Visibility in Finitely Oriented Polygons (Extended Abstract)

*Vladimir Estivill-Castro** and *Venkatesh Raman**

Abstract

A polygon is said to be C-oriented if the set of orientations of its edges is a subset of a set C. We study visibility in C-oriented polygons under C-oriented rays and explore the combinatorial structure of their edge visibility graphs. We present efficient algorithms to compute the portions of the polygon visible to a point or an edge of the polygon. We also present an interesting art gallery theorem concerning the number of edges needed to cover an orthogonal polygon.

1 Introduction

The concept of visibility in computational geometry has proven useful for answering a variety of questions[8, 2]. Application areas like VLSI design, computer aided design, digital picture processing have traditionally placed heavy emphasis on orthogonally oriented objects. However, recent advances allow objects to have more than the usual two orientations and as a result interest has grown for studying objects formed with lines oriented in a fixed set of directions [5, 10, 13, 14].

The *orientation* of a line is the smaller of the two positive angles it makes with the positive x-axis. Let C be a finite set of orientations in a plane. A collection of lines, segments and rays is *C-oriented* if the set of orientations of the elements in the collection is a subset of C. Thus, we speak of C-lines, C-segments and C-rays to mean C-oriented lines, segments and rays respectively. Then, *C-oriented polygons* [4] (C-polygons) are polygons in which the orientation of each edge is in C. If $|C|$ is c, then the polygon is said to be *c-oriented.* For example, orthogonal polygons are 2-oriented where the two orientations in C are orthogonal.

Let P be a C-oriented polygon. We now define *C-oriented visibility.* We say that two points x and y of P are *visible* to each other if the segment joining them is C-oriented and does not intersect the exterior of P. This definition of visibility is the natural extension to orthogonal visibility explored by Booth and O'Rourke[9, Section 7.3]. Two edges e_1 and e_2 of P are *visible* to each other if there exist points $x \in e_1$ and $y \in e_2$ (x and y are not endpoints of e_1 and e_2) that are visible to each other in P. This definition of edge visibility corresponds to the well studied notion of (weak) edge visibility[1].

*Data Structuring Group, Department of Computer Science, University of Waterloo, WATERLOO, Ontario N2L 3G1, Canada.

Other definitions of visibility in C-oriented polygons have been proposed before in the context of C-convex sets, but none of them generalize orthogonal visibility. According to the visibility definition of Rawlins and Wood [11], a (non-end) point in a horizontal edge in an orthogonal polygon can see a (non-end) point of a vertical edge through a "stair line". Schuierer, Rawlins and Wood [12] use the convex hull operator (denoted by *C-hull*) in an abstract convexity space and redefine visibility: given points x and y in P, they see each other if $C\text{-}hull(\{x,y\}) \subset P$. In an orthogonal polygon, this definition allows $C\text{-}hull(\{x,y\})$ to be just $\{x,y\}$, thus allowing a (non-end) point in a horizontal edge to see a (non-end) point of a vertical edge. If we insist that *C-hull* should be connected, then their visibility definition coincides with ours and thus naturally generalizes orthogonal visibility. Throughout the following sections visibility refers to C-oriented visibility.

In the next section, we define and study some properties of the edge visibility graph of C-oriented polygons.[1] Section 3 outlines algorithms to construct portions of the polygon visible from a point or an edge of the polygon. Finally Section 4 presents an art gallery theorem for orthogonal polygons.

2 On the Edge Visibility Graph

The edge *visibility graph* of a polygon consists of a node for every edge in the polygon. There is an edge joining two nodes, if the two corresponding edges are visible to each other in the polygon. We say that a C-oriented polygon P is in *general position* if no two vertices can be connected by a C-segment interior to P. Equivalently, a C-oriented polygon with n vertices is in general position if the vertex visibility graph of the polygon (where visibility rays are C-segments between the vertices of the polygon never in the exterior of the polygon) is the cycle graph C_n. Booth and O'Rourke[9, Section 7.3] show that the edge visibility graph of an orthogonal polygon of n edges in general position consists of two disconnected trees with $n/2$ nodes each. We extend their result to the following theorem.

Theorem 2.1 *The edge visibility graph of a C-polygon with n edges in general position*

- *consists of two disconnected trees with $n/2$ nodes each, if $|C| = 2$.*
- *If $|C| > 2$, it is connected with at least one cycle, need not be biconnected, and any cut vertex (if exists) of the edge visibility graph divides the graph into two components, one of which is a single node.*

Furthermore, since any collection of lines, segments and rays in (one, two or) three orientations in the plane can be mapped onto another collection having the same incidence structure as the first but with (one, two or) three completely different orientations [10], we can generalize immediately all the results for orthogonal polygons [9, Section 7.3] to 2-oriented polygons.

[1] In this extended abstract we omit all proofs. They are available in the full version [3].

In general, the recognition problem for visibility graphs of simple polygons is open. A characterization of a class of graphs may be used to design efficient recognition algorithms. Recall that a graph is *realizable* if there is a polygon P such that G is the edge visibility graph of P. For finitely oriented geometry we say that a graph is c-realizable if there exists a set C of c orientations and a C-oriented polygon P such that G is the edge visibility graph of P. Clearly, any c-realizable graph is $(c+1)$-realizable, and any realizable graph with n nodes is c-realizable where c is $O(n^2)$. Considering finitely oriented polygons results in a hierarchy of graph classes as the following theorem shows.

Theorem 2.2 *For each $c \geq 2$, there is a graph G that is $(c+1)$-realizable but not c-realizable.*

This result follows directly from the following lemma. Observe that C-oriented visibility makes certain graphs, which are otherwise realizable, non-realizable.

Lemma 2.3 *The complete graph on $n+1$ nodes, is $(n+1)$-realizable, but not n-realizable.*

3 Visibility Algorithms

The notion of C-oriented visibility leads to two central algorithmic questions.

- Given a point x in a C-oriented polygon P, compute $V_C(x)$, the portion of P visible from x through C-rays. $V_C(x)$ consists of at most $2|C|$ line segments.
- Given an edge e (or a C-segment) in a C-oriented polygon P, compute $V_C(e)$ the portion of P visible from some point of P trough C-rays. The topological closure of $V_C(e)$ is a C-oriented polygon.

We can solve the first problem by preprocessing the polygon[9, Section 8.7] in $O(n \log\log n)$ time and building a $O(n)$ data structure which can answer queries of the following form in $O(\log n)$ time: given a point p in P and a direction θ, find the first edge of P hit by a θ-ray through p. Thus, performing a query for each direction gives a $O(n \log\log n + c \log n)$ algorithm. We obtain a linear algorithm by modifying Lee's linear algorithm [7, 6] for constructing the visibility region from a point inside a polygon. Notice that we do not require P to be C-oriented.

Theorem 3.1 *Let C be a given set of c orientations and P a simple polygon with n vertices. If the orientations are sorted, $V_C(x)$ for a point $x \in P$ can be computed in $O(n+c)$ time and $O(n+c)$ space in the worst case.*

To solve the second problem, one could use an algorithm to find the edge visibility region for general visibility and modify the result accordingly. Since the best known algorithm for this task (even for orthogonal polygons) is based on trapezoidization, it would require $O(n \log\log n)$ time. However, for 2-oriented polygons we can improve this to linear time.

Theorem 3.2 *Let C be a set of 2 orientations and P a simple C-polygon with n vertices. $V_C(e)$ for an edge $e \in P$ can be computed in $O(n)$ time and $O(n)$ space in the worst case.*

4 Art Gallery Theorems

Art Gallery Theorems have practical applications and answer natural combinatorial questions on visibility. One variant that corresponds to visibility from a line is when mobile guards are allowed to patrol an interior line segment of the polygon[9, Chapter 3]. We insist that such mobile guards can patrol only an edge of the polygon. By examining the structure of a partition of an orthogonal polygon into L-shaped pieces[9, Section 2.5], we obtain the following result for 2-oriented polygons.

Theorem 4.1 *Let $C = \{\theta_1, \theta_2\}$ be a set of two orientations. Then,*

- *$\lfloor n/4 \rfloor$ edges are always sufficient and sometimes necessary to cover the interior of a C-oriented polygon of n edges with C-rays. Moreover, these edges can be chosen to be all θ_1-oriented or all θ_2-oriented.*
- *$\lfloor n/2 \rfloor$ edges are always sufficient and sometimes necessary to cover the interior and the boundary of a C-oriented polygon of n edges with C-rays.*

Moreover, the corresponding partitioning algorithm based on trapezoidization[9, section 2.6] can be applied here to find the set of covering edges.

Theorem 4.2 *Given $C = \{\theta_1, \theta_2\}$ a set of 2 orientations, a set of $\lfloor n/4 \rfloor$ edges to cover the interior of a C-polygon using C-rays, can be found in $O(n \log\log n)$ time.*

References

[1] D Avis and G.T. Toussaint. An optimal algorithm for determining the visibility of a polygon from an edge. *IEEE Trsansactions on Computing*, C-30:910–914, 1981.

[2] D. Dobkin, H. Edelsbruner, and M Overmars. Searching for empty convex polygons. In *Proc. 4th ACM Symposium on Computational Geometry*, pages 224–228, 1988.

[3] V. Estivill-Castro and V. Raman. Edge visibility for finitely oriented figures. Technical Report in preparation, Department of Computer Science, University of Waterloo, 1990.

[4] R.H. Guting. Stabbing C-Oriented polygons. *IPL*, 16:35–40, 1983.

[5] R.H. Guting. Dynamic C-Oriented polygonal intersection searching. *Information and Control*, 63:143–163, 1984.

[6] B. Joe and R.B. Simpson. Corrections to Lee's visibility polygon algorithm. *BIT*, 27:458–473, 1987.

[7] D.T. Lee. Visibility of a simple polygon. *Comput. Vision, Graphics, and Image Proc.*, 22:207–221, 1983.

[8] T. Lozano-Perez and M.A. Wesley. An algorithms for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22:560–570, 1979.

[9] J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford University Press, New York, 1987.

[10] G. Rawlins and D. Wood. Optimal computation of finitely oriented convex hulls. *Information and Control*, 72:150–166, 1987.

[11] G. Rawlins and D. Wood. Restricted-oriented convex sets. Technical Report CS-89-01, Department of Computer Science, University of Waterloo, 1989.

[12] S. Schuierer, G. Rawlins, and D. Wood. A general theory of visibility. Technical Report in preparation, Department of Computer Science, University of Waterloo, 1989.

[13] P. Widemayer, Y. F. Wu, and C. K. Wong. Distance problems in computational geometry with fixed orientations. In *Proceedings of the First Annual ACM Conference in Computational Geometry*, pages 186–195, Baltimore, Maryland, 1985.

[14] P. Widmayer, Y.F. Wu, M.D.F. Schlag, and C.K. Wong. On some union and intersection problems for polygons with fixed orientation. *Computing*, 36:183–197, 1986.