March 2015

Reconstructing Geometric Structures from Combinatorial and Metric Information

Md Ashraful Alam
University of Massachusetts - Amherst

RECONSTRUCTING GEOMETRIC STRUCTURES FROM COMBINATORIAL AND METRIC INFORMATION

A Dissertation Presented

by

MD ASHRAFUL ALAM

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

February 2015

School of Computer Science

RECONSTRUCTING GEOMETRIC STRUCTURES FROM COMBINATORIAL AND METRIC INFORMATION

A Dissertation Presented

by

MD ASHRAFUL ALAM

Approved as to style and content by:

Ileana Streinu, Chair

David A. Mix Barrington, Member

Andrew McGregor, Member

J MacGregor Smith, Member

Lori A. Clarke, Chair
School of Computer Science

To my wife and my daughter

ACKNOWLEDGMENTS

First and foremost, I would like to express my heartfelt gratitude to my advisor, Professor Ileana Streinu for her continuous patience, guidance and encouragement during my graduate study and dissertation research. She has been kind and patient when I made mistakes and taught me how to learn from them. I have found a great mentor in her.

I would also like to thank my committee members - Professor David A. Mix Barrington, Professor Andrew McGregor and Professor J Macgregor Smith for their support and helpful comments. In addition, I would like to thank my former and current lab partners - Naomi Fox, Filip Jagodzinski and John Bowers for their help and support.

Last but not the least, I thank my family for their unconditional support. My parents have encouraged me all my life to pursue my dreams. My wife, Sharmin was the person who provided me with consistent motivation and sacrificed the most during my graduate student life. And finally, there is Sumaira, my daughter, who is a source of unlimited inspiration.

ABSTRACT

RECONSTRUCTING GEOMETRIC STRUCTURES FROM COMBINATORIAL AND METRIC INFORMATION

FEBRUARY 2015

MD ASHRAFUL ALAM

B.Sc., BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Ileana Streinu

In this dissertation, we address **three** reconstruction problems.

First, we address the problem of reconstructing a *Delaunay triangulation* from a *maximal planar graph*. A maximal planar graph G is *Delaunay realizable* if there exists a realization of G as a Delaunay triangulation on the plane. Several classes of graphs with particular graph-theoretic properties are known to be Delaunay realizable. One such class of graphs is *outerplanar graph*. In this dissertation, we present a new proof that an outerplanar graph is Delaunay realizable.

Given a convex polyhedron P and a point s on the surface (the source), the *ridge tree* or *cut locus* is a collection of points with multiple shortest paths from s on the surface of P. If we compute the shortest paths from s to all polyhedral vertices of P and cut the surface along these paths, we obtain a planar polygon called the *shortest*

path star (sp-star) unfolding. It is known that for any convex polyhedron and a source point, the ridge tree is contained in the sp-star unfolding polygon [8]. Given a combinatorial structure of a ridge tree, we show how to construct the ridge tree and the sp-star unfolding in which it lies. In this process, we address several problems concerning the existence of sp-star unfoldings on specified source point sets.

Finally, we introduce and study a new variant of the sp-star unfolding called *(geodesic) star unfolding*. In this unfolding, we cut the surface of the convex polyhedron along a set of non-crossing geodesics (not-necessarily the shortest). We study its properties and address its realization problem. Finally, we consider the following problem: given a geodesic star unfolding of some convex polyhedron and a source point, how can we derive the sp-star unfolding of the same polyhedron and the source point? We introduce a new algorithmic operation and perform experiments using that operation on a large number of geodesic star unfolding polygons. Experimental data provides strong evidence that the successive applications of this operation on geodesic star unfoldings will lead us to the sp-star unfolding.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

CHAPTER 1

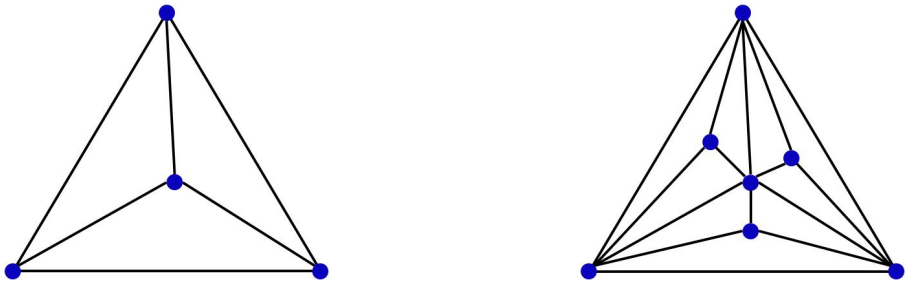
INTRODUCTION

In this dissertation, we focus on two well known structures in Computational Geometry - *Delaunay tessellation* and the *shortest path star unfolding* and introduce a new structure - *geodesic star unfolding*. We address **three** reconstruction problems related to these structures. In this chapter, we briefly discuss the concepts, problems and our contributions, which will occupy the rest of the dissertation.

Delaunay tessellation. A *Delaunay tessellation* of a set of points on the plane is a planar subdivision where the circumcircle of any face does not contain any other points inside it. If all faces, except possibly the outer face, are triangles, then it is called the *Delaunay triangulation*. Because of their unique structural properties, Delaunay triangulations have many applications, such as computing the minimum spanning tree [41], or the alpha hull of a point set [25], or generating a mesh on a polyhedral surface [9, 40, 26].

The problem of characterizing and reconstructing arbitrary Delaunay tessellations (in two- or higher dimensions) is an old problem. A closely related problem, going back to Steiner [29] asks for a characterization of the graphs of inscribable or circumscribable polyhedra[1]. The best result to date is due to Rivin[44], who proved necessary and sufficient conditions for a polyhedral graph to be of inscribable or circumscribable type. Dillencourt and Smith [21] linked inscribability of a graph to its

[1]Inscribable and circumscribable polyhedra are those whose vertices lie on a sphere or whose faces are tangent to a sphere, respectively



(a) (b)

Figure 1.1: (a) An example of a Delaunay realizable graph. (b) An example of a graph that is not Delaunay realizable.

realizability as a Delaunay tessellation and identified several classes of graphs that can be realized as Delaunay tessellations. See Fig 1.1.

An *outerplanar graph* is a graph with all vertices on the outer face. Dillencourt proved, constructively, that any maximal outerplanar graph can be realized as a Delaunay triangulation [19]. His proof used a simple and natural criterion on the angles of the triangles in a Delaunay triangulation, and the algorithm takes $O(n^2)$ time. The run time of this algorithm was later improved by Lambert [34]. However, none of these algorithms used the inscribability criteria set by Rivin [44]. Moreover, these algorithms fail when the outerplanar graph is not maximal, i.e, all faces are not cycles of length 3.

Contributions. We show that every outerplanar graph can be realized as a Delaunay tessellation. Our proof uses Rivin's [44] inscribability criterion and constructs an explicit witness of this inscribability, in the form of certain weights assigned to the edges of the graph. Our proof technique covers all outerplanar graphs, including those which are not maximal; by contrast, Dillencourt's algorithm works only for maximal outerplanar graphs.

Unfolding. An *unfolding* of a polyhedron P in R^3 is obtained by cutting the surface of P along a collection of non-crossing geodesic arcs that span the vertices of

(a) (b)

Figure 1.2: A classical example of an edge unfolding (right), obtained by cutting the cube (left) along a spanning tree of its 1-skeleton.

P and then immersing the surface into the 2D plane such that it is locally flat. This set of edges, known as *cut edges*, forms a tree (called a *cut tree*) whose leaves are the polyhedral vertices. The unfolding is locally flat and non-overlapping. Globally, the surface may self-overlap: its boundary is a planar polygon, which, in general, may not be simple.

Different types of unfoldings are distinguished by the choice of the cut tree. If the cut tree is a spanning tree of the 1-skeleton of the polyhedral surface, then the resulting unfolding is called *edge unfolding*. The uncut edges appear as the diagonals of the unfolding polygon. An example of an edge unfolding of a cube is shown in Fig 1.2.

Shortest path star unfolding. Another well known unfolding can be generated by fixing a point s (the *source vertex*) on the surface of the polyhedron P and cutting along the shortest paths from s to all vertices $v_1, v_2, \ldots, v_n$ of P. The resulting unfolding is called a *shortest-path star (sp-star) unfolding*. Since the cut tree is a star, and the resulting unfolding is non-overlapping, the boundary of the unfolding is known as an *sp-star unfolding polygon* [8, 47]. See Fig 1.3.

Along with its application in computing shortest paths in polyhedral surfaces [1, 16], the sp-star unfolding of a convex polyhedron P exhibits some interesting structural properties. For example, if P has n vertices, then its sp-star unfolding has

(a) (b)

Figure 1.3: (a) A tetrahedron and the shortest paths from a source vertex placed on a face. (b) The corresponding sp-star unfolding polygon, with source and polyhedral vertices colored, respectively, in red and blue.

$2n$ vertices when the source is not placed at one of the polyhedral vertices, and $2(n-1)$ otherwise. To streamline the presentation, we work under the assumption that the first case holds. Along the boundary of the unfolding, the n vertices corresponding to the polyhedral vertices appear in alternation with n vertices which are copies of the source vertex. The latter are known as *source images*. To fix the notation, we label the source images as $(s_i)_{i=1,..,n}$ and the polyhedral vertices as $(v_i)_{i=1,..,n}$. The sum of all the angles, interior to the surface, at the source images is exactly 2π under our assumption that the source is not placed at a vertex of the polyhedron. Each polyhedral vertex is placed on the perpendicular bisector of its adjacent two source images. A detailed exposition on sp-star unfolding can be found in [18].

Ridge tree. Given a convex polyhedron P and a source point s on P, a *ridge tree* is a set of points with multiple shortest paths from s on P. It is an embedded tree on the surface of P and the vertices of P are its leaves. It is known that the ridge tree is contained in the sp-star unfolding and is the Voronoi diagram of the source images [8]. Since ridge trees are combinatorial trees, it will be interesting to know whether all the combinatorial types of trees are represented as sp-star ridge trees. In other words, given a combinatorial tree, we would like to know if it can be realized as the ridge tree of some convex polyhedron and with respect to some source point.

Contributions. We address the following problems related to the sp-star unfolding and the ridge tree:

- We present the necessary and sufficient conditions for a polygon to arise as the sp-star unfolding of some convex polyhedron and a source vertex. In other words, we characterize the sp-star unfolding polygons. The necessary conditions are implicit in [8]. We show that those conditions are also sufficient and thus complete the characterization.

- Based on this characterization, we show that there may not exist an sp-star unfolding polygon on a given set of cyclically ordered points designated as source images. We present some examples and give some positive and negative results on the existence of the sp-star unfolding on specified point sets.

- Finally, we show how to reconstruct a ridge tree and its underlying sp-star unfolding from its combinatorial structure.

Geodesic star unfolding. We generalize the sp-star unfolding by cutting along a star-tree so that the cut edges are non-crossing geodesics (i.e. not necessarily the *shortest* geodesics) from the source vertex (Fig. 1.4). The result is a curvature zero disk-like surface with a polygonal boundary. We refer to this more general setting as a *geodesic star unfolding* and use the term *geodesic star unfolding polygon* to refer to its polygonal boundary.

Geodesic star unfoldings share all properties of sp-star unfoldings as stated above. However, it has two basic differences with an sp-star unfolding. First, all of its boundary edges are not the shortest paths on the corresponding polyhedral surface, where as the boundary edges of the sp-star unfolding are, by design, the shortest paths on the polyhedral surface. Second, the source angles of the sp-star unfolding are all acute angles but this is not necessarily true for geodesic star unfoldings. Although,

(a) (b)

Figure 1.4: (a) One of the shortest paths on the tetrahedron from Fig. 1.3(a) is replaced by a geodesic. (b) The corresponding geodesic star unfolding polygon.

it is possible that all of the source angles of a geodesic star unfolding may be acute, there are cases when one angle is larger than π. See, for example, Fig 1.4(b).

Because of the close relationship between the geodesic and sp-star unfolding, it is natural to ask the following question: given a geodesic star unfolding of a convex polyhedron with respect to a source vertex, can we convert it into the sp-star unfolding of the same polyhedron and the source vertex? A positive answer will give us an alternative way to compute the shortest paths on polyhedral surfaces without using the complex traditional algorithms. To the best of our knowledge, there are no existing results on this problem.

Embedding the surface of an arbitrary polyhedron, particularly convex polyhedron, on the plane is sometimes necessary to compute some geometric objects like shortest paths, triangular mesh etc. One way to do this is to compute an arbitrary convex polyhedron in R^3, cut the surface according to certain rules and unfold it onto the plane. All of these operations are complex and computationally intensive. Therefore, it is desirable to have an algorithm that will directly generate the polyhedral surface as a planar polygon. However, any arbitrary polygon does not represent a convex polyhedral surface - Alexandrov [6] presented the characterizations of such polygons. One of the main focuses of this dissertation is to present algorithms to generate convex polyhedral surfaces in the forms of geodesic and sp-star unfoldings without using unfolding operations.

Contributions. We address the following problems related to the geodesic star unfolding:

- We study the properties of the geodesic star unfolding of a convex polyhedron. We compare these properties with those of the sp-star unfolding. We discover several contrasting properties which are specific to the geodesic star unfolding.

- We consider the problem of constructing the geodesic star unfolding from a set of cyclically ordered planar points. We discuss the challenges of this problem and present algorithms to generate the geodesic star unfolding from a special class of points.

- We address the problem of deriving an sp-star unfolding from the geodesic star unfolding of the same convex polyhedron and the source vertex. We use a new type of algorithmic operation, called *cut-paste* operation on geodesic star unfolding polygons. Under this operation, we select and cut along a diagonal from a polyhedral vertex to a source image and paste along the current cut edges emanating from that polyhedral vertex. The result is a new geodesic star unfolding (possibly sp-star) of the same convex polyhedron and the source vertex. We have implemented this operation in Mathematica and performed experiments on a large number of geodesic star unfolding polygons. Experimental data provides strong evidence that successive applications of this operation on geodesic unfolding polygons always lead us to the sp-star unfolding. However, at the time this dissertation has been completed, only elements of the proof have been completed.

1.1 Outline

The dissertation is organized as follows: the results on the *Delaunay realization* problem are presented in Chapter 2. Chapter 3 addresses the ridge tree reconstruction

problem and presents the characterization of sp-star unfolding polygons. Chapter 4 introduces the geodesic star unfolding and presents results, theoretical and experimental, on deriving the sp-star unfolding from a geodesic star unfolding. Finally, Chapter 5 summarizes the results with remaining open problems.

CHAPTER 2

DELAUNAY REALIZATION OF OUTERPLANAR GRAPHS

2.1 Introduction

A *triangulation* of set of planar points P is a planar subdivision where all faces, except possibly the outerface, are triangles. If the circumcircle of any triangle does not contain any other points inside it, it is called a *Delaunay triangulation*. Delaunay triangulations possess some nice structural properties. For example, the minimum angle among the angles of the triangles in a Delaunay triangulation is larger than that of any possible triangulation of the same set of points. Due to these structural properties, it is widely used in generating triangular meshes on 3D surfaces for surface reconstruction [11, 23, 22, 28]. Besides, it is also used to compute several other geometric objects like the Euclidean minimum spanning tree [41], the alpha hull [25], Gabriel graphs [27] and relative neighborhood graphs [48].

Because of their importance in different fields, characterizing arbitrary Delaunay triangulations has received significant attention. Although the problem is substantially difficult, some progresss has been made. For example, Dillencourt [20] has shown that all Delaunay triangulations are 1-tough[1] and have perfect matching. A more closely related problem, due to Steiner [30], asks to characterize the graph of inscribable and circumscribable polyhedra. The most prominent result is due to Rivin [44, 45] who presented a characterization of polyhedral graphs of inscribable and circumscribable type. Dillencourt and Smith [21] linked combinatorial characteristics

[1] A graph is 1-tough if removing k vertices splits the graph into at most k components.

of inscribable graphs to their realizability as Delaunay triangulations. Using this technique, they have identified a set of graphs which are Delaunay realizable. For example, they have proved that all 4-connected maximal planar graphs are Delaunay realizable. Recently, Ziegler et al [50] linked the inscribability of a set of polyhedral graphs, called 1-skeletons of stacked polytopes, to Delaunay realizability.

The Delaunay realizability of an outerplanar graph was proved almost 25 years ago by Dillencourt [19]. He used a simple and constructive proof using the natural criterion on the angles of triangles in a Delaunay triangulation. The algorithm incrementally calculates the angles of each triangle and takes $O(n^2)$ time. Later, the run time of the algorithm was improved by Lambert [34]. To the best of our knowledge, no proofs or algorithms exploited the inscribability criterion to show that the outerplanar graph is Delaunay realizable.

2.1.1 Results

The main result in this chapter is a new proof of the following theorem:

Theorem 1 *Any outerplanar graph can be realized as a Delaunay tessellation.*

Our proof links the Delaunay realizability of the outerplanar graph with the inscribability criterion set by Rivin [44]. We show, using a constructive proof, that there exists a set of weights on the edges of any outerplanar graph that satisfies the inscribability criterion.

2.2 Preliminaries

Given a graph $G = (V, E)$, two paths between two vertices are *independent* if they do not share other vertices besides the end-points. A graph is connected if there is a path between any two vertices and it is *k-connected* if there are k independent paths between any two vertices. All outerplanar graphs are 2-connected.

A graph is *planar* if it can be drawn in the plane (or, equivalently, on the sphere) with no crossings of endpoint-disjoint edges. A drawing of a planar graph on the sphere, called a *spherical graph*, subdivides the sphere into regions called faces. For 2-connected spherical graph drawings, the faces are topological disks. In the plane, exactly one face, called the *outer face*, is unbounded. A spherical graph is specified by its sets of vertices, edges and faces: $G = (V, E, F)$. A *plane graph* is obtained from a spherical graph by specifying a face f as the outer face: $G = (V, E, F, f)$.

The stellation G_s of a plane graph $G = (V, E, F, f)$ is the graph obtained by adding one new vertex (the *stellating* vertex s) and connecting it to all the vertices of f through edges called *stellating* edges. Stellation does not violate the planarity property: a stellated planar graph remains planar, and a plane realization of it is obtained by placing the stellating vertex inside the face f.

The dual $G^* = (V^*, E^*, F^*)$ of a spherical graph $G = (V, E, F)$ is obtained by switching the roles of vertices and faces: $V^* = F$, $E^* = E$, $F^* = V$.

A plane graph where *all* vertices lie on the outer face is called an *outerplanar graph*. In a *maximal outerplanar graph*, all faces are triangular except the outer face. A *wheel graph* is obtained by stellating a cycle.

A *cutset* of a graph is the minimal set of edges whose removal disconnects the graph. A cutset is *coterminous* if all the edges emanate from a single vertex and *noncoterminous* if its edges do not have a common endpoint. In the dual graph G^* of a spherical graph G, a coterminous cut set of G becomes the set of edges of a face in G^*; a noncoterminous cut set of G becomes a non-facial cycle of G^* (a cycle which is not a face).

A graph is *polyhedral* if it is planar and 3-connected. In this case, the faces of a spherical realization are uniquely determined. Any polyhedral graph can be realized as the 1-skeleton of a convex polyhedron in dimension 3 (Steinitz theorem,

see Grünbaum [30]). A polyhedral graph is *inscribable* if it can be realized as the 1-skeleton of an convex polyhedron inscribed in a sphere.

Given a set P of points in the Euclidean plane, a *triangulation* of P is a plane graph where all faces, with the possible exception of the outer face, are triangles. The *Delaunay triangulation* of a point set P is a triangulation of P with the property that the circumcircle of any face contains no other point of P inside. A *Delaunay tessellation* is a geometric planar graph which has the circumcircle property for each of its faces but not all faces need to be triangles.

A planar graph $G = (V, E, F)$ is *Delaunay realizable* if there exists an embedding of the graph on the plane as Delaunay tessellation.

2.3 Delaunay realizability of outerplanar graph

In this section, we prove Theorem 1. For our convenience, we first reformulate the inscribability criterion with respect to outerplanar graphs.

2.3.1 Inscribability criterion for outerplanar graphs

Igor Rivin [44, 45] gave a complete characterization of the graphs which are of inscribable type :

Theorem 2 *[Rivin] A planar graph $G = (V, E)$ is of inscribable type if and only if:*

1. *G is 3-connected, and*
2. *There exists an assignment $w : E \to \mathbb{R}$ of weights $w(e)$ to the edges $e \in E$ such that :*
   (a) ***Edge condition*** *For each edge e, $0 < w(e) \leq 1/2$.*
   (b) ***Vertex condition*** *For each vertex v, the sum of all weights of edges incident to v is 1.*

(c) ***Cutset condition*** *For each non-coterminous cutset $C \subseteq E$, the sum of all the weights of edges of C must exceed 1.*

Theorem 2 can be found in [21] (page 65), as a reformulation in the Euclidean space of the general, hyperbolic space result of Rivin et al [33] (page 247).

We also have the following lemma from [21]:

Lemma 3 *A plane graph G = (V,E,F,f) is realizable as a Delaunay Tessellation if and only if its stellation G_s is of inscribable type.*

Combining Theorem 2 with this lemma, we have to prove that the stellation G_s of any outerplanar graph G is 3-connected and has the weight assignment properties from Theorem 2. We prove 3-connectivity first.

Lemma 4 *Any outerplanar graph is 2-connected.*

Proof In an outerplanar graph, all the vertices lie on the unbounded face f. If we label the vertices as $1, 2, \cdots, n$ in the order in which they appear on the outer face, there are two independent paths between any pair of vertices i and j: one from $i, i+1, \cdots, j$ and another is $i, i-1, \cdots, j$. ∎

Lemma 5 *The stellation of G_s of an outerplanar graph G = (V,E,F,f) is 3-connected.*

Proof By definition, the stellation of a planar graph is also a planar graph. We must show that there exist three independent paths between any two vertices i and j of G_s. If i and j are the vertices of the outerplanar graph, Lemma 4 gives two independent paths between i and j. A third independent path is (i, s, j) where s is the stellating vertex. If $i \neq s$ and $j = s$, then we obtain three independent paths $(s, i), (s, i-1, i)$ and $(s, i+1, i)$ (index arithmetic is done modulo n in the range 1,. . . ,n) ∎

For the convenience, we restate the weight assignment conditions in terms of the dual graph $G^* = (V^*, E^*, F^*)$.

Theorem 6 ***[Dual formulation]*** *A 3-connected planar graph $G = (V, E)$ is of inscribable type if and only if its dual G^* has an assignment $w : E \to \mathbb{R}$ of weights on its edges $e \in E^* = E$ such that*

1. ***Edge condition*** *For each edge $e \in E^*, 0 < w(e) \leq 1/2$.*
2. ***Face condition*** *For each face $f \in F^*$, the sum of its edge wights is 1.*
3. ***Cycle condition*** *For each non-facial cycle $C \subseteq E^*$, the sum of its edge weights is at least 1.*

With this formulation, and using Lemma 3, the proof of Theorem 1 is reduced to proving:

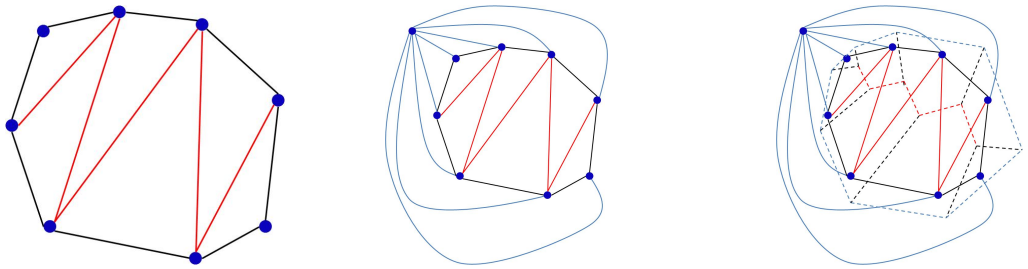
Theorem 7 *Let G_s be the stellation of an outerplanar graph. Then there exists a weight assignment on the edges of its dual graph G_s^* such that the three edge, face and cycle conditions of Theorem 6 are satisfied.*

In the next two sections, we show the existence of a weight assignment satisfying the face, edge and cycle conditions as required in Theorem 7.

2.3.2 Weight assignment

We start by having a closer look at the structure of the dual of a stellated outerplanar graph G_s^*. If we remove the cycle C made of the duals of the stellating edges of G_s, what remains is a tree, whose leaves lie on the cycle C. Removing the leaf edges of the tree, what remains is a smaller tree called the *backbone*; this is actually the dual of the outerplanar graph without the outerface. We thus partition the edges of G_s^* into three classes; cycle, backbone and leaf edges. See Fig 2.1.

Edge contraction and expansion: The graph obtained by the *contraction* of an edge ij has the two vertices i and j merged into one new vertex v and the edges



(a) (b) (c)

Figure 2.1: (a) A maximal outerplanar graph, with outer face cycle (black edges) and interior edges (red). (b) Its stellation. Blue edges are the stellating edges. (c) The dual graph of the stellated outerplanar graph. The dual edges are dashed. Red, black and blue edges are backbone, leaf and cycle edges, respectively.

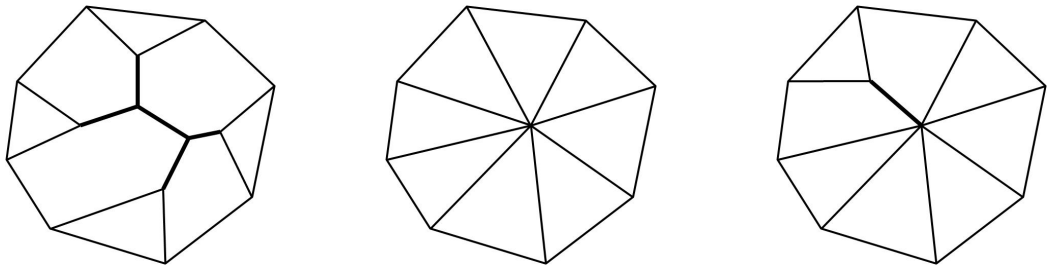
incident to either i or j become incident to v. The opposite operation is called *edge expansion*.

A graph G has an *edge-expansion inductive construction*, starting from a base graph G_0 if there exists a sequence of graphs $G_0, G_1, \ldots, G_k$ such that G_{i+1} is obtained from G_i by an edge expansion and $G_k = G$.

Lemma 8 *The dual of a stellated outerplanar graph G_s^* has an edge-expansive inductive construction starting from a wheel graph.*

Proof We perform the contraction and expansion operations on G_s^*. A contraction is applied on a backbone edge, one at a time, in an arbitrary order. When all backbone edges are contracted, we obtain a wheel graph where boundary edges are cycle edges and remaining edges are leaf edges of G_s^*. See Fig 2.2. This sequence of contraction, taken in reverse, gives an edge expansion inductive construction for G_s^*. ∎

Next we show that we can assign weights on the edges of G_s^* to meet the condition 2 (the face condition) of Theorem 6.



(a) (b) (c)

Figure 2.2: (a) The dual of a stellated outerplanar graph. Thick edges are backbone edges. (b) Dual graph after contraction of all backbone edges. (c) Expansion of a single backbone edge.

2.3.3 The face condition

We show that there exists a weight assignment scheme which will satisfy the face condition of Theorem 6. In the following lemma, we show how we can assign those weights.

Lemma 9 *There exists a weight assignment on the dual of a stellated outerplanar graph G_s^* such that the sum of the edge weights of each face is 1.*

Proof We assign weights on G_s^* in an inductive fashion, based on an edge expansion sequence $G_0, \ldots, G_{n-3}$ for G_s^*.

Base case: G_0 is the wheel graph. We assign $\frac{1}{n}$ to the cycle edge and $\frac{n-1}{2n}$ to each leaf edge. The sum of weights on the edges of each face is 1. See Fig 2.2(b).

Inductive step: Let $f_1, \ldots, f_n$ be the faces dual to the n vertices of G, labeled in counter clockwise order. Assuming that we have completed a weight assignment for G_k, let G_{k+1} be obtained from G_k by the expansion of edge ij between faces f_i and f_j. We assign a weight ϵ, (for a value of $0 < \epsilon \leq 1/2$ that will be determined later) on the edge ij. As a result, the sum of the weights on faces f_i or f_j is imbalanced. We remove the imbalance by subtracting $\frac{\epsilon}{4}$ from each of the two leaf edges of f_i and f_j, respectively and by also subtracting $\frac{\epsilon}{2}$ from the cycle edges of these faces. Although

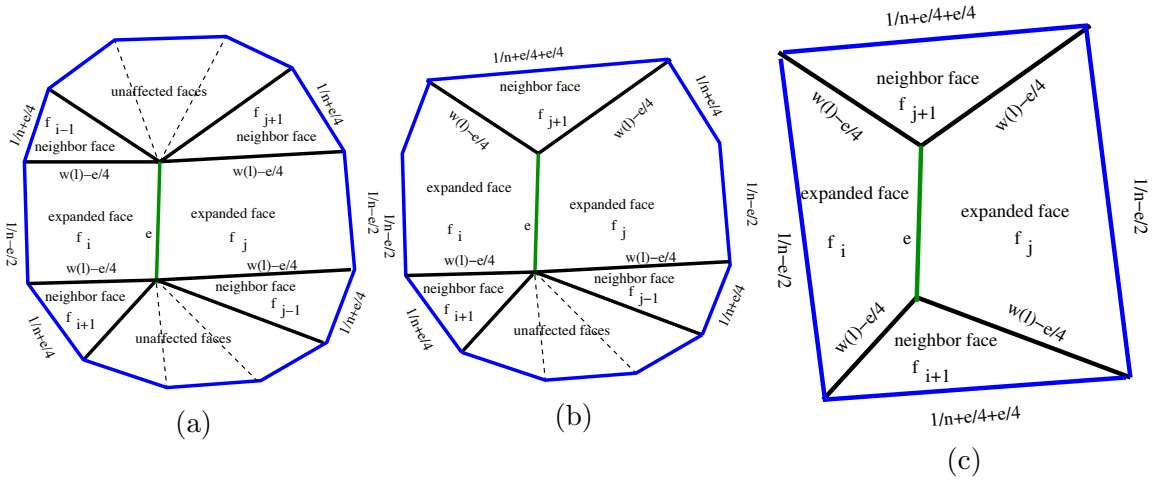


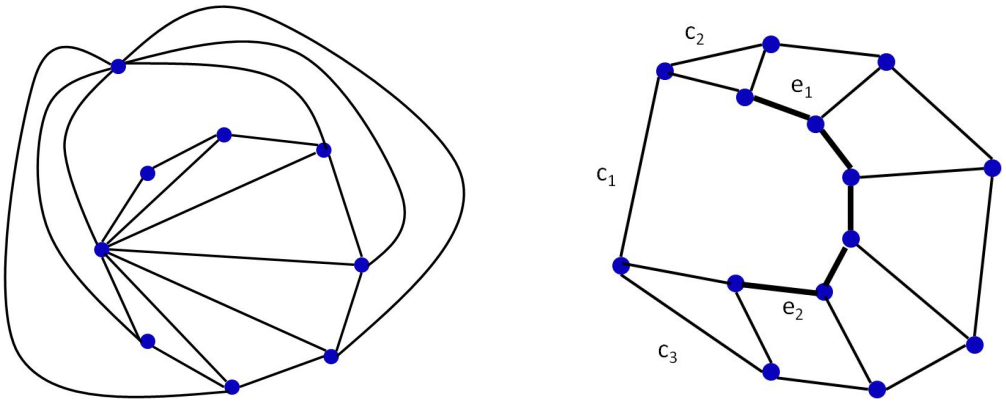
Figure 2.3: Three possible cases when a backbone edge is expanded, leading to the expansion of two faces, which have, as neighbors, (a) four distinct faces, (b) two distinct faces and one common face and (c) two common faces. When a backbone edge is expanded, only the weights of these neighbor faces have to be adjusted; all the other faces remain unchanged.

it restores the balance of weights in face f_i and f_j, it destroys the balances to the faces adjacent to f_i and f_j, namely $f_{i-1}, f_{i+1}, f_{j-1}, f_{j+1}$ and the cycle edges. To fully balance the weights, we add $\frac{\epsilon}{4}$ to the cycle edges of these four faces. See Fig 2.3. Now the sum of the weights of the edges of each face is 1. This weight assignment obviously satisfies the face condition. ∎

2.3.4 Bounds on weights

In previous section, we showed how to assign weights on the edges of G_s^*. Based on this weight assignment, we now try to calculate the maximum and minimum possible weights on the edges of G_s^* in terms of ϵ. These bounds will help us when we will calculate the valid range of values for ϵ.

Lemma 10 *Over all the stellated outerplanar graphs, the maximum possible weight is* $\frac{1}{n} + \frac{(n-2)\epsilon}{4}$*, and the minimum is* $\frac{1}{n} - \frac{(n-3)\epsilon}{2}$.



(a) (b)

Figure 2.4: A stellated outerplanar graph where all diagonals emanate from a single vertex. (b) Its dual. Backbone edges are thickened. The cycle edge whose weight is decremented by $\epsilon/2$ at each backbone edge expansion step is labeled by c_1. The cycle edges whose weights are increased by $\epsilon/4$ for each expansion of the backbone edge are c_2 and c_3. A special case occurs when e_1 and e_2 are expanded: the weights of c_2 and c_3 are then increased by $\epsilon/2$.

Proof As described in the inductive step of the proof for Lemma 9, when a backbone edge e incident to faces f_i and f_j is expanded, the weights of the two cycle edges of these faces are decremented by $\frac{\epsilon}{2}$. Similarly, the weights of the cycle edges of all adjacent faces of these two faces are increased by $\frac{\epsilon}{4}$. However, if the expanded backbone edge e is a leaf of the backbone tree, then $f_{i-1} = f_{j+1}$ (and/or $f_{i+1} = f_{j-1}$), as illustrated in Fig 2.3(b) and 2.3(c). In this case, the weight of the cycle edge of the face f_{j+1} and/or f_{i+1} is increased twice, each time by $\frac{\epsilon}{4}$.

The minimum, resp. maximum weight of an edge is attained when its weight is reduced by $\frac{\epsilon}{2}$, resp, increased by $\frac{\epsilon}{4}$, at each backbone edge expansion step. This happens when the original outerplanar graph G has all diagonals emanating from a single vertex, or, equivalently, when the dual graph G^* has one face incident to all backbone edges. See Fig 2.4. Consider a face f in G_s^* corresponding to such a vertex in G.

Each time a non-leaf edge of the backbone tree is expanded, the weight of each of the two cycle edges on the two faces adjacent to f is increased by $\frac{\epsilon}{4}$. However, if the expanded edge is a leaf edge of the backbone tree, then one of the cycle edges' weight is increased twice, each time by$\frac{\epsilon}{4}$. See Fig 2.4(b). Thus the weight of each such edge is increased by at most $\frac{\epsilon(k+1)}{4}$, where k is the number of backbone edges in G_s^* or, equivalently, diagonals in G. Since k is at most $(n-3)$ ($(n-3)$ when G is maximal, otherwise less than $(n-3)$), we obtain the maximum weight on any edge of G_s^* as being $\frac{1}{n} + \frac{\epsilon(n-2)}{4}$ ($\frac{1}{n}$ is the initial weight on cycle edges).

Similarly, each time a backbone edge is expanded, the weight of the cycle edge of f is reduced by $\frac{\epsilon}{2}$. When all backbone edges are expanded, the weight of this edge is at least $\frac{1}{n} - \frac{\epsilon(n-3)}{2}$. This gives the minimum possible weight on any edge of G_s^*. ∎

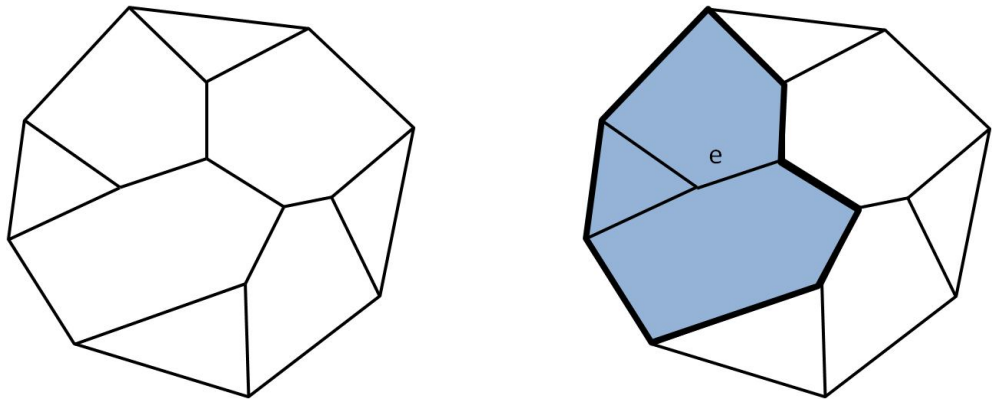
2.3.5 The edge and cycle conditions

To conclude the proof of Theorem 1, we show now how to choose ϵ such that conditions 1 (edge) and 3 (cycle) of Theorem 6 are also satisfied. Obviously, ϵ has to be strictly greater than 0 to be a valid weight on backbone edges. Now we establish an upper bound of ϵ.

Lemma 11 *If* $0 < \epsilon < \frac{2}{n(n-3)}$, *then* $0 < w(e) \leq 1/2$ *for any edge of* e *of* G_s^*.

Proof We find an upper bound on ϵ from the constraint that the weight on any edge should be between 0 and 1/2. Bounding from the below the minimum possible weight on an edge of G_s^* (Lemma 10) and solving $\frac{1}{n} - \frac{(n-3)\epsilon}{2} > 0$, we obtain $\epsilon < \frac{2}{n(n-3)}$. Similarly, for the maximum, bounded from the above by 1/2: solving $\frac{1}{n} + \frac{(n-2)\epsilon}{4} \leq 1/2$ results in $\epsilon \leq \frac{2}{n}$. The final resulting bounds are $0 < \epsilon < min\{\frac{2}{n(n-3)}, \frac{2}{n}\}$ or $0 < \epsilon < \frac{2}{n(n-3)}$. ∎

Lemma 12 *If* $0 < \epsilon < \frac{4(n^2-4n+9)}{3n(n-2)(n-3)}$, *then the sum of the weights on the edges of a non-facial cycle of* G_s^* *is stricly greater than 1.*



(a) (b)

Figure 2.5: (a) A dual G_s^* of a stellate outerplanar graph. (b) A non-facial cycle C of G_s^*, shown in thick lines, divides the plane into two regions. The shaded region, containing e as an internal edge, has the smaller number of faces.

Proof Any non-facial cycle C of G_s^* divides the plane into two regions, each one containing exactly one of the $n+1$ faces of G_s^*. Let R_i be the region containing the smallest number $k \leq \frac{n+1}{2}$ of faces. An edge is called *internal* to a region if it lies inside it, i.e. not in the complementary region and not on the boundary cycle. See Fig 2.5. Since the sum of the weights on the cycle C is k (sum of the weights on the internal k faces) minus twice the sum of the internal edges of R_i (because these edges are shared by two faces in R_i). Since k faces are dual of k vertices in the stellated outerplanar graph G_s and the subgraph induced by k vertices are planar, there are at most $3k-6$ internal edges of R_i. Since $k \leq \frac{n+1}{2}$, there are at most $\frac{3(n+1)}{2} - 6$ or $\frac{3n-9}{2}$ internal edges of R_i. This bounds the sum of the weights on C by at most $\frac{n+1}{2}$ minus twice the sum of the weights on $\frac{3n-9}{2}$ internal edges. Let w_{max} be the maximum possible weight on any of these internal edges. Then it suffices to show that $\frac{n+1}{2} - \frac{2w_{max}(3n-9)}{2} > 1$. Rearranging the terms, we get $w_{max} < \frac{n-1}{6(n-3)}$.

The maximum possible weight on any edge of G_s^* is $\frac{1}{n} + \frac{(n-2)\epsilon}{4}$ (Lemma 10. Thus $\frac{1}{n} + \frac{(n-2)\epsilon}{4} < \frac{n-1}{6(n-3}$. Solving this equation, we get $\epsilon < \frac{2(n^2-7n+18)}{3n(n-2)(n-3)}$. Therefore, the cycle conditions is satisfied if $\epsilon < \frac{2(n^2-7n+18)}{3n(n-2)(n-3)}$. ∎

To satisfy both the edge and cycle conditions, we will choose an ϵ such that $0 < \epsilon < min\{\frac{2}{n(n-3)}, \frac{2(n^2-7n+18}{3n(n-2)(n-3)}\}$. This completes the proof of Theorem 1.

2.4 Conclusion

We have presented a new proof of the Delaunay realizability of the outerplanar graph. We have used the inscribability criterion, set by Rivin [44] using weights on the edges of the graph. We have proved that such a weight assignment exists and shown how to compute those weights.

CHAPTER 3

SHORTEST PATH STAR UNFOLDING

3.1 Introduction

The shortest paths between two points on the polyhedral surface is the path with the smallest length measured on the surface. Finding shortest paths between two points on a surface is a well studied problem because of its wide range of applications in the field of robotics, geographic information systems and computer graphics [5, 43, 38]. In two dimensions, the problem has been thoroughly studied and a number of efficient algorithms are known [49, 36, 47]. However, the problem becomes significantly complex in three dimensions. Canny and Reif have shown that it is an NP-hard problem [14] and only exponential time algorithms are known in such cases [42, 13].

In certain special cases, however, the problem of computing shortest paths in three dimensions becomes easier. One such case is computing shortest paths between two points on the surface of a convex polyhedron P. Sharir and Schorr [47] proposed an algorithm that computes the exact shortest path between two points on the surface of P. The proposed algorithm is mainly based on three observations. First, any shortest path intersecting an edge e of P enters and leaves e under the same angle. Second, no shortest path on a convex polyhedron P can pass through a vertex p of P unless p is the destination or source of the shortest path. Third, if the sequence of edges of P intersected by the shortest path between s and p is known, the shortest path can be computed as the straight line joining s and p after unfolding the faces, adjacent to the edge sequence, on the plane. Their algorithm runs in $O(n^3 logn)$. An improved

algorithm for a convex polyhedron was given by Mount [39, 37] which reduces the running time to $O(n^2 logn)$. Chen and Han further improved the running time to $O(n^2)$ [15]. Recently, Schreiber and Sharir [46] proposed an algorithm which runs in $O(nlogn)$ time and takes $O(nlogn)$ space. Since all these algorithms to compute the exact shortest paths are quite computationally intensive and complex, interest has been growing in developing approximation algorithms to compute the shortest paths. For a given $\epsilon > 0$, a path between two points is $(1+\epsilon)$-approximate if its length is at most $(1+\epsilon)$ times the actual shortest paths. A lot of work has been going on in this direction [3, 4, 31, 32].

An unfolding of a polyhedron P is an operation where we cut its surface and immerse the bounded surface on the plane. Based on how we choose to cut the surface, there are different types of unfoldings. One such unfolding is *edge unfolding*, in which, the surface is cut along the polyhedral edges spanning its vertices. Polyhedral edges are, in particular, shortest paths between the vertices. Since the cut edges are non-crossing, the unfolding yields a planar polygon with an interior.

The edge unfolding of a polyhedron is a well-studied problem. A famous problem on edge unfolding, named in the honor of Albrecht Dürer [24], is the following: does every convex polyhedron have an edge unfolding to a simple, non-overlapping polygon? The problem appears to be very difficult and is still open to date. As related problems, researchers study different combinatorial and structural properties of the cut tree to see if they yield non-overlapping unfoldings. For non-convex polytopes, it is shown in [10] that there exist certain types of non-convex polyhedra which do not unfold to simple non-overlapping polygons.

The two well known variants of the unfolding of a polyhedron are source unfolding and shortest-path star unfolding. In source unfolding, the surface of the polyhedron is cut along the ridge tree with respect to a given point on its surface. The source unfolding of a convex polyhedron with respect to a source vertex always yields a non-

overlapping simple polygon [18]. It has been used in computing shortest paths on polyhedral surfaces [47]. The shortest path star unfolding, where the cut edges are the shortest paths from the given source vertex to all polyhedral vertices, was first mentioned in [7] and is also known as *Alexandrov unfolding* [35]. The non-overlapping property of the shortest path star unfolding is studied in [8]. A number of applications of this unfolding are presented in [1, 16].

An interesting direction of research leads us to the problem of reconstructing polygons that arise as the unfolding of polytopes, particularly convex polyhedra. This will enable us to generate polyhedral surfaces without actually realizing the polyhedra. Alexandrov formulated a characterization of polygons that can be folded into convex polyhedra [6]. However, this characterization does not give us methods to generate such polygons algorithmically. Even, if we know that a polygon can be folded into a polyhedron, the actual folding process is still difficult. The difficulty arises from the fact that the edges along which a polygon will fold into a polyhedron are not, usually, given. Finding those edges or creases is hard. Recently, Bobenko gave a constructive proof of the Alexandrov's theorem, which answers some of these questions related to crease pattern [12].

3.2 Preliminaries

A *polygon* in two dimensions is the region bounded by a simple cycle formed by finite number of line segments called *edges*. A polygon is *convex* if a line segment connecting any of its two points lies fully inside its region, otherwise, it is non-convex. A *polyhedron* is the region bounded by a finite number of polygons called *faces* such that i) if two faces intersect, then it is only at a common edge or a vertex, ii) every edge of every face is an edge of exactly one other face, and iii) faces surrounding each vertex form a simple cycle [17]. A polyhedron is *convex* if a line segment connecting any of its two points lies entirely in its interior. In other words, a convex polygon

is a region bounded by finite number of half planes and a convex polyhedron is the bounded intersection of a finite number of half-spaces.

A *shortest* path between two points on P is the smallest of all possible curves between these two points, measured on P. Different properties of the shortest paths in polyhedral surfaces can be found in [47, 39].

The *unfolding* of a convex polyhedron P is obtained by cutting its surface along the edges of a tree where its leaves are the vertices of P. The *shortest path star unfolding* or *sp-star unfolding* in short, is obtained by cutting its surface along the shortest paths from a given point to all vertices of P. This point is called *source vertex, s.* The sp-star unfolding has $2n$ vertices – n polyhedral vertices of P and n copies of source vertex, known as *source images.* The source images and the polyhedral vertices are placed alternately, meaning each source image has one polyhedral vertex on its right and one on its left. The angle at each source image is called *source angle*. The sum of the source angles of an sp- star unfolding is 2π when the source vertex is on a polyhedral face or an edge and less than 2π when it is placed on a polyhedral vertex.

Given a set of n points P_n on the plane, the *Voronoi* diagram V_D of P_n is the planar subdivision composed of n regions, one for each point such that any point x lies in the region corresponding to a point y in P_n if the distance of x from y is smaller than its distance from any other point in P_n. The points of P_n are called the *Voronoi sites*, or simply *sites*, and their corresponding planar regions are called the *Voronoi regions* of V_D. Voronoi regions meet at edges and vertices, called *Voronoi edges* and *Voronoi vertices* respectively. There are two kinds of Voronoi edges in any Voronoi diagram – one is bounded at both ends by two Voronoi vertices and the other is bounded by a Voronoi vertex at one end but the other end is unbounded or infinite. We call these edges *bounded Voronoi edges* and *infinite Voronoi edges*, respectively. We call the bounded end of an infinite Voronoi edge the *close end* and the unbounded end the *open end*.

3.2.1 Su-Polygon

By definition, an *su-polygon* (short for *abstract star-unfolding polygon*[1]) is the boundary of a flat disk-like polygon in the plane, satisfying the abstract version of the following properties: (a) it has $2n$ vertices, alternately labeled as s_i and v_i and referred to as s-, resp. v-vertices. The angle (interior to the surface) at an s-vertex is refered to as an s-angle; (b) the sum of all the s-angles is 2π, and (c) the v-vertices are placed on the perpendicular bisectors of the two neighboring s-vertices. In particular, this last property implies that the two edges incident to a v-vertex have equal lengths. Intuitively, the s-vertices, v-vertices and s-angles are analogous to source images, polyhedral vertices and source angles for the sp-star unfolding polygons.

Since the boundary edges of an sp-star-unfolding polygon must be the shortest paths on the corresponding polyhedral surface, not all su-polygons arise as sp-star-unfoldings. We remark that the definition allows for the self-overlap of the unfolded surface, i.e. the su-polygon is not necessarily a simple polygon.

3.2.2 Polyhedral metric

A *convex polyhedral metric* is a presentation of the surface of a convex polyhedron as a collection of (one or more) planar polygonal pieces together with rules for glueing them, along pieces of their boundaries, into a surface that is topologically a sphere. The glueing may result in a finite set of points of non-zero Gaussian curvature, called the vertices of the surface. The metric is convex if the sum of the surface angles at each vertex is at most 2π, i.e. if the curvature is positive. Alexandrov's Theorem [6] guarantees that any convex polyhedral metric has an isometric realization as a convex polyhedron. As a side note, we remark that the edges of the polygonal pieces need not be related in any way to the edges of the convex polyhedral realization.

[1]The modifier *"abstract"* is used to emphasize that, *a priori*, such polygons are not guaranteed to arise from star unfoldings of 3D convex polyhedra.

Su-polygons are special cases of polyhedral metrics, under the rule that the two equal sized edges incident to a v-vertex are glued together.

3.2.3 Ridge tree

Given a convex polyhedron P and a source vertex s, a *ridge point* is a point on the surface of P connected to s by multiple shortest paths. The collection of all such points is a tree T_s, known as the *ridge tree* from s. In an sp-star unfolding of P on the plane, this ridge tree becomes the part of the Voronoi diagram of the source images contained inside the sp-star unfolding polygon, which is, in this case, non-overlapping (see [47, 8] for details of these properties).

3.3 Results

We organize the results presented in this chapter in the following way:

3.3.1 Reconstruction of ridge tree

The ridge tree of a convex polyhedron P with respect to a source vertex s inherits the combinatorics of a tree. We would like to know whether all combinatorial types of such trees are represented among sp-star ridge trees. We prove the following geometric reconstruction theorem for combinatorial types of ridge trees.

Theorem 13 *Any combinatorial type of a topologically embedded tree can be realized by the ridge tree of a convex polyhedron together with a source vertex.*

3.3.2 Characterization of sp-star unfolding polygons

In order to prove Theorem 13, we first need to characterize the polygons that arise as the sp-star unfolding polygons. Since the ridge tree is the Voronoi diagram of the source images, the following characterization with respect to Voronoi diagram will be very helpful:

Theorem 14 *A simple su-polygon is an sp-star unfolding polygon if and only if (a) the perpendicular bisectors of the pairs of consecutive s-vertices are all present in the Voronoi diagram of all the s-vertices; and (b) any v-vertex lies precisely on the Voronoi edge corresponding to its two s-vertex neighbors.*

The relationship between the ridge tree and the Voronoi diagram of the source vertices has been identified and used in [47, 8], in the context of sp-star unfoldings. Our contribution is to prove that these conditions are sufficient to characterize the class of sp-star unfoldings. The fact that (in our formulation) they are also necessary is implicit in [8]. We show that if either one of the (a) or (b) conditions is violated in an su-polygon S, then S is *not* an sp-star unfolding.

3.3.3 Point sets supporting sp-star unfolding

We want to use Theorem 14 to construct examples of su-polygons which are, or are not, sp-star unfolding polygons. We say that a cyclically ordered point set *supports* an sp-star unfolding if there exists an sp-star unfolding polygon using these points as s-vertices. In other words, we want to know whether it is always possible to place v-vertices on the corresponding Voronoi edges to satisfy the *source angle sum* property of an su-polygon. We present an algorithm to decide the question.

Theorem 15 *There exist cyclically ordered point sets which do not support sp-star unfoldings, and others that do. For a given point set, the corresponding decision problem can be answered in $O(n \log n)$ time.*

A special situation appears when all the s-vertices of an su-polygon that arises as the sp-star unfolding are in convex position. In this case, condition (a) in Theorem 14 always holds. The Voronoi diagram is combinatorially a tree, and its leaf edges are infinite rays separating the Voronoi regions of the s-vertices. We are now asking: is it true that an su-polygon whose s-vertices are in convex position is in fact a *shortest*

path star unfolding? We will answer this problem in the negative. The decision problem can be answered by an algorithm whose structure is similar to the one for general position, but whose individual components are computationally faster.

Theorem 16 *There exist cyclically ordered point sets in convex position which do not support sp-star unfoldings, and others that do. For a given convex point set, the corresponding decision problem can be answered in $O(n)$ time.*

Organization of the proofs: We first characterize the polygons that arise as the sp-star unfoldings. Then, using that characterization, we move to prove Theorem 15 and 16. Finally, we present the construction of ridge tree as presented in Theorem 13.

3.4 Characterization of sp-star unfolding polygons

In this section, we give a complete characterization of polygons that arise as the sp-star unfolding of some convex polyhedron with respect to some source vertex. This characterization will be pivotal in attaining our goal of reconstructing ridge tree and its underlying sp-star star unfolding.

In Theorem 14, one direction is already implicit in [8]. We only need to show:

Theorem 17 *Let S be a simple su-polygon satisfying two conditions: (a) the perpendicular bisectors of the pairs of consecutive s-vertices are all present in the Voronoi diagram of all the s-vertices; and (b) any v-vertex lies precisely on the Voronoi edge corresponding to its two s-vertex neighbors. Then S is an sp-star unfolding polygon.*

Proof The proof proceeds by contradiction. Let v_i be a v-vertex with two adjacent s-vertices s_{i-1} and s_i. We assume that the segment $v_i s_i$ (and its glueing mate of equal length $v_i s_{i-1}$) does not correspond to the shortest geodesic on the polyhedral surface P. Then, on the polyhedral surface, the shortest path is located (in the cut-star

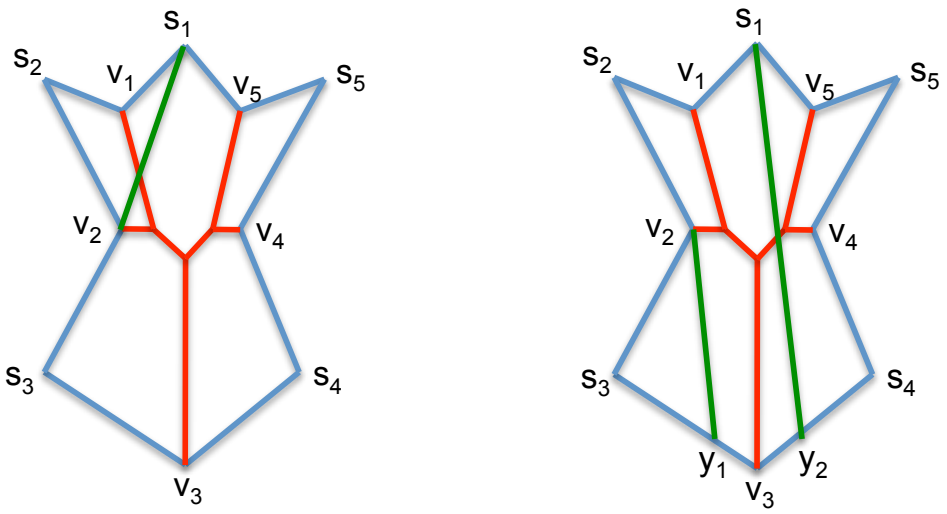


Figure 3.1: The two cases appearing in the proof of Theorem 14.

rotation around the source point s) between some pair of cut edges from the source to v_j and v_{j+1}. In the su-polygon S, the shortest path induces a line segment starting from the s-vertex s_j and going towards the interior of the su-polygon.

Two scenarios are possible (Fig. 3.1): (a) Either the shortest path lies fully inside the su-polygon and it is a straight line joining v_i and s_j, or (b) the shortest path is *broken* and represented by a series of k line segments that exit the su-polygon through an edge and re-enters it again through its corresponding gluing edge.

In case (a) the proof is straightforward. Since v_i is on the Voronoi edge of s_{i-1} and s_i, then v_i is closer to s_i or s_{i-1} than any other s-vertices on the plane. Thus $v_i s_j$ cannot be the shortest path. See Fig. 3.1(a) for an example illustrating this case.

In case (b), we use the fact that if $v_k s_k$ is an edge of the su-polygon, then any point on this edge lies in the Voronoi region of s_k. Let first segment of the broken shortest path be $v_i y_1$, where y_1 is the point through which it exits the polygon; similarly, let the last segment be $y_n s_j$. We know that y_n is on the polygonal edge $v_k s_k$, for $k \neq i$. But then we must have $s_k \neq s_j$, because otherwise $y_n s_j$ will make zero angle with the

polygonal edge $v_k s_k$ (each line segment of the shortest path makes a non-zero angle when it exits and enters the polygon). Since y_n is on the edge of $v_k s_k$, it is on the Voronoi region of s_k, hence $|y_n s_k| < |y_n s_j|$. Therefore, there exists another s-vertex s_k which has a shorter length path from v_i. Thus the path from v_i to s_j is not the shortest. See Fig. 3.1(b) for an example illustrating this case.

The derived contradictions conclude the proof that each polygonal edge of the su-polygon comes from the shortest path unfolding of some convex polyhedron with respect to some source vertex. ∎

We turn now to applications for this characterization of sp-star unfolding polygons.

3.5 Points supporting star unfoldings

In this section we assume that a cyclically ordered point set has been given. We want to decide if it supports an sp-star unfolding polygon and, if so, to construct one.

Flap polygon. This technical concept, needed to formulate and prove the results in this section, is obtained by relaxing the third condition defining an su-polygon: we no longer ask for the s-angle sum to be at most 2π (it can be anything).

3.5.1 Source points in convex position

We start with the simpler situation when the s-vertices lie in convex position. In this case, their Voronoi diagram is a tree, and the leaf edges are infinite rays separating consecutive s-vertices s_i and s_{i+1}. The conditions in Theorem 14 are violated only when some v-vertex is placed on the *extension* of a Voronoi ray and not on the ray itself.

To prove Theorem 16 we will construct a flap polygon with s-vertices in convex position and show that there is no readjustment of its v-vertices, keeping them on the Voronoi rays, so that the s-angle sum condition is satisfied. We need the following simple lemma:

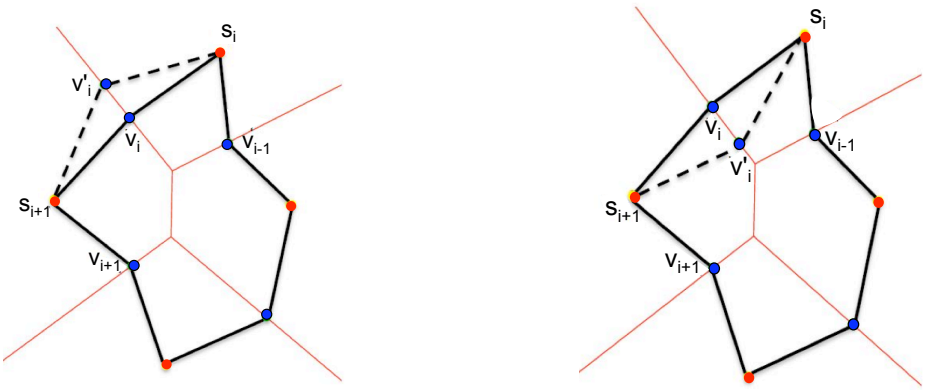


Figure 3.2: Dashed lines indicate the increase/decrease behavior of the two source angles incident to a v-vertex v_i, when displaced from its position: (Left) increase, when the displacement is towards the open end of the infinite Voronoi edge, and (Right) decrease, when moved towards the closed end.

Lemma 18 *Given a flap polygon, if a v-vertex moves towards the open end of the Voronoi ray, the sum of the s-angles increases and if it moves towards the closed end, the sum decreases.*

Proof Let s_i and s_{i+1} be two consecutive s-vertices on a flap polygon P_x. Both of these vertices are joined to a polyhedral vertex v_i. Refer to Fig. 3.2. The sum of the s-angles at these two s-vertices is $\angle v_i s_i v_{i-1}$ and $\angle v_i s_{i+1} v_{i+1}$. Now we move v_i towards the open end of the Voronoi edge. Let this new position of v_i be v'_i. This movement of v_i only affects the angles at s_i and s_{i+1}. The new angles at these two vertices are $\angle v_i s_i v_{i-1} + v_i s_i v'_i$ and $v_i s_{i+1} v_{i+1} + v_i s_{i+1} v'_i$ respectively. These two new angles are clearly larger than the old ones. So the sum of s-angles is increased. Similarly, we can prove that the sum of the s-angles decreases when a v-vertex moves towards the closed end. ∎

When the s-vertices of a flap polygon are in convex position, the bisector of any consecutive s-vertices is an infinite edge (a ray) of the Voronoi diagram of the s-

vertices. The v-vertices are placed on these rays. An *extreme flap polygon* arises when the v-vertices are placed on the Voronoi vertices at the closed end of the corresponding Voronoi rays. Note that an extreme flap polygon always has at least two pairs of adjacent overlapping edges, as in Fig. 3.3. This can be seen on the dual Delaunay triangulation, which is in this case a maximal outerplanar graph; thus, it has at least two vertices of degree two. One such vertex gives rise to overlapping adjacent edges in the extreme flap polygon.

The following is a straightforward consequence of Lemma 18:

Corollary 19 *For a given set of s-points in convex position and variable v-vertices placed on the Voronoi rays, the sum of the source angles is minimized when the corresponding flap polygon is extreme.*

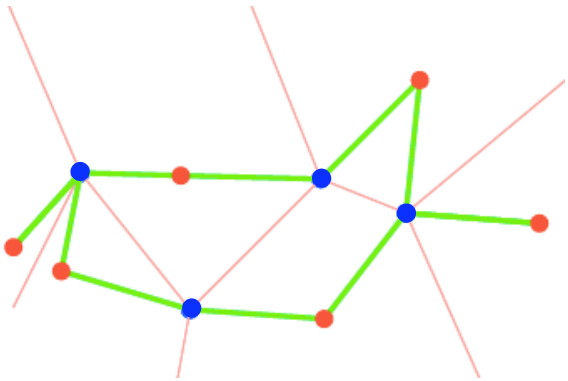


Figure 3.3: An extreme flap polygon (in green) together with the Voronoi diagram of the s-vertices (in red).

Lemma 20 *If the sum of the s-angles of an extreme flap polygon with s-vertices in convex position is larger than 2π, then there are no valid placements for the v-vertices that would result in an su-polygon supporting an sp-star unfolding.*

Proof A flap polygon does not support a sp-star unfolding if the sum of its s-angles is larger than 2π or if the v-vertices are not placed on the Voronoi rays. But in

an extreme flap polygon, the sum of the s-angles cannot be reduced (Corollary 19) without moving the v-vertices further inward, thus violating the condition that they must lie on the Voronoi rays. ∎

Proof of Theorem 16. Using Lemma 20, we only need to find a set of points in convex position whose extreme flap polygon has total s-angle sum larger than 2π. This is not unique, and not rare either. An example where the sum of the s-angles is larger than 2π is illustrated in Fig. 3.3.

Algorithm. Given a set of points in convex position, the following simple algorithm decides if they support an sp-star unfolding. We first compute the Voronoi diagram, then we generate the extreme flap polygon by placing the v-vertices at the corresponding Voronoi vertices on the infinite rays. If the sum of the s-angles exceeds 2π, the given point set does not support an sp-star unfolding; otherwise, it does.

Computing the Voronoi diagram of n points in convex position can be done in $O(n)$ time using the algorithm in [2]. Constructing the extreme flap polygon takes linear time. The whole algorithm is therefore linear. ∎

3.5.2 Source points in arbitrary position

Not all sp-star unfolding polygons have s-vertices in convex position. An example is given in Fig. 3.4(a).

We assume now that we are given a circularly ordered point set in non-convex position. The polygon induced by the ordering need not be simple. Such a point set may not even support a flap polygon, but when the perpendicular bisectors of consecutive pairs of points are present in the Voronoi diagram of the point set, there always exists at least one, namely the extreme flap polygon.

The definition of an *extreme flap polygon* can be extended naturally from the convex case: instead of requiring that the v-vertices be on the closed end of a Voronoi ray, we ask that it be placed on the endpoint of the Voronoi edge that is towards

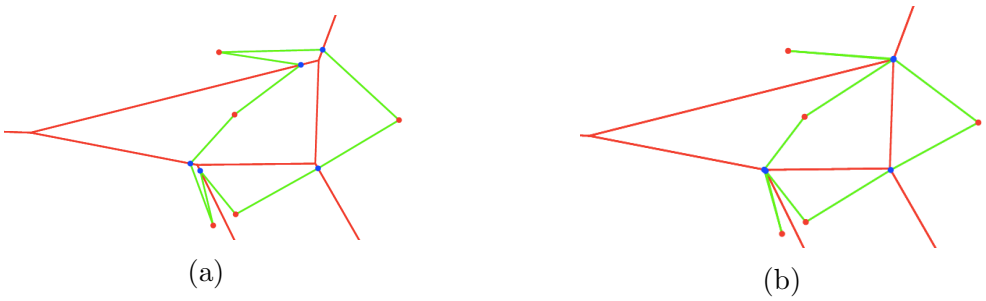


Figure 3.4: (a) An sp-star unfolding polygon (in green) together with the Voronoi diagram of its s-vertices (in red). The s-vertices are in non-convex position. (b) The corresponding extreme flap polygon on the s-vertices.

the interior of the flap polygon. Fig. 3.4(b) illustrates an extreme flap polygon on s-vertices in non-convex position. Lemmas 18 and 19 above can be extended in a straightforward manner to the non-convex case.

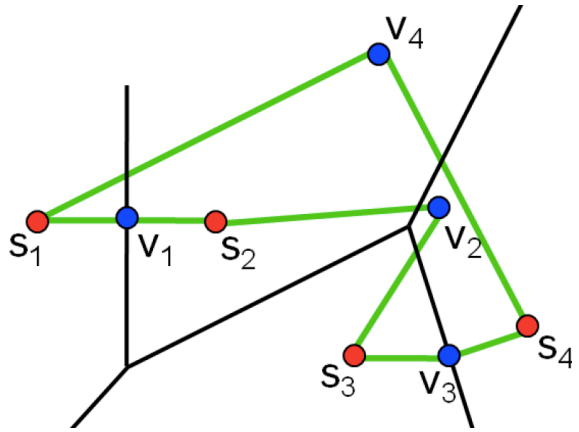


Figure 3.5: An example of an su-polygon which violates condition (a): the bisector of $\{s_4, s_1\}$ is not in the Voronoi diagram.

The example illustrated in Fig. 3.5 has the property that *not all* of the consecutive perpendicular bisectors of the s-vertices are part of the Voronoi diagram. This proves that:

Lemma 21 *There exist circularly given point sets in non-convex position that do not support sp-star unfolding polygons by violating condition (a) of Theorem 17.*

For the example in Fig. 3.6, the consecutive perpendicular bisectors are present in the Voronoi diagram of the s-vertices, but the extreme flap polygon does not satisfy the s-angle condition: its s-angle sum is $380^0 > 2\pi$. This proves:

Lemma 22 *There exist circularly given point sets in non-convex position that satisfy condition (a) but violate condition (b) of Theorem 17: they do not support sp-star unfolding polygons.*

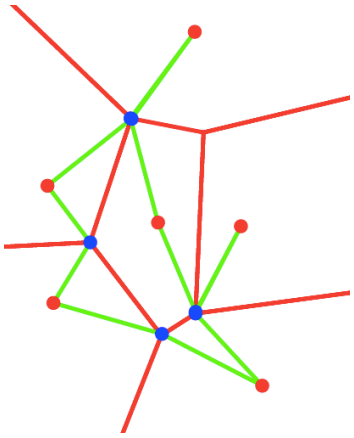


Figure 3.6: An example proving Lemma 22.

We complete the proof of Theorem 15 by giving an algorithm to decide whether a cyclically ordered point set in non-convex position supports an sp-star unfolding polygon. This is a straightforward extension of the one previously presented for the convex case.

Algorithm. Given a cyclically ordered point set $(s_i)_{i=1,..,n}$, construct its Voronoi diagram in $O(n \log n)$ time and verify, in linear time, whether the perpendicular bisectors of consecutive pairs of points appear in the Voronoi diagram. If not, stop and return *False*. Otherwise, construct (in linear time) the extreme flap polygon and compute the sum of the s-angles. If they exceed 2π, return *False*, otherwise return *True*. The entire calculation takes $O(n \log n)$ time. ∎

3.6 Realizations of combinatorial ridge trees

In this section, we turn to realizations of sp-star unfolding polygons with prescribed combinatorics for their ridge trees, and prove Theorem 13. The techniques described in the previous sections reduce it to the following reformulation.

Theorem 23 *Any combinatorial tree T arises as the Voronoi diagram of a point set in convex position whose extreme flap polygon can be relaxed to an su-polygon that supports an sp-star unfolding.*

Proof overview. We first consider the maximal outerplanar graph which is the graph dual of the given tree. We realize this graph as the Delaunay triangulation D_T using Dillencourt's algorithm [19]. The dual of this Delaunay triangulation D_T gives the realization of the tree as the Voronoi diagram V_D. The vertices of D_T are the sites of V_D. Since the sites of V_D are in convex position, there is an infinite Voronoi segment in between two consecutive sites. We place one new vertex on the Voronoi vertex located on the closed end of each of these infinite segments. If we alternately join the sites and new vertices, we obtain an extreme flap polygon. We show that this flap polygon supports an sp-star unfolding. For this, we establish a relationship between the s-angles of this extreme flap polygon and the angles of the Delaunay triangulation. We then use this relationship to prove that the sum of s-angles of this extreme flap polygon is always smaller than 2π.

The proof details occupy the rest of this Section.

3.6.1 Source angles and the angles of the Delaunay triangulation

We start by establishing a relationship between the sum of the source angles of an extreme flap polygon and the angles of the Delaunay triangulation of its *s-vertices*.

Let D_T be the Delaunay triangulation of the *s-vertices* and let SA be the sum of the *s-angles* of the extreme flap polygon. If V_D is the Voronoi diagram of the *s-vertices*, its internal edges are duals of the diagonals of D_T.

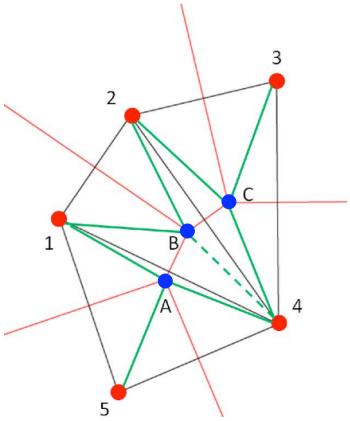


Figure 3.7: A point set in convex position, together with its Voronoi diagram (red), Delaunay triangulation (gray) and the extreme flap polygon (green).

Each s-angle in an extreme flap polygon is spanned by one or more internal edges of the Voronoi diagram. Therefore, the sum of the s-angles SA is actually the sum of the angles spanned by each internal edge of the Voronoi diagram. We establish the relationship between s-angles and the diagonals of the D_T, which are dual of the internal edges of V_D. See Fig 3.7.

Given a Delaunay realization D_T of a maximal outerplanar graph, we can classify the angles of D_T into two sets. One is the set of angles opposite to all diagonals, known as *internal angles* and the other is the set of angles opposite to the boundary edges of D_T, known as *boundary angles*. Let α and β denote the sum of each of these two sets of angles respectively. Since these two sets of angles comprise all the angles of the convex polygon formed by the boundary edges of D_T, $\alpha + \beta = (n-2)\pi$.

Lemma 24 *Let D_T be the Delaunay triangulation of n s-vertices and let SA_n be the sum of the s-angles of its extreme flap polygon. Then:*

$$SA_n = (n-3)2\pi - 2\alpha_n$$

where α_n is the sum of the angles opposite to the diagonals of D_T.

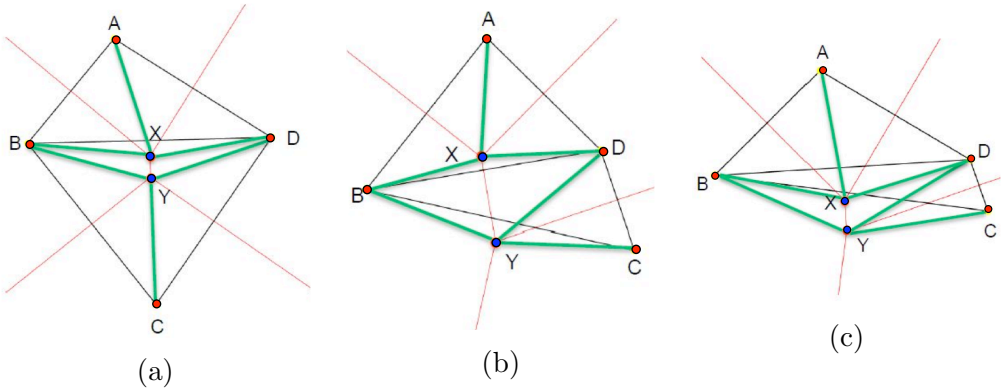


Figure 3.8: The base case analysis for the inductive proof of Lemma 24. Shown is an extreme flap polygon for four convex points A, B, C and D. The triangles $\triangle ABD$ and $\triangle BCD$ are Delaunay.

Proof. By induction on the number n of s-vertices.

Base case, n = 4: In this case we have one diagonal in D_T and one internal edge in V_D. Out of four s-angles, two of them are zero and the remaining two are spanned by the internal edge of V_D. We have to show that:

$$SA_4 = 2\pi - 2\alpha_4$$

Let A, B, C and D be the four s-vertices in convex position and let XY be the Voronoi edge dual to the diagonal BD. The source angles at A and B, the opposite vertices of the diagonal BD, are both zero. The non-zero source angles are $\angle XBY$ and $\angle XDY$. In $\triangle AXB$, $\angle BAX = \angle ABX$ (since X is on the perpendicular bisector of A and B). Similarly,$\angle DAX = \angle ADX$, $\angle YBC = \angle BCY$ and $\angle DCY = \angle CDY$. We show that $\angle XBY + \angle XDY = 2\pi - 2(A + C)$.

We consider three possible cases, illustrated in Fig. 3.8: (a) $\angle XBY$ is contained in $\angle ABC$, (b) $\angle XBY$ is partially contained in $\angle ABC$ and (c) $\angle XBY$ is outside $\angle ABC$. Three other cases are possible for the other source angle, but the same arguments apply.

Case 1: The angle sum $\angle ABC + \angle BCD + \angle CDA + \angle DAB$ of the quadrilateral ABCD is 2π, hence $\angle ABC + \angle CDA = 2\pi - (\angle DAB + \angle BCD)$. Decomposing the angles $\angle ABC$ and $\angle CDA$ we obtain:

$\angle ABX + \angle XBY + \angle YBC + \angle CDY + \angle XDY + \angle ADX = 2\pi - (\angle DAB + \angle BCD)$

Using the bisector properties in the isosceles triangles $\triangle AXB$ and $\triangle BYC$ yields:

$\angle BAX + \angle XBY + \angle BCY + \angle DCY + \angle XDY + \angle DAX = 2\pi - (\angle DAB + \angle BCD)$

From here, we get $\angle XBY + \angle XDY + \angle DAB + \angle BCD = 2\pi - (\angle DAB + \angle BCD)$, which implies the desired equality: $\angle XBY + \angle XDY = 2\pi - 2(\angle DAB + \angle BCD)$.

Cases 2 and 3: We have $\angle ABC = \angle ABX + \angle XBY - \angle YBC$. Therefore:

$$\angle ABC + \angle BCD + \angle CDA + \angle DAB = 2\pi$$

$$\angle ABC + \angle CDA = 2\pi - (\angle DAB + \angle BCD)$$

$$\angle ABX + \angle XBY - \angle YBC + \angle CDY + \angle XDY + \angle ADX = 2\pi - (\angle DAB + \angle BCD)$$

$$\angle BAX + \angle XBY - \angle BCY + \angle DCY + \angle XDY + \angle DAX = 2\pi - (\angle DAB + \angle BCD)$$

$$\angle XBY + \angle XDY + \angle DAB + \angle BCD = 2\pi - (\angle DAB + \angle BCD)$$

which yields the desired equality $\angle XBY + \angle XDY = 2\pi - 2(\angle DAB + \angle BCD)$. Since A and C are the opposite angles of the diagonal, we obtain:

$$SA_4 = 2\pi - 2\alpha_4$$

Inductive step. The induction hypothesis for $n-1$ s-vertices is that:

$$SA_{n-1} = (n-4)2\pi - 2\alpha_{n-1}$$

We now add a new s-vertex in convex position. As a result, a new triangle will be formed and one of the boundary edge will become a diagonal. Let e be this new

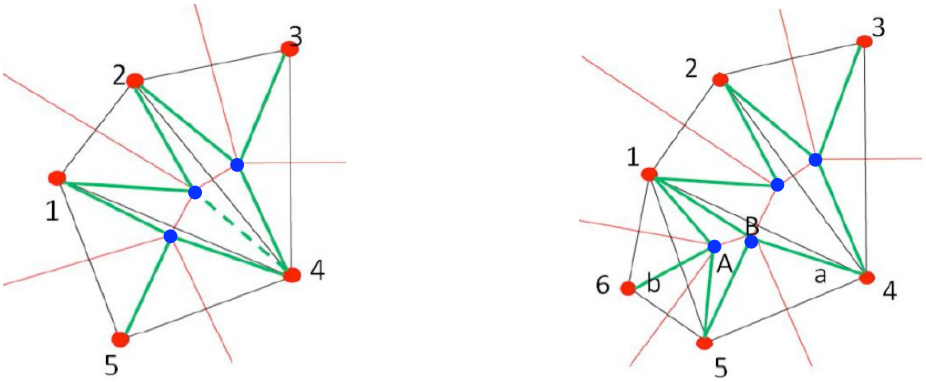


Figure 3.9: The inductive step in the proof of Lemma 24. (a) An extreme flap polygon with 5 s-vertices. (b) A new s-vertex is added.

diagonal. One of the two opposite angles of e was a boundary angle which has now become internal angle. The other opposite angle is a newly formed internal angle of D_T. Let us denote these angles by a_e and b_e, respectively. All other internal angles of D_T remain the same.

The addition of the new s-vertex leads to a new diagonal or a new internal Voronoi edge. The sum of s-angles increases by SA', the sum of angles spanned by the new Voronoi edge. See Fig 3.9. Using a similar argument as in the base case, we obtain:

$$SA' = 2\pi - 2(a_e + b_e)$$

The final SA_n is:

$$\begin{aligned} SA_n &= SA_{n-1} + SA' = (n-4)2\pi - 2\alpha_{n-1} + SA' \\ &= (n-4)2\pi - 2\alpha_{n-1} + 2\pi - 2(a_e + b_e) = (n-3)2\pi - 2\alpha_n \end{aligned}$$

Corollary 25 *$SA = 2\beta - 2\pi$, where β is the sum of boundary angles of the Delaunay triangulation of D_T.*

Proof In a Delaunay triangulation we have $\alpha + \beta = (n-2)\pi$. From Lemma 24 we infer that $SA = (n-3)2\pi - 2\alpha$. Substituting $\alpha = (n-2)\pi - \beta$ gives the result.

3.6.2 Realization of a Delaunay triangulation

The angles of the triangles in a Delaunay realization of an outerplanar graph satisfy a set of constraints [19]. To each angle at a vertex of a triangle in a Delaunay triangulation, associate a variable a_i. The constraints are:

1. **Positive**: All angles are positive : $\forall i, a_i > 0$.

2. **Face angles**: The three angles a_i, a_j, a_k of a triangle add up to π:

$$a_i + a_j + a_k = \pi$$

3. **Convex**: All the points are in convex position, hence the sum of the angles a_{i_j} at a vertex does not exceed π:

$$\sum_{j=1}^{k} a_{i_j} < \pi$$

4. **Locally Delaunay**: an edge shared by two triangles is locally Delaunay, i.e. the sum of the two angles opposite to the edge does not exceed π. If a_i and a_j are two angles opposite of an edge e, then $a_i + a_j < \pi$.

These constraints can be turned into a linear programming system on variables a_i. If the system has a feasible solution, it gives a set of angles from which the Delaunay triangulation can be realized. Dillencourt [19] showed that, for any maximal outerplanar graph, the linear programming system always has a feasible solution. However, a maximal outerplanar graph may have multiple Delaunay triangulation realizations,

but not all of them yield valid sp-star unfoldings. We now show that in fact Dillencourt's algorithm always yields a valid sp-star unfolding. Before proceeding with the proof we outline the algorithm.

3.6.3 Dillencourt's Algorithm.

A variable s is assigned to the whole triangulation T and a variable b_i to each angle of the triangulation, such that the following conditions are satisfied:

1. Each $b_i \geq 1$.
2. For each triangle, the sum of its angles is s: $b_i + b_j + b_k = s$.
3. For each vertex, the sum of all angles incident on it is less than or equal to s : $\sum_{j=1}^{k} b_{i_j} \leq s$.
4. If b_i and b_j are two opposite angles of an internal edge ij, then : $b_i + b_j \leq s - 1$.

By setting each $a_i = \frac{\pi . b_i}{s}$, these conditions become equivalent to the angle constraints. An inductive argument now shows that such an assignment always exists.

Lemma 26 *Dillencourt's algorithm yields a valid Delaunay triangulation, with vertices in convex position.*

Proof. If a triangulation T has only one triangle, then we assign 1 to each angle and set $s = 3$. This assignment satisfies all of the four conditions. We now proceed inductively, visiting new triangles adjacent with the already visited ones. Each time a new triangle is visited, we assign values to its angles and update s. Then we traverse again the already visited triangles and modify their angles to reinstate the above conditions. For example, if we have assigned values to the angles of the triangle ijk, we may be moving to one of its adjacent triangle ijl whose angles will have to be initialized (see Fig. 3.10(a)). Let x, y and z be, respectively, the angles at the vertices i, j, k of $\triangle ijk$. Then we assign angle values of $z+1, z+1$ and $s-z-1$ to the vertices

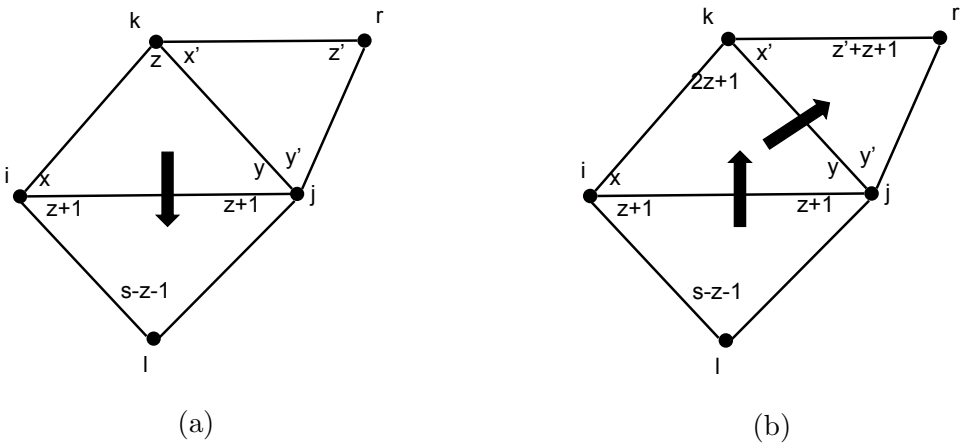


Figure 3.10: Illustration of the steps in Dillencourt's algorithm: (a) First visit of a triangle; (b) Revisiting previously visited triangles and readjusting the variables.

i, j, and l of $\triangle ijl$. Then we update the value of s by setting $s := s + z + 1$. Since this new value of s causes the second condition to fail (where sum of angles of a triangle is equal to s), we traverse each of the visited triangles from our current $\triangle ijl$. When we enter a visited triangle by crossing an internal edge (i.e. a diagonal of the outerplanar graph), we increase the angle opposite to the internal edge by $z + 1$. For example, when we move from the triangle ijl to ijk, we cross the edge ij and increase the value of the angle at the vertex k by $z + 1$ (see Fig. 3.10(b)). We continue in this fashion until we re-visit all the previously visited triangles. This whole process is repeated each time we visit and initialize the angles of a new triangle.

These assignments guarantee that all four conditions are satisfied. Indeed, all assignments are ≥ 1. Every time we assign angles to a new triangle, s is increased by $z + 1$. We increase exactly one angle of each visited triangle by $z + 1$ and second condition is satisfied. When we update the angles of all visited triangles, the sum of the angles at each vertex is increased by $z+1$ and at the same time s is also increased by $z + 1$. So, the inequality of condition 3 still holds. Finally, when we visit a new triangle, one of its angle is opposite to a diagonal (like angle at vertex l of triangle

ijl). This angle is assigned $s - z - 1$ and its opposite angle in the just visited triangle (like angle at vertex k of triangle ijk) is assigned $z + (z + 1) = 2z + 1$. So, their sum $s + z \leq (s + z + 1) - 1$. After these assignments and modifications of these two triangles, we increase the angles of each visited triangle and this inequality remains valid. ∎

We retain for further use the following properties of this algorithm:

1. Each time a new triangle is visited, the value of s is increased by $z + 1$.

2. After each iteration, one boundary angle (the one opposite to a boundary edge) becomes an internal angle (i.e. opposite to a diagonal), and two new boundary angles are introduced.

3. During the update phase, only internal angles are updated. Boundary angles remain unchanged.

4. The time complexity of this algorithm is $O(n)$ [34]

3.6.4 Delaunay realization supporting the sp-star unfolding

In this section, we show that if a maximal outerplanar graph is realized as a Delaunay triangulation using Dillencourt's algorithm, then the sum of the s-angles in the extreme flap polygon is less than 2π.

Let D_T, SA and β denote the resulting Delaunay triangulation using Dillencourt's algorithm, the sum of the s-angles in the extreme flap polygon and the sum of the boundary angles (whose opposite edges are the boundary edges of the Delaunay triangulation) respectively. We know from Corollary 25 that, $SA = 2\beta - 2\pi$. If β is larger than π but smaller than 2π, then the resulting extreme flap polygon will support an sp-star unfolding.

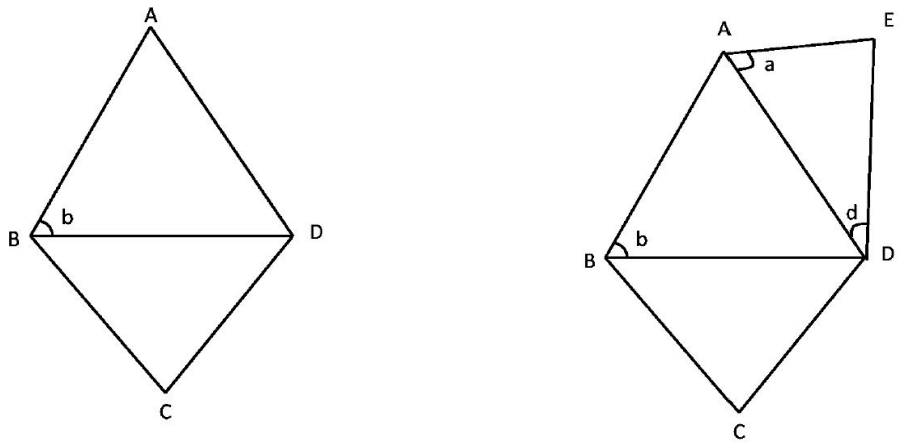


Figure 3.11: Illustration of the inductive step in the proof of Lemma 27. (a) In this triangulation, b is a boundary angle. (b) At the next iteration, a new triangle AED is added and b becomes an internal angle; a and d are new boundary angles.

Let us assume that Dillencourt's algorithm assigns a total integer value a to the boundary angles of D_T. Let $\beta = \frac{a.\pi}{s}$. We remind the reader that s is the sum of integer values assigned to any triangle.

Lemma 27 *Let a_n be the sum of the integer values assigned to the boundary angles and s_n be the sum of the integer values assigned to the angles of any triangle, where n is the number of the triangles in D_T. Then, $s_n < a_n < 2s_n$.*

Proof By induction on the number of triangles of D_T.

Base case. The conditions are satisfied when $n = 2$, since $s_2 = 5$ and $a_2 = 6$.

Induction Step. We assume as induction hypothesis that $s_{n-1} < a_{n-1} < 2s_{n-1}$. Whenever we add a new triangle, one boundary angle becomes internal (whose opposite edge is a diagonal shared by the new triangle) and two new boundary angles are added to the triangulation. See Fig 3.11. Each of the two new boundary angles is assigned $(z + 1)$ where z was the value of the now-converted internal angle. New

value of s_n is obtained by adding $(z+1)$ to s_{n-1}. All other boundary angles remain unchanged. Therefore:

$$a_n = a_{n-1} - z + 2(z+1) = a_{n-1} + z + 2$$

Similarly, $s_n = s_{n-1} + z + 1$. Using the induction hypothesis that $s_{n-1} < a_{n-1} < 2s_{n-1}$ and replacing s_{n-1} and a_{n-1} in the above equation, we obtain:

$$\begin{array}{rrr} s_{n-1} & < a_{n-1} & < 2s_{n-1} \\ s_n - z - 1 & < a_n - z - 2 & < 2s_n - 2z - 2 \\ s_n & < a_n - 1 & < 2s_n - z - 1 \\ s_n + 1 & < a_n & < 2s_n - z \\ s_n & < a_n & < 2s_n \end{array}$$

This completes the proof. ∎

Algorithm. The s-angles obtained from the Dillencourt's algorithm sum up to less than 2π. In a valid sp-star unfolding, the sum of these angles will have to be exactly 2π. In this section, we show how we can increase the angles such that they add up to 2π.

We define the slack angle s_a as the difference between 2π and the sum of the s-angles SA obtained from Dillencourt's algorithm, i.e, $a = 2\pi - SA$. All the v-vertices V are placed on the Voronoi vertices when we apply Dillencourt's algorithm. Now we move each v-vertex $v \in V$ along the infinite edge of the Voronoi diagram away the corresponding Voronoi vertex (towards the open end). When we move v, each of the two s-angles at its neighboring s-vertices increases equally. Therefore, we move each v-vertex v such that each of the s-angles at its two neighboring s-vertices is increased by exactly $\frac{s_a}{2n}$.

The construction of the Delaunay triangulation from a maximal outerplanar graph takes $O(n)$ time [34]. The time complexity of moving the v-vertices to make the sum of the s-angles equal to 2π is also $O(n)$. Therefore, the whole algorithm runs in $O(n)$ time.

3.7 Tools for experimentation

We have written two programs, both in Mathematica, to facilitate the exploration of several properties of the sp-star unfolding. First one computes the sp-star unfolding of a convex polyhedron. It first computes a random convex polyhedron from the number of vertices given as input. Based on the user defined source vertex, it computes the shortest paths from the source vertex to all polyhedral vertex and then lay out the unfolding. For shortest path computation, we have used an existing implementation, written in C, found at http://cs.smith.edu/ orourke/ShortestPaths/.

In the second tool, also written in Mathematica, we implemented the Dillencourt's algorithm. The program takes, as input, a tree as graph. Then it computes the Delaunay triangulation and extreme flap polygon from that. Finally, we move all the v-vertices to complete the sum of the s-angles to 2π.

3.8 Conclusion

We have presented several results on the characterization of sp-star unfolding polygons and reconstruction of ridge trees. We have studied sp-star unfoldings with source images in arbitrary position and shown that not all point sets, even if they are in convex position, support an sp-star unfolding. We gave a reconstruction algorithm, showing that any maximal outerplanar graph arises as the dual of the ridge tree of a sp-star unfolding. Finally, we presented algorithms to check whether a given set of points can be realized as the source images of an sp-star unfolding.

CHAPTER 4

GEODESIC STAR UNFOLDING

4.1 Introduction

In the sp-star unfolding, we use the shortest paths from a source vertex s to the vertices of a convex polyhedron P as the cut edges. We now present the following version of the sp-star unfolding: we take a set of non-crossing geodesic paths, all of which are not necessarily the shortest and cut along these paths. The result is a curvature-zero disk like surface on the plane. We refer to this more general setting as *geodesic star unfolding*. Since it is a relaxed version of the sp-star unfolding, it shares some of the properties of the sp-star unfolding, and for the same reason, also possesses some contrasting properties.

Geodesic star unfolding polygons are polyhedral metric. Therefore, any polygon that is geometrically equivalent to a geodesic star unfolding of a convex polyhedron is also a convex polyhedral metric. In Chapter 3, we have seen how we can generate an sp-star unfolding polygon. One might naturally wonder if the same techniques can be applied here to generate geodesic star unfolding polygons, which will give us another way of generating polyhedral metric.

Because of the close relationship between these two types of unfolding, the following interesting question arises: given a geodesic star unfolding of a convex polyhedron with respect to a source vertex, can we convert it to the sp-star unfolding of the same polyhedron and the source vertex? If it is possible, then we can, in fact, compute the shortest paths from the source vertex to all polyhedral vertices using a geodesic star unfolding.

4.1.1 Results

The results presented in this chapter are the following :

First, we study the properties of the geodesic star unfolding of a convex polyhedron. We compare these properties with those of the sp-star unfolding and try to answer the following question:

Problem 1 *Does the geodesic star unfolding have the same properties as the sp-star unfolding?*

We find that the statement is partially true and identify some properties unique to the geodesic star unfolding.

Second, we discuss the reconstruction problem of the geodesic star unfolding polygons. We are asking the following question:

Problem 2 *Can we construct a geodesic star unfolding polygon from a given set of cyclically ordered points, designated as the source images?*

We discuss this problem in the context of the su-polygon and prove that su-polygons are, by definition, equivalent to geodesic star unfoldings (refer to Section 3.2.1 for detailed definition of the su-polygon). Therefore, this problem reduces to the reconstruction problem of the su-polygon from a set of s-vertices. We define some challenges of this problem and present an algorithm to generate an su-polygon when all but one s-vertices are on a circle.

Finally, we address the problem of computing the sp-star unfolding polygon from the geodesic star unfolding of the same polyhedron. We formulate the problem as follows:

Problem 3 *Given a geodesic star unfolding of a convex polyhedron with respect to some source vertex, can we obtain the sp-star unfolding of the same polyhedron from it?*

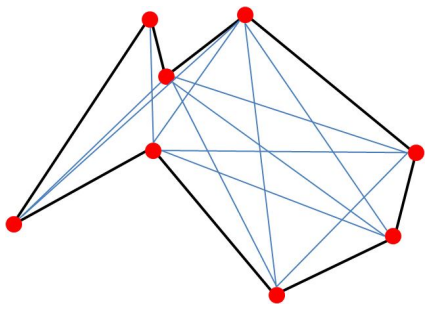


Figure 4.1: Diagonals of a simple polygon

We show, using experimental data, that we can always derive the sp-star unfolding from the geodesic star unfolding of the same convex polyhedron and the source vertex.

4.2 Preliminaries

A *geodesic path* between two points on the surface is a locally shortest path. Given two points on the polyhedral surface, there may be more than one, even infinite number of geodesic paths between the points. In the remainder of the text, we consider the geodesic paths between a source vertex and the polyhedral vertices of a convex polyhedron. In this setting, the geodesic path cannot pass through a polyhedral vertex. Although the shortest paths, by definition, are geodesics, we use geodesics, in this text, to refer to non-shortest paths.

Given a convex polyhedron and a source vertex, the *geodesic star unfolding* is obtained by cutting along a set of non-crossing paths where some or all of these paths are geodesics. The result is a flat disk-like surface with a polygonal boundary.

Given a polyhedron and a source vetex, we can angularly sort the geodesic paths from the source vertex to all polyhedral vertices around the source vertex. This map of geodesic paths gives us an ordering of the polyhedral vertices. The boundary of the *core* is the lines joining these vertices in this order.

(a) (b)

Figure 4.2: (a)Face angles of one vertex of the cube are shown. The curvature of this point is 2π minus the sum of these face angles.(b) When flattened on the plane, curvature of a point is the angle exterior to the unfolding polygon.

The *diagonal* of a simple polygon joins two non-consecutive vertices and lies entirely inside the polygon. Any simple polygon has at least $(n-3)$ (excluding boundary edges) and at most $O(n^2)$ diagonals. See Fig 4.1.

The *Gaussian curvature* or *curvature*, in short, of any point p on a polyhedral surface is the angle deficit at p: 2π minus the sum of the face angles incident to p [18].The vertices of a convex polyhedron always have positive curvature. See Fig 4.2.

4.3 Properties

Let n be the number of the vertices of the convex polyhedron P. The geodesic star unfolding S_s, obtained by cutting along a set of non-crossing geodesic paths, is a polygon with an interior. It has $2n$ vertices if the source vertex is on the face or on an edge, and $2(n-1)$ otherwise. Along the boundary of the polygon, the polyhedral vertices and the source images appear alternately. The sum of the angles, interior to the surface, at the source images is exactly 2π under the assumption that the source vertex is on a face or on an edge of the polyhedron. Each polyhedral vertex lies on the perpendicular bisectors of its two neighboring source images.

(a) (b)

Figure 4.3: (a) One of the shortest paths on the tetrahedron from Fig. 1.3(a) is replaced by a geodesic. (b) The corresponding geodesic star unfolding polygon. Red and blue vertices are source images and polyhedral vertices respectively. One an source angle is larger than π.

4.4 Contrasts with sp-star unfolding

Although geodesic star unfoldings share some basic properties with sp-star unfoldings, there are some surprising differences between them. One immediate difference is on the source angle - in an sp-star unfolding, all source angles are acute. Although it is possible for all source angles to be acute in a geodesic star unfolding, in some instances, however, one of the angles is larger than π. See Fig 4.3. Some of the other, non-obvious differences are described below.

4.4.1 Non-simple, self-overlapping polygon

Unlike sp-star unfolding polygons, geodesic star unfolding polygons may be self-overlapping. We found an example where the geodesic star unfolding polygon of a convex polyhedron has an overlap.

Theorem 28 *Not all geodesic star unfolding polygons of a convex polyhedron and a source vertex are simple, non-overlapping.*

Proof We present a counter example of a geodesic star unfolding, in Fig 4.4, which shows that it is not always a simple polygon. ∎

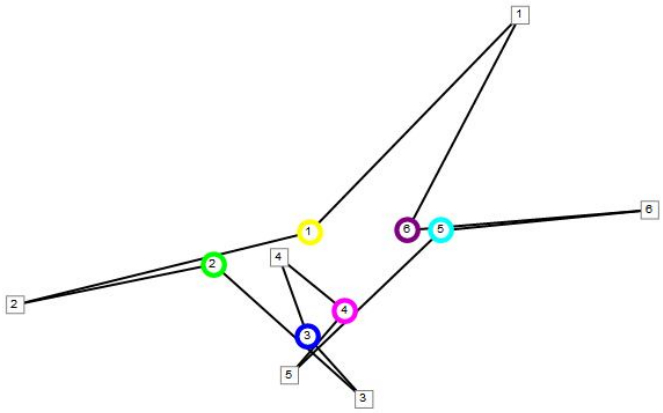


Figure 4.4: An example of a self-overlapping geodesic star unfolding polygon. The circular and square vertices are polyhedral vertices and source images respectively.

4.4.2 Placement of polyhedral vertices

We know, from the characterization of the sp-star unfolding, that each polyhedral vertex lies exactly on the corresponding Voronoi segment. One might expect that this property extends to the geodesic star unfolding. On the contrary, we find that, in certain occasions, the polyhedral vertices lie on the extension of the corresponding Voronoi segments. Even surprisingly, we discover that some of the perpendicular bisectors on which the polyhedral vertices lie are not even part of the Voronoi diagram. Both cases are illustrated in Fig 4.5.

In the sp-star unfolding, we know that the ridge tree is the Voronoi diagram of the source images. But the counter example shown here indicates that we cannot relate the Voronoi diagram of the source images of the geodesic star unfolding to similar type of geometric objects.

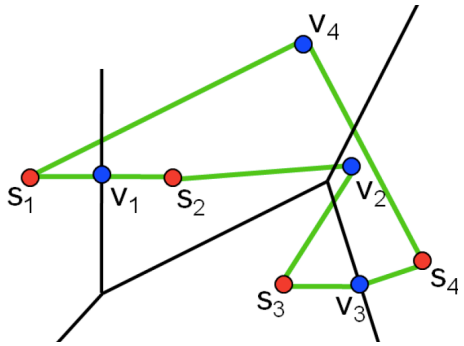


Figure 4.5: An example of a geodesic star unfolding. Black edges denote the Voronoi diagram. Polyhedral vertex v_2 is on the extension of a Voronoi segment and the bisector on which v_4 lies is not present in the Voronoi diagram.

4.4.3 Self-intersecting core

It is known that the core of an sp-star unfolding is a simple polygon [18]. In fact, the core contains the ridge tree in the unfolding. This property does not extend to the geodesic star unfolding and we discover an example where the core is not simple.

Theorem 29 *There exist a convex polyhedron P, a source vertex s and a set of non-crossing geodesic paths from s to the vertices of P such that the core of the resulting geodesic star unfolding polygon is self-intersecting.*

Proof An example of a self-intersecting core of a geodesic star unfolding is shown in Fig 4.6. ∎

4.5 Relation with su-polygon

An su-polygon is the boundary of a flat disk-like polygon satisfying the following properties: (a) it has $2n$ vertices, alternately labeled as s_i and v_i and referred to as s-, resp. v-vertices. The angle (interior to the surface) at an s-vertex is referred to as an s-angle; (b) the sum of all the s-angles is 2π, and (c) the v-vertices are placed on the perpendicular bisectors of the two neighboring s-vertices. In particular, this last property implies that the two edges incident to a v-vertex have equal lengths.

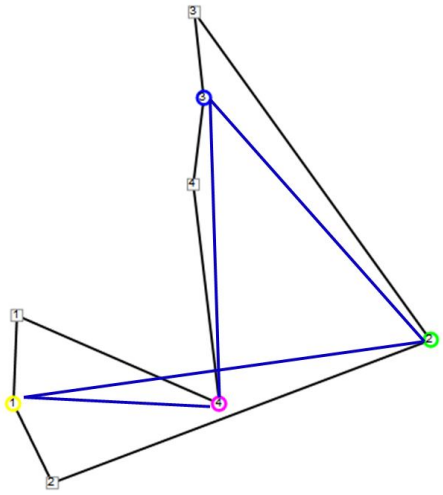


Figure 4.6: An example of a geodesic star unfolding where the core, shown in blue edges, is self-intersecting.

We show that, every su-polygon arises from a geodesic star unfolding of some convex polyhedron with respect to some source vertex. Before that, we need to explain Alexandrov's theorem on polyhedral metric.

Alexandrov's theorem: A. D. Alexandrov proved a significant theorem on the existence of convex polyhedra [6] as stated in [18]:

Theorem 30 *For every convex polyhedral metric, there exists a unique convex polyhedron (up to translation or a translation with a symmetry) realizing this metric.*

A *convex polyhedral metric* is a presentation of the surface of a convex polyhedron as a collection of (one or more) planar polygonal pieces together with rules for glueing them, along pieces of their boundaries, into a surface that is topologically a sphere. The glueing must satisfy the following properties : i) Each edge of the polygon can be glued to exactly one other polygonal edge. ii) the curvature at any point is non-negative and iii) the glueing results in a complex homeomorphic to sphere, or equivalently, the total curvature of the surface is 4π.

Now we show that every su-polygon is a convex polyhedral metric.

Theorem 31 *Any su-polygon is a convex polyhedral metric*

Proof Since each v-vertex of an su-polygon P is on the perpendicular bisector of its two neighboring s-vertices, the two edges emanating from a v-vertex can be glued together. So, each edge of P glues to exactly one other edge of P.

All points except v-vertices on P have zero curvature. Each v-vertex has positive curvature which is the the angle at the v-vertex exterior to P. We show that the sum of the curvature at all v-vertices is 4π.

The sum of the angles at all v- and s-vertices interior to P is $(2n-2)\pi$, where $2n$ is the number of vertices of P (n s-vertices and n v-vertices). Since the sum of the s-angles is 2π (by the definition of su-polygon), the sum of the angles at v-vertices interior to P is $2n\pi - 4\pi$. The sum of all the angles, interior and exterior to P, at v-vertices is $2n\pi$. Therefore, the sum of the angles exterior to P at v-vertices, or equivalently, the total curvature of P is 4π. Therefore, P is a convex polyhedral metric. ∎

Since an su-polygon is a convex polyhedral metric, it can be folded back into a convex polyhedron. After folding, its boundary edges represent the cut edges on the polyhedral surface. Since these cut edges are straight lines, these are also geodesics (some of which may be the shortest) from the source to the polyhedral vertices. Therefore, we have the following result:

Theorem 32 *Any su-polygon is a geodesic star unfolding of some convex polyhedron and a source vertex.*

From now on, we will be using su-polygons and geodesic star unfolding polygons interchangeably when there is no chance of confusion.

4.6 Construction of geodesic star unfolding polygons

In Chapter 3, we characterized polygons that arise as the sp-star unfoldings and used that characteristics to present an algorithm to reconstruct the ridge tree and its underlying sp-star unfolding. Since the geodesic star unfolding is a relaxed version of the sp-star unfolding, one might expect that the similar technique can be applied here to its reconstruction. However, we have a different experience and describe some challenges in this regard.

4.6.1 Challenges

We showed, in Chapter 3, that some set of s-vertices do not support sp-star unfolding polygons. In those cases, the sum of the s-angles of the extreme flap polygon, built upon those s-vertices, exceeds 2π. Although that poses a problem for the sp-star unfolding reconstruction, but for the geodesic star unfolding, that is in fact, not prohibitive. In a geodesic star unfolding, all of the polyhedral vertices can *not* be on the corresponding Voronoi segments. If the sum of the s-angles of an extreme flap polygon is larger than 2π, we should be able to push some v-vertices inwards until the s-angles sum up to 2π. It will cause some v-vertices to be placed on the extension of their corresponding Voronoi segments. Then, according to Theorem 14 and 32, the resulting polygon is a geodesic star unfolding.

However, the problem with this approach is that we do not know how far we can push each v-vertex inwards. There is no geometric properties that will guarantee that we can push a set of v-vertices inwards enough to make the sum of s-angles to 2π without crossing through other vertices or edges. See Fig 4.7. This complicates the development of an algorithm for the construction of a geodesic star unfolding.

4.6.2 Point sets supporting geodesic star unfoldings

Due to the problem discussed in the previous section, we need to have a set of points, designated as s-vertices, such that the v-vertices can freely move on the

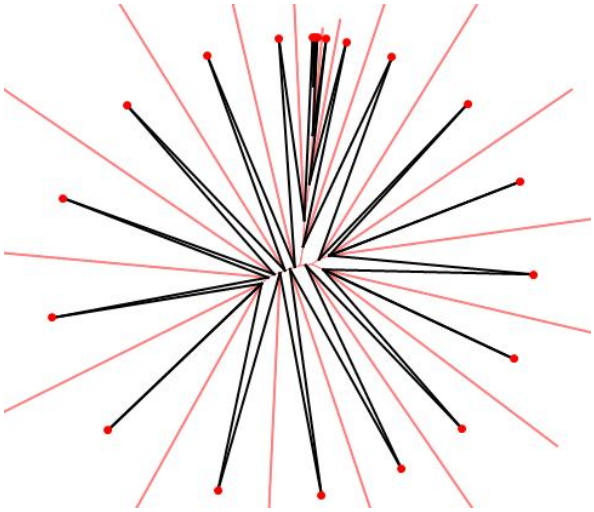
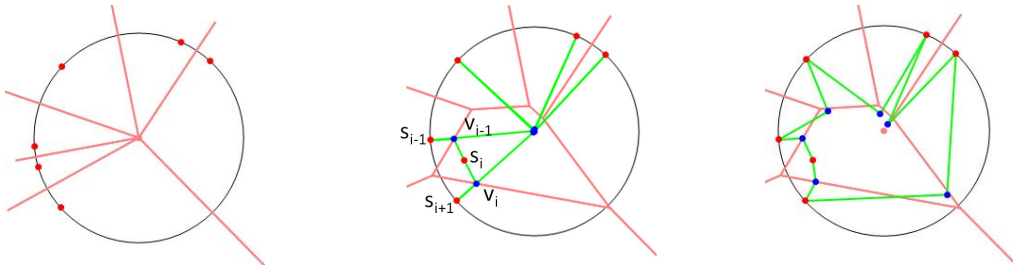


Figure 4.7: An example of a extreme flap polygon with larger than 2π s-angles. It is difficult to push the v-vertices inwards to make the sum of the s-angles equal to 2π while maintaining a valid geodesic star unfolding.

extension of the Voronoi segments. One such configuration is to place all s-vertices on the circle. The bisectors of all pair of consecutive s-vertices, also part of the Voronoi diagram, meet each other exactly at the center of the circle. See Fig 4.8(a). In other words, the Voronoi diagram of these points has n line segments which meet at the center of the circle. The sum of s-angle of the extreme flap polygon is zero.

We move one s-vertex to the interior of the circle as follows : let c be the center of the circle and s_{i-1}, s_i, s_{i+1} be three consecutive s-vertices on the circle. We move s_i somewhere inside the triangle (c, s_{i-1}, s_{i+1}). We place the v-vertices on the corresponding Voronoi segments and build the flap polygon by joining the s- and v-vertices alternately. Let the vertices of su-polygon be $s_1, v_1, s_2, v_2, \ldots, s_i, v_i, \ldots, s_n, v_n$. We move all v-vertices to the center of the circle except v_{i-1} and v_i. The lines $s_{i-1}c$ and $s_{i+1}c$ will intersect the Voronoi segments corresponding to v_{i-1} and v_i respectively. Finally, we place v_{i-1} and v_i on those intersection points respectively. See Fig 4.8(b). At this configuration, the s-angles at all s-vertices except s_i are zero. Therefore, the sum of the s-angles at s-vertices is less than 2π and all v-vertices except v_{i-1} and



(a) (b) (c)

Figure 4.8: (a) Six s-vertices on the circle. The Voronoi segments meet at the center of the circle. (b) A flap polygon where all s-angles but one (at s_i) are zero. (c) An su-polygon obtained from the given s-vertices.

v_i are on the extension of their corresponding Voronoi segments. We now move the v-vertices, placed on the center of the circle, outwards by enough amount such that the sum of the s-angles become 2π. This results in a geodesic star unfolding polygon. Based on the construction described here, we have the following result:

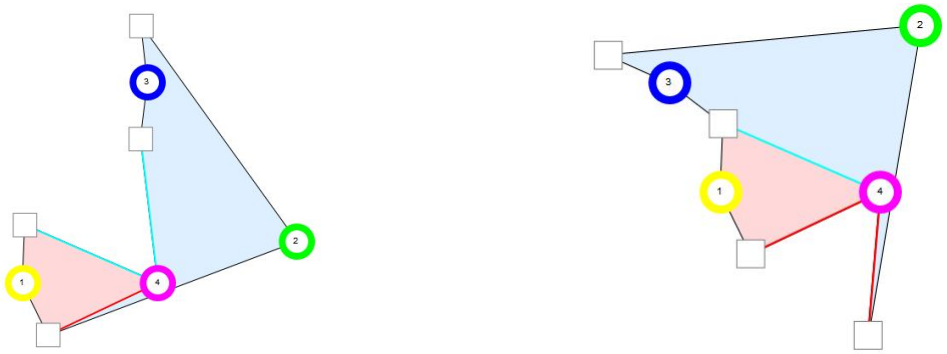
Theorem 33 *We can always construct a geodesic star unfolding where all but one s-vertices are on a circle.*

4.7 Conversion to sp-star unfolding

In this section, we address the following problem: given a geodesic star unfolding of a convex polyhedron, how can we convert it into the sp-star unfolding of the same polyhedron? To address this problem, we introduce a new operation on geodesic star unfolding polygons called *cut and paste* operation.

4.7.1 Cut and paste operation

Let S be an *su-polygon* with v-vertices $v_1, \ldots, v_n$ and s-vertices $s_1, \ldots, s_n$. Since S is a polygon, it has at least $(n-3)$ diagonals, fully contained in the interior of the polygon. We are not interested in all diagonals, only in those diagonals that connect



(a) (b)

Figure 4.9: (a) An su-polygon where circled and squared vertices are v- and s-vertices respectively. The cut-paste operation will be applied on the red diagonal. The polygon will be cut along the red diagonal and pasted along the two cut edges emanating from the v_2. (b) New su-polygon after the application of the cut-paste operation. Blue diagonal was the old cut edge.

s-vertices to v-vertices. We call those *v-s diagonals* or *diagonals* in short. Let d be one such diagonal that connects v_i to s_j. The v-vertex v_i is incident on the cut edges from s_i and s_{i+1}. If we cut along the diagonal d and paste along $v_i s_i$ and $v_i s_{i+1}$, we will obtain a new su-polygon with the same polyhedral metric. We call this *cut-paste operation*. See Fig 4.9.

Each cut-paste operation replaces the current geodesic cut edge with a new geodesic cut edge. If the diagonal d is longer than the cut edge the v-vertex is currently on, the cut-paste operation will yield an su-polygon where the length of the corresponding new cut edge will increase. This means that the perimeter of the new su-polygon will increase. On the other hand, if d is shorter than the current cut-edge, then, after the cut-paste operation, the corresponding v-vertex will be on a shorter cut edge, meaning the perimeter of the su-polygon will decrease. In the former case, d is called a *negative-gain diagonal*, whereas, in latter case, it is called a *positive-gain diagonal*. It should be noted that there is no positive gain diagonal in an sp-star unfolding polygon.

Each application of the cut-paste operation on a positive gain diagonal will replace the current geodesic with a shorter one. Let S_1 be the initial su-polygon of a fixed convex polyhedron and a source vertex and S_s be the sp-star unfolding of that polyhedron and source vertex. We argue that we can reach from S_1 to S_s after finite number of application of the cut-paste operation, assuming that there exists at least one positive gain diagonal at every step. If there are multiple positive gain diagonals at any step, we choose the one with the highest gain (the difference between the length of the diagonal and the length of the cut edge it replaces). We have the following proof:

Lemma 34 *We can obtain the sp-star polygon of a convex polyhedron with respect to a source vertex from an su-polygon S of the same polyhedron and the source vertex using cut-paste operation in finite number of steps, provided that there exists at least one positive gain diagonal at every step.*

Proof If S is the sp-star unfolding polygon, then we are done. Otherwise, Let $v_1, v_2, \ldots, v_k$ are the v-vertices of S which are not incident on shortest paths. Let gl_i and sl_i denote the lengths of the current geodesic path and the actual shortest path of v_i from the source vertex, respectively. There exist only finite number of geodesic paths from the source vertex to v_i with lengths in between gl_i and sl_i. (if there are infinite geodesic paths, the lengths of those geodesic paths extend to infinity contradicting our condition). Let this number be m_i. Since we perform the cut-paste operation only on positive gain diagonals, every time we perform such operation on v_i, we obtain a geodesic cut edge with shorter length. Therefore, after at most m_i such operations on v_i, it will be on the shortest path. Similar arguement holds for other v-vertices which are on non-shortest geodesic paths. ∎

We use an experimental method to verify that there exists at least one positive gain diagonal at each step until we obtain the sp-star unfolding. Our experimental

result shows that we can move from the given su-polygon to the sp-star unfolding using only positive gain diagonals.

4.7.2 Experimental setup

We conduct experimentation to answer the following two questions:

1. Starting from a geodesic star unfolding S_1 of a convex polyhedron and a source point, can we obtain the sp-star unfolding of the same polyhedron and the source vertex by repeatedly applying the cut-paste operation on positive gain diagonals?

2. If there exists multiple positive gain diagonals, does the choice of the diagonals affect the process? For example, if we choose the highest gain diagonals at every step, does it lead us to the sp-star unfolding faster?

We conduct experiments on large number of geodesic star unfolding polygons. For each such polygon, we apply cut-paste operation using both highest positive gain and lowest positive gain diagonals. When there exists no positive gain diagonal, we check if the final polygon is the sp-star unfolding (using Voronoi diagram property). We also keep note of the number of steps required to reach the sp-star unfolding using both types of positive gain diagonals.

4.7.2.1 Tools used

We have used three tools, all written in Mathematica, to conduct the experiments.

The first program is written to perform the cut-paste opertion on an su-polygon. It takes an su-polygon as input and compute all v-s diagonals. It classifies these diagonals into positive gain and negative gain diagonals and sort them in ascending order, within these classes. Based on the chosen diagonal, the program performs the cut-paste operation and computes a new geodesic star unfolding polygon.

The second program computes the su-polygon from some input s-vertices. Given a cyclically ordered s-vertices, the program computes the flap polygon (an su-polygon where the sum of the s-angles is not 2π). The user then can move the v-vertices along the corresponding Voronoi edges (on the segments and their extensions). This way the user can manually generate an su-polygon, even with s-vertices in non-convex position.

The third program is the implementation of the algorithm to generate su-polygons from co-circular point sets presented in Section 4.6.2. It takes a set of points on the circle and compute the su-polygon.

4.7.2.2 Generation of input su-polygons

To conduct the experiments, we have generated 3500 su-polygons with 10,15 and 20 v-vertices (or s-vertices) in three different ways:

First, we used the second program stated in Section 4.7.2.1 to manually generate su-polygons.

Second, we used the third program to generate su-polygons with all s-vertices but one in co-circular positions.

Third, we first generated the sp-star unfolding polygons with the algorithm and tools discussed in Chapter 3. Then we repeatedly apply the cut-paste operation on each of these sp-star unfoldings with randomly chosen *negative gain* diagonals. Because of the negative gain diagonals, some or all of the shortest paths in the sp-star unfolding will be replaced by longer geodesics. As a result, we will obtain geodesic unfolding polygons from sp-star unfolding polygons.

4.7.2.3 Experimental results

After conducting the experiments, we observe the following results:

1. For every input su-polygon, there exists at least one positive gain diagonal at every step.

2. Irrespective of our choice of the highest or the lowest positive gain diagonals, we can always obtain the sp-star unfolding from each of the input su-polygon.

3. There is no guarantee that the highest positive gain diagonals will lead us to the sp-star unfolding polygons faster. Sometimes, we reach to the sp-star unfolding in a smaller number of steps using only the lowest gain positive diagonals.

The immediate consequence of these observations leads us to a simple algorithm to compute the sp-star unfolding from the geodesic star unfolding.

4.7.2.4 Algorithm

Given an su-polygon, we compute the v-s diagonals and select the diagonal with the highest positive gain. We cut along this diagonal and paste along the cut edges the corresponding v-vertex is on. We keep performing this operation, until there is no diagonal with positive gain. The final polygon is the sp-star unfolding.

4.8 Conclusion

We have introduced a new kind of unfolding called geodesic star unfolding. We have examined its properties and found that it has some unique properties. We have discussed some challenges to construct the geodesic star unfolding from a given point set and presented an algorithm to construct the geodesic star unfolding from all but one co-circular s-vertices. Finally, we have introduced the cut and paste operation. We have conducted experiments which show that we can obtain the sp-star unfolding from any given geodesic star unfolding using this operation. We have given heuristics on how we can choose diagonals for the cut and paste operation and find the performance of the algorithm based on these heuristics.

CHAPTER 5

CONCLUSIONS AND OPEN PROBLEMS

In this dissertation, we have addressed several problems for two well-known geometric structures: Delaunay tessellation and the shortest path star unfolding. We have also introduced a new structure, called geodesic star unfolding, and investigated several problems on this structure. In the rest of this chapter, we summarize our results and state some related open problems.

5.1 Realization of Delaunay triangulations

We have considered the decision problem if an outerplanar graph is realizable as a Delaunay triangulation. We have used a new proof technique, utilizing the criterion of inscribable graphs set by Rivin [44, 45], to show that any outerplanar graph can be realized as a Delaunay tessellation. This inscribability criterion requires a set of weights on the edges of the graph such that they satisfy some conditions. We have shown, inductively, how we can assign weights to the edges of an outerplanar graph while maintaining these conditions.

Although we know certain classes of planar graphs are Delaunay realizable, the combinatorial characterization of all Delaunay realizable planar graphs is still unknown. Therefore, we have the following problem:

Problem 4 *Give a complete combinatorial characterization of the planar graphs which are Delaunay realizable.*

5.2 Ridge tree reconstruction

We have shown that every combinatorial tree can be embedded as a ridge tree of some convex polyhedron with respect to some source vertex. We have presented a complete characterization of the polygons that arise as the shortest path star unfoldings. In the process, we have addressed several problems concerning the realization of the sp-star unfolding polygons on a specified point set. We have shown, using counter examples, that arbitrary set of points cannot be used to construct this unfolding. We have connected Dillencourt's algorithm for Delaunay realization to the realization problem of the sp-star unfolding and its underlying ridge tree. Our reconstruction algorithm, however, puts the source images in convex position. Since the source images of an sp-star unfolding are not always placed in convex position, developing an algorithm that puts the source images in arbitrary position for any combinatorial ridge tree will address the reconstruction problem for the whole class of the sp-star unfolding. An open problem is the following:

Problem 5 *Given a tree, can we realize it as the ridge tree of an sp-star unfolding where the source images are not necessarily in convex position?*

5.3 Geodesic star unfolding

Being inspired by the sp-star unfolding, we have introduced a new type of unfolding called geodesic star unfolding. In this unfolding, we take a set of non-crossing geodesic paths, one for each polyhedral vertex, from the source vertex to the vertices of the convex polyhedron. All of these paths are not necessarily the shortest. The resulting unfolding is called a geodesic star unfolding. We have found that this unfolding has several unique properties: a geodesic star unfolding polygon may be self-overlapping, its polyhedral vertices are not necessarily placed exactly on the corresponding Voronoi segments and its core may be self-intersecting. We have also considered the reconstruction problem for geodesic star unfolding. We discussed some

challenges of this problem and presented a reconstruction algorithm for a small class of s-vertices. Finally, we introduced a new operation, called cut-paste operation. This operation cuts an su-polygon along a diagonal joining a v-vertex to an s-vertex and pastes along the current cut edges of this v-vertex. The result is a new su-polygon. We have shown, using experimental data, that using this operation on geodesic star unfoldings, we can always obtain the sp-star unfolding of the same convex polyhedron and the source vertex. However, we would like to see our experimental result being backed by a theoretical proof. Therefore we have the following open problem:

Problem 6 *Does there always exist a diagonal connecting a polyhedral vertex to a source image and lying entirely inside the geodesic star unfolding polygon such that its length is smaller than the length of the cut edge of the corresponding v-vertex?*

We believe a solution to this problem requires the exploration of the properties of geodesic paths on the polyhedral surface. Therefore, we find that some of the following problems are relevant:

Problem 7 *What is the bound on the number of geodesic paths between two points on the convex polyhedral surface whose lengths are equal or smaller than l, $l \in \mathbb{R}$?*

Problem 8 *What are the necessary and sufficient conditions for two points on convex polyhedral surface to have infinite number of geodesic paths?*

BIBLIOGRAPHY

[1] Agarwal, Pankaj K., Aronov, Boris, O'Rourke, Joseph, and Schevon, Catherine A. Star unfolding of a polytope with applications. *SIAM J. Computing 26* (1997), 1689–1713.

[2] Agarwal, Pankaj K., Guibas, Leonidas J., Saxe, James B., and Shor, Peter W. A linear-time algorithm for computing the voronoi diagram of a convex polygon. *Discrete Computational Geometry 4*, 1 (1989), 591–604.

[3] Agarwal, Pankaj K., Har-Peled, Sariel, and Karia, Meetesh. Computing approximate shortest paths on convex polytopes. In *Proceedings of the Sixteenth Annual Symposium on Computational Geometry* (New York, NY, USA, 2000), SCG '00, ACM, pp. 270–279.

[4] Aleksandrov, Lyudmil, Lanthier, Mark, Maheshwari, Anil, and Sack, Jrg-R. An approximation algorithm for weighted shortest paths on polyhedral surfaces. In *Algorithm Theory SWAT'98*, Stefan Arnborg and Lars Ivansson, Eds., vol. 1432 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1998, pp. 11–22.

[5] Alexander, R, and Rowe, N. Path planning by optimal-path-map construction for homogeneous-cost two-dimensional regions. In *IEEE International Conferance on Robotics and Automation* (1990).

[6] Alexandrov, Aleksandr Danilovich. *Convex Polyhedra*. Springer Monographs in Mathematics. Springer Verlag, Berlin Heidelberg, 2005. English Translation of Russian edition, Gosudarstv. Izdat.Tekhn.-Teor.Lit., Moscow-Leningrad, 1950.

[7] Alexandrov, Aleksandr Danilovich. *Intrinsic Geometry of Convex Surfaces*. Chapman and Hill/CRC, Taylor and Francis Group, Boca Raton, Florida, 2006. Selected Works II, translated from Russian by S. S. Kutateladze.

[8] Aronov, Boris, and O'Rourke, Joseph. Non-overlap of the star unfolding. *Discrete and Computational Geometry 8* (1992), 219–250.

[9] Bern, Marshall Wayne, and Eppstein, David. Mesh generation and optimal triangulation. In *Computing in Euclidean Geometry*, Ding-Zhu Du and Frank Kwang-Ming Hwang, Eds., no. 1 in Lecture Notes Series on Computing. World Scientific, 1992, pp. 23–90.

[10] Biedl, Therese C., Demaine, Erik D., Demaine, Martin L., Lubiw, Anna, Overmars, Mark, O'Rourke, Joseph, Robbins, Steven M., and Whitesides, Sue. Unfolding some classes of orthogonal polyhedra. In *Proceedings of the 10th Canadian Conference on Computational Geometry (CCCG'98)* (Montréal, Québec, Canada, 1998).

[11] Bobenko, AlexanderI., and Springborn, BorisA. A discrete laplacebeltrami operator for simplicial surfaces. *Discrete & Computational Geometry 38*, 4 (2007), 740–756.

[12] Bobenko, Alexander I., Izmestiev Ivan. Alexandrovs theorem, weighted delaunay triangulations, and mixed volumes. *Annales de linstitut Fourier 58*, 2 (2008), 447–505.

[13] Canny, John. A new algebraic method for robot motion planning and real geometry. In *Proc IEEE FOCS* (1987), pp. 39–48.

[14] Canny, John, and Reif, J. New lower bound techniques for robot motion planning problems. In *Proc. 28th IEEE Symp. Found. Comput. Sci.* (1987), IEEE Computer Society Press, pp. 49–60.

[15] Chen, Jindong, and Han, Yijie. Shortest paths on a polyhedron. *Int. J. Comput. Geom. Appl 6* (1996), 127–144.

[16] Cook, AtlasF., IV, and Wenk, Carola. Shortest path problems on a polyhedral surface. *Algorithmica 69*, 1 (2014), 58–77.

[17] Coxeter, Harold Scott McDonald. *Regular Polytopes*. Macmillan, New York, 1963.

[18] Demaine, Erik D., and O'Rourke, Joseph. *Geometric Folding Algorithms: Linkages, Origami, and Polyhedra*. Cambridge University Press, 2007.

[19] Dillencourt, Michael B. Realizability of delaunay triangulations. *Information Processing Letters 33*, 6 (1990), 283–287.

[20] Dillencourt, Michael B. Toughness and delaunay triangulations. *Discrete & Computational Geometry 5*, 1 (1990), 575–601.

[21] Dillencourt, Michael B., and Smith, Warren D. Graph-theoretic conditions for inscribability and delaunay realizability. *Discrete Mathematics 161*, 1-3 (1996), 63–77.

[22] Du, Qiang, Faber, Vance, and Gunzburger, Max. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Rev. 41*, 4 (Dec. 1999), 637–676.

[23] Du, Qiang, and Wang, Desheng. Boundary recovery for three dimensional conforming delaunay triangulation. *Computer Methods in Applied Mechanics and Engineering 193*, 2326 (2004), 2547 – 2563.

[24] Dürer, Albrecht. Unterweysung der messung mit dem zyrkel und rychtscheyd, 1525. English translation with commentary by Walter L. Strauss The Painter's Manual, New York (1977).

[25] Edelsbrunner, H., Kirkpatrick, D. G., and Seidel, R. On the shape of a set of points in the plane. 551–558.

[26] Edelsbrunner, Herbert, Ablowitz, M. J., Davis, S. H., Hinch, E. J., Iserles, A., Ockendon, J., and Olver, P. J. *Geometry and Topology for Mesh Generation (Cambridge Monographs on Applied and Computational Mathematics)*. Cambridge University Press, New York, NY, USA, 2006.

[27] Gabriel, K. Ruben, and Sokal, Robert R. A new statistical approach to geographic variation analysis. *Systematic Zoology 18*, 3 (1969), pp. 259–278.

[28] George, P.L., Hecht, F., and Saltel, E. Automatic mesh generator with specified boundary. *Computer Methods in Applied Mechanics and Engineering 92*, 3 (1991), 269 – 288.

[29] Grünbaum, Branko. On steinitz's theorem about non-inscribable polyhedra. *Ned. Akad. Wetenschap. Proc. Ser. A 66* (1964), 452–455.

[30] Grünbaum, Branko. *Convex Polytopes*. John Wiley and Sons, 1967.

[31] Har-Peled, Sariel. Approximate shortest paths and geodesic diameters on convex polytopes in three dimensions. In *Proceedings of the Thirteenth Annual Symposium on Computational Geometry* (New York, NY, USA, 1997), SCG '97, ACM, pp. 359–365.

[32] Har-Peled, Sariel. Constructing approximate shortest path maps in three dimensions. In *Proceedings of the Fourteenth Annual Symposium on Computational Geometry* (New York, NY, USA, 1998), SCG '98, ACM, pp. 383–391.

[33] Hodgson, Craig D., Rivin, Igor, and Smith, Warren D. A characterization of convex hyperbolic polyhedra and of convex polyhedra inscribed in the sphere. *Bulletin of the AMS 27*, 2 (October 1992), 246–251.

[34] Lambert, Timothy. An optimal algorithm for realizing a delaunay triangulation. *Information Processing Letters 62*, 5 (1997), 245–250.

[35] Miller, Ezra, and Pak, Igor. Metric combinatorics of convex polyhedra: Cut loci and nonoverlapping unfoldings. *Discrete Comput. Geom. 39*, 1 (2008), 339–388.

[36] Mitchell, Joseph S. B. Shortest paths among obtacles in the plane. In *Proc 9th Annual ACM Sympos. Comput. Geom.* (1993), pp. 308–317.

[37] Mitchell, Joseph S. B., Mount, David, and Papadimitriou, Christos. The discrete geodesic problem. *SIAM Journal on Computing 16*, 4 (1987), 647–668.

[38] Mitchell, Joseph S.B. An algorithmic approach to some problems in terrain navigation. *Artificial Intelligence 37*, 13 (1988), 171 – 201.

[39] Mount, David. On finding shortest path on convex polyhedra. Tech. Rep. 1495, University of Maryland, 1985.

[40] Plantinga, Simon, and Vegter, Gert. Isotopic meshing of implicit surfaces. *The Visual Computer 23*, 1 (2007), 45–58.

[41] Preparata, Franco P., and Shamos, Michael I. *Computational Geometry: An Introduction*. Springer-Verlag New York, Inc., New York, NY, USA, 1985.

[42] Reif, J., and Storer, J. 3-dimensional shortest paths in presence of polyhedral obstacles. In *Proc Foundation of Computer Sci* (1988), pp. 85–92.

[43] Richbourg, R., Rowe, N.C., Zyda, M.J., and McGhee, R.B. Solving global two-dimensional routing problems using snell's law and a search. In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on* (Mar 1987), vol. 4, pp. 1631–1636.

[44] Rivin, Igor. Euclidean structures on simplicial surfaces and hyperbolic volume. *Annals of Mathematics 139*, 3 (May 1994), 553–580.

[45] Rivin, Igor. A characterization of ideal polyhedra in hyperbolic 3-space. *The Annals of Mathematics 143*, 1 (January 1996), 51–70.

[46] Schreiber, Yevgeny, and Sharir, Micha. An optimal-time algorithm for shortest paths on a convex polytope in three dimensions. *Discrete and Computational Geometry 39* (2008), 500–579.

[47] Sharir, Micha, and Schorr, Amir. On shortest path in polyhedral spaces. *SIAM Journal on Computing 15*, 1 (February 1986), 193–215.

[48] Toussaint, Godfried T. The relative neighbourhood graph of a finite planar set. *Pattern Recognition 12*, 4 (1980), 261 – 268.

[49] Welzl, Emo. Constructing the visibility graph for n-line segments in $o(n^2)$ time. *Information Processing Letters 20* (1985), 167–171.

[50] Ziegler, Günter M., and Gonska, Bernd. Inscribable stacked polytopes. *Advances in Geometry 13*, 4 (October 2013), 723–740.