An Output-Sensitive Algorithm for Computing the s-Kernel*

Leonidas Palios†

Abstract

Two points p, q of an orthogonal polygon P are s-visible from one another if there exists a staircase path (i.e., an x- and y-monotone chain of horizontal and vertical line segments) from p to q that lies in P. The s-kernel of P is the (possibly empty) set of points of P from which all points of P are s-visible.

We are interested in the problem of computing the s-kernel of a given orthogonal polygon (on n vertices) possibly with holes. The problem has been considered by Gewali [1] who described an $O(n)$-time algorithm for orthogonal polygons without holes and an $O(n^2)$-time algorithm for orthogonal polygons with holes. The problem is a special case of the problem considered by Schuierer and Wood [5], whose work implies an $O(n)$-time algorithm for orthogonal polygons without holes and an $O(n \log n + h^2)$-time algorithm for orthogonal polygons with $h \geq 1$ holes.

In this paper, we give a simple output-sensitive algorithm for the problem. For an n-vertex orthogonal polygon P that has h holes, our algorithm runs in $O(n + h \log h + k)$ time where $k = O(1 + h^2)$ is the number of connected components of the s-kernel of P. Additionally, a modified version of our algorithm enables us to compute the number k of connected components of the s-kernel in $O(n + h \log h)$ time.

Keywords: s-kernel, visibility, orthogonal polygon, output-sensitive algorithm.

1 Introduction

A polygon is *orthogonal* if its edges are either horizontal or vertical; an edge e of such a polygon is a N-edge (S-edge, E-edge, and W-edge, resp.) if the outward-pointing normal vector to e is directed towards the North (South, East, and West, resp.); see Figure 1(a). Of particular importance are the *dents*, i.e., edges whose endpoints are reflex vertices of the polygon, characterized as N-dents, S-dents, E-dents, and W-dents (see Figure 1(a)); the dents are a measure of non-convexity of an orthogonal polygon.

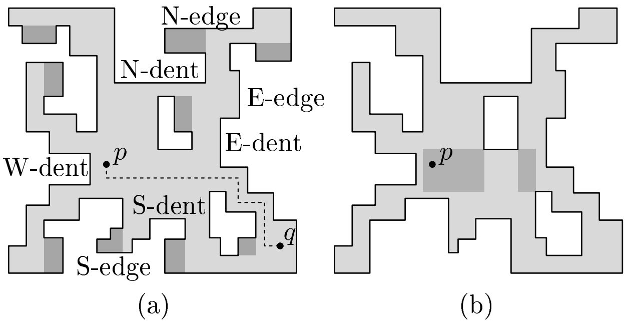


Figure 1: (a) Illustration of the main definitions (the portions of the polygon not s-visible from p are shown dark); (b) the s-visibility polygon of p, which is an s-star, with its s-kernel shown darker.

A set of points is *x-monotone* (*y-monotone*, resp.) if its intersection with any line perpendicular to the x-axis (y-axis, resp.) is a connected set. A *staircase path* is a chain of horizontal and vertical segments that is both x- and y-monotone.

Then, two points p, q of an orthogonal polygon P are *s-visible* from one another if there exists a staircase path from p to q that lies in P (Figure 1(a) shows two such points p and q). The set of points that are s-visible from a point p form the *s-visibility polygon* of p. The *s-kernel* of P is the (possibly empty) set of points of P whose s-visibility polygon is equal to P, i.e., the set of points from which all points of P are s-visible (the s-kernel of the orthogonal polygon in Figure 1(b) is shown darker); note that the s-kernel may be disconnected. An orthogonal polygon is an *s-star* if it has non-empty s-kernel. The orthogonal polygon in Figure 1(b) is an s-star; as can be seen in the figure, an s-star may have holes.

Visibility problems are closely related to reachability and to covering problems. The s-kernel of a polygon is the set of points from which all other points of the polygon can be reached by means of x- and y-monotone paths. So, if a robot restricted to move parallel to the coordinate axes is considered to "guard" a point p in an orthogonal polygon provided that it can get to p along a monotone path, then the polygons that can be "guarded" are those with non-empty s-kernel. Additionally, because the s-stars may be highly non-convex

* This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALIS UOA (MIS 375891) - Investing in knowledge society through the European Social Fund.

†Department of Computer Science and Engineering, University of Ioannina, Greece, palios@cs.uoi.gr

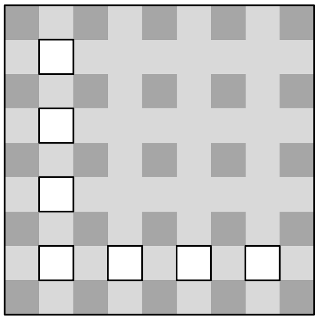


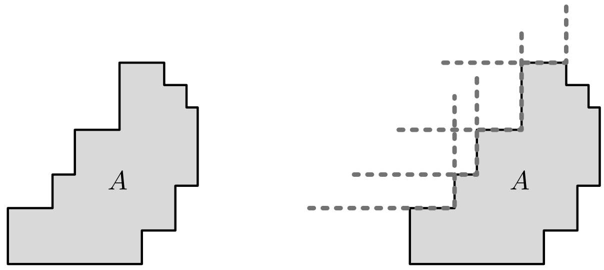
Figure 2: An orthogonal polygon with $\Theta(n)$ holes whose s-kernel (shown darker) has $\Theta(n^2)$ size.

(see Figure 1(b)), a minimum cover of an orthogonal polygon using s-stars (see [3] for an algorithm) is expected to involve a smaller number of pieces compared to other minimum covers. (Note also that in the usual sense of visibility, the kernel of a polygon with holes is empty and that the kernel of an n-vertex polygon can be computed in $O(n)$ time [2].)

Gewali [1] has considered the problem of computing the s-kernel of an orthogonal polygon; he described an $O(n)$-time algorithm for an orthogonal polygon without holes and an $O(n^2)$-time algorithm for orthogonal polygons with holes where n is the number of vertices of the polygon. He also showed that the latter algorithm is worst-case optimal since the s-kernel of an orthogonal polygon with holes may be of $\Theta(n^2)$ size; Figure 2 shows an orthogonal polygon with $\Theta(n)$ holes whose s-kernel has $\Theta(n^2)$ size [1]. Gewali used this result to give an $O(n \log n)$-time algorithm for recognizing whether an orthogonal polygon with holes is an s-star.

Schuierer and Wood [5] studied the notion of $\mathcal{O}$-visibility, that is, visibility along a set $\mathcal{O}$ of orientations and gave an $O(n \log |\mathcal{O}|)$-time algorithm for the computation of the $\mathcal{O}$-kernel of an orthogonal polygon without holes and an $O(n(\log |\mathcal{O}| + \log n) + h(|\mathcal{O}| + h))$-time algorithm for polygons with h holes, respectively. Their algorithms imply $O(n)$-time and $O(n \log n + h^2)$-time algorithms for the s-kernel of orthogonal polygons without holes and of orthogonal polygons with $h \geq 1$ holes, respectively.

In this paper, we present a simple output-sensitive $O(n + h \log h + k)$-time and $O(n)$-space algorithm for computing the s-kernel of an orthogonal polygon having n vertices, $h \geq 0$ holes, and an s-kernel consisting of k connected components; as we will see $k = O(1 + h^2)$. The algorithm also enables us to count the number k of connected components of the s-kernel of such a polygon in $O(n + h \log h)$ time using $O(n)$ space (i.e., without computing the s-kernel), and thus we can determine if an orthogonal polygon is an s-star in the same time and space complexity.



(a) (b)

Figure 3: (a) An orthogonal polygon A that is orthogonally convex; (b) some of the quadrants whose union is equal to the complement of A.

2 Theoretical Framework

For an edge e of an orthogonal polygon P, let D_e be a small enough disk centered at the midpoint of e; we define the *in-halfplane* of e as the *closed* halfplane that is defined by the line supporting e and contains the portion of D_e that lies in P.

An orthogonal polygon is *orthogonally convex* if it is both x-monotone and y-monotone. For simplicity and since we deal with orthogonal polygons, in the following, an orthogonally convex orthogonal polygon will be referred to as "orthogonally convex polygon." Clearly, an orthogonally convex polygon cannot have dents. The reverse also works, and we have:

Observation 1 *An orthogonal polygon is orthogonally convex if and only if it has no dents.*

Therefore, the boundary of an orthogonally convex polygon consists of x- and y- monotone chains connecting the leftmost edge of the polygon, to the uppermost edge, to the rightmost edge, to the bottommost edge, and back to the leftmost edge (see Figure 3(a)); any one of these chains may degenerate to a single point. Moreover, it is important to observe that the following lemma holds.

Lemma 1 *Let A be an orthogonally convex polygon having n vertices. Then, the complement of A can be expressed as the union of* $\Theta(n)$ open *quadrants.*

The lemma follows from the fact that the complement of an orthogonally convex polygon is equal to the union of as many open quadrants as the polygon's reflex vertices (for a reflex vertex, the corresponding quadrant is the complement of the union of the in-halfplanes of the edges incident on the reflex vertex) plus 4 more (one for each of the leftmost, topmost, rightmost, bottommost edge); Figure 3(b) shows the quadrants belonging to the complement of an orthogonally convex polygon that are associated with the boundary chain from the leftmost

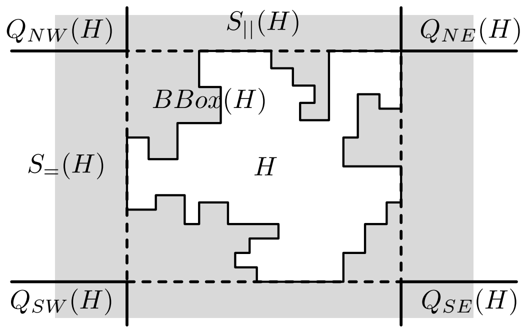


Figure 4: Illustration of $BBox(H)$, $S_{=}(H)$, $S_{||}(H)$, $Q_{NW}(H)$, $Q_{NE}(H)$, $Q_{SE}(H)$, and $Q_{SW}(H)$ for a hole H.

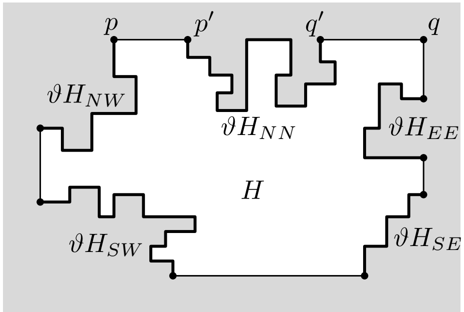


Figure 5: Illustration of the boundary subchain notation for a hole H (the subchain ϑH_{NE} is point q; no $\vartheta H_{WW}, \vartheta H_{SS}$ exist).

to the topmost edge (the remaining three chains contribute additional quadrants in a similar fashion). As a result, the total number of quadrants is nearly half the number of vertices of the polygon.

2.1 The s-kernel of orthogonal polygons without holes

The algorithm of Gewali [1] computes the s-kernel of an orthogonal polygon P without holes by intersecting P with the in-halfplanes of the lowermost N-dent, the rightmost W-dent, the topmost S-dent, and the leftmost E-dent. This implies the following result.

Lemma 2 *Let P be an orthogonal polygon without holes that has n vertices. The s-kernel of P is an orthogonally convex polygon of $O(n)$ size.*

2.2 Notation for orthogonal polygons with holes

Let D be an orthogonal polygon or a hole in an orthogonal polygon. Then, we define:

ϑD : the boundary of D;

$BBox(D)$: the smallest axes-aligned rectangle containing D.

Additionally, for a hole H, we have:

$S_{=}(H)$: the smallest *open* horizontal strip containing the interior of H;

$S_{||}(H)$: the smallest *open* vertical strip containing the interior of H;

$Q_{NW}(H)$: the *closed* axes-aligned quadrant that is the complement of the union of the interiors of the in-halfplanes of the top and left edges of the rectangle $BBox(H)$ (see Figure 4) — similarly, we define $Q_{NE}(H)$, $Q_{SW}(H)$, and $Q_{SE}(H)$;

ϑH_{NW} : the part of the boundary of H in counterclockwise direction from the leftmost among the points of H with maximum y-coordinate to the topmost among the points of H with minimum x-coordinate (see Figure 5) — similarly, we define ϑH_{NE}, ϑH_{SW}, and ϑH_{SE};

ϑH_{NN} : let p, q be the leftmost and rightmost, resp., vertices of H with maximum y-coordinate; if p, q are adjacent in H then no ϑH_{NN} exists; otherwise, if p' (q', resp.) is the other endpoint of the horizontal edge incident on p (q, resp.), ϑH_{NN} is the part of the boundary of H connecting p' and q' after the edges pp' and qq' have been removed (see Figure 5) — similarly, we define ϑH_{WW}, ϑH_{SS}, and ϑH_{EE}.

The following lemma provides important properties of the s-kernel of orthogonal polygons with holes.

Lemma 3 *Let H be a hole of an orthogonal polygon P. Then:*

(i) No point of the strips $S_{=}(H)$ and $S_{||}(H)$ belongs to the s-kernel of P.

(ii) If ϑH_{NW} is not a single point, then no point of the quadrant $Q_{SE}(H)$ belongs to the s-kernel of P. Moreover:
if ϑH_{NW} contains a S-dent or an W-dent, then no point of the quadrant $Q_{SW}(H)$ belongs to the s-kernel of P (see Figures 6 and 7);
if ϑH_{NW} contains a N-dent or an E-dent, then no point of the quadrant $Q_{NE}(H)$ belongs to the s-kernel of P;
if ϑH_{NW} contains a N-dent or an W-dent, then no point of the quadrant $Q_{NW}(H)$ belongs to the s-kernel of P.
Similar results hold for the boundary subchains ϑH_{NE}, ϑH_{SW}, and ϑH_{SE}.

(iii) If the boundary of H contains a subchain ϑH_{NN}, then no point of the quadrants $Q_{SW}(H) \cup Q_{SE}(H)$

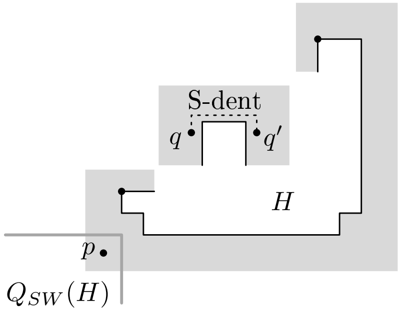


Figure 6: If ϑH_{NW} contains a S-dent, then no point of the quadrant $Q_{SW}(H)$ belongs to the s-kernel.

belongs to the s-kernel of P. Moreover:
if ϑH_{NN} contains a N-dent or an E-dent, then no point of the quadrant $Q_{NE}(H)$ belongs to the s-kernel of P;
if ϑH_{NN} contains a N-dent or an W-dent, then no point of the quadrant $Q_{NW}(H)$ belongs to the s-kernel of P.
Similar results hold for the boundary subchains ϑH_{WW}, ϑH_{SS}, and ϑH_{EE}.

The fact that if ϑH_{NW} contains a S-dent, then no point of the quadrant $Q_{SW}(H)$ belongs to the s-kernel of P (statement (ii) of Lemma 3) follows from the fact that there cannot exist x- and y-monotone paths from any point p of $Q_{SW}(H)$ to both points q, q' on either side of the S-dent; see Figure 6. Figure 7 shows examples of subchains ϑH_{NW} containing a S-dent but no W-dents (at left) and an W-dent but no S-dents (at right).

Lemma 3 implies that for a hole H of the given orthogonal polygon P, points of the s-kernel of P belong to all, some, or none of the four quadrants $Q_{NW}(H)$, $Q_{NE}(H)$, $Q_{SW}(H)$, and $Q_{SE}(H)$.

3 Computing the s-Kernel

Let P be an orthogonal polygon. In [5], the s-kernel of an orthogonal polygon P with h holes is computed as the intersection of the s-kernel A of P after having ignored the holes in P with the *external* s-kernels of all of P's holes. However, as the external s-kernel of each hole contains a horizontal and a vertical strip, the intersection of the external s-kernels may result to computing a partial s-kernel of quadratic (in h) size, most of which may be clipped in the end. So, in order to get a faster algorithm, we need to avoid this. Hence, we process the horizontal strips $S_{=}(\ )$ of the holes separately, computing the horizontal "in"-strips, i.e., the horizontal strips that form the complement of the strips $S_{=}(\ )$; these strips thus contain the entire s-kernel of P (see Lemma 3(i)). We work similarly for the vertical strips $S_{||}(\ )$. Next, we clip the complement of the union U_Q of all the quadrants not containing points of the s-kernel

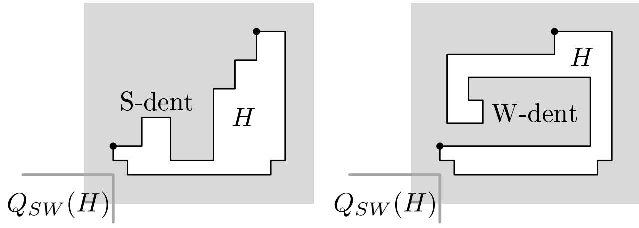


Figure 7: No point of the quadrant $Q_{SW}(H)$ belongs to the s-kernel if ϑH_{NW} contains: (left) a S-dent or (right) an W-dent.

(resulting from the holes as described in Lemma 3(ii) and (iii)) about the polygon A. Finally, we intersect the clipped complement of U_Q with the vertical and horizontal "in"-strips. A detailed description of the algorithm is given in Algorithm s-Kernel below.

Algorithm s-Kernel(P)
Input : an orthogonal polygon P possibly with holes
Output: the s-kernel of P

---

1. compute the s-kernel A of the orthogonal polygon bounded only by P's outer boundary component;
   if P has no holes
   then return A as the s-kernel of P;
   exit;
   let $x_{min}, x_{max}, y_{min}, y_{max}$ be the extreme values of x- and y-coordinates of the bounding rectangle $BBox(A)$ of A;

2. process the holes of P to determine the (open) strips and (closed) quadrants that do not contain points of the s-kernel of P (see Lemma 3);
   if all 4 quadrants $Q_{NW}(H)$, $Q_{NE}(H)$, $Q_{SW}(H)$, $Q_{SE}(H)$ of a hole H do not contain points of the s-kernel of P
   then print("The s-kernel of P is empty.");
   exit;
   let $\mathcal{C}_{=}$ ($\mathcal{C}_{||}$, $\mathcal{C}_Q$, resp.) be the set of horizontal strips (vertical strips, quadrants, resp.) not containing points of the s-kernel of P;

3. {*process the strips in $\mathcal{C}_{=}$ and $\mathcal{C}_{||}$*}
   compute the union of the horizontal strips in $\mathcal{C}_{=}$, clip it about the range $[y_{min}, y_{max}]$, and store it in a y-ordered array $M_{=}$ of alternating *closed* "in"-strips (containing points of the s-kernel) and *open* "out"-strips (not containing points of the s-kernel); work similarly for the vertical strips in $\mathcal{C}_{||}$ using the range $[x_{min}, x_{max}]$, producing an x-ordered array $M_{||}$;

4. {*process the quadrants in $\mathcal{C}_Q$*}
   compute the union U_Q of all the quadrants in $\mathcal{C}_Q$, and clip its complement about the boundary of the

polygon A computed in Step 1;
if the clipped complement of the union U_Q of the quadrants in $\mathcal{C}_Q$ is empty
then print("The s-kernel of P is empty.");
exit;

5. **for** each polygon B_j in the clipped complement of U_Q in y-order **do**
   for each horizontal "in"-strip I intersecting B_j in y-order **do**
   compute the boundary $\vartheta B_j(I) = \vartheta B_j \cap I$;
   locate a leftmost point of $\vartheta B_j(I)$ in the vertical strips array $M_{||}$;
   walk on $\vartheta B_j(I)$ and in $M_{||}$ until a rightmost point of $\vartheta B_j(I)$ is found, printing each polygon (if any) contributed by $B_j \cap I$ and each "in"-strip of $M_{||}$;

(Note that the clipped complement of the union U_Q at the completion of Step 4 does not contain its entire boundary; it contains the edges that resulted from the clipping about A but it does not contain the edges that resulted from the quadrants in $\mathcal{C}_Q$.)

The correctness of Algorithm s-Kernel follows from Lemma 3 and the fact that the s-kernel of P indeed is the intersection of polygon A (see Step 1) with the complement of the union of the collected strips and quadrants from the holes of P.

Time and Space Complexity. Let n and h be the number of vertices and holes of the input orthogonal polygon P. In the following lemma, we show that the complement of the union of axes-aligned quadrants has some very interesting properties; two polygons are *horizontally* (*vertically*, resp.) *separated* if no horizontal (vertical, resp.) line intersects both them.

Lemma 4 *(i) Each halfline bounding a quadrant in $\mathcal{C}_Q$ contributes at most one edge to the polygons forming the complement of the union U_Q of all the quadrants in $\mathcal{C}_Q$.*

(ii) The complement of U_Q consists of $O(h)$ orthogonally convex polygons that are horizontally and vertically separated and have $O(h)$ total size.

(iii) The clipped complement of U_Q computed upon completion of Step 4 of Algorithm s-Kernel consists of $O(h)$ horizontally and vertically separated orthogonally convex polygons of $O(n)$ total size.

Lemma 4(iii) and the fact that the intersection of $O(h)$ horizontal strips with $O(h)$ vertical strips consists of $O(h^2)$ connected components of $O(h^2)$ total size imply the following corollary.

Corollary 5 *The s-kernel of an n-vertex orthogonal polygon that has h holes consists of $O(1+h^2)$ orthogonally convex polygons of $O(n+h^2)$ total size.*

The number of orthogonally convex polygons and the size of a s-kernel given in Corollary 5 are tight; a lower bound can be obtained by a generalization of the polygon in Figure 2.

The computation of the s-kernel in Step 1 takes $O(n)$ time [1] and so does the entire Step 1. Step 2 takes $O(n)$ time as well by traversing the boundary of each hole H of P, computing the subchains ϑH_{NW}, ϑH_{NW}, ϑH_{NE}, ϑH_{SW}, ϑH_{SE}, ϑH_{NN}, ϑH_{WW}, ϑH_{SS}, and ϑH_{EE}, determining whether they contain dents, and applying Lemma 3. The processing of the h horizontal strips in $\mathcal{C}_=$ in Step 3 can be completed in $O(h \log h)$ time by sorting them by non-decreasing bottom side and then processing them from bottom to top; similarly, the processing of the vertical strips in $\mathcal{C}_{||}$ takes $O(h \log h)$ time. In Step 4, we sort the quadrants in y-order in $O(h \log h)$ time and compute the right-bounding line of the union of quadrants $Q_{NW}(H_i)$ and $Q_{SW}(H_{i'})$ in $\mathcal{C}_Q$ and the left-bounding line of the union of quadrants $Q_{NE}(H_i)$ and $Q_{SE}(H_{i'})$ in $O(h)$ time. The complement of these unions is clipped about polygon A and by traversing their boundaries from top to bottom we compute the clipped complement of U_Q in $O(n)$ time. In total, Step 4 takes $O(n+h \log h)$ time. For Step 5, let t_j be the number of horizontal "in"-strips intersecting polygon B_j. Because the polygons in the clipped complement of U_Q are horizontally separated (Lemma 4(iii)), then any other polygon may be intersected only by the topmost or bottommost of these t_j "in"-strips. Then, the number of pairs of polygons and "in"-strips considered is $\sum_j t_j = \sum_j 2 + \sum_j (t_j - 2) = O(h)$ since the total number of polygons B_j (see Lemma 4(iii)) and the total number of "in"-strips are both $O(h)$. Thus, if the s-kernel of P has k connected components, Step 5 takes $O(n + h \log h + k)$ time by using binary search in the x-sorted array $M_{||}$ for locating leftmost points. Therefore:

Theorem 6 *Let P be an orthogonal polygon having n vertices and $h = O(n)$ holes. Algorithm s-Kernel computes the s-kernel of P in $O(n + h \log h + k)$ time using $O(n)$ space where k is the number of connected components of the s-kernel of P.*

4 Computing the Number of Components of the s-Kernel

Algorithm s-Kernel can be modified to help us compute the number k of connected components of the s-kernel of a given orthogonal polygon P; it suffices to modify Step 1 so that if P has no holes it returns 0 if A is empty and 1 otherwise, Steps 2 and 4 to return

0 if the s-kernel is found empty, and Step 5 as follows: for each polygon B_j and each horizontal "in"-strip I intersecting B_j, we compute a leftmost point a and a rightmost point z of the boundary of B_j in I, and locate them in the vertical strips array $M_{||}$ using binary search; then, depending on the indices of the strips to which a, z belong and whether they are "in"- or "out"-strips, we compute the number $\kappa(B_j, I)$ of "in"-strips (if any) between (and including) the strips of a and of z. The total number of components of the s-kernel of P is the sum of all the computed $\kappa(B_j, I)$.

The correctness of the modified algorithm follows immediately from the fact that for each polygon B_j and each horizontal "in"-strip I, each "in"-strip between (and including) the strips containing a and z contributes a separate component to the s-kernel of P. The complexity analysis of Step 5 of Algorithm s-Kernel and the fact that $\kappa(B_j, I)$ can be computed in constant time after the strips containing a and z have been determined imply that the modified Step 5 takes $O(n+h \log h)$ time. Recall that the number k of connected components of the s-kernel may be as large as $\Theta(1 + h^2)$; see Corollary 5.

Therefore, we have:

Theorem 7 *Let P be an orthogonal polygon having n vertices and $h = O(n)$ holes. The described modified algorithm computes the number of connected components of the s-kernel of P in $O(n + h \log h)$ time using $O(n)$ space.*

5 Recognizing s-Stars

The modified algorithm of Section 4 to recognize whether a polygon P is an s-star (i.e., its s-kernel consists of at least 1 component) or not. A simpler version that does not compute the number k of components simply checks in Step 5 whether a and z fall in the same vertical "out"-strip of $M_{||}$; if they don't, then there exists a point in $B_j \cap I$ belonging to the s-kernel of P and hence P is an s-star (the algorithm can be augmented to return such a point as a certificate of its decision). If the above condition for a, z does not hold for all polygons B_j and "in"-strips I, then clearly the s-kernel of P is empty, and hence P is not an s-star.

Theorem 8 *Let P be an orthogonal polygon having n vertices and $h = O(n)$ holes. It can be decided whether P is an s-star in $O(n + h \log h)$ time using $O(n)$ space.*

6 Concluding Remarks

In this paper, we presented a simple output-sensitive algorithm for computing the s-kernel of an orthogonal polygon possibly with holes. The algorithm runs in $O(n + h \log h + k)$-time using $O(n)$ space, where n and h are the numbers of vertices and holes, respectively, of the input polygon, and k is the number of connected components of the computed s-kernel. Modifications of our algorithm enable us to compute the number k of connected components and to recognize if an orthogonal polygon is an s-star in $O(n + h \log h)$ time using $O(n)$ space.

Schuierer and Wood [5] mention that Rawlins in his PhD thesis [4] showed that the computation of the kernel of a multiply connected polygon under restricted orientation visibility has a lower bound of $\Omega(n \log n)$. This may imply that our s-kernel algorithm is optimal.

It is interesting to investigate the complexity status of the s-star recognition problem, i.e., can there be an algorithm running in $o(n + h \log h)$ time or is there an $\Omega(n + h \log h)$ lower bound? Additionally, it would be interesting to study extensions of the problem to 3-dimensional space.

References

[1] L.P. Gewali. Recognizing s-Star Polygons. *Pattern Recognition* 28(7), 1019-1032, 1995.

[2] D.T. Lee and F.P. Preparata. An Optimal Algorithm for Finding the Kernel of a Simple Polygon. *J. ACM* 26, 415-421, 1979.

[3] R. Motwani, A. Raghunathan, and H. Saran. Covering Orthogonal Polygons with Star Polygons: the Perfect Graph Approach. *J. Comput. Systems Science* 40, 19-48, 1990.

[4] G. Rawlins. *Explorations in Restricted-Orientation Geometry.* PhD Thesis, University of Waterloo, 1987.

[5] S. Schuierer and D. Wood. Generalized Kernels of Polygons with Holes. *Proc. 5th Canadian Conf. on Computational Geometry*, 222-227, 1993.