

A CONSTANT-FACTOR APPROXIMATION ALGORITHM FOR VERTEX GUARDING A WV-POLYGON *

Stav Ashur,[†] Omrit Filtser,[‡] and Matthew J. Katz[§]

ABSTRACT. The problem of vertex guarding a simple polygon was first studied by Subir K. Ghosh (1987), who presented a polynomial-time $O(\log n)$ -approximation algorithm for placing as few guards as possible at vertices of a simple n -gon P , such that every point in P is visible to at least one of the guards. Ghosh also conjectured that this problem admits a polynomial-time algorithm with constant approximation ratio. Due to the centrality of guarding problems in the field of computational geometry, much effort has been invested throughout the years in trying to resolve this conjecture. Despite some progress (surveyed below), the conjecture remains unresolved to date. In this paper, we confirm the conjecture for the important case of weakly visible polygons, by presenting a $(2 + \varepsilon)$ -approximation algorithm for guarding such a polygon using vertex guards. A simple polygon P is *weakly visible* if it has an edge e , such that every point in P is visible from some point on e . We also present a $(2 + \varepsilon)$ -approximation algorithm for guarding a weakly visible polygon P , where guards may be placed anywhere on P 's boundary (except in the interior of the edge e). Finally, we present an $O(1)$ -approximation algorithm for vertex guarding a polygon P that is weakly visible from a chord.

Our algorithms are based on an in-depth analysis of the geometric properties of the regions that remain unguarded after placing guards at the vertices to guard the polygon's boundary. Finally, our algorithms may become useful as part of the grand attempt of Bhattacharya et al. to prove the original conjecture, as their approach is based on partitioning the underlying simple polygon into a hierarchy of weakly visible polygons.

1 Introduction

The Art Gallery Problem is a classical problem in computational geometry, posed by Victor Klee in 1973: Place a minimum number of points (representing guards) in a given simple polygon P (representing an art gallery), so that every point in P is seen by at least one of the placed points. We say that a point $p \in P$ *sees* (or *guards*) a point $q \in P$ if the line segment \overline{pq} is contained in P . We say that a subset $G \subseteq P$ *guards* P , if every point in P is seen by at least one point in G .

*An earlier version of this paper (excluding the proof of [Theorem 16](#)) was presented at WAOA'20 [3]. O. Filtser was supported by the Eric and Wendy Schmidt Fund for Strategic Innovation, by the Council for Higher Education of Israel, and by Ben-Gurion University of the Negev. M. Katz was supported by grant 1884/16 from the Israel Science Foundation.

[†]Ben-Gurion University of the Negev, Beer-Sheva, Israel, stavshe@post.bgu.ac.il

[‡]The Open University of Israel, Raanana, Israel, omrit.filtser@gmail.com

[§]Ben-Gurion University of the Negev, Beer-Sheva, Israel, matya@cs.bgu.ac.il

There are numerous variants of the art gallery problem, which are also referred to as the art gallery problem. These variants differ from one another in (i) the underlying domain, e.g., simple polygon, polygon with holes, orthogonal polygon, or terrain, (ii) which parts of the domain must be guarded, e.g., only its vertices, only its boundary, or the entire domain, (iii) the type of guards, e.g., static, mobile, or with various restrictions on their coverage area such as limited range, (iv) the restrictions on the location of the guards, e.g., only at vertices (vertex-guards), only on the boundary (boundary-guards), or anywhere (point-guards), and (v) the underlying notion of visibility, e.g., line of sight, rectangle visibility, or staircase visibility. It is impossible to survey here the vast literature on the art gallery problem, ranging from combinatorial and optimization results to hardness of computation results, so we only mention the book by O'Rourke [28] and a small sample of recent papers [1, 9].

In this paper, we deal with the version of the art gallery problem, where the guards are confined to the boundary of the underlying polygon, and in particular to its vertices. Such guards are referred to as *boundary* guards or *vertex* guards, respectively. The first to present results for this version was Ghosh [15, 16], who gave a polynomial-time $O(\log n)$ -approximation algorithm for guarding either a simple polygon or a polygon with holes using vertex (or edge) guards (see below for the definition of an edge guard). In the related work paragraph below we survey many of the subsequent results for this version.

We consider an important family of simple polygons, namely, the family of *weakly visible* polygons. A simple polygon P is *weakly visible* if it has an edge e , such that every point in P is visible from some point on e , or, in other words, a guard patrolling along e can see the entire polygon. We also consider polygons that are weakly visible from a chord, rather than an edge, where a *chord* in a polygon P is a line segment whose endpoints are on the boundary of P and whose interior is contained in the interior of P .

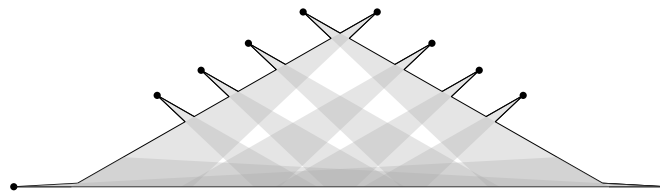


Figure 1: A WV-polygon P and a subset of its vertices that guards P 's boundary but not its interior.

The problem of guarding a weakly visible polygon (WV-polygon) P by vertex guards was studied by Bhattacharya et al. [7]. They first present a 4-approximation algorithm for vertex guarding *only* the vertices of P . Next, they claim that this algorithm places the guards at vertices in such a way that each of the remaining unguarded regions of P has a boundary edge which is contained in an edge of P . Based on this claim, they devise a 6-approximation algorithm for vertex guarding P 's boundary (by adding vertex guards to their set of vertex guards that guards P 's vertices), and present it as a 6-approximation algorithm for guarding P (boundary plus interior). Unfortunately, this claim is false, i.e., the interior of P might still contain unguarded regions; counterexamples were constructed and approved by the authors (who are now attempting to fix their algorithm, so as to obtain an algorithm for vertex guarding P entirely) [8]. Thus, the challenge of obtaining a constant-factor approximation

algorithm for guarding a WV-polygon with vertex guards or boundary guards is still on. Figure 1 depicts a WV-polygon P and a subset of its vertices (which is not necessarily the one returned by the algorithm of Bhattacharya et al. [7]) that guards P 's boundary but not its interior.

The main result of this paper is such an algorithm. Specifically, denote by OPT the size of a minimum-cardinality subset of the vertices of P that guards P . We present a polynomial-time algorithm that finds a subset I of the vertices of P , such that I guards P and $|I| \leq (2 + \varepsilon)\text{OPT}$, for any constant $\varepsilon > 0$.

Already in 1987, Ghosh conjectured that there exists a constant-factor approximation algorithm for vertex guarding a simple polygon. Recently, Bhattacharya et al. [6] managed to devise such an algorithm for vertex guarding the *vertices* or the *boundary* of a simple polygon P , by first partitioning P into a hierarchy of WV-polygons according to the link distance from a starting vertex. They also present such an algorithm for vertex guarding P (boundary plus interior), however, this algorithm is erroneous, since it relies on the false statement mentioned above. Thus, our result may prove useful towards resolving Ghosh's conjecture.

We note that our algorithm for vertex guarding a WV-polygon is more general than that of Bhattacharya et al. [7] (assuming it can be fixed), since unlike theirs, it is not based on a *specific* procedure for placing guards at vertices so as to see all the polygon's vertices. This we believe makes our algorithm a more suitable tool for dealing with the general problem (i.e., for simple polygons), and will enable a better approximation ratio, if successful.

Prior to our result, the only (non-trivial) family of polygons for which a constant-factor approximation algorithm for guarding a member of the family was known, is the family of monotone polygons. Specifically, Krohn and Nilsson [24] showed that vertex guarding a monotone polygon is NP-hard, and presented a constant-factor approximation algorithm for *point* guarding such a polygon (as well as an $O(\text{OPT}^2)$ -approximation algorithm for point guarding an orthogonal polygon).

Related work. Several improvements to Ghosh's $O(\log n)$ -approximation algorithm were obtained over the years. In 2006, Efrat and Har-Peled [12] described a randomized polynomial-time $O(\log \text{OPT})$ -approximation algorithm for vertex guarding a simple polygon, where the approximation factor is correct with high probability. Moreover, they considered the version in which the guards are restricted to the points of an arbitrarily dense grid, and presented a randomized algorithm which returns an $O(\log \text{OPT})$ -approximation with high probability, where OPT is the size of an optimal solution to the modified problem and the running time depends on the ratio between the diameter of the polygon and the grid size. Combining ideas from the latter algorithm and from the work of Deshpande et al. [11], Bonnet and Miltzow [9] presented a randomized polynomial-time $O(\log \text{OPT})$ -approximation algorithm for point guards, assuming integer coordinates and a general position assumption on the vertices of the underlying simple polygon.

In 2011, King and Kirkpatrick [22] observed that by applying methods that were developed for the Set Cover problem after the publication of Ghosh's algorithm, one can obtain an $O(\log \text{OPT})$ approximation factor for vertex guarding a simple polygon (and an

$O(\log h \log \text{OPT})$ factor for vertex guarding a polygon with h holes). Moreover, they improved the approximation factor to $O(\log \log \text{OPT})$ for guarding a simple polygon either with vertex guards or boundary guards, where in the former case the running time is polynomial in n and in the latter case it is polynomial in n and the spread of the vertices; see also [23].

Most of the variants of the art gallery problem are NP-hard. O'Rourke and Supowit [29] proved this for polygons with holes and point guards, Lee and Lin [26] proved this for simple polygons and vertex guards, and Aggarwal [2] generalized the latter proof to simple polygons and point guards. Eidenbenz et al. [13] presented a collection of hardness results. In particular, they proved that it is unlikely that a PTAS exists for vertex guarding or point guarding a simple polygon. Recently, Abrahamsen et al. [1] proved that the art gallery problem is $\exists\mathbb{R}$ -complete, for simple polygons and point guards.

WV-polygons were defined and studied in the context of mobile guards [28]. An *edge guard* is a guard that traverses an edge of the polygon. Thus, a simple polygon is weakly visible if and only if it can be guarded by a single edge guard, and the problem of vertex guarding a WV-polygon is equivalent to the problem of replacing the single edge guard by as few vertex guards as possible. Avis and Toussaint [5] presented a linear-time algorithm for detecting whether a polygon is weakly visible from a given edge. Subsequently, Sack and Suri [30] and Das et al. [10] devised linear-time algorithms which output all the edges (if any) from which the polygon is weakly visible. Algorithms for finding either an edge or a chord from which the polygon is weakly visible were given by Ke [21] and by Ghosh et al. [17]. Finally, Bhattacharya et al. [7] proved that the problem of point guarding a WV-polygon is NP-hard, and that there does not exist a polynomial-time algorithm for vertex guarding a WV-polygon with holes with approximation factor better than $((1 - \varepsilon)/12) \ln n$, unless $\text{P}=\text{NP}$.

Our algorithm for vertex guarding a WV-polygon uses a solution to the problem of guarding the boundary of a WV-polygon using vertex guards. This problem admits a local-search-based PTAS (see [4, 20]), which is similar to the local-search-based PTAS of Gibson et al. [18] for vertex guarding the vertices of a 1.5D-terrain. The proof of both these PTASs is based on the proof scheme of Mustafa and Ray [27].

Results. Our algorithm for vertex guarding a WV-polygon P (presented in Section 3) consists of two main parts. In the first part, it computes a subset G of the vertices of P that guards P 's boundary. This is done by applying a known algorithm for this task. In the second part, it computes a subset G' of the vertices of P of size at most $|G|$, such that $G \cup G'$ guards P (boundary plus interior). Thus, if we apply the algorithm of [4] for computing G , then the approximation ratio of our algorithm is $2 + \varepsilon$, since the former algorithm guarantees that $|G| \leq (1 + \varepsilon/2)\text{OPT}_\partial$, where OPT_∂ is the size of a minimum-cardinality subset of the vertices of P that guards P 's boundary, and clearly $\text{OPT}_\partial \leq \text{OPT}$.

Let x be a vertex in G and let $\text{Vis}(x)$ be the visibility polygon of x (i.e., $\text{Vis}(x)$ is the set of all points of P that are visible from x), then $P \setminus \text{Vis}(x)$ is a set of connected regions, which we refer to as *pockets*. Moreover, a connected subset H of P is a *hole* in P w.r.t. G if (i) there is no point in H that is visible from G , and (ii) H is maximal in the sense that any connected subset of P that strictly contains H has a point that is visible from G . The

second part of our algorithm (and its proof) are based on a deep structural analysis and characterization of the pockets and holes in P (presented in [Section 2](#)).

The requirement that G is a subset of the vertices of P is actually not necessary; the second part of our algorithm only needs a set of boundary points that guards P 's boundary. This observation enables us to use a smaller number of guards, assuming that boundary guards are allowed.

Finally, in [Section 4](#), we consider the more general family of polygons, those that are weakly visible from a chord. Notice that a chord \overline{uv} in P slices P into two polygons, such that each of them is weakly visible w.r.t. \overline{uv} (which is an edge in each of them). After updating two of the geometric claims presented in [Section 2](#), we show how to apply our algorithm to a polygon that is weakly visible from a chord. The approximation ratio in this case is $3|G|$ (rather than $2|G|$). However, we are not aware of any specialized algorithm for computing G in this case. We could apply though the (yet unpublished) c -approximation algorithm of Bhattacharya et al. [6] for vertex guarding the boundary of a simple polygon, where c is a relatively large constant, to obtain an approximation ratio of $3c$.

2 Structural analysis

For two points $x, y \in \mathbb{R}^2$, we denote by \overline{xy} the line segment whose endpoints are x and y , and by ℓ_{xy} the line through x and y . We denote by ℓ_x the horizontal line through x .

Let P be a polygon whose set of vertices is $V = \{u = v_1, v_2, \dots, v_n = v\}$, and which is weakly visible from its edge $e = \overline{uv}$. We denote the boundary of P by ∂P . The edges of P are the segments $\overline{v_1v_2}, \overline{v_2v_3}, \dots, \overline{v_{n-1}v_n}$ and $\overline{v_1v_n} = \overline{uv}$. We assume w.l.o.g. that \overline{uv} is contained in the x -axis, and u is to the left of v . Furthermore, we assume that P is contained in the (closed) halfplane above the x -axis; in particular, the angles at u and v are convex. This assumption can be easily removed, as we show towards the end of [Section 3](#). Finally, for simplicity we assume that there are no two vertices of P with the same x -coordinate.

2.1 Visibility polygons

For a point $p \in P$, let $\text{Vis}(p) = \{q \mid \overline{pq} \subseteq P\}$ be the *visibility polygon* of p . In other words, $\text{Vis}(p)$ is the set of all points of P that are visible from p . By definition $\text{Vis}(p)$ is a star-shaped polygon, and thus clearly also a simple polygon, contained in P (see [Figure 2](#)).

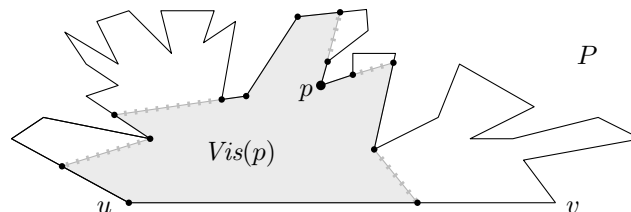


Figure 2: The visibility polygon of a point $p \in P$.

Any vertex of P that belongs to $\text{Vis}(p)$ is also considered a vertex of $\text{Vis}(p)$. Consider

the set of edges of $\text{Vis}(p)$. Some of these edges are fully contained in ∂P . The edges that are not contained in ∂P are **constructed edges**: these are edges whose endpoints are on ∂P and whose interior is contained in the interior of P (the gray dotted edges in Figure 2).

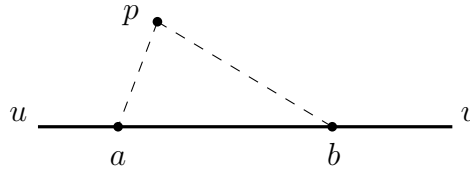


Figure 3: If p sees both a and b , then p sees the entire segment \overline{ab} .

Claim 1. *For any $p \in P$, there exists a single edge of $\text{Vis}(p)$ that is contained in \overline{uv} .*

Proof. Since p is visible from \overline{uv} , $\overline{uv} \cap \text{Vis}(p) \neq \emptyset$. Let a (resp. b) be the leftmost (resp. rightmost) point on \overline{uv} that belongs to $\text{Vis}(p)$ (see Figure 3). The triangle $\triangle pab$ cannot contain points of ∂P in its interior, because ∂P cannot cross the segments \overline{pa} and \overline{pb} . Therefore, p sees every point in $\triangle pab$, and in particular it sees every point between a and b on \overline{uv} . \square

2.2 Pockets

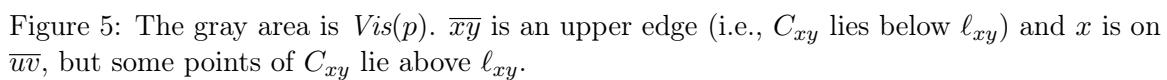
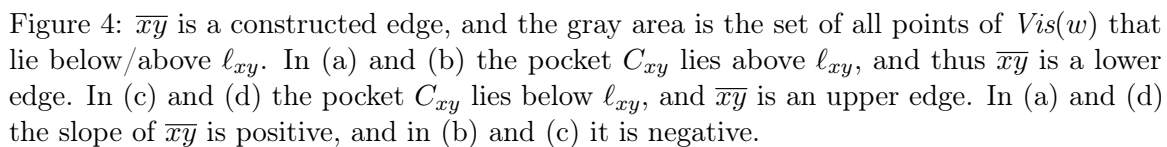
Consider $P \setminus \text{Vis}(p)$ (see Figure 2). This is a set of connected regions, which we refer to as **pockets**. Since P is a simple polygon, each pocket C is adjacent to a single constructed edge, and therefore the intersection of ∂P and C is connected. We refer to $\partial P \cap C$ as the boundary of the pocket C , and denote it by ∂C . (Thus the constructed edge itself is not part of ∂C .)

Let \overline{xy} be a constructed edge such that x is below y (w.r.t. the y -axis), and denote by C_{xy} the pocket adjacent to \overline{xy} . We say that C_{xy} lies above (resp. below) ℓ_{xy} , if for any point w in the interior of \overline{xy} , all the points that w sees (points of $\text{Vis}(w)$) that lie above (resp. below) ℓ_{xy} , belong to C_{xy} (see Figure 4 (a)/(b) for “above” and (c)/(d) for “below”). Notice that since \overline{xy} divides P into two parts, C_{xy} and $P \setminus C_{xy}$, ‘stepping off’ \overline{xy} to one side, places us in the interior of C_{xy} (and ‘stepping off’ \overline{xy} to the other side places us in the interior of $P \setminus C_{xy}$).

Note that if C_{xy} lies above (resp. below) ℓ_{xy} , it does not necessarily mean that all the points of C_{xy} lie above (resp. below) ℓ_{xy} . Indeed, when $x \in \overline{uv}$, C_{xy} may have points on both sides of ℓ_{xy} (see Figure 5).

We say that \overline{xy} is an **upper edge** (resp. **lower edge**), if C_{xy} lies below (resp. above) ℓ_{xy} . Notice that by Claim 1, $\text{Vis}(p)$ has at most two constructed edges with an endpoint in \overline{uv} . Moreover, at most one of these edges is an upper edge.

Claim 2. *Let \overline{xy} and $\overline{x'y'}$ be two constructed edges (which belong to two different visibility polygons), such that \overline{xy} crosses $\overline{x'y'}$, and y' is on the same side of ℓ_{xy} as C_{xy} . Then, $y' \in \partial C_{xy}$ (see Figure 6).*



Claim 3. *Let \overline{xy} be a constructed edge of $\text{Vis}(p)$ such that x is below y , then C_{xy} lies above ℓ_x (in the weak sense), i.e., every point in C_{xy} is either above or on ℓ_x . Moreover, if $x \notin \overline{uv}$, then C_{xy} lies strictly above ℓ_x , i.e., every point in C_{xy} is above ℓ_x .*

If $x \notin \overline{uv}$, then x is above ℓ_{uv} (see Figure 4). By the observation above, $\partial C_{xy} \cap \overline{uv} = \emptyset$ (and thus of course $C_{xy} \cap \overline{uv} = \emptyset$). Let z be any point in C_{xy} . Since P is a WV-polygon, there exists a point z' on \overline{uv} such that $\overline{zz'} \subseteq P$. The segment $\overline{zz'}$ has to cross \overline{xy} , because

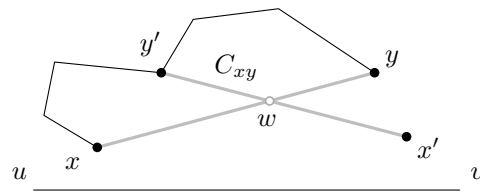


Figure 6: Two constructed edges that cross each other.

$z \in C_{xy}$ and $z' \notin C_{xy}$. Since x is below y , this crossing point is above x , which in turn is above z' , so we get that z is above x . We conclude that C_{xy} lies strictly above ℓ_x . \square

2.3 Holes

Let $G \subseteq P$ be a set of points that guards ∂P . A connected subset H of P is a **hole** in P w.r.t. G if (i) there is no point in H that is visible from G , and (ii) H is maximal in the sense that any connected subset of P that strictly contains H has a point that is visible from G .

Let H be a hole in P w.r.t. G , then clearly H is a simple polygon. Each edge of H lies on some constructed edge e , and we say that H (and this edge of H) **lean** on the edge e and that e **supports** H (and this edge of H). Notice that H is fully contained in the pocket adjacent to e . Moreover, since $H \cap \partial P = \emptyset$, we can view H as an intersection of halfplanes (defined by the lines containing the constructed edges supporting the edges of H), and thus obtain the following observation.

Observation 4. Any hole H in P w.r.t. G is a convex polygon.

Another immediate but useful observation is the following.

Observation 5. Any hole H in P w.r.t. G leans on at least one upper edge and at least one lower edge.

Next, we show that any hole can be guarded by a single vertex, and that such a vertex can be found in polynomial time. We actually prove a slightly more general claim.

Lemma 6. Let S be a simple convex polygonal region contained in P . Then, there exists a vertex w of P , such that S is visible from w , i.e., every point in S is visible from w . Moreover, w can be found in polynomial time.

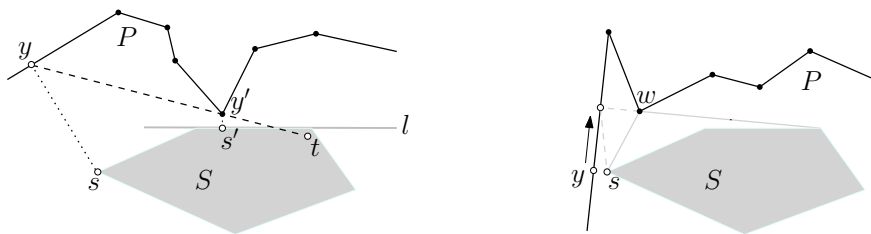


Figure 7: Proof of Lemma 6.

Proof. If P and S share a vertex, then we are done. Otherwise, let y be a point on ∂P that is closest to S , where the distance between a point p and S is $\text{dist}(p, S) = \min\{\|p - s\| \mid s \in S\}$, and let s be a point in S closest to y . We first prove that S is visible from y . If y is also on ∂S , then y clearly sees S , so assume that y is not on ∂S and that there exists a point $t \in S$ that is not visible from y . Then, there exists a point y' on ∂P that lies in the interior of the segment \overline{yt} . Let s' be a point in S closest to y' , and let l be the line through s' and perpendicular to $\overline{y's'}$. Since S is convex, it is contained in one of the two halfplanes supported by l . Assume w.l.o.g. that l is horizontal and that S is contained in the bottom halfplane supported by l (see Figure 7, left). Then y' is above l and its projection onto l is s' . Now, it is impossible that y is not above y' (in terms of y -coordinate), since this would imply that t is above l . So y must be above y' , but then $\|y - s\| > \|y' - s'\|$ and we have reached a contradiction.

Now, if y is a vertex of P , then we are done. Otherwise, we slide y along ∂P , in any one of the directions, until the convex hull of $S \cup y$ meets a vertex w of P (see Figure 7, right). The vertex w is either an endpoint of the edge of P on which we slide y , or it is another vertex of P that lies on one of the tangents to S through y . In both cases, w clearly sees S . \square

3 The Algorithm

We show that given a set G of vertices that guards ∂P , one can find a set of vertices G' of size at most $|G|$ such that $G \cup G'$ guards P (boundary plus interior).

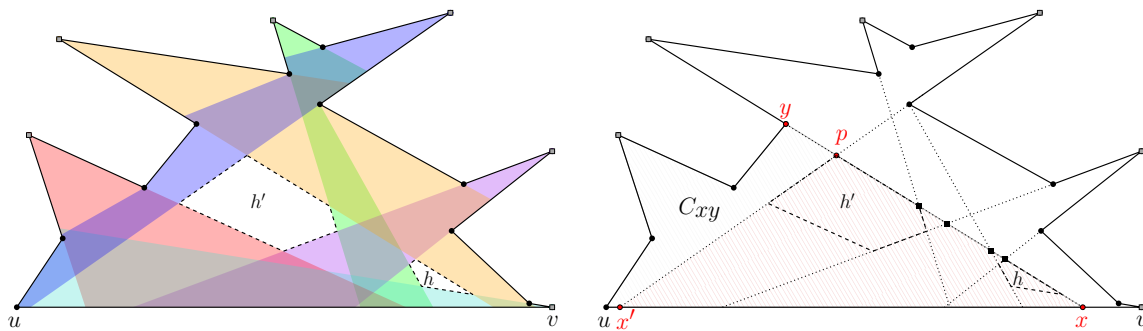


Figure 8: Left: The polygon P and the set of guards G . Right: In the iteration for the edge \overline{xy} , p is the topmost among the five relevant intersection points, and a vertex that guards $\triangle xpx'$ is added to G' .

Algorithm 1 Vertex Guarding a WV-polygon

Input: A polygon P weakly visible from its edge \overline{uv} , and a set G of vertices of P that guards ∂P (see Figure 8, left).

Output: A set of vertices G' , such that $G \cup G'$ guards P .

Compute the set E of the constructed edges of the visibility polygons of the vertices in G .

For each upper edge $e = \overline{xy}$ in E with $x \in \overline{uv}$:

1. Find the topmost intersection point p of e with an edge $e' = \overline{x'y'}$ in E with $x' \in \overline{uv}$ such that x' is on the same side of ℓ_{xy} as the pocket C_{xy} (see Figure 8, right).
2. If such a point p exists (then the triangle $\triangle xpx'$ is contained in P), find a vertex that guards $\triangle xpx'$ (see description in the proof of Lemma 6) and add it to G' .

Now our goal is to show that for any hole H in P w.r.t. G there exists a vertex in G' that guards it. More precisely, we show that H is contained in one of the triangles considered by the algorithm.

We first show in Claim 8 that H leans on an upper edge with an endpoint in \overline{uv} . Recall that H leans on at least one upper edge (Observation 5).

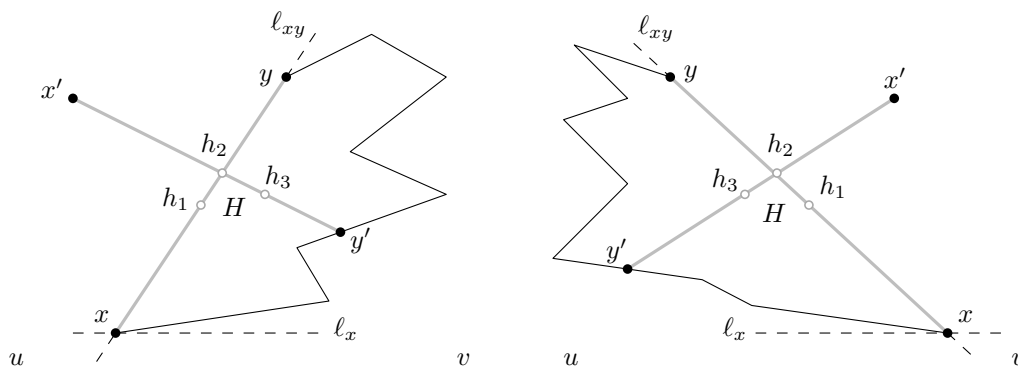


Figure 9: Two edges of the hole H : h_1h_2 and h_2h_3 . The edge h_1h_2 leans on an upper edge and h_1 is below h_2 , thus h_2h_3 has to lean on an upper edge such that h_2 is below h_3 (and not as drawn in the figure).

Claim 7. Let $H = (h_1, \dots, h_k)$ be a hole in P w.r.t. G . Assume that the edge h_1h_2 of H leans on an upper edge \overline{xy} , such that x is below y , $x \notin \overline{uv}$, and h_1 is below h_2 . Then h_2h_3 also leans on an upper edge and h_2 is below h_3 .

Proof. Since \overline{xy} is an upper edge, H (which is contained in C_{xy}) lies below ℓ_{xy} , and by Claim 3, H lies strictly above ℓ_x (see Figure 9).

Let $\overline{x'y'}$ be the constructed edge supporting h_2h_3 (i.e., h_2h_3 leans on $\overline{x'y'}$), and assume that y' is on the same side of ℓ_{xy} as h_3 . (h_2 is the crossing point of \overline{xy} and $\overline{x'y'}$, so

x' and y' are on different sides of ℓ_{xy} .) Hence, y' is on the same side of ℓ_{xy} as C_{xy} , and by [Claim 2](#), $y' \in \partial C_{xy}$. Moreover, since $x \notin \overline{uv}$, we know by [Claim 3](#) that y' is strictly above x (and in particular, $y' \notin \overline{uv}$).

Next we show that h_2 is below h_3 , by showing that x' is below y' . Indeed, if x' is above y' (as in [Figure 9](#)), then since x is on the same side of $\ell_{x'y'}$ as h_1 , it is also on the same side of $\ell_{x'y'}$ as $C_{x'y'}$. Now, by [Claim 2](#) again, $x \in \partial C_{x'y'}$, and therefore, since $y' \notin \overline{uv}$, we know by [Claim 3](#) that x is strictly above y' , a contradiction.

Finally, $\overline{x'y'}$ is clearly an upper edge, since h_1 (which is in $C_{x'y'}$) is below $\overline{x'y'}$. \square

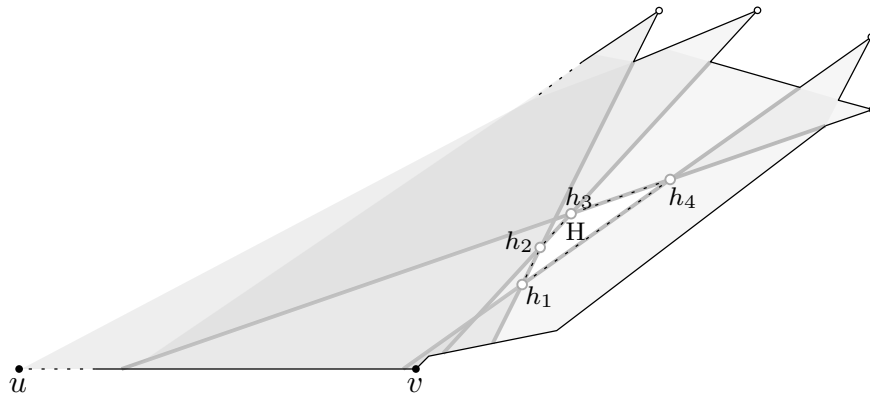


Figure 10: The edges of H lean on a sequence of upper edges.

Claim 8. *Let H be a hole in P w.r.t. G . Then H leans on an upper edge with an endpoint in \overline{uv} .*

Proof. H has at least one edge h_1h_2 that leans on an upper edge $\overline{x_1y_1}$, and assume w.l.o.g. that h_1 is below h_2 . If x_1 or y_1 are on \overline{uv} then we are done. Otherwise, by [Claim 7](#), the edge h_2h_3 also leans on an upper edge $\overline{x_2y_2}$, and h_2 is below h_3 . Again, if one of x_2, y_2 is on \overline{uv} then we are done, otherwise by [Claim 7](#), h_3h_4 leans on an upper edge and h_3 is below h_4 . This process must end since $H = (h_1, \dots, h_k)$ is a convex polygon, and the only way for it to end is if $\overline{x_iy_i}$ is an upper edge such that one of x_i, y_i is on \overline{uv} , for some $1 \leq i \leq k-1$ (see [Figure 10](#)). \square

Claim 9. *Assume that h_1h_2 leans on an upper edge \overline{xy} such that x is below y and $x \in \overline{uv}$, and that h_1 is below h_2 . Then h_2h_3 leans on a constructed edge $\overline{x'y'}$ (where x' is below y') such that $x' \in \overline{uv}$.*

Proof. Let $\overline{x'y'}$ be the constructed edge supporting h_2h_3 such that x' is below y' . The constructed edges \overline{xy} and $\overline{x'y'}$ cross each other at the point h_2 , so by [Claim 2](#) we get that $x \in \partial C_{x'y'}$, because h_1 and x are on the same side of $\ell_{x'y'}$. Now, by [Claim 3](#), x lies (weakly) above $\ell_{x'}$, but $x \in \overline{uv}$, so $x' \in \ell_{uv}$. \square

We are now ready to prove the correctness of our algorithm.

Claim 10. Any hole H in P w.r.t. G is contained in one of the triangles considered by Algorithm 1.

Proof. By Claim 8, H has an edge which leans on an upper edge with an endpoint in \overline{uv} . Let h_1h_2 , where h_1 is below h_2 , be such an edge of H of minimum (absolute) slope, and let \overline{xy} , where $x \in \overline{uv}$, be the upper edge supporting h_1h_2 . By Claim 9, the edge h_2h_3 leans on a constructed edge $\overline{x'y'}$ such that $x' \in \overline{uv}$.

Assume first that the algorithm found a point p on \overline{xy} . If p is not below h_2 , then the triangle corresponding to p contains H and we are done. If p is below h_1 (or $p = h_1$), then x' is not on the same side of \overline{xy} as C_{xy} (because otherwise, p would not be a point below h_2). But, if so, then the (absolute) slope of h_2h_3 is smaller than that of h_1h_2 , a contradiction. Now, if the algorithm did not find such a point p on \overline{xy} , then as before this means that x' is not on the same side of \overline{xy} as C_{xy} and we reach a contradiction. \square

Approximation ratio. Notice that Algorithm 1 only uses a subset E' of the constructed edges, namely, the subset of upper edges with an endpoint in \overline{uv} . Moreover, by Claim 1, we get that $|E'| \leq |G|$, since for any vertex $w \in G$, at most one of the (at most two) constructed edges of $\text{Vis}(w)$ with an endpoint in \overline{uv} is an upper edge. For example, in Figure 2 the sole constructed edge of p with an endpoint in \overline{uv} is not an upper edge. Finally, in each iteration, at most one vertex is added to G' . We conclude that $|G'| \leq |E'| \leq |G|$. Now, since $|G| \leq (1 + \varepsilon/2)\text{OPT}$ (assuming G is computed by the algorithm of [4]), the approximation ratio of Algorithm 1 is $2 + \varepsilon$, for any $\varepsilon > 0$.

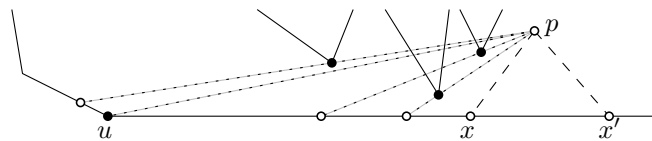
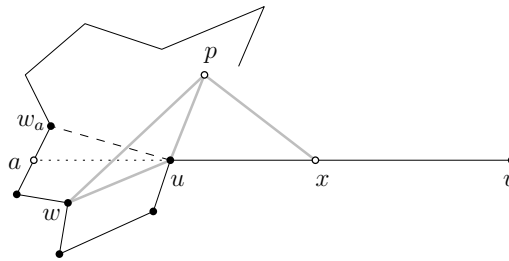


Figure 11: Finding a vertex of P that guards the triangle $\triangle xpx'$.

Running time. The running time of the algorithm of [4] for finding a set of vertices G of size $(1 + \varepsilon/2)\text{OPT}$ that guards ∂P is $O(n^{O(1/\varepsilon^2)})$, for any $\varepsilon > 0$. The set of constructed edges E can be computed in $O(|G|n)$ time [19, 25]. Since $|E'| \leq |G|$, the total number of intersection points (Step 1 of Algorithm 1) is $O(|G|^2)$. Moreover, a vertex that guards the triangle $\triangle xpx'$ (Step 2 of Algorithm 1) can be found in linear time by the following simple algorithm. (We could use the algorithm described in the proof of Lemma 6, but in this special case it is not necessary.) Assume, w.l.o.g., that x is to the left of x' . Then, for each vertex w of P such that w is below p , compute the crossing point (if it exists) between \overline{uv} and the ray emanating from p and passing through w . Now, among the vertices whose corresponding crossing point is between u and x , pick the one closest to x (see Figure 11). Thus, the total running time of Algorithm 1 (given the set G) is $O(|G|n) = O(n^2)$. Note that for a sufficiently small ε the total running time is dominated by $O(n^{O(1/\varepsilon^2)})$.

Figure 12: A polygon with a concave angle at u .

Removing the convexity assumption. Up to now, we have assumed that the angles at u and at v are convex. As in [4], this assumption can be easily removed. Assume, e.g., that the angle at u is concave, and let a be the first point on P 's boundary (moving clockwise from u) that lies on the x -axis (see Figure 12). Then, every point in the open portion of the boundary between u and a is visible from u and is not visible from any other point on \overline{uv} . Moreover, for any vertex w in this portion of P 's boundary, if w sees some point p in P above the x -axis, then so does u . Indeed, since P is weakly-visible from \overline{uv} , there exists a point $x \in \overline{uv}$ that sees p . In other words, \overline{xp} is contained in P , as well as \overline{uw} and \overline{wp} . Thus, the quadrilateral $uwp x$ does not contain points of ∂P in its interior, and since \overline{up} is contained in it, we conclude that u sees p . Therefore, we may assume that an optimal guarding set does not include a vertex from this portion. Moreover, we may assume that the size of an optimal guarding set is greater than some appropriate constant, since otherwise we can find such a set in polynomial time. Now, let w_a be the first vertex following a . We place a guard at u and replace the portion of P 's boundary between u and w_a by the edge $\overline{uw_a}$. Notice that every point in the removed region is visible from u . Similarly, if the angle at v is concave, we define the point b and the vertex w_b (by moving counterclockwise from v), place a guard at v , and replace the portion of P 's boundary between v and w_b by the edge $\overline{vw_b}$. Finally, we apply the algorithm of [4] to the resulting polygon, after adjusting its parameters so that together with u and v we still get a $(1 + \varepsilon/2)$ -approximation of an optimal guarding set for ∂P .

Theorem 11. *Given a WV-polygon P with n vertices and $\varepsilon > 0$, one can find in $O(n^{\max\{2, O(1/\varepsilon^2)\}})$ time a subset G of the vertices of P , such that G guards P (boundary plus interior) and G is of size at most $(2 + \varepsilon)OPT$, where OPT is the size of a minimum-cardinality such set.*

Boundary guards. Let OPT^B be the size of a minimum-cardinality set of points on P 's boundary (except the interior of the edge \overline{uv}) that guards P (boundary plus interior), and let OPT_∂^B be the size of a minimum-cardinality such set that guards P 's boundary; clearly, $OPT_\partial^B \leq OPT^B \leq OPT$. A PTAS for finding a set G^B of points on $(\partial P \setminus \overline{uv}) \cup \{u, v\}$ that guards ∂P is described in [4], that is, $|G^B| \leq (1 + \varepsilon)OPT_\partial^B$, for any $\varepsilon > 0$. Its running time is $O(n^{O(1/\varepsilon^2)})$, and it is similar to the corresponding PTAS of Friedrichs et al. [14] for the case of 1.5D-terrains. Given the set G^B as input, we can apply our algorithm as is and obtain a set G' of boundary points of size at most $|G^B|$ such that $G^B \cup G'$ guards P . Thus, we have $|G^B| + |G'| \leq 2|G^B| \leq (2 + \varepsilon)OPT^B$.

Corollary 12. *Given a WV-polygon P (w.r.t to edge \overline{uv}) with n vertices and $\varepsilon > 0$, one can find in $O(n^{\max\{2, O(1/\varepsilon^2)\}})$ time a set G of points on $(\partial P \setminus \overline{uv}) \cup \{u, v\}$, such that G guards P (boundary plus interior) and G is of size at most $(2 + \varepsilon)OPT^B$, where OPT^B is the size of a minimum-cardinality such set.*

4 Polygons weakly visible from a chord

A *chord* in a simple polygon P is a line segment whose endpoints are on the boundary of P and whose interior is contained in the interior of P . In particular, any diagonal of P is a chord in P .

In this section, we show that our method can be extended to the case where P is weakly visible from a chord \overline{uv} (in P), i.e., every point in P is visible from some point on \overline{uv} . More precisely, we show that given a set G of vertices that guards the boundary of such a polygon P , one can find a set G' of size at most $2|G|$ such that $I = G \cup G'$ guards P (boundary plus interior). Thus, given a c -approximation algorithm for vertex guarding the boundary of a polygon P weakly visible from a chord, we provide a $3c$ -approximation algorithm for guarding P . The only known (though yet unpublished) such c -approximation algorithm is that of Bhattacharya et al. [6] (for simple polygons), and there $c = 18$.

The chord \overline{uv} slices P into two (sub)polygons, where each of them is a WV-polygon w.r.t. the edge uv (see Figure 13). Thus, we would like to apply Algorithm 1 to each of these polygons separately. We assume, w.l.o.g., that \overline{uv} is horizontal, and refer to the (sub)polygons above and below \overline{uv} as P_1 and P_2 , respectively.

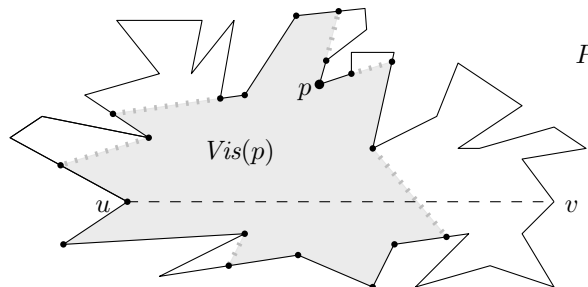


Figure 13: The visibility polygon of a point $p \in P$.

The definitions of visibility polygons, pockets, and holes, are still appropriate for the case of polygons weakly visible from a chord. However, we need to prove a few simple claims, before applying Algorithm 1 to each of the two (sub)polygons separately. The proof of the next claim is similar to that of Claim 1.

Claim 13. *For any point $p \in P$, $Vis(p) \cap \overline{uv}$ is connected.*

So, by Claim 13, $Vis(p)$ has at most two constructed edges that cross \overline{uv} .

Claim 14. *For any point $p \in P$, let \overline{xy} be a constructed edge of $Vis(p)$, such that x is below y and \overline{xy} is strictly above \overline{uv} , then C_{xy} lies strictly above ℓ_x , i.e., every point in C_{xy} is above ℓ_x .*

Proof. As in the proof of [Claim 3](#), we first observe that $C_{xy} \cap \overline{uv} = \emptyset$, since, otherwise, $\overline{uv} \subseteq C_{xy}$ and p is not visible from \overline{uv} .

Now, again as in the proof of [Claim 3](#), let z be any point in C_{xy} and let z' be a point on \overline{uv} such that $\overline{zz'} \subseteq P$. The segment $\overline{zz'}$ has to cross \overline{xy} , because $z \in C_{xy}$ and $z' \notin C_{xy}$. Since x is below y , this crossing point is above x , which in turn is above z' , so we get that z is above x . We conclude that C_{xy} lies strictly above ℓ_x . \square

Claim 15. *Let \overline{xy} be a constructed edge of $\text{Vis}(p)$, such that x is below y and \overline{xy} crosses \overline{uv} , then ℓ_{uv} intersects ∂C_{xy} at a single point which is either u or v .*

Proof. \overline{uv} intersects ∂C_{xy} at a single point which is either u or v , since $\partial C_{xy} \subseteq \partial P$ and \overline{uv} is a chord. Moreover, if ℓ_{uv} intersects ∂P at a point different than u, v , then this point is not visible from \overline{uv} . \square

As in the previous section, let E be the set of the constructed edges of the visibility polygons of the vertices in G . We run [Algorithm 1](#) on P_1 and P_2 separately, to find a set of vertices G_1 (of P_1) such that $G \cup G_1$ guards P_1 (boundary plus interior), and a set of vertices G_2 (of P_2) such that $G \cup G_2$ guards P_2 . We then set $G' = G_1 \cup G_2$ and show that $|G'| \leq 2|G|$. Thus $G \cup G'$ is a set of vertices of size at most $3|G|$ that guards P .

When running the algorithm on P_1 , we ignore the existence of P_2 . That is, if an edge $\overline{xy} \in E$ (associated with a vertex of G from either P_1 or P_2) crosses \overline{uv} , then we only consider its part that is contained in P_1 (and the part of its associated pocket that is contained in P_1). Similarly, if a hole H lies on both sides of \overline{uv} , we only consider its part that is contained in P_1 . Moreover, in this case we get a hole in P_1 that has an edge which is not a constructed edge of a visibility polygon. But this does not cause a problem, since such holes must also have upper edges, and furthermore, they must have an upper edge with endpoint on \overline{uv} .

Notice that the role of an edge $\overline{xy} \in E$ that crosses \overline{uv} switches from an upper edge in P_1 to a lower edge in P_2 , or vice versa. So, each such edge is an upper edge in either P_1 or P_2 but not in both, and therefore $|G'| \leq |G_1| + |G_2| \leq 2|G|$. Finally, notice that after computing E , we do not use the set G in our algorithm and proofs; it only reappears as one of the two sets (i.e., G and G') whose union is the final guarding set.

Theorem 16. *Given a polygon P with n vertices that is weakly visible from a chord, and a set G of vertices of P (or a set G of points on ∂P) that guards ∂P , one can find in polynomial time a set G' of size at most $2|G|$ such that $G \cup G'$ guards P (boundary plus interior). If G is a c -approximation for guarding ∂P , then $G \cup G'$ is a $3c$ -approximation for guarding the entire polygon P .*

References

- [1] M. Abrahamsen, A. Adamaszek, and T. Miltzow. The art gallery problem is $\exists \mathbb{R}$ -complete. In *Proc. 50th ACM SIGACT Sympos. on Theory of Computing, STOC*, pages 65–73, 2018.

- [2] A. Aggarwal. *The art gallery theorem: its variations, applications and algorithmic aspects*. PhD thesis, Johns Hopkins Univ., 1984.
- [3] S. Ashur, O. Filtser, and M. J. Katz. A constant-factor approximation algorithm for vertex guarding a wv-polygon. In *Proc. 18th Workshop on Approximation and Online Algorithms, WAOA*, 2020.
- [4] S. Ashur, O. Filtser, M. J. Katz, and R. Saban. Terrain-like graphs: PTASs for guarding weakly-visible polygons and terrains. *Comput. Geom.*, 101:101832, 2022.
- [5] D. Avis and G. T. Toussaint. An optimal algorithm for determining the visibility of a polygon from an edge. *IEEE Trans. Computers*, 30(12):910–914, 1981.
- [6] P. Bhattacharya, S. K. Ghosh, and S. Pal. Constant approximation algorithms for guarding simple polygons using vertex guards. 2017. [arXiv:1712.05492](https://arxiv.org/abs/1712.05492).
- [7] P. Bhattacharya, S. K. Ghosh, and B. Roy. Approximability of guarding weak visibility polygons. *Discrete Applied Mathematics*, 228:109–129, 2017.
- [8] P. Bhattacharya, S. K. Ghosh, and B. Roy. Personal communication, 2018.
- [9] É. Bonnet and T. Miltzow. An approximation algorithm for the art gallery problem. In *Proc. 33rd Internat. Sympos. on Computational Geometry, SoCG*, pages 20:1–20:15, 2017.
- [10] G. Das, P. J. Heffernan, and G. Narasimhan. Finding all weakly-visible chords of a polygon in linear time. *Nord. J. Comput.*, 1(4):433–457, 1994.
- [11] A. Deshpande, T. Kim, E. D. Demaine, and S. E. Sarma. A pseudopolynomial time $O(\log n)$ -approximation algorithm for art gallery problems. In *Algorithms and Data Structures, 10th Internat. Workshop, WADS*, pages 163–174, 2007.
- [12] A. Efrat and S. Har-Peled. Guarding galleries and terrains. *Inf. Process. Lett.*, 100(6):238–245, 2006.
- [13] S. Eidenbenz, C. Stamm, and P. Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):79–113, 2001.
- [14] S. Friedrichs, M. Hemmer, J. King, and C. Schmidt. The continuous 1.5D terrain guarding problem: Discretization, optimal solutions, and PTAS. *JoCG*, 7(1):256–284, 2016.
- [15] S. K. Ghosh. Approximation algorithms for art gallery problems. In *Proc. of the Canadian Information Processing Society Congress*, pages 429–434, 1987.
- [16] S. K. Ghosh. Approximation algorithms for art gallery problems in polygons. *Discrete Applied Mathematics*, 158(6):718–722, 2010.
- [17] S. K. Ghosh, A. Maheshwari, S. P. Pal, S. Saluja, and C. E. Veni Madhavan. Characterizing and recognizing weak visibility polygons. *Comput. Geom.*, 3:213–233, 1993.

- [18] M. Gibson, G. Kanade, E. Krohn, and K. Varadarajan. Guarding terrains via local search. *JoCG*, 5(1):168–178, 2014.
- [19] B. Joe and R. B. Simpson. Corrections to Lee’s visibility polygon algorithm. *BIT*, 27(4):458–473, 1987.
- [20] M. J. Katz. A PTAS for vertex guarding weakly-visible polygons — an extended abstract. 2018. [arXiv:1803.02160](https://arxiv.org/abs/1803.02160).
- [21] Y. Ke. Detecting the weak visibility of a simple polygon and related problems. Technical report, Johns Hopkins Univ., 1987.
- [22] J. King and D. G. Kirkpatrick. Improved approximation for guarding simple galleries from the perimeter. *Discret. Comput. Geom.*, 46(2):252–269, 2011.
- [23] David G. Kirkpatrick. An $o(\lg \lg \text{opt})$ -approximation algorithm for multi-guarding galleries. *Discret. Comput. Geom.*, 53(2):327–343, 2015.
- [24] E. Krohn and B. J. Nilsson. Approximate guarding of monotone and rectilinear polygons. *Algorithmica*, 66(3):564–594, 2013.
- [25] D. T. Lee. Visibility of a simple polygon. *Computer Vision, Graphics, and Image Processing*, 22(2):207–221, 1983.
- [26] D. T. Lee and A. K. Lin. Computational complexity of art gallery problems. *IEEE Trans. Information Theory*, 32:276–282, 03 1986.
- [27] N. H. Mustafa and S. Ray. PTAS for geometric hitting set problems via local search. In *Proc. 25th Sympos. on Computational Geometry, SoCG*, pages 17–22, 2009.
- [28] J. O’Rourke. *Art gallery theorems and algorithms*. Oxford Univ. Press, 1987.
- [29] J. O’Rourke and K. J. Supowit. Some NP-hard polygon decomposition problems. *IEEE Trans. Information Theory*, 29(2):181–189, 1983.
- [30] J.-R. Sack and S. Suri. An optimal algorithm for detecting weak visibility of a polygon. *IEEE Trans. Computers*, 39(10):1213–1219, 1990.