Comput. Math. Applic. Vol. 13, No. 4, pp. 401–405, 1987
Printed in Great Britain.

0097-4943/87 $3.00 + 0.00

LINEAR PROGRAMMING IN $\mathbb{R}^3$ AND THE SKELETON AND LARGEST INCIRCLE OF A CONVEX POLYGON

M. ORLOWSKI and M. PACHTER
NRIMS/CSIR, P.O. Box 395, Pretoria 0001, South Africa

(*Received* 27 *August* 1986)

Communicated by E. Y. Rodin

Abstract—In this paper the geometrical problem of constructing the largest circle inscribed in a (given) convex polygon is solved in $0(n)$ time. This problem is related to the construction of the skeleton of the polygon, which construction is shown to be accomplishable in $0(n \log n)$ time.

1. INTRODUCTION

The coloring in of a two-dimensional (polygonal) figure in a raster graphics environment is accomplished by projecting the basic colors onto the planar domain of the figure. The footprint of the CRT (cathode ray tube) projection is circular, which gives rise to the theoretical and also the practical problems of how to inscribe the largest possible circle in a given convex polygon.

In conformity with the paradigm of computational geometry, we tacitly assume the convex polygon to be given in the form of a counterclockwise listing of the planar coordinates of the vertices of the n-gon consideration. Obviously, the largest possible incircle of the convex polygon will be in contact with three edges of the polygon.† Furthermore, the three straight lines that are determined by the above three edges and that are oriented in the direction of the edges must satisfy a "Triangle Condition"—namely, by moving in the direction of the lines a closed triangular circuit ensues. Thus, the largest incircle of the polygon corresponds to the incircle of the above triangle. Indeed, the validity is readily verifiable for the $0(n^3)$-time brute force method where one considers all those $\approx\binom{n}{3}$ edge triplets that in addition satisfy the Triangle Condition, and in each triangle one then inscribes a circle; the largest incircle of the polygon is the smallest circle in a triangle.

In Section 2 of this paper we propose an efficient $0(n)$ time algorithm for the construction of the largest incircle of the convex polygon. We employ a linear programming (LP) formulation and exploit the recent result of Megiddo [1] and Dyer [2] concerning linear–time algorithms for LP in $\mathbb{R}^3$. The largest incircle of a convex polygon is intimately related to the "skeleton" of the polygon. Roughly speaking, the skeleton is a generalized (medial) axis of symmetry of the polygon. The skeleton idea was introduced by Blum [3], and it was further developed by Montanari [4]. Thus, in Section 3 we give an $0(n \log n)$–time algorithm for the construction of the skeleton of a convex polygon. In Section 4 we present an additional $0(n \log n)$–time approach to the construction of skeletons of convex polygons, and we also comment on the related dual problem of the smallest enclosing circle.

2. THE LARGEST INCIRCLE OF A CONVEX POLYGON

Assume that our convex n-gon is given by the set of n linear inequalities‡

$$a_{i1}x_1 + a_{i2}x_2 \leqslant b_i \quad i = 1, 2, \ldots, n, \tag{1}$$

†This is so for a polygon in general position. Without loss of generality, we shall ignore, in the sequel, the rather exceptional case of a rectangle, illustrated in Fig. 2a; thus, if two sides of the polygon are parallel, then some incircles will be in contact with the two parallel sides of the polygon, and the largest possible incircle will not be unique.

‡Note that formulation (1) can obviously be produced in $0(n)$ time from the polygon vertices listing.

where the coefficients satisfy, for all $1 \leqslant i \leqslant n$,

$$a_{i1}^2 + a_{i2}^2 = 1. \tag{2}$$

Then b_i is the distance of the ith line

$$a_{i1}x_1 + a_{i2}x_2 = b_i$$

from the origin and the expression,

$$b_i - (a_{i1}x_1 + a_{i2}x_2)$$

yields the distance from the ith line of the planar point (x_1, x_2). Furthermore,

$$a_{i1}x_1 + a_{i2}x_2 = b_i - x_3$$

is the equation of a straight line that runs parallel to the ith line, and it is displaced a distance x_3 below the ith line, where $x_3 \geqslant 0$. Hence, the set (x_1, x_2) that is delimited by

$$a_{i1}x_1 + a_{i2}x_2 + x_3 \leqslant b_i \tag{3}$$

is the half-plane below the x_3-displaced ith line. The reduced polygon (3) formed by the n x_3-displaced half-planes, where $i = 1, 2, \ldots, n$ and $x_3 > 0$ and fixed, is shown in Fig. 1. In general, the reduced polygon is an m-gon, where $m \leqslant n$.

Next, we observe that the reduced polygon corresponds to the curve traced out by the center of a disc of radius x_3 inside the polygon that is rolled around the perimeter of the polygon. This then leads to the LP formulation for the determination of the radius x_3 and the center (x_1, x_2) of the largest incircle of the convex polygon (1) [where the coefficients in inequalities (1) are normalized to satisfy equation (2)]:

$$\max_{(x_1, x_2, x_3)} x_3, \tag{4}$$

subject to

$$a_{i1}x_1 + a_{i2}x_2 + x_3 \leqslant b_i \quad i = 1, 2, \ldots, n \quad x_3 \geqslant 0. \tag{5}$$

This is a linear program in $\mathbb{R}^3$. We can therefore employ the result of Megiddo [1] and Dyer [2] who give a linear-time algorithm for the solution of linear programming in $\mathbb{R}^3$. Hence, the computational effort to determine the largest incircle of the convex polygon (1) is $0(n)$. Also note that, by construction, any intermediate output of the linear programming algorithm yields a suboptimal solution—that is, we obtain the radius $x_3 > 0$ and the center (x_1, x_2) of a circle that we know is contained in the polygon.

Further insight into the problem is gained by the following geometrical observation. The feasible polyhedron (5) "grows out" of the convex polygon (1) if through each edge of the polygon (1), we pass a plane that is tilted at 45° to the horizontal plane $x_3 = 0$, where the polygon lies. Hence the polyhedron (5) is akin to a multi-slope pitched "roof" based on the convex polygon (1). The optimal point (x_1, x_i, x_2) then designates the vertex point of the "roof"—that is, the point of highest altitude x_3 above the polygonal base of the "roof". The projection of the "roof" of the polyhedron onto the x_3 plane is illustrated in Fig. 2(a–c) for a rectangle, a regular hexagon and for a rather general polygon, respectively.

The straight line segments in the polygon are thus the projections of the edges of the feasible polyhedron onto the plane x_3. They emanate from the vertices of the polygon and move toward

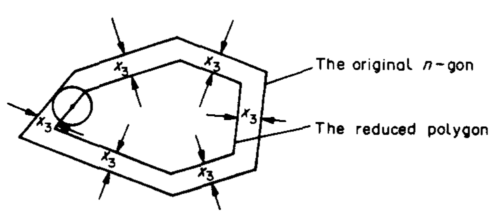

Fig. 1. The reduced polygon.

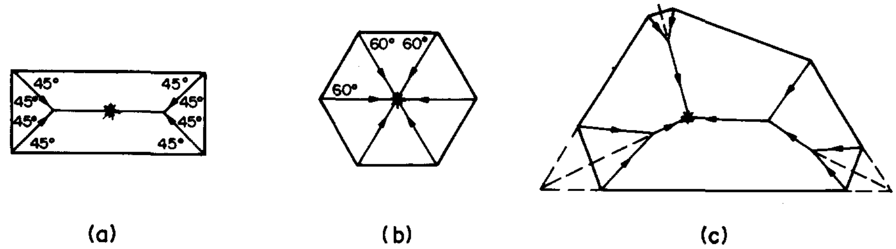

Fig. 2. The skeletons of convex polygons: (a)* center of incircle and ridge line of "roof"; (b)* (c)* center of incircle and center point of "roof".

the vertex point of the "roof", namely toward the center of the incircle. The straight line segments that emanate from the vertices are the bisectors of the internal (vertex) angles of the polygon; when two bisectors meet, they merge into a straight line segment that lies on the bisector of the angle formed by two non-adjacent edges of the polygon, as is illustrated in Fig. 2(c).

We finally remark that in Fig. 2(a) the rather exceptional case is illustrated where the center of the largest incircle of the polygon is not unique.

3. THE SKELETON OF A CONVEX POLYGON

The tree-like structure in the polygons in Fig. 2(a–c) corresponds to the skeletons of these (special) polygons. Strictly speaking, we have the following:

Definition. The skeleton of a simple polygon is the locus S of its internal points such that each point $p \in S$ is equidistant from at least two distinct points on the boundary (edges) of the polygon.

From a physical point of view, one can think of the boundary of the polygon as an initial wavefront that subsequently propagates toward the interior of the polygon; in this analogy, there will exist line segments where the wavefront "intersects itself". This locus is the skeleton S of the polygon.

In the sequel we give $0(n \log n)$–time algorithms for the efficient construction of the skeleton S of a given convex polygon. The algorithm is based on the definition of S, and it can be outlined as follows:

Input

—The list V of pairs formed by the consecutive vertices of the polygon P and their associates; initially, each vertex is its own associate.
—The list E of consecutive edges of the polygon P.
—The list T of triangles sorted according to an increasing order of the length of their altitudes; if two altitudes are equal, then the triangle with the shorter basis is the first.
—The list S of segments that form the skeleton; S is initially empty.

The performance of the algorithm consists of the following steps:

(1) If T consists of more than three triangles consider the triangle t from the bottom of the list T; otherwise go to Step (3). Add to S the segments lying on the sides of the triangle t between the top vertex of the triangle and the associates of the vertices of the base of this triangle.

(2) Update the lists V, E and T as follows:

(a) remove the triangle t from the list T;
(b) remove its base from the list E;
(c) remove two triangles, the bases of which are edges adjacent to the base of the triangle t;

(d) extend these two edges (to the point of intersection); replace these two edges by their extensions on the list E (new edges);
(e) replace, in the list P, the consecutive pairs formed by vertices of the base of the triangle t by the common point of "new edges" and appoint the top vertex of the triangle t as its associate;
(f) construct two new triangles with new edges as their bases, and bisectors of adjacent angles as their sides;
(g) insert these triangles into the list T according to their altitudes and go to Step (1).

(3) Add to S the three segments the common end of which is a top vertex of a triangle from T and the other ends are the associates of the remaining vertices.
(4) STOP.

Note. For the sake of simplicity we assume that the processed polygon is in general position, that is no two edges are parallel. To cover this case a little effort is required to extend the instruction (3) in the algorithm on the case where Step (2d) does not yield a point. This easy task is left to the reader.

The proof of the correctness of the algorithm follows from the following observation. If the triangle t (see Fig. 3) is the smallest triangle on the list T for the original polygon, then each of the segments of the bisectors of the angles of P, between P and the $(n-1)$-gon P_1, obtained by parallel translations of edges of P towards its interior by a distance equal to the altitude of t. Now, one can repeat this for the consecutive $(n-1)$-gon P_1, $(n-2)$-gon P_2 etc.; see Fig. 2. If t_1 is the smallest triangle for P_1, then the triangle t' is the smallest triangle on the updated list T after the first loop has been performed.

We also point out that the pieces of bisectors between consecutive polygons are not added gradually, but in one aggregated piece at the moment when this piece is lying on the side of the current smallest triangle.

From the above analysis one can infer that the tree S has at most as many vertices of degree $\geqslant 3$ as the number of polygons P; and hence the number of edges in S is $<2n$. Hence, and since $|T|$, $|E|$, $|P| \leqslant n$, one can infer that the space needed is of range $0(n)$.

The computational complexity of the algorithms is $0(n \log n)$. The lists V and E are virtually given. The computation of the triangles and the preparation of the list T requires $0(n \log n)$ time. The number of performed loops $\rightarrow (1) \rightarrow (2) \rightarrow (1)$ is $<n$ and each Step requires constant time except the Step (2g) which needs $0(\log n)$ time. Thus the overall complexity of the algorithm is $0(n \log n)$.

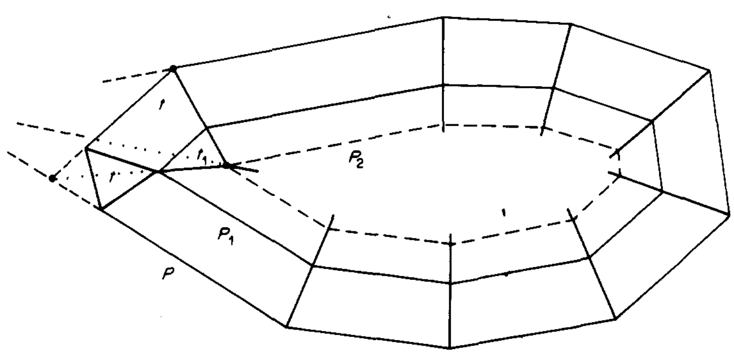

Fig. 3. Illustration to the proof of correctness of the algorithm.

4. ADDITIONAL APPROACH AND REMARK

The following idea for an alternative construction of S is based on our discussion in Section 2 concerning the polyhedron (5), namely:

Proposition. The skeleton S of the convex n-gon in the plane (x_1, x_2), which is given by the set of n-linear inequalities (1) and where the coefficients satisfy equation (2), coincides with the projection onto the plane $x_3 = 0$ of the edges of the polyhedron in $\mathbb{R}^3 = \{(x_1, x_2, x_3)\}$ that is given by the set of $n + 1$ linear inequalities (5).

Hence, in order to obtain S it is sufficient to construct the polyhedron (5). Thus, we must evoke the $0(n \log n)$ algorithm of Preparata and Muller [5] for finding the intersection of n half-spaces in three-dimensional space.

There is a certain duality relationship between the smallest enclosing circle for a set of points A [1] and the largest incircle in a convex polygon. Suppose that the polygon P is given by the equations of its m sides

$$\ell_i : a_i x + b_i y = 1 \quad i = 1, \ldots, n, \tag{6}$$

and the largest incircle S_r for P has the center at the origin of the coordinates system and the radius r. It is obvious that the distances of the lines in equation (6) from the origin are not less than r. By the transformation

$$\ell_i \rightarrow (a_i, b_i).$$

One obtains a set $A = \{(a_i, b_i);\ i = 1, \ldots, n\}$ of points, which distances from the origin are not bigger than $1/r$. Only the images of the lines that were tangent to S_r are at a $1/r$ distance from the origin so that they lie on the circle $S_{1/r}$ with the center at the origin and radius $1/r$. One can easily see that $S_{1/r}$ is the smallest enclosing circle for the set A.

The connection between the Voronoi diagram of A and the skeleton of P, as well as other related notions is the subject of our further studies.

Acknowledgements—The authors would like to thank Mr P. J. Smit of the Computer Science Division of NRIMS for presenting them with the problem of inscribing a circle in a polygon. The authors would also like to thank Dr D. H. Martin for an enlightening discussion.

REFERENCES

1. N. Megiddo, Linear time algorithms for linear programming in $\mathbb{R}^3$ and related problems. *SIAM Jl Comput.* **12**(4), (1983).
2. M. E. Dyer, Linear time algorithms for two- and three-variable linear programs. *SIAM Jl Comput.* **13**(1), (1984).
3. H. Blum, A transformation for extracting new descriptors of shape. In *Symp. on Models for the Perception of Speech and Visual Form*, Boston, Mass. (Edited by W. Wathen-Dunn). MIT Press, Cambridge, Mass. (1964).
4. V. Montanari, Continuous skeletons from digitized images. *J. Ass. Comput. Mach.*, **16**(4), (1969).
5. F. P. Preparata and D. E. Muller, Finding the intersection of n half-spaces in time $0(n \log n)$. *Theor. Comput. Sci.* **8**, 45–55 (1978).