Guarding curvilinear art galleries with vertex or point guards

Menelaos I. Karavelas\textsuperscript{a,b}, Csaba D. Tóth\textsuperscript{c}, Elias P. Tsigaridas\textsuperscript{d}

\textsuperscript{a}University of Crete, Department of Applied Mathematics, GR-714 09 Heraklion, Greece
\textsuperscript{b}Foundation for Research and Technology - Hellas, Institute of Applied and Computational Mathematics, P.O. Box 1385, GR-711 10 Heraklion, Greece
\textsuperscript{c}University of Calgary, Department of Mathematics and Statistics, 2500 University Drive NW, Calgary, AB, Canada T2N 1N4
\textsuperscript{d}INRIA-LORIA Lorraine, 615 rue du Jardin Botanique, BP 101, 54602 Villers-dès-Nancy cedex, France

Abstract

We study a variant of the classical art gallery problem, where an art gallery is modeled by a polygon with curvilinear sides. We focus on piecewise-convex and piecewise-concave polygons, which are polygons whose sides are convex and concave arcs, respectively. It is shown that for monitoring a piecewise-convex polygon with \( n \geq 2 \) vertices, \( \lfloor \frac{2n}{3} \rfloor \) vertex guards are always sufficient and sometimes necessary. We also present an algorithm for computing at most \( \lfloor \frac{2n}{3} \rfloor \) vertex guards in \( O(n \log n) \) time and \( O(n) \) space. For the number of point guards that can be stationed at any point in the polygon, our upper bound \( \lfloor \frac{2n}{3} \rfloor \) carries over and we prove a lower bound of \( \lceil \frac{n}{2} \rceil \). For monitoring a piecewise-concave polygon with \( n \geq 3 \) vertices, \( 2n - 4 \) point guards are always sufficient and sometimes necessary, whereas there are piecewise-concave polygons where some points in the interior are hidden from all vertices, hence they cannot be monitored by vertex guards. We conclude with bounds for some special types of curvilinear polygons.

Key words: art gallery, curvilinear polygons, vertex guards, point guards, piecewise-convex polygons, piecewise-concave polygons

2000 MSC: 68U05, 68W40

1. Introduction

In the classical art gallery problem, an art gallery is represented by a simply connected closed polygonal domain (for short polygon) \( P \). The art gallery is monitored by a set of guards, each represented by a point in \( P \), if every point in \( P \) is visible to at least one of the guards. Two points see each other if they are visible to each other, i.e., if the closed line segment connecting them lies in \( P \). Victor Klee asked what is the minimum number of guards that can monitor any polygon with \( n \geq 3 \) vertices. Art gallery-type problems have found applications in robotics [1, 2], motion planning [3, 4], computer vision and pattern recognition [5, 6, 7, 8], graphics [9, 10], CAD/CAM [11, 12] and wireless networks [13]. Curvilinear objects were typically modeled with straight-line polygonal approximations. Starting from the late 80s, some geometric algorithms were extended to curvilinear polygons [14]. Refer to the recent book edited by Boissonnat and Teillaud [15] for a collection of computational-geometry results for curves and surfaces. In this context this paper addresses the classical art gallery problem for various classes of polygonal regions bounded by curvilinear edges. To the best of our knowledge this is the first time that the art gallery problem is considered in this context.

The first results on art gallery-type problems date back to the 1970’s. Chvátal [16] proved that every simple polygon with \( n \) vertices can be monitored by \( \lfloor \frac{n}{4} \rfloor \) vertex guards; this bound is tight in the worst case. Later Fisk [17]...
gave an elegant algorithmic proof using a 3-coloring of a triangulation of the polygon. Fisk’s algorithm runs in $O(n)$ time for a triangulated polygon with $n$ vertices, and the time complexity of the triangulation is $O(n)$ based on Chazelle’s algorithm [18]. Lee and Lin [19] showed that finding the minimum number of vertex guards for a given simple polygon is NP-hard, which was extended to point guards by Aggarwal [20]. Other types of art galleries have also been considered. Kahn, Klawe and Kleitman [21] showed that finding the minimum number of vertex guards for a given simple orthogonal polygon, i.e., simple polygon with axes-aligned edges, with $n$ vertices can be monitored by $\lfloor \frac{n}{4} \rfloor$ vertex guards, and this bound is best possible. Several $O(n)$ time algorithms have been proposed for placing the guards in this variation of the problem, notably by Sack [22] and later by Lubiw [23]. Edelsbrunner, O’Rourke and Welzl [24] gave an $O(n)$ time algorithm for placing $\lfloor \frac{n}{4} \rfloor$ point guards that jointly monitor an orthogonal polygon with $n$ vertices. Other types of guarding problems have also been studied in the literature. For a detailed discussion of these variations and the corresponding results the interested reader should refer to the book by O’Rourke [25], or the survey papers by Shermer [26] and by Urrutia [27].

The main focus of this paper is the class of polygons that are either locally convex or locally concave (except possibly at the vertices), the edges of which are convex arcs (defined below); we call such polygons piecewise-convex and piecewise-concave polygons, respectively.

We show that every piecewise-convex polygon with $n \geq 2$ vertices can be monitored by at most $\lfloor \frac{2n}{3} \rfloor$ vertex guards. This bound is tight: there are piecewise-convex polygons with $n$ vertices, for every $n \geq 2$, that cannot be monitored by fewer than $\lfloor \frac{2n}{3} \rfloor$ vertex guards. Our upper bound is based on an algorithm for placing vertex guards, which can be implemented in $O(n \log n)$ time and $O(n)$ space. Our algorithm is a generalization of Fisk’s algorithm [17]; in fact, when applied to a straight-line polygon with $n \geq 3$ vertices, it produces at most $\lfloor \frac{n}{3} \rfloor$ vertex guards. For the purposes of our complexity analysis and results, we assume, throughout the paper, that the curvilinear edges of our polygons are arcs of algebraic curves of constant degree. As a result, all predicates required by the algorithms described in this paper take $O(1)$ time in the real RAM model of computation model. The central idea for our upper bound is the approximation of a piecewise-convex polygon by a straight-line polygon by adding Steiner vertices on the boundary of the curvilinear polygon. The resulting polygonal approximation is a simple straight-line polygon. We compute a guard set for the polygonal approximation by a slightly modified version of Fisk’s algorithm [17]. This guard set monitors the original curvilinear polygon, however, vertex guards may be located at Steiner vertices. The final step of our algorithm maps the vertex guards of the polygonal approximation to vertex guards of the curvilinear polygon. Our upper bound of $\lfloor \frac{2n}{3} \rfloor$ also applies to point guards. However, it does not match the best lower bound we have found.

There are piecewise-convex polygons with $n$ vertices, for every $n \geq 2$, that cannot be monitored by fewer than $\lceil \frac{n}{4} \rceil$ point guards.

Some piecewise-concave polygons have interior points hidden from all vertices (see Fig. 14(a)), and hence they cannot be monitored by vertex guards alone. We thus turn our attention to point guards, and we show that $2n - 4$ point guards are always sufficient and sometimes necessary for monitoring a piecewise-concave polygon with $n \geq 3$ vertices. Our upper bound proof is based on Fejes Tóth’s technique for illuminating sets of disjoint convex objects in the plane [28]. Given a piecewise-concave polygon $P$, we subdivide $P$ into crescents (bounded by a convex and a concave arc), each adjacent to an edge of $P$, and into convex polygonal holes. Using Fejes Tóth’s argument, if we place guards at points incident to at least three crescents, at two vertices of each triangular hole and all vertices at holes with 4 or more vertices, we obtain a guard set that monitors all holes and all crescents, hence the entire piecewise-concave polygon $P$. Since the intersection graph of the crescents is outerplanar, whose faces correspond to the holes, it is easy to show that the number of point guards is at most $2n - 4$.

The rest of the paper is structured as follows. In Section 2 we define curvilinear polygons, including piecewise-convex and piecewise-concave polygons. In Section 3 we present our algorithm for computing a vertex guard set, of size $\lfloor \frac{2n}{3} \rfloor$, for a piecewise-convex polygon with $n$ vertices, and present families of piecewise-convex polygons that require a minimum of $\lfloor \frac{n}{4} \rfloor$ vertex or $\lceil \frac{n}{3} \rceil$ point guards in order to be monitored. In Section 4 we present our results for piecewise-concave polygons, namely, that $2n - 4$ point guards are always necessary and sometimes sufficient for this class of polygons. The final section of the paper, Section 5, discusses further results and states open problems.
2. Definitions

Types of curvilinear polygons. Let $V$ be a sequence of points $v_1, \ldots, v_n$, $n \geq 2$, and $A$ a set of curvilinear arcs $a_1, \ldots, a_n$, such that the endpoints of $a_i$ are $v_i$ and $v_{i+1}$\(^1\). We assume that the arcs $a_i$ and $a_j$, $i \neq j$, do not intersect, except when $j = i - 1$ or $j = i + 1$, in which case they intersect only at the points $v_i$ and $v_{i+1}$, respectively. We define a curvilinear polygon $P$ to be the closed region delimited by the arcs $a_i$. The points $v_i$ are called the vertices of $P$. An arc $a_i$ is a convex arc if every line on the plane intersects $a_i$ at either at most two points or along a line segment.

A polygon $P$ is a straight-line polygon if its edges are line segments (see Fig. 1(a)). A polygon $P$ is locally convex (see Fig. 1(c)), (resp., locally concave (see Fig. 1(e))), if for every point $p$ on the boundary of $P$, with the possible exception of $P$’s vertices, there exists a disk centered at $p$, say $D_p$, such that $P \cap D_p$ is convex (resp., concave). A polygon $P$ is piecewise-convex (see Fig. 1(b)), (resp., piecewise-concave (see Fig. 1(d))), if it is locally convex (resp., concave), and the portion of the boundary between every two consecutive vertices is a convex arc. Finally, a polygon is said to be a general polygon if we impose no restrictions on the type of its edges (see Fig. 1(f)). We use the term curvilinear polygon to refer to a polygon the edges of which are either line or curve segments.

Guards and guard sets. In our setting, a guard or point guard is a point in the interior or on the boundary of a curvilinear polygon $P$. A guard of $P$ that is also a vertex of $P$ is called a vertex guard. We say that a curvilinear polygon $P$ is monitored by a set $G$ of guards if every point in $P$ is visible from at least one point in $G$, where two points $p$ and $q$ in $P$ are visible from each other if the line segment $pq$ lies entirely in $P$. The set $G$ that has this property is called a guard set for $P$. A guard set that consists solely of vertices of $P$ is called a vertex guard set.

3. Piecewise-convex polygons

In this section we present an algorithm which, given a piecewise-convex polygon $P$ with $n$ vertices, computes a vertex guard set $G$ of size $\lceil \frac{2n}{3} \rceil$. The basic steps of the algorithm are as follows:

\(^1\)Indices are evaluated modulo $n$. 

Figure 1: Different types of curvilinear polygons: (a) a straight-line polygon, (b) a piecewise-convex polygon, (c) a locally convex polygon, (d) a piecewise-concave polygon, (e) a locally concave polygon and (f) a general polygon.
1. Compute the polygonal approximation $\tilde{P}$ of $P$.
2. Compute a constrained triangulation $T(\tilde{P})$ of $\tilde{P}$.
3. Compute a guard set $G_3$ for $\tilde{P}$, by 3-coloring the vertices of $T(\tilde{P})$.

3.1. Polygonalization of a piecewise-convex polygon

Let $a_i$ be a convex arc with endpoints $v_i$ and $v_{i+1}$. We call the convex region $r_i$ delimited by $a_i$ and the line segment $v_i v_{i+1}$ a room. A room is called degenerate if the arc $a_i$ is a line segment. A line segment $pq$, where $p, q \in a_i$ is called a chord, and the region delimited by the chord $pq$ and $a_i$ is called a sector. The chord of a room $r_i$ is defined to be the line segment $v_i v_{i+1}$ connecting the endpoints of the corresponding arc $a_i$. A degenerate sector is a sector with empty interior. We distinguish between two types of rooms (see Fig. 2):

1. a room is empty if it is non-degenerate and does not contain any vertex of $P$ in its interior or in the interior of its chord.
2. a room is non-empty if it is non-degenerate and contains at least one vertex of $P$ in its interior or in the interior of its chord.

In order to polygonalize $P$ we add Steiner vertices in the interior of non-linear convex arcs. More specifically, for each empty room $r_i$ we add a vertex $w_{i+1}$ (anywhere) in the interior of the arc $a_i$ (see Fig. 3). For each non-empty room $r_i$, let $X_i$ be the set of vertices of $P$ that lie in the interior of the chord $v_i v_{i+1}$ of $r_i$, and $R_i$ be the set of vertices of $P$ that are contained in the interior of $r_i$ or belong to $X_i$ (by assumption $R_i \neq \emptyset$). If $R_i = X_i$, let $C_i$ be the set of vertices on the convex hull of the vertex set $(R_i \setminus X_i) \cup \{v_i, v_{i+1}\}$; if $R_i = X_i$, let $C_i = X_i \cup \{v_i, v_{i+1}\}$. Finally, let $C_i = C_i \setminus \{v_i, v_{i+1}\}$.

Clearly, $v_i$ and $v_{i+1}$ belong to the set $C_i$, and, furthermore, $C_i \neq \emptyset$.

Let $m_i$ be the midpoint of $v_i v_{i+1}$ and $l_i^+(p)$ the line perpendicular to $v_i v_{i+1}$ passing through a point $p$. If $C_i^+ \neq X_i$, then, for each $v_k \in C_i^+$, let $w_{ki}$, $1 \leq k \leq |C_i^+|$, be the (unique) intersection of the line $m_i v_k$ with the arc $a_i$; if $C_i^+ = X_i$, then, for each $v_k \in C_i^+$, let $w_{ki}$, $1 \leq k \leq |C_i^+|$, be the (unique) intersection of the line $l_i^+(v_k)$ with the arc $a_i$.

Now consider the sequence $V$ of the original vertices of $P$ augmented by the Steiner vertices added to empty and non-empty rooms; the order of the vertices in $V$ is the order in which we encounter them as we traverse the boundary of $P$ counterclockwise. The straight-line polygon defined by the sequence $V$ of vertices is denoted by $\tilde{P}$ (see Fig. 4(a)). It is easy to show that:

**Lemma 1.** The straight-line polygon $\tilde{P}$ is a simple polygon.

**Proof.** It suffices show that the line segments replacing the curvilinear segments of $P$ do not intersect other edges of $P$ or $\tilde{P}$.

Let $r_i$ be an empty room, and let $w_{i+1}$ be the point added in the interior of $a_i$. The interior of the line segments $v_i w_{i+1}$ and $w_{i+1} v_{i+1}$ lie in the interior of $r_i$. Since $P$ is a piecewise-convex polygon, and $r_i$ is an empty room, no edge of $P$ could
vertices of the polygonal approximation $\tilde{Q}$ then $w$ is a point in the interior of $a_3$. $m_5$ is the midpoint of the line segment $v_5v_6$, whereas $w_{3,1}$ and $w_{5,2}$ are the intersections of the lines $m_5v_2$ and $m_5v_1$ with the arc $a_5$, respectively. In this example $R_3 = \{v_1, v_2, v_3\}$, whereas $C_5^* = \{v_1, v_2\}$.

Figure 3: The Steiner vertices (white points) for rooms $r_5$ (empty) and $r_3$ (non-empty). $w_{3,1}$ is a point in the interior of $a_3$. $m_5$ is the midpoint of the line segment $v_5v_6$, whereas $w_{5,1}$ and $w_{5,2}$ are the intersections of the lines $m_5v_2$ and $m_5v_1$ with the arc $a_5$, respectively. In this example $R_5 = \{v_1, v_2, v_3\}$, whereas $C_5^* = \{v_1, v_2\}$.

potentially intersect $v_iw_{i,1}$ or $w_{i,1}v_{i+1}$. Hence replacing $a_i$ by the polyline $v_iw_{i,1}v_{i+1}$ gives us a new piecewise-convex polygon.

Let $r_i$ be a non-empty room. Let $w_{i,1}, \ldots, w_{i,K_i}$ be the points added on $a_i$, where $K_i$ is the cardinality of $C_i^*$. By construction, every point $w_{i,k}$ is visible from $w_{i,k+1}$, $k = 1, \ldots, K_i - 1$, and every point $w_{i,k}$ is visible from $w_{i,k-1}$, $k = 2, \ldots, K_i$. Moreover, $w_{i,1}$ is visible from $v_i$ and $w_{i,K_i}$ is visible from $v_{i+1}$. Therefore, the interior of the segments in the polyline $v_iw_{i,1}, \ldots, w_{i,K_i}v_{i+1}$ lie in the interior of $r_i$ and do not intersect any arc in $P$. Hence, substituting $a_i$ by the polyline $v_iw_{i,1}, \ldots, w_{i,K_i}v_{i+1}$ gives us a new piecewise-convex polygon.

As a result, the straight-line polygon $\tilde{P}$ is a simple polygon. \hfill \Box

We call the straight-line polygon $\tilde{P}$, defined by $\tilde{V}$, the straight-line polygonal approximation of $P$, or simply the polygonal approximation of $P$. An obvious result for $\tilde{P}$ is the following:

**Corollary 2.** If $P$ is a piecewise-convex polygon the polygonal approximation $\tilde{P}$ of $P$ is a straight-line polygon that is contained in $P$.

We end this subsection by proving a tight upper bound on the size of the polygonal approximation of a piecewise-convex polygon. We start with an intermediate result, namely that the sets $C_i^*$ are pairwise disjoint.

**Lemma 3.** Let $i, j$, with $1 \leq i < j \leq n$. Then $C_i^* \cap C_j^* = \emptyset$.

**Proof.** Consider an arc $a_i$ of $P$, delimited by the vertices $v_i$ and $v_{i+1}$ and let $\pi_i$ denote the shortest path in $P$ between them. Note that $\pi_i$ is a straight-line polygonal path, the internal vertices of which are the vertices of $C_i^*$. Since $a_i$ is a convex arc, $\pi_i$ is also a convex arc. $a_i$ and $\pi_i$ bound a (curvilinear) polygon, that we denote by $Q_i$, for which $\pi_i$ is locally concave. That is, every point in $C_i^*$ is a reflex vertex of $Q_i$, and so every point in $C_i^*$ is a reflex (i.e., locally concave) vertex of $P$ as well. At every vertex $w \in C_i^*$, the bisector of the internal angle of $P$ enters the polygon $Q_i$ and leaves $Q_i$ (and $P$) at some point along $a_i$.

Consider the bisector of the internal angle at every reflex vertex $w$ of $P$. If the bisector intersects some arc $a_j$, then $w$ can belong to the set $C_j^*$ only. Since every bisector intersects at most one arc $a_j$ (we are referring to the first intersection of the bisector while walking on it away from $w$), every vertex $w$ belongs to at most one set $C_j^*$. \hfill \Box

An immediate consequence of Lemma 3 is the following corollary that gives us a tight bound on the number of vertices of the polygonal approximation $\tilde{P}$ of $P$. 

5
Corollary 4. The number of vertices of the polygonal approximation \( \tilde{P} \) of a piecewise-convex polygon \( P \) with \( n \) vertices is at most \( 3n \). This bound is tight (up to an additive constant).

Proof. Let \( a_i \) be a convex arc of \( P \), and let \( r_i \) be the corresponding room. If \( r_i \) is an empty room, then \( \tilde{P} \) contains one Steiner vertex due to \( a_i \). Hence \( \tilde{P} \) contains at most \( n \) Steiner vertices attributed to empty rooms in \( P \). If \( r_i \) is a non-empty room, then \( \tilde{P} \) contains \( |C_i^*| \) Steiner vertices due to \( a_i \). By Lemma 3 the sets \( C_i^*, i = 1, \ldots, n \) are pairwise disjoint, which implies that \( \sum_{i=1}^{n} |C_i^*| \leq |V| = n \). Therefore \( \tilde{P} \) contains the \( n \) vertices of \( P \), contains at most \( n \) vertices in empty rooms of \( P \), and at most \( n \) vertices in non-empty rooms of \( P \). We thus conclude that the size of \( \tilde{V} \) is at most \( 3n \).

The upper bound of the paragraph above is tight up to an additive constant. Consider the piecewise-convex polygon \( P \) of Fig. 5. It consists of \( n - 1 \) empty rooms and one non-empty room \( r_1 \), such that \( |C_1^*| = n-2 \). It is easy to see that \( |\tilde{V}| = 3n - 3 \).

3.2. Triangulating the polygonal approximation

Let \( P \) be a piecewise-convex polygon, \( \tilde{P} \) its polygonal approximation, and \( S_{\tilde{P}} \) the set of Steiner vertices in \( \tilde{P} \). We construct a constrained triangulation of \( \tilde{P} \), i.e., we triangulate \( \tilde{P} \), while imposing some triangles to be part of this.
Lemma 6. The set \( G \) for the three possible cases for \( \Pi \) power set 2 \( T \) guard set for \( P \).

Proof. for \( \tilde{\Pi} \) Lemma 5. Each one of the sets \( K_p, \Pi_p \) and \( M_p \) is a guard set for \( P \).

Proof. Let \( G_p \) be one of \( K_p, \Pi_p \) and \( M_p \). By construction, \( G_p \) monitors all triangles in \( T(\tilde{P}) \). To show that \( G_p \) is a guard set for \( P \), it suffices to show that \( G_p \) also monitors the non-degenerate sectors defined by the edges of \( \tilde{P} \) and the corresponding convex subparts of \( P \).

Indeed, let \( s \) be a non-degenerate sector associated with the convex arc \( a_i \), and let \( T \in T(\tilde{P}) \) be the triangle incident to the chord of \( s \). If \( r_i \) is an empty room, each of the three vertices of \( T \) monitors \( r_i \) (and therefore also \( s \)). If \( r_i \) is a non-empty room, the vertex of \( T \) that is not an endpoint of the chord of \( s \) is a vertex in \( C_i \) and monitors \( s \) by construction. Clearly, one of the three vertices of \( T \) belongs to \( G_p \).}

Let as now assume, without loss of generality, that \( |K_p| \leq |\Pi_p| \leq |M_p| \). Define the mapping \( f \) from \( K_{S_p} \) to the power set \( 2^{\Pi_p} \) of \( \Pi_p \) by mapping a vertex \( x \) in \( K_{S_p} \) to all the neighboring vertices of \( x \) in \( T(\tilde{P}) \) that belong to \( \Pi_p \) (see Fig. 7 for the three possible cases for \( x \)). Notice that \( 1 \leq |f(x)| \leq 2 \).

Finally, define the set \( G_p = K_p \cup f(K_{S_p}) \), where \( f(K_{S_p}) = \bigcup_{x \in K_{S_p}} f(x) \). We claim that \( G_p \) is a guard set for \( P \).

Lemma 6. The set \( G_p = K_p \cup f(K_{S_p}) \) is a guard set for \( P \).
The regions in $P \setminus \tilde{P}$ are sectors bounded by a curvilinear arc, which is a subarc of an edge of $P$, and the corresponding chord connecting the endpoints of this subarc. To show that $G_P$ is a guard set for $P$, it suffices show that every triangle in $\mathcal{T}(\tilde{P})$ and every sector in $P \setminus \tilde{P}$ is monitored by at least one vertex in $G_P$.

If all three vertices of a triangle $T \in \mathcal{T}(\tilde{P})$ are vertices of $P$, one of the vertices of $T$ is in $K_P \subseteq G_P$. If $T$ is a triangle in an empty room (see Fig. 8(left)), or a boundary crescent triangle (see Fig. 8(middle)), either the unique Steiner vertex $z$ of $T$ is in $K_{SP}$, in which case one of the other two vertices of $T$ belongs to $f(K_{SP})$, or $z$ is not in $K_{SP}$, in which case one of the other two vertices of $T$ belongs to $K_P$. Moreover, the sector/sectors adjacent to an edge of $T$ in $r_j$ is/are visible by both vertices of $T$ in $P$ and thus monitored by one of them. Finally, upper and lower crescent triangles come in pairs. Let $T$ be an upper crescent triangle in a non-empty room $r_i$ (see Fig. 8(right)). Let $x, y$ be the vertices of $T$ in $P$, and let $z$ be its vertex in $S_P$; it is assumed here that $z$ is the intersection of $m_{i,j}$ with $a_i$. Let $T'$ be the lower crescent triangle adjacent to $T$ along the edge $xz$, $w$ be the third vertex of $T'$, and $s$ be the sector in $P \setminus \tilde{P}$ adjacent to $zw$. Since $x$ and $y$ belong to $C_i$, either $x$ or $y$ monitors $T$, $T'$ and $s$. We end the proof by claiming that either $x$ or $y$ belongs to $G_P$: if $x$ or $y$ belongs to $K_P$, the claim is obvious; if neither $x$ nor $y$ belongs to $K_P$, then $z \in K_{SP}$, in which case one of $x$ and $y$ belongs to $f(K_{SP})$.

Since $f(K_{SP}) \subseteq \Pi_P$ we get that $G_P \subseteq K_P \cup \Pi_P$. Since $K_P$ and $\Pi_P$ are the two sets of smallest cardinality among $K_P, \Pi_P$ and $M_P$, we conclude that $|G_P| \leq |K_P| + |\Pi_P| \leq \frac{2n}{3}$, and thus arrive at the following theorem.

**Theorem 7.** Let $P$ be a piecewise-convex polygon with $n \geq 2$ vertices. $P$ can be monitored with at most $\lfloor \frac{2n}{3} \rfloor$ vertex guards.

We close this subsection by making two remarks:

**Remark 1.** When the input to our algorithm is a straight-line polygon all rooms are degenerate: consequently, no Steiner vertices are created, and the guard set computed corresponds to the set of colored vertices of smallest cardinality, hence producing a vertex guard set of size at most $\lfloor \frac{n}{2} \rfloor$. In that respect, our algorithm can be viewed as a generalization of Fisk’s algorithm [17] to the class of piecewise-convex polygons.

**Remark 2.** Given a straight-line polygon $P$ with $r \geq 2$ reflex vertices, we can view $P$ as a piecewise-convex polygon the edges of which are $c$ convex polylines, where $c \geq r$. In this context Theorem 7 can be “translated” as follows:

If the boundary of a simple straight-line polygon $P$ can be partitioned into $c \geq 2$ convex polylines such that $P$ is a piecewise-convex polygon with its edges being the $c$ convex polylines, then $P$ can be monitored with at most $\lfloor \frac{2n}{3} \rfloor$ vertex guards.
Figure 7: The three cases in the definition of the mapping \( f \). Case (a): \( x \) is a Steiner vertex in an empty room. Case (b): \( x \) is an Steiner vertex in a non-empty room and is not the last Steiner vertex added on the curvilinear arc. Cases (c) and (d): \( x \) is the last Steiner vertex added on the curvilinear arc of a non-empty room (in (c) \( |f(x)| = 1 \), whereas in (d) \( |f(x)| = 2 \)).

3.4. Time and space complexity

In this subsection we show how to compute the vertex guard set \( G_P \) in \( O(n \log n) \) time and \( O(n) \) space. It is straightforward to show that Steps 2–4 of our algorithm (see beginning of Section 3) can be implemented in linear time and space. To complete our time and space complexity analysis, we need to show how to compute the polygonal approximation \( \tilde{P} \) of \( P \) in \( O(n \log n) \) time and linear space. In order to compute \( \tilde{P} \), it suffices to compute for each room \( r_i \) the set of vertices \( C^*_i \). If \( C^*_i = \emptyset \), then \( r_i \) is empty, otherwise we have the set of vertices we wanted. From \( C^*_i \) we can compute the points \( w_{i,k} \) and the straight-line polygon \( \tilde{P} \) in \( O(n) \) time and space.

The underlying idea is to split \( P \) into \( y \)-monotone piecewise-convex subpolygons. For each room \( r_i \) within each such \( y \)-monotone subpolygon we then compute the corresponding set \( C^*_i \). This is done by first computing a subset \( S_i \) of the set \( R_i \) of the points in the room \( r_i \), such that \( S_i \supseteq C^*_i \), and then applying an optimal time and space convex hull algorithm to the set \( S_i \cup \{v_i, v_{i+1}\} \) in order to compute \( C_i \), and subsequently from that \( C^*_i \). In the discussion that follows, we assume that for each convex arc \( a_i \) of \( P \) we associate a set \( S_i \), which is initialized to be the empty set. The sets \( S_i \) are progressively filled with vertices of \( P \), so that in the end they fulfill the containment property mentioned above.

Splitting \( P \) into \( y \)-monotone piecewise-convex subpolygons is done in two steps:

1. First we split each convex arc \( a_i \) into \( y \)-monotone pieces. Let \( P' \) be the piecewise-convex polygon we get by introducing the \( y \)-extremal points for each \( a_i \) and let \( V' \) be the vertex set of \( P' \). Since each \( a_i \) can yield up to three \( y \)-monotone convex pieces, we conclude that \( |V'| \leq 3n \). Obviously splitting the convex arcs \( a_i \) into \( y \)-monotone pieces takes \( O(n) \) time and space. A vertex added to split a convex arc into \( y \)-monotone pieces are called an added extremal vertex.
2. Second, we apply to $P'$ the standard algorithm for computing $y$-monotone subpolygons of a straight-line polygon (cf. [30] or [31]). The algorithm in [30] (or [31]) is valid not only for line segments, but also for piecewise-convex polygons consisting of $y$-monotone arcs (such as $P'$). Since $|V'| \leq 3n$, we conclude that computing the $y$-monotone subpolygons of $P'$ takes $O(n \log n)$ time and requires $O(n)$ space.

Note that a non-split arc of $P$ belongs to exactly one $y$-monotone subpolygon. $y$-monotone pieces of a split arc of $P$ may belong to at most three $y$-monotone subpolygons (see Fig. 9).

Suppose now that we have a $y$-monotone polygon $Q$. The edges of $Q$ are either convex arcs of $P$, or pieces of convex arcs of $P$, or line segments between mutually visible vertices of $P$, added in order to form the $y$-monotone subpolygons of $P$; we call these line segments bridges (see Fig. 9). For each non-bridge edge $e_i$ of $Q$, we want to compute the set $C_i^*$. This is done by sweeping $Q$ in the negative $y$-direction (i.e., by moving the sweep line from $+\infty$ to $-\infty$). The events of the sweep correspond to the $y$ coordinates of the vertices of $Q$, which are all known before-hand and can be put in a decreasing sorted list. There are four different types of events:

1. the first event: corresponds to the top-most vertex of $Q$,
2. the last event: corresponds to the bottom-most vertex of $Q$,
3. a left event: corresponds to a vertex of the left $y$-monotone chain of $Q$, and
4. a right event: corresponds to a vertex of the right $y$-monotone chain of $Q$.

Our sweep algorithm proceeds as follows. Let $\ell$ be the sweep line parallel to the $x$-axis at some $y$. For each $y$ in between the $y$-maximal and $y$-minimal values of $Q$, $\ell$ intersects $Q$ at two points which belong to either a left edge $e_l$
or a left vertex $v_l$ (i.e., an edge or vertex on the left $y$-monotone chain of $Q$), and either a right edge $e_r$ or a right vertex $v_r$ (i.e., a edge or vertex on the right $y$-monotone chain of $Q$). We associate the current left edge $e_l$ at position $y$ to a point set $S_L$ and the current right edge at position $y$ to a point set $S_R$. If the edge $e_l$ (resp., $e_r$) is a non-bridge edge, the set $S_L$ (resp., $S_R$) contains vertices of $Q$ that are in the room of the convex arc of $P$ corresponding to $e_l$ (resp., $e_r$).

When the $y$-maximal vertex $v_{\text{max}}$ is encountered, i.e., during the first event, we initialize $S_L$ and $S_R$ to be the empty set. When a left event is encountered due a vertex $v_l$, let $e_{\text{left}}$ be the left edge above $v_l$ and $e_{\text{down}}$ be the left edge below $v_l$ and let $e_i$ be the current right edge. If $e_{\text{left}}$ is an non-bridge edge, and $a_i$ is the corresponding convex arc of $P$, we augment the set $S_i$ by the vertices in $S_L$. Then, irrespectively of whether or not $e_{\text{left}}$ is a bridge edge, we re-initialize $S_L$ to be the empty set. Finally, if $e_i$ is a non-bridge edge, and $a_k$ is the corresponding convex arc in $P$, we check if $v_l$ is in the room of $a_k$; if this is the case we add $v_l$ to $S_R$. When a right event is encountered our sweep algorithm behaves symmetrically. When the last event is encountered due to the $y$-minimal vertex $v_{\text{min}}$, let $e_l$ (resp., $e_r$) be the left (resp., right) edge of $Q$ above $v_{\text{min}}$. If $e_l$ (resp., $e_r$) is a non-bridge edge, let $a_i$ (resp., $a_j$) be the corresponding convex arc in $P$. In this case we simply augment $S_j$ (resp., $S_i$) by the vertices in $S_L$ (resp., $S_R$).

We claim that our sweep-line algorithm computes a set $S_i$ such that $S_i \supseteq C_i$. To prove this we need the following intermediate result:

**Lemma 8.** Given a non-empty room $r_i$ of $P$, with $a_i$ the corresponding convex arc, the vertices of the set $C_i$ belong to the $y$-monotone subpolygons of $P'$ computed via the algorithm in [30] (or [31]), which either contain the entire arc $a_i$ or $y$-monotone pieces of $a_i$.

**Proof.** Let $u$ be a vertex of $P$ in $C_i$ that is not a vertex of any of the $y$-monotone subpolygons of $P'$ (computed by the algorithm in [30] or [31]) that contain either the entire arc $a_i$ or $y$-monotone pieces of $a_i$. Let $v_{\text{max}}$ (resp., $v_{\text{min}}$) be the vertex of $P$ of maximum (resp., minimum) $y$-coordinate in $C_i$; ties are broken lexicographically. Let $\ell_u$ be the line parallel to the $x$-axis passing through $u$. Consider the following cases:

1. $u \in C_i \setminus \{v_{\text{min}}, v_{\text{max}}\}$. Without loss of generality we can assume that $u$ is a vertex in the right $y$-monotone chain of $C_i$ (see Figs. 10(a) and 10(b)). Let $\ell_u'$ be the intersection of $\ell_u$ with $a_i$. Let $Q$ (resp., $Q'$) be the $y$-monotone subpolygon of $P'$ that contains $u$ (resp., $u'$); by our assumption $Q \neq Q'$. Finally, let $u_+$ (resp., $u_-$) be the vertex of $C_i$ above (resp., below) $u$ in the right $y$-monotone chain of $C_i$. The line segment $uu'$ cannot intersect any edges of $P$, since this would contradict the fact that $u \in C_i$. Similarly, $uu'$ cannot contain any vertices of $P'$: if $v$ is a vertex of $P$ in the interior of $uv$, $u_-$ would be in the triangle $u_-uv$, which contradicts the fact that $u \in C_i$. Whereas, if $v$ is a vertex of $V' \setminus V$ in the interior of $uv'$, $P$ would not be locally convex at $v$, a contradiction with the fact that $Q$ is a piecewise-convex polygon. As a result, and since $Q \neq Q'$, there exists a bridge edge $e$ intersecting $uu'$. Let $w_+$ and $w_-$ be the endpoints of $e$ in $P'$, where $w_+$ lies above the line $\ell_u$ and $w_-$ lies below the line $\ell_u$. In fact neither $w_+$ nor $w_-$ can be a vertex in $V' \setminus V$, since the algorithm in [30] (or [31]) connects a vertex in $V' \setminus V$ in a room $r_k$ with either the $y$-maximal or the $y$-minimal vertex of $C_i$ only. Let $\ell_{w_+}$ (resp., $\ell_{w_-}$) be the line passing through the vertices $u$ and $u_+$ (resp., $u_-$ and $u$). Finally, let $b$ be the point delimited by the lines $\ell_u$, $\ell_{w_+}$ and $a_i$. Now, if $w_+$ or $w_-$ lies in $s$, then $u$ is in the triangle $w_+w_-u_+$ or in the triangle $w_-w_+u_-$ respectively (see Fig. 10(a)). In either case we get a contradiction with the fact that $u \in C_i$. If neither $w_+$ nor $w_-$ lie in $s$, then both $w_+$ and $w_-$ have to be vertices in $r_i$, and moreover $u$ lies in the convex quadrilateral $w_+w_-u_+u_-$; again this contradicts the fact that $u \in C_i$ (see Fig. 10(b)).

2. $u \equiv v_{\text{max}}$. By the maximality of the $y$-coordinate of $u$ in $C_i$, we have that the $y$-coordinate of $u$ is larger than or equal to the $y$-coordinates of both $v_i$ and $v_{i+1}$. Therefore, the line $\ell_u$ intersects the arc $a_i$ exactly twice and, moreover, $a_i$ has a $y$-maximal vertex of $V' \setminus V$ in its interior, which we denote by $v_{\text{max}}'$. (see Fig. 10(c)). Let $\ell_{u'}$ be the intersection of $\ell_u$ with $a_i$ that lies to the right of $u$, and let $Q$ (resp., $Q'$) be the $y$-monotone subpolygon of $P'$ that contains $u$ (resp., $u'$). By assumption $Q \neq Q'$, which implies that there exists a bridge edge $e$ intersecting the line segment $uu'$. Notice, that, as in the case $u \in C_i \setminus \{v_{\text{min}}, v_{\text{max}}\}$, the line segment $uu'$ cannot intersect any edges of $P$, or cannot contain any vertex $v'$ of $V' \setminus V$; the former would contradict the fact that $u \in C_i$, whereas the latter would contradict the fact that $P$ is piecewise-convex. Furthermore, $uu'$ cannot contain vertices of $P$ since this would contradict the maximality of the $y$-coordinate of $u$ in $C_i$. Let $w_+$ and $w_-$ be the endpoints of $e$ above and below $\ell_u$, respectively. Notice that $e$ cannot have $v_{\text{max}}'$ as endpoint, since the only bridge edge that has $v_{\text{max}}'$ as endpoint is the bridge edge $v_{\text{max}}'w_+$. But then $w_+$ must be a vertex of $P$ lying in $r_i$; this contradicts the maximality of the $y$-coordinate of $u$ among the vertices in $C_i$. \[\]
Corollary 9. For each convex arc $a_i$ of $P$, the set $S_i$ computed by the sweep algorithm described above is a superset of the set $C_i^*$. 

Let us now analyze the time and space complexity of Step 1 of the algorithm sketched at the beginning of this subsection. Computing the polygonal approximation $\tilde{P}$ of $P$ requires subdividing $P$ into $y$-monotone subpolygons. This subdivision takes $O(n \log n)$ time and $O(n)$ space. Then we need to compute the sets $S_i$ for each convex arc $a_i$ of $P$. The sets $S_i$ can be implemented as red-black trees. During the course of our algorithm we only perform insertions in $S_i$'s and the $y$-monotone decomposition of $P$ is $O(n)$, we conclude that the total size of the $S_i$'s is $O(n)$ and that we perform $O(n)$ insertions on the $S_i$'s. Therefore we need $O(n \log n)$ time and $O(n)$ space to compute the $S_i$'s and the $C_i^*$'s. The analysis above thus yields the following:

Theorem 10. Let $P$ be a piecewise-convex polygon with $n \geq 2$ vertices. We can compute a guard set for $P$ of size at most $\lceil \frac{3n}{4} \rceil$ in $O(n \log n)$ time and $O(n)$ space.

3. $u \equiv v_{\text{min}}$. This case is entirely symmetric to the case $u \equiv v_{\text{max}}$. $\square$

An immediate corollary of the above lemma is the following:

Corollary 9. For each convex arc $a_i$ of $P$, the set $S_i$ computed by the sweep algorithm described above is a superset of the set $C_i^*$. 

Let us now analyze the time and space complexity of Step 1 of the algorithm sketched at the beginning of this subsection. Computing the polygonal approximation $\tilde{P}$ of $P$ requires subdividing $P$ into $y$-monotone subpolygons. This subdivision takes $O(n \log n)$ time and $O(n)$ space. Then we need to compute the sets $S_i$ for each convex arc $a_i$ of $P$. The sets $S_i$ can be implemented as red-black trees. During the course of our algorithm we only perform insertions in $S_i$'s and the $y$-monotone decomposition of $P$ is $O(n)$, we conclude that the total size of the $S_i$'s is $O(n)$ and that we perform $O(n)$ insertions on the $S_i$'s. Therefore we need $O(n \log n)$ time and $O(n)$ space to compute the $S_i$'s and the $C_i^*$'s. The analysis above thus yields the following:

Theorem 10. Let $P$ be a piecewise-convex polygon with $n \geq 2$ vertices. We can compute a guard set for $P$ of size at most $\lceil \frac{3n}{4} \rceil$ in $O(n \log n)$ time and $O(n)$ space.

3.5. Lower bound constructions

In this subsection we present an $n$-vertex piecewise-convex polygon, for every $n \geq 2$, that cannot be monitored by fewer than $\lceil \frac{n}{2} \rceil$ vertex guards (resp., $\lceil \frac{n}{3} \rceil$ point guards).

It is clear that a piecewise-convex 2-gon (e.g., Fig. 11(a)) requires 1 vertex guard. Fig. 11(b) depicts a piecewise-convex triangle that cannot be monitored by 2 vertex or point guards.

For every integer $n \geq 4$, we give a construction based on a regular $k$-gon $a_1a_2\ldots a_k$, where $k = \lceil \frac{n}{2} \rceil \geq 2$ (in particular, for $k = 2$, a 2-gon is a line segment $a_1a_2$). First assume that $n = 3k$ for an integer $k \geq 2$. Let $\kappa$ denote the circumscribed circle of $a_1a_2\ldots a_k$. Replace each edge $a_ia_{i+1}$, $i = 1,2,\ldots,k$, by a piecewise-convex path $(a_i,b_i,c_i,a_{i+1})$ depicted in Fig. 12(b), to obtain a piecewise-convex $n$-gon $P$. The vertices $b_i$ and $c_i$ are in the left open halfplane delimited by the directed line $\overrightarrow{a_i a_{i+1}}$ and they are separated from the polygon $a_1a_2\ldots a_k$ by the tangent of $\kappa$ at $a_i$. The patterns $(a_i,b_i,c_i,a_{i+1})$ are designed such that at each vertex of $P$, the tangents of the two adjacent edges are the same, which we call the common tangent at the vertex. The common tangent at $a_i$ is also tangent to the circle $\kappa$ at $a_i$; the common tangent at $b_i$ is parallel to the common tangent at $a_i$; and the common tangent at $c_i$ is perpendicular to the common tangents at $a_i$ and $b_i$. Let $A_i$ and $B_i$ denote the empty rooms bounded by $a_i b_i$ and $b_i c_i$, respectively. Let $C_i$ denote the part of the (non-empty) room bounded by $c_i a_{i+1}$ that lies on the left side of both directed lines $\overrightarrow{a_i a_{i+1}}$ and $\overrightarrow{a_{i+1} c_i}$. Note that the regions $A_i$, $B_i$, and $C_i$ are hidden from any vertex of $P$ other than $a_i$, $b_i$, and $c_i$. However, none of $a_i$, $b_i$, and $c_i$ sees all three regions $A_i$, $B_i$, and $C_i$ entirely (in particular, $a_i$ does not see $B_i$ entirely; $b_i$ does not see $C_i$ entirely; and $c_i$
does not see $A_i$ entirely). Hence each triple of regions $\{A_i, B_i, C_i\}$ requires at least two vertex guards at $\{a_i, b_i, c_i\}$. This gives a lower bound of $2k = \frac{2n}{3}$, if $n = 3k$, $k \geq 2$.

Now assume that $n = 3k - 2$ for an integer $k \geq 2$. Replace every edge $a_ia_{i+1}$, for $i = 1, 2, \ldots, k-1$, by a piecewise-convex path $(a_i, b_i, c_i, a_{i+1})$ depicted in Fig. 12(b). The previous argument shows that the resulting piecewise-convex $n$-gon requires $2(k-1) = \lfloor \frac{2n}{3} \rfloor$ vertex guards. Finally, assume that $n = 3k - 1$ for $k \geq 2$. Replace every edge $a_ia_{i+1}$, for $i = 1, 2, \ldots, k-1$, by a piecewise-convex path $(a_i, b_i, c_i, a_{i+1})$ depicted in Fig. 12(b); and replace edge $a_ka_1$ by $(a_k, b_k, a_1)$ depicted in Fig. 12(c). The common tangent at $b_k$ in Fig. 12(c) passes through side $a_ka_1$. The empty room bounded by $a_kb_k$ is not visible from any other vertex but $a_k$ and $b_k$, hence there must be a guard at one of these vertices. Combined with the previous argument, the resulting piecewise-convex $n$-gon requires $2(k-1) + 1 = 2k - 1 = \lfloor \frac{2n}{3} \rfloor$ vertex guards.

**Theorem 11.** For every integer $n \geq 2$, there is a piecewise-convex polygon with $n$ vertices that cannot be monitored by fewer than $\lfloor \frac{2n}{3} \rfloor$ vertex guards.

The lower bound for point guards can be established much more easily. Consider the $n$-vertex piecewise-convex
polygon $C$ shown in Fig. 11(c). It can be readily seen that we need one point guard for any two consecutive prongs of $C$; since $C$ contains $n$ prongs, a minimum of $\lceil \frac{n}{3} \rceil$ point guards are necessary for monitoring $C$.

**Theorem 12.** For every integer $n \geq 2$, there is a piecewise-convex polygon with $n$ vertices that cannot be monitored by fewer than $\lceil \frac{n}{4} \rceil$ point guards.

4. **Piecewise-concave polygons**

In this section we address the problem of finding the minimum number of guards that can jointly monitor any piecewise-concave polygon with $n \geq 3$ vertices. Monitoring a piecewise-concave polygon with vertex guards may be impossible even for very simple configurations (see Fig. 14(a)). In particular we prove the following:

**Theorem 13.** For every integer $n \geq 3$, the minimum number of point guards that can jointly monitor any piecewise-concave polygon with $n$ vertices is $2n - 4$.

To prove the sufficiency of $2n - 4$ point guards we adapt a technique due to Fejes Tóth [28] to our case. Fejes Tóth proved that the free space around $n$ pairwise disjoint compact convex sets can be monitored by $\max(2n, 4n - 7)$ point guards. The edges of a piecewise-convex polygon $P$ are the boundaries of compact convex sets in the plane; these sets however are not necessarily disjoint. The proof in [28] is based on a tessellation of the free space; here we compute a tessellation restricted to $P$.

**Proof.** We are given a piecewise-concave polygon $P$ with $n$ vertices and $n$ concave arcs (see Fig. 13). Successively replace each concave arc $a_i$ by another concave arc $\kappa_i$ with the same endpoints that decreases the polygon maximally. Formally, we construct a sequence of piecewise-concave polygons $P_0 = P, P_1, P_2, \ldots, P_n$. For $i = 1, 2, \ldots, n$, we obtain $P_i$ from $P_{i-1}$ by replacing the concave arc $a_i$ by a concave arc $\kappa_i$ between $v_i$ and $v_{i+1}$ such that $P_i$ is minimal (for containment), that is, there is no piecewise-convex polygon $P'_i$ with $n$ vertices such that $P'_i \subseteq P_i$ and the boundary of $P'_i$ differs from $P_i$ only in the edge between $v_i$ and $v_{i+1}$. Let $\mathcal{K} = \{\kappa_i : 1 \leq i \leq n\}$.

Let us call the region bounded by $a_i$ and $\kappa_i$ the **crescent** of edge $a_i$. Fejes Tóth proved that each arc $\kappa_i$ is a polygonal path, and the arcs $\kappa_i$ partition $P$ into $n$ crescents (one for each edge) and convex polygons, which he called **gaps**. The crescents and convex gaps are the **faces** of a tessellation $T$ of $P$. A vertex of this tessellation is a point incident to at least three faces. Note that every vertex of a gap is a vertex of $T$. Fejes Tóth showed that we can monitor all crescents and all gaps (hence, the entire $P$) if we place point guards as follows:

- place a point guard at every vertex of $T$ incident to at least 3 crescents;
- place two guards at two arbitrary vertices of every triangular gap;
- place a guard at each vertex of every gap with 4 or more vertices.

Construct, now, a planar graph $\Gamma$ with vertex set $\mathcal{K}$. Two vertices $\kappa_i$ and $\kappa_j$ of $\Gamma$ are connected via an edge if $\kappa_i$ and $\kappa_j$ are adjacent. The graph $\Gamma$ is a planar graph combinatorially equivalent to an outerplanar graph $R$ with $n$ vertices. The edges of $\Gamma$ connecting consecutive arcs $\kappa_i, \kappa_{i+1}, 1 \leq i \leq n$, correspond to the boundary edges of $R$, whereas all other edges of $\Gamma$ correspond to diagonals in $R$. Every gap of the tessellation incident to $k$ crescents corresponds to a bounded $k$-gon face of $R$. Every ordinary vertex of the tessellation which is incident to $k$ crescents but no gap corresponds to a bounded $k$-gon face of $R$.

Denote by $d_k$ the number of $k$-gon faces of $R$. Every triangular face of $R$ corresponds to at most 2 point guards, and every $k$-gon face, $k \geq 4$ corresponds to at most $k$ point guards. The total number of point guards is $2d_3 + \sum_{k=4}^{n} kd_k$. This quantity does not decrease if we subdivide a bounded face with $k \geq 4$ vertices into $k - 2$ triangles. In the worst case, all faces are triangles. An outerplanar graph with $n$ vertices has at most $n - 2$ triangular faces, hence the number of point guards is bounded by $2(n - 2)$.

To prove the necessity, refer to the piecewise-concave polygon $P$ in Fig. 14(b). Each one of the pseudo-triangular regions in the interior of $P$ requires exactly two point guards in order to be monitored. Consider for example the pseudo-triangle $\tau$ shown in gray in Fig. 14(b). We need one point along each one of the lines $l_1$, $l_2$ and $l_3$ in order to monitor the regions near the corners of $\tau$, which implies that we need at least two points in order to monitor $\tau$ (two out of the three points of intersection of the lines $l_1$, $l_2$ and $l_3$). The number of such pseudo-triangular regions is exactly $n - 2$, thus we need a total of $2n - 4$ point guards to monitor $P$. 

\[\square\]
5. Discussion and open problems

Every piecewise-convex polygon with \( n \geq 3 \) vertices can be monitored by \( \left\lfloor \frac{2n}{3} \right\rfloor \) vertex guards, which is best possible. Furthermore, we presented an \( O(n \log n) \) time and \( O(n) \) space algorithm for computing a vertex guard set of size at most \( \left\lfloor \frac{2n}{3} \right\rfloor \). Every piecewise-concave polygons with \( n \geq 3 \) vertices can be monitored by \( 2n - 4 \) point guards, which is also best possible. We have not found a piecewise-convex polygon that requires more than \( \left\lceil \frac{n}{2} \right\rceil \) point guards. Closing the gap between the upper and lower bounds, for the case of point guards, remains an open problem.

Beyond the two classes of polygons considered in this paper, it is straightforward to prove the following results (the details are available in a preliminary version of this paper [32]):

1. Given a monotone piecewise-convex polygon \( P \) with \( n \) vertices (i.e., a piecewise-convex polygon \( P \) for which there exists a line \( L \) such that any line \( L^\perp \) perpendicular to \( L \) intersects the boundary of \( P \) at most twice), \( \left\lceil \frac{n}{2} \right\rceil + 1 \) vertex (resp., \( \left\lceil \frac{n}{2} \right\rceil \) point) guards are always sufficient and sometimes necessary in order to monitor \( P \).
2. Given a locally convex polygon \( P \) with \( n \) vertices, \( n \) point guards are always sufficient and sometimes necessary in order to monitor \( P \). In particular, the \( n \) vertices of \( P \) are a guard set for \( P \).
3. Given a monotone locally convex polygon (defined in direct analogy to monotone piecewise-convex polygons), \( \left\lceil \frac{n}{2} \right\rceil + 1 \) vertex or point guards are always sufficient and sometimes necessary.
4. Finally, there exist general polygons that cannot be monitored with a finite number of point guards.

Karavelas [33, 34] has recently shown that every piecewise-convex polygon with \( n \) vertices can be monitored by \( \left\lfloor \frac{2n+1}{4} \right\rfloor \) edge guards or by \( \left\lfloor \frac{n+1}{2} \right\rfloor \) guards each of which is either an edge or a straight-line diagonal of the polygon; whereas \( \left\lfloor \frac{n}{4} \right\rfloor \) edges or straight-line diagonals are sometimes necessary. Other types of guarding problems have been studied in the literature, which either differ on the type of guards, the topology of the polygons considered (e.g., polygons with holes) or the guarding model; see the book by O’Rourke [25], the surveys by Shermer [26] and by Urrutia [27] for an extensive list of the variations of the art gallery problem with respect to the types of guards or the guarding model. It would be interesting to extend these results to the families of curvilinear polygons presented in this paper.
Figure 14: (a) A piecewise-concave polygon $P$ that cannot be monitored solely by vertex guards. Two consecutive edges of $P$ have a common tangent at the common vertex and as a result the three vertices of $P$ see only the points along the dashed segments. (b) A piecewise-concave polygon $P$ that requires $2n - 4$ point guards in order to be monitored.

Acknowledgements

The authors wish to thank Ioannis Z. Emiris, Hazel Everett and Günter Rote for useful discussions about the problem. We are also thankful to Valentin Polishchuk as well as an anonymous referee for their comments and suggestions. Work partially supported by the IST Programme of the EU (FET Open) Project under Contract No IST-006413 – (ACS - Algorithms for Complex Shapes with Certified Numerics and Topology).

References