Pergamon

Comput. & Graphics, Vol. 19, No. 4, pp. 595–600, 1995

0097-8493/95 $9.50 + 0.00

0097-8493(95)00037-2

Technical Section

ORIENTATION, SIMPLICITY, AND INCLUSION TEST FOR PLANAR POLYGONS

F. FEITO,[1] J. C. TORRES[2†] and A. UREÑA[1]

[1]Departmento Informática, Universidad de Jaén, Spain and
[2]Departmento Lenguajes y Sistemas Informáticos, E.T.S.I. Informática,
Universidad de Granada, E-18071 Spain

Abstract—This paper presents a collection of robust and simple algorithms to decide the orientation, simplicity, and inclusion of planar polygons, without solving any equation systems and without using trigonometric functions.

INTRODUCTION

Many graphic applications work with planar polygons, and most of the algorithms which carry out operations with polygons require some properties to ensure that the sequence of lines really makes a polygon.

These properties are [4]:

1. Closure: Every segment must be delimited by exactly two vertices, and every vertex must be at the intersection of exactly two segments. (Fig. 1a)
2. No self-intersection: Any two segments can intersect only if they are adjacent, in this case the intersection point is the shared vertex (Fig. 1b).
3. Orientation: Every segment must have a direction, and the directions of all the segments must be consistent (Fig. 1c).

Many algorithms have been proposed to test these conditions. Most of them solve these problems by working out equation systems, which implies problems of stability [6]. Balbes and Siegel have proposed, to avoid the resolution of equation systems, some algorithms based on the computation of the winding number [1]. These algorithms use trigonometric functions which result in two important drawbacks: the rise of computation errors, and a loss of efficiency.

This paper presents a formal definition of some robust and simple algorithms to decide the orientation and simplicity of polygons, and a new inclusion algorithm, which do not require any equation systems or the use of trigonometric functions.

The following section introduces some basic definitions and theorems, many of which can be found in the text on geometry [4]. The third section examines the simplicity problem, the fourth solves the orientation problem, and the last one studies the inclusion problem.

† Author for correspondence.

BASIC CONCEPTS

This section introduces some basic definitions and concepts.

Definition 1. (Polyline, Polygon). A *polyline* is a sequence of points $A, B, C, D, \ldots, K, L$. The polyline can be identified using the points sequence $ABCD \ldots KL$. A *polygon* is a polyline for which the two extreme points coincide. The polygon can be identified using the points sequence $ABCD \ldots KL$. The polygon segments, $AB, BC, CD, \ldots, LA$, are called *edges* and the polygon points, $A, B, C, \ldots, L$, are called *vertices* [4].

Definition 2. (Simple polygon). A polygon, $P_1 P_2 \ldots P_n$, which holds the following properties:

- All the vertices are different: $\forall i \neq j \;\; P_i \neq P_j$
- Any vertex is over an edge
- There is no intersection between any two edges

is called *simple* [4].

A segment AB splits the plane into two regions, one on the right of the vector AB, and the second on the left.

Definition 3. (Orientation of a triangle). A triangle, ABC, has positive orientation if its interior points yield to the left of all its edges, in the opposite case the triangle has negative orientaton [4].

Definition 4. (Area of a triangle). The area of any triangle, ABC, is half of the product of the length of any edge multiplied by its distance to the opposite vertex; the area of the triangle is denoted as $[ABC]$.

Definition 5. (Signed area of a triangle). The signed area of a triangle, ABC, is denoted by $[ABC]$, and is computed as its area $[ABC]$ if it has positive orientation, and as the opposite of its area, $-[ABC]$, when it has negative orientation [4].

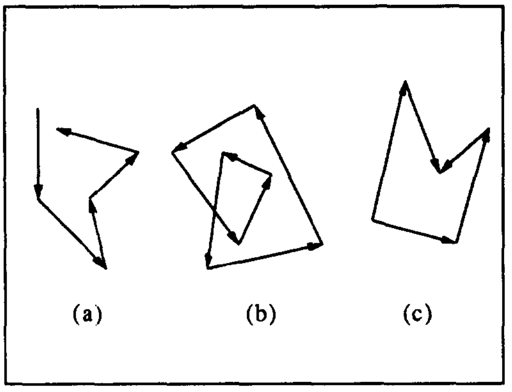

Fig. 1. Some not correct polygons. (a) Not closed. (b) Auto-intersected. (c) Not well oriented.

Note that the sign of the area of a triangle, sign$[ABC]$, determines the orientation of the triangle, and that for any given triangle $[CBA] = -[ABC]$.

For any given triangle ABC and any given point O, the following condition holds [4]:

$$[ABC] = [OAB] + [OBC] + [OCA].$$

For any given triangle, $P = P_1P_2P_3$, its signed area can be computed as [6]:

$$[P] = \frac{1}{2}\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}.$$

The orientation of a polygon depends on the order in which its vertices are arranged. A polygon may be oriented in clockwise or counter clockwise order. The order of a polygon is important because it may be used to establish certain properties, such as the internal and external sides of a polyhedron face.

Usually the orientation is mathematically defined by means of the winding number of an interior point [3]. The winding number of a point, Q, in respect to a polygon, P, is defined using the angle between the lines of every consecutive polygon vertex at a given point, denoted by $\{P_i, Q, P_{i+1}\}$.

Definition 6. (Winding number). For any given polygon, $P = P_1P_2, \ldots P_n$, and any given point Q, the winding number of Q with respect to the polygon, P, is defined as [3]:

$$w(P, Q) = \sum_{i=1}^{n} \{P_i, Q, P_{i\oplus 1}\} \quad (\text{where } n \oplus 1 = 1).$$

The winding number of a polygon, P, for an interior point, Q, is $W(P, Q) = -2^*PI$ if P is oriented clockwise, and $w(P, Q) = 2^*PI$ if P is oriented counter clockwise (for any external point $w(P, Q) = 0$).

Theorem 1. (Orientation of a polygon). Any polygon, P, is oriented clockwise, or negatively, if for any interior point, Q, $W(P, Q) = -2^*PI$, and is oriented counter clockwise, or positively, if $w(P, Q) = 2^*PI$ [5].

Definition 7. (Area of a polygon). The area of any simple polygon, P, is the addition of the areas of any set of triangles in which it can be decomposed, and is denoted by $[P]$.

Definition 8. (Signed area of a polygon). The signed area of a polygon, P, is denoted as $[P]$, and is computed as its area for a positively oriented polygon, and as minus for a negatively oriented polygon.

SIMPLICITY

Many graphic algorithms work only with simple polygons, thus graphic systems need to test the simplicity of the polygons defined. Here we present an effortless algorithm to test polygon simplicity.

Let $P = P_1P_2 \ldots P_n$ be a polygon given with positive orientation, where every P_i is a 2D point $P_i = (x_i, y_i)$, $i = 1, \ldots n$. According to Definition 2, the polygon is simple if it holds that:

- All the vertices are different:

$$\forall i \neq j \quad P_i \neq P_j.$$

- Any vertex is at an edge (see Fig. 2a).
- There is no intersection between any two edges (see Fig. 2b).

Let us study each condition independently. The first one can be easily tested by comparing each pair of vertices, that is:

$$\forall i, j \in [1, n] \subset \mathbb{N}, i \neq j => x_i \neq x_j \vee y_i \neq y_j.$$

To test the second condition it is necessary to see if any vertex is inside an edge (except its own edges), which can be carried out by checking that:

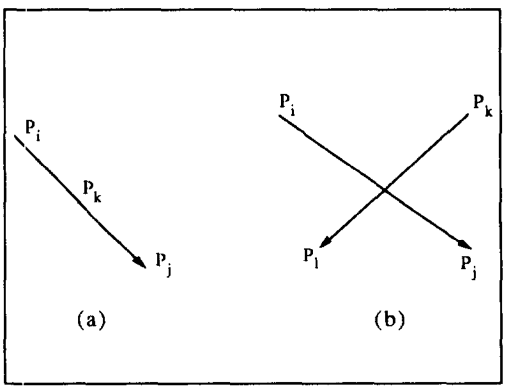

Fig. 2. Abnormal situations for a polygon.

$\forall i, k \in [1, n] \subset \mathbb{N}, i \neq k, i \oplus 1 \neq k, j = i \oplus 1$

$$=> (x_j - x_k)/(x_i - x_k) \neq (y_j - y_k)/(y_i - y_k).$$

which implies that the line (i, k) is not parallel to the edge $(i, i \oplus 1)$ (notice that the indices are used in a circular way, that is $n \oplus 1 = 1$). When these two terms are equal, it is necessary to verify that the vertex k is outside the edge, that is, the slope of the lines (i, k) and (j, k) must be the same:

$$\mathrm{sign}\big((x_i - x_k)/(y_i - y_k)\big) = \mathrm{sign}\big((x_j - x_k)/(y_j - y_k)\big).$$

Where sign(x) if the sign function defined as

$$\mathrm{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}.$$

Notice that when one part of the partitions is zero the condition only has to be tested with the other part.

The third condition is usually checked by solving linear equation systems: for every pair of edges the intersection point is computed by matching the line equations of the two edges, and then a test is carried out to see if the intersection point is inside one of the edges. This method has several drawbacks caused by computational errors.

We will now present, in a formal way, a previously known robust algorithm to test this condition. The algorithm does not compute the intersection point. We will assume that the polygons to be tested have positive orientation.

Theorem 2. The necessary and sufficient conditions for any two segments, P_iP_j and P_kP_l, to intersect is:

$$\mathrm{sign}(P_kP_iP_j) \neq \mathrm{sign}(P_lP_iP_j) \wedge$$
$$\mathrm{sign}(P_iP_kP_l) \neq \mathrm{sign}(P_jP_kP_l).$$

Proof: Let us build the triangles $P_kP_iP_j$ and $P_lP_iP_j$, their difference in sign means that they have different orientations, and thus p_k is on one side of the line defined by P_iP_j and P_l is on the other. This implies that the line defined by P_iP_j and the segment P_lP_k have an intersection point, but this intersection point may be outside the segment P_iP_j. In order to ensure that the intersection point is inside this segment we need P_i and P_j to be at different sides of the line defined by P_kP_l, that is $P_iP_kP_l$ and $P_jP_kP_l$ must have different signs.

Notice that the sign of the triangle can be easily computed solving a determinant.

Previously, Balbes and Siegel presented an algorithm which did not compute the intersection point [1], based on the computation of the winding number; this implies the use of trigonometric functions.

ORIENTATION OF POLYGONS

Usually the orientation is mathematically defined by means of the winding number of an interior point; the main drawback of this method is that the winding number is computed using trigonometric functions, which requires lot of extra computation. We will formalize a method, already known, to compute the orientation of a simple polygon based on the triangle decomposition which requires only the computation of additions and products [2] (see Fig. 3).

Theorem 3. (Computation of the signed area of a polygon). The signed area of a simple polygon, $P = P_1P_2 \dots P_n$ where $P_i = (x_i, y_i)$ $i = 1, \dots n$, can be computed using the expression:

$$A(P) = \sum_{i=1}^{n} (x_i * y_{i\oplus 1} - x_{i\oplus 1} * y_i).$$

Proof. According to Definition 7 the polygon must be split into triangles in order to compute its area. Suppose that the polygon has been decomposed into a set of triangles, $\{T_i | j = 1 \dots m\}$. Then

$$|P| = \sum_{i=1}^{m} |T_i|.$$

It is clear that this equation also holds for the signed area of the polygon if the triangles have the same orientation as the polygon. And so, assuming that T_i has the orientation of the polygon

$$[P] = \sum_{i=1}^{m} [T_i].$$

The signed area of every triangle, $T_j = P_{1j}P_{2j}P_{3j}$, may be computed as:

$$[T_j] = [OP_{1j}P_{2j}] + [OP_{2j}P_{3j}] + [OP_{3j}P_{1j}].$$

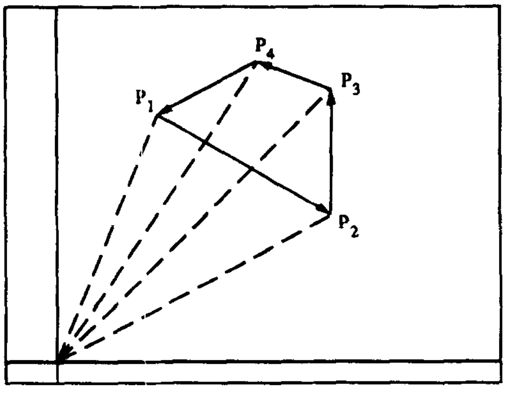

Fig. 3. Computation of the signed area of a polygon.

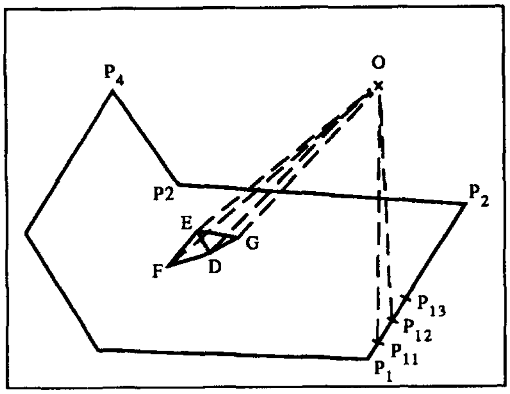

Fig. 4. Decomposition of a polygon into triangles.

Let us now look at the piece of decomposition in Fig. 4. The triangles DEF and GED are part of the decomposition of the polygon, its areas are computed using the previous decomposition. That is

$$[DEF] = [ODE] + [OEF] + [OFD]$$

$$[GED] = [OGE] + [OED] + [ODG].$$

So, in the computation of the signed area of the triangle we will have the term $[ODE]$ and $[OED]$, whose addition is obviously zero.

For the same reason all the terms corresponding to interior edges of any triangle will be neutralized, and so the only effective terms in the computation of the signed area of the polygon will be those corresponding to triangle edges on the polygon boundary (see Fig. 4).

Let $P_{k1}, P_{k2}, \ldots, P_{kl}$ be the sequence of triangle vertices located on the polygon edges $P_k P_{k+1}$, then their contribution to the polygon signed area will be

$$A_k = [OP_{k1}P_{k2}] + [OP_{k2}P_{k3}] + \cdots + [OP_{kl-1}P_{kl}].$$

Which is obviously equal to $[OP_1P_2]$, and therefore the polygon signed area will be equal to

$$A(P) = \sum_{i=1}^{n} (x_i * y_{i\oplus 1} - x_{i\oplus 1} * y_i).$$

This theorem can also be easily probed by using the trapezoidal rule [6].

Collorary 1. Any simple polygon, P, is oriented counterclockwise if, and only if, the expression

$$A(P) = \sum_{i=1}^{n} (x_i * y_{i\oplus 1} - x_{i\oplus 1} * y_i).$$

is positive. The polygon is oriented clockwise if, and only if, this expression is negative.

INCLUSION TEST

Several algorithms have been proposed to carry out the inclusion test [2], some of them for specific kinds of polygons (star, convex, *etc.*).

One method frequently used is to count the number of intersections between a semiinfinite line starting at Q with the polygon edges. This number is odd for any interior point, and even when the point is exterior. This method implies the resolution of an equation system, which may have stability problems.

As we have previously said, another possible method to decide whether a point Q is interior to a polygon P is to compute its winding number $W(P, Q)$; this number is $2 * PI$ for any interior point and zero when the point is exterior [3]. This method has the drawback of having to work with trigonometric functions.

The results presented above can be used to decide the inclusion of the point in the polygon in two different ways. The first method can be slightly modified using a line segment delimited by Q at some point external to the polygon instead of a semiinfinite line. Such a point can always be found, for example the point $(\max(x_i) + 1, \max(y_i + 1))$ can be used ((x_i, y_i) being the vertices of the polygon). This will allow us to use Theorem 2 to find out the number of intersections, this avoids the computation of the coordinates of the intersection point, and so the stability problems posed above are omitted.

Another possibility is to use the underlined ideas of triangle decomposition used in this paper. That is, if we decompose the polygon into triangles, in seeing that the point belongs to the triangles, we can decide whether the point belongs to the polygon.

Lemma 1. Let $P = ABC$ be a triangle with positive orientation, and Q a point. Then Q is interior to the triangle if, and only if:

$$\text{sign}([QAB]) \geq 0 \quad \text{and} \quad \text{sign}([QBC]) \geq 0$$

$$\text{and} \quad \text{sign}([QCA]) \geq 0.$$

Proof. Q is an interior point if, and only if, it yields to the left of the segments AB, BC and CA, which implies that $[QAB]$, $[QBC]$ and $[QCA]$ are positive (see Fig. 5). When some of these signed areas are negative, the point yields to the right of the segment, and so it is outside the polygon. When some area is zero then the point is on the segment.

Theorem 4. Let $P = P_1P_2 \ldots P_n$ be a positive simple polygon and Q a point. Let us define the sorted sets $s(Q)$ as follows

$$s(Q) = \{(j_1, \ldots, j_i, \ldots, j_n) | j_i = \text{sign}([OP_iP_{i\oplus 1}])$$

$$\text{if} \quad Q \in OP_iP_{i\oplus 1} \quad \text{and} \quad j_i = 0\}.$$

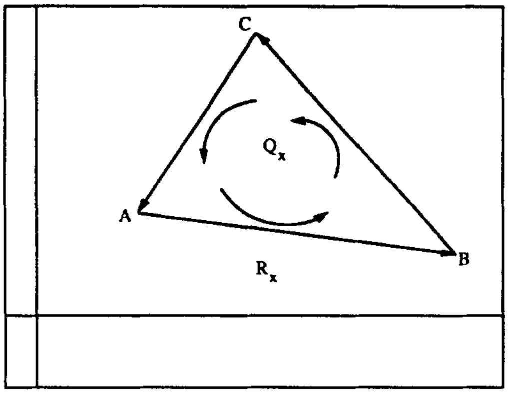

Fig. 5. Interior point of a triangle.

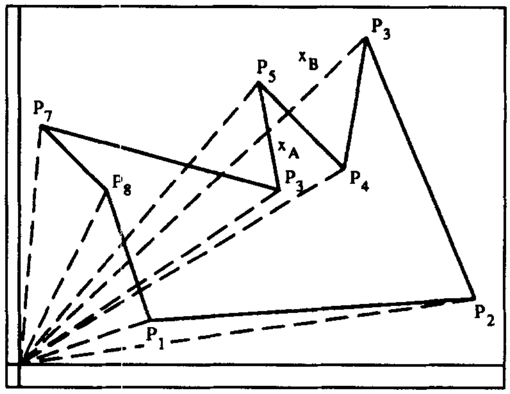

Fig. 6. Decomposition of a polygon into original triangles.

The set $s'(Q)$ is defined as

$$s'(Q) = \{(k_1, \dots, k_i, \dots, k_n) | k_i = j_i$$

$$\text{if} \quad j_i <> j_{i\oplus 1} \quad \text{otherwise} \quad k_i = 0\}.$$

where $1 \ominus 1 = n$. Then $Q \in P$ if and only if

$$\sum_{i=1}^{n} k_i > 0.$$

Proof. Let us study the term of the $s(Q)$ sequence. Every j_i is not zero when the point Q is on the triangle OP_iP_{i+1}; in this case j_i would be either $+1$ or -1 according to the orientation of the triangle OP_iP_{i+1}. It is important to note that, as the polygon has positive orientation, the orientation of the triangle OP_iP_{i+1} will be positive for those triangles overlapping the polygons (consider that only in this case the edge P_iP_{i+1} appears at the same side for the polygon and the triangle).

With these considerations it is easy to see that the sequence $s(Q)$ will contain a $+1$ for every triangle which overlaps the polygon and which contains the point Q, and a -1 for every triangle which does not overlap the polygon and which contains the point Q.

The point Q is inside the polygon if any line starting at Q has an odd number of intersection with the polygon edges. Point A in Fig. 6 is inside the polygon. Let us build the line starting at the origin and containing Q. For every positive triangle OP_iP_{i+1} containing Q this line will have an intersection with the edge P_iP_{i+1}, which implies necessarily that the line advances from the inside to the outside of the polygon. In the same way, when the triangle is negative the line advances from the outside to the inside of the polygon. So, counting intersections is the same as counting the sign of each triangle, with the only problem of avoiding to passing through any vertex.

The difference between the sequences $s(Q)$ and $s'(Q)$ is that we have changed every subsequence of consecutive $+1$ or -1 by only one $+1$ or -1 and zeros. A pair of consecutive $+1$ necessarily implies that the point Q is the common edge of two consecutive positive triangles, and so the subordinate line would have an intersection with a vertex.

EFFICIENCY

The computational complexity of the algorithms proposed is the same as other algorithms previously proposed ($O(n)$ for the inclusion test and $O(n)$ [1, 3, 5] for the simplicity test). The computational cost for a given problem has been satisfactorily reduced in respect to [1, 5]. The reason for this reduction is that we do not use trigonometric functions, which must be computed by algorithms based on the winding number evaluation.

In order to confirm this assertion we have implemented the inclusion test algorithm and the orientation algorithm, comparing the runtimes with previous algorithms. The inclusion test has been compared with a well known algorithm [5], and the orientation test has been compared with the algorithm of Balbes and Siegel [3].

The algorithms have been used to test 1000 simple polygons previously generated on a file. The polygons have between 6 and 15 edges, and have been generated using our simplicity test (based on Theorem 2). All the agorithms have been implemented using *C* and have been run on a 386 computer with a mathematical coprocessor.

Table 1. Inclusion test

N	Theorem 4	Kalay
6	1651.33	3151.89
7	1971.60	3764.59
8	2295.22	4257,.9
9	2626.26	4993.71
10	2966,.0	5581,.4
11	3280.91	6180.27
12	3608.71	6663,.7
13	3945.90	7148.55
14	4269.35	8065.64
15	4581.88	8672.35

Table 2. Orientation test

N	Corollorary 1	Winding n
6	239.30	3758.00
7	281.71	4356.59
8	325.40	4868.69
9	368.22	5585.37
10	411.60	6174.22
11	454.12	6782.63
12	496.52	7273.78
13	539.80	7698.13
14	581.99	8658.39
15	625.93	9287.79

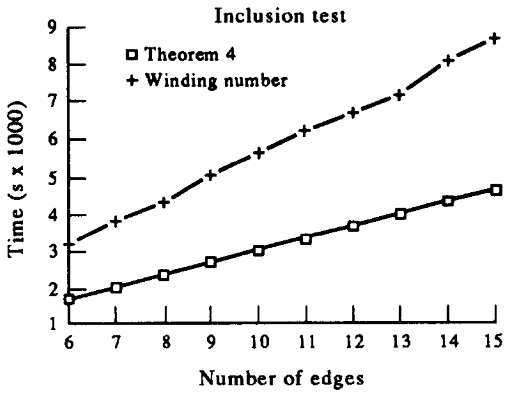

Fig. 7. Comparing runtimes for the inclusion test.

The test has shown a reduction of 50% in time for the inclusion test (see Table 1 and Fig. 7), and a reduction of 85% for the orientation test (see Table 2 and Fig. 8).

The comparison with "exclusive or method" is more complicated [3]. The "exclusive or method" can be implemented with a preprocessing stage that precomputes the equation of each edge. When this preprocessing is performed the operations required of every edge are two multiplications, two additions and one comparison. This would not be the normal situation for a real application, because it implies a big extra memory occupancy: it needs to store three extra real parameters for every edge.

When the preprocessing stage is not performed the "exclusive or method" requires the computation of the sign of the point of study with respect to every

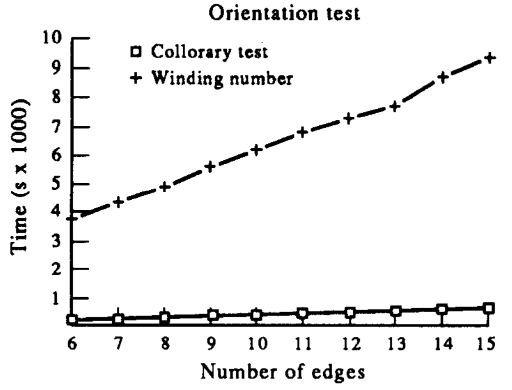

Fig. 8. Comparing runtimes for the orientation text.

edge, which may imply the computation of six multiplications and two additions. Our algorithm needs, for every edge, to perform four products, two additions and two comparations.

CONCLUSION

This paper has presented a collection of robust and simple algorithms to calculate the orientation, simplicity and inclusion for planar polygons, without solving any equation systems and without using trigonometric functions. The algorithms proposed have been shown to be more efficient than previous algorithms.

Acknowledgements—We want to express our thanks to referees for his comments and suggestions.

REFERENCES

1. R. Balbes and J. Siegel, A robust method for calculating the simplicity and orientation of planar polygons. *Comp. Aided Geom. Design* **8**, 327–335 (1991).
2. J. Foley, D. A. Vam, S. Feiner and J. Hughes, *Computer Graphics: Principles and Practice* (2nd Ed.), Addison Wesley, Reading, MA (1990).
3. L. Guibas, L. Ranshaw and J. Stolfi, A kinitec framework for computatinal geometry. In *24th Annual Symp. Foundations of Computer Science*, IEEE, New York, 100–111 (1983).
4. D. Hilbert, *Grundlagen der Geometrie* (7th Ed.), Teubner, Leipzig-Berlin (1930).
5. Y. E. Kalay, *Modelling Objects and Environments.* Wiley, New York (1989).
6. F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction.* Springer-Verlag, New York (1985).