THE ALL NEAREST-NEIGHBOR PROBLEM FOR CONVEX POLYGONS *

D.T. LEE and F.P. PREPARATA

Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, IL, USA

Received 29 August 1977; revised version received 20 March 1978

Nearest neighbor, computational geometry, computational complexity, analysis of algorithms, diameter of a convex polygon, Voronoi diagram

Introduction

The problem of finding the nearest neighbor for each of N arbitrary points in the Euclidean plane has been shown [1] to require time $O(N \log N)$, and algorithms achieving the lower bound have also been given in [1,2]. However, the lower bound does not apply to the same problem when the given points, rather than being arbitrarily placed, are the vertices of a convex polygon [4]. In this paper, we show that this additional property indeed enables us to obtain a linear time algorithm, whose running time is obviously optimal within a multiplicative constant.

Main result

Let a convex polygon P be denoted by a sequence of vertices $(p_0, p_1, ..., p_{N-1})$ in which $\overline{p_i p_{i+1}}$ **, $0 \leqslant i < N$, is an edge. Define an index set $I = \{0, 1, ..., N-1\}$. Let $d(p_i, p_j)$, $i, j \in I$, denote the distance between p_i and p_j and $D(P)$ denote the diameter of P, i.e.,

$$D(P) = \max_{i,j \in I} d(p_i, p_j),$$

the largest distance between the vertices of P. The nearest neighbor $NN(p_i)$ pf p_i is p_j such that

$$d(p_i, p_j) = \min_{k \in I - \{i\}} d(p_i, p_k).$$

Consider now the following conditions:

Condition (i): The two farthest points of P are the extremes of an edge, i.e., $D(P) = d(p_i, p_{i+1})$ for some i.

Condition (ii): all vertices of P lie inside a circle with diameter $D(P)$. A convex polygon P that satisfies both (i) and (ii) is said to have the *semi-circle property*. Fig. 1 shows a convex polygon having the semi-circle property.

Lemma 1. *Given a convex polygon $P = (p_0, p_1, ..., p_{N-1})$, there exists a linear time algorithm to decompose it into at most four convex polygons which have the semi-circle property.*

Proof. We first apply the linear time algorithm [3] to find the diameter. Let $D(P) = d(p_u, p_v)$. The chord $\overline{p_u p_v}$ will, in general, divide P into two convex poly-

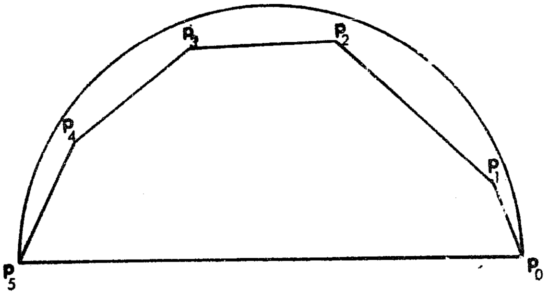

Fig. 1. A convex polygon with the semi-circle property.

* This work was supported in part by the National Science Foundation under Grant MCS-76-17321 and in part by the Joint Services Electronics Program under Contract DAAB-07-72-C-0259.

** All indices in the text are taken modulo N.

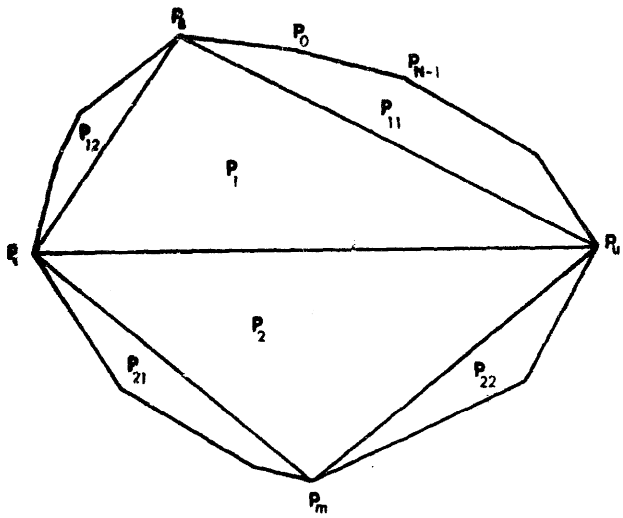

Fig. 2. Decomposition of a convex polygon into four convex polygons satisfying semi-circle property.

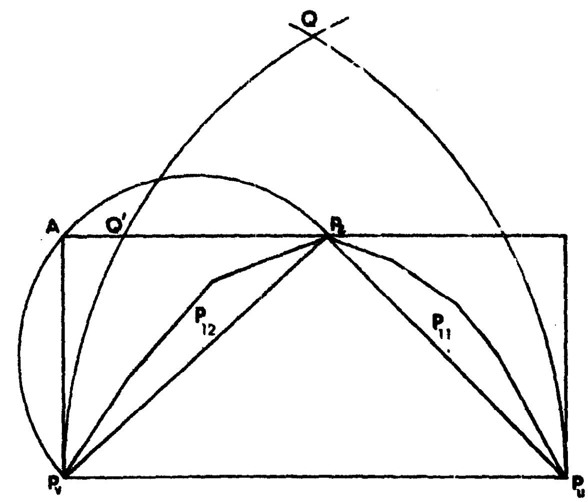

Fig. 3. Proof that P_{12} has semi-circle property.

gons

$$P_1 = (p_u, p_{u+1}, ..., p_v)$$

and

$$P_2 = (p_v, p_{v+1}, ..., p_{N-1}, p_0, ..., p_u)$$

(Fig. 2), where $D(P_1) = D(P_2) = d(p_u, p_v)$. Let $p_l \in P_1$ be the vertex with largest distance from the chord $\overline{p_u p_v}$. Let $p_m \in P_2$ be defined similarly. It is obvious that p_l and p_m can be found in O(N) time. p_l will determine two convex polygons

$$P_{11} = (p_u, p_{u+1}, ..., p_l)$$

and

$$P_{12} = (p_l, p_{l+1}, ..., p_v).$$

Similarly, p_m determines the two polygons

$$P_{21} = (p_v, p_{v+1}, ..., p_m)$$

and

$$P_{22} = (p_m, p_{m+1}, ..., p_{N-1}, p_0, ..., p_u).$$

We claim that P_{11}, P_{12}, P_{21}, and P_{22} satisfy the semi-circle property. Without loss of generality we shall just consider P_{12}.

In Fig. 3, since $p_u p_v$ is the longest chord, all vertices $p_u, p_{u+1}, ..., p_v$ must lie within the region $p_u Q p_v$, where Q is the intersection of the two circles with radius $d(p_u, p_v)$ and centered at p_u and p_v respectively. Let Q' be the intersection of the circular arc $\widehat{p_v Q}$ and the line through p_l parallel to $\overline{p_u p_v}$. By the definitions of $\overline{p_u p_v}$ (longest chord) and of p_l (vertex of P_1 farthest from $\overline{p_u p_v}$) and by convexity the vertices of P_{12} lie in the region $\mathcal{A}$ delimited by the straight-line segments $\overline{p_v p_l}$ and $\overline{p_l Q}'$, and by the arc $\widehat{p_v Q}'$. Let A' be the intersection of the normal to $\overline{p_u p_v}$ in p_v and the line containing $\overline{p_l Q}'$; clearly region $\mathcal{A}$ is contained in the right triangle $p_v A' p_l$, which is in turn contained in the semi-circle having $\overline{p_l p_v}$ as its diametral chord. This proves that P_{12} has the semi-circle property.

Lemma 2. *Given a convex polygon $P = (p_0, p_1, ..., p_{N-1})$ with the semi-circle property, for any vertex p_i, its nearest neighbor p_j is adjacent to p_i, i.e., either $j = i + 1$ or $j = i - 1$.*

Proof. Without loss of generality, we may assume that $D(P) = d(p_0, p_{N-1})$. Suppose for some p_i, $NN(p_i) = p_k$ where $k > i + 1$ (Fig. 4). Consider the triangle $p_i p_{i+1} p_k$. Since $d(p_i, p_k) < d(p_i, p_{i+1})$ by assumption, the angle $\measuredangle\, p_i p_{i+1} p_k$ is less than the angle $\measuredangle\, p_i p_k p_{i+1}$. By convexity, p_i and p_k are external to the triangle $p_0 p_{i+1} p_{N-1}$. Thus $\measuredangle\, p_i p_{i+1} p_k$ must be greater than $\measuredangle\, p_0 p_{i+1} p_{N-1}$, which is not smaller than $\pi/2$ by the semi-circle property of the given polygon. That is, $\measuredangle\, p_i p_k p_{i+1} > \measuredangle\, p_i p_{i+1} p_k > \pi/2$ which is impossible. Therefore $NN(p_i)$ must be adjacent to p_i, for all $i \in I$.

Lemma 3. *Given a convex polygon $P = (p_0, p_1, ..., p_{N-1})$ satisfying condition* (i) *above, the set of nearest*

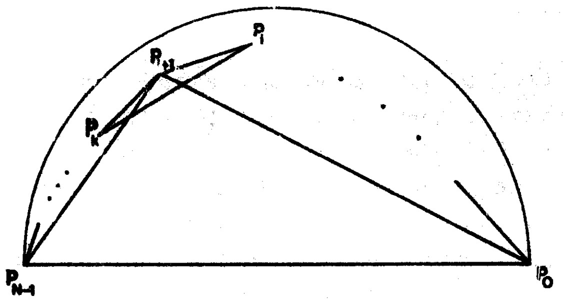

Fig. 4. Illustration of the proof of Lemma 2.

neighbors $\{NN(p_i) | 0 \leq i \leq N-1\}$ *can be found in* O(N) *time.*

Proof. Suppose $D(P) = d(p_0, p_{N-1})$. Let p_i be the vertex with the largest distance from the chord $\overline{p_0 p_{N-1}}$. By Lemma 1, the two convex polygons

$P_1 = (p_0, p_1, ..., p_i)$

and

$P_2 = (p_i, p_{i+1}, ..., p_{N-1})$ have the semi-circle property.

have the semi-circle property. The nearest neighbor of each vertex in P_s, (s = 1, 2), can be found separately by a simple scan through the vertices of P_s by Lemma 2, in time O(N). We must still check whether the nearest neighbor of a vertex in, say P_1, belongs to P_2, and this can be done as follows. Let p_{N-1} be the origin of the plane, and let the chord $\overline{p_{N-1} p_0}$, directed from p_{N-1} to p_0, define the positive x-axis; also let l denote the vertical line through p_i (Fig. 5). Notice that, for $0 \leq j < k \leq i$, $y(p_j) \leq y(p_k)$; while for $i \leq j < k \leq N-1$, $y(p_k) \leq y(p_j)$.

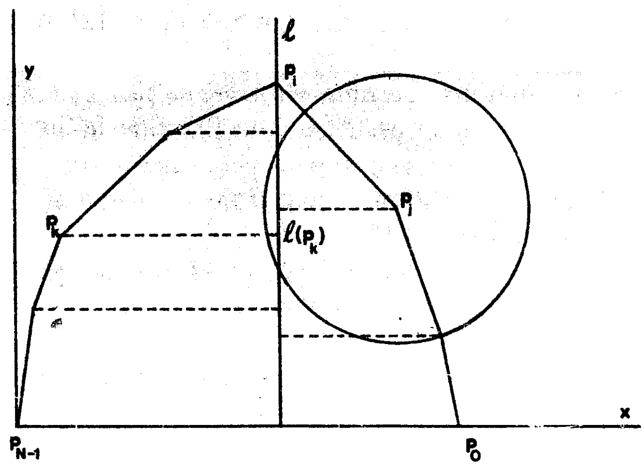

Fig. 5. The vertices of P_1 and P_2 are projected on l (and they are ordered as in the corresponding polygons).

Without loss of generality, let $p_j \in P_1$, and let $\delta(p_j)$ be the distance between p_j and its nearest neighbor in P_1; also, let δ*-circle*(p_j) denote the circle with radius $\delta(p_j)$ centered at p_j. A vertex $p_k \in P_2$ is a candidate for being the nearest neighbor $NN(p_j)$ of p_j in P only if the projection $l(p_k)$ of p_k on l is contained in δ-circle(p_j). This is the basis for the following algorithm, which determines $NN(p_j)$ in P for each $p_j \in P_1$, and uses a function, UPDATE(p_j, p_k), which compares $d(p_j, p_k)$ with $\delta(p_j)$ and updates $NN(p_j)$ accordingly. Index s is used to scan the sequence $(p_i, p_{i-1}, ..., p_0)$ of the vertices of P_1. Clearly (step 3), if δ-circle(p_s) does not intersect l, $NN(p_s)$ has already been determined and we decrement s (step 11); otherwise, the conditions of the *while* loops (lines 5 and 8) simply inspect those $p_k \in P_2$ such that $y(v) \leq y(p_k) \leq y(u)$, as required. This proves the correctness of the proce-

```
1.  begin s ← i; s_F ← i + 1;
2.    while s ≥ 0 do (Comment: scan of P )
3.      begin if δ-circle(p_s) intersects l in u and v (y(u) ≥ y(v)) then
4.        begin s_B ← s_F − 1;
5.          while y(p_{s_F}) ≥ y(v) and s_F < N do (Comment: forward scan of P_2)
6.            begin if y(p_{s_F}) ≤ y(u) then NN(p_s) ← UPDATE(p_s, p_{s_F});
7.              s_F ← s_F + 1
              end;
8.          while y(p_{s_B}) ≤ y(u) and s_B > i do (Comment: backward scan of P_2)
9.            begin NN(p_s) ← UPDATE(p_s, p_{s_B});
10.             s_B ← s_B − 1
              end
          end;
11.     s ← s − 1
      end
  end
```

dure; as to its performance, it has been noted [2], that each $l(p_k)$, for $p_k \in P_2$, is contained in at most *four* δ-circles; thus the algorithm will examine each $p_k \in P_2$ at most four times (specifically, the first time in the forward scan of P_2, and all remaining times in the backward scan). We conclude that the running time of the algorithm is $O(N)$.

Analogously, in time $O(N)$ we can determine the nearest neighbor $NN(p_k)$ in P for each $p_k \in P_2$.

Based on the above lemmas, we have the following theorem.

Theorem. *Given a convex polygon* $P = (p_0, p_1, ..., p_{N-1})$, *the nearest neighbor of each vertex can be found in* $O(N)$ *time.*

Proof. Let $D(P) = d(p_u, p_v)$. The chord $\overline{p_u p_v}$ divides the polygon P into two polygons

$$P_1 = (p_u, p_{u+1}, ..., p_v)$$

and

$$P_2 = (p_v, p_{v+1}, ..., p_{N-1}, p_0, ..., p_u).$$

By Lemma 3, the nearest neighbor $NN(p_j) \in P_s$ of $p_j \in P_s$, $(s = 1, 2)$ can be found in $O(N)$ time. Now, we project all the vertices in P_s, $(s = 1, 2)$ onto the chord $\overline{p_u p_v}$. Since the projections of the vertices in P_s, $(s = 1, 2)$ are ordered, by a technique similar to that described in Lemma 3, we can find for each vertex $p_i \in P$, its nearest neighbor in $O(N)$ time. Since the diameter of P can be found in $O(N)$ time, the total running time is $O(N)$.

Conclusion

It is rather interesting that the nearest-neighbor problem for a set of N arbitrary points requires time $\Omega(N \log N)$, whereas the problem can be solved in linear time if the given set of points forms a convex polygon. In [1], the nearest neighbor problem was solved by the Voronoi diagram technique. The construction of the Voronoi diagram for a set of N points has also been shown to require $\Omega(N \log N)$ time [1]. But whether the construction of the Voronoi diagram for the set of vertices of a convex polygon can be solved in less than $O(N \log N)$ time still remains an open problem. However, we know at least that the nearest neighbor problem for the set of vertices of a convex polygon is not as time-consuming as the presently known techniques for constructing the Voronoi diagram for it.

References

[1] M.I. Shamos and D.J. Hoey, Closest-Point Problems, Proc. 16th Annual IEEE Symposium on Foundations of Computer Science (Oct. 1975) 151–162.

[2] J.L. Bentley and M.I. Shamos, Divide-and-Conquer in Multidimensional Space, Proc. 8th Annual ACM Symposium on Theory of Computing (May 1976) 220–230.

[3] M.I. Shamos, Geometric complexity, Proc. 7th Annual ACM Symposium on Theory of Computing (May 1975) 224–233.

[4] M.I. Shamos, Problems in Computational Geometry, Dept. of Computer Science, Yale University, New Haven, Conn. (May 1975) (to be published by Springer-Verlag).