RECOGNIZING POLYGONS, OR HOW TO SPY

by James A. Dean, Andrzej Lingas
and Jörg-R. Sack

SCS-TR-117

August 1987

School of Computer Science
Carleton University
Ottawa, Ontario
CANADA K1S 5B6

This research was supported in part by NSERC.

Recognizing Polygons, or How to Spy+

James A. Dean[1], Andrzej Lingas[2] and Jörg-R. Sack[3]

[1] Bell Northern Research, Ottawa, Canada; research for this paper was done while the author was at Carleton University.

[2] Department of Computer and Information Science, University of Linköping, Linköping, Sweden; Research for this paper was done in part while the author was visiting Carleton University.

[3] School of Computer Science, Carleton University, Ottawa, Canada. This research was supported in part by NSERC under grant No. A0392 and by Carleton University.

+ A preliminary version of this paper has been presented at the 24th Allerton Conference on Communication, Control and Computing, Monticello, Ill. Oct. 1986.

Key words: Computational geometry - Geometric algorithms - Complexity - Polygon recognition - Star-shapedness

Abstract

A new class of so-called pseudo-star-shaped polygons is introduced. A polygon is pseudo-star-shaped if there exists a point from which the whole interior of the polygon can be seen, provided it is possible to see through single edges. We show that the class of pseudo-star-shaped polygons unifies and generalizes the well-known classes of convex, monotone and pseudo-star-shaped polygons. We give algorithms for testing whether a polygon is pseudo-star-shaped from a given point in linear time, and for constructing all regions from which the polygon is pseudo-star-shaped in quadratic time. We show the latter algorithm to be worst-case optimal. Also, we give efficient algorithms solving standard geometrical problems such as point-location and triangulation for pseudo-star-shaped polygons.

1 Introduction

For the task of spying on an apartment whose floor-plan is given as a simple polygon P, we are equiped with an infra-red camera and a microphone. The equipment is strong enough to receive signals which have penetrated through at most one wall. The task is to find a possible location for the equipment such that we can pick-up signals originating anywhere inside the apartment. If such a location exists then P is called *pseudo-star-shaped* + (from this location) and the set of all such locations is referred to as the *pseudo-kernel* for P.

Two geometrical problems naturally arise and their solutions will be discussed in Section 3:

(1) Given a specific location u, is P pseudo-star-shaped from u?

(2) Construct the pseudo-kernel for P, i.e. find all loci (regions) from which P is pseudo-star-shaped.

While the first problem can easily be solved in linear time, the second problem is harder. It requires quadratic time, in the number of vertices of P, in the worst case, as will be shown in Section 2. Thus our quadratic-time solution to this problem, as presented in Section 3, is worst-case optimal.

While any convex, star-shaped, or monotone polygon can be shown to be pseudo-star-shaped, there are pseudo-star-shaped polygons which do not fall into any of these classes. Through the notion of pseudo-kernel the common polygon classes of convex, monotone and star-shaped polygons can be characterized (see Section 2). A unifiying and generalizing view of the three notions of convexity, star-shapedness and monotonicity is thus provided.

In the literature, many efficient algorithms solving geometrical problems for the common polygons classes are known, e.g.

(a) triangulation and point location (computational geometry),

(b) hidden-line elimination (computer graphics), and

(c) motion planning (robotics).

As examplified in Section 4 on problem-set (a), efficient algorithms designed originally for the common polygon classes, can often be easily adapted to work for the more general class of pseudo-star-shaped polygons.

+ Keil (1985) considers a very special class of pseudo-star-shaped polygons arising from the problem of decomposing simple polygons into the minimum number of star-shaped components.

2 Pseudo-star-shaped polygons and their relationship to common polygon classes

In this section, we introduce some notation and then establish the relationship between the class of pseudo-star-shaped polygons and the standard classes of convex, star-shaped and monotone polygons.

2.1 Definitions

It is assumed that all polygons P are specified by a list of their vertices $(p_0,\ldots,p_{n-1})$ given in counter clockwise order around the boundary of P. This induces an orientation on the edges, $e_i=p_{i-1}p_i$, of P such that the interior of P lies always to the left of e_i. Futhermore, we assume that P is simple, i.e. non-self intersecting, and given in standard form (Preparata and Shamos 1985). Operations on indices are taken modulo n, such that e.g. $p_n=p_0$. Two points u,v are mutually *visible* if the open line-segment joining them does not (properly) intersect the boundary of the polygon. We allow points, u, to be located at infinity, in which case a unique direction is associated with u and the line-segment (half-line) connecting u to any point v in P is parallel to that direction.

We can now give a formal definition of pseudo-star-shapedness: A polygon P is *pseudo-star-shaped* if there exists a point u such that for each point v in P the open line-segment (half-line) connecting u to v properly intersects the boundary of P at most once.

A polygon is *star-shaped* if there exists a point in P which can see each point in P. The set of all such points is called the *kernel* of P. A polygon is *convex* if for i=0,...,n-1 the interior angle p_{i-1},p_i,p_{i+1} at vertex p_i is less than 180^o.

A *polygonal chain* C_{ij} of P is the list of its vertices $(p_i,\ldots, p_j)$, for i=j we define C_{ii} to be $(p_i,p_{i+1},\ldots,p_{i-1},p_i)$. A polygonal chain C_{ij} is *monotone with respect to a line d* if the perpendicular projections of the vertices p_k, k=i,...,j, on d are in the same order as the vertices on C_{ij} (here we assume that overlapping projections obey the polygonal order). A polygonal chain is *completely visible from a point u* if each point on the chain is visible from u.

Let H[u,x] denote the directed straight-line originating at u and passing through point x (where u may possibly be located at infinity). If u is on the plane then H[u,x] is a half-line. A vertex p_i of P is called a *(generalized) support vertex* of P with respect to u if no point in P except boundary points are located on $H[u,p_i]$. This definition of support vertex is more general than that given in (Preparata and Shamos 1985), since the point u is not restricted to be exterior to the convex hull of P.

2.2 Characterizations of pseudo-star-shaped polygons

In the following lemma we give alternate characterizations of pseudo-star-shapedness which contribute to a further understanding of the structure of this polygon class and will be of importance in the subsequent sections.

Lemma 2.1 *Let $P=(p_0,\dots,p_{n-1})$ be a simple polygon and u be a point located either on the plane, or at infinity. Then the following conditions are equivalent:*

(1) P is pseudo-star-shaped from u.

(2) There exists an index-pair (i,j) such that both polygons $P_{ij} = (p_i,\dots, p_j, u)$ and $P_{ji} = (p_j,\dots, p_i, u)$ are star-shaped from u, if u is on the plane, and both chains $C_{ij} = (p_i,\dots, p_j)$ and $C_{ji} = (p_j,\dots, p_i)$ are monotone with respect to the line perpendicular to the direction of u, if u is at infinity.

(3) The boundary of P can be partitioned into two chains C_{ij} and C_{ji} such that C_{ji} is completely visible from u, even in the presence of C_{ij}, and C_{ij} is completely visible from u in the absence of C_{ji}.

Proof "(1) => (2)" If u is inside P then any open line-segment connecting u to any point v inside P intersects the boundary of P an even number of times. Since P is pseudo-star-shaped from u, the number of intersections is always zero. Hence, P is a star-shaped polygon, in that case, and the second condition trivially follows. In case that the points u,v and some vertex p_i of P are co-linear, we count the number of intersections in the standard manner, see e.g. (Preparata and Shamos 1985). If u is outside P, or located on the boundary of P, then the support vertices p_i, p_j for u and P partition the boundary into the two chains C_{ij} and C_{ji}. Since u sees both p_i and p_j without properly crossing the boundary of P, it completely sees each of the two chains if we remove the other. Otherwise, an open segment (half-ray) connecting u with some point on one of the two chains would cross the chain twice. We conclude that the polygons $P_{ij} = (p_i,\dots,p_j,u)$ and $P_{ji} = (p_j,\dots,p_i,u)$ are star-shaped from u, if u is on the plane and otherwise, the chains C_{ij}, C_{ji} are monotone to the direction perpendicular to u.

"(2) => (3)" The two lines $H[u,p_j]$ and $H[u,p_i]$ partition the plane into two (three) infinite regions. From the Jordan Curve Theorem (discussed e.g. in Preparata and Shamos 1985), it follows that since both P_{ij} and P_{ji} are star-shaped from u (or C_{ij} and C_{ji} are monotone to the line perpendicular to u, if u is at infinity), neither of the two chains C_{ij} and C_{ji} can be located in two different regions.

(a) The chains C_{ij} and C_{ji} are in different regions.

In this case, u is inside P and clearly the visibility of one chain is unchanged by the presence of the other.

(b) Both chains are in the same region.

In this case, since both P_{ij} and P_{ji} do not intersect properly, one polygonal chain, say C_{ij}, will be completely in front of the other, i.e. any straight-line originating at u and intersecting the chain C_{ji} at a point v has an intersection point on C_{ij} which is closer to u than v. Thus, in this case, C_{ij} is completely visible from u even in the presence of the other chain.

"(3) => (1)" It is sufficient to observe that any straight-line starting at u can intersect each of the chains at most once.
q.e.d.

Let P, C_{ij}, C_{ji} and u satisfy the third condition of Lemma 2.1. Then, the vertices p_i and p_j are called the *chain-defining vertices for point u.*

Lemma 2.2 *Let P be a pseudo-star-shaped polygon from a point u and p_i, p_j be the two chain-defining vertices for u then the vertices of each chain C_{ij} and C_{ji} are sorted by polar angle with u as origin if u is on the plane, or sorted along the line perpendicular to the direction associated with u, if u is at infinity.*
Proof Follows from Lemma 2.1 (2). q.e.d.

Lemma 2.3 *For any point u outside a polygon P, every pair of chain-defining vertices for u is a pair of support vertices with respect to u.*
Proof Assume that p_i is a chain-defining vertex but is not a support vertex for P. Then H[u, p_i] intersects at least one of the chains at a point different from p_i. Such a chain cannot be completely visible from u which violates Lemma 2.1 (3). q.e.d.

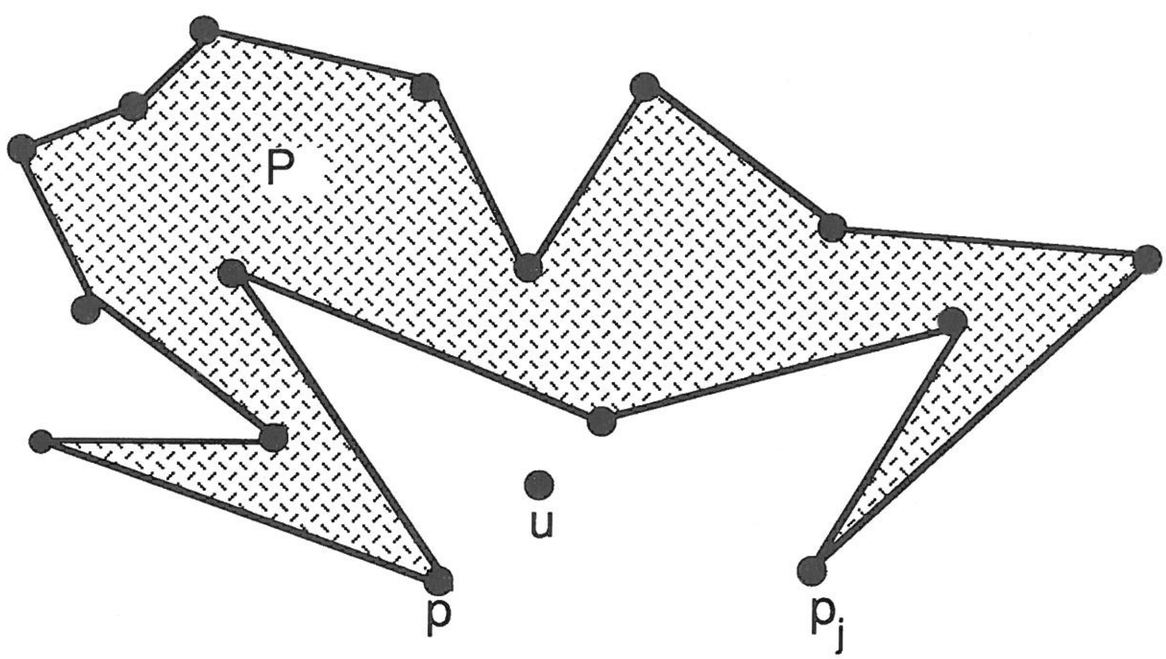

Fig. 1. A polygon pseudo-star-shaped from a point u.

2.3 The characterization of convex and star-shaped polygons through pseudo-star-shapedness

Whereas the kernel of a star-shaped, n-vertex polygon is a bounded, convex polygon with O(n) vertices located inside P (see e.g. Lee and Preparata 1985), the pseudo-kernel of a pseudo-star-shaped polygon may be unbounded, disconnected, possibly partially inside and partially outside P, and having $\Omega(n^2)$ vertices, as will be shown in Section 3.

It is, however, possible to relate star-shapedness to pseudo-star-shapedness by examining the portion of the pseudo-kernel located inside P called the *inner pseudo-kernel* of P. The remaining portions of the pseudo-kernel are referred to as the *outer pseudo-kernel*. The following characterization of star-shapedness through pseudo-star-shapedness becomes now quite clear:

Theorem 2.4 *For any polygon P, kernel(P) = inner pseudo-kernel(P) holds.*
Proof For any u in kernel(P) the second condition of Lemma 2.1 is satisfied by taking e.g. i=j=1. Conversely, if u is in inner pseudo-kernel(P) then there are two indices i,j such that the polygons $P_{ij} = (p_i,\ldots,p_j,u)$ and $P_{ji} = (p_j,\ldots,p_i,u)$ are star-shaped from u, again by Lemma 2.1 (2). Now, it is sufficient to observe that the two polygons form a partition of P since u is inside P. Therefore P is also star-shaped from u. q.e.d.

Corollary 2.5 *A polygon is star-shaped iff its inner pseudo-kernel is non-empty.*

We will now establish the relationship between convexity and pseudo-star -shapedness.

Theorem 2.6 *A polygon P is pseudo-star-shaped from all points (the entire plane including all points at infinity) if and only if P is convex.*
Proof "<=" Since the kernel of a convex polygon P is equal to P (Preparata and Shamos 1985), all points in P are in the inner pseudo-kernel (by Theorem 2.4). Given a point u outside P, the support vertices, say p_i, p_j, for P with respect to u, divide the boundary of P into two chains C_{ij}, C_{ji} satisfying the third condition of Lemma 2.1. Thus P is pseudo-star-shaped from all points.
"=>" If P had a concave vertex p_i, then any straight-line properly intersecting the two edges of P incident to p_i would cross the boundary of P at least four times. We obtain a contradiction since P could not be pseudo-star-shaped from any point on that line outside P. q.e.d.

2.4 The characterization of polygon monotonicity through pseudo-star-shapedness

In this section, we define polygon monotonicity and then relate it to pseudo-star-shapedness.
A polygon is *monotone with respect to a line d* if there exists a pair of vertices p_i, p_j such the resulting chains C_{ij} and C_{ji} are each monotone with respect to d.

A polygon P is *monotone* if there exists a line d with respect to which P is monotone.

To see the relation between pseudo-star-shapedness and monotonicity, we examine the unbounded regions of the pseudo-kernels. An unbounded region of the pseudo-kernel may

(a) contain all points at infinity and all points of the plane,
(b) contain some points at infinity and some on the plane, or
(c) contain some points at infinity, but no point on the plane.

In case (a), using the Theorem 2.6, P is convex and is therefore monotone with respect to all directions. An example for (c) is given in Figure 2. In case (b), each unbounded region of the pseudo-kernel may be described by some polygonal chain and two half-lines say e_a and e_b. Lines perpendicular to these half-lines define directions of monotonicity.Thus, unbounded kernel-loci define directions of monotonicity (Figure 3 gives an example). This situation is further examined in Theorem 2.7.

Theorem 2.7 *A polygon P is pseudo-star-shaped from u at infinity iff P is monotone with respect to the line perpendicular to the direction associated with u.*

Proof Follows immediately from Lemma 2.1. q.e.d.

Preparata and Supowit (1981) observed that an n-vertex polygon may be monotone to as many as $\Omega(n)$ disjoint direction-regions and thus there may exist at least $\Omega(n)$ unbounded disjoint outer pseudo-kernels (see also Fig. 8).

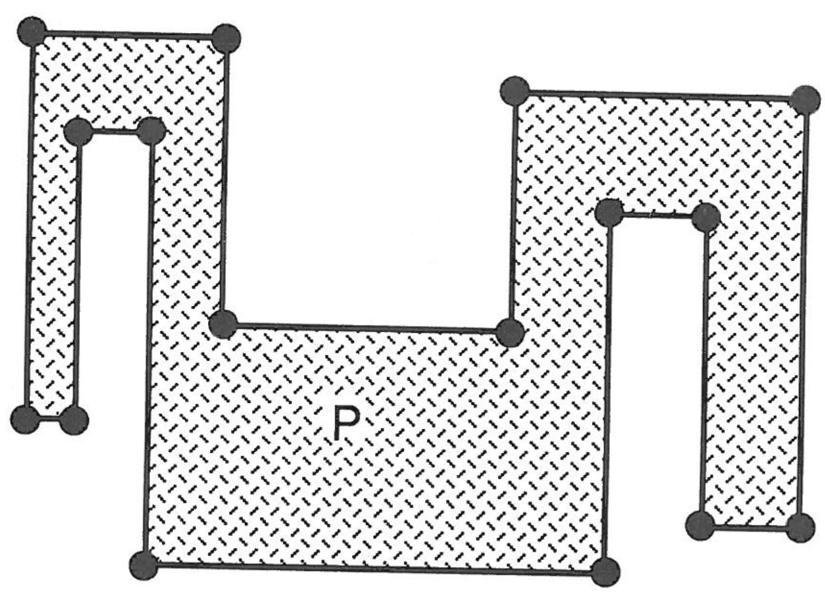

Fig. 2. A pseudo-star-shaped polygon P whose (unbounded) pseudo-kernel contains only two points, located at + and - infinity on the y-axis.

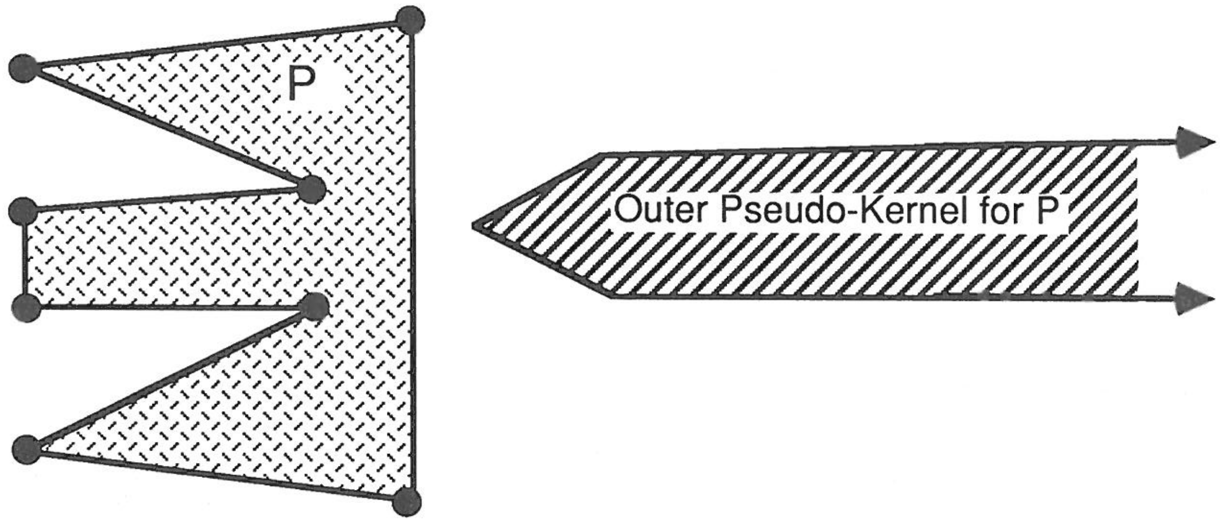

Fig. 3. A pseudo-star-shaped polygon with one of its two unbounded outer pseudo-kernels.

2.5 Summary of polygon-class relationships

By introducing the notion of pseudo-kernel of a polygon, convex, star-shaped and monotone polygons become just special instances of pseudo-star-shaped polygons. We can imagine the point u from which a pseudo-star-shaped polygon is pseudo-star-shaped to travel from u inside P (in the case of convex or of star-shaped polygons) through the portion of the plane exterior to P to reach infinity (in case of monotone polygons). Notice, here, that this constitutes a continuous transition between the perspective visibility (point u on the plane) and parallel visibility (point u at infinity) (for a discussion of these models see e.g. Newman and Sproull 1978). The known relationships among the four polygon classes are summarized below. The reader is referred to (Toussaint 1985) for a discussion of a hierarchy of additional polygon classes.

Any pseudo-star-shaped polygon that is neither monotone, nor convex, nor star-shaped is called *truely pseudo-star-shaped.* The following theorem describes the truely pseudo-star-shaped polygons.

Theorem 2.8 *A polygon is truely pseudo-star-shaped iff*
(1) its inner pseudo-kernel is empty and
(2) its outer pseudo-kernel is composed of only non-empty bounded polygonal regions.

Proof Follows from Theorems 4.5-4.7. q.e.d.

In the construction of the pseudo-kernel as discussed in the next section, the truely pseudo-star-shaped polygons represent the hardest problem instances.

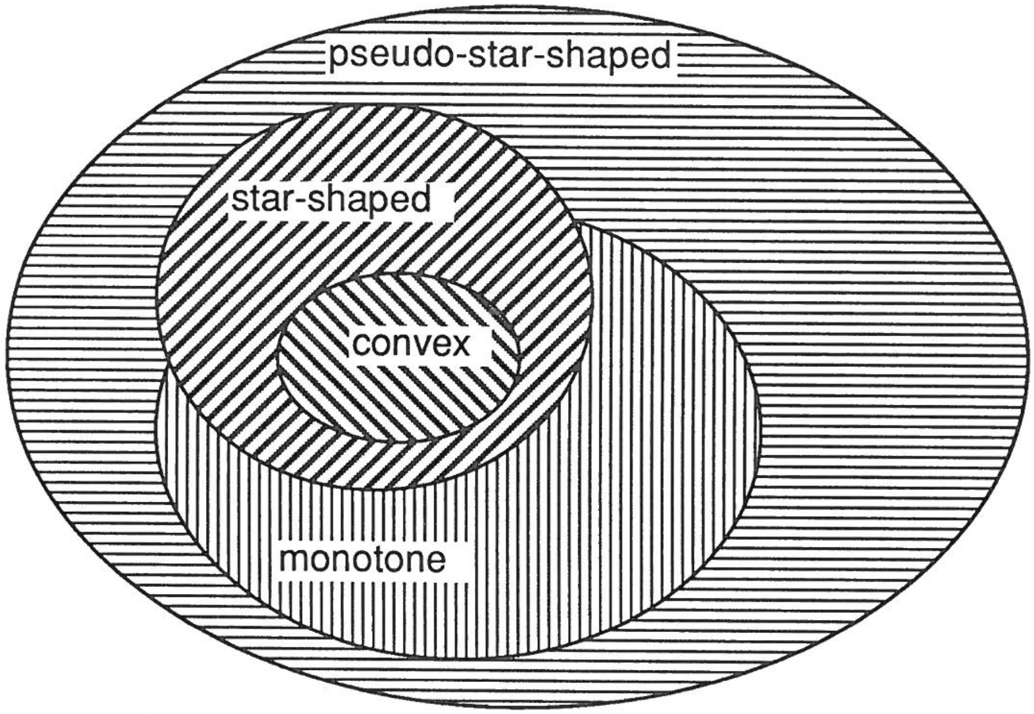

Fig. 4. Relationships between polygon classes: convex, star-shaped, monotone and pseudo-star-shaped.

Polygon Class	Pseudo-Kernel
convex	entire plane including infinity
star-shaped	non-empty inner pseudo-kernel
monotone	unbounded outer pseudo-kernel
pseudo-star-shaped	non-empty pseudo-kernel

Fig. 5. Characterization of polygon-classes via pseudo-kernels.

3 Solutions to the spy problems

Of the two spy-problems discussed in Section 1, we first solve the problem of determining whether a given location is satisfactory to perform our spying task, i.e. whether P is pseudo-star-shaped from that location. Then we construct all possible locations for the equipment, i.e. we construct the pseudo-kernel for P. A desirable location for placing the equipment can subsequently be chosen from these possible locations subject to other, possibly non-geometrical constraints.

3.1 Testing for pseudo-star-shapedness from a given point

As we have seen earlier, testing a polygon P for pseudo-star-shapedness from a given point u in the interior of P is equivalent to testing P for star-shapedness from this point. We could do this by constructing the kernel of P using the linear-time algorithm of Lee and Preparata (1979), and subsequently testing u for inclusion in the kernel. Alternately, without explicitly constructing the kernel, we can test whether all vertices of P are angularly sorted around u, i.e. whether their polar angles are sorted, where the polar angles are taken with u as origin.

When u is outside P we need to test at most two chain pairs for being angularly sorted around u. To find such a pair (if any), we construct a pair of support vertices for u (see Lemma 2.3). We distinguish between u outside the convex hull CH(P) and u in CH(P)\P (where "\" denotes the set difference); the latter case is more interesting. Notice that for a point u outside CH(P) a support vertex pair will always exist, whereas this is not necessarily true for points inside CH(P)\P. The following describes how to determine if a support vertex pair for a point u exists and how it can be found.

Lemma 3.1 *Given a point u outside P then the existence and location of a support vertex pair can be determined in linear time.*
Proof If u is outside CH(P) then support vertices can be found in linear time (Preparata and Shamos 1985). Otherwise, we compute the region of P that is visible from u. This is obtained, in linear time, from the solution to the hidden-line elimination problem for polygon P with view-point u (see e.g. Lee 1983; Dean and Sack 1985). If the solution is a (closed) polygon no support vertex can exist. Otherwise, the solution is a polygonal chain, C_{ij}, whose endpoints, p_i and p_j are the support vertices of P for u. q.e.d.

If support vertices for points u outside CH(P) need be found repeatedly then, as shown e.g. in (Preparata and Shamos 1985), this can be done in O(log n) time per query, where n is the number of vertices of P. Using a technique developed Sack and Toussaint (1985) this bound can also be achieved in case the point u is inside CH(P)\P.

Theorem 3.2 *A polygon can be tested for pseudo-star-shapedness from a given point in linear time.*
Proof By Lemma 2.3 it is sufficient to find support vertices for u and subsequently test whether these are chain defining. Support vertices p_i, p_j can be found in linear time by

using Lemma 3.1. To verify whether the support vertices are chain-defining it suffices to test whether the vertices on each of the two chains are sorted either angularly around the point or along a straight-line perpendicular to the direction of the point. Also this test can be performed in linear-time. q.e.d.

3.2 Constructing pseudo-kernels

In this section, we shall address the problem of constructing the pseudo-kernel of a simple polygon. By Theorem 2.2, the inner pseudo-kernel of a simple polygon P is equal to the kernel of P. Since the latter can be constructed by the algorithm of Lee and Preparata (1979) in linear time, the most difficult part of the construction of the pseudo-kernel of polygon is the construction of its outer part. It requires, in the worst case, quadratic time.

We approach the problem of constructing the pseudo-kernel of a simple polygon $P=(p_0, p_1,\ldots, p_{n-1})$ by considering for each pair (i, j), where $0 \le i,j \le n-1$ and $i \ne j$, the locus PK(i,j) of all points in the outer pseudo-kernel of P such that p_i and p_j are the support vertices for u, and u can completely see the chains C_{ij} and C_{ji}, the former even in the presence of the latter. Thus, for any $0 \le i \le n-1$, PK(i,i) is the inner pseudo-kernel of P. Clearly, the pseudo-kernel of P is the union of all loci PK(i,j), where $0 \le i,j \le n-1$ and $i \ne j$, or i=0 and j=0.

Denote by e_i the edge of P directed from $p_{i-1 \bmod n}$ to p_i (in this section, modulo operations on indices are taken only when specified). Next, let $LP(e_j)$ and $RP(e_j)$, respectively, denote the left and the right half-plane induced by e_j. The following lemma yields an obvious way of constructing the sets PK(i,j): it will also be used for establishing the lower bound on the number of disjoint PK(i,j)'s as derived in Theorem 3.4

Lemma 3.3 *Given a polygon $P=(p_0, p_1,\ldots,p_{n-1})$, for $0 \le i,j \le n-1$ and $i \ne j$, or i=0 and j=0, the set PK(i,j) is the common intersection of $LP(e_{i+1 \bmod n})$, $LP(e_{i+2 \bmod n}),\ldots,LP(e_{j \bmod n})$ with $RP(e_{j+1 \bmod n})$, $RP(e_{j+2 \bmod n}),\ldots,RP(e_i)$.*

Proof Lee and Preparata (1979) proved that the kernel of P is the intersection of $LP(e_1)$, $LP(e_2),\ldots,LP(e_{n-1})$, $LP(e_0)$. Hence, by Theorem 2.4, we conclude that the thesis holds for i=0 and j=0, when PK(0,0) is the inner pseudo-kernel of P.

Suppose that $(0,0) \ne (i,j)$. By the definition of PK(i,j), any point u in PK(i,j) can completely see C_{ji} in the presence of C_{ij}.

Assume that there is an edge e_k of C_{ji} such that u lies in $LP(e_k)$. Then, any straight-line segment (or half-line, if u is at infinity) connecting u with the inside of e_k intersects the interior of P. Since u is outside P, we obtain a contradiction to the definition of PK(i,j). We infer that u lies in the common intersection of $RP(e_{j+1 \bmod n})$, $RP(e_{j+2 \bmod n}),\ldots,RP(e_i)$.

Assume in turn that there is an edge e_k of C_{ij} such that u lies in $RP(e_k)$. Then, any straight-line originating at u and intersecting e_k has to intersect the boundary of P again

after passing through e_k. We obtain a contradiction to the visibility assumptions in both cases, when the next intersection point is on C_{ij}, or on C_{ji}, respectively.
Conversely, suppose that u lies in the common intersection of $LP(e_{i+1 \bmod n})$, $LP(e_{i+2 \bmod n}),\ldots,LP(e_{j \bmod n})$ with $RP(e_{j+1 \bmod n})$, $RP(e_{j+2 \bmod n}),\ldots,RP(e_i)$. Assume that there is a point q on an edge of C_{ji} that cannot see u. Then, the segment (or half-line) [u,q] passes through some point p of the boundary of P different from q such that p can see u. We may assume without loss of generality that the open segment (or half-line) (p,u) does not intersect any edge of P. Let e_k be an edge of P on which p lies. It follows that u is in $RP(e_k)$ and hence the edge e_k is on C_{ji}. Consequently, the open segment (p,q) has to intersect at least one edge e_m on C_{ji} such that u is in $LP(e_m)$. We obtain a contradiction.
The proof that u can see C_{ij} in the absence of C_{ji} is analogous. q.e.d.

Suppose that the intersection of the half-planes from Lemma 3.3 can be found in linear time. Let r be the number of convex (from inside) vertices of P. Then, by separately constructing each of the regions PK(i,j) where p_i and p_j are different convex vertices of P, or i=0 and j=0, we could construct the pseudo-kernel of P in time $O(nr^2)$. Before improving on this upper bound we will derive a lower bound on the number of disjoint components of the outer pseudo kernel.

Theorem 3.4 *For all $n \geq 3$, there is a polygon with n vertices whose outer pseudo kernel consists of $\Omega(n^2)$ pairwise disjoint components.*
Proof Consider a semi-circular polygon $P=(p_0,\ldots,p_{n-1})$ with base p_0p_{n-1} (see Figure 6). Draw all lines co-linear with edges of P. These lines intersect in $\Omega(n^2)$ points partitioning the outside of P into $\Theta(n^2)$ quadrilaterals and $\Theta(n)$ triangles. It is easy to see that each of the quadrilaterals is a region PK(i,j) for P, where i<j-1.
Now consider a second polygon $P'=(p_0,p'_1,p_1,p'_2,p_2,\ldots,p'_{n-1},p_{n-1})$, where for $k=1,\ldots,n-1$, p'_k is the point on the perpendicular bisector of the edge $p_{k-1}p_k$ in the distance ε from the edge $p_{k-1}p_k$ inside P (see Figure 7). It is clear that any point u outside P that completely sees the edges $p_{k-1}p'_k$, and p'_kp_k has to be outside the two thin cones induced by the above edges (see Figure 7). By choosing ε small enough, we can ensure that for each of the discussed quadrilaterals PK(i,j) there is a non-empty part PK'(i,j) of PK(i,j) outside all the 2n-2 forbidden cones. An example of such a polygon P' and its pseudo-kernel is given in Figure 8. Using Lemma 3.3, we can conclude that any point u in PK'(i,j), i<j-1, is in the pseudo-kernel of P' since u is in PK(i,j) and is not in the forbidden cones.
On the other hand, no point ouside P on any of the lines co-linear with the edges of P can be in the outer pseudo-kernel of P'. Suppose otherwise, and let u be a point on the line L co-linear with the edge p_jp_{j+1} of P. Turn L slightly around u such that it intersects the edges incident to p_j and p_{j+1}. L, in the new position, intersects the boundary of P' in at least four points and therefore P cannot be pseudo-star-shaped from u by Lemma 2.1 (3). q.e.d.

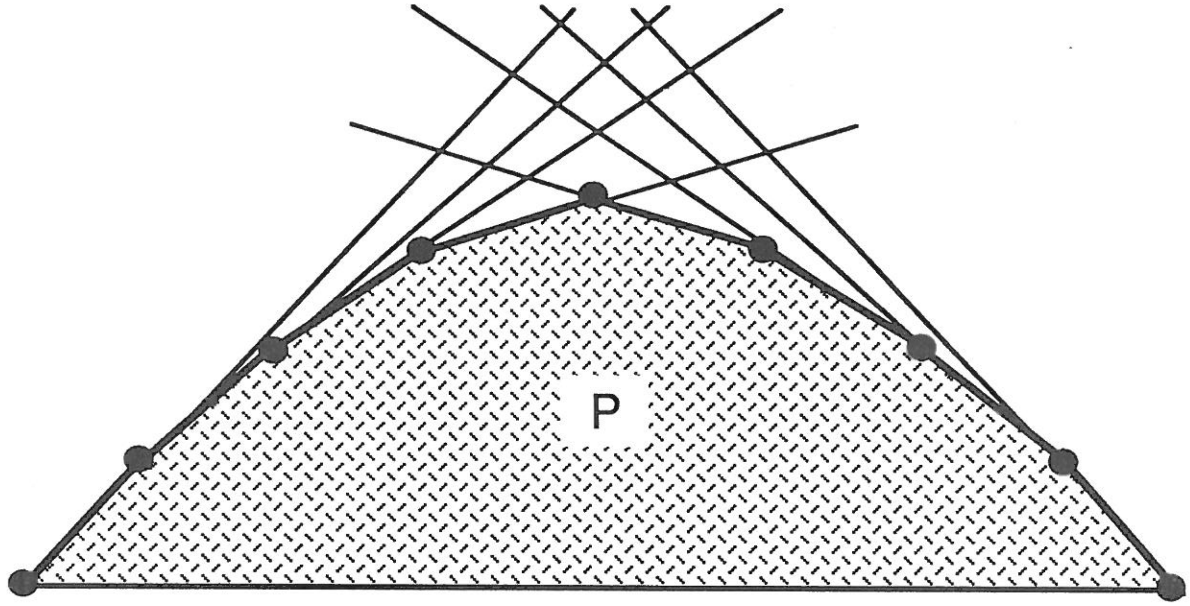

Fig. 6. A semi-circular polygon partitioning the exterior of P into $\theta(n^2)$ qudarilaterals.

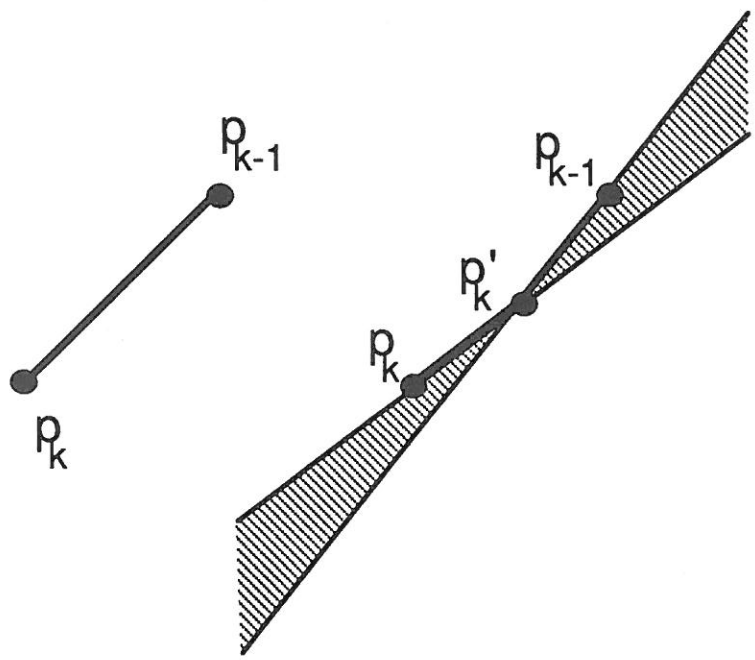

Fig. 7. "Bending" and edge $p_{k-1}p_k$ to create $p_{k-1}p'_kp_k$ and the induced two cones.

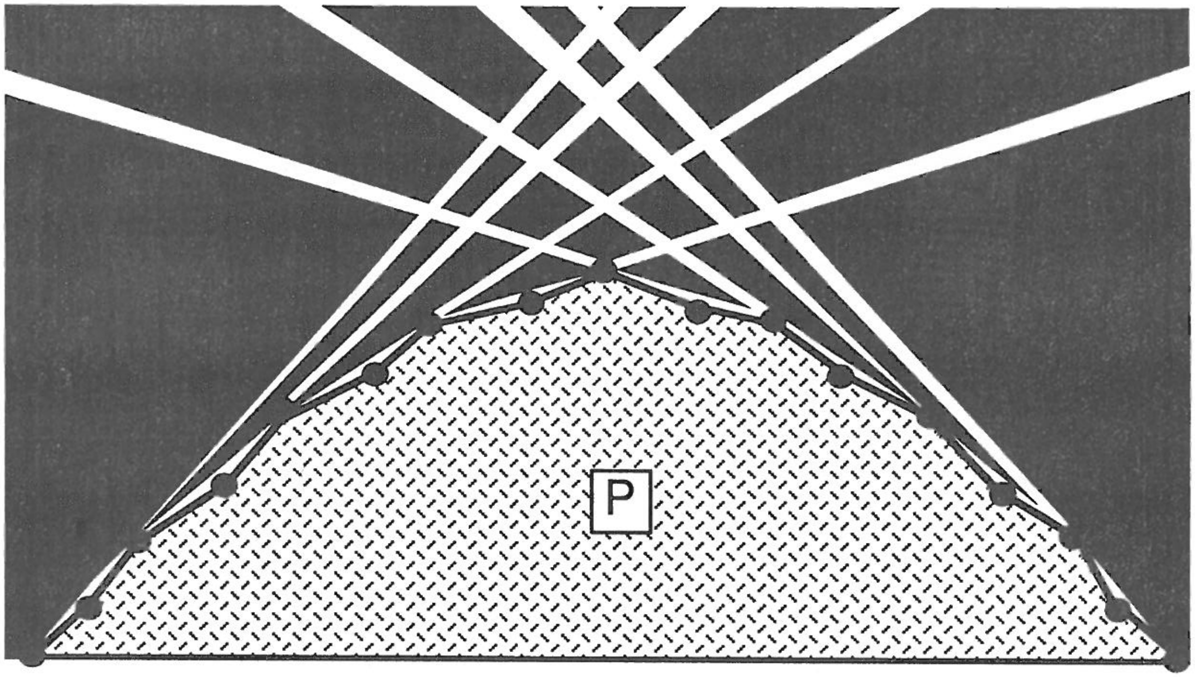

Fig. 8. A pseudo-star-shaped polygon with $\theta(n^2)$ loci in the outer pseudo-kernel.

By Theorem 3.4 any algorithm for constructing the pseudo-kernel of P which constructs all the non-empty regions PK(i,j) takes $\Omega(n^2)$ time, in the worst case. We shall show how to construct all loci PK(i,j) and the outer and inner pseudo-kernels of P in time O(nr). The solution we shall outline, reduces the original problem for P to r subproblems of the following form:

> Given a convex vertex p_k of P, construct PK(k,j) for all convex vertices v_j of P, where $j \neq k$.

To obtain the O(nr)-time solution, we need to solve the subproblem in linear time. We may assume without loss of generality that k=0.
For j=1,...,n, let OK(j) denote the common intersection of $LP(e_1)$, $LP(e_2)$,...,$LP(e_{j \bmod n})$, and for j=0,1,...,n-1, let IK(j) denote the common intersection of $RP(e_{j+1})$, $RP(e_{j+2 \bmod n})$,...,$RP(e_0)$. Note that OK(n) is the kernel of P. In a natural way, we can extend the above definition assuming that OK(0) and IK(n) stand for the plane. Then, it holds:

- i) for j=0,..,n-1, OK(j+1) is a convex subset of $LP(e_{j+1}) \cap OK(j)$,
- ii) for j=1,2,...,n-1, IK(j) is a convex subset of $RP(e_{j+1 \bmod n}) \cap IK(j+1)$,
- iii) for j=1,2,..,n, $OK(j) \cap IK(j) = PK(0, j \bmod n)$.

The first two properties follow directly from the above definitions, the third one follows from Lemma 3.3.
Actually, the linear-time algorithm of Lee and Preparata (1979) for finding the kernel of a polygon applied to P explicitely constructs the sequence OK(1), OK(2),...,OK(n). The region OK(j+1) is constructed on the basis of OK(j) by finding a polygonal chain of the

form $w''w_{s-1}...w_tw'$ or $w''w_s...w_{t-1}w'$, where w' is a point on the straight-line co-linear with e_j and w" is either another point on the line or represents the empty symbol. The chain is cut off from OK(j) along the straight-line (see pp. 417,418 in Lee and Preparata 1979). Since OK(j) is convex, the cut-off polygonal region is also convex. We can interpret the cut-off region as the set difference, od(j), between OK(j) and OK(j+1) extended by the whole segment of the cutting line within OK(j), for j=1,..,n-1. In a natural way, we can extend the above definitions assuming that od(n) is the inner pseudo-kernel of P, i.e. it equals OK(n). For j=0,1,...,n-1, the boundary of od(j) is formed by the mentioned chain plus the segment (w',w") in case od(j) is bounded, or a half-line co-linear with e_j and starting from the end of the chain marked with w" otherwise. Since the algorithm of Lee and Preparata (1979) constructs the chain by scanning all its vertices and finds the closing segment or the half-line and its end, we can trivially modify their algorithm to produce the regions od(j), j=0,1,...,n, keeping its linear time performance. Thus, by (i), the definition of the regions od(j) and the above argumentation, we obtain the following properties of the sets od(0), od(1), od(2),..., od(n):

a) od(j) is a convex subset of $RP(e_{j+1 \bmod n})$,
b) the total number of vertices of od(0), od(1), od(2),..., od(n) is O(n),
c) all the polygonal regions od(0), od(1), od(2),..., od(n) can be constructed in linear time.

Similarly, for j=0,...,n-1, let us define id(j+1) as the set difference between ID(j) and ID(j+1) extended by the cutting segment or half-line. Using the symmetry between the definition of OK(j) , od(j) and that of ID(j), id(j), we can easily adjust our modification of the algorithm of Lee and Preparata (1979) to construct the sequences IK(n-1),...,IK(1) and id(n), id(n-1),...,id(1), in linear time. The adjustment consists in respectively exchanging the adjectives "left" and "right" in the body of the algorithm. In consequence, we obtain the following, symmetric list of properties of the polygonal regions id(1),...,id(n):

d) id(j) is a convex subset of $LP(e_{j+1 \bmod n})$,
e) the total number of vertices of id(1),...,id(n) is O(n),
f) all the polygonal regions id(1),...,id(n) can be constructed in linear time.

In the context of the properties (iii), (b) and (e), the following lemma explains the usefulness of the polygonal regions od(j) and id(j) for finding outer pseudo-kernels of P.

Lemma 3.5 *For j=1,...,n-1, it holds* $OK(j) \cap IK(j) = od(j) \cap id(j)$.

Proof By the definition of od(j), we have $OK(j) \cap IK(j) = (OK(j+1) \cup od(j)) \cap IK(j)$. Since OK(j+1) is a subset of $LP(e_{j+1})$ by (i), and IK(j) is a subset of $RP(e_{j+1})$ by (ii), we conclude that $OK(j) \cap IK(j) = od(j) \cap IK(j)$. By the definition of id(j), the latter equality is equivalent to $OK(j) \cap IK(j) = od(j) \cap (IK(j-1) \cup id(j))$. Since od(j) is a subset of OK(j) and OK(j) is a subset of $LP(e_j)$ by (i), the region od(j) is a subset of $LP(e_j)$. On the other hand, IK(j-1) is a subset of $RP(e_j)$ by (ii). Hence, we conclude that $od(j) \cap IK(j) = od(j)$

$\cap$ id(j) and using the above equation OK(j) $\cap$ IK(j) = od(j) $\cap$ IK(j) the result follows. q.e.d.

By (iii) and Lemma 3.5, given the sequences od(0), od(1),..., od(n-1) and id(1),..., id(n-1), to solve the subproblem for the vertex p_0 it is sufficient to find the intersections od(j) $\cap$ id(j) for j=1,...,n-1. Thus, the entire algorithm for constructing the pseudo-kernel of a simple polygon can be outlined as follows:

Algorithm 3.1 *Constructing the loci PK(i,j) for a simple polygon*
input. a simple polygon $Q=(w_0, w_1,\dots,w_{n-1})$
output. all non-empty loci PK(i,j) for Q

```
for each convex vertex w_k of Q do
  begin
    rename the vertices of Q such that ( w_k, w_{k+1 mod n},...,w_{k+n-1 mod n})=( p_0, p_1,..., p_{n-1});
    find the sequences od(0), od(1),..., od(n) and id(1),...,id(n);
    for j=1,...,n-1 do
    if w_{k+j mod n} is convex then
      begin
        PK(k,k+j mod n) := od(j) ∩ id(j);
        return PK(k,k+j mod n)
      end
  end
PK(0,0) := od(n);
return PK(0,0);
and i ≠ j.
```

Theorem 3.6 *Algorithm 3.1 constructs all non-empty loci PK(i,j) for a simple polygon with r convex vertices and n-r concave vertices in time O(nr).*
Proof The correctness of the algorithm follows from the correctness of the reduction to the r subproblems and then from (iii) and Lemma 3.1. By (c), (f), the convex polygonal regions od(1),..., od(n) and id(1),...,id(n-1) for a given vertex w_k of Q can be constructed in time O(n). Since the intersection of two convex polygons can be found in time linear in the total number of their vertices (Preparata and Shamos 1985), the intersections od(j) $\cap$ id(j) , j=1,...,n-1, for a given vertex w_k of Q, can be constructed in time O(n) by (b) and (e). Thus, the whole algorithm can be implemented in time O(nr). q.e.d.

To construct the outer pseudo-kernel of Q it is sufficient to find the union of the regions PK(i,j) where p_i and p_j are convex vertices of Q. By Theorem 3.6, the above regions are of total size O(nr) and therefore we could find their union in time O(nr logn) by applying a standard sweep-line technique. We can even do it in time O(nr) by observing the following consequences of Lemma 3.3:

The regions PK(i,j) are convex, pairwise disjoint and their boundaries lie on the n lines co-linear with the edges of Q. Moreover, if |i-i'|>1 or |j-j'|>1 then, since no two consecutive edges of Q are co-linear, the regions PK(i,j) and PK(i',j') cannot share a non-degenerate boundary segment.

Thus, each of the regions PK(i,j) may share a boundary segment only with a constant number of other regions PK(i',j') along a constant number of lines co-linear with the edges of Q. By scanning the boundary of each region PK(i,j), we form a constant size list of edges that lie on the above lines. We assign to each edge on such a list, the indices i', j' of the potentially adjacent regions PK(i',j'). Using the constant size lists with the additional information, we detect all pairs of adjacent regions PK(i,j), PK(i',j') and link them appropriately. The last stage consists in scanning and reporting the boundaries of the linked regions. Whenever a link from PK(i,j) to PK(i',j') is encountered, we pass along the link to the boundary of PK(i',j') continuing the edge scanning and reporting. If the corresponding link from PK(i',j') to PK(i,j) has not yet been scanned we push it on a special stack. It will be popped from the stack if we encounter it during the continued scan. Otherwise, the boundary piece of PK(i,j) following the corresponding link lies on another boundary of the union and we shall use the link to initialize its scan later. We skip here lengthy details of the union procedure as not relevant to the topic of the paper. The whole procedure clearly can be implemented in time proportional to the number of scanned edges of non-empty regions PK(i,j), i.e. O(nr) by Theorem 3.6.

Proposition 3.7 *Given a simple polygon with r convex vertices and n-r concave vertices, we can construct its outer pseudo-kernel in time O(nr).*

4 Geometrical algorithms for pseudo-star-shaped polygons

We have seen how convexity, star-shapedness and monotonicity are related to pseudo-star-shapedness; monotonicity implies a linear ordering of the vertices on the two chains along a straight-line (which we refer to as sorted around u at infinity), while convexity and star-shapedness imply an angular ordering around point u on the plane. Pseudo-star-shapedness encompasses both these orderings. In making use of this more general ordering relation, several other problems studied in computational geometry, like point location, triangulation, union and intersection problems, can be solved efficiently. We will sketch how two of the existing geometrical algorithms can be generalized to work on pseudo-star-shaped polygons and refer the reader to (Dean 1985) for further details.

4.1 Point location

Shamos and Preparata (1985) describe a method for solving the *point location problem* for monotone polygons, i.e. the problem of determining whether a given query point q is inside or outside a polygon P. For convex or star-shaped polygons a "wedge"-method (see e.g. Preparata and Shamos 1985) has been designed for the same problem. The methods are easily generalized to produce a solution to the point location problem for a given n-vertex polygon pseudo-star-shaped from some point u. We describe the case when u is outside P and is not at infinity. We draw half-lines connecting u and every vertex p_i of P. Thereby, a partition of P into O(n) regions angularly sorted around u is created. The region of the plane enclosed by two adjacent such half-lines is called a *wedge*. One of these wedges is empty (two, in case of u at infinity). While all other wedges intersect two edges of P, one on the chain C_{ij} the other on the chain C_{ji}, for the chain-defining vertices p_i and p_j. The intersection of such a wedge with P is a convex quadrilateral. Now a query point q is inside P iff q is inside one of the quadrilaterals (possibly degenerated to a triangle). Using binary search on the polar angles defined by the rays followed by a point location test in the so determined quadrilateral the point location query can be answered in O(log n) time.

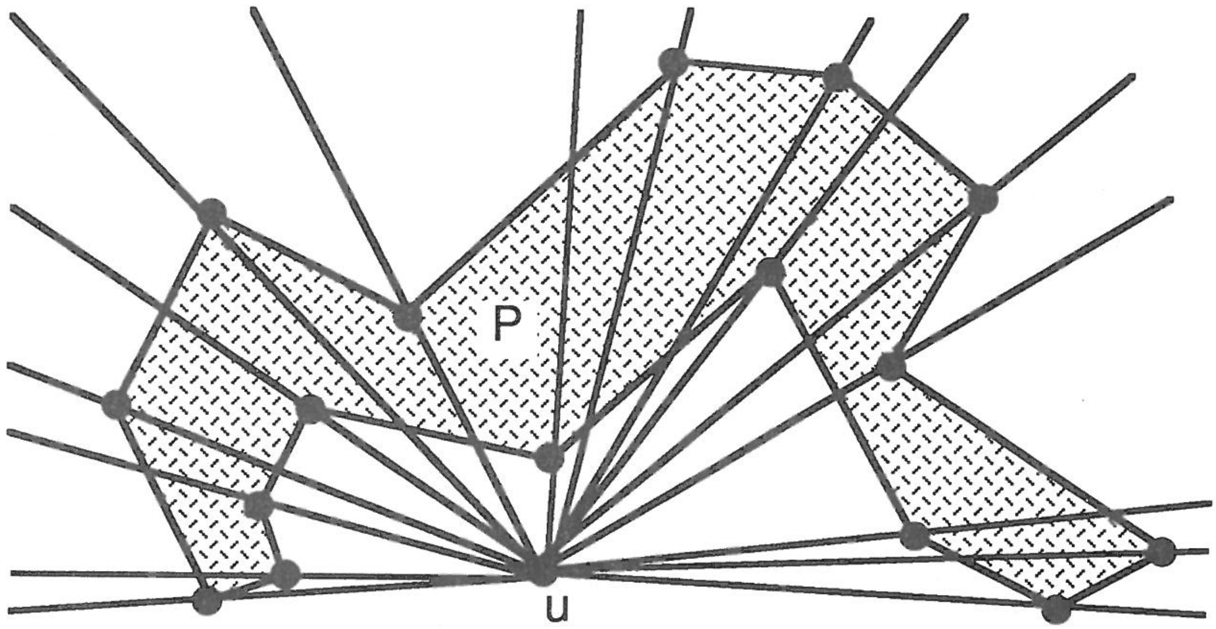

Fig. 9. A pseudo-star-shaped polygon preprocessed for efficient point location tests.

4.2 A simple triangulation algorithm for pseudo-star-shaped polygons

Toussaint (1983) describes a simple, linear-time algorithm to triangulate a monotone polygon. A given monotone polygon is first decomposed into weakly edge-visible polygons and subsequently these are triangulated using a method (Toussaint and Avis 1982) derived from Sklansky's convex hull algorithm. We show that this idea can be generalized to obtain an efficient triangulation algorithm for pseudo-star-shaped polygons. We will give an algorithm for partitioning a pseudo-star-shaped polygon into *weakly edge-visible polygons,* i.e. polygons in which each point can be seen from some point of a fixed edge. The remaining part of the triangulation algorithm is then analogous to that of Toussaint's.

Algorithm 4.1 *Partitioning a Pseudo-Star-Shaped Polygon into Weakly Edge-Visible Polygons*

input. A pseudo-star-shaped polygon P and a point w from which P is pseudo-star-shaped.
output. A partition of P into weakly edge-visible polygons.

begin

1. Determine support vertices p_i and p_j with respect to w;
   Store the resulting chains as lists C_{ij} and C_{ji}, sorted angularly with respect to w; call the vertices of C_{ij}, p-vertices and those of C_{ji}, q-vertices;

2. Merge the two sorted lists to obtain a new sorted list;

3. Scan the new list and whenever a p (or q) vertex follows a q (or p) vertex add a diagonal between them.

end.

Figure 10 illustrates the partitioning of a pseudo star-shaped polygon into weakly edge-visible polygons as produced by the algorithm.

Lemma 4.1 *Algorithm 4.1 correctly partitions a pseudo-star-shaped polygon into weakly edge-visible polygons. The running-time of the algorithm is linear in the number of vertices of P.*
Proof By the fact that each diagonal is defined by a q-vertex and a p-vertex adjacent in the sorted list, it is clear that no two diagonals intersect and furthermore that each diagonal is inside P.The proof that each piece is weakly edge-visible as well as the complexity analysis follow directly from (Toussaint 1983) and (Toussaint and Avis 1982). It is thus omitted here. q.e.d.

Theorem 4.2 *Any polygon P pseudo-star-shaped from a given point u can be triangulated in linear time.*
Proof The theorem follows from Lemma 4.1 and the fact that a weakly edge-visible polygon can be triangulated in linear time (Toussaint and Avis 1982). q.e.d.

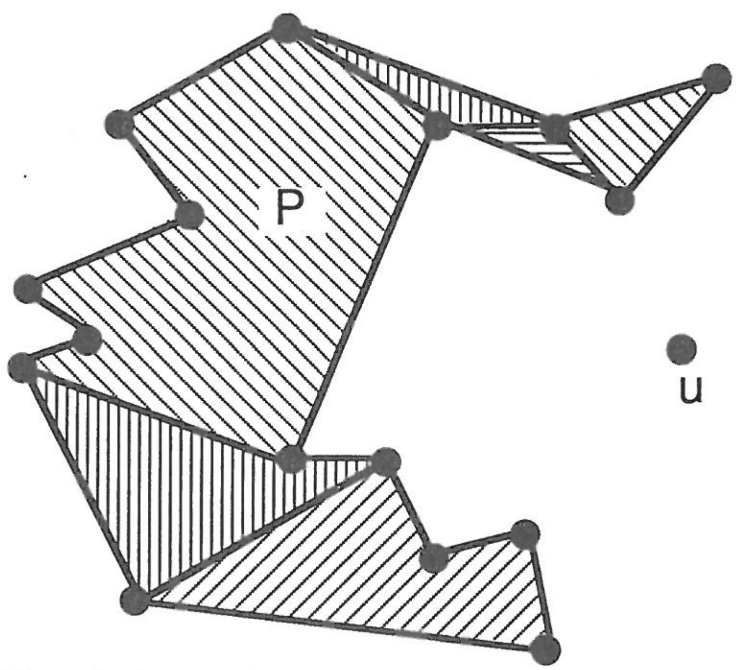

Fig. 10. A pseudo-star-shaped polygon partitioned into weakly edge-visible polygons.

5 Conclusions and open problems

A new class of polygons has been presented which unifies and generalizes three traditional computational geometry polygon classes (convex, star-shaped and monotone). Two worst-case optimal algorithms for testing a given polygon for pseudo-star-shapedness from a point and for constructing its pseudo-kernel have been presented. The optimality of the latter has been based on the quadratic lower bound on the number of disjoint polygonal components of the pseudo-kernel. However, this lower bound argument cannot be used for the following important open problems arising from this work:

(1) Determine whether a given polygon is pseudo-star-shaped (yes/no answer).
(2) Find a point (if any) from which a given polygon is pseudo-star-shaped.

The applications presented in Section 4 require knowledge of at least one point from which the input polygon is pseudo-star-shaped. Therefore it would be desirable to solve the above problems in linear time. To possibly achieve this goal, one might have to discover, additional properties of the new polygon class.

The concept of polygon pseudo-star-shapenedess can be generalized in several ways.
(1) By allowing visibility through at most k edges a proper hierachy of polygon classes P(0), P(1), P(2),... could be defined. For a non-negative integer k, P(k) is the class of polygons each of which can be seen from a single point by looking through at most k edges.
(2) By allowing more than one observation point, questions similar to those arizing in the study of guard problems (O'Rourke 1987) could be posed. In this case, the polygons are not necessarily simple and may contain polygonal holes.

Acknowledgement The authors are greatly indebted to Mark Keil for suggesting the n-vertex polygon leading to the lower bound established in Section 4.

References

Dean JA (1985) Structural properties of polygons. Master's thesis, Carleton University Ottawa, Canada

Dean JA and J-R Sack (1985) Efficient hidden-line elimination by capturing winding information. Proc 23rd Allerton Conference on Communication, Control and Computing, Monticello Ill., pp 496-505

Keil JM (1985) Decompositions of polygons into simpler components. SIAM Journal of Computing 14: 799-817

Lee DT and FP Preparata (1979) An optimal algorithm for finding the kernel of a simple polygon. Journal of the ACM 26 (3): 415-421

Lee DT (1983) Visibility of a simple polygon. Computer Vision, Graphics and Image Processing 22: 207-221

Newman W and R Sproull (1978) Principles of interactive computer graphics. McGraw Hill, New York
O'Rourke J (1987) Art-Gallery Theorems. Oxford University Press
Preparata FP and KJ Supowit (1981) Testing a simple polygon for monotonicity. Information Processing Letters 12: 161-164
Preparata FP and MI Shamos (1985) Computational geometry: an introduction. Springer, Berlin Heidelberg New York Tokyo
Sack J-R and GT Toussaint (1985) Translating polygons in the plane. Proc STACS85, Saarbrücken FRG, Lecture Notes in Computer Science 182, Springer, Berlin Heidelberg New York Tokyo, pp 310-321; also (1987) *Robotica* 5: 55-63.
Toussaint GT and D Avis (1982) On a convex hull algorithm for polygons and its applications to triangulation. Pattern Recognition 15 (1): 23-29
Toussaint GT (1983) A new linear algorithm for triangulating monotone polygons. Tech. Rept. SOCS-83.9, School of Computer Science, McGill University, Montréal, Canada
Toussaint GT (1985) Movable separability of sets. In Computational Geometry, GT Toussaint (Editor) North-Holland Amsterdam New York Oxford pp. 335-376

Carleton University, School of Computer Science
Bibliography of Technical Reports
Publications List (1985 -->)

School of Computer Science
Carleton University
Ottawa, Ontario, Canada
KIS 5B6

SCS-TR-66 — **On the Futility of Arbitrarily Increasing Memory Capabilities of Stochastic Learning Automata**
B.J. Oommen, October 1984. Revised May 1985.

SCS-TR-67 — **Heaps in Heaps**
T. Strothotte, J.-R. Sack, November 1984. Revised April 1985.

SCS-TR-68 out-of-print — **Partial Orders and Comparison Problems**
M.D. Atkinson, November 1984. See Congressus Numerantium 47 ('86), 77-88

SCS-TR-69 — **On the Expected Communication Complexity of Distributed Selection**
N. Santoro, J.B. Sidney, S.J. Sidney, February 1985.

SCS-TR-70 — **Features of Fifth Generation Languages: A Panoramic View**
Wilf R. LaLonde, John R. Pugh, March 1985.

SCS-TR-71 — **Actra: The Design of an Industrial Fifth Generation Smalltalk System**
David A. Thomas, Wilf R. LaLonde, April 1985.

SCS-TR-72 — **Minmaxheaps, Orderstatisticstrees and their Application to the Coursemarks Problem**
M.D. Atkinson, J.-R. Sack, N. Santoro, T. Strothotte, March 1985.

SCS-TR-73 — **Designing Communities of Data Types**
Wilf R. LaLonde, May 1985.
Replaced by SCS-TR-108

SCS-TR-74 out-of-print — **Absorbing and Ergodic Discretized Two Action Learning Automata**
B. John Oommen, May 1985. See IEEE Trans. on Systems, Man and Cybernetics, March/April 1986, pp. 282-293.

SCS-TR-75 — **Optimal Parallel Merging Without Memory Conflicts**
Selim Akl and Nicola Santoro, May 1985

SCS-TR-76 — **List Organizing Strategies Using Stochastic Move-to-Front and Stochastic Move-to-Rear Operations**
B. John Oommen, May 1985.

SCS-TR-77 — **Linearizing the Directory Growth in Order Preserving Extendible Hashing**
E.J. Otoo, July 1985.

SCS-TR-78 — **Improving Semijoin Evaluation in Distributed Query Processing**
E.J. Otoo, N. Santoro, D. Rotem, July 1985.

SCS-TR-79	**On the Problem of Translating an Elliptic Object Through a Workspace of Elliptic Obstacles** B.J. Oommen, I. Reichstein, July 1985.
SCS-TR-80	**Smalltalk - Discovering the System** W. LaLonde, J. Pugh, D. Thomas, October l985.
SCS-TR-81	**A Learning Automation Solution to the Stochastic Minimum Spanning Circle Problem** B.J. Oommen, October 1985.
SCS-TR-82	**Separability of Sets of Polygons** Frank Dehne, Jörg-R. Sack, October 1985.
SCS-TR-83 out-of-print	**Extensions of Partial Orders of Bounded Width** M.D. Atkinson and H.W. Chang, November 1985. See Congressus Numerantium, Vol. 52 (May 1986), pp. 21-35.
SCS-TR-84	**Deterministic Learning Automata Solutions to the Object Partitioning Problem** B. John Oommen, D.C.Y. Ma, November 1985
SCS-TR-85 out-of-print	**Selecting Subsets of the Correct Density** M.D. Atkinson, December 1985. To appear in Congressus Numerantium, Proceedings of the 1986 South-Eastern conference on Graph theory, combinatorics and Computing.
SCS-TR-86	**Robot Navigation in Unknown Terrains Using Learned Visiblity Graphs. Part I: The Disjoint Convex Obstacles Case** B. J. Oommen, S.S. Iyengar, S.V.N. Rao, R.L. Kashyap, February 1986
SCS-TR-87	**Breaking Symmetry in Synchronous Networks** Greg N. Frederickson, Nicola Santoro, April 1986
SCS-TR-88	**Data Structures and Data Types: An Object-Oriented Approach** John R. Pugh, Wilf R. LaLonde and David A. Thomas, April 1986
SCS-TR-89	**Ergodic Learning Automata Capable of Incorporating Apriori Information** B. J. Oommen, May 1986
SCS-TR-90	**Iterative Decomposition of Digital Systems and Its Applications** Vaclav Dvorak, May 1986.
SCS-TR-91	**Actors in a Smalltalk Multiprocessor: A Case for Limited Parallelism** Wilf R. LaLonde, Dave A. Thomas and John R. Pugh, May 1986
SCS-TR-92	**ACTRA - A Multitasking/Multiprocessing Smalltalk** David A. Thomas, Wilf R. LaLonde, and John R. Pugh, May 1986
SCS-TR-93	**Why Exemplars are Better Than Classes** Wilf R. LaLonde, May 1986
SCS-TR-94	**An Exemplar Based Smalltalk** Wilf R. LaLonde, Dave A. Thomas and John R. Pugh, May 1986
SCS-TR-95	**Recognition of Noisy Subsequences Using Constrained Edit Distances** B. John Oommen, June 1986

SCS-TR-96 **Guessing Games and Distributed Computations in Synchronous Networks**
J. van Leeuwen, N. Santoro, J. Urrutia and S. Zaks, June 1986.

SCS-TR-97 **Bit vs. Time Tradeoffs for Distributed Elections in Synchronous Rings**
M. Overmars and N. Santoro, June 1986.

SCS-TR-98 **Reduction Techniques for Distributed Selection**
N. Santoro and E. Suen, June 1986.

SCS-TR-99 **A Note on Lower Bounds for Min-Max Heaps**
A. Hasham and J.-R. Sack, June 1986.

SCS-TR-100 **Sums of Lexicographically Ordered Sets**
M.D. Atkinson, A. Negro, and N. Santoro, May 1987.

SCS-TR-102 **Computing on a Systolic Screen: Hulls, Contours, and Applications**
F. Dehne, J.-R. Sack and N. Santoro, October 1986.

SCS-TR-103 **Stochastic Automata Solutions to the Object Partitioning Problem**
B.J. Oommen and D.C.Y. Ma, November 1986.

SCS-TR-104 **Parallel Computational Geometry and Clustering Methods**
F. Dehne, December 1986.

SCS-TR-105 **On Adding *Constraint Accumulation* to Prolog**
Wilf R. LaLonde, January 1987.

SCS-TR-107 **On the Problem of Multiple Mobile Robots Cluttering a Workspace**
B. J. Oommen and I. Reichstein, January 1987.

SCS-TR-108 **Designing Families of Data Types Using Exemplars**
Wilf R. LaLonde, February 1987.

SCS-TR-109 **From Rings to Complete Graphs - Θ(n logn) to Θ(n) Distributed Leader Election**
Hagit Attiya, Nicola Santoro and Shmuel Zaks, March 1987.

SCS-TR-110 **A Transputer Based Adaptable Pipeline**
Anirban Basu, March 1987.

SCS-TR-111 **Impact of Prediction Accuracy on the Performance of a Pipeline Computer**
Anirban Basu, March 1987.

SCS-TR-112 **ε-Optimal Discretized Linear Reward-Penalty Learning Automata**
B.J. Oommen and J.P.R. Christensen, May 1987.

SCS-TR-113 **Angle Orders, Regular n-gon Orders and the Crossing Number of a Partial Order**
N. Santoro and J. Urrutia, June 1987.

SCS-TR-115 **Time is Not a Healer: Impossibility of Distributed Agreement in Synchronous Systems with Random Omissions**
N. Santoro, June 1987.

SCS-TR-117 **Recognizing Polygons, or How to Spy**
James A. Dean, Andrzej Lingas and Jörg-R. Sack, August 1987.

SCS-TR-118 **Stochastic Rendezvous Network Performance - Fast, First-Order Approximations**
J.E. Neilson, C.M. Woodside, J.W. Miernik, D.C. Petriu, August 1987.