k-Star-shaped Polygons*

William Evans† Jeff Sember†

Abstract

We introduce k-star-shaped polygons, polygons for which there exists at least one kernel point x such that for any point y of the polygon, segment $\overline{xy}$ crosses the polygon's boundary at most k times. We show that a polygon's kernel has $O(n^2)$ complexity (if $k = 2$), and $O(n^4)$ complexity (if $k \geq 4$). We prove that both bounds are tight, and give a worst-case optimal algorithm for constructing this kernel. Finally, we show how k-convex polygons can be recognized in $O(n^2 \cdot \min(1 + k, \log n))$ time and $O(n)$ space.[1]

1 Introduction

The kernel of a polygon P is the set of points x such that $\overline{xy} \subset P$ for all $y \in P$. In other words, the kernel is the set of points that can see all of P when the boundary of P blocks all lines of sight. In some applications, lines of sight may cross the boundary of P to a limitied extent. We say that two points x and y are *mutually k-visible* if $\overline{xy}$ crosses the boundary of P at most k times, and define the *k-kernel* of P to be the set of points x that are k-visible to every point of P. Note that points in the k-kernel may be outside[2] of P for $k \geq 1$. We denote the k-kernel of P by $M^k(P)$ (or, when k and P are clear from the context, M). P is k-convex if $P \subseteq M^k(P)$.

Lee and Preparata [6] describe an optimal $O(n)$ algorithm to find $M^0(P)$. Aicholzer et al. [1] introduce the notion of k-convexity (using slightly different definitions) and give an $O(n \log n)$ algorithm for recognizing 2-convex polygons, and an $O((1 + k)n)$ algorithm for triangulating k-convex polygons.

Dean, Lingas, and Sack [5] give algorithms that determine if a point is in the 1-kernel (which they call the psuedokernel) of an n-vertex polygon P in $O(n)$ time and that calculate the 1-kernel in $O(n^2)$ time. They show that the latter algorithm is optimal by demonstrating that the 1-kernel may have $\Omega(n^2)$ complexity.

In this paper, we investigate the concept of k-star-shaped polygons: polygons with nonempty k-kernels. We present an efficient algorithm for constructing k-kernels, and for recognizing k-convex polygons.

*Research supported by NSERC and Institute for Computing, Information and Cognitive Systems (ICICS) at UBC

†Department of Computer Science, University of British Columbia, `[will,jpsember]@cs.ubc.ca`

[1]An applet demonstrating these results can be found at `http://www.cs.ubc.ca/~jpsember/ss.html`.

[2]For example, a transmitter that can penetrate a building's walls may 'see' the entire building from an outside location.

2 Properties

Before continuing, we will require some terminology. Polygons are simple, closed, and bounded by a ccw sequence of directed edges. The predecessor and successor vertices of a vertex s of a polygon are denoted s^- and s^+ respectively.

To fully define k-visibility, we must define what constitutes a segment / polygon boundary crossing. The number of crossings that a segment $\overline{xy}$ makes with the boundary of P is equal to the number of edges that intersect $\overline{xy}$, where (i) edges of P collinear with $\overline{xy}$ are excluded, and (ii) if a vertex of P lies on $\overline{xy}$, and the edges of P incident to the vertex lie on opposite sides of $\overline{xy}$, then only one of the edges is counted. Figure 1 illustrates these conditions.

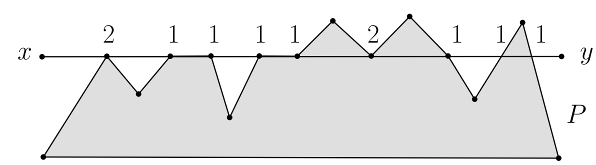

Figure 1: Segment / polygon boundary crossings (crossing counts are indicated)

Lemma 1 *Every point on the boundary of $M^k(P)$ lies on a line containing two vertices of P.*

Proof. Let x be a point of M that does not lie on a line containing two vertices of P. We will show that x cannot lie on the boundary of M. Let r be any ray from x, and suppose r contains no vertices of P. Let y be any point on $r \setminus \{x\}$. There must exist a point x' arbitrarily close to and in any direction λ from x such that ray $\overrightarrow{x'y}$ will have the same number of crossings with the boundary of P as ray $\overrightarrow{xy}$. If instead, r does contain a vertex v of P, then (since v must be unique) we can choose y to be v to get the same result. Since this is true for any choice of r and λ, $x' \in M$; hence x cannot lie on the boundary of M. $\square$

Theorem 2 *The k-kernel of an n-vertex polygon P has $O(n^4)$ complexity.*

Proof. By Lemma 1, there are at most $O(n^2)$ lines containing edges of $M^k(P)$, and these lines can intersect at most $O(n^4)$ times. □

Theorem 3 *For $k \geq 4$, there exist polygons whose k-kernels have $\Theta(n^4)$ complexity.*

Proof. Consider the polygon P of Figure 2. It includes four sequences of $\Theta(n)$ 'Z'-shaped edge sections, which induce $\Theta(n^2)$ aperature pairs. Each aperature generates a narrow gap in M, and these gaps intersect $\Theta(n^4)$ times in the top left of P. □

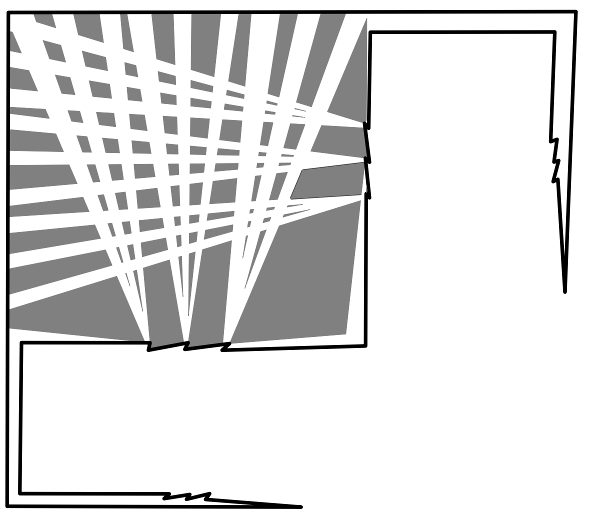

Figure 2: $M^4(P)$ has $\Theta(n^4)$ complexity (P is bold, $M^4(P)$ is shaded; some details omitted for clarity)

3 Constructing the k-kernel

We first define a *v-region*, a structure associated with a polygon's vertex. We will show that a polygon's k-kernel is equal to the intersection of the v-regions of the vertices of the polygon, and provide an efficient algorithm to construct a v-region. This in turn will lead to an algorithm to construct $M^k(P)$.

Definition 1 *The v-region for vertex s of a polygon P, denoted V_s, is the set of points x for which x is k-visible to every point of P on ray $\overrightarrow{xs}$.*

An example of a v-region is shown in Figure 3.

Theorem 4 *$M^k(P)$ is equal to the intersection of the v-regions of P.*

Proof. Suppose some point x is not in M. Then there exists some point $y \in P$ that is not k-visible to x, which implies that segment $\overline{xy}$ contains at least $k+1$ crossings. If x is a vertex of P, then $x \notin V_x$. Otherwise, we can rotate ray $\overrightarrow{xy}$ around x until it contains a vertex s of P and some $y' \in P$, where $\overline{xy'}$ contains at least $k+1$ crossings. Hence, $x \notin V_s$.

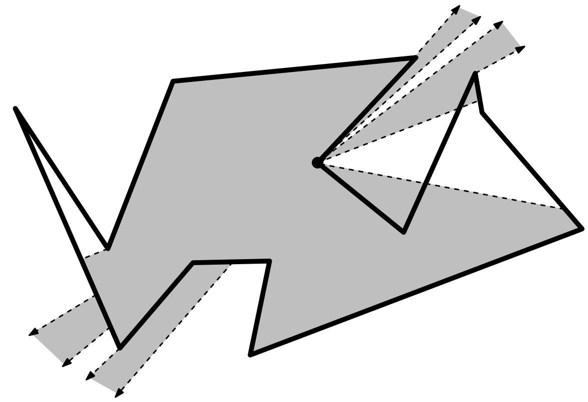

Figure 3: v-region ($k = 2$)

Now suppose there exists a vertex s of P where $x \notin V_s$. Then the ray $\overrightarrow{xs}$ contains some point of P that is not k-visible to x, which implies $x \notin M$. □

Let us investigate how v-regions might be constructed. Suppose s is a vertex of polygon P. Draw lines through s and every other vertex of P. These lines partition the plane into (closed) wedges (2D cones) that contain no vertex of P in their interiors. Each wedge A in the partition has a symmetric 'dual' wedge $\tilde{A}$ in the partition that is bounded by the same lines as those bounding A, and the two wedges are separated by regions A_L and A_R; see Figure 4.

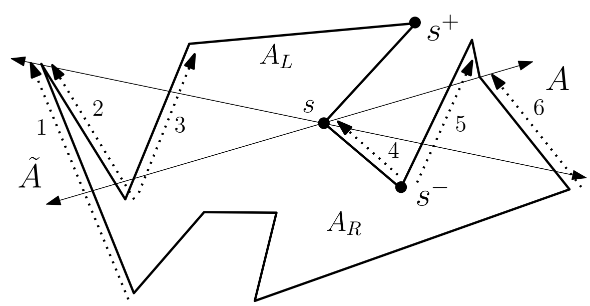

Figure 4: A wedge

We define the *clipping list* $E(A)$ to be the sequence of edges of P that cross A or $\tilde{A}$. We orient the edges in this list to cross from A_R to A_L. We include the edge (s^-, s), and also the edge (s, s^+) if s^+ and s^- both lie in A_L or both lie in A_R. We ignore all remaining edges of P, including those coincident with the lines bounding A and $\tilde{A}$. We order the elements of $E(A)$ according to their signed distance from s, as shown in Figure 4. $E(A)_i$ denotes the i^{th} element of $E(A)$.

We say that a point x is *k-clipped* by a wedge A of vertex s if (i) $x \in A$, and (ii) x is strictly to the right of

$E(A)_{k+2}$ (if k is even), or on or to the right of $E(A)_{k+2}$ (if k is odd).

Lemma 5 *If s is a vertex of polygon P, and x is a point in the interior of some wedge A of s, then x is within V_s iff A does not k-clip x.*

Proof. Suppose A does not k-clip x. Then ray $\overrightarrow{xs}$ will cross at most $k+1$ edges of $E(A)$, which implies that $\overrightarrow{xs}$ crosses the boundary of P at most $k+1$ times. Note also that no part of P lies to the left of $E(A)_1$, so every point of P on the ray is k-visible to x; hence $x \in V_s$. If, on the other hand, A does k-clip x, then ray $\overrightarrow{xs}$ crosses at least $k+1$ edges of $E(A)$, and hence crosses the boundary of P at least $k+1$ times to reach some point of P; thus, the point is not k-visible to x, and $x \notin V_s$. □

For points x on the boundary of wedges A and B, we can derive a lemma similar to Lemma 5 that uses a clipping list incorporating edges of $E(A)$ and $E(B)$; we omit the details. These lemmas then imply that the boundary of V_s is a union of subsets of wedges, where each subset is either unbounded, or is bounded by *p-edges*: edges of P with s to their left. These subsets are bounded on the sides by *r-edges*, which lie on lines through s. If the vertices of P are not in general position, then r-edges can induce 'cracks' in the kernel; see Figure 5.

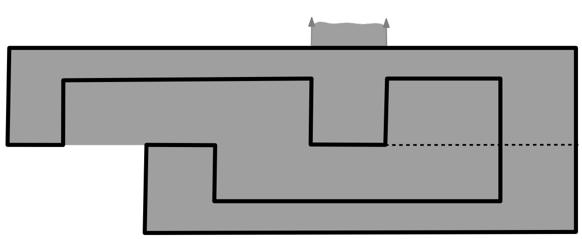

Figure 5: Shaded region is $M^4(P)$, dotted line is a crack

Lemma 6 *The v-region for a vertex of a polygon P with n vertices has $O(n)$ complexity, and can be constructed in $O(n \log n)$ time.*

Proof. Each v-region has $O(n)$ wedges, and by Lemma 5 each wedge is bounded by at most three segments (or rays); hence a v-region has $O(n)$ size. To construct a v-region, we use a sweep line algorithm [2]. The sweep line rotates around s, and stops when it encounters a polygon vertex. Active lists maintain the clipping lists for the current wedge. At each event point, the appropriate boundary p-edge and r-edge can be found in $O(\log n)$ time; we omit the details. If a suitable tree structure (e.g., [4]) is used for the event queues and active lists, a v-region can be generated in $O(n \log n)$ time. □

Theorem 7 *The k-kernel of a polygon P of n vertices can be constructed in $O(n^2 \log n + \kappa)$ time, where κ is the number of intersections between edges of the v-regions of P.*

Proof. We first use the algorithm given in the proof of Lemma 6 to construct, in $O(n^2 \log n)$ time, the v-regions for the vertices of P. Next, we construct the trapezoidal decomposition of the edges of these v-regions. This can be done in $O(n^2 \log n + \kappa)$ (deterministic) time [3], though a more practical randomized algorithm with the same (expected) running time exists [7]. Finally, we perform a linear traversal of this decomposition to find the edges bounding the common intersection of the n v-regions, which (by Theorem 4) are the edges bounding M. The running time of the complete algorithm is thus dominated by the time spent in the second step. It is worst-case optimal, since κ can be $\Omega(n^4)$, matching the lower bound of Theorem 3. □

4 Complexity of the 2-kernel

There exist polygons whose 2-kernels have quadratic complexity [1]. In this section we show that no polygon has a 2-kernel with more than quadratic complexity.

By Theorem 4, the boundary of M is some number of p-edges and r-edges. Since every vertex of M is the intersection of two lines that are coincident with p-edges or r-edges, it suffices to show that there are a linear number of these lines.

Since there are n edges of P, there are at most a linear number of lines containing p-edges, as well as r-edges collinear with edges of P (it can be shown that this includes cracks). If we ignore symmetric cases, and categorize an r-edge by the orientation of the polygon edges and vertices that intersect the line containing the r-edge, then each remaining r-edge is one of the three types of Figure 6.

Each of these r-edges, r, is associated with two vertices, u and v. Both u and v are convex in type (1) and reflex in type (2). In type (3), u is convex, v is reflex, and an additional *parity* edge[3] of P crosses the line containing r between u and v.

Consider type (1). Point x is a point interior to M, arbitrarily close to r. Now, suppose some vertex v' of P, together with u, induces a second r-edge r' of type (1) (and an analogous point x'). We can assume, without loss of generality, that v' is right of $\overrightarrow{uv}$; see Figure 7. Suppose some edge of P crosses ray $\overrightarrow{uv}$. Then, to satisfy parity, there must be two such edges crossing $\overrightarrow{uv}$; but then the ray from x through v will cross the boundary of P four times, implying $x \notin M$, a contradiction. By

[3]We can think of these edges as enforcing a parity condition: the polygon edges that cross a particular line, when ordered by crossing position along the line, will alternate between crossing from right to left and crossing from left to right.

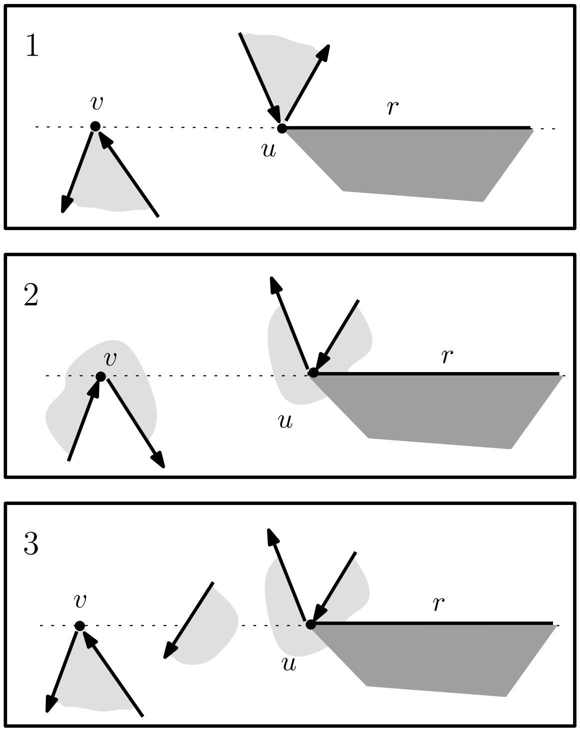

Figure 6: Types of r-edge (M is shaded)

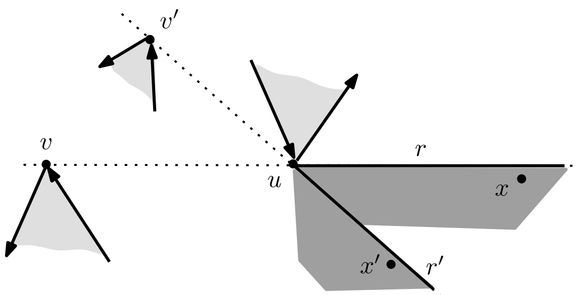

Figure 7: Type (1) r-edge

a similar argument, we can show that no edge of P can cross ray $\overrightarrow{uv'}$, otherwise $x' \notin M$. We now have a contradiction, since P is no longer connected (e.g., there is no path from v' to u). Thus v is the only vertex inducing a type (1) r-edge with u.

Let us examine type (2). Suppose some vertex v' of P, together with u, induces a second r-edge r' of type (2). We can assume v' is right of $\overrightarrow{uv}$; see Figure 8. We can use an argument similar to that for the type (1) edge to show that no additional edges of P can cross rays $\overrightarrow{uv}$ or $\overrightarrow{uv'}$, or segment $\overline{xu}$. Hence $x \in P$. Consider ray $\overrightarrow{xv'}$. Since $\overline{xu} \subset P$ and $\overline{uv'} \subset P$, we cannot have $\overline{xv'} \subset P$, otherwise u is on the boundary of a hole in P. This implies that $\overrightarrow{xv'}$ crosses the boundary of P at least five times, so that $x \notin M$, a contradiction.

If r is of type (3), then suppose u and some vertex

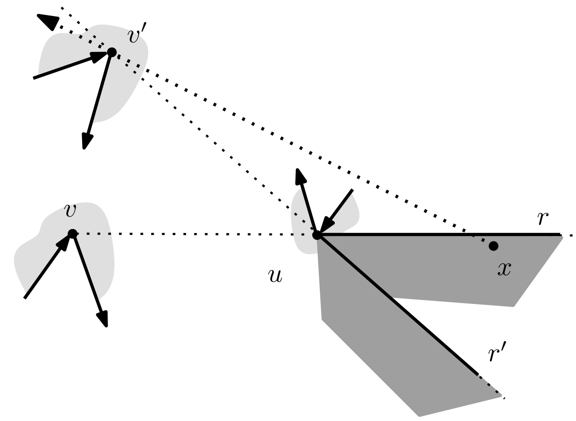

Figure 8: Type (2) r-edge

v' of P induce a second r-edge r' of type (3). We can assume v' is right of $\overrightarrow{uv}$; see Figure 9. Let t and t' be

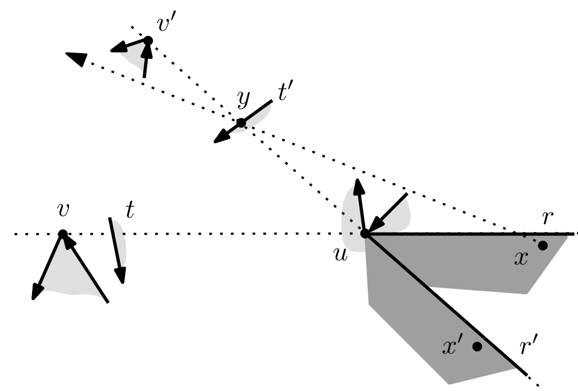

Figure 9: Type (3) r-edge

the parity edges associated with the two r-edges. We can use an argument similar to that for the type (1) edge to show that no additional edges of P can cross rays $\overrightarrow{uv}$ or $\overrightarrow{uv'}$, or segment $\overline{xu}$. Let y be the point where edge t' crosses ray $\overrightarrow{uv'}$. We can use an argument similar to that for the type (2) edge to show that ray $\overrightarrow{xv'}$ must cross the boundary of P at least twice before crossing it at y. We can also show that it must cross the boundary at least twice after y, for v' to remain connected to P. Hence $\overrightarrow{xy}$ crosses the boundary of P at least five times, implying $x \notin M$.

We have thus shown that each vertex of P can play the role of vertex u in (including symmetric cases) at most $O(1)$ r-edges of types (1), (2), or (3). This implies that there are $O(n)$ of these r-edges, lying on $O(n)$ distinct lines. We therefore conclude:

Theorem 8 *$M^2(P)$ has $O(n^2)$ complexity.*

We leave as an open problem whether or not there exist polygons whose 3-kernels have greater than quadratic complexity.

5 k-Convexity

We now show how the k-convexity of a polygon can be determined by examining its v-regions.

Lemma 9 *It is possible to determine if the v-region for a vertex s of a polygon P contains P in $O(n \cdot \min(1 + k, \log n))$ time.*

Proof. We present two algorithms for determining if the v-region of s contains P, which when run in parallel, yield the stated running time. They are motivated by the following insight: to determine if $P \subseteq V_s$, only the size of a clipping list is significant, not its elements, since some $x \in P$ will lie outside of V_s if and only if there exists some $(k+3)^{\text{rd}}$ element of a clipping list of a wedge of s.

The first algorithm is simply that of Lemma 6, modified so that if a clipping list for a wedge ever has more than $k + 2$ edges, it returns false; otherwise, it returns true. Its running time is $O(n \log n)$.

The second algorithm performs ccw traversal of P, starting from s, and uses a doubly-linked circular list of nodes to determine the maximum number of crossings of any line through s. There are two types of nodes: vertex nodes, which are ordered by the vertex's polar angle around s, and edge nodes, which connect neighboring vertex nodes. Each node has a dual whose angle is offset by π. The nodes include a crossing count, and the sum of the crossing counts for a primal / dual node pair represents the number of crossings that a line through the node's vertex or edge and s will make with that portion of the boundary of P traversed so far.

Initially, there are four vertex nodes, corresponding to the two vertices incident with s and their duals, plus four connecting edge nodes; see Figure 10. The algorithm traverses edges of P, maintaining pointers to the current primal and dual nodes, and moves ccw or cw around the node list, depending upon the direction of the current edge of P with respect to s (for ease of exposition, we assume no edges not incident with s lie on rays from s). The crossing count of the primal is incremented every time the node is traversed. As each new vertex of P is reached, the current edge node is split and primal and dual nodes for the new vertex are inserted. If the vertex represents a change in ccw / cw direction, the crossing count of the primal vertex is incremented, in accordance with Figure 1.

If the sum of the crossing counts for a primal / dual pair ever exceeds $k + 2$, then this is evidence of a pair of points of P that are not mutually k-visible, and the algorithm returns false; otherwise, when the traversal is complete, it returns true. Observe that (i) each traversal step increments some node's crossing count; (ii) the algorithm halts if any such count exceeds $k + 2$; (iii) each node can be traversed, and new nodes can be inserted, in constant time; and (iv) at most $O(n)$ nodes

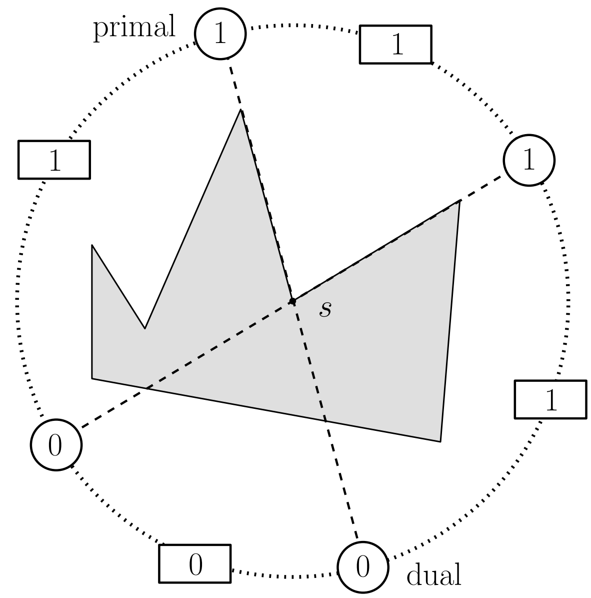

Figure 10: Initial vertex (circle) and edge (rectangle) nodes, with crossing counts

are created in total. Hence the algorithm performs at most $O(n \cdot (k + 1))$ steps. $\square$

By Theorem 4, polygon P is k-convex iff every v-region of P contains P. Hence, by applying the algorithm of Lemma 9 (which requires $O(n)$ space) to each vertex of P, we get the following result.

Theorem 10 *k-convex polygons can be recognized in $O(n^2 \cdot \min(1 + k, \log n))$ time and $O(n)$ space.*

Observe that if k is fixed, k-convex polygons can be recognized in $O(n^2)$ time.

Acknowledgment

The authors would like to thank Maarten Löffler for helpful discussions, and the anonymous reviewers for their comments.

References

[1] O. Aichholzer, F. Aurenhammer, F. Hurtado, P. Ramos, and J. Urrutia. Two-convex polygons. In *Proc. 25th European Workshop on Computational Geometry EuroCG'09*, pages 117–120, Brussels, Belgium, 2009.

[2] J. L. Bentley and T. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Computers*, 28(9):643–647, 1979.

[3] B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *J. ACM*, 39(1):1–54, 1992.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to algorithms, second edition, 2001.

[5] J. A. Dean, A. Lingas, and J. R. Sack. Recognizing polygons, or how to spy. *The Visual Computer*, 3(6):344–355, 1988.

[6] D. T. Lee and F. P. Preparata. An optimal algorithm for finding the kernel of a polygon. *J. ACM*, 26(3):415–421, 1979.

[7] K. Mulmuley. A fast planar partition algorithm. I. In *Foundations of Computer Science, 1988., 29th Annual Symposium on*, pages 580–589, Oct 1988.