Optimal Guarding of Polygons and Monotone Chains (Extended Abstract)

Danny Z. Chen* Vladimir Estivill-Castro† Jorge Urrutia‡

Abstract

In this paper we study several problems concerning the guarding of a polygon or a x-monotone polygonal chain P with n vertices from a set of points lying on it. Our results are: (1) An $O(n \log n)$ time sequential algorithm for computing the shortest guarding boundary chain of a polygon P. (2) An $O(n \log n)$ time sequential algorithm for computing the smallest set of consecutive edges guarding a polygon P. (3) Parallel algorithms for each of the two previous problems that run in $O(\log n)$ time using $O(n)$ processors in the CREW-PRAM computational model. (4) A linear sequential algorithm for computing the smallest left-guarding set of vertices of an x-monotone polygonal chain P. (5) An optimal $\Theta(n \log n)$ sequential algorithm for computing the smallest guarding set of relays of an x-monotone polygonal chain P. (6) Finally, we consider the problem of placing, on a x-monotone polygonal chain P, a minimum set of vertex guards which collectively cover the entire surface and show that this problem is NP-complete. The previously best known sequential algorithms for problems (1) and (2) take $O(n^2 \log n)$ time.

1 Introduction

In this paper we study two types of guarding problems for simple polygons and monotone chains. To motivate our first problem, suppose that we have a broadcasting station s that must transmit a radio signal to all points on a monotone chain C that models the skyline of a digital terrain. If a point p on the chain C is not visible form s, no direct transmission is possible from s to p. Therefore, we need to introduce a set of relay stations that will rebroadcast this signal originating in s until p is eventually reached; refer to Figure 1. For technical reasons, including avoiding noise in the broadcasting created by multiple coverage, all relay stations must broadcast only in the direction opposite to the location of s. Thus, if a relay station is placed to the left of s, it is allowed to broadcast only to its left, whereas if it is located to the right of s, it will broadcast only to its right. Our first problem is then, that of determining the minimum number of relay stations needed to cover all of C when the position of s is given. A more interesting variation of this problem is that of finding the optimal location where a broadcast station has to be built on C such that the number of relay stations needed to cover C is minimized. We give $\Theta(n)$ and $\Theta(n \log n)$ algorithms, respectively, to solve both problems.

We also study here the problem of finding the shortest guarding boundary chain of a polygon P; that is, the shortest boundary segment T of a polygon P such that every point of P is visible from a point T. A closely related problem, that of finding the smallest set of consecutive edges of a polygon that guard P, is also considered. We give $\Theta(n \log n)$ time algorithms to solve both of these problems. Parallel algorithms that solve this problems in $O(\log n)$ time using n processors in the CREW-PRAM model are given in the full version of the paper. Previous sequential algorithms for these problems require $O(n^2 \log n)$ time [1].

The common feature to most of the results presented here is that they are solved using a similar technique, that is: shoot rays from the vertices of a polygon to create a set of pockets, define a corresponding interval for each pocket and finally, from the set of intervals, find the optimal solution to the problem.

The covering problems we consider are motivated by applications of the notion of visibility in polyhedral terrains. These applications include the configuration of line-of-sight transmission networks for TV and radio broadcasting, cellular telephony, microwave relays, and other telecommunication technologies [2, 6, 8]. Computing the smallest vertex set (SVS) guarding a polyhedral terrain is known to be NP-complete [5] as well as the SVS guarding a poly-

*Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, U.S.A. chen@cse.nd.edu.

†Laboratorio Nacional de Informática Avanzada, Rébsamen 80, Xalapa, Veracruz 91000, México. vlad@xalapa.lania.mx.

‡Department of Computer Science, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada. jorge@csi.uottawa.ca.

gon [9]. However, applications of polyhedral terrains like planning of micro-wave links, TV-broadcasting relays and telecommunications relays require that a communication path is established between two points p_1 and p_2 of the terrain (two cities, two satellite-antennas, etc). Consideration of the problem in the vertical plane E defined by the line joining the two points p_1 and p_2 reduces the problem back to two dimensions (because a skyline of the terrain is now considered). In the full paper, we demonstrate that finding the SVS covering an x-monotone polygonal chain is NP-complete for general visibility. Thus, the algorithms presented here provide efficient solutions to those polynomial variants of covering problems in the plane.

2 Optimizing Relays

In this section we solve the problem of finding the minimum number of relay stations for our broadcasting problem when the position of the station s is fixed. We can divide this problem into two symmetric problems, covering the part of the chain that is to the left of s, and covering the part that is to the right. Now, the problem can be modeled as follows. Let $P = \langle v_1, v_2, \ldots, v_n \rangle$ be a sequence of points in the plane defining an x-monotone polygonal chain; that is, the orthogonal projections of $v_1, \ldots, v_n$ onto the x-axis are in the same order as in the chain. We say that a point v is to the *left* of a point u if the x-coordinate of v is smaller than the x-coordinate of u (see Figure 1). In particular, in the x-monotone chain P, v_i is to the left of v_j, for all $i < j$ $(i, j \in \{1, \ldots, n\})$. We say that a point v is *under* the line defined by $v_i v_{i+1}$ if the 2-line path $v_i v_{i+1} v$ makes a right turn at v_{i+1}. A point $p = (x, y)$ is *under* the polygonal chain $P = \langle v_1, v_2, \ldots, v_n \rangle$ if there is $i \in 1, \ldots, n-1$ such that $x \leq x$ coordinate of v_{i+1}; $x \geq x$ coordinate of v_i, and p is under $v_i v_{i+1}$. We say that a point v to the left of a point u is *left-visible* in the terrain defined by the polygonal chain $P = \langle v_1, v_2, \ldots, v_n \rangle$ if the line vu never intersects the set of points under the polygonal chain P.

A set of points G in the x-monotone chain P, such that any point v on the chain is left-visible from at least one point in G is called a left-cover of the chain. We consider the problem of computing the smallest left-cover of an x-monotone polygonal chain. Clearly, the set of vertices $\{v_2, v_3, \ldots, v_n\}$ is a left-cover of size $n-1$. Thus, the smallest cover has a finite set of points. The points of a cover will be called guards.

Before we give an algorithm for computing a minimum cover we present some properties of covers. Let

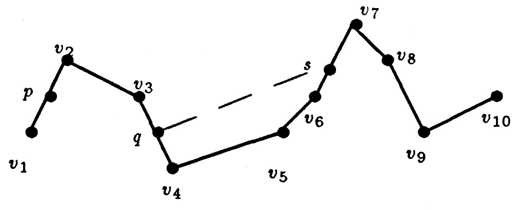

Figure 1: Points p and q are to the left of s. Point q is left-visible from s, but point p in not left-visible from s. A guard at s left-covers the subchain $\langle v_2, v_3, v_4, v_5, v_6, s \rangle$.

G be an minimum cover that has guards on edges of the polygonal chain P. For each guard g in an edge $v_i v_{i+1}$, replace g with the left most vertex g' that is a right turn and is to the right of g (see Figure 2). The new set of points N has the same cardinality as G. It is not hard to see that the new set is also a cover. The reader may verify that the portion of P left-visible from g is contained in the portion of P left-visible from g'. Thus, N will be a minimum cover that consists of guards placed only at vertices of the polygonal chain P that are right turns. Note that, in any cover $G = \langle g_1, g_2, \ldots, g_m \rangle$, the guard g_i $(i \in \{1, \ldots, m-1\})$ is always left visible from a guard g_j with $i < j$. Thus, guards form a link of visibility from right to left. Let $LV(g_i)$ be the set of points on the polygonal chain left-visible from g_i. Observe that, for all minimum covers $G = \langle g_1, \ldots, g_m \rangle$, the set of points $LV(g_i)$ is not contained in the set $LV(g_j)$, for all $i < j$. Otherwise, we could remove g_i from G to obtain a smaller cover.

We are now ready to present an overview of an algorithm for computing the minimum left-cover. The algorithm works incrementally, traversing the polygonal chain from left to right, starting with v_1. The algorithm repeatedly finds the next vertex v_i that is a right turn, and places a guard g at v_i. It analyzes all previously placed guards. If there is a previous guard g' such that $LV(g')$ is contained in $LV(g)$, then the algorithm removes g' from the set of guards. The algorithm terminates when it reaches v_n and in this last step, v_n is added to the set of guards and also, previous guards that guard regions guarded by v_n are removed from the set of guards. We call this algorithm the *Army-Withdraw* algorithm.

Theorem 2.1 ***The Army-Withdraw algorithm computes a minimum left-cover of an x-monotone polygonal chain.***

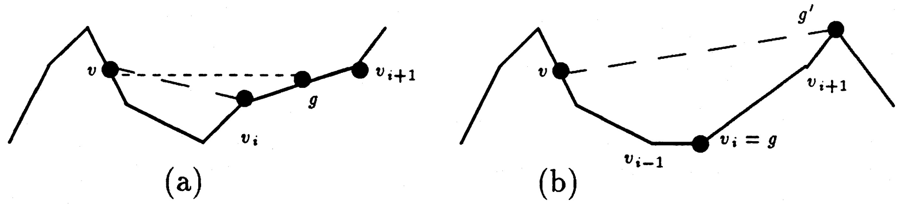

Figure 2: Guards in a minimal cover can be placed at vertices that are right turns.

Proof: Let $G = \langle g_1, g_2, \ldots, g_m \rangle$ be a minimum left-cover such that g_i is a right turn of the polygonal chain P, for $i = i, \ldots, m-1$. Moreover, with out loss of generality, we may assume that, for each g_i there is no right turn vertex v such that v is to the right of g_i and $LV(g_i)$ is contained in $LV(v)$.

Clearly, the set computed by the *Army Withdraw* algorithm is a left cover $R = \langle r_1, \ldots, r_{m'} \rangle$, with $m' \geq m$. We now probe that $m' = m$. We probe this by contradiction. We assume that $m' > m$, and since $r_{m'} = g_m = v_n$, there must be a t such that $r_t \neq g_t$. Let t_0 be the first t with this property. Since r_{t_0} and g_{t_0} are right turn vertices on the polygonal chain, we have two cases:

Case 1: The guard g_{t_0} is to the left of r_{t_0}. Since g_{t_0} is a right turn vertex, the *Army Withdraw* algorithm must have placed a guard r' at g_{t_0} at some stage (at least to cover the edge of the polygonal chain with right endpoint at g_{t_0}). Moreover, at a later stage, the *Army Withdraw* algorithm must have found a right turn vertex r to the right of r' and such that $LV(r')$ is contained in $LV(r)$ (to remove r', since r' is not r_{t_0}). But this contradicts the choice of G, since this implies that there is a right turn vertex to the right of g_{t_0} that covers at least what g_{t_0} covers.

Case 2: The guard r_{t_0} is to the left of g_{t_0}. In this case, the region $LV(r_{t_0})$ must be covered by two or more guards of G (if $LV(r_{t_0})$ was covered by only one guard $g \in G$ to the right of r_{t_0}, the *Army Withdraw* algorithm would have found g and removed r_{t_0}). Thus, there must exists different points u and v in $LV(r_{t_0})$ and guards g_i and g_j in G to the right of r_{t_0} such that:

- point v is in $LV(g_i)$ and not in $LV(g_j)$, and
- point u is in $LV(g_j)$ and not in $LV(g_i)$.

Without loss of generality assume g_i is to the left of g_j.

Subcase 2a: Point v is to the left of u. In this case, we have visibility rays as in Figure 3 (a), because $v \in LV(g_i)$ and $u \in LV(g_j)$. This implies that the polygonal chain is under the line segment vg_i and under the line segment ug_j. Therefore, v is left visible from g_j. This contradicts $v \notin LV(g_j)$.

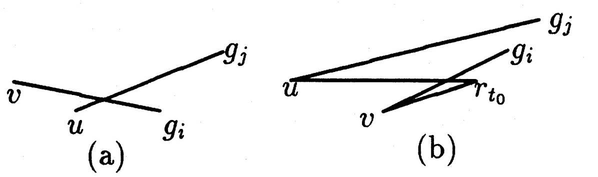

Figure 3: Subcases for proof of correctness of *Army Withdraw* algorithm.

Subcase 2b: Point u is to the right of v. Since u and v are in $LV(r_{t_0})$ we have visibility rays as in Figure3 (b). But then, $u \in LV(g_i)$, a contradiction. This completes the proof. □

We now show how to carry out the *Army Withdraw* algorithm in $O(n)$ time. The idea for a linear algorithm starts by characterizing those guards g' that the *Army Withdraw* algorithm will remove at a later step because it finds a right turn vertex g to the right of g' and such that $LV(g)$ contains $LV(g')$. We call these guards removable guards. The algorithm will preprocess the polygonal chain P to obtain the necessary information, such that, in the left to right pass over the polygonal chain P, the *Army Withdraw* algorithm will identify removable guards in constant time, thus the *Army Withdraw* algorithm will not have to insert them just to remove them later. The *Army Withdraw* algorithm reduces to a scan from left to right that requires linear time.

First, note that a removable guard g' is left visible from the replacement g. This is the main observation in the proof of the following proposition.

Proposition 2.2 *If g' at v_i is a removable point, the right most point that causes the removal of g' is the first point, from left to right, in the upper chain on the convex hull of the set of points $v_i, v_{i+1}, \ldots, v_n$.*

Proof: Let g' be a removable guard at v_i and let g be the right most point that covers $LV(g')$. A guard at g must see g'. If the line segment $g'g$ is not the first edge of the upper chain of the convex hull of $\{v_i, \ldots, v_n\}$, then there is a vertex v_j to the right of g and that sees g'. But, then is not hard to see that $LV(g') \subset LV(v_j)$ which contradicts that g is the right most point with this property. □

We are now ready to present the main result of this section.

Theorem 2.3 *An optimum left-cover for an x-monotone polygonal chain $P = \langle v_1, \ldots, v_n \rangle$ can be computed in $\Theta(n)$ time.*

Proof: That any algorithm requires to examine all vertices of the chain trivialy gives a lower bound of $\Omega(n)$ time. An $O(n)$ time algorithm is now obtained by an $O(n)$ preprocessing step for the *Army Withdraw* algorithm. Before performing the left to right scan of the *Army Withdraw* algorithm, perform a right to left Graham scan [10] recording, for each vertex v_i (with $i = n, n-1, \ldots, v_1$), the vertex v_j to the right of v_i such that the line segment v_iv_j is the first edge of the upper chain of the convex hull of the set $\{v_i, \ldots, v_n\}$. The vertex v_j associated in this way to a vertex v_i will be called its remover. This preprocessing will require linear time.

The *Army Withdraw* algorithm is performed next. Each vertex v_i that is a right turn of the polygonal chain will be added to the set of guards only if the remover of v_i is above the line that includes the line segment $v_{i-1}v_i$. This test can be performed in constant time, thus the *Army Withdraw* scan requires also linear time. □

3 Locating the Broadcaster

In this section we solve the problem of finding the optimal location for the broadcasting station s. We model this problem as follows: given a polygonal chain, we are required to find the position on the polygonal chain of a broadcasting station such that the polygonal chain is covered and the number of relays is minimized. However, we have the restriction that no relay can broadcast towards the station, since this would create interferencing signals.

To solve this problem we first establish some properties of the solution. Again, let $P = \langle v_1, v_2, \ldots, v_n \rangle$ be the sequence of points in the plane that defines the x-monotone chain that represents the skyline of the terrain. Let s be the point where the broadcasting station minimizes the number of relays. If we were given s, the relays can be computed in linear time by covering the chain $v_1, v_2, \ldots, s$ from the left as in the previous section and covering the chain $s, \ldots, v_{n-1}, v_n$ from the right by a symmetric algorithm. Thus, the problem reduces to finding s. Note that if the station s is to the right of a vertex v_i, and the y-coordinate of v_i is larger than the y-coordinate of v_{i-1}, then a relay will be forced at v_i if the ray that extends the edge $v_{i-1}v_i$ does not intersect P; that is, if v_i is the highest point in P to the right of v_{i-1} that sees v_{i-1}.

Moreover, if the ray that extends the edge $v_{i-1}v_i$ intersects P at a point u to the left of s, there is no need for a relay at v_i because the relay that covers u also covers the edge $v_{i-1}v_i$ and anything that v_i covers. If u is to the right of s, then v_i will require a relay.

Now we are ready for the description of out algorithm. For each vertex $v_i = (x_i, y_i)$ (with y-coordinate larger than v_{i-1}) we associate an interval in the real line. The interval associated with v_i has lower end point the x-coordinate of v_i and upper endpoint the x-coordinate of u (the first intersection of P with the extension of the edge $v_{i-1}v_i$), and $+\infty$ if this extension does not intersect P.

Note that the number $l(s)$ of relays required to the left of a point s in the x-monotone polygonal chain P is the number of intervals that contain the x-coordinate of s. Applying the argument symmetrically (reversing left for right), we can compute the number $r(s)$ of relays required to the right of a point s in the x-monotone polygonal chain P. The station that minimizes the total number of relays can be placed in any vertex s such that $l(s) + r(s)$ is minimum.

We require $O(n \log n)$ time to compute all the intervals and to sort their endpoints as to compute $l(s)$ and $r(s)$ for all vertices. All other steps require linear time; thus, we have the following result.

Theorem 3.1 *Finding the position of a broadcasting station on a x-monotone polygonal chain with n vertices representing the intersection of a polyhedral terrain and a vertical plane, such that the chain is covered, the relays are minimized and there is no interference can be done in $\Theta(n \log n)$.*

Proof: The proof of the lower bound is based on a reduction from integer sorting to the broadcasting station placement problem. The details are left to the full paper. □

4 Chains Guarding Polygons

The problems in Sections 2 and 3 are closely related to visibility and stationary guarding. Essentially, the algorithm in Section 3 for finding an optimal placement for the broadcasting station s is based on the following framework: (1) Perform ray-shooting to obtain a set of "pockets" on the input polygonal chain, (2) define an interval corresponding to each "pocket", (3) from the set of intervals, find an optimal solution to the problem. We generalize the techniques of this later algorithm to study two problems on the *weakly visibility* of simple polygons. Let P be an n-vertex simple polygon. For a point p and an object C in P, p is said to be *weakly visible* from C if and only if p is visible from some point on C (depending on p). Polygon P is said to be *weakly visible* from C if and only if every point $p \in P$ is weakly visible from C.

We consider the problem of computing the shortest weakly visible chain of a simple polygon (called it the SWVC problem) and the problem of computing a chain on the polygonal boundary that contains the minimum number of consecutive edges and from which the polygon is weakly visible, also called the *consecutive edge guards (CEG) problem* [1]. Assuming that the exterior of polygon P is "opaque", we would like to find a chain C on the boundary of P such that (i) P is weakly visible from C, and (ii) for the SWCV, the length of C is the shortest among all such chains on the boundary of P, while for CEG, the number of edges in C is the smallest. Intuitively, if P represents a house whose interior is that of a simple polygon, then C is the contiguous portion along the walls of P by which a mobile guard has to patrol back and forth in order to keep the inside of P completely under surveillance and satisfies a minimality condition.

For these two problems, we provide sequential algorithms that run in $O(n \log n)$ time, and parallel algorithm that run in $O(\log n)$ time using $O(n)$ processors in the CREW PRAM computational model. Our sequential solutions to these problems improve the previously best known sequential $O(n^2 \log n)$ time algorithms [1].

Suppose polygon P is defined by a sequence of its vertices $(v_1, v_2, \ldots, v_n)$ in the counterclockwise order along the boundary $B(P)$ of P. A vertex v_i of P is said to be *reflex* if the path $v_{i-1}v_iv_{i+1}$ makes a right turn at v_i (with the convention that $v_{n+1} = v_1$ and $v_0 = v_n$). Our algorithms crucially rely on the notion of polygon "pockets" which are defined with respect to reflex vertices.

Definition 4.1 *Let v_i be a reflex vertex of P. The* clockwise *(resp.,* counterclockwise*)* pocket *of v_i is defined as follows: Shoot a ray starting at v_{i-1} (resp., v_{i+1}) and passing v_i, and let the ray hit $B(P) - v_{i-1}v_i$ (resp., $B(P) - v_iv_{i+1}$) at a point p; then the chain along $B(P)$ from v_i* clockwise *(resp.,* counterclockwise*) to p is called the* clockwise *(resp.,* counterclockwise*)* pocket *of v_i and is denoted by $PK_c(v_i)$ (resp., $PK_{cc}(v_i)$).*

Proposition 4.1 *Polygon P is weakly visible from a chain C on its boundary if and only if C intersects every pocket of P.*

Proof: If the chain C does not intersect a pocket of a vertex v_i (say, its clockwise pocket $PK_c(v_i)$), then $C \cap PK_c(v_i)$ is empty and clearly the vertex v_{i-1} cannot be weakly visible from C. If C intersects every pocket of P, then we show that every point of P is weakly visible from C. This is proved by contradiction, as follows. Suppose there is a point p in P that is not weakly visible from C. Then for any point q on C, the shortest path from p to q inside P consists of at least two line segments. Let pp' be the first segment on the shortest p-to-q path inside P, and assume that the shortest path makes a right turn at p' (the other case is proved similarly). Now shoot a ray starting at p and passing p', and let the ray hit $B(P) - pp'$ at a point h. Then the segment $p'h$ partitions P into two subpolygons P_1 and P_2, with $p \in P_1$ and $C \subset P_2$. The fact that the shortest p-to-q path makes a right turn at p' implies that the chain C is completely contained in the interior of the chain along $B(P_2)$ from p' counterclockwise to h, and that p' is a reflex vertex of P. It is now easy to see that the pocket $PK_c(p')$ is completely contained in P_1 and hence $PK_c(p') \cap C$ is empty, contradicting that C intersects every pocket of P. □

The two corollaries below follow immediately from Proposition 4.1.

Corollary 4.2 *The shortest weakly visible chain of P must intersect every pocket of P.*

Corollary 4.3 *For two distinct pockets PK' and PK'' of P, if $PK'' \subset PK'$, then PK' can be removed from the set of pockets of P without affecting the structure of the shortest weakly visible chain of P.*

Proof: This is because any chain on $B(P)$ intersecting PK'' must also intersect PK'. □

Based on the observations discussed above, we can map the points on $B(P)$ to points on a unit circle *Circle* (a bijection function for such a mapping can be defined trivially). Hence, every pocket of P is mapped to an arc on *Circle*. Because P has $O(n)$ pockets, there are $O(n)$ corresponding arcs on *Circle* to consider. Let A denote the set of arcs so obtained. Hence the SWVC problem is reduced to the problem of first eliminating the arcs in A that contain some other arcs of A (Corollary 4.3) and then finding an arc a^* on *Circle* (a^* is not necessarily in A) that intersects all the remaining arcs of A and has the shortest length.

Proposition 4.4 *Both endpoints of an arc a^* on Circle that intersects all the arcs of A and has the shortest length can be chosen to be endpoints of some arcs in A.*

Proof: If a^* consists of only a single point, then we can easily let a^* coincide with an endpoint of an arc in A. When a^* has two distinct endpoints, if one endpoint of a^* were not an endpoint of some arc in A, then a^* could have been made shorter, a contradiction. □

We are now ready to present the algorithm for finding the shortest weakly visible chain of P.

(1) For every reflex vertex of P, compute its two pockets. This can be done in $O(n\log n)$ time by using ray shooting algorithms in simple polygons [3, 4, 7].

(2) Map the set of pockets of P to a set A of arcs on a unit circle *Circle*, and sort the endpoints of the arcs in A.

(3) Eliminate the arcs in A that contain some other arcs in A. Let A' be the set containing the remaining arcs in A.

(4) Compute an arc a^* on *Circle* that intersects all the arcs in A' and has the shortest length, based on Proposition 4.4. Map a^* back to $B(P)$.

The correctness of the SWVC algorithm follows from the observations given above. The time complexity of the algorithm is $O(n\log n)$ because Steps (3) and (4) can be performed in $O(n)$ time.

A parallel implementation of the sequential SWVC algorithm takes $O(\log n)$ time using $O(n)$ CREW PRAM processors. The details of this parallel algorithm are left to the full paper.

Our techniques can also be used to solve the related problem of computing a chain on the polygon boundary that contains the minimum number of edges and from which the polygon is weakly visible (called the *consecutive edge guards problem* [1]). Our sequential and parallel solutions to this problem have the same complexity bounds as those for the SWVC problem, improving the previously best known sequential $O(n^2\log n)$ time algorithm for the consecutive edge guards problem [1]. The details are left to the full paper.

References

[1] M. Abellanas, J. Garcia-Lopez, and F. Hurtado. "Consecutive guards," *Proc. 6th CCCG*, 1994, pp. 393–398.

[2] L. Boithias. *Radio Wave Propagation.* North Oxford Academic Publishers Ltd, London, England, English translation edition, 1987.

[3] B. Chazelle, H. Edelsbrunner, M. Grigni, L. Guibas, J. Hershberger, M. Sharir, and J. Snoeyink. "Ray shooting in polygons using geodesic triangulations," *Proc. 18th ICALP*, 1991, pp. 661–673.

[4] B. Chazelle and L. J. Guibas. "Visibility and intersection problems in plane geometry," *Discrete and Computational Geometry*, 4 (1989), pp. 551–581.

[5] R. Cole and M. Sharir. Visibility problems for polyhedral terrains. *J. Symbolic Comput.*, 7:11–30, 1989.

[6] M.F. Goodchild and J. Lee. Coverage problems an visibility regions on topographic surfaces. *Annals of Operations Research*, 18:175–186, 1989.

[7] L. J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. E. Tarjan. "Linear time algorithms for visibility and shortest path problems inside triangulated simple polygons," *Algorithmica*, 2 (1987), pp. 209–233.

[8] J. Lee. Analysis of visibility sites on topographic surfaces. *Int. J. of Geographic Information Systems*, 5:413–429, 1991.

[9] D. T. Lee and A. K. Lin. "Computational Complexity of art gallery problems," *IEEE Trans. Info. Th*, **IT-32** (1986), pp. 276–282.

[10] F. P. Preparata and M. I. Shamos. *Computational Geometry: an Introduction.* Springer-Verlag, New York, NY, 1985.