Covering grids and orthogonal polygons with periscope guards

Laxmi P. Gewali*

Department of Computer Science, University of Nevada, Las Vegas, NV 89154, USA

Simeon Ntafos**

Computer Science Program, University of Texas at Dallas, Richardson, TX 75080, USA

Communicated by Godfried Toussaint
Submitted 10 July 1992
Accepted 27 October 1992

Abstract


The problem of finding minimum guard covers is NP-hard for simple polygons and open for simple orthogonal polygons. Alternative definitions of visibility have been considered for orthogonal polygons. In this paper we try to determine the complexity of finding guard covers in orthogonal polygons by considering periscope visibility. Under periscope visibility, two points in an orthogonal polygon are visible if there is an orthogonal path with at most one bend that connects them without intersecting the exterior of the polygon. We show that finding minimum periscope guard (as well as k-periscope and s-guard) covers is NP-hard for 3-d grids. We present an O(n^2) algorithm for finding minimum periscope guard covers for simple grids and discuss how to extend the algorithm to obtain minimum k periscope guard covers. We show that this algorithm can be applied to obtain minimum periscope guard covers for a class of simple orthogonal polygon in O(n^2).

1. Introduction

The problem of covering a polygon with simpler polygons has gained the interest of many researchers [1, 3, 16]. One such problem is finding the minimum number of star polygons needed to cover a given polygon (a star polygon is such...
that there exists a point in the polygon from which all points in the polygon are visible). Covering a polygon with the minimum number of star polygons (minimum star cover) is equivalent to the placement of the minimum number of point guards (minimum guard cover) so that each point inside the polygon is visible to some guard. This problem was shown to be intractable (NP-hard [6]) for polygons with holes in [15]. The problem was shown to remain intractable for simple polygons in [10, 1]. The complexity of finding minimum guard covers for simple orthogonal polygons remains an open question [21]. Many of the results on guard covers and the related Art Gallery problem can be found in [16].

Because of the intractability of most minimum guard cover problems, restricted classes of polygons and different definitions of visibility have been considered [9, 16, 18, 7, 13, 20]. In the standard definition of visibility, two points are said to be visible if the straight line segment joining them does not intersect the exterior of the polygon. A generalized notion of visibility is the notion of k-visibility [19, 20, 9], in which two points are visible (k-visible) if they can be connected by a path of k or fewer segments without intersecting the exterior of the polygon. An important class of polygons is that of orthogonal polygons. A polygon P is an orthogonal polygon if all its edges are parallel to the major axes. A polygon P is orthogonally convex if it is orthogonal and any horizontal or vertical line (that is not co-linear with an edge) intersects the boundary of P in at most two points. Similarly, P is horizontally (vertically) convex if any horizontal (vertical) line that is not co-linear with an edge intersects P in at most two points. A path inside P is orthogonally convex if it consists of orthogonal segments and any horizontal or vertical line that is not co-linear with a segment intersects the path in at most one point. Two points inside an orthogonal polygon are said to be s-visible [13, 18] if they can be joined by an orthogonally convex staircase path that does not intersect the exterior of the polygon (note that the shortest such path is not unique). Two points are said to be r-visible [7, 13] if they can be placed inside an orthogonal rectangle that is completely contained in the polygon. The notions of s-visibility and r-visibility directly lead to s-star and r-star polygons.

Polynomial time algorithms for solving restricted versions of the guard cover problem in simple orthogonal polygons have been reported in [7, 13]. An optimal O(n) time algorithm for minimally covering a horizontally convex orthogonal polygons with the minimum number of r-star polygons is reported in [7]. In [13], an O(n^3) time algorithm for covering an orthogonal polygon that has only three (out of the possible four) dent orientations with the minimum number of s-star polygons is presented. They also give an O(n^8) time algorithm for the case when the polygon has dents in all four directions.

A structure that is often associated with polygons and has found many applications in Computational Geometry is a grid (e.g., the grid induced by the polygon's edges/vertices; note that the vertices of this grid do not necessarily lie at integer coordinates). If we think of the grid edges as corridors, the star cover problem in a grid is to find the minimum number of guards that need to be
Covering grids

stationed in the grid so that each point in the grid is visible to some guard. Finding a minimum guard cover in a three dimensional grid is NP-hard but a minimum cover for a two dimensional grid can be found in \(O(n^{2.5})\) time [14] (where \(n\) is the number of segments in the grid).

In this paper we address the problem of finding minimum star covers for grids and orthogonal polygons under periscope visibility. Two points are visible under periscope visibility if there is an orthogonal path with at most one bend connecting them without intersecting the exterior of the polygon. Generalizing, \(k\)-periscope visibility allows orthogonal staircase paths with at most \(k\) bends (periscope visibility is the same as 1-bend visibility). The definition of \(k\)-periscope visibility allows orthogonal paths that are not necessarily horizontally, vertically convex. For example, a \(k\)-visibility path may contain a subpath that traces the boundary of the polygon. In the next section we show that finding a minimum periscope guard cover for a three dimensional grid is NP-hard. Also, finding minimum \(k\)-periscope guard covers and \(s\)-guard covers are NP-hard problems. In Section 3, we present an \(O(n^3)\) algorithm for finding the minimum number of periscope guards needed to cover a simple 2-d grid (simple grids are closely related to simple orthogonal polygons). We discuss how to extend this algorithm to include the general case of covering simple 2-d grids by \(k\)-periscope guards. In Section 4, we adapt the grid covering algorithm to develop an \(O(n^3)\) algorithms for finding minimum periscope guard covers for a class of orthogonal polygons called turret-less polygons [22] that includes monotone and orthogonally convex polygons (turret-less polygons are similar to Chung’s polygons discussed in [11]).

2. Periscope guard covers for 3-d grids

The complete two-dimensional grid of size \(n\) is the graph with vertex set \(V = \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}\) and edge set \(E = \{(i, j), (k, m)\} : |i - k| + |j - m| = 1\}. The complete 3-d grid is defined similarly. A (partial) grid is any subgraph of the complete grid. In a geometric setting we think of the grid edges as corridors and the grid vertices as intersections of corridors. We also assume that the grid edges are parallel to the major axes. Although we define a grid so that its vertices are at integer coordinates, this is only for convenience; all the results apply equally well to grids with vertices at arbitrary coordinates.

Finding a minimum set of guards to cover (under normal visibility) a 3-d grid is NP-hard [14]. The reduction is from the vertex cover problem for graphs with maximum degree three [6,16]. We use a similar approach to show that the minimum cover problems for periscope guards, \(k\)-bend guards, and \(s\)-guards are NP-hard for 3-d grids.

Vertex Cover

Instance: Graph \(G = (V, E)\) with all vertices having degree three or less, positive integer \(k < |V|\).
Question: Is there a vertex cover of size $\leq k$ for $G$? (i.e., a set of vertices such that each edge in $G$ is incident on at least one vertex in the set).

Periscope guard cover for 3-d grid

Instance: A three dimensional grid with $n$ segments, integer $r$.

Question: Is there a positioning of $r$ periscope guards in the grid so that every point in the grid is visible to at least one guard?

Given an instance of vertex cover we construct an instance of guard cover as follows. Index the vertices of $G$ arbitrarily from 1 to $|V|$. We construct a three dimensional grid $Q$ such that $Q$ can be covered by less than or equal to $r$ guards ($r$ to be determined later) if and only if there is a vertex cover with at most $k$ vertices in $G$. We start with a full 3-d grid of size $10|V|$ and assign the vertices of $G$ to grid vertices so that the vertex $v_i$ is assigned to the grid vertex $(10i, 10i, 10i)$. This assignment forces each vertex to occupy a distinct plane parallel to each of the major axes. An edge $(v_i, v_j)$ of $G$ is represented by a grid path connecting the corresponding grid points. For our construction, we want the grid paths for the edges of $G$ to be disjoint (this forces any visibility paths that connect points in two distinct grid paths to go through a grid vertex that corresponds to a vertex in $G$). Also, we want each grid path to consist of $4x + 2$ segments (for some integer $x \geq 1$). Then $x$ periscope guards are needed along the grid path. If we place a periscope guard at one of the endpoints of the grid path, the remaining $x - 1$ guards are placed at every 4th bend and the last guard can see the other endpoint of the grid path. To prevent this last guard from seeing any segments in other grid paths, we further require that the three paths corresponding to the three edges incident on a vertex of $G$ need to be orthogonal to each other in the immediate neighborhood of the grid vertex where they meet. Once a path that satisfies these requirements is constructed for each edge, all grid edges and vertices that are not used in these paths are removed. The resulting grid is $Q$.

The generic grid path shown in Fig. 1(a) has ten bends ($x = 2$) and is sufficient to make all the connections (the actual paths we use retain the same structure but may involve rotations and scaling of portions of the generic path). We assign a direction to each edge $(v_i, v_j)$ in $G$ from the vertex with the lower index to the one with the higher index. We use the first three segments of a path for the edges directed out of $v_i$. These three segments take the (at most three) paths out of $v_i$ to the three grid points $(10i + m, 10i + m, 10i + m)$, $m = 1, 2, 3$ ($m$ is distinct for each path). We make these paths disjoint by making them orthogonal to each other at $v_i$, e.g., if all three exist, the path for the first edge moves in the $+x$ direction first, then in the $+y$ and finally in the $+z$ direction while the path for the second edge moves in the $+y$, $+x$, $+z$ order and the third path moves in the $+z$, $+x$, $+y$ order. We use the last three segments of a path to connect $(10j - m', 10j - m', 10j - m')$, $m' = 1, 2, 3$ to the point $(10j, 10j, 10j)$; again the (at most three) paths coming into the grid point $(10j, 10j, 10j)$ are made
Covering grids

Fig. 1. Showing a generic ten-segment path.

orthogonal to each other. We use the mid four segments of the generic path to complete the connection. The length of the fourth segment is always three. The fourth and seventh segments are parallel to each other and in the direction orthogonal to both the third and the eighth segment. The other two segments complete the connection (see Fig. 1(a)).
Since the maximum degree in \( G \) is three, the total number of paths (both coming into and going out of) a grid point that corresponds to a vertex of \( G \) is three. Thus we can always make these paths orthogonal to each other at that grid point (e.g., if there are two incoming and one outgoing edge at \( v_i \), we can bring the incoming paths along the \(-x, -y\) directions and take the outgoing path out along the \(+z\) direction). In constructing paths for the edges of \( G \), we begin by constructing the first three and last three segments of the paths associated with each vertex of \( G \). Then we complete the connections by constructing the mid four segments of each path.

**Theorem 1.** There is a vertex cover of size \( k \) in \( G \) if and only if there is a solution to the corresponding grid cover problem of size \( k + 2|E| \).

**Proof.** Assume there is a vertex cover of size \( k \). Position \( k \) guards at grid vertices that correspond to the vertices in the vertex cover. Then position the remaining guards at every fourth corner in the ten bend paths (Fig. 1(c)) starting from the position of the guards corresponding to the vertex cover (if both ends of a path correspond to vertices in the vertex cover, arbitrarily select one of them; see Fig. 1(b)). Two additional guards are needed in every path.

Conversely, assume that a solution of size \( k + 2|E| \) for the grid cover problem exists. We can obtain a solution to the vertex cover problem from the solution of the grid cover problem as follows. Note that at least two guards are needed in the interior of each path to cover it (in addition to guards at endpoints of paths). If we spread these guards as shown in Fig. 1(b), the remaining \( k \) guards can be shifted to grid vertices corresponding to vertices of \( G \) without disturbing the cover. These vertices must constitute a vertex cover in \( G \). If not, there must be an edge in \( G \) such that the corresponding grid path is guarded by just two guards which is impossible since the path consists of ten segments. 

**Corollary 1.** Finding the minimum number of periscope guards needed to cover a 3-d grid is NP-hard.

Consider now \( k \)-bend guards. We can use the same approach but use grid paths consisting of \( 2(k + 1)x + k + 1(x \geq 1) \) segments for each edge in \( G \). For \( s \)-guards, there is no upper bound on the size of the staircase connecting two points (so that they are visible). However, the staircase is restricted to be orthogonally convex. We can use this requirement to construct grid paths for the edges in \( G \) that will consist of \( 4x + 2 \) orthogonally convex staircases connected so that no sequence of two or more staircases is orthogonally convex. Then the same arguments used in Theorem 1 can be used to show that the minimum cover problem for \( s \)-guards in 3-d grids is NP-hard.

**Corollary 2.** The minimum cover problems for \( k \)-bend guards and \( s \)-guards in 3-d grids are NP-hard.
3. Minimum covers for simple 2-d grids

In this section we consider the periscope guard cover problem for a class of 2-d grids (simple grids) which can be used to model the periscope guard problem in simple orthogonal polygons.

A grid segment is a succession of grid edges along a straight line bounded at either end by a missing edge. A grid is called a simple grid if all the endpoints of its segments lie on the outer face of the planar subdivision formed by the grid (as in Fig. 2(a)); otherwise, the grid is called a general grid (a general grid may have holes as in Fig. 2(b)). Two points on a grid are visible under periscope visibility if they lie on the same segment or the segments on which they lie intersect. Points lying on the same segment are defined to be directly visible while points on intersecting segments are said to be indirectly visible.

The crossing set $C_i$ of a segment $s_i$ is the set of segments that intersect $s_i$. A segment $s_1$ is said to be dominated by another segment $s_2$ if $C_1$ is a proper subset of $C_2$ (note that dominance is irreflexive, i.e., $s_i$ does not dominate itself). In Fig. 2(a), segments $c$ and $d$ are dominated by segment $e$. A segment $s$ is called a cross if there exists a segment $s_1$ such that the crossing set of $s_1$ contains only $s$.

A segment is called a pseudo cross if it becomes a cross by removing zero or more segments dominated by it (note that every cross is also a pseudo cross). A segment is called prime if it is neither a pseudo cross nor is dominated by any other segment. Two segments are equivalent if their crossing sets are the same. In Fig. 2(a), segments 3 and 8 are crosses, segment $e$ is a pseudo cross, segment 7 is prime, and segments 5, 6 are equivalent. Note that a periscope guard that can see a segment $s$ can see all segments equivalent to $s$. Therefore, without loss of generality, we assume that no two segments in the grid are equivalent (we can
The importance of domination is illustrated in Fig. 2(a). A guard located on a dominated segment can be moved to the dominating segment and still see all segments visible from its original position (as well as some additional ones). For example, a guard $g$ placed at point $x$ (Fig. 2(a)) can see segments 1, 2, 3, 4, 5, 6, 7, $b$, $e$, $f$. If $g$ is moved to point $y$ then $g$ can still see all segments visible from $x$ plus segment $a$. This indicates that certain segments are more important than others. We capture this idea by defining a reduced grid.

Let $G$ be a simple grid. Mark all segments that are dominated in $G$. The grid obtained by removing all marked segments is called the reduced grid $G_r$. Fig. 3 shows a grid (the lighter segments are the dominated segments) and the reduced grid obtained by removing them. Since all segments that are connected by a dominated segment in $G$ are also connected by the dominating segment, we have the following.

**Remark.** The reduced grid of any simple and connected grid is simple and connected.

The reduced grid is itself a grid and the definitions of crossing sets, dominated segments, crosses, pseudo crosses, prime, and equivalence are applicable to it. To avoid confusion, we will identify the grid to which these terms refer to.

At this point it is interesting to note that a straightforward greedy strategy for placing guards does not give minimum solution. We can construct an instance of $G$ where repeatedly (i) selecting as a guard an intersection point from which the count of periscope visible segments is maximum and (ii) deleting all segments covered by this guard, does not necessarily yield minimum guard cover. In Fig. 4, the point from which the maximum number of segments are periscope visible is $x$. 

\begin{figure}
\centering
\includegraphics[width=\textwidth]{fig3}
\caption{Fig. 3. Illustrating the construction of the reduced grid.}
\end{figure}
If we place a guard at $x$ and remove segments visible to it then four more guards and hence a total of five guards are needed to cover the whole grid. But, only four guards are enough to cover the grid by placing them at points $a$, $b$, $c$, and $d$.

Our approach for finding a minimum guard cover for a simple grid is to identify places where any minimum solution should have a guard, place a guard, remove a portion of the grid and repeat until all of the grid is visible to guards. Let $R$, $R'$ be two guard covers for a simple grid. We say that a guard $g_i$ in $R$ is equivalent to a guard $g_j$ in $R'$ if the two guards see exactly the same set of grid segments. A guard $g_i$ covers a guard $g_j$ if the set of segments visible to $g_i$ contains the set of segments visible to $g_j$. To obtain a minimum guard cover we locate each guard so that any minimum guard cover will contain a guard equivalent to it or covered by it.

Lemma 1. There exists a minimum periscope guard cover in which all guards are either at the intersection of two segments or (in the case where a segment does not intersect any other segment) at end points of the segment.

Proof. We can adjust any guard cover to obtain one that has an equal (or smaller) number of guards and satisfies the conditions. If a guard $g_i$ is located in the interior of a grid edge, a guard $g_j$ located at either endpoint of that edge covers $g_i$ and we can replace $g_i$ with $g_j$ in the guard cover without increasing its size. □

Lemma 2. There is a minimum periscope guard cover for $G$ in which every pseudo cross in $G$ has a guard along it.
Proof. From the definition of a pseudo cross we have that there exists a segment \( l \) that intersects only the pseudo cross or the pseudo cross and segments dominated by it in \( G \). In order to cover \( l \), a guard must be positioned either along \( l \) or along one of the segments that intersect \( l \). In the first case we move the guard to the intersection of the segment with the pseudo cross. In the second case the guard is on a dominated segment in \( G \). If we move the guard to the pseudo cross (dominating segment), the guard still sees all segments seen before (and some additional ones). In either case we maintain a cover without increasing the number of guards.

Let \( C = s_1s_2, \ldots, s_k \) be the crossing set of a segment \( s \) in the reduced grid \( G_r \). The crossing set \( C \) is said to form a group if there exists a segment \( s' \in C \) such that \( s' \) is dominated (in the reduced grid) by all segments in \( C \) that are not equivalent to \( s' \). Then \( s' \) is called a junior segment in \( C \). In Fig. 3(b), the crossing set for segment \( c \) forms a group and segment \( 2 \) is the junior segment in this group. Another group is formed by the crossing set for segment \( 1 \) with segment \( a \) or segment \( d \) as the junior segment (junior segments are not unique).

Lemma 3. Let \( s_l \) (respectively \( s_r \)) be the left most (respectively right most) segment in the crossing set of a horizontal segment \( s \) in the reduced grid. Let \( s' \) be a segment with smallest crossing set in the crossing set of \( s \). Let \( s_t \) (\( s_b \)) be the top most (bottom most) segment in the crossing set of \( s' \). Then the crossing set of \( s \) forms a group with junior segment \( s' \) if and only if segments \( s_l, s_r, s_t, s_b \) intersect to form a rectangle (as in Fig. 5). A similar property holds for any vertical segment. Note that the rectangle may be degenerate, i.e., it may be a segment or a point.

![Fig. 5. Illustrating Lemma 3.](image-url)
Proof. A junior segment in a group in G, must have the smallest crossing set in that group. Since it is dominated (in $G_r$) by all other segments in the crossing set of s, we must have that $s_1, s_2$ (and all other segments intersecting $s'$) must intersect all segments in the crossing set of $s$; thus $s_1, s_2, s_3, s_4$ intersect to form a rectangle. The rectangle degenerates to a segment when $s_r, s_b$ are the same and to a point when the crossing set of $s$ contains only $s'$. □

Lemma 4. Let s be a prime segment in G. Then a crossing set of s in $G_r$ can not form a group.

Proof. Assume to the contrary that the crossing set of s in $G_r$ forms a group. Let the junior segment in the crossing set be $s'$. If $s'$ intersects only s in $G_r$ then s cannot be prime (it will be a pseudo cross), a contradiction. If $s'$ intersects s and some other segment $s_1$ (in $G_r$) then the crossing set of $s_1$ must contain the crossing set of s (otherwise $s'$ will not be the junior segment). But then $s_1$ will dominate s implying that s is not a prime segment, a contradiction. □

Lemma 5. Let $G_r$ be the reduced grid of a simple grid G. Let s be a pseudo cross segment in G such that its crossing set $C = s_1, s_2, \ldots, s_k$ forms a group in $G_r$. Let $s_i \in C$ be a junior segment in the group. Then there exists a minimum guard cover for G that contains a guard equivalent to a guard placed at the intersection of s and $s_i$.

Proof. Let r be the minimum number of guards needed to cover the grid and let Q be a placement of r guards that covers the grid. We prove the lemma by showing that Q can be rearranged so that a guard is placed at the intersection of s and $s_i$ and the resulting guard set still covers the grid. Note that $s_i$ must be a pseudo cross in the original grid. Now consider the crossing set of $s_i$.

Case 1: The crossing set of $s_i$ contains only s.

Since $s_i$ is a pseudo cross in G, Lemma 2 assures that any minimum cover has a guard along $s_i$. We can move this guard to the intersection of s and $s_i$ (see Fig. 6(a)) without affecting the coverage.

Case 2: The crossing set of $s_i$ contains more than one segment.

Since both s and $s_i$ are pseudo crosses, Lemma 2 implies that Q has at least one guard $g_1$ along $s_i$ and one guard $g_2$ along s. Now observe that $g_1$ and $g_2$ must be at opposite corners of the rectangle formed by the segments $s, s_i, s'$ and $s''$ (Fig. 6(b)). If such a rectangle does not exist then $s_i$ can not be a junior segment (contradicting Lemma 3). We can move $g_1$ and $g_2$ to the other two corners of the rectangle without affecting the set of segments guarded. □

From $G_r$ we can construct two planar graphs, the horizontal segment graph $T_h = (V_h, E_h)$ and the vertical segment graph $T_v = (V_v, E_v)$. $T_h$ is constructed as follows (the construction for $T_v$ is similar): If two or more horizontal grid
segments are equivalent (in \( G_r \)) then we treat them as a single horizontal segment. The set of vertices \( V_h \) is \( V_h = \{ v \mid v \text{ is a horizontal segment in } G_r \} \). Two vertices \( v_1, v_2 \) are connected by an edge \((v_1, v_2)\) if the corresponding horizontal segments are neighbors, i.e., there is a vertical segment that intersects both of them without intersecting any other horizontal segments between them. Clearly \( T_h, T_u \) are planar and connected graphs. If \( T_h \) (similarly, \( T_u \)) contains a cycle, it also contains a cordless cycle. Then the region bounded by the grid edges corresponding to the edges of this chordless cycle must contain one end point of each horizontal segment corresponding to vertices of the cycle, which implies that the grid is not simple, a contradiction. Thus \( T_h, T_u \) can not contain any cycles, i.e., they are trees.

We use \( T_h \) (or \( T_u \)) to find a segment whose crossing set forms a group. We construct a modified tree \( T_{hm} \) from \( T_h \) as follows: A branching node in \( T_h \) is a node that has degree three or higher. A leaf node \( s_l \) of \( T_h \) may directly adjoin its nearest branching node \( b \) or it may be connected to \( b \) through a path of length greater than one. If the later is the case, we identify the largest subpath \( \{s_1, s_2, \ldots, s_l\} \) such that \( s_1 \) dominates \( s_2, s_2 \) dominates \( s_3, \ldots, s_{l-1} \) dominates \( s_l \), and the parent \( s_0 \) of \( s_l \) does not dominate \( s_1 \) and we make all the vertices in this subpath directly adjoin \( s_0 \). We obtain \( T_{hm} \) by repeating this process to all leaves in \( T_h \). Fig. 7(b)) shows \( T_h \) for the grid of Fig. 7(a), and Fig. 7(c) shows the modified tree \( T_{hm} \).

**Lemma 6.** The reduced grid \( G_r \) of any simple grid \( G \) contains a segment whose crossing set forms a group.
Proof. Every leaf segment of $T_h$ (or $T_v$) is a pseudo cross in the original grid $G$. Consider a leaf segment $l$ in $T_{hm}$. If $l$ is neither dominated nor equivalent to any other segment in $G_r$ then there must be a segment $s$ such that $s$ intersects with $l$ but not with the parent of $l$. This means that the crossing set of $l$ forms a group with $s$ as the junior segment. If $l$ is not dominated but is equivalent to another segment in $G_r$, again there is a segment $s$ that intersects $l$ and segments equivalent to it only. Then $l$ forms a group with $s$ as the junior segment.

Now consider the case when all leaf segments in $T_{hm}$ are dominated. Fig. 8 shows an example of such a grid. The parent segment $f$ of a leaf segment in $T_{hm}$ is called a peripheral segment if $f$ becomes a leaf when all leaves of $T_{hm}$ are removed. In Fig. 8, segments 1, 13, and 20 are peripheral segments, whereas segments 16...
and 17 are not. Consider the set of grid points $D$ that are on the outer boundary of the grid $G$, and are the intersections of a peripheral segment $f$ and its children in $T_{km}$ with their crossing sets. If we trace the points in $D$ along the boundary, we obtain two Manhattan Sky Line (MSL) structures (shown by the gray boundary in Fig. 8; if such a trace of $D$ does not form a MSL then some child segment of $f$ will not be dominated by $f$, a contradiction). The MSL above $f$ is the top MSL and the other below $f$ is the bottom MSL. Consider the leaf segment $t$ corresponding to the left most (right most) peak in either MSL. In general, there will be two left (right) peaks, one in the top MSL and another in the bottom MSL. The left most
(rightmost) peak in either MSL is the one that extends further to the left (right); 
in case of ties, the one that starts further to the left (right). In Fig. 8, the leaf 
segment corresponding to the left most peak is segment 7 and the one 
corresponding to the right most peak is segment 11. Let $C$ be the crossing set of $t$. 
Then the crossing set $C$ of $t$ must form a group (otherwise $t$ will not correspond to 
the left most (right most) peak, a contradiction; note that segment 3 in Fig. 8 
does not form a group). □

We are now ready to describe the algorithm for finding a minimum periscope 
guard cover for a simple 2-d grid. The dominated segments in the given grid $G$ 
are marked and a reduced grid $G_r$ is obtained from $G$ by removing dominated 
segments. Lemma 6 guarantees that at least one segment of $G_r$ is such that its 
crossing set forms a group. Once such a segment is found, the location of a guard 
$g$ that corresponds to an optimum solution is determined by using Lemma 5 and 
visible segments in $G$ are marked. We could remove from $G$ the segments that 
are visible to the current guard set and repeat the process on the resulting smaller 
grid until the grid becomes empty. However a problem arises when we remove a 
visible segment that may be used by a guard (positioned later) to see some other 
segments indirectly. For example, when segments visible to $g$ in Fig. 9(a) are 
removed, the grid shown in Fig. 9(b) results. Two additional guards are needed 
for a total of three whereas two guards are enough to cover the original grid. The 
problem arises due to the removal of segments that are indirectly visible to the 
current guard set but may have a guard (positioned later) along them. There are 
cases when it is safe to remove indirectly visible segments. The following 
observation lists the kinds of visible segments that can be safely removed.

**Observation 1.** It is safe to remove a segment $s$ from the grid when:
(a) $s$ is directly visible (to a guard in the current guard set),
(b) $s$ is indirectly visible to a guard placed on segment $s'$ and $s$ is dominated by $s'$,
(c) $s$ and all segments in the crossing set of $s$ are visible to a guard in the current guard set,
(d) $s$ is visible to a guard in the current guard set and intersects with only one segment.

The algorithm therefore removes from $G$ those segments that are visible to a guard in the current guard set and are safe to remove (making use of the above observation). Note that the resulting grid $G''$ may contain some segments that may be indirectly visible. When a grid has some segments marked visible then the definition of domination has to be extended accordingly. We say that segment $s_1$ dominates segment $s_2$ if the set of invisible segments in $C_2$ is a subset of the set of invisible segments in $C_1$. In Fig. 10, segment $s_1$ is dominated by segment $s_2$. Construct a reduced grid $G'$ from $G$' by removing dominated segments. Now observe that cross (or pseudo cross) segments are of two kinds as illustrated by segments $s_5$ and $s_6$ in Fig. 11 (the gray segments are those that are visible to the current guard set). Note that segment $s_6$ is a cross but segment $s_4$ is not. Both kinds of crosses always need a guard along them. From this it follows that Lemmas 5, 6 hold even when the reduced grid contains some visible segments. The algorithm therefore finds a segment that forms a group in the reduced grid of $G''$ and places the next guard using Lemma 5. This process is repeated until the
remaining grid is fully visible. A formal description of the algorithm is given below.

Algorithm GRID-COVER

*Step 1:* Find all dominated segments in the simple grid \( G \) and remove them. Let \( G_r \) be the reduced grid.

*Step 2:* Find a segment \( s \) in \( G_r \) such that its crossing set forms a group. Let \( s' \) denote the junior segment in the crossing set of \( s \).

*Step 3:* Place a guard at the intersection of \( s \) and \( s' \) and remove all segments that are safe to remove (Observation 1) from the reduced grid.

*Step 4:* Repeat step 1 through step 3 until all segments of the grid are visible to a guard.

The execution of the above algorithm is illustrated by an example shown in Fig. 12. Consider the simple 2-d grid shown in Fig. 12(a). All dominated segments are
marked by \( V \). The crossing set of segment \( a \) forms a group with segment 2 as a junior segment. A guard is placed at the intersection of \( a \) and 2. The segments that are visible to \( g_1 \) are drawn light. Visible segments \( a, b, 1, \) and 2 are safe to remove. In the grid obtained by removing these segments the algorithm finds that it is safe to remove visible segments 3, 4, 5, 7 and 8 (each of them intersects with only one invisible segment). The location of the next four guards is found
similarly (Fig. 12(c)). Note that segment $k$ is not safe to remove (it connects invisible segments 14, 15, and 16). When the guard $g_8$ is placed, all the grid segments are visible and the algorithm terminates. Fig. 12(g) shows the final guard placement.

**Theorem 2.** A minimum periscope guard cover for a simple grid can be found in $O(n^3)$ time.

**Proof.** We can construct the grid graph $G$ formed by the intersection of grid segments in $O(n^3)$ time using the arrangement of lines algorithm given in [2] or [5]. This graph is represented by using a quad-edge data structure (as described in [8]) to facilitate traversal of the faces of the graph. We can find the crossing set of one segment in $O(n)$ time by simply traversing the graph along the segment. Then the crossing sets of all segments can be determined in time $O(n^2)$. A segment $s$ dominates a segment $s'$ if the crossing set of $s$ contains the crossing set of $s'$. Note that if a segment is dominated then it is dominated by one of its neighbors. Therefore dominated segments can be identified by comparing the crossing sets of neighboring pairs in $O(n^2)$ time. We can form the reduced graph $G_r$ from $G$ by removing the dominated segments in $O(n^2)$ time by traversing the graph on faces containing the segments, deleting the dominated segments and updating the quad-edge data structure. Whether or not the crossing set of a segment $s$ in $G_r$ forms a group can be determined by using Lemma 3 in $O(n)$ time. Therefore we can find in $O(n^3)$ time a segment whose crossing set form a
group (by applying Lemma 3 at most $O(n)$ times). Visible segments that are safe to remove can be removed by traversing the graph in $O(n^2)$ time. Since the minimum guard cover has size at most $O(n)$, all of the above steps will be repeated at most $O(n)$ times and the overall time complexity of algorithm GRID COVER is $O(n^3)$. □

The algorithm GRID-COVER can be extended in a straightforward way to minimally cover a 2-d grid by $k$-periscope guard as follows. By removing dominated segments from the reduced grid (1-reduced grid) a 2-reduced grid is obtained. Repeating this process $k$ times, a $k$-reduced grid is identified. Next, a guard is placed along a grid segment $s$ such that the crossing set of $s$ forms a group in the $k$-reduced grid. Segments that are visible to the guard and safe to remove are removed from the grid to obtain a smaller grid, in the same way as in GRID-COVER. This process is repeated until the whole grid is covered by guards.

The algorithms we have described work on simple grids. It is interesting to consider what happens in general 2-d grids. It is easy to construct 2-d grids in which no segment forms a group (e.g., consider the grid in Fig. 13). If all the segments in the reduced grid are pseudo crosses, we can find a minimum guard cover using the matching approach in [14] (since each pseudo-cross must have a guard along it). However, the reduced grid may also contain prime segments and it is not clear that the approach we use for simple grids can be extended to general grids.

4. Covering orthogonal polygons

In this section we consider the problem of finding the minimum number of periscope guards needed to cover an orthogonal polygon. We show that the polygon cover problem can be converted into an appropriate grid cover problem for a class of orthogonal polygons.
Consider the subdivision formed by extending the edges of an orthogonal polygon into its interior. The polygon now consists of rows and columns of rectangles. Without loss of generality, we assume that no two polygon edges that face in opposite directions are co-linear (if such a pair exists, we can shift one of the edges slightly so that we create a new row/column of rectangles where a degenerate row/column was before). Each row/column consists of a number of sequences of rectangles separated by portions of the exterior of the polygon. We construct a grid $G$ to represent an orthogonal polygon $P$ by associating a grid segment with each sequence of rectangles. Then the internal grid vertices represent individual rectangles in the polygon. Fig. 14(a) shows an orthogonal polygon and Fig. 14(b) shows the grid corresponding to it.

**Lemma 8.** The grid $G$ is a simple and connected grid.

**Lemma 9.** Let $X$ be a guard cover for the grid $G$. Then $X$ is a guard cover for the underlying polygon $P$.

**Proof.** Suppose that there is a point in $P$ that is not visible to any guard in $X$. Consider the rectangle that contains this point. There must be two grid segments that go through this rectangle and both of them must be visible to a guard in $X$. Then any guard that sees one of these edges in $G$ sees all points in the rectangle in $P$. □

The difficulty with the grid $G$ is that the reverse of Lemma 9 is not true, i.e., a guard cover for the polygon does not always correspond to a guard cover in the grid. For example, consider the polygon of Fig. 15(a) and the corresponding grid
Two guards are needed to cover the grid but one guard is sufficient to cover the polygon. The problem here results from the two rectangles at the top of the polygon. Note that the whole area in these rectangles is indirectly visible to a guard located at point $x$ but a distinct guard for each rectangle is needed in the grid (the two horizontal segments in the top rectangles are not visible to any single guard). Suppose that we place a grid guard at point $x$ in both the polygon and the grid. Then the grid segment corresponding to the left side of the polygon is indirectly visible to the guard. Consider then the infinite sequence of vertical segments obtained by sliding segment $a$ along the boundary of $P$ so that it sweeps through the top rectangle. All these segments are indirectly visible from point $x$; also, all the points in the horizontal segment $b$ (that is not visible to the grid
guard at \( x \) are included in this sweep. That is, the segment \( b \) need not be included in the grid because any guard that sees the left edge will automatically cover \( b \) as well. In effect, the horizontal segment \( b \) can be replaced by its intersection with the sweeping segment without affecting Lemma 9.

We say that a grid segment is a *swept segment* if there is a grid segment that intersects it and can be moved through the entire span of the swept segment without intersecting the exterior of the polygon and while both its ends remain in contact with the boundary of the polygon. This definition can be applied recursively by removing swept rectangles from \( P \) each time. All but three horizontal segments in Fig. 15(d) are swept (recursively). Replacing corresponding segments in the grid with their intersections with the sweeping segment results in the grids of Fig. 15(c), (e). Note that a single guard can cover the grid of Fig. 15(c), while two guards are needed for the grid of Fig. 15(e). We call the grid resulting from sweeping the *swept grid* \( G \), for the orthogonal polygon.

There is one more problem with both grids \( G \) and \( G_s \). It arises when the polygon contains corners like the one shown in Fig. 16. There are two orthogonal grid segments that enter the corner and a guard placed on either one of them can see the whole corner. The problem is that it is not clear locally which of the two choices is the best. We refer to this type of corner as a *turret corner* and a polygon containing such a corner is called a *turret polygon* [22]. Turret polygons are similar to Chung’s polygons discussed in [11]. Note that choosing to place a guard along horizontal grid line \( h \) (respectively, vertical grid line \( v \)) in Fig. 16 is equivalent to adding a vertical (horizontal) notch into the corner so that the notch is not visible from the vertical (horizontal) edge that enters the corner. Also, note that addition of such a notch eliminates one of the choices, i.e., the grid cover now corresponds to a polygon cover.

**Lemma 10.** A grid cover in the grid \( G \) obtained from a simple polygon without

![Fig. 16. A turret corner in a polygon.](image)
turret corners after replacing swept segments with points is equivalent to a polygon
guard cover.

**Proof.** A guard cover for the grid is clearly a guard cover for the polygon. Given
a guard cover for the polygon, we first obtain an equivalent cover in which all
guards are in the interior of some rectangle. To do this, we need to shift guards
that lie at the border between two or more rectangles (i.e., they are co-linear with
some polygon edge) without affecting their visibility. If a guard is at the border
between two rectangles, we note that all the polygon edges that are co-linear with
this border are facing in the same direction. Then moving the guard to the
interior of the rectangle that lies in that same direction does not affect the
visibility of the guard. Guards that lie at the (point) border between three or four
rectangles are handled similarly.

From the polygon guard cover with all guards in the interior of rectangles, we
obtain an equivalent grid cover by shifting each guard to the nearest grid vertex
(i.e., the intersection of the grid segments that intersect the rectangle containing
the guard). Suppose that the resulting grid guard set does not cover the grid.
Then there is a segment in $G$ that is not visible to any of the guards. That means
that none of the rectangles intersected by this segment and none of the rectangles
intersected by segments in its crossing set contained a guard in the original
polygon. But then the set of polygon guards did not cover the polygon, a
contradiction. □

From Lemma 10 it is clear that the minimum periscope guard cover for the
swept grid $G_s$ of a turretless polygon $P$ is also the minimal periscope guard cover
for $P$ as well. Hence we can use algorithm GRID-COVER to the swept grid and
obtain solution for polygon $P$.

**Theorem 3.** A minimum periscope guard cover for a simple orthogonal polygon
with a fixed number of turret corners can be constructed in $O(n^3)$.

**Proof.** Each corner represents two choices. Making the choice is equivalent to
removing the corner. As long as the number of corners is fixed, there are $O(1)$
possible choices to consider. For each choice we can apply the algorithm
GRID-COVER of the previous section to obtain a minimum grid cover. Then we
select the best of these grid covers and that will be an optimum polygon
cover. □

**Corollary 2.** We can find minimum polygon covers in $O(n^3)$ for simple
orthogonally convex polygons, orthogonal monotone polygons, and orthogonal
spiral polygons.
5. Concluding Remarks

We presented $O(n^3)$ algorithms for finding optimum periscope guard covers for simple grids and simple orthogonal polygons with a constant number of turret corners. We showed that finding minimum periscope guard covers for general 3-d grids is NP-Hard. We conjecture that the perfect graph approach [13] is applicable for computing minimum periscope guard covers for simple orthogonal polygons (but this approach yields an algorithm of high complexity). It would be interesting to extend our algorithm (for simple 2-d grids) to include general grids. Similarly, it would be interesting to extend our algorithm (for simple orthogonal polygons with constant number of turret corners) to include all simple orthogonal polygons. Our motivation for considering periscope guards is to help determine the complexity of the guard cover problem for orthogonal polygons which remains a well known open problem.

Acknowledgement

We thank the referees for their helpful suggestions.

References


[22] G. Toussaint, Personal communication.