On r-Guarding Thin Orthogonal Polygons*

Therese Biedl [1] and Saeed Mehrabi[2]

1 **Cheriton School of Computer Science, University of Waterloo, Canada**
biedl@cs.uwaterloo.ca
2 **Cheriton School of Computer Science, University of Waterloo, Canada**
smehrabi@uwaterloo.ca

Abstract

Guarding a polygon with few guards is an old and well-studied problem in computational geometry. Here we consider the following variant: We assume that the polygon is orthogonal and *thin* in some sense, and we consider a point p to guard a point q if and only if the minimum axis-aligned rectangle spanned by p and q is inside the polygon.

A simple proof shows that this problem is NP-hard on orthogonal polygons with holes, even if the polygon is thin. If there are no holes, then a thin polygon becomes a *tree* polygon in the sense that the so-called dual graph of the polygon is a tree. It was known that finding the minimum set of r-guards is polynomial for tree polygons (and in fact for all orthogonal polygons), but the run-time was $\tilde{O}(n^{17})$. We show here that with a different approach one can find the minimum set of r-guards can be found in tree polygons in linear time, answering a question posed by Biedl et al. (SoCG 2011). Furthermore, the approach is much more general, allowing to specify subsets of points to guard and guards to use, and it generalizes to polygons with h holes or thickness K, becoming fixed-parameter tractable in $h + K$.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Art Gallery Problem, Orthogonal Polygons, r-Guarding, Treewidth, Fixed-parameter Tractable

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2016.17

1 Introduction

The art gallery problem is one of the oldest problems studied in computational geometry. In the standard art gallery, introduced by Klee in 1973 [21], the objective is to observe a simple polygon P in the plane with the minimum number of point guards, where a point $p \in P$ is seen by a guard if the line segment connecting p to the guard lies entirely inside the polygon. Chvátal [4] proved that $\lfloor n/3 \rfloor$ point guards are always sufficient and sometimes necessary to guard a simple polygon with n vertices. The art gallery problem is known to be NP-hard on arbitrary polygons [18] and orthogonal polygons [24]. Even severely restricting the shape of the polygon does not help: the problem remains NP-hard for simple monotone polygons [17] and for orthogonal tree polygons (defined precisely below) if guards must be at vertices [26]. Further, the art gallery problem is APX-hard on simple polygons [9], but some approximation algorithms have been developed [12, 17]. A number of other types of guards have been studied, especially for orthogonal polygons. See for example guarding with sliding cameras [15, 8], guarding with rectangles [10] or with orthogonally convex polygons [20]. Also, different types of visibility have been studied, especially for orthogonal polygons: guards

* This work was supported by Natural Sciences and Engineering Research Council of Canada (NSERC).

27th International Symposium on Algorithms and Computation (ISAAC 2016).
Editor: Seok-Hee Hong; Article No. 17; pp. 17:1–17:13
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

could be only seeing along horizontal or vertical lines inside P, or along an orthogonal staircase path inside P [20], or use r-visibility (defined below).

Definitions and Model. Let P be an orthogonal polygon with n vertices. The *pixelation* of P (also called *dent diagram* [6] and related to a *rectangleomino* [1]) is the partition of P obtained by extending a horizontal and a vertical ray inward at every reflex vertex, and expand it until it hits the boundary. Let Ψ be the resulting set of rectangles that we call *pixels* (also called *basic regions* [27]). See Figure 1 for an example. Note that $|\Psi|$ could be quadratic in general. We will sometimes interpret the pixelation as a planar graph, with one vertex at every corner of a pixel and an edge for each side of a pixel. Define the *dual graph* D of a polygon P to be the weak dual graph of the pixelation of P, i.e., D has a vertex for every pixel and two pixels are adjacent in D if and only if they have a common side.

An orthogonal polygon P is called a *thin polygon* if any pixel-corner lies on the boundary of P. It is called a *tree polygon* if its dual graph is a tree. One can easily see that a tree polygon is the same as a thin polygon that has no holes (see also Lemma 9). For most of this paper, polygons are assumed to be thin polygons.

We say that point g *r-guards* a point p if the minimum axis-aligned rectangle $R(g, p)$ containing g and p is a subset of P. The (standard) *rGuarding problem* hence consists of finding a minimum set S of points such that any point in P is r-guarded by a point in S. However, our results work for a broader problem as follows. Let $U \subseteq P$ be the region that we wish to guard. In particular, we could choose to guard only the vertices of P, or only the boundary, or only those parts of the art gallery that truly need to be watched. Let Γ be the set of guards that are allowed to be used (in particular, we could choose to use only vertices as guards). In the standard problem, Γ is the set of all points in P. Biedl et al. [1] introduced *pixel-guards*, where one guard consists of all the points that belong to one pixel (see Figure 1). Our approach allows pixel-guards, so $\Gamma \subset P \cup \Psi$. Now the *$(U, \Gamma, P)$-rGuarding problem* consists of finding a minimum set S of guards in Γ such that all of U is r-guarded by some guard in S (or to report that no such set exists).

Restricting the region that needs to be guarded exacerbates some degeneracy-issues for r-guarding. Previous papers were silent about what happens if rectangle $R(g, p)$ (in the definition of r-guarding) is a line segment. For example, in Figure 1, does g guard u_4? Does u_1 guard u_4? This issue can be avoided by assuming that only the interior of pixels must be guarded (as seems to have been done by Keil and Worman [27], e.g. their Lemma 1 is false for point u_4 located in the pixel ψ_{10} in Figure 1, because u_4 sees $q \in P$ but not all points in ψ_{10} do). When the entire polygon needs to be guarded, then this is a reasonable restriction since the guards that see the interior also see the boundary in the limit. But if only a subset of P must be guarded, then we must clarify how degeneracies are to be handled. We say that an axis-aligned rectangle R is *degenerate* if it has area 0 (i.e., is a line segment) and there exists no rectangle R' with positive area and $R \subset R' \subseteq P$. In Figure 1, $R(g, u_4)$ is degenerate while $R(u_1, u_4)$ is not. Our approach is broad enough that it can handle both allowing and disallowing the use of degenerate rectangles when defining r-guarding.

Related Results. The problem of guarding orthogonal polygons using r-guards was introduced by Keil [16] in 1986. He gave an $O(n^2)$-time exact algorithm for the rGuarding problem for horizontally convex orthogonal polygons. The complexity of rGuarding in simple orthogonal polygons was a long-standing open problem until 2007 when Worman and Keil [27] gave a polynomial-time algorithm for it. However, the algorithm by Worman and Keil is quite slow: it runs in $\tilde{O}(n^{17})$-time, where n denotes the number of the vertices of P and $\tilde{O}$

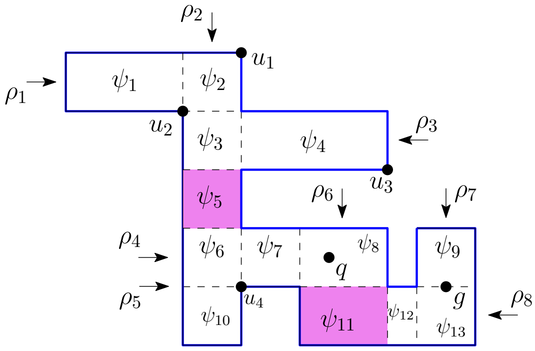

Figure 1 A tree polygon with pixels $\{\psi_1, \ldots, \psi_{13}\}$ and maximal axis-aligned rectangles $\{\rho_1, \ldots, \rho_8\}$; rectangle ρ_5 is degenerate. Pixel-guard ψ_5 guards u_3 via its top-right corner.

hides a poly-logarithmic factor. As such, Lingas et al. [19] gave a linear-time 3-approximation algorithm for rGuarding in simple polygons. Faster exact algorithms are known for a number of special cases of orthogonal polygons [16, 6, 22]. All these algorithms require the polygon to be simple. We are not aware of any results concerning the rGuarding problem for polygons with holes, or if only the vertices or only the boundary need to be guarded or used as guards.

The first results on guarding thin polygons were (to our knowledge) in [1]; they studied guarding pixelations and asked whether this can be done more easily if the dual graph is a tree. However, no better results than applying [27] were found. Later, Tomas [26] showed that indeed guarding tree polygons[1] is NP-hard in the traditional guarding-model (i.e. g guards p if the line segment gp is in P), and if all guards must be at vertices. The complexity of guarding thin polygons in the r-guarding model remained open. Paper [1] was also (apparently) the first paper to consider pixel-guards in place of point-guards.

Our Results. In this paper, we resolve the complexity of the rGuarding problem on thin polygons. We show with a simple reduction from Vertex Cover in planar graphs that this problem is NP-hard on polygons with holes, even if the polygon is thin. As our main result, we show that the rGuarding problem is linear-time solvable on thin polygons without holes.

Comparing our results to the one by Worman and Keil [27], their algorithm works for a broader class of polygons (they do not require thinness), but is slower. Moreover, their approach crucially needs that the polygon is simple, that the entire polygon needs to be guarded, and that any point in the polygon can guard. In contrast to this, our approach generalizes easily to a number of other scenarios. First of all, it is not crucial that the polygon is simple; we can deal with any constant number h of holes. Secondly, we can choose what has to be guarded and what to guard with; we can hence also solve all art gallery variants where only the vertices or only the boundary need to be guarded, or where only guards at the vertices or the boundary are allowed to be used. Finally, the restriction on thinness can be relaxed. We use thinness only to bound the treewidth of the dual graph of the polygon, and as long as the treewidth is bounded the approach works. In particular, if the polygon is K-thin in some sense, and has at most h holes, then for constants h and K our algorithm is still linear, and the rGuarding problem hence is fixed-parameter tractable in $h + K$.

[1] Tomas constructs only simple polygons and hence used the term "thin polygon" for tree polygons.

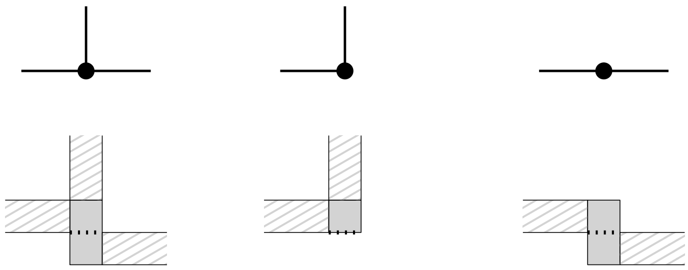

Figure 2 Converting an orthogonal drawing without bends into a polygon for rGuarding. R_e is hatched, R_v is gray, and s_v is dotted.

2 NP-hardness

In this section, we prove that rGuarding is NP-hard in polygons with holes. The reduction is from Vertex Cover in planar graphs with maximum degree 3; it is well-known that this is NP-hard [11]. So let $G = (V, E)$ be a planar graph with maximum degree 3. Let G^s be the graph obtained from G by subdividing every edge twice. It is folklore (see e.g. [23]) that G has a vertex cover of size k if and only if G^s has a vertex cover of size $|E| + k$. G has a planar orthogonal drawing with at most one bend per edge (see e.g. [14]). By placing one subdivision vertex of each edge at such a bend (if any) and placing the other subdivision vertex arbitrarily, we hence obtain a drawing Γ of G^s where every vertex is a point, every edge is a horizontal or vertical line segment, and edges are disjoint except at common endpoints.

We construct a polygon P as a "thickened" version of Γ. After possible scaling, we may assume that Γ resides in an integer grid with consecutive grid-lines at least $2n$ units apart, where $n = |V|$. Replace each horizontal edge e by a rectangle R_e of unit height, spanning between the points corresponding to the ends of e. Similarly replace each vertical edge by a rectangle of unit width. These rectangles will get moved later, but never so far that they would overlap edge-rectangles from other rows or columns.

We replace vertex-points by small gadgets as illustrated in Figure 2. Thus, let v be a vertex of degree 3 in G^s; up to rotation it has incident edges e_1, e_2, e_3 on the left, right and top in Γ. Replace v by two adjacent pixels, one above the other; we denote the resulting gadget by R_v. Then, attach R_{e_3} at the top of the upper pixel, R_{e_1} at the left side of the upper pixel and R_{e_2} at the right side of the lower pixel. Let s_v be the side common to the two pixels of R_v. Rectangles R_{e_1} and R_{e_2} are not quite horizontally aligned, resulting in one of them being offset from the grid-line. However, in total over all vertices in the row, there are at most n offsets, and so edge-rectangles remain disjoint. For any vertex of degree 2, omit the third rectangle and also any pixel that is not needed.

▶ **Observation 1.** *For any vertex v, any point in s_v guards the rectangles R_e of any incident edge $e = (v, w)$, as well as the pixel of w where R_e attaches. Moreover, for any edge $e = (v, w)$, if any point in R_e is r-guarded from a point q, then q belongs to R_e, R_v or R_w.*

Using this observation, the reduction is immediate. Namely, let C be a vertex cover of G_s of size k. For any $v \in C$, place a guard anywhere along s_v. Since C was a vertex cover, this r-guards R_e for all edges, and also R_w for all $w \notin C$ since each pixel of R_w is attached to some R_e. Vice versa, if we have a set S of r-guards, then we can create a set C as follows: For any vertex v, if R_v contains a guard in S, then add v to C. For any edge $e = (v, w)$, if R_e contains a guard in S that is in neither R_v nor R_w, then arbitrarily add one of v, w to C. Clearly $|C| \leq |S|$, and since any rectangle R_e was guarded, any edge in E is covered by C.

Inspection of Figure 2 shows that the constructed polygon is thin. Observe that it has holes, namely, one per face of G. Since rGuarding is clearly in NP, we can conclude:

▶ **Theorem 2.** *The rGuarding problem is* NP*-complete on thin polygons.*

3 Polygons Whose Dual Has Bounded Treewidth

We now show how to solve the rGuarding problem in a tree polygon in linear time. In fact, we show something stronger, and prove that the rGuarding problem can be solved in linear time in any polygon for which the dual graph D has bounded treewidth, and under any restriction on the set U to be guarded and the set Γ that may serve as guards.

The approach is to construct an auxiliary graph H, and argue that solving the rGuarding problem reduces to a graph problem in H. Then we argue that the treewidth of H satisfies $tw(H) \in O(tw(D))$ and that the graph problem is linear-time solvable in bounded treewidth graphs. This auxiliary graph is different from the so-called region-visibility-graph used by Worman and Keil [27] in that it encodes who can guard what, rather than who can be guarded by a common guard.

3.1 Simplifying U and Γ

We first show that we can simplify the points to guard and the point-guards to use such that only a constant number of each occur at each pixel.

▶ **Lemma 3.** *Let $U \subseteq P$ be any (possibly infinite) set of points in P. Then there exists a finite set of points $U' \subseteq U$ such that U' is r-guarded by a set S if and only if U is. Moreover, for any pixel ψ, at most 4 points in U' belong to ψ.*

Proof. We construct the set U' as follows.

- For every pixel ψ, if the interior of ψ intersects U, then add one point from this intersection into U'.
- For every pixel-side e, if neither incident pixel has a point of U in its interior, but the open set e intersects U, then add one point from this intersection to U',
- For every pixel-corner c, if $c \in U$, and if none of the incident pixels or pixel-sides has added a point to U', then add c to U'.

Correctness can be shown easily (see the full version [2]), by arguing that any two points in the strict interior of one pixel ψ are guarded by the same set of guards, and similarly for points in the relative interior of a side of a pixel. ◀

▶ **Lemma 4.** *Let $\Gamma \subseteq P$ be any (possibly infinite) set of points in P. Then there exists a finite set of points $\Gamma' \subseteq \Gamma$ such that for any pixel ψ, at most 4 points in Γ' belong to ψ. Moreover, if some set $S \subseteq \Gamma$ r-guards a set $U \subseteq P$, then there exists a set $S' \subseteq \Gamma'$ with $|S'| \leq |S|$ that also r-guards U.*

Proof. We construct the set Γ' follows:

- For every pixel-corner c, if $c \in \Gamma$ then add c to Γ'.
- For every pixel-side e, if neither endpoint of e is in Γ, but some interior point of e is in Γ, then add one such point to Γ'.
- Finally, for every pixel ψ, if no corner is in Γ and no side has a point in Γ, but the interior of ψ contains points in Γ, then add one such point to Γ'.

Correctness can be shown easily (see the full version [2]). ◀

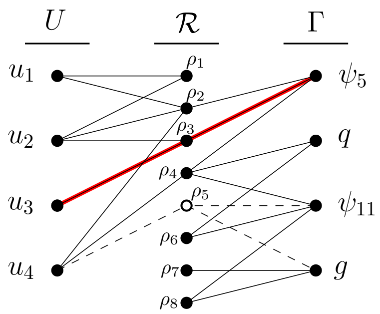

Figure 3 The graph H corresponding to Figure 1 for the chosen U and Γ. The thick red path corresponds to the pixel-guard ψ_5 seeing the point u_3 since both intersect rectangle ρ_3. Rectangle ρ_5 and its incident edges are included in H only if we allow degenerate rectangles.

3.2 Maximal Rectangles and an Auxiliary Graph

Assume we are given a polygon P, a region $U \subseteq P$ to be guarded, and a set Γ of guards allowed to be used. In what follows, we treat any element $\gamma \in \Gamma$ as a set, so either $\gamma = \psi$ is a pixel-guard or $\gamma = \{p\}$ is a point-guard.

As a first step, apply Lemmas 3 and 4 to reduce U and the point-guards in Γ so that they are finite sets, each pixel contains at most 4 points of U, and at most 4 point-guards of Γ.

Let $\mathcal{R}$ be the set of maximal axis-aligned rectangles in P, i.e., $\rho \in \mathcal{R}$ if and only if $\rho \subseteq P$ and there is no axis-aligned rectangle ρ' with $\rho \subset \rho' \subseteq P$. In this definition of $\mathcal{R}$, we use the one that was meant for r-guarding, i.e., we include degenerate rectangles in $\mathcal{R}$ if and only if a degenerate rectangle $R(g, p)$ is sufficient for g to r-guard p. Now define graph H as follows. The vertices of H are $U \cup \mathcal{R} \cup \Gamma$, i.e., we have one vertex for every point that needs guarding, one for every maximal rectangle in P, and one for every potential guard. We define edges of H via containment as follows (see also Figure 3).

(i) There is an edge from a point $u \in U$ to a rectangle $\rho \in \mathcal{R}$ if and only if $u \in \rho$.
(ii) There is an edge from a potential guard $\gamma \in \Gamma$ to a rectangle $\rho \in \mathcal{R}$ if and only if their intersection is non-empty.

▶ **Lemma 5.** *A point $u \in U$ is r-guarded by $\gamma \in \Gamma$ if and only if there exists a path of length 2 from u to γ in H.*

Proof. If u is r-guarded by γ, then there exists some $g \in \gamma$ such that the axis-aligned rectangle R spanned by p and g is inside P. Expand R until it is maximal to obtain $\rho \in \mathcal{R}$. More precisely, if R is non-degenerate, then use as ρ some maximal rectangle that has non-zero area and contains R. If R is degenerate, then obviously degenerate rectangles were allowed for r-guarding, and so expanding R into a maximal line segment within P gives an element ρ of $\mathcal{R}$. Either way $u \in R \subseteq \rho$ and $g \in R \subseteq \rho$ and we have a path $u - \rho - g$ in H.

Vice versa, if there exists such a path, then it must have the form $u - \rho - \gamma$ for some maximal rectangle ρ by construction of H. By definition of the edges, $u \in \rho$ and some point $g \in \gamma$ satisfies $g \in \rho$, which means that the axis-aligned rectangle spanned by u and g is inside $\rho \subseteq P$ and so g (and with it γ) guards u. ◀

So, the rGuarding problem reduces to finding the minimum subset $S \subseteq \Gamma$ such that all $u \in U$ have a path of length 2 to some $\gamma \in S$, or reporting that no such S exists. We call this the *restricted distance-2-dominating set* since this is the distance-2-dominating set [25] with restrictions on who can be chosen and who must be dominated. Therefore, we have:

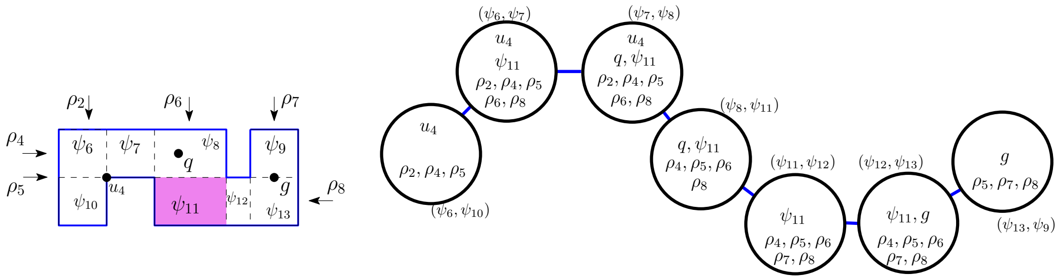

Figure 4 The tree decomposition $\mathcal{T}^H = (I, \mathcal{X}^H)$ of graph H corresponding to a sub-polygon of the one in Figure 1. We label the bags with the edges of the tree they correspond to.

▶ **Lemma 6.** *The (U, Γ, P)-rGuarding problem has a solution of size k if and only if the restricted distance-2-dominating set in H has a solution of size k.*

3.3 Constructing a Tree Decomposition

Recall graph D, the weak dual graph of the pixelation of polygon P. Assume now that the dual graph D has small treewidth, defined as follows. A *tree decomposition* of a graph D consists of a tree I and an assignment $\mathcal{X} : I \to 2^{V(D)}$ of *bags* to the nodes of I such that (a) for any vertex v of D, the bags containing v form a connected subtree of I and (b) for any edge (v, w) of D, some bag contains both v and w. The width of such a decomposition is $\max_{X \in \mathcal{X}} |X| - 1$, and the *treewidth* $tw(D)$ of D is the minimum width over all tree decompositions of D.

Fix a tree decomposition $\mathcal{T} = (I, \mathcal{X})$ of D that has width $tw(D)$. We now construct a tree decomposition of H from $\mathcal{T}$ while increasing the bag-size by a constant factor. Any bag $X \in \mathcal{X}$ consists of vertices of D, i.e., pixels of P. To obtain $\mathcal{T}' = (I, \mathcal{X}')$, modify any bag $X \in \mathcal{X}$ to get X' as follows: For any pixel $\psi \in X$, add to X'

- any point of U that is in ψ,
- any guard of Γ that intersects ψ, and
- any rectangle in $\mathcal{R}$ that intersects ψ.

Finally we may (optionally) delete all pixels from all bags, since these are not vertices of H. We call the final construction $\mathcal{T}^H = (I, \mathcal{X}^H)$. See also Figure 4.

▶ **Lemma 7.** *For any polygon, $\mathcal{T}^H = (I, \mathcal{X}^H)$ is a tree decomposition of H. If P is thin, then the tree decomposition has width $O(tw(D))$.*

Proof. First we argue that for any vertex of H the bags containing it are connected. Crucial for this is that for any pixel ψ, the bags that used to contain ψ in $\mathcal{T}$ are a connected subtree since $\mathcal{T}$ was a tree decomposition. First consider a point p. (We use p for both the point and for the vertex in H representing it.) Vertex p was added to all bags that contained a pixel ψ with $p \in \psi$. There may be multiple such pixels (if p is on the side or the corner of a pixel), but the union of them is a connected subgraph of D. For any connected subgraph, the bags containing vertices of it form a connected subtree. So the bags to which p has been added form a connected subtree of the tree I of the tree decomposition as required.

The connectivity-argument is identical for a point-guard, and similar for pixel-guards and rectangles. Namely, consider a vertex of H representing a pixel-guard γ. This guard was added to all the bags that contained a pixel ψ that intersects γ. Again there may be many such pixels (up to 9), but they are connected via ψ and so the bags to which γ is added are connected. Finally, consider a rectangle $\rho \in \mathcal{R}$ which was added to all bags of pixels intersecting ρ. The pixels that ρ intersects form a connected subset of P (because they are connected along ρ), and hence correspond to a connected subgraph of D. So the bags containing ρ form a connected subtree. Now we must verify that for any edge of H, both endpoints appear in a bag. Let (u, ρ) be an edge from some point u to some rectangle ρ. Let ψ be a pixel containing u. Then $\rho \cap \psi \supseteq \{u\}$ is non-empty and so ρ was added to any bag containing ψ. We also added u to any bag containing ψ, so u and ρ appear in one bag. Now consider some edge (γ, ρ) from a guard γ to some rectangle ρ. This edge exists because some point $g \in \gamma$ belongs to ρ. Again fix some pixel ψ that contains g and observe that any bag that contained ψ has both g and ρ added to it.

It remains to discuss the width of the tree decomposition. Consider a bag X of $\mathcal{T}$ and one pixel ψ in X. Since we reduced U and Γ with Lemma 3 and 4, pixel ψ intersects at most 4 points in U and at most 4 point-guards. It also intersects at most 9 pixel-guards. Finally, one can show that in a thin polygon ψ intersects at most 6 maximal rectangles. (A more general statement will be proved in Lemma 14.) Thus when creating bag X' from bag X we add $O(1)$ new items per pixel and hence $|X'| \in O(|X|)$ and $\mathcal{T}^H$ has width $O(tw(D))$. ◀

3.4 Solving 2-dominating Set

To solve the restricted distance-2-dominating set problem on H, we first show that the problem can be expressed as a monadic second-order logic formula [5]. In particular, a set S is a feasible solution for this problem if and only if

$$S \subseteq \Gamma \quad \wedge \quad \forall u \in U\ \exists \rho \in \mathcal{R}\ \exists \gamma \in S:\ \mathrm{adj}(u, \rho) \wedge \mathrm{adj}(\rho, \gamma)$$

where adj is a logic formula to encode that its two parameters are adjacent in H. Since H has bounded treewidth, we can find the smallest set S that satisfies this or report that no such S exists in linear time using Courcelle's theorem [5]. Here "linear" refers to the number of bags and hides a term that only depends on the treewidth. One can show that a thin polygon has $O(n)$ pixels (we will show something more general in Lemma 13). Therefore graph D has $O(n)$ vertices and hence a tree decomposition with $O(n)$ bags. In consequence the run-time is hence $O(f(tw(D))n)$ for some computable function f.

3.5 Run-time considerations

We briefly discuss here how to do all other steps in linear time, under some reasonable assumptions. The first step is to find the pixels. To do so, we need to compute the vertical decomposition (i.e., the partition obtained by extending only vertical rays from reflex vertices), which can be done in $O(n)$ time [3]. Likewise, compute the horizontal decomposition. Since (in a thin polygon) none of the rays intersect, we can obtain the pixels (and with it, the pixelation-graph and D) in linear time. Since D is planar, we can compute an $O(1)$-approximation of its treewidth in linear time [13], and hence can find $\mathcal{T}$ with width $O(tw(D))$. Next we need to simplify U and Γ. The run-time to do so depends on the exact form of the original U and Γ, but as long as those have a simple enough form that we can answer queries such as "does the interior of pixel ψ intersect U" in constant time, the overall time is $O(1)$ per pixel and hence overall linear.

Next we need to find the rectangles $\mathcal{R}$. In a thin polygon, all maximal rectangles are either a "slice" defined by the vertical or horizontal decomposition, or are a maximal line segment composed of pixel sides. All such slices and maximal line segments can be found from the pixelation in linear time, and there are $O(n)$ of them. This may yield some rectangles that are not maximal, but we can retain those without harm since even then any pixel intersects $O(1)$ rectangles. Constructing H from these three sets, and building $\mathcal{T}^H$ given $\mathcal{T}$, can also clearly be done in linear time. Putting everything together, we hence have:

▶ **Theorem 8.** *Let P be a thin polygon for which the dual graph has treewidth k. Then for any set $U \subseteq P$ and $\Gamma \subseteq P \cup \Psi$, we can solve the (U, Γ, P)-rGuarding problem in time $O(f(k)n)$ time for some computable function f.*

4 Generalizations

In this section, we give some applications and generalizations of Theorem 8.

4.1 Thin Polygons with Few Holes

We claimed earlier that a simple thin polygon is a tree polygon, and give here a formal proof because it will be useful later.

▶ **Lemma 9.** *Let P be a thin polygon. If P has no holes, then the dual graph D of the pixelation of P is a tree.*

Proof. Assume for contradiction that D contains a cycle. By tracing along the midpoints of the pixels-sides corresponding to this cycle, we can create a simple closed curve C that is inside P, yet has pixel-corners both inside and outside C. In a thin polygon, all pixel-corners are on the boundary of P, so the boundary of P has points both inside and outside a simple closed curve that is strictly within P. This is possible only if P has holes. ◀

Since every tree has treewidth 1, we hence have:

▶ **Corollary 10.** *Let P be a thin polygon that has no holes. Then for any sets $U \subseteq P$ and $\Gamma \subseteq P \cup \Psi$, we can solve the (U, Γ, P)-rGuarding problem in $O(n)$ time.*

Inspecting the proof of Lemma 9, we see that in fact every cycle of D gives rise to a hole that is inside the curve defined by the cycle. If D has f inner faces, then each face defines a cycle in D, and the insides of these cycles are disjoint. Therefore, D has at least f holes. Turning things around, if the polygon has h holes, then D has at most h inner faces. In consequence, D is a so-called h-outerplanar graph (i.e., if we remove all vertices from the outer-face and repeat h times, then all vertices have been removed). It is well-known that h-outerplanar graphs have treewidth $O(h)$ (see e.g. [7]).

▶ **Corollary 11.** *Let P be a thin polygon with h holes. Then for any sets $U \subseteq P$ and $\Gamma \subseteq P \cup \Psi$, we can solve the (U, Γ, P)-rGuarding problem in time $O(f(h)n)$ time for some computable function f.*

4.2 Polygons That are not Thin

The construction of the tree decomposition of H in Section 3.3 works even if P is not thin. However, the bound on the resulting treewidth, and the claim on the linear run-time both used that the polygon is thin. We can generalize these results to polygons that are somewhat thicker. More precisely, we say that a polygon is *K-thin* (for some integer $K \geq 1$) if the dual graph D of P contains no induced $(K+1) \times (K+1)$-grid. A thin polygon is a 1-thin polygon in this terminology, because a pixel-corner is in the interior if and only if the four pixels around it form a 4-cycle, hence a 2×2-grid, in D. Notice that K-thin is equivalent to saying that the pixelation-graph has no induced $(K+2) \times (K+2)$-grid. We need some observations:

▶ **Lemma 12.** *Let P be a K-thin polygon. Then, for any pixel-corner p, there exists a point on the boundary of P that is in the first quadrant relative to p and has distance at most $2K+1$ from p, where distance is measured by the length of the path in the pixelation-graph.*

Proof. Consider any path in the pixelation graph that starts at p and goes upward or rightward for at most $K+1$ edges each. If some such path reaches a point on the boundary after at most $2K+1$ edges, then we are done. Else the union of these paths forms a $(K+2) \times (K+2)$-grid in the pixelation-graph, and P is not K-thin. ◀

▶ **Lemma 13.** *The pixelation of a K-thin polygon with n vertices has $O(K^2 n)$ pixels.*

Proof. There are $O(n)$ boundary vertices: one for each vertex of P, and one whenever a ray hits the boundary (of which there are at most $n-4$ since there are $n/2-2$ reflex vertices and each emits two rays). Each vertex on the boundary has $O(K^2)$ pixel-corners within distance $2K+1$. By the previous lemma all pixel-corners must be within such distance, so there are $O(K^2 n)$ pixel-corners, and hence $O(K^2 n)$ pixels. ◀

Since a K-thin polygon contains no $(K+2) \times (K+2)$-grid in the pixelation, one can also show the following (details are in the full version [2]):

▶ **Lemma 14.** *Any pixel ψ in a K-thin polygon P is intersected by $O(K^2)$ maximal axis-aligned rectangles inside P.*

▶ **Theorem 15.** *Let P be a K-thin simple polygon. Then for any set $U \subseteq P$ and $\Gamma \subseteq P \cup \Psi$, the (U, Γ, P)-rGuarding problem can be solved in $O(f(K^3)K^2 n)$ time for some computable function $f(.)$.*

Proof. The pixelation of P has $O(k^2 n)$ vertices by Lemma 13, and can be constructed in $O(k^2 n)$ time by constructing the vertical decomposition and then ray-shooting along the horizontal rays emitted from reflex vertices. For any pixel-corner p, there exists a point on the boundary of P that and has distance at most $2K+1$ from p. It follows that the pixelation graph is $(2K+1)$-outerplanar, and hence it (and also its dual graph D) have treewidth $O(k)$. Find a tree decomposition of D with treewidth $O(k)$ and $O(k^2 n)$ bags; this can be done in linear time since D is planar [13]. Replace each pixel in each bag of $\mathcal{T}$ by points, guards and rectangles as explained in Section 3.3. Since each pixel belongs to $O(k^2)$ rectangles, the resulting tree decomposition has width $O(k^3)$. Now solve the restricted 2-dominating set problem using Courcelle's theorem. The run-time is as desired since we have $O(k^2 n)$ bags and treewidth $O(k^3)$. ◀

4.3 K-Thin Polygons with Few Holes

Both of the above generalizations can be combined, creating an algorithm that is fixed-parameter tractable in both the thinness and the number of holes.

▶ **Lemma 16.** *Let P be a polygon that is K-thin and that has h holes. Then the dual graph of P has treewidth $O(K(h+1))$.*

Proof. Let D' be the (full) dual graph of the pixelation graph, i.e., it is graph D plus a vertex for each hole and for the outerface, connected to all incident pixels. We claim that all vertices in D' have distance $O(K(h+1))$ from the outerface-vertex. This implies that D' (and hence also D) is $O(K(h+1))$-outerplanar and so has treewidth $O(K(h+1))$.

To prove the distances, we first connect holes as follows. If H is a hole, then let c be a corner of H that maximizes the sum of the coordinates (breaking ties arbitrarily). Let ψ be a pixel incident to c and let c' be some other corner of ψ. By Lemma 12, there exists a pixel-corner p on the boundary of P within distance $2K+1$ from c'. Moreover, the path from c' to p goes only up and right. Thus p is incident to the outer-face or to a hole H', where $H' \neq H$ by choice of c. Following this path, we can hence find a path in D of length $O(K)$ from the vertex representing H to the vertex representing H' or the outer-face. Combining all these paths, we can reach the outer-face from any hole in a path of length $O(K(h+1))$.

Now for any other vertex in D (hence pixel ψ), let c be one pixel-corner, and find a path in the pixelation of length at most $2K+1$ from c to some point on the boundary. Following this path, we can find a path of length $O(K)$ in D from ψ to some hole or the outer-face, and hence reach the outer-face along a path of length $O(K(h+1))$. The result follows. ◀

The following summarizes this approach, and includes all previous results.

▶ **Theorem 17.** *Let P be a polygon that is K-thin and has h holes. Then for any set $U \subseteq P$ and $\Gamma \subseteq P \cup \Psi$, the (U, Γ, P)-rGuarding problem can be solved in $O(f((K(h+1))^3)(K(h+1))^2 n)$ time for some computable function $f(.)$. In particular, the rGuarding problem is fixed-parameter tractable in $K+h$.*

5 Conclusion

In this paper, we studied the problem of guarding a thin polygon under the model that a guard can only see a point if the entire axis-aligned rectangle spanned by them is inside the polygon. We showed that this problem is NP-hard, even in thin polygons, if there are holes. If there are few holes or, more generally, the dual graph of the polygon has bounded treewidth, then we solved the problem in linear time. Our approach is quite flexible in that we can specify which points must be guarded and which points/pixels are allowed to be used as guards. In fact, with minor modifications even more flexibility is possible. We could allow any guard that consists of a connected union of pixels (as long as any pixel is intersected by $O(1)$ guards). We could even consider other guarding models by replacing the rectangles in $\mathcal{R}$ by arbitrary connected unions of pixels and pixel-sides (again as long as any pixel is intersected by $O(1)$ such shapes). For all these, the (naturally defined) auxiliary graph H has treewidth $O(tw(D))$ in thin polygons, and we can hence solve r-guarding by solving the restricted distance-2-dominating set.

Our results mean that the complexity of r-guarding is nearly resolved, with the exception of polygons that have $O(1)$ holes but are not K-thin for a constant number K. For such polygons, is the problem still NP-hard? Also, for polygons that have a large number of holes, is the problem APX-hard, or can we develop a PTAS?

Acknowledgments. The authors would like to thank Justin Iwerks and Joe Mitchell; the discussions with them made us consider thin polygons in the first place.

References

1 T. Biedl, M. T. Irfan, J. Iwerks, J. Kim, and J. S. B. Mitchell. Guarding polyominoes. In *Proc. of the ACM Symp. on Computational Geometry (SoCG'11)*, pages 387–396, 2011.

2 T. C. Biedl and S. Mehrabi. On r-guarding thin orthogonal polygons. *CoRR*, abs/1604.07100, 2016. URL: `http://arxiv.org/abs/1604.07100`.

3 B. Chazelle. Triangulating a simple polygon in linear time. *Disc. Comp. Geom.*, 6(5):485–524, 1991.

4 V. Chvátal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory, Series B*, 18:39–41, 1975. `doi:10.1137/S0097539796302531`.

5 B. Courcelle. The monadic second-order logic of graphs I: Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.

6 J. Culberson and R. A. Reckhow. Orthogonally convex coverings of orthogonal polygons without holes. *Journal of Computer and System Sciences*, 39(2):166–204, 1989.

7 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Mark, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, Heidelberg, Germany, 2015.

8 S. Durocher and S. Mehrabi. Guarding orthogonal art galleries using sliding cameras: algorithmic and hardness results. In *Proceedings of Mathematical Foundations of Computer Science (MFCS 2013)*, volume 8087 of *LNCS*, pages 314–324, 2013.

9 S. Eidenbenz, C. Stamm, and P. Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):79–113, 2001. `doi:10.1007/s00453-001-0040-8`.

10 D. S. Franzblau and D. J. Kleitman. An algorithm for constructing regions with rectangles: Independence and minimum generating sets for collections of intervals. In *Proceedings of the ACM Symposium on Theory of Computing (STOC 1984)*, pages 167–174, 1984.

11 M. R. Garey and D. S. Johnson. The Rectilinear Steiner Tree Problem is NP-complete. *SIAM Journal of Applied Mathematics*, 32:826–834, 1977.

12 S. K. Ghosh. Approximation algorithms for art gallery problems in polygons. *Disc. App. Math.*, 158(6):718–722, 2010.

13 F. Kammer and T. Tholey. Approximate tree decompositions of planar graphs in linear time. In *Proc. of the ACM-SIAM Symp. on Discrete Algorithms (SODA 2012)*, pages 683–698, 2012.

14 G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16:4–32, 1996.

15 M. J. Katz and G. Morgenstern. Guarding orthogonal art galleries with sliding cameras. *Int. J. Comput. Geometry Appl.*, 21(2):241–250, 2011.

16 J. M. Keil. Minimally covering a horizontally convex orthogonal polygon. In *Proceedings of the ACM Symposium on Computational Geometry (SoCG 1986)*, pages 43–51, 1986.

17 E. Krohn and B. J. Nilsson. Approximate guarding of monotone and rectilinear polygons. *Algorithmica*, 66(3):564–594, 2013.

18 D. T. Lee and Arthur K. Lin. Computational complexity of art gallery problems. *IEEE Trans. on Information Theory*, 32(2):276–282, 1986. `doi:10.1109/TIT.1986.1057165`.

19 A. Lingas, A. Wasylewicz, and P. Zylinski. Linear-time 3-approximation algorithm for the r-star covering problem. *Int. J. Comput. Geometry Appl.*, 22(2):103–142, 2012.

20 R. Motwani, A. Raghunathan, and H. Saran. Perfect graphs and orthogonally convex covers. *SIAM J. Discrete Math.*, 2(3):371–392, 1989.

21 J. O'Rourke. *Art Gallery Theorems and Algorithms*. The International Series of Monographs on Computer Science. Oxford University Press, New York, NY, 1987.

22 L. Palios and P. Tzimas. Minimum r-star cover of class-3 orthogonal polygons. In *Proceedings of the International Workshop on Combinatorial Algorithms (IWOCA 2014)*, volume 8986 of *LNCS*, pages 286–297. Springer, 2015.

23 S. Poljak. A note on stable sets and colorings of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 15(2):307–309, 1974.

24 D. Schuchardt and H.-D. Hecker. Two NP-hard art-gallery problems for ortho-polygons. *Mathematical Logic Quarterly*, 41(2):261–267, 1995.

25 P. J. Slater. R-domination in graphs. *J. ACM*, 23(3):446–450, 1976.

26 A. P. Tomás. Guarding thin orthogonal polygons is hard. In *Proceedings of Fundamentals of Computation Theory (FCT 2013)*, volume 8070 of *LNCS*, pages 305–316, 2013.

27 C. Worman and J. M. Keil. Polygon decomposition and the orthogonal art gallery problem. *Int. J. Comput. Geometry Appl.*, 17(2):105–138, 2007. `doi:10.1142/S0218195907002264`.