Information Processing Letters 25 (1987) 11–12
North-Holland
20 April 1987

ON-LINE CONSTRUCTION OF THE CONVEX HULL OF A SIMPLE POLYLINE

Avraham A. MELKMAN *

Ben Gurion University of the Negev, Beer Sheva, Israel

Communicated by Alan Shaw
Received 2 April 1985
Revised 27 November 1985 and 9 September 1986

Keywords: Convex hull, simple polygon, analysis of algorithms

1. Introduction and description of the algorithm

After McCallum and Avis [4] showed that the convex hull of a simple polygon P with n vertices can be constructed in O(n) time, several authors [1,2,3] devised simplified algorithms for this problem. Graham and Yao [2] presented a particularly simple and elegant one. After finding two points of the convex hull, their algorithm generated all other hull vertices using only one stack for intermediate storage. It is the purpose of this short article to show that a slightly modified version of their algorithm constructs, on-line, the convex hull of any simple polyline in O(n) time. In contrast, the on-line construction of a nonsimple polyline requires O(n log n) time, as shown by Preparata [5]. In the special case of a simple polygon our algorithm produces the convex hull without first identifying two of the hull vertices, as was required in [2]. The price that we pay is the use of a deque instead of a queue. After this article was submitted, we learned of a similar approach taken by Tor and Middleditch [6], who embed it in an algorithm for the convex decomposition of a simple polygon.

To keep this article short, we use where possible the definitions and notations of Graham and Yao. The polyline $P = \langle v_1, \ldots, v_m \rangle$ is assumed to be given by an input list of its vertices as they are encountered in an ordered traversal, and the algorithm will consider the vertices in that order. For an ordered triple of points x, y, z, the function (x, y, z) assumes values 1, 0, or -1 depending on whether z is to the right of, collinear with, or to the left of the directed line from x to y.

The main data structure used is a deque of vertices, $D = \langle d_b, \ldots, d_t \rangle$, where the variable b points to the bottom of the deque and t to its top, and d_b and d_t will always refer to the same vertex. Thus, D describes a closed curve; the polygon enclosed by this curve will also be referred to as D. Variable v refers to the input vertex under consideration. 'Pushing v' means executing $[t \leftarrow t+1; d_t \leftarrow v]$, 'popping d_t' means setting $[t \leftarrow t-1]$, 'inserting v' means executing $[b \leftarrow b-1; d_b \leftarrow v]$, and 'removing d_b' means setting $[b \leftarrow b+1]$. The algorithm halts when its input is exhausted.

Algorithm Hull

1. $t \leftarrow -1; b \leftarrow 0;$
   $v_1 \leftarrow$ input; $v_2 \leftarrow$ input; $v_3 \leftarrow$ input;
   if $(v_1, v_2, v_3) > 0$
      then begin push v_1; push v_2; **end**
      else begin push v_2; push v_1; **end**
   push v_3; insert v_3;
2. $v \leftarrow$ input;
   until $(v, d_b, d_{b+1}) < 0$ **or** $(d_{t-1}, d_t, v) < 0$
      do $v \leftarrow$ input **end**;

* Present affiliation: Department of Computer Science, Columbia University, New York, NY 10027, U.S.A.

3. until $(d_{t-1}, d_t, v) > 0$ **do** pop d_t **end**;
push v;
4. until $(v, d_b, d_{b+1}) > 0$ **do** remove d_b **end**;
insert v;
goto 2.

2. Correctness of Algorithm Hull

In this section we prove the following theorem.

Theorem. *Algorithm Hull finds the convex hull of* P *correctly and in linear time.*

Proof. Evidently, the algorithm is linear in the number of vertices of P, as each vertex is pushed (inserted) at most once and popped (removed) at most once. We use the following characterization of the convex hull H of P:

(i) H is convex,
(ii) P is contained in H,
(iii) the set of vertices of H is a subset of the set of vertices of P.

The last property is built in to the algorithm. Thus, we will prove by induction that the following hypothesis is true each time a new vertex is read in step 2:

(H) The deque $D = \langle d_b, \ldots, d_t \rangle$, considered as a polygon, is convex and contains that part of the polyline P seen so far.

First of all, observe the following.

Claim. *All vertices rejected in step* 2 *are in, or on, the current convex hull contained in* D.

Indeed, if a vertex v is rejected, it is because both $(d_b, d_{b+1}, v) \geqslant 0$ and $(d_{t-1}, d_t, v) \geqslant 0$, i.e., v is to the right of both the edges $\langle d_{t-1}, d_t \rangle$ and $\langle d_b, d_{b+1} \rangle$. Moreover, the polyline connects v with d_t and, because it is simple, does not cross the part of the polyline connecting d_{b+1} and d_{t-1}. Hence, v must lie within the convex polygon described by D.

Step 1 ensures that (H) holds when three vertices have been read (assuming they are not collinear). Suppose then inductively that (H) holds when k vertices have been read. Because of the Claim it may be assumed that v satisfies either $(d_b, d_{b+1}, v) < 0$ or $(d_{t-1}, d_t, v) < 0$ (or both), and that steps 3 and 4 are executed immediately after v is read and before the next vertex is read.

Suppose that, before execution of steps 3 and 4, the deque is $D = \langle d_b, \ldots, d_t \rangle$ and that after their execution the deque is $D' = \langle d_k, \ldots, d_m \rangle$, with $v = d_k = d_m$. To prove D′ to be convex we show that it describes a closed simple curve and that $(d_i, d_{i+1}, d_{i+2}) > 0$, $i = k, \ldots, m-2$, and $(d_{m-1}, d_m, d_{k+1}) > 0$. Because D is simple, D′ can only be nonsimple if the edges $\langle v, d_{k+1} \rangle$ and $\langle d_{m-1}, v \rangle$ intersect edges of D; this cannot be the case since the polyline P is simple. Now, $(d_i, d_{i+1}, d_{i+2}) > 0$, $i = k, \ldots, m-2$, is either already true for D or else it is the result of steps 3 and 4. Suppose then by contradiction that $(d_{m-1}, d_m, d_{k+1}) \leqslant 0$. Then, $v = d_m$ is on or to the right of the directed line from d_{m-1} to d_{k+1}, and at the same time it must be to the right of the polyline $\langle d_{k+1}, \ldots, d_{m+1} \rangle$. Thus, v is within D, a contradiction.

Finally, it is easily seen by induction that P is contained in D′, because it is contained in D and each time a vertex d_j is popped, the polygon $\langle v, d_i, \ldots, d_j, v \rangle$ is contained within the polygon $\langle v, d_i, \ldots, d_{j+1}, v \rangle$; similarly, the removal of a vertex only enlarges the polygon. □

Acknowledgment

Karl Abrahamson and David Kirkpatrick devised a similar algorithm as part of their refereeing. The helpful suggestions of an anonymous referee, which simplified the algorithm and its proof, are also gratefully acknowledged.

References

[1] B.K. Bhattacharya and H. El Gindy, A new linear convex hull algorithm for simple polygons, IEEE Trans. Inform. Theory IT-30 (1984) 85–88.
[2] R.L. Graham and F.F. Yao, Finding the convex hull of a simple polygon, J. Algorithms 4 (1984) 324–331.
[3] D.T. Lee, On finding the convex hull of a simple polygon, Internat. J. Comput. Inform. Sci. 12 (1983) 87–98.
[4] D. McCallum and D. Avis, A linear algorithm for finding the convex hull of a simple polygon, Inform. Process. Lett. 9 (1979) 201–206.
[5] F.P. Preparata, An optimal real time algorithm for planar convex hulls, Comm. ACM 22 (1979) 402–405.
[6] S.B. Tor and A.E. Middleditch, Convex decomposition of simple polygons, ACM Trans. Graphics 3 (1984) 244–265.