CITY GUARDING AND PATH CHECKING: SOME STEPS TOWARDS SMART CITIES

by

Hemant Malik

APPROVED BY SUPERVISORY COMMITTEE:

Dr. Ovidiu Daescu, Chair

Dr. R. Chandrasekaran

Dr. Ding-Zhu Du

Dr. Kyle Fox

Dedicated to

my parents, brother, sister in law, wife,

and all the people who inspired and motivated to reach this point of my life.

CITY GUARDING AND PATH CHECKING: SOME STEPS TOWARDS SMART CITIES

by

HEMANT MALIK, BS, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS

August 2020

ACKNOWLEDGMENTS

The completion of my dissertation would not have been possible without the support and nurturing of my advisor Dr. Ovidiu Daescu, Professor and Assistant Head, Department of Computer Science, The University of Texas at Dallas. I am grateful for his continuous support during my PhD study. He guides me well throughout the research work from the title's selection to finding the results. His immense knowledge, motivation, and patience have given me more power and spirit to excel in research.

I would like to express my most profound appreciation to Dr. R. Chandrasekaran for giving encouragement and sharing insightful suggestions. He played a significant role in polishing my research skills. His endless guidance is hard to forget throughout my life, which provided me extensive personal and professional advice and taught me a great deal about both scientific research and life in general.

Besides these, I would like to thank the rest of my dissertation committee: Dr. Ding-Zhu Du, and Dr. Kyle Fox, for their insightful comments and encouragement. Without their precious support, it would not be possible to conduct this research.

My sincere thanks go to my fellow lab mates for the stimulating discussions. It was great sharing laboratory with all of you during the last five years.

I very much appreciate the friendships of Alisha Khoja, Rashika Mishra, Mayank Chauhan, Abhishek Mishra, Harsh Agrawal, Partha De, Riti Gour, and Sruthi Chappidi and their support over the last five years. They have directly and indirectly helped me in completing this dissertation.

Nobody has been more important to me in the pursuit of this research work than the members of my family. I would like to thank my parents, whose love and guidance are with me in whatever I pursue. I am grateful to my brother, Jayant Malik and sister-in-law, Upasana for supporting me spiritually throughout writing this thesis and my life in general. Most

importantly, I wish to thank my loving and supportive wife, Anjali, who provided unending inspiration. I owe it all to you. Many Thanks!

June 2020

CITY GUARDING AND PATH CHECKING: SOME STEPS

TOWARDS SMART CITIES

Hemant Malik, PhD
The University of Texas at Dallas, 2020

Supervising Professor: Dr. Ovidiu Daescu, Chair

With drones and other small unmanned aerial vehicles starting to get permission to fly within city limits, monitoring the aerial space of big cities is becoming a critical problem that yet has to be addressed. While video cameras are easily available in most cities, their purpose is to guard the streets at ground level. Guarding the aerial space of a city with video cameras is a problem that so far has been largely ignored even in a limited way of all three dimensions. In this dissertation, we address various issues that set a necessary foundation for drone surveillance, which are as follows:

1. City Guarding with Limited Field of View.

2. Path Checking in $\Re^2$

In the first problem, we present bounds on the number of cameras needed to guard a city's aerial space (roofs, walls, and ground) using cameras with 180° range of vision (the region in front of the guard), which is common for most commercial cameras. Each camera is placed at the top corner of a building. We considered the following cases:

1. All buildings are vertical and have a rectangular base.

2. All buildings are vertical and have an orthogonal base.

For each case, we further considered the following two sub-cases:

1. Buildings have an axis-aligned ground base and,

2. Buildings have an arbitrary orientation.

Unlike previous studies on guarding polygons with holes, a key subproblem we encounter is to guard a simple shaped polygon with holes by placing guards only at the vertices of the holes.

We further address the following path checking problem: Given a set S of m disjoint simple polygons in the plane, with a total of n vertices, preprocess them so that for a query consisting of a positive constant c and a simple polygonal path π with k vertices, from a point u to a point v in free space, where k is much smaller than n, one can quickly decide whether π has clearance at least c (that is, there is no polygonal obstacle within distance c of π). To do so, we show how to solve the following related problem: Given a set S of m simple polygons in $\Re^2$, preprocess S into a data structure so that the polygon in S closest to a query line segment s can be reported quickly.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The concept of **smart city** (air, 2020) is to build a city where all public services are interconnected with each other. The vision behind the concept focuses on reducing the cost and consumption of resources, and improving public safety. UAVs, also fondly called "drones", are going to be a significant contributors in this endeavor. Some applications using drones include carrying an object from one place to another, surveillance and security, especially for locations which are tough to monitor, smart traffic management, and pollution control.

Having an eye in the sky helps both people and police on the ground. For people, it helps to monitor the routes while commuting from one place to another, by avoiding the roads with jams. One can build a map that takes the input from cameras and helps in smart traffic management. For police, it is helpful in better monitoring of crowds or UAVs. For particular areas that humans cannot reach, drones can help in analyzing the situation correctly.

For city guarding with cameras problem, we consider that the given city is rectangular, and all the buildings have an orthogonal base. We discuss the issue of placing video cameras at the top corners of some buildings, to guard the aerial space of the city, where the range of visibility of each camera is bounded to 180°. In this version, the cameras/guards are static.

A problem that arises while using drones for surveillance or carrying an object from one place to another is path planning. The drones are assigned a specific path where they need to monitor the aerial space or need to deliver a package from a source to a destination, and the path needs to have a predefined clearance depending on the size of the drone or the delivery box. We discuss the path checking problem in the plane, $\Re^2$, where given a set of disjoint simple polygons, a simple polygonal path, and a constant c, one needs to decide if the path has a clearance of c or not. We need to extend this approach for 3D, as drones can fly at different altitudes.

In the rest of this chapter, we provide more in-depth knowledge of both problems.

1.1 City Guarding with Limited Field of View

Drones and other small unmanned aerial vehicles (UAVs) are already allowed to experimentally fly within some cities and are starting to get permission to commercially fly within city limits. For example, in August 2019, Uber announced it has selected the city of Dallas to experiment with flying drones and small UAVs, within the city limits. Monitoring the aerial space of big cities is thus becoming a critical problem that yet has to be addressed. Video cameras are easily available in most cities, but their purpose is to guard the streets at ground level. Guarding the aerial space of a city with cameras is a problem that has been largely ignored.

City guarding is related to the famous *art gallery problem* (Klee, 1969) and its many variations (O'Rourke, 1987; Urrutia, 2000, 2004) studied in the past few decades. In almost all these studies, the art gallery lies in the plane (2D), assuming a polygonal shape with or without holes. In the art gallery problem, the goal is to determine the minimum number of point guards sufficient to see every point of the interior of a simple polygon. A point q is visible to guard g if the line segment joining q and g lies completely within the polygon. When the guards are restricted to vertices of the polygon only, they are referred to as vertex guards.

In the orthogonal art gallery problem, all edges of the polygon are either horizontal or vertical. In some versions of the art gallery problem, the polygon is allowed to have h holes. When guarding such polygons, it is allowed to place the guards at the vertices of the enclosing polygon and the vertices of the holes.

For guarding an *orthogonal polyhedron*, point guards are less effective. There exist examples of polyhedra with n vertices where guards placed at every vertex do not cover the whole interior of the polyhedra; instead $O(n^{3/2})$ non-vertex guards are required to guard the whole interior. Refer to O'Rourke (O'Rourke, 1987) for more details.

The problem is also related to the following problem (Blanco et al., 1994): Given k pairwise disjoint isothetic rectangles (rectangles are *isothetic* if all their sides are parallel to the coordinate axes) in the plane, place vertex guards on rectangles such that every point in free space (plane area except the interior of quadrilaterals) is visible to at least one guard.

The city guarding problem was introduced in (Bao et al., 2008), and it is a 2.5D variant of the 2D orthogonal art gallery with holes. The input consists of k buildings, within an area bounded by an axis-parallel rectangle, with each building being vertical and having an axis parallel rectangular base (a vertical rectangular prism), and the goal is to place the minimum number of guards that can see in any direction (referred as 360° field of vision), at the top corners (vertices) of some buildings, to guard the aerial space of the city. The height of a building is a strictly positive real number. In (Bao et al., 2008), they consider three variations of city guarding: (i) *Roof Guarding:* determine the minimum number of vertex guards required to guard the roofs (ii) *Ground and Wall Guarding:* determine the minimum number of vertex guards necessary to guard the ground and the walls, and (iii) *City Guarding:* determine the minimum number of vertex guards required to guard the (aerial space of the) city, which means the roofs, walls, and the ground. As with the 2D art gallery problem, the 2.5D city guarding problems are NP-hard and, by a simple reduction, so are the corresponding versions studied in this dissertation.

We consider the three variations of the city guarding problem with a restriction on the visibility range of the guards. Specifically, a guard is only able to see the region in front of it, i.e., the range of vision of a guard is bounded by 180°, instead of the 360° in (Bao et al., 2008). This corresponds to the capabilities of most existing commercial cameras. The guards are placed at the top corners of a building. In our constructions and proofs, the seen and unseen regions of a guard are separated by a vertical plane parallel with one of the sides of the building where the guard is placed.

1.2 Path Checking in $\Re^2$

Path planning problems among polygonal obstacles in the plane usually ask to find a path that avoids the obstacles and is optimal with respect to some measure or a combination of measures, for example, a shortest u-to-v path (Storer and Reif, 1994; Mitchell, 2000, 1996) or an u-to-v shortest path of clearance at least c (Wein et al., 2008; Weina et al., 2007), where u and v are points in the free space, and c is a positive constant. In some practical applications, however, such as emergency interventions/evacuations and medical treatment planning, a number of u-to-v (polygonal or circular arc) paths are suggested by experts and the question is whether such paths satisfy specific requirements, such as a given clearance from the polygonal obstacles. We address the following path query problem:

Path-Obstacles Proximity Queries: Given a set S of m disjoint simple polygons in the plane, with a total of n vertices, preprocess it to quickly answer queries of the following type: for a positive constant c and a simple polygonal path π, from a point u to a point v in free space, decide whether π has clearance at least c, that is, there is no polygonal obstacle within distance c of π.

Somehow surprisingly, it seems there is little related work on this problem. To solve it, we show how to solve the following related problem:

Object-Obstacles Proximity Queries: Given a set S of m polygonal obstacles with a total of n vertices, preprocess S into a data structure so that the obstacle in S closest to a query object ρ can be reported quickly.

We consider the set S as a collection of disjoint simple polygons, and the query object corresponds to a line segment (or line). Once the segment-polygon proximity problem is solved, one can check for each of the segments of the given path π whether the segment has a clearance of c or not, and also report the *minimum clearance of the path*, defined as the minimum of the clearances of the line segments along the path.

CHAPTER 2

CITY GUARDING WITH LIMITED FIELD OF VIEW

In 1973, Victor Klee posed the following question: *How many guards are necessary, and how many are sufficient to patrol the paintings and works of art in an art gallery with n walls?*. The problem is known as the *art-gallery problem*. This question has stimulated a plethora of research papers, surveys, and books where various versions of the *art-gallery* problem are studied.

In this chapter, we consider the problem of guarding the aerial space of a given city. The input consists of a rectangular city with k rectangular/orthogonal buildings where all buildings are vertical, of arbitrary positive height, and the range of vision of a guard is bounded by 180°. Each guard is placed at the top corner of a building and is oriented such that the seen and unseen regions of the guard are separated by a vertical plane parallel with one of the sides of the building where the camera is placed. Thus, when building bases are isothetic (axis-aligned) rectangles, a camera will face in one of four directions, East, West, North, or South (E, W, N, S). From now on, we assume cameras are placed as stated here, unless otherwise specified, and may omit to mention camera orientation throughout the chapter.

We consider the following four versions of the city guarding problem:

1. Buildings are axis-aligned and have a rectangular base,
2. Buildings are axis-aligned and have an orthogonal base,
3. Buildings have arbitrary orientation and have a rectangular base,
4. Buildings have arbitrary orientation and have an orthogonal base.

To solve the four versions, we solve the following variations of the art-gallery problem:

(V1) Given an axis-aligned rectangle P with k disjoint axis-aligned rectangular holes, place vertex guards on holes such that every point inside P is visible to at least one guard, where the range of vision of a guard is 180°.

(V2) Given an axis-aligned rectangle P with k disjoint axis-aligned orthogonal holes, $H_1, .., H_k$ with $m_1, m_2, \ldots, m_k$ vertices respectively such that $\sum_{i=1}^{k} m_i = m$, place vertex guards on holes such that every point inside P is visible to at least one guard, where the range of vision of guards is 180°.

(V3) Given an axis-aligned rectangle P with k disjoint (arbitrary oriented) rectangular holes, place vertex guards on holes such that every point inside P is visible to at least one guard, where the range of vision of a guard is 180°.

(V4) Given an axis-aligned rectangle P with k disjoint orthogonal holes, $H_1, H_2, \ldots, H_k$ with $m_1, m_2, \ldots, m_k$ vertices respectively such that $\sum_{i=1}^{k} m_i = m$, place vertex guards on holes such that every point inside P is visible to at least one guard, where the range of vision of guards is 180°.

Remark: Our results also hold if we remove P or if P is a convex polygon, or a star-shaped polygon, such that every building is in the kernel of P.

For the first problem (V1), we present an upper bound of $2k + \lfloor \frac{k}{4} \rfloor + 4$ on the number of vertex guards. For the second problem (V2), we present an upper bound of $\frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 4$ on the number of vertex guards. To obtain these bounds, we provide a divide and conquer algorithm that departs from standard approaches (triangulation, coloring) used in most art gallery solutions.

For the third problem (V3), we show that $3k + 1$ vertex guards are sometimes necessary and conjecture that the bound is tight. For the fourth problem(V4), we show that $r - k + 1$ vertex guards are sometimes necessary to guard a rectangular polygon with k holes and r reflex vertices. We conjecture that the bound is tight.

A comparison of our sufficiency and necessity results with those in (Bao et al., 2008) is shown in Table 2.1. Our solutions implicitly provide bounds for guarding the aerial space of the city and set a basic foundation for monitoring drones flying within city limits.

Table 2.1: Sufficient and necessary results comparisons. A tight bound is shown in blue color(bold), a sufficiency bound in red color and a necessary bound in green color(italics) .

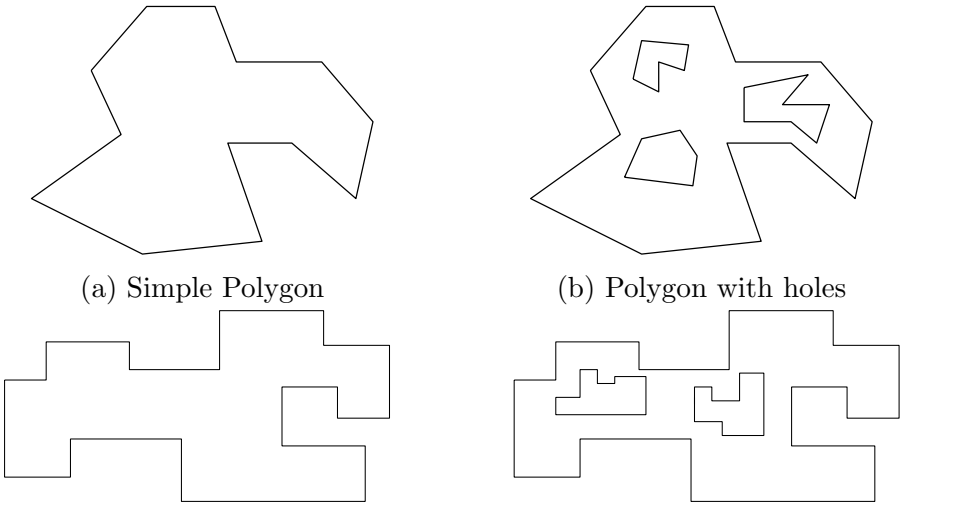
	(Bao et al., 2008) Guard vision range: 360° (axis-aligned rectangular buildings)	Guard vision range: 180° (axis-aligned rectangular buildings)	Guard vision range: 180° (non-axis-aligned rectangular buildings)
Roof Guarding	$\mathbf{\lfloor \frac{2(k-1)}{3} \rfloor + 1}$	$\mathbf{k}$	$\mathbf{k}$
Ground and Wall Guarding	$k + \lfloor \frac{k}{4} \rfloor + 1$	$2k + \lfloor \frac{k}{4} \rfloor + 4$	$3k + 1$
City Guarding	$k + \lfloor \frac{k}{2} \rfloor + 1$	$2k + \lfloor \frac{k}{4} \rfloor + 4$	$3k + 1$

2.1 Basic Terminology

Among all the variations that have been considered for the art-gallery problem, there are various types of restrictions that have been imposed on the structure of polygons and placement of guards.

A **polygon** is a two-dimension shape formed by a finite set of line segments connected to form a closed polygonal chain. The segments are known as edges, and their intersection points are known as vertices. A polygon is **simple** if it divides the plane into two regions. A **simple polygon** has no holes and does not intersect itself (see Figure 2.1a). A polygon is **orthogonal** if all of its edges intersect at a right angle (refer Figure 2.1c). A **polyhedron** is a three-dimensional closed shape made by joining together polygons. Each such polygon corresponds to a face of polyhedron. Edges are the line-segments where faces meet each other, and vertices are the corners of the polyhedron. A **hole** is a polygon inside the polygon or a polyhedron inside the polyhedron.

A **triangulation** of a polygon P is a partitioning of P into a set of disjoint interior triangles, where triangle edges are either edges of P or internal diagonals joining two distinct vertices of P. Triangulation played a crucial role in the study of the Art-Gallery problem, as a lot of bounds are proven based on polygon triangulation.



(a) Simple Polygon

(b) Polygon with holes

(c) Orthogonal Polygon

(d) Orthogonal Polygon with orthogonal holes

Figure 2.1: Type of polygons.

The following main types of guards have been studied in the literature:

1. **Point Guards:** These guards are placed anywhere inside a polygon/ polyhedron.

2. **Vertex Guards:** These guards are only placed at the vertices of a polygon/polyhedron.

3. **Edge Guards:** These guards are only placed on the edges of a polygon/polyhedron.

4. **Face Guards:** Face guards are only defined for polyhedron and are only placed on the faces of a polyhedron.

5. **Mobile Guards:** These guards are allowed to move along closed line segments contained in a polygon/polyhedron.

2.2 Variations of the Art Gallery Problem

2.2.1 Simple Polygon Results

Given a simple polygon P in the plane, with n vertices, Chvatal (Chvatal, 1975) proved that $\lfloor n/3 \rfloor$ vertex guards are always sufficient and sometimes necessary to guard P. Chvatal's proof was later simplified by Fisk (Fisk, 1978) using the existence of a three-coloring of a triangulated polygon. The polygon is triangulated by adding $n-2$ interior diagonals, and the resultant triangulated polygon vertices are colored using three colors such that any two vertices joined by an edge or a diagonal receive different colors. This process partitions the vertices into three chromatic classes and a guard is placed at each vertex of the smallest chromatic class (see Figure 2.2).

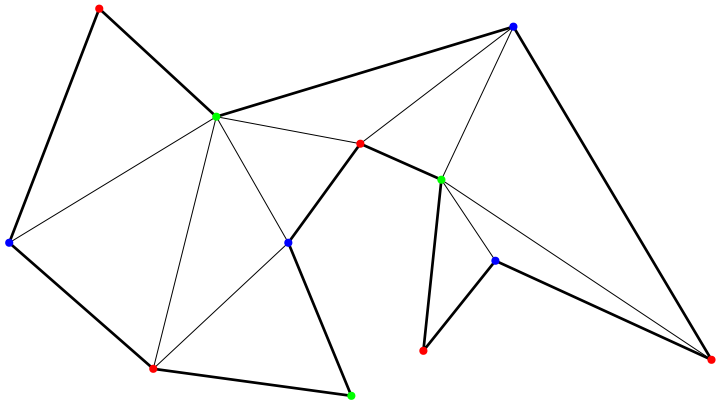


Figure 2.2: Triangulation and 3-coloring of a polygon.

Theorem 1. *(Chvatal, 1975), (Fisk, 1978) Given a simple polygon in a plane with n vertices,* $\lfloor \frac{n}{3} \rfloor$ *vertex guards are sometimes necessary and always sufficient to guard the polygon.*

When the view of the guard is limited to 180°, Toth (Tóth, 2000) showed that $\lfloor \frac{n}{3} \rfloor$ point guards are always sufficient to cover the interior of P (thus, moving from 360 to 180 range of vision keeps the same sufficiency number). F. Santos (Santos, 1995) conjecture that $\lfloor \frac{3n-3}{5} \rfloor \pi$ vertex guards are always sufficient and occasionally necessary to cover any polygon with n

vertices. Later in 2002, Toth (Tóth, 2002) provided a lower bound on the number of point guards when the range of vision α is less than 180°. When $\alpha < 180°$, there exist a polygon P that cannot be guarded by $\frac{2n}{3} - 3$ guards. For $\alpha < 90°$ there exist P that cannot be guarded by $\frac{3n}{4} - 1$ guards, and for $\alpha < 60°$ there exist P where the number of guards needed to cover P is at least $\lfloor \frac{60}{\alpha} \rfloor \frac{(n-1)}{2}$.

2.2.2 Orthogonal Polygon Results

In 1983, Kahn et al. (Kahn et al., 1983) showed that if every pair of adjacent sides of the polygon form a right angle, then $\lfloor \frac{n}{4} \rfloor$ vertex guards are occasionally necessary and always sufficient to guard a polygon with n vertices (refer to Figure 2.4). The proof uses the observation that any finite region bounded by a finite number of edges, each of which lies parallel to one of a fixed pair of perpendicular axes, has a partition into disjoint convex quadrilaterals. The resultant polygon vertices are colored using four colors such that any two vertices joined by an edge or a diagonal receive different colors. This process partitions the vertices into four chromatic classes, and a guard is placed at each vertex of the smallest chromatic class.

Significant research has been carried out to solve the problem of decomposing an orthogonal polygon into disjoint convex quadrilaterals. In 1982, Sack and Toussaint (Sack and An, 1982) developed a $O(n \log n)$ time algorithm for quadrilateralizing an orthogonal polygon. In 1985, Lubiw (Lubiw, 1985) also provided an $O(n \log n)$ time quadrilateralization algorithm for orthogonal polygons. They use a two-step process. First, the orthogonal polygon is partitioned into a specific type of monotone polygons, which are, in turn, partitioned into quadrilaterals. This is similar to the Fisk (Fisk, 1978) proof for Theorem 1. Refer to Figure 2.3.

Theorem 2. *(Kahn et al., 1983) Given an orthogonal polygon in the plane with n vertices,* $\lfloor \frac{n}{4} \rfloor$ *vertex guards are sometimes necessary and always sufficient to guard the polygon.*

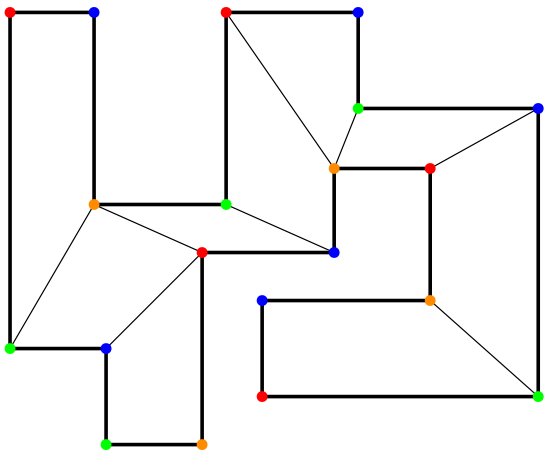


Figure 2.3: Convex quadrilateralization and 4-coloring of an orthogonal polygon

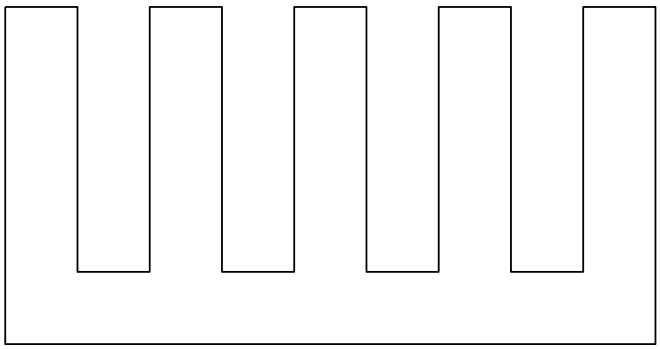


Figure 2.4: An orthogonal polygon that requires $\lfloor \frac{n}{4} \rfloor$ guards

In 1983, O'Rourke (O'Rourke, 1983a) showed that $1 + \lfloor \frac{r}{2} \rfloor$ vertex guards are necessary and sufficient to cover the interior of an orthogonal polygon with r reflex vertices. The algorithm is based on the partition of the polygon into L-shaped pieces. An L-shaped piece is an orthogonal polygon with six vertices that can be guarded with one guard. The proof is by induction.

Theorem 3. *(O'Rourke, 1983a)* $\lfloor \frac{r}{2} \rfloor + 1$ *vertex guards are always sufficient and sometimes necessary to cover the interior of an orthogonal polygon with* r *reflex vertices.*

Castro and Urrutia (Estivill-Castro and Urrutia, 1994) provided a tight bound of $\lfloor \frac{3(n-1)}{8} \rfloor$ on the number of orthogonal guards placed on the vertices, sufficient to cover an orthogonal polygon with n vertices.

Theorem 4. *(Estivill-Castro and Urrutia, 1994)* $\lfloor \frac{3(n-1)}{8} \rfloor$ *orthogonal guards are always sufficient and sometimes necessary to cover the interior of an orthogonal polygon with* n *vertices.*

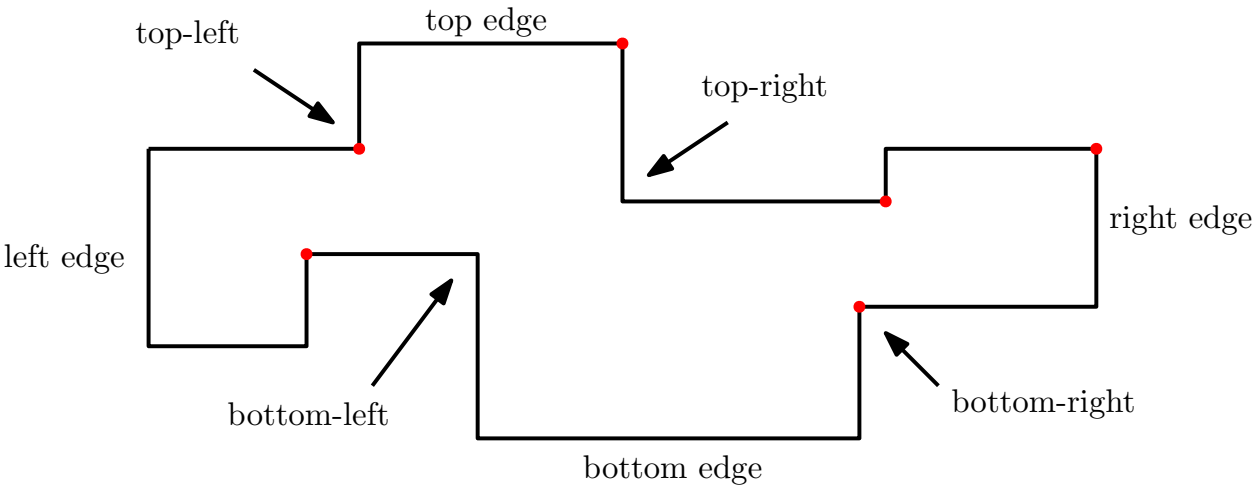


Figure 2.5: Classifying the edges and vertices of an orthogonal polygon; guards are placed according to top-right rule.

The proof idea of the above theorem is as follows: Each edge is classified as *top*, *bottom*, *right* and *left* edge in the natural way. Similarly, each angle is classified as top-right, top-left, bottom-right, and bottom-left, depending on the label of corresponding edges. Refer to Figure 2.5. The guards are then placed according to one of the following rules: (i) top-right rule, (ii) top-left rule, (iii) bottom-right rule, (iv) bottom-left rule. In the *top-right* rule, a guard is placed at the right end of all the top edges, and a guard is placed at the top end of all the right edges. The polygon can be guarded by placing guards according to any of the rules.

2.2.3 Polygon with Holes Results

For a polygon P with n vertices and h holes, the value n is the sum of the number of vertices of P and the number of vertices of the holes. Let $g(n, h)$ be the minimum number of point

guards and $g^v(n,h)$ be the minimum number of vertex guards necessary to cover any polygon with n vertices and h holes.

O'Rourke (O'Rourke, 1983b) gave a first proof on guarding polygons with holes and showed that $g^v(n,h) \leq \lfloor \frac{n+2h}{3} \rfloor$. The proof was based on removing one hole at a time and converting a polygon with holes into a polygon without holes. This process introduces an additional $2h$ vertices, and the polygon obtained has a total of $n+2h$ vertices.

Theorem 5. *(O'Rourke, 1983b) Any polygon with n vertices and h holes can always be guarded with $\lfloor \frac{n+2h}{3} \rfloor$ vertex guards.*

Shermer conjectured that $g^v(n,h) \leq \lfloor \frac{n+h}{3} \rfloor$ and this is a tight bound. He was able to prove that, for $h = 1$, $g^v(n,1) = \lfloor \frac{n+1}{3} \rfloor$. However, for $h > 1$ the conjecture remains open. Shermer's result can be found in (O'Rourke, 1987; Shermer, 1992). Refer to Figure 2.6.

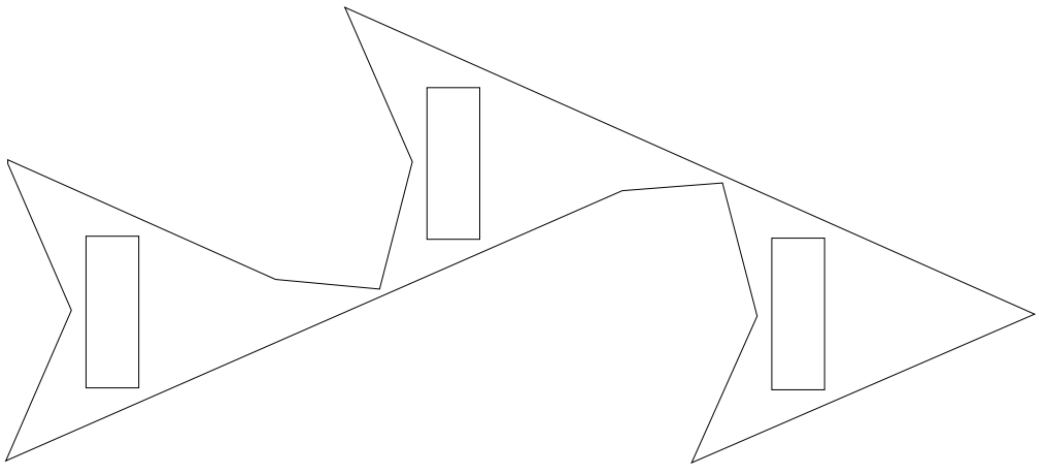


Figure 2.6: (O'Rourke, 1987) A polygon that require $\lfloor \frac{n+h}{3} \rfloor$ guards

Conjecture 1. *(Shermer, 1992) Any polygon with n vertices and h holes can always be guarded with $\lfloor \frac{n+h}{3} \rfloor$ vertex guards.*

Sachs and Souvaine (Bjorling-Sachs and Souvaine, 1995) and Hoffmann et al. (Hoffmann et al., 1991) showed that no art gallery problem with n vertices and h holes requires more than $\lfloor \frac{n+h}{3} \rfloor$ point guards and provided an $O(n^2)$ algorithm to find such placement, which is based on triangulation and 3-coloring. The main idea in Sachs and Souvaine (Bjorling-Sachs and Souvaine, 1995) paper is to connect each hole of the polygon to the exterior with a quadrilateral "channel" and triangulate the hole-free version of the polygon. The channels are such that (i) There exists a triangle T in the remaining polygon such that any point in it sees all of the channel and (ii) Only one new vertex is introduced per channel. Note that T is then forced to be in a triangulation of the remaining polygon. Guards are then placed according to Fisk's (Fisk, 1978) proof.

Theorem 6. *(Bjorling-Sachs and Souvaine, 1995; Hoffmann et al., 1991)* $\lceil \frac{n+h}{3} \rceil$ *point guards are always sufficient and occasionally necessary to guard any polygon with* n *vertices and* h *holes.*

2.2.4 Orthogonal Polygon with Holes Results

For this version, all polygons and holes are orthogonal and axis-aligned. Let $orth(n, h)$ be the minimum number of point guards and $orth^v(n, h)$ be the minimum number of vertex guards necessary to guard any orthogonal polygon with n vertices and h holes. Note that $orth(n, h) \leq orth^v(n, h)$.

O'Rourke's method in Theorem 5 extends to show that: $orth^v(n, h) \leq \lfloor \frac{n+2h}{4} \rfloor$.

Theorem 7. *(O'Rourke, 1983b)* $\lfloor \frac{n+2h}{4} \rfloor$ *vertex guards are always sufficient to guard any orthogonal polygon with* n *vertices and* h *holes.*

Shermer (O'Rourke, 1987) conjectured, using the example in Figure 2.7, that $orth^v(n, h) \leq \lfloor \frac{n+h}{4} \rfloor$ which Aggarwal (Aggarwal, 1984) established for $h = 1$ and $h = 2$. Zylinski (Żyliński,

2006) showed that $\lfloor \frac{n+h}{4} \rfloor$ vertex guards are always sufficient to guard any orthogonal polygon with n vertices and h holes, provided that there exists a quadrilateralization whose dual graph is a cactus.

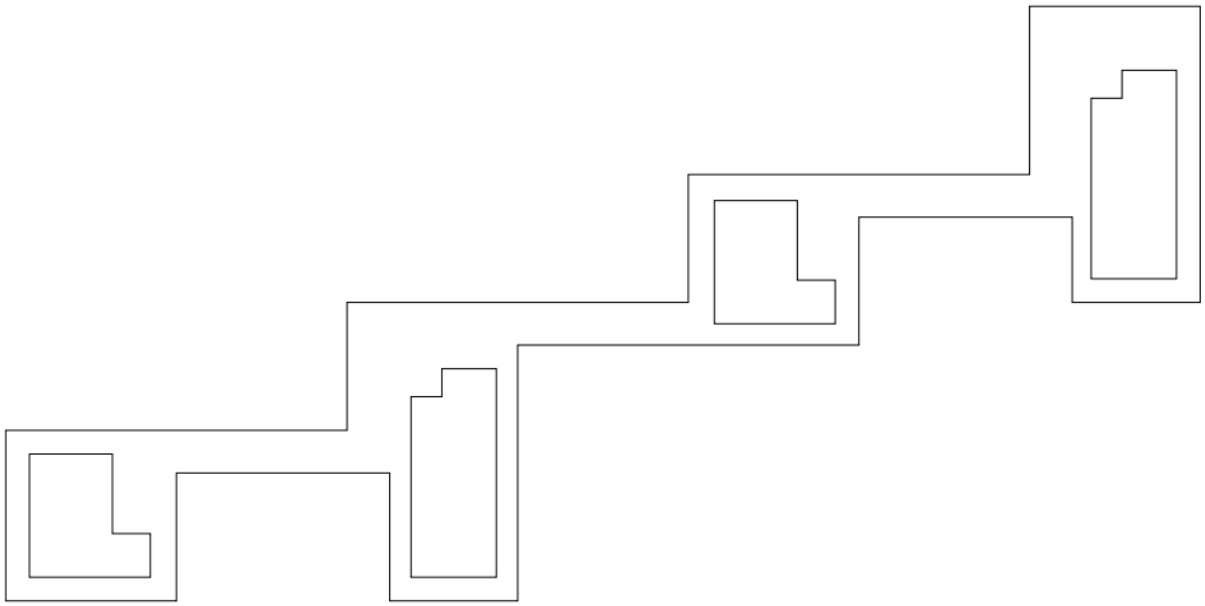


Figure 2.7: (O'Rourke, 1987) An orthogonal polygon that require $\lfloor \frac{n+h}{4} \rfloor$ guards.

Theorem 8. *(Żyliński, 2006)* $\lfloor \frac{n+h}{4} \rfloor$ *vertex guards are always sufficient and sometimes necessary to guard any orthogonal polygon with* n *vertices and* h *holes, when there exists a quadrilateralization of the polygon whose dual graph is a cactus.*

O'Rourke also conjectured that $orth(n, h)$ is independent of h: $orth(n, h) = \lfloor \frac{n}{4} \rfloor$, which was verified by Hoffmann (Hoffmann, 1990). In 1990, Hoffmann (Hoffmann, 1990) showed that $\lfloor \frac{n}{4} \rfloor$ point guards are always sufficient and sometimes necessary to guard an orthogonal polygon with n vertices and an arbitrary number of holes. The guards are placed on points other than the vertices or the boundary. The solution corresponds to partitioning the polygon into $\leq \lfloor \frac{n}{4} \rfloor$ r-stars, each of size at most 16.

Theorem 9. *(Hoffmann, 1990)* $\lfloor \frac{n}{4} \rfloor$ *point guards are always sufficient to guard any orthogonal polygon with n vertices and h holes.*

In 1991, Hoffmann and Kaufmann (Hoffmann and Kaufmann, 1990) provided an $O(n^{3/2} \log^2 n \log \log n)$ time algorithm for the placement of guards for orthogonal art galleries with an arbitrary number of holes. In 1996, Hoffmann and Kriegel (Hoffmann and Kriegel, 1996) showed that $\leq \lfloor \frac{n}{3} \rfloor$ vertex guards are sufficient to guard the interior of an orthogonal polygon with holes. The proof is based on the partition of the orthogonal polygon into convex quadrilaterals, that are further triangulated, and uses 3-coloring on the resultant graph.

Theorem 10. *(Hoffmann and Kriegel, 1996)* $\lfloor \frac{n}{3} \rfloor$ *vertex guards are always sufficient to guard an orthogonal polygon with n vertices and arbitrary number of holes.*

Consider $orth^v(n, .)$ as the maximum of $orth^v(n, h)$ over all h. Hoffmann conjectured that $orth^v(n, .) \leq \lfloor \frac{2n}{7} \rfloor$ (refer to Figure 2.8), disproving the earlier conjecture of Aggarwal (Aggarwal, 1984) that $orth^v(n, .) \leq \lfloor \frac{3n}{11} \rfloor$. In 2013, Michael and Pinciu Michael and Pinciu (Michael and Pinciu) improved this bound and showed that an orthogonal gallery with n vertices and an unspecified number of holes can be guarded by at most $\frac{17n-8}{52}$ vertex guards ($17 / 52 \approx 0.3269$).

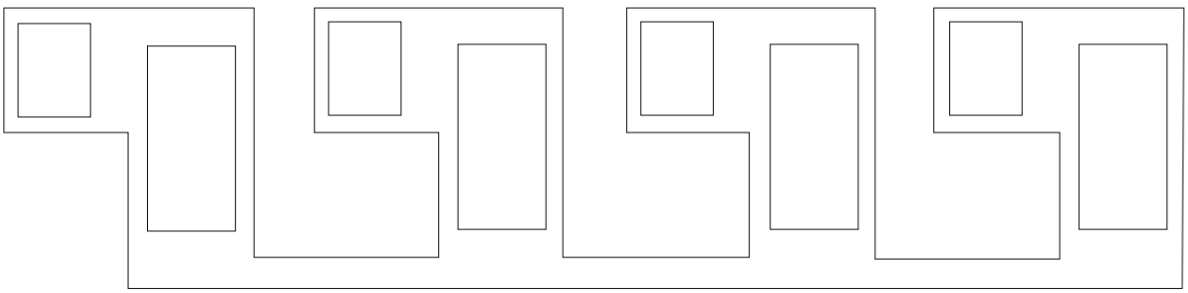


Figure 2.8: (O'Rourke, 1987)An orthogonal polygon that require $\lfloor \frac{2n}{7} \rfloor$ guards.

In 1998, Abello et al. (Abello et al., 1998) provided a first tight bound of $\lfloor \frac{3n+4(h-1))}{8} \rfloor$ for the number of guards placed at the vertices of an orthogonal polygon with n vertices

and h holes, which are sufficient to guard the polygon, and described a simple linear-time algorithm to find the guard placement for an orthogonal polygon (with or without holes). The algorithm consists of a traversal of the boundary of the polygon. The proof is similar to the proof of Theorem 4.

Theorem 11. *(Abello et al., 1998)* $\lfloor \frac{3n+4(h-1))}{8} \rfloor$ *orthogonal guards are always sufficient and sometimes necessary to cover any orthogonal polygon with* n *vertices and* h *holes.*

2.2.5 Families of Convex Sets (Triangles and Quadrilaterals) on the Plane Results

In 1977, Toth (Tóth, 1977) considered the following problem: Given a set F of n disjoint compact convex sets in a plane, how many guards are sufficient to cover every point in the boundary of each set in F. Toth proved that $max\{2n, 4n-7\}$ point guards are always sufficient to cover n disjoint compact convex sets in a plane. Everett and Toussaint (Everett and Toussaint, 1990) proved that the families of n disjoint squares $n > 4$, can always be guarded with n point guards. For families of disjoint isothetic rectangles (rectangles are *isothetic* if all their sides are parallel to the coordinate axes.), Czyzowicz et al. (Czyzowicz et al., 1993) proved that $\lfloor \frac{4n+4}{3} \rfloor$ point guards suffice and conjectured that, for a constant c, $n+c$ point guards would suffice. If the rectangles have equal width, then $n+1$ point guards suffice, and $n-1$ point guards are occasionally necessary. Refer (Martini and Soltan, 1999) for more details.

In 1994, Blanco et al. (Blanco et al., 1994) considered the problem of guarding the region of the plane, excluding the interior of the quadrilaterals (free space). Given n pairwise disjoint quadrilaterals in the plane whose convex hull has no cut-off quadrilaterals, they showed that $2n$ vertex guards are always sufficient to cover the free space and all locations could be found on $O(n^2)$ time. If the quadrilaterals are isothetic rectangles, all locations can be placed in $O(n)$ time.

Garcia-Lopez (de la Calle, 1995) proved that $\lfloor \frac{5m}{9} \rfloor$ vertex lights are always sufficient and $\lfloor \frac{m}{2} \rfloor$ vertex guards are occasionally necessary to guard the free space generated by a family of disjoint polygons with m vertices. To cover the free space generated by any family of n disjoint quadrilaterals, he proved that $2n$ vertex lights are always sufficient and occasionally necessary and that $\lfloor \frac{5n+3}{3} \rfloor$ point guards are always sufficient. He conjectured that $n+c$ point lights can always cover the free space generated by m disjoint quadrilaterals, c is a constant which was proved false by Czyzowicz and Urrutia (J. Czyzowicz, 1996).

Czyzowicz et al. (Czyzowicz et al., 1994) proposed the following problem: Given a set F of n disjoint compact convex sets in a plane, how many guards are sufficient to protect each set in F. A set F is protected by a guard g if at least one point in the boundary of F is visible from g. They prove that $\lfloor \frac{2(n-2)}{3} \rfloor$ point guards are always sufficient and occasionally necessary to protect any family of n disjoint convex sets, $n > 2$. To protect any family of n isothetic rectangles, $\lceil \frac{n}{2} \rceil$ point guards are always sufficient, and $\lfloor \frac{n}{2} \rfloor$ point guards are sometimes necessary.

Urrutia (Urrutia, 2004) showed that any family of n disjoint rectangles can be guarded with at most $n+1$ point guards. A big rectangle encloses the elements of F, and consider this as an orthogonal polygon with holes. The total number of vertices is now $4n+4$. Using the results from guarding orthogonal polygon with holes, this can be guarded with $n+1$ guards.

Czyzowicz et al. (Czyzowicz et al., 1993) showed that any family of n disjoint triangles can be guarded with at most $\lfloor \frac{4n+4}{3} \rfloor$ point guards and $n-1$ are occasionally necessary. They also showed that $n+1$ guards are always sufficient and $n-1$ guards are occasionally necessary to illuminate any family of n homothetic triangles and conjectured that there is a constant c such that $n+c$ point guards sufficient to guard any collection of n triangles. Later, Toth (Tóth, 2003) showed that $\lfloor \frac{5n+2}{4} \rfloor$ guards can monitor the boundaries and the free space of n disjoint triangles.

2.2.6 Polyhedral Terrain Results

In 1989, Goodchild and Lee (Goodchild and Lee, 1989) proposed coverage problems and visibility regions on topological surfaces. The problem is to locate the minimum number of viewpoints to see the entire surface. In the same year, Cole and Sharir (Cole and Sharir, 1989) proposed various problems considering the visibility of a polyhedral terrain from a point lying above it.

A polyhedral terrain is a polyhedral surface in three dimensions such that its intersection with any vertical line is either empty or a point. A polyhedral terrain is triangulated if each of its faces is a triangle. Notice that a polyhedral terrain has a different structure than a city with vertical buildings. In 1994, Everett and Rivera-Campo (Everett and Rivera-Campo, 1994) showed that $\lfloor \frac{n}{3} \rfloor$ edge guards (i.e., guards free to patrol an entire edge of the terrain) are always sufficient to guard a triangulated polyhedral terrain with n vertices.

The problem of guarding in 2.5D has been studied in (Bose et al., 1997) for polyhedral terrains, where a terrain corresponds to a triangulation of a set of points in the plane. In 1997, Bose et al. (Bose et al., 1997) proved that $\lfloor \frac{n}{2} \rfloor$ vertex guards are always sufficient and sometimes necessary to guard an n-vertex polyhedral terrain. Concerning edge-guards, they showed that $\lfloor \frac{4n-4}{13} \rfloor$ edge guards are sometimes necessary to guard the surface of a n-vertex triangulated polyhedral terrain. They also presented a linear time algorithm to place $\lfloor \frac{3n}{5} \rfloor$ vertex guards and $\lfloor \frac{2n}{5} \rfloor$ edge guards to cover an n-vertex triangulated polyhedral terrain. Cole (Cole and Sharir, 1989) proved that the minimum vertex guard problem is NP-hard and Batista et al. (Batista, 2010) proved that the minimum edge-guard problem is NP-hard. In 2003, Bose et al. (Bose et al., 2003) showed that at most $\lfloor \frac{n}{2} \rfloor$ vertex guards, and at most $\lfloor \frac{n}{3} \rfloor$ edge guards, are required to guard an n-vertex triangulated polyhedral terrain.

In 2012, Iwamoto et al. (Iwamoto et al., 2012) showed an upper bound of $\lfloor \frac{n}{3} \rfloor$ and a lower bound of $\lfloor \frac{2n-5}{7} \rfloor$ for the number of face guards needed to guard an n-vertex triangulated

polyhedral terrain. Later, Iwamoto et al. (Iwamoto and Kuranobu, 2012) improved these bounds to $\lfloor \frac{n-1}{3} \rfloor$.

2.2.7 Polyhedron Results

There exists a polyhedron P_n with n vertices such that even placing a guard on every vertex does not guard the entire interior of P_n and $O(n^{3/2})$ non-vertex guards are necessary to cover the interior (O'Rourke, 1987). Urrutia (Urrutia, 2004) conjectured that any polyhedron of genus zero with e edges can be guarded with at most $\lfloor \frac{e}{6} \rfloor + O(1)$ edge guards. In 2012, Cano et al. (Cano et al., 2012) proved that any polyhedron with e edges can be guarded by at most $\lfloor \frac{27e}{32} \rfloor$ edge guards. They also showed that every polyhedron in R^3 with e edges and a connected 1-skelton can be guarded with at most $\frac{5e}{6} + \frac{1}{12}$ edge guards.

Let $g(P)$ be the minimum number of face guards needed for a polyhedron P and let $g(f)$ be the minimum of $g(P)$ over all polyhedra P with exactly f faces. In 2011, Souvaine et al. (Souvaine et al., 2011) showed that $\lfloor \frac{f}{5} \rfloor \leq g(f) \leq \lfloor \frac{f}{2} \rfloor$ for general polyhedra and $\lfloor \frac{f}{7} \rfloor \leq g(f) \leq \lfloor \frac{f}{6} \rfloor$ for orthogonal polyhedra.

In 2015, Viglietta (Viglietta, 2015) showed that $\lfloor \frac{f}{4} \rfloor \leq g(f) \leq \lfloor \frac{f}{2} \rfloor - 1$. He mentioned a tight bound of $\lfloor \frac{f}{6} \rfloor$ for open face guards for orthogonal polyhedrons, where f is total number of faces of a given polyhedron. Urrutia (Urrutia, 2004) conjectured that any orthogonal polyhedron in 3-dimension with e edges can always be guarded with at most $\lfloor \frac{e}{12} \rfloor + O(1)$ edge guards. In 2011, Benbernou (Benbernou et al., 2011) showed that an orthogonal polygon with e edges and genus g can be guarded by $\lfloor \frac{11e}{72} + \frac{g}{6} \rfloor - 1$ edge guards, while if the polyhedron has r reflex edges, then $\lfloor \frac{7r}{12} \rfloor - g + 1$ edge guards are sufficient, and further conjectured that $\frac{r}{2} + O(1)$ edge guards always suffice.

Benbernou et al. (Benbernou et al., 2011) also introduced a new model, of open edge guards. Given an orthogonal polyhedron with e edges, out of which r are reflex, (Benbernou et al., 2011) improved the upper bound on the edge guards to show that $\lfloor \frac{11e}{72} \rfloor$ (open or closed) edge guards suffice.

Recently, in 2017, Viglietta (Viglietta, 2017) conjectured that any non-convex orthogonal polyhedron with e edges and genus g can be guarded by at most $\lfloor \frac{e}{12} + \frac{g}{2} \rfloor$ (open or closed) reflex edge guards. He also stated that any orthogonal polyhedron with $r > 0$ reflex edges and genus g can be guarded by at most $\lfloor \frac{r-g}{2} \rfloor + 1$ (open or closed) reflex edge guards and showed that this bound is tight for $g = 0$. He provided an upper bound of $\lfloor \frac{e-4}{8} \rfloor + g$ (open or closed) reflex edge guards, that are sufficient to cover any non-convex 2-reflex orthogonal polyhedron with e edges, and also presented an $O(n \log n)$ algorithm to compute guard locations matching the above bounds.

2.2.8 City Guarding Results

In 2008, Bao et al. (Bao et al., 2008) proposed the city guarding problem where one is given a rectangular city with k vertical buildings, each having an axis-aligned rectangular base. The guards are to be placed only at the top vertices of the buildings. They showed that $\lfloor \frac{2(k-1)}{3} \rfloor + 1$ vertex guards are sometimes necessary and always sufficient to guard the roofs (Roof Guarding Problem). They further proved that $k + \lfloor \frac{k}{4} \rfloor + 1$ vertex guards are always sufficient to guard the ground and the walls and $k + \lfloor \frac{k}{2} \rfloor + 1$ vertex guards are always sufficient to guard the aerial space, which includes all roofs and walls of the buildings, and the ground.

We start with the following theorem, that allows us to limit our attention to guarding the roofs and walls of the buildings, and the ground.

Theorem 12. *If guards are placed so that the roofs, walls, and the ground of the city are guarded, then every point in the aerial space of the city is guarded.*

Proof. Let p be a point in the aerial space of the city and assume p is not guarded. Let p' be the vertical projection of p onto the ground (or a building roof) and let g be a guard that sees p' (such g exists since the ground of the city is guarded). Then g, p and p' define a vertical plane π. Consider the vertical triangle defined by g, p, and p'. If any portion of a building intersects the triangle side gp' at some point q then the line segment qq' is part of that building, where q' is the vertical projection of q onto the ground. Since the line segment qq' intersects the triangle side gp' it then follows that p' is not visible from g, a contradiction. □

A similar result holds if we aim for walls and ground guarding only (no roof guarding requirement), including the space between the buildings.

2.3 City Guarding Problem for Axis-Aligned Rectangle Buildings

Given a rectangular city with k disjoint axis-aligned vertical buildings, each having a rectangular base, the goal is to place the minimum number of cameras that can see only the half-space in front of them (denoted as 180° range of vision), at the top corners (vertices) of the buildings to guard the city (roofs, walls, ground, and aerial space). Thus, when a guard (camera) is aligned with a wall of the building, it is placed on the half-space seen by the guard is bounded by a vertical plane containing that wall.

2.3.1 Roof Guarding

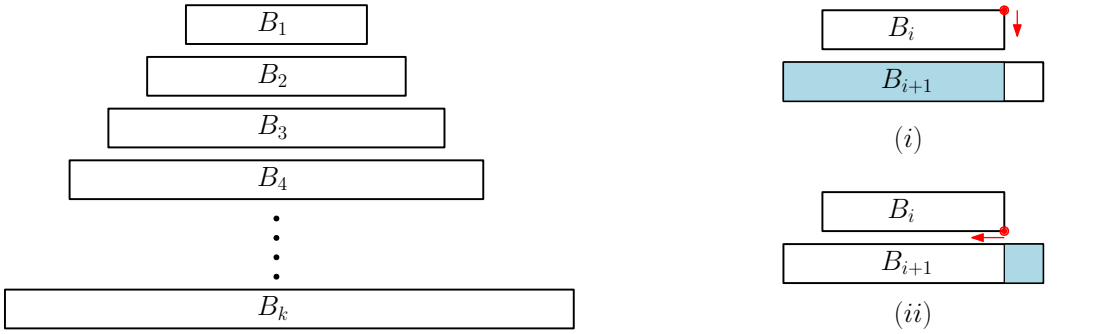
In this version of the problem, the guards (i.e., cameras) are placed at top vertices of the buildings, the visibility range of a guard is the half-space in front of it (denoted as 180° range of vision), and each guard is oriented to see in North, South, East, or West direction. The input consists of k vertical buildings with a rectangular, axis-aligned base, all within some axis-aligned rectangle P (city limits).

Theorem 13. *Given a city with k disjoint axis-aligned rectangular buildings, k vertex guards are always sufficient and sometimes necessary to guard the roofs.*

Proof. The sufficiency bound is trivial (place one guard at each roof). For the necessary part, consider a set $S = \{B_1, B_2, B_3, \ldots, B_k\}$ of k buildings, as shown in Figure 2.9a, with the following setup:

1. the height h_{B_i} of building B_i is greater than the height h_{B_j} of building B_j, $\forall\, i, j$ such that $1 \leq i < j \leq k$, and

2. $\forall\, i < j - 1$, building B_{j-1} totally blocks the visibility between B_i and B_j.



(a) City Setup (b) Possible guards position

Figure 2.9: k guards are needed to guard the roofs.

We need to place the first guard on building B_1 to guard its roof, because $h_{B_1} > h_{B_i}, \forall i > 1$. There are four possible positions to place a vertex guard on building B_1. Let the guard be placed on one of the right vertices (placing the guard on one of the left vertices results in symmetric cases). Consider the two relevant orientations of the guard, shown in Figure 2.9b, out of three possible positions that can see the roof of B_1, where the arrow corresponds to the direction in which the guard is guarding the roof. Notice that a guard facing West is placed at the lower vertex of B_1 rather than at the top vertex. Due to the limited visibility of the guard, there is no vertex on building B_1 from where roofs of both building B_1 and B_2

are completely visible. Therefore, the next guard should either be placed on building B_2 or on B_1, such that the roof of building B_2 is completely visible after placing this guard. If we place the second guard on B_1, then the next guard must be placed on building B_3 because no point on the roof of building B_3 is visible by the previously placed guards. Thus, we can place the next guard on building B_2. The rest follows by induction on the number of buildings, as we are left with a similar problem on $k-1$ buildings. □

In the rest of this section, we provide a greedy algorithm to find a placement of guards to guard all roofs. The algorithm assigns a roof to a guard only when that guard entirely guards the roof. For each top corner of a building B_i, there are only four possible orientations for the guard: (i) Guard facing North (ii) Guard facing South (iii) Guard facing East and (iv) Guard facing West of the building. If we consider all four vertices, then there are 16 such guard positions, and for k buildings, we have $16k$ guard positions.

We convert the problem of roof guarding to a graph problem. Assuming no two heights are the same (we can use small perturbations to guaranty this condition), we construct a directed acyclic graph $G = [V, E]$ as follows:

1. For each building B_i, add vertex v_i to V;

2. For each possible position of a guard on the corners of building B_i we add vertices $v_i^1, v_i^2, \ldots, v_i^{16}$ to V. Let roof of building B_i be visible to the guards that correspond to vertices $v_i^1, v_i^2, \ldots, v_i^8$;

3. Add a directed edge (v_i^a, v_j) if the roof of B_j is completely visible to the guard corresponding to vertex v_i^a.

After performing the above steps, we obtain a directed disconnected graph $G = [V; E]$. Let $deg(i, j)$ be the out-degree of vertex v_i^j in graph G. The graph obtained for the structure in Figure 2.9 is shown in Figure 2.10. We optimize the graph by removing all vertices v_i^j

with degree zero, as no roof is visible by the corresponding guards. For a building B_i, if $deg(i, j) = 1$ for some $j \in [1, 8]$ then we remove v_i^j if the in-degree of $v_i \neq 0$ (see Figure 2.11). From the optimized graph it is easy to see that k guards are required to guard the city in Figure 2.9.

The graph obtained by performing the steps described is a directed acyclic graph, assuming buildings have distinct heights. Let H be the set which contains vertices $v_1, v_2, \ldots, v_k$ and let H' be the set which contains vertices $v_1^1, v_1^2, ...$ (corresponding to a potential guard position), sorted according to the out-degree. We use the soft heap (Chazelle, 2000) data structure to store the values of set H'. The first guard is placed at a vertex with maximum out-degree. Let v_c^d be the vertex with max out-degree and g_c^d be the guard placed at this vertex. We remove all vertices $v_i \subseteq H$ if there is a directed edge from v_c^d to vertex v_i and then update the graph and continue placing the guards in a similar way. This is illustrated in Algorithm 1.

Algorithm 1 Greedy algorithm for guard placement

```
1: procedure RoofGuardSet(G, H, H')
2:     guardSet ← φ
3:     while H is not empty do
4:         v_c^d ← H'.extractMax
5:         Add v_c^d to guardSet
6:         for each vertex v_i ∈ G.Adj[v_c^d] do
7:             Remove vertex v_i from set H
8:             Remove all edges incident to v_i in graph G
```

In Algorithm 1, graph G has $O(k)$ vertices and $O(k^2)$ edges. The soft-heap allows for insertions in $O(\log 1/\epsilon)$ time while constant time is needed for updating and deletion (alternately, we can use a standard binary heap). We can use a linked list data structure to store the *guardSet*, with two pointers, *head* and *tail*. Set H contains the vertices corresponding to the buildings of the city and has size $O(k)$. The while loop in line 3 executes at most $O(k)$ times. Line 4 extracts a vertex (potential guard position) of maximum degree (maximum

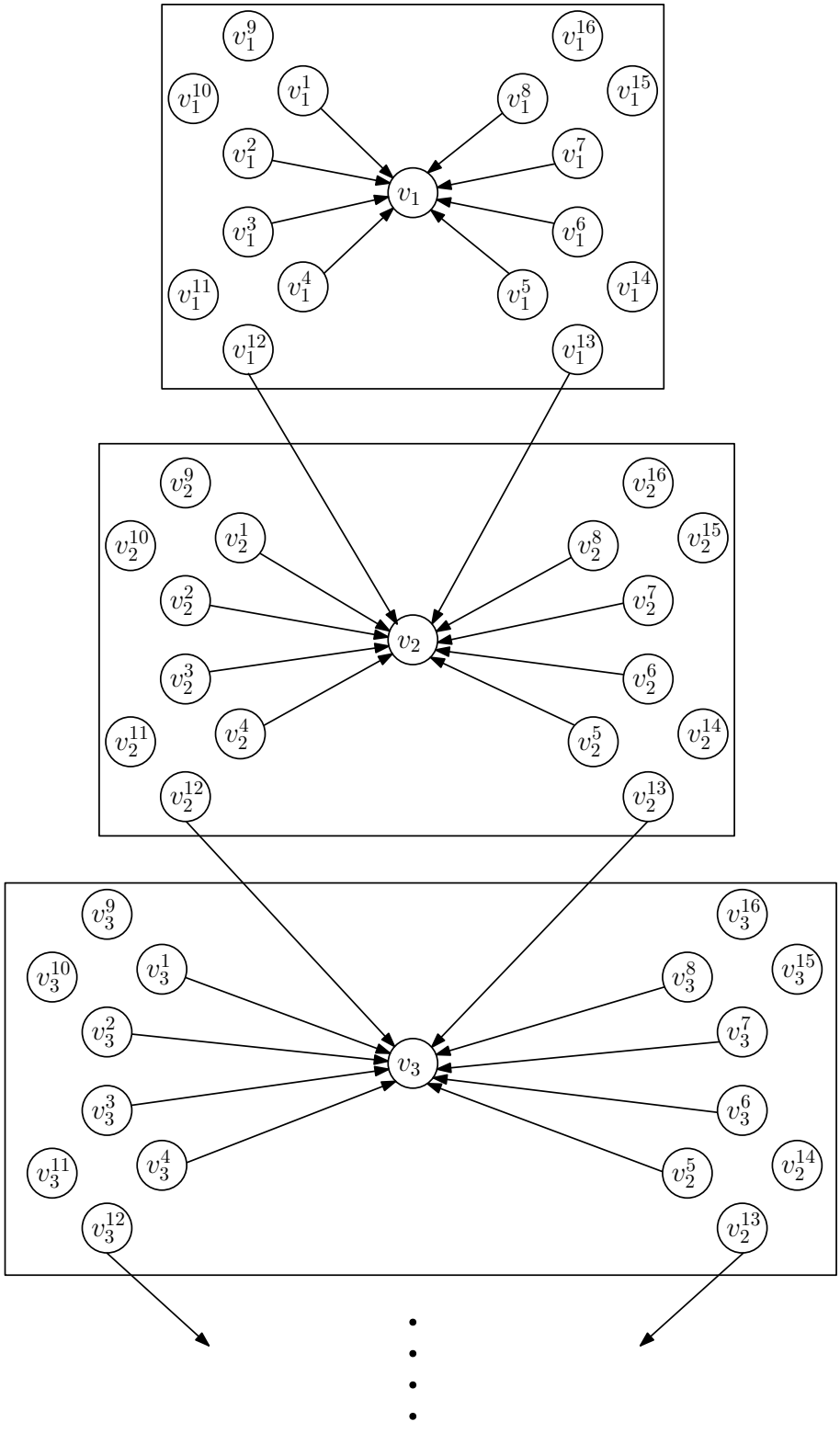


Figure 2.10: Graph corresponding to the structure in Figure 2.9

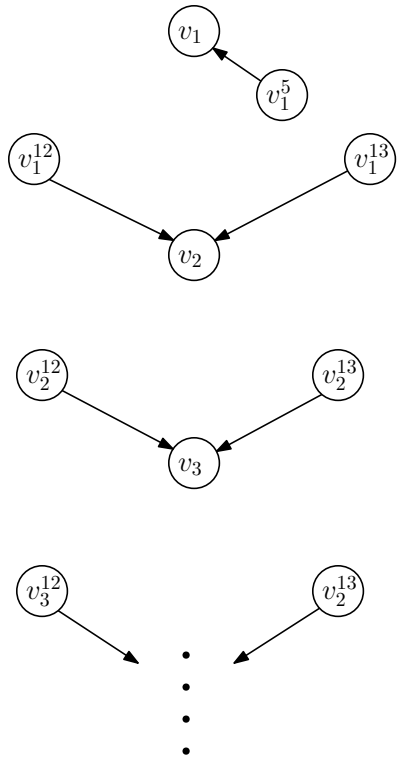


Figure 2.11: Optimized Graph

visibility) from set *H'* and adds it into *guardSet* in line 5. The time taken to extract the maximum value from *H'* is $O(1)$ and time taken to add this vertex to *guardSet* is $O(1)$. Therefore, the total time taken by steps 4 and 5 is $O(k)$. In line 7, we remove the vertices in H which are visible to the guard placed at v_c^d. Since the size of H is $O(k)$ the total time taken by step 7 is $O(k)$. In line 8, we remove all incident edges of vertices $v_i \in G.Adj[v_c^d]$, which can be done in $O(\sum_{i=1}^{b} deg_{v_i} G)$ where $v_1, v_2, \ldots, v_b \in G.Adj[v_c^d]$. Total time taken by step-8 is $O(k^2)$. The running time of Algorithm 1 is $O(k^2)$ and the space complexity is $O(k^2)$. The mentioned algorithm is an implementation of the greedy set cover algorithm. We get an $O(\log\ k)$ approximation on the minimum number of guards needed.

2.3.2 Ground and Wall Guarding

Vertically projecting the city on the ground results in a rectangle polygon with k rectangular holes. As mentioned earlier, the number of guards required to guard the walls and ground is no larger than the number of guards needed for the following problem:

SubProblem 1. *(V1)Given an axis-aligned rectangle P with k disjoint axis-aligned rectangular holes, place vertex guards on holes such that every point inside P is visible to at least one guard, where the range of vision of guards is* 180° *and each guard faces East, West, North, or South.*

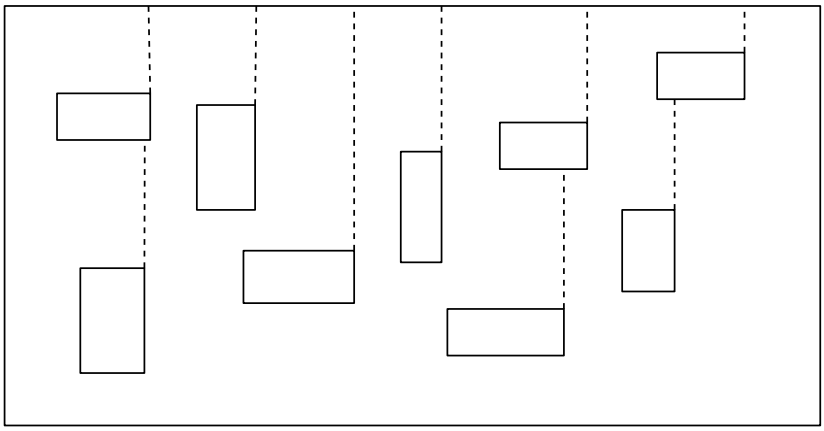
On the other hand, it is easy to see that a lower bound on the number of guards for Subproblem 1 can be used to obtain a lower bound for guarding the walls and the ground of a city with k rectangular buildings: map the holes to buildings of the same height.

It is worth noticing though that the two problems are not equivalent, that is, for a given input, fewer guards might be needed to guard the walls and ground than the number needed to guard the holes defined by projecting the buildings to the ground.

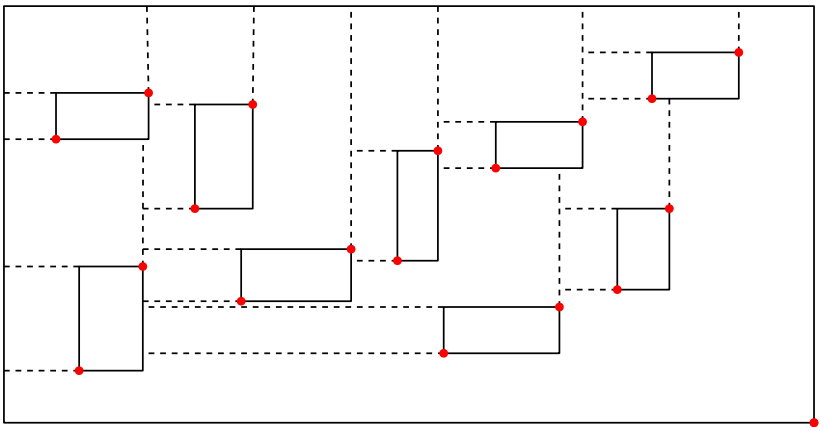
Observe that $2k + \lfloor \frac{k}{2} \rfloor + 2$ vertex guards, placed on holes, can be obtained from (Bao et al., 2008) by replacing a 360° guard with two 180° guards.

In what follows, we show how to improve this bound. For each hole, extend the right vertical edge in the upward (North) direction through the interior of the polygon until it encounters some horizontal edge of a hole or the outer rectangle. After extending the vertical edges, extend both horizontal edges of each hole in the left (West) direction through the interior of the polygon until it encounters some vertical edge or extended vertical edge of a hole or the outer rectangle (see Figure 2.12). The steps above divide the polygon into $2k+1$ shapes. Each shape corresponds to a monotone staircase (both in x and y-direction), as shown in Figure 2.12b. Only one guard is required to guard each staircase, placed at the South-East corner of the staircase, facing West. Guard positions are shown in Figure 2.12b.

Out of $2k+1$ guards, $2k$ guards are placed on the vertices of the holes while one guard is placed on a vertex of the rectangle P. Refer to Figure 2.12b for visual details.



(a) Extension of the right vertical edge of each hole until it encounters a horizontal edge.



(b) Extension of horizontal edges of each hole until they encounter a vertical edge. Guard positions are shown using red dots. All guards are facing West.

Figure 2.12: $2k+1$ guards are always sufficient to guard the walls and ground.

Theorem 14. $2k+1$ *guards are always sufficient to guard the walls and ground of a rectangular city with* k *disjoint axis-aligned rectangular buildings, with at most one guard placed at a corner of the bounding rectangle.*

If however, we do not allow a guard to be placed at a corner of the enclosing rectangle P, then the number of guards needed could increase significantly. In the rest of this section, we prove an upper bound on the number of vertex guards, placed only on vertices of the

holes (the setup in (Blanco et al., 1994; Bao et al., 2008)). To this end, we derive a worst-case upper bound on the number of vertex guards required to cover the portion of the city guarded by the guard placed at a vertex of the rectangle P.

Let S be the set of k buildings in the city, contained in the axis-aligned rectangle P defined by the points $[0, 0; x, y]$. Let x_s^M, x_f^M be the starting and finishing boundry sequence along x-axis and y_s^M, y_f^M be the starting and finishing boundry sequence along y-axis.

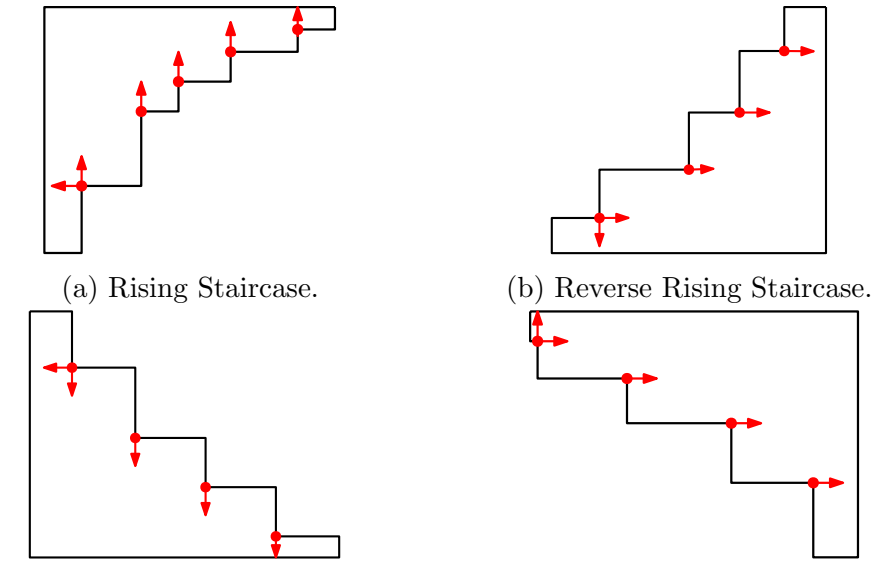
We define four types of staircases (see Figure 2.13):

1. Rising staircase (RS): $x_s^M = 0$, $y_f^M = y$, x_f^M is non decreasing along the positive y-axis, and y_s^M is non decreasing along the positive x-axis

2. Falling staircase (FS): $x_f^M = x$, $y_f^M = y$, x_s^M is non increasing along the positive y-axis, and y_s^M is non increasing along the positive x-axis

3. Reverse rising staircase (RRS): $x_f^M = x$, $y_s^M = 0$, x_s^M is non decreasing along the positive y-axis, and y_f^M is non decreasing along the positive x-axis

4. Reverse falling staircase (RFS): $x_s^M = 0$, $y_s^M = 0$, x_f^M is non increasing along the positive y-axis, and y_f^M is non increasing along the positive x-axis



(a) Rising Staircase. (b) Reverse Rising Staircase.

(c) Reverse Falling Staircase (d) Falling Staircase

Figure 2.13: Type of staircases and placement of guards

A rising staircase is constructed as follows: extend the horizontal edges of each hole towards the right (East) direction, then extend the vertical edges towards the South direction. The closed orthogonal polygon formed by the top edge and the left edge of P, and the extended edges of the holes, corresponds to a rising staircase. Falling, reverse rising, and reverse falling staircases are constructed similarly. Note that for each staircase, a reflex vertex corresponds to a vertex of a hole. Thus, the number of buildings involved in the construction of a staircase is equal to the number of reflex vertices on the staircase.

We find the staircase comprising the minimum number of buildings. WLOG assume the staircase involving the minimum number of buildings is RRS (otherwise, we can rotate the input, so the staircase corresponds to RRS). For this staircase, place a guard on each reflex vertex, facing right (East), and an additional guard on the first (bottom) stair, with the guard facing down (South), as shown in Figure 2.13. The number of guards required to cover the staircase is one more than the number of stairs in it. In the worst case, each of the four staircases must have the same number of stairs, otherwise we can use one with the smallest number as RRS.

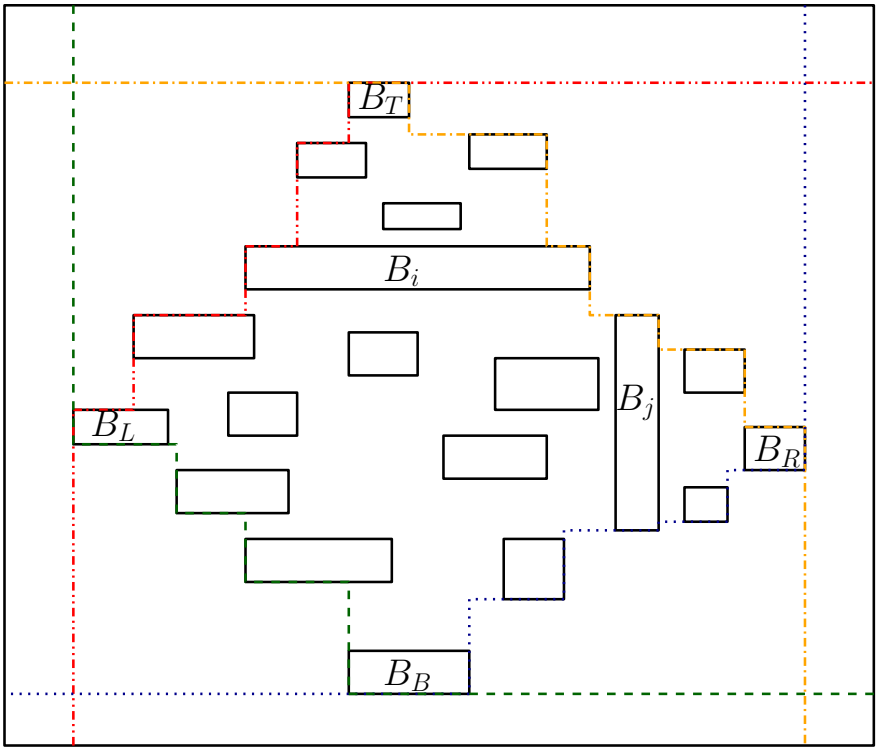


Figure 2.14: Staircases RS (dash dot / red), FS(dash dot / orange), RRS (dotted / blue) and RFS(dashed / green). Buildings above B_i lie in its vertical span and buildings right of B_j lie in its horizontal span.

In what follows, we provide a divide and conquer approach to find an upper bound on the number of guards. For some building B, the vertical span of B is the parallel strip defined by the vertical sides of B and containing B. The horizontal span is defined accordingly. Let B_L be the leftmost building, B_T be the topmost building, B_R be the rightmost building, and B_B be the bottom-most building of the city. Assume two adjacent staircases, say RS and FS, share the same building $B_i \notin \{B_L, B_T, B_R, B_B\}$. Then, all buildings that lie in the upper half-plane defined by the line supporting the upper horizontal edge of B_i (buildings above B_i) are in the vertical span of B_i(see Figure 2.14). Similarly, if staircases RRS and FS include the walls of the same building $B_j \notin \{B_L, B_T, B_R, B_B\}$, then all buildings that lie on the right of B_j are in the horizontal span of B_j.

A building B_i is called an *internal building* if $B_i \notin \{B_L, B_T, B_R, B_B\}$. The pair of staircases (i) RS, FS (ii) FS, RRS (iii) RRS, RFS and (iv) RFS, RS are called *adjacent staircases* while the pairs (v) RS, RRS, and (vi) FS, RFS are called *opposite staircases*.

Consider the four staircases RS, FS, RFS, and RRS. We can have four cases:

Case 1: There does not exist any internal building that is shared by either adjacent staircases or opposite staircases.

Case 2: There exists an internal building that is shared by opposite staircases, and none of the internal buildings are shared by adjacent staircases.

Case 3: There exists an internal building that is shared by adjacent staircases, and none of the internal buildings are shared by opposite staircases.

Case 4: There exists an internal building shared by a pair of adjacent staircases and a pair of opposite staircases.

Before addressing the case above, we notice it is possible that, from the set of building pairs $(B_L, B_T), (B_T, B_R), (B_R, B_B)$, and (B_B, B_L), the pair in one of the sets corresponds to the same building. In this situation (call it Case 0), one of the staircases consists of only one stair, and two guards are required to guard the staircase (see Figure 2.15). Using a placement of guards like in Theorem 14, $2k + 2$ guards are required to cover the walls and ground of such a rectangular city. Thus, from now on, we assume this is not the case.

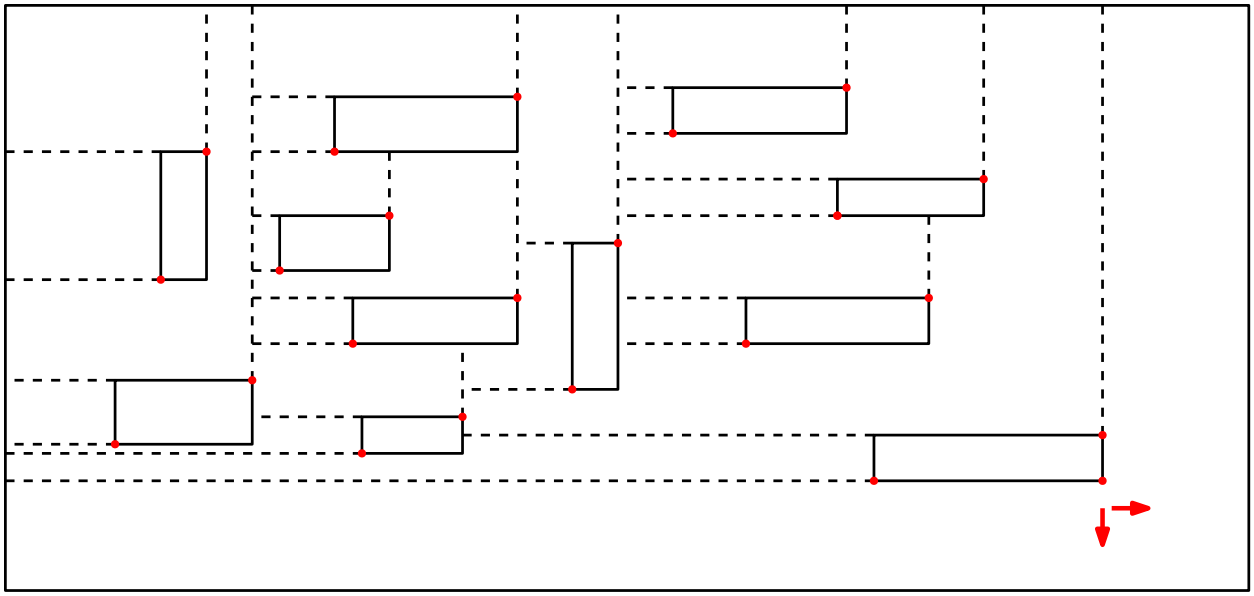


Figure 2.15: Situation where B_R and B_B correspond to the same building

Case 1: No adjacent pair of staircases share the same internal building, and no opposite pair of staircases share the same building.

WLOG assume that RRS contains the minimum number of stairs; place the guards in a similar fashion as in Theorem 14. Overall, we place two guards on each building and the rest on the reflex vertices of the staircase RRS.

The upper bound on the number of vertex guards required to cover the staircase of P is achieved when the number of buildings involved in the construction of each staircase is the same. Let the staircases RS and RRS contain δ distinct buildings. Staircase FS contains $\delta - 2$ distinct buildings because building B_T is already counted in staircase RS and building B_R is counted in staircase RRS. Similarly, RFS contains $\delta - 2$ distinct buildings. Note that $\delta + \delta + \delta - 2 + \delta - 2 = k$ and thus $\delta = \lfloor \frac{k}{4} \rfloor + 1$. Therefore, to guard each staircase, we require $\delta + 1 = \lfloor \frac{k}{4} \rfloor + 1 + 1 = \lfloor \frac{k}{4} \rfloor + 2$ guards.

We placed 2 guards on each building and $\lfloor \frac{k}{4} \rfloor + 2$ guards to cover the staircase. Therefore, $2k + \lfloor \frac{k}{4} \rfloor + 2$ guards are required (sufficient) to cover the walls and ground.

Case 2: At least one pair of opposite staircases shares the same building, and no adjacent staircases share the same interior building.

Let the staircases RS and RRS share a building B_i. Refer to Figure 2.16 and note that extending the top edge of B_i to the left until it hits P will not result in an intersection with the other buildings. Similarly, extending the bottom edge of B_i to the right until it hits P will not result in an intersection with the other buildings. We divide the city into two sub-cities, $city_1$ and $city_2$ (green and orange boundaries in Figure 2.16), by extending the top edge of B_i towards left and the bottom edge towards the right. Let B_i be included in both sub-cities.

All buildings in $city_1$ lie either above or towards the right of B_i. We place two guards on each building, one at the North-West corner and one at the South-East corner, both facing East. We further place a third guard on the North-West corner of B_i, facing West. All buildings in $city_2$ lie either below or towards the left of B_i. We place two guards on each building, one at the South-East corner and one at the North-West corner, both facing West. We further place an additional guard on the South-East corner of B_i facing East. In total, we have placed six guards on B_i. However, two guards are duplicates, so we only have four guards on B_i. Thus, $2k + 2$ guards are required (sufficient) to cover the walls and ground of the city.

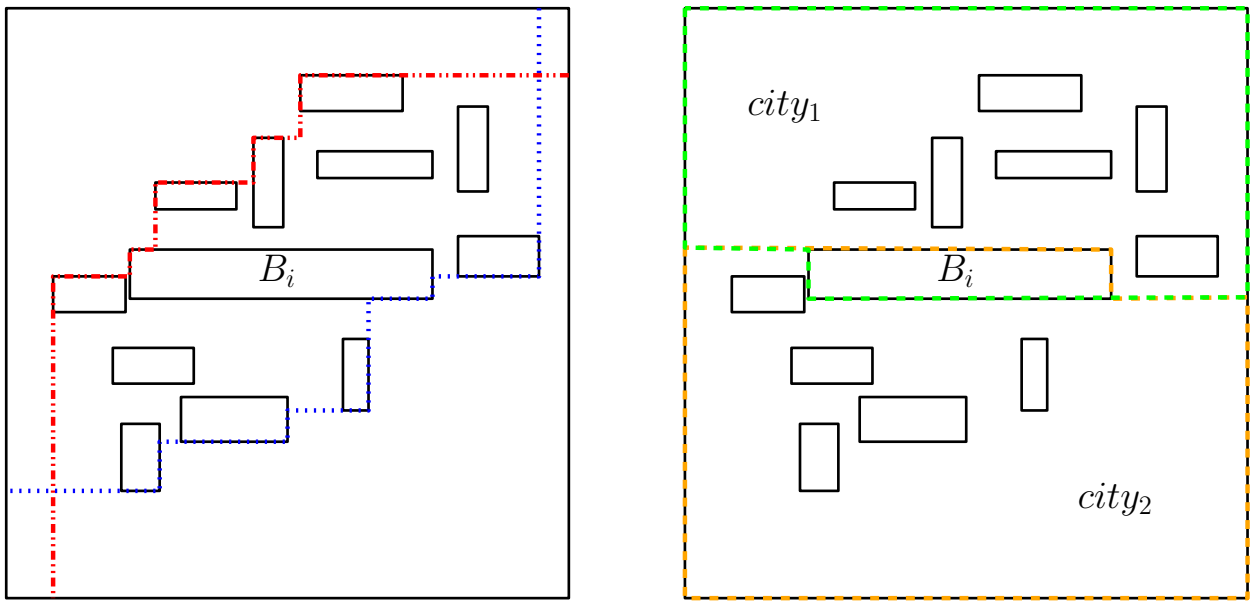


Figure 2.16: Building B_i is shared by staircases RS and RRS. Staircase RS is shown in dash dot / red color and staircase RRS is shown in dotted / blue color.

Case 3: At least one pair of adjacent staircases shares the same interior building, and no pair of opposite staircases shares the same building.

We use the following recursive approach to compute an upper bound on the number of guards. Let B_i be a building shared by two adjacent staircases, say RS and FS, as shown in Figure 2.17a. Let α_i be the number of buildings that lie above B_i and β_i be the number of buildings (excluding B_i), that lie in the lower half-plane defined by the line supporting the upper horizontal edge of B_i. Note that $\alpha_i + \beta_i + 1 = k$. Let C be the set containing all such buildings B_i (walls included in more than one staircase). Let $B_j \in C$ the the building that minimizes the value $|\alpha_j - \beta_j|$, such that $\alpha_j, \beta_j \geq 3$.

If such building does not exist then each building B_j in set C has $\alpha_j \leq 3$ or $\beta_j \leq 3$. We can use a similar argument as the one discussed in Case 2. Recall that in the worst case all staircases should have an equal number of buildings/stairs. It is easy to notice that there exist at most three buildings shared by a staircase pair (i) RS, FS (ii) FS, RRS (iii) RRS, RFS, or (iv) RFS, RS, as $\alpha_i < 3$ or $\beta_i < 3$. Let each of RS, RRS contain δ distinct

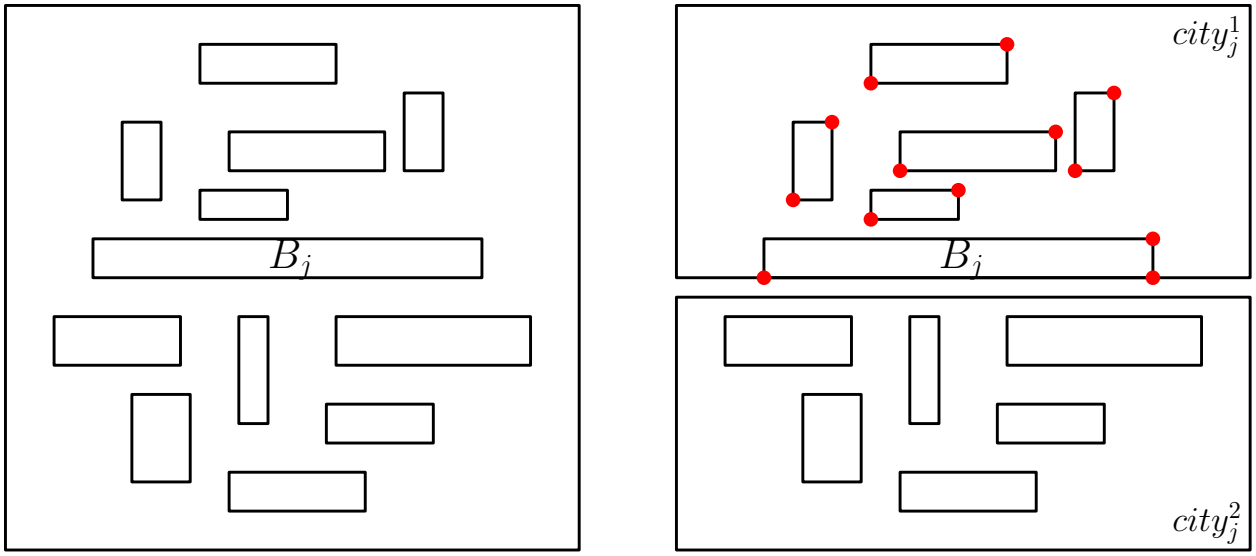
buildings. Staircases FS, RFS contain $\delta - 6$ additional buildings, as three buildings are included in each of RS and RRS. There are k buildings, $2 \times \delta + 2 \times (\delta - 6) = k$, thus $4 \times \delta - 12 = k$, and $\delta = \lfloor \frac{k}{4} \rfloor + 3$. To guard the staircase we need at most $\lfloor \frac{k}{4} \rfloor + 4$ guards, resulting in $2k + \lfloor \frac{k}{4} \rfloor + 4$ guards overall.

If there exists a building B_j such that $\alpha_j, \beta_j \geq 3$, we proceed as follows.

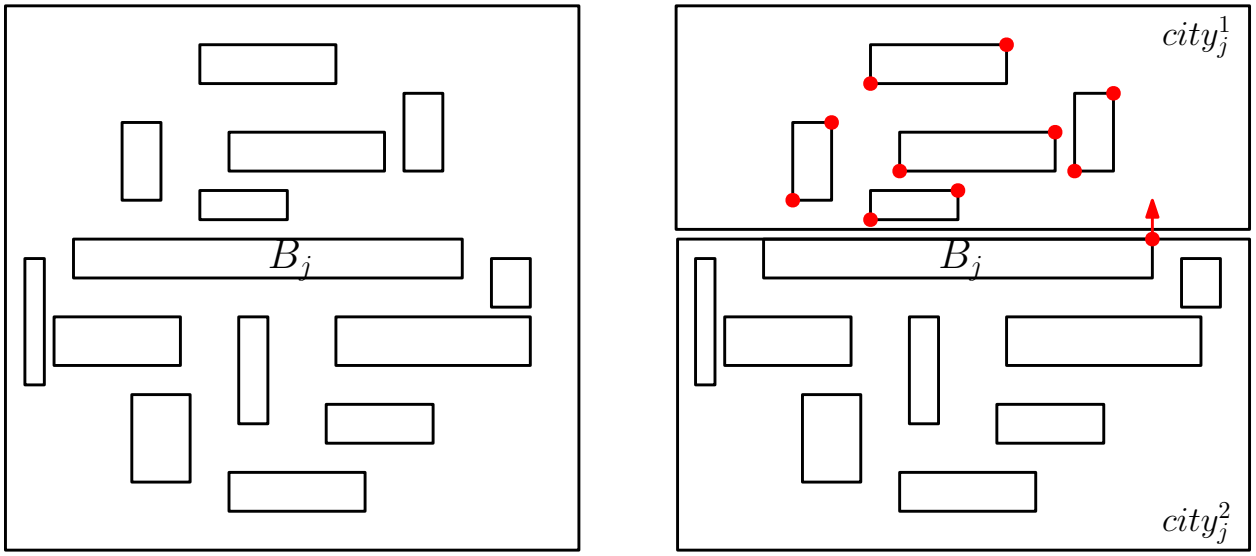
Let the staircases RS and FS share building B_j. There can be two cases: (i) there are no buildings within the horizontal span of B_j (Refer Figure 2.17a) and (ii) there exist buildings within the horizontal span of B_j (Refer to Figure 2.17b).

Consider the first case where none of the buildings lie within the horizontal span of B_j. We divide the city into two sub-cities, $city_j^1$ with α_j buildings and $city_j^2$ with β_j buildings. All the buildings in one of the sub-cities lie inside the vertical span of B_j. Let this sub-city be $city_j^1$. We add B_j to $city_j^1$, which results in a total of $(\alpha_j + 1)$ buildings in $city_j^1$, and follow Case 0, which results in a total of $2(\alpha_i + 1) + 1$ guards to guard the walls and ground of this sub-city, as shown in Figure 2.17a. For $city_j^2$, we consider the Case it falls in and places the guards accordingly. For the placement of guards, we treat the two sub-cities as independent cities. It is important to notice that only one of $city_j^1$ and $city_j^2$ above can be in Case 3, while the other one is in Case 0. Thus, only one of the two cities could need further divisions.

Assume Case 3 keeps occurring, and we need to divide the city m times. During each division, the sub-city with corresponding building B_j contains greater than or equal to four buildings and one additional guard is required to guard such sub-city. Let the first division divide the city into two sub-cities with $k_1, k - k_1$ buildings each. The second division splits the sub-city with $k - k_1$ buildings into $k_2, k - k_1 - k_2$ buildings and so on, down to sub-cities with $k_m, k - \sum_{i=1}^{m} k_i$ buildings. Each resulting sub-city does not need to be divided further. Let $k' = \sum_{i=1}^{m} k_i$. For each sub-city with $k_1, k_2, \ldots k_m$ buildings, we require $2k_i + 1$ guards, where $k_i \geq 4$, and for the last sub city we need at most $4 + \lfloor (k - k')/4 \rfloor$ additional guards. Thus, the total number of guards required to guard the city is:



(a) No buildings lies within the horizontal span of B_j



(b) Buildings lie within the horizontal span of B_j

Figure 2.17: (a) Building B_j is shared by RS and FS. (b) Divide city into two sub-cities, $city_j^1$ and $city_j^2$.

$2k_1 + 1 + 2k_2 + 1 + \cdots + 2k_m + 1 + 2(k - k') + 4 + \lfloor \frac{k-k'}{4} \rfloor = 2k + m + 4 + \lfloor \frac{k-k'}{4} \rfloor \leq$

$2k + 4 + k'/4 + \lfloor \frac{k-k'}{4} \rfloor \leq 2k + \lfloor \frac{k}{4} \rfloor + 4$

Consider the second case where buildings lie within the horizontal span of B_j. We divide the city into two sub-cities, $city_j^1$ with α_j buildings and $city_j^2$ with β_j buildings, such that all

buildings in one of the sub-cities lie inside the vertical span of B_j. Let this sub-city be $city_j^1$. We add B_j to $city_j^2$, which results in a total of $(\beta_j + 1)$ buildings in $city_j^2$. For the placement of guards, $city_j^1$ is in Case 0 and we place two guards on each building of $city_j^1$, and one guard on building B_j facing towards $city_j^1$, which results in a total of $2(\alpha_i) + 1$ guards to cover the walls and ground of $city_j^1$, as shown in Figure 2.17b. For $city_j^2$, we consider the case it falls in and place the guards accordingly. For the placement of guards, we treat the two sub-cities as independent cities. Using a similar explanation as in the first case, we obtain the upper bound of $2k + \lfloor \frac{k}{4} \rfloor + 4$.

Case 4: At least one pair of opposite staircases and one pair of adjacent staircases share the same building.

We place guards according to Case 2 and conclude that $2k + 2$ guards are sufficient to cover the walls and ground of the city.

Noticing that the derived upper bound then holds when the vertex guards are allowed to have *arbitrary* orientation we obtain:

Theorem 15. *$2k + \lfloor \frac{k}{4} \rfloor + 4$ guards are always sufficient to guard the walls and ground of a rectangular city with k disjoint axis-aligned rectangular buildings, with all guards placed on vertices of the buildings.*

We also obtain the following Theorem, restricting the visibility of guards in (Blanco et al., 1994) to 180°:

Theorem 16. *Given k pairwise disjoint isothetic rectangles in the plane, $2k + \lfloor \frac{k}{4} \rfloor + 4$ vertex guards are always sufficient to guard the free space.*

2.3.3 City Guarding

Theorem 17. *Given a city with k disjoint axis-aligned buildings within an axis-aligned rectangle P, the number of cameras with 180° range of vision needed to guard the city (roofs,*

walls, and ground) is upper bounded by (i) $2k + \lfloor \frac{k}{4} \rfloor + 4$ *when cameras are placed on buildings only, and (ii)* $2k + 1$ *when at most one camera can be placed at a corner of* P*.*

Proof. The solution described earlier in Section 2.3.2 established either $2k + \lfloor \frac{k}{4} \rfloor + 4$ guards in case (i) and or $2k + 1$ guards in case (ii) to cover the ground and the walls of the city. In both (i) and (ii), on each building, we place at least two guards, on diagonal corners, facing in the same direction. Thus, one of these two guards also guards the roof of the building. Therefore, $2k + \lfloor \frac{k}{4} \rfloor + 4$ guards are always sufficient to guard the city in case (i) and $2k + 1$ in case (ii). □

To find a guard set for (i) Roof guarding, (ii) Ground and Wall Guarding, and (iii) City Guarding, we can use the $O(\log n)$ greedy approximation for the set-cover problem. For roof guarding, we have implemented the optimized graph-based version described earlier in Algorithm 1.

2.4 City Guarding with Arbitrary Oriented Rectangle Buildings

Given a rectangular city with k vertical buildings, each having a rectangular base, the goal is to place the minimum number of cameras with 180° range of vision, at the top corners (vertices) of the buildings to guard the city (roofs, walls, and ground). In all our proofs each guard is aligned with a wall of the building it is placed on, similar to the axis-aligned version.

2.4.1 Roof Guarding

The same city structure in Theorem 13 leads to:

Theorem 18. *Given a city with k disjoint rectangular buildings, k vertex guards are always sufficient and sometimes necessary to guard the roofs.*

2.4.2 Ground and Wall Guarding

As before, the problem of guarding the ground and the walls reduces to:

SubProblem 2. *(V3) Given an axis-aligned rectangle P with k disjoint rectangular holes, place vertex guards on holes such that every point inside P is visible to at least one guard, where the range of vision of guards is* $180°$.

O'Rourke (O'Rourke, 1983b) proved that $\lfloor \frac{n+2k}{3} \rfloor$ vertex guards are always sufficient to guard any polygon with n vertices and k holes when the range of vision is $360°$. We can replace a $360°$ guard with at most two guards with $180°$ range of vision. Thus, $4k+2$ vertex guards are always sufficient to guard a rectangular polygon P with k disjoint rectangular holes when guards may be placed at the corners of P.

Our main result is stated in the following theorem.

Theorem 19. $3k+1$ *vertex guards are sometimes necessary to guard a rectangular polygon P with k disjoint rectangular holes, where guards are placed only at the corners of the holes. We conjecture the bound is tight.*

Proof. Consider the input in Figure 2.18, with the following properties:

1. B_i lies within the span of B_j $, \forall j < i$.

2. None of the edges of B_i is partially or completely visible from any vertex of B_j, $\forall\ j < i-1$ and from any vertex of B_m, $\forall\ m > i+1$.

3. From each potential position of a vertex guard on B_i, the guard is able to see at most one edge of B_{i+1}.

4. There is no guard position on B_i from where an edge of B_{i-1} and an edge of B_i are visible.

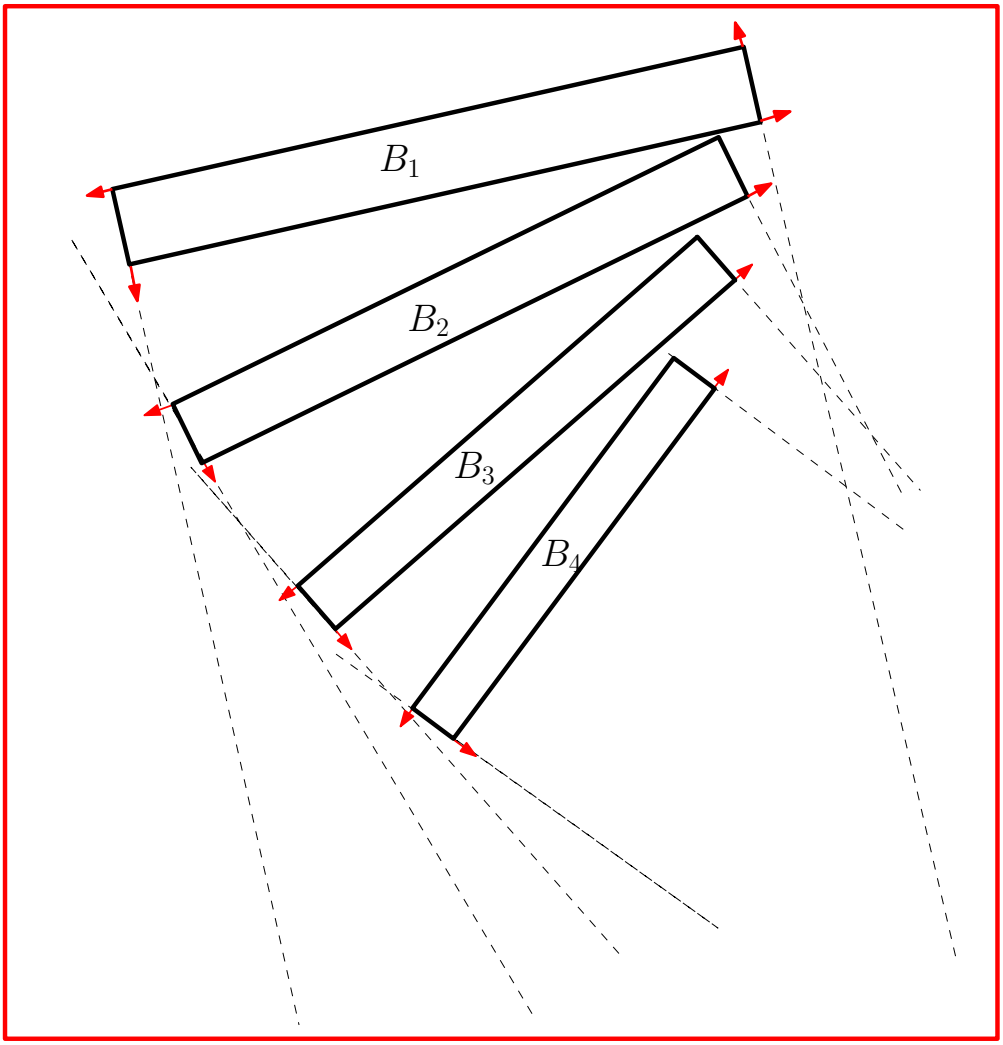


Figure 2.18: City Structure where $3k + 1$ guards are necessary to guard the polygon

Consider the space $\wp_i$ between two consecutive holes B_i and B_{i+1}, as shown in Figure 2.19. Because of property 2, $\wp_i$ is not visible to any guard placed on hole B_j where $j \in [1, i) \cup (i + 1, k]$. Therefore, $\wp_i$ is only visible to the guards either placed on B_i or B_{i+1}. It is easy to notice that there are twelve possible guard positions on B_i and B_{i+1} from where $\wp_i$ is visible (partially from each positions, see Figure 2.19), and these guards cover three walls, one wall of B_i and two walls of B_{i+1}. Note that these guards do not cover any other wall (partially or entirely), and the mentioned three walls are not visible (partially or entirely) to any other potential guard. Out of twelve possible guard positions, the minimum number of guards

required to guard $\wp_i$ is two (either both placed on B_i or one placed on B_i and another on B_{i+1}. Therefore the space between any two consecutive holes is only guarded by the guards placed on these holes, and the minimum number of guards required to guard such space is two.

Consider the left wall, w_L^i of hole B_i. Because of the structure of the city, w_L^i is not visible to any guard placed on B_j for $j \neq i$. Hence, w_L^i can only be guarded by a guard placed on B_i, and there are four possible guard positions from where w_L^i is visible(see Figure 2.19). However, none of these guards positions cover any other wall in the city. Therefore, we need one guard to cover the left wall of each hole.

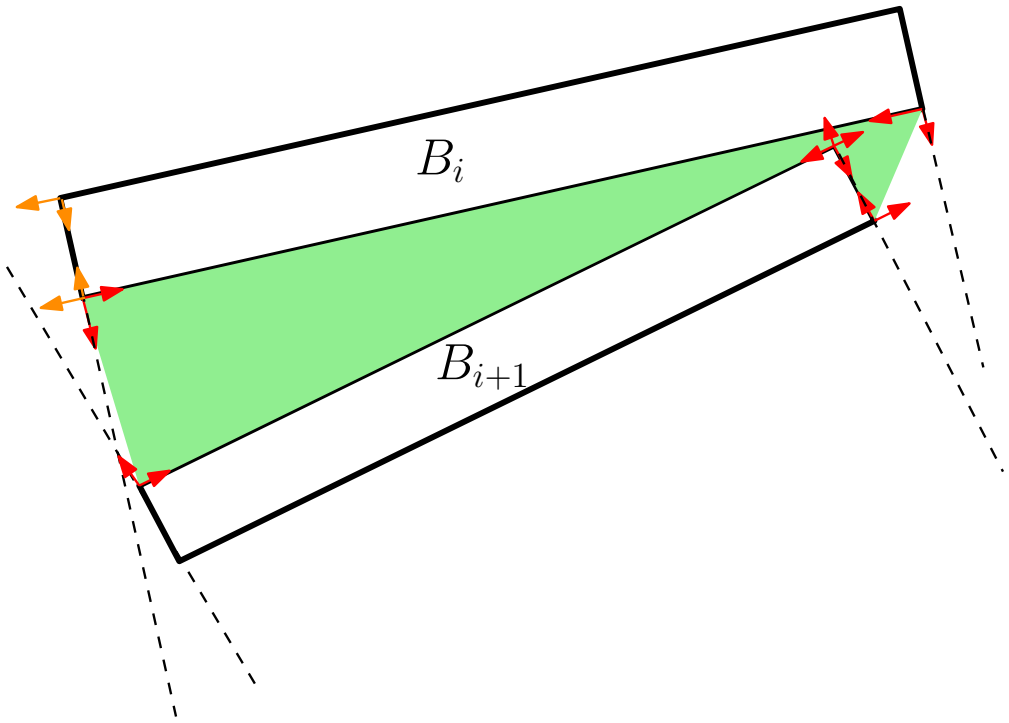


Figure 2.19: $\wp$ is shaded in green. Potential guard positions to cover $\wp$ are shown in red and potential guard positions to cover the w_L^i is shown in orange.

There are $k-1$ spaces $\wp_i$ in total, between consecutive holes ($i = 1, 2, \ldots, k-1$). Thus, $2(k-1)$ guards are needed to guard their union. As argued, each left edge of a hole needs an

additional guard, resulting in $3k-2$ guards. The top and right edges of B_1 and the bottom edge of B_k are not guarded, so in total at least $3k+1$ guards are needed.

A $3k+1$ guard placement, for $k=4$, is shown in Figure 2.18 and is obtained as follows. We start placing guards on B_1. Three of its walls (left, top and right) are not visible by any potential guard placed on B_i $, \forall i > 1$ (property 1). We place three guards to cover these walls. Consider the space $\wp_1$ between B_1 and B_2. We need two guards to cover $\wp_1$; let one of these guards be placed on B_1 and the other on B_2 (see Figure 2.18). After placing these two guards, all walls of B_1 are visible, and two walls of B_2 (top and right) are visible. We need an additional guard to cover the left wall of B_2, and this guard does not cover any other wall in the city. Consider now the space $\wp_2$ between the hole B_2 and B_3 and place two guards to cover $\wp_2$; let one of these guards be placed on B_2 and the other on B_3. After placing these two guards, all walls of B_1 and B_2 are guarded, and two walls of B_3 are guarded. We placed four guards on B_1 and three guards on B_2. We continue this process, and three guards are required to guard each hole B_i $\forall i > 2$. This results in $3k+1$ guards as we place three guards on each hole B_i $, \forall i \in (1,k]$, and four guards on B_1. Guard locations and directions are shown in Figure 2.18. □

2.4.3 City Guarding

Theorem 20. $3k+1$ *vertex guards are sometimes necessary to guard a city with* k *vertical buildings with rectangular base, where guards are placed only at the top vertices of the buildings. We conjecture the bound is tight.*

The proof follows immediately from Theorem 19.

2.5 City Guarding Problem for Axis-Aligned Orthogonal Buildings

Given a rectangular city with k disjoint axis-aligned vertical buildings, each having an orthogonal base, the goal is to place the minimum number of cameras that can see only the half-space in front of them (denoted as 180° range of vision), at the top corners (vertices) of the buildings to guard the city (roofs, walls, ground, and aerial space). Thus, when a guard (camera) is aligned with a wall of the building it is placed on, the half-space seen by the guard is bounded by a vertical plane containing that wall.

2.5.1 Roof Guarding

The same city structure in Theorem 13 leads to:

Theorem 21. *Given a city with k disjoint axis-aligned vertical buildings, each having an orthogonal base, k vertex guards are always sufficient and sometimes necessary to guard the roofs.*

2.5.2 Ground and Wall Guarding

Vertically projecting the city on the ground results in a rectangle polygon with k orthogonal holes. As mentioned earlier, the number of guards required to guard the walls and ground is no larger than the number of guards needed for the following problem:

SubProblem 3. *(V2) Given an axis-aligned rectangle P with k disjoint axis-aligned orthogonal holes, $H_1, H_2, \ldots, H_k$ with $m_1, m_2, \ldots, m_k$ vertices respectively such that $\sum_{i=1}^{k} m_i = m$, place vertex guards on holes such that every point inside P is visible to at least one guard, where the range of vision of guards is* 180°.

In all our proofs each guard faces East, West, North, or South only.

Each vertex of an orthogonal polygon is either 270° (reflex) or 90° (convex) which is defined as follows: For each orthogonal hole H_i, pick an arbitrary vertex, say u, and start

traversing H_i in clockwise direction. For a vertex $v \in H_i$, if the turn made at v while traversing H_i is right then v is a *reflex* vertex, else v is *convex* vertex. For each hole H_i, let r_i and c_i be the total reflex and convex vertices. From (O'Rourke, 1987), $r_i = \frac{m_i+4}{2} = \frac{m_i}{2} + 2$ and $c_i = \frac{m_i-4}{2} = \frac{m_i}{2} - 2$. Note that all the holes are orthogonal and every orthogonal polygon has even number of vertices. Therefore, $\forall i \in [1, k]$, the value of $\frac{m_i}{2}$ is an integer.

Lemma 1. *(Urrutia, 2004) Given an axis-aligned rectangle P with k disjoint axis-aligned orthogonal holes, P can be guarded by placing one guard at each of the reflex vertices of holes.*

To guard the polygon P, traverse each hole H_i in clockwise direction. While traversing each edge $(u, v) \in H_i$, if v is a reflex vertex, we place a guard at v facing away from edge (u, v) (guard facing away from edge (u, v)).

Total guards required is equal to the summation of the total reflex vertices on each hole.

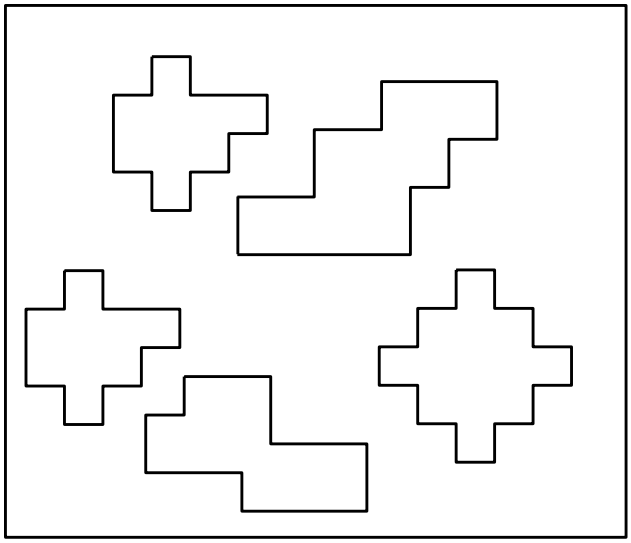
$$\textit{Total reflex vertices} = \sum_{i=1}^{k} r_i = \sum_{i=1}^{k} (\frac{m_i}{2} + 2) = \sum_{i=1}^{k} \frac{m_i}{2} + \sum_{i=1}^{k} 2 = \frac{m}{2} + 2k \tag{2.1}$$

Similarly,

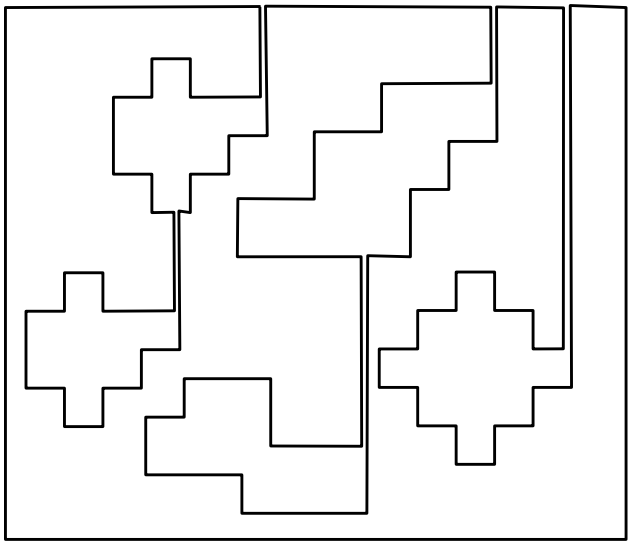
$$\textit{Total convex vertices} = \frac{m}{2} - 2k$$

From Theorem 3, $\lfloor \frac{r}{2} \rfloor + 1$ vertex guards are always sufficient and sometimes necessary to cover the interior of an orthogonal polygon with r reflex vertices. We can convert the given orthogonal polygon with k axis-aligned orthogonal holes and r reflex vertices into an orthogonal polygon without holes using O'Rourke's (O'Rourke, 1987) method such that the resultant polygon has $r - k$ reflex vertices (see Figure 2.20).

Theorem 22. *Given an axis-aligned rectangle polygon P with k disjoint axis aligned orthogonal holes with a total of m vertices, the number of guards with 180° range of vision needed to guard P is upper bounded by $\frac{m}{2} + k + 2$, with all guards placed at vertices of the holes.*



(a) A rectangular polygon with axis-aligned orthogonal holes



(b) An orthogonal polygon without holes

Figure 2.20: Converting an orthogonal polygon with five axis-aligned orthogonal holes into an orthogonal polygon without holes

In what follows, we show how to improve the above bound by using an extension of the Algorithm discussed in Section 2.3.2. Initially assume that each hole H_i is xy-monotone.

For each hole H_i, let E_R^i be the rightmost edge, E_L^i be the leftmost edge, E_T^i be the topmost edge, and E_B^i be the bottom-most edge. First we perform the following two steps (shown in Figure 2.21):

1. Traverse H_i from E_R^i to E_L^i in an counter-clockwise direction and extend each encountered vertical edge (including E_L^i and E_R^i) towards North (upward) direction until it hits the polygon P or any other hole. This process is shown in Figure 2.21a.

2. Traverse H_i from E_L^i to E_B^i in an counter-clockwise direction and extend each horizontal edge (including E_B^i) towards West (left) direction until it hits the polygon P, any other hole, or any of the extended vertical edges. This process is shown in Figure 2.21b.

The above steps divide the polygon into shapes where each shape corresponds to a monotone staircase (both in x and y direction), and one guard, placed at the South-East corner of each shape, is required to guard each shape. The total number of guards needed to guard the polygon is equal to the number of shapes constructed while performing the two steps.

For each hole H_i, we define the following staircases (shown in Figure 2.22):

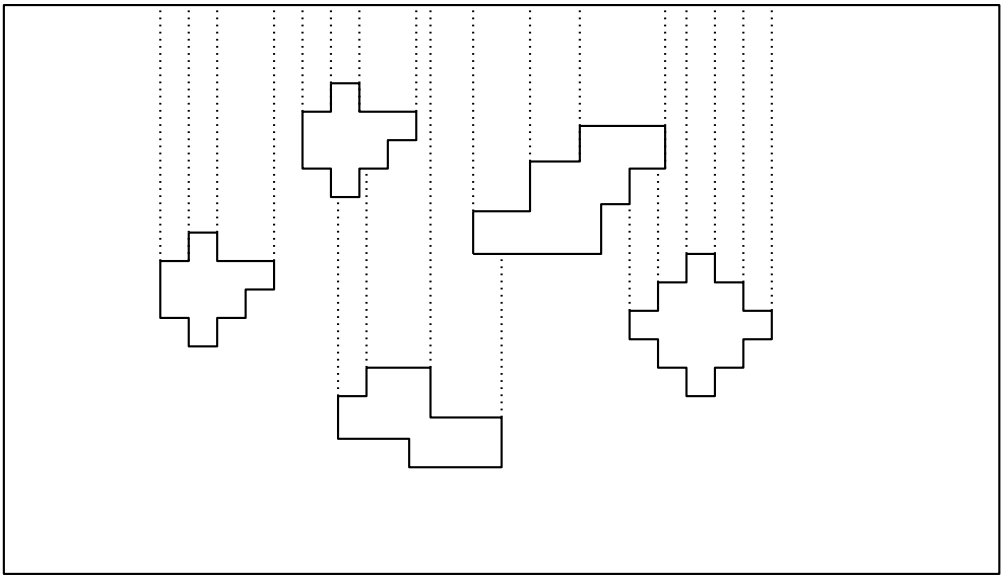
1. Rising Staircase RS_i: Monotone staircase formed by traversing H_i in clockwise order from E_L^i to E_T^i (including E_L^i and E_T^i).

2. Falling Staircase FS_i: Monotone staircase formed by traversing H_i in clockwise order from E_T^i to E_R^i (including E_T^i and E_R^i).

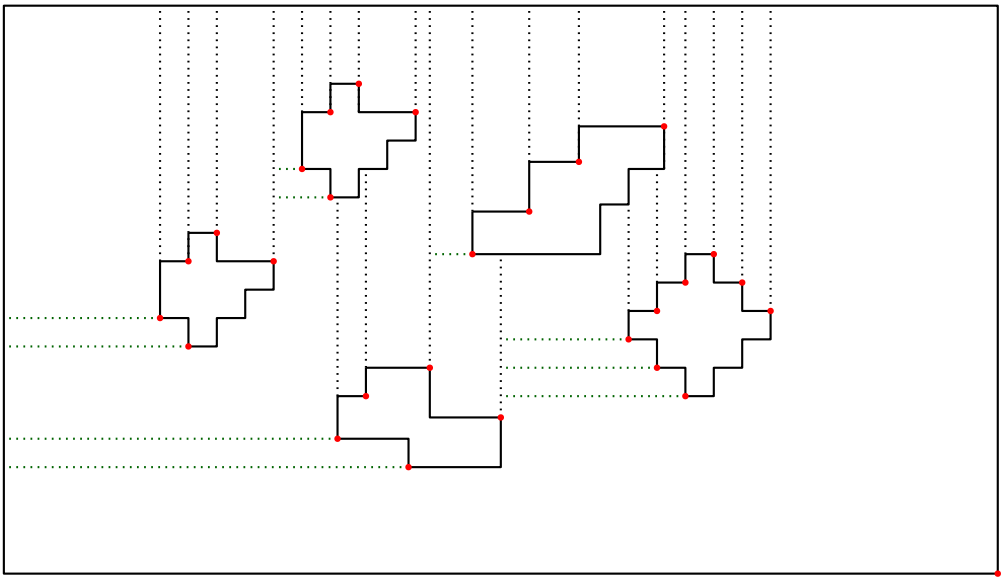
3. Reverse Rising Staircase RRS_i: Monotone staircase formed by traversing H_i in clockwise order from E_R^i to E_B^i (including E_R^i and E_B^i).

4. Reverse Falling Staircase RFS_i: Monotone staircase formed by traversing H_i in clockwise order from E_B^i to E_L^i (including E_B^i and E_L^i).

For hole H_i and staircase RS_i, let RS_i^r be the number of reflex vertices (excluding the first and last vertex of RS_i), RS_i^c be the number of convex vertices and RS_i^s be the number of stairs on staircase RS_i. Similarly, we define $FS_i^r, FS_i^c, FS_i^s, RFS_i^r, RFS_i^c, RFS_i^s, RRS_i^r, RRS_i^c$, and RRS_i^s



(a) For each hole H_i, extend each vertical edge encountered while traversing H_i in anti-clockwise direction from E_R^i to E_L^i (including edges E_R^i and E_L^i) in North (upward) direction until it hit any other hole or the polygon P.



(b) For each hole H_i, extend each horizontal edge encountered while traversing H_i in anti-clockwise direction from E_L^i to E_B^i(including R_B^i) in West(left) direction until it hit any other hole or the polygon P or any of the extended horizontal edges.

Figure 2.21: The process to divide the polygon P into shapes (monotone staircases). One guard is required to guard each shape (shown with red dot).

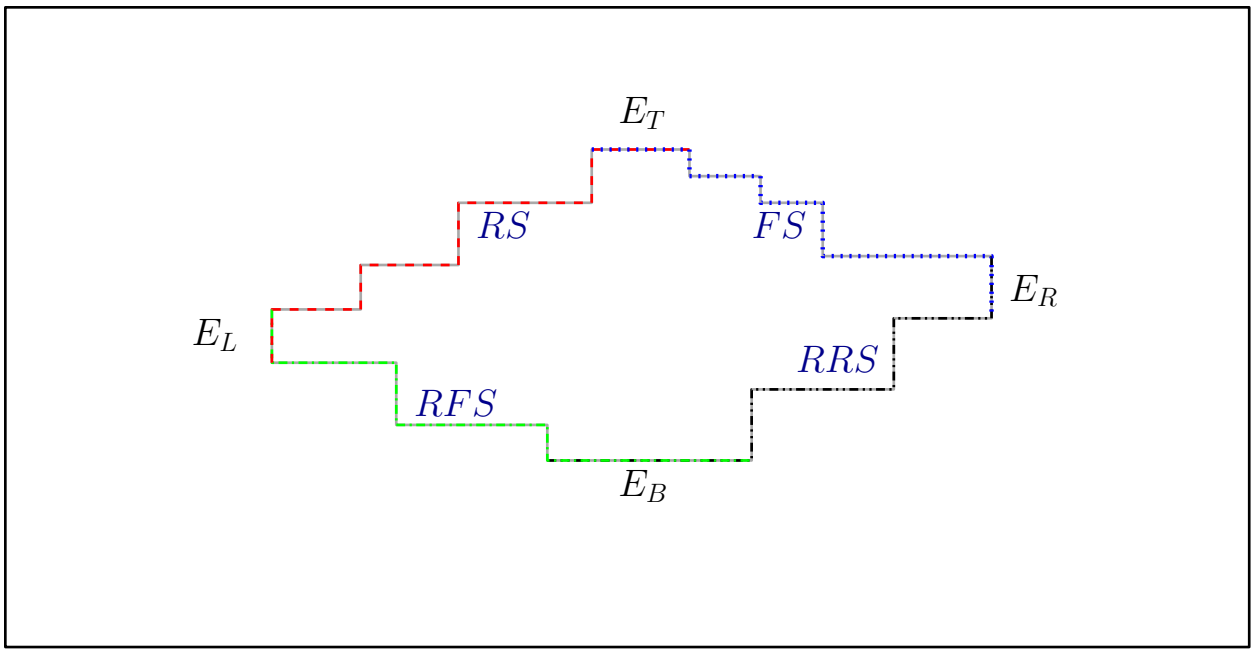


Figure 2.22: E_L, E_R, E_T and E_B are the leftmost, rightmost, topmost and bottom-most edge of the orthogonal hole. Rising staircase is shown in Red (dashed), Falling staircase is shown in Blue (dotted), Reverse Rising Staircase is shown in Black (dash dot dotted) and Reverse Falling Staircase is shown in Green (dash dotted).

Each hole contributes towards the formation of shapes. It is easy to notice that the total number of shapes contributed by a hole H_i is equal to the sum of:

1. number of reflex vertices FS_i^r on FS_i,

2. number of convex vertices RS_i^c on RS_i, and

3. number of reflex vertices RFS_i^r on RFS_i.

The total number of shapes is then $1 + \sum_{i=1}^{k}(FS_i^r + RS_i^c + RFS_i^r)$, where one guard is placed at a corner of the bounding rectangle P.

For each hole H_i, the following conditions hold:

1. On any staircase, total reflex vertices = total convex vertices + 1. Therefore, $RS_i^r = RS_i^c + 1$, $FS_i^r = FS_i^c + 1$, $RRS_i^r = RRS_i^c + 1$ and $RFS_i^r = RFS_i^c + 1$

2. Total reflex vertices = total convex vertices + 4, i.e. $r_i = c_i + 4$

Over all the holes, the relation between total reflex vertices and total convex vertices is as follows, $\sum_{i=1}^{k} r_i = \sum_{i=1}^{k} c_i + 4k$.

From Equation 2.1,

$$\sum_{i=1}^{k}(FS_i^r + RS_i^r + RFS_i^r + RRS_i^r) = \frac{m}{2} + 2k$$

Substituting the value of RS_i^r as $RS_i^c + 1$

$$\sum_{i=1}^{k}(FS_i^r + RS_i^c + 1 + RFS_i^r + RRS_i^r) = \frac{m}{2} + 2k$$

$$\sum_{i=1}^{k}(FS_i^r + RS_i^c + RFS_i^r) = \frac{m}{2} + k - \sum_{i=1}^{k} RRS_i^r \tag{2.2}$$

Total number of shapes = $1 + \frac{m}{2} + k - \sum_{i=1}^{k} RRS_i^r$

The bound on the total guards required to guard P depends upon the value of $\sum_{i=1}^{k} RRS_i^r$. We can always rotate the polygon to select the best bound: the bound can be improved by selecting the rotation which maximizes the value of $\sum_{i=1}^{k} RRS_i^r$. The rotation refers to the combined rotation of the given polygon and the holes in the clockwise direction, either by 0°, 90°, 180° or by 270° so that the polygon and holes remain axis-aligned. Therefore, we have four possible rotations and among the four possible rotations, we select the one which maximizes the value of $\sum_{i=1}^{k} RRS_i^r$.

From Equation 2.1, the total number of reflex vertices is $\frac{m}{2} + 2k$. In the worst case, all rotations result in the same value of $\sum_{i=1}^{k} RRS_i^r$, thus $\sum_{i=1}^{k} RRS_i^r \geq \frac{1}{4} \times \lfloor \frac{m}{2} + 2k \rfloor = \lfloor \frac{m}{8} + \frac{k}{2} \rfloor$.

The total number of shapes is $1 + \frac{m}{2} + k - \sum_{i=1}^{k} RRS_i^r \leq 1 + \frac{m}{2} + k - (\lfloor \frac{m}{8} + \frac{k}{2} \rfloor) = \lfloor \frac{3m}{8} + \frac{k}{2} \rfloor + 1$. Note that this bound also matches with the bound obtained for the special case considered in Section 2.3.2.

Theorem 23. $\lfloor \frac{3m}{8} + \frac{k}{2} \rfloor + 1$ *guards are always sufficient to guard an axis-aligned rectangle polygon with* k *disjoint axis-aligned monotone orthogonal holes, where* m *is the total number of vertices of the holes, with at most one guard placed at a corner of the bounding rectangle.*

If however, we do not allow a guard to be placed at a corner of the enclosing rectangle P, then the number of guards needed could increase significantly. In the rest of this section, we prove an upper bound on the number of vertex guards, placed only on vertices of the holes. To this end, we derive a worst-case upper bound on the number of vertex guards required to cover the portion of the city guarded by the guard placed at a vertex of the rectangle P.

We introduce staircases $RS_{staircase}, FS_{staircase}, RRS_{staircase}$ and $RFS_{staircase}$ (refer to Figure 2.23). These staircases are similar as of staircases in Section 2.3.2. $RS_{staircase}$ is constructed as follows: $\forall i \in [1, k]$, extend the horizontal edges of each hole towards the right (East) direction, then extend the vertical edges towards the South direction. The closed orthogonal polygon formed by the top edge, left edge of P, and the extended edges of the holes, corresponds to a rising staircase, $RS_{staircase}$. Similarly, we can construct $RRS_{staircase}, FS_{staircase}$, and $RFS_{staircase}$. As discussed in Section 2.3.2, we can replace the guard placed on the corner of P by placing guards at the reflex vertices of $RRS_{staircase}$. The total number of guards required to guard such staircase is $1 + RRS^r_{staircase}$.

$$\textit{Number of guards required to guard } P = 1 + \sum_{i=1}^{k} (FS^r_i + RS^c_i + RFS^r_i) + RRS^r_{staircase} \quad (2.3)$$

Substituting the value of $\sum_{i=1}^{k} (FS^r_i + RS^c_i + RFS^r_i)$ from Equation 2.2:

$$\textit{Number of guards required to guard } P = 1 + \frac{m}{2} + k - \sum_{i=1}^{k} RRS^r_i + RRS^r_{staircase} \quad (2.4)$$

The bound on the total number of guards required to guard P depends upon the two terms in Equation 2.4. We can always rotate the polygon to find the rotation which maximizes the value of $\sum_{i=1}^{k} RRS^r_i - RRS^r_{staircase}$.

In the rest of this section, our main aim is to maximize the value of $\sum_{i=1}^{k} RRS_i^r - RRS_{staircase}^r$ among all rotations (out of 4 possible). Note that the staircases RRS_i , $\forall i \in [1,k]$, do not contribute towards the total guards, but the staircase $RRS_{staircase}$ does, and its stairs are stairs from the RRS_i , $i \in [1,k]$, staircases.

In what follows, we provide a divide and conquer approach to improve the upper bound provided earlier.

For some hole H_i, the vertical span of H_i is the parallel strip (extension of E_L^i and E_R^i) and containing H_i. The horizontal span is defined accordingly.

Let H_L be the leftmost hole, H_T be the topmost hole, H_R be the rightmost hole, and H_B be the bottom-most hole of the polygon, as defined by their leftmost, topmost, rightmost, and bottom-most edges. Assume two adjacent staircases, say $RS_{staircase}$ and $FS_{staircase}$, include edges of the same hole $H_i \notin \{H_L, H_T, H_R, H_B\}$. Then, all the holes that lie entirelly inside the vertical span of H_i are in the vertical span of H_i. Similarly, if staircases $RRS_{staircase}$ and $FS_{staircase}$ include the edges of the same hole $H_i \notin \{H_L, H_T, H_R, H_B\}$. Then, all the holes that lie entirelly inside the horizontal span of H_i are in the horizontal span of H_i.

A hole H_i is called an *internal hole* if $H_i \notin \{H_L, H_T, H_R, H_B\}$. The pair of staircases $(RS_{staircase}, FS_{staircase})$, $(FS_{staircase}, RRS_{staircase})$, $(RRS_{staircase}, RFS_{staircase})$ and $(RFS_{staircase}, RS_{staircase})$ are *adjacent staircases* while the pairs $(RS_{staircase}, RRS_{staircase})$ and $(FS_{staircase}, RFS_{staircase})$ are *opposite staircases.*

Consider the four staircases $RS_{staircase}, FS_{staircase}, RFS_{staircase}$, and $RRS_{staircase}$. We can have four possible situations (cases):

Situation 1: No internal hole is shared by either adjacent staircases or opposite staircases.

Situation 2: There exists an internal hole that is shared by opposite staircases and no internal hole is shared by adjacent staircases.

Situation 3: There exists an internal hole that is shared by adjacent staircases and no internal hole is shared by opposite staircases.

Situation 4: There exists an internal hole shared by a pair of adjacent staircases and a pair of opposite staircases.

Before addressing the cases above we notice it is possible that, from the set of holes $(H_L, H_T), (H_T, H_R), (H_R, H_B)$, and (H_B, H_L), the pair in one of the sets corresponds to the same hole.

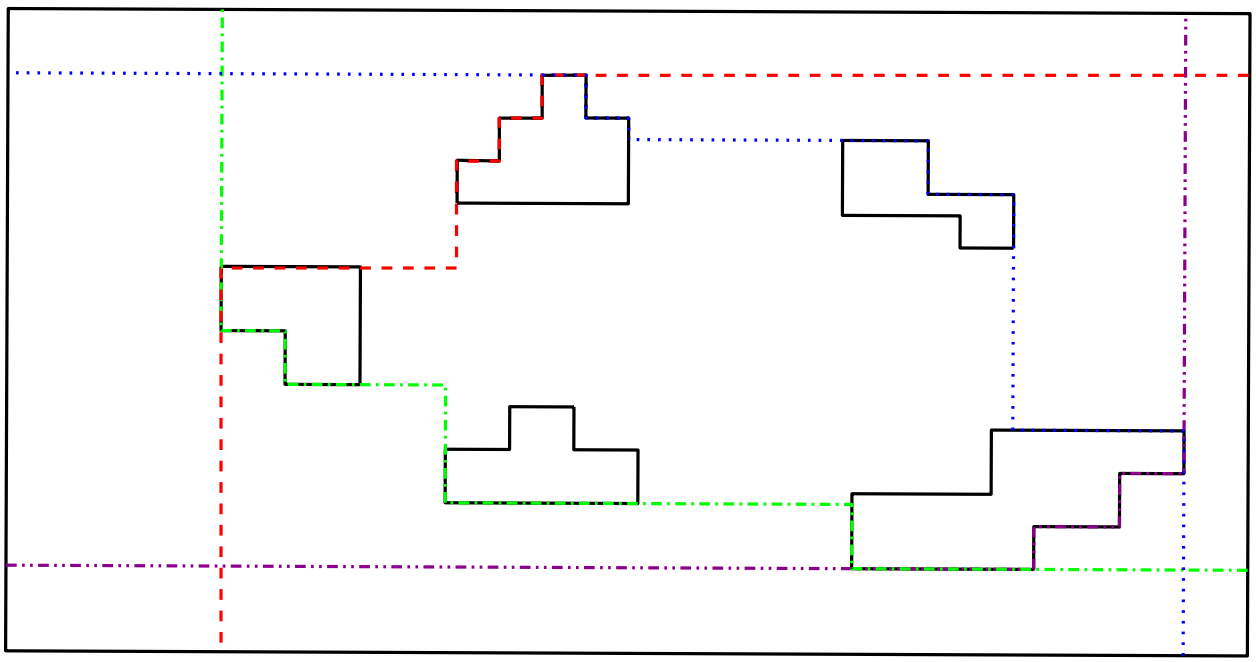


Figure 2.23: The four staircases $RS_{staircase}$ is shown in Red (dashed), $FS_{staircase}$ is shown in Blue (dotted), $RRS_{staircase}$ is shown in Magenta (dash dot dotted) and $RFS_{staircase}$ is shown in Green (dash dotted). The staircase $RRS_{staircase}$ consist of only one hole. Among all the holes H_i, not participating in the formation of $RRS_{staircase}$, $RRS_i^r = 1$.

In this situation(call it Situation 0), one of the staircases $RS_{staircase}, FS_{staircase}, RRS_{staircase}$, and $RFS_{staircase}$ consists of only one hole. Therefore, there exists a rotation where $k-1$ holes do not participate in the formation of $RRS_{staircase}$. We rotate the polygon such that the $RRS_{staircase}$ consist of one hole. All the orthogonal holes are of arbitrary shape. Among all the holes H_i, not participating in the formation of $RRS_{staircase}$, $RRS_i^r \geq 1$ (see Figure 2.23).

$$\sum_{i=1}^{k} RRS_i^r - RRS_{staircase}^r \geq k - 1 \tag{2.5}$$

From Equation 2.4,

$$\textit{total guards required to guard } P = 1 + \frac{m}{2} + k - \sum_{i=1}^{k} RRS_i^r + RRS_{staircase}^r$$

From Equation 2.4 and Equation 2.5,

$$\textit{total guards required to guard } P \leq 1 + \frac{m}{2} + k - (k-1)$$

$$\textit{total guards required to guard } P \leq \frac{m}{2} + 2 \tag{2.6}$$

Thus, from now on, we assume this is not the case.

Situation 1. *No internal hole is shared by either adjacent staircases or opposite staircases.*

The staircases $RS_{staircase}, FS_{staircase}, RRS_{staircase}$, and $RFS_{staircase}$ do not share any hole other than H_T, H_R, H_B, and H_L. We want to find a large lower bound for $\sum_{i=1}^{k} RRS_i^r - RRS_{staircase}^r$ and note that each hole not in $RRS_{staircase}^r$ contributes at least 1 reflex vertex for that bound and each hole on $RRS_{staircase}^r$ subtracts at least 1 reflex vertex from it. We can always rotate the polygon such that $RRS_{staircase}$ contains the smallest number of holes among all possible rotations. The largest possible value for the minimum number of holes involved in $RRS_{staircase}$ is when the number of holes involved in the construction $RS_{staircase}, FS_{staircase}$, $RRS_{staircase}$, and $RFS_{staircase}$, is the same. Let the staircases $RS_{staircase}$ and $RRS_{staircase}$ contain δ distinct holes. Staircase $FS_{staircase}$ contains $\delta - 2$ additional holes because hole H_T is already counted in staircase $RS_{staircase}$ and hole H_R is counted in staircase $RRS_{staircase}$. Similarly, $RFS_{staircase}$ contains $\delta - 2$ additional holes. Note that $\delta + \delta + \delta - 2 + \delta - 2 = k$ and thus $\delta = \lfloor \frac{k}{4} \rfloor + 1$. Therefore, $k - \delta = k - (\lfloor \frac{k}{4} \rfloor + 1) = \lceil \frac{3k}{4} \rceil - 1$ holes do not participate in the formation of $RRS_{staircase}$. Among all the holes H_i, not participating in the formation of $RRS_{staircase}$, $RRS_i^r \geq 1$. Therefore,

$$\sum_{i=1}^{k} RRS_i^r - RRS_{staircase}^r \geq \lceil \frac{3k}{4} \rceil - 1 \tag{2.7}$$

From Equation 2.4,

$$\textit{total guards required to guard } P = 1 + \frac{m}{2} + k - \sum_{i=1}^{k} RRS_i^r + RRS_{staircase}^r$$

From Equation 2.4 and Equation 2.7,

$$\textit{total guards required to guard } P \leq 1 + \frac{m}{2} + k - (\lceil \frac{3k}{4} \rceil - 1)$$

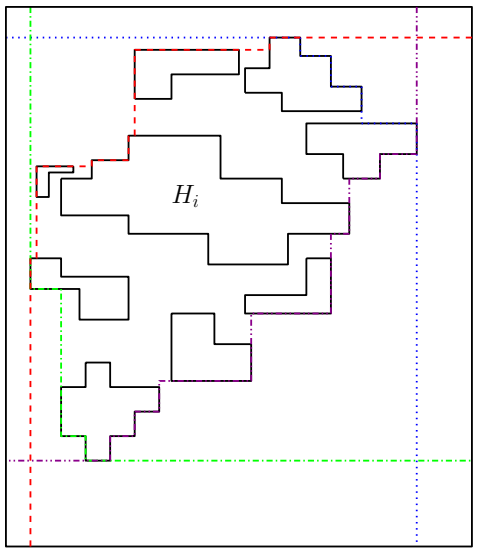
$$\textit{total guards required to guard } P \leq \frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 2 \tag{2.8}$$

Situation 2. *There exists an internal hole that is shared by opposite staircases and no internal hole is shared by adjacent staircases.*

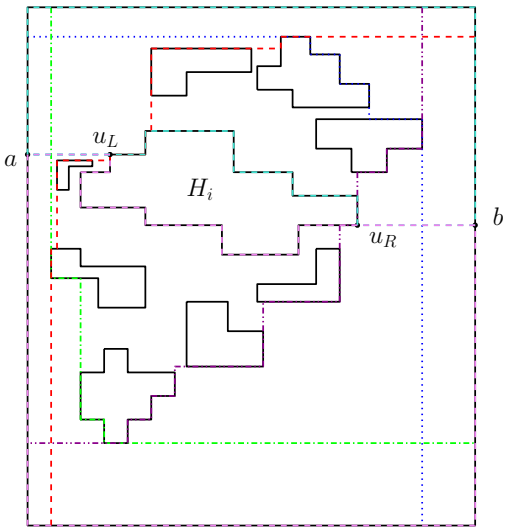
Let the staircases $RS_{staircase}$ and $RRS_{staircase}$ share a hole H_i and refer to Figure 2.24. We define two polygons, P_1 and P_2 (turquoise and violet boundaries in Figure 2.24b), included in P and such that H_i is included in both P_1 and P_2.

P_1 and P_2 are constructed as follows: Traverse H_i from E_L^i to E_R^i in clockwise direction and let u_L be the first vertex of H_i which is part of $RS_{staircase}$. Let e_L be the horizontal edge incident to u_L. We extend e_L to the left until it hits the left boundary of P at point a. Traverse H_i from E_R^i to E_L^i in clockwise direction and let u_R be the first vertex of H_i which is part of $RRS_{staircase}$. Let e_R be the horizontal edge incident to u_R. We extend e_R to the right until it hits the right boundary of P at point b.

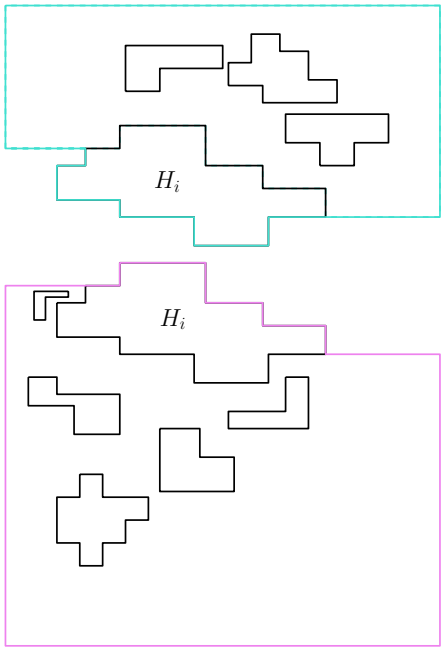
P_1 is the polygon traced by going from u_L to a, top left vertex of P, top right vertex of P, b, u_R then counterclockwise along the boundary of H_i from u_R to u_L. P_2 is defined on the lower part of H_i in a similar way. Note that P_1, H_i, and P_2 are interior disjoint (see Figure 2.24b). We then make $P_1 = P_1 \cup H_i$ and $P_2 = P_2 \cup H_i$ (see Figure 2.24c).



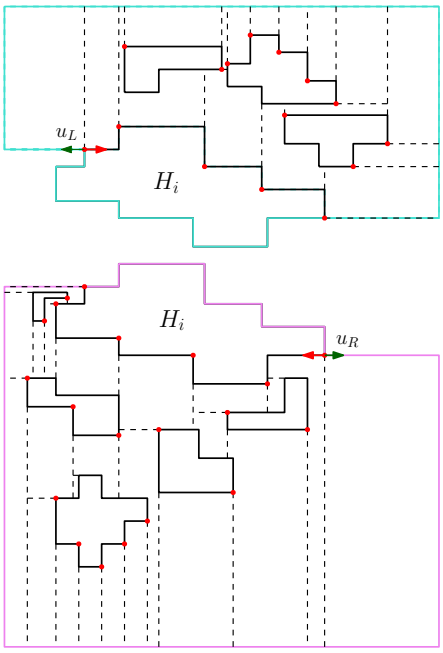
(a) Hole H_i edges are included in staircases $RS_{staircase}$ and $RRS_{staircase}$.



(b) Polygon is divided along H_i. P_1 is shown in dotted Turquoise and P_2 is shown in dashed violet



(c) Two polygons created around hole H_i



(d) Extension of edges and placement of guards (red dots). We place an additional guard on u_L in P_1 and on u_R in P_2. Each red guard in P_1 is facing East and in P_2 its facing West.

Figure 2.24: The four staircases. $RS_{staircase}$ is shown in Red (dashed), $FS_{staircase}$ is shown in Blue (dotted), $RRS_{staircase}$ is shown in Magenta (dash dot dotted) and $RFS_{staircase}$ is shown in Green (dash dotted). P_1 is shown in violet and P_2 is shown in Turquoise

We guard P_1 and P_2 separately. Let P_1 contain α number of holes (excluding H_i), and $m_\alpha = \sum_{i=1}^{\alpha} m_i$, and let P_2 contain β number of holes (excluding H_i), with $m_\beta = \sum_{i=1}^{\beta} m_i$. Note that $\alpha + \beta + 1 = k$.

For each hole H_j in P_1,

1. Traverse H_j from E_R^j to E_L^j in an counter-clockwise direction and extend each encountered vertical edge (including E_L^j and E_R^j) towards North (upward) direction until it hits the polygon P_1 or any other hole.

2. Traverse H_j from E_R^j to E_B^j in a clockwise direction and extend each horizontal edge (including E_B^j) towards East (right) direction until it hits the polygon P_1, any other hole, or any of the extended vertical edges.

For each hole H_j in P_2,

1. Traverse H_j from E_R^j to E_L^j in an clockwise direction and extend each encountered vertical edge (including E_L^j and E_R^j) towards South (downward) direction until it hits the polygon P_2 or any other hole.

2. Traverse H_j from E_L^j to E_T^j in an clockwise direction and extend each horizontal edge (including E_T^j) towards West (left) direction until it hits the polygon P_2, any other hole, or any of the extended vertical edges.

The above steps divide the polygon P_1 and P_2 into shapes, where each shape corresponds to a monotone staircase, and one guard is required to guard each shape (see Figure 2.24d).

Each hole contributes towards the formation of shapes. Let S_{P1} be the total number of shapes contributed by holes in P_1 excluding H_i. Total number of shapes contributed by each hole H_j (excluding H_i) in P_1 is equal to the sum of:

1. number of reflex vertices RS_j^r on RS_j,

2. number of convex vertices FS_j^c on FS_j, and

3. number of reflex vertices RRS_j^r on RRS_j.

$S_{P1} = \sum_{i=1}^{\alpha} FS_j^c + RS_j^r + RRS_j^r$.

Let S_{P2} be the total number of shapes contributed by holes in P_2 excluding H_i. Total number of shapes contributed by each hole H_j (excluding H_i) in P_2 is equal to the sum of:

1. number of reflex vertices RS_j^r on RS_j,

2. number of convex vertices RFS_j^c on RFS_j, and

3. number of reflex vertices RRS_j^r on RRS_j.

$S_{P2} = \sum_{i=1}^{\beta} RFS_j^c + RS_j^r + RRS_j^r$.

Let S_{Hi} be the total number of shapes contributed by hole H_i in both P_1 and P_2, which is equal to the sum of:

1. number of reflex vertices RS_i^r on RS_i,

2. number of reflex vertices RRS_i^r on RRS_i,

3. number of convex vertices FS_i^c on FS_i,

4. number of convex vertices RFS_i^c on RFS_i, and

5. Two extra shapes, one formed at vertex u_L and another at vertex u_R as these vertices are shared by both polygons, P_1 and P_2.

$S_{Hi} = \frac{m_i}{2} + 2$.

We have one additional rectangular shape in each of the two polygons. In P_1, it is the leftmost shape and in P_2, it is the rightmost shape.

The total number of shapes in P_1 and P_2 is $2 + S_{P1} + S_{P2} + S_{Hi}$

From Equation 2.1,

$$\sum_{i=1}^{\alpha}(FS_i^r + RS_i^r + RFS_i^r + RRS_i^r) = \frac{m_\alpha}{2} + 2\alpha$$

Substituting the value of FS_i^r as $FS_i^c + 1$

$$\sum_{i=1}^{\alpha}(RS_i^r + FS_i^c + 1 + RFS_i^r + RRS_i^r) = \frac{m_\alpha}{2} + 2\alpha$$

$$\sum_{i=1}^{\alpha}(RS_i^r + FS_i^c + RRS_i^r) = \frac{m_\alpha}{2} + \alpha - \sum_{i=1}^{\alpha} RFS_i^r$$

Therefore, $S_{P1} = \frac{m_\alpha}{2} + \alpha - \sum_{i=1}^{\alpha} RFS_i^r$. For each hole, $RFS_i^r \geq 1$ and thus we have $S_{P1} \leq \frac{m_\alpha}{2}$. Similarly, $S_{P2} \leq \frac{m_\beta}{2}$.

The total number of shapes is $S_{P1} + S_{P2} + S_{Hi} + 2 \leq \frac{m_\alpha}{2} + \frac{m_\beta}{2} + \frac{m_i}{2} + 4 = \frac{m}{2} + 4$.

It follows that the total number of shapes contributed by the k holes of P is no more than $\frac{m}{2} + 2$. In order to guard $\frac{m}{2} + 2$ shapes, we place one guard in each shape. Let us assume that all guards placed in P_1 are facing East (right) and all the guards placed in P_2 are facing West (left). Note that two guards are placed on each of vertex u_L and vertex u_R, one facing East and one facing West. One of the guards placed on u_L (resp. u_R) is used to cover the side rectangular shape associated with u_L (resp. u_R). Hence,

$$\textit{total number of guards required to guard } P \leq \frac{m}{2} + 2 \tag{2.9}$$

Situation 3. *There exists an internal hole that is shared by adjacent staircases and no internal hole is shared by opposite staircases.*

Let H_i be a hole whose edges are included in more than one staircase. Assume the pair $RS_{staircase}, FS_{staircase}$ shares H_i, as shown in Figure 2.25. Traverse H_i from E_L^i to E_R^i in clockwise direction and let v_1^i be the first vertex encountered while traversing H_i which is part of $RS_{staircase}$. Similarly, let v_2^i be the last vertex encountered while traversing H_i which

is part of $FS_{staircase}$. Let E_1^i be the vertical edge incident to v_1 and E_2^i be the vertical edge incident to v_2. Extend E_1^i and E_2^i in upward direction until they hit the polygon P. Let C_1 be the set of holes which lie above H_i and inside the parallel strip defined by extended edges E_1^i and E_2^i . Let α_i be the number of holes in C_1. Note that all holes in C_1 lie within the vertical span of H_i. Let C_2 be the set containing the holes which are not in C_1 (excluding H_i) and let β_i be the number of holes in C_2. Note that $\alpha_i + \beta_i + 1 = k$. Let C be the set containing all holes H_i that fall within Situation 3. From set C, find the hole H_j that minimizes the value $|\alpha_j - \beta_j|$, such that $\alpha_j, \beta_j \geq 3$.

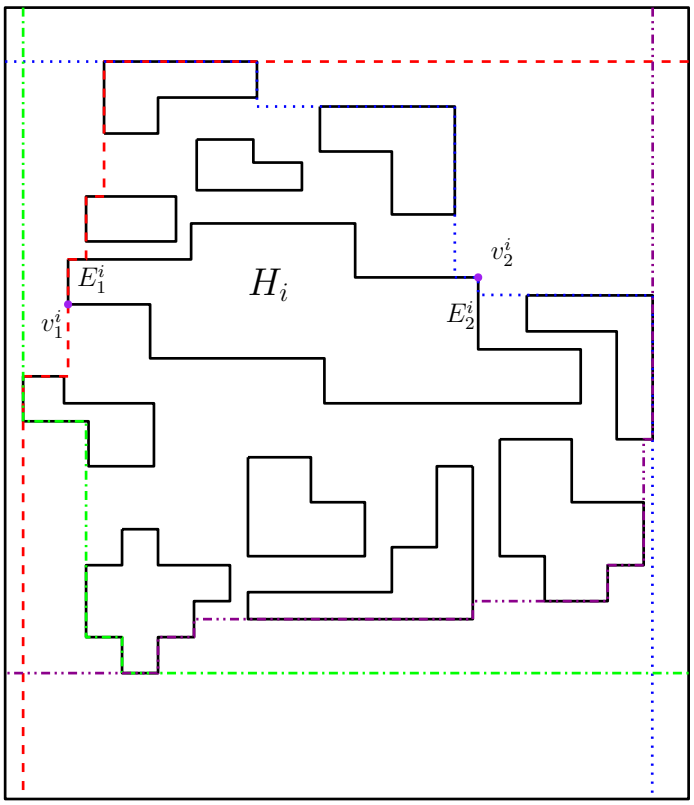


Figure 2.25: The four staircases $RS_{staircase}$ is shown in Red(dashed), $FS_{staircase}$ is shown in Blue(dotted), $RRS_{staircase}$ is shown in Magenta(dash dot dotted) and $RFS_{staircase}$ is shown in Green(dash dotted).

If such hole does not exist then each hole H_j in set C has $\alpha_j \leq 3$ or $\beta_j \leq 3$. We can use a similar argument as the one discussed in Situation 2. Recall that in the worst case

all staircases should involve an equal number of holes. It is easy to notice that there exist at most three holes whose edges are included by a staircase pair (i) $RS_{staircase}, FS_{staircase}$ (ii) $FS_{staircase}, RRS_{staircase}$ (iii) $RRS_{staircase}, RFS_{staircase}$, or (iv) $RFS_{staircase}, RS_{staircase}$, as $\alpha_i < 3$ or $\beta_i < 3$. Let each of $RS_{staircase}, RRS_{staircase}$ contain δ distinct holes. Staircases $FS_{staircase}, RFS_{staircase}$ contain $\delta - 6$ additional holes, as three holes are included in each of $RS_{staircase}$ and $RRS_{staircase}$. There are k holes, $2 \times \delta + 2 \times (\delta - 6) = k$, and $4 \times \delta - 12 = k$, which gives $\delta = \lfloor \frac{k}{4} \rfloor + 3$. In the worst case, $k - \delta = k - (\lfloor \frac{k}{4} \rfloor + 3) = \lceil \frac{3k}{4} \rceil - 3$ holes do not participate in the formation of $RRS_{staircase}$. Among all the holes H_i not participating in the formation of $RRS_{staircase}$, $RRS_i^r \geq 1$. Therefore,

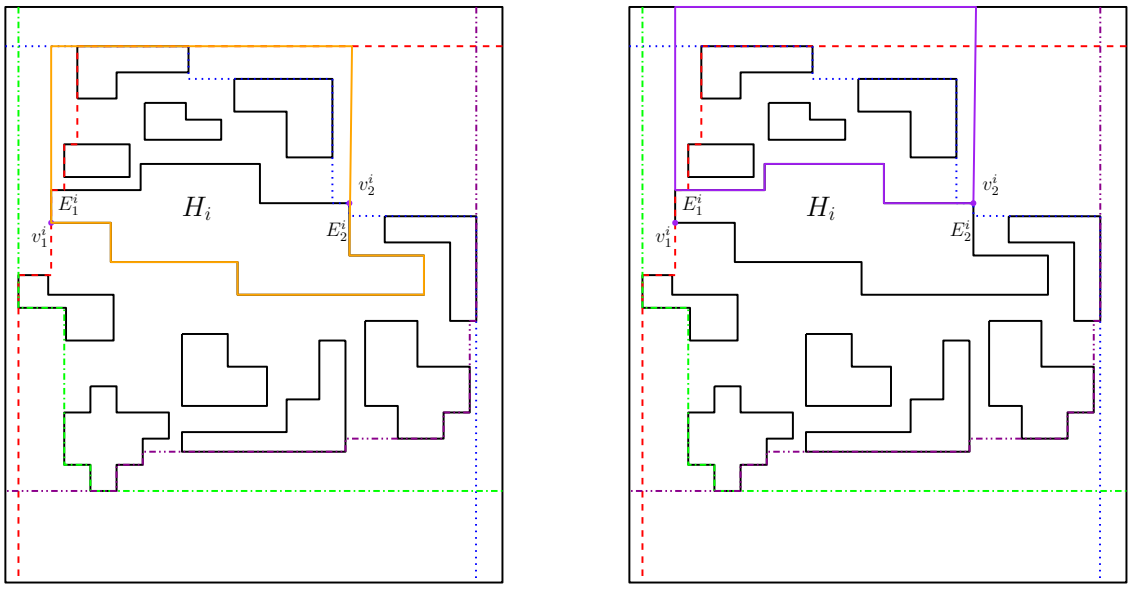
$$\sum_{i=1}^{k} RRS_i^r - RRS_{staircase}^r \geq \lceil \frac{3k}{4} \rceil - 3 \tag{2.10}$$

If there exists a hole H_j such that $\alpha_j, \beta_j \geq 3$, we proceed as follows. Let H_j edges be included in $RS_{staircase}$ and $FS_{staircase}$. Extend the topmost edge of H_T in both directions (East and West). Let e be this extended edge. Extend E_1^j and E_2^j North (upward) until they intersect e. Let the extension of E_1^j intersect e at a and the extension of E_2^j intersect e at b.

We introduce a new orthogonal hole N_1, such that the hole H_j and the all the holes in C_1 lie inside N_1 and all the holes in C_2 lies outside N_1. N_1 is an orthogonal polygon traced by going from v_1^j to a, b, v_2^j then clockwise along the boundary of H_j from v_2^j to v_1^j (see Figure 2.26).

The process of introducing a new hole is known as *contraction* as we have contracted $\alpha_j + 1$ holes into another hole, N_1. The reverse process of *contraction* is known as *expansion*. The updated polygon after performing the contraction consist of $(k - \alpha_j)$ holes. We keep contracting the holes if the updated polygon lies within situation 3, and there exists a hole B_d such that $\alpha_d, \beta_d \geq 3$.

Assume situation 3 keeps occurring, and the process is repeated x times. Let N_i be the hole introduced during the i^{th} contraction. The initial polygon has k holes. After



(a) The new axis-aligned orthogonal hole N_1 formed by the contraction process is shown in orange.

(b) P_1 is shown in purple.

Figure 2.26: Contraction and Expansion

repeating the process for l iterations, let the updated polygon have $k - \sum_{i=1}^{l} k_i$ holes. Let $k^0 = k - \sum_{i=1}^{x} k_i$. At i^{th} contraction, $k_i + 1$ holes are contracted into N_i. After the last contraction, the polygon can't be divided further. Therefore, from Equation 2.10,

$$\sum_{i=1}^{k^0} RRS_i^r - RRS_{staircase}^r \geq \lceil \frac{3k^0}{4} \rceil - 3 \tag{2.11}$$

The polygon P has k^0 holes and let $m^0 = \sum_{i=1}^{k^0} m_i$. The number of guards required to guard $P \leq 1 + \frac{m^0}{2} + k^0 - (\lceil \frac{3k^0}{4} \rceil - 3) = \frac{m^0}{2} + \lfloor \frac{k^0}{4} \rfloor + 4$.

N_x is the last contracted hole and let N_x be the topmost hole in P. We perform the expansion process on N_x. Remove hole N_x from the polygon and add all the $k_x + 1$ holes contracted inside N_x. Let H_x be one of the $k_x + 1$ holes contracted inside N_x such that the rest of k_x holes lies inside the vertical span of H_x. After expansion, guards placed on N_x are placed on the corresponding vertices of H_x with the same orientation. If a guard is placed

at a, place it on the top vertex of E_1^x and if a guard is placed b, place it on the top vertex of E_2^x with the same orientation.

Extend E_1^x and E_2^x in upward direction until hit P. Let E_1^x intersect P at u_1^P and E_2^x intersect P at u_2^P. Let P_1 be the closed polygon traced by going from upper vertex of E_1^x to u_1^P, u_2^P, upper vertex of E_2^x then counterclockwise along the boundary of H_x (see Figure 2.26b). $P_2 = P \setminus P_1$. Note that after expansion, P_2 remains guarded.

Note that there is a guard placed on the topmost edge of N_x facing North (upward). Let this guard be placed at a. After expansion, this guard is placed on the top vertex of E_1^x facing North.

Consider the polygon P. Let m^1 be the total number of vertices and let $k^1 = k - \sum_{i=1}^{x-1} k_i$ be the total number of holes inside P after expanding N_x. Let the k_x holes inside N_x (excluding H_x) have a total of m_x vertices. Let $m^{'} = m^1 - m^0 - m_x$. We extend the edges of P_1 in a similar way as we did for situation 2. Total number of shapes formed $= 1 + \frac{m^{'}}{2} + \sum_{i=1}^{k_x} FS_j^c + RS_j^r + RRS_j^r \leq 1 + \frac{m^{'}}{2} + \frac{m_x}{2}$. We need one guard to guard each shape, however one of the shapes is already guarded by the guard placed on the top vertex of E_1^x.

Total guards required to guard the updated $P \leq \frac{m^0}{2} + \lfloor \frac{k^0}{4} \rfloor + 4 + \frac{m_x}{2} + \frac{m^{'}}{2} = \frac{m^1}{2} + \lfloor \frac{k^0}{4} \rfloor + 4$
After expanding all the contracted holes, total guards required to guard $P = \frac{m}{2} + \lfloor \frac{k^0}{4} \rfloor + 4 \leq \frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 4$

$$\textit{total guards required to guard } P \leq \frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 4 \tag{2.12}$$

Situation 4. *There exists an internal hole shared by a pair of adjacent staircases and a pair of opposite staircases.*

We place guards according to Situation 2 and conclude that $\frac{m}{2} + 2$ guards are sufficient to cover P.

$$\textit{total guards required to guard } P \leq \frac{m}{2} + 2 \tag{2.13}$$

From Equation 2.6, 2.8, 2.9, 2.12 and 2.13, we obtain the following bound:

$$\textit{total guards required to guard } P \leq \frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 4 \tag{2.14}$$

Theorem 24. $\frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 4$ *vertex guards are always sufficient to guard an axis-aligned rectangle polygon with* k *disjoint axis-aligned monotone orthogonal holes where* m *is the total number of vertices of the holes, with all guards placed on vertices of the holes.*

The term $\frac{m}{2}$ arises because for each hole H_i with m_i vertices, we place at least $\frac{m_i}{2}$ guards on H_i.

Consider an orthogonal hole H_i. As defined at the starting of this section, let E_L^i, E_T^i, E_R^i, and E_B^i be the leftmost, topmost, rightmost, and bottom-most edges of H_i. The monotone orthogonal hole M_i of H_i is obtained from H_i as follows (see Figure 2.27):

1. Traverse H_i from E_L^i to E_T^i in clockwise direction and extend each horizontal edge towards right (except E_T^i) and extend each vertical edge towards South (except E_L^i) until it hit another edge of H_i or the extended edge,

2. Traverse H_i from E_T^i to E_R^i in clockwise direction and extend each vertical edge towards South (except E_R^i) and each horizontal edge towards left (except E_T^i) until it hit another edge of H_i or the extended edge,

3. Traverse H_i from E_R^i to E_B^i in clockwise direction and extend each horizontal edge towards the left (except E_B^i) and each vertical edge towards North (except E_R^i) until it hit another edge of H_i or the extended edge,

4. Traverse H_i from E_B^i to E_L^i in clockwise direction and extend each horizontal edge towards right (except E_B^i) and each vertical edge towards North (except E_L^i) until it hit another edge of H_i or the extended edge.

.

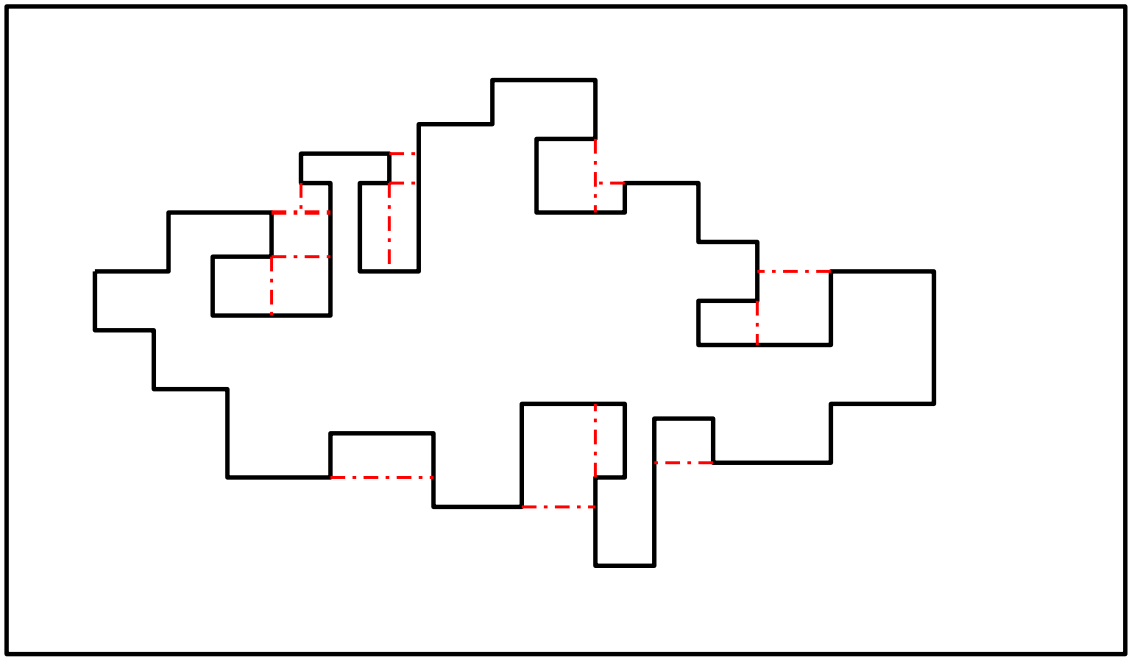


Figure 2.27: Converting an orthogonal polygon into a monotone orthogonal polygon (a).

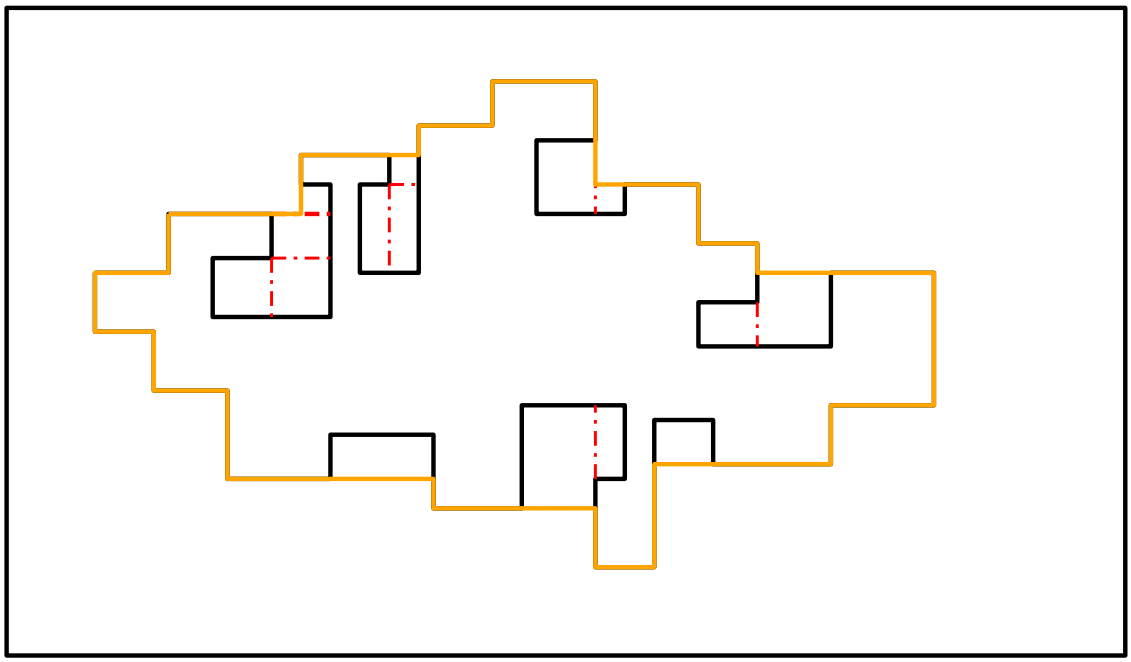


Figure 2.28: Converting an orthogonal polygon into a monotone orthogonal polygon (b).

Lemma 2. *After performing the above steps, the outer boundary of H_i corresponds to a monotone orthogonal hole M_i.*

Proof. Let us assume that M_i is not monotone and there exists a vertical line l that intersects M_i at more that two points. Let l intersect M_i at four points, say a, b, c and d. Let the point a be encountered while traversing M_i from E_B^i to E_L^i in clockwise direction and the rest of the points be encountered while traversing M_i from E_L^i to E_T^i in clockwise direction (see Figure 2.29).

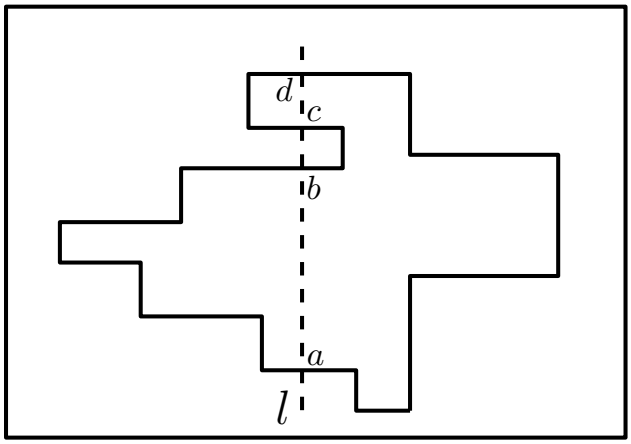


Figure 2.29: Boundary of H_i and a vertical line l that intersect M_i at four points.

Let the points encountered while traversing M_i from E_L^i to E_T^i in clockwise direction is in the following order: b, c, d. Let the point b lie on edge e_b, point c lie in edge e_c and the point d lie on edge e_d. Let e^v be the vertical edge incident to the left vertex of e_c.

While traversing H_i from E_L^i to E_T^i, we extend all the vertical edges towards the South until they hit any other edge of H_i or any extended edge. It means that we extended the edge e^v towards the South until it hit another edge of H_i or any other extended edge. As e_v is not the left most edge and is encountered while traversing E_L^i to E_T^i in clockwise direction, extension of e_v must exist a horizontal edge or an extended horizontal edge. Hence the points b and c does not lie on the outer boundary of H_i after performing the four steps, and line l intersect M_i at only two points, a and d.

Hence, there is no vertical line that intersects M_i at more than two points after performing the above steps. Similarly, we can prove that there is no horizontal line that intersects M_i at more than two points after performing the above steps. Therefore, M_i is a monotone orthogonal polygon. □

The outer boundary obtained from the construction above corresponds to a monotone orthogonal hole M_i (see Figure 2.27). $M_i \setminus H_i$ results in a set of orthogonal polygons, and we can use the result from (Estivill-Castro and Urrutia, 1994) to guard each such polygon. Using (Estivill-Castro and Urrutia, 1994), we need at most $\lfloor \frac{3(n-1)}{8} \rfloor$ guards to guard any orthogonal polygon with n vertices.

While constructing M_i from H_i, we introduced new vertices, and all the new vertices are convex. Note that, we might need to change the position of a placed guard. If a guard is placed on a reflex vertex, we do not change its position or orientation. If a guard is placed on a convex vertex of M_i, which is not a vertex of H_i, we update the position of the guard. The position is updated as follows(see Figure 2.30):

1. Let e_M be a horizontal edge of M_i on RS_i and a guard is placed on the right end of e_M.

2. If e_M is obtained by extending the horizontal edge of H_i towards the right, say e_H.

   (a) Let v_R^M be the right end of e_M and v_R^H be the right end of e_H.

   (b) Remove a guard placed on v_R^M and place it on v_R^H with the same orientation.

   (c) If v_R^M is not a point on H_i, then the vertex v_R^M is obtained by extending a horizontal edge(e_H) towards the right and extending a vertical edge towards South, say e_V. Place another guard on the lower end of e_V facing away from the edge.

3. If e_M is sub-edge of the horizontal edge of H_i

(a) Let v_R^M be the right end of e_M, and the vertex v_R^M is obtained by extending a vertical edge towards the South, say e_V.

(b) Remove a guard placed on v_R^M and place it on the lower end of e_V with the same orientation.

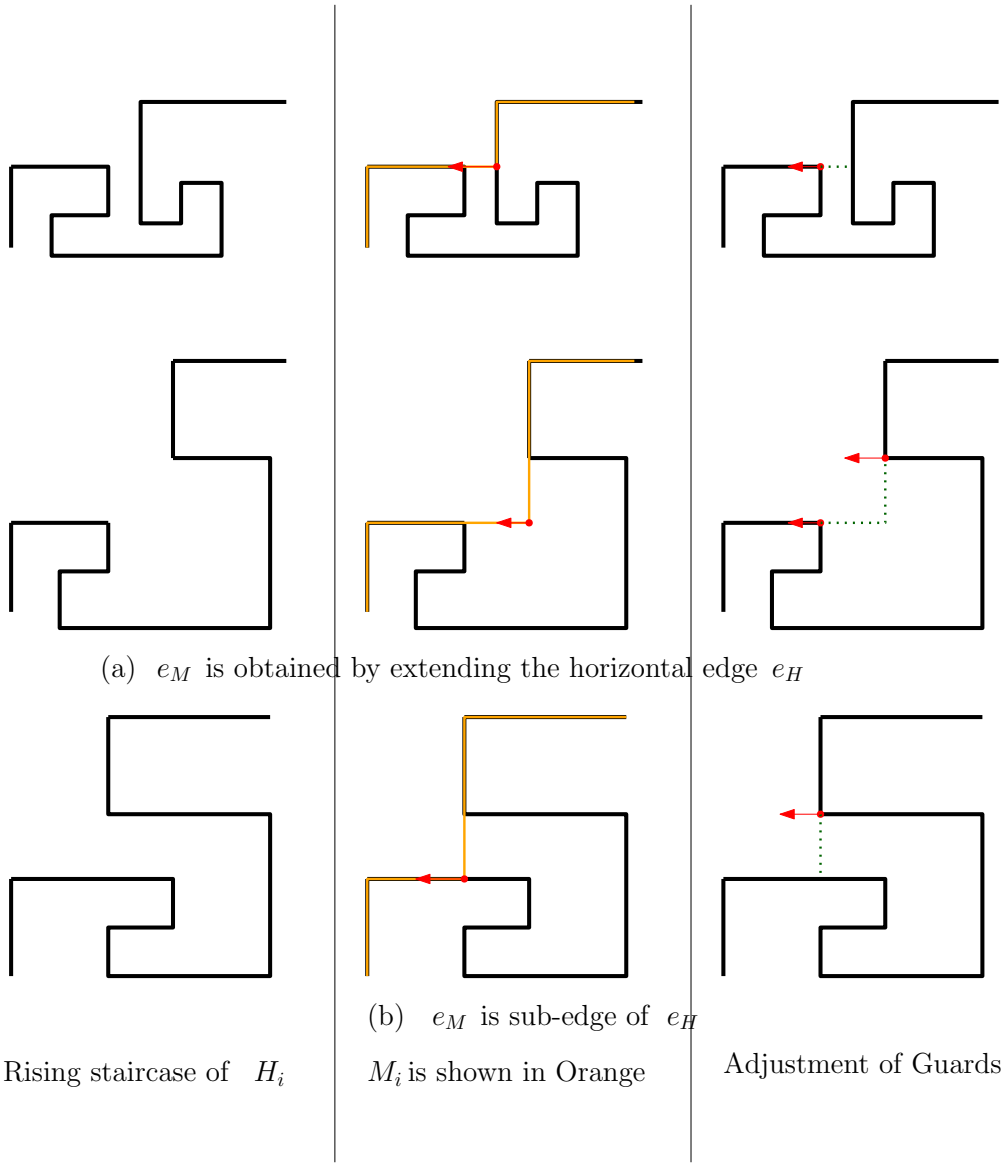


Figure 2.30: The guard positions is shown in Red and the boundary of M_i is shown in Orange.

Consider the monotone orthogonal hole M_i with m_i vertices. We know that we placed at least $\frac{m_i}{2}$ guards on M_i. Consider an orthogonal hole O in $M_i \setminus H_i$. Let O consists of n vertices. Note that out of n vertices, $n-1$ are the vertices of H_i. Also, M_i and O have one common vertex which is not a vertex of H_i. Let M_i^1 be the orthogonal polygon obtained by merging M_i and O into each other (removing the common boundary and vertex of M_i and O from $M_i \cup O$). Note that M_i^1 has $m_i + n - 2$ vertices. We can have the following cases:

Case 1: No additional guard is placed while updating the position of guards as shown above

We place the guards according to one of the rules from (Estivill-Castro and Urrutia, 1994) to guard the orthogonal polygon O. We need at most $\lfloor \frac{3(n-1)}{8} \rfloor$ guards to guard O. Total number of vertices in $M_i^1 = m_i + n - 2$. Note that n is even as O is an orthogonal polygon.

Let us assume that M_i has $\frac{m_i}{2}$ guards placed on it. Therefore, M_i^1 has at most $\frac{m_i}{2} + \lfloor \frac{3(n-1)}{8} \rfloor$ guards.

$\frac{m_i}{2} + \lfloor \frac{3(n-1)}{8} \rfloor = \frac{m_i}{2} + \lfloor \frac{3n}{8} - \frac{3}{8} \rfloor = \frac{m_i}{2} + \lfloor \frac{3n}{8} - \frac{3}{8} + \frac{n}{8} - \frac{n}{8} + \frac{5}{8} - \frac{5}{8} \rfloor = \frac{m_i}{2} + \lfloor \frac{n}{2} - \frac{2}{2} - \frac{n}{8} + \frac{5}{8} \rfloor$
$= \frac{m_i+n-2}{2} + \lfloor \frac{5}{8} - \frac{n}{8} \rfloor \leq \frac{m_i+n-2}{2}$ because $n \geq 4$

That is, we added $n-2$ vertices to M_i, and to guard the added orthogonal polygon, we added at most $\frac{n-2}{2}$ guards.

Case 2: One additional guard is placed while updating the position of guards as shown above

Let us consider that O is encountered while traversing H_i from E_L^i to E_T^i. Let e_1 be the horizontal edge of O which is not part of H_i and e_2 be the vertical edge of O which is not part of H_i. Note that e_1 and e_2 are adjacent edges. Let e_3 be the wall adjacent to e_1 (other than e_2) and e_4 be the wall adjacent to e_2 other than e_1.

Each edge of O is classified as the top, bottom, right, and left edge in the natural way. Note that e_1, e_4 are top edge of O and e_2, e_3 are left edge of O. Now we place guards

according to the bottom-right rule from (Estivill-Castro and Urrutia, 1994), which is as follows: a guard is placed at the right end of all the bottom edges, and a guard is placed on the bottom end of all the right edges facing away from the edge. According to this placement rule, we do not place any guard on e_1, e_2, e_3, and e_4. On the rest of the horizontal edges of O, we place at most one guard. Total number of guards placed on $O \leq \frac{n}{2} - 2 = \frac{n-4}{2}$.

Total number of vertices in $M_i^1 = m_i + n - 2$. The total guards placed on M_i^1 is $\frac{m_i}{2} + \frac{n-4}{2} = \frac{m_i+n-2}{2} - 1 \leq \frac{m_i+n-2}{2}$. That is, we added $n-2$ vertices to M_i and to guard the added orthogonal polygon, we added at most $\frac{n-2}{2}$ guards.

From the above two cases, we obtain the following theorem:

Theorem 25. *Given an axis-aligned rectangle P with k disjoint axis-aligned orthogonal holes, $H_1, H_2, \ldots, H_k$ with $m_1, m_2, \ldots, m_k$ vertices respectively such that $\sum_{i=1}^{k} m_i = m$ and the monotone orthogonal polygon of $H_1, \ldots, H_k$ are disjoint. $\frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 4$ vertex guards are always sufficient to guard P with all guards placed on vertices of the holes, where the range of vision of guards is* $180°$.

Let M_i, M_j be two monotone orthogonal holes of H_i, H_j respectively such that M_i and M_j are not disjoint. Let H_{ij} be the orthogonal polygon obtained by traversing the outer boundary of M_i and M_j. We construct another monotone orthogonal polygon, say M_{ij} from H_{ij}. We continue this process such that all the holes in P are disjoint and monotone.

Using the results from Theorem 24 and Theorem 25, we obtain the following theorem:

Theorem 26. *$\frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 4$ vertex guards are always sufficient to guard an axis-aligned rectangle polygon with k disjoint axis-aligned orthogonal holes where m is total vertices of the holes, with all guards placed on vertices of the holes.*

We also obtain the following Theorem:

Theorem 27. *Given k pairwise disjoint axis aligned orthogonal polygons in the plane, $\frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 4$ vertex guards are always sufficient to guard the free space, where m is total vertices.*

2.5.3 City Guarding

Theorem 28. *Given a city with k disjoint axis-aligned monotone orthogonal buildings with a total of m vertices, within an axis-aligned rectangle P, the number of axis-aligned cameras with 180° range of vision needed to guard the city (roofs, walls, and ground) is upper bounded by $\frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 4$, where cameras are placed only on top vertices of the buildings.*

Proof. The term $\frac{m}{2}$ arises because for each hole H_i with m_i vertices, we place at least $\frac{m_i}{2}$ guards on H_i. In situation 0, situation 1, and situation 3, these guards are placed as follows:

1. For the staircase FS_i, guards are placed at FS_i^r,
2. For the staircase RS_i, guards are placed at RS_i^c,
3. For the staircase RFS_i, guards are placed at RFS_i^r, and
4. For the staircase RRS_i, guards are placed at RRS_i^r except the reflex vertex at B_R.

Let the visibility of each of the $\frac{m_i}{2}$ guards placed on H_i be towards East. We show that these $\frac{m_i}{2}$ guards also cover the roof of H_i. We divide the polygon's roof into different shapes (rectangle and staircase) and show that the placed guards cover each shape. For each hole, H_i, extend all the vertical edges of H_i, which are part of staircases RS_i, FS_i, and RRS_i towards the interior of H_i until they hit another edge of H_i. Refer to Figure 2.31. The process divides the polygon into smaller rectangles and staircases. For each rectangle constructed, a guard is placed at one of its rightmost vertices facing East (left), and it guards the rectangle. The staircase formed is a Reverse Falling Staircase, and a guard is placed at the top right corner of the staircase, which guards the staircase. Hence the roof of each hole is guarded by the placed guards.

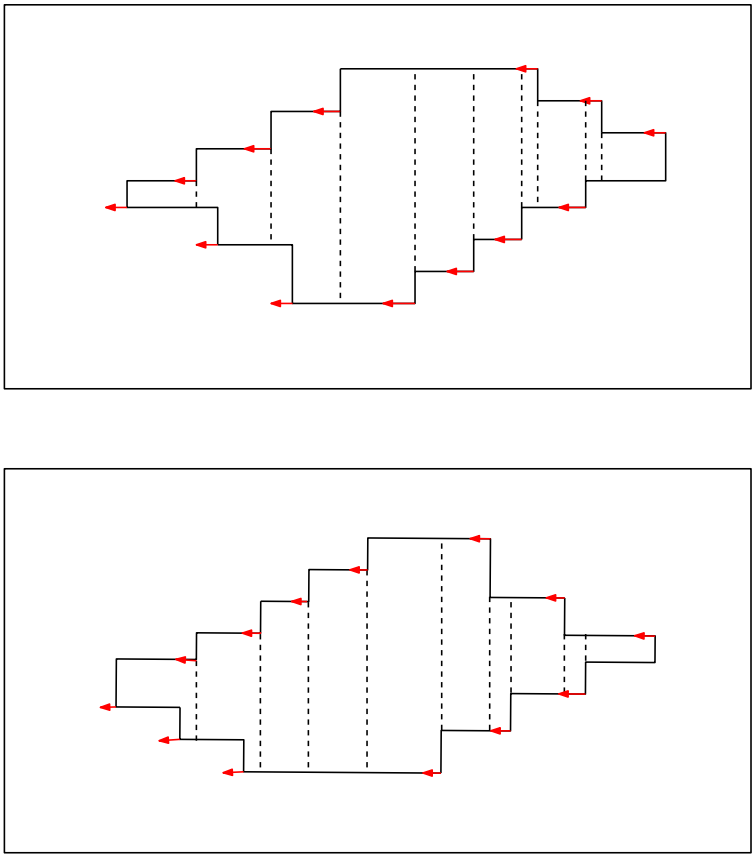


Figure 2.31: Partition of a hole into shapes (rectangular and staircases) with each shape guarded by an already placed guard. Guards location and direction are shown in red arrow.

For situation 2 and situation 4, the guards placed on all the holes except one (hole H_i that is shared by opposite staircases) is placed in a similar way as explained above. From situation 2, we know that $\frac{m_i}{2} + 2$ guards are placed on H_i. Let vertices u_L be the vertex on RS_i and u_R be the vertex on RRS_i as defined in situation 2. We place guards on H_i as follows and all the guards are not facing in the same direction:

1. For the staircase FS_i, guards are placed at FS_i^c,
2. For the staircase RS_i, guards are placed at RS_i^r,
3. For the staircase RFS_i, guards are placed at RFS_i^c, and
4. For the staircase RRS_i, guards are placed at RRS_i^r

5. Two extra guards, one at u_L and another at u_R

Let the visibility of the guards encountered while traversing H_i from u_L on u_R in a clockwise direction be towards the right. Let the visibility of guards encountered while traversing H_i from u_R on u_L in the clockwise direction be towards left (refer Situation 2). On vertex u_L and u_R, we place two guards where visibility of one guard is towards left and visibility of another guard is towards right. We divide the roof of the polygon into different shapes and show that the placed guards cover each shape. Refer to Figure 2.32.

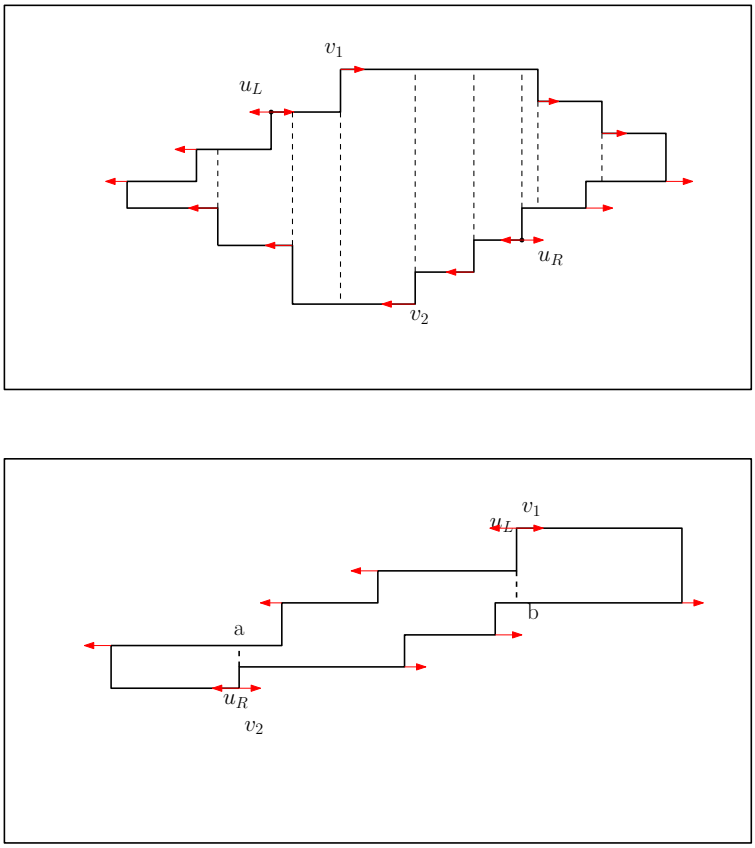


Figure 2.32: Partition of a hole into shapes (rectangular and staircases) with each shape guarded by an already placed guard. Guards location and direction are shown in red arrow.

Let v_1 be the left vertex of E_T^i and v_2 be the right vertex of E_B^i. Traverse H_i from v_1 to E_R^i in clockwise direction and extend each vertical edge towards the interior of H_i(including vertical edge incident at v_1). This step divides the roof on the right of v_1 into staircases and

a guard is placed on the top-left corner guard the staircase. Similarly, traverse H_i from v_2 to E_L^i in clockwise direction and extend each vertical edge towards the interior of H_i(including vertical edge incident at v_2). Note that if v_1 lies towards the left of v_2, then the roof is guarded by the placed guards.

Let us assume that v_1 lies towards the right of v_2. Let us consider that u_R lies toward the right of v_1. In such case, we extend the vertical edges encountered while traversing H_i from u_R to v_2 in clockwise direction including vertical edges incident at u_R and v_2. The space between the extended vertical lines at u_R and u_1 is already guarded. The space between the vertical lines at v_1 and v_2 is guarded by the guard placed at the convex vertices encountered while traversing H_i from u_R to v_2. Hence the roof is completely guarded. Similar is the case if u_L lies towards the left of v_2.

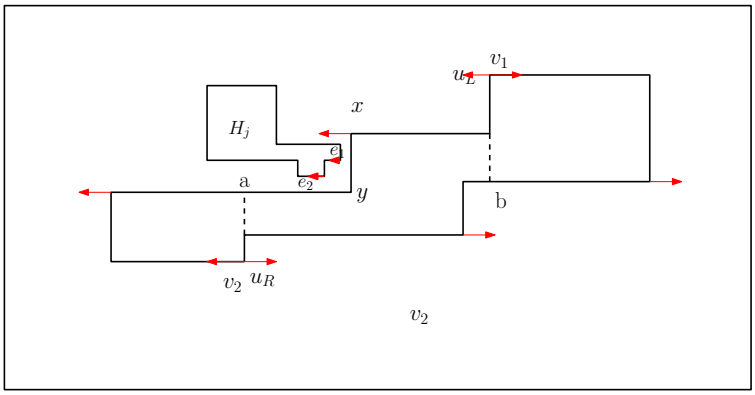


Figure 2.33: Case where v_1 lies towards the right of v_2, u_R lies toward the left of v_1 and u_L lies toward the right of v_2.

Let us consider that v_1 lies towards the right of v_2, u_R lies toward the left of v_1 and u_L lies toward the right of v_2 (see Figure 2.33). The roof area, which is not guarded by the placed guards, lies between the extended vertical edges at u_L and u_R. Let us define this area by A. Let R be the minimum area of a rectangle that can be formed from these points. If the height of all the buildings in R is less than equal to the height of building H_i, then A is guarded by the guards placed on H_i (placed at reflex vertices). Let us assume that there

exists another hole H_j whose height is greater than H_i and blocks the view of one of the guards placed on a reflex vertex of H_i. Let x be the first reflex vertex and y be the first convex vertex encountered while traversing H_i from a to u_L in clockwise direction and the visibility of the guard at x is blocked by another H_j. Let $A^{'}$ be the sub-area of A, which is blocked by H_j. Now we show that $A^{'}$ is guarded by the guards placed on H_j.

Let e_1 be the vertical edge of H_j closest to the vertical edge incident to x and e_2 be the horizontal edge closest to the sub edge ay. Note that in the path from e_1 to e_2, the guards are placed on the reflex vertices, and the path is part of a staircase RRS_j (refer Situation 2). We extend the vertical edges encountered while traversing H_j from e_1 to e_2 in clockwise direction(including edge e_1) in South until it hit RRS_i. The step divided $A^{'}$ into shapes, and the guards placed on H_j guard $A^{'}$. Similarly, if any other hole blocks the view, the guard placed on that hole guards the blocked area.

Hence the roof of H_i is entirely guarded.

□

Using Theorem 25 and Theorem 28, we conclude the following theorem:

Theorem 29. *Given an axis-aligned rectangle P with k disjoint axis-aligned orthogonal holes, $H_1, H_2, \ldots, H_k$ with $m_1, m_2, \ldots, m_k$ vertices respectively such that $\sum_{i=1}^{k} m_i = m$ and the monotone orthogonal polygon of $H_1, \ldots, H_k$ are disjoint. $\frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 4$ vertex guards are always sufficient to guard the city (roofs, walls, and ground) with all guards placed on vertices of the holes, where the range of vision of guards is* $180°$.

Using Theorem 26 and Theorem 29, we conclude the following theorem:

Theorem 30. *Given a city with k disjoint axis-aligned orthogonal buildings with a total of m vertices, within an axis-aligned rectangle P, the number of axis-aligned cameras with $180°$ range of vision needed to guard the city (roofs, walls, and ground) is upper bounded by $\frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 4$, when cameras are placed only on top vertices of the buildings.*

2.6 City Guarding Problem with Arbitrary Oriented Orthogonal Buildings

Given a rectangular city with k disjoint vertical buildings, each having an orthogonal base, the goal is to place the minimum number of cameras with 180° range of vision, at the top corners (vertices) of the buildings to guard the city (roofs, walls, and ground).

2.6.1 Roof Guarding

For roof guarding, the same city structure in Theorem 13 gives:

Theorem 31. *Given a city with k disjoint buildings, k vertex guards are always sufficient and sometimes necessary to guard the roofs.*

2.6.2 Ground and Wall Guarding

As before, the problem of guarding the ground and the walls reduces to:

SubProblem 4. *(V4) Given an axis-aligned rectangle P with k disjoint orthogonal holes $H_1, \ldots, H_k$, with $m_1, m_2, \ldots, m_k$ vertices, respectively, such that $\sum_{i=1}^{k} m_i = m$, place vertex guards on holes such that every point inside P is visible to at least one guard, where the range of vision of guards is* 180°.

Theorem 32. *$r - k + 1$ guards are sometimes necessary to guard an axis-aligned rectangle polygon with k disjoint orthogonal holes and r reflex vertices, with all guards placed on vertices of the holes. We conjecture the bound is tight.*

For the necessity part, consider the structure in Figure 2.34, with the following properties:

1. B_i lies within the span of B_j $, \forall j < i$.

2. None of the edges of B_i is partially or completely visible from any vertex of B_j $, \forall j < i-1$ and from any vertex B_m $, \forall m > i+1$.

3. From each potential position of a vertex guard on B_i, the guard is able to see at most one edge of B_{i+1}.

4. There is no guard position on B_i from where an edge of B_{i-1} and an edge of B_i are visible.

Let E_L^i be the leftmost edge of B_i such that the interior of B_i lies towards the right of it. Similarly, we define E_T^i as the topmost edge, E_R^i as the rightmost edge and E_B^i as the bottom-most edge. For each hole H_i, we define staircases RS_i, FS_i, RRS_i and RFS_i as defined in Section 2.5.2 and shown in Figure 2.22. For each building B_i, let r_i be the total reflex vertices of B_i and c_i be the total convex vertices of B_i such that $\sum_{i=1}^{k} r_i = r$ and $\sum_{i=1}^{k} c_i = c$.

Consider the space $\wp_i$ between two consecutive holes B_i and B_{i+1}, as shown in Figure 2.35. $\wp_i$ is not visible to any guard placed on building B_j where $j \in [1, i) \cup (i+1, k]$ (property 2). Therefore, $\wp_i$ is only visible to the guards either placed on B_i or B_{i+1}. Total possible guard positions on B_i and B_{i+1} from where $\wp_i$ is visible (partially from each positions, see Figure 2.35) are $4(r_{i+1} - 1) + 2c_{i+1}$. These guards cover $m_{i+1} - 1$ walls, one wall of B_i, and $m_{i+1} - 2$ walls of B_{i+1}. Note that these guards do not cover any other wall (partially or entirely), and the mentioned $m_{i+1} - 1$ walls are not visible (partially or entirely) to any other potential guard. Out of $4(r_{i+1} - 1) + 2c_{i+1}$ possible guard positions, the minimum number of guards required to guard $\wp_i$ is $r_{i+1} - 2$. Therefore the space between any two consecutive holes is only guarded by the guards placed on these holes, and the minimum number of guards required to guard such space is $r_{i+1} - 2$.

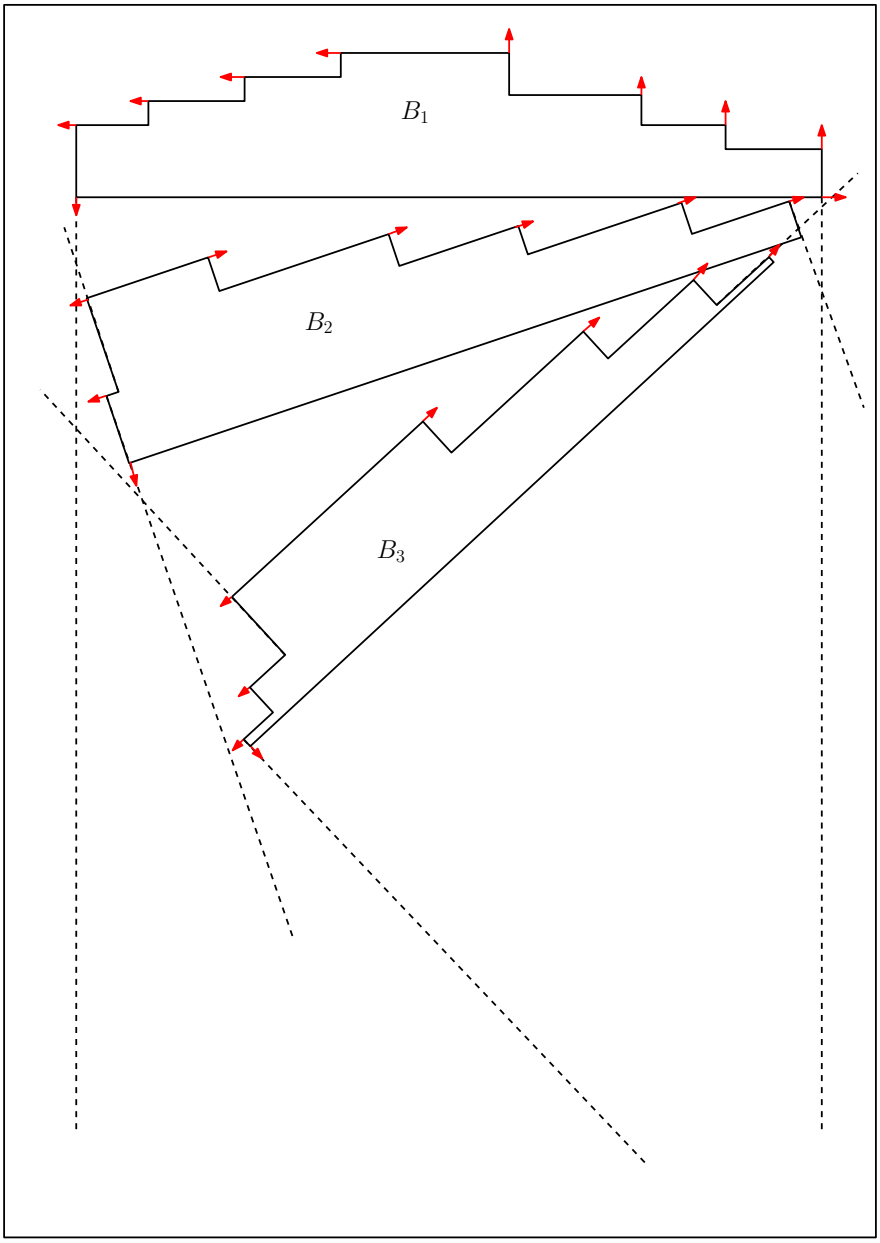


Figure 2.34: City Structure where for each hole $B_i, i \in (1, k]$, $r_i - 1$ guards are placed on B_i and r_1 guards are placed on B_1, where r_i is the number of reflex vertices on hole B_i.

Consider the left wall, w_L^i of hole B_i. Because of the structure of the city, E_L^i is not visible to any guard placed on B_j for $j \neq i$. Hence, E_L^i can only be guarded by a guard placed on B_i, and there are four possible guard positions from where E_L^i is visible (see Figure 2.35). However, none of these guards positions cover (partially or entirely) any other wall in the city. Therefore, we need one guard to cover the left wall of each hole.

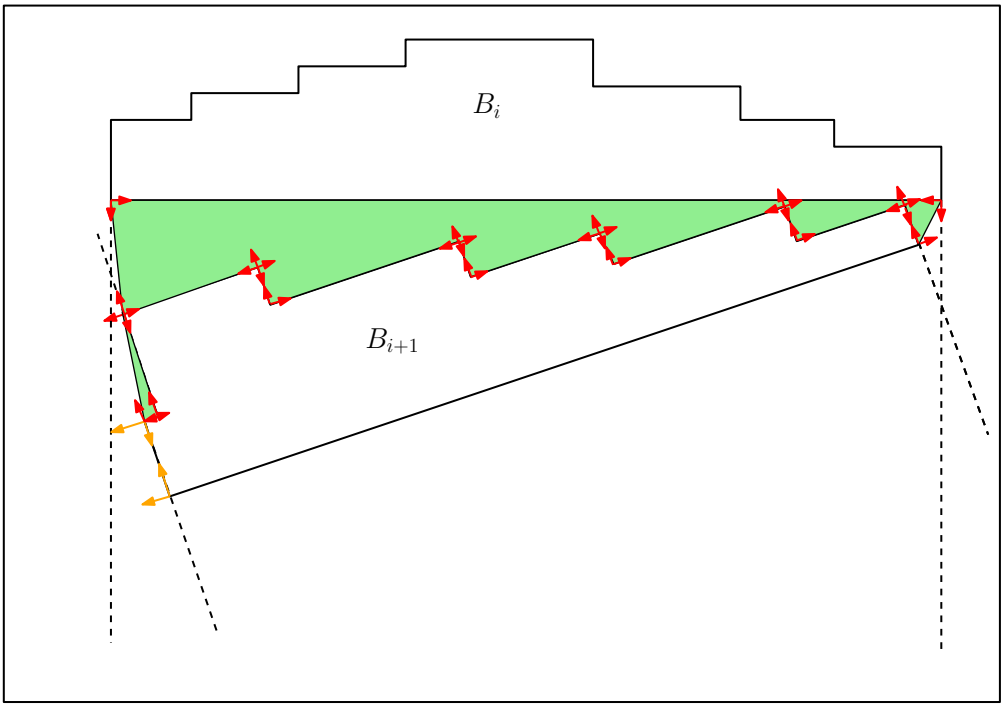


Figure 2.35: $\wp$ is shaded in green. Potential guard positions to cover $\wp$ are shown in red and potential guard positions to cover the w_L^{i+1} is shown in orange.

There are $k-1$ spaces $\wp_i$ in total, between consecutive holes ($i = 1, 2, \ldots, k-1$). Thus, $\sum_{i=1}^{k-1}(r_{i+1} - 2)$ guards are needed to guard their union. As argued, each left edge of a hole needs an additional guard.

For B_k, one guard is required to guard the bottom-most wall. For B_1, only two walls (left most and bottom most) are guarded by the above mentioned guards. In order to guard the rest of the walls, the minimum number of guards required is $r_1 - 2$. Therefore, in total at least $r_1 - 1 + \sum_{i=2}^{k}(r_i - 2) + k = r - k + 1$ guards are needed.

A $r-k+1$ guard placement, for $k=3$, is shown in Figure 2.34 and is obtained as follows. We start placing guards on B_1. All the edges involved in the construction of staircases FS_1 and RS_1 are not visible from any possible guard position on building $B_i, \forall i > 1$. Therefore, we place $r_1 - 1$ guards to cover these edges. Consider the space $\wp_1$ between B_1 and B_2. We need $r_2 - 2$ guards to cover $\wp_1$; let one of these guards be placed on B_1 and rest on B_2 (see Figure 2.34). After placing these guards, all walls of B_1 are visible, and $m_2 - 2$ walls of B_2 are visible. We need an additional guard to cover the left wall of B_2, and this guard does not cover any other wall in the city. Consider the space $\wp_2$ between the hole B_2 and B_3 and place $r_3 - 2$ guards to cover $\wp_2$; let one of these guards be placed on B_2 and rest on B_3. We continue this process, and $r_i - 1$ guards are placed on each hole $B_i \ \forall i > 2$. This results in $r-k+1$ guards as we place $r_i - 1$ guards on each hole $B_i \ , \forall i \in (1, k]$, and r_1 guards on B_1. Guard locations and directions are shown in Figure 2.34.

2.6.3 City Guarding

Theorem 33. *$r-k+1$ guards are sometimes necessary to guard an axis-aligned rectangle polygon with k disjoint orthogonal holes and r reflex vertices, with all guards placed on vertices of the holes.*

We conjecture the bound is tight. The proof follows from the proof of Theorem 32 by assuming that all the guards are facing along the wall instead of facing away from the wall.

CHAPTER 3

PATH CHECKING IN $\Re^2$

Given a set S of m disjoint simple polygonal obstacles in the plane, with a total of n vertices, the *path checking* problem is to preprocess S so that given a positive constant c and an u-to-v polygonal path with k edges, where k is much smaller than n (i.e., $k = o(n)$) one can quickly answer whether the path has clearance at least c.

3.1 Related Work

A simple, brute force solution to the path clearance checking problem would be to take each line segment along the path and find its distance (zero in case of intersections) to each of the line segments defining the boundary of the polygonal obstacles, which can be done in constant time per pair of segments. The clearance of the path would be reported as the minimum clearance over its line segments. For a path π with k line segments, among a set of polygonal obstacles with a total of n vertices, this leads to an $O(nk)$ time, $O(n+k)$ space solution that requires no preprocessing. This is linear in n and thus inefficient for a query type problem. It is good however to contrast this with results that can be extracted from using complex data structures, such as the Visibility-Voronoi Complex (VVC) (Wein et al., 2008). The Visibility-Voronoi diagram for clearance c, $VV^{(c)}$, introduced in (Wein et al., 2008), encodes the visibility graph of the obstacles dilated with a disc of radius c and can be used to compute paths of clearance c and other desired properties between two points u and v by a search in this graph. The Visibility-Voronoi complex is a generalization of $VV^{(c)}$, that allows to find u-to-v paths for any given clearance value c without having to first construct the $VV^{(c)}$, by performing a Dijkstra like search on the graph encoding the VVC. $VV^{(c)}$ and the VVC require $O(n \log n + n_1)$ preprocessing time and can report an u-to-v path of clearance at least c in $O(n \log n + n_2)$ time, where n_1 is the number of visibility edges and

n_2 is the number of edges of the diagram encountered during the search; both n_1 and n_2 are $O(n^2)$ in the worst case. However, neither $VV^{(c)}$ or VVC can be used directly to check whether a given path has clearance at least c, since the edges of the path are in general not encoded by the underlying graphs.

For finding a closest point to a query line, Cole and Yap (Cole and Yap, 1983) and Lee and Ching (Lee et al., 1985) reported a solution with preprocessing time and space in $O(n^2)$ and query time in $O(\log n)$. Mitra and Chaudhuri (Mitra and Chaudhuri, 1998) presented an algorithm with $O(n \log n)$ preprocessing time, $O(n)$ space, and $O(n^{0.695})$ query time. Mukhopadhyay (Mukhopadhyay, 2003), used the simplicial partition technique of Matoušek (Matoušek, 1992) to improve the query time to $O(n^{1/2+\epsilon}))$ for arbitrary $\epsilon > 0$, with $O(n^{1+\epsilon}))$ preprocessing time and $O(n \log n)$ space.

The problem of locating the nearest point to a query line segment among a set P of n points in the plane was addressed in (Bespamyatnikh and Snoeyink, 1999). If the query line segment is known to lie outside the convex hull of P, an $O(n)$ size data structure can be constructed in $O(n \log n)$ time, which can answer the nearest neighbor of a line segment in $O(\log n)$ time. If k non-intersecting line segments are given at a time, then the nearest neighbors of all these line segments can be reported in $O(k \log^3 n + n \log^2 n + k \log k)$ time using divide and conquer and the data structure for queries outside the convex hull. Later on, in (Bespamyatnikh, 2003), the time was reduced to $O(n \log^2 n)$ when $n = k$. Moreover, given n disjoint red segments and k disjoint blue segments in the plane, the algorithm in (Bespamyatnikh, 2003) can be used to find the closest pair of segments of a different color in $O((n + k) \log^2(n + k))$ time. Thus, with the red segments the edges of polygons in S and the blue segments the segments along the query path, the path-polygon proximity problem we study can be solved within $O((n + k) \log^2(n + k))$ time, without any preprocessing. Our goal is to obtain a query time that is sublinear in n, and thus more efficient for small values of k.

Goswami et al. (Goswami et al., 2004) reported an algorithm for closest point to line segment queries with $O(\log^2 n)$ query time and $O(n^2)$ preprocessing time and space, based on simplex range searching. Segal and Zeitlin (Segal and Zeitlin, 2008) provided an algorithm which takes $O(\log^2 n \log\log n)$ query time, using $O(n^2/\log n)$ space and $O(n^2)$ preprocessing time. However, these algorithms do not answer the segment-polygon proximity query problem as described here, as it is not enough to consider only the vertices of the polygons in S.

3.2 Results

For a set S of m disjoint simple polygonal obstacles in the plane, with a total of n vertices, the goal is to preprocess S so that given a positive constant c and an u-to-v polygonal path with k edges, where k is much smaller than n (i.e., $k = o(n)$) one can quickly answer whether the path has clearance at least c. We have the following results.

- We present an $O(t \log n)$ time, $O(t)$ space preprocessing, $O((n/\sqrt{t}) \log^{7/2} n)$ query time solution, for any $n^{1+\epsilon} \leq t \leq n^2$, to report the closest polygon in S to a query line segment.

- For a path π with k segments, we obtain $O((nk/\sqrt{t}) \log^{7/2} n)$ query time, with $O(t \log n)$ time, $O(t)$ space preprocessing, using segment-polygon proximity queries. When $t = n$ this gives $O(\sqrt{n}k \log^{7/2} n)$ query time with linear space and $O(n \log n)$ time preprocessing, improving over previous methods whenever k is small (i.e. $k = o(\sqrt{n})$). When $t = \Theta(n^2)$ it gives a query time of $O(k \log^{7/2} n)$, which is an $O(n)$ time improvement in query time over applying the solution derived from (Bespamyatnikh and Snoeyink, 1999; Bespamyatnikh, 2003), when k is small. Moreover, unlike (Bespamyatnikh and Snoeyink, 1999; Bespamyatnikh, 2003), our result is easily parallelizable, since queries with line segments along the query path are independent of each other. Thus, with k

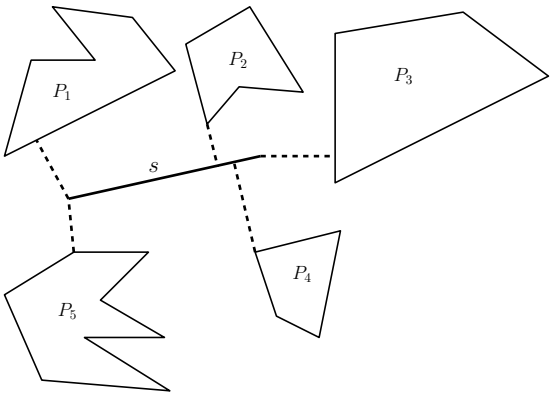


Figure 3.1: A set S of polygons, to be preprocessed for closest polygon to a query line segment (or line).

processors available, a query with a k segment path would take time proportional to the time to answer a line segment query. Assuming k is small, this can be easily implemented by multithreading (JAVA, C++) on modern laptop and desktop computers.

Our solutions differ from algorithms that could possibly be derived from existing visibility graph or Voronoi diagram based methods. They offer a preprocessing-query time trade-off and result in significant improvements when k is much smaller than n.

3.3 Line segment proximity queries

In a **nearest neighbor query** problem a set S of n geometric objects in $\Re^d$, d a positive integer constant, is preprocessed into a data structure so that the object of S closest to a query object (point, line, line segment, etc.) can be reported quickly. In this section we address nearest-neighbor queries in the plane, where the input S corresponds to a set of disjoint simple polygons and the query object corresponds to a line segment. This is illustrated in Figure 3.1.

Obviously, if the query object intersects a polygon in S then that polygon is a closest polygon and the closest distance from the query object to S is zero. Following this observation, a query can be divided into two parts, executed in this order:

1. **Emptiness Query**: Query if any polygon of S is intersected by the query object. If there is such polygon, then report it as the answer, with a distance of zero.

2. **Proximity Query**: (No polygon in S intersect the query object) Query for the closest polygon in S.

Thus, one can separately develop data structures for the two steps above, aiming for the best trade-offs on preprocessing-space-query on both structures.

Emptiness queries have been addressed in the context of ray shooting among polygonal obstacles in the plane. Chazelle et al. (Chazelle et al., 1994) gave an algorithm for ray shooting queries among m disjoint simple polygons with a total of n edges, with $O(n\sqrt{m} + m^{3/2}\log m + n\log n)$ preprocessing time, $O(\sqrt{m}\log n)$ query time, and $O(n)$ space. Obviously, ray shooting queries can be used to answer emptiness queries for both lines and line segments, within the same time and space bounds, by replacing each such query with two, respectively one, ray shooting queries.

Agarwal and Sharir (Agarwal and Sharir, 1996) develop data structures for ray shooting queries by first building data structures for line and line segment intersection queries. They first address line intersection queries and show that a set of m simple polygons with a total of n vertices can be preprocessed in time $O((m^2 + n\log m)\log n)$ into a data structure of size $O(m^2 + n)$ so that an intersection between a query line and the polygons can be detected in $O(\log n)$ time. Alternately, they give a data structure with $O(n\log n)$ preprocessing time, $O(n)$ space, and $O(\lceil m/\sqrt{n}\rceil^{1+\epsilon}\log n)$ query time. When $m \leq \sqrt{n}$ the query time becomes $O(\log n)$ while when $m \geq \sqrt{n}$ a query can be answered in time $O(\lceil m/\sqrt{t}\rceil^{1+\epsilon})$ with space t such that $n \leq t \leq m^2$. For line segment intersection queries for disjoint simple polygons they give a data structure of size $O((m^2 + n)\log m)$, that can be constructed in $O((m^2 + n)\log n\log m)$ time and can answer whether a query segment intersects any of the polygons in $O(\log m\log n)$ time. Alternately, they gave a data structure with $O(n\log^2 m)$

preprocessing time, $O(n \log m)$ space, and $O(\lceil m/\sqrt{n} \rceil^{1+\epsilon} \log^2 n)$ query time. For ray shooting among pairwise disjoint polygons, they give a data structure with $O(n \log n \log m)$ preprocessing time, $O(n)$ space, and $O(\lceil m/\sqrt{n} \rceil^{1+\epsilon} \log^5 n)$ query time.

We first warm up by providing a simple solution to finding the closest polygon to a query line in the following subsection. We then extend this approach to find the closest polygon to a query line segment.

3.3.1 Closest polygon to a query line

Given a set S of m disjoint simple polygons, with a total of n vertices, to find the closest polygon to a query line l we first perform an emptiness query with l, as described earlier. Using the result in (Agarwal and Sharir, 1996), this can be done with $O(n \log n)$ preprocessing time, $O(n)$ space, and $O(\lceil m/\sqrt{n} \rceil^{1+\epsilon} \log n)$ query time.

Observation 1. *Given a simple polygon P and a line l such that l does not intersect P, the closest point of P from l is a vertex of P.*

Assume that none of the polygons in S intersect the query line l. Based on Observation 1 to find the closest polygon in S to the query line l reduces to finding the closest point to a query line problem, where points corresponds to the vertices of the polygons in S. We further preprocess S by computing the convex hull of each polygon in S and taking the vertices of the convex hulls as the set of points. This requires only an additional $O(n)$ time and storage. Thus, we have a set of $n' \leq n$ points, where n' could be much smaller than n in practice.

We can then use the results in (Mukhopadhyay, 2003; Mitra and Chaudhuri, 1998) for closest point to line queries. Putting things together we obtain the following result.

Lemma 3. *A set S of m polygons, with a total of n vertices, can be preprocessed in $O(n^{1+\epsilon})$ time into a data structure of size $O(n \log n)$, that can report the closest polygon to a query line in $O(n^{(1/2)+\epsilon} + \lceil m/\sqrt{n} \rceil^{1+\epsilon} \log n)$ time, for arbitrary $\epsilon > 0$. Alternately, with $O(n \log n)$*

time preprocessing one can construct a data structure of size $O(n)$ that can report the closest polygon to a query line in $O(n^{0.695})$ time.

3.3.2 Closest polygon to a query line segment

Given a set S of m polygons, with a total of n vertices, to find the closest polygon to a query line segment s we first perform an emptiness query with s. To facilitate that, we preprocess S into a data structure for planar point location queries, which requires $O(n \log n)$ time and $O(n)$ space (Kirkpatrick, 1983). Given a segment s, we locate the endpoints of s in this data structure, in $O(\log n)$ query time; if any of the two endpoints is inside a polygon in S then we are done.

If both endpoints of s are in free space we proceed with a segment intersection query. As described earlier, this can be done with $O((m^2+n)\log m)$ space, $O((m^2+n)\log n \log m)$ preprocessing time, and $O(\log m \log n)$ query time or, alternately, with $O(\lceil m/\sqrt{n} \rceil^{1+\epsilon} \log^2 n)$ query time, $O(n \log^2 m)$ preprocessing time and $O(n \log m)$ space.

Assume that the line segment s does not intersect any polygon in S. Obviously, the closest distance from s to S is attained by a point on s and a point on a line segment on the boundary of some polygon in S.

Observation 2. *Consider a line segment e on the boundary of some polygon in S. The closest distance between s and e is either along:*

1. *A line perpendicular to s and passing through an endpoint of e, or*
2. *A line perpendicular to e and passing through an endpoint of s, or*
3. *A line joining an endpoint of s and an endpoint of e.*

This is illustrated in Figure 3.2 where AB is the query segment s and s_1, s_2, s_3, s_4 and s_5 are five line segments (see also (Wein et al., 2008)). Lines l_1 and l_2 are perpendicular to segment AB and passing through points A and B, respectively.

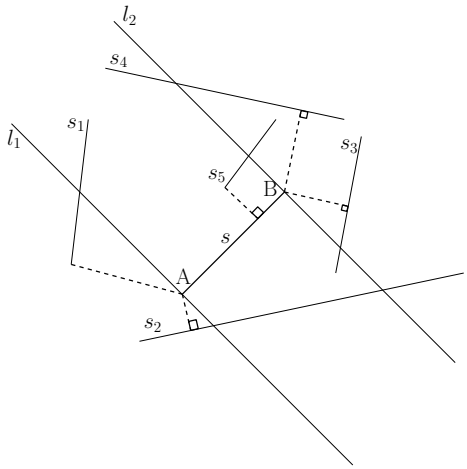


Figure 3.2: Illustrating the closest distance between query line segment AB and some other line segments.

Thus, we can focus on the n line segments on the boundaries of the polygons in S. Given a set M of n non-intersecting line segments, we want to build a data structure so that, for a query segment s with endpoints A and B, the closest segment of M can be quickly determined. From Observation 2, it is clear that the minimum distance involves an end point of at least one line segment. Therefore we can decompose this problem into the following two subproblems:

1. Find the line segment of set M closest to point A or point B.

2. Let l_1 and l_2 be the lines perpendicular to s at its endpoints and consider the endpoints of the line segments in M which lie between l_1 and l_2. Find the closest such endpoint to s.

Notice that for our purpose we could relax the second subproblem, and ask instead for finding the closest endpoint of M to the query segment s.

Subproblem 1: Given a set M of n line segments, preprocess M so that we can efficiently find the closest line segment to a query point q.

To answer Subproblem 1 we construct the Voronoi diagram of the line segments in M and preprocess it for point location queries. Yap (Yap, 1987) provided an $O(n \log n)$ time algorithm to construct the Voronoi diagram of non intersecting line segments. After constructing the Voronoi diagram, preprocessing for point location takes $O(n \log n)$ time with $O(n)$ storage, and a point location query can be answered in $O(\log n)$ time (Kirkpatrick, 1983).

Lemma 4. *A set M of n non-intersecting line segments can be preprocessed in $O(n \log n)$ time into a data structure of size $O(n)$ that can report the closest line segment to a query point in $O(\log n)$ time.*

For a given line segment s, *slab(s)* is defined as the region bounded by the lines l_1 and l_2 perpendicular to the endpoints of s and containing s (see (Bespamyatnikh and Snoeyink, 1999)).

Subproblem 2: Given a set P of n points, preprocess P into a data structure so that one can efficiently answer the following query: For a line segment s, find the closest point in $P \cap slab(s)$ to s. This is illustrated in Figure 3.3.

To solve this problem, we use a multilevel data structure based on Matousek's (Matoušek, 1993) decomposition scheme. Refer to Figure 3.3. Specifically, the first level is for halfplane range queries, to separates the points that are on the side of l_1 that contains s, and the second level is for halfplane range queries on the resulting points to separate those that are on the side of l_2 and contain s. These two levels are used to isolate the points in $P \cap slab(s)$. The third level is for halfplane range queries bounded by the line supporting s, to isolate the subsets of $P \cap slab(s)$ that are on either side of s. The subsets on this level are further processed for closest point to line segment queries, when the query segment is outside the convex hull of the points, as in (Bespamyatnikh and Snoeyink, 1999).

Using Theorem 6.1 from (Matoušek, 1993), we obtain the following trade-off.

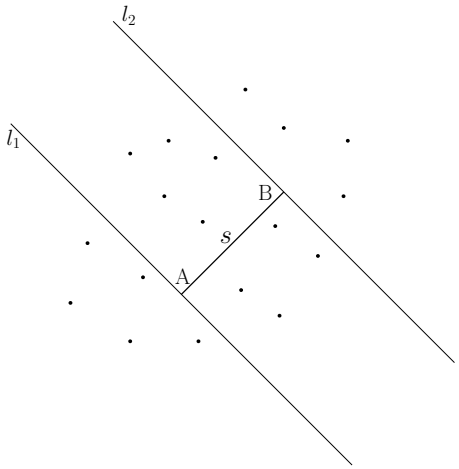


Figure 3.3: slab(s) and the points in $P \in slab(s)$.

Lemma 5. *A set P of n points in the plane can be preprocessed in $O(t \log n)$ time and $O(t)$ space into a data structure that can answer the query in Subproblem 2 in $O((n/\sqrt{t}) \log^{7/2} n)$ time, for any $n^{1+\epsilon} \leq t \leq n^2$ and $\epsilon > 0$.*

Summing up, we have the following result.

Lemma 6. *Given a set S of m disjoint simple polygons, it can be preprocessed in $O(t \log n)$ time into a data structure of size $O(t)$ that can report the closest polygon in S to a query line segment in $O((n/\sqrt{t}) \log^{7/2} n + \lceil m/\sqrt{n} \rceil^{1+\epsilon} \log^2 n)$ time, for any $n^{1+\epsilon} \leq t \leq n^2$ and $\epsilon > 0$.*

3.4 Closest polygon to path queries

We now turn our attention to finding the closest polygon to a query path. Given a set S of m disjoint simple polygons in the plane, with a total of n vertices, we want to preprocess S into a data structure so that for a query consisting of a positive constant c and a simple polygonal path π with k vertices, from a point u to a point v in free space, one can quickly decide whether there is no polygonal obstacle within distance c of π.

Our solution actually works even if π has self intersections, however we do not see the practical aspect of such paths.

To solve the path-polygons proximity query problem we proceed as follows.

Preprocessing. We build the following data structures.

1. A point location data structure D_1 for the polygons is S. It can be built with $O(n)$ space and $O(n \log n)$ time, and can answer point location queries in $O(\log n)$ time.

2. A segment intersection data structure D_2 for the polygons of S. As described earlier, this can be done with $O((m^2 + n) \log m)$ space, $O((m^2 + n) \log n \log m)$ preprocessing time, and $O(\log m \log n)$ query time or, alternately, with $O(\lceil m/\sqrt{n} \rceil^{1+\epsilon} \log^2 n)$ query time, $O(n \log^2 m)$ preprocessing time and $O(n \log m)$ space.

3. The Voronoi diagram of the polygons in S, enhanced with a point location data structure, D_3. It can be built with $O(n)$ space and $O(n \log n)$ time, and can answer point location and closest polygon to point queries in $O(\log n)$ time.

4. With P the set of vertices of the polygons in S, a data structure D_4 to find the closest point of $P \cap \mathit{slab}(s)$ to s, for a query segment s, as described in the previous section. It can be built with $O(t)$ space and $O(t \log n)$ time, and can answer a query in $O((n/\sqrt{t}) \log^{7/2} n)$ time, for any $n^{1+\epsilon} \leq t \leq n^2$ and $\epsilon > 0$.

Query. Given a simple polygonal path π with k vertices, to answer a query we proceed as follows.

1. Query D_1 with the vertices of π. If any such vertex is inside some polygon of S we stop and report it as the closest polygon to π, with a zero distance (or an *intersection* flag). Otherwise, all vertices of π are in free space and we proceed with the next step. This step takes $O(\log n)$ time per query and thus $O(k \log n)$ time overall.

2. Query D_2 with the line segments on π. If it is found that a line segment intersects a polygon in S then stop and report it as the closest polygon to π, with a zero distance (or an *intersection* flag).

3. Query $D3$ with the vertices of π and keep tract of the closest distance found. That distance gives the closest polygon of S to the vertices of π. This step takes $O(\log n)$ time per query and thus $O(k \log n)$ time overall.

4. Query $D4$ to find the closest obstacle vertex in $slab(s)$, for each segment s of π. This takes $O((n/\sqrt{t}) \log^{7/2} n)$ time per query and $O((kn/\sqrt{t}) \log^{7/2} n)$ time overall, for any $n^{1+\epsilon} \leq t \leq n^2$ and $\epsilon > 0$.

Theorem 34. *A set S of m disjoint simple polygons in the plane, with a total of n vertices, can be preprocessed in $O(t \log n)$ time into a data structure of size $O(t)$ so that given a query consisting of a positive value c and a simple polygonal path π with k vertices one can answer if π has clearance at least c in $O(k((n/\sqrt{t}) \log^{7/2} n + \lceil m/\sqrt{n} \rceil^{1+\epsilon} \log^2 n))$ time, for any $n^{1+\epsilon} \leq t \leq n^2$ and $\epsilon > 0$.*

When $t = n^2$ and $m = o(\sqrt{n})$ the query time becomes $o(k \log^{7/2} n)$, which is a linear time faster than what can be obtained from previous algorithms (Bespamyatnikh and Snoeyink, 1999; Bespamyatnikh, 2003) when k is much smaller than n. When $t = n^{1+\epsilon}$ the query time is $O(k(\sqrt{n} \log^{7/2} n + \lceil m/\sqrt{n} \rceil^{1+\epsilon} \log^2 n))$, which is asymptotically faster than what can be obtained from previous algorithms (Bespamyatnikh and Snoeyink, 1999; Bespamyatnikh, 2003) when $k = o(\sqrt{n})$.

3.5 Future Work

A possible extension of our work, that we leave as an open problem, is to query with paths that are not polygonal, but formed of, or including, circular arcs. This version has direct applications in minimally invasive surgery, for instruments formed of circular tubes (Morimoto et al., 2017). This problem seems significantly harder, even when the clearance c (diameter of the tube) is known in advance. We sketch a possible approach here and underline the missing data structures needed to address this version.

It is easy to see that the minimum distance between a circular arc σ and a line segment s could be achieved by a point $p \in \sigma$ and a point $q \in s$ neither of which is an endpoint of σ or s.

For the general version, with clearance given at query time one would need data structures for the following two types of queries: (1) circular arc intersection queries for disjoint polygons and (2) circular arc proximity queries for disjoint polygons. So far, neither of these data structures have been described in the computational geometry literature. There are however data structures for ray shooting queries among circular arcs (Agarwal et al., 1991), so if k is comparable to n one could instead answer ray shooting queries against π at query time. Such method however seems inefficient.

Consider now the case when the clearance c is known at preprocessing time. As before, we have a set S of m disjoint simple polygons in the plane, with a total of n vertices. In addition, we also know the clearance c, given as a positive real value. A query consists of a path π with k circular arcs and asks whether π has clearance at least c. To solve it, one can proceed as follows.

Preprocessing. Build the following data structures.

1. Find the Minkowski sum of the obstacles in S with a disk of radius c and compute the union Γ of the resulting objects, which can be done with $O(n)$ space and $O(n \log^2 n)$ time (Wein et al., 2008). The boundary of Γ consists of both line segments and circular arcs. Further process Γ for point location queries, for an additional $O(n)$ space and $O(n \log n)$ time. A point location query in the resulting data structure D_1 can be answered in $O(\log n)$ time.

2. Preprocess Γ for circular ray shooting queries: given a circular arc σ determine the first line segment or arc on the boundary of Γ hit by σ. Notice that the radius for the query circular arc is known in advance.

Thus, in this case, we are dealing only with a special case of circular ray shooting queries among disjoint line segments and circular arcs, where the radius of all circular arcs given at preprocessing time is the same. Still, we are not aware of any data structure that can efficiently handle such queries.

CHAPTER 4

CONCLUSION

In this dissertation, we considered city guarding with limited field of view and path clearance verification in $\Re^2$.

Firstly, we considered the problem of guarding a city consisting of k vertical buildings $B_1, \ldots, B_k$, each having $m_1, \ldots, m_k$ roof vertices, respectively, with $\sum_{i=1}^{k} m_i = m$, by placing guards at the top corners of the buildings, where a guard can only see the space in front of it. We considered the following versions of the city guarding problem:

1. Buildings are axis-aligned and have a rectangular base,
2. Buildings are axis-aligned and have an orthogonal base,
3. Buildings have arbitrary orientation and have a rectangular base,
4. Buildings have arbitrary orientation and have an orthogonal base.

For all the versions, we showed that k guards are always sufficient and sometimes necessary to guard the roofs of a city.

For axis-aligned buildings, having a rectangular base, we proved that $2k + \lfloor \frac{k}{4} \rfloor + 4$ guards are always sufficient to guard the whole city (walls, ground, and the roofs). For axis-aligned buildings, each having an orthogonal base, we proved that $\frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 4$ guards are always sufficient to guard the whole city (walls, ground, and the roofs).

We also obtained the following results:

1. Given k pairwise disjoint isothetic rectangles in the plane, $2k + \lfloor \frac{k}{4} \rfloor + 4$ vertex guards are always sufficient to guard the free space.
2. Given k pairwise disjoint axis-aligned orthogonal polygons in the plane, $\frac{m}{2} + \lfloor \frac{k}{4} \rfloor + 4$ vertex guards are always sufficient to guard the free space.

We provided a divide and conquer approach to find the upper bound on the number of guards for a special case of the art gallery with holes.

For non-axis-aligned buildings, each having a rectangular base, we showed that $3k + 1$ guards are sometimes necessary to guard the walls and ground, and also the whole city. For non-axis-aligned buildings, each having an orthogonal base, we showed that $r - k + 1$ guards are sometimes necessary to guard the city with k buildings having r reflex vertices and conjecture that the bound is tight.

Secondly, we considered the problem of finding the closest polygon of a set S of disjoint simple polygons to a query line segment or simple polygonal path. We proposed solutions that are significantly better in query time, when k is small relative to n, than what could be obtained from existing, non-query based approaches. Since queries with line segments along the query path are independent of each other, our result is easily parallelizable: with k processors available, a query with a k segment path would take time proportional to the time to answer a line segment query. When k is small, this can be easily implemented by multithreading (JAVA, C++) on modern laptop and desktop computers.

REFERENCES

(2020, Feb). Drones and the smart city.

Abello, J., V. Estivill-Castro, T. Shermer, and J. Urrutia (1998). Illumination of orthogonal polygons with orthogonal floodlights. *International Journal of Computational Geometry & Applications 8*(01), 25–38.

Agarwal, P. K. and M. Sharir (1996). Ray shooting amidst convex polygons in 2d. *Journal of Algorithms 21*(3), 508–519.

Agarwal, P. K., M. van Kreveld, and M. Overmars (1991). Intersection queries for curved objects. In *Proceedings of the seventh annual symposium on Computational geometry*, pp. 41–50. ACM.

Aggarwal, A. (1984). The art gallery theorem: its variations, applications and algorithmic aspects.

Bao, L., S. Bereg, O. Daescu, S. Ntafos, and J. Zhou (2008). On some city guarding problems. In *International Computing and Combinatorics Conference*, pp. 600–610. Springer.

Batista, V. (2010). On the complexity of the edge guarding problem. In *Proc. 26th European Workshop on Computational Geometry, Dortmund, Germany, 2010*, pp. 53–56.

Benbernou, N., E. D. Demaine, M. L. Demaine, A. Kurdia, J. O'Rourke, G. Toussaint, J. Urrutia, and G. Viglietta (2011). Edge-guarding orthogonal polyhedra. In *Proceedings of the 23rd Canadian Conference on Computational Geometry*, pp. 461–466.

Bespamyatnikh, S. (2003). Computing closest points for segments. *International Journal of Computational Geometry & Applications 13*(05), 419–438.

Bespamyatnikh, S. and J. Snoeyink (1999). Queries with segments in voronoi diagrams. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 122–129. Society for Industrial and Applied Mathematics.

Bjorling-Sachs, I. and D. L. Souvaine (1995). An efficient algorithm for guard placement in polygons with holes. *Discrete & Computational Geometry 13*(1), 77–109.

Blanco, G., H. Everett, J. G. Lopez, and G. Toussaint (1994). Illuminating the free space between quadrilaterals with point light sources. In *PROC. COMPUTER GRAPHICS INT.,. WORLD SCIENTIFIC*. Citeseer.

Bose, P., D. Kirkpatrick, and Z. Li (2003). Worst-case-optimal algorithms for guarding planar graphs and polyhedral surfaces. *Computational Geometry 26*(3), 209–219.

Bose, P., T. Shermer, G. Toussaint, and B. Zhu (1997). Guarding polyhedral terrains. *Computational Geometry 7*(3), 173–185.

Cano, J., C. D. Tóth, and J. Urrutia (2012). Edge guards for polyhedra in three-space. In *Proceedings of 24th Canadian Conference on Computational Geometry (CCCG)*, pp. 163–167.

Chazelle, B. (2000). The soft heap: an approximate priority queue with optimal error rate. *Journal of the ACM (JACM) 47*(6), 1012–1027.

Chazelle, B., H. Edelsbrunner, M. Grigni, L. Guibas, J. Hershberger, M. Sharir, and J. Snoeyink (1994). Ray shooting in polygons using geodesic triangulations. *Algorithmica 12*(1), 54–68.

Chvatal, V. (1975). A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory, Series B 18*(1), 39–41.

Cole, R. and M. Sharir (1989). Visibility problems for polyhedral terrains. *Journal of symbolic Computation 7*(1), 11–30.

Cole, R. and C. K. Yap (1983). Geometric retrieval problems. In *Foundations of Computer Science, 1983., 24th Annual Symposium on*, pp. 112–121. IEEE.

Czyzowicz, J., E. Rivera-Campo, J. Urrutia, and J. Zaks (1994). Protecting convex sets. *Graphs and Combinatorics 10*(2-4), 311–321.

Czyzowicz, J., E. Riveracampo, and J. Urrutia (1993). Illuminating rectangles and triangles in the plane. *Journal of Combinatorial Theory, Series B 57*(1), 1–17.

de la Calle, J. G. L. (1995). *Problemas algorítmico-combinatorios de visibilidad*. Ph. D. thesis, Universidad Politécnica de Madrid.

Estivill-Castro, V. and J. Urrutia (1994). Optimal floodlight illumination of orthogonal art galleries. In *CCCG*, pp. 81–86.

Everett, H. and E. Rivera-Campo (1994). Edge guarding a triangulated polyhedral terrain. In *CCCG*, pp. 293–295.

Everett, H. and G. Toussaint (1990). On illuminating isothetic rectangles in the plane,".

Fisk, S. (1978). A short proof of chvátal's watchman theorem. *J. Combinatorial Theory (B) 24*, 374.

Goodchild, M. F. and J. Lee (1989). Coverage problems and visibility regions on topographic surfaces. *Annals of Operations Research 18*(1), 175–186.

Goswami, P. P., S. Das, and S. C. Nandy (2004). Triangular range counting query in 2d and its application in finding k nearest neighbors of a line segment. *Computational Geometry 29*(3), 163–175.

Hoffmann, F. (1990). On the rectilinear art gallery problem. In *International Colloquium on Automata, Languages, and Programming*, pp. 717–728. Springer.

Hoffmann, F. and M. Kaufmann (1990). On the rectilinear art gallery problem algorithmic aspects. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pp. 239–250. Springer.

Hoffmann, F., M. Kaufmann, and K. Kriegel (1991). The art gallery theorem for polygons with holes. In *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pp. 39–48. IEEE.

Hoffmann, F. and K. Kriegel (1996). A graph-coloring result and its consequences for polygon-guarding problems. *SIAM Journal on Discrete Mathematics 9*(2), 210–224.

Iwamoto, C., J. Kishi, and K. Morita (2012). Lower bound of face guards of polyhedral terrains. *Information and Media Technologies 7*(2), 737–739.

Iwamoto, C. and T. Kuranobu (2012). Improved lower and upper bounds of face guards of polyhedral terrains. *IEICE Trans. Inf. & Syst.(Japanese Edition) 95*, 1869–1872.

J. Czyzowicz, J. U. (1996). Personal communication.

Kahn, J., M. Klawe, and D. Kleitman (1983). Traditional galleries require fewer watchmen. *SIAM Journal on Algebraic Discrete Methods 4*(2), 194–206.

Kirkpatrick, D. (1983). Optimal search in planar subdivisions. *SIAM Journal on Computing 12*(1), 28–35.

Klee, V. (1969). Is every polygonal region illuminable from some point? *The American Mathematical Monthly 76*(2), 180–180.

Lee, D., Y. Ching, et al. (1985). The power of geometric duality revisited. *Info. Processing Letters 21*, 117–122.

Lubiw, A. (1985). Decomposing polygonal regions into convex quadrilaterals. In *Proceedings of the first annual symposium on Computational geometry*, pp. 97–106. ACM.

Martini, H. and V. Soltan (1999). Survey paper. *Aequationes Math 57*, 121–152.

Matoušek, J. (1992). Efficient partition trees. *Discrete & Computational Geometry 8*(3), 315–334.

Matoušek, J. (1993). Range searching with efficient hierarchical cuttings. *Discrete & Computational Geometry 10*(2), 157–182.

Michael, T. and V. Pinciu. Guarding orthogonal polygons with h holes: A bound independent of h.

Mitchell, J. S. (1996). Shortest paths among obstacles in the plane. *International Journal of Computational Geometry & Applications 6*(03), 309–332.

Mitchell, J. S. (2000). Geometric shortest paths and network optimization. *Handbook of computational geometry 334*, 633–702.

Mitra, P. and B. Chaudhuri (1998). Efficiently computing the closest point to a query line1. *Pattern Recognition Letters 19*(11), 1027–1035.

Morimoto, T. K., J. J. Cerrolaza, M. H. Hsieh, K. Cleary, A. M. Okamura, and M. G. Linguraru (2017). Design of patient-specific concentric tube robots using path planning from 3-d ultrasound. In *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE*, pp. 165–168. IEEE.

Mukhopadhyay, A. (2003). Using simplicial partitions to determine a closest point to a query line. *Pattern Recognition Letters 24*(12), 1915–1920.

O'Rourke, J. (1983a). An alternate proof of the rectilinear art gallery theorem. *Journal of Geometry 21*(1), 118–130.

O'Rourke, J. (1983b). Galleries need fewer mobile guards: a variation on chvátal's theorem. *Geometriae Dedicata 14*(3), 273–283.

O'Rourke, J. (1987). *Art gallery theorems and algorithms*, Volume 57. Oxford University Press Oxford.

Sack, J. and O. An (1982). Algorithm for decomposing simple rectilinear polygons into convex quadrilaterals,". In *Proceedings of the 20th Allerton Conference on Communication, Control, and Computing, Monticello*, pp. 64–74.

Santos, F. (1995). Personal communication.

Segal, M. and E. Zeitlin (2008). Computing closest and farthest points for a query segment. *Theoretical Computer Science 393*(1-3), 294–300.

Shermer, T. C. (1992). Recent results in art galleries. *PROCEEDINGS-IEEE 80*, 1384–1384.

Souvaine, D. L., R. Veroy, and A. Winslow (2011). Face guards for art galleries. In *Proceedings of the XIV Spanish Meeting on Computational Geometry*, pp. 39–42. Citeseer.

Storer, J. A. and J. H. Reif (1994). Shortest paths in the plane with polygonal obstacles. *Journal of the ACM (JACM) 41*(5), 982–1012.

Tóth, C. D. (2000). Art gallery problem with guards whose range of vision is 180. *Computational Geometry: Theory and Applications 3*(17), 121–134.

Tóth, C. D. (2002). Art galleries with guards of uniform range of vision. *Computational Geometry 21*(3), 185–192.

Tóth, C. D. (2003). Guarding disjoint triangles and claws in the plane. *Computational Geometry 25*(1-2), 51–65.

Tóth, L. F. (1977). Illumination of convex discs. *Acta Mathematica Hungarica 29*(3-4), 355–360.

Urrutia, J. (2000). Art gallery and illumination problems. In *Handbook of computational geometry*, pp. 973–1027. Elsevier.

Urrutia, J. (2004). Art gallery and illumination problems.

Viglietta, G. (2015). Reprint of: Face-guarding polyhedra. *Computational Geometry: Theory and Applications 48*(5), 415–428.

Viglietta, G. (2017). Optimally guarding 2-reflex orthogonal polyhedra by reflex edge guards. *arXiv preprint arXiv:1708.05469*.

Wein, R., J. Van Den Berg, and D. Halperin (2008). Planning high-quality paths and corridors amidst obstacles. *The International Journal of Robotics Research 27*(11-12), 1213–1231.

Weina, R., J. P. van den Bergb, and D. Halperina (2007). The visibility–voronoi complex and its applications. *Computational Geometry 36*, 66–87.

Yap, C. K. (1987). Ano (n logn) algorithm for the voronoi diagram of a set of simple curve segments. *Discrete & Computational Geometry 2*(4), 365–393.

Żyliński, P. (2006). Orthogonal art galleries with holes: A coloring proof of aggarwal's theorem. *the electronic journal of combinatorics 13*(1), 20.

BIOGRAPHICAL SKETCH

Hemant Malik completed his PhD in Computer Science at The University of Texas at Dallas under the supervision of Dr. Ovidiu Daescu. His research focuses on Combinatorial Optimization, Graph Algorithms, and Computational Geometry. He has worked on various problems that set an essential foundation of surveillance and security using drones.

Along with his research, he was an instructor for Automata Theory (CS 4384) and awarded the best teaching assistant during 2018-2019 for his outstanding performance. During Summer 2019, he interned at Facebook as a PhD Research Software Engineer.

He came to the United States for further education after completing his bachelor in Computer Science from the Indian Institute of Information Technology Design and Manufacturing. He started his master's in Computer Science from The University of Texas at Dallas. During his master's, he felt very passionate about algorithms and decided to pursue a PhD. He feels very excited while solving his research problems and works very hard to find the solution. According to him, there is no set recipe for success. It is a series of trying, failing, and adapting.

CURRICULUM VITAE

Hemant Malik

malik@utdallas.edu · https://www.linkedin.com/in/malikhemant/ ·

Publications

1. Daescu, Ovidiu, and Hemant Malik. City Guarding with Limited Field of View. [Accepted at CCCG-2020]

2. Daescu, Ovidiu, Stephan Friedrichs, Hemant Malik, Valentin Polishchuk, and Christiane Schmidt. "Altitude terrain guarding and guarding uni-monotone polygons." Computational Geometry (2019).

3. Daescu O., Malik H. (2019) k-Maximum Subarrays for Small k: Divide-and-Conquer Made Simpler. Analysis of Experimental Algorithms. SEA 2019. Lecture Notes in Computer Science, vol 11544. Springer, Cham

4. Daescu, Ovidiu, and Hemant Malik. "Does a robot path have clearance c?." International Conference on Combinatorial Optimization and Applications. Springer, Cham, 2018.

Education

- PhD in Computer Science 2016-2020
  The University of Texas at Dallas

- Masters in Computer Science 2014-2020
  The University of Texas at Dallas

- Bachelors in Computer Science 2010-2014
  Indian Institute of Information and Technology, Jabalpur, India

Experience

1. PhD Research Software Engineer Internship May 2019-August 2019
   Facebook, Menlo Park-CA

2. Instructor for Automata Theory 2018 – May 2019, August 2019 – May 2020
   The University of Texas at Dallas

   (a) Awarded outstanding performance during 2018-2019 for best Teaching Assistant.