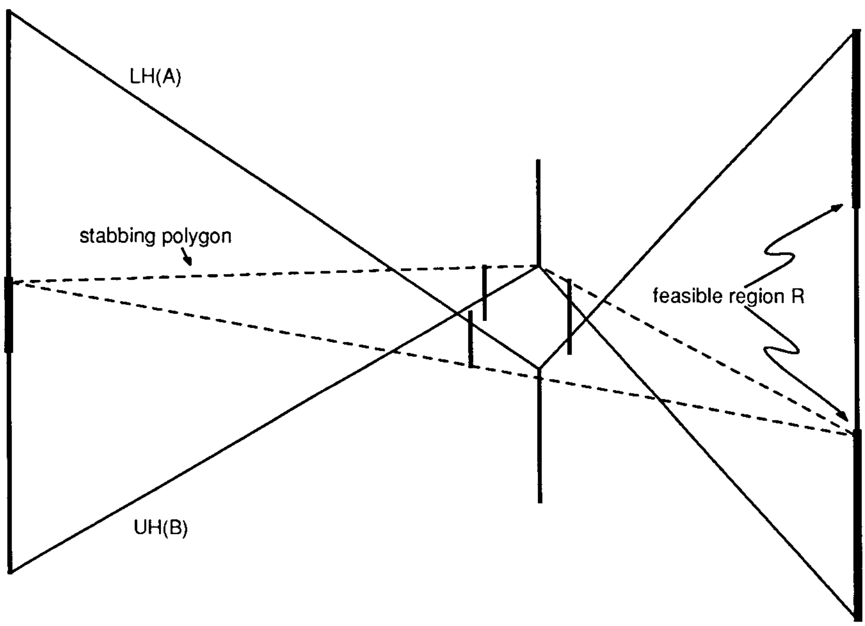


FIG. 4. An example with no two-sided stabbing polygon.

wrappers. Assume that we wish to look for a $W(p, q)$ in which $\langle p, q \rangle$ is the lower chain. The other case is symmetric. The problem is still to find points on $s_l$ and $s_r$ such that any segments dropped by one polygonal chain are picked up by the other. We will again be able to transform this case of the problem into a bipartite stabbing problem, and find a stabber in $O(n \log n)$ steps if one exists.

Since we have assumed that the lower chain is a single segment, and, in Lemma 2.2, we have eliminated the case when all the segments can be stabbed by a single line, we know that the upper chain is non-trivial. We can use the events from the upper wrapper to define intervals. Let $L_1, L_2, \ldots, L_m$ ($m = O(n)$) be closed intervals between consecutive drop events from the upper wrapper; $L_j = \langle (x_l, d_{i_{j-1}}^{\mathrm{lu}}), (x_l, d_{i_j}^{\mathrm{lu}}) \rangle$. Note that these are slightly different intervals than those of case one—they do not use unwrap events, for example.

As before, let $\mathscr{D}(L_i)$ be the set of segments dropped by the upper wrapper when its left endpoint is in $L_i$. Similarly define $R_1, R_2, \ldots, R_k$ and $\mathscr{D}(R_j)$. Let $\mathscr{D}_{i,j}$ denote the union $\mathscr{D}(L_i) \cup \mathscr{D}(R_j)$. If the points $p \in L_i$ and $q \in R_j$ are to induce a one-sided stabbing polygon $W(p, q)$ then the line through $p$ and $q$ must intersect every segment in $\mathscr{D}_{i,j}$. This allows us to make the following trivial reduction to an instance of the bipartite stabbing problem (BSP):

LEMMA 3.6. *The instance of BSP with $\mathscr{U}_i = \{L_i\} \cup \mathscr{D}(L_i)$ and $\mathscr{V}_j = \{R_j\} \cup \mathscr{D}(R_j)$ has a solution if and only if the collection $\mathscr{S}$ has a one-sided convex stabbing polygon $W(p, q)$ with points $p \in s_l$ and $q \in s_r$ such that the segment $\langle p, q \rangle$ is the lower chain of $W(p, q)$.*

*Proof.* Assume that the line $l$ is a solution; $l$ stabs $\mathscr{U}_i \cup \mathscr{V}_j$ for some $i$ and $j$. We construct a one-sided stabber. Let $p = l \cap s_1$ and $q = l \cap s_r$ and notice that $p \in L_i$ and $q \in R_j$. Suppose a segment $s_k$ is dropped by the upper wrapper of $W(p, q)$. Then $s_k$ is in $\mathscr{D}(L_i)$ or in $\mathscr{D}(R_j)$. But since $l$ stabs $\mathscr{U}_i \cup \mathscr{V}_j$ the segment $\langle p, q \rangle$ stabs the union $\mathscr{D}_{i,j}$. Therefore, the upper wrapper of $W(p, q)$ and the segment $\langle p, q \rangle$ stab the collection $\mathscr{S}$.

We must show that $W(p, q)$ is a convex stabbing polygon with $\langle p, q \rangle$ as its lower chain. Since the upper wrapper of $W(p, q)$ is upper convex from $p$ to $q$, the union of this wrapper and the segment $\langle p, q \rangle$ is a convex polygon that stabs $\mathscr{S}$. By Lemma 2.1 the lower chain of any convex polygon stabbing $\mathscr{S}$ through $p$ and $q$ must lie on or below the lower chain of $W(p, q)$. Therefore, the segment $\langle p, q \rangle$ is the lower chain of $W(p, q)$.

Conversely, assume $W(p, q)$ is the desired stabber. Let $p$ and $q$ lie in $L_i$ and $R_j$, respectively. The segment $\langle p, q \rangle$ stabs $\mathscr{D}(L_i) \cup \mathscr{D}(R_j)$, therefore the line through $p$ and $q$ stabs $\mathscr{U}_i \cup \mathscr{V}_j$ and provides a solution for the instance of BSP. □

This gives us a method of finding one-sided stabbers; unfortunately, the reduction uses $\Omega(n^2)$ segments. Solving the bipartite stabbing problem would take $O(n^2 \log n)$ time and quadratic space rather than the desired $O(n \log n)$ time and linear space. We must identify a few crucial segments of $\mathscr{D}(L_i)$ and $\mathscr{D}(R_j)$ and include only those in our BSP instance—we begin to do so by looking at the conditions that the stabbing line must satisfy.

Recall that $A(\mathscr{S}')$ and $B(\mathscr{S}')$ denote the sets of upper and lower endpoints of the collection of vertical segments $\mathscr{S}'$. We abbreviate the notation for the upper hull of the lower endpoints of the dropped segments, $\mathrm{UH}(B(\mathscr{D}(L_i)))$, by $\mathrm{UD}(L_i)$.

By Lemma 2.1, the line containing the segment $\langle p, q \rangle$ must be below or tangent to $\mathrm{LH}(A)$, the lower wrapper of $\mathscr{S}$. In addition, since it stabs every segment in $\mathscr{D}_{i,j}$ it must be above the upper hull $\mathrm{UH}(B(\mathscr{D}_{i,j}))$. But the line lies above this hull if and only if it is above both $\mathrm{UD}(L_i)$ and $\mathrm{UD}(R_j)$. We want to find two small sets of line segments, $\sigma(L_i)$ and $\rho(L_i)$, with the property that a line $l$ through $L_i$ lies between the convex chains $\mathrm{LH}(A)$ and $\mathrm{UD}(L_i)$ if and only if $l$ stabs $L_i$, $\sigma(L_i)$, and $\rho(L_i)$.

Let $\alpha$ and $\beta$ be the tangent lines from the upper and lower endpoints of the interval $L_i$ to the convex chain $\mathrm{LH}(A)$. Let $\{a_{i_1}, a_{i_2}, \ldots, a_{i_c}\}$ be the vertices on the portion of $\mathrm{LH}(A)$ between the points of tangency of $\alpha$ and $\beta$ with $\mathrm{LH}(A)$, inclusive. A line that passes through $L_i$ lies below $\mathrm{LH}(A)$ if and only if it lies below

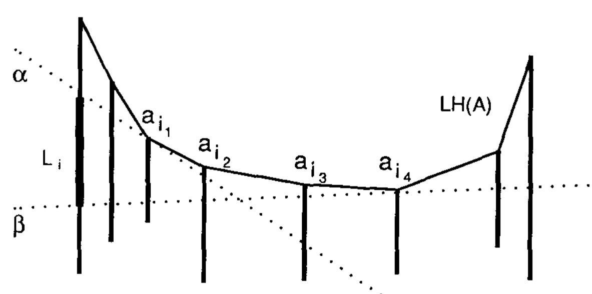


FIG. 5. A line through $L_i$ lies below $\mathrm{LH}(A)$ if it lies below $a_{i_1}, \ldots, a_{i_4}$.

the portion of LH($A$) between the two points of tangency, as illustrated in Fig. 5. Similarly, let $\alpha'$ and $\beta'$ be the tangent lines from the ends of the interval $L_i$ to UD($L_i$). Let $\{b_{j_1}, b_{j_2}, \ldots, b_{j_d}\}$ be the vertices on the portion of this hull between the points of tangency of $\alpha'$ and $\beta'$ with UD($L_i$), inclusive. A line through $L_i$ lies above UD($L_i$) if and only if it lies above the portion between the two points of tangency.

Let $\sigma(L_i)$ denote the set of segments $\{s_{i_1}, s_{i_2}, \ldots, s_{i_c}\}$ with endpoints between the points of tangency on LH($A$). Let $\rho(L_i)$ denote the segments $\{s_{j_1}, s_{j_2}, \ldots, s_{j_d}\}$ with lower endpoints between the points of tangency on UD($L_i$). Note that $\sigma(L_i) \cup \rho(L_i) \subseteq \mathscr{D}(L_i)$. Define $\sigma(R_j)$ and $\rho(R_j)$ analogously. For a new instance of BSP, take $\mathscr{U}_i = \{L_i\} \cup \sigma(L_i) \cup \rho(L_i)$ and $\mathscr{V}_i = \{R_i\} \cup \sigma(R_i) \cup \rho(R_i)$. We show later how to construct all the $\mathscr{U}_i$'s and $\mathscr{V}_j$'s in $O(n \log n)$ time. Because each $\mathscr{U}_i$ contains an $L_i$ and each $\mathscr{V}_i$ contains an $R_i$, both collections satisfy the pairwise-independence condition. We summarize the above discussion in the following theorem.

THEOREM 3.7. *Let $\mathscr{S}$ be a collection of segments with leftmost $s_1$ and rightmost $s_r$. There is a pair $(p, q)$ with $p \in s_1$ and $q \in s_r$ such that $W(p, q)$ is a one-sided convex stabber with lower chain $\langle p, q\rangle$ if and only if there is a solution to the* BSP *instance with $\mathscr{U}_i = \{L_i\} \cup \sigma(L_i) \cup \rho(L_i)$ and $\mathscr{V}_i = \{R_i\} \cup \sigma(R_i) \cup \rho(R_i)$.*

*Proof.* By Lemma 3.6 it is enough to show that the above instance of BSP is equivalent to the one with $\mathscr{U}_i' = \{L_i\} \cup \mathscr{D}(L_i)$ and $\mathscr{V}_j' = \{R_j\} \cup \mathscr{D}(R_j)$. That is, $l$ is a solution to one iff it is a solution to the other.

Assume $l$ stabs $\mathscr{U}_i \cup \mathscr{V}_j$ for some $i$ and $j$. Then $l$ passes through interval $L_i$ and lies above the portion of UD($L_i$) between the points of tangency of $\alpha'$ and $\beta'$. By the preceding discussion, $l$ lies above UD($L_i$). Also, $l$ passes through $R_j$ and lies above UD($R_j$)—thus $l$ lies above UH($B(\mathscr{D}_{i,j})$). Similarly, $l$ lies below LH($A$). As a result, $l$ intersects every segment in $\mathscr{D}_{i,j}$ and stabs $\mathscr{U}_i' \cup \mathscr{V}_j'$.

Assume $l$ stabs $\mathscr{U}_i' \cup \mathscr{V}_j'$ for some $i$ and $j$. Since $\mathscr{U}_i \subseteq \mathscr{U}_i'$ and $\mathscr{V}_j \subseteq \mathscr{V}_j'$, $l$ also stabs $\mathscr{U}_i \cup \mathscr{V}_j$. Thus we have shown the two instances are equivalent and established the theorem. □

Thus we have a smaller instance of bipartite stabbing that gives us a solution to the convex stabbing problem. To complete the algorithm we must find the segments of **U** and **V** in $O(n \log n)$ time and show that there are at most $O(n)$ segments in total.

To count or to compute the segments of $\sigma(L_i)$ for all $L_i$ is not hard. Each interval $L_i$ contributes at least the segment whose endpoint is the point of tangency of the line $\alpha$; additional segments are contributed because of, and only because of, edges of LH($A$) that intersect $L_i$ when extended. (See Fig. 5.) But these intersections correspond precisely to the unwrap events $u^{\text{ll}}$ contained in interval $L_i$. Since the intervals are disjoint, each unwrap event contributes only one segment to **U**—in total there are $O(n)$ segments. We can find $\sigma(L_i)$ for each interval $L_i$ in $O(n)$ time by "walking" along LH($A$) from left to right and processing the intervals in order of decreasing $y$ coordinate.

To compute and count the segments of $\rho(L_i)$ requires more machinery—the hulls change from interval to interval. If we process the intervals in order of increasing $y$ coordinate, however, the only difference between $\mathscr{D}(L_{i-1})$ and $\mathscr{D}(L_i)$ is that a single segment is added to the latter—no segments are ever removed. This means that we can use the on-line convex hull algorithm of Preparata [13] to compute the sequence of hulls $\mathrm{UD}(L_1), \ldots, \mathrm{UD}(L_m)$.

Preparata's hull algorithm relies on the fact that, when a new point is added to a convex hull, it causes at most two new hull edges. These edges are the two tangent lines, or supporting lines, through the new point to the hull; they can be found by binary search in $O(\log n)$ time. Thus, the total time to find all the hulls in the sequence is $O(n \log n)$. The tangents $\alpha'$ and $\beta'$ can also be found in $O(\log n)$ time per interval, and, if we show that the total number of segments in $\rho(L_i)$ for all intervals $L_i$ is $O(n)$, then they can all be computed in $O(n \log n)$ total time.

The number of vertical segments in the $\rho$'s is one for each interval, plus the number of unwrap events of $\mathrm{UD}(L_i)$ that fall in $L_i$ for each interval $L_i$. This is certainly less than $m$ plus the total number of unwrap events over all hulls $\mathrm{UD}(L_1), \ldots, \mathrm{UD}(L_m)$. But adding a vertex to a hull adds only two new hull edges, thus there are at most $2n$ unwrap events and $m + 2n = O(n)$ segments associated with the intervals.

Therefore, the instance of bipartite stabbing in Theorem 3.7 has $O(n)$ segments and can be solved in $O(n \log n)$ time. This completes case two—finding a one-sided stabbing polygon. We summarize the results of this section in the following theorem.

THEOREM 3.8. *Given a set $\mathcal{S}$ of $n$ vertical line segments in the plane, we can reduce the problem of finding a convex polygon stabbing $\mathcal{S}$ to three bipartite stabbing problems of size $O(n)$ in $O(n \log n)$ time and $O(n)$ space.*

*Proof.* There is one bipartite stabbing problem for case one, and two for case two—one each for the two kinds of one-sided polygons. □

In the next section we show how to solve the bipartite stabbing problem in $O(n \log n)$ time in $O(n)$ space.

## 4. THE BIPARTITE STABBING PROBLEM

Recall the problem we wish to solve: Suppose we are given two sets $\mathbf{U} = \{\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_l\}$ and $\mathbf{V} = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_m\}$, where each $\mathcal{U}_i$ (resp. $\mathcal{V}_i$) is a collection of line segments in $\Re^2$ that are pairwise independent in the following sense: for any $i$ and $j$ it is either the case that no line stabs $\mathcal{U}_i \cup \mathcal{U}_j$ (resp. $\mathcal{V}_i \cup \mathcal{V}_j$) or all the lines that stab $\mathcal{U}_i \cup \mathcal{U}_j$ (resp. $\mathcal{V}_i \cup \mathcal{V}_j$) intersect in a common point. We wish to report a line $l$ that stabs $\mathcal{U}_i \cup \mathcal{V}_j$, for some $i$ and $j$. If no such line exists, we report that fact. Let $n$ be the total number of segments represented in $\mathbf{U}$ and $\mathbf{V}$. In this section, we show how to use the notion of geometric duality [3, 12] to map BSP to a problem that we can solve in $O(n \log n)$ time and $O(n)$ space. This is optimal in the algebraic computation tree model [14] by a trivial reduction from the *set disjointness problem*.

We solve an instance of BSP using three steps:

1. For each set of segments $\mathcal{U}_i$, form the stabbing region $Stab(\mathcal{U}_i)$: a representation of the lines that stab $\mathcal{U}_i$. Form $Stab(\mathcal{V}_i)$ similarly.
2. Form $Stab(\mathbf{U})$ and $Stab(\mathbf{V})$ from the stabbing regions for the $\mathcal{U}_i$'s and $\mathcal{V}_i$'s, respectively.

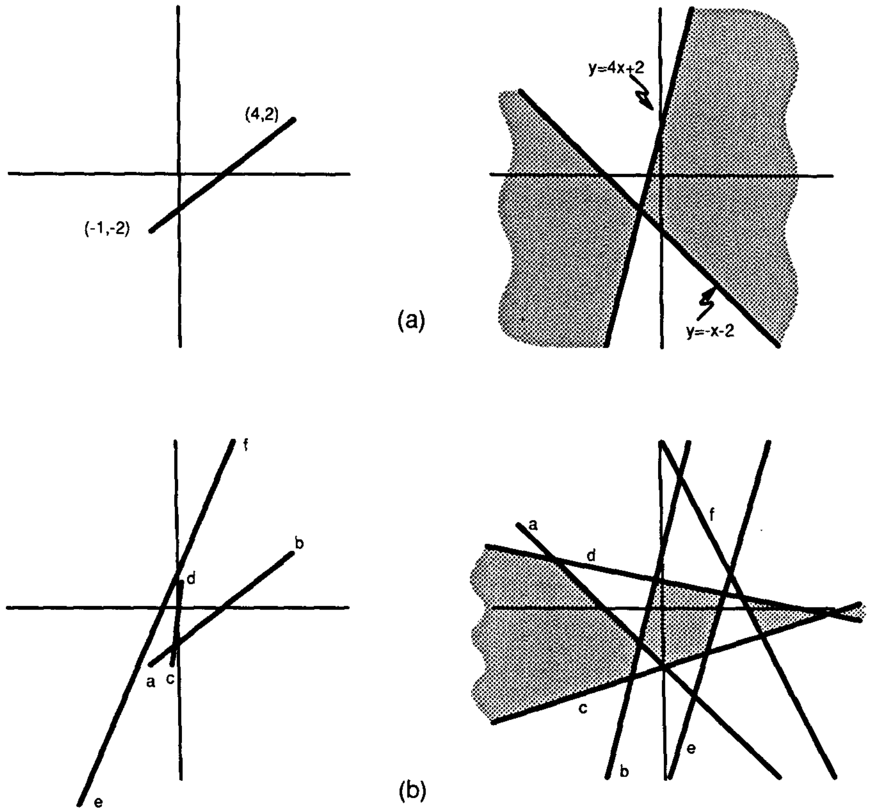


FIG. 6. A line segment maps to a double wedge, a set of segments to a stabbing region.

3. Find a point in the intersection $Stab(\mathbf{U}) \cap Stab(\mathbf{V})$— this will correspond to a line that stabs both $\mathcal{U}_i$ and $\mathcal{V}_j$ for some $i$ and $j$.

We will elaborate on these steps in the following paragraphs.

Geometric duality maps problems on one kind of geometric variety to another kind. Frequently, our intuition is better for the transformed problem [15, 3]. In our case, we use a duality transform $T$ that maps points to lines and lines to points and, in the process, maps lines stabbing a set of segments to points in a region. Specifically, $T$ maps the point $(a, b)$ to the line $y = ax + b$, and the line $y = kx + d$ to the point $(-k, d)$. Stolfi, with his work on oriented projective geometry [12], shows how vertical lines can map to points at infinity to avoid special cases.

We know that a line $l$ stabs a segment $s$ if it lies between the endpoints of $s$. In the dual plane, the endpoints of $s$ map to a pair of intersecting lines that partition the plane into four wedges. The primal line $l$ stabs $s$ if and only if the point $T_l$ lies above one dual line and below the other. Thus, $T$ maps the line segment $s$ onto a *double wedge* $T_s$; two opposite wedges that do not contain a vertical line (see Fig. 6a). The following facts are contained in [3]:

FACT 4.1. *A line $l$ intersects a line segment $s$ if and only if the point $T_l$ lies in the double wedge $T_s$.*

FACT 4.2. *The stabbing lines for a set of line segments stand in one-to-one correspondence with the set's stabbing region—the points in the intersection of their double wedges (Fig. 6b).*

FACT 4.3. *The stabbing region of m segments has no more than* $8m + 4$ *edges and can be computed in* $O(m \log m)$ *time.*

Now we can perform step 1 of the algorithm and analyze its cost. Each set $\mathcal{U}_i$ determines a stabbing region, $Stab(\mathcal{U}_i)$, which is the intersection in the dual plane of all the double wedges for the segments in $\mathcal{U}_i$. So, $Stab(\mathcal{U}_i)$ is a collection of polygons and every point in $Stab(\mathcal{U}_i)$ corresponds to a line in the original plane that stabs all the segments in $\mathcal{U}_i$. Since the total number of segments is $O(n)$, we can find the stabbing regions for all the $\mathcal{U}_i$'s in $O(n \log n)$ steps by the divide-and-conquer method used by Edelsbrunner *et al.* [3] to establish Fact 4.3. We find the stabbing regions for the $\mathcal{V}_i$'s in a similar fashion.

The pairwise independence condition makes step 2 easy; it implies that the intersection of $Stab(\mathcal{U}_i)$ and $Stab(\mathcal{U}_j)$, for any $i \neq j$ is at most a collection of co-linear line segments. Define $Stab(\mathbf{U}) = \bigcup_{i=1}^{l} Stab(\mathcal{U}_i)$. Then $Stab(\mathbf{U})$ consists of a collection of (possibly unbounded) polygons, no two of which intersect except possibly at a vertex or along an edge. Moreover, the total number of edges in $Stab(\mathbf{U})$ is $O(n)$ by Fact 4.3. Thus $Stab(\mathbf{U})$ can be constructed in linear time by simply grouping together the stabbing regions for the individual $\mathcal{U}_i$'s. Again, we form $Stab(\mathbf{V})$ similarly.

We can consider $Stab(\mathbf{U})$ and $Stab(\mathbf{V})$ as two subdivisions of the plane or as two collections of polygons. In either case, any point $p$ that lies in the intersection $Stab(\mathbf{U}) \cap Stab(\mathbf{V})$ must lie in $Stab(\mathcal{U}_i)$ and $Stab(\mathcal{V}_j)$ for some indices $i$ and $j$. This point maps to a line $T_p$ that stabs both $\mathcal{U}_i$ and $\mathcal{V}_j$ and solves the instance of BSP. If we can find such a point quickly, we can perform step 3 of the algorithm.

Shamos and Hoey [16] developed a sweep-line algorithm that finds a point common to two sets of line segments if the segments in each set are disjoint. Their algorithm places the endpoints of the segments in priority queue, ordered by increasing $x$ coordinate. Then it passes a sweeping line over the plane from left to right, maintaining the order in which the segments cross the sweep. By testing for intersections between adjacent line segments whenever a segment is hit or dropped by the sweep, they find an intersecting pair of segments or report that none exist in $O(n \log n)$ time.

It is not hard to modify this algorithm to find a point $p$ in $Stab(\mathbf{U}) \cap Stab(\mathbf{V})$ in $O(n \log n)$ time or show that no such point exists. Simply treat the regions as polygons defined by line segments, use the same priority queue, and maintain the intervals of the sweep that are contained in $Stab(\mathbf{U})$ and $Stab(\mathbf{V})$. If, during the sweep, a segment bounding $Stab(\mathbf{U})$ appears in an interval of $Stab(\mathbf{V})$ or vice versa, then the point $p$ at which it first appears may be reported as the solution. The algorithm of Shamos and Hoey must be modified slightly to handle the case when two segments in $Stab(\mathbf{U})$ (or $Stab(\mathbf{V})$) intersect at a vertex or partially overlap, but this can be done.

This completes the description of the algorithm and allows us to end this section with the following theorem.

THEOREM 4.4. *Given collections* $\mathbf{U}$ *and* $\mathbf{V}$ *of sets of line segments in the plane that satisfy the pairwise-independence condition and contain n segments in total, the* *bipartite stabbing problem for* **U** *and* **V** *can be solved in* $O(n \log n)$ *time and* $O(n)$ *space.* □

## 5. MINIMAL STABBING POLYGONS

The framework we have established allows us to find convex stabbing polygons that have minimum area or perimeter. (For the latter we give our computational model the ability to compute square roots). In this section we will show that the minimal polygons are among the wrappers $W(p, q)$. Then we will add a measure function to a bipartite stabbing problem so that the minimal polygon will be a solution with minimal measure. Finally, we show how to solve this modified BSP in $O(n^2)$ time.

The proof of theorem 2.3 shows that if a set of segments $\mathcal{S}$ has a convex stabbing polygon $P$, then $\mathcal{S}$ is stabbed by a convex polygon $W(p, q)$ that is contained inside $P$. From this we know that the polygons with minimum area or perimeter will be the double wrappers $W(p, q)$ of $\mathcal{S}$.

In an instance of BSP, define the measure $m_{ij}$ of a pair $\mathcal{U}_i$ and $\mathcal{V}_j$ to be the minimal perimeter or area of the all double wrappers $W(p, q)$ such that the line from $p$ to $q$ stabs $\mathcal{U}_i \cup \mathcal{V}_j$. The measure is defined to be infinite if no wrapper exists. With a small amount of preprocessing, we can obtain a formula for finite measures $m_{ij}$ that can be evaluated in constant time.

Consider first the instance of BSP used in Theorem 3.5 to solve the two-sided case. Since $\mathcal{U}_i$ and $\mathcal{V}_j$ include intervals of $s_l$ and $s_r$ that contain no unwrap events, all wrappers $W(p, q)$ such that $\langle p, q \rangle$ stabs $\mathcal{U}_i \cup \mathcal{V}_j$ have the same vertices—only the placement of the points $p$ and $q$ varies. The perimeter of such a wrapper is a constant plus a term that depends on the lengths of the segments adjacent to $p$ and $q$; the area depends on the area of the triangles defined by the pairs of the segments adjacent to $p$ and $q$. It is not hard to find in constant time the placement of points that minimizes the appropriate measure $m_{ij}$. Furthermore, if, at each chain vertex, we store the length of the chain to the left of the vertex, then we can compute the perimeter measure in constant time. (Here we assume that we can compute and store square roots.) Similarly, if we store the area under a chain and to the left of a vertex, then we can compute the areas under the upper and lower chains and subtract to find the area measure in constant time.

Next, consider the instance of BSP used in Theorem 3.7 to solve the one-sided case. If we use the unwrap events from the upper wrapper to define intervals for this case, as we did in the two-sided case, then we can compute finite measures $m_{ij}$ in constant time in the manner of the previous paragraph.

Since finite measures $m_{ij}$ can be computed in constant time, we can find a minimal area or perimeter polygon by enumerating all pairs $\mathcal{U}_i$ and $\mathcal{V}_j$ with finite measure and taking the smallest. Using the notation of the previous section, the pairs with finite measure are exactly those pairs satisfying $Stab(\mathcal{U}_i) \cap Stab(\mathcal{V}_j) \neq \varnothing$. Thus, we must compute the intersection $Stab(\mathbf{U}) \cap Stab(\mathbf{V})$, which is a planar subdivision with at most $O(n^2)$ polygonal regions, and find the minimal measure over all regions. This can be accomplished in $O(n^2)$ time and linear space by a sweeping algorithm [16].

## 6. CONCLUSION

In this paper we have investigated problems of stabbing parallel line segments. We gave an algorithm for solving the following problem: given a set $\mathcal{S}$ of $n$ vertical line segments in the plane, find a convex polygon whose boundary intersects each segment in $\mathscr{S}$, if such a polygon exists, and report failure otherwise. Our algorithm runs in $O(n \log n)$ time and $O(n)$ space, which is optimal. Our solution involved reducing two different cases of the convex stabbing problem to a problem we called bipartite stabbing, which is an interesting problem in its own right. We also show how to use our algorithm to stab parallel line segments with a polygon of minimum area or perimeter in $O(n^2)$ time and $O(n)$ space.

There are several directions for further work. We are most interested in stabbing arbitrary (non-parallel) segments in the plane—this is a simplification of Tamir's original problem [8] that is still open. An algorithm for stabbing the maximum number of segments would also be interesting for certain pattern matching applications.

## ACKNOWLEDGMENTS

We thank John Hershberger, Joseph O'Rourke, and Subhash Suri for helpful discussions.

## REFERENCES

1. T. Pavlidis, A vectorizer and feature extractor for document recognition, *Comput. Vision Graphics Image Process.* **35**, 1986, 111–127.
2. M. R. Ward, L. Rossol, and S. W. Holland, CONSIGHT: An adaptive robot with vision, *Rob. Today*, Summer 1979, 26–32.
3. H. Edelsbrunner, H. A. Maurer, F. P. Preparata, A. L. Rosenberg, E. Welzl, and D. Wood, Stabbing line segments, *BIT* **22**, 1982, 274–281.
4 B. Grünbaum, On common transversals, *Arch. Math.*, **9**, 1958, 465–469.
5. M. Katchalski, T. Lewis, and A. Liu, Geometric permutations and common transversals, *Discrete and Computa. Geom.* **1**, 1986, 371–377.
6. H. Edelsbrunner, Finding transversals for sets of simple geometric figures, *Theoret. Comput. Sci.* **35**, 1985, 55–69.
7. M. Atallah and C. Bajaj, Efficient algorithms for common transversals, *Inform. Process. Lett.* **25**, No. 2, 1987, 87–90.
8. A. Tamir, Problem 4-2, Dept. of Statistics and Operations Research, New York University, in *Problems Presented at the Fourth NYU Computational Geometry Day (3 / 13 / 87)*, 1987.
9. J. O'Rourke, Computational geometry column No. 3, *Comput. Graphics* **21**, No. 5, 1987, 314–315.
10. F. P. Preparata and M. I. Shamos, *Computational Geometry*, Springer-Verlag, New York, 1985.
11. H. Edesbrunner, Computing the extreme distances between two convex polygons, *J. Algorithms* **6**, 1985, 213–224.
12. J. Stolfi, Oriented projective geometry, in *Proceedings, 3rd ACM Symp. on Computational Geometry, Waterloo, 1987*, pp. 76–85.
13. F. P. Preparata, An optimal real time algorithm for planar convex hulls, *Commun. ACM* **22**, No. 7, 1979, 402–405.
14. M. Ben-Or, Lower bounds for algebraic computation trees in, *Proceedings 15th ACM Symp. on Theory Comput., Boston, 1983*, pp. 80–86.
15. K. Q. Brown, Geometric transforms for fast geometric algorithms, Ph.D. thesis, Rep. CMU-CS-80-101, Dept. Computer Science, Carnegie-Mellon University, Pittsburg, PA, 1980.
16. M. I. Shamos and D. Hoey, Geometric intersection problems, in *Proceedings, 17th IEEE Symp. on Foundations of Computer Science, Houston, 1976*, pp. 208–215.