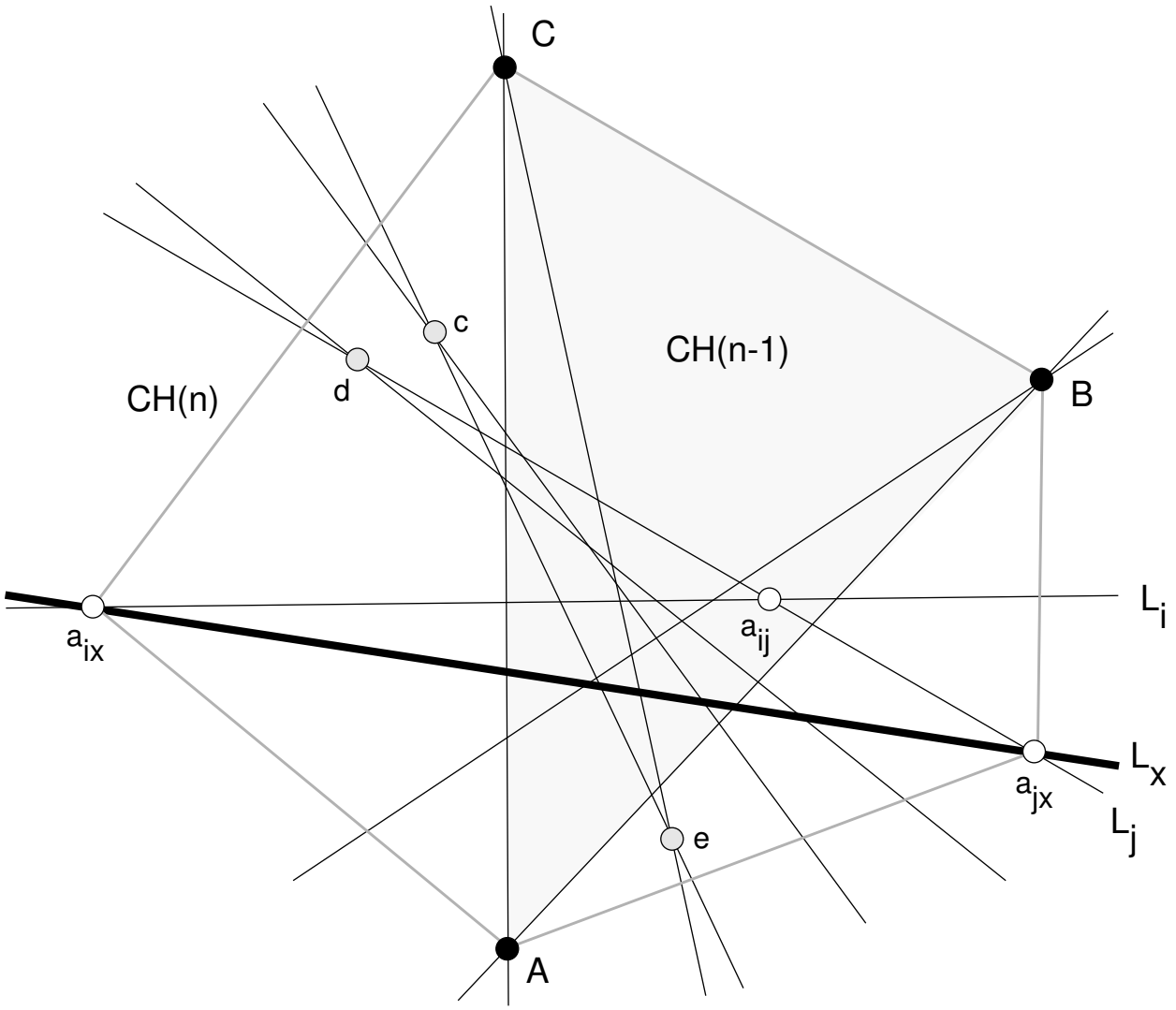

Fig. 2 Illustrating the counterexample for the line deletion algorithm.

berg, 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.

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

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

Step 3: Compute the pairwise intersection points of the input line and the enclosing lines.

Step 4: Test if the intersection points (a_{ix} and a_{jx}) lie outside the CH(n) and delete the point(s) from CH(n) that lie outside it.

Step 5: Insert the intersection point (a_{ij}) of the two enclosing lines if it lies outside the boundary of the resulting convex hull.

Step 6: Delete the input line from the list of straight lines.

3. The Counterexample

Consider the arrangement of lines in Fig. 2 and step through the above algorithm. Let L_x be the line to be deleted. We must update the convex hull of **I** accordingly. By step 2 we find the two lines enclosing L_x, that is lines L_i and L_j. In step 3 we compute the three intersection points a_{ix}, a_{jx} and a_{ij}. According to step 4 we must delete either a_{ix} or a_{jx} or both if they affect the present convex hull to obtain the new hull. In the example of Fig. 2 both of these vertices are thus deleted from CH(n) to obtain triangle ABC as the candidate for the new convex hull of all lines except L_x, CH(n-1). In step 5 we test whether a_{ij} lies outside the boundary of the candidate convex hull ABC and insert it into the convex hull if it does. In our example a_{ij} lies in the interior of triangle ABC and thus the algorithm exits with triangle ABC as CH(n-1). However, just because a_{ij} does not lie outside the candidate convex hull does not mean that all other intersection points don't either. Indeed, in our example there remain three, c, d and e. Furthermore CH(n-1) must include two of these intersection points, d and e, as extreme vertices.

4. References

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

[Bo90] Boreddy, J., "An incremental computation of 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.

[DT90] Devroye, L. and Toussaint, G. T., "Convex hulls for random lines," Tech. Rept. SOCS-90.11, School of Computer Science, McGill University, May 1990.

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

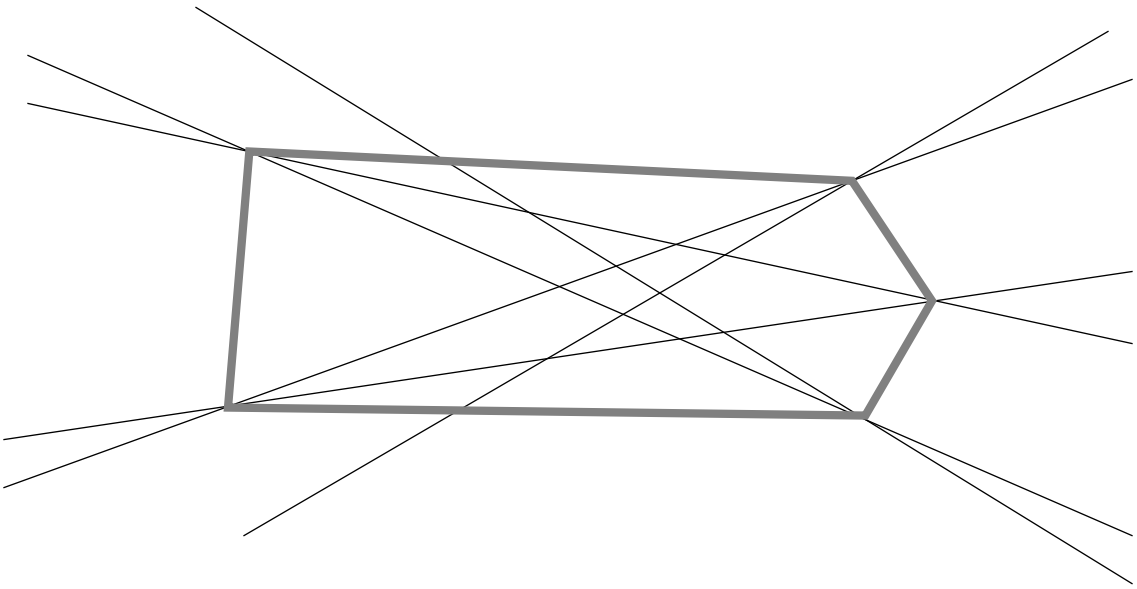

Fig. 1 An arrangement of six lines and their convex hull.

only if p_{ij} is extreme with respect to both lines L_i and L_j.

It is known that the convex hull of n points in the plane can be computed in O(n log n) time [To85]. Straightforward application of such algorithms to all the points in **I** thus leads to $O(n^2 \log n)$ time algorithms. Surprisingly, Ching & Lee [CL85] present an O(n log n) time algorithm for this problem and also establish an Ω(n log n) lower bound. Atallah [At86] independently rediscovered the same convex hull algorithm. The crucial lemma in [CL85] and [At86] that leads to the result is the fact that the vertices of the convex hull of **I** are determined by the critical intersection points of **I** only and there are O(n) of these. It is also interesting to note that for a *random* arrangement of lines, under almost any definition of random line, the expected number of vertices on the convex hull of **I** is O(1) [DT90]. An arrangement of six lines and their convex hull is illustrated in Fig. 1. The above lemma suggests sorting the lines by angle in O(n log n) time and then computing the critical intersection points by intersecting adjacent lines on the sorted list.

2. The Deletion Algorithm

We use the terminology of [Bo90]. Accordingly, let the line to be deleted be L_x and its adjacent lines on the sorted list be denoted by L_i and L_j. These lines are called the enclosing lines of L_x. The three pairwise intersection points of these three lines are denoted by a_{ix}, a_{jx} and a_{ij}. The algorithm proposed in [Bo90] is described in the following six steps. The convex hull of the intersection points of n lines is denoted by CH(n).

Step 1: If the input line does not exist in the list of lines, then exit.

Step 2: Determine the enclosing lines of the input line.

A Counter-example to a Dynamic Algorithm for Convex Hulls of Line Arrangements

Binay Bhattacharya
Hazel Everett
Godfried Toussaint

McGill University
Canada

Published in: ***Pattern Recognition Letters***, vol. 12, March 1991, pp. 145-147.

ABSTRACT

An algorithm was recently published to maintain dynamically the convex hull of the set **I** of intersection points of a set **L** of *n* lines in the plane [Bo90]. 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, we show in this note that the deletion algorithm proposed is incorrect.

1. Introduction

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 $\mathbf{I} = \{p_{ij} \mid 1 \leq i,j \leq n\}$ 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] whereas a survey of recent research results can be found in [EGS90]. 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}. The intersection point p_{ij} for $i \neq j$ and $1 \leq i,j \leq n$ is said to be *critical* if and