M. Sc. Engineering Thesis

Algorithms for Minimum Length Sliding Camera Problem

By Mohammad Al-
Mahmud Student no.:
0413052096

Submitted to

Department of Computer Science and Engineering

in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science & Engineering

Department of Computer Science and Engineering Bangladesh
University of Engineering and Technology(BUET) Dhaka-
1000, Bangladesh

February, 2016

The thesis titled "Algorithms for Minimum Length Sliding Camera Problem", submitted by Mohammad Al Mahmud, Roll No. 0413052096, Session April 2013, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science and Engineering and approved as to its style and contents. Examination held on 19 March 2016.

Board of Examiners

1. ______________________________

Dr. Md. Saidur Rahman — Chairman
Professor — (Supervisor)
Department of CSE, BUET, Dhaka 1000.

2. ______________________________

Dr. M. Sohel Rahman — Member
Professor & Head — (Ex-officio)
Department of CSE, BUET, Dhaka 1000.

3. ______________________________

Dr. Md. Abul Kashem Mia — Member
Professor
Department of CSE, BUET, Dhaka 1000.

4. ______________________________

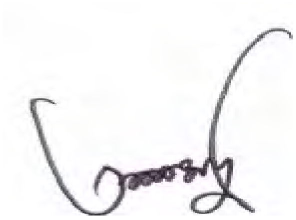
Dr. Sadia Sharmin — Member
Assistant Professor
Department of CSE, BUET, Dhaka 1000.

5. ______________________________

Dr. Md. Rafiqul Islam — Member
Professor — (External)
Department of CSE,
Khulna University, Khulna

Candidate's Declaration

This is to certify that the work presented in this thesis entitled "Algorithms for Minimum Length Sliding Camera Problem" is the outcome of the investigation carried out by me under the supervision of Professor Dr. Md. Saidur Rahman in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka. It is also declared that neither this thesis nor any part thereof has been submitted or is being currently submitted anywhere else for the award of any degree or diploma.



Mohammad Al Mahmud
Candidate

Contents

List of Figures

Acknowledgments

First I would like to thank my supervisor Professor Dr. Md. Saidur Rahman for introducing me to the field of graph theory. I have learned a lot on research methodology, scientific writing and many other research relevant matters from him. I thank him also for his patience in reviewing my work.

I would like to thank all the members of the examination board, Dr. Md. Mahfuzul Islam, Dr. Md. Abul Kashem Mia, Dr. Sadia Sharmin and Dr. Md. Rafiqul Islam, for their valuable comments.

I would also like to thank Dr. Md. Iqbal Hossain, Shahina Sultana, Nazmun Nessa Moon, Abu Reyan ahmed and all the members of my research group for their valuable suggestions, assistance and continual encouragements.

Special gratitude goes to my departed mother who during her life helped me in my personal commitment and encouraged me to give full concentration on my work. My wife and friends also supported me to their best. My heart-felt gratitude goes to them also.

Abstract

Art gallery problem is a well known visibility problem where objective is to cover the whole gallery by the minimum number of cameras. In computa- tional geometry galleries are represented by two dimensional polygons. There are many variants of the art gallery problem. One such variant is the minimum length sliding camera (MLSC) problem which uses sliding cameras that travel along the boundary of the polygon and cover orthogonally inside it. Here the objective is to cover the whole polygon by traversing the minimum length in the polygon's boundary. There is an algorithm which solves MLSC problem in $O(n^2)$ time for orthogonal polygons. In this thesis we show that for some subclasses of orthogonal polygons one major step of that algorithm shows lower time complexity. So far, all the results of the art gallery problem with sliding cameras are for orthogonal polygons. But in reality some non-orthogonal edges may be incorporated in the polygon. Based on this requirement we develop an algorithm that solves MLSC problem for some subclasses of semi-orthogonal polygons in $O(n^2)$ time. The class of semi-orthogonal polygons is a superclass of orthogonal polygons. As a byproduct of our work, we establish some relations among different components of an orthogonal polygon after it is being rectan- gulated. Advancement in the wireless technology introduces new variants in the art gallery problem. One such variant is MLSCk problem. We consider few modifications in the MLSCk problem and termed it as modified MLSCk prob- lem. Here sliding k-transmitters are used which have infinite broadcast range, can penetrate at most k number of walls (k is an integer and $k > 0$), travel along the boundaries of an orthogonal polygon and can cover orthogonally inside the polygon. The objective is to find the minimum-length sliding k-transmitters that cover the entire orthogonal polygon. We develop an algorithm which finds the minimum length cover in $O(n^2)$ time.

Chapter 1

Introduction

Computational Geometry involves study of algorithms for solving geometric problems by a computer. The field of computational geometry focuses mostly on problems in 2-dimensional space and to a lesser extent in 3-dimensional space. It primarily deals with straight or flat objects like lines, line segments, polygons, planes and polyhedra or simple curved objects such as circles. The main impetus for the development of computational geometry as a discipline was progress in computer graphics, computer aided design and computer aided manufacturing (CAD/CAM). But many problem in computational geometry are classical in nature, and may come from mathematical visualization.

Art gallery problem or museum problem is a well-studied visibility problem in computational geometry. It originates from a real-world problem of guarding an art gallery with the minimum number of guards who together can observe the whole gallery. Figure 1.1 illustrates the art gallery problem. Two dimen-

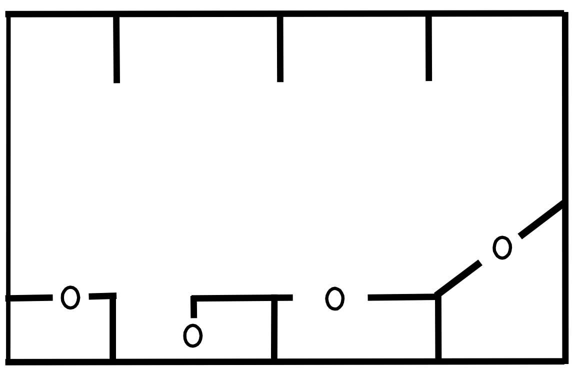


Figure 1.1: An illustration of art gallery problem.

sional drawing in this figure represents an art gallery. Whole gallery divided into some rooms. Objective of the problem is to guard the gallery by the minimum number of cameras. The circles inside the gallery represents cameras to guard. For this particular gallery, at least four cameras are required to cover the entire region. In the computational geometry version of the problem the layout of a art gallery is represented by a simple polygon and each guard is represented by a point in the polygon. A set S of points is said to guard a polygon if, for every point p in the polygon, there is some $q \in S$ such that the line segment between p and q does not leave the polygon. The rest of this chapter is organized as follows. In the next section we present some historical aspects of art gallery problem. In Section 1.2, we define the MSC and MLSC problems, which is the central idea of this thesis. Section 1.3 describes some interesting applications of MLSC problem. In Section 1.4, we present relevant results on MSC and MLSC problems. In Section 1.5, we describe the scope and objective of this thesis. The last section i.e. Section 1.6, presents the organization of this thesis.

1.1 Historical Background

The original art gallery problem, presented to Chvtal by Victor Klee [15] to find the smallest number of guards require to cover a polygon of n edges. In 1975 Chvtal [6] stated, occasionally necessary and always sufficient number of cameras to cover an n vertex polygon. In 1975 Steve Fisk gave a proof on Chvtal theorem by polygon triangulation and graph coloring. Avis and Toussaint [3] in 1981 using the basic idea of Fisk, developed an O(nlogn) algorithm for lo- cating those stationary guards. Khan, Klaw and Kleitman [11] in 1983 first investigate the art gallery problem for orthogonal or rectilinear polygons. In

1995 Schuchardt and Hecker [18] showed that art gallery problem is NP-Hard for orthogonal polygons.

There are numerous variations in the original problem that are also referred to as art gallery problem. One such variants of art gallery involve mobile guards. Here the requirement is that every point of gallery is visible by some guards at some points along his path. Kay and Guay [10] gave an algorithm for the problem of determining whether a given polygon can be guarded by a single guard patrolling along a single line segment. In 2011 Katz and Morgenstern [12] introduced a new problem on mobile guard i.e. the minimum sliding camera (MSC) problem. Later in 2013 Durocher and Mehrabi [8] changed the objective of MSC problem and introduced another problem which they termed as the minimum length sliding camera (MLSC) problem. Our research work is focused on art gallery problem with sliding camera variant.

1.2 Covering Polygon by Sliding Cameras

Let P be a simple orthogonal polygon. An orthogonal line segment $s \subseteq P$ is called segment guard or seguard. A guard s sees a point $p \in P$ if there exist a point $q \in S$, such that the line segment $\overline{pq}$ is orthogonal and contained in P. Let $v(s)$ the region of P that is seen by a seguard s and say a seguard set S guards P if $\bigcup v(s)|s \in S = P$. The objective of MSC problem is to find such the set of minimum cardinality. We can consider seguard s as a security camera sliding back and forth along horizontal or vertical track. It can see orthogonally inside the polygon along its traveling track. Figure 1.2 (b) describes the MSC problem for an input orthogonal polygon given in Figure 1.2 (a). Here only two cameras are required to cover the whole polygon. In

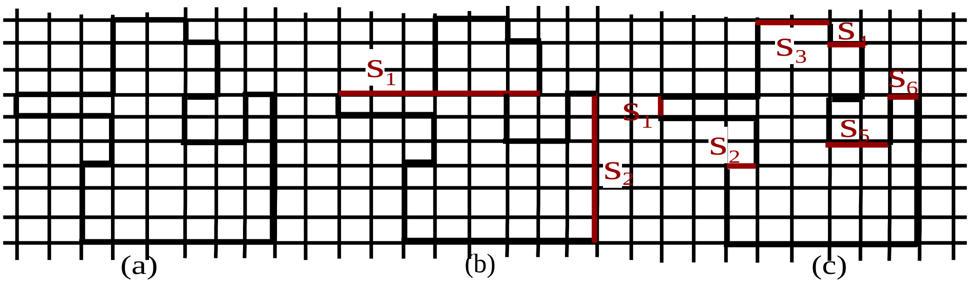


Figure 1.2: An illustration of MSC and MLSC problem. Each grid cell has size 1x1. (a) a simple orthogonal polygon P. (b) two sliding cameras can cover the whole polygon in MSC problem (c) six sliding cameras required for MLSC problem.

the MLSC problem objective changed to the minimum sliding length from the minimum number of sliding cameras (MSC). Here the minimum number of guards is not the constraint rather total traveling length by the cameras is the constraint factor. A sliding camera travels back and forth along an orthogonal line segment $s \subseteq P$. The camera (i.e. the guarding line segment s) can see a point $p \in P$ (equivalently, p is orthogonally visible to s) if and only if there exists a point q on s such that pq is normal to s and is completely contained in P . Objective of the problem is to cover the whole polygon minimizing the total length of trajectories along which the camera travels. Figure 1.2(c) describes the MLSC problem for an input polygon given in Figure 1.2(a). For MLSC problem total six cameras are required to cover the whole polygon. In the

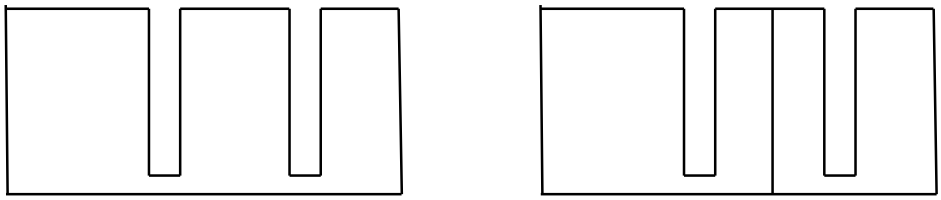


Figure 1.3: An illustration of MLSCk problem for $k = 2$. (a) a simple orthogonal polygon P . (b) position of minimum-length sliding k-transmitters(dotted lines).

MLSCk problem sliding k-transmitters are used. Sliding k-transmitters travel back and forth along axis aligned line segment s inside P , have infinite omni-directional broadcast range and can penetrate up to k number of walls(for a fixed integer $k > 0$). A point p is covered by this guard if there exist a point $q \in s$ such that $\overline{pq}$ is a line segment normal to s and is completely inside P . Again $\overline{pq}$ has at most k intersections with the polygon's boundary walls. The objective is to find minimum-length sliding k-transmitters that cover the entire P . In other words the goal is to find the minimum total length of trajectories on which the sliding k-transmitters travel to cover the entire polygon. Figure 1.3(b) indicates the position of minimum-length sliding k-transmitters by the dotted lines for the input orthogonal polygon described in Figure 1.3(a)

1.3 Applications

The most extensive use of MSC and MLSC problems is in the field of robotics. Scientist are always been interested by the idea of interaction between a robot and its environment. This kind of robots are called autonomous robot. Close and effective interaction between robots, environment and human are very important to the success of autonomous robot [20]. Sliding sensors can be used to control the movement of autonomous robot inside a museum or gallery. While moving these sensors can instantly gather surrounding informations also. By the minimum sliding length we can find the shortest path movement of an au- tonomous robot. Again by the minimum sliding cameras we can reduce the number of sensors to control the autonomous robot.

Beside robotics MSC and MLSC problems have several many applications in practice like placing security cameras in showroom, arranging the lighting source, placing sensors or placing radar station in mountain area.

1.4 Previous Results

As stated in Section 1.1 Katz and Morgenstern [12] in 2011 gave a $O(n^2)$ time solution for the MSC problem using only vertical or horizontal track. Beside that they presented a 3-approximation algorithm for the same problem if both the orientations are allowed. In 2013 Durocher et al. [7] presented a $O(n^{5/2})$- time and (7/2)-approximation algorithm for this problem. Later Durocher and Mehrabi [8] introduced the MLSC problem with a $O(n^2)$ time solution even for orthogonal polygons with holes. In that work they showed that the MSC prob- lem is NP-hard when P is allowed with holes. De Berg et al. [5] gave a linear- time dynamic programming algorithm for the MSC problem on x-monotone orthogonal polygons. More generally their algorithm can be used to solve the MSC problem in linear time on a simple orthogonal polygon P for which the dual graph induced by the vertical decomposition of P is a path. There result was first polynomial-time exact algorithm for MSC problems on a non-trivial subclass of orthogonal polygons. Durocher and Mehrabi in 2014 [9] gave an improvement from their earlier solution of MSC problem. They gave an $O(n^3)$- time 3-approximation algorithm for the MSC problem on any simple orthogonal polygon with n vertices. Mahdavi et al. [13] in 2014 introduced sliding cameras with k-transmitter. Such a guard can travel back and forth along and line seg- ments like earlier sliding camera but difference is that it can penetrate k no of boundary walls of the polygon. The objective is to minimize the sum of traveling length of the sliding k-transmitters to cover the entire polygon. They showed that this problem is NP-complete and presented a 2-approximation algorithm for it. In 2009 Fabila-Monroy et al. [17] first gave the concept of k -transmitter. In 2013 Ballinger et al. [4] develop a lower and upper bounds for the number of k -transmitters that are necessary and sufficient to cover a given collection of line segments, polygonal chains and polygons. In 2015 Aichholzer et al. [2] gave some restrictions on the number of k-transmitters for monotone polygons and monotone orthogonal polygons.

1.5 Scope of this Thesis

Though the existing algorithm solves the MLSC problem in $O(n^2)$ time but during our research we found that few of the steps of the existing algorithm can be improved for some subclasses of orthogonal polygons. Beside this we notice that all the existing work on sliding camera problem take orthogonal polygons as input. But in practice geometric shapes are not always orthogonal. Some non-orthogonal edges may be incorporated there. It encourages us to give an effort to increase the type of input polygon for the same problem. While working on MLSC problem we found some important relations among different components of an orthogonal polygon, after it is being rectangulated by a rectangulation technique. We present those relations expecting that these relations may contribute in the future research work. We introduce a new problem modifying the MLSCk problem and we develop an algorithm for the new problem. The main results of this thesis are as follows.

- An improved algorithm that solves MLSC problem for monotone orthogonal polygons.
- More improved algorithm that solves MLSC problem for FAT and MIN AREA grid n-ogons.
- An algorithm that solves MLSC problem for sub-classes of semi-orthogonal polygons.
- Relations among different components of an orthogonal polygon after it is being rectangulated by a rectangulation technique.
- An algorithm that solves modified MLSCk problem for orthogonal polygons.

1.6 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, we give some basic terminologies of polygon, graph theory and algorithm analysis. In Chapter 3, we present the improvement of the existing algorithm for some subclasses of orthogonal polygons. In Chapter 4, we describe the algorithm on MLSC problem for semi-orthogonal polygons as input. In this chapter we also present some relations between different components of orthogonal polygons when it is be- ing rectangulated by a rectangulation technique. In Chapter 5 we present an algorithm on MLSCk problem with few modifications. Finally, Chapter 6 sum- marizes our contribution, discusses the open problem in this field and gives this thesis an ending.

Chapter 2

Preliminaries

In this chapter, we define some basic terminologies of polygons, graph theory, graph algorithms, algorithm theory that will be used throughout this thesis. Definitions which are not included in this chapter will be introduced as they are needed. We review, in Section 2.1, polygons and their different classes, different partitioning techniques of polygons, some standard definitions on polygons and sweeping technique of plane. In Section 2.3, we discuss about basic terminologies on graphs. In Section 2.4 we give a brief description on different subclasses of graphs which are used in later chapters. Section 2.5 describes some graph related problems those are important for the ideas and concepts used in the later parts of this thesis. Finally, we introduce different terminologies on algorithms and their complexity in Section 2.6.

2.1 Basic Terminologies of Polygons

In this section we give some definitions on polygon, its subclasses and their components.

2.1.1 Polygon

A polygon is a plane that is bounded by a finite chain of straight line segments closing in a loop to form a closed chain or circuit. These segments are called its edges or sides and the points where two edges meet are the polygon's vertices or corners. The interior of the polygon sometimes called its body. A polygon is a 2-dimensional example of the more general polytope in any number of dimensions.

There are many different types of polygons basing on the construction of their edges and bodies. Few of such polygons are defined below. A simple polygon is defined as a flat shape consisting of straight non-intersecting line segments or sides that are joined pair-wise to form a closed path. Figure 2.1 shows some
examples of simple polygons.

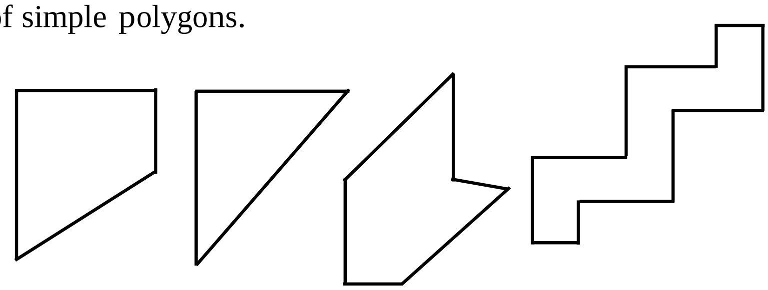


Figure 2.1: Different simple polygons.

If the sides of a polygon intersects then the polygon is not simple. A simple polygon ensures following properties:

1 Encloses a region which always has a measurable area;
2 The line segments that make up such a polygon only meet at their end-points;
3 Exactly two edges meet at each vertex;
4 The number of edges is always equals the number of vertices.

2.1.2 Orthogonal Polygon

An orthogonal polygon or rectilinear polygon is a polygon where all of whose edges meet at right angles. Thus the interior angle at each vertex is either 90° or 270°. It can be defined in another way, an orthogonal polygon is a polygon with sides parallel to the axis of cartesian coordinates. Figure 2.2 shows some examples of orthogonal polygons.

From the second definition it can be easily said that an orthogonal polygon has two types of edges i.e. horizontal edges and vertical edges. It has also two types of vertices. If the interior angle of a vertex is 90° then it is called a convex vertex. On the other hand if the angle is 270° then it is called a concave or a reflex vertex. A knob in an orthogonal polygon is an edge whose two endpoints are convex corners. An antiknob is an edge whose two endpoints are concave corners. An orthogonal polygon is simple if it does not have any hole inside.

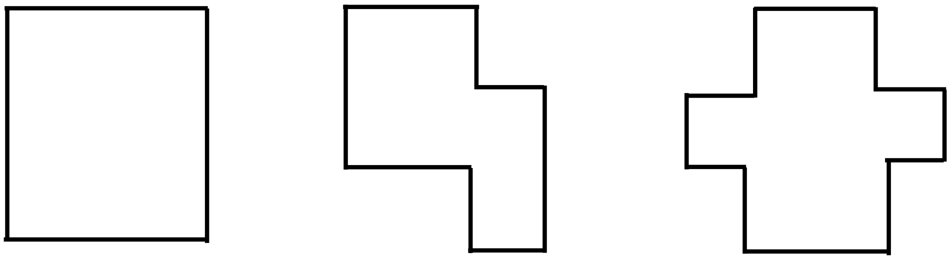


Figure 2.2: Different examples of orthogonal polygons.

2.1.3 Semi-orthogonal Polygon

A semi-orthogonal polygon is a polygon that contains two types of edges i.e. orthogonal edges and non-orthogonal edges. All the orthogonal edges (i.e. horizontal and vertical edges) meet each other at right angles. Thus interior angles of vertices those contains horizontal and vertical edges, are always 90° or 270°. Non-orthogonal edges of a semi-orthogonal polygon always meet with the orthogonal edges and maintain an interior angle less than 270°. That means two neighboring edges of a non-orthogonal edge are always orthogonal. Again interior angles of the vetices of a non-orthogonal edge are always less than 270°. The class of semi-orthogonal polygons is a superclass of orthogonal polygons. Figure 2.3 shows an example of a semi-orthogonal polygon.

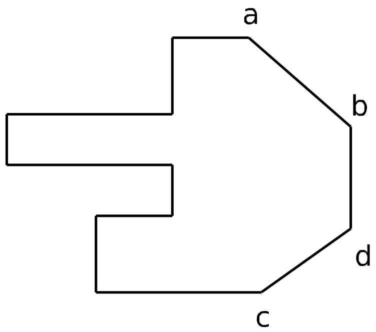


Figure 2.3: Example of a semi-orthogonal polygon.

In this polygon there are two non-orthogonal edges, i.e. ab and cd. Two neighboring edges of ab and cd are orthogonal. Interior angle at a, b, c and d are less than 270°. Beside that all the orthogonal edges meet each other at right angle.

2.1.4 Monotone and Non-Monotone Orthogonal Polygon

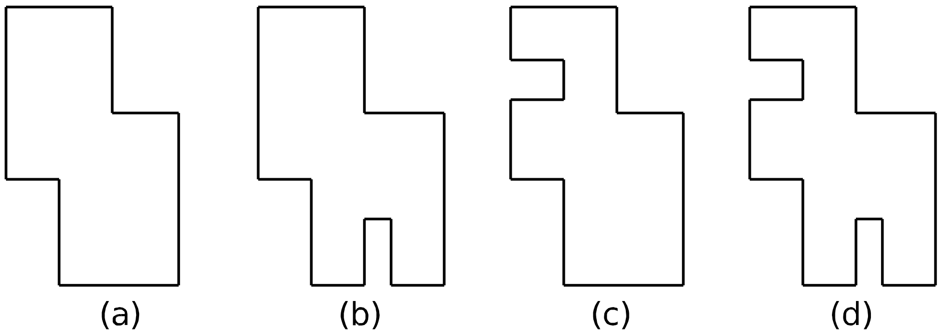


Figure 2.4: Different monotonicity in orthogonal polygons. (a) xy-monotone (b) y-non monotone/x-monotone (c) x-non monotone/y-monotone (d)xy-non monotone.

A polygon P in the plane is called monotone with respect to a straight line L, if every line orthogonal to L intersects P at most twice. Here the reference line L can be X-axis or Y-axis or both. If the polygon is both orthogonal and monotone then the polygon is called a monotone orthogonal polygon. If an orthogonal line drawn from X-axis intersects an orthogonal polygon at most twice then the orthogonal polygon is called a x-monotone polygon. If an orthogonal line drawn from X-axis intersects the polygon more than twice then the orthogonal polygon is a x non-monotone polygon. Again if the orthogonal line drawn from Y-axis intersects an orthogonal polygon at most twice then the orthogonal polygon is y-monotone . If an orthogonal line drawn from Y-axis intersects the polygon more than twice then the orthogonal polygon is y non-monotone. If orthogonal line drawn from both X and Y axis intersects the orthogonal polygon at most twice then it is termed as xy-monotone orthogonal polygon. Figure 2.4 shows different types of monotonicity in orthogonal polygon.

2.1.5 Grid n-ogon

An n vertex orthogonal polygon is called an n-ogon. An n-ogon P is in general position if and only if every horizontal and vertical line contain at most one edge of P, i.e. if and only if P has no collinear edges. If an n-ogon is in general position and if it can be defined in a $\frac{n}{2} \times \frac{n}{2}$ square grid then it is called

grid n-ogon. A rectilinear cut (r-cut) of an n-ogon P is obtained by extending each edge incident to reflex vertices of P towards interior of P until it hits the boundary of P . This partition is denoted by $\Pi(P)$ and the number of its el-
ements (pieces) by $|\Pi(P)|$. A grid n-ogon Q is called FAT grid n-ogon if and only if $|\Pi(Q)| \geq |\Pi(P)|$, for all grid n-ogons P . Similarly, a grid n-ogon Q is called THIN grid n-ogon if and only if $|\Pi(Q)| \leq |\Pi(P)|$, for all grid n-ogons P . Figure 2.5 shows different subclasses of grid n-ogons.

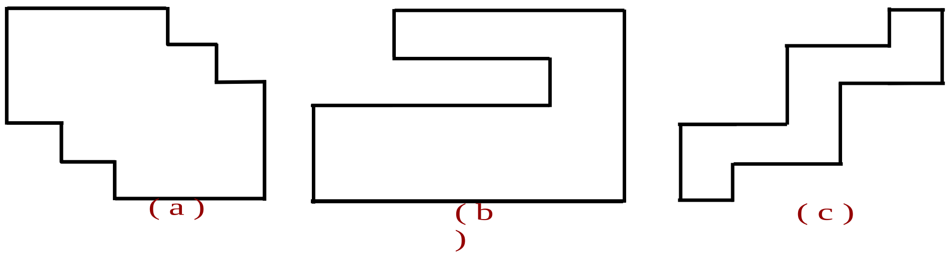


Figure 2.5: Examples of different grid n-ogons. (a) FAT grid n-ogon (b) THIN grid n-ogon (c) MIN AREA grid n-ogon.

O´ Rourke [15] gave a relation between number of convex and reflex vertices in an orthogonal polygon. The relation is $n = 2r + 4$, where n is the number of convex vertices and r is the number of reflex vertices. Bajuelos et al [1] gave
the following result for FAT grid n-ogon on number of r-cut.

$$|\Pi(P)| = \begin{cases} \frac{3r^2+6r+4}{4} & \text{for } r \text{ is even} \\ \frac{3(r+1)^2}{4} & \text{for } r \text{ is odd} \end{cases}$$

If orthogonal polygon P is THIN grid n-ogon then $|\Pi(P)| = 2r + 1$
The area of a grid n-ogon P represented by $A(P)$ and it is the number of grid cells in interior of P . Bajuelos et al.[1] also gave following relations between the maximum and the minimum number of grid cells with the number of reflex
vertex for a grid n-ogon: $2r + 1 \leq A(P) \leq r^2 + 3$ for $r \geq 2$. If for a grid n-ogon
$A(P) = r^2 + 3$ then that grid n-ogon is called MAX AREA grid n-ogon. Again if $A(P) = 2r + 1$ then that grid n-ogon is called MIN AREA grid n-ogon.

2.2 Polygon Partitioning and Plane Sweeping

In computational geometry polygons are usually triangulated to perform different computational tasks. Polygon triangulation is the decomposition of a polygonal area (simple polygon) P into a set of triangles i.e. finding a set of triangles with pairwise non-intersecting interiors whose union is P . Over time a number of algorithms have been proposed to triangulate a polygon. If the input polygon is orthogonal then sometimes the polygons are rectangulated. In rectangulation technique orthogonal polygons are decomposed of some adjacent, non-overlapping rectangles that fully cover the input orthogonal polygon. In the subsequent paragraph we describe some triangulation and rectangulation algorithm of polygons.

2.2.1 Convex Polygon Triangulation

A convex polygon is a simple polygon (not self-intersecting) where all interior angles are less than 180°. A convex polygon is trivial to triangulate in linear time, by adding diagonals from one vertex to all other vertices. The total number of ways to triangulate a convex n-gon by non-intersecting diagonals is
the $n-2$-th catalan number which equals, $\frac{n(n+1)....(2n-4)}{(n-2)!}$. In the Figure 2.6 bellow a convex polygon is triangulated.

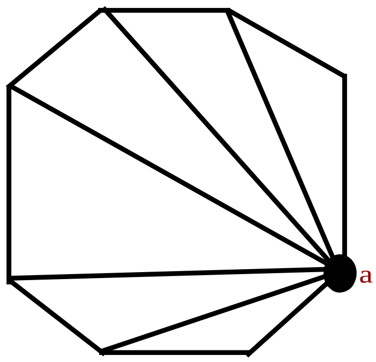


Figure 2.6: Triangulation of a convex polygon.

Here diagonals are drawn to all other vertices except the neighbor vertices from the vertex marked by a circle a. All the diagonals decomposed the orthogonal polygon in to some adjacent non-overlapping triangles.

2.2.2 Ear Clipping Method

Any simple polygon with at least 4 vertices without holes has at least two ears. An ear is a triangle with two sides being the edges of the polygon and the third one remains completely inside the polygon. Ear clipping method triangulate a polygon based on ears. The algorithm finds such an ear, remove it from the polygon (which results in a new polygon that still meets the conditions) and repeating until there is only one triangle left. Figure 2.7 describes triangulation by ear clipping method.

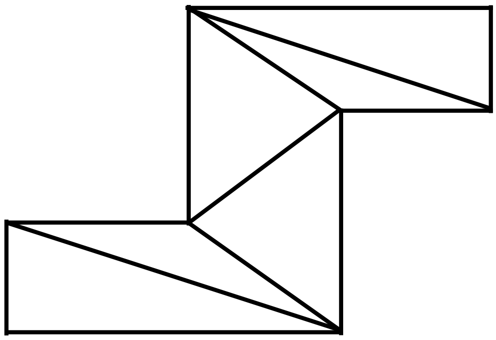


Figure 2.7: Triangulation by ear clipping method.

This algorithm is easy to implement, but slower than some other algorithms, and it only works on polygons without holes. An implementation that keeps separate lists of convex and concave vertices will run in $O(n^2)$ time. This method is known as ear clipping and sometimes ear trimming.

2.2.3 Rectangulation using Reflex vertices

Two vertices are co-grid if they do not share the same edges in the orthogonal polygon but lie on the same horizontal or vertical line. A chord is a line segment fully contained in P that connect two co-grid reflex vertices. If the orthogonal polygon is without co-grid then following algorithm can be used to rectangulate:

Algorithm Rectangulation without Chord

Input : A n vertex simple orthogonal polygon P .

Output : Polygon P with rectangulation.

1 For each reflex vertex, select one of its incident edges.(Two edges are incident to each reflex vertex.)

2 Extend this edge until it hits another such extended edge, or a bound- ary edge of P .

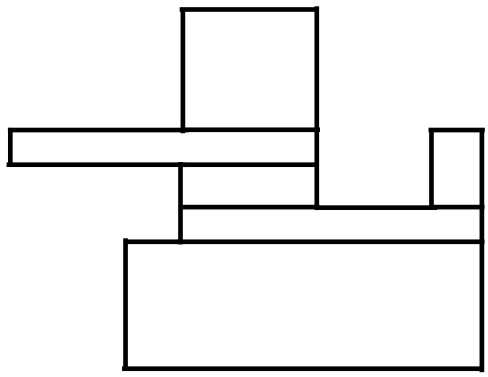


Figure 2.8: An illustration of rectangulation of a chord free orthogonal polygon.

Figure 2.8 describes the rectangulation of an orthogonal polygon without chord. In this
figure all horizontal edges of reflex vertices are extended. As all the horizontal edges of the reflex vertices are extended, all the extensions meet at the boundary of the orthogonal polygon.

If the orthogonal polygon has co-grid then we can use the following algorithm:

Algorithm Rectangulation with Chord

Input : A n vertex simple orthogonal polygon P .

Output : An orthogonal polygon P with rectangulation.

1 Find chords of P .

2 Construct a bipartite graph with edges between vertices in the sets V and H , where each vertex in V corresponds to a vertical chord, and each vertex in H corresponds to a horizontal chord. Draw an edge between vertices $v \in V$ and $h \in H$ iff the chords corresponding to v and h intersect.

3 Find the maximum matching M of bipartite graph.

4 Use M to find the maximum independent set S of vertices of the bipartite graph. (This corresponds to the maximum set of non intersecting chords of P .)

5 Draw the chords corresponding to S in P .This subdivides P into S + 1 smaller polygons, none of which contains a chord.

6 Use earlier algorithm to rectangulate each of the chord less polygons.

7 Output the union of the rectangulations of the previous step.

Figure 2.9 illustrates the above algorithm. In this algorithm all the chords

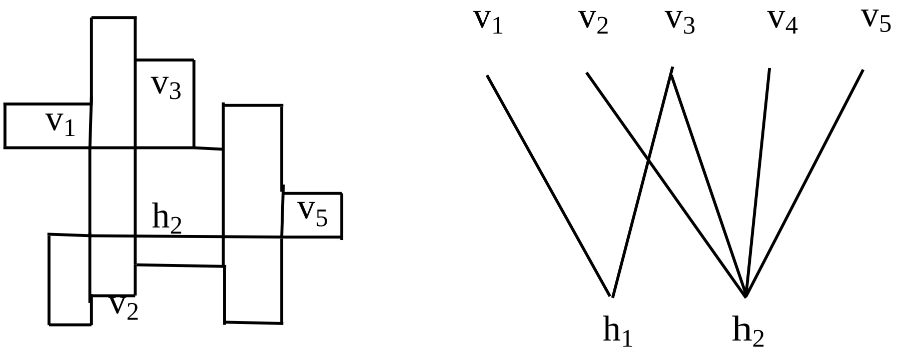


Figure 2.9: An illustration of rectangulation of an orthogonal polygon with chords

are first identified. There are two types of chords i.e. horizontal and verti- cal. In the Figure 2.9(a) h_1, h_2 are horizontal chords and v_1, v_2, v_3, v_4 , v_5 are vertical chords. This algorithm constructs a bipartite graph taking h_1, h_2 and v_1, v_2 , v_3 , v_4 , v_5 are two partite set of vertices. Two vertices are neighbor if the corresponding cords intersect each other. Figure 2.9(b) is constructed from Fig- ure 2.9(a). In the Figure 2.9(b), v_1 and h_1 are neighbor as they intersected in the polygon. All other connectivity between the vertices of two partite set are made similar way. Algorithm then finds the maximum matching M from the bipartite graph. This M is used to find the maximum independent set S of vertices of that graph. Then corresponding chords in S are drawn to P . At the end earlier algorithm is utilized to rectangulate each of the chord less polygon.

2.2.4 Sweep Line Algorithm

A sweep line algorithm uses a conceptual sweep line or sweep surface to solve various problem in euclidean space. It is one of the key technique in computational geometry. The idea behind this algorithm is to imagine a line which moves across a plane and stops at some points. Different geometric operations are executed whenever it stops and the complete solution is available when the

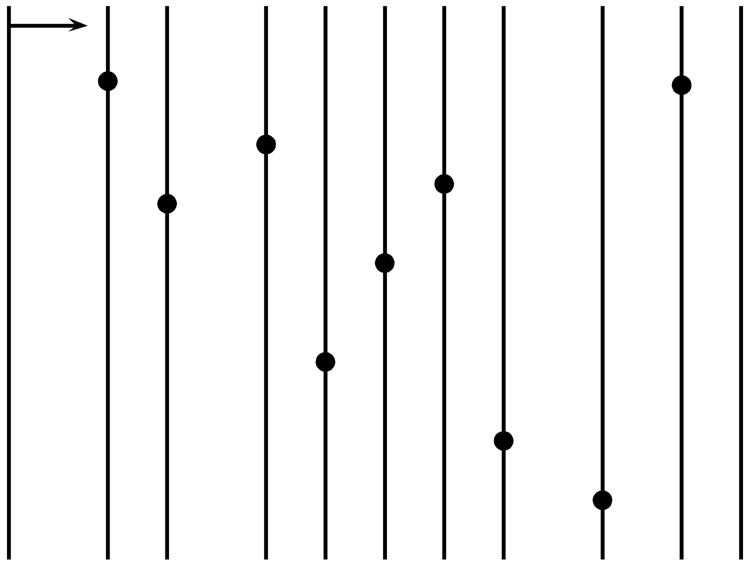


Figure 2.10: An illustration of sweep line algorithms

line pass over all the objects in the plane. Figure 2.10 describes the sweep line algorithm. In this figure the continuous vertical line moves towards right over a surface. Circles are representing objects on the surface. While moving towards right when the vertical line found any object then it stops and execute some operations. The dotted lines represent the stop of the vertical line on its move.

2.3 Basic Terminology of Graph

In this section we give some definitions of standard graph-theoretical terms used throughout this thesis.

2.3.1 Graphs

A graph G is a tuple (V, E) which consists of a finite set V of vertices and a finite set E of edges; each edge being an unordered pair of vertices.

Figure 2.11 depicts a graph $G = (V, E)$ where each vertex in $V = \{v_1, v_2, \cdots, v_6\}$ is drawn as a small circle and each edge in $E = \{e_1, e_2, \cdots, e_8\}$ is drawn by a line segment.

We denote an edge joining two vertices u and v of the graph $G = (V, E)$ by (u, v) or simply by uv. If $uv \in E$ then the two vertices u and v of the graph G are said to be adjacent; the edge uv is then said to be incident to the vertices u

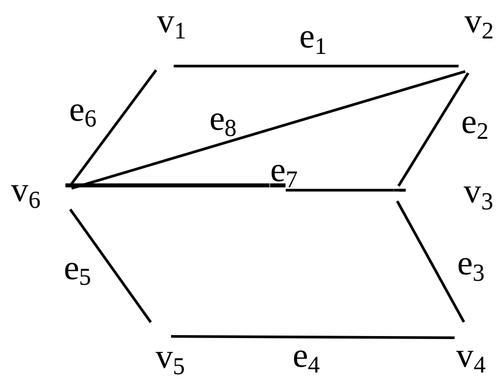


Figure 2.11: A graph with six vertices and eight edges.

and v; also the vertex u is said to be a neighbor of the vertex v (and vice versa). The degree of a vertex v in G, denoted by d(v) is the number of edges incident to v. In the graph shown in Figure 2.11 vertices v_1 and v_2 are adjacent, and $d(v_6) = 4$, since four of the edges, namely e_5, e_6, e_7 and e_8 are incident to v_6.

2.3.2 Simple Graphs and Multigraphs

If a graph G has no "multiple edges" or "loops", then G is said to be a simple graph. Multiple edges join the same pair of vertices, while a loop joins a vertex with itself. The graph in Figure 2.11 is a simple graph.

A graph in which loops and multiple edges are allowed is called a multigraph. Multi graphs can arise from various application. One example is the "call graph" that represents the telephone call history of a network. Figure 2.12 illustrates multigraphs with multiple edges and loops.

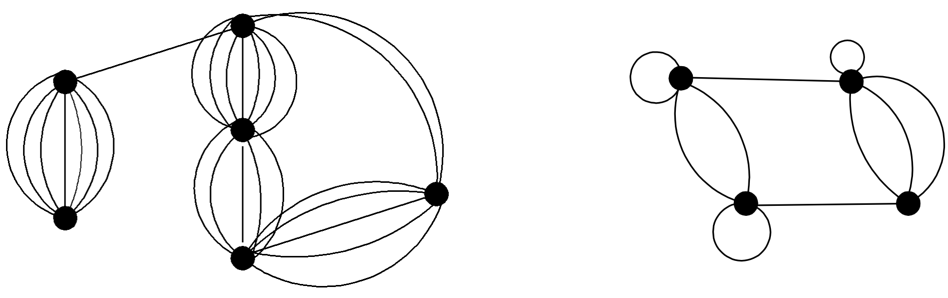


Figure 2.12: Multigraphs.

Often it is clear from the context that the graph is simple. In such cases, a simple graph is called a graph. In the remainder of thesis we will only concern about simple graphs.

2.3.3 Directed and Undirected Graphs

In a directed graph, the edges do have a direction but in an undirected graph, the edges are undirected. Mathematically, the edges in a directed graphs are
2-tuple while for undirected graphs they are 2-member subset of the vertex set. In Figure 2.13(a) and (b), we show an undirected and a directed graphs

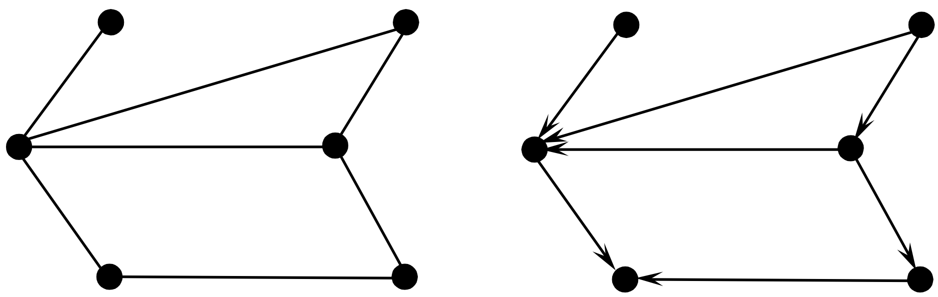


Figure 2.13: Directed and undirected graphs.

respectively.

2.3.4 Weighted and Unweighted Graphs

Each vertex or each edge in a graph can be assigned by a label which is called weight. This kind of graph are called weighted graph. Weights are usually a real number. They may be restricted to rational number or integers. This numbers are assigned basing on the characteristics of the vertices and edges in the real scenario. Figure 2.14(a) and 2.14(b) describe the node weighted and edge weighted graph respectively. Numbers associated with the vertices and edges are the weights of that vertices and edges.

Certain algorithms require further restrictions on weights; for instance, Dijkstra's algorithm works properly only for positive weights. Many extended problems are introduced basing on the weights of graph like the minimum weight vertex cover etc. The weight of a path or the weight of a tree in a weighted graph is the sum of the weights of the selected edges. Sometimes a non-edge (a vertex pair with no connecting edge) is indicated by labeling it with a spe- cial weight representing infinity. Sometimes the word cost is used instead of weight. When stated without any qualification, a graph is always assumed to

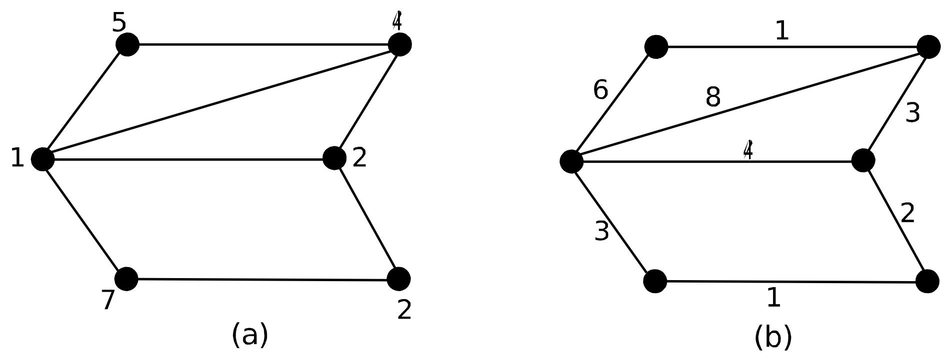


Figure 2.14: Node and edge weighted graphs.

be unweighted graph. In some writing on graph theory the term network is a synonym for a weighted graph.

2.3.5 Subgraphs

A subgraph of a graph $G = (V, E)$ is a graph $G' = (V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$. If G' contains all the edges of G that join two vertices in V', then G' is said to be the subgraph induced by V'. Figure 2.15 depicts a subgraph of G in Figure 2.11.

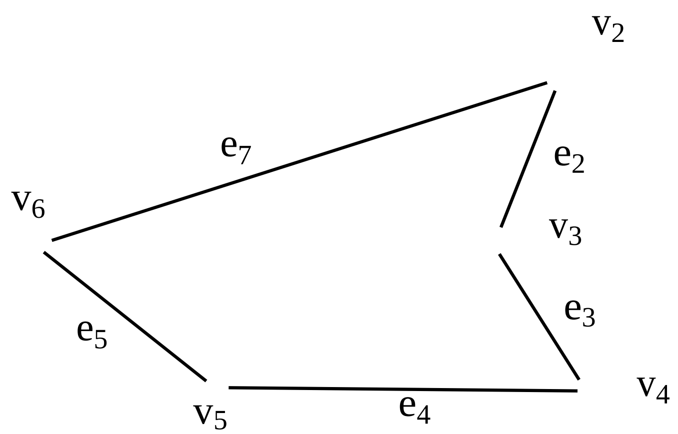


Figure 2.15: A subgraph of the graph in the Figure 2.11.

We often construct new graphs from old ones by deleting some vertices or edges. If v is a vertex of a given graph $G = (V, E)$, then $G - v$ is the subgraph of G obtained by deleting the vertex v and all the edges incident to v. More generally, if V' is a subset of V, then $G - V'$ is the subgraph of G obtained by deleting the vertices in V' and all the edges incident to them. Then $G - V'$ is a subgraph of G induced by $V - V'$. Similarly, if e is an edge of a G, then $G - e$ is the subgraph of G obtained by deleting the edge e. More generally, if $E' \subseteq E$, then $G - E'$ is the subgraph of G obtained by deleting the edges in E'.

2.4 Different Classes of Graphs

In this section we give some definitions of special classes of graphs related to planar graphs and non planar graphs used in this thesis. For readers interested in planar graphs

2.4.1 Planar Graphs

A planar graph is a graph that can be embedded in the plane, i.e., it can be drawn on the plane in such a way that its edges intersect only at their endpoints. In other words, it can be drawn in such a way that no edges cross each other. Such a drawing is called a plane graph or planar embedding of the graph. Figure 2.16(a),(b) and (c) show three different examples of planer and non planar graph

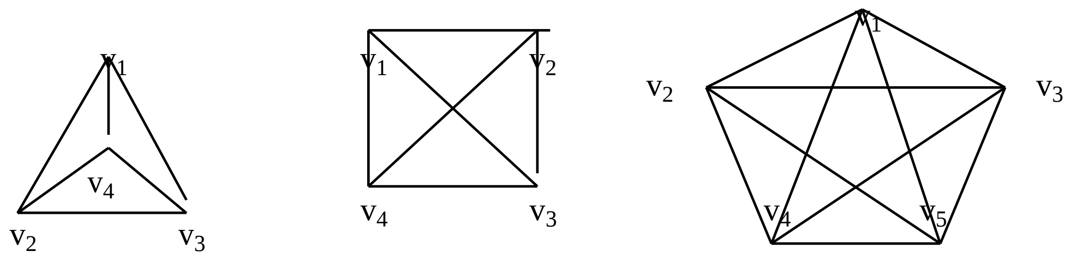


Figure 2.16: Example of planner and non planner graphs.

A plane graph can be defined as a planar graph with a mapping from every node to a point on a plane, and from every edge to a plane curve on that plane, such that the extreme points of each curve are the points mapped from its end nodes, and all curves are disjoint except on their extreme points.

2.4.2 Eular's Theorem on Planar Graphs

Euler's formula states that if a finite, connected, planar graph is drawn in the plane without any edge intersections, and v is the number of vertices, e is

the number of edges and f is the number of faces (regions bounded by edges, including the outer, infinitely large region), then the relations among them is
$v - e + f = 2$. Figure 2.17 illustrates the Eular's formula. Here the graph is a

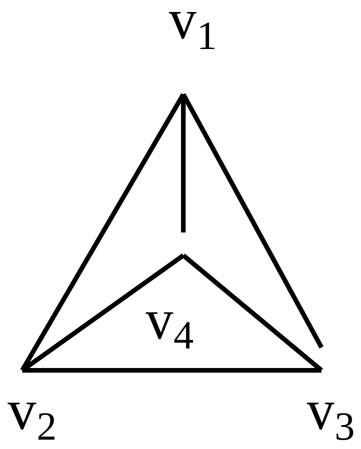


Figure 2.17: An illustration of Eular's formual.

planer graph. In this planer graph total number of vertices $v = 4$, total number of edges $e = 6$ and total number of faces (including outer face) $f = 4$. After executing the formula we found the result 2.

2.4.3 Bipartite Graphs

A bipartite graph is a graph whose vertices can be divided into two disjoint sets U and V (i.e. U and V are each independent sets) such that every edge connects a vertex in U to one in V. Vertex set U and V are often denoted as partite sets. In bipartite graph there will be no edge connecting the two vertices within one partite set. Figure 2.18 bellow is an example of bipartite graph

In this graph vetices are divided in to two groups i.e. U and V. Each vertex from U is connected to the vertices of V. No two vertices from the same group are neighbor.

2.4.4 Convex Bipartite Graphs

A convex bipartite graph is a bipartite graph with specific properties. A bipartite graph $(U \cup V, E)$ is said to be convex over the vertex set U, if U can be enumerated such that for all $v \in V$ the vertices adjacent to v are consecutive.

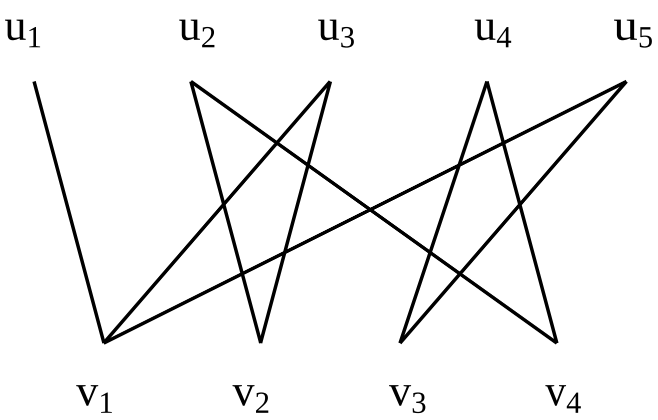


Figure 2.18: Example of a bipartite graph.

Convexity over V can be defined analogously. A bipartite graph $(U \cup V, E)$ that is convex over both U and V is said to be biconvex or doubly convex bipartite graph. Figure 2.19 illustrate the structure of convex bipartite graph

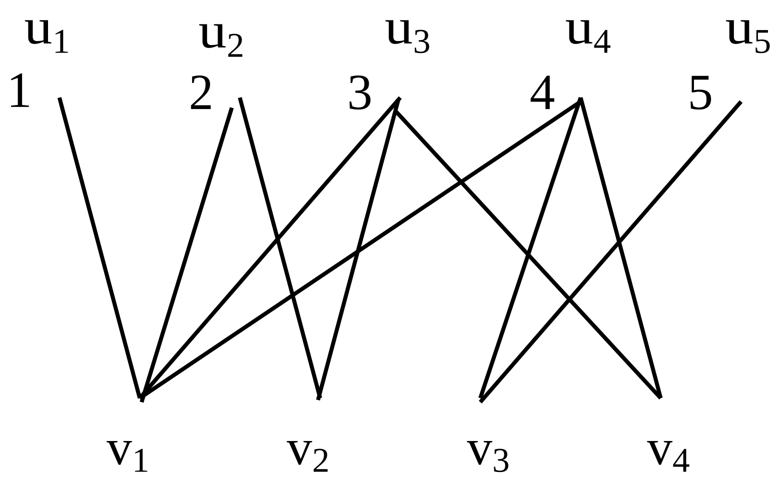


Figure 2.19: Example of a convex bipartite graph.

In this figure vertices of partite set U maintains a sequence, represented by a sequence number(1,2,3,4) Neighbors of any vertex from V are always consecutive. Figure 2.20 depicts one non convex bipartite graph.

Here in the figure vertex v1 has a neighbor set i.e. u1, u2, u4 which are not consecutive.

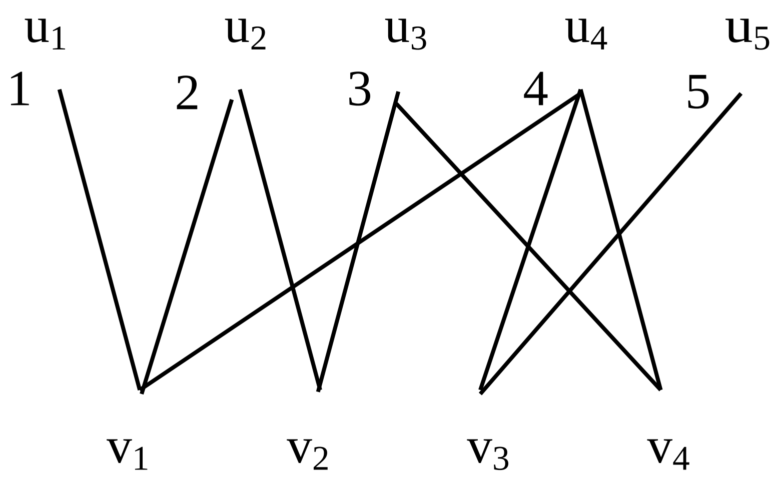


Figure 2.20: Example of a non-convex bipartite graph.

2.5 Different Graph Problems

In this section we will describe some graph problem and some of their existing solution which are relevant to our work.

2.5.1 Vertex Cover

A vertex cover of a graph is a set of vertices such that each edge of the graph is incident to at least one vertex of the set. Formally a vertex cover $V^{'}$ of an undirected graph $G = (V, E)$ is a subset of V such that $\forall_{uv} \in E : u \in V^{'} \vee v \in V^{'}$. It is a set of vertices $V^{'}$ where every edges at least one endpoint in the vertex cover $V^{'}$. Such a set is said to cover the edges of G. The above figure shows two examples of vertex cover. A minimum vertex cover is a vertex cover of the smallest possible size. The vertex cover number τ is the size of the minimum vertex cover, i.e. $\tau = |V^{'}|$. Figure 2.21(a) and 2.21 (b) shows examples of vertex cover and the minimum vertex cover of the same graph. In this figure white circles are selected in the vertex cover. The problem of finding the minimum vertex cover is a typical example of an NP-hard optimization problem that has an approximation algorithm.The vertex cover problem, was one of Karp's 21 NP-complete problems and is therefore a classical NP-complete problem as well. But the minimum vertex cover problem for bipartite graph has a polynomial time solution using Konig's theorem. Konig's theorem states that if $G = (V, E)$ is a bipartite graph then size of the minimum vertex cover is

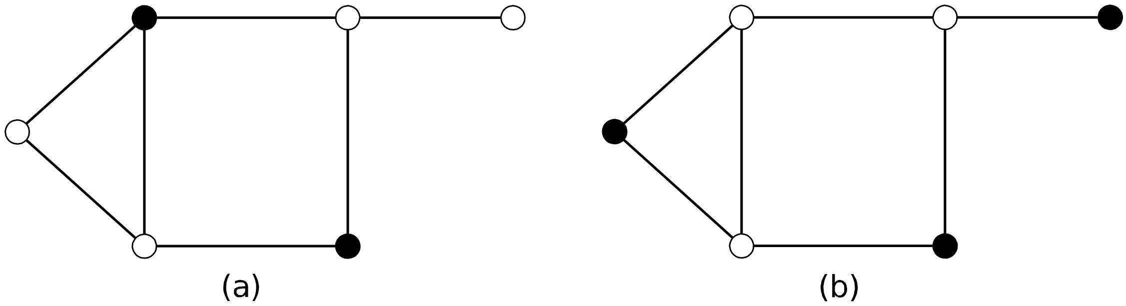


Figure 2.21: (a) A vertex Cover (b) the minimum Vertex Cover.

equal to the size of the maximum matching. The theorem not only established the similarity between the size of the minimum vertex cover with the size of the maximum matching but proof of this theorem gives an efficient algorithm for finding the minimum vertex cover of a bipartite graph from the maximum matching. Finding the maximum matching in a bipartite graph G, construct a forest F of alternating trees and then the minimum vertex cover C .

In the weighted vertex cover problem each node $i \in V$ has an associated weight $w_i \geq 0$ and we want to minimize the total weight of the set S. The objective is to minimize the total weight of vertices those have all the edges connected but not to minimize the number of vertics. The minimum vertex cover problem for bipartite graph is solvable in polynomial time implies that the minimum weight vertex cover problem for bipartite graph is also solvable in polynomial time.

2.5.2 Matching

A matching or independent edge set in a graph is a set of edges without common vertices. It may also be an entire graph consisting of edges without common vertices. Given a graph G = (V, E), a matching M in G is a set of pairwise non-adjacent edges; that is, no two edges share a common vertex. A vertex is matched or saturated if it is an endpoint of one of the edges in the matching. Otherwise the vertex is unmatched .

A maximal matching is a matching M of a graph G with the property that if any edge not in M is added to M , it is no longer a matching, that is, M is maximal if it is not a proper subset of any other matching in graph G. In other words, a matching M of a graph G is maximal if every edge in G has a non-empty intersection with at least one edge in M . A maximum matching (also known as maximum-cardinality matching) is a matching that contains the largest possible number of edges. There may be many maximum matchings. The matching number ν (G) of a graph G is the size of a maximum matching. Every maximum matching is maximal, but not every maximal matching is a maximum matching. A perfect matching is a matching which matches all vertices of the graph i.e. every vertex of the graph is incident to exactly one edge of the matching. Every perfect matching is maximum and hence maximal. Sometimes it is termed as complete matching. A perfect matching is also a minimum-size edge cover.

Thus, $\nu(G) \leq \rho(G)$ i.e. the size of a maximum matching is no larger than the size of a minimum edge cover. Figure 2.22(a) describes maximal matching and Figure 2.22(b) describes maximum matching.

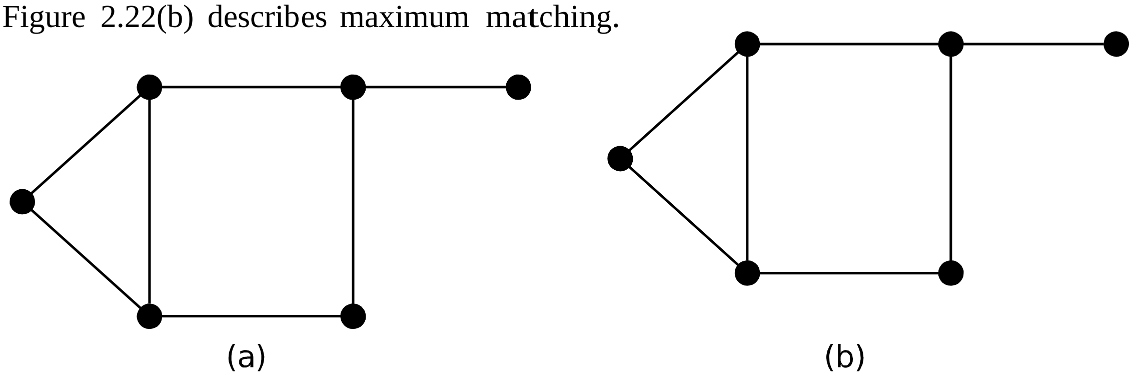


Figure 2.22: (a) A maximal matching and (b) A maximum and perfect matching.

In the figure broken lines are the lines selected for matching. A near-perfect matching is one in which exactly one vertex is unmatched. This can only occur when the graph has an odd number of vertices, and such a matching must be maximum. If, for every vertex in a graph, there is a near-perfect matching that omits only that vertex, the graph is also called factor-critical.

2.6 Complexity of Algorithms

In this section we briefly introduce some terminologies related to complexity of algorithms. For interested readers, we refer the book of Garey and Johnson.

The most widely accepted complexity measure for an algorithm is the running time, which is expressed by the number of operations it performs before producing the final answer. The number of operations required by an algorithm is not the same for all problem instances. Thus, we consider all inputs of a given size together, and we define the complexity of the algorithm for that input size to be the worst case behavior of the algorithm on any of these inputs. Then the running time is a function of size n of the input.

2.6.1 The Notation O(n)

In analyzing the complexity of an algorithm, we are often interested only in the "asymptotic behavior", that is, the behavior of the algorithm when applied to very large inputs. To deal with such a property of functions we shall use the following notations for asymptotic running time. Let $f(n)$ and $g(n)$ are the functions from the positive integers to the positive reals, then we write $f(n) = O(g(n))$ if there exists positive constants c_1 and c_2 such that $f(n) \leq c_1 g(n) + c_2$ for all n. Thus the running time of an algorithm may be bounded from above by phrasing like "takes time $O(n^2)$".

2.6.2 Polynomial Algorithms

An algorithm is said to be polynomially bounded (or simply polynomial) if its complexity is bounded by a polynomial of the size of a problem instance. Examples of such complexities are $O(n)$, $O(n\log n)$, $O(n^{100})$, etc. The remaining algorithms are usually referred as exponential or nonpolynomial. Examples of such complexity are $O(2^n)$, $O(n!)$, etc. When the running time of an algorithm is bounded by $O(n)$, we call it a linear-time algorithm or simply a linear algorithm.

2.6.3 NP-complete Problems

There are a number of interesting computational problems for which it has not been proved whether there is a polynomial time algorithm or not. Most of them are "NP-complete", which we will briefly explain in this section.

The state of algorithms consists of the current values of all the variables and the location of the current instruction to be executed. A deterministic algorithm is one for which each state, upon execution of the instruction, uniquely determines at most one of the following state (next state). All computers, which exist now, run deterministically. A problem Q is in the class P if there exists a deterministic polynomial-time algorithm which solves Q. In contrast, a nondeterministic algorithm is one for which a state may determine many next states simultaneously. We may regard a nondeterministic algorithm as having the capability of branching off into many copies of itself, one for the each next state. Thus, while a deterministic algorithm must explore a set of alternatives one at a time, a nondeterministic algorithm examines all alternatives at the same time. A problem Q is in the class NP if there exists a nondeterministic polynomial-time algorithm which solves Q. Clearly, $P \subseteq NP$.

Among the problems in NP are those that are hardest in the sense that if one can be solved in polynomial-time then so can every problem in NP. These are called NP-complete problems. The class of NP-complete problems has the following interesting properties.

(a) No NP-complete problem can be solved by any known polynomial algorithm.

(b) If there is a polynomial algorithm for any NP-complete problem, then there are polynomial algorithms for all NP-complete problems.

Sometimes we may be able to show that, if problem Q is solvable in polynomial time, all problems in NP are so, but we are unable to argue that $Q \in NP$. So Q does not qualify to be called NP-complete. Yet, undoubtedly Q is as hard as any problem in NP. Such a problem Q is called NP-hard.

Chapter 3

Algorithms for MLSC Problem

3.1 Introduction

Let P be a simple orthogonal polygon. Consider sliding cameras travels back and forth along the orthogonal line segment of P and it can see inside the polygon orthogonally. Minimum length sliding camera (MLSC) problem asks to cover the whole polygon by sliding cameras so that sliding length becomes minimum. Durocher and Mehrabi [8] gave an $O(n^2)$ time algorithm to solve the MLSC problem. Throughout this chapter we call this algorithm as Algorithm DM. We show that some steps of Algorithm DM takes $O(n \log^2 n)$ time for monotone orthogonal polygons and $O(n)$ time for some subclasses of grid n-ogons. The rest of the chapter is organized as follows. Section 3.2 contains the outline of the Algorithm DM and few known properties of some sub classes of grid n-ogon. Section 3.3 describes how some steps of the Algorithm DM takes $O(n \log^2 n)$ time for monotone orthogonal polygons. Section 3.4 shows how same steps of the Algorithm DM takes $O(n)$ time for some subclasses of grid n-ogon. Finally, we summarize our contribution in Section 3.5.

3.2 Preliminaries

In section 3.2.1 we present the outline of the Algorithm DM which solves the MLSC problem . Related existing results on grid n-ogon are described Section 3.2.2

3.2.1 Algorithm DM On MLSC Problem

Algorithm DM first partitions the interior of the orthogonal polygon into some rectangles, then it constructs a bipartite graph from that partition. At the end it utilizes the the minimum weight vertex cover algorithm to solve the MLSC problem. The detail description of each step is given below.

Step 1. Partitioning: The partition technique used in Algorithm DM was first used by Katz and Morgenstern [12]. In this technique two incident edges of all the reflex vertices of P are extended inward until they hit the next boundary. The extended edges partitions the interior and boundary of P into a set of rectangles and line segments respectively. Each such rectangles is called r-cut. Let TP, LP and $R(P)$ be the set of all top facing segments, left facing segments and r-cuts of P respectively. All the r-cuts covered by a single left facing and right facing segment is called row of r-cut. Similarly all the r-cuts covered by a single top facing and bottom facing segment is called column of r-cut. Figure 3.1 is the input orthogonal polygon. Figure 3.2 shows the partitioning of Figure 3.1.

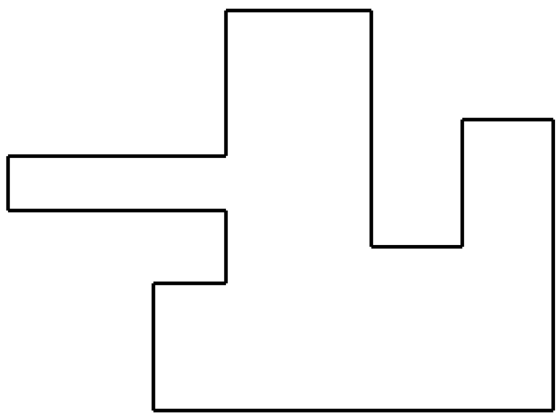


Figure 3.1: An input orthogonal polygon P for the MLSC problem.

In the Figure 3.2, dotted lines are the extension of edges incident to reflex vertices and r_1 is a single r-cut created by some of those extensions. Again $tp_n \in TP$ and $lp_n \in LP$ indicates single top and left facing segments respectively. Number associated with each tp_n and lp_n is the length of those segments. Durocher and Mehrabi showed that using sliding camera every orthogonal polygon P can be optimally covered by the segments of TP and LP. Partitioning of input orthogonal polygons requires $O(n)$ time complexity.

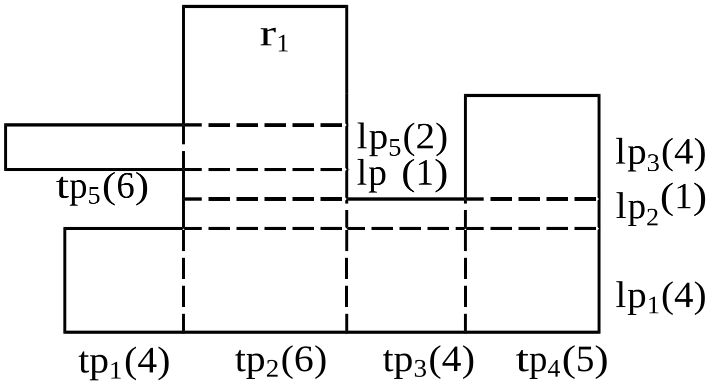


Figure 3.2: Rectangulation of the orthogonal polygon P in the Figure 3.1.

Step 2. Reducing to a Bipartite Graph: After partitioning Algorithm DM constructs an undirected weighted graph G_P where each segment in P corresponds to a vertex of G_P . Let T and L are the two sets of vertices in G_P corresponds to T P and LP of P respectively. Length of each segment is the weight of the corresponding vertex. Two vertices are adjacent in G_P if and only if both the corresponding segment in P can see same single r-cut. The graph in Figure 3.3 bellow is constructed from Figure 3.2.

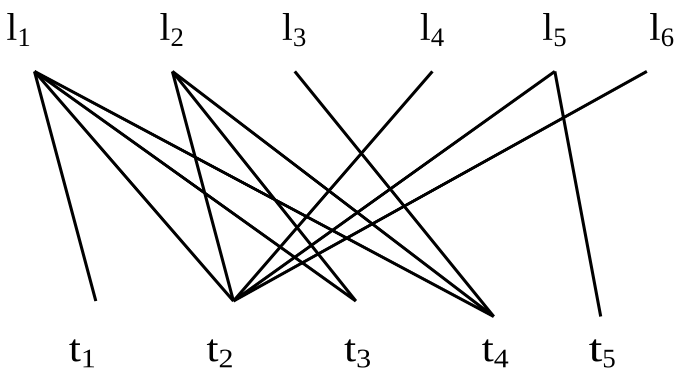


Figure 3.3: The bipartite graph G_P corresponding to the rectangulated polygon in the Figure 3.2.

In Figure 3.3 $t_n \in T$ and $l_n \in L$ indicates two single vertex of G_P which corresponds to a top facing segment $tp_n \in TP$ and $lp_n \in LP$ in P respectively. Two vertices t_n and l_n are adjacent in G_P if and only if corresponding tp_n and lp_n in P can see same single r-cut $R_n \in R(P)$. Number associated with each vertex in Figure 3.3 is the weight of that vertex which is same as the length of the corresponding segments in P. Durocher and Mehrabi showed that G_P is a bipartite graph and there is a bijection between the r-cut in P and the edges in G_P. After partitioning P number of segments increases linearly depending on the number of vertices. Therefore Step 3 requires $O(n^2)$ time complexity

Step 3. Executing Minimum Weight Vertex Cover Algorithm:

Durocher and Mehrabi show that MLSC problem on P is equivalent to the minimum weight vertex cover problem on G_P. Minimum weight vertex cover algorithm is
$O(n^2)$ time complex. Therefore overall time complexity of Algorithm DM is $O(n^2)$.

3.2.2 Existing Results on Grid n-ogon

A grid n-ogon is a n vertex orthogonal polygon which is in general position. An orthogonal polygon is in general position if it can be defined in a $\frac{n}{2} \times \frac{n}{2}$ square grid. A grid n-ogon Q is called FAT grid n-ogon if and only if $|\Pi(Q)| \geq |\Pi(P)|$, for all grid n-ogons P. Similarly, a grid n-ogon Q is called THIN grid n-ogon if and only if $|\Pi(Q)| \leq |\Pi(P)|$, for all grid n-ogons P. Here $\Pi(P)$ is the number of r-cut inside the orthogonal polygon P. The Area $A(P)$ of a grid n-ogon P is the number of grid cells inside P. If for a grid n-ogon $A(P) = 2r + 1$, then it is called MIN AREA grid n-ogon. In this section we state some results of Bajuelos et al. [1] on FAT and MIN AREA grid n-ogon.

Bajuelos et al. showed that there is single FAT grid n-ogon (except for symmetries of the grid) and its form is in Figure 3.4. They also show that each r-cut of a FAT grid n-ogon has area 1.

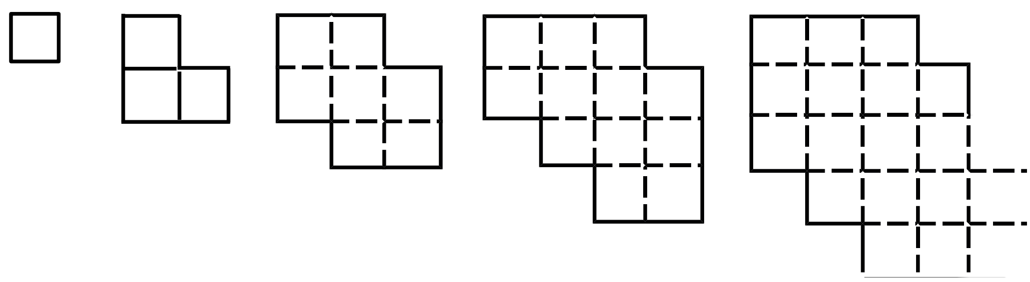


Figure 3.4: The unique FAT grid n-ogons (symmetries excluded), for $n = 4, 6, 8, 10, 12$.

For MIN AREA grid n-ogon Bajuelos et al. give similar result like FAT grid n-ogon. They found that there is single MIN-AREA grid n-ogon (except for

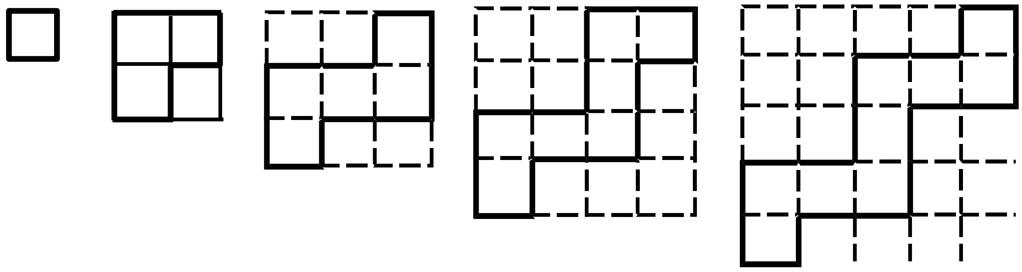


Figure 3.5: The unique MIN AREA grid n-ogons (symmetries excluded) for $r = 0, 1, 2, 3, 4$.

symmetries of the grid) and its form is like in Figure 3.5. They also state that all the MIN AREA grid n-ogon are THIN grid n-ogon but all the THIN grid n-ogon are not MIN AREA grid n-ogon.

3.3 Algorithm DM on Monotone Orthogonal Polygons

In this section we show that if the input is a monotone orthogonal polygon then Algorithm DM always construct a weighted convex bipartite graph. Minimum weight vertex cover problem for weighted convex bipartite graph has $O(n\log^2 n)$ time solution [16].

From the definition of monotone orthogonal polygons we can write the following observation:

Observation 3.3.1 Inside an orthogonal polygon P if there exist at least one pair of points which have same Y coordinates but require two separate $lp \in LP$ to cover the points then the orthogonal polygon is Y non monotone. Similarly, if there exist at least one pair of points which have same X coordinates but require two separate $tp \in TP$ to cover those points then the orthogonal polygon is X non monotone.

We assign one label from a set of consecutive labels to each $tp \in TP$ of an orthogonal polygon P. First label assigned to the leftmost and bottommost $tp \in TP$ of the polygon. The progress of assignment follow a path from left to right. At the end of one row of segment we move from bottom to top and again assignment progresses from left to right along the polygon boundary until it

reaches the topmost and right most $tp \in TP$. We do similar kind of assignment to each $lp \in LP$ on the same orthogonal polygon P where another set of consecutive labels used. This time the assignment starts from the rightmost and bottommost $lp \in LP$, and progress of assignment follows a path from bottom to top. At the end of one column of segment we move from left to right and again assignment progresses from bottom to top along the polygon boundary
until it reaches topmost and leftmost $lp \in LP$. A set of segments is said to be consecutive if they have consecutive labels otherwise the segment set is non consecutive. Vertices of graph G_P also receive same label to its corresponding segment of the polygon P. Let$[a_1, a_2, a_3, a_4, a_5]$ and $[b_1, b_2, b_3, b_4, b_5, b_6]$ be two sets of consecutive labels. Figure 3.6 below describes the assignment of these consecutive sets of labels to top and left facing segments of the orthogonal poly- gon P
First top facing segment tp_1 gets first label a_1, similarly second top facing seg-

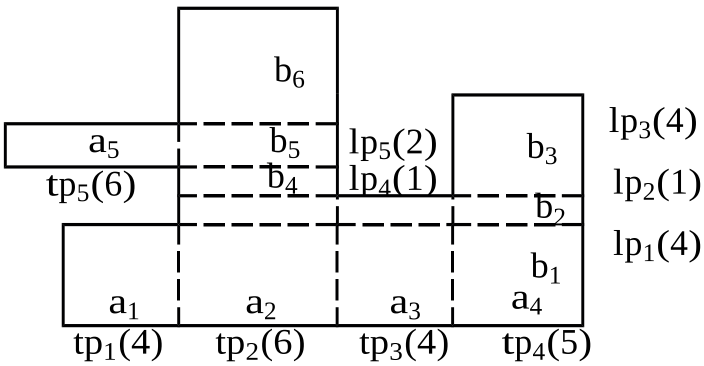


Figure 3.6: An illustration of assigning two sets of consecutive labels to top and left facing segments of an orthogonal polygon P.

ment tp_2 gets second label a_2. In this way topmost rightmost top facing segment tp_5 gets last label a_5. Second consecutive sets of labels are assigned to the left facing segments. Following the same assignment rules first left facing segment lp_1 gets b_1 and finally leftmost topmost segment lp_6 gets last label b_6. Figure 3.7 bellow illustrates the assignment of labels to the vertices of bipartite graph G_P from the polygon P. Label of tp_1 is assigned to the corresponding vertex t_1 of graph G_P. Thus vertex t_1 gets the label a_1. All other vertices receive labels in the same way.
We can write the following observation on guarding a pair of points by consecutive and missing segments of an orthogonal polygon.

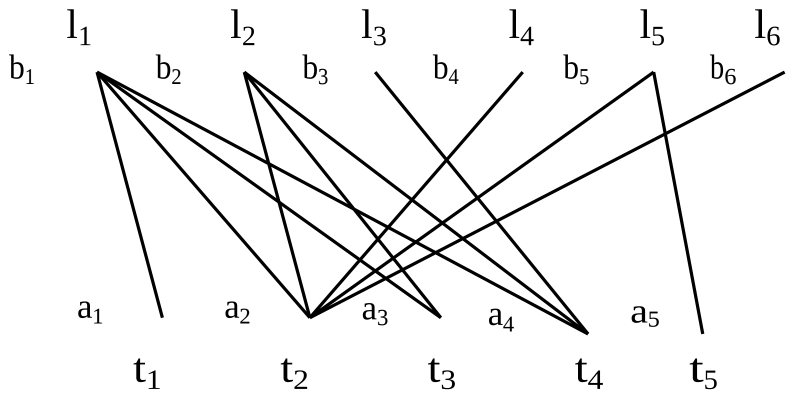


Figure 3.7: An illustration of receiving same labels in vertices of the graph G_P from the segments of the orthogonal polygon P in the Figure 3.6.

Observation 3.3.2 Let $LP_1 \subseteq LP$ be a set of non consecutive left facing segments with respect to a single $tp_1 \in TP$ that covers same set of r-cut of an orthogonal polygon P . Then at least one segment from the set of segments those are missing from the sequence of LP_1 and one segment from LP_1 always guard one pair of points separately whose Y coordinate are same.

We can write following lemma relating the consecutiveness of LP with the monotonicity of the input orthogonal polygon.

Lemma 3.3.3 Let P be a n vertices orthogonal polygon. P becomes Y non monotone if and only if there exists at least one non consecutive set of left facing segments LP_1 with respect to a single top facing segment tp such that both LP_1 and tp cover same set of r-cut.

Proof. Necessity: Assume that the input orthogonal polygon P is Y non monotone. Then there exists one pair of points inside P that have same Y coordinate but require two separate $lp \in LP$ to cover those points by Observation 3.3.1. Let (x1, y) and (x2, y) be those points. One point (x1, y) must remain inside a single r-cut which is guarded by a single top facing segment. Let $tp_1 \in TP$ be that top facing segment. Beside that r-cut, tp_1 also guards some other r-cuts. Let $R_1(P) \subseteq R(P)$ be the set of r-cuts guarded by tp_1. $R_1(P)$ also guarded by as set of left facing segments. Let $LP_1 \subseteq LP$ be the set of left facing segments. Similarly other point (x2, y) will also guarded by a set of left facing segments. Let $LP_2 \subseteq LP$ be that set of segment for point (x2, y). As (x1, y) and (x2, y) have same Y coordinate but guarded by two different

$lp \in LP$ therefore, either LP_1 or LP_2 or both are non consecutive. At least one point of such a pair of point is always creating such set of non consecutive segments in LP.

Sufficiency: Assume that an orthogonal polygon P has a non consecutive set of left facing segments LP_1, with respect to a single top facing segment tp_1, such that both LP_1 and tp_1 covers same set of r-cuts. Since LP_1 is non consecutive, it misses a set of segments. Let $LP_2 \subseteq LP$ be the missing set of segments. Therefore by Observation 3.3.2 LP_1 and LP_2 guard one pair of points separately whose Y coordinate are same. It implies that inside P we are getting at least one pair of points which have same Y coordinate but guarded by two separate $lp \in LP$ (one from LP_1 and other from LP_2). By Observation 3.3.1 the polygon is Y non monotone. Q.E.D.

We Now can write the following lemma relating the monotonicity of an orthogonal polygon with the convexity of bipartite graph.

Lemma 3.3.4 Given a monotone orthogonal polygon P with n vertices, graph G_P constructed from P will always be a convex bipartite graph.

Proof. For an X monotone orthogonal polygon P we construct a graph G_P from P using the algorithm described in Section 3.2.1. Let graph G_P is a non convex bipartite graph. Then by the definition of a non convex bipartite graph, there are two vertices (one from each partite set) in G_P whose neighbors are not consecutive. Let $l \in L$ be one such vertex which has a neighbor set $T_1 \subseteq T$, that are not consecutive. Again let $t \in T$ be the other vertex which has a neighbors set $L_1 \subseteq L$, that are not also consecutive. For such set T_1 and L_1 there are two sets of segments in the orthogonal polygon P. Let the two sets of segments be $TP_1 \subseteq TP$ and $LP_1 \subseteq LP$ respectively. As the vertex set are not consecutive therefore the segment sets are also not consecutive . But by Lemma 3.3.3 if there are two such non consecutive sets of segments (one from LFB and one from TFB) remains inside an orthogonal polygon then the polygon is XY non monotone which contradicts our initial assumption. We can get similarly contradiction for Y and XY monotone orthogonal polygons also. Now it is evident that for a given a monotone orthogonal polygon P, graph G_P constructed from P is will always be a convex bipartite graph. Q.E.D.

Construction of graph G_P from orthogonal polygon P can be completed in

$O(n^2)$ time. From Lemma 3.3.4 we can say that for a monotone orthogonal polygon P, constructed graph G_P from P will be a convex bipartite graph. Minimum weight vertex cover problem for convex bipartite graph can be solvable in $n\log^2 n$ time complexity [16]. Where n is the number of vertices. If the input orthogonal polygon is monotone then we can utilize the algorithm of the the minimum weight vertex cover for convex bipartite graph instead of only bipartite graph. It obviously reduce the execution time of solving MLSC problem. We now can write the following theorem on the efficiency of Algorithm DM.

Theorem 3.3.5 For monotone orthogonal polygons Step 3 of Algorithm DM takes $O(n\log^2 n)$ time.

3.4 MLSC Problem on Subclasses of Grid n-ogon

In this section we show that Step 3 of Algorithm DM takes $O(n)$ time for FAT and MIN AREA grid n-ogon if they are drawn on an equidistant grid. The bipartite graph constructed from a FAT or MIN AREA grid n-ogon drawn on an equidistant grid becomes an unweighted convex bipartite graph. That means an MLSC problem can be reducible to a vertex cover problem for convex bipartite graphs. It is known that the vertex cover problem can be solved in $O(n)$ time on convex bipartite graphs [19]. In the subsequent paragraph we show that FAT and MIN AREA grid n-ogons are monotone orthogonal polygons. Moreover if a FAT or a MIN AREA grid n-ogon is drawn on an equidistant grid, the lengths of the segments become equal.

We first show that each r-cut of a MIN AREA grid n-ogon has area 1 as in Lemma 3.4.1.

Lemma 3.4.1 If P is a MIN AREA grid n-ogon then each r-cut $\pi(P)$ has area 1.

Proof. Since all MIN AREA grid n-ogons are THIN grid n-ogons, then MIN AREA grid n-ogons have $|\pi(P)| = 2r + 1$ r-cuts. Again from the definition, MIN AREA grid n-ogon has area $A(P) = 2r + 1$. Therefore, for MIN AREA grid n-ogon $A(P) = |\pi(P)|$, which implies that each r-cut has area 1. Q.E.D.

Observation 3.4.2 In an equidistant grid height of all the cells are equal. Similarly width of all the cells are also equal. Furthermore height of one cell is equal
to the width of that cell.

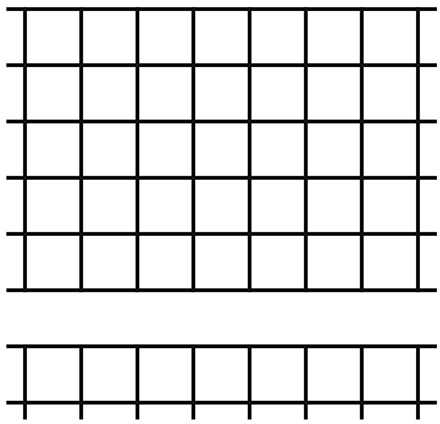


Figure 3.8: An example of 8x8 equidistant grid.

Figure 3.8 shows an example of 8 x 8 equidistant grid. Here all the horizontal and vertical lines are equidistant from each other. Therefore all the grid cells has equal height and length.

Lemma 3.4.3 If a FAT grid n-ogon P is drawn in an equidistant grid, all the segments of P get equal length.

Proof. Bajuelos et al. showed that each r-cut of a FAT grid n-ogon is equal to one grid cell. Therefore, if we draw a FAT grid n-ogon on an equidistant grid, by Observation 3.4.2 height of all the r-cuts become equal. Similarly width of all the r-cuts also become equal. Furthermore height of one r-cut is equal to the width of that r-cut.
For an orthogonal polygon length of a left facing segment depends on the height of the row of r-cut covered by that segment. A row of r-cut is the set of r-cuts those are covered by same left facing segment. Since all the r-cuts of a FAT grid n-ogon, drawn on an equidistant grid, have equal height then all the left facing segments also have equal length. Similarly length of a top facing segment of a FAT grid n-ogon depends on the width of the column of r-cut covered by that segment. A column of r-cut is the set of r-cuts those are

covered by same top facing segment. Since all the r-cuts of a FAT grid n-ogon, drawn on an equidistant grid, have equal width then all the top facing segments also have equal length. Furthermore for a FAT grid n-ogon drawn on an equidistant grid height of a r-cut is equal to the width of that r-cut. Therefore length of all the top and left facing segments are also equal.

It implies that if FAT grid n-ogons are drawn in an equidistant grid, all its segments get equal length. Q.E.D.

Lemma 3.4.4 If a MIN AREA grid n-ogon P is drawn in an equidistant grid, all the segments of P get equal length.

Proof.

By Lemma 3.4.1 each r-cut of a MIN AREA grid n-ogon is equivalent to one grid cell. Therefore if we draw a MIN AREA grid n-ogon in an equidistant grid, by Observation 3.4.2 height of all the r-cuts become equal. Similarly width of all the r-cuts are also become equal. Furthermore height one r-cut is equivalent to the width of that r-cut.

For an orthogonal polygon length of a left facing segment depends on the height of the row of r-cut covered by that segment. Since all the r-cuts of a MIN AREA grid n-ogon, drawn on an equidistant grid, have equal height then all the left facing segments also have equal length. Similarly length of a top facing segment of a MIN AREA grid n-ogon depends on the width of the column of r-cut covered by that segment. Since all the r-cuts of a MIN AREA grid n-ogon, drawn on an equidistant grid, have equal width then all the top facing segments also have equal length. Furthermore for a MIN AREA grid n-ogon drawn on an equidistant grid height of a r-cut is equal to the width of that r-cut. Therefore length of all the top and left facing segments are equal.

It implies that if FAT grid n-ogons are drawn in an equidistant grid, all its segments gets equal length. Q.E.D.

We next show that FAT and MIN AREA grid n-ogons are monotone orthogonal polygons as in the following lemma

Lemma 3.4.5 FAT and MIN AREA grid n-ogons are monotone orthogonal polygon.

Proof. Bajuelos et al. showed that except symmetries of grid, construction of a FAT and MIN AREA grid n-ogon is always like in the Figure 3.4 and 3.5

respectively. If we draw vertical lines from x-axis and horizontal lines from y-axis to a FAT and MIN AREA grid n-ogon, all the lines intersects that FAT and MIN ARE grid n-ogon at most twice. It satisfies the conditions of a polygon to become X Y -monotone. Therefore, FAT and MIN AREA grid n-ogons are
xy-monotone orthogonal polygon.

Q.E .D.

Lemmas 3.4.3, 3.4.4 and 3.4.5 immediately prove the following theorem.

Theorem 3.4.6 For FAT and MIN AREA grid n-ogon Step 3 of Algorithm DM takes O(n) time.

3.5 Conclusion

In this chapter, we have shown that a monotone orthogonal polygon can be represented by a convex bipartite graph. Using this findings we have improve some steps of the existing algorithm of MLSC problem. We also have shown that this improvement is more for some subclasses of grid n-ogon like FAT and MIN AREA grid n-ogons, if they are drawn on an equidistant grid.

Chapter 4

Guarding Semi-orthogonal Polygons

4.1 Introduction

All the results found on art gallery problem with sliding camera so far considers input polygons as orthogonal polygons. But geometric shapes obtained from real art gallery may not always be orthogonal. They may contain some non orthogonal edges. Based on this requirement we consider the MLSC problem for a semi-orthogonal polygon with some constraints. The class of semi-orthogonal polygon is a superclass to the class of orthogonal polygon . Detail description of semi-orthogonal polygons is given in Chapter 2. In this chapter we also show some relationship among different components of orthogonal polygons after being partitioned by one partitioning technique which use reflex vertices.

The rest of the chapter is organized as follows. We start with a brief review of some preliminary concepts and some basic definition on semi-orthogonal polygon in Section 4.2. In this section we also present some relations among different components of an orthogonal polygon after it is being rectangulated by a rectangulation technique. In Section 4.3 we give the result of MLSC problem on semi-orthogonal polygon. Finally, in Section 4.4 we summarize our contributions.

4.2 Preliminaries

4.2.1 Covering Semi-orthogonal Polygons by Sliding Cameras

Let P be a semi-orthogonal polygon. Sliding cameras travel back and forth along the boundary of a semi-orthogonal polygon. The cameras can see inside the polygon orthogonally. The objective of this problem is to find the minimum boundary length where sliding camera travels and all the sliding cameras together cover the whole P . Figure 4.1 illustrates the MLSC problem for a semi-orthogonal polygon as input. For an input polygon in Figure 4.1(a), Fig-

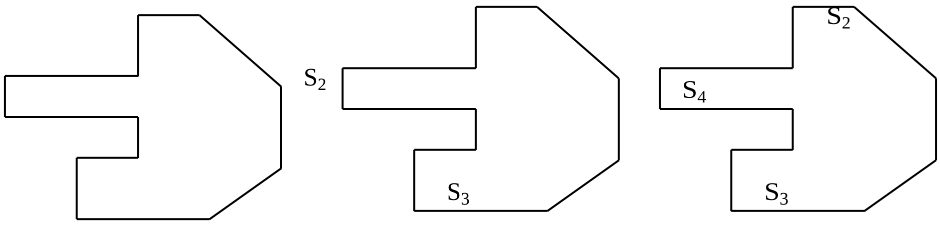


Figure 4.1: Covering a semi-orthogonal polygon by sliding cameras. (a) input semi-orthogonal polygon (b) minimum length sliding camera locations (C) not provide the minimum length sliding cameras locations.

ure 4.1(b) indicates the minimum travel length by sliding cameras S_1 , S_2 and S_3. Cameras S_1, S_2 and S_3 also covers the whole polygon. Figure 4.1(c) is an example where all the cameras cover the whole polygon but the total length traveled by the cameras is not the minimum.

4.2.2 Semi-orthogonal Polygons with Constraints

Let P be a semi-orthogonal polygon which do not have pair of edges such that one of the edge is non orthogonal and they remain close to each other externally maintaining a distance, lower than the higher length edge. Figure 4.2 describes these classes of semi-orthogonal polygons. Polygons in the Figure 4.2(a) and (b) are valid but Figure 4.2(c) is not a valid polygon. In Figure 4.2(c) external distance between the two non-orthogonal edges is less than the length of the higher length edge.

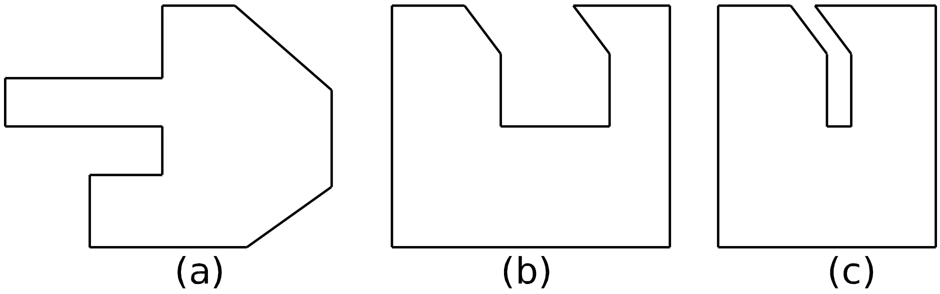


Figure 4.2: Constraints on semi-orthogonal Polygons (a) valid (b) valid (c) invalid.

4.2.3 Co-grid and Chord of a orthogonal polygon

Let u and v be two reflex vertices of an orthogonal polygon P that do not share the same edges of P . u and v are co-grid if they lie on the same horizontal or vertical line. A chord is a line segment fully contained in P that connects two cogrid reflex vertices. In Figure 4.3 black circles are cogrid and the dotted line is a chord

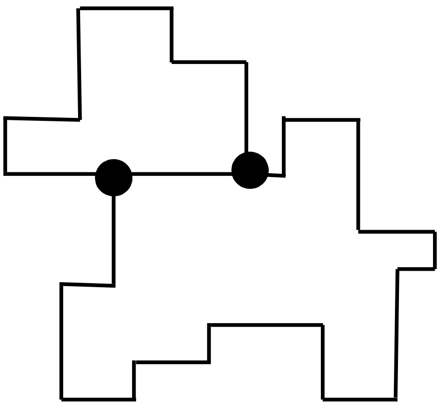


Figure 4.3: An illustration of a co-grid and chord.

4.2.4 Rectangulation of Orthogonal Polygons

We partition an orthogonal polygon using its reflex vertices. If we extend two incident edges of each reflex vertex inward of an orthogonal polygon until it hits the next boundary then the extended edges partition both the interior of the orthogonal polygon as well as the boundary. Generally there are two types of vertices in an orthogonal polygon. If the interior angle of the vertex is 90° then the vertex is a convex vertex and if it is 270° then the vertex is a reflex vertex. The lines that connects two vertices are called edges. Beside vertices and edges this partitioning technique creates several new components in the orthogonal polygon described as follows.

1 Extension of the edges incident to reflex vertices partition the original edges of the orthogonal polygon in to some small edges. Each such small edges is called segment.

2 Extended portion of the edges incident to reflex vertices are called extended edges.

3 Extended edges also partition itself into some small segments, those are called extended segment.

4 Extended edges create some new vertices at the intersection point with other extended edges. Those intersection points are called intersection vertices.

5 Extended edges also create some new vertices at the intersection point with the edges of the orthogonal polygon. Those intersection points are called boundary intersection vertices.

6 Extended edges create some rectangles to the interior of orthogonal polygons. Those rectangles are called r-cut.

In the Figure 4.4 bellow c, r, x and bx represents convex vertex, reflex vertex, intersection vertex and boundary intersection vertex respectively. s and es represent segment and extended segment respectively. Interior rectangle inside P denoted by rc, is an example of r-cut.

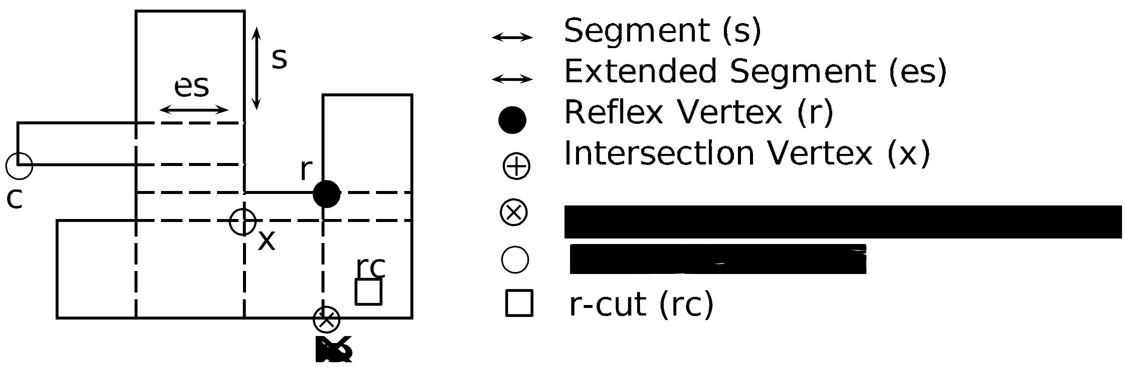


Figure 4.4: Different components of an orthogonal polygon after rectangulation by reflex vertices.

We now show the relations among the above components of an orthogonal polygon.

Lemma 4.2.1 Let P be a simple orthogonal polygon which is rectangulated by extending the incident edges of all the reflex vertices inward of P until they hit the next boundary. If c, r, x, bx, s, es and rc be the total number of convex ver- tex, reflex vertex, intersection vertex , boundary intersection vertex, segment, extended segment and r-cut in P respectively then the components always main- tains the following relation $c + r + x + bx - s - es + rc = 1$

Proof. Let P be a simple orthogonal polygon which is rectangulated by a technique described in Section 3.2.1. We now reduce P into a graph G using the following rules.

1 Each convex and reflex vertex in P corresponds to a vertex in G.

2 Each intersection of any edge and any extended edge in P corresponds to a vertex in G.

3 Each intersection of any two extended edges in P corresponds to a vertex in G.

4 Two vertices in G are connected if there is a connectivity in the corre- sponding vertices and intersection points in P .

Figure 4.5 illustrates the construction of graph G from the orthogonal polygon P in Figure 4.4. We consider all the intersection points in P as vertices of G.

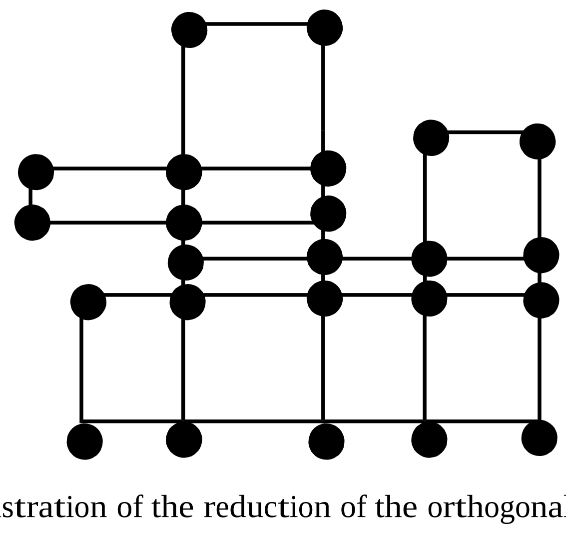


Figure 4.5: An illustration of the reduction of the orthogonal polygon in Figure 4.4 to a graph.

Again in G we connect the vertices based on the connectivity of intersection points in P. It implies that there is no edge crossing in G. So, G is a planer graph. Let n, m and f be the number of vertices, number of edges and num- ber of faces in G respectively. Then by Eular's formula for G we can write

$n - m + f = 2$.

Vertices V of G corresponds to four types of components in P i.e convex, reflex, intersection and boundary intersection vertices. Again edges of G corre- sponds to two types of components in P i.e. segments and extended segments. Similarly faces of G corresponds to the r-cuts and exterior portion of P. Let c, r, x, bx, s, es and rc be the total number of convex vertex, reflex vertex, inter- section vertex , boundary intersection vertex, segment, extended segment and r-cut in P respectively. We now reconstruct the polygon P from graph G. We also reconstruct the Eular's formula in G by the corresponding components of P. Then for P Eular's formula becomes $c + r + x + bx - s - es + rc + 1 = 2$, i.e. $c + r + x + bx - s - es + rc = 1$ Q.E.D.

Lemma 4.2.2 Let P be a simple orthogonal polygon which is rectangulated by extending the incident edges of all the reflex vertices inward of P until they hit the next boundary. If P is chord free and has R number reflex vertices, then P has total $4r + 4$ number of segments. But if P has ch number of chord then total number of segments in P is $4r + 4 - 2ch$.

Proof. By properties of polygons described in Chapter 2 any n vertex orthogonal polygon P have n number of edges. O'Rourke [14] showed that an orthogonal polygon has total number of vertices, $n = 2r + 4$ where r is the number of reflex vertices. Therefore total number of edges in an orthogonal polygon, $E = 2r + 4$. In the rectangulation technique we extend two incident edges of reflex vertices inward until they hit the next boundary. If the polygon is chord free then the extended edges always bisect another edge of the polygon and creates two additional edges. r number of reflex vertices creates 2r additional edges. Therefore total number of edges becomes $E = 2r + 2r + 4 = 4r + 4$. In fact here the edges are the segments created by reflex vertices.

By definition a chord is a line segment fully contained in P that connect two cogrid vertices or reflex vertice. Therefore a chord is an extended segment. But

this extended segment do not bisect the opposite boundary as two ends of a chord are two reflex vertices. Therefore, after rectangulation for each chord two additional segments are not added in the orthogonal polygon. If we introduce a chord in a rectangulated chord free orthogonal polygon, the chord reduces two segments from the total number of segments in that chord free orthogonal polygon. Following example gives explanation of this statement. Let P be a chord free orthogonal polygon. After rectangulaion, P has total $4r + 4$ segments. Adjust the reflex vertices of P in such a way that total number of edges in P remains same but it introduces total ch number of chord in P. Then after
rectangulation, the new orthogonal polygon has $4r+4-2ch$ number of segments.

Q.E.D.

Theorem 4.2.3 Let P be a simple orthogonal polygon which is rectangulated by extending the incident edges of all the reflex vertices inward of P until they hit the next boundary. Let x, es and rc be the total number of intersection vertex, extended segment and r-cut in P respectively. Then the components of P always maintains the following relation, $x - es + rc = 1$.

Proof. Let P be a simple orthogonal polygon which is rectangulated by a technique described in Section 3.2.1. Let c, r, x, bx, s, es and rc be the total number of convex vertex, reflex vertex, intersection vertex , boundary intersection vertex, segment, extended segment and r-cut in P respectively. By Lemma 4.2.1 we get the following relation among the above components $c + r + x + bx - s - es + rc = 1$. If P is chord free then by Lemma 4.2.2
$S = 4r + 4$. Again each reflex vertex creates two boundary intersection ver- tices. Therefore we can write $bx = 2r$. We now replace the value of s and bx in the equation of Lemma 4.2.1 by the new value. The equation becomes
$c + r + x + 2r - 4r - 4 - es + rc = 1$, i.e. $c - r + x - es + rc = 5$. If P has n number of vertices then $n = 2r + 4$, i.e. $c + r = 2r + 4$, i.e. $c - r = 4$. If we put
the value of $C - R$ in the earlier equation, the equation becomes $x - es + rc = 1$.

But if P has ch no of chord then by Lemma 4.2.2 $s = 4r + 4 - 2ch$. If we introduce a chord in a rectangulated chord free orthogonal polygon, the chord reduces two boundary intersection vertices from the total number of boundary intersection vertices in that chord free orthogonal polygon. Therefore for orthogonal polygon with chord $bx = 2r - 2ch$. We now replace the value of s and bx in the equation of Lemma 4.2.1 by the new value. The equation becomes c + r + x + 2r − 2ch − 4r − 4 + 2ch − es + rc = 1, i.e. c − r + x − es + rc = 5. Replacing the value of c − r we get x − es + rc = 1 Q.E.D.

4.3 MLSC Problem on Semi-orthogonal Polygons

Let P be a semi-orthogonal polygon where non orthogonal edges do not have external proximity edges. In our algorithm we first reduces P to an orthogonal polygon. Then we execute a new algorithm on reduced orthogonal polygon to solve the MLSC problem. At the end we reconstruct the semi-orthogonal polygon from the reduced orthogonal polygon and we show that results achieved from the reduced orthogonal polygon is also hold for the semi-orthogonal polygon. Details of the procedure above is described in the subsequent paragraph. We first show the reduction technique.

4.3.1 Reduction to Orthogonal Polygons

In the reduction technique first we partition the P based the non-orthogonal edges then we reduce it to an orthogonal polygon. At the end we assign length to the boundaries of P . The detail description of each step is given below.

Step 1. Partitioning by non orthogonal edges: We partition P by drawing a horizontal or a vertical or both (whichever possible) lines interior to P from both vertices of all the non orthogonal edges. If the vertical or horizontal line intersects other non orthogonal edge then again draw a horizontal or a vertical or both (whichever possible) lines interior to P from that intersection point. Orthogonal edges in semi-orthogonal polygons have two types of visibility. Horizontal edges has vertical visibility on the other hand vertical edges has horizontal visibility. But non orthogonal edges can have any one type of visibility from these two types. After drawing horizontal and vertical lines from two vertices of a non-orthogonal edge, it gets two corresponding segments from the opposite horizontal and vertical boundary. If length of the vertical segment is smaller than the length of horizontal segment then the non orthogonal edge gets horizontal visibility. Otherwise it gets vertical visibility. Figure 4.6(b) illustrates the partitioning of the input semi-orthogonal polygon in 4.6(a) based on the vertices of non orthogonal edge.

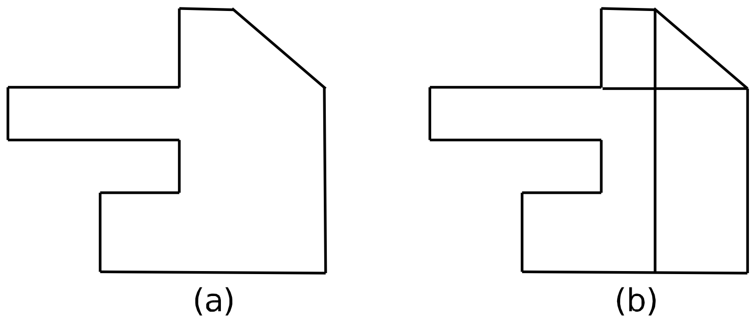


Figure 4.6: Rectangulating a semi-orthogonal polygon based on non orthogonal edges. (a) the input Semi-orthogonal polygon (b) a semi- orthogonal polygon after rectangulation.

Step 2. Omitting non orthogonal edges: After partitioning we replace all the non orthogonal edges of P by drawing right angle triangles exterior to P , taking each non-orthogonal edges as hypotenuse. Figure 4.7 illustrates the the omitting procedure of non orthogonal edges from a semi-orthogonal polygon and reducing it to another polygon.

Step 3. Assigning weights: Segments of the reduced orthogonal polygon

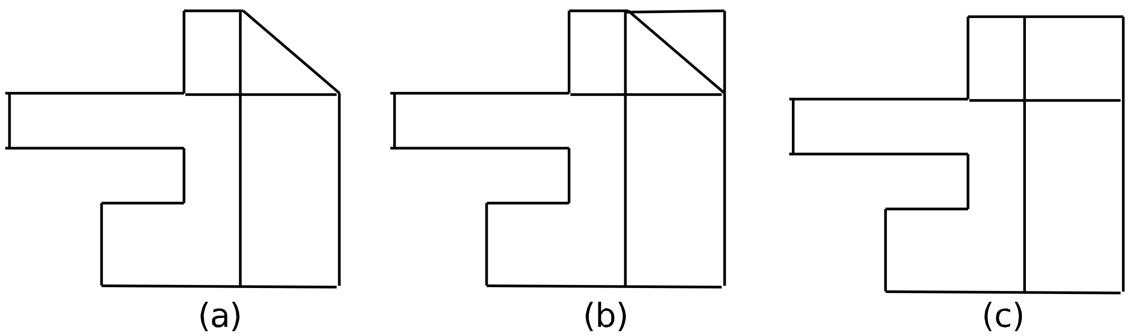


Figure 4.7: Omitting non orthogonal edges from a semi-orthogonal polygon. (a) a semi-orthogonal polygon after rectangulation (b) drawing right angle triangle (c) the reduced polygon.

are assigned with a weight based on the length of the segments of the semi-orthogonal polygon. If the non-orthogonal edges have vertical visibility then bases of right angle triangles gets the length of the respective hypotenuses as their weight where perpendiculars receives an infinite weight. On the other hand if the non-orthogonal edges have horizontal visibility then perpendiculars of right angle triangles gets the length of the respective hypotenuses as their weight where bases receives an infinite weight. All other edges in the reduced polygon get their weights from the length of respective segments of the semi-orthogonal polygon. Figure 4.8(b) illustrates the assignment of weight to the reduced orthogonal polygon from the input semi-orthogonal polygon in Figure 4.8(a).

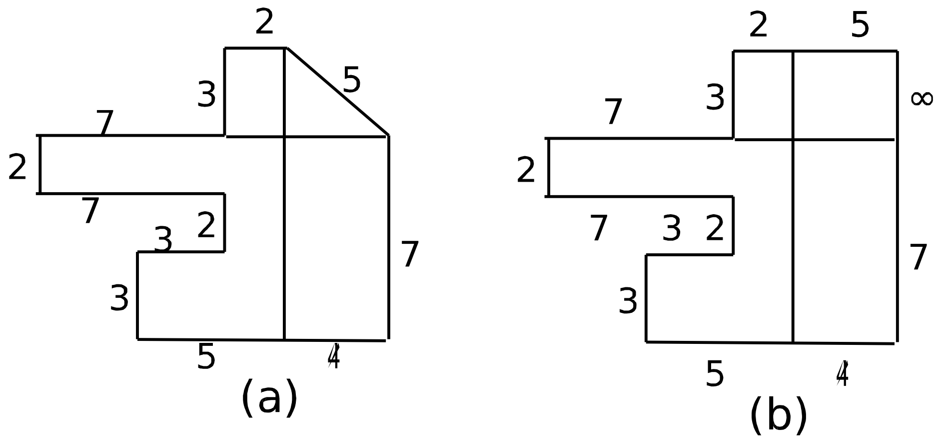


Figure 4.8: Assignment of weight to the boundaries of a reduced polygon. (a) an input semi-orthogonal polygon with boundary lengths (b) different boundary weights of the reduced polygon from the semi-orthogonal polygon in Figure 4.8(a).

We can write the following lemma on the reduced polygon P'

Lemma 4.3.1 The reduction technique mentioned above always reduces a semi-
orthogonal polygon P to an orthogonal polygon P'.

Proof. In this proof we first show that the reduction technique described above is always possible for all semi-orthogonal polygons that do not have external proximity edges with non orthogonal edges, and then we show that the reduced polygon is an orthogonal polygon. At the end we show the correctness of the reduction.

Non-orthogonal edges of a semi-orthogonal polygon has always orthogonal neighbors. Again vertices of non-orthogonal edges always has an interior an- gle less than 270°. Furthermore we only consider a subclass of semi-orthogonal polygons which do not have external proximity edges with non-orthogonal edges. All the above three conditions always ensures sufficient space outside a semi-orthogonal polygon to draw right angle triangles taking non-orthogonal edges as hypotenuse.

Except the non-orthogonal edges, all other edges of P are orthogonal. In the reduction technique we replace each such non-orthogonal edges by two edges i.e. the bases and perpendiculars of the right angle triangles. Interior angles between these two edges are always 90°. Which implies that all the non-orthogonal edges are replaced by two orthogonal edges. Therefore the reduced polygon be- come orthogonal.

In the reduction technique the right angle triangles are always drawn outside P therefore no interior portion of the semi-orthogonal polygon are excluded in the reduced polygon. Rather reduced polygon includes some exterior portion along with the whole interior portion of P . Since visibility of the sliding camera are orthogonal and its direction is towards interior to the polygon, therefore each base of the right angle triangle can cover both the newly included exterior portion and the earlier interior portion covered by each hypotenuse (i.e. the non-orthogonal edges). The perpendiculars of right angle triangles are the extra edges which does not exist in the input semi-orthogonal polygon. But they gets an ∞ weight which always excluded them from consideration in the subsequent part of algorithm.
Q.E .D.

Let P' be the reduced orthogonal polygon. After reduction we rectangulate P' using the rectangulation technique described in Section 3.2.1 of Chapter 3. If the rectangulation intersects an non orthogonal edge then draw a horizontal or a vertical or both (whichever possible) lines interior to P from that intersection point as described in Section 4.3.1.Figure 4.9 describes the rectangulation by reflex vertices and boundary segment's weight calculation of the reduced orthogonal polygon.

This rectangulation technique partition the interior and boundary of P' into a set of r-cut and segments respectively like before. Let R(P) be the set of r-cut. The weight of the segments in the bases are calculated according to the ratio of the weight of the entire bases. Each segments in the perpendicular edges again

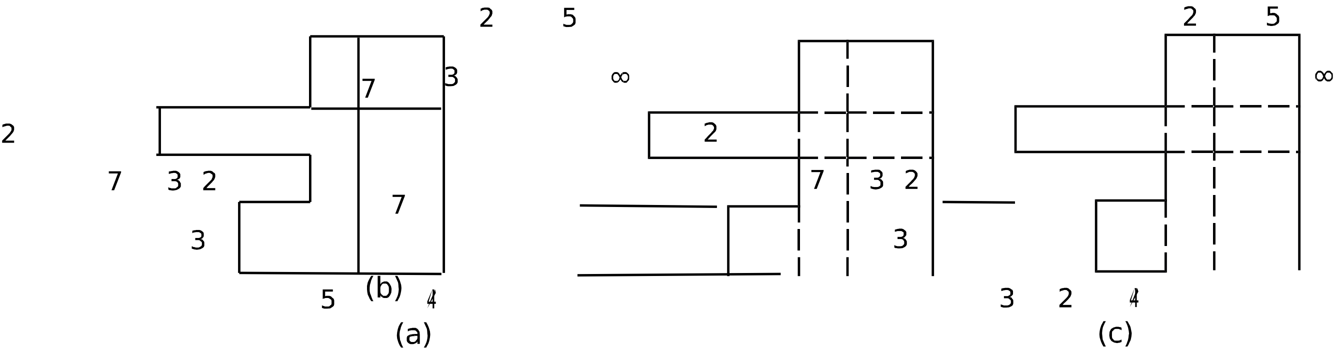


Figure 4.9: Illustration of rectangulation of the reduced orthogonal polygon by reflex vertices. (a) a reduced orthogonal polygon with boundary weights (b) rectangulaiton by reflex vertices (c) after rectangulation weight of the boundary segments.

get the ∞ value as weight. All other segments get their respective values as weight.

We can write the following observation on the r-cut and segments of P'.

Observation 4.3.2 In P' weight of a $lp \in LP$ and a $rp \in RP$ that covers same row of r-cut are not always equal. Similarly weight of a $tp \in TP$ and a $bp \in BP$ that covers same column of r-cut are not always equal.

In the Figure 4.9(C) top facing segment tp and bottom facing segment bp covers the same column of r-cut but their weight are not equal.

We now can write the following lemma on the reduced orthogonal polygon P'.

Lemma 4.3.3 Let C be a cover of P'. Let C' be the line segments obtained from C by executing the following operations on P'

1 Replace each lp (Resp. rp) by rp (Resp. lp) if both the segments cover same row of r-cut and weight of rp (Resp. lp) is smaller than weight of lp (Resp. rp).

2 Replace each tp (Resp. bp) by bp (Resp. tp) if both the segments cover same column of r-cut and weight of bp (Resp.tp) is smaller than weight of tp (Resp. bp)

Then C' is also a cover of P' and the sum of weight of the segments in C' is always less than or equal to the sum of the weight of the segments in C.

Proof. By Observation 4.3.2 weight of a left and a right facing segment that covers same row of r-cut are not always equal. Similarly weight of a top and a bottom facing segment that covers same column of r-cut are not always equal. In the above replacement operation, for each row of r-cut we choose smaller weight segment from the left and right facing segment that cover the same row r-cut. Similarly for each column of r-cut we choose a smaller weight segment from top and bottom facing segment that guards the same column of r-cut. Thus all the row of r-cuts and column of r-cuts are guarded by at least one segment. Guarding all the row of r- cuts and column of r-cuts means guarding the whole polygon. Again for each row and column of r-cut we choose the segment whose weight is smaller than other. Therefore it makes the sum of weight of segments smaller than before. If all the smaller segments are chosen at the beginning then no replacement operation takes place. Q.E.D.

We consider C' as a regular cover of P'. Let $S(P')$ be the set of segments those are in C'.

Durocher and Mehrabi in [8] showed that a single r-cut can be entirely covered by only horizontal or only vertical line segments. From that findings we can write the following lemma for the reduced orthogonal polygon P'.

Lemma 4.3.4 Let $R \in R(P')$ be a r-cut and C be a cover P' then, there exist a set $C' \subseteq C$ such that all line segments in C' have the same orientation and (i.e., they all are horizontal or they all are vertical) and they collectively guard R entirely.

The following Lemma follows from Lemma 4.3.3 and 4.3.4

Lemma 4.3.5 Reduced orthogonal polygon P' has an optimal cover C from $S(P')$.

Observation 4.3.6 Let P' be a reduced orthogonal polygon and $R(P')$ be the r-cuts created using the partition technique described above. Then each r-cut in $R(P')$ is seen by exactly one vertical and one horizontal line segment from $S(P')$. Again if $C \subseteq S(P')$ is a cover of P' then every r-cut in $R(P')$ must be seen by at lest one horizontal or one vertical line segment in C.

In Figure 4.10(a) let $S(P') = tp_1, tp_2, tp_3, lp_1, lp_2, lp_3, rp_4$ be the segment set.

Here r-cut rc_1 covered by one horizontal segment tp_3 and one vertical segment

rp_4. Segments $tp_3, rp_4 \in S(P^{'}$. If $C \subseteq S(P^{'})$ is an optimal cover of $P^{'}$ then at least one segment from C covers the r-cut rc_1. We now construct an undirected weighted graph $G_{P^{'}} = (V, E)$ by the segments $S(P^{'})$ and r-cuts of $P^{'}$ following the same rules described in Section 3.2.1 of chapter 3. The graph in Figure 4.10 (b) is constructed from the reduced orthogonal polygon in Figure 4.10(a). We

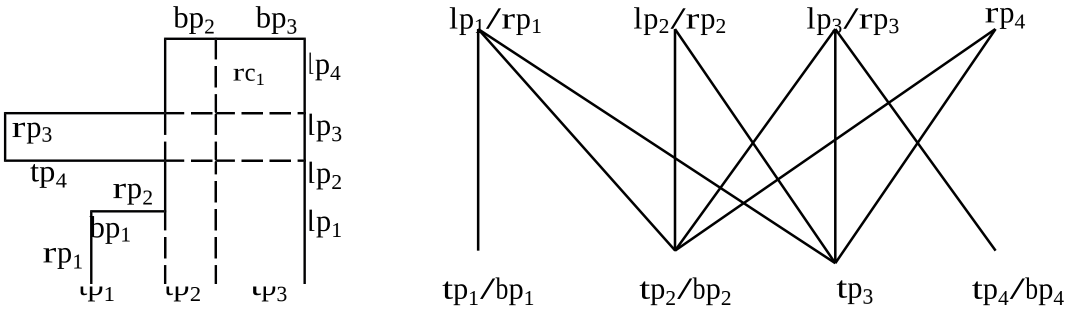


Figure 4.10: An illustration of constructing a bipartite graph from the reduced orthogonal polygon.

next show that MLSC problem on the semi-orthogonal polygon P reduces to the minimum weight vertex cover problem on $G_{P^{'}}$. For this we first need the following lemma

Lemma 4.3.7 Let $T(P^{'})$ be the set of all the perpendiculars of the right angle triangle drawn in $P^{'}$ taking non orthogonal edges as hypotenuse. Then $S(P^{'}) \cap T(P^{'}) = \emptyset$.

Proof. By Lemma 4.3.3, $S(P^{'})$ comprises with the segments each of whose weight is smaller than the corresponding segment that cover same row or column of r-cuts. Each time the perpendiculars are compared with other segment, perpendiculars are always excluded as the weight of perpendiculars considered as ∞. At the end no perpendicular edges is included in the set $S(P^{'})$. Which implies that $S(P^{'}) \cap T(P^{'}) = \emptyset$. Q.E.D.

Durocher and Mehrabi in [8] showed that MLSC problem on an orthogonal polygon reduces to the minimum weight vertex cover problem on a graph which is constructed from that orthogonal polygon using the algorithm described earlier. Since the reduced polygon $P^{'}$ is an orthogonal polygon and graph $G_{P^{'}}$ is constructed from P' using the said algorithm then directly we can write the following Lemma using the above result

Lemma 4.3.8 MLSC problem on reduced orthogonal polygon P' reduces to the minimum weight vertex cover problem on $G_{P'}$.

We now show the equivalencey of the MLSC problem on a semi-orthogonal polygon and the corresponding reduced orthogonal polygon.

Lemma 4.3.9 MLSC problem on semi-orthogonal polygon P reduces to MLSC problem on reduced orthogonal polygon P'

Proof. Let C be the minimum length camera positions in the reduced orthogonal polygon P'. We now reconstruct the semi-orthogonal polygon P from the reduced orthogonal polygon P' by replacing the perpendiculars and bases with the hypotenuses. Then we set the minimum length camera positions C , which is obtained from P', to P . By Lemma 4.3.7 additionally drawn perpendicular edges of P' are never included in C . Weight of the bases are equal to the length of the corresponding hypotenuses. Again all other segment's weight in P' is equal to the length of the segments in P . Therefore, we directly can set the the minimum length camera positions from P' to P . Q.E.D.

To find out the computational complexity, next we show that graph $G_{P'}$ is a bipartite graph.

Lemma 4.3.10 Graph $G_{P'}$ constructed from the reduced orthogonal polygon P' is a bipartite graph.

Proof. In $G_{P'}$ there are two types of vertices. One type corresponds to the vertical segments and other type is the horizontal segments of $S(P')$. In $S(P')$ for each row and column of r-cut there is one vertical and one horizontal segment respectively to cover. Therefore no two vertical or horizontal segments can see the same r-cut simultaneously. It implies that no two vertices of the same type are neighbor. Vertices are connected only among the two types. Its satisfies the both the conditions (two types of vertices and No connectivity among the same type of vertices) for a graph to become a bipartite graph. Q.E.D.

Regarding the computational complexity of the algorithm we can write the following lemma

Lemma 4.3.11 Algorithm for the MLSC problem on semi-orthogonal polygons takes $O(n^2)$ time.

Proof. Following operations are required for an optimal solution of MLSC problem on semi orthogonal polygons

Algorithm MLSC

Input : n vertex semi-orthogonal polygon.

Output : The minimum length boundaries that cover the whole polygon.

1. Partition the polygon by the vertices of non orthogonal edges.
2. Draw right angle triangle taking each non orthogonal edges as hypotenuse assigning weight to each edges i.e reducing the semi-orthogonal
   polygon P into an orthogonal polygon P'.
3. Partition the reduced orthogonal polygon by extending two edges of reflex vertices.
4. Compare each $tp \in TP$ with corresponding $bp \in BP$ that covers same set of r-cut $R_1(P') \subseteq R(P')$. Similarly comparing each $lp \in LP$ with corresponding $rp \in RP$ that covers the same set of r-cut. Select the segments which have smaller length from each comparison. Create a set $S(P')$ with those segments.
5. Construct a bipartite graph $G_{P'}$ from the reduced orthogonal polygon P' considering the segments set$S(P')$ as vertices.
6. Execute the minimum weight vertex cover algorithm on bipartite graph $G_{P'}$.
7. Reconstruct the semi-orthogonal polygon P from the reduced orthogonal polygon P' and also set the minimum length camera positions to P.

Figure 4.11 and 4.12 illustrates all the complete algorithm. Steps 1, 2, 3, 4 and 7 require linear time complexity, steps 5 and 6 require $O(n^2)$ time complexity. Overall we can say that the algorithm requires $O(n^2)$ time complexity. Q.E.D.

Now from Lemmas 4.3.9 and 4.3.11 we can write the following theorem.

Theorem 4.3.12 Given a semi-orthogonal polygon P with n vertices, there exist an algorithm that solves MLSC problem on P in $O(n^2)$ time.

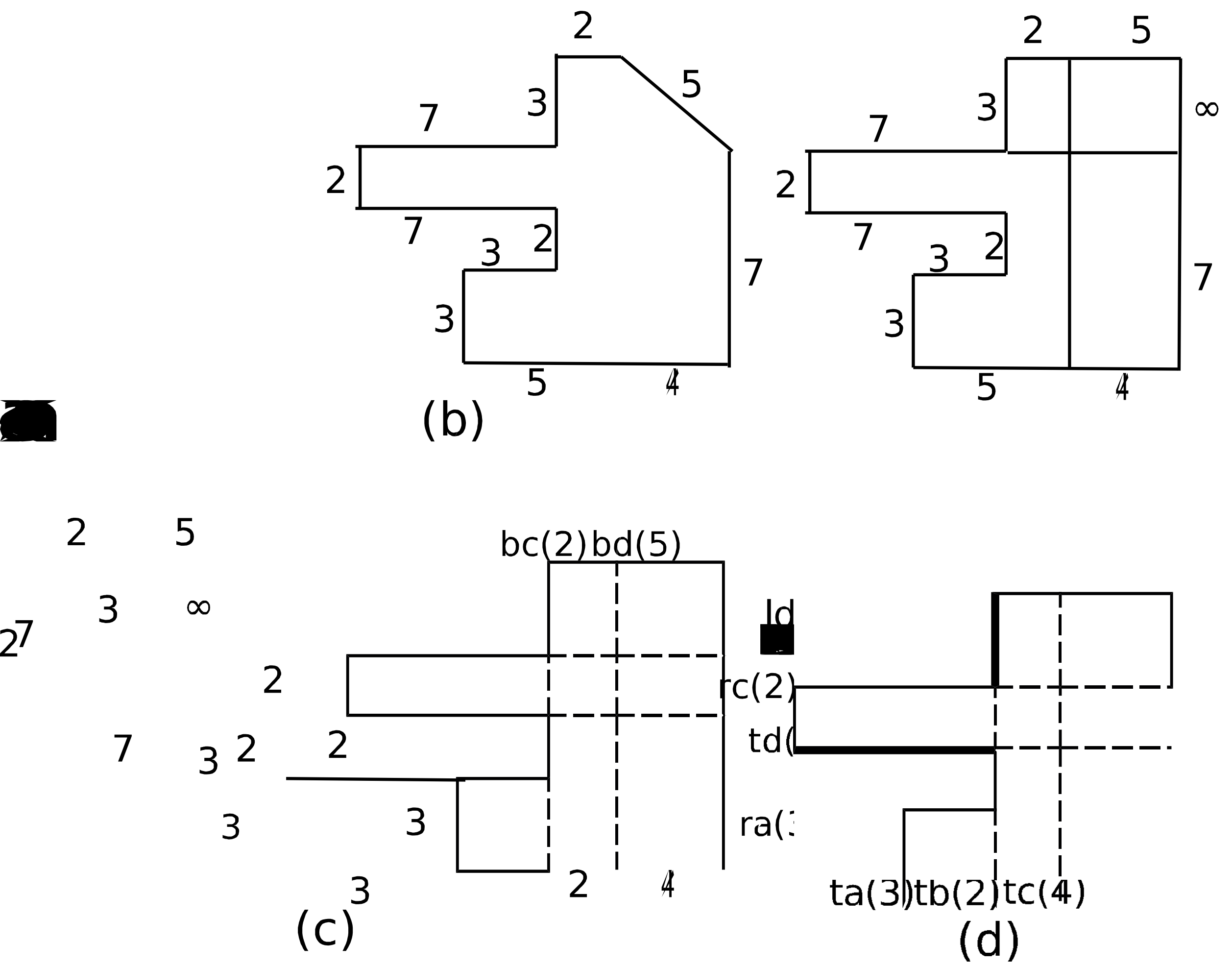


Figure 4.11: An illustration of the algorithm of MLSC problem for semi-orthogonal polygons. (a) an input semi-orthogonal polygon with boundary lengths (b) partitioning by non-orthogonal edge, reducing to orthogonal polygon P' and assigning weights to the segments of P' (c) partitioning by reflex vertices (d) assigning labels to all the segments and constructing $S(P')$ (the bold line segments).

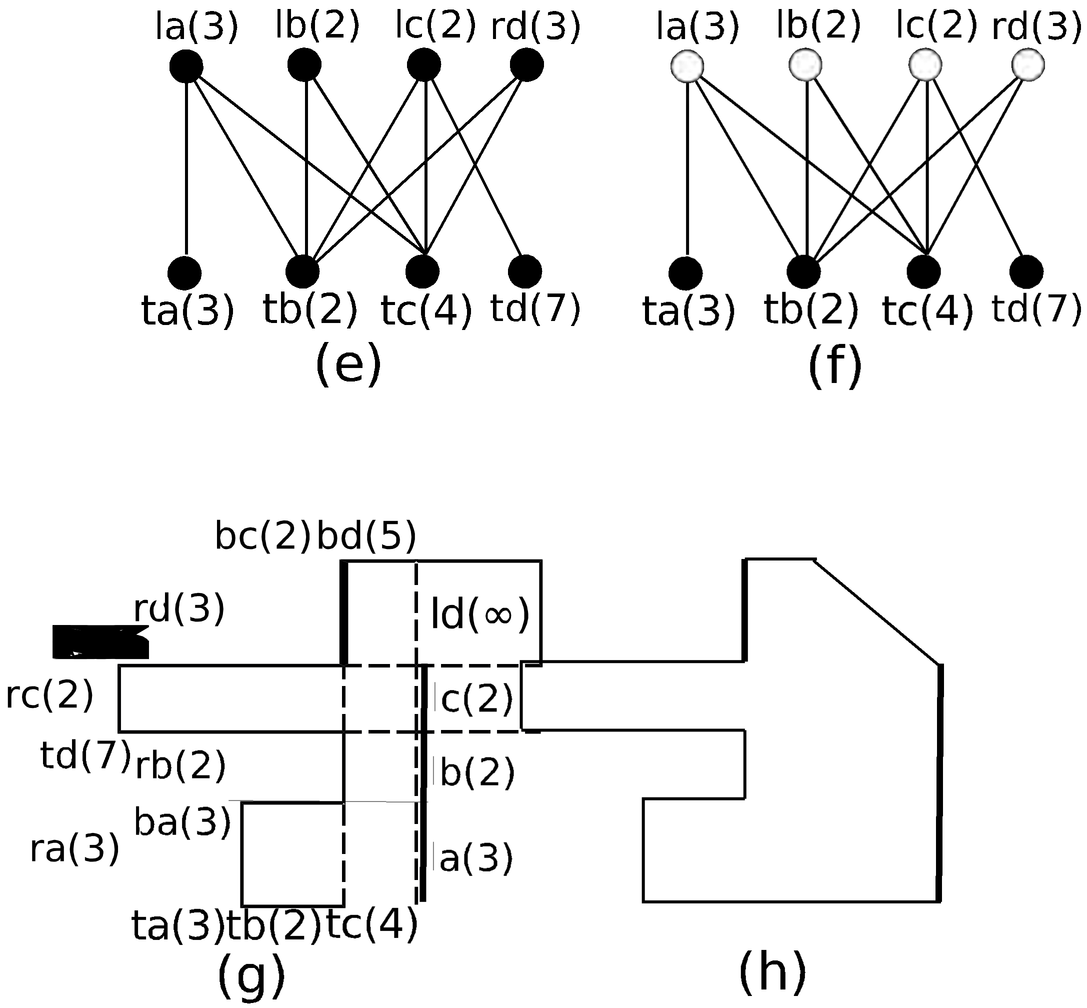


Figure 4.12: An illustration of the algorithm of MLSC problem for semi-orthogonal polygons(continued). (e) construction of bipartite graph $G(P')$ from P' (f) the minimum weight vertex cover of bipartite graph $G(P')$(white vertices). (g) locating the vertices of minimum weight cover in P' (bold line segments) (h) reconstructing the semi-orthogonal polygon P from P' with the minimum length camera positions.

4.4 Conclusion

In this chapter we have developed an algorithm that finds the minimum length cover using sliding cameras for a semi-orthogonal polygon. We also have established some relations among the number of different components of an orthogonal polygon after it is being rectangulated by a rectangulation technique.

Chapter 5

MLSC Problem with k-transmitters

5.1 Introduction

Advancement in the wireless technology introduces another new variation in the art gallery problem. Placing wireless devices or modems inside a building in such a way that receiving strong enough signal by any computer or similar kind of device from that modem is a common problem nowadays. For this the main limitation is not the distance but the number of walls that separate them. Modems has infinite broadcast range and can penetrate up to k number of walls to reach a client. Here k is some fixed integer and $k > 0$. Such kind of modem is called k-transmitter. In computational geometry walls are represented by line segments in polygons. For this thesis we only consider orthogonal polygons. If k- transmitters can slide back and forth then they are called sliding k-transmitters. In this chapter for orthogonal polygons we develop an algorithm that finds the minimum-length sliding k-transmitters with one directional transmission capability and cover the whole polygon also.

The rest of the chapter is organized as follows. We start with problem definition with some relevant terminologies in Section 5.2. In Section 5.3 we give the result of MLSC problem on orthogonal polygons with one directional sliding k-transmitter. Finally, in Section 5.4 we summarize our contributions.

5.2 Preliminaries

5.2.1 Guarding by Sliding k-transmitters

Let P be an orthogonal polygon. Sliding k-transmitters travel back and forth along axis-aligned line segment s inside P and cover omni-directionally. A point p is covered by this guard if there exist a point $q \in s$ such that $\overline{pq}$ is a line segment normal to s and is completely inside P. Again $\overline{pq}$ has at most k intersections with the polygon's boundary walls. The objective is to find minimum-length sliding k-transmitters that cover the entire P. In other words the goal is to find the minimum total length of trajectories on which the sliding k-transmitters travel to cover the entire polygon. The problem is denoted as MLSCk problem. In this chapter we consider some variations of this problem. We consider that sliding k-transmitters travel along the boundary of P and it cover one directionally inside P. We call this problem as modified MLSCk problem. For the orthogonal polygon in Figure 5.1(a),Figure 5.1(b) indicates

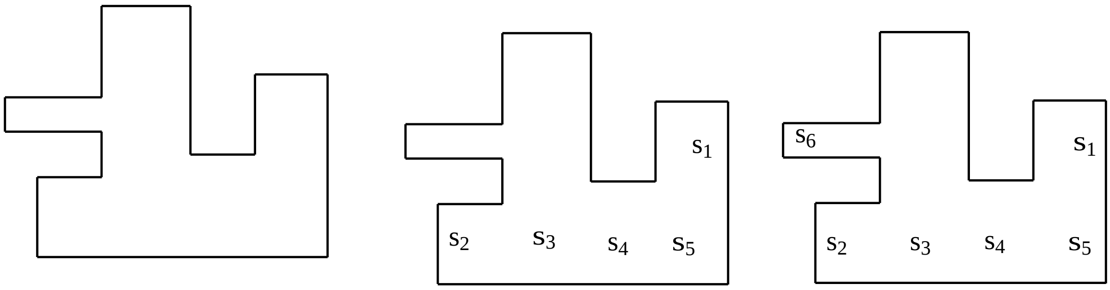


Figure 5.1: MLSCk problem with variation for k = 2. (a) an input orthogonal polygon (b) provide the minimum-lenngth sliding k-transmitters (c) not provide the minimum-length sliding k-transmitters.

the minimum travel length by sliding k-transmitters s_1, s_2, s_3, s_4 and s_5. They also cover the whole polygon. Figure 5.1(c) is an example where all the sliding k-transmitter cover the whole polygon but the total length traveled by sliding k-transmitters is not the minimum.

5.2.2 Classification of Convex and Reflex Vertices

Interior angles of convex vertices of an orthogonal polygon is always 90° but it is 270° for reflex vertices. We classify convex and reflex vertices based on their

monotonicity. Let P be an orthogonal polygon. Draw the smallest possible rectangle exterior to P so that entire P remains inside the rectangle. Extend the two incident edges of each convex vertex externally to the orthogonal polygon until they hit the outer rectangle. Similarly extend the two incident edges of each reflex vertex internally to the orthogonal polygon until they hit the outer rectangle. For a convex vertex if both the extended edges intersects at most one boundary of P or no boundary of P then the convex vertex is called monotone convex vertex. But if any of the extended edge of a convex vertex intersects more than one boundary of P then the convex vertex is called non monotone
convex vertex. Again for a reflex vertex if both the extended edges intersects

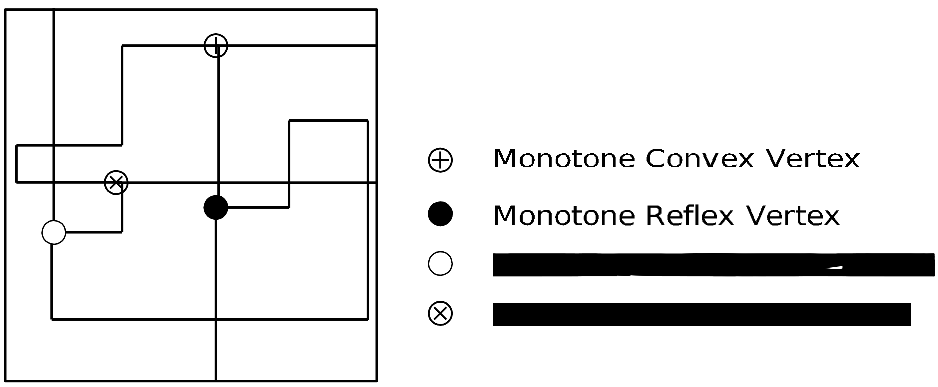


Figure 5.2: Classification of convex and reflex vertices.

at most one boundary of P then the reflex vertex is called monotone reflex vertex. But if any of the extended edge of a reflex vertex intersects more than one boundary of P then the reflex vertex is called non monotone reflex vertex. Figure 5.2 illustrates different classes of convex and reflex vertices.

5.3 Algorithm for Modified MLSCk Problem

In this section we give an algorithm for the modified MLSCk problem. In this algorithm we first rectangulate the input orthogonal polygon by a new rectangulation technique. After rectangulation we create groups among segments by plane sweep algorithm. Then for a fixed k we construct a graph from the rectangulated orthogonal polygon. After that we eliminate redundant vertices and edges from the graph using the group of segments and construct a new graph. Finally we execute minimum weight vertex cover algorithm on the new graph to get the minimum-length sliding k-transmitter. Detail of this algorithm is in the subsequent paragraph.

5.3.1 Rectangulation of the Orthogonal Polygons

Let P be the input orthogonal polygon. We first rectanglulate P using a new rectangulation technique. In this technique a rectangle is drawn exterior to the orthogonal polygon which is described in section 5.2.2. Then extend the

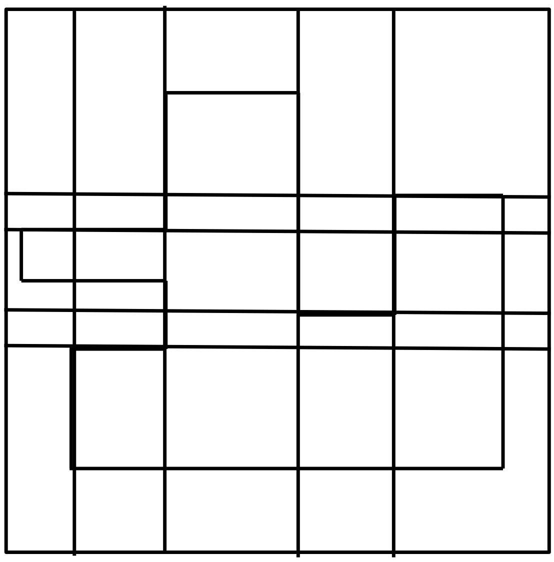


Figure 5.3: Rectangulation of an orthogonal polygon.

two incident edges of all the non monotone convex vertices and all the reflex vertices in both direction until they hit the exterior rectangles. During this extension some of the extended edges may pass along the edges of the orthogonal polygon or extended edges of some other veritces. Beside rectangulating the interior of the orthogonal polygon, this rectangulation technique partitions the boundaries of the orthogonal polygon and the boundaries of the outer rectangle also. Figure 5.3 illustrates the rectangulation technique. We call each of the interior rectangle of the orthogonal polygon as rc-cut. We observe that in order to guard the entire orthogonal polygon, it suffices to guard all rc-cut. The following observation is straight forward

Observation 5.3.1 Let P be an orthogonal polygon and C be a cover of P .
Moreover, let C' be the set of line segments obtained from C by translating every

vertical line segment in C horizontally penetrating at most k boundaries to its right and every horizontal line segment in C vertically penetrating at most k boundaries below it until they reach the nearest boundary of P from the exterior rectangle. Then C' is also a cover of P and the respective sums of the lengths of line segments in C and C' are equal.

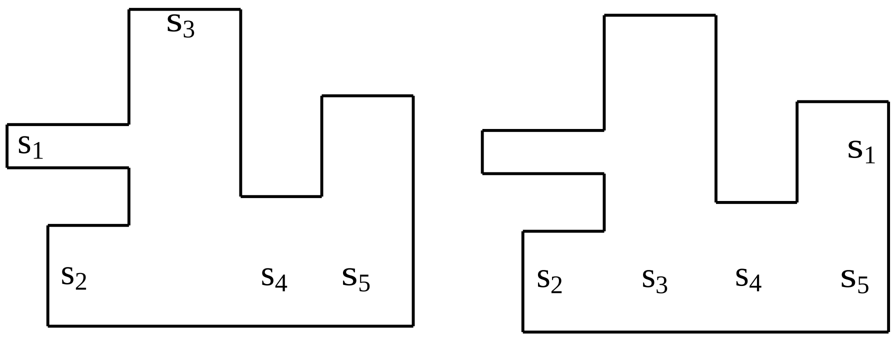


Figure 5.4: (a) Before translating right and bottom facing segments (b) after translating right and bottom facing segments.

Figure 5.4 illustrates the Observation 5.3.1. Figure 5.4(a) uses bottom and right facing boundary for k-transmitters. On the other hand Figure 5.4(b) found after translating Figure 5.4(a). Both 5.4(a) and 5.4(b) provide the same result.

For MLSC problem Durocher and Mehrabi [8] gave a lemma where they showed that each r-cut is entirely covered by either horizontal or vertical line segments. Using that lemma we can directly write the following lemma for modified MLSCk problem

Lemma 5.3.2 Let R be a rc-cut and let C be a cover of P then there exist a set $C' \subseteq C$ such that all the line segments in C' have the same orientation (i.e. they are all vertical or they are all horizontal) and they collectively guard R entirely.

Let B(P) be the set of all top and left facing segments in P created by the rectangulation technique described in Section 5.3.1. Then following lemma follows from Observation 5.3.1 and Lemma 5.3.2

Lemma 5.3.3 Using sliding k-transmitters every orthogonal polygon P has an optimal cover $C \subseteq B(P)$.

5.3.2 Assigning Labels and Weights to Segments

By Lemma 5.3.3, only top and left facing segments are required to optimally cover an orthogonal polygon. This is why we assign labels to only top and left facing segments. Length of each segment is assigned as its weight.

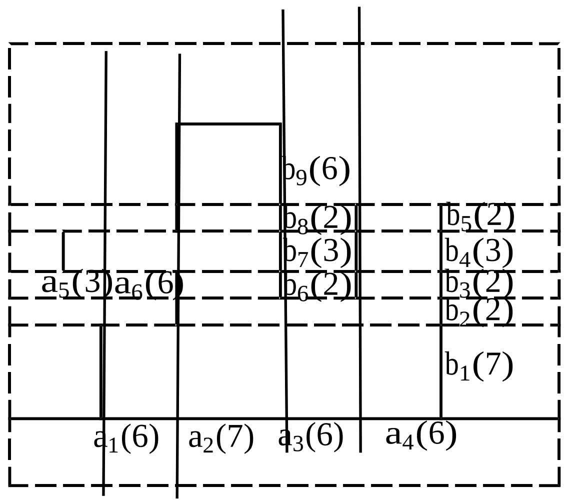


Figure 5.5: An illustration of assigning labels and weights to left and top facing segments.

In the Figure 5.5, labels a_1, a_2, ..., a_6 are assigned to each top facing segments and labels b_1 , b_2 , ..., b_9 are assigned to left facing segments. Numbers associated with each label is the length of that segment.

As we are using k-transmitters as guard, each rc-cut may be covered by more than one left facing or top facing segments. We next group the left and top facing segments those may cover same rc-cut.

5.3.3 Creating Groups among Segments

The rectangulation technique described in Section 5.3 partitions the boundary of the exterior rectangle into some segments. In the Figure 5.6, the notations $s_1, s_2, \cdots, s_{11}$ indicates those segments in the exterior rectangle. Run plane sweep algorithm once for each top facing and for each left facing segment of the exterior rectangle. It starts from the top and left facing segments of the exterior rectangle and ends at the opposite boundary of the exterior rectangle. In each run from a top facing segment of the exterior rectangle, group all top facing segments of the orthogonal polygon. Similarly in each run from a left facing

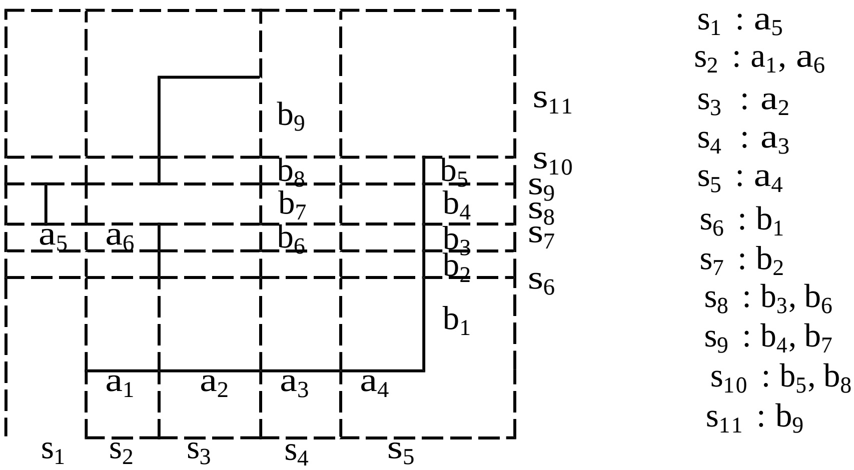


Figure 5.6: An illustration of grouping among segments of an orthogonal polygon.

segment of the exterior rectangle, group all left facing segments of the orthogonal polygon. Each such group is called a segment group. Due to rectangulation technique segments in the same group have equal length. In the Figure 5.6 the plane sweep algorithm starts from segments s_2, s_8, s_9 and s_{10} found more than one top or left facing segments.

We next reduce the orthogonal polygon to a graph for a fixed value of k.

5.3.4 Reduction to Graph for a Fixed k

We construct an undirected weighted graph $G_P = (V, E)$ from P using the following rules:

1 Each segment $s \in B(P)$ corresponds to a vertex $v_s \in V$ such that weight of the v_s is the length of s.

2 Two vertices $v_s, v_{s'} \in V$ are adjacent in G_P, if s and s' do not have same orientation(i.e. if one of the segment is vertical then other segment must be horizontal or vice versa) and s & s' cover a common rc-cut.

Figure 5.8 is the reduced graph from the rectangulated orthogonal polygon in Figure 5.7 for $k = 2$. Here in the Figure 5.7 segments b_3 and b_6 cover one common rc-cut. But vertices b_3 and b_6 in Figure 5.8 are not connected as they have same orientation (i.e. both of them are left facing segments).

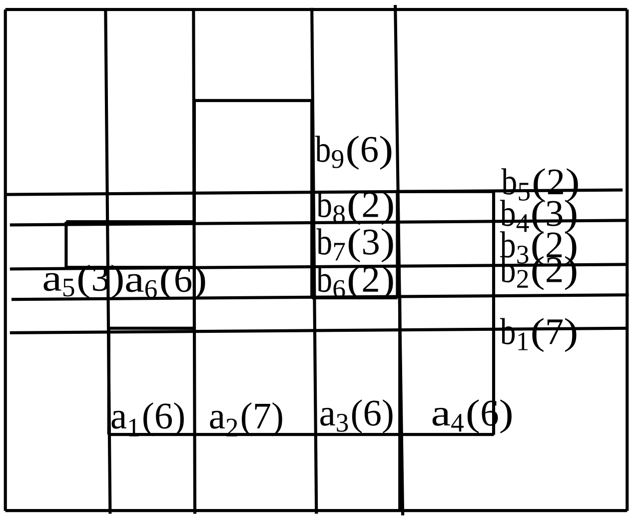


Figure 5.7: A rectangulated orthogonal polygon for reducing to a graph.

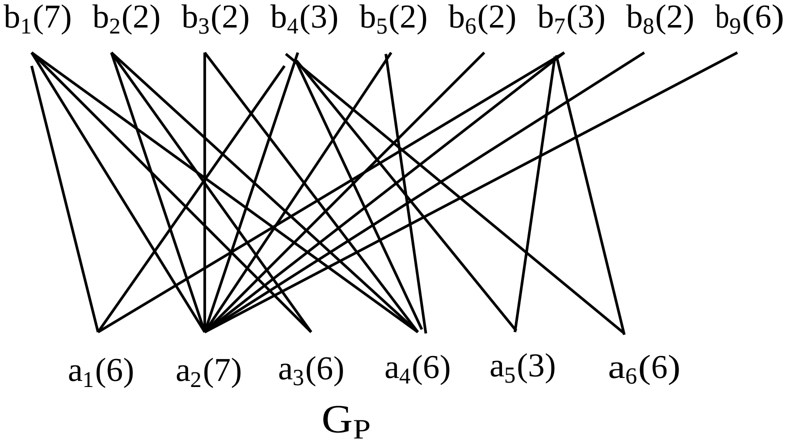


Figure 5.8: An illustration of constructing a graph from a rectangulated orthogonal polygon in Figure 5.7 for $k = 2$.

We next show that graph G_P constructed from P is a bipartite graph.

Lemma 5.3.4 Graph G_P constructed from the orthogonal polygon P is a bipartite graph.

Proof. In G_P there are two types of vertices. One type corresponds to the top facing segments and other type is the left facing segments. Due to the use of k-transmitters two top facing segments or two left facing segments can cover a common rc-cut. Though two segments of same orientation can cover a common rc-cut, but during construction of graph G_P corresponding vertices of such segments are kept disjoint. It ensures that no two vertices of same orientation are neighbors in G_P. Vertices are connected only among the two types. Its satisfies the both conditions (two types of vertices and no connectivity among the same type of vertices) for a graph to become a bipartite graph.

Q.E .D.

5.3.5 Redundant Edges and Vertices in Graph G_P

k-transmitters cover penetrating walls which may causes one rc-cut covered by more than one left facing or more than one top facing segments or both. Therefore G_P can have more than one edge for an rc-cut. Due to the rectangulation technique length of all the left facing segments that cover same rc-cuts are equal. Similarly length of all the top facing segments that cover same rc-cuts are also equal. As length of the segments in P are equal, weight of the vertices in G_P are also equal. Therefore, only one edge is sufficient in G_P to represent an rc-cut. Keeping one edge for an rc-cut if we eliminate all other edges it will not create any impact on minimum-length cover. Therefore, for an rc-cut except one edge all other edges are redundant. For an rc-cut rc_1 in Figure 5.9 there are four edges in G_P (in Figure 5.10)i.e. $a_1 b_4$, $a_1 b_7$, $a_6 b_4$, $a_6 b_7$ (dotted lines in Figure 5.10). Except $a_1 b_4$ all the remaining edges are redundant in G_P.

Furthermore for the same characteristics of k-transmitters there are some segments in P whose covering area is fully covered by another segment of same orientation and equal length. Eliminating corresponding vertices of those segments form G_P does not create any impact on minimum length cover. Therefore such vertices are redundant in G_P. In Figure 5.9, rc-cut covered by the segment b_6 is also covered by the segment b_3. Both b_3 and b_6 are of equal length. b_6 also

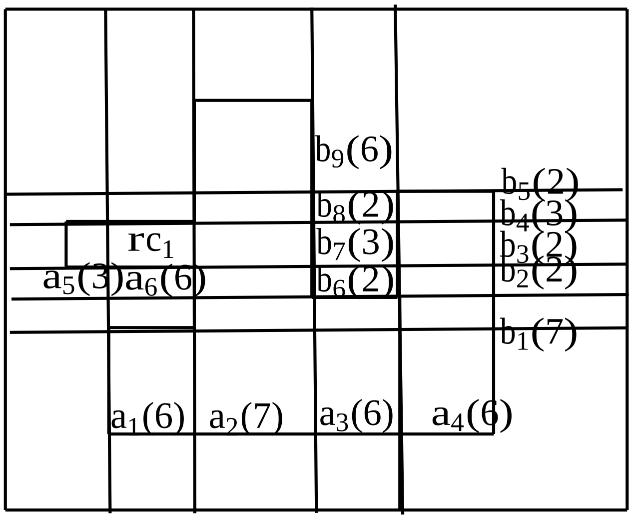


Figure 5.9: A rectangulated orthogonal polygon for redundant vertices and edges.

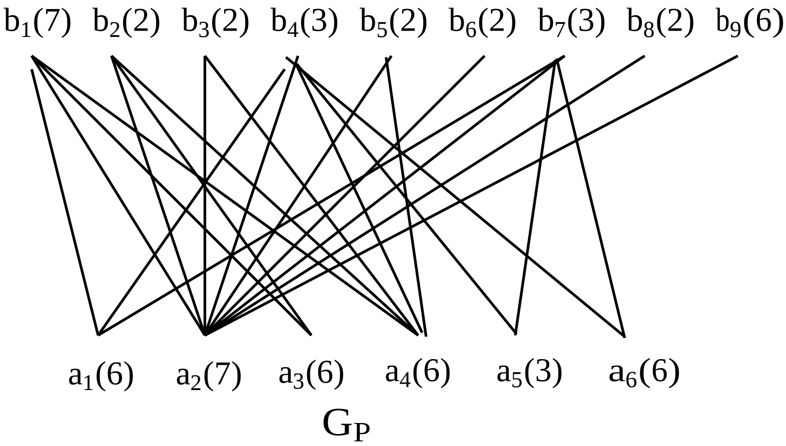


Figure 5.10: An illustration of redundant vertices and edges in graph G_P .

cover one additional rc-cut. Therefore the corresponding vertex of segment b_3 in graph G_P is redundant.

There is a vertex in G_P for each segment in P. Furthermore weight of a vertex in G_P is the length of the corresponding segment in P. Therefore grouping of segments in P can be considered in G_P also. We call each such group in G_P as vertex group. We need to eliminate all such redundant vertices and edges from graph G_P. For this we first need the following observation.

Observation 5.3.5 Redundant vertices and edges are created only from vertex groups which have more than one member in the group.

We next construct a graph G_P' form G_P after eliminating all redundant vertices and edges from G_P. This elimination work is done by executing two set operations in all the vertex groups which have more than one members.

5.3.6 Eliminating Redundant Vertices and Edges from G_P

By Lemma 5.3.4 graph G_P is a bipartite graph. Therefore each vertex has a neighbor set from the opposite partite set. If a vertex group have more than one member, construct neighbor sets of all the vertices in that group. Let NS_{a_m} and NS_{a_n} be two neighbor sets of two vertices of same group a_m and a_n respectively. Then we follow the bellow mention rules to eliminate redundant vertices and edges from G_P

1 If NS_{a_n} (Resp. NS_{a_m}) is subset of NS_{a_m} (Resp. NS_{a_n}) then we can eliminate vertex a_n (Resp. a_m) and its associated edges from G_P.

2 If NS_{a_n} and NS_{a_m} are not subset of each other but have common element b_x then we can eliminate either edge $a_m b_x$ or $a_n b_x$ from G_P.

In the Figure 5.11 neighbor set of vertex a_1 is b_1, b_4, b_7. Again neighbor set of same group vertex a_6 is b_4, b_7. As neighbor set of vertex a_6 is subset of same group vertex a_1, we eliminate vertex a_6 and its associated edges from G_P. Similarly vertices b_6, b_7 & b_8 and their associated edges are also eliminated from graph G_P. Figure 5.13 shows the graph G'_P which is constructed after eliminating redundant vertices and edges from G_P (shows in Figure 5.12). We

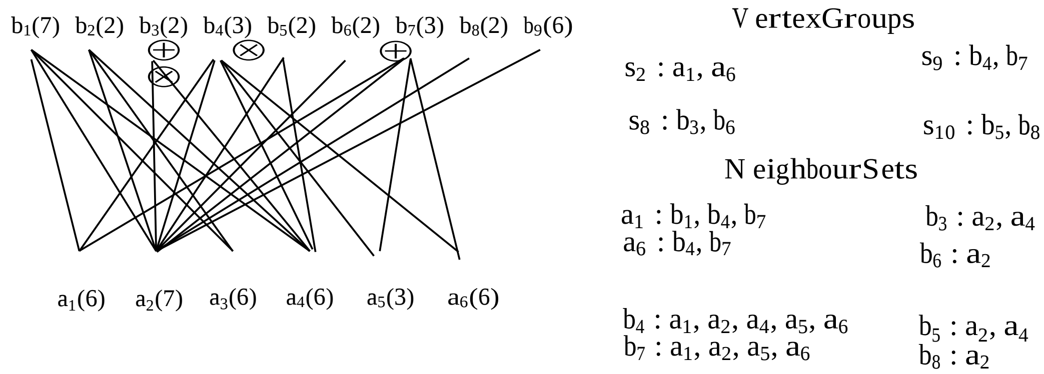


Figure 5.11: An illustration of neighbor sets of group vertices for $k = 2$.

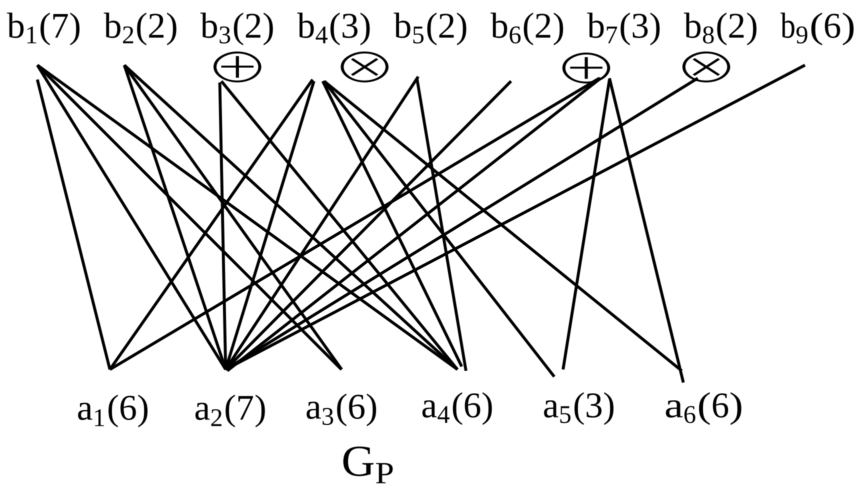


Figure 5.12: Graph before eliminating redundant vertices and edges.

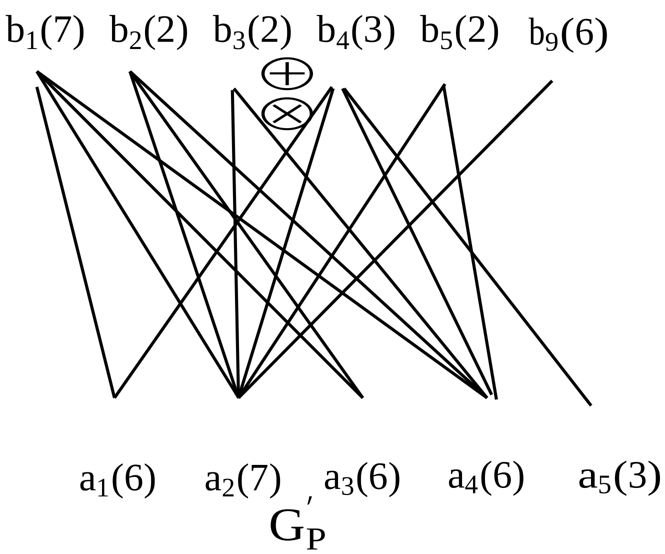


Figure 5.13: Graph after eliminating redundant vertices and edges.

next show that set operations described in this section eliminate all redundant vertices and edges from graph G_P .

Lemma 5.3.6 G'_P is constructed after eliminating all the redundant vertices and edges from G_P .

Proof. Let $NS_{a_m} \subseteq NS_{a_n}$. It implies that NS_{a_n} contains all the elements of NS_{a_m} and also some additional vertices. It indicates that the corresponding segment of the vertex a_n in P covers all the rc-cuts, covered by the corresponding segment of the vertex a_m in P . Beside those rc-cuts, the corresponding segment of a_n in P covers some additional rc-cuts also. As a_m and a_n are in the same group, their corresponding segments in P have equal length and both of them are either top facing or left facing segment. For these two reasons if we eliminate the vertex a_m and its associated edges from G_P it does not effect the minimum length covering. Therefore vertex a_m and its associated edges are redundant in G_P and we can eliminate them.

Let NS_{a_m} and NS_{a_n} are not complete of each other but they have a common vertex. Let b_x be the common vertex. Edges $a_m b_x$ and $a_n b_x$ represents common rc cut in P . As a_m and a_n vertices are in the same group, their corresponding segments in P have equal length and both of them are either top facing or left facing segments. If we eliminate either $a_m b_x$ or $a_n b_x$ from G_P , it does not create any impact on the minimum length cover. Therefore either $a_m b_x$ or $a_n b_x$ is redundant in G_P and we can eliminate any one of them.

By Observation 5.3.5, all the redundant vertices and edges are among the group which have multiple members. Elimination operations are also executed among the group which have more than one member. Each elimination operations among each group ensure that in that group there are no redundant vertices and edges. Therefore executing elimination operations in all groups with multiple members in graph G_P ensures that there are no redundant vertices and edges in graph G'_P. Q.E.D.

Following lemma follows from Lemma 5.3.6

Lemma 5.3.7 There is a bijection between rc-cut in P and edges in G'_P .

We next show the equivalence between modified mlsck problem in P and minimum weight vertex cover problem on G'_P

Lemma 5.3.8 The modified MLSCk problem on P reduces to minimum weight vertex cover problem on G'_P

Proof. Let graph G'_P is constructed from an orthogonal polygon P with k-transmitters having penetration capacity k. Let C_{G1} be a vertex cover of G', and let C_{P1} be a cover of P defined in terms of C_{G1} (mapping illustrated later). For each vertex v in G'_P let w(v) denote the weight of v and for each line segment $l \in C_{P1}$ let len(l) denote the length of l. We need to prove that C_{G1} is a minimum weight vertex cover for G'_P if and only if C_{P1} is an optimal cover of P for any value of k. For any value of k we show the following necessity and sufficiency statements respectively:

1 For any vertex cover C_G of G'_P, there exist a cover C_P of P such that $\sum_{l \in C_P} len(l) = \sum_{v \in C_G} w(v)$.

2 For any cover C_P of P, there exist a vertex cover C_G of G'_P such that

$$\sum_{v \in C_G} w(v) = \sum_{l \in C_P} len(l).$$

Necessity: Consider any vertex cover C_G of G'_P. We next find a cover C_P for P. For each edge $(v_{a_n}, v_{b_n}) \in E$, if $v_{a_n} \in C_G$ we locate a guarding line segment on the boundary of P that is aligned with a line segment $a_n \in B(P)$. Otherwise, we locate a guarding line segment on the boundary of P that is aligned with line segment $b_n \in B(P)$. Since either v_{a_n} or v_{b_n} is in C_G, by Lemma 5.3.7 we can say that every rc-cut is guarded by at least one line segment located on the boundary of P. Therefore C_P is a cover of P. Furthermore for each vertex in C_G we locate exactly one guarding line segment on the boundary of P whose length is same as the weight of the vertex. Therefore, $\sum_{l \in C_P} len(l) = \sum_{v \in C_G} w(v)$.

Sufficiency: Consider any cover C_P of P. We next construct a vertex cover C_G for G'_P From the cover C_P of P for any value of k. By Observation 5.3.1, let C' be a regular cover obtained from C. Moreover let M be the partition of C' into line segments induced by the rectangulation technique describe in Section 5.3.1. By Lemma 5.3.2 for any rc-cut there exist a set $C'_R \subseteq C$ such that all

$\mathbb{R}$ line segment in $\mathbb{C}'$

have same orientation
and collectively guard
that RC -cut.

Therefore M is also a cover of P. Let C_G be the subset of vertices of G_P such that $v_{a_n} \in C_G$ if and only if $a_n \in M$. Since M is a cover of G_P, by Lemma 5.3.7

we can write that C_G is a vertex cover of G_P. Furthermore it is observed that

$$\sum_{v \in C_G} w(v) = \sum_{l \in M} len(l) = \sum_{l \in C_P} len(l)$$ Q.E.D.

Regarding the computational complexity of the algorithm we can write the following lemma

Lemma 5.3.9 Algorithm for the modified MLSCk problem on orthogonal polygon P takes $O(n^2)$ time.

Proof. Following operations are required for an optimal solution of the modified MLSCk problem on orthogonal polygons

Algorithm MLSC

Input : An n vertex orthogonal polygon.
Output : Minimum length boundaries that cover the whole polygon.

1 Rectangulate the orthogonal polygon P using non monotone convex and reflex vertices.

2 Assign labels and weights to segments of P.

3 Create groups among segments by plane sweep algorithm.

4 Reduce P to a bipartite graph G_P for a fixed value of k.

5 Construct neighbor set of all vertices those are member of vertex group with multiple vertices.

6 Find out the all the neighbor sets those are complete subset to other neighbor set of the same group. Then eliminate the corresponding vertices and edges of those neighbor sets from graph G_P.

7 Find out the common members among neighbor sets of same groups by set intersection operation. Then eliminate one of the corresponding edge from G_P. After executing steps 6 and 7 graph G'_P is constructed from graph G_P.

8 Execute minimum weight vertex cover algorithm on G'_P.

Steps 1, 2, 3 and 5 require linear time complexity, Steps 6,7 and 8 require $O(n^2)$ time complexity. Step 4 require $O(n^2) + q$, where q depends on the value of k and number of rc-cut covered by the segments those are in the segment groups that have multiple members. q is always less than $O(n^2)$. Therefore, Step 4 also require $O(n^2)$ time complexity. Overall we can say that the algorithm requires $O(n^2)$ time complexity. Q.E.D.

Now from Lemmas 5.3.8 and 5.3.9 we can write the following theorem.

Theorem 5.3.10 Given an orthogonal polygon P with n vertices, there exists an algorithm that solves the modified MLSCk problem on P in $O(n^2)$ time.

5.4 Conclusion

In this chapter we have developed an algorithm for modified MLSCk problem. The algorithm finds the minimum length cover using sliding k-transmitters.

Chapter 6

Conclusion

In this thesis, we have considered the art gallery problem with sliding camera variant. Here the cameras travel back and forth along the polygon boundary and can see orthogonally inside the polygon. This problem asks to find out the minimum length of polygon's boundary where sliding cameras travel, such that those cameras can guard the whole polygon. In literature this problem is termed as minimum length sliding camera (MLSC) problem. There exist a $O(n^2)$ time algorithm which finds the minimum length boundary for orthogonal polygons. We have shown that one major step of existing algorithm has $O(n\log^2 n)$ time complexity for monotone orthogonal polygons and has $O(n)$ time complexity for FAT and MIN AREA grid n-ogons. In the existing algorithm that step takes $O(n^2)$ time. For the same problem we also have given an algorithm for semi- orthogonal polygon with $O(n^2)$ time complexity. The class of semi-orthogonal polygon is a superclass to the class of orthogonal polygon. While solving differ- ent art gallery problems with orthogonal polygons, input polygons are usually rectangulated by extending the incident edges of reflex vertices. In this thesis as a byproduct, we have established few relations among different components of orthogonal polygons after it is rectangulated by extending the incident edges of reflex vertices.

In Chapter 1, we have presented the historical background of the art gallery problem. Then we have described few art gallery problems with sliding camera along with some previous results. At the end of this chapter we have focused on the scopes of this thesis.

In Chapter 2, we have described some basic terminologies of polygons, polygon partitioning, graph theory, graph related problems and complexity of algrithms.

In Chapter 3, we have shown that one major step of existing algorithm of MLSC problem has $O(n\log^2 n)$ time complexity for monotone orthogonal polygons and has $O(n)$ time complexity for FAT & MIN AREA grid n-ogons. But in the existing algorithm that major step takes $O(n^2)$ time complexity for orthogonal polygons.

In Chapter 4, we have presented an algorithm on MLSC problem for semi-orthogonal polygons. The class of semi-orthogonal polygon is superclass to the class of orthogonal polygon. In this chapter we have also shown some relations among different components of orthogonal polygons after they are being rectangulated by a rectangulation technique.

Finally in Chapter 5 we have introduced a new problem based on the MLSCk problem. We have presented an algorithm for the new problem which has $O(n^2)$ time complexity.

Now we discuss some of the related open problems in this field.

- Scopes are open for working on modified MLSCk problem with omnidirectional k-transmitters.
- Scopes are open to work on modified MLSCk problem for semi-orthogonal polygons.
- Scopes are open to find out some other relations among the components of a rectangulated orthogonal polygon.

References

[1] A. P. Toma´s, A. L. Bajuelos and F. Marques. Partitioning orthogonal polygons by extension of all edges incident to reflex vertices: Lower and upper bounds on the number of pieces. In Computational Science and Its Applications - ICCSA 2004, International Conference, Assisi, Italy, May 14-17, 2004, Proceedings, Part III, pages 127–136, 2004.

[2] O. Aichholzer, R. F. Monroy, D. Flores-Pen˜aloza, T. Hackl, J. Urrutia, and B. Vogtenhuber. Modem illumination of monotone polygons. CoRR, abs/1503.05062, 2015.

[3] D. Avis and G. Toussaint. An efficient algorithm for decomposing a polygon into star-shaped polygons. Pattern Recognition, 13:395–398, 1981.

[4] P. Bose, M. Damian, E. D. Demaine, V. Dujmovi, R. Flatland, F. Hurtado, J. Iacono, A. Lubiw, P. Morin, V. Sacristn, D. Souvaine, B. Ballinger, N. Benbernou and R. Uehara. Coverage with k-transmitters in the presence of obstacles. In Weili Wu and Ovidiu Daescu, editors, Combinatorial Optimization and Applications, volume 6509 of Lecture Notes in Computer Science, pages 1–15. Springer Berlin Heidelberg, 2010.

[5] M. D. Berg, S. Durocher and S. Mehrabi. Guarding monotone art galleries with sliding cameras in linear time. In Combinatorial Optimization and Applications - 8th International Conference,COCOA 2014, Wailea, Maui, HI, USA, December 19-21, 2014, Proceedings, pages 113–125, 2014.

[6] V. Chvtal. A combinatorial theorem in plane geometry. Journal of Combinatorial Theory, Series B, 18(1):39 – 41, 1975.

[7] S. Durocher, O. Filtser, R. Fraser, A. D. Mehrabi, and S. Mehrabi. A (7/2)-approximation algorithm for guarding orthogonal art galleries with sliding

cameras. In LATIN 2014: Theoretical Informatics - 11th Latin American Symposium,Montevideo, Uruguay, March 31 - April 4, 2014. Proceedings, pages 294–305, 2014.

[8] S. Durocher and S. Mehrabi. Guarding orthogonal art galleries using sliding cameras: Algorithmic and hardness results. In Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013.Proceedings, pages 314–324, 2013.

[9] S. Durocher and S. Mehrabi. A 3-approximation algorithm for guarding orthogonal art galleries with sliding cameras. In Combinatorial Algorithms - 25th International Workshop, IWOCA 2014,Duluth, MN, USA, October 15-17, 2014,Revised Selected Papers, pages 140–152, 2014.

[10] C. David, G. Kay and D. Merle. Convexity and a certain propertypm. Israel Journal of Mathematics, 8(1):39–52, 1970.

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

[12] M. J. Katz and G. Morgenstern. Guarding orthogonal art galleries with sliding cameras. Int. J. Comput. Geom. Appl., 21(2):241–250, 2011.

[13] S. S. Mahdavi, S. Seddighin, and M. Ghodsi. Covering orthogonal polygons with sliding k-transmitters. In Proceedings of the 26th Canadian Conference on Computational Geometry,CCCG 2014, Halifax, Nova Scotia, Canada, 2014, 2014.

[14] J. O'Rourke. An alternate proof of the rectilinear art gallery theorem. Journal of Geomentry, pages 118–130, 1983.

[15] J. O'Rourke. Art Gallery Theorems and Algorithms. Oxford University Press, Inc., New York, NY, USA, 1987.

[16] C. G. Plaxton. Vertex-weighted matching in two-directional orthogonal ray graphs. In Algorithms and Computation, volume 8283 of Lecture Notes in Computer Science, pages 524–534. 2013.

[17] A. R. Vargas, R. Fabila-Monroy and J. Urrutia. On modem illumination problems. In XIII Encuentros de Geometria Computacional, 2009.

[18] D. Schuchardt and H. D. Hecker. Two np-hard art-gallery problems for ortho-polygons. Math. Log. Q., 41:261–267, 1995.

[19] G. Steiner and J. S. Yeomans. A linear time algorithm for maximum matchings in convex, bipartite graphs. Computers and Mathematics with Applications, 31(12):91–96, 1996.

[20] I. Nourbakhsh, T. W. Fong and K. Dautenhahn. A survey of socially interactive robots: Concepts, design, and applications. Technical Report CMU-RI-TR-02-29, Robotics Institute, Pittsburgh, PA, December 2002.

Index