Guarding Orthogonal Art Galleries With Sliding Cameras☆

Stephane Durocher[a,1], Omrit Filtser[b], Robert Fraser[a], Ali D. Mehrabi[c,2], Saeed Mehrabi[a,3,*]

[a] Department of Computer Science, University of Manitoba, Canada.
[b] Department of Computer Science, Ben-Gurion University, Israel.
[c] Department of Computer Science, Eindhoven University of Technology, the Netherlands.

Abstract

Let P be an orthogonal polygon with n vertices. A sliding camera travels back and forth along an orthogonal line segment $s \subseteq P$ corresponding to its *trajectory*. The camera sees a point $p \in P$ if there is a point $q \in s$ such that $\overline{pq}$ is a line segment normal to s that is completely contained in P. In the *Minimum-Cardinality Sliding Cameras (MCSC) problem*, the objective is to find a set S of sliding cameras of minimum cardinality to guard P (i.e., every point in P can be seen by some sliding camera in S), while in the *Minimum-Length Sliding Cameras (MLSC) problem* the goal is to find such a set S so as to minimize the total length of trajectories along which the cameras in S travel.

In this paper, we answer questions posed by Katz and Morgenstern (2011) by presenting the following results: (i) the MLSC problem is polynomially tractable even for orthogonal polygons with holes, (ii) the MCSC problem is NP-complete when P is allowed to have holes, and (iii) an $O(n^3 \log n)$-time 2-approximation algorithm for the MCSC problem on *[NE]-star-shaped* orthogonal polygons with n vertices (similarly, [NW]-, [SE]-, or [SW]-star-shaped orthogonal polygons).

Keywords: Orthogonal Art Galleries, Sliding Cameras, Approximation Algorithms.

☆A preliminary version of some of these results have appeared in the 38th International Symposium on Mathematical Foundations of Computer Science (MFCS 2013) [1].

*Corresponding author
Email addresses: durocher@cs.umanitoba.ca (Stephane Durocher), omritna@cs.bgu.ac.il (Omrit Filtser), fraser@cs.umanitoba.ca (Robert Fraser), amehrabi@win.tue.nl (Ali D. Mehrabi), mehrabi@cs.umanitoba.ca (Saeed Mehrabi)

[1] Work of the author is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).
[2] Work of the author is supported by the Dutch Science Foundation (NWO) under grant 612.001.118
[3] Work of the author is supported in part by a Manitoba Graduate Scholarship (MGS).

1. Introduction

The art gallery problem is well known in computational geometry, where the objective is to cover a geometric shape (e.g., a polygon) with the union of the visibility regions of a set of point guards while minimizing the number of guards. The problem's multiple variants have been examined extensively (e.g., see [2, 3, 4]) and can be classified based on the type of guards (e.g., points or line segments), the type of visibility model, and the geometric shape (e.g., simple polygons, orthogonal polygons [5], or polyominoes [6]).

Recently, Katz and Morgenstern [7] introduced a variant of the orthogonal art gallery problem in which *sliding cameras* are used to guard the gallery. Let P be an orthogonal polygon with n vertices. A sliding camera travels back and forth along an orthogonal line segment $s \subseteq P$ called its *trajectory.* The camera (i.e., the guarding line segment s) can *see* a point $p \in P$ (equivalently, p is *orthogonally visible* to s) if and only if there exists a point q on s such that $\overline{pq}$ is normal to s and is completely contained in P. There are two variants of this problem: in the *minimum-cardinality sliding cameras (***MCSC***) problem*, we wish to minimize the number of sliding cameras so as to guard P entirely, while in the *minimum-length sliding cameras (***MLSC***) problem* the objective is to minimize the total length of trajectories along which the cameras travel; we assume that in both variants of the problem, the polygon P and all sliding cameras are constrained to be orthogonal. In both problems, every point in P must be visible to some camera (see Figure 1). In this paper, we answer questions posed by Katz and Morgenstern [7] by presenting the following results, some of which appeared in [1] (see also [8]):

- We show that the MLSC problem is solvable in $O(n^{2.3727})$ time even for orthogonal polygons with holes, where n is the number of the vertices of the polygon.
- We show that the MCSC problem is NP-complete for orthogonal polygons with holes.
- We give an $O(n^3 \log n)$-time 2-approximation algorithm to the MCSC problem on *[X]-star-shaped* orthogonal polygons with n vertices, where $X \in \{NE, NW, SE, SW\}$ (see Section 5 for the definition of [X]-star-shaped polygons).

Throughout the paper, we denote an orthogonal polygon with n vertices by P. A vertex u of P is *reflex*, if the angle at u that is interior to P is 270°. We denote the set of reflex vertices and the set of edges of P by $V(P)$ and $E(P)$, respectively. We consider P to be a closed region; therefore, the trajectory of a camera may include an edge of P. We also assume that a camera can see all points on its trajectory. Let H_u and V_u be the maximum-length horizontal and vertical line segments, respectively, inside P through a vertex $u \in V(P)$. Let $L(P) = \{H_u \mid u \in V(P)\} \cup \{V_u \mid u \in V(P)\}$. Moreover, let L be an orthogonal line segment (with respect to P) inside P; the *visibility region* of L is the union

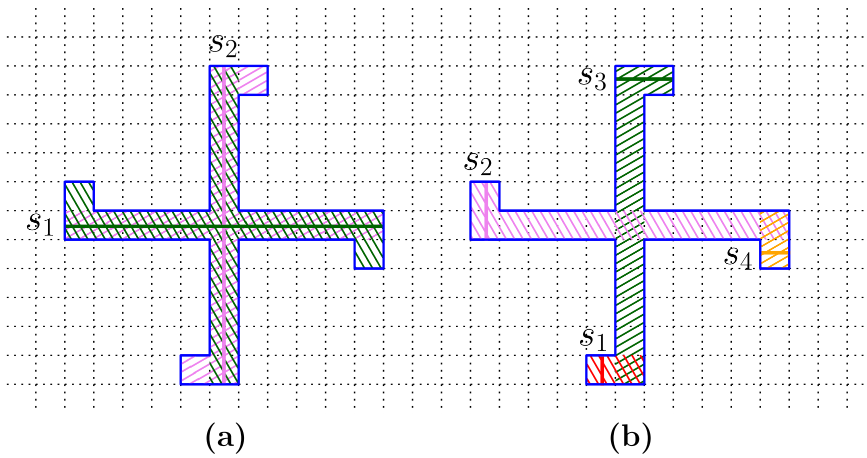


Figure 1: An illustration of the MCSC and MLSC problems. Each grid cell has size 1×1. (a) The set of two sliding cameras s_1 and s_2 as an optimal solution for the MCSC problem on P; each shaded region indicates the visibility region of the corresponding camera. (b) A set of four sliding cameras whose total length is 6, which is an optimal solution for the MLSC problem on P.

of the points in P that are seen by the sliding camera that travels along L. We say that a set T of orthogonal line segments contained in P is a *cover* of P, if the corresponding cameras can collectively see all points in P; equivalently, we say that the line segments in T *guard* P entirely.

The rest of the paper is organized as follows. Section 2 presents related work. In Section 3, we describe a polynomial-time algorithm for the MLSC problem. Section 4 presents our hardness result for the MCSC problem on polygons with holes. Finally, we present our 2-approximation algorithm in Section 5. We conclude the paper in Section 6.

2. Related Work

The art gallery problem was first introduced by Klee in 1973 [9]. Two years later, Chvátal [10] gave an upper bound proving that $\lfloor n/3 \rfloor$ point guards are always sufficient and sometimes necessary to guard a simple polygon with n vertices. The orthogonal art gallery problem was first studied by Kahn et al. [11] who proved that $\lfloor n/4 \rfloor$ guards are always sufficient and sometimes necessary to guard the interior of a simple orthogonal polygon. Lee and Lin [12] showed that the problem of guarding a simple polygon using the minimum number of guards is NP-hard. Moreover, the problem was also shown to be NP-hard for orthogonal polygons [13]. Even the problem of guarding the vertices of an orthogonal polygon using the minimum number of guards is NP-hard [14].

Limiting visibility allows some versions of the problem to be solved in polynomial time. Motwani et al. [15] studied the art gallery problem under s-visibility, where a point guard $p \in P$ can see all points in P that can be reached from p

by an orthogonal staircase path contained in P. They use a perfect graph approach to solve the problem in polynomial time. Worman and Keil [16] defined r-visibility, in which a point guard $p \in P$ can see all points $q \in P$ such that the bounding rectangle of p and q (i.e., the axis-parallel rectangle with diagonal $\overline{pq}$) is contained in P. Given that P has n vertices, they use a similar approach to Motwani et al. [15] to solve this problem in $\widetilde{O}(n^{17})$ time, where $\widetilde{O}()$ hides poly-logarithmic factors. Moreover, Lingas et al. [17] presented a linear-time 3-approximation algorithm for this problem.

Katz and Morgenstern [7] introduced the MCSC problem. They first considered a restricted version of the problem, where cameras are constrained to travel only vertically inside the polygon. Using a similar approach to Motwani et al. [15] they constructed a graph G corresponding to P and then showed that (i) solving this problem on P is equivalent to solving the minimum clique cover problem on G, and that (ii) G is chordal. Since the minimum clique cover problem is polynomial-time solvable on chordal graphs, they solved the vertical-camera MCSC problem in polynomial time. They also generalized the problem definition such that both vertical and horizontal cameras are allowed (i.e., the MCSC problem) for which they presented a 2-approximation algorithm for the case in which the input is an x-monotone orthogonal polygon. They left open the complexity of the problem and mentioned studying the MLSC problem as future work. Seddighin [18] considered the sliding cameras problem under *k-visibility*, where a camera's line of sight can pass through k edges of the polygon. He first proved that the MLSC problem is NP-hard under k-visibility and then gave a 2-approximation algorithm for this problem.

A *histogram* H is a simple orthogonal polygon whose boundary contains some edge, called the *base*, whose length is equal to the sum of the lengths of the edges of H that are parallel to the base. Moreover, a *double-sided histogram* is the union of two histograms that share a common base edge and that are located on opposite sides of the base. The MCSC problem is equivalent to the problem of covering P with the minimum number of double-sided histograms, since the visibility region of a sliding camera is exactly a double-sided histogram. Fekete and Mitchell [19] proved that partitioning an orthogonal polygon (possibly with holes) into a minimum number of histograms is NP-hard. Note that in general, the fact that an orthogonal polygon can be covered by k histograms does not imply that it can be partitioned by k histograms. Therefore, their proof does not directly imply that covering an orthogonal polygon with minimum number of double-sided histograms is NP-hard, leaving open the question of whether the MCSC problem is also NP-hard for orthogonal polygons with holes.

3. The MLSC Problem: An Exact Algorithm

In this section, we give an algorithm that solves the MLSC problem exactly in polynomial time even when P has holes. Let T be a cover of P. In this section, we say that T is an *optimal cover for P* if the total length of trajectories along which the cameras in T travel is minimum over that of all covers of P. Our algorithm relies on reducing the MLSC problem to the *minimum-weight vertex cover problem* in bipartite graphs. We remind the reader of the definition of the minimum-weight vertex cover problem:

Definition 1. Given a graph $G = (V, E)$ with positive vertex weights, the *minimum-weight vertex cover* problem is to find a subset $V' \subseteq V$ that is a vertex cover of G (i.e., every edge in E has at least one endpoint in V') such that the sum of the weights of vertices in V' is minimized.

The minimum-weight vertex cover problem is NP-hard in general [20]. However, König's theorem [21] that describes the equivalence between maximum matching and vertex cover in bipartite graphs implies that the minimum-weight vertex cover problem in bipartite graphs is solvable in polynomial time. Given P, we first construct a vertex-weighted graph G_P and then we show (i) that the MLSC problem on P is equivalent to the minimum-weight vertex cover problem on G_P, and (ii) that graph G_P is bipartite.

Similar to Katz and Morgenstern [7], we define a partition of an orthogonal polygon P into rectangles as follows. Extend the two edges of P incident to every reflex vertex in $V(P)$ inward until they hit the boundary of P. Let $S(P)$ be the set of the extended edges and the edges of P whose endpoints are both non-reflex vertices of P. We refer to elements of $S(P)$ simply as *edges*. The edges in $S(P)$ partition P into a set of rectangles; let $R(P)$ denote the set of resulting rectangles. We observe that in order to guard P entirely, it suffices to guard all rectangles in $R(P)$. The following observations are straightforward:

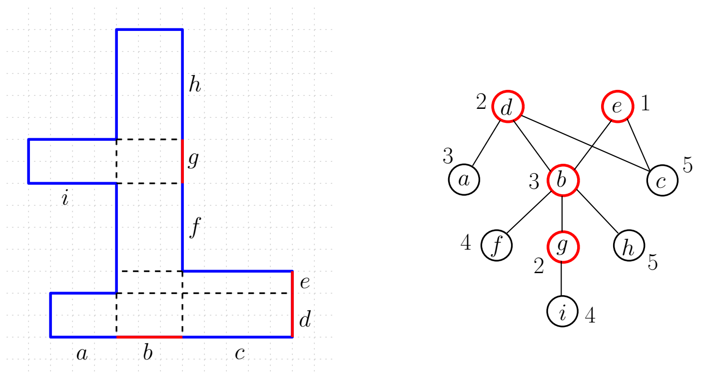
Observation 1. Let T be a cover of P and let s be an orthogonal line segment in T. Then, for any partition of s into line segments $s_1, s_2, \ldots, s_k$ the set $T' = (T \setminus \{s\}) \cup \{s_1, \ldots, s_k\}$ is also a cover of P and the respective sums of the lengths of segments in T and T' are equal.

Observation 2. Let T be a cover of P. Moreover, let T' be the set of line segments obtained from T by translating every vertical line segment in T horizontally to the nearest boundary of P to its right and every horizontal line segment in T vertically to the nearest boundary of P below it. Then, T' is also a cover of P and the respective sums of the lengths of line segments in T and T' are equal. We call T' a *regular cover* of P.

Given P, let $H(P)$ denote the subset of the boundary of P consisting of line segments that are immediately to the right of or below P; in other words, for each edge $e \in H(P)$, the region of the plane immediately to the right of or below e does not belong to the interior of P. Let $B(P)$ denote the partition of $H(P)$ into line segments induced by the edges in $S(P)$. The following lemma follows from Observations 1 and 2:

Lemma 1. *Every orthogonal polygon P has an optimal cover $T \subseteq B(P)$.*

Observation 3. Let P be an orthogonal polygon and consider its corresponding set $R(P)$ of rectangles induced by edges in $S(P)$. Every rectangle $R \in R(P)$



(a) **(b)**

Figure 2: An illustration of the reduction; each grid cell has size 1×1. (a) An orthogonal polygon P along with the elements of $B(P)$ labelled as $a, b, c, \ldots, i$. (b) The graph G_P associated with P; the integer value besides each vertex indicates the weight of the vertex. The vertices of a vertex cover on G_P and their corresponding guarding line segments for P are shown in red.

is seen by exactly one vertical line segment in $B(P)$ and exactly one horizontal line segment in $B(P)$. Furthermore, if $T \subseteq B(P)$ is a cover of P, then every rectangle in $R(P)$ must be seen by at least one horizontal or one vertical line segment in T.

We denote the horizontal and vertical line segments in $B(P)$ that can see a rectangle $R \in R(P)$ by R_H and R_V, respectively. Using Observation 3, we now describe a reduction of the MLSC problem to the minimum-weight vertex cover problem. We construct an undirected weighted graph $G_P = (V, E)$ associated with P as follows: each line segment $s \in B(P)$ corresponds to a vertex $v_s \in V$ such that the weight of v_s is the length of s. Two vertices $v_s, v_{s'} \in V$ are adjacent in G_P if and only if the line segments s and s' can both see a common rectangle $R \in R(P)$. See Figure 2. By Observation 3 the following result is straightforward:

Observation 4. There is a bijection between rectangles in $R(P)$ and edges in G_P.

Next we show equivalence between the two problems and then prove that graph G_P is bipartite. To this end, we first need the following result.

Lemma 2. *Let $R \in R(P)$ be a rectangle and let T be a cover of P. Then, there exists a set $T' \subseteq T$ such that all line segments in T' have the same orientation (i.e., they are all vertical or they are all horizontal) and they collectively guard R entirely.*

Proof. Suppose no such set T' exists. Let R_v (resp., R_h) be the subregion of R that is guarded by the union of the vertical (resp., horizontal) line segments in T and let $R_v^c = R \setminus R_v$ (resp., $R_h^c = R \setminus R_h$). Since R cannot be guarded exclusively by vertical line segments (resp., horizontal line segments), we have $R_v^c \neq \emptyset$ (resp., $R_h^c \neq \emptyset$). Choose any point $p \in R_v^c$ and let L_h be the maximal horizontal line segment inside R that crosses p. Since no vertical line segment in T can guard p, we conclude that no point on L_h is guarded by a vertical line segment in T. Similarly, choose any point $q \in R_h^c$ and let L_v be the maximal vertical line segment inside R that contains q. By an analogous argument, we conclude that no point on L_v is guarded by a horizontal line segment. Since L_h and L_v are maximal and have perpendicular orientations, L_h and L_v intersect inside R. Therefore, no orthogonal line segment in T can guard the intersection point of L_h and L_v, which is a contradiction. □

Theorem 3. *The MLSC problem on P reduces to the minimum-weight vertex cover problem on G_P.*

PROOF. Let S_0 be a vertex cover of G_P and let C_0 be a cover of P defined in terms of S_0; the mapping from S_0 to C_0 will be defined later. Moreover, for each vertex v of G_P let $w(v)$ denote the weight of v and for each line segment $s \in C_0$ let $len(s)$ denote the length of s. We need to prove that S_0 is a minimum-weight vertex cover of G_P if and only if C_0 is an optimal cover of P. We show the following stronger statements: (i) for any vertex cover S of G_P, there exists a cover C of P such that

$$\sum_{s \in C} len(s) = \sum_{v \in S} w(v),$$

and (ii) for any cover C of P, there exists a vertex cover S of G_P such that

$$\sum_{v \in S} w(v) = \sum_{s \in C} len(s).$$

Proof of (i)**.** Choose any vertex cover S of G_P. We find a cover C for P as follows: for each edge $(v_s, v_{s'}) \in E$, if $v_s \in S$ we locate a guarding line segment on the boundary of P that is aligned with the line segment $s \in B(P)$. Otherwise, we locate a guarding line segment on the boundary of P that is aligned with the line segment $s' \in B(P)$. If both v_s and $v_{s'}$ are in S, then both of the corresponding guarding line segments are located on the boundary of P. Since at least one of v_s and $v_{s'}$ is in S, we conclude by Observation 4 that every rectangle in $R(P)$ is guarded by at least one line segment located on the boundary of P and so C is a cover of P. Moreover, for each vertex in S we locate exactly one guarding line segment on the boundary of P whose length is the same as the weight of the vertex. Therefore,

$$\sum_{s \in C} len(s) = \sum_{v \in S} w(v).$$

Proof of (ii)**.** Choose any cover C of P. We construct a vertex cover S for G_P as follows. By Observation 2, let T' be the regular cover obtained from C.

Moreover, let M be the partition of T' into line segments induced by the edges in $S(P)$. By Observation 2, M is also a cover of P. Now, let S be the subset of the vertices of G_P such that $v_s \in S$ if and only if $s \in M$. By Lemma 2 and the fact that M is a cover of P, for any rectangle $R \in R(P)$, there exists a set $C'_R \subseteq C$ such that all line segments in C'_R have the same orientation and collectively guard R. Moreover, since M is obtained from the regular cover T' (which is in turn induced by C), a segment in M that covers R is in $B(P)$. Therefore, by Observation 4 and the fact that M is a cover of P, we conclude that S is a vertex cover of G_P. Moreover, we observe that

$$\sum_{v \in S} w(v) = \sum_{s \in M} len(s) = \sum_{s \in C} len(s). \square$$

We now show that graph G_P is bipartite.

Lemma 4. *Graph G_P is bipartite.*

Proof. The proof follows from the facts that (i) we have two types of vertices in G_P; those that correspond to the vertical line segments in $B(P)$ and those that correspond to the horizontal line segments in $B(P)$, and that (ii) no two vertical line segments in $B(P)$ nor any two horizontal line segments in $B(P)$ can see a fixed rectangle in $R(P)$. $\square$

We now examine the running time of the algorithm. For the running time of Part 1 of the proof of Theorem 3, the described procedure can be completed in $O(n)$ time; note that this is used to construct the cover after computing the vertex cover. For the running time of the construction described in Part 2 of the proof of Theorem 3, we first compute the running time of constructing graph G. Graph G_P can be constructed in $O(n^2 \log n)$ time as follows (recall that n is the number of the vertices of P). For each line segment $s \in B(P)$, let $r(s)$ be a ray that is normal to s and starts from a point on s. For every pair $s_1, s_2 \in B(P)$, where s_1 is horizontal and s_2 is vertical, we check to see if $r(s_1)$ and $r(s_2)$ intersect each other inside P. If so, then the line segments s_1 and s_2 can see a common rectangle and, therefore, we add an edge between the corresponding vertices in G_P. Considering all pairs $s_1, s_2 \in B(P)$ takes $O(n^2)$ time and the ray shooting queries can be answered in $O(\log n)$ time [22]. Therefore, graph G_P is constructed in $O(n^2 \log n)$ time. Next, note that by locating a guard on every edge of the polygon P, we obtain a feasible solution for the MLSC problem; hence, we can assume that $|C| \in O(n)$, where C is a cover of P. So, we can compute a regular cover of P from C in $O(n^2)$ time, by moving each line segment in C down or to the right (in $O(n)$ time) until it hits the boundary of P. The set M can also be obtained in $O(n^2)$ time. Thus, the construction in the second part of the proof can be completed in $O(n^2)$ time. A minimum vertex cover of G_P can be found by solving the maximum matching problem on G_P since these two problems are equivalent on bipartite graphs by Konig's theorem [21]. The maximum matching on G_P can be solved in $O(T(n))$ time, where $T(n)$ is the time required to multiply two $n \times n$ matrices. Since the best known algorithm for multiplying two $n \times n$ matrix runs in $O(n^{2.3727})$ [23], our algorithm runs in $O(n^{2.3727})$ overall time. Therefore, by Theorem 3, Lemma 4 and the fact that minimum-weight vertex cover is solvable in $O(n^{2.3727})$ time on bipartite graphs, we have the main result of this section:

Theorem 5. *Given an orthogonal polygon P, there exists an $O(n^{2.3727})$-time algorithm that finds an optimal cover of P, where n is the number of the vertices of P.*

4. The MCSC Problem: Hardness Result

In this section, we show that the MCSC problem is NP-complete for orthogonal polygons with holes; that is, we show that the following problem is NP-complete:

MCSC With Holes
Input: An orthogonal polygon P, possibly with holes and an integer k.
Output: Yes, if there exist k orthogonal line segments inside P that guard P entirely; No, otherwise.

Given a candidate solution for the MCSC With Holes problem, we can verify the solution in polynomial time by checking whether the union of the visibility regions of cameras in the solution is P. Therefore, the problem is in NP. We show NP-hardness by a reduction from the *minimum hitting of horizontal unit segments* problem, which we call the Min Segment Hitting problem, defined as follows [24]:

Min Segment Hitting
Input: n pairs (a_i, b_i), $i = 1, \ldots, n$, of integers and an integer k.
Output: Yes, if there exist k orthogonal lines $l_1, \ldots, l_k$ in the plane, i.e., for each i, l_i is horizontal or vertical, such that each line segment $[(a_i, b_i), (a_i+1, b_i)]$ is hit by at least one of the lines; No, otherwise.

Hassin and Megiddo [24] proved that the Min Segment Hitting problem is NP-complete. Let I be an instance of the Min Segment Hitting problem, where I is a set of n horizontal unit-length segments with integer coordinates. We construct an orthogonal polygon P (with holes) such that there exists a set of k orthogonal lines that hit the segments in I if and only if there exists a set C of $k+4$ orthogonal line segments inside P that collectively guard P. Throughout this section, we refer to the segments in I as *unit segments* and to the segments in C as *line segments*.

Gadgets. Without loss of generality, assume that no two unit segments overlap each other. We observe that any two unit segments in I can share at most one point, which must be a common endpoint of the two unit segments. For each unit segment $s_i \in I$, $1 \leq i \leq n$, we denote the left endpoint of s_i by (a_i, b_i)

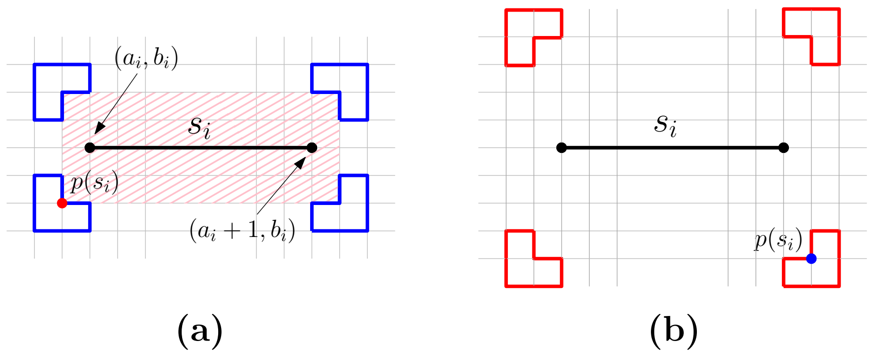


Figure 3: The L-holes associated with a line segment $s_i \in I$, where (a) a_i is even, and (b) a_i is odd.

and, therefore, the right endpoint of s_i is $(a_i + 1, b_i)$. Let $L(s_i)$ denote the set of unit segments in I for which the x-coordinate of their left endpoints is equal to a_i. Moreover, let $N(s_i)$ denote the set of unit segments in I that have at least one endpoint with x-coordinate equal to a_i or $a_i + 1$. Our reduction refers to an L-hole, which we define as an orthogonal polygon with six vertices such that exactly one of them is reflex. We constrain each grid cell to have size $\frac{1}{12} \times \frac{1}{12}$. The L-holes have variable size; in order to specify the size of L-holes, we first need the following notation.

Let $I = \{O_1, O_2, \ldots, O_r\}$ be a partition of I such that the left endpoints of all the unit segments in O_m, for each $1 \leq m \leq r$, have the same x-coordinate. Consider the set O_m, for some $1 \leq m \leq r$, and let $|O_m| = t$, where $t = |L(s_i)|$ for any unit segment $s_i \in O_m$. The idea is to associate exactly four L-holes for each unit segment $s \in O_m$ depending on t and the parity of the x-coordinate of the left endpoint of s.

Case 1. $t = 1$. Let s_i be the only unit segment in O_m. If a_i is even, then Figure 3(a) shows the L-holes associated with s_i. If a_i is odd, then Figure 3(b) shows the L-holes associated with s_i. In both cases, each L-hole has height and width of 1/6. However, in the case a_i is even (resp., is odd), the L-holes are located such that the vertical distance between any point on an L-hole and s_i is at least 1/12 (resp., 3/12). Note the red vertex on the bottom left L-hole of s_i in Figure 3(a) and the blue vertex on the bottom right L-hole of s_i in Figure 3(b); we call this vertex the *visibility vertex* of s_i, which we denote $p(s_i)$. The L-holes associated with s_i do not interfere with the L-holes associated with the line segments in $N(s_i)$ because for any unit segment $s_j \in N(s_i)$ the vertical distance d between s_i and s_j is either zero or at least one. If $d \geq 1$, then it is trivial that the L-holes of s_i do not interfere with those of s_j. Now, suppose that s_i and s_j share a common endpoint; that is $d = 0$. Since s_i and s_j have unit lengths, and a_i and a_j have different parities, the L-holes associated with s_i and s_j do not interfere with each other. Figure 4 shows an example of such two unit segments and their corresponding L-holes.

Case 2. $t \geq 2$. Let $\{s_{i_1}, s_{i_2}, \ldots, s_{i_t}\}$ be the unit segments in O_m ordered from top to bottom. We associate each unit segment in O_m with four L-holes similar to Case 1: two left L-holes around its left endpoint and two right L-holes around its right endpoint. Here, each L-hole has height and width equal to $\frac{1}{6t}$; note that the size of L-holes remain polynomial in the size of the input. Informally, the idea is to locate the L-holes in such a way that the L-holes of every two unit segments in O_m do not block the vertical visibility of their horizontal edges. To this end, consider the vertical slab of size two grid units to the left of the unit segments in O_m; see Figure 5(a) for an illustration. We first partition the slab into t vertical subslabs; note that the total width of all L-holes on one side is $1/6$. Consider the subslabs from left to right. Then, we consider the unit segments in O_m from top to bottom and locate their left L-holes in separate subslabs from left to right; see Figure 5(a). Note that the left L-holes of *each* unit segment in O_m lie in the same slab (i.e., they are aligned with each other). Next, we locate the right L-holes of the unit segments in O_m analogously by considering the vertical slab of size two grid units to the right of the unit segments in O_m, and then partitioning the slab into t vertical subslabs. Then, consider the subslabs from right to left: we locate the right L-holes of the unit segments in O_m from top to bottom in separate subslabs from *right to left* (as opposed to the left L-holes that were located in separate subslabs from *left to right*). See Figure 5(b) for an illustration. We emphasize that, similar to Case 1, the vertical distance between any point on an L-hole and its corresponding unit segment is at least $1/12$ or $3/12$ depending on the parity of the x-coordinate of the left endpoint of the unit segment. Note that Figure 5 does not show this distance properly due to space constraints. Our construction ensures the following observation:

Observation 5. Let s be a unit segment in O_m. Then, the visibility region of the topmost horizontal edge of the upper left L-hole (resp., the lowest horizontal edge of the lower left L-hole) of s is unbounded from above (resp., from below); i.e., it is not blocked by any other L-hole. Similarly, the visibility region of the topmost horizontal edge of the upper right L-hole (resp., the lowest horizontal edge of the lower right L-hole) of s is unbounded from above (resp., from below). See the shaded regions in Figure 5 for an example.

Observation 5 holds even if the unit segments in Case 2 share a common endpoint; this is illustrated in Figure 6. We now describe the reduction.

Reduction. Given an instance I of the Min Segment Hitting problem, we first associate each unit segment in $s_i \in I$ with four L-holes as described above. After adding the corresponding L-holes, we enclose I a rectangle such that all unit segments and the L-holes associated with them lie in its interior. Finally, we create four small rectangles, each located on one corner of the bigger rectangle as shown in Figure 7; note the *visibility vertex* of each smaller rectangle shown in red. The visibility vertex of a smaller rectangle is only visible to the line segments that guard the interior of the smaller rectangle. Moreover, (i) any orthogonal line segment that guards one of the smaller rectangles cannot intersect any of the unit segments in I, and (ii) there exists a dent on the entrance to each smaller rectangle to ensure that no orthogonal line segment in

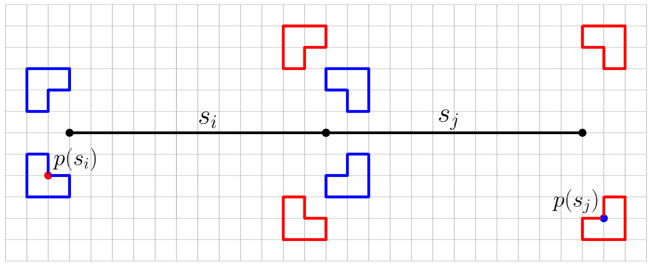


Figure 4: An illustration of the L-holes associated with two line segments s_i and s_j in I that share a common endpoint such that a_i is even and $L(s_i) = \emptyset$.

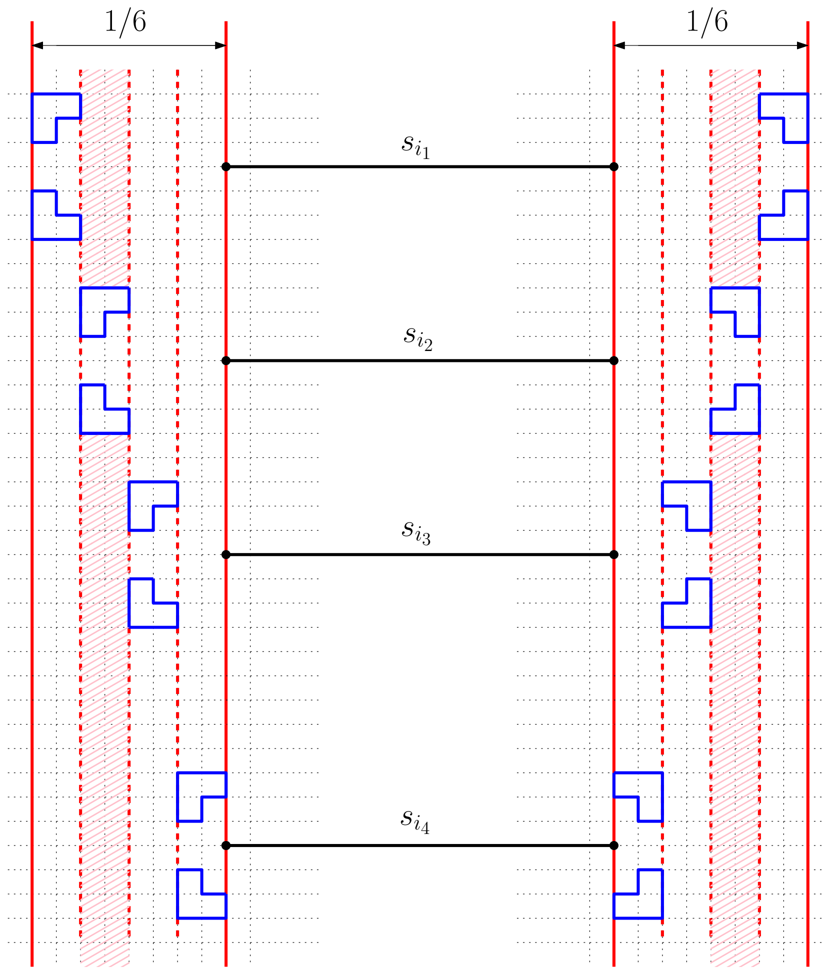
P can see more than one visibility vertex of the smaller rectangles. See Figure 7 for a complete example of the reduction. Let P be the resulting orthogonal polygon. We now show the following lemma.

Lemma 6. *There exist k orthogonal lines such that each unit segment in I is hit by one of the lines if and only if there exists $k+4$ orthogonal line segments inside P that collectively guard P.*

PROOF. ($\Rightarrow$) Suppose that there exists a set S of k lines such that each unit segment in I is hit by at least one line in S. We can assume that every line $L \in S$ hits at least one unit segment in I; otherwise, we can remove L from S without affecting the feasibility of S. Let $L \in S$ and let $L_P = L \cap P$. If L is horizontal, then L, and therefore L_P, does not cross any L-hole inside P. Similarly, if L is vertical and passes through an endpoint of some unit segment(s) in I, then neither L nor L_P passes through the interior of any L-hole in P.[4] Now, suppose that L is vertical and passes through the interior of a unit segment $s \in I$. Translate L_P horizontally such that it passes through the midpoint of s. Since unit segments have endpoints on adjacent integer grid point, L_P still crosses the same set of unit segments of I as it did before this move. Moreover, this ensures that L_P does not cross any L-hole inside P. Consider the set $S' = \{L_P \mid L \in S\}$.

The line segments in S' cannot guard the visibility vertex of any of the smaller rectangles. Moreover, if all line segments in S' are vertical or all are horizontal, then they cannot collectively guard the bigger rectangle entirely. Next, we add four orthogonal line segments into S' each of which guards one of the smaller rectangles entirely. We choose each of these line segments in such way that it guards one smaller rectangle and hits the dent that is on the other corner of the bigger rectangle; see the red dashed line segments in Figure 7. We call these four line segments the *boundary guards*. Let S'' be the union of S' and the boundary guards. Clearly, $|S''| = k + 4$. We now show that S'' is a feasible solution for the MCSC problem on P. Let p be a point in P. If p is inside one of the smaller rectangles, then it must be guarded by one of the boundary guards. Recall the vertical subslabs obtained by dividing the vertical

[4]Note that it is possible for L to pass through the boundary of some L-hole.



(a) **(b)**

Figure 5: The unit segments in O_m, where $|O_m| = 4$, and the associated L-holes. (a) The left slab (shown as the two solid vertical red lines) is divided into four subslabs (separated by the three dashed vertical red lines) and the left L-holes of the unit segments in O_m (ordered from top to bottom) are located from left to right in separate subslabs. (b) The right slab is divided into four subslabs and the right L-holes of the unit segments in O_m (ordered from top to bottom) are located from right to left in separate subslabs. Note the *unbounded* shaded regions below and above the L-holes associated with unit segment s_{i_2}.

slabs when locating the L-holes; note that each of such subslabs contains exactly two L-holes.

- Suppose that p lies in one of the subslabs. If it is between the L-holes associated with a unit segment $s \in I$, then p is guarded by the line segment in S' whose corresponding line intersects s (if the line is horizontal), or p

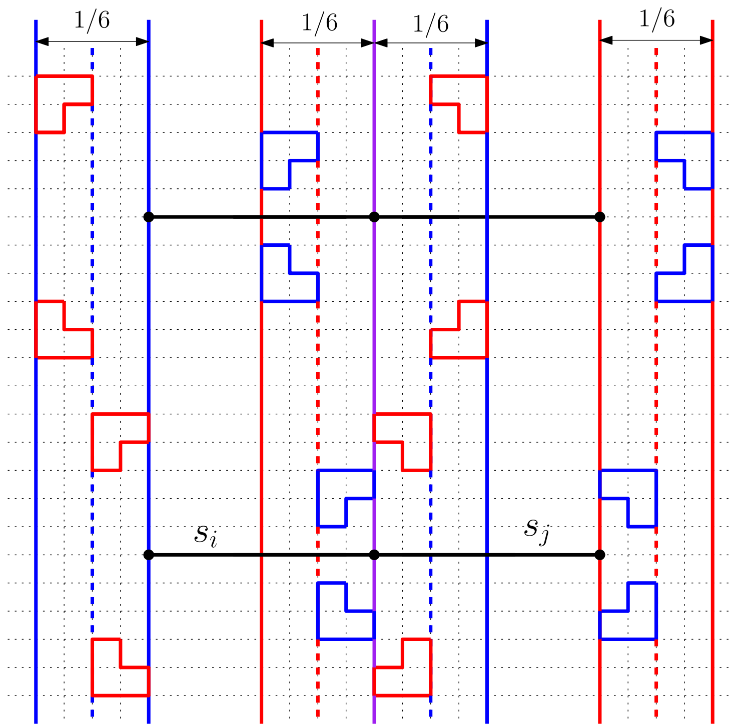
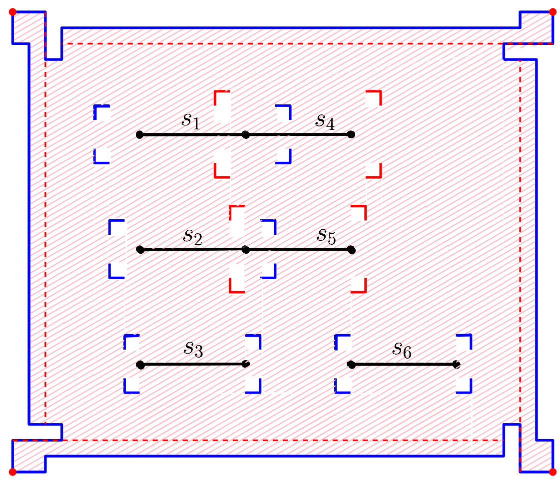


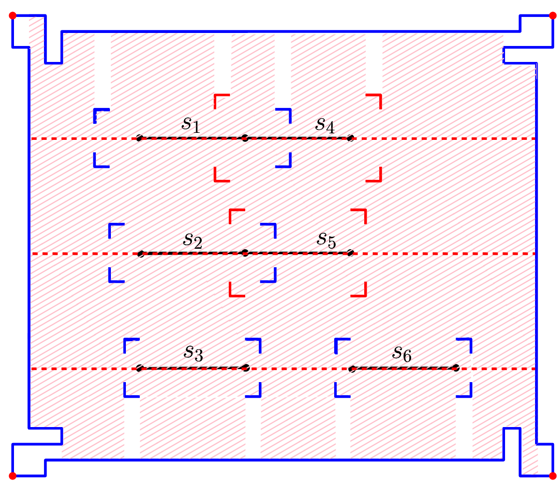
Figure 6: An illustration of the L-holes associated with two line segments s_i and s_j in I that share a common endpoint such that a_i is odd and $|L(s_i)| = 1$ (i.e., the only unit segment in $L(s_i)$ is the one that is directly above s_i).

is guarded by the line segment in S' whose corresponding line intersects s or the unit segment(s) to the left or right of s (if the line is vertical). Otherwise, by Observation 5, one of the horizontal boundary guards must see p.

- Now, suppose that p does not lie in a vertical subslab. Then, if p lies inside the rectangle induced by the reflex vertices of the L-holes associated with s (i.e., the shaded rectangle in Figure 3(a) for instance), for some unit segment $s \in I$, then there are four cases: either (i) p is guarded by the line segment in S' whose corresponding line intersects s (if the line is horizontal), (ii) p is guarded by a line segment in S' whose corresponding line intersects a unit segment in $L(s)$ that is above s (if that line is horizontal), (iii) p is guarded by the line segment in S' whose corresponding line intersects the unit segment to the left or right of s (if the line is vertical), or (iv) p is guarded by one of the boundary guards if neither the line segment described in (i) nor the one described in (iii) can see p. This is also true for two unit segments with a common endpoint because for those regions that the L-holes associated with such two unit segments block the visibility, point p is visible to at least one of the boundary guards (iv). If



(a)



(b)

Figure 7: A complete example of the reduction, where $I = \{s_1, s_2, \ldots, s_6\}$, with the assumption that a_1 is even. Each line segment that has a bend represents an L-hole associated with a unit segment. Note the red vertex inside each smaller rectangle. This vertex, which we call the *visibility vertex* of the smaller rectangle, is only visible to the line segments that guard the interior of the smaller rectangle, which in turn cannot intersect any unit segment in I. The shaded regions indicate (a) the visibility region of the boundary guards, and (b) the visibility region of three horizontal sliding cameras induced by a solution to the Min Segment Hitting problem.

p is not interior to any of such rectangles, then it is seen by the boundary of the polygon in at least one orthogonal direction and, therefore, one of the boundary guards sees p.

Therefore, the set S'' is a feasible solution for the MCSC problem on P such that $|S''| = k + 4$.

($\Leftarrow$) Now, suppose that there exists a set M of $k+4$ orthogonal line segments contained in P that collectively guard P. For any line segment $c \in P$, we denote the line induced by c by L_c. We now describe how to find k lines that form a solution to instance I by moving the line segments in M accordingly such that each unit segment in I is hit by at least one of the corresponding lines. From the construction of polygon P, no line segment in M can see more than one visibility vertex of the smaller rectangles. Thus, let M' be the set of the four line segments in M each of which guards a visibility vertex of a smaller rectangle. We know that no line segment in M' can see $p(s)$, for all $s \in I$. Therefore, for each unit segment $s \in I$ in order, consider a line segment $\ell \in M \setminus M'$ that guards $p(s)$; let ℓ' be the maximal line segment inside P that is aligned with ℓ. Note that ℓ' must intersect R, the rectangle induced by the reflex vertices of the L-holes associated with unit segment s (see the shaded rectangle in Figure 3 for an example). If ℓ' is horizontal and $L_{\ell'}$ does not align with s, then move ℓ' accordingly up or down until it aligns with s. Thus, $L_{\ell'}$ is a line that hits s. Now, suppose that ℓ' is vertical. If ℓ' intersects s, then $L_{\ell'}$ also intersects s. If ℓ' does not intersect s, then the endpoints of ℓ' must be on the boundary of two of the L-holes associated with s; this is because the only way for a maximal line segment to see $p(s)$ is to intersect R and, therefore, either intersect s or be bounded by the L-holes of s. Thus, we move ℓ' horizontally to the left or to the right until it hits s. Therefore, $L_{\ell'}$ is a line that hits s after this move.

Therefore, we have obtained exactly one line from each line segment in $M \setminus M'$ such that each unit segment in I is hit by at least one of the lines. This completes the proof of the lemma. $\square$

By Lemma 6 we obtain the main result of this section:

Theorem 7. *The* MCSC WITH HOLES *is* NP-*complete.*

5. A 2-Approximation Algorithm for [X]-Star-Shaped Orthogonal Polygons

In this section we present a polynomial-time 2-approximation algorithm for the MCSC problem on [X]-star-shaped orthogonal polygons. Let $X \in \{NE, NW, SE, SW\}$, where N, S, E and W denote the four main compass directions. An orthogonal polygon P is called [X]-star-shaped, for some $X \in \{NE, NW, SE, SW\}$, if there exist a convex vertex $p \in P$ such that, for all points $q \in P$, there exist an orthogonal path from p to q using only compass directions defined by X. For example, the polygon shown in Figure 8 is a [NE]-star-shaped orthogonal polygon because there exist an orthogonal path from p to q with only the North (N) and East (E) compass directions for all points $q \in P$; by definition this path is x- and y-monotone. Note that such a point p is unique because otherwise there cannot be an orthogonal path from at least one such point p to the other point p' with only the compass directions defined by X; we call such a point p in P the *kernel* of P.

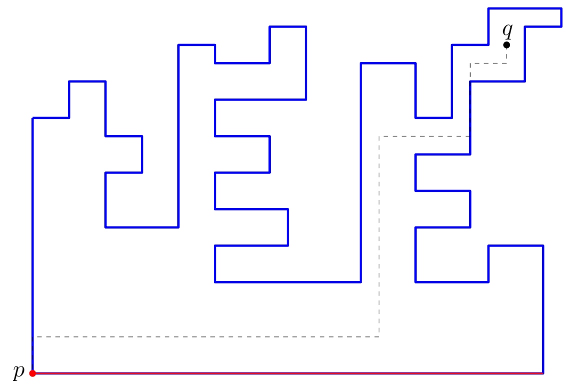


Figure 8: An example of a [NE]-star-shaped orthogonal polygon P. Point p is the kernel point and the bottom most edge of P is the kernel segment of P.

5.1. Notation

Let s be the maximal horizontal line segment inside P that passes through the kernel of P. For the rest of this section, we refer to the kernel of P as the *kernel point* of P and to s as the *kernel segment* of P. For example, the point p in Figure 8 is the kernel point of P and the bottom most horizontal edge of P is the kernel segment of P. The kernel point of P is the left endpoint of the kernel segment of P. We denote the line and the maximal line segment in P aligned with an edge e of P by $\ell(e)$ and $s(e)$, respectively. For a subpolygon P' of P, we denote the portion of P' that is seen by an orthogonal line segment s in P by $V_{P'}(s)$. For a point $p \in P$, we denote the x- and y-coordinates of p by $x(p)$ and $y(p)$, respectively. For a horizontal line segment s inside P, we denote its left and right endpoints by $left(s)$ and $right(s)$, respectively. For two points $\{p, q\} \in P$, we denote *shortest orthogonal path* between p and q in P by $SOP(p, q)$. The following observation is by Katz and Morgenstern [7]:

Observation 6. Let p and q be two points inside P such that all $SOP(p, q)$ have at least three bends. Then, no orthogonal line segment inside P can see both p and q.

Throughout this section, we assume that P is a [NE]-star-shaped orthogonal polygon; the algorithm for other values of $X \in \{NE, NW, SE, SW\}$ is analogous. In the following, we describe the algorithm in which we denote the solution set being constructed by S, which is initially empty.

5.2. Algorithm

Let p be the kernel point of P and let s be the kernel segment of P. Let s' be the highest horizontal line segment (i.e., the one with largest y-coordinate) in P such that $V_P(s) \subseteq V_P(s')$. Add s' to S. Next, if $V_P(s') = P$, then terminate. Otherwise, note that $P \setminus V_P(s')$ is a set of subpolygons $M = \{P_1, P_2, \dots\}$ each of which is a [NE]-star-shaped orthogonal polygon and which has exactly one edge in common with the visibility polygon $V_P(s')$. Let $M' \subseteq M$ denote a maximal set of all $P_i \in M$ such that the union of all P_i in M' is guarded by a single vertical line segment $s_i \in P$ that intersects s'. For each such subset M', we add the leftmost such line segment s_i to S. Then, we solve the problem recursively on each subpolygon $P_j \in M \setminus M'$. See Figure 9 for an illustration of the algorithm.

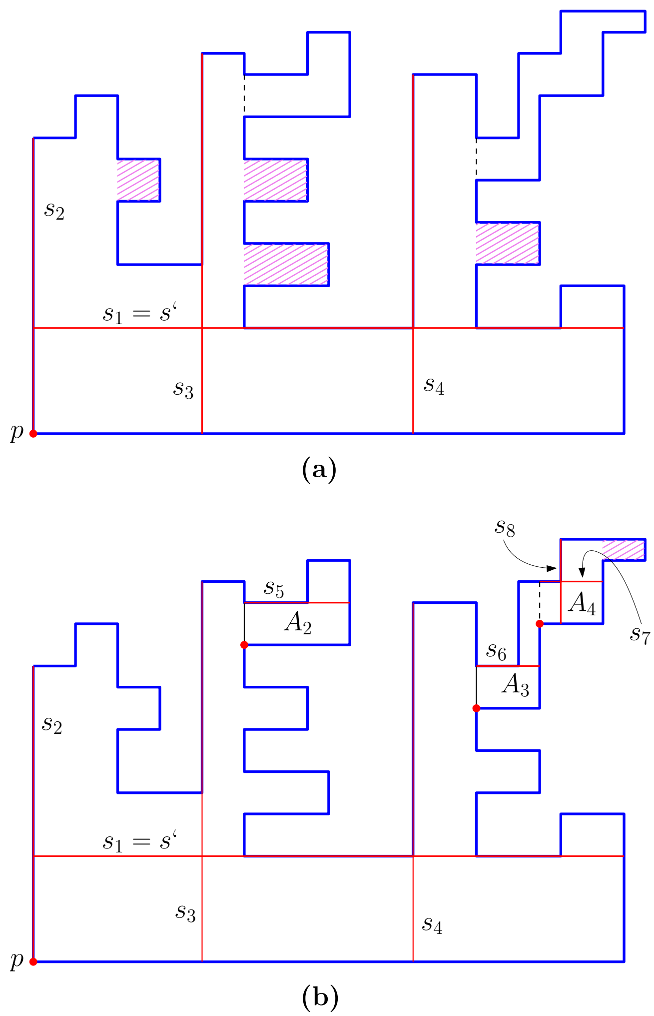


Figure 9: An example illustrating the execution of the algorithm on a [NE]-star-shaped polygon P. (a) In the first iteration of the algorithm the line segments s_1, s_2, s_3 and s_4 are added to S. (b) The second iteration of the algorithm in which we first guard subpolygon A_2 by s_5 and then the subpolygon A_3 by s_6. Finally, in the third iteration of the algorithm, the line segments s_7 and s_8 are added to guard the subpolygon A_4. The solution $S = \{s_1, s_2, \ldots, s_8\}$ is returned at the end; note that any feasible solution requires at least 7 sliding cameras.

5.3. Running Time

We now examine the running time of the algorithm. Consider the first iteration of the algorithm. The kernel point of P can be computed in $O(n)$ time, where n is the number of the vertices of P [25]. Moreover, the visibility polygon of the kernel segment of P and so that of s' is also computed in $O(n)$ time [26]. So, we can then compute each unguarded subpolygon P_i in $O(n)$ time [26]. Next, for each unguarded subpolygon P_i, we can check to see if P_i is guarded entirely by a single vertical line segment s_i in $O(n_i)$ time, where n_i is the number of the vertices of P_i; such a line segment s_i can be computed in $O(n_i \log n_i)$ time using a sweep line algorithm. Therefore, the first iteration of the algorithm can be completed in $O(n^2 \log n)$ overall time. Since the subpolygon in each recursive step of the algorithm has at least one vertex less than its parent subpolygon, the algorithm has at most $O(n)$ recursive steps. Therefore, the algorithm can be completed in $O(n^3 \log n)$ overall time.

5.4. Algorithm Correctness

We first show that S, the solution computed by the algorithm is a feasible solution. In each recursive step, the visibility polygon of a horizontal line segment inside P is guarded. Potentially some of the unguarded subpolygons are also guarded by the vertical sweeping line, and then the remaining unguarded subpolygons will be guarded recursively until the entire polygon is guarded. Therefore, we have the following lemma:

Lemma 8. *The set S is a feasible solution for the MCSC problem on P.*

We next prove that $|S| \leq 2 \cdot |OPT|$, where OPT is an optimal solution for the MCSC problem on P. To this end, we first associate a rooted tree T with the solution S as follows: the root r of T corresponds to the first horizontal line segment s' added to S. Each subpolygon $P_i \in M$ of $P \setminus V_P(s')$ has exactly one edge in common with the subpolygon $P_0 = V_P(s')$ (that is not an edge of P); we call this edge the *window segment* between P_i and P_0. The children of r are defined as follows. First, the root r has a child that is a leaf node for each vertical line segment s_i added to S to guard $P_i \in M'$. Next, for each subpolygon P_j that remains unguarded, let T_j be the tree whose root corresponds to the first horizontal line segment located in P_j. Then, the root of T_j becomes a child of T. Figure 10 shows the tree T associated with the solution S of the polygon in Figure 9(b).

We denote the parent of a node $v \in T$ by $par(v)$. Let $N(T)$ denote the number of nodes in T; note that $N(T) \in O(n)$. For each node $u \in T$, let $s(u)$ be the line segment in S that corresponds to u.

Observation 7. Let u be a node in T. If u is a non-leaf node, then $s(u)$ is horizontal.

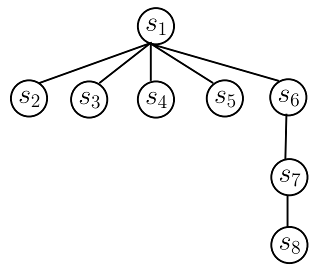


Figure 10: The tree T associated with the solution S of the polygon in Figure 9(b). Note that the subpolygon $P(u_i)$, where u_i is the node in T corresponding to s_i, for all $2 \leq i \leq 4$, is shaded in Figure 9(a).

Observation 7 implies that if $s(u)$ is vertical, then u must be a leaf node in T. We associate a subpolygon $P(u)$ with each node $u \in T$ as follows. If $s(u)$ is horizontal, then $P(u)$ is the visibility polygon of $s(u)$ in P. If $s(u)$ is vertical, then

$$P(u) = V_P(s(u)) \setminus \bigcup_{\substack{s \in S, \text{ and} \\ s \text{ is horizontal}}} V_P(s).$$

That is, $P(u)$ is the set of all points in P that are visible to $s(u)$, but not to any horizontal line segment in S. Note that $P(u)$ is not necessarily connected. We now define

$$A_i = \bigcup_{u \in I_i} P(u),$$

where I_i is the set of all line segments in S that are computed in ith step of the recursion. See A_2, A_3 and A_4 in Figure 9(b) for an example. We now show that $|OPT| \geq 1/2 \cdot N(T)$.

Lemma 9. *Let OPT be an optimal solution for the MCSC problem on P. Then, $|OPT| \geq 1/2 \cdot N(T)$*

Proof. Let h denote the height of the tree T, where the root r is at height 0. To prove the lemma, we first compute a subset U of nodes in T whose cardinality is at least half of $N(T)$ and will then show that $|OPT| \geq |U|$; that is, $|OPT| \geq |U| \geq 1/2 \cdot N(T)$.

Consider the following recursive operation on T. Let u be a leaf node at height h and let v denote its parent. First, add u and all of its siblings to U; note that node v is not added to U. Next, remove v and all of its children from T and let T' be the resulting tree. Perform the operation recursively on T' until $N(T') = 0$. For example, $U = \{s_8, s_6, s_5, s_4, s_3, s_2\}$ for the tree T shown in Figure 10. Consider the first step of this recursive operation. Observe that the node u and all of its siblings that are added to U in this step are all the leaf nodes of T because u is at height h of T. Moreover, we remove the node v from T only because it has at least one child u, which has been already added to U. In general, considering tree T' in some step of the recursion, at least one node u of T' is added to U for every node v that is removed from T'. Therefore,

$$|U| \geq 1/2 \cdot N(T). \tag{1}$$

We now prove that $|OPT| \geq |U|$. To this end, we first associate a fixed point $p_u \in P(u)$ with each node $u \in U$, and then will show that for every pair of nodes $u, v \in U$, $SOP(p_u, p_v)$ has at least three bends. By Observation 6, this implies that $|OPT| \geq |U|$. For each node $u \in U$, if $s(u)$ is vertical, then let p_u

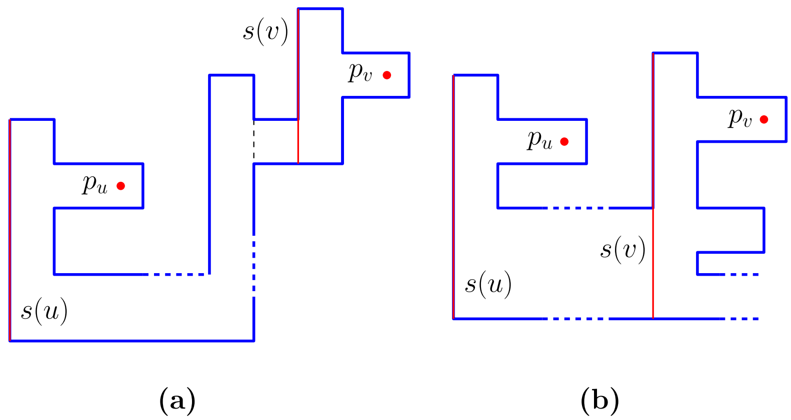


Figure 11: An illustration in support of the first case of Lemma 9.

be a point in $P(u)$ and $p_u \notin P(v)$ for all siblings v of u in T. Note that at least one such point p_u must exist because otherwise we would have not added $s(u)$ into S.

Now, suppose that $s(u)$ is horizontal. If P_i is the input polygon P, then let p_u be the kernel point of P. Otherwise, let w be the parent of u (note that w exists as u is not the root node), and consider the window segment between $P(u)$ and $P(w)$. Then, let p_u be a point in $P(u)$ that is not visible to this window segment; at least one such point exists as otherwise $s(u)$ would have not been horizontal.

We now prove that $SOP(p_u, p_v)$ has at least three bends, for every pair of nodes $u, v \in U$. To this end, let u and v be two nodes in U. We consider three cases depending on the orientations of $s(u)$ and $s(v)$.

- $s(u)$ and $s(v)$ are both vertical. Then, either $s(u)$ and $s(v)$ lie in the same subpolygon A_i, for some i, or they lie in different subpolygons. If $s(u)$ and $s(v)$ lie in different subpolygons, then Figure 11(a) shows that $SOP(p_u, p_v)$ must have at least three bends. Now, suppose that $s(u)$ and $s(v)$ lie in the same subpolygon A_i, for some i. Since p_u is not visible to $s(v)$ and p_v is not visible to $s(u)$, it is easy to see that $SOP(p_u, p_v)$ must have at least three bends; see Figure 11(b) for an illustration.

- $s(u)$ and $s(v)$ are both horizontal. First, note that neither u is the parent of v in T nor v is the parent of u in T because we never add a node and its parent into U. Therefore, w.l.o.g., suppose that the path between u and v in T passes through $par(u)$, the parent of u. By Observation 7, line segment $s(par(u))$ is horizontal and, therefore, $s(u), s(v)$ and $s(par(u))$ are all horizontal. Consider $SOP(p_u, p_v)$; clearly, $SOP(p_u, p_v)$ must intersect $P(par(u))$, the subpolygon corresponding to node $par(u)$. Let P^u and P^v denote the portion of $SOP(p_u, p_v)$ that lie in $P(u)$ and $P(v)$, respectively. Since p_u is not visible to the window segment between $P(u)$ and $P(par(u))$,

P^u must have at least one bend before entering $P(par(u))$. Similarly, P^v must have at least one bend before entering $P(par(u))$. Since P is [NE]-star-shaped, P^u and P^v can enter $P(par(u))$ either both from right or one from right and the other from left. (i) If both P^u and P^v enter $P(par(u))$ from right, then clearly they require at least one bend to reach each other and hence $SOP(p_u, p_v)$ has at least three bends in total. (ii) If P^u and P^v enter $P(par(u))$ from opposite directions, then we show in the following that they still require at least one bend to reach each other and so $SOP(p_u, p_v)$ has at least three bends in total.

Now, suppose that P^u and P^v could be connected to each other inside $P(par(u))$ with no additional bend. Since they enter $P(par(u))$ from opposite directions, the portion of $SOP(p_u, p_v)$ inside $P(par(u))$ is just a single horizontal line segment s. But, then s could be extended into $P(u)$ and spans the kernel segments of both $P(par(u))$ and $P(u)$. That is, the algorithm would have merged $P(par(u))$ and $P(u)$ into one subpolygon by selecting s as the line segment s' in the algorithm's description — a contradiction.

- $s(u)$ and $s(v)$ have different orientations. Suppose w.l.o.g. that $s(u)$ is vertical and $s(v)$ is horizontal. Since $s(u)$ is vertical, by Observation 7, node u must be a leaf node in T. Moreover, node v is an internal node and it cannot be the parent of u. Therefore, $s(par(u))$ is horizontal and different than $s(v)$. Consider $SOP(p_u, p_v)$, and let P^u and P^v denote the portion of $SOP(p_u, p_v)$ that lies in $P(u)$ and $P(v)$, respectively. First, it is clear that $SOP(p_u, p_v)$ must intersect $P(par(u))$. Now, similar to the previous case, we can show that P^v must have at least one bend before entering $P(par(u))$, but P^u may have no bends. Nevertheless, we show that P^u and P^v require at least two bends inside $P(par(u))$ in order to reach each other. To this end, we first note that $SOP(p_u, p_v)$ cannot have only one bend inside $P(par(u))$ because both P^u and P^v enter $P(par(u))$ horizontally (i.e., $SOP(p_u, p_v)$ is normal to both window segments between $P(u)$ and $P(par(u))$, and between $P(v)$ and $P(par(v))$); hence, they need an even number of bends to reach each other. Now, suppose for a contradiction that they reach each other with no bends. By an argument analogous to the one shown in the previous case, we can conclude that the portion of $SOP(p_u, p_v)$ inside $P(par(u))$ is just a single horizontal line segment s, and so the algorithm would have merged $P(v)$ and $P(par(u))$ into one subpolygon by selecting s as the line segment s' in the algorithm's description — a contradiction. This means that $SOP(p_u, p_v)$ has at least two bends inside $P(par(u))$, and hence three bends in total.

Therefore, $SOP(p_u, p_v)$ has at least three bends for every pair of nodes $u, v \in U$, and so by Observation 6,

$$|OPT| \geq |U|. \tag{2}$$

By (1) and (2), $|OPT| \geq 1/2 \cdot N(T)$. □

By the definition of tree T, the number of nodes in T corresponds to the number of line segments in S; that is, $|S| = N(T)$. Therefore, by Lemma 9, we have the following result:

Theorem 10. *There exist an $O(n^3 \log n)$-time 2-approximation algorithm for the MCSC problem on any [NE]-star-shaped orthogonal polygon with n vertices.*

Our algorithm can be applied analogously to get a 2-approximation algorithm for [X]-star-shaped orthogonal polygons for all $X = \{NE, NW, SE, SW\}$. Therefore, we obtain the main result of this section:

Theorem 11. *There exist an $O(n^3 \log n)$-time 2-approximation algorithm for the MCSC problem on any [X]-star-shaped orthogonal polygon with n vertices, where $X \in \{NE, NW, SE, SW\}$.*

6. Conclusion

In this paper, we studied the problem of guarding an orthogonal polygon P using sliding cameras that was introduced by Katz and Morgenstern [7]. We considered two variants of this problem: the MCSC problem (in which the objective is to minimize the number of sliding cameras used to guard P) and the MLSC problem (in which the objective is to minimize the total length of trajectories along which the cameras travel).

We gave a polynomial-time algorithm that solves the MLSC problem exactly even for orthogonal polygons with holes, answering a question posed by Katz and Morgenstern [7]. We also showed that the MCSC problem is NP-complete when P contains holes, which partially answers another question posed by Katz and Morgenstern [7]. Furthermore, we gave an $O(n^3 \log n)$-time 2-approximation algorithm for the MCSC problem on [X]-star-shaped orthogonal polygons, where n is the number of the vertices.

Although we settled the complexity of the MLSC problem, the complexity of the MCSC problem on simple orthogonal polygons remains open. For small constants $\alpha > 0$, giving α-approximation algorithms for the MCSC problem on any orthogonal polygon is another direction for future work. Finally, does the MCSC problem admit a PTAS or the problem is APX-hard?

Acknowledgement

The authors thank Mark de Berg and Matya Katz for insightful discussions on the sliding cameras problem.

References

[1] S. Durocher, S. Mehrabi, Guarding orthogonal art galleries using sliding cameras: algorithmic and hardness results., in: Proc. MFCS, 2013, pp. 314–324, LNCS 8087.

[2] Y. Amit, J. S. B. Mitchell, E. Packer, Locating guards for visibility coverage of polygons, Int. J. Comput. Geometry Appl. 20 (5) (2010) 601–630.

[3] J. O'Rourke, Art gallery theorems and algorithms, Oxford University Press, 1987.

[4] J. Urrutia, Art gallery and illumination problems, in: Handbook of Comp. Geom., North-Holland, 2000, pp. 973–1027.

[5] F. Hoffmann, On the rectilinear art gallery problem, in: Proc. ICALP, 1990, pp. 717–728.

[6] T. C. Biedl, M. T. Irfan, J. Iwerks, J. Kim, J. S. B. Mitchell, The art gallery theorem for polyominoes, Disc. & Comp. Geom. 48 (3) (2012) 711–720.

[7] M. J. Katz, G. Morgenstern, Guarding orthogonal art galleries with sliding cameras, Int. J. of Comp. Geom. & App. 21 (2) (2011) 241–250.

[8] S. Mehrabi, Geometric optimization problems on orthogonal polygons: hardness results and approximation algorithms, Ph.D. thesis, University of Manitoba, Winnipeg, Canada (August 2015).

[9] J. O'Rourke, Art gallery theorems and algorithms, Oxford University Press, Inc., New York, NY, USA, 1987.

[10] V. Chvátal, A combinatorial theorem in plane geometry, J. Comb. Theory, Ser. B 18 (1975) 39–41.

[11] J. Kahn, M. M. Klawe, D. J. Kleitman, Traditional galleries require fewer watchmen, SIAM J. on Algebraic Disc. Methods 4 (2) (1983) 194–206.

[12] D. T. Lee, A. K. Lin, Computational complexity of art gallery problems, IEEE Trans. on Info. Theory 32 (2) (1986) 276–282.

[13] D. Schuchardt, H. Hecker, Two NP-hard art-gallery problems for ortho-polygons, Math. Logic Quarterly 41 (2) (1995) 261–267.

[14] M. J. Katz, G. S. Roisman, On guarding the vertices of rectilinear domains, Comput. Geom. 39 (3) (2008) 219–228.

[15] R. Motwani, A. Raghunathan, H. Saran, Covering orthogonal polygons with star polygons: the perfect graph approach, in: Proc. ACM SoCG, 1988, pp. 211–223.

[16] C. Worman, J. M. Keil, Polygon decomposition and the orthogonal art gallery problem, Int. J. of Comp. Geom. & App. 17 (2) (2007) 105–138.

[17] A. Lingas, A. Wasylewicz, P. Zylinski, Linear-time 3-approximation algorithm for the r-star covering problem, in: Proc. WALCOM, 2008, pp. 157–168.

[18] S. Seddighin, Guarding polygons with sliding cameras, Master's thesis, Sharif University of Technology (2014).

[19] S. P. Fekete, J. S. B. Mitchell, Terrain decomposition and layered manufacturing, Int. J. of Comp. Geom. & App. 11 (6) (2001) 647–668.

[20] R. M. Karp, Reducibility among combinatorial problems, in: Complexity of Computer Computations, 1972, pp. 85–103.

[21] D. König, Gráfok és mátrixok, Matematikai és Fizikai Lapok 38 (1931) 116–119.

[22] D. Z. Chen, H. Wang, Visibility and ray shooting queries in polygonal domains, in: Algorithms and Data Structures - 13th International Symposium, WADS London, ON, Canada, August 12-14, 2013. Proceedings, 2013, pp. 244–255.

[23] V. V. Williams, Multiplying matrices faster than coppersmith-winograd, in: STOC, 2012, pp. 887–898.

[24] R. Hassin, N. Megiddo, Approximation algorithms for hitting objects with straight lines, Disc. App. Math. 30 (1) (1991) 29–42.

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

[26] D. Avis, G. T. Toussaint, An optimal algorithm for determining the visibility of a polygon from an edge, IEEE Trans. Computers 30 (12) (1981) 910–914.