INFORMATION PROCESSING LETTERS 2 (1973) 18–21. NORTH-HOLLAND PUBLISHING COMPANY

ON THE IDENTIFICATION OF THE CONVEX HULL OF A FINITE SET OF POINTS IN THE PLANE

R.A. JARVIS

The Australian National University, Department of Statistics, Box 4, Canberra, A.C.T. 2600, Australia

Received 6 December 1972

convex hull algorithm

1. Introduction

This paper presents an extremely simple algorithm for identifying the convex hull of a finite set of points in the plane in essentially, at most $n(m+1)$ operations for n points in the set and $m \leq n$ points on the convex hull. In most cases far less than $n(m+1)$ operations are necessary because of a powerful point deletion mechanism that can easily be included. The operations are themselves trivial (computationally inexpensive) and consist of angle comparisons only. Even these angle comparisons need not be actually carried out if an improvement suggested in a later section is implemented. Although Graham's algorithm [1] requires no more than $(n \log n)/\log 2 + Cn$ operations*, the operations are themselves more complex than those of the method presented here; in particular, Graham's method would not be as efficient for low m.

2. Geometric interpretation

The underlying method of the algorithm can be described simply: find an origin point outside the point set and swing a radius arm in an arbitrary direction until a point of the set is met; this point becomes

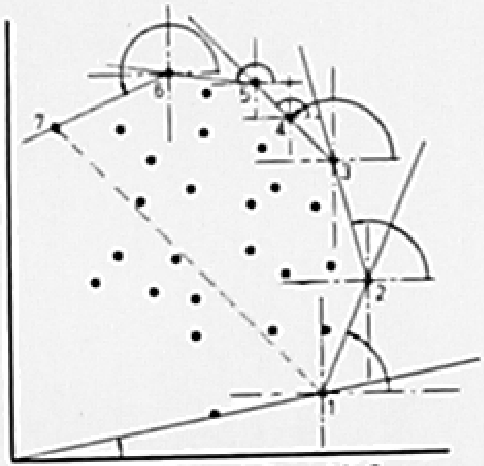

Fig. 1. Geometric interpretation of the algorithm.

the first point on the hull. Make this the new origin point and swing a radius arm from this point in the same direction as before till the next hull point is found. Repeat until the points are enclosed by the convex hull. Delete points from further consideration if

(i) they have already been identified as being on the convex hull,

(ii) they lie in the area enclosed by a line from the first to the last convex hull point found and the lines joining the convex hull points in the sequence found.

Fig. 1 illustrates this geometric interpretation.

* To quote Graham, "*C* is a small positive constant which depends on what is meant by an 'operation'". In fact, *C* is distributed over the five basic steps of Graham's algorithm and his paper should be consulted for detailed interpretation.

Fig. 2. Finding an origin outside the finite point set.

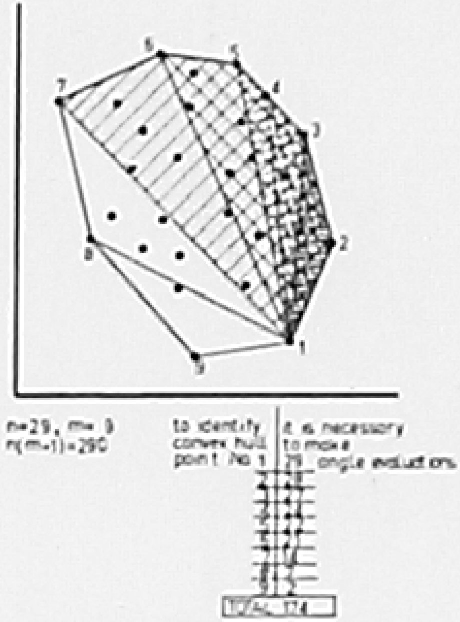

Fig. 3. Point deletion mechanism.

3. Convex hull algorithm

The following notation is adopted:

$$S \equiv \{S_i\}, \quad i = 1, 2, ..., n,$$

is the finite set of points in the plane.

$$S_i \equiv (X_i, Y_i), \quad i = 1, 2, ..., n,$$

where X_i and Y_i are the cartesian co-ordinates of the ith point in the set.

Step 1. Find an origin point outside the set (for example by picking

$$X_{\text{origin}} < \min_i \{X_i\} \quad \text{and} \quad Y_{\text{origin}} < \min_i \{Y_i\}$$

(see fig. 2)).

Step 2. Find S_k such that

$$\theta_{0k} \leq \min_i \{\theta_{0i}\}, \quad i = 1, 2, ..., n,$$

where θ_{0i} is the angle measured with respect to a radial arm from the origin in an arbitrary direction (e.g. anti-clockwise) from an arbitrary zero outside the set (e.g. horizontal to right). For equal minimum angles pick the point closest to the origin.

Step 3. Shift origin to S_k and repeat step 2 with consistent angle direction and origin until first convex hull point is re-found.

There will be $(m+1)$ iterations of step 2 for m convex hull points so that the number of angle evaluating and comparing operations is less than $n(m+1)$. (Only $(n-1)$ angles need be evaluated for each iteration after the first.) Step 1 is trivial and the computational cost has not been included; in many cases a point outside the set may be known 'a priori' and would not have to be found.

In many cases, depending on the distribution of points in the given set, considerable computational savings result by including the following deletion check:

(a) After step 2 above has been completed once and the first convex hull point identified, evaluate and store the angles of lines from the first hull point to the $(n-1)$ other points in the set consistent with whatever convention is adopted for angle measurements.

(b) Within step 2 after the first iteration: any point whose angle calculated and stored above is less than the angle of the last found convex hull point (also stored above) cannot be a valid candidate for a convex hull point and can be passed over for all future iterations of one algorithm (see fig. 3).

The slight extra complexity introduced by including (a) and (b) above can be very worthwhile for large n and evenly distributed points. If however, many points are clustered near the last convex hull point, (a

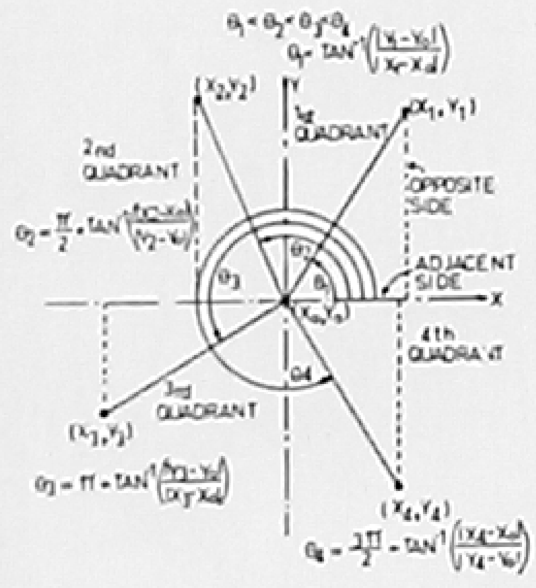

Fig. 4. Obviating actual angle evaluation.

somewhat unusual case) the advantage is lost (unless one reverses the direction of scan).

4. Improvements

It is possible to obviate the need to actually calculate the $n(m+1)$ angles of the basic algorithm by keeping a four item list of opposite side/adjacent side ratios for the right angled triangles related to the angle measures (see fig. 4):

Suppose angles are measured anticlockwise from the horizontal to the right; first quadrant angles are smaller than 2nd quadrant angles < 3rd quadrant angles < 4th quadrant angles. By checking the signs of $(X_i - X_0)$ and $(Y_i - Y_0)$, $i = 1, 2, ..., n$, quadrant determination can be made. Keep only the lowest ratio (opposite side/adjacent side ratio for quadrants 1 and 3, inverse ratio for quadrants 2 and 4) for the lowest numbered quadrant yet detected while working through the points. In many cases the ratio division will not be necessary because sign checks will be sufficient to allow a point to be discarded. The smallest ratio for the lowest quadrant detected will identify the next convex hull point within step 2 of the basic algorithm.

Angles of $\pi/2$, π and $3\pi/2$ will have to be considered as special cases but they can be accommodated relatively easily. The above substitute for actual angle evaluations can be used with the point deletion addition to the basic algorithm.

5. Illustrations

Fig. 5a, b, c, d shows a number of convex hull identification results for sets of points in the plane. Colinear points are considered members of the convex hull if they occur along the convex hull boundary. Examples (c) and (d) are meant to be vaguely suggestive of possible applications in the picture processing field.

6. Conclusions

The method of convex hull identification presented in this paper is simple, direct and easily computed. Inclusion of point deletion checks and use of opposite side/adjacent side ratios to replace angle evaluation and comparison makes the algorithms a fast and widely applicable one which is especially effective when the number of points on the convex hull is relatively small compared to the number of points in the finite set ($m < n$). The algorithm presented can be extended to more than two dimensions but the details have not yet been investigated by the author. The algorithm also lends itself to parallel processing in that clockwise and anti-clockwise wrap arounds can be started at a number of points simultaneously; overlaps can be detected to terminate the various processes.

Reference

[1] R.L. Graham, An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set, Information Processing Letters 1 (1972) 132–133.

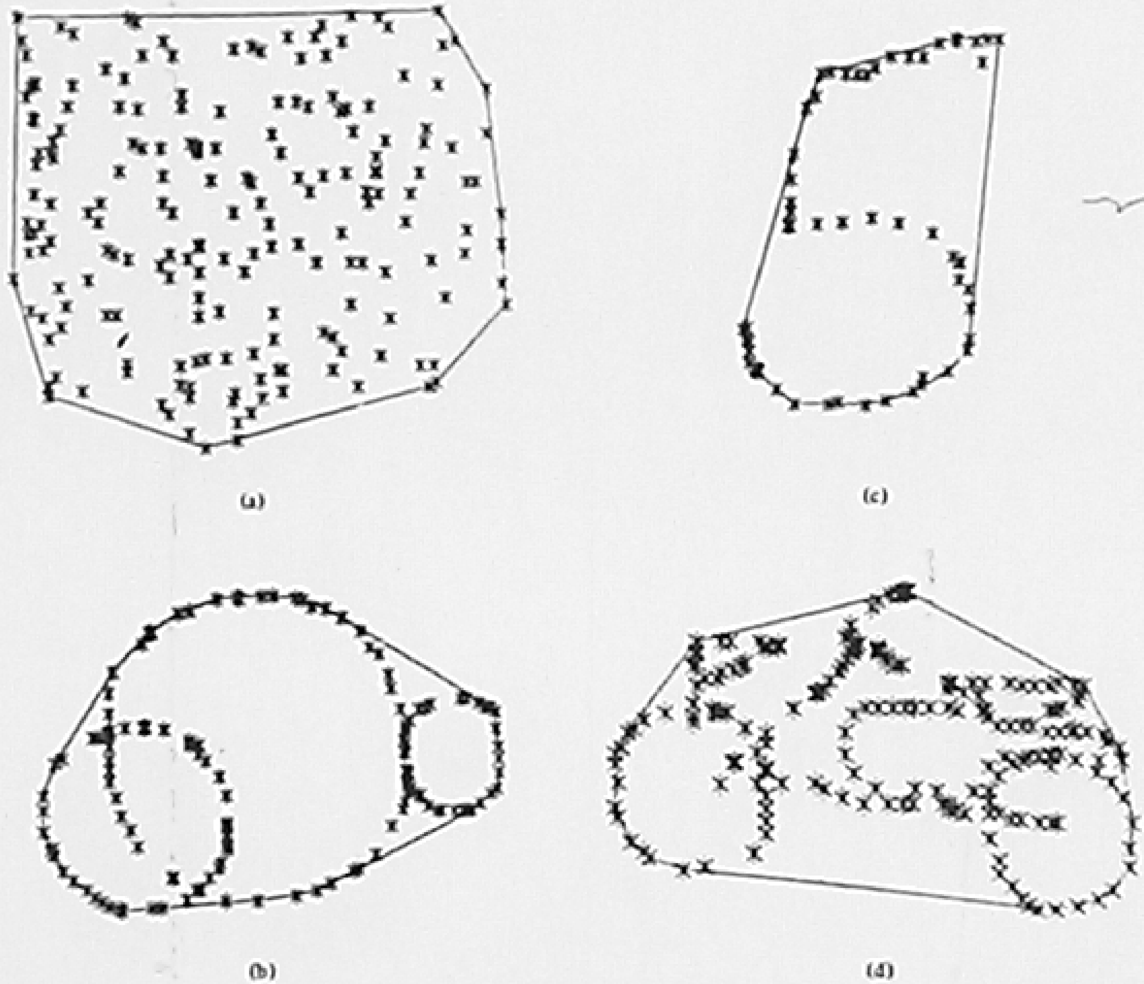

Fig. 5. Convex hull identification examples.