Recognizing S-Star Polygons

Laxmi P Gewali
Department of Computer Science
University of Nevada, Las Vegas
4505 Maryland Parkway, Las Vegas, Nevada 89154

Abstract

We consider the problem of recognizing star-polygons under stair-case visibility (**s-visibility**). We present an algorithm for computing the s-kernel of a polygon in $O(n)$ time for simple orthogonal polygons and in $O(n^2)$ time for orthogonal polygons with holes; both complexities are optimal in the worst case. Finally, we report the main result of this paper: we show that even though the s-kernel of a polygon with holes may have $\Omega(n^2)$ components it is possible to recognize such polygons in $O(n \log n)$ time.

1 Introduction

We consider the problem of computing visibility polygon, the problem of computing kernel, and the problem of recognizing star polygons under stair-case visibility (**s-visibility**). One motivation for considering these problems is the usefulness of s-visibility for developing polygon covering algorithms. For example, the minimum star cover of a simple orthogonal polygon under s-visibility can be computed in $O(n^8)$ time [MRS90], although the same problem is NP-hard under standard visibility [O87]. Other motivation comes from the applicability of monotone paths in manufacturing [BT92], and s-visibility is based on monotonicity in two perpendicular directions.

Perhaps the main contribution of this paper is the demonstration that a s-star polygon, possibly containing holes, can be recognized in $O(n \log n)$ time even though its s-kernel can have $\Omega(n^2)$ components. Our results can be summerized as follows: (a) We present an $O(n)$ time algorithm for computing the s-visibility polygon from a point inside a simple orthogonal polygon. (b) For orthogonal polygons with holes, we show that s-visibility polygon from an interior point can be computed in $O(n \log n)$ time. We prove this time complexity to be optimal by linear time reduction from the sorting problem. (c) We give a simple construction for converting an instance of computing s-visibility from an exterior point to an instance of computing s-visibility from an interior point. (d) We present an algorithm for computing the s-kernel of orthogonal polygons. The algorithm computes the s-kernel in $O(n)$ time for simple orthogonal polygons and in $O(n^2)$ time for orthogonal polygons with holes. Both complexities are optimal in the worst case. (e) Finally, we show that s-star polygons with holes can be recognized in $O(n \log n)$ time.

2 Preliminaries

The input orthogonal polygon Q, whose edges are parallel to coordinate axes, is given as an ordered sequence of vertices $v_0, v_1, v_2, v_3, ..., v_{n-1}$, in the order in which they occur along the clockwise traversal of its boundary. A **stair-case path** is a path consisting of line segments parallel to the x or y coordinate axes such that any horizontal or vertical line intersect with at most one line segment of the path. Stair-case paths can be categorized into two types: (a) a **type I stair-case path** extends from south-west to north-east, and (b) a **type II stair-case path** extends from south-east to north-west (Figure 1a-b). Two points are said to be (**s-visible**) if they can be connected by a stair-case path without intersecting the exterior of the polygon. A polygonal path is **monotone** along a given direction d if any line perpendicular to d intersects with at most one line segment of the path. An orthogonal path having monotonicity along the x-axis direction is called an **x-monotone path.** Similarly, **y-monotone path** is defined. In term of monotonicity, a stair-case path can be viewed as an orthogonal path having monotonicity in both x-axis and y-axis directions. A maximal boundary chain of an orthogonal polygon which is also a type I stair-case path is referred to as a **type I boundary s-chain.** Similarly, a **type II boundary s-chain** is defined. In Figure 1e, chain $(v_8, v_9, ..., v_{14})$ is a type I boundary s-chain and chain (v_{17}, v_{18}, v_{19}) is a type II boundary s-chain. The boundary of an orthogonal polygon can be partitioned into the above two types of s-chains. A horizontal or vertical line segment inside the polygon with end points at its boundary is a **chord** of the polygon.

The visibility polygon from a point can be defined in terms of x-monotone path, y-monotone path, or stair-case path. The **x-monotone visibility polygon** (respectively, the **y-monotone visibility polygon**) from a point q is the set of points that can be connected to q by a x-monotone path (respectively, y-monotone path). Similarly, **s-visibility polygon** from a point is the set of points that can be connected to q by a stair-case path. Figure 1 illustrates the above three types of visibility polygons. Let $V_q^s(Q)$, $V_q^x(Q)$, and $V_q^y(Q)$ denote the s-visibility polygon, x-monotone visibility polygon, and y-monotone visibility polygon, respectively, from a point q inside polygon Q. Due to space limitation, we omit proofs and running examples in this extended abstract.

Lemma 1: [GG93] The s-visibility polygon $V_q^s(Q)$ is given by the intersection of the x-monotone visibility polygon $V_q^x(Q)$ and the y-monotone visibility polygon $V_q^y(Q)$. (Proof omitted)

Lemma 2: [GG93] $V_q^s(Q)$, the s-visibility polygon from a point q inside an orthogonal polygon Q, is an orthogonal polygon.(Proof omitted)

3 Algorithms for Computing s-Visibility Polygon

In this section we develop an algorithm for computing the s-visibility polygon from a point inside an orthogonal polygon. An overview of the algorithm can be stated as follows: (i) We first compute $R = V_q^x(Q)$, the x-monotone visibility polygon from point q for polygon Q. (ii) We then compute the y-monotone visibility polygon from point q for polygon R, which is precisely the required s-visibility polygon $V_q^s(Q)$ (Lemma 1).

Theorem 1: [GG93] The s-visibility polygon from a point inside a simple orthogonal polygon can be computed in $O(n)$ time. (Proof omitted)

The results of Theorem 1 can be generalized in a straightforward way to include polygons with holes. When holes are present trapezoidization can not be computed in $O(n)$ time. We need to use an $O(n \log n)$ time algorithm [FM84]. All other steps have the same time complexity as for the case of a polygon without holes and the total time complexity becomes $O(n \log n)$.

Theorem 2: The s-visibility polygon from a point inside a polygon with holes can be computed in $O(n \log n)$ time. (Proof omitted)

The time complexity stated in Theorem 2 is optimal, within a constant factor. This fact can be established by demonstrating that the sorting problem can be reduced in linear time to the problem of computing s-visibility polygon from a point inside a polygon with holes.

Theorem 3: Sorting is linear time transformable to the problem of computing the s-visibility polygon from a point inside a polygon with holes. Therefore, finding the s-visibility polygon from a point inside a polygon of n vertices with holes requires $\Omega(n \log n)$ time. (Proof omitted)

4 Computing s-Kernel and Recognizing s-Star Polygons

Under standard visibility, the **kernel** of a polygon Q is the set of points from which all points inside Q are visible. The kernel of a polygon may be empty or non-empty and a polygon with non-empty kernel is called a **star polygon**. A linear time algorithm for computing the kernel of a polygon under standard visibility has been reported [LP79]. We consider the star-shape property under s-visibility. An orthogonal polygon is a **s-star** polygon (or simply s-star) if it contains a point from which all other points inside the polygon can be connected by a stair-case path, and the set of such points is its **s-kernel.** shows examples of s-star polygons. It is interesting to note that whereas star polygons are always simple, s-star polygons may contain holes (Figure 2b). Also, although the kernel of a polygon is always connected and convex, a s-kernel need not be connected.

Imagine partitioning polygon Q into rectangles by extending its edges into its interior. Observe that the s-visibility polygons from points on the same rectangle are identical. Hence a straight forward way to recognize an s-star polygon is to compute the s-visibility polygon from a point on each rectangle and accept the polygon as s-star if the whole polygon is visible from at least one rectangle. Since there can be potentially $\Omega(n^2)$ rectangles, this approach requires $O(n^3)$ time, which is rather expensive. To develop an efficient algorithm we start with few definitions. If we traverse the boundary of a simple orthogonal polygon in the clockwise direction, keeping the interior to the right, then at the vertex of the polygon we either turn 90^0 right (outside corner) or 90^0 left (inside corner). An edge of an orthogonal polygon whose both ends are inside corners is referred to as a **dent** [RC87]. The direction of dent traversal gives its orientation which we indicate as **N, S, E,** and **W** dents. In Figure 4a, edges (a,b), (c,d), (e,f), and (g,h) are N, E, S, and W-dents, respectively. The s-kernel can be captured in terms of critical chords. An **east critical chord** is the chord passing through the left most E-dent; if there is no E-dent then the unique right most vertical edge is taken as the east critical chord. Similarly, **west critical chord, south critical chord**, and **north critical chord** are defined.

Lemma 3: A simple orthogonal polygon is an s-star polygon if and only if the following two conditions are satisfied: (a) The x-coordinate of the east critical chord is greater than the x-coordinate of the west critical chord, and (b) The y-coordinate of the north critical chord is greater than the y-coordinate of the south critical chord. (Proof omitted)

Theorem 4: The s-kernel of a simple orthogonal polygon can be computed in $O(n)$ time. (Proof omitted)

We next consider s-visibility in the exterior of the polygon. A simple orthogonal polygon is an **external s-star** if there exist a point from which all points in the exterior of the polygon are s-visible.

Theorem 5: An externally s-star simple orthogonal polygon can be recognized in $O(n)$ time. (Proof omitted)

We now consider the problem of computing the s-kernel of orthogonal polygons that may contain holes. For a polygon with holes we use the term "chord" to indicate only those horizontal or vertical line segments inside the polygon whose both end points are on the outer boundary of the polygon. This means that horizontal or vertical line segments inside the polygon with end points on the boundary of holes are not taken as chords. A **prime chord** is a chord that contains an edge of the polygon. Prime chords partition the polygon into rows and columns of rectangular blocks and each such block is referred to as an **elementary block**. The partitioning itself is referred to as **prime chord partitioning**. A row of elementary blocks is called a **horizontal corridor** if no block in the row encloses a hole; if some elementary block of the row encloses a hole then that row is called a **horizontal strip**. Similarly, **vertical corridor** and **vertical strip** are defined. We number rows from top to bottom. The i^{th} horizontal corridor and the i^{th} horizontal strip are denoted by C_i and H_i, respectively. The j^{th} elementary block in C_i and H_i are denoted by $C_{i,j}$

and $H_{i,j}$, respectively. A **corridor intersection** is an elementary block formed by the intersection of a vertical and horizontal corridor. Note that corridors and strips occur alternately. Corridor intersections are feasible regions to be a part of the s-kernel.

Lemma 4: An orthogonal polygon containing holes is s-star if and only if it is s-visible from one of the corridor intersections. (Proof omitted)

We may use Lemma 4 to check for the s-star shape property of polygon Q by computing the s-visibility polygon from a point on each corridor intersection and we report the polygon to be s-star if the s-visibility polygon from some corridor intersection is the entire polygon. The time complexity of the algorithm designed by employing this strategy is rather high. Since there can be $O(n^2)$ corridor intersections and the s-visibility polygon computation from a point takes $O(n \log n)$ time, the total time becomes $O(n^3 \log n)$.

To develop a faster algorithm we identify corridor intersections satisfying certain visibility properties. A corridor intersection $C_{i,j}$ is said to **cover** H_k if all points in H_k are s-visible from $C_{i,j}$. Let $S(i,j) = \{C_{i,j} \mid C_{i,j}$ is a cover for $H_j\}$. A corridor intersection is a **top covering rectangle** if all points in the polygon lying above it are s-visible from it. Similarly, a **bottom covering rectangle** is defined. We use T_i and B_i to denote the set of top covering rectangles and the set of bottom covering rectangles, respectively, in corridor C_i. Let *BT(Q)* and *TB(Q)* denote the corridor rows of Q containing the bottom most top covering rectangles and the top most bottom covering rectangles, respectively.

Lemma 5: An orthogonal polygon Q containing holes is s-star if and only if *BT(Q)* is not above *TB(Q)*. (Proof omitted)

Theorem 6: The s-kernel of an orthogonal polygon with holes can be computed in $O(n^2)$ time. (Proof omitted)

Examples can be constructed to show that an orthogonal polygon with holes may contain $\Omega(n^2)$ components. The $O(n^2)$ algorithm for computing s-kernel could be used to recognize s-star polygons with holes. It is interesting to develop sub-quadratic time complexity algorithms to recognize s-star polygons with holes, without having to output its s-kernel. The main bottle neck in this direction is the size of the planar graph formed by the polygon and the intersection of prime chords. The approach used to compute s-kernel is the computation of top covering rectangles (and bottom covering rectangles) for each horizontal corridor. Since there can be $O(n^2)$ top covering (or bottom covering) rectangles in total, it is not possible to come up with a sub-quadratic time algorithm by explicitly maintaining covering rectangles.

We next consider how to develop a sub-quadratic time complexity algorithm to recognize s-star polygons with holes. We define the **upper x-extent** (respectively, **lower x-extent**) of a polygon to be the upper horizontal edge (respectively, lower horizontal edge) of its bounding box. Our algorithm is based on computing the lower x-extents of top covering rectangles (and upper x-extents of bottom covering rectangles) without constructing an $O(n^2)$ planar graph. Without going into further detail (due to space limitation), we state the main result of this paper:

Theorem 7: S-Star polygon, possibly containing holes, can be recognized in $O(n \log n)$ time. (Proof omitted)

5 Discussions

We presented an optimum $O(n)$ time algorithm for computing the s-kernel of simple orthogonal polygons. We showed that the s-kernel of an orthogonal polygon with holes can be computed in $O(n^2)$ time. Even though the s-kernel of a polygon with holes may have $\Omega(n^2)$ components we showed that it is possible to recognize s-star polygons in $O(n \log n)$ time. The algorithm can be generalized to include non-orthogonal polygons without increasing complexity: reflex vertices play the role of dent lines and the algorithm generalizes naturally. It is not clear if the $O(n \log n)$ time algorithm to recognize s-star polygons with holes is optimal or not. It would be therefore interesting to obtain a faster recognition algorithm or to prove that the complexity is optimal.

References

[A85] T. Asano, "An Efficient Algorithm for Finding the Visibility Polygon for a Polygonal Region With Holes", *Transactions of IECE of Japan*, E 68(9), pp. 557-559, 1985.

[BT92] P. Bose and G. Toussaint, "Geometric and Computational Aspects of Injection Molding", Technical Report No. SOCS 92.16, School of Computer Science, McGill University, Canada, 1992.

[EA81] H. ElGindy and D. Avis, "A Linear Time Algorithm for Computing the Visibility Polygon From a Point", Journal of Algorithms, Vol. 2, pp. 186-197, 1981.

[FM84] A. Fournier and D. Y. Moutuno, "Triangulating Simple Polygons and Equivalent Polygons", *ACM Transactions on Graphics*, Vol. 3, No. 2, pp. 153-174, 1984.

[GG93] L. Gewali, and D. Glasser, "On Computing Stair-Case Visibility Polygon", Proceedings of the 1993 Systems Science Conference, Las Vegas, 1993.

[GN93] L. Gewali, and S. Ntafos, "Covering Grids and Orthogonal Polygons With Periscope Guards", *Computational Geometry: Theory and Applications*, Vol. 2, 1993, pp. 309-334, 1993.

[HA92] Y. K. Hwang and N. Ahuza, "Gross Motion Planning - A Survey", *ACM Computing Survey*, Vol. 24, No. 3, 1992, pp. 219-291.

[LP79] D. T. Lee and F. P. Preparata, "An optimal algorithm for finding the kernel of a simple polygon", *Journal of the ACM*, Vol. 26, 1979, pp. 415-421.

[MRS90] R. A. Motwani, Raghunathan and H. Saran, "Covering Orthogonal Polygons With Star Polygons: The Perfect Graph Approach", *Journal of Computer and Systems Sciences*, Vol. 40, 1990, pp. 19-48.

[O87] J. O,Rourke, "Art Gallery Theorem and Applications", Oxford University Press, 1987.

[PS85] F. P. Preparata, F. P. and M. I. Shamos, "Computational Geometry - An Introduction", Springer Verlag, 1985.

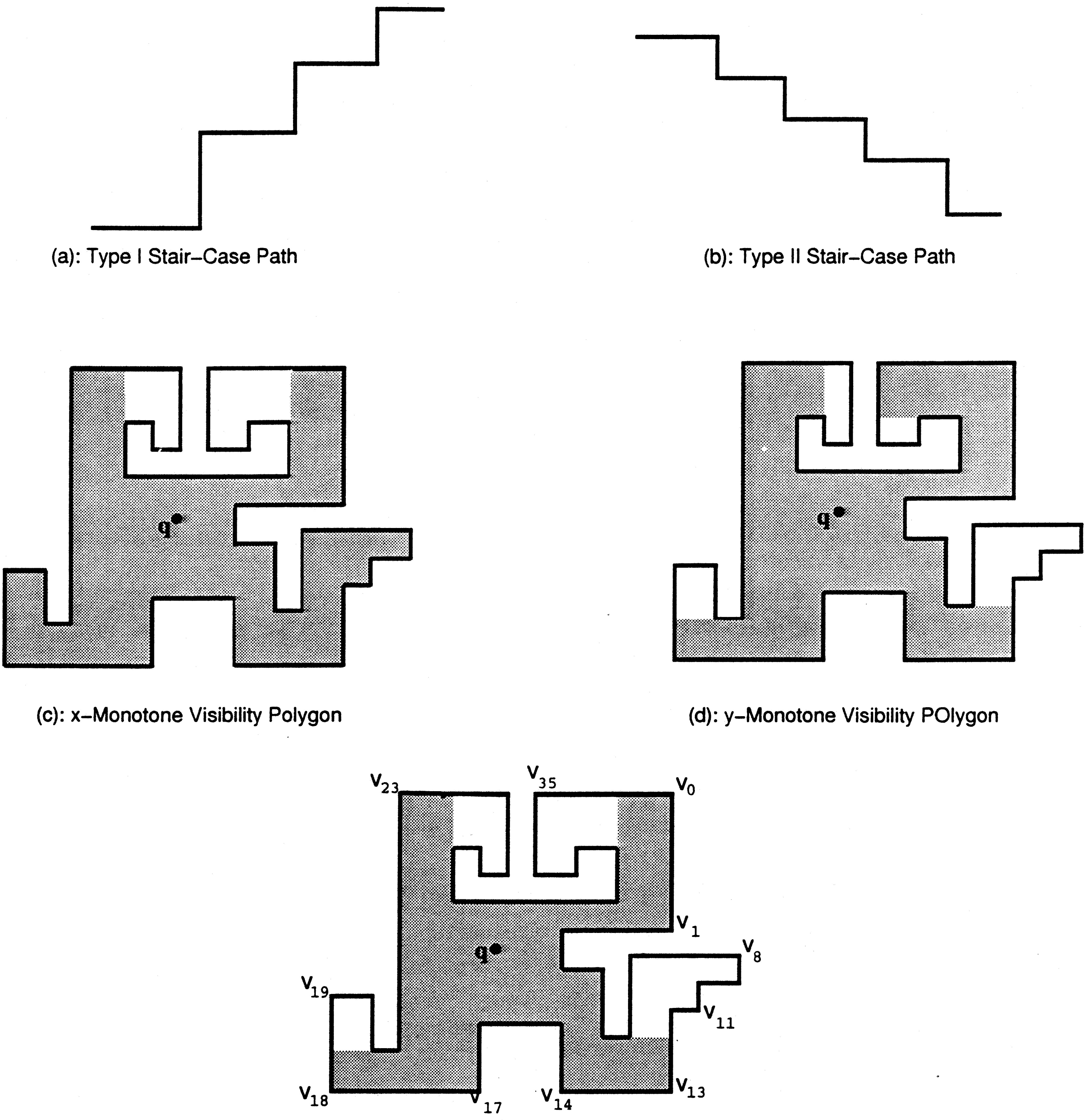
[RC87] R. Reckhow and J. Culberson, "Covering Simple Orthogonal Polygons With a Minimum Number of Orthogonally Convex Polygons", *Proceedings of the Third Annual Symposium on Computational Geometry* (1987), pp. 268-277.

[S92] T. Shermer, "Recent Results in Art Galleries", *Proceedings of The IEEE*, Vol. 80, No. 9, September 1992.

[SRW91] S. Schuierer, G. Rawlins, and D. Wood, "A Generalization of Stair-Case Visibility", *Proceedings of the Third Canadian Conference on Computational Geometry*, 1991.

[SW93] S. Schuierer and D. Wood, "Generalized Kernels of Polygons With Holes", *Proc. of The Fifth Canadian Conference on Computational Geometry*, 1993.

[T88] G. Toussaint, (Editor), Computational Morphology, North Holland, 1988.



(a): Type I Stair–Case Path

(b): Type II Stair–Case Path

(c): x–Monotone Visibility Polygon

(d): y–Monotone Visibility POlygon

(e): s–Visibility Polygon

Figure 1: Illustrating Stair–Case Path and s–Visibility Polygon