4. Bobrow, D.G., Burchfiel, J.D., Murphy, D.L., and Tomlinson, R.S. TENEX, a paged time sharing system for the PDP-10. *Comm. ACM 15*, 3 (March 1972), 135–143.
5. Bobrow, D.G., and Murphy, D.L. The structure of a LISP system using two level storage. *Comm. ACM 10*, 3 (March 1967). 155–159.
6. Cheney, C.J. A nonrecursive list compacting algorithm. *Comm. ACM 13*, 11 (Nov. 1970), 677–678.
7. Deutsch, L.P. An interactive program verifier. Ph.D. Th., Comptr. Sci. Dep., U. of California, Berkeley, Calif., May 1973.
8. Deutsch, L.P. A LISP machine with very compact programs. Third Int. Joint Conf. on Artificial Intelligence, Stanford, Calif., 1973, pp. 697–703.
9. Fenichel, R.R., and Yochelson, J.C. A LISP garbage-collector for virtual-memory computer systems. *Comm. ACM 12*, 11 (Nov. 1969), 611–612.
10. Hansen, W.J. Compact list representation: Definition, garbage collection, and system implementation. *Comm. ACM 12*, 9 (Sept. 1969), 499-507.
11. Hehner, E.C.R. Matching program and data representations to a computing environment. Ph.D. Th., Comptr. Systems Res. Group, U. of Toronto, Toronto, Canada, Nov. 1974.
12. McCarthy, J., et al. *LISP 1.5 Programmer's Manual*. M.I.T. Press, Cambridge, Mass., 1962.
13. Minsky, M.L. A LISP garbage collector algorithm using serial secondary storage. Artificial Intelligence Proj. Memo 58 (rev.), Project MAC, M.I.T., Cambridge, Mass., Dec. 1963.
14. Quam, L.H. Stanford LISP 1.6 manual. Artificial Intelligence Proj., Stanford University, Stanford, Calif., Sept. 1969.
15. Reboh, R., and Sacerdoti, E. A preliminary QLISP manual. Tech. Note 81, Stanford Res. Inst. AI Center, Menlo Park, Calif., Aug. 1973.
16. Sacerdoti, E. The nonlinear nature of plans. Fourth Int. Joint Conf. on Artificial Intelligence, Tbilisi, Georgia, U.S.S.R., 1975, pp. 206–214.
17. Shannon, C.E. A mathematic theory of communication. *Bell System Tech. J. 27* (July 1948), 379–423.
18. Smith, D.H., Masinter, L.M., and Sridharan, N.S. Heuristic DENDRAL: Analysis of molecular structure. In *Computer Representation and Manipulation of Chemical Information*, W.T. Wipke, S. Heller, R. Feldman, and E. Hyde, Eds., Wiley, New York, 1974.
19. Teitelman, W. INTERLISP Reference manual. Xerox Palo Alto Res. Center, Palo Alto, Calif., 1974.
20. Van der Poel, W.L. A Manual of HISP for the PDP-9. Technical U., Delft, Netherlands.
21. Weissman, C. *LISP 1.5 Primer*. Dickenson Pub. Co., Belmont, Calif., 1967.
22. Wilner, W.T. Design of the B1700. Proc. AFIPS 1972 FJCC, Vol. 41, AFIPS Press, Montvale, N.J., pp. 579–586.
23. Zipf, G.K. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Reading, Mass., 1949.

Programming Techniques

G. Manacher S.L. Graham Editors

Convex Hulls of Finite Sets of Points in Two and Three Dimensions

F. P. Preparata and S. J. Hong
University of Illinois at Urbana-Champaign

The convex hulls of sets of n points in two and three dimensions can be determined with O(n log n) operations. The presented algorithms use the "divide and conquer" technique and recursively apply a merge procedure for two nonintersecting convex hulls. Since any convex hull algorithm requires at least O(n log n) operations, the time complexity of the proposed algorithms is optimal within a multiplicative constant.

Key Words and Phrases: computational complexity, convex hull, optimal algorithms, planar set of points, spatial set of points

CR Categories: 4.49, 5.25, 5.32

1. Introduction

The determination of the convex hull of a finite set of points is relevant to several problems in computer graphics, design automation, pattern recognition and operations research: references [3, 4, 10]—just to cite a few—discuss some interesting applications in these areas, which require convex hull computation.

This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-72-C-0259.

Authors' addresses: F.P. Preparata, Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801; S.J. Hong, IBM Systems Product Division, Poughkeepsie, NY 12602. This work was completed while the second author was on leave at the University of Illinois.

We also mention that other important questions, such as set separability or existence of linear decision rules, are easily solved through the determination of convex hulls.

This problem has received some attention in recent times. Chand and Kapur [2] described a convex hull algorithm for a finite set of n points in a space with an arbitrary number of dimensions. Their approach is based on the so-called "gift wrapping" principle and requires a number of operations $O(n^2)$. The first convex hull algorithm to run in time less than $O(n^2)$ was found by R.L. Graham [5] for a set of points in the plane. Graham's method, based on representing the points in polar coordinates and sorting them according to their azimuth, has a running time $O(n \log n + Cn)$, for some constant C determined by the Cartesian-to-polar coordinate conversion. More recently, R.A. Jarvis [7] presented an alternative algorithm for the planar case, which is the two-dimensional specialization of the Chand-Kapur method and runs in time $O(nm)$, where m is the number of points on the hull. Subsequent to the original submission of this paper, we learned of the enlightening doctoral thesis of M.I. Shamos, which contains several convex hull algorithms with running time $O(n \log n)$ for a set of n points in the plane ([9], problems P3, P15, POL5c).

A problem of long standing has been the computation of convex hulls in more than two dimensions in time less than $O(n^2)$. In this paper we show that the convex hull of a finite set of n points in three dimensions can be computed with at most $O(n \log n)$ operations. We also report an algorithm for the plane, which is original and can be viewed as the two-dimensional specialization of the three-dimensional algorithm. The latter is based on the property that the number of edges of the convex hull of n points is at most linear in n. For this reason, its generalization based on edges is impossible beyond three dimensions, since in more than three dimensions it is known that there exist convex polytopes with n vertices whose numbers of edges are $O(n^2)$ (see [6, p. 193]).

The computation model we shall refer to is that of a random access machine (RAM) in the sense of Aho, Hopcroft, and Ullman ([1, pp. 5–24]), with the only modification that real number arithmetic replaces integer arithmetic. Our algorithms are based on the well-known technique called "divide and conquer." Let E^d denote the d-dimensional Euclidean space and let the set $S = \{a_1, \cdots, a_n \mid a_j \in E^d\}$ be given. Without loss of generality, the points of S are assumed to be given in Cartesian coordinates: in fact, in the adopted computation model, a conversion between coordinate systems can be done in a constant time per point. By $x_i(a)$ we denote the ith coordinate of $a \in E^d$, for $i = 1, \cdots, d$. Here and hereafter we assume that for any two points u and v in E^d we have $x_i(u) \neq x_i(v)$, for $i = 1, \cdots, d$. This simplification helps bring out the basic ideas of the algorithms to be described, while the modifications required for the unrestricted case are straightforward.

As a preliminary step we sort the elements of S according to the coordinate x_1, and relabel them if necessary so that we may assume $x_1(a_i) < x_1(a_j) \Leftrightarrow i < j$. We can now give the following general algorithm for a set of n d-dimensional points.

Algorithm CH

```
Input. A set $S = \{a_1, \ldots, a_n\}$, where $a_i \in E^d$ and $x_1(a_i) < x_1(a_j) \Leftrightarrow i < j$ for $i, j = 1, \ldots, n$.
Output. The convex hull $CH(S)$ of $S$.
Step 1. Subdivide $S$ into $S_1 = \{a_1, \ldots, a_{\lfloor \frac{1}{2}n \rfloor}\}$ and $S_2 = \{a_{\lfloor \frac{1}{2}n \rfloor + 1}, \ldots, a_n\}$.
Step 2. Apply recursively Algorithm CH to $S_1$ and $S_2$ to obtain $CH(S_1)$ and $CH(S_2)$.
Step 3. Apply a merge algorithm to $CH(S_1)$ and $CH(S_2)$ to obtain $CH(S)$ and halt.
```

The initial sorting of the x_1 coordinates of the elements of S requires $O(n \log n)$ operations. Notice that, because of this sorting and of step 1, the sets $CH(S_1)$ and $CH(S_2)$ will define two nonintersecting convex domains. Now, if the merging of two convex hulls with at most n d-dimensional extreme points in total requires at most $P_d(n)$ operations, an upper-bound to the number $C_d(n)$ of operations required by Algorithm CH is given by the equation

$$C_d(n) = 2C_d(\tfrac{1}{2}n) + P_d(n).$$

(Notice that we have assumed that n be even for simplicity, but practically without loss of generality). Thus, if we can show that $P_d(n)$ is $O(n)$, we shall obtain that $C_d(n)$ is $O(n \log n)$, and, taking into account the initial sorting pass, an overall complexity $O(n \log n)$ results for the convex hull determination.

In Sections 3 and 4 we shall show that merging algorithms with number of operations $O(n)$ can be designed for $d = 2, 3$. In the next section we shall establish a lower bound to the number of operations performed by any algorithm for finding the convex hull of a set of n points. Since this computational work is of at least the same order as that of an algorithm for sorting n numbers, i.e. it is $O(n \log n)$, we reach the interesting conclusion that the proposed convex hull algorithms for finite sets in two and three dimensions are optimal in their order of complexity, within a multiplicative constant.

2. Lower Bound

In this section we present a lower bound to the running time of any algorithm which computes the convex hull of a set of n points.

The arguments presented are similar to those developed in connection with finding the maxima of a set of vectors ([8, 11]), which is a problem related to the one being investigated. Incidentally, we note that