STUDIES OF SEVERAL TETRAHEDRALIZATION PROBLEMS

BOTING YANG

Your file Votre référence
ISBN: 0-612-89705-2
Our file Notre référence
ISBN: 0-612-89705-2

Studies of Several Tetrahedralization Problems

by

Boting Yang

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

Doctor of Philosophy

Department of Computer Science
Memorial University of Newfoundland

2002

St. John's Newfoundland

Abstract

The main purpose of decomposing an object into simpler components is to simplify a problem involving the complex object into a number of subproblems having simpler components. In particular, a *tetrahedralization* is a partition of the input domain in R^3 into a number of tetrahedra that meet only at shared faces. Tetrahedralizations have applications in the finite element method, mesh generation, computer graphics, and robotics.

This thesis investigates four problems in tetrahedralizations and triangulations. The first problem is on the computational complexity of tetrahedralization detections. We present an $O(nm \log n)$ algorithm to determine whether a set of line segments $\mathcal{L}$ is the edge set of a tetrahedralization, where m is the number of segments and n is the number of endpoints in $\mathcal{L}$. We show that it is NP-complete to decide whether $\mathcal{L}$ contains the edge set of a tetrahedralization. We also show that it is NP-complete to decide whether $\mathcal{L}$ is tetrahedralizable. The second problem is on minimal tetrahedralizations. After deriving some properties of the graph of polyhedra, we identify a class

of polyhedra and show that this class of polyhedra can be minimally tetrahedralized in $O(n^2)$ time. The third problem is on the tetrahedralization of two nested convex polyhedra. We give a method to tetrahedralize the region between two nested convex polyhedra into a linear number of tetrahedra without introducing Steiner points. This result answers an open problem raised by Bern [16]. The fourth problem is on the lower bound for β-skeletons belonging to minimum weight triangulations. We prove a lower bound on β ($\beta = \frac{1}{6}\sqrt{2\sqrt{3}+45}$) such that if β is less than this value, the β-skeleton of a point set may not always be a subgraph of the minimum weight triangulation of this point set. This result settles Keil's conjecture [62].

Acknowledgments

My deepest thanks go to Cao An Wang, my supervisor, for years of guidance, support, and encouragement. He has been a source of excellent ideas and advice. I have greatly benefitted from my fruitful discussions with him.

I would also like to thank Paul Gillard, Manrique Mata-Montero, Todd Wareham and Xiaobu Yuan for their advice and support. Special thanks go to Paul Gillard, Todd Wareham, Donald Craig, Elaine Boone and Iona Bulgin for providing many helpful comments and corrections on various drafts of this work.

I am also grateful to the School of Graduate Studies and the Department of Computer Science at Memorial University of Newfoundland for providing me with a fellowship and other financial support. Thanks also go to the graduate students in the Department of Computer Science for making my graduate student experience very pleasurable.

Finally, I would like to thank my wife Zhe Su for her love, patience, and constant support; this work is dedicated to her.

Contents

List of Figures

Chapter 1

Introduction

1.1 General definitions

In this section, we introduce some definitions of the geometric objects that are used in this thesis.

Euclidean distance: Let R^d denote the d-dimensional Euclidean space, and $p = (p_1, p_2, ..., p_d)$ and $q = (q_1, q_2, ..., q_d)$ be two points in R^d. The *Euclidean distance* between p and q, denoted as $\text{dist}(p, q)$, is measured by the L_2-metric:

$$\text{dist}(p, q) = (\sum_{i=1}^{d} |p_i - q_i|^2)^{\frac{1}{2}}.$$

The *Euclidean length* of the straight line segment (edge) joining p and q, denoted as

$|pq|$, is equal to $\text{dist}(p, q)$. In this thesis, we will use *distance (length)* as a shorthand for the Euclidean distance (length).

Convex hull: A domain D in R^d is *convex* if and only if, for any pair of points p and q in D, the line segment pq is completely contained in D. The *convex hull* of a set of geometric objects S in R^d, denoted by $CH(S)$, is the smallest convex set in R^d containing S. Since the intersection of convex sets is a convex set, $CH(S)$ is the intersection of all convex sets that contain S. Note that the term convex hull is used in this thesis to denote the union of the boundary and the interior.

Polygon: In R^2, a *polygon* is defined by a finite set of line segments such that each endpoint of a segment is shared by exactly two segments. The segments are the *edges* and their endpoints are *vertices* of the polygon.

A polygon is *simple* if any pair of nonconsecutive edges do not share a point. A simple polygon is homeomorphic to a closed disc, whose boundary partitions the plane into two disjoint areas. The bounded area is called the *interior*, and the unbounded area is called the *exterior* of the polygon.

Similarly, we can define the polygon in a piecewise linear surface in R^3, such as the surface of a polyhedron defined as follows.

Polyhedron: In R^3, a *polyhedron* is defined by a finite set of plane polygons such that each edge of the polygons is shared by exactly two polygons. These polygons and their

edges and vertices are the $2, 1, 0$-dimensional *faces* of the polyhedron, respectively. In particular, the vertices and edges of the polygons are the *vertices* and *edges* of the polyhedron respectively; the polygons are the *facets* of the polyhedron.

A polyhedron is *simple* if it is homeomorphic to a closed 3-ball. Therefore, a simple polyhedron does not meet itself in a handle, or touch itself at a point or an edge (see Figure 1.1). The boundary of such a polyhedron is a piecewise linear 3-manifold which forms a connected planar graph. A simple polyhedron partitions the space into two disjoint regions. The bounded region is called the *interior*, and the unbounded region is called the *exterior* of the polyhedron. A simple polyhedron is *simplicial* if all of its facets are triangles.

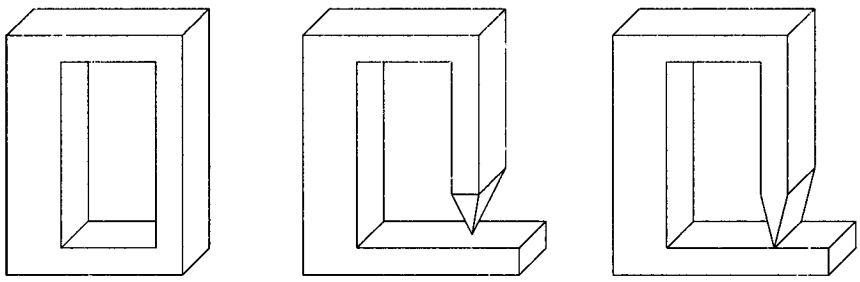

Figure 1.1: Three examples of non-simple polyhedra.

Note that when we use the term polygon or polyhedron, we usually mean the union of the boundary and the interior.

A *diagonal* of a polyhedron (or polygon) is a line segment between two vertices that lies inside the polyhedron (or polygon) and does not intersect the boundary of the polyhedron (or polygon) except at its endpoints.

A polyhedron (or polygon) is *star-shaped* if the entire polyhedron (or polygon) is visible from some point inside the polyhedron (or polygon).

Tetrahedralization/Triangulation: A *tetrahedralization* (resp. *triangulation*) is a partition of the input domain in R^3 (resp. R^2) into a number of tetrahedra (resp. triangles) that meet only at shared faces (see Figure 1.2).

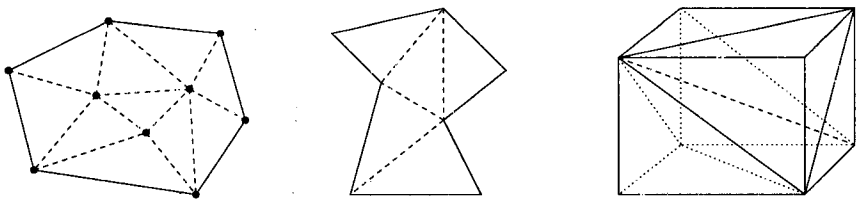

Figure 1.2: The triangulations of a point set (left) and a simple polygon (middle). The tetrahedralization of a polyhedron (right). The solid and dotted lines show input; the dashed lines show the added edges.

In R^3, let S be a piecewise linear 3-manifold in which each facet is a triangle. If S is topologically equivalent to a disc, then S is called a *triangulation surface.* If S is topologically equivalent to a 3-sphere, then S is called a *closed triangulation surface.*

A tetrahedralization is significantly more complicated than a triangulation. While different triangulations of the same input must contain the same number of triangles, different tetrahedralizations of the same input may contain different numbers of tetrahedra (see Figure 1.3). There even exists a convex n-vertex polyhedron that can be tetrahedralized with $\binom{n-2}{2}$ tetrahedra. The equation $t = e_i + n - 3$ [39] describes the relationship between the number of tetrahedra t and the number of interior edges e_i in a tetrahedralization of an n-vertex polyhedron.

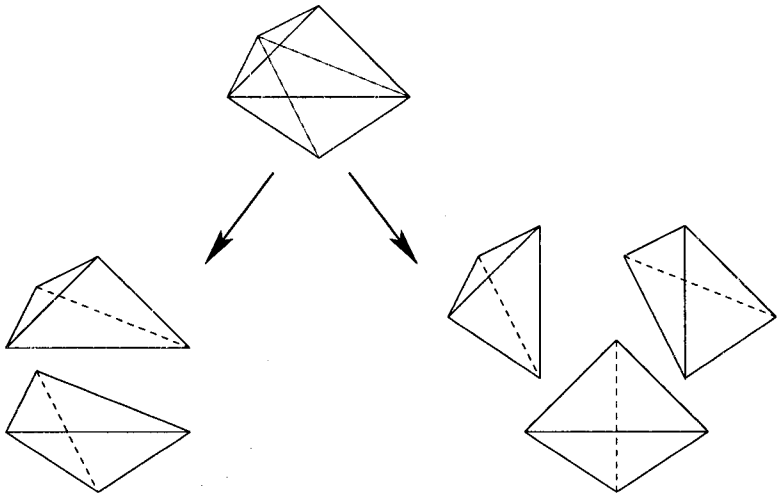

Figure 1.3: Two ways of tetrahedralizing a polyhedron.

Note that the word *tetrahedralization* we used in this thesis has a variety of names in the literature, such as triangulation [26] and tetrahedrization [48]. The word *triangulation* usually refers to the 2-dimensional case or the d-dimensional case [89]. The word *tetrahedralization* is commonly used for the 3-dimensional case [16, 27].

Tetrahedralizations and triangulations have applications in the finite element method, CAD/CAM, computer graphics, and robotics [6, 17, 22, 48, 82].

1.2 The computational model and complexity

Before we can analyze algorithms, we must have a computational model. In this thesis, the computational model for all the algorithms is the *real random access machine* (RAM) [1, 82]. In this model each of the following primitive operations can be performed in a unit cost:

1. The four arithmetic operations $(+, -, \times, \div)$.

2. Comparisons between two real numbers $(<, \leq, =, \neq, \geq, >)$.

3. Indirect addressing of memory (integer addresses only).

Given a geometric problem, each instance of the problem is specified by a set of data called the *input* to the problem. For a given input, an algorithm that solves the problem under the input yields a result called the *output.* The *size* of the input (resp. output) equals the number of memory cells needed to store this input (resp. output). For geometric problems, this is usually the number of points and segments in the input or output. The *running time* of an algorithm on a given input is the number of primitive operations executed in solving the input. Obviously, the running time of an algorithm depends not only on the size of the input, but also on the input itself, e.g., how points and segments are placed in space and what are the lengths of segments. For an algorithm, we usually concentrate on its *worst-case time complexity*, or time complexity *in the worst case*, that is, a function $t(n)$ that represents an asymptotic upper bound on the number of primitive operations executed by the algorithm when the input size is n. Similarly, we can define the *worst-case space complexity.* In this thesis, we will use *complexity* as a shorthand for the worst-case time (or space) complexity. When an asymptotic upper bound to the worst-case time complexity of an algorithm is $t(n)$, we also say that the algorithm runs in time $t(n)$.

1.3 Basic notations

In this section, we introduce some basic notations that are used throughout this thesis.

For a set A of geometrical objects (points, edges, facets, or polyhedra) in the d-dimensional Euclidean space R^d, let $V(A)$, $E(A)$ and $F(A)$ be the vertex set, edge set and facet set of A, respectively. Similarly, let $\text{int}(A)$, $\text{cl}(A)$ and $CH(A)$ be the interior, closure and convex hull of A, respectively. Let $|A|$ be the cardinality of A if A is a set according to the context. Note that the Euclidean length of a straight line segment (edge) pq is also denoted as $|pq|$.

Throughout this thesis, every set of geometrical objects (points, edges, facets, or polyhedra) is a finite set unless otherwise stated.

Given arbitrary points a, b, c, d in R^2 or R^3, unless otherwise stated, we use ab to denote the line segment (edge) with endpoints a and b in R^2 or R^3, abc to denote the triangle with vertices a, b and c in R^2 or R^3, and $abcd$ to denote the tetrahedron with vertices a, b, c and d in R^3.

We will compare the growth of different functions by using the notations introduced by Knuth [66]. Let f and g be two positive real-valued functions of the integer-valued variable n.

$$O(g(n)) = \{f(n) \quad | \quad \text{there exist positive constants } c \text{ and } n_0 \text{ such that}$$

$$f(n) \leq cg(n) \text{ for all } n \geq n_0\}.$$

$$\Omega(g(n)) = \{f(n) \mid \text{there exist positive constants } c \text{ and } n_0 \text{ such that}$$

$$cg(n) \leq f(n) \text{ for all } n \geq n_0\}.$$

$$\Theta(g(n)) = \{f(n) \mid \text{there exist positive constants } c_1, c_2 \text{ and } n_0 \text{ such that}$$

$$c_1 g(n) \leq f(n) \leq c_2 g(n) \text{ for all } n \geq n_0\}.$$

In particular, a function $f(n)$ is $O(1)$ if and only if it is bounded above by a constant.

1.4 Outline of the thesis

The rest of this thesis is organized as follows.

In Chapter 2, we investigate the computational complexity of the tetrahedralization detection problem. Let $\mathcal{L}$ be a set of line segments in three dimensional Euclidean space. We prove several characterizations of the tetrahedralizations. We present an $O(nm \log n)$ algorithm to determine whether $\mathcal{L}$ is the edge set of a tetrahedralization, where m is the number of segments and n is the number of endpoints in $\mathcal{L}$. We show that it is NP-complete to decide whether $\mathcal{L}$ contains the edge set of a tetrahedralization as a subset. We also show that it is NP-complete to decide whether $\mathcal{L}$ is tetrahedralizable. The results from this chapter are due to joint work with C. A. Wang.

In Chapter 3, we present some properties of the graph of polyhedron. We identify a class of polyhedra and show that this class of polyhedra can be minimally tetrahedralized in $O(n^2)$ time. The results from this chapter also are due to joint work with C. A. Wang, and have appeared as [110].

In Chapter 4, we present a method to tetrahedralize the region between two nested convex polyhedra without introducing Steiner points. This method produces at most $9n - 6$ tetrahedra, where n is the number of the vertices in the two given polyhedra. Thus, we answer the open problem as to whether the region between two nested convex polyhedra can be tetrahedralized into a linear number of tetrahedra without Steiner points [16]. The results from this chapter are also a joint work with C. A. Wang, and have appeared in [107].

In Chapter 5, we study the relationship between β-skeletons and minimum weight triangulations. We prove a lower bound on β value ($\beta = \frac{1}{6}\sqrt{2\sqrt{3} + 45}$) such that if β is less than this value, the β-skeleton of a point set may not be always a subgraph of the minimum weight triangulation of the point set. Thus, we disprove Keil's conjecture that, for $\beta = \frac{2}{3}\sqrt{3}$, the β-skeleton is a subgraph of the minimum weight triangulation [62]. The results from this chapter are also a joint work with C. A. Wang; it appeared originally in [105], and in final form in [108].

Chapter 2

Detections of the Tetrahedralizations

Let $\mathcal{L}$ be a set of line segments in three dimensional Euclidean space, and m be the number of segments and n be the number of endpoints in $\mathcal{L}$. In Section 2.1, we review the literature regarding tetrahedralizations. In Section 2.2, we give some notations and assumptions. In Section 2.3, we present three characterizations of tetrahedralizations. They describe a tetrahedralization from the facet, vertex, and combinatorial viewpoints, respectively. In Section 2.4, we describe an $O(nm \log n)$ algorithm to determine whether $\mathcal{L}$ forms a tetrahedralization. In Section 2.5, we show that the problem of deciding if $\mathcal{L}$ contains a tetrahedralization is NP-complete by a reduction from the two dimensional analog of this problem [74]. In Section 2.6,

we prove that the problem of deciding if $\mathcal{L}$ is tetrahedralizable is NP-complete by a reduction from the Satisfiability problem [51]. Finally, we list some open problems in Section 2.7.

2.1 Introduction

Given a set V of points in R^3, a *tetrahedralization* of V is a partition of the convex hull of V into a number of tetrahedra such that (i) each vertex of the tetrahedra belongs to V, (ii) each point of V is the vertex of a tetrahedron, and (iii) the intersection of any two tetrahedra is either empty or a shared face.

Edelsbrunner *et al.* [48] studied the problem of tetrahedralizing a set of points. They presented several combinatorial results on extremum problems concerning the number of tetrahedra in a tetrahedralization. They also presented an algorithm that tetrahedralizes a set of n points in $O(n \log n)$ time. A similar algorithm was given in [6]. If additional vertices, the so-called *Steiner points*, are allowed when constructing the tetrahedralization, then any simple polyhedron of n vertices can be partitioned into $O(n^2)$ tetrahedra. Chazelle and Palios [26] described a tetrahedralization algorithm for decomposing a simple polyhedron with n vertices and r reflex edges into $O(n + r^2)$ tetrahedra using $O(n + r^2)$ Steiner points. Chazelle and Shouraboura [27] showed that this algorithm works just the same for polyhedra of arbitrary genus. Dey *et al.* [42] described an algorithm that constructs a tetrahedralization avoiding the

creation of flat or long and thin tetrahedra. Other algorithms for tetrahedralizing polyhedra with Steiner points are presented in [7, 41, 57]. Delaunay tetrahedralizations are discussed in [59, 60, 61, 83, 96, 97].

On the issue of NP-completeness, Ruppert and Seidel [88] proved that finding a tetrahedralization of a nonconvex polyhedron without Steiner points is NP-complete; this problem remains NP-complete even for star-shaped polyhedra. It follows that the problem of deciding how many Steiner points are needed to tetrahedralize a polyhedron is also NP-complete.

Given a set of line segments $\mathcal{L}$ in R^3, we investigate the algorithmic complexity of finding tetrahedralizations regarding $\mathcal{L}$. If $\mathcal{L}$ is the edge set of a tetrahedralization, we say that $\mathcal{L}$ *forms* a tetrahedralization. If $\mathcal{L}$ is a superset of the edge set of a tetrahedralization of $V(\mathcal{L})$, we say that $\mathcal{L}$ *contains* a tetrahedralization. If $\mathcal{L}$ is a subset of the edge set of a tetrahedralization of $V(\mathcal{L})$, we say that $\mathcal{L}$ is tetrahedralizable. We present an $O(nm \log n)$ algorithm to determine whether $\mathcal{L}$ forms a tetrahedralization, where m is the number of segments and n is the number of endpoints in $\mathcal{L}$. If $\mathcal{L}$ does not form a tetrahedralization, we prove that it is NP-complete to decide whether $\mathcal{L}$ contains a tetrahedralization of $V(\mathcal{L})$, and we prove that it is also NP-complete to decide whether $\mathcal{L}$ is tetrahedralizable.

2.2 Preliminaries

Definition 2.1 *For a face or polyhedron P and a line segment set L, if no segment of L intersects the interior of P, then we say that P is L-*`empty`*, as shown in Figure 2.1.*

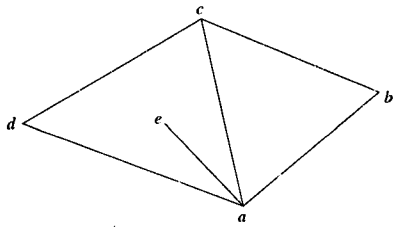

Figure 2.1: $L = \{ab, ac, ad, ae, bc, cd\}$; face abc is L-empty and face acd is not L-empty.

Throughout this chapter, let $\mathcal{L}$ be a set of m line segments with n endpoints in R^3; let $\Delta_{\mathcal{L}} = \{abc \mid ab, bc, ca \in \mathcal{L}, \text{int}(abc) \cap \mathcal{L} = \emptyset\}$ be the set of $\mathcal{L}$-empty triangles with edges in $\mathcal{L}$; and Δ_{CH} be the set of facets in $CH(\mathcal{L})$. Let $\Delta_I = \Delta_{\mathcal{L}} - \Delta_{CH}$.

For simplicity, we assume throughout this chapter that no four vertices of $V(\mathcal{L})$ are coplanar. This assumption implies that each facet of $CH(\mathcal{L})$ is a triangle and any pair of segments in $\mathcal{L}$ do not intersect.

2.3 Characterizations of tetrahedralizations

For each triangle $abc \in \Delta_I$, let abc^+ and abc^- denote its two oriented versions with respect to the opposite normal directions. For the plane H containing abc, let H^+ and H^- denote its two half spaces corresponding to abc^+ and abc^- respectively. Let

$n_t(abc^+)$ (resp. $n_t(abc^-)$) be the number of vertices in $V(\mathcal{L})$ that lie in H^+ (resp. H^-) and form $\mathcal{L}$-empty tetrahedra with abc.

The following theorem gives characterizations of a tetrahedralization from the facet viewpoint.

Theorem 2.1 (i) *If $\Delta_I = \emptyset$, then $\mathcal{L}$ forms a tetrahedralization if and only if $\mathcal{L}$ forms a tetrahedron.*

(ii) *If $\Delta_I \neq \emptyset$, then $\mathcal{L}$ forms a tetrahedralization if and only if $n_t(abc^+) = 1$ and $n_t(abc^-) = 1$, for each $abc \in \Delta_I$.*

Proof (i) Sufficiency is trivial. We only prove necessity. Suppose $\mathcal{L}$ forms a tetrahedralization. If $\mathcal{L}$ does not form a tetrahedron, then there are at least two tetrahedra in $\mathcal{L}$ which share one face. This face is an inner triangle. This contradicts the assumption that $\Delta_I = \emptyset$.

(ii) (Necessity). Suppose $\mathcal{L}$ forms a tetrahedralization. For each $abc \in \Delta_I$, consider $n_t(abc^+)$. Let H be the plane containing abc and H^+ be the half space corresponding to abc^+. If $n_t(abc^+) > 1$, let d and d' be two vertices of $V(\mathcal{L})$ which lie in H^+ and form the $\mathcal{L}$-empty tetrahedra $abcd$ and $abcd'$. Since $abcd$ and $abcd'$ are $\mathcal{L}$-empty tetrahedra, $\text{int}(abcd) \cap \{ad', bd', cd'\} = \emptyset$, and $\text{int}(abcd') \cap \{ad, bd, cd\} = \emptyset$ (see Figure 2.2). This is a contradiction. Hence, $n_t(abc^+) \leq 1$. Assume $n_t(abc^+) = 0$. Let e be an arbitrary interior point in abc. Since abc is an $\mathcal{L}$-empty inner face, there must exist a small

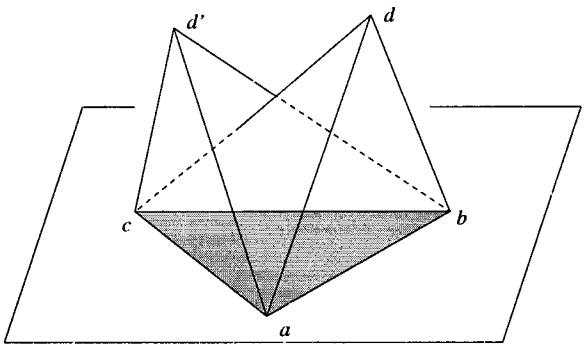

Figure 2.2: The interiors of the tetrahedra must intersect if $n_t(abc^+) > 1$.

open ball $O(e)$ such that $O(e) \cap abc \subset \text{int}(abc)$ and $O(e) \subset \text{int}(CH(\mathcal{L}))$. Let $e_1 \in O(e) \cap \text{int}(H^+)$. Since $\mathcal{L}$ forms a tetrahedralization, there exists a tetrahedron t_1 containing e_1. If t_1 also contains e, then abc must be a face of t_1, which contradicts the assumption that $n_t(abc^+) = 0$. Thus, t_1 does not contain e. Let ee_1 intersect the boundary of t_1 at e_1', and e_2 be the middle point of edge ee_1'. Then e_2 is outside t_1. There must exist a tetrahedron t_2 containing e_2. By using the same argument, we get an infinite sequence of points $e_1, e_2, ..., e_i, ...$ such that each e_i is contained in a different tetrahedron t_i. However, there are only a finite number of tetrahedra in a tetrahedralization. This is a contradiction. Therefore $n_t(abc^+) = 1$. Similarly, we can prove that $n_t(abc^-) = 1$ for each $abc \in \Delta_I$.

(Sufficiency). Suppose that $n_t(abc^+) = 1$ and $n_t(abc^-) = 1$, for each $abc \in \Delta_I$. Thus, for each $\mathcal{L}$-empty inner face abc, there are exactly two vertices d and d' such that $abcd$ and $abcd'$ are two $\mathcal{L}$-empty tetrahedra. Let T denote the set of all these $\mathcal{L}$-empty tetrahedra. From the assumption, we know that T contains at least two tetrahedra. For any $t, t' \in T$, $\text{int}(t) \cap \text{int}(t') = \emptyset$ since they are $\mathcal{L}$-empty. Thus, in

order to prove that T is a tetrahedralization, we only need to show that the union of all the tetrahedra in T is $CH(\mathcal{L})$. We prove this by contradiction. Suppose there exists a point $p \in CH(\mathcal{L})$ that is not contained in any tetrahedron of T. Let $t \in T$ and select an interior point q in t such that $\text{int}(pq) \cap \mathcal{L} = \emptyset$. Let $F_{pq} = \{f \mid f \in \Delta_I, \text{int}(f) \cap pq \neq \emptyset\}$. So one facet of t belongs to F_{pq}. Since F_{pq} is a nonempty finite set, there must exist a facet $p_1p_2p_3$ that is the nearest facet to p along edge pq among facets of F_{pq}. Let H be the plane containing $p_1p_2p_3$. Without loss of generality, suppose that p lies in the halfspace H^+. Since $n_t(p_1p_2p_3^+) = 1$, there exists a point p_4 such that $p_1p_2p_3p_4$ is an $\mathcal{L}$-empty tetrahedra. Since $p \notin \bigcup_{t\in T} t$, pq must intersect two faces of $p_1p_2p_3p_4$, say $p_1p_2p_3$ and $p_2p_3p_4$. Then $p_2p_3p_4$ is nearer than $p_1p_2p_3$ to p along pq. This is a contradiction. □

For each vertex $v \in V(\mathcal{L})$, let $\mathcal{L}(v)$ be the edge set induced from vertex v and all its adjacent vertices, and $T(v)$ be a set of tetrahedra in $\mathcal{L}$ with v as a vertex such that each tetrahedron in $T(v)$ is $\mathcal{L}(v)$-empty. For each tetrahedron $t \in T(v)$, the three facets incident to v are called *side facets* of v and the fourth facet is called the *bottom facet* of v. The union of all the bottom facets of v is denoted by $B(v)$. From Section 1.1, we know that the triangulation surface can be considered as a portion of the boundary of a simplicial polyhedron, and the closed triangulation surface can be considered as the whole boundary of a simplicial polyhedron.

Given the above, we have the following theorem which is the characterization of

a tetrahedralization from the vertex viewpoint.

Theorem 2.2 *$\mathcal{L}$ forms a tetrahedralization if and only if the following conditions hold:*

1. *for each vertex $v \in V(CH(\mathcal{L}))$, $B(v)$ is a triangulation surface whose boundary edges are contained in $E(CH(\mathcal{L}))$; and*
2. *for each vertex $v \in \text{int}(CH(\mathcal{L}))$, $B(v)$ is a closed triangulation surface with v in its interior.*

Proof (Necessity). Let $\mathcal{L}$ form a tetrahedralization and v be an arbitrary vertex in $V(\mathcal{L})$. Since each tetrahedron in $T(v)$ is $\mathcal{L}$-empty, each face of $B(v)$ is a triangle. Suppose ab is an arbitrary boundary edge of $B(v)$. If vab is an inner face of $CH(\mathcal{L})$, from Theorem 2.1 we know that there exist vertices c and c' such that $vabc$ and $vabc'$ are two $\mathcal{L}$-empty tetrahedra. So ab is an inner edge in $B(v)$. This is a contradiction. Thus, if $v \in V(CH(\mathcal{L}))$, each boundary edge of $B(v)$ belongs to $E(CH(\mathcal{L}))$; if $v \in \text{int}(CH(\mathcal{L}))$, $B(v)$ is a closed triangulation surface with v in its interior.

(Sufficiency). Let $T = \bigcup_{v \in V(\mathcal{L})} T(v)$. We will prove that T is a tetrahedralization. We first show that for any point p in $CH(\mathcal{L})$, there exists at least one vertex $v \in V(\mathcal{L})$ such that $p \in T(v)$. If p is on the boundary of $CH(\mathcal{L})$, then there exists a face abc containing p. Thus $p \in T(a)$. Suppose there exists an interior point $p \in \text{int}(CH(\mathcal{L}))$ which is not contained in any $T(v), v \in V(\mathcal{L})$. Select an arbitrary tetrahedron $t \in T$

and an interior point q of t such that $\text{int}(pq) \cap \mathcal{L} = \emptyset$. Let $F_{pq} = \{f \mid \text{int}(f) \cap pq \neq \emptyset, f \in B(v), v \in V(\mathcal{L})\}$. Since F_{pq} is a nonempty finite set (at least one facet of t belongs to F_{pq}), there must exist a facet abc that is the nearest facet to p along edge pq among facets of F_{pq}. Since abc is an inner face and $p \notin T(a)$, pq must intersect a face of $B(a)$ which is nearer than abc to p along pq. This is a contradiction.

We now show that each tetrahedron in $T(v)$ is $\mathcal{L}$-empty. For any $v \in V(\mathcal{L})$, since each face of $B(v)$ is a triangle, any pair of tetrahedra in $T(v)$ are interior disjoint. Suppose there is a tetrahedron $vabc \in T(v)$ which is not $\mathcal{L}$-empty. There are two cases regarding $vabc$.

1. The four facets of $vabc$ are $\mathcal{L}$-empty. Let S be the set of vertices of $V(\mathcal{L})$ inside $vabc$. So the vertices of S are interior points of $CH(\mathcal{L})$. Since abc is a face in $B(v)$, there is no edge between v and S. For any $u \in S$, there exists a $T(u)$ such that u is an interior point of the simplicial polyhedron bounded by $B(u)$. Let $F_{uv} = \{f \mid \text{int}(f) \cap uv \neq \emptyset, f \in B(x), x \in V(\mathcal{L})\}$. Using an argument similar to the above, we can derive a contradiction for F_{uv}.

2. There exists an edge pq which intersects at least one facet of $vabc$. Suppose p is outside of $vabc$ and xyz is the nearest facet to p along pq among all the facets intersected by pq (See Figure 2.3(a)). If $pxyz \notin T(x)$, since xyz is a side facet of a tetrahedron in $T(x)$, pq must intersect a triangle face of $B(x)$ which is nearer than xyz to p along pq. This is a contradiction. Thus, $pxyz \in$

$T(x)$. The adjacent vertices of x in $B(p)$ form a cycle ($x \in \text{int}(CH(\mathcal{L}))$) or a chain ($x \in V(CH(\mathcal{L}))$). If $xy \notin E(B(p))$, there must exist an edge $x'y'$ in the cycle (or chain) such that $x'y' \cap \text{int}(pxy) \neq \emptyset$, and $xx'y'$ is a face in $B(p)$(See Figure 2.3(b)). Since $x'y' \in \mathcal{L}(x)$, $pxyz$ is not $\mathcal{L}(x)$-empty. This is a contradiction because $pxyz \in T(x)$. Thus $xy \in E(B(p))$. Similarly, $yz, zx \in E(B(p))$. Because $B(p)$ is a triangulation surface and $pq \cap \text{int}(xyz) \neq \emptyset$, there must exist a vertex $p' \in V(\mathcal{L}(p))$ such that $p'xy$ is a face in $B(p)$ and the line containing pp' intersects the interior of xyz (See Figure 2.3(c)). Since $pp' \in \mathcal{L}(x)$ and $pp' \cap \text{int}(pxyz) \neq \emptyset$, this is a contradiction because $pxyz \in T(x)$.

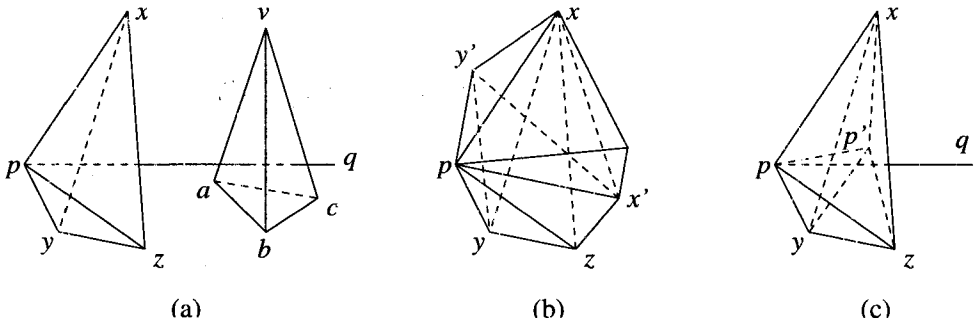

Figure 2.3: (a) pq intersects face xyz. (b) $x'y'$ intersects face pxy. (c) $p'xy$ is a face in $B(p)$ and the line containing pp' intersects face xyz.

From cases 1 and 2 we know that for any $vabc \in T(v), v \in V(\mathcal{L})$, $vabc$ is $\mathcal{L}$-empty. Hence T is a tetrahedralization. Finally, we show that $\mathcal{L} = E(T)$. For any $uv \in \mathcal{L}$, if u is not a vertex of $B(v)$, then uv will intersect a face abc of $B(v)$ or will be contained in $vabc$. So $vabc$ is not $\mathcal{L}(v)$-empty. Thus $vabc \notin T(v)$ and $abc \notin B(v)$. This is a contradiction. Hence u is a vertex of $B(v)$ and $uv \in E(T(v)) \subset E(T)$. Thus $\mathcal{L} \subseteq E(T)$. On the other hand, it is obvious that $E(T) \subseteq \mathcal{L}$. Therefore $\mathcal{L} = E(T)$. $\square$

Consider the space divided by the $\mathcal{L}$-empty triangles in $\Delta_{\mathcal{L}}$: $C = R^3 - \{\mathrm{cl}(abc) \mid abc \in \Delta_{\mathcal{L}}\}$. Let n_c be the number of the bounded connected components in C. Each connected component of C is called a *cell.* Each bounded cell is denoted by $C_i (1 \leq i \leq n_c)$, and the unbounded cell is denoted by C_0. So each $\mathrm{cl}(C_i)(0 \leq i \leq n_c)$ can be considered as a polyhedron. Recall that $|A|$ stands for the cardinality of A if A is a set. Then we have the following theorem which is the characterization of a tetrahedralization from the combinatorial viewpoint.

Theorem 2.3 *$\mathcal{L}$ forms a tetrahedralization if and only if the following conditions hold:*

1. *$(V(\mathcal{L}), \mathcal{L})$ is a connected graph;*
2. *each $\mathrm{cl}(C_i)(0 \leq i \leq n_c)$ is a simple polyhedron and $\mathrm{cl}(C_0)$ is $\mathcal{L}$-empty; and*
3. *$n_c = |\mathcal{L}| - |V(\mathcal{L})| - |V(CH(\mathcal{L}))| + 3$.*

Proof If $\mathcal{L}$ forms a tetrahedralization, it is easy to see that conditions 1 and 2 hold. From [48, Lemma 2.1] we know that condition 3 also holds. We now prove the sufficiency. Suppose the three conditions hold. Let $F(\mathrm{cl}(C_i))(0 \leq i \leq n_c)$ denote the facet set on the boundary of the polyhedron $\mathrm{cl}(C_i)$. Since the boundary of the polyhedron $\mathrm{cl}(C_0)$ is a maximal planar graph, it has $2|V(\mathrm{cl}(C_0))| - 4$ facets. If the number of facets on the boundary of a cell is summed over all cells, then such a sum counts each facet twice. Thus, $\sum_{i=1}^{n_c} |F(\mathrm{cl}(C_i))| + 2|V(\mathrm{cl}(C_0))| - 4 = 2n_f$, where n_f

is the total number of facets on the boundary of $\mathrm{cl}(C_i)(0 \leq i \leq n_c)$. Since each polyhedron $\mathrm{cl}(C_i)$ has at least 4 facets, we have

$$4n_c + 2|V(\mathrm{cl}(C_0))| - 4 \leq 2n_f, \tag{2.1}$$

where equality holds if and only if every $\mathrm{cl}(C_i)(1 \leq i \leq n_c)$ is a tetrahedron. Let $L' = \bigcup_{1 \leq i \leq n_c} E(\mathrm{cl}(C_i))$. From condition 2 we know that Euler's formula is valid for the cell complex consisting of $\mathrm{cl}(C_i)$. So $|V(L')| - |L'| + n_f - n_c = 1$. Substituting it into (2.1), we have

$$n_c \leq |L'| - |V(L')| - |V(\mathrm{cl}(C_0))| + 3. \tag{2.2}$$

In order to show (2.2) holds for $(V(\mathcal{L}), \mathcal{L})$, we now construct $(V(\mathcal{L}), \mathcal{L})$ from $(V(L'), L')$. Note that $(V(L'), L')$ is a subgraph of $(V(\mathcal{L}), \mathcal{L})$. In the construction process, we add edges one at a time so that the current graph is always connected. When we add an edge ab into $\mathrm{cl}(C_i)$, the left side of (2.2) does not change; the right side of (2.2) increases by 1 if a and b are already in the current graph, and does not change if a or b is a new vertex added to the current graph. Note that both a and b cannot be newly added vertices because we keep the current graph connected. Hence, for graph $(V(\mathcal{L}), \mathcal{L})$, we have

$$n_c \leq |\mathcal{L}| - |V(\mathcal{L})| - |V(\mathrm{cl}(C_0))| + 3, \tag{2.3}$$

where equality holds if and only if every $\mathrm{cl}(C_i)(1 \leq i \leq n_c)$ is an $\mathcal{L}$-empty tetrahedron. Since $\mathrm{cl}(C_0)$ is $\mathcal{L}$-empty, we have $V(CH(\mathcal{L})) \subseteq V(\mathrm{cl}(C_0))$. Thus, $|V(CH(\mathcal{L}))| \leq |V(\mathrm{cl}(C_0))|$. Hence, from (2.3) we obtain $n_c \leq |\mathcal{L}| - |V(\mathcal{L})| - |V(CH(\mathcal{L}))| + 3$, where equality holds if and only if each bounded cell is an $\mathcal{L}$-empty tetrahedron and the unbounded cell is the exterior of $CH(\mathcal{L})$. Therefore, it follows from condition 3 of the theorem that $\mathcal{L}$ forms a tetrahedralization. □

Similarly, we can prove the two dimensional counterpart of Theorem 2.3; see Appendix A for the proof.

Theorem 2.4 *Let* E *be a set of line segments in* R^2. E *is a triangulation if and only if* E *is a connected plane graph, and* $|\mathsf{E}| = 3|V(\mathsf{E})| - |V(CH(\mathsf{E}))| - 3$ *(or* $|F(\mathsf{E})| = 2|V(\mathsf{E})| - |V(CH(\mathsf{E}))| - 3$*).*

Hopcroft and Tarjan [58] showed that $O(n)$ time is sufficient to decide planarity on a conventional random access machine. From Theorem 2.4 we obtain the following theorem.

Theorem 2.5 *Given a set of edges* E *in* R^2*, determining whether* E *is a triangulation can be done in* $O(n)$ *time.*

2.4 Deciding whether $\mathcal{L}$ forms a tetrahedralization

2.4.1 Outline of the algorithm

In this section, let us consider a tetrahedralization detection problem — that is, how to design an efficient algorithm to decide if $\mathcal{L}$ forms a tetrahedralization.

Since there are $\binom{n}{3}$ triangles and $\binom{n}{4}$ tetrahedra in $\mathcal{L}$, testing whether each triple of vertices form an $\mathcal{L}$-empty triangle requires $O(n^3m)$ time and testing whether each quadruple of vertices form an $\mathcal{L}$-empty tetrahedron requires $O(n^4m)$ time. Hence, if we use a brute-force method to find all the $\mathcal{L}$-empty tetrahedra and check if their union is a tetrahedralization, the running time is at least $O(n^4m)$. If we apply Theorem 2.1, 2.2 or 2.3 directly to detect the tetrahedralization, the time complexity is as follows:

- In Theorem 2.1, since all the tetrahedra in Δ_I are $\mathcal{L}$-empty, this approach requires at least $O(n^3m)$ time.
- In Theorem 2.2, since each tetrahedra of $T(v)$ is $\mathcal{L}(v)$-empty, every side facets and bottom facets of v is $\mathcal{L}(v)$-empty, where the bottom facets form $B(v)$. Notice that $\mathcal{L}(v) = O(m)$, $V(\mathcal{L}(v)) = O(n)$, and in the worst case, the two bounds are tight (For example, if $(V(\mathcal{L}), \mathcal{L})$ is a complete graph, then $\mathcal{L}(v) = m$, for any $v \in V(\mathcal{L})$). Thus, the total running time of this approach is at least $O(n^3m)$.

- In Theorem 2.3, since each $\mathrm{cl}(C_i)$ is a simple polyhedron, every triangle in the boundary of $\mathrm{cl}(C_i)$ is $\mathcal{L}$-empty. This approach also requires at least $O(n^3m)$ time.

From the above analysis, we know that we should avoid computing too many empty triangles in our algorithms. Motivated by Theorem 2.2, we compute a triangulation surface B_v to be specified later, which is easier to compute than $B(v)$. Connecting v with each vertex of B_v, we obtain a set of tetrahedra T_v. As the triangles in B_v may not be $\mathcal{L}(v)$-empty, this modification cannot guarantee that the union of T_v ($v \in V(\mathcal{L})$), denoted as T, is a tetrahedralization; thus we need to add an extra step to check T. Fortunately, this step can be completed in $O(n^2)$ time by testing if intersections exist between tetrahedra in T. For any vertex $v \in V(\mathcal{L})$, let $\mathrm{adj}(v)$ be the set of adjacent vertices of v, and $\mathcal{L}_v = \{ab \mid ab \in \mathcal{L} \text{ and } a, b \in \mathrm{adj}(v)\}$. Note that the relationship between $\mathcal{L}_v$ and $\mathcal{L}(v)$ is $\mathcal{L}(v) = \mathcal{L}_v \cup \{va \mid a \in \mathrm{adj}(v)\}$. Before we present the algorithm, let us give some definitions.

Definition 2.2 *We say that two segments aa' and bb'* `cross` *one another if and only if their interiors intersect at a single point. This point is called a* `cross-point`*. We say that an edge* `punctures` *a face (facet or polyhedron) if and only if their interiors have a nonempty intersection.*

The outline of our algorithm is as follows:

Algorithm TETRAHEDRALIZATION-DETECTION($\mathcal{L}$) (outline)

Input: A set of line segments $\mathcal{L}$ in R^3.

Output: If $\mathcal{L}$ does not form a tetrahedralization, return (NO). Otherwise, return (YES, T), where T is the set of tetrahedra in the tetrahedralization $\mathcal{L}$.

Step 1 Compute $CH(\mathcal{L})$. If $E(CH(\mathcal{L})) \not\subseteq \mathcal{L}$, then return (NO).

Step 2 (CHECK-TRIANGULATION) For each vertex v on the boundary of $CH(\mathcal{L})$, check if $\mathcal{L}_v$ contains a specified triangulation surface:

Step 2.1 For each segment $pp' \in \mathcal{L}_v$, if there exists a segment $qq' \in \mathcal{L}_v$ such that qq' punctures face vpp', then delete pp' from $\mathcal{L}_v$.

Step 2.2 If the updated $\mathcal{L}_v$ is the edge set of a triangulation surface (denoted as B_v) whose boundary edges are contained in $E(CH(\mathcal{L}))$, then let T_v be the set of tetrahedra $vabc$, where abc is a bounded face in B_v. Otherwise, return (NO).

Step 3 (CHECK-CLOSED-TRIANGULATION) For each vertex v inside $CH(\mathcal{L})$, check if $\mathcal{L}_v$ contains a specified closed triangulation surface with v in its interior:

Step 3.1 Same as Step 2.1.

Step 3.2 If the updated $\mathcal{L}_v$ is the edge set of a closed triangulation surface (denoted as B_v) with v in its interior, then let T_v be the set of tetrahedra $vabc$, where abc is a face in B_v. Otherwise, return (NO).

Step 4 Let $T = \cup_{v \in V(\mathcal{L})} T_v$. For each tetrahedron $abcd \in T$, if $abcd \in T_a \cap T_b \cap T_c \cap T_d$, then return (YES, T); otherwise return (NO).

The details of this algorithm are given in Section 2.4.2. The following theorem shows the characterization of a tetrahedralization from the algorithmic viewpoint. This characterization will be used in this section.

Theorem 2.6 *$\mathcal{L}$ forms a tetrahedralization if and only if* TETRAHEDRALIZATION-DETECTION*($\mathcal{L}$) (outline) returns (YES, T).*

Proof (Necessity). Let $\mathcal{L}$ form a tetrahedralization. From Theorem 2.2 we know that for each vertex $v \in V(CH(\mathcal{L}))$, $B(v)$ is a triangulation surface, and for each vertex $v \in \text{int}(CH(\mathcal{L}))$, $B(v)$ is a closed triangulation surface with v in its interior. For each

face abc in $B(v)$, we have $vabc \in T(v)$ which is defined prior to Theorem 2.2. Since $T(v)$ is $\mathcal{L}(v)$-empty, vab, vbc and vca are $\mathcal{L}(v)$-empty. Thus, ab, bc and ca cannot be deleted in Steps 2.1 and 3.1. Since each face abc is still in the updated $\mathcal{L}_v$, we have $B_v = B(v)$ and $T_v = T(v)$. For each $vabc \in T_v$, since $\mathcal{L}$ forms a tetrahedralization, we know that $vabc$ also belongs to T_a, T_b and T_c. Therefore TETRAHEDRALIZATION-DETECTION($\mathcal{L}$) returns (YES, T), where $T = \cup_{v \in V(\mathcal{L})} T_v$.

(Sufficiency). Suppose TETRAHEDRALIZATION-DETECTION($\mathcal{L}$) returns (YES, T). We shall prove that $\mathcal{L}$ forms a tetrahedralization. We first show that $\mathcal{L} = E(T)$. For any $uv \in \mathcal{L}$, since $uv \notin \mathcal{L}_v$, segment uv and vertex u cannot be deleted in Steps 2.1 and 3.1. So $uv \in E(T_v) \subseteq E(T)$. Hence $\mathcal{L} \subseteq E(T)$. On the other hand, since we never add new segments in the algorithm, we have $E(T) \subseteq \mathcal{L}$. Therefore $\mathcal{L} = E(T)$. We now show that T is a tetrahedralization.

From Step 1, we know that the boundary of $CH(\mathcal{L})$ is contained in T. If there exists an interior point of $CH(\mathcal{L})$ which is not contained in T, similar to the argument in the proof of Theorem 2.2, we can derive a contradiction. Thus, $CH(\mathcal{L})$ is covered by T. We finally show that each tetrahedron in T is $\mathcal{L}$-empty. Suppose there is a tetrahedron $abcd \in T$ which is not $\mathcal{L}$-empty. There are two cases concerning the nonemptiness of $abcd$.

1. The four facets of $abcd$ are $\mathcal{L}$-empty. Let S be the set of vertices of $V(\mathcal{L})$ inside $abcd$. Since $abcd \in T_a \cap T_b \cap T_c \cap T_d$, there is no edge between $\{a, b, c, d\}$ and S.

For a vertex $s \in V(CH(S))$, we have $s \in \text{int}(CH(\mathcal{L}))$; but $\mathcal{L}_s$ cannot contain a closed triangulation surface with s in its interior. Thus Step 3.2 returns (NO). This is a contradiction.

2. There exists a segment pq which punctures at least one facet of $abcd$. Suppose p lies outside $abcd$ and xyz is the nearest facet to p among all the facets punctured by pq. If $pxyz \notin T_x$, since xyz is a side facet of a tetrahedron in T_x, pq must intersect a triangle face of B_x which is nearer than xyz to p along pq. This is a contradiction. Thus, $pxyz \in T_x \subset T$. From Step 4 we know that $pxyz \in T_p$. Hence xyz is a triangle face in B_p. This contradicts the assumption that pq punctures xyz.

Therefore, for any $abcd \in T$, $abcd$ is $\mathcal{L}$-empty. This completes the proof. □

2.4.2 Details of the algorithm

Let $\prec$ denote the lexicographical order on points. We also use $\prec$ to define the lexicographical order on tetrahedra. Let $abcd$ and $a'b'c'd'$ be two tetrahedra. Then $abcd \prec a'b'c'd'$ if and only if $a \prec a'$; or $a = a'$ and $b \prec b'$; or $a = a'$, $b = b'$, and $c \prec c'$; or $a = a'$, $b = b'$, $c = c'$, and $d \prec d'$. For the tetrahedra of T produced in the algorithm, we store them in a red-black tree [36], ordered according to $\prec$. In order to search tetrahedra easily, we constrain the vertices in the representation of the tetrahedron also in $\prec$ order, that is, $a \prec b \prec c \prec d$ for any $abcd \in T$. We store a

number count(t) with each tetrahedron $t \in T$, which indicates the number of times when t is qualified to be inserted into T. According to the deletion process in Steps 2.1 and 3.1, we know that for each segment pp' in B_v, face vpp' is not punctured by any segment in the updated $\mathcal{L}_v$. Thus we have the following definition.

Definition 2.3 *The (closed) triangulation surface B_v produced in Steps 2 and 3 of* TETRAHEDRALIZATION-DETECTION*($\mathcal{L}$) (outline) is called* `puncture-free`.

Note that the puncture-free property is sensitive to the order in which segments are deleted from $\mathcal{L}_v$ — that is, the puncture-free (closed) triangulation surface may be different if the order of segment deletion is changed.

In TETRAHEDRALIZATION-DETECTION($\mathcal{L}$), lines 2–5 corresponds to Step 1 in the outline; lines 7–11 corresponds to Step 2; lines 12–16 corresponds to Step 3; and lines 17–31 corresponds to Step 4.

Algorithm TETRAHEDRALIZATION-DETECTION($\mathcal{L}$)
Input: A set of line segments $\mathcal{L}$ in R^3.
Output: If $\mathcal{L}$ does not form a tetrahedralization, return (NO). Otherwise, return (YES, T), where T is the set of tetrahedra in the tetrahedralization $\mathcal{L}$.

```
1  T ← ∅.
   (* T is implemented by a red-black tree. *)
2  Compute CH(L).
3  if E(CH(L)) ⊈ L
4      then return (NO) and stop.
5  endif
6  for each vertex v ∈ V(L)
7      do if v ∈ V(CH(L))
8             then Pick a plane H_v through v that is tangent to CH(L) and
```

translate H_v to a position H'_v such that $CH(\mathcal{L})$ is between the old H_v and H'_v.
9 Project each vertex of adj(v) and each segment of $\mathcal{L}_v$ from v onto plane H'_v; let L'_v be the image of $\mathcal{L}_v$ under the projection.
10 CHECK-TRIANGULATION($v, L'_v; B_v$).
(* Check if $\mathcal{L}_v$ contains a puncture-free triangulation surface B_v. *)
11 Let T_v be the set of tetrahedra $vabc$, where $v \prec a \prec b \prec c$ and abc is a bounded face in the triangulation surface B_v.
12 **else** Pick a sphere S_v with center v such that $\mathcal{L}$ is contained in S_v.
13 Project each vertex of adj(v) and each segment of $\mathcal{L}_v$ from v onto sphere S_v; let L'_v be the image of $\mathcal{L}_v$ under the projection.
14 CHECK-CLOSED-TRIANGULATION($v, L'_v; B_v$).
(* Check if $\mathcal{L}_v$ contains a puncture-free closed triangulation surface B_v with v in its interior. *)
15 Let T_v be the set of tetrahedra $vabc$, where $v \prec a \prec b \prec c$ and abc is a face in the closed triangulation surface B_v.
16 **endif**
17 **for** each tetrahedron $t \in T_v$
18 **do if** $t \notin T$
19 **then** Insert t into T
20 count(t) $\leftarrow 1$
21 **else** count(t) $\leftarrow$ count(t) $+ 1$
22 **endif**
23 **endfor**
24 **endfor**
25 **for** each tetrahedron $t \in T$
26 **do if** count(t) < 4
27 **then return** (NO) and stop.
28 **endif**
29 **endfor**
30 **return** (YES, T) and stop.
31 **end**

In CHECK-TRIANGULATION($v, L'_v; B_v$), we use the plane sweep algorithm to check if $\mathcal{L}_v$ contains a puncture-free triangulation surface by using L'_v. Plane sweep algorithms are very suitable for finding intersections in a set of geometric objects

[93, 15, 23, 80]. From Steps 2.1 and 3.1 of the outline we know that our aim is to find the crosses in L'_v and remove these crosses by deleting one of the crossing segments according to their puncture relationship. Hence, in our plane sweep algorithm, an imaginary vertical sweep line passes through L'_v from left to right. The status of the sweep line is a set of segments intersecting it. The status changes while the sweep line moves. We place the status into a dynamic data structure Y. The status structure Y is updated only at the endpoints of L'_v. Note that the definition of the cross-point in this thesis is different from the intersection point in [93, 15, 23, 80]. In their definitions, two segments pp' and qq' intersect if and only if $pp' \cap qq'$ is a singleton, which is different from the cross-point in the endpoints. This small difference causes a significant difference in the conditions for deciding crosses.

In CHECK-TRIANGULATION$(v, L'_v; B_v)$, we choose an xy-coordinate system such that each segment in L'_v is not vertical to the x-axis. Lines 1–11 check if L'_v satisfies two necessary conditions for containing a triangulation: one is left$(p) \neq \emptyset$; the other is $E(CH(L'_v)) \subseteq L'_v$. Lines 12–16 is our plane sweep algorithm for removing cross-points. Lines 17–22 check if the updated L'_v is a triangulation and produce the triangulation surface B_v.

CHECK-TRIANGULATION$(v, L'_v; B_v)$
1 **for** each point p in $V(L'_v)$ except the endpoint with the largest x-coordinate
2 **do** Let left(p) be a set of segments whose left endpoint is p; store left(p) with the endpoint p.
3 **if** left$(p) = \emptyset$
4 **then return** (NO) and stop.

5 **endif**
6 **endfor**
7 Compute $CH(L'_v)$.
8 **if** $E(CH(L'_v)) \subseteq L'_v$,
9 **then** Calculate $n_v = |V(L'_v)|$ and $n'_v = |V(CH(L'_v))|$.
10 **else return** (NO) and stop.
11 **endif**
12 $Y \leftarrow \emptyset$
(* The status structure Y is implemented by a red-black tree. *)
13 Sort all the endpoints of segments in L'_v from left to right, breaking ties by putting points with lower y-coordinates first.
14 **for** each point a in the sorted list of endpoints
15 **do** UPDATE-STATUS(a)
16 **endfor**
17 Calculate $|L'_v|$ (L'_v has been updated).
18 **if** $|L'_v| = 3n_v - n'_v - 3$
19 **then** L'_v is a plane triangulation.
20 Let B_v be the triangulation surface whose edge set (namely, the updated $\mathcal{L}_v$) corresponds to L'_v under the projection.
(* B_v is puncture-free according to DELETE-SEGMENT to be specified later. *)
21 **else return** (NO) and stop.
22 **endif**

UPDATE-STATUS(a) describes how to update the status structure Y at endpoint a. Line 1 deletes the segments whose right endpoint is a from Y since the sweep line has passed through these segments. Line 2 inserts the segments whose left endpoint is a into Y since the sweep line will intersect these segments. Lines 3-9 find and remove the cross-points between the new inserted segments in left(a) and the old segments in above(a) and below(a).

UPDATE-STATUS(a)
1 Delete the segments whose right endpoint is a from Y.
2 Insert the segments of left(a) into Y. The order of these segments in Y should correspond to the order in which they are crossed by a sweep line

```
    (or longitude in CHECK-CLOSED-TRIANGULATION to be specified later)
    just to the right of a.
3   Let above(a) be the set of segments above left(a) in Y, and below(a) be
    the set of segments below left(a) in Y.
4   if above(a) ≠ ∅,
5       then HANDLE-ABOVE-SEGMENTS(left(a), above(a))
6   endif
7   if below(a) ≠ ∅,
8       then HANDLE-BELOW-SEGMENTS(left(a), below(a))
9   endif
```

In HANDLE-ABOVE-SEGMENTS(left(a), above(a)), line 1 finds a pair of neighbour segments, one in left(a), the other in above(a). If these two segments cross each other, then delete one of them (line 2). If necessary, repeat the above steps recursively (lines 3-15).

```
HANDLE-ABOVE-SEGMENTS(left(a), above(a))
1   Let aa' be the uppermost segment in left(a) and bb' be the lowest
    segment in above(a).
2   DELETE-SEGMENT(aa', bb')
3   if aa' is deleted from Y
4       then Delete aa' from left(a)
5            if left(a) = ∅
6                then return (NO) and stop.
7                else HANDLE-ABOVE-SEGMENTS(left(a), above(a))
8            endif
9   endif
10  if bb' is deleted from Y
11      then Delete bb' from above(a)
12           if above(a) ≠ ∅
13               then HANDLE-ABOVE-SEGMENTS(left(a), above(a))
14           endif
15  endif
```

Similar to HANDLE-ABOVE-SEGMENTS, we have the following procedure.

HANDLE-BELOW-SEGMENTS(left(a), below(a))
1 Let aa'' be the lowest segment in left(a) and cc' be the uppermost segment in below(a).
2 DELETE-SEGMENT(aa'', cc')
3 **if** aa'' is deleted from Y
4 **then** Delete aa'' from left(a)
5 **if** left(a) $= \emptyset$
6 **then return** (NO) and stop.
7 **else** HANDLE-BELOW-SEGMENTS(left(a), below(a))
8 **endif**
9 **endif**
10 **if** cc' is deleted from Y
11 **then** Delete cc' from below(a)
12 **if** below(a) $\neq \emptyset$
13 **then** HANDLE-BELOW-SEGMENTS(left(a), below(a))
14 **endif**
15 **endif**

DELETE-SEGMENT describes how to remove the cross-point by deleting one segment according to their puncture relationship.

DELETE-SEGMENT(p_1p_2, q_1q_2)
1 Let $p'_1p'_2 \in \mathcal{L}_v$ be the preimage of p_1p_2 and $q'_1q'_2 \in \mathcal{L}_v$ be the preimage of q_1q_2 under the projection.
2 **if** $p'_1p'_2$ punctures face $vq'_1q'_2$
3 **then** Delete q_1q_2 from Y and L'_v, and delete $q'_1q'_2$ from $\mathcal{L}_v$.
4 **endif**
5 **if** $q'_1q'_2$ punctures face $vp'_1p'_2$
6 **then** Delete p_1p_2 from Y and L'_v, and delete $p'_1p'_2$ from $\mathcal{L}_v$.
7 **endif**

In CHECK-CLOSED-TRIANGULATION($v, L'_v; B_v$), we extend the plane sweep idea to a sphere. We will sweep the sphere with a rotating semi-circle on the sphere to check if $\mathcal{L}_v$ contains a puncture-free closed triangulation surface by using L'_v. Notice that each element $pq \in L'_v$ is a segment of a circle; we simply call it segment. Since

pq is the image of a line segment projected from the center v of the sphere, the arc length of pq is less than that of half of a big circle on the sphere, where a *big circle* is a circle with the same center and radius as the sphere. Any two different big circles cross each other at two points which divide the two circles into four semi-circles. Thus, the intersection of any two different segments of L'_v is at most a single point. Similar to Definition 2.2, we define that two circle segments of L'_v *cross* if and only if their interiors intersect at a single point. In order to use an argument similar to the plane sweep algorithm, we choose an xy-coordinate system on the sphere S_v. Without loss of generality, we suppose that S_v is a unit sphere. Select a pair of points $y, y' \notin V(L'_v)$ as the north and south poles so that no segment of L'_v lies on the longitude. Choose the equator as the x-axis and an arbitrary longitude as the y-axis. The intersection of the xy-axes is the origin. For an arbitrary point $p \in S_v - \{y, y'\}$, let the longitude through p intersect the x-axis at a point whose x-coordinate is p_x; and let the latitude through p intersect the y-axis at a point whose y-coordinate is p_y. Then the coordinates of p is defined as (p_x, p_y). For example, the coordinates of y and y' are $(0, \pi/2)$ and $(0, -\pi/2)$, respectively. Note that the domain of the x-coordinate is $[0, 2\pi)$; and the domain of the y-coordinate is $(-\pi/2, \pi/2)$ (refer to Figure 2.4). For any segment $pq \in L'_v$, since pq does not lie on any longitude, there must exist a point $p' \in pq$ such that $\text{int}(pp') \cap y\text{-axis} = \emptyset$. If $p_x < p'_x$, then p is called the left endpoint of pq; if $p_x > p'_x$, then p is called the right endpoint of pq; as shown in Figure 2.4. We say that point p is above (below) q if and only if $p_y > q_y$ ($p_y < q_y$).

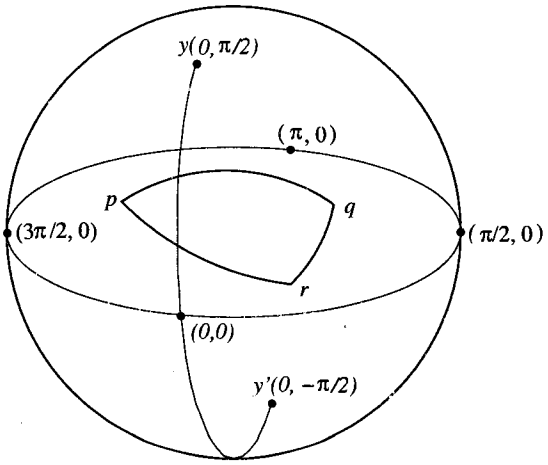

Figure 2.4: p is the left endpoint, q and r are the right endpoints of pq and pr.

In the next procedure, lines 1–6 check if L'_v satisfies a necessary condition for containing a triangulation on the sphere. Lines 7–16 implement the sphere sweep algorithm for removing cross-points. Lines 17–22 check if the updated L'_v is a triangulation on S_v and produce the closed triangulation surface B_v.

CHECK-CLOSED-TRIANGULATION$(v, L'_v; B_v)$.

```
1   for each point p in V(L'_v)
2       do Let left(p) be a set of segments whose left endpoint is p; store left(p)
           with the endpoint p.
3          if left(p) = ∅
4              then return (NO) and stop.
5          endif
6   endfor
7   Y ← ∅
    (* The status structure Y is implemented by a red-black tree. *)
8   Sort the endpoints of V(L'_v) lexicographically by nondecreasing coordinates.
    Let X be the sorted list of the endpoints.
9   for each point a ∈ X in the sorted order
10      do UPDATE-STATUS(a)
11  endfor
12  for each point a ∈ X in the sorted order
13      do while a_x < π
```

14 **do** UPDATE-STATUS(a)
15 **endwhile**
16 **endfor**
17 Calculate $|L'_v|$ and $|V(L'_v)|$ (L'_v has been updated).
18 **if** $|L'_v| = 3|V(L'_v)| - 6$
19 **then** L'_v is a maximal planar graph (triangulation) on S_v.
20 Let B_v be the closed triangulation surface whose edge set (namely, the updated $\mathcal{L}_v$) corresponds to L'_v under the projection.
(* B_v is puncture-free according to DELETE-SEGMENT. *)
21 **else return** (NO) and stop.
22 **endif**

2.4.3 Correctness

The following lemma shows that CHECK-TRIANGULATION is correct.

Lemma 2.1 *Let* CHECK-TRIANGULATION *be run on a vertex v and segment set L'_v.*

(i) If the procedure produces B_v, then B_v is a puncture-free triangulation surface.

(ii) If the procedure returns (NO), then $\mathcal{L}$ cannot form a tetrahedralization.

Proof (i) Let B_v be produced by the procedure. We will prove that B_v is a puncture-free triangulation surface as it is claimed in line 20. We first show that B_v is a plane graph by proving that the plane sweep algorithm of lines 12–16 removes all the cross-points of segments. Let p_1p_2 and q_1q_2 be two segments in L'_v such that $p_1 \prec p_2$, $q_1 \prec q_2$, and p_1p_2 crosses q_1q_2 at point x. Without loss of generality, we assume that $p_1 \prec q_1$ and p_1p_2 is above q_1q_2 in Y. Thus, when q_1q_2 is inserted into Y in UPDATE-STATUS(q_1), p_1p_2 is already in Y. We distinguish the following two cases concerning the relative position of p_1p_2 and q_1q_2 in Y.

1. p_1p_2 and q_1q_2 are adjacent in Y. p_1p_2 or q_1q_2 is deleted when DELETE-SEGMENT (p_1p_2, q_1q_2) is run. Thus, the cross-point x is removed.

2. p_1p_2 and q_1q_2 are not adjacent in Y. We distinguish two subcases when UPDATE-STATUS(q_1) is running, namely,

   (a) All the segments between p_1p_2 and q_1q_2 in Y are deleted by running HANDLE-ABOVE-SEGMENTS(left(q_1), above(q_1)) recursively, as illustrated in Figure 2.5. So p_1p_2 and q_1q_2 become neighbours in Y. Hence, by using the same argument as case 1, the cross-point x can be removed.

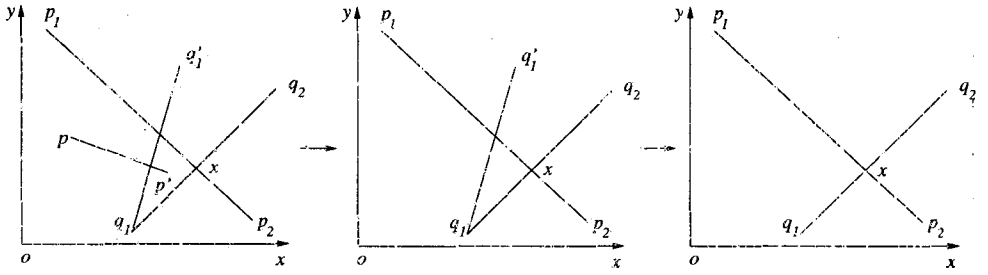

Figure 2.5: Suppose the deletion order of segments in Y between p_1p_2 and q_1q_2 is pp', q_1q_1'.

   (b) After UPDATE-STATUS(q_1) is completed, there exists a set E_x of segments in Y between p_1p_2 and q_1q_2. Note that $q_1 \prec a'$ for any segment $aa' \in E_x$ $(a \prec a')$. From case 1, we know that each pair of neighbour segments in $E_x \cup \{p_1p_2, q_1q_2\}$ do not cross each other. Let $a_1a_2 \in E_x$ $(a_1 \prec a_2)$ be adjacent to p_1p_2, $b_1b_2 \in E_x$ $(b_1 \prec b_2)$ be adjacent to q_1q_2, and $V_x \subset V(L_v')$ be the set of vertices which lie in the triangle p_1xq_1. We now prove that $V_x \neq \emptyset$. If both a_2 and b_2 lie outside the triangle p_1xq_1, a_1a_2 must cross

b_1b_2; as shown in Figure 2.6(a). Then consider the set of segments in Y between a_1a_2 and b_1b_2. This segment set is smaller than E_x by two segments. Since each pair of neighbour segments in E_x do not cross each other, by narrowing E_x recursively, we can finally find a segment $bb' \in E_x$ ($b \prec b'$) which does not cross any other segments in E_x. Since $q_1 \prec b'$, we know that b' lies in the triangle p_1xq_1. Hence, $V_x \neq \emptyset$.

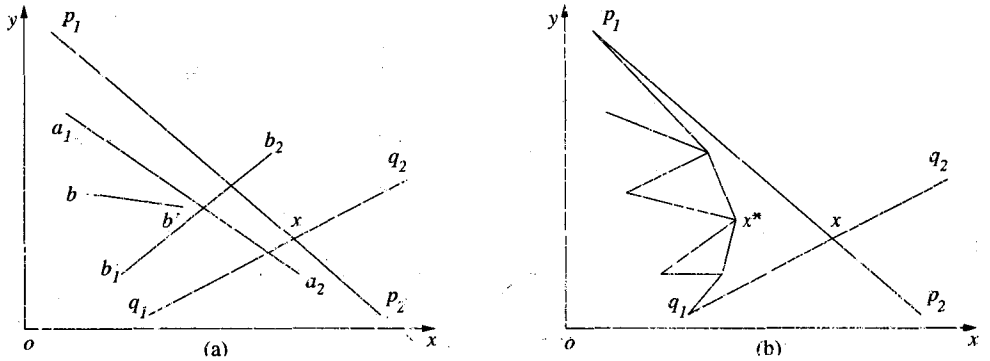

Figure 2.6: (a) $E_x = \{a_1a_2, b_1b_2, bb'\}$. (b) $\text{left}(x^*) = \emptyset$.

For each vertex $x' \in V_x$, since the procedure produces B_v, we always have $\text{left}(x') \neq \emptyset$ in line 5 of HANDLE-ABOVE-SEGMENTS(left(x'), above(x')). Thus, there must exist a segment $ab \in E_x$ such that ab crosses p_1p_2 or q_1q_2, say p_1p_2, and ab is adjacent to p_1p_2 in Y when running HANDLE-ABOVE-SEGMENTS. If p_1p_2 is deleted, then the cross-point x is removed; otherwise, if all the segments in E_x which cross p_1p_2 or q_1q_2 are deleted, there must exist a vertex $x^* \in V_x$ such that $\text{left}(x^*) = \emptyset$ (refer to Figure 2.6(b)). This is a contradiction. Hence, after each UPDATE-STATUS(x'), $x' \in V_x$, is run, p_1p_2 or q_1q_2 or both of them will be deleted and the cross-point x is removed.

Since lines 12–16 remove all cross-points, the updated L'_v must be a plane graph. Since the equality in line 18 holds, it follows from Theorem 2.4 that the updated L'_v is a plane triangulation. Therefore B_v is a triangulation surface. From the deletion process in DELETE-SEGMENT, we know that B_v is puncture-free.

(ii) Let $\mathcal{L}$ form a tetrahedralization. From Theorem 2.2 we know that for each vertex $v \in V(CH(\mathcal{L}))$, $B(v)$ is a triangulation surface. Since each tetrahedron in $T(v)$ is $\mathcal{L}(v)$-empty and $\mathcal{L}_v \subset \mathcal{L}(v)$ ($T(v)$ and $\mathcal{L}(v)$ are defined prior to Theorem 2.2), $B(v)$ is puncture-free under any order of segment deletion for $\mathcal{L}_v$. Let L''_v be the segment set in H'_v which corresponds to the edge set of $B(v)$ under the projection. So L''_v is a plane triangulation. Notice that $V(L''_v) = V(L'_v)$ and $L''_v \subseteq L'_v$. Thus, $E(CH(L'_v)) = E(CH(L''_v)) \subseteq L'_v$; the procedure does not return (NO) in line 10. For each point $p \in V(L''_v)$ except the vertex with the largest x-coordinate, since each angle in any triangles of L''_v is less than π, we have left$(p) \neq \emptyset$. So the procedure does not return (NO) in line 4. Since $B(v)$ is puncture-free under any order of segment deletion, L''_v is the updated L'_v after lines 12–16 is run. So the procedure does not return (NO) in line 6 of HANDLE-ABOVE-SEGMENTS and HANDLE-BELOW-SEGMENTS. Since L''_v is a triangulation, the equality in line 18 holds. Thus the procedure does not return (NO) in line 21.

Hence, if $\mathcal{L}$ forms a tetrahedralization, the procedure does not return (NO). Therefore, if the procedure returns (NO), then $\mathcal{L}$ cannot form a tetrahedralization. □

In the procedure CHECK-CLOSED-TRIANGULATION, lines 7–11 is the sphere sweep algorithm which is similar to the plane sweep algorithm of lines 12–16 in CHECK-TRIANGULATION. When we insert segments into Y whose left endpoints are near 2π, these segments may cross the y-axis. These segments may cross the segments which have been deleted from Y but belong to the updated L'_v, as illustrated in Figure 2.7. Hence, we must remove these crosses. For any segment $pp' \in L'_v$, since the arc length of pp' is less than π, we repeat the sphere sweep algorithm in lines 12–16 for the vertices whose x-coordinate is less than π. Using a similar argument to that used in the proof of Lemma 2.1, we can prove the correctness of CHECK-CLOSED-TRIANGULATION.

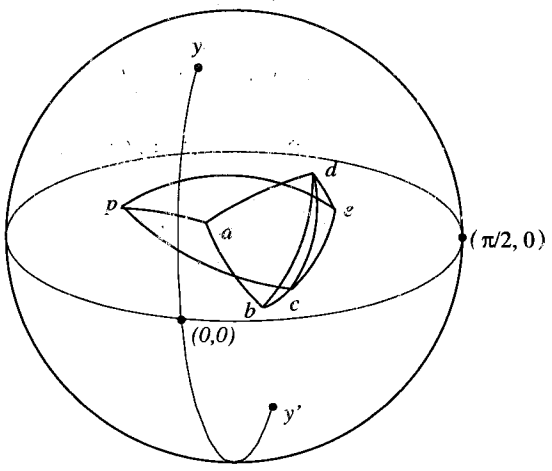

Figure 2.7: ab, ad, bd, cd belong to the updated L'_v, but pc, pe cross them after UPDATE-STATUS(p) is run.

Lemma 2.2 *Let* CHECK-CLOSED-TRIANGULATION *be run on a vertex v and segment set L'_v.*

(i) If the procedure produces B_v, then B_v is a puncture-free closed triangulation surface.

(ii) If the procedure returns (NO), then $\mathcal{L}$ cannot form a tetrahedralization.

Comparing TETRAHEDRALIZATION-DETECTION with its outline, we know that Step 1 is the same as lines 2–5. It follows from Lemma 2.1 that Step 2 is implemented by lines 7–11 correctly. It follows from Lemma 2.2 that Step 3 is implemented by lines 12–16 correctly. Step 4 corresponds to lines 17–29. For any four points $a, b, c, d \in V(\mathcal{L})$, from the definition of T_v we know that *abcd* belongs to at most four T_v, namely, T_a, T_b, T_c and T_d. Thus, in lines 17–23, we count the number of T_v that *abcd* belongs to. In lines 25–29, we decide if *abcd* belongs to exactly four T_v. Hence, Step 4 is implemented by lines 17–29 correctly.

Therefore, from Theorem 2.6 we know that TETRAHEDRALIZATION-DETECTION is correct.

Theorem 2.7 *If* TETRAHEDRALIZATION-DETECTION *is run on a set of line segments $\mathcal{L}$, then the algorithm returns (YES, T) if and only if $\mathcal{L}$ forms a tetrahedralization.*

2.4.4 Running time

The following lemmas show the upper bounds of the running times of the major procedures.

Lemma 2.3 *If m_a segments are deleted in* HANDLE-ABOVE-SEGMENTS *(or* HANDLE-

BELOW-SEGMENTS*), then the running time of the procedure is* $O(m_a \log m)$.

Proof Recall that Y is implemented by a red-black tree, and each red-black tree operation on Y, such as insertion, deletion and neighbour finding, takes $O(\log m)$ time. Thus, for each segment deleted in HANDLE-ABOVE-SEGMENTS (or HANDLE-BELOW-SEGMENTS), lines 1,2,4 and 11 take $O(\log m)$ time each. Hence, if m_a segments are deleted, the total running time is $O(m_a \log m)$. □

Lemma 2.4 *If* m_i *segments are inserted and* m_d *segments are deleted in* UPDATE-STATUS, *then the running time of the procedure is* $O((m_i + m_d) \log m)$.

Proof Suppose m_a segments are deleted from Y and L'_v in lines 5 and 8. From Lemma 2.3 we know that the running time to delete these segments is $O(m_a \log m)$. The remaining $m_d - m_a$ segments are deleted from Y in line 1; this takes $O((m_d - m_a) \log m)$ time. For each segment of left(a), it takes $O(\log m)$ time to insert the segment into Y. The total time is thus $O((m_i + m_d) \log m)$. □

Lemma 2.5 *The running time of* CHECK-TRIANGULATION *and* CHECK-CLOSED-TRIANGULATION *is* $O(m \log m)$.

Proof We first analyze the running time of CHECK-TRIANGULATION. Lines 1–6 take $O(m)$ time. Line 7 takes $O(n \log n)$ time by using the prune-and-search algorithm of Kirkpatrick and Seidel [64]. Lines 8–11 take $O(m)$ time. Line 12 takes $O(1)$

time. Line 13 takes $O(n \log n)$ time by using either merge sort or heapsort. In the **for** loop of lines 14–16, each segment is inserted into Y one time (line 2 of UPDATE-STATUS), and deleted from Y one time (line 1 of UPDATE-STATUS or line 3 or 6 of DELETE-SEGMENT). From Lemma 2.4 we know that the running time of lines 14–16 is $O(m \log m)$. Lines 17–22 take $O(m)$ time. Hence, the total time is $O(m \log m)$.

Similarly, we can prove the running time of CHECK-CLOSED-TRIANGULATION is also $O(m \log m)$. □

Remark: Although L'_v lies on the sphere in CHECK-CLOSED-TRIANGULATION, we need not solve nonlinear equations for finding cross-points of circle segments. In fact, when we project $\mathcal{L}_v$ from v onto sphere S_v, we only need to store the two endpoints of each circle segment and its preimage. Notice that computing cross-points of circle segments is only done at one point in the algorithm, namely, line 2 of UPDATE-STATUS. Suppose we want to insert segment aa' into Y in UPDATE-STATUS(a). We need to compute the cross-points of longitude yay' and some segments $pq \in Y$. Let $p'q'$ be the preimage of pq. We solve a system of linear equations consisting of the line $p'q'$ and the plane containing points y, a, y' to find the intersection x_a, and we then project x_a onto S_v. This image of x_a is the intersection of longitude yay' and pq. Therefore, in the whole algorithm, the only computation involving a nonlinear function is the projection of points onto a sphere. This can be done easily.

From the above lemmas, we can finally prove the following theorem.

Theorem 2.8 *If* TETRAHEDRALIZATION-DETECTION *is run on m line segments with n endpoints, then the running time is $O(nm \log n)$ and the running space is $O(n^2)$.*

Proof Line 1 takes $O(1)$ time. Line 2 takes $O(n \log n)$ time. Lines 3–5 take $O(m)$ time. The **for** loop of lines 6–24 iterates at most n times. It follows from Lemma 2.5 that each iteration takes $O(m \log m)$ time. Thus, the running time of lines 6–24 is $O(nm \log m)$. Since a triangulation of n points has $O(n)$ edges and faces, we have $|T_v| = O(n)$. It follows that $|T| = O(n^2)$. Thus, lines 25–29 take $O(n^2)$ time. Therefore, the total time is $O(n \log n + m + nm \log m + n^2) = O(nm \log n)$. Since the space for T is $O(n^2)$, and this dominates all other spaces, the total space is $O(n^2)$. □

Remark: Since there exists a tetrahedralization whose graph is complete [48], from Theorem 2.3 we know that $|T| = |\mathcal{L}| - |V(\mathcal{L})| - |V(CH(\mathcal{L}))| + 3 = O(n^2)$. So the running space of the algorithm is worst-case optimal. As the lower bound for the problem of reporting all line segment intersections is $\Omega(m \log m + k)$, where k is the number of intersection points [9, 25], the sweep algorithms described in CHECK-TRIANGULATION and CHECK-CLOSED-TRIANGULATION are optimal. We conjecture that the running time of TETRAHEDRALIZATION-DETECTION is also optimal in the worst case.

2.5 Deciding whether $\mathcal{L}$ contains a tetrahedralization

In this section, we discuss the complexity of deciding if a set of line segments $\mathcal{L}$ in R^3 contains a tetrahedralization (the TETRAD problem). There are two NP-complete problems related to TETRAD. Ruppert and Seidel proved that the problem of deciding if a 3-dimensional nonconvex polyhedron can be tetrahedralized is NP-complete [88] (the NCP problem). Lloyd proved that the problem of deciding if a line segment set E in R^2 contains a triangulation is NP-complete [74] (the TRID problem). Either of the problems can reduce to TETRAD; and the reduction of NCP to TETRAD is easier than that of TRID to TETRAD. However, we will show a reduction of TRID to TETRAD because the two lemmas that are used to prove this reduction may have other applications independent of this proof.

Let E, a set of line segments in R^2, be an instance of TRID. If we apply the coplanarities of segments, we can easily transform E to an instance of TETRAD by placing a vertex above the plane containing E and connecting this vertex with each endpoint of E. Hence, we only consider the nondegenerate case in our proof. Our aim is to construct a segment set $\mathcal{L}$ without four coplanar endpoints. We first project the vertices of $V(\mathsf{E})$ onto a semi-sphere above E; the new vertex set is denoted by V_U. Secondly, we symmetrically construct another convex point set V_L which is a mirror image of V_U w.r.t. the plane containing E, and we then twist V_L by a small

amount such that no four points are coplanar. Thirdly, we place a vertex x^* above the semi-sphere. Finally, for each segment $ab \in \mathsf{E}$, we add a line segment between the two corresponding points of a and b in V_U (denoted as E_U); we also add edges to each pair of vertices in V_L (denoted as E_L), add edges to each pair of vertices between V_U and V_L (denoted as E_{UL}), and add edges between x^* and each vertex of V_U (denoted as L^*). Let $\mathcal{L} = L^* \cup E_U \cup E_L \cup E_{UL}$. This completes the construction (for more details, see the proof of Theorem 2.9). The following two lemmas solve the problems related to the existence of tetrahedralization in E_L and E_{UL}, respectively.

Lemma 2.6 *Let Q be a simple polyhedron satisfying $V(Q) = V(CH(Q))$. The region R between Q and $CH(Q)$ is tetrahedralizable.*

Proof If Q is convex, then $R = \emptyset$; otherwise, there must exist at least one reflex edge, say ab, on the boundary of Q (denoted as $\mathrm{bd}(Q)$). The two faces on $\mathrm{bd}(Q)$ which share ab are denoted as abc and abd. If $\mathrm{int}(cd) \cap Q \neq \emptyset$, there are two cases as follows.

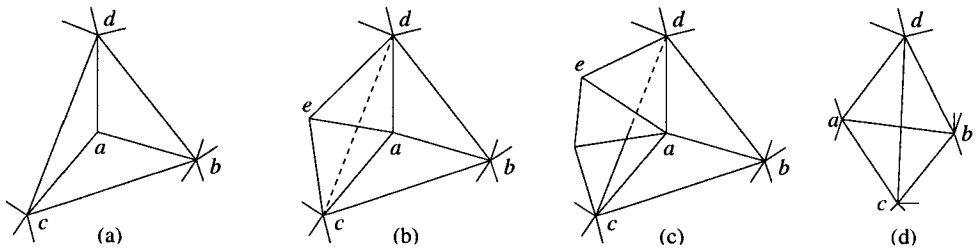

Figure 2.8: (a) $cd \in E(Q)$. (b) (c) $\mathrm{int}(cd) \cap \mathrm{int}(Q) \neq \emptyset$. (d) $\mathrm{int}(cd) \cap Q = \emptyset$.

1. $cd \in E(Q)$. Refer to Figure 2.8(a). If acd and bcd are the two faces on $\mathrm{bd}(Q)$

which share cd, then the four faces abc, abd, acd and bcd form a tetrahedron. This contradicts the assumption that ab is a reflex edge. Thus either acd or bcd is not a face of $\mathrm{bd}(Q)$. Without loss of generality, assume bcd is not a face of $\mathrm{bd}(Q)$ and let $H(bcd)$ be the plane containing it. Suppose a' is an interior point of Q and lies in the neighbourhood of a. Since ab is reflex, the vertex a will be on the same side of $H(bcd)$ as a'. This contradicts the assumption that $a \in V(CH(Q))$.

2. $\mathrm{int}(cd) \cap \mathrm{int}(Q) \neq \emptyset$. Refer to Figure 2.8(b)(c). Let $H(bcd)$ be the plane containing face bcd. Since ab is reflex, there exists at least one edge, say ae, which intersects face bcd. Thus vertices a and e lie on the opposite sides of $H(bcd)$. Because abc and abd lie on $\mathrm{bd}(Q)$, we have $a \notin V(CH(Q))$. This is a contradiction.

Thus $\mathrm{int}(cd) \cap Q = \emptyset$. Refer to Figure 2.8(d). Now we can construct a new polyhedron Q_1 from Q by flipping ab to cd. Thus, $E(Q_1) = (E(Q) - \{ab\}) \cup \{cd\}$ and $F(Q_1) = (F(Q) - \{abc, abd\}) \cup \{acd, bcd\}$. Hence $Q \subset Q_1$, and $Q_1 - Q = \{abcd\}$. Since $V(Q_1) = V(Q) = V(CH(Q))$, we know that $V(Q_1) = V(CH(Q_1))$. Let R_1 denote the region between Q_1 and $CH(Q_1)$. If Q_1 is convex, then $R_1 = \emptyset$ and $R = \{abcd\}$; otherwise, using the same argument, we can construct a polyhedron Q_2 such that $V(Q_2) = V(CH(Q_2))$, $Q_1 \subset Q_2$, and $Q_2 - Q_1 = \{a_1b_1c_1d_1\}$. Since the number of edges is finite, we can perform this procedure until we find a convex polyhedron Q_k

in step k. Since $R_k = CH(Q_k) - Q_k = \emptyset$, we know that R can be tetrahedralized into tetrahedra $abcd, a_1b_1c_1d_1, ..., a_kb_kc_kd_k$. □

Remark: Note that the condition $V(Q) = V(CH(Q))$ is necessary. Without this condition, even if Q is a star-shaped polyhedron, the region $R = CH(Q) - Q$ may be untetrahedralizable. Figure 2.9(a) serves as a counterexample, where $P_A = x^*abc$ is a tetrahedron and $P_B = abca'b'c'$ is a Schönhardt's polyhedron [90] (refer to Section 2.6) inside P_A. Both of them share the face abc. If we remove the face abc, the other faces of P_A and P_B may bound a star-shaped polyhedron, denoted as Q. It is obvious that $V(Q) \neq V(CH(Q))$. Since Schönhardt's polyhedron is untetrahedralizable [90], the region $R = CH(Q) - Q = P_B$ is untetrahedralizable.

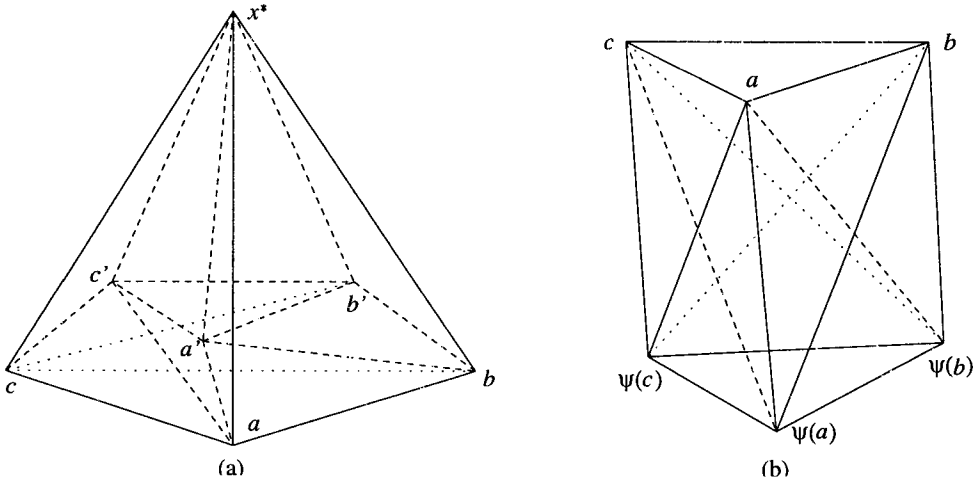

Figure 2.9: (a) $abca'b'c'$ is an instance of Schönhardt's polyhedron, which cannot be untetrahedralized. (b) A regular quasi-prism.

Now let us consider how to tetrahedralize the region between two isomorphic piecewise linear surfaces, which can be applied to find a tetrahedralization in E_{UL}.

Let S_1 and S_2 be two piecewise triangle surfaces (also called triangulations in R^3) such that $V(S_1) = V_U$, $V(S_2) = V_L$ and S_1 is isomorphic to S_2. That is, there exists a bijection $\psi : V(S_1) \rightarrow V(S_2)$ with $xy \in E(S_1) \Leftrightarrow \psi(x)\psi(y) \in E(S_2)$. Construct a geometric graph S_{12} such that $V(S_{12}) = V(S_1) \cup V(S_2)$ and $E(S_{12}) = E(S_1) \cup E(S_2) \cup \{x\psi(x) \mid x \in V(S_1)\}$. If ab is on the boundary of S_1, then we call $ab\psi(b)\psi(a)$ a boundary quadrilateral; moreover, if $a\psi(b) \in E(CH(V(S_{12})))$, then we call $a\psi(b)$ a convex diagonal of the boundary quadrilateral $ab\psi(b)\psi(a)$. By adding the convex diagonal of each boundary quadrilateral of S_{12}, we obtain a new geometric graph S, called a *quasi-prism*. For each face abc on S_1, if $\overline{CH}(\{a, b, \psi(a), \psi(b)\}) \cap \overline{CH}(\{b, c, \psi(b), \psi(c)\}) = \{b\psi(b)\}$, $\overline{CH}(\{a, b, \psi(a), \psi(b)\}) \cap \overline{CH}(\{a, c, \psi(a), \psi(c)\}) = \{a\psi(a)\}$ and $\overline{CH}(\{b, c, \psi(b), \psi(c)\}) \cap \overline{CH}(\{a, c, \psi(a), \psi(c)\}) = \{c\psi(c)\}$, then we call S *regular* (see Figure 2.9(b)).

We can use the following algorithm to tetrahedralize any regular quasi-prism S. This tetrahedralization can be considered as a constrained tetrahedralization since all the edges in S must be used to build the tetrahedralization.

Algorithm QUASI-PRISM-TETRAHEDRALIZATION
Input: A regular quasi-prism S.
Output: A tetrahedralization T of S.
1 $S' \leftarrow S_1$, $T \leftarrow E(S)$, and $A \leftarrow \{x \mid x$ is a boundary vertex in $S_1\}$
2 **while** $A \neq \emptyset$, **do**
3 Choose an arbitrary vertex $a \in A$.
* $\text{adj}_{S'}(a)$ denotes the set of vertices adjacent to a in S' *
4 **if** $\text{adj}_{S'}(a) \neq \emptyset$
5 **then** $T \leftarrow T \cup \{a\psi(x) \mid x \in \text{adj}_{S'}(a)\}$,
6 $A \leftarrow (A - \{a\}) \cup \text{adj}_{S'}(a)$

```
7            else A ← (A − {a})
8         endif
9         S' ← S' − a
10     endwhile
```

The following lemma shows that QUASI-PRISM-TETRAHEDRALIZATION is correct.

Lemma 2.7 *If S is a regular quasi-prism, then* QUASI-PRISM-TETRAHEDRALIZATION *always outputs a tetrahedralization of S.*

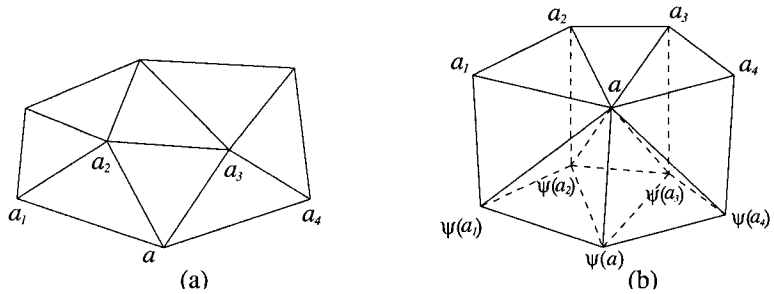

Figure 2.10: (a) The vertices of $\text{adj}_{S'}(a)$ form a chain $a_1a_2a_3a_4$. (b) Six pyramids with apex a.

Proof Let $a \in A$ be chosen in line 3 of the algorithm and suppose $\text{adj}_{S'}(a) \neq \emptyset$ in line 4. If S' is a triangulation, then the vertices in $\text{adj}_{S'}(a)$ form a chain, denoted by $C(a) = a_1a_2...a_k$ (see Figure 2.10(a)). So $T(a) = E(C(a)) \cup \{aa_i \mid 1 \leq i \leq k\}$ is a sub-triangulation in S_1. Similarly, we know that $T(\psi(a)) = E(C(\psi(a))) \cup \{\psi(a)\psi(a_i) \mid 1 \leq i \leq k\}$ is a sub-triangulation in S_2. Thus $S(a) = T(a) \cup T(\psi(a)) \cup \{a\psi(a), a_i\psi(a_i) \mid 1 \leq i \leq k\}$ is a sub-quasi-prism of S. There are two phases to tetrahedralize $S(a)$. In the first phase (line 5), we partition $S(a)$ into a number of pyramids, each with apex a and the base equal to a face of $S(a)$ not incident to a (see Figure 2.10(b)).

There are two kinds of bases, triangular and quadrilateral. Each triangular pyramid will be a tetrahedron in the resulting tetrahedralization. Each quadrilateral pyramid with base $a_i a_{i+1}\psi(a_{i+1})\psi(a_i), 1 \leq i \leq k-1$, can be divided into two tetrahedra in the second phase. That is, when we choose a_i or a_{i+1} in line 3, we will insert edge $a_i\psi(a_{i+1})$ or $a_{i+1}\psi(a_i)$ in line 5. This partitions the quadrilateral pyramid into two triangular pyramids.

If S' is a chain, then $\text{adj}_{S'}(a)$ contains one or two vertices. Let $a_1 \in \text{adj}_{S'}(a)$. When we insert edge $a\psi(a_1)$ (line 5), it divides the quadrilateral $aa_1\psi(a_1)\psi(a)$ into two triangles. This partitions a quadrilateral pyramid produced earlier into two tetrahedra.

The above argument can also apply to other cases in which S' consists of several components. We have proven that the algorithm divides S into a number of triangular pyramids. Now we show that these pyramids intersect only in their shared vertices, edges or facets. Since S can be considered as a set of triangular quasi-prisms, we only need to show that each of them has this property.

Let $aa_1a_2\psi(a)\psi(a_1)\psi(a_2)$ be an arbitrary triangular quasi-prism in the above procedure when we choose a in line 3 (refer to Figure 2.10(b)). Since S is regular, the three tetrahedra $aa_1\psi(a_1)\psi(a)$, $aa_2\psi(a_2)\psi(a)$ and $a_1a_2\psi(a_2)\psi(a_1)$ are pairwise disjoint except at the shared edges. Thus, the three triangular pyramids with bases $\psi(a)\psi(a_1)\psi(a_2)$, $a_1a_2\psi(a_1)$, $a_2\psi(a_1)\psi(a_2)$, or $\psi(a)\psi(a_1)\psi(a_2)$, $a_1a_2\psi(a_2)$, $a_1\psi(a_1)\psi(a_2)$

are disjoint except at the shared vertices, edges, or faces. □

Theorem 2.9 *The* TETRAD *problem is NP-complete.*

Proof We first show that the TETRAD problem belongs to NP. Suppose we are given a set of segments $\mathcal{L}$ in R^3. A nondeterministic algorithm needs only guess a subset $L' \subseteq \mathcal{L}$. From Theorem 2.8 we know that checking whether L' is a tetrahedralization can be accomplished in polynomial time.

We shall show that the TETRAD problem is NP-hard by proving a reduction of TRID to TETRAD. Given a set of line segments E in a plane H (an instance of TRID), the reduction algorithm constructs a set of line segments $\mathcal{L}$ in R^3 with no four endpoints coplanar (an instance of TETRAD) such that E contains a triangulation if and only if $\mathcal{L}$ contains a tetrahedralization.

Let x be the center of the smallest circle containing E and r be the radius of the circle. Construct a big sphere S with center x and a very large radius, say $100r$. The plane H cuts S into two parts. The upper semi-sphere is denoted as S_U and the lower semi-sphere is denoted as S_L. Vertically project the vertices of $V(\mathsf{E})$ onto S_U and S_L, and denote them as V_U and V_L, respectively. For each $ab \in \mathsf{E}$, add a line segment between the two corresponding points of a and b in V_U. So we obtain a new edge set E_U corresponding to E. Let E_L be the set of all the edges with endpoints in V_L, and E_{UL} be the set of all the edges between V_U and V_L. Place a vertex x^* in the vertical

line through x and far above S_U such that it can see all the vertices of V_U, that is, for any $u \in V_U$, $\text{int}(x^*u) \cap S_U = \emptyset$. Let $L^* = \{x^*u \mid u \in V_U\}$.

Let $\mathcal{L} = L^* \cup E_U \cup E_L \cup E_{UL}$. Notice that each triangle face in E_U and its corresponding triangle face in E_L form a triangular prism in $\mathcal{L}$. In order to prevent four vertices from lying in the same plane, we twist S_L (with V_L together) by a small amount such that no four vertices are coplanar and each prism become a regular quasi-prism.

We have constructed a set of segments $\mathcal{L}$ with no four endpoints coplanar. It is easy to see that this construction can be done in polynomial time. Since the vertical projection of x^* onto H is the center x, any edges that connect x^* with vertices of V_L must lie inside $CH(\mathcal{L})$. Thus, $\mathcal{L}$ has the property that $E(CH(\mathcal{L})) \subseteq \mathcal{L}$.

We now show that E contains a triangulation if and only if $\mathcal{L}$ contains a tetrahedralization. First, suppose that E contains a triangulation T. Then there must exist two corresponding triangulations T_U in E_U and T_L in E_L, respectively. The three triangulations are isomorphic to each other. Thus, there exists a regular quasi-prism with the upper surface T_U and the lower surface T_L. From Lemma 2.7 we know that this regular quasi-prism can be tetrahedralized by using edges in E_{UL}. This tetrahedralization is denoted as T_{UL}. On the other hand, since the boundary of T_{UL} is a simple polyhedron satisfying $V(T_{UL}) = V(CH(T_{UL}))$, it follows from Lemma 2.6 that the region between $CH(T_L)$ and the lower surface T_L can be tetrahedralized by

using edges in E_L. This tetrahedralization is denoted as T_{LR}. Since x^* can see all the vertices of V_U, T_U and L^* form a tetrahedralization denoted as T_{L^*}. Therefore T_{UL}, T_{LR} and T_{L^*} constitute a tetrahedralization in $\mathcal{L}$.

Conversely, suppose that $\mathcal{L}$ contains a tetrahedralization Γ. We first prove that $L^* \subset E(\Gamma)$. Since $E(CH(\mathcal{L})) \subset E(\Gamma)$, the edges incident to x^* in $E(CH(L^*))$ belong to $E(\Gamma)$. If there exists a segment $x^*v \in L^*$ which does not belong to $E(\Gamma)$, then x^*v must intersect a triangle face in the tetrahedralization Γ. This is a contradiction because v lies on the upper semi-sphere. Thus, $L^* \subset E(\Gamma)$. In the tetrahedralization Γ, the bases of the triangular pyramids with apex x^* form a triangulation T_{x^*} in E_U. Hence E also contains a triangulation that corresponds to T_{x^*}. □

2.6 Constrained tetrahedralizations

Let S be a planar straight line graph (PSLG) which is defined as a set of vertices and noncrossing (that is, intersecting only at endpoints) line segments in the plane. A *constrained triangulation* of S is a triangulation of $V(S)$ such that all the edges of $E(S)$ are used as edges in the triangulation. For any PSLG, there exist several knids of constrained triangulations; and among these types, the constrained Delaunay triangulations have many applications [35, 38, 50, 77]. Similarly, we can define a *constrained tetrahedralization* for a set G of vertices and noncrossing line segments in the space as a tetrahedralization of the vertices and endpoints $V(G)$ such that all the

segments $E(G)$ are used as edges in the tetrahedralization [8, 20, 34, 95, 98]. We also say that G can be tetrahedralized. In this section, we discuss the complexity problem of deciding if a set of vertices and line segments $\mathcal{G}$ in R^3 can be tetrahedralized (briefly, the CT problem).

Ruppert and Seidel proved that it is NP-complete to tetrahedralize a 3-dimensional nonconvex polyhedron [88]. We shall extend their proof to show that the CT problem is also NP-complete. They proved this NP-completeness result by using a transformation from the Satisfiability problem [51]. That is, for any Boolean formula in conjunctive normal form, a nonconvex polyhedron is constructed such that it can be tetrahedralized if and only if the Boolean formula is satisfiable. The main tool in their construction is a gadget, called a *niche*, which is derived from Schönhardt's polyhedron [90]. As illustrated in Figure 2.11(a), the way to construct Schönhardt's polyhedron is as follows. Start with a triangular prism; fix the bottom face $a_1b_1c_1$ so that it cannot move; twist the top face abc so that each rectangular face of the prism folds into two triangles with a reflex edge between them. If the top face abc is removed from the Schönhardt's polyhedron, then the resulting figure that consists of seven triangular faces on six vertices is called a *niche*; and face $a_1b_1c_1$ is called the *base* of the niche. In this section, we use S_N to denote a niche, and $\overline{S}_N$ to denote the corresponding Schönhardt's polyhedron.

Definition 2.4 *A point p can* see *a point q in the base $a_1b_1c_1$ of a niche S_N if the*

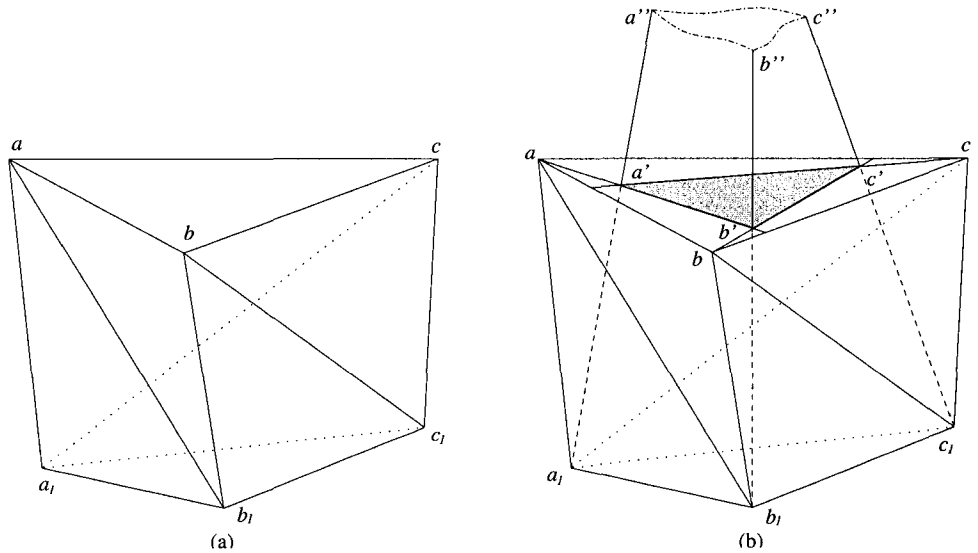

Figure 2.11: (a) A Schönhardt's polyhedron that cannot be tetrahedralized. (b) aa_1b_1b', bb_1c_1c' and cc_1a_1a' are plane quadrilaterals. view$(a_1b_1c_1)$ is the truncated cone $a_1b_1c_1a''b''c''$; illum$(a_1b_1c_1)$ is the truncated cone $a'b'c'a''b''c''$.

interior of the segment pq intersects the interior of $\overline{S}_N$ *but does not intersect any faces of* S_N. *The set of points that can see the entire base* $a_1b_1c_1$ *of* S_N *is called the* `view` *of* S_N, *denoted as* view$(a_1b_1c_1)$. *The subset of* view$(a_1b_1c_1)$ *in the exterior of* $\overline{S}_N$ *is called the* `illuminant` *of* S_N, *denoted as* illum$(a_1b_1c_1)$.

Let $S_N = \{a_1b_1c_1, aa_1b_1, bb_1c_1, cc_1a_1, abb_1, bcc_1, caa_1\}$ with base $a_1b_1c_1$. Suppose the three planes passing through the faces aa_1b_1, bb_1c_1, and cc_1a_1, respectively, intersect at point v. They produce eight cones with v as the apex. We are interested in the cone containing base $a_1b_1c_1$, which is called the *view cone* of $a_1b_1c_1$. The objects view$(a_1b_1c_1)$ and illum$(a_1b_1c_1)$ can be considered as truncated cones (see Figure 2.11(b)). Note that each point of view$(a_1b_1c_1)$ can see the seven faces of S_N from the inside; moreover, if the apex v is below the base $a_1b_1c_1$, then all

the points in the symmetric cone can see the seven faces of S_N from the outside as shown in Figure 2.12(a). If we twist the top face abc of S_N further, view$(a_1b_1c_1)$ and illum$(a_1b_1c_1)$ will shrink and view$(a_1b_1c_1)$ will become a tetrahedron inside $\overline{S}_N$ and illum$(a_1b_1c_1) = \emptyset$, as shown in Figure 2.12(b).

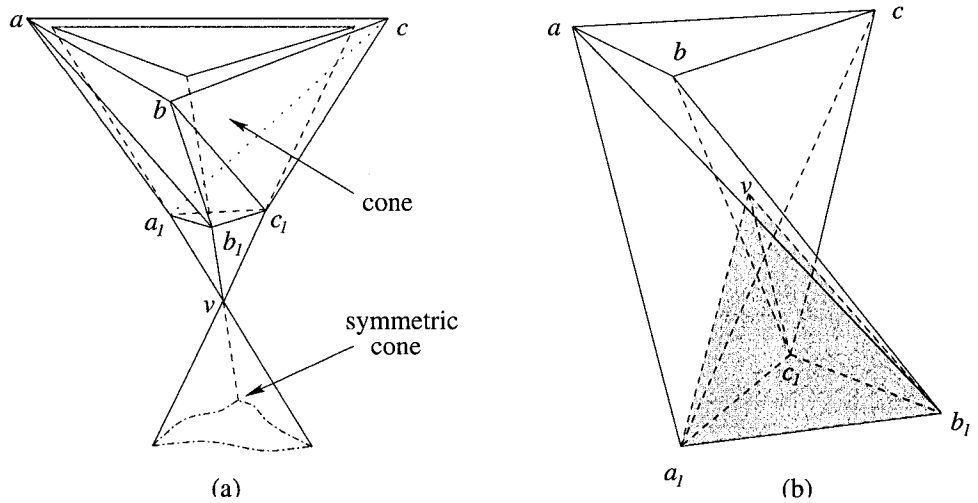

Figure 2.12: (a) v is the apex of the double cone. (b) illum$(a_1b_1c_1) = \emptyset$ and view$(a_1b_1c_1) =$ int$(va_1b_1c_1)$ (the shaded tetrahedron).

Lemma 2.8 *Let* $S_N = \{a_1b_1c_1, aa_1b_1, bb_1c_1, cc_1a_1, abb_1, bcc_1, caa_1\}$ *be a niche with base* $a_1b_1c_1$, *as shown in Figure 2.13,* $a_ib_jc_k$ *be a triangle such that* a_ib_j, a_ic_k *puncture face* aa_1b_1, b_ja_i, b_jc_k *puncture face* bb_1c_1, *and* c_kb_j, c_ka_i *puncture face* cc_1a_1. *Then* $S'_N = \{a_ib_jc_k, aa_ib_j, bb_jc_k, cc_ka_i, abb_j, bcc_k, caa_i\}$ *is a niche and* illum$(a_ib_jc_k) \subset$ illum$(a_1b_1c_1)$.

Proof Refer to Figure 2.13. Since a_ib lies outside $\overline{S}_N$ and ab_j punctures face bb_1c_1, we know that a_ib lies outside $\overline{S}'_N$. Similarly, b_jc and c_ka lie outside $\overline{S}'_N$. So $\overline{S}'_N$ is a niche. We now show that illum$(a_ib_jc_k) \subset$ illum$(a_1b_1c_1)$. If illum$(a_ib_jc_k) = \emptyset$, this

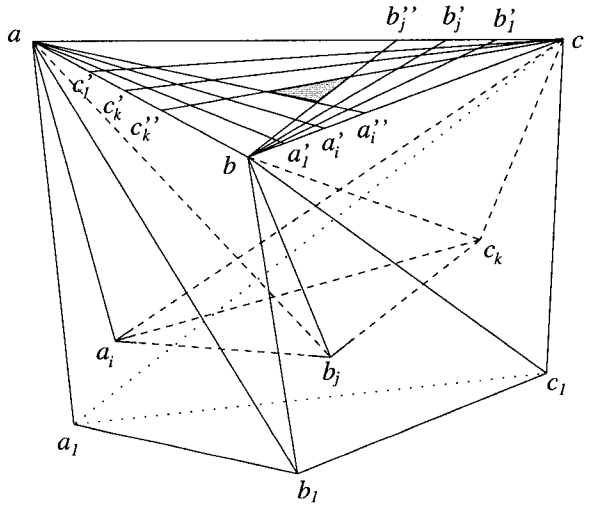

Figure 2.13: $\{a_1b_1c_1, aa_1b_1, bb_1c_1, cc_1a_1, abb_1, bcc_1, caa_1\}$ and $\{a_ib_jc_k, aa_ib_j, bb_jc_k, cc_ka_i, abb_j, bcc_k, caa_i\}$ form two niches respectively.

is trivial. So we assume that $\text{illum}(a_ib_jc_k) \neq \emptyset$. Let aa_1', aa_i', aa_i'' be the intersection segments of face abc and the planes passing through $aa_1b_1, aa_ib_1, aa_ib_j$ respectively. Since a_i lies outside $\overline{S}_N$ and the planes containing aa_1b_1 and aa_ib_1 intersect at line ab_1, we have $\text{dist}(b, a_1') < \text{dist}(b, a_i')$. Since a_ib_j punctures aa_1b_1 and the planes containing aa_ib_1 and aa_ib_j intersect at line aa_i, we have $\text{dist}(b, a_i') < \text{dist}(b, a_i'')$. Thus $\text{dist}(b, a_1') < \text{dist}(b, a_i'')$. Similarly, $\text{dist}(c, b_1') < \text{dist}(c, b_j'')$ and $\text{dist}(a, c_1') < \text{dist}(a, c_k'')$. Hence, the triangle face bounded by aa_1', bb_1', cc_1' contains the triangle face bounded by aa_i'', bb_j'', cc_k''. Note that $\text{illum}(a_1b_1c_1)$ is bounded by face abc and the three planes containing ab_1a_1', bc_1b_1' and ca_1c_1', and $\text{illum}(a_ib_jc_k)$ is bounded by face abc and the three planes containing ab_ja_i'', bc_kb_j'' and ca_ic_k''. It is easy to see that $\text{illum}(a_ib_jc_k) \subset \text{illum}(a_1b_1c_1)$. □

Remark: Although $\text{illum}(a_ib_jc_k) \subset \text{illum}(a_1b_1c_1)$, $\text{view}(a_ib_jc_k)$ may not be contained

in view($a_1b_1c_1$). Since a_i lies outside $\overline{S}_N$, there exists a point $q \in$ view($a_ib_jc_k$) which is in the neighbourhood of a_i such that q lies outside $\overline{S}_N$. Thus view($a_ib_jc_k$) $\not\subset$ view($a_1b_1c_1$).

Theorem 2.10 *Let $\mathcal{L}$ be a set of segments containing a niche S_N with base $a_1b_1c_1$. If* view($a_1b_1c_1$) $\cap\, V(\mathcal{L}) = \emptyset$, *then $\mathcal{L}$ cannot be tetrahedralized.*

Proof Suppose $\mathcal{L}$ can be tetrahedralized and T is the tetrahedron set of the tetrahedralization. Let $S_N = \{a_1b_1c_1, aa_1b_1, bb_1c_1, cc_1a_1, abb_1, bcc_1, caa_1\}$ with base $a_1b_1c_1$, as shown in Figure 2.14. Consider face $a_1b_1c_1$. From Theorem 2.1, we know that there must exist a vertex a_2 in the niche side of the plane containing $a_1b_1c_1$ such that tetrahedron $a_2a_1b_1c_1 \in T$. Since view($a_1b_1c_1$)$\cap V(\mathcal{L}) = \emptyset$, a_2 lies outside view($a_1b_1c_1$). So a_2a_1, a_2b_1 or a_2c_1 must puncture a facet of $\overline{S}_N$. Since $a_2a_1b_1c_1$ is $\mathcal{L}$-empty, a_2 cannot lie between view($a_1b_1c_1$) and $\overline{S}_N$, and a_2a_1, a_2b_1, or a_2c_1 cannot puncture facets abc, abb_1, bcc_1, or caa_1. Thus, without loss of generality, suppose a_2c_1 punctures aa_1b_1. Consider face $a_2b_1c_1$. There exists a vertex b_2 such that tetrahedron $a_2b_2b_1c_1 \in T$. It follows from Lemma 2.8 that $\{a_2b_1c_1, aa_2b_1, bb_1c_1, cc_1a_2, abb_1, bcc_1, caa_2\}$ is a niche with base $a_2b_1c_1$, and illum($a_2b_1c_1$) $\subset$ illum($a_1b_1c_1$). Similar to the above argument, we can suppose a_2b_2 punctures bb_1c_1. Then consider $a_2b_2c_1$, and so on, as shown in Figure 2.14. In general, consider $a_ib_jc_k$ with a_ib_j, a_ic_k puncturing aa_1b_1, b_ja_i, b_jc_k puncturing bb_1c_1, and c_kb_j, c_ka_i puncturing cc_1a_1. From Theorem 2.1 we know that there exists a vertex p such that tetrahedron $pa_ib_jc_k \in T$. It follows from

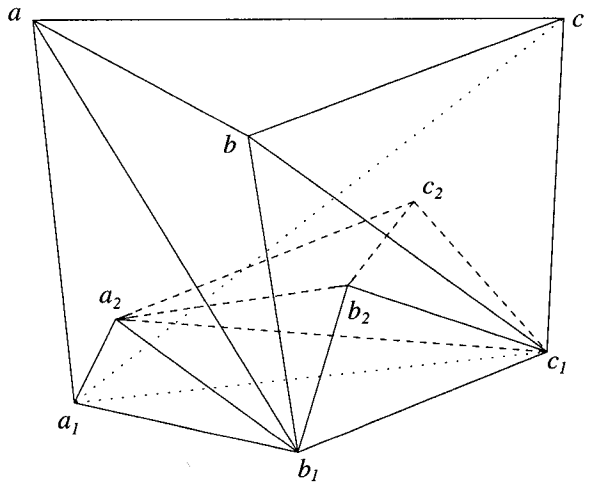

Figure 2.14: view$(a_1b_1c_1) \cap V(\mathcal{L}) = \emptyset$ and tetrahedra $a_2a_1b_1c_1, a_2b_2b_1c_1, a_2b_2c_2c_1$ are $\mathcal{L}$-empty.

Lemma 2.8 that $\{a_ib_jc_k, aa_ib_j, bb_jc_k, cc_ka_i, abb_j, bcc_k, caa_i\}$ is a niche with base $a_ib_jc_k$ and illum$(a_ib_jc_k) \subset$ illum$(a_1b_1c_1)$. Since view$(a_1b_1c_1) \cap V(\mathcal{L}) = \emptyset$, p must lie outside view$(a_1b_1c_1)$. Since a_i, b_j, c_k lie outside $\overline{S}_N$, pa_i, pb_j, or pc_k must puncture a facet of $\overline{S}_N$. Since $pa_ib_jc_k$ is $\mathcal{L}$-empty, p cannot lie between view$(a_1b_1c_1)$ and $\overline{S}_N$, and pa_i, pb_j, or pc_k cannot puncture faces abc, abb_1, bcc_1 or caa_1. Thus, if pc_k punctures aa_1b_1, denote p by a_{i+1} and then consider $a_{i+1}b_jc_k$; if pa_i punctures bb_1c_1, denote p by b_{j+1} and then consider $a_ib_{j+1}c_k$; if pb_j punctures cc_1a_1, denote p by c_{k+1} and then consider $a_ib_jc_{k+1}$. Since the number of vertices in $V(\mathcal{L})$ is finite, this process will stop when we consider a face such as $a_{i'}b_{j'}c_{k'}$. That is, there does not exist a vertex p such that $pa_{i'}b_{j'}c_{k'}$ is an $\mathcal{L}$-empty tetrahedron. Since $a_{i'}b_{j'}c_{k'}$ is an inner triangle, this contradicts Theorem 2.1. Therefore $\mathcal{L}$ cannot be tetrahedralized. □

Let $\mathcal{P}$ be the nonconvex polyhedron constructed in [88] for showing the NP-

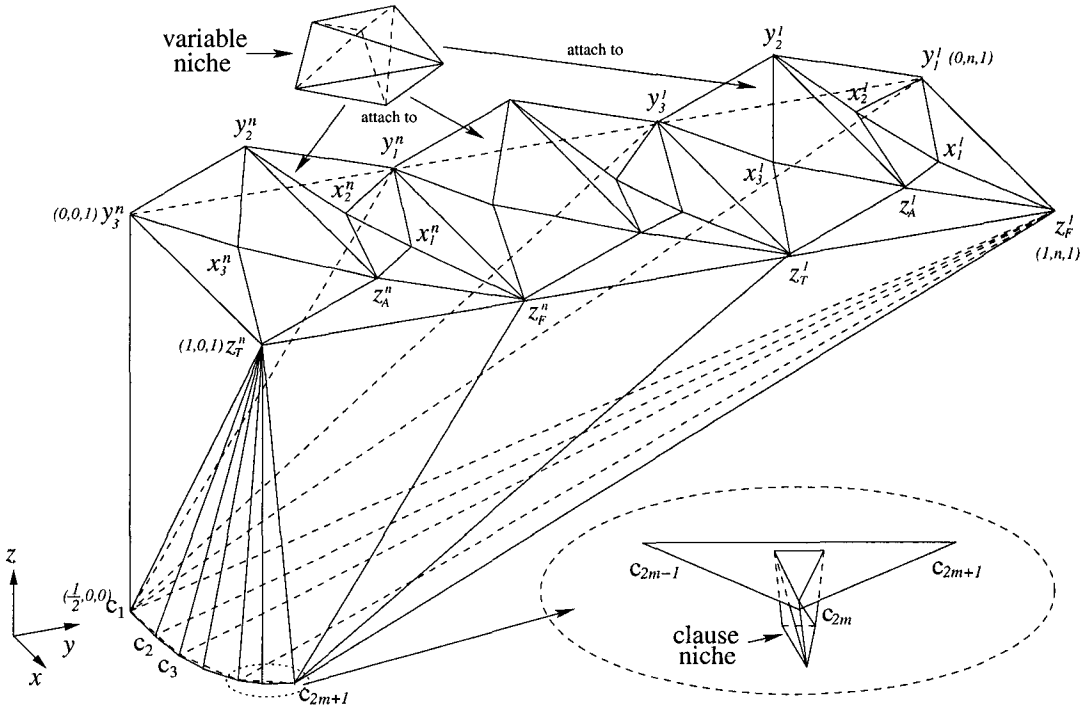

Figure 2.15: Polyhedron $\mathcal{P}$ constructed in [88] (not to scale).

completeness of tetrahedralizing nonconvex polyhedra. As illustrated in Figure 2.15, the starting point for constructing $\mathcal{P}$ is the polyhedron consisting of a row of n squares $y_1^i y_3^i z_T^i z_F^i (1 \leq i \leq n)$ in plane $z = 1$ and m triangles $c_{2k-1}c_{2k}c_{2k+1} (1 \leq k \leq m)$ in plane $z = 0$. The coordinates of the squares' vertices are $y_1^i(0, n+1-i, 1)$, $y_3^i(0, n-i, 1)$, $z_T^i(1, n-i, 1)$, and $z_F^i(1, n+1-i, 1)$. Note that $y_3^i = y_1^{i+1}$ and $z_T^i = z_F^{i+1}$. Vertices $c_k (1 \leq k \leq m)$ lie on a parabola in the xy-plane. The coordinates of c_k are $(\alpha_k, \beta_k, 0)$, where $\alpha_k = 0.5 + 0.005(k-1)/m$ and $\beta_k = 0.25\alpha_k^2 - 0.25\alpha_k + 0.0625$. To each square $y_1^i y_3^i z_T^i z_F^i$, a roof is attached which contains the variable's three literal vertices x_1^i, x_2^i and x_3^i. As mentioned above, niches are used as gadgets that force any tetrahedralization to have certain properties; for example, they can force tetrahedra to appear. For each variable, there is a variable niche attached to $y_1^i y_2^i y_3^i$ of the roof. For each

clause, there is a clause niche attached to $c_{2k-1}c_{2k}c_{2k+1}$ in the bottom. Refer to [88] for the detailed placement and coordinates of the roofs and niches.

The ideas of our proof of NP-completeness is as follows. Since the boundary of the constrained tetrahedralization is a convex hull, we intend to construct a convex polyhedron that contains $\mathcal{P}$. If we simply use $CH(\mathcal{P})$ as our convex polyhedron, it is not clear how to tetrahedralize the region between $\mathcal{P}$ and $CH(\mathcal{P})$ because the boundary of $\mathcal{P}$ is too complicated. In order to simplify the boundary of $\mathcal{P}$, we cover each niche by using a *cap* that is defined as a pyramid with base $y_1^i y_2^i y_3^i (1 \leq i \leq n)$ or $c_{2k-1}c_{2k}c_{2k+1} (1 \leq k \leq m)$ (see Figure 2.16 or 2.17). From the structure of $\mathcal{P}$, we know that the apex of the view cone lies outside $\mathcal{P}$. So we can place the cap's tip inside the symmetric cone of the view cone such that it can see the seven faces of the niche from the niche's "outside" (see Figure 2.12(a)). Hence, the region between the niche and its cap can be tetrahedralized by adding segments between the cap's tip and all other vertices. The boundary of $\mathcal{P}$ is updated by these caps. Since the new boundary is simple, we can tetrahedralize the region between it and its convex hull. Note that in our construction, each face of the niche is an inner face. So segments can puncture these faces. We avoid these cases by using a similar argument to the proof of Theorem 2.10.

Theorem 2.11 *The CT problem is NP-complete.*

Proof We first show that CT is in NP. For a given set $\mathcal{G}$ of vertices and segments in

R^3, a nondeterministic algorithm need only guess a tetrahedralization of $V(\mathcal{G})$. It is easy to see that checking whether $\mathcal{G}$ is a subset of the tetrahedralization can be done in polynomial time.

We next prove that CT is NP-hard by showing that a restricted version of the Satisfiability problem can be reduced to CT in polynomial time. The reduction algorithm begins with an instance of the Satisfiability problem. Let ϕ be a Boolean formula with m clauses over n variables. We restrict ϕ to conjunctive normal form in which each variable appears exactly three times in three different clauses, once as a negative literal, and twice as a positive literal. We shall construct a set of segments $\mathcal{G}$ such that ϕ is satisfiable if and only if $\mathcal{G}$ is tetrahedralizable. Our construction is based on the nonconvex polyhedron $\mathcal{P}$ described above.

For any variable niche with base $q_1^i q_2^i q_3^i (1 \leq i \leq n)$, let q^i be the apex of the view cone of $q_1^i q_2^i q_3^i$ which lies outside $\mathcal{P}$. Note that in the construction of $\mathcal{P}$ in [88], the variable niche is specified in detail because its base $q_1^i q_2^i q_3^i$ must see the two truth-setting vertices z_T^i and z_F^i, and the three literal vertices x_1^i, x_2^i, x_3^i on the roof of the variable must satisfy the "blocking" condition for the clause niches. From the placement of the variable niche, we know that the x-coordinate of q^i is greater than -1. So we can select a vertex a^i in the plane $x = -1$ such that a^i can see the seven faces of the niche from the outside. Then connect a^i with the six vertices of the niche and the three vertices of face $y_1^i y_2^i y_3^i$ which contains the niche such that the 13 facets of $\mathcal{P}$ bounded by triangle $y_1^i y_2^i y_3^i$ form 13 tetrahedra with a^i (see Figure 2.16). So we

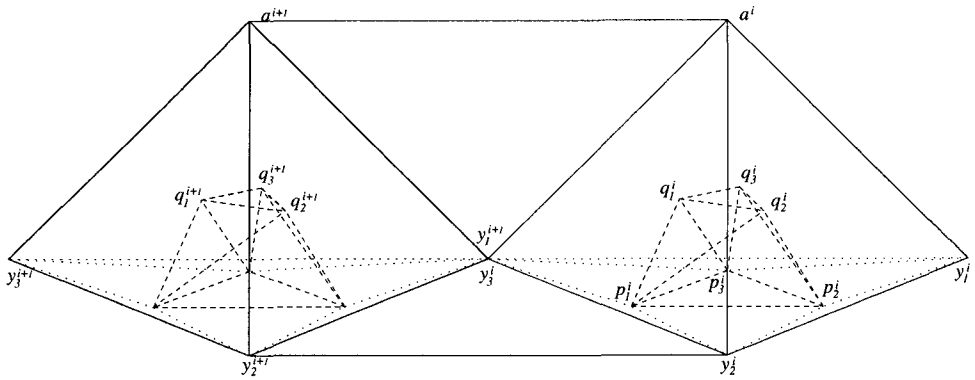

Figure 2.16: $a^i y_1^i y_2^i y_3^i$ is the cap of the niche. a^i can see the seven faces of the niche and six faces on face $y_1^i y_2^i y_3^i$.

can update the boundary of $\mathcal{P}$ by using facets $a^i y_1^i y_2^i$, $a^i y_2^i y_3^i$, $a^i y_3^i y_1^i$ $(1 \leq i \leq n)$ to replace the 13 facets in each cap. In particular, from the structure of $\mathcal{P}$, we can select a row of points in the plane $x = -1$ such that segments $a^i a^{i+1}$, $y_2^i y_2^{i+1}$ $(1 \leq i \leq n-1)$ and $a^i y_2^i (1 \leq i \leq n)$ lie on the convex hull of $V(\mathcal{P}) \cup \{a^i \mid 1 \leq i \leq n\}$. Let $\mathcal{P}_1$ be the updated boundary of $\mathcal{P}$.

The construction of the clause niche is much simpler than that of the variable niche. For the kth clause niche, its illuminant contains the segment with endpoints $(\alpha_{2k}, 0, 1)$ and $(\alpha_{2k}, n, 1)$, where $\alpha_{2k} = 0.5 + 0.005(2k-1)/m$. Since this segment intersects $y_1^1 z_F^1$ and $y_3^n z_T^n$, the kth niche must be contained in the region bounded by face $c_{2k-1}c_{2k}c_{2k+1}$ and the planes passing through $z_F^1 c_{2k-1}c_{2k+1}$, $z_T^n c_{2k-1}c_{2k}$ and $z_T^n c_{2k}c_{2k+1}$ respectively. From the illuminant lemma [88] we can construct the niche in this region as small as possible such that $a^i (1 \leq i \leq n)$ cannot see $b_k (1 \leq k \leq m)$ (to be specified later). Similarly to the placement of a^i, for the kth clause niche, select a vertex b_k in the plane $z = -0.00001$ such that b_k can see the seven faces of the niche

from the outside but b_k cannot see $a^i (1 \leq i \leq n)$. Then connect b_k with the six vertices of the niche and the three vertices of the triangle face to which the niche is attached. Then we update the current boundary of $\mathcal{P}_1$ by using three facets to replace the 13 facets in each cap. In particular, we place $b_k (1 \leq k \leq m)$ in the plane $z = -0.00001$ such that faces $b_k c_{2k-1} c_{2k}$, $b_k c_{2k} c_{2k+1} (1 \leq k \leq m)$ and $b_k b_{k+1} c_{2k+1} (1 \leq k \leq m-1)$ lie on the convex hull of $\mathcal{P} \cup \{a^i \mid 1 \leq i \leq n\} \cup \{b_k \mid 1 \leq k \leq m\}$ (see Figure 2.17). Let $\mathcal{P}_2$ be the updated boundary of $\mathcal{P}_1$.

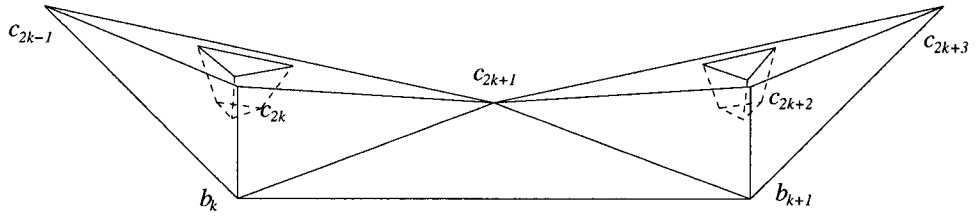

Figure 2.17: $b_k c_{2k-1} c_{2k} c_{2k+1}$ is the cap of the niche. b_k can see the seven faces of the niche and six faces inside $c_{2k-1} c_{2k} c_{2k+1}$.

It is easy to see that the region between $\mathcal{P}_2$ and $CH(\mathcal{P}_2)$ consists of three parts. The first part is the convex polyhedron with vertex set $\{a^i \mid 1 \leq i \leq n\} \cup \{y_1^i \mid 1 \leq i \leq n\} \cup \{y_3^n, c_1\}$. The second part is the convex polyhedron with vertex set $\{b_k \mid 1 \leq k \leq m\} \cup \{c_{2k+1} \mid 0 \leq k \leq m\} \cup \{y_1^1, z_F^1\}$. So these two parts can be tetrahedralized easily by adding some segments. The third part consists of $n-1$ wedge-shaped polyhedra between the n variable roofs. Since each roof is convex, each wedge-shaped polyhedron can be tetrahedralized by adding some segments [54].

Let $\mathcal{G}$ be the segment set consisting of $E(\mathcal{P})$, $E(CH(\mathcal{P}_2))$ and all the segments we have added between $\mathcal{P}$ and $CH(\mathcal{P}_2)$. $\mathcal{G}$ can be considered as the edge set of

the tetrahedralization between $\mathcal{P}$ and $CH(\mathcal{P}_2)$. Since $\mathcal{P}$ and $\mathcal{P}_2$ can be constructed in polynomial time, and $n-1$ wedge-shaped polyhedra can be tetrahedralized in polynomial time, $\mathcal{G}$ can be constructed in polynomial time.

We claim that formula ϕ is satisfiable if and only if $\mathcal{G}$ is tetrahedralizable. We first suppose that there exists a constrained tetrahedralization T for $\mathcal{G}$. For any variable niche $p_1p_2p_3q_1q_2q_3$ (see Figure 2.16) in $\mathcal{P}$ with base $q_1q_2q_3$ (for simplicity, we drop the i superscripts), it follows from Theorem 2.1 that there exists a vertex q_4 in the niche side of the plane passing through $q_1q_2q_3$ such that $q_1q_2q_3q_4 \in T$. Note that $\text{view}(q_1q_2q_3) \cap V(\mathcal{G}) = \{z_T, z_F\}$. If $q_4 \notin \{z_T, z_F\}$, then using an argument similar to the proof of Theorem 2.10, q_1q_4, q_2q_4, or q_3q_4 can only puncture the faces $p_1q_1q_2$, $p_2q_2q_3$, or $p_3q_3q_1$. Without loss of generality, suppose q_3q_4 punctures $p_1q_1q_2$. From the structure of $\mathcal{G}$, we know that no other points can form an $\mathcal{L}$-empty tetrahedron with face $q_2q_3q_4$. This is a contradiction to Theorem 2.1. Thus q_4 must lie in the illuminant of the niche. Similarly, for any clause niche in $\mathcal{P}$, the vertex which forms a tetrahedron with the base of the niche must lie inside its illuminant. From the proof of Theorem 1 in [88], we can obtain a truth assignment for ϕ. Conversely, suppose ϕ is satisfiable. Using the method of [88], we can add some segments to tetrahedralize $\mathcal{P}$. These added segments together with $\mathcal{G}$ form a constrained tetrahedralization of $V(\mathcal{G})$. □

Remark: Note that the NP-completeness of tetrahedralizing nonconvex polyhedra does not depend on coplanarities of faces or other degeneracies. The structure of $\mathcal{P}$

can be modified easily so that the vertices of $\mathcal{P}$ are in a nondegenerate position [88, page 250]. Consequently, we can also modify the structure of $\mathcal{G}$ so that the vertices of $\mathcal{G}$ are in a nondegenerate position. Thus, the NP-completeness result is still valid for tetrahedralizing a set of vertices and segments in the general position.

2.7 Conclusions

In this chapter, we have given some new results on the computational complexity of tetrahedralization detections. We have presented an $O(nm \log n)$ algorithm to determine whether $\mathcal{L}$ is a tetrahedralization. If $\mathcal{L}$ is not a tetrahedralization, we have proven that it is NP-complete to decide whether $\mathcal{L}$ contains a tetrahedralization of $V(\mathcal{L})$, and we have also proven that it is NP-complete to decide whether $\mathcal{L}$ is a subset of a tetrahedralization of $V(\mathcal{L})$. The proofs of the NP-completeness results are constructive. The former is constructed from its two dimensional analog [74]. The latter is constructed from the nonconvex polyhedron used in [88].

We conclude this chapter with some open problems that in one way or the other relate to the material discussed in this chapter.

- Given a set of line segments $\mathcal{L}$ in R^3, how can we efficiently detect if $\mathcal{L}$ forms a simple polyhedron ?

- How hard is it to decide if a collection of *sticks* (i.e., line segments subject to

rigid motions) can be joined to form a triangulation (or tetrahedralization)?

- Given a collection of triangles (resp. tetrahedra) without adjacency information, how hard is it to decide if they can be assembled into a triangulation (resp. tetrahedralization) ?

- How hard is it to decide if a collection of sticks can be joined to form a convex polyhedron ?

- Given a collection of polygons without adjacency information, how hard is it to decide if they can be assembled into a convex polyhedron ?

Chapter 3

Minimal Tetrahedralizations of a Class of Polyhedra

Given a simple polyhedron in three dimensional Euclidean space, different tetrahedralizations of that polyhedron may contain different numbers of tetrahedra. A minimal tetrahedralization is a tetrahedralization with the minimum number of tetrahedra. In Section 3.1, we survey related results in this area. In Section 3.2, we present several properties of the graph of polyhedra. In Section 3.3, we prove various properties of the structure of the minimal tetrahedralizations of BP-polyhedra and some special two-level BP-polyhedra defined in Section 3.3.1. In Section 3.4, we present a quadratic time algorithm for identifying and tetrahedralizing an augmented BP-polyhedra. Finally, we discuss issues arising from these results in Section 3.5.

3.1 Introduction

A *tetrahedralization* of a polyhedron is a partition of that polyhedron into a number of tetrahedra that meet only at shared vertices, edges, or triangles. All convex polyhedra are tetrahedralizable, but not all nonconvex polyhedra can be tetrahedralized [71, 90]. If Steiner points are allowed, then all polyhedra are tetrahedralizable. Chazelle [24] showed that a simple polyhedron of n vertices can always be partitioned into $O(n^2)$ convex regions if Steiner points are allowed, and this bound is tight in the worst case. Clearly, this is also a tight bound for tetrahedralization in the worst case.

For optimal tetrahedralization problems, very few results are known. Since different tetrahedralizations of the same polyhedron may contain different numbers of tetrahedra, a natural optimization problem is how to tetrahedralize a polyhedron with the minimum number of tetrahedra. A tetrahedralization with the minimum number of tetrahedra is referred to as a *minimal tetrahedralization.* Recently, Below *et al.* [12] proved that the minimum number of tetrahedra in the tetrahedralization of a convex polyhedron can be decreased if Steiner points are allowed. Chin *et al.* [33, 32] studied approximation algorithms for computing the minimal tetrahedralization of a convex polyhedron. Richter-Gebert [84] proved that finding a minimal tetrahedralization of the boundary of a 4-polytope is NP-complete. Furthermore, Below *et al.* [13, 14] proved that finding a minimal tetrahedralization of a convex polyhedron is NP-complete. Thus, it becomes significant to study which kind of polyhedra can be

minimally tetrahedralized in polynomial time. It is obvious that stacked polyhedra can be minimally tetrahedralized in $O(n^2)$ time [86, 39]. Wang and Yang [106] presented another kind of special polyhedra, the so-called k-vet polyhedra, and proved that the k-vet polyhedra can be minimally tetrahedralized under some conditions.

In this chapter, we present some properties of the graph of polyhedra. Then we identify a class of polyhedra called BP-polyhedra and show that this class of polyhedra can be minimally tetrahedralized in $O(n^2)$ time.

3.2 Properties of the graph of polyhedra

3.2.1 Definitions and assumptions

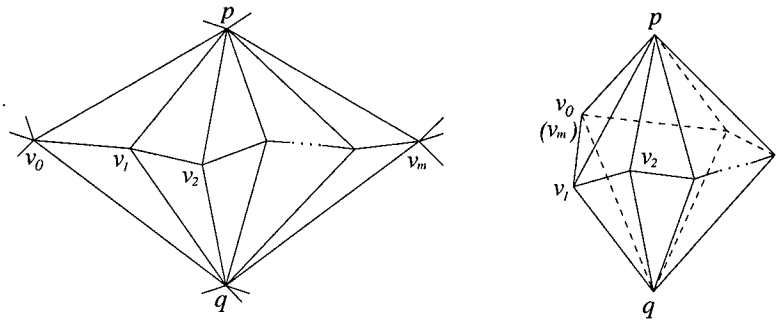

Figure 3.1: A BP-net (left) and a bipyramid (right).

Let $\mathcal{P}$ be a simple polyhedron. As in Chapter 2, we assume throughout this chapter that the vertices of $\mathcal{P}$ are in a general position such that no four vertices of $\mathcal{P}$ are coplanar. Note that vertices not in a general position can be perturbed by various

methods [48, 91]. This assumption implies that each facet of $\mathcal{P}$ is a triangle and that no diagonals intersect. A minimal tetrahedralization of $\mathcal{P}$ is a tetrahedralization that contains the minimum number of tetrahedra among all tetrahedralizations of $\mathcal{P}$. The graph of $\mathcal{P}$, denoted by $G(\mathcal{P})$, is the graph with vertex set $V(\mathcal{P})$ and edge set $E(\mathcal{P})$. Throughout this chapter, we study properties of $G(\mathcal{P})$ and the minimal tetrahedralization of $\mathcal{P}$.

Let $v_0v_1...v_m$ be a path on $G(\mathcal{P})$. If $m > 2$ and there exist two vertices $p, q \in V(\mathcal{P})$ such that all v_i $(0 \leq i \leq m)$ are adjacent to p and q, then the subgraph induced from p, q, v_i $(0 \leq i \leq m)$ is called a *bipyramid net* (for short, *BP-net*), denoted by $B(pv_0v_1...v_mq)$ (see Figure 3.1(left)). Furthermore, if $v_0 = v_m$, that is, if $v_0v_1...v_m$ is a cycle, then $B(pv_0v_1...v_mq)$ is called a *bipyramid* (see Figure 3.1(right)). Vertices p and q are called *poles* of the BP-net or bipyramid. The edge pq is the *polar edge.* Obviously, a BP-net can be obtained from a bipyramid by splitting a path pv_iq into two paths pv_iq and $pv_i'q$, illustrated in Figure 3.2.

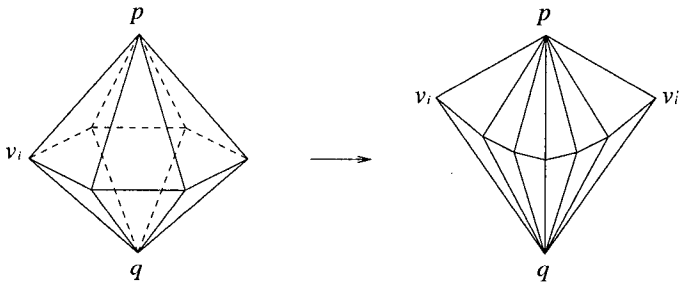

Figure 3.2: The relationship between a bipyramid (left) and a BP-net (right).

3.2.2 Properties of $G(\mathcal{P})$

For each edge ab on the surface of $\mathcal{P}$, there must exist two vertices c and d such that abc and abd are two facets on the surface of $\mathcal{P}$. Let $\text{tri}(ab)$ denote the set of two vertices that determine the two triangles sharing ab. The set of all adjacent vertices of a vertex v in $G(\mathcal{P})$ is denoted by $\text{adj}(v)$. The degree of v is denoted by $\deg(v)$.

Theorem 3.1 *If $|V(\mathcal{P})| > 4$, then any two vertices of degree 3 are not adjacent in $G(\mathcal{P})$.*

Proof Refer to Figure 3.3. Suppose $ab \in E(\mathcal{P})$, and $\deg(a) = \deg(b) = 3$. Let $\text{tri}(ab) = \{c, d\}$. Since $\deg(a) = 3$, $\text{adj}(a) = \{b, c, d\}$; hence $\text{tri}(ac) = \{b, d\}$. Similarly, $\text{tri}(bc) = \{a, d\}$, $\text{tri}(ad) = \{b, c\}$, $\text{tri}(bd) = \{a, c\}$, and $\text{tri}(cd) = \{a, b\}$. Thus, $\mathcal{P} = abcd$ is a tetrahedron and $|V(\mathcal{P})| = 4$. This is a contradiction. Therefore any two vertices of degree 3 are not adjacent if $|V(\mathcal{P})| > 4$. □

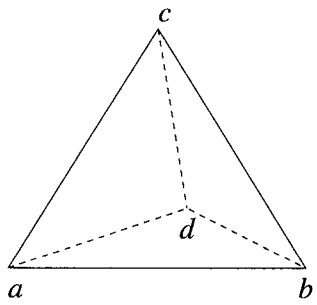

Figure 3.3: The illustration for Theorem 3.1 and Corollary 3.1.

Corollary 3.1 *Let $a, b \in V(\mathcal{P})$, and $\deg(a) = \deg(b) = 3$. $\mathcal{P}$ is a tetrahedron if and only if $ab \in E(\mathcal{P})$ (see Figure 3.3).*

Theorem 3.2 *Let* $\deg(a) = 3$, $a \in V(\mathcal{P})$. *If* $|V(\mathcal{P})| > 5$, *then there is at most one vertex in* $\mathrm{adj}(a)$ *whose degree is less than 5.*

Proof Refer to Figure 3.4(left). Let $\mathrm{adj}(a) = \{a_1, a_2, a_3\}$. It follows from Theorem 3.1 that $\deg(a_i) > 3, i = 1, 2, 3$. Suppose there exists one vertex a_1 whose degree is 4, and $\mathrm{adj}(a_1) = \{a, a_2, a_3, b\}$. Then $a_2b, a_3b \in E(\mathcal{P})$. Now consider the edge a_2a_3. If $\mathrm{tri}(a_2a_3) = \{a, b\}$, then $\mathcal{P} = aa_1a_2a_3b$, and $|V(\mathcal{P})| = 5$, which contradicts the condition of the theorem. Thus, $\mathrm{tri}(a_2a_3) = \{a, c\}$. Therefore, $\deg(a_i) \geq 5, i = 2, 3$. This completes the proof. □

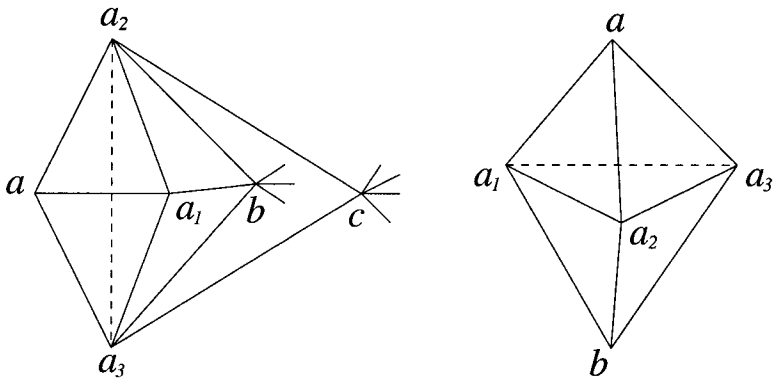

Figure 3.4: The illustration for Theorem 3.2 (left) and Corollary 3.2 (right).

Corollary 3.2 *Let* $a, a_1, a_2 \in V(\mathcal{P})$, $\deg(a) = 3$, *and* $\deg(a_1) = \deg(a_2) = 4$. $\mathcal{P}$ *is a hexahedron if and only if* $a_i \in \mathrm{adj}(a)$, $i = 1, 2$ *(see Figure 3.4).*

For the vertex of degree 4 in $\mathcal{P}$, its adjacent vertices have the following property:

Theorem 3.3 *Let* $\deg(a) = 4, a \in V(\mathcal{P})$. *If* $|V(\mathcal{P})| > 6$, *then there are at most two nonadjacent vertices in* $\mathrm{adj}(a)$ *whose degrees are less than 5.*

Proof Let $\text{adj}(a) = \{a_1, a_2, a_3, a_4\}$ and a_1a_2, a_2a_3, a_3a_4, $a_4a_1 \in E(\mathcal{P})$. Consider the following two cases:

1. $\deg(a_1) = 3$ (Refer to Figure 3.5(left)). Then $a_2a_4 \in E(\mathcal{P})$. Thus, $\deg(a_i) \geq 4, i = 2, 4$. If $\deg(a_2) = 4$ or $\deg(a_4) = 4$, then $\mathcal{P} = aa_1a_2a_3a_4$, and $|V(\mathcal{P})| = 5$, which contradicts the condition of the theorem. Thus, $\deg(a_i) > 4, i = 2, 4$. If $\deg(a_3) = 3$, then $\mathcal{P} = aa_1a_2a_3a_4$, and $|V(\mathcal{P})| = 5$, which contradicts the condition of the theorem. The degree of a_3 can be equal to 4.

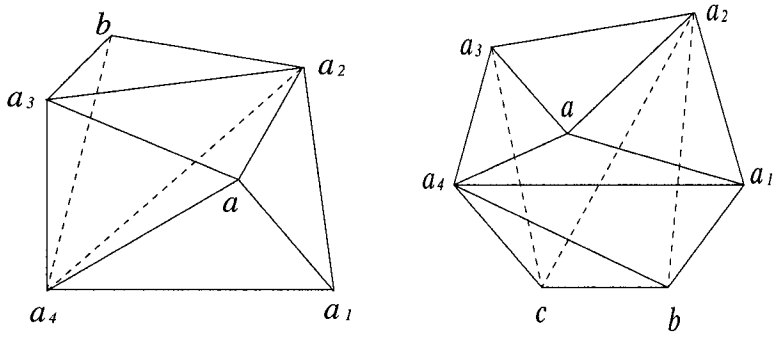

Figure 3.5: Case 1 (left) and Case 2 (right) of the proof of Theorem 3.3. The left figure is also a type I octahedron.

2. $\deg(a_1) = 4$ (Refer to Figure 3.5(right)). We know that $a, a_2, a_4 \in \text{adj}(a_1)$. If $a_3 \in \text{adj}(a_1)$, then $\mathcal{P} = aa_1a_2a_3a_4$, which contradicts the condition $|V(\mathcal{P})| > 6$. Thus $a_3 \notin \text{adj}(a_1)$. Let $\text{adj}(a_1) = \{a, a_2, a_4, b\}$, then $a_2b, a_4b \in E(\mathcal{P})$. If $\deg(a_4) = 4$, then $\text{adj}(a_4) = \{a, a_1, a_3, b\}$, and $\text{tri}(a_4b) = \{a_1, a_3\}$. Thus, $\mathcal{P} = aa_1a_2a_3a_4b$, and $|V(\mathcal{P})| = 6$ which contradicts the condition. Therefore, $\deg(a_4) > 4$. Similarly, we can prove that $\deg(a_2) > 4$. The degree of a_3 can be equal to 4. This ends the proof.

□

From the definition of a BP-net, we know that for a BP-net $B(pv_0v_1...v_mq)$, $\deg(v_i) = 4$ $(1 \leq i \leq m-1)$. Now we consider the inverse case.

Theorem 3.4 *Let $a_1a_2...a_k$ be a path in $G(\mathcal{P}), k > 1$. If $\deg(a_i) = 4$ $(1 \leq i \leq k)$, then the subgraph induced from a_i $(1 \leq i \leq k)$ and their adjacent vertices is a bipyramid or a BP-net.*

Proof Consider the following four cases of k.

1. $k = 2$ (see Figure 3.6 (left)). Let $\text{tri}(a_1a_2) = \{b, c\}$, $\text{adj}(a_1) = \{a_2, b, c, d\}$ and $\text{adj}(a_2) = \{a_1, b, c, e\}$. Then $bd, cd, be, ce \in E(\mathcal{P})$. Thus, the graph induced from $\{a_1, a_2, b, c, d, e\}$ is a BP-net if $d \neq e$ or a bipyramid if $d = e$.

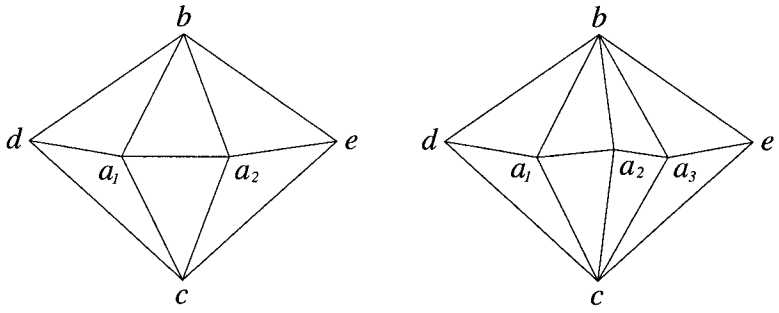

Figure 3.6: Case 1 (left) and Case 2(a) (right) of the proof of Theorem 3.4.

2. k=3. Consider the following two cases of $\text{tri}(a_1a_2)$.

   (a) $\text{tri}(a_1a_2) = \{b, c\}$ (see Figure 3.6 (right)). Then $\text{adj}(a_2) = \{a_1, a_3, b, c\}$. Thus, $a_3b, a_3c \in E(\mathcal{P})$. Let $d \in \text{adj}(a_1), e \in \text{adj}(a_3)$, then $bd, cd, be, ce \in E(\mathcal{P})$. Therefore, the graph induced from $\{a_1, a_2, a_3, b, c, d, e\}$ is a BP-net if $d \neq e$ or a bipyramid if $d = e$.

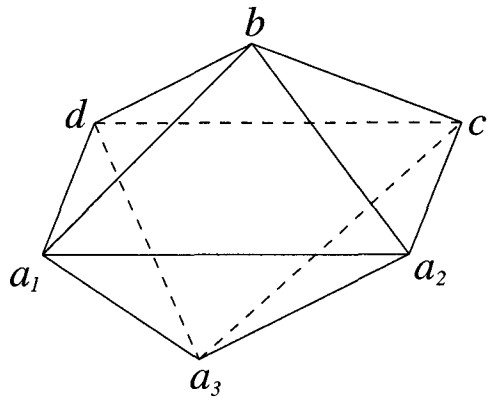

Figure 3.7: Case 2(b) of the proof of Theorem 3.4 (a type II octahedron).

(b) $\text{tri}(a_1a_2) = \{a_3, b\}$. Let $\text{adj}(a_2) = \{a_1, a_3, b, c\}$. Hence, $a_3c, bc \in E(\mathcal{P})$. If $c \in \text{adj}(a_1)$, then $\mathcal{P} = a_1a_2a_3bc$ and $\deg(a_3) = 3$, which is a contradiction. Thus, $\text{adj}(a_1) = \{a_2, a_3, b, d\}$. Therefore $a_3d, bd \in E(\mathcal{P})$. Since $\text{adj}(a_3) = \{a_1, a_2, c, d\}$, $cd \in E(\mathcal{P})$. Thus, the graph induced from $\{a_1, a_2, a_3, b, c, d\}$ is a bipyramid (in particular, an octahedron), illustrated in Figure 3.7.

3. $4 \leq k \leq 6$. Consider the following two cases of $\text{tri}(a_1a_2)$.

(a) $\text{tri}(a_1a_2) = \{b, c\}$ (refer to Figure 3.6 (right)). Then $\text{adj}(a_2) = \{a_1, a_3, b, c\}$. It follows that $\text{adj}(a_i) = \{a_{i-1}, a_{i+1}, b, c\}$, $3 \leq i \leq k-1$. Let $d \in \text{adj}(a_1)$, $e \in \text{adj}(a_k)$. Then $bd, cd, be, ce \in E(\mathcal{P})$. Thus the graph induced from $\{a_1, ..., a_k, b, c, d, e\}$ is a BP-net if $d \neq e$ or a bipyramid if $d = e$.

(b) $\text{tri}(a_1a_2) = \{a_i, b\}$, $3 \leq i \leq k$. Using the same argument as Case 2(b), we can prove that the graph induced from $a_1, ..., a_k$ and their adjacent vertices is an octahedron (bipyramid). For example, we can set $c = a_4, d = a_5, b = a_6$ in Figure 3.7.

4. $k > 6$. In this case, the graph induced from $a_1, ..., a_k$ and their adjacent vertices cannot be an octahedron. Thus, we can prove that this graph is a BP-net or bipyramid using the same argument as Case 2(a) and Case 3(a).

□

Theorems 3.3 and 3.4 imply the following three properties of octahedra.

Corollary 3.3 *There are only two different types for the graphs of octahedra.*

Call the two types of graphs of octahedra in Corollary 3.3 a type I octahedron (illustrated in Figure 3.5(left)) and a type II octahedron (illustrated in Figure 3.7).

Corollary 3.4 *Let* $a, a_1, a_2 \in V(\mathcal{P})$, $a_1a_2 \in E(\mathcal{P})$, $\deg(a_1) = 3$, $\deg(a) = 4$ *and* $\deg(a_2) = 5$. $\mathcal{P}$ *is a type I octahedron if and only if* $a_i \in \text{adj}(a)$, $i = 1, 2$.

Corollary 3.5 *Let* $a, a_1, a_2 \in V(\mathcal{P})$, $a_1a_2 \in E(\mathcal{P})$, *and* $\deg(a) = \deg(a_1) = \deg(a_2) = 4$. $\mathcal{P}$ *is a type II octahedron if and only if* $a_i \in \text{adj}(a)$, $i = 1, 2$.

3.3 Minimal tetrahedralizations of BP-polyhedra

3.3.1 Definitions

For a simple polyhedron, the tetrahedralization is a partition of the polyhedron into a number of tetrahedra that meet only at shared faces. The tetrahedralization consists of all the edges of the polyhedron and some of its diagonals. If $\mathcal{P}$ has a tetrahedralization in which there are no diagonals, then $\mathcal{P}$ is called a *stacked polyhedron* [86, 39]. Obviously, if $\mathcal{P}$ is a stacked polyhedron, then $\mathcal{P}$ can be obtained from tetrahedra by successive additions of tetrahedra on the triangle facets.

Let $B(pv_0v_1...v_mq)$ be a BP-net on the surface of $\mathcal{P}$. $B(pv_0v_1...v_mq)$ is *proper* if it satisfies the following two conditions:

1. pq is a diagonal or $pq \in E(\mathcal{P})$, and

2. each pqv_jv_{j+1} $(0 \leq j \leq m-1)$ is a tetrahedron which is included in the polyhedron $\mathcal{P}$.

If $B(pv_0v_1...v_mq)$ is a proper BP-net, then the union of all the tetrahedra pqv_jv_{j+1} $(0 \leq j \leq m-1)$ is called a *tetrahedralized BP-net* (see Figure 3.8(a)). It is clear that the tetrahedralized BP-net is a stacked polyhedron. For two proper BP-nets on the surface of $\mathcal{P}$, if the intersection of the interiors of their tetrahedralized BP-nets is empty, then the two proper BP-nets are *disjoint* (see Figure 3.8(b)).

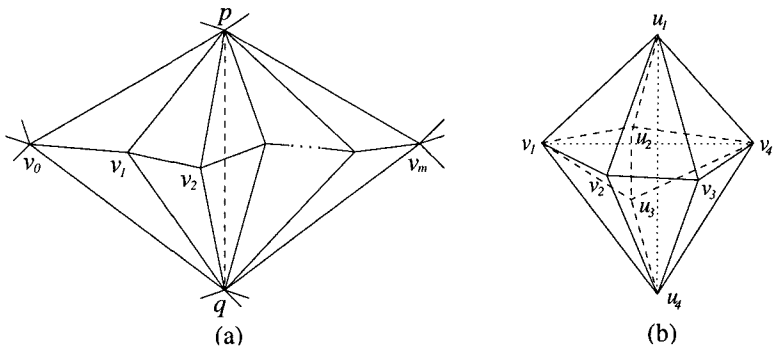

Figure 3.8: (a) The tetrahedralized BP-net with tetrahedra pqv_jv_{j+1} $(0 \leq j \leq m-1)$. It can be considered as a stacked polyhedron. (b) The BP-nets $B(u_1v_1v_2v_3v_4u_4)$ and $B(v_1u_1u_2u_3u_4v_4)$ are disjoint. After trimming off the two BP-nets, we obtain the tetrahedron $u_1v_1u_4v_4$.

Let $B(p^{(i)}v_0^{(i)}v_1^{(i)}...v_{m_i}^{(i)}q^{(i)})$ $(1 \leq i \leq k)$ be all the disjoint proper BP-nets on the surface of $\mathcal{P}$. After removing the vertices $v_j^{(i)}$ $(j = 1, ..., m_i - 1, i = 1, ..., k)$ and their adjacent edges and adding edges $p^{(i)}q^{(i)}$, $i = 1, ..., k$, we obtain a new polyhedron $\mathcal{P}_1$. When there is no confusion, this process of removing vertices and adding edges is called *trimming off* BP-nets. If $\mathcal{P}_1$ is a stacked polyhedron (the empty polyhedron is regarded as a stacked polyhedron), then $\mathcal{P}$ is called a *BP-polyhedron* (see Figure 3.8(b)). Recursively, if $\mathcal{P}_1$ is a BP-polyhedron, then $\mathcal{P}$ is called a *two-level* BP-polyhedron, and $\mathcal{P}_1$ is called the *internal* BP-polyhedron of $\mathcal{P}$. Let $\mathcal{P}'$ be a BP-polyhedron (or two-level BP-polyhedron). If $\mathcal{P}$ is constructed from $\mathcal{P}'$ by successive additions of tetrahedra on the triangle facets, then $\mathcal{P}$ is called an *augmented* BP-polyhedron (or two-level BP-polyhedron).

In the definition of a BP-polyhedron, since BP-nets $B(p^{(i)}v_0^{(i)}v_1^{(i)}...v_{m_i}^{(i)}q^{(i)})$ $(1 \leq i \leq k)$ are disjoint and proper, the order in which the process of trimming off is done

does not affect the resulting polyhedron $\mathcal{P}_1$.

Let $\mathcal{D}$ be the set of diagonals in a tetrahedralization of $\mathcal{P}$. Then we say that $\mathcal{D}$ *creates* a tetrahedralization of $\mathcal{P}$. Let $\mathcal{A}$ be an area on the surface of $\mathcal{P}$ such that the boundary of $\mathcal{A}$ is a simple polygon on the surface of $\mathcal{P}$. Consider $ab \in \mathcal{D}$. If $a, b \in V(\mathcal{A})$, then ab is called a *type-I diagonal* (or *diag-I*, for short) on $\mathcal{A}$. We also say that $\mathcal{A}$ associates with a diag-I of $\mathcal{D}$. If a is an interior vertex of $\mathcal{A}$, and $b \notin V(\mathcal{A})$, then ab is called a *type-II diagonal* (or *diag-II*, for short) on $\mathcal{A}$. We also say that $\mathcal{A}$ associates with a diag-II of $\mathcal{D}$. As illustrated in Figure 3.9, the polyhedron $\mathcal{P}$ consists of the solid edges and the dotted edge. $\mathcal{P}$ is constructed as follows: Start with a bipyramid $B(pv_0v_1v_2v_3v_4v_0q)$, attach a tetrahedron apv_0v_4 to the facet pv_0v_4 of the bipyramid so that edge aq is contained in the interior of $\mathcal{P}$. Let $\mathcal{D} = \{v_0v_2, v_2v_4, av_2, aq\}$ be the set of diagonals which creates the tetrahedralization with tetrahedra $\{pv_0v_1v_2, qv_0v_1v_2, pv_2v_3v_4, qv_2v_3v_4, apv_0v_2, aqv_0v_2, apv_2v_4, aqv_2v_4, aqv_0v_4\}$. Consider the BP-net $B(pv_0v_1v_2v_3v_4q)$ as an area on the surface of $\mathcal{P}$. This BP-net associates with two diag-Is of $\mathcal{D}$ (i.e., v_0v_2, v_2v_4) and one diag-II of $\mathcal{D}$ (i.e., av_2).

Let tetra(abc) denote the set of vertices that form the tetrahedra with abc in a tetrahedralization of $\mathcal{P}$. Obviously, $|\text{tetra}(abc)| = 1$ or 2. Each vertex in tetra(abc) is called the *tetra-vertex* of abc.

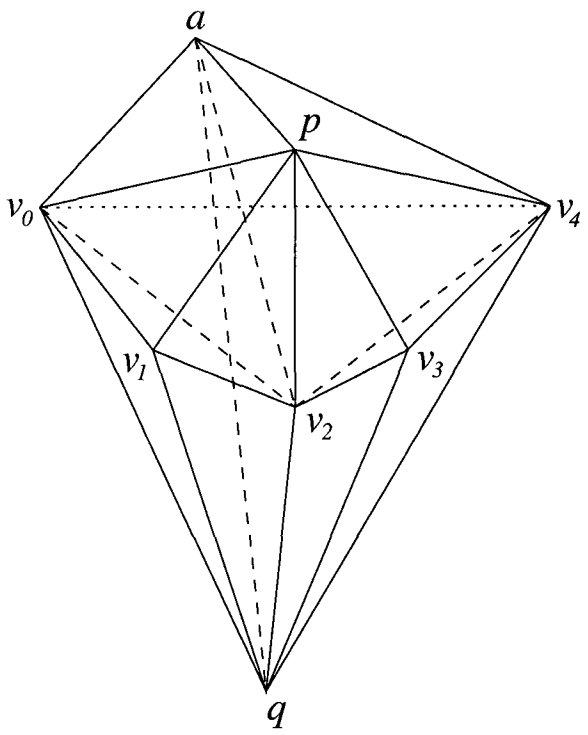

Figure 3.9: The solid edges and the dotted edge form the edge set of a polyhedron $\mathcal{P}$. The dashed edges are diagonals of $\mathcal{P}$ which create a tetrahedralization of $\mathcal{P}$.

3.3.2 BP-polyhedra

In this section, we prove the structure of the minimal tetrahedralizations of BP-polyhedra. The main result is the following:

Theorem 3.5 *Let $\mathcal{P}$ be a BP-polyhedron, and $p^{(i)}q^{(i)}$ $(1 \leq i \leq k)$ be all the polar edges of the disjoint proper BP-nets on the surface of $\mathcal{P}$. The set $\{p^{(i)}q^{(i)} \mid 1 \leq i \leq k\}$ creates a unique minimal tetrahedralization of $\mathcal{P}$.*

Proof Let $B(p^{(i)}v_0^{(i)}v_1^{(i)}...v_{m_i}^{(i)}q^{(i)})$ $(1 \leq i \leq k)$ be all the disjoint proper BP-nets on the surface of $\mathcal{P}$. After removing the vertices $v_j^{(i)}$ $(j = 1, ..., m_i - 1, i = 1, ..., k)$ and

their adjacent edges from $\mathcal{P}$, and adding edges $p^{(i)}q^{(i)}$ $(1 \leq i \leq k)$, we obtain a new polyhedron Q. From the definition of a BP-polyhedron we know that Q is a stacked polyhedron. Since each $B(p^{(i)}v_0^{(i)}v_1^{(i)}...v_{m_i}^{(i)}q^{(i)})$ and its polar edge $p^{(i)}q^{(i)}$ also construct a stacked polyhedron, the edge set $\{p^{(i)}q^{(i)} \mid 1 \leq i \leq k\}$ creates a tetrahedralization of $\mathcal{P}$. $\{p^{(i)}q^{(i)} \mid 1 \leq i \leq k\}$ may contain edges that belong to $E(\mathcal{P})$ or edges that are coincident. Without loss of generality, we can assume that all the edges $p^{(i)}q^{(i)}$ $(1 \leq i \leq k)$ are different diagonals. Now we show that $\{p^{(i)}q^{(i)} \mid 1 \leq i \leq k\}$ creates a unique minimal tetrahedralization by contradiction.

Suppose that the diagonal set T creates a minimal tetrahedralization of $\mathcal{P}$ such that $T \neq \{p^{(i)}q^{(i)} \mid 1 \leq i \leq k\}$ and $|T| \leq k$. Let $T_0 = T \cap \{p^{(i)}q^{(i)} \mid 1 \leq i \leq k\}$ and $T_1 = T - T_0$. Since the BP-net is not a tetrahedralization, there must exist some edges of T whose endpoints belong to the BP-net. Thus, we can classify the BP-nets into two sets β_0 and β_1, which are defined as follows: for each BP-net in β_0, its polar edge belongs to T_0, and for each BP-net in β_1, its polar edge does not belong to T. First, we prove the following lemma:

Lemma 3.1 *Each BP-net $B(pv_0v_1...v_mq)$ in β_1 must associate with at least:*

(1) two diag-Is of T, or

(2) one diag-I and one diag-II of T, or

(3) three diag-IIs of T, or

(4) two diag-IIs of T with a common endpoint.

Proof Each triangle facet on the surface of $\mathcal{P}$ has only one tetra-vertex and each triangle facet in the interior of $\mathcal{P}$ has exactly two tetra-vertices. From the definition of a BP-net, we know that $m \geq 3$. Consider the triangle facet v_0pv_1. Its tetra-vertex cannot be q since $B(pv_0v_1...v_mq)$ is in β_1; that is, $pq \notin T$. We have the following three cases:

1. $v_2 \in \text{tetra}(v_0pv_1)$ (see Figure 3.10(left)). Then $v_0v_2 \in T$, which is a diag-I. Consider the interior facet v_0v_2p. v_1 is one tetra-vertex. The other tetra-vertex may be $v_i (3 \leq i \leq m)$, or the vertex a which does not belong to $B(pv_0v_1...v_mq)$. Thus, $v_0v_i \in T$ or $v_2a \in T$. In total, the BP-net associates with at least two diag-Is, or one diag-I and one diag-II.

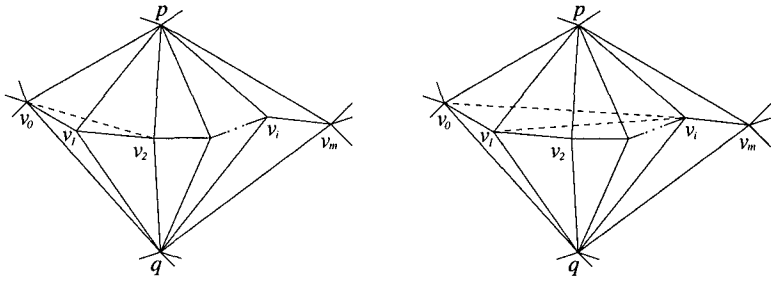

Figure 3.10: Case 1 (left) and Case 2 (right) of the proof of Lemma 3.1.

2. $v_i \in \text{tetra}(v_0pv_1), 3 \leq i \leq m$ (see Figure 3.10(right)). Then $v_0v_i, v_1v_i \in T$, which are diag-Is. Thus, the BP-net associates with at least two diag-Is.

3. $b \in \text{tetra}(v_0pv_1)$ and $b \notin V(B(pv_0...v_mq))$. Then $v_1b \in T$, which is a diag-II. There are three cases for the tetra-vertex of v_1pv_2.

(a) $v_i \in \text{tetra}(v_1pv_2), 3 \leq i \leq m$. Then $v_1v_i \in T$, which is a diag-I (see Figure 3.11(left)). Thus, the BP-net associates with at least one diag-I and one diag-II.

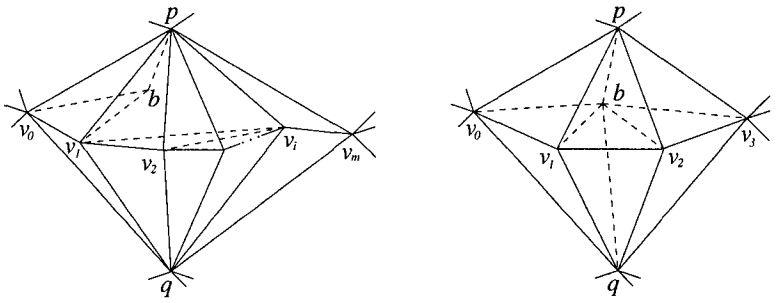

Figure 3.11: Case 3(a) (left) and Case 3(c)ii (right) of the proof of Lemma 3.1.

(b) $c \in \text{tetra}(v_1pv_2)$ and $c \notin V(B(pv_0...v_mq))$. Then $v_1c, v_2c \in T$, which are diag-IIs. Thus, the BP-net associates with at least three diag-IIs.

(c) $b \in \text{tetra}(v_1pv_2)$. Then $v_2b \in T$. Consider v_2pv_3. If $m > 3$, the tetra-vertex of v_2pv_3 may be $v_i (4 \leq i \leq m)$, or the vertex b, or the vertex $b' \notin V(B(pv_0...v_mq))$. Thus, $v_2v_i \in T$, or $v_3b \in T$, or $v_2b', v_3b' \in T$. Therefore, $B(pv_0...v_mq)$ associates with at least one diag-I and two diag-IIs, or at least three diag-IIs. If m=3, we have the following two cases:

i. $d \in \text{tetra}(v_2pv_3)$ and $d \notin V(B(pv_0...v_mq))$. Then $v_2d \in T$, which is a diag-II. Thus, $B(pv_0...v_mq)$ associates with at least three diag-IIs.

ii. $b \in \text{tetra}(v_2pv_3)$. Then $v_3b \in T$. We can discuss the symmetric triangle facets v_0qv_1, v_1qv_2, v_2qv_3 using the above argument. Either we prove the lemma, or we get the same case as above; that is, $qb \in T$ (see Figure 3.11(right)). Thus, $B(pv_0...v_mq)$ associates with two diag-IIs

$\{v_1b, v_2b\}$ with a common endpoint b. This completes the proof of Lemma 3.1.

□

Note that each diag-I associates with only one BP-net and each diag-II associates with at most two BP-nets. Also note that each pair of diag-IIs with a common endpoint cannot be used to tetrahedralize two BP-nets. Since $T \neq \{p^{(i)}q^{(i)} \mid 1 \leq i \leq k\}$, we know that $T_1 \neq \emptyset$. It follows from Lemma 3.1 that the total number of edges in T_1 that associate with the BP-nets in β_1 is greater than $|\beta_1|$. Since we assumed that all the edges $p^{(i)}q^{(i)}, 1 \leq i \leq k$ are different, the total number of edges in T_0 is $|\beta_0|$. Thus

$$|T| = |T_1| + |T_0| > |\beta_1| + |\beta_0| = k.$$

This contradicts the assumption that $|T| \leq k$. Therefore, $\{p^{(i)}q^{(i)} \mid 1 \leq i \leq k\}$ creates a unique minimal tetrahedralization of $\mathcal{P}$. This ends the proof of Theorem 3.5. □

From an extension of Lemma 3.1, we can derive the following corollary on the minimal tetrahedralizations of augmented BP-polyhedra.

Corollary 3.6 *Let $\mathcal{P}$ be a simple polyhedron. After removing all the degree 3 vertices in $G(\mathcal{P})$ recursively, we obtain a new polyhedron $\mathcal{P}'$. If $\mathcal{P}'$ is a BP-polyhedron, then $\{p^{(i)}q^{(i)} \mid 1 \leq i \leq k\}$ creates a unique minimal tetrahedralization of $\mathcal{P}$, where $p^{(i)}q^{(i)}$ $(1 \leq i \leq k)$ are the polar edges of all the disjoint proper BP-nets on the surface of*

$\mathcal{P}'$.

Proof Let B'_i $(1 \leq i \leq k)$ be all the disjoint proper BP-nets on the surface of $\mathcal{P}'$. Since $G(\mathcal{P}')$ is obtained by trimming off some stacked polyhedra from $G(\mathcal{P})$, for each B'_i there must exist B_i on the surface of $\mathcal{P}$ such that B'_i is obtained by trimming off stacked polyhedra from B_i (here, an empty polyhedron is also considered as a stacked polyhedron). It is obvious that Lemma 3.1 still holds for such B_i. Therefore, we can prove the corollary with an argument similar to that used in the proof of Theorem 3.5.

□

3.3.3 Two-level BP-polyhedra

Now we consider the minimal tetrahedralization of two-level BP-polyhedra. Let $\mathcal{P}$ be a two-level BP-polyhedron, $\mathcal{P}_1$ be the internal BP-polyhedron of $\mathcal{P}$, and BP-net($\mathcal{P}$) (resp. BP-net($\mathcal{P}_1$)) denote the set of all BP-nets on the surface of $\mathcal{P}$ (resp. $\mathcal{P}_1$). From the definition of a two-level BP-polyhedron, we know that the BP-nets in BP-net($\mathcal{P}$) and BP-net($\mathcal{P}_1$) are proper and disjoint, respectively. As $\mathcal{P}_1$ is obtained by trimming off BP-net($\mathcal{P}$) from $\mathcal{P}$, each BP-net in BP-net($\mathcal{P}_1$) has at least one edge that is the polar edge of a BP-net in BP-net($\mathcal{P}$) and does not belong to $E(\mathcal{P})$. In this section, we only prove the structure of the minimal tetrahedralization for a special two-level BP-polyhedron, that is, each BP-net in BP-net($\mathcal{P}_1$) contains only one edge that is the polar edge of a BP-net in BP-net($\mathcal{P}$). The following five lemmas give properties

of the relationship between BP-nets and the minimal tetrahedralization.

Lemma 3.2 *Let $B = B(xz_0z_1...z_my)$ be a proper BP-net on the surface of $\mathcal{P}$. By deleting the interior edges of B and adding edge xy, we obtain a new proper BP-net, denoted by $\tilde{B}_1 = B(pv_0v_1v_2v_3q)$, where $v_1 = x, v_2 = y$, $p = z_0, q = z_m$ (see Figure 3.12). Let T create a tetrahedralization of $\mathcal{P}$, and $\overline{B}_1$ be the area on the surface of $\mathcal{P}$ which contains $v_0, v_1, v_2, v_3, z_i (0 \leq i \leq m)$ and is bounded by pv_0, pv_3, qv_0, qv_3. If $xy, pq \notin T$, then T must have at least:*

(1) three diag-Is on $\overline{B}_1$, or

(2) two diag-Is and one diag-II on $\overline{B}_1$, or

(3) one diag-I and three diag-IIs on $\overline{B}_1$, or

(4) five diag-IIs on $\overline{B}_1$, or

(5) four diag-IIs on $\overline{B}_1$ with a common endpoint.

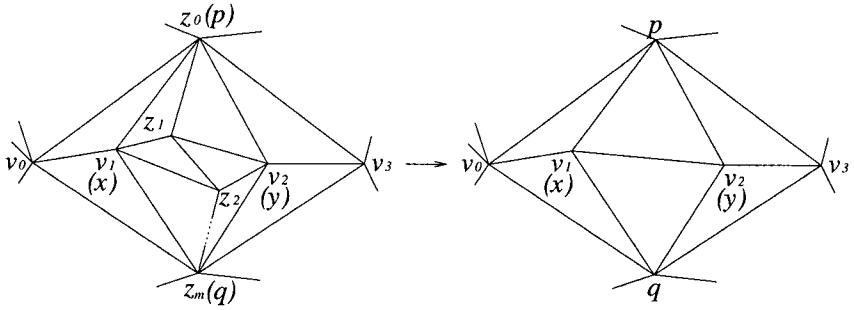

Figure 3.12: After trimming off BP-net $B = B(xz_0z_1...z_my)$ from $\overline{B}_1$ (left), we obtain a new BP-net $\tilde{B}_1 = B(pv_0v_1v_2v_3q)$ (right).

Proof We only prove the lemma for $m = 3$. It can be similarly proved for $m > 3$.

Consider the facets $z_0v_1z_1$. $v_2, z_3 \notin \text{tetra}(z_0v_1z_1)$ since $v_1v_2, z_0z_3 \notin T$. Thus, $\text{tetra}(z_0v_1z_1)$ may contain z_2, or v_0, or v_3, or the vertex c which does not belong to $\overline{B}_1$. Therefore, we have the following cases:

1. $z_2 \in \text{tetra}(z_0v_1z_1)$. Then $z_0z_2 \in T$, which is a diag-I (see Figure 3.13(left)). Consider the interior facet $z_0v_1z_2$. $z_1 \in \text{tetra}(z_0v_1z_2)$, and the other vertex in $\text{tetra}(z_0v_1z_2)$ may be v_0, or v_3, or the vertex a which does not belong to $V(\overline{B}_1)$.

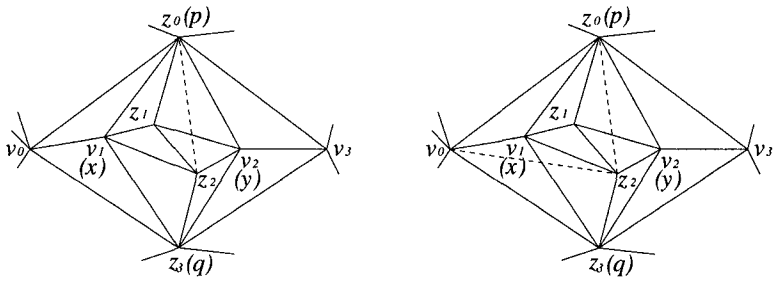

Figure 3.13: Case 1 (left) and Case 1(a) (right) of the proof of Lemma 3.2.

   (a) $v_0 \in \text{tetra}(z_0v_1z_2)$. Then $z_2v_0 \in T$, which is a diag-I (see Figure 3.13(right)). Consider $z_0z_2v_0$.

      i. If v_2 or v_3 belongs to $\text{tetra}(z_0z_2v_0)$, then v_0v_2 or $v_0v_3 \in T$, which is a diag-I. In total, there are at least three diag-Is on $\overline{B}_1$.

      ii. If $a' \in \text{tetra}(z_0z_2v_0)$ which does not belong to $V(\overline{B}_1)$, then $z_2a' \in T$, which is a diag-II. In total, there are at least two diag-Is and one diag-II on $\overline{B}_1$.

   (b) $v_3 \in \text{tetra}(z_0v_1z_2)$. Then $v_1v_3, z_2v_3 \in T$, which are diag-Is (see Figure 3.14(left)). Thus, there are at least three diag-Is on $\overline{B}_1$.

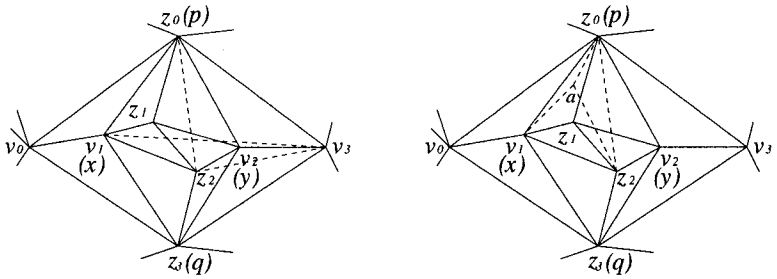

Figure 3.14: Case 1(b) (left) and Case 1(c) (right) of the proof of Lemma 3.2.

(c) $a \in \text{tetra}(z_0 v_1 z_2)$. Then $v_1 a, z_2 a \in T$, which are diag-IIs (see Figure 3.14 (right)). Considering $z_0 z_2 v_2$, at least one diag-I or diag-II is incident on z_0, or z_2, or v_2. In total, there are at least two diag-Is and two diag-IIs, or one diag-I and three diag-IIs on $\overline{B}_1$.

2. $v_0 \in \text{tetra}(z_0 v_1 z_1)$. Then $z_1 v_0 \in T$, which is a diag-I (see Figure 3.15(left)). Consider the interior facet $z_1 v_0 v_1$. $z_0 \in \text{tetra}(z_1 v_0 v_1)$, and the other vertex in $\text{tetra}(z_1 v_0 v_1)$ may be z_2, or z_3, or v_3, or the vertex b which does not belong to $V(\overline{B}_1)$.

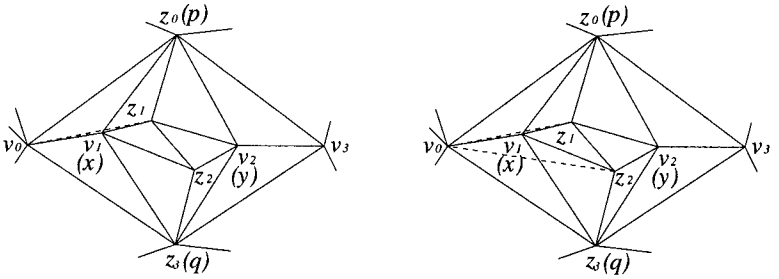

Figure 3.15: Case 2 (left) and Case 2(a) (right) of the proof of Lemma 3.2.

(a) $z_2 \in \text{tetra}(z_1 v_0 v_1)$. Then $z_2 v_0 \in T$, which is a diag-I (see Figure 3.15(right)). Consider $z_1 z_2 v_0$.

i. If z_0, or z_3, or v_2, or v_3 belongs to tetra($z_1z_2v_0$), then at least one diag-I is incident on z_1, or z_2, or v_0. In total, there are at least three diag-Is on $\overline{B}_1$.

ii. If $b' \in$ tetra($z_1z_2v_0$) which does not belong to $V(\overline{B}_1)$, then $z_1b', z_2b' \in T$, which are diag-IIs. In total, there are at least two diag-Is and two diag-IIs on $\overline{B}_1$.

(b) $z_3 \in$ tetra($z_1v_0v_1$). Then $z_1z_3 \in T$, which is a diag-I. Considering $z_1z_3v_0$, at least one diag-I or one diag-II is incident on z_1, or z_3, or v_0. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_1$.

(c) $v_3 \in$ tetra($z_1v_0v_1$). Then $z_1v_3, v_0v_3, v_1v_3 \in T$, which are diag-Is. In total, there are at least four diag-Is on $\overline{B}_1$.

(d) $b \in$ tetra($z_1v_0v_1$). Then $z_1b, v_1b \in T$, which are diag-IIs. Considering $z_1z_2v_2$, at least one diag-I or diag-II is incident on z_1, or z_2, or v_2. In total, there are at least two diag-Is and two diag-IIs, or one diag-I and three diag-IIs on $\overline{B}_1$.

3. $v_3 \in$ tetra($z_0v_1z_1$). Then $z_1v_3, v_1v_3 \in T$, which are diag-Is (see Figure 3.16(left)). Considering the interior facet $z_1v_1v_3$, at least one diag-I or diag-II is incident on z_1, or v_1, or v_3. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_1$.

4. $c \in$ tetra($z_0v_1z_1$) and $c \notin V(\overline{B}_1)$. Then $z_1c, v_1c \in T$, which are diag-IIs (see

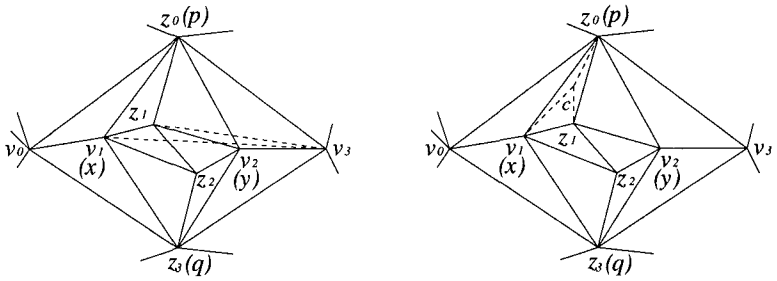

Figure 3.16: Case 3 (left) and Case 4 (right) of the proof of Lemma 3.2.

Figure 3.16(right)). Consider $z_1v_1z_2$. If $z_0 \in \text{tetra}(z_1v_1z_2)$, then the intersection of the interiors of tetrahedra $z_0z_1v_1z_2$ and $z_0v_1z_1c$ is not empty. Thus $z_0 \notin \text{tetra}(z_1v_1z_2)$.

(a) $z_3 \in \text{tetra}(z_1v_1z_2)$. Then $z_1z_3 \in T$, which is a diag-I. Considering $z_1z_2v_2$, at least one diag-I or diag-II is incident on z_1, or z_2, or v_2. In total, there are at least two diag-Is and two diag-IIs, or one diag-I and three diag-IIs on $\overline{B}_1$.

(b) $v_3 \in \text{tetra}(z_1v_1z_2)$. Then $v_1v_3, z_1v_3, z_2v_3 \in T$, which are diag-Is. In total, there are at least three diag-Is and two diag-IIs on $\overline{B}_1$.

(c) $v_0 \in \text{tetra}(z_1v_1z_2)$. Then $z_1v_0, z_2v_0 \in T$, which are diag-Is. In total, there are at least two diag-Is and two diag-IIs on $\overline{B}_1$.

(d) $c' \in \text{tetra}(z_1v_1z_2)$. Then $v_1c', z_1c', z_2c' \in T$, which are diag-IIs. In total, there are at least five diag-IIs on $\overline{B}_1$.

(e) $c \in \text{tetra}(z_1v_1z_2)$. Then $z_2c \in T$, which is a diag-II. Consider $z_1z_2v_2$.

   i. z_0, z_3, v_0 or v_3 belongs to $\text{tetra}(z_1z_2v_2)$. At least one diag-I is incident

on z_1, or z_2, or v_2. In total, there are at least one diag-I and three diag-IIs on $\overline{B}_1$.

ii. $c'' \in \text{tetra}(z_1 z_2 v_2)$ and $c'' \notin V(\overline{B}_1)$. Then $z_1c'', z_2c'', v_2c'' \in T$. In total, there are at least six diag-IIs on $\overline{B}_1$.

iii. $c \in \text{tetra}(z_1 v_2 z_2)$. Then $v_2c \in T$. Consider the other triangle facets in $\overline{B}_1$ (except $z_0v_1z_1, z_1v_1z_2, z_1v_2z_2$). If there is at least one triangle facet whose tetra-vertex is $d \neq c$, then $\overline{B}_1$ associates with at least one diag-I and four diag-IIs, or five diag-IIs; otherwise, if all the vertices in $\overline{B}_1$ are adjacent to c, then there are four diag-IIs on $\overline{B}_1$ with a common endpoint c. This completes the proof of Lemma 3.2.

□

Using a similar method to that in the above proof, we can prove the following three lemmas. For their proofs, see Appendix B, C and D, respectively.

Lemma 3.3 *Let $B = B(xz_0z_1...z_my)$ be a proper BP-net on the surface of $\mathcal{P}$. By deleting the interior edges of B and adding edge xy, we obtain a new proper BP-net, denoted by $\tilde{B}_2 = B(pv_0v_1v_2v_3q)$, where $v_2 = x, v_3 = y$, $p = z_0, q = z_m$ (see Figure 3.17). Let T create a tetrahedralization of $\mathcal{P}$, and $\overline{B}_2$ be the area on the surface of $\mathcal{P}$ which contains $v_0, v_1, v_2, v_3, z_i (0 \leq i \leq m)$ and is bounded by pv_0, pv_3, qv_0, qv_3. If $xy, pq \notin T$, then T must have at least:*

(1) three diag-Is on $\overline{B}_2$, or

(2) two diag-Is and one diag-II on $\overline{B}_2$*, or*

(3) one diag-I and three diag-IIs on $\overline{B}_2$*, or*

(4) five diag-IIs on $\overline{B}_2$*, or*

(5) four diag-IIs on $\overline{B}_2$ *with a common endpoint.*

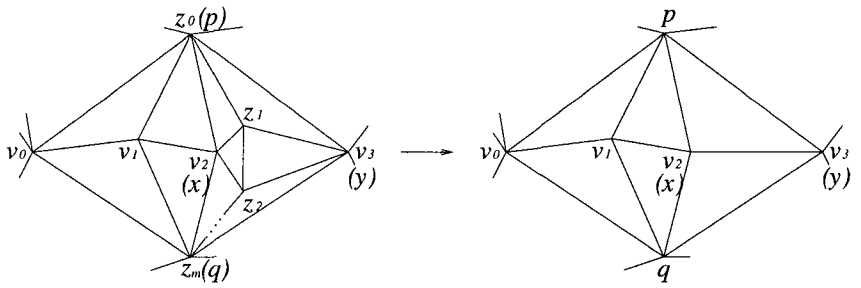

Figure 3.17: After trimming off BP-net $B = B(xz_0z_1...z_my)$ from $\overline{B}_2$ (left), we obtain a new BP-net $\tilde{B}_2 = B(pv_0v_1v_2v_3q)$ (right).

Lemma 3.4 *Let* $B = B(xz_0z_1...z_my)$ *be a proper BP-net on the surface of* $\mathcal{P}$*. By deleting the interior edges of* B *and adding edge* xy*, we obtain a new proper BP-net, denoted by* $\tilde{B}_3 = B(pv_0v_1v_2v_3q)$*, where* $p = x, v_2 = y$*,* $v_1 = z_0, v_3 = z_m$ *(see Figure 3.18). Let* T *create a tetrahedralization of* $\mathcal{P}$*, and* $\overline{B}_3$ *be the area on the surface of* $\mathcal{P}$ *which contains* $v_0, v_1, v_2, v_3, z_i (0 \leq i \leq m)$ *and is bounded by* pv_0, pv_3, qv_0, qv_3*. If* $xy, pq \notin T$*, then* T *must have at least:*

(1) three diag-Is on $\overline{B}_3$*, or*

(2) two diag-Is and one diag-II on $\overline{B}_3$*, or*

(3) one diag-I and three diag-IIs on $\overline{B}_3$*, or*

(4) five diag-IIs on $\overline{B}_3$*, or*

(5) four diag-IIs on $\overline{B}_3$ *with a common endpoint.*

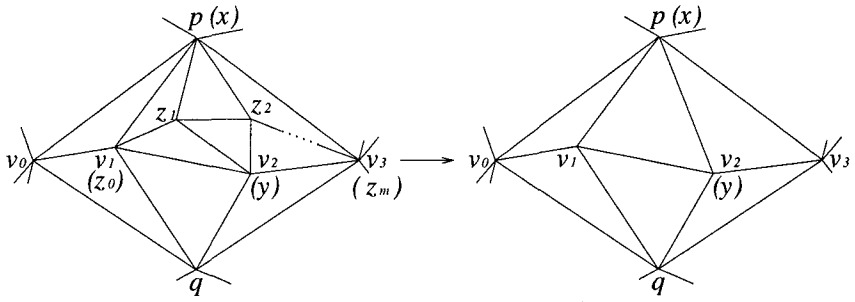

Figure 3.18: After trimming off BP-net $B = B(xz_0z_1...z_my)$ from $\overline{B}_3$ (left), we obtain a new BP-net $\tilde{B}_3 = B(pv_0v_1v_2v_3q)$ (right).

Lemma 3.5 *Let $B = B(xz_0z_1...z_my)$ be a proper BP-net on the surface of $\mathcal{P}$. By deleting the interior edges of B and adding edge xy, we obtain a new proper BP-net, denoted by $\tilde{B}_4 = B(pv_0v_1v_2v_3q)$, where $p = x, v_3 = y$, $v_2 = z_0$ (see Figure 3.19). Let T create a tetrahedralization of $\mathcal{P}$, and $\overline{B}_4$ be the area on the surface of $\mathcal{P}$ which contains $v_0, v_1, v_2, v_3, z_i (0 \leq i \leq m)$ and is bounded by $pv_0, pz_m, qv_0, qv_3, z_mv_3$. If $xy, pq \notin T$, then T must have at least:*

(1) three diag-Is on $\overline{B}_4$, or

(2) two diag-Is and one diag-II on $\overline{B}_4$, or

(3) one diag-I and three diag-IIs on $\overline{B}_4$, or

(4) five diag-IIs on $\overline{B}_4$, or

(5) four diag-IIs on $\overline{B}_4$ with a common endpoint.

From Lemmas 3.2-3.5, we can derive the following property:

Lemma 3.6 *Let $B = B(xz_0z_1...z_my)$ be a proper BP-net on the surface of $\mathcal{P}$. By deleting the interior edges of B and adding edge xy, we obtain a new proper BP-net,*

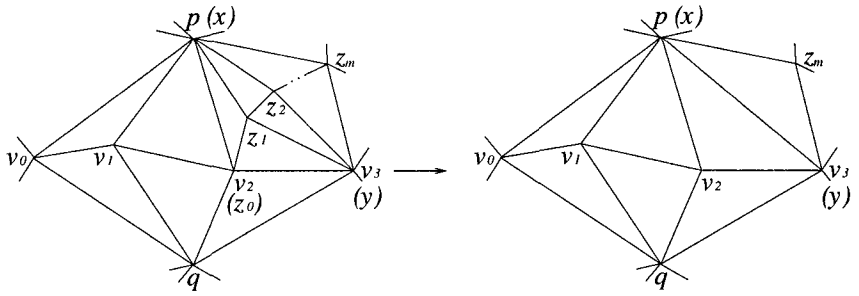

Figure 3.19: After trimming off BP-net $B = B(xz_0z_1...z_my)$ from $\overline{B}_4$ (left), we obtain a new BP-net $\tilde{B}_4 = B(pv_0v_1v_2v_3q)$ (right).

denoted by $\tilde{B} = B(pv_0v_1...v_lq)$, *where* xy *is an edge of* $\tilde{B}$. *Let* T *create a tetrahedralization of* $\mathcal{P}$, *and* $\overline{B}$ *be the graph induced from the vertices of* B *and* $\tilde{B}$. *If* $xy, pq \notin T$, *then* T *must have at least:*

(1) three diag-Is on $\overline{B}$, *or*

(2) two diag-Is and one diag-II on $\overline{B}$, *or*

(3) one diag-I and three diag-IIs on $\overline{B}$, *or*

(4) five diag-IIs on $\overline{B}$, *or*

(5) four diag-IIs on $\overline{B}$ *with a common endpoint.*

Proof From the structure of $\tilde{B}_1, \tilde{B}_2, \tilde{B}_3$, and $\tilde{B}_4$ in Lemmas 3.2-3.5, we know that $\tilde{B}$ must contain at least one of $\tilde{B}_i, i = 1, 2, 3, 4$ in its structure. Thus, Lemma 3.6 is implied by Lemmas 3.2-3.5. □

Now we prove the structure of the minimal tetrahedralization for the special two-level BP-polyhedron mentioned prior to Lemma 3.2.

Theorem 3.6 *Let* $\mathcal{P}$ *be a two-level BP-polyhedron, and* $\mathcal{P}_1$ *be the internal BP-polyhedron*

of $\mathcal{P}$. Let $PE(\mathcal{P})$ be the set of all the polar edges of the disjoint proper BP-nets on the surface of $\mathcal{P}$, and $PE(\mathcal{P}_1)$ be the set of all the polar edges of the disjoint proper BP-nets on the surface of $\mathcal{P}_1$. If each BP-net of $G(\mathcal{P}_1)$ contains only one edge of $PE(\mathcal{P})$, then $PE(\mathcal{P}) \cup PE(\mathcal{P}_1)$ creates a unique minimal tetrahedralization of $\mathcal{P}$.

Proof Let $\tilde{B}_j = B(p^{(j)}v_0^{(j)}v_1^{(j)}...v_{l_j}^{(j)}q^{(j)})$ $(1 \leq j \leq k_1)$ be the disjoint proper BP-nets on the surface of $\mathcal{P}_1$. Notice that each $\tilde{B}_j$ contains a polar edge of a BP-net on the surface of $\mathcal{P}$. Let $B_i = B(x^{(i)}z_0^{(i)}z_1^{(i)}...z_{m_i}^{(i)}y^{(i)})$ $(1 \leq i \leq k_2)$ be the disjoint proper BP-nets on the surface of $\mathcal{P}$ such that each $\tilde{B}_j$ $(1 \leq j \leq k_1)$ contains the polar edge of B_j with the same subscripts. Note that after deleting the interior edges of B_i $(1 \leq i \leq k_2)$ and adding polar edges $x^{(i)}y^{(i)}$, we obtain the internal BP-polyhedron $\mathcal{P}_1$. From the above notation, we know that $PE(\mathcal{P}) = \{x^{(i)}y^{(i)} \mid 1 \leq i \leq k_2\}$ and $PE(\mathcal{P}_1) = \{p^{(j)}q^{(j)} \mid 1 \leq j \leq k_1\}$. It is obvious that $PE(\mathcal{P}) \cup PE(\mathcal{P}_1)$ is a tetrahedralization of $\mathcal{P}$. Without loss of generality, we can assume that all the edges in $PE(\mathcal{P})$ and $PE(\mathcal{P}_1)$ are different diagonals. Now we prove, by contradiction, that $PE(\mathcal{P}) \cup PE(\mathcal{P}_1)$ creates a unique minimal tetrahedralization.

Suppose that the diagonal set T creates a minimal tetrahedralization of $\mathcal{P}$ such that $T \neq PE(\mathcal{P}) \cup PE(\mathcal{P}_1)$ and $|T| \leq k_1 + k_2$. Let $\overline{B}_j$ $(1 \leq j \leq k_1)$ be the graph induced from the vertices of B_j and $\tilde{B}_j$. Define

$$\beta_0 = \{B_i \mid k_1 < i \leq k_2, x^{(i)}y^{(i)} \in T\},$$

$$\beta_1 = \{B_i \mid k_1 < i \leq k_2, x^{(i)}y^{(i)} \notin T\},$$

$$\beta_{00} = \{\overline{B}_j \mid 1 \leq j \leq k_1, x^{(j)}y^{(j)}, p^{(j)}q^{(j)} \in T\},$$

$$\beta_{11} = \{\overline{B}_j \mid 1 \leq j \leq k_1, x^{(j)}y^{(j)}, p^{(j)}q^{(j)} \notin T\},$$

$$\beta_{01} = \{\overline{B}_j \mid 1 \leq j \leq k_1, \text{ only one of } x^{(j)}y^{(j)} \text{ and } p^{(j)}q^{(j)} \text{ belongs to } T\}.$$

We now discuss the three sets β_1, β_{01} and β_{11} respectively. Note that Lemmas 3.1 and 3.6 hold for any diagonal set that creates a tetrahedralization of $\mathcal{P}$.

1. For each $B_i \in \beta_1$, since $x^{(i)}y^{(i)} \notin T$, from Lemma 3.1 we know that the BP-net must associate with at least two diag-Is of T, or one diag-I and one diag-II of T, or three diag-IIs of T, or two diag-IIs of T with a common endpoint.

2. For each $\overline{B}_j \in \beta_{01}$, if $x^{(j)}y^{(j)} \notin T$, then B_j (contained in $\overline{B}_j$) must associate with the same number of diagonals as Case 1. If $x^{(j)}y^{(j)} \in T$ and $p^{(j)}q^{(j)} \notin T$, then $B_j \cup \{x^{(j)}y^{(j)}\}$ is a tetrahedralized BP-net, and can be considered as a stacked polyhedron attached on the surface of $\tilde{B}_j$. It is obvious that Lemma 3.1 still holds for $\tilde{B}_j \cup B_j \cup \{x^{(j)}y^{(j)}\}$. Thus, in total, $\overline{B}_j$ must associate with at least three diag-Is of T, or two diag-Is and one diag-II of T, or one diag-I and three diag-IIs of T, or one diag-I and two diag-IIs of T with a common endpoint.

3. For each $\overline{B}_j \in \beta_{11}$, it follows from Lemma 3.6 that $\overline{B}_j$ must associate with at least three diag-Is of T, or two diag-Is and one diag-II of T, or one diag-I and

three diag-IIs of T, or five diag-IIs of T, or four diag-IIs of T with a common endpoint.

Since each BP-net is not a tetrahedralization, there must exist some edges of T whose endpoints belong to the BP-net. Note that each diag-I associates with only one BP-net; each diag-II associates with at most two BP-nets. Also note that each pair of diag-IIs with a common endpoint cannot be used to tetrahedralize two BP-nets; and each quadruple of diag-IIs with a common endpoint cannot be used to tetrahedralize two elements in β_{11}. It follows from the above cases that

$$|T| > |\beta_0| + |\beta_1| + 2(|\beta_{00}| + |\beta_{01}| + |\beta_{11}|) = k_2 - k_1 + 2k_1 = k_1 + k_2.$$

This contradicts the assumption that $|T| \leq k_1 + k_2$. Therefore, $PE(\mathcal{P}) \cup PE(\mathcal{P}_1)$ creates a unique minimal tetrahedralization of $\mathcal{P}$. □

Using a similar argument to that in the proof of Corollary 3.6, we can prove the following corollary on minimal tetrahedralizations of augmented two-level BP-polyhedra.

Corollary 3.7 *Let $\overline{\mathcal{P}}$ be a simple polyhedron. By removing all the degree 3 vertices in $G(\overline{\mathcal{P}})$ recursively, we obtain a new polyhedron $\mathcal{P}$. If $\mathcal{P}$ is the two-level BP-polyhedron described in Theorem 3.6, then $PE(\mathcal{P}) \cup PE(\mathcal{P}_1)$ creates a unique minimal tetrahedralization of $\mathcal{P}$.*

3.4 Algorithm

In this section, we present an algorithm for computing the minimal tetrahedralization of augmented BP-polyhedra. This algorithm is mainly based on Corollary 3.6. The input of the algorithm is a simple polyhedron $\mathcal{P}$. There are two kinds of output: if $\mathcal{P}$ is an augmented BP-polyhedron, then the algorithm generates the set of all the tetrahedra in the minimal tetrahedralization of $\mathcal{P}$; otherwise, the algorithm reports that $\mathcal{P}$ is not an augmented BP-polyhedron. Corollary 3.6 can be rewritten as an algorithm as follows:

1. Trim off all the stacked polyhedra from $\mathcal{P}$ and produce a new polyhedron $\mathcal{P}_1$.
2. Trim off all the BP-nets from $\mathcal{P}_1$ and produce a new polyhedron $\mathcal{P}_2$.
3. Trim off all the stacked polyhedra from $\mathcal{P}_2$ and produce a new polyhedron $\mathcal{P}_3$.
4. If $\mathcal{P}_3$ is empty, then the minimal tetrahedralization of $\mathcal{P}$ consists of all the tetrahedra trimmed off from $\mathcal{P}$; otherwise, $\mathcal{P}$ is not an augmented BP-polyhedron.

In order to design an efficient algorithm, we mix the above steps; that is, trim off the stacked tetrahedron or BP-net when it is found as in the following algorithm. Step 1 of this algorithm does the necessary preprocessing. The main body of this algorithm (Step 2) is an iterative procedure that handles each vertex of $\mathcal{P}$ to find a tetrahedron or BP-net. Here is an outline of the algorithm.

Algorithm MINTETRA.
Input: A simple polyhedron $\mathcal{P}$.
Output: If $\mathcal{P}$ is not an augmented BP-polyhedron, then return (NO). Otherwise, return (YES, T), where T is the tetrahedron set of the unique minimal tetrahedralization of $\mathcal{P}$.

1. Let adj(u) denote the adjacency list of the vertex u. $V \leftarrow V(P)$ and $U \leftarrow \emptyset$.
2. **for** each vertex $u \in V$, **do**
   (* Check the degree of u. *)

2.1. **if** $\deg(u) = 3$ (suppose $\text{adj}(u) = \{u_1, u_2, u_3\}$), **then**

2.1(a). **if** there exists one $u_i (i = 1, 2, 3)$ of degree 3, **then**
$T \leftarrow \{uu_1u_2u_3\}$. **return** (YES, T) and stop (Theorem 3.1).
endif

2.1(b). **if** $\deg(u_i) > 3, i = 1, 2, 3$, **then**
$T \leftarrow \{uu_1u_2u_3\}$, $V \leftarrow V - \{u\}$.
Update the adjacency list of $u_i, i = 1, 2, 3$.
for $i = 1, 2, 3$,
if $u_i \in U$ and $\deg(u_i) \leq 4$, **then**
$U \leftarrow U - \{u_i\}$, $V \leftarrow \{u_i\}$.
endif
endfor
endif
endif

2.2. **if** $\deg(u) > 4$, **then**
$U \leftarrow \{u\}$, $V \leftarrow V - \{u\}$.
if $V = \emptyset$, **then**
return (NO) and stop.
endif
endif

2.3. **if** $\deg(u) = 4$ (suppose $\text{adj}(u) = \{u_1, u_2, u_3, u_4\}$), **then**

2.3(a). **if** there exists at least one u_i of degree 4 (suppose $\deg(u_1) = 4$), **then**
there exists a BP-net $B(pu'uu_1u''q)$ (Theorem 3.4).
if pq is a diagonal, **then**
$V \leftarrow V - \{u, u_1\}$, $T \leftarrow \{pqu'u, pquu_1, pqu_1u''\}$.
if $|V \cup U| = \emptyset$, **then**
return (YES, T) and stop.
endif
if $|V \cup U| \neq \emptyset$, **then**
Update the adjacency list of p, q, u', u'' (Note that pq is a new edge). Move p, q, u' or u'' from U to V if it is in U and its degree is less than 5.
endif
endif

endif

2.3(b). **if** there exists one u_i of degree 3 (suppose $\deg(u_1) = 3$), **then**
Change u into u_1, go to step 2.1.
endif

2.3(c). **if** $\deg(u_i) > 4, i = 1, 2, 3, 4$, **then**
$U \leftarrow \{u, u_1, u_2, u_3, u_4\}$, $V \leftarrow V - \{u, u_1, u_2, u_3, u_4\}$.
if $V = \emptyset$, **then**
return (NO) and stop.
endif
endif
endif
endfor

In Algorithm MINTETRA, we keep track of three lists. For each vertex u of $\mathcal{P}$, an adjacency list adj(u) contains all the adjacent vertices of u in the current polyhedron. The current polyhedron means the polyhedron in the current iteration after trimming off some tetrahedra from $\mathcal{P}$ (these tetrahedra are stored in T). U is a list of vertices each of whose degree has been checked in Step 2 and is greater than 4 in the current polyhedron except those vertices entering U in Step 2.3(c). V is a list of candidate vertices in the current polyhedron. There are two cases for a vertex u staying in V:

1. The degree of u has not been checked in Step 2.

2. When the degree of u is checked the first time, its degree is greater than 4. Thus, u is moved from V to U in Step 2.2. However, after trimming off the tetrahedra in Steps 2.1(b) and 2.3(a), the degrees of some vertices in U may decrease. If the new degree is less than 5, then the algorithm moves the vertex from U to V.

In order to prove the correctness of Algorithm MINTETRA, let us first assume that the input $\mathcal{P}$ is an augmented BP-polyhedron. As mentioned above, the main body (Step 2) of the algorithm performs the vertex trimming described by Corollary 3.6. In Step 2, the degree of each vertex $u \in V$ is checked. If a stacked tetrahedron is found (i.e., $\deg(u) = 3$), then trim it off and add it to T in Step 2.1. If a BP-net is found (i.e., $\deg(u) = \deg(u_1) = 4$), then trim it off and add the tetrahedralized BP-net to T in Step 2.3(a). Let $B(pv_0v_1...v_mq)$ be a proper BP-net in $\mathcal{P}$. If we only trim off a part of the BP-net, say, $B(pv_iv_{i+1}v_{i+2}v_{i+3}q)$, $0 \leq i \leq m-3$, then the remaining part of the BP-net, that is, $B(pv_0...v_iq)$ and $B(pv_{i+3}...v_mq)$, consists of two stacked polyhedra in the current polyhedron (because pq has been added). All tetrahedra in these stacked polyhedra will be trimmed off one at a time in the subsequent iterations. Thus, Step 2.3(a) actually trims off a BP-net and produces at most two stacked tetrahedra in the current polyhedron; this does not influence the other BP-nets because all the BP-nets are proper and disjoint. From Corollary 3.6 we know that the minimal tetrahedralization of an augmented BP-polyhedron can be considered as a set of stacked polyhedra and tetrahedralized BP-nets. Therefore, the output T of the algorithm is the tetrahedron set of the minimal tetrahedralization of $\mathcal{P}$. If the algorithm returns (YES, T), it is easy to see that $\mathcal{P}$ is an augmented BP-polyhedron since we add only the polar edges in the algorithm.

From the above discussion, we have proven that Algorithm MINTETRA is correct.

Theorem 3.7 *If Algorithm* MINTETRA *is run on a simple polyhedron $\mathcal{P}$, then the*

algorithm returns (YES, T) if and only if $\mathcal{P}$ is an augmented BP-polyhedron.

For the complexity of the algorithm, we have the following theorem.

Theorem 3.8 *If Algorithm* MINTETRA *is run on a simple polyhedron $\mathcal{P}$, then its running time is $O(n^2)$ and its space is $O(n)$.*

Proof Since $G(\mathcal{P})$ is a planar graph, we have $|E(\mathcal{P})| = O(n)$. Thus, Step 1 requires $O(n)$ time and $O(n)$ space for the adjacency list. Step 2 is an iterative step. Step 2.1(a) requires constant time. Step 2.1(b) requires $O(n)$ time and $O(n)$ space for updating the adjacency list and checking if $u_i \in U$. Step 2.2 requires constant time. Step 2.3(a) requires $O(n)$ time and $O(n)$ space for updating the adjacency list and checking if pq is a diagonal and if $u_i \in U$. Steps 2.3(b) and 2.3(c) require constant time.

In Step 2, each vertex $u \in V(\mathcal{P})$ is checked at most two times. If $\deg(u) = 3$, it is checked only one time because it is trimmed off in Step 2.1. If $\deg(u) \geq 4$, it is checked at most two times, the first time when it moves from V to U in Steps 2.2 and 2.3(c), the second time when it moves from U to V in Steps 2.1(b) and 2.3(a). So the **for** loop iterates $O(n)$ times. Thus, the total running time in Step 2 is $O(n^2)$. Therefore, computing the minimal tetrahedralization of an augmented BP-polyhedron takes $O(n^2)$ time and $O(n)$ space. □

3.5 Discussion

By definition, a BP-polyhedron must satisfy two conditions, one regarding the BP-net (i.e., the BP-nets are proper and disjoint) and the other regarding the stacked polyhedron (i.e., the new polyhedron generated is a stacked polyhedron). For a polyhedron $\mathcal{P}$ containing BP-nets on its surface, if either of the two conditions cannot be satisfied, then the polar edges of these BP-nets may not create any tetrahedralization of $\mathcal{P}$. Two counterexamples are presented below.

The first counterexample is Schonhardt's polyhedron depicted in Figure 2.11(a). The construction of this polyhedron was described in Section 2.6. Schonhardt's

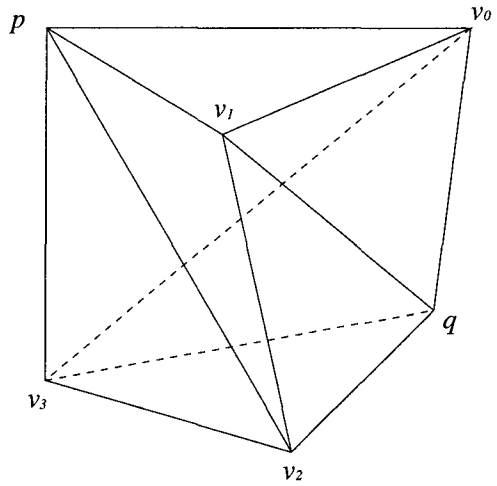

Figure 3.20: Schonhardt's polyhedron is a BP-net.

polyhedron is also a BP-net, specifically a bipyramid. As shown in Figure 3.20, $B(pv_0v_1v_2v_3q)$ is a BP-net. However, because the polar edge pq lies outside of the polyhedron, it is not a proper BP-net. Thus, we cannot obtain the result of Theo-

rem 3.5. In fact, it is well known that Schonhardt's polyhedron cannot be tetrahedralized [90].

The second counterexample is a convex polyhedron constructed from Schonhardt's polyhedron. Let $p_1p_2p_3q_1q_2q_3$ be a Schonhardt's polyhedron, where $p_1p_2p_3$ is the top triangle, $q_1q_2q_3$ is the bottom triangle, and p_1q_2,p_2q_3,p_3q_1 are three reflex edges. Let $B(p_1q_1a_1a_2p_2q_2)$, $B(p_2q_2b_1b_2p_3q_3)$ and $B(p_3q_3c_1c_2p_1q_1)$ be disjoint proper BP-nets added to the polyhedron such that p_1q_2,p_2q_3,p_3q_1 are the polar edges of the BP-nets respectively (Figure 3.21). We can arrange a_i,b_i,c_i, $i = 1, 2$ such that the constructed polyhedron $\mathcal{P}$ is convex.

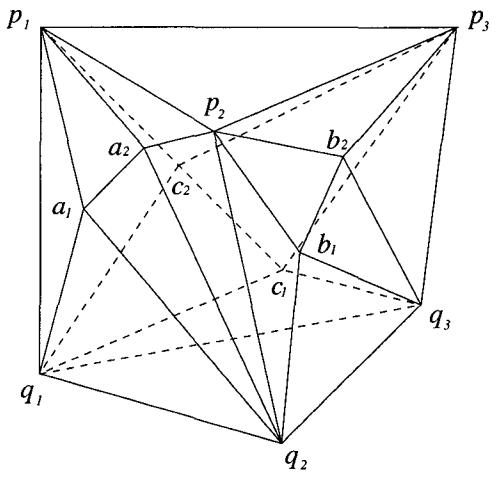

Figure 3.21: Schonhardt's polyhedron with three BP-nets.

After removing the vertices a_i, b_i, c_i and their adjacent edges from $\mathcal{P}$ and adding the polar edges p_1q_2, p_2q_3, p_3q_1, we get Schonhardt's polyhedron which is not a stacked polyhedron. Thus, we cannot guarantee that $\{p_1q_2, p_2q_3, p_3q_1\}$ creates a minimal tetrahedralization of $\mathcal{P}$.

In Section 3.3.1, we defined the two-level BP-polyhedron. Similarly, we can define the h-level BP-polyhedron recursively. Let $B(p^{(i)}v_0^{(i)}v_1^{(i)}...v_{m_i}^{(i)}q^{(i)})$ $1 \leq i \leq k$ be all the disjoint proper BP-nets on the surface of $\mathcal{P}$. After removing the vertices $v_j^{(i)}$ $(j = 1, ..., m_i - 1,\ i = 1, ..., k)$ and their adjacent edges from $\mathcal{P}$, and adding edges $p^{(i)}q^{(i)}$ $(1 \leq i \leq k)$, we obtain a new polyhedron $\mathcal{P}_{h-1}$. If $\mathcal{P}_{h-1}$ is a $(h-1)$-level BP-polyhedron, then $\mathcal{P}$ is called an *h-level* BP-polyhedron, and $\mathcal{P}_{h-1}$ is called the *internal* BP-polyhedron of $\mathcal{P}$, where h is a positive integer. The stacked polyhedron can be defined as the 0-level BP-polyhedron. Obviously, the 1-level BP-polyhedron is a BP-polyhedron as defined in Section 3.3.1. Let $\mathcal{P}'$ be an h-level BP-polyhedron. If $\mathcal{P}$ is constructed from $\mathcal{P}'$ by successive additions of tetrahedra on the triangle facets, then $\mathcal{P}$ is called an *augmented* h-level BP-polyhedron.

In Theorems 3.5 and 3.6, we proved that all polar edges create a unique minimal tetrahedralization of the BP-polyhedron and the special two-level BP-polyhedron respectively. Motivated by these results, we make the following conjecture:

Conjecture. If $\mathcal{P}$ is an augmented h-level BP-polyhedron $(h \geq 2)$, then all polar edges of the BP-nets on the surface of $\mathcal{P}$ and $\mathcal{P}_i$ $(1 \leq i \leq h-1)$ create a unique minimal tetrahedralization of $\mathcal{P}$, where $\mathcal{P}_i$ is the internal BP-polyhedron of $\mathcal{P}_{i+1}$.

By applying a similar method to that used in the proof of Theorems 3.5 and 3.6, we can prove most cases of this conjecture; however, the proofs involved are long. It

is our hope that a simpler proof exists; however, we suspect that such a proof will probably be based on a very different strategy.

Chapter 4

A Linear Size Tetrahedralization of Two Nested Convex Polyhedra

In this chapter, we present a method to tetrahedralize the region between two nested convex polyhedra without introducing Steiner points. In Section 4.1, we survey the literature regarding the tetrahedralizations of two nested convex polyhedra and give the basic idea of our algorithm. In Section 4.2, we give some notations and definitions. In Section 4.3, we describe our tetrahedralization algorithm. In Section 4.4, we prove that the number of tetrahedra produced by our algorithm is at most $9n - 6$. Finally, we conclude our work in Section 4.5.

4.1 Introduction

Tetrahedralizations are significantly more complicated than triangulations. While a simple polygon with n vertices is always triangulatable and each triangulation contains exactly $n-2$ triangles, a simple polyhedron with n vertices may not be tetrahedralizable [90] and the number of tetrahedra in its tetrahedralizations may vary from $O(n)$ to $O(n^2)$ even if it is tetrahedralizable [24]. Therefore, the following two problems need to be investigated: to identify classes of polyhedra which are tetrahedralizable and to find tetrahedralization methods for these polyhedra which produce a linear or even an optimal number of tetrahedra.

So far, very few non-trivial classes of simple polyhedra are known to be tetrahedralizable without introducing Steiner points. Goodman and Pach [54] showed the following two kinds of polyhedra are tetrahedralizable: the polyhedra induced by the region between two *side-by-side* convex polyhedra, and the polyhedra induced by the region between two *nested* convex polyhedra. Let n be the number of vertices in the two polyhedra. In the side-by-side case, Bern [16] proved that $\Omega(n^2)$ tetrahedra are necessary to tetrahedralize the region between two side-by-side convex polyhedra. In the nested case, Bern [16] presented a method to tetrahedralize the region between two nested convex polyhedra with $O(n \log n)$ tetrahedra and no Steiner points in $O(n \log n)$ time. He proposed an open question as to whether the region between two nested convex polyhedra can be tetrahedralized with a linear number of tetrahedra

without using Steiner points. Chazelle and Shouraboura [27] presented an algorithm which produces $O(n)$ tetrahedra by introducing Steiner points, so they only partially solved Bern's problem.

In this chapter, we answer Bern's question in the affirmative by presenting an algorithm to tetrahedralize the region between two nested convex polyhedra with a linear number of tetrahedra without introducing any Steiner points. The basic idea comes from the following analysis. The "cap removal" method invented by Chazelle and Palios [26] has been pushed to its limit in the two nested convex polyhedra case by Bern [16]. Let P and Q be two convex polyhedra such that Q is contained in P. A "cap" with tip v of P is the closed region between P and the convex hull of all vertices of P and Q except v. After a cap is removed, some vertices $S_Q \subset V(Q)$ may lie on the surface of $CH((V(P) - \{v\}) \cup V(Q))$. Thus the subsequent cap removal process will avoid choosing a tip vertex belonging to S_Q. For simplicity, Bern uses $P - Q$ to denote the region between P and Q as well as the remainder after the subsequent removals of the caps. His method is to remove one cap at a time with the tip vertices in $V(P)$ until $P - Q$ contains only the vertices of Q. Note that a cap can be decomposed into k tetrahedra if it contains k triangle facets in its bottom. Thus, the number of tetrahedra produced by this method depends on the number of vertices in the caps. Bern's method may result in $O(n \log n)$ tetrahedra by removing the caps whose tips have the smallest degree among the remaining vertices of $V(P)$ on the surface of $P - Q$, where n is the number of vertices in P and Q. The reason why

the above bound is difficult to improve is as follows. In the worst case, the vertices with small degrees on the surface of $P - Q$ may lie on the surface of Q. This forces the method to choose the vertices of P with large degrees. Bern showed that if the number of $P - Q$ vertices is i, then the degree of the minimum-degree vertex of P can be $\frac{6n}{i}$. Overall the bound will be $n(2 + 6H_n)$, where H_n is the n-th Harmonic number.

Our observation is that Bern's method depends on the degree of the tip vertices of the caps. We will avoid this by removing a "big cap" (called *bigcap*) which contains some vertices of P and Q visible from v. Our method is to remove one bigcap at a time with the tip vertices in $V(P)$ until $P - Q$ contains only the vertices of Q. Notice that the number of tetrahedra is bounded by the summation of the number of interior edges and the number of vertices in P and Q. We can prove that the total number of interior edges is at most $8n$. Thus, our bigcap-removal method will produce at most $9n - 6$ tetrahedra for tetrahedralizing $P - Q$.

4.2 Preliminaries

Let P and Q be two convex polyhedra. We say that Q is *nested* in P if Q is entirely contained in P, where they may share some common facets, edges, or vertices. Let $P - Q$ denote the closed region between P and Q, and n denote the number of vertices in $P - Q$. A vertex $u \in V(P) \cup V(Q)$ is *visible* to (or can *see*) a point v in $P - Q$ if

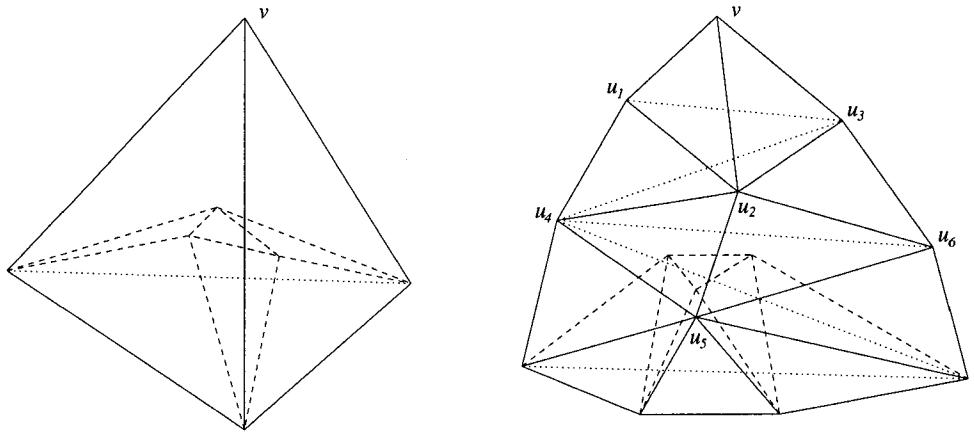

Figure 4.1: A cap with tip v (left) and a bigcap with tip v (right), where the shaded area is on the surface of Q.

and only if the interior of the line segment uv lies inside $P - Q$ and does not intersect Q. We define that u is visible to itself. Similarly, a vertex $u \in V(P) \cup V(Q)$ is visible to (or can see) a facet whose vertices belong to $V(P) \cup V(Q)$ if and only if u can see every point in the facet including its boundary.

For a vertex $v \in V(P)$, a *cap* with tip v is the closed region between P and $CH((V(P) - \{v\}) \cup V(Q))$ (see Figure 4.1(left)). A *bigcap* with tip v is the closed region between P and $CH((V(P) - U_v) \cup V(Q))$, where

$$U_v = \{u \mid u \in V(P), u \text{ and its incident facets are visible to } v\}$$

(see Figure 4.1(right)). Note that a bigcap with tip v is "bigger" than a cap with tip v. As our algorithm removes one cap or constrained bigcap at a time with the tip vertices in $V(P)$ until all vertices of P are removed, let $\overline{P}$ always denote the current convex hull of the remaining vertices of P and Q during the processing of the algorithm.

Now let us consider a constrained bigcap that will be used in our algorithm. Let S be the union of some chains on the surface of $\overline{P}$ such that $V(S) \subset V(P)$. Define

$$U_v^S = \{u \mid u \in V(P) - V(S), u \text{ and its incident facets are visible to } v\}$$

If the closed region between $\overline{P}$ and $CH((V(\overline{P}) - U_v^S) \cup V(Q))$ is a simple polyhedron, then it is called an *S-bigcap* of v, denoted as S-bigcap(v). S-bigcap(v) can be considered as a bigcap restricted by S (see Figure 4.2). Note that a bigcap with

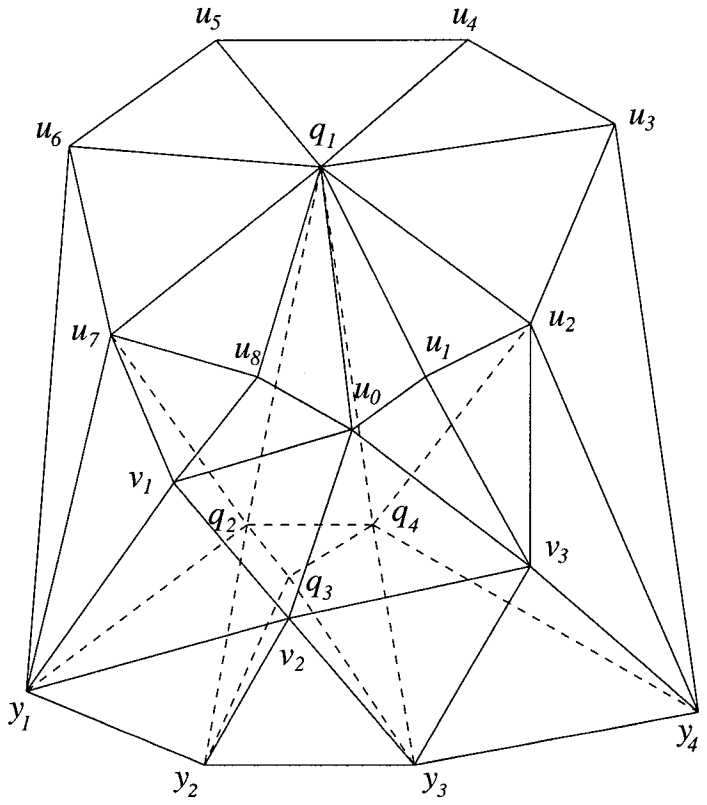

Figure 4.2: This figure illustrates an S-bigcap with tip u_0, where $S = u_2u_3...u_7$ and $U_{u_0}^S = \{u_1, u_8, v_1, v_2, v_3\}$. The vertices q_1, q_2, q_3, q_4 belong to $V(Q)$, and all the other vertices belong to $V(P)$. After S-bigcap(u_0) is removed, the dashed edges lie on bottom(u_0), where the shaded area is on the surface of Q. The solid edges inside polygon $q_1u_7y_1y_2y_3y_4u_2$ form dome(u_0).

tip v is the S-bigcap of v with $V(S) = \emptyset$. The intersection of S-bigcap(v) and $CH((V(\overline{P}) - U_v^S) \cup V(Q))$ is called the *bottom* of S-bigcap(v), denoted as bottom(v), and the intersection of S-bigcap(v) and $\overline{P}$ is called the *dome* of S-bigcap(v), denoted as dome(v).

Lemma 4.1 *S-bigcap(v) is a star-shaped polyhedron with v in its kernel.*

Proof Since S-bigcap(v) is a simple polyhedron, any interior vertex of dome(v) does not lie on bottom(v), and any interior vertex of bottom(v) does not lie on dome(v). From the definition of U_v^S, we know that v can see each facet on dome(v). Suppose that there exists a point x on bottom(v) which is not visible to v. Since bottom(v) lies on the surface of the convex polyhedron $CH((V(\overline{P}) - U_v^S) \cup V(Q))$, the portion of bottom($v$) which prevents v from seeing x also blocks v from seeing some points on the boundary of bottom(v). It follows that v cannot see some points on the boundary of dome(v). This is a contradiction. Thus, v can see every points on bottom(v). Therefore, S-bigcap(v) is a star-shaped polyhedron with v in its kernel. □

From Lemma 4.1, we know that S-bigcap(v) is a star-shaped polyhedron consisting of a convex surface (i.e., dome(v)) and a concave surface (i.e., bottom(v)) (refer to Figures 4.1 and 4.2). Figure 4.3 illustrates how an S-bigcap with tip vertex v is removed from convex polyhedron P.

The union of all facets on the surface of $CH((V(\overline{P}) - U_v^S) \cup V(Q))$ which is visible to v if S-bigcap(v) has not been removed is called the *view* of v, denoted as

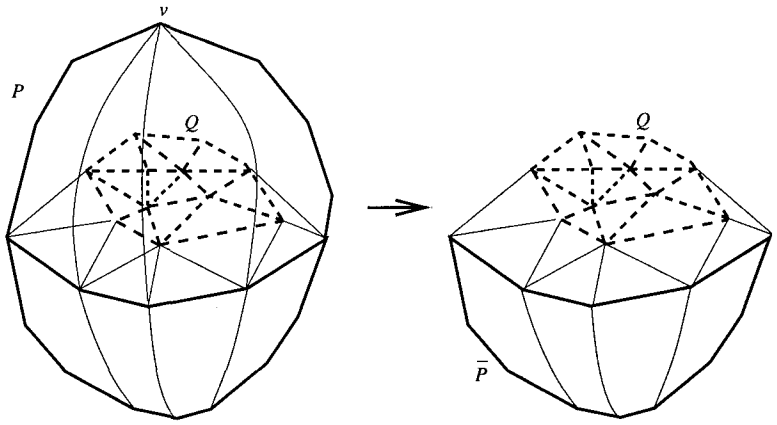

Figure 4.3: An S-bigcap with tip v is removed. The dashed edges lie on the surface of Q and the shaded area lies on the surface of P and $\overline{P}$.

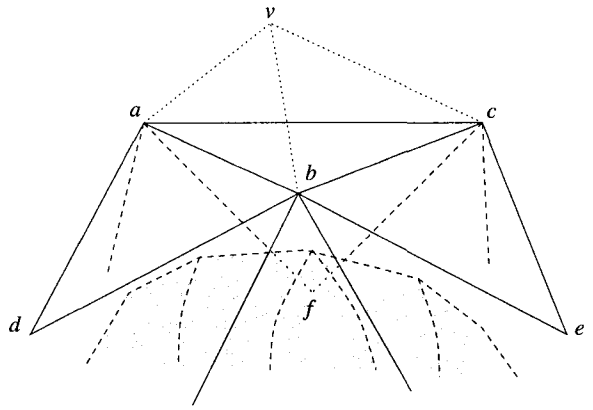

Figure 4.4: The tetrahedron $vabc$ is S-bigcap(v). The triangle face abc is bottom(v). The surface consisting of triangle facets abc, abd, bce and caf is the view(v). The shaded region is a portion of Q.

view(v). bottom(v) may be a subset of view(v), as illustrated in Figure 4.4. Let $\{v_i \mid 1 \leq i \leq m\}$ be a set of vertices of P. If we remove $S(v_i)$-bigcap(v_i) $(1 \leq i \leq m)$ in different vertex orders, the resulting polyhedron $\overline{P}$ may be different. For a given vertex order, say, $i = 1, 2, ..., m$, after all $S(v_i)$-bigcap(v_i) are removed, the intersection of the surface of the resulting $\overline{P}$ and $\bigcup_{i=1}^{m}$ view(v_i) is called the view of v_i in the removal order $i = 1, 2, ..., m$, and is denoted by view($v_1, v_2, ..., v_m$).

Lemma 4.2 view(v) *does not contain any vertices of* $V(P) - V(S)$ *in its interior.*

Proof Consider an arbitrary S-bigcap(v). Suppose that view(v) contains a vertex $p \in V(P) - V(S)$ in its interior. Then $p \notin V(S)$. Since $\overline{P}$ and $CH((V(\overline{P}) - U_v^S) \cup V(Q))$ are convex, p and its incident facets are visible to v. It follows that $p \in U_v^S$. Thus, p cannot lie on the surface of $CH((V(\overline{P}) - U_v^S) \cup V(Q))$. This contradicts the definition of view(v). □

From this lemma, we can derive the following result.

Corollary 4.1 view($v_1, v_2, ..., v_m$) *does not contain any vertices of* $V(P) - \bigcup_{i=1}^{m} V(S(v_i))$ *in its interior.*

4.3 Algorithm

In this section, we describe an algorithm to tetrahedralize $P - Q$ into a linear number of tetrahedra. The idea of our algorithm is as follows. First we remove caps with tip vertices of small degree until some vertices of Q appear on the current surface of $\overline{P}$. We find the boundary vertices of the portion of Q which appear on the surface of $\overline{P}$, denoted as Q_a. We also find a set of polygons on the surface of $\overline{P}$, denoted as P_a, such that the polygons of P_a surround Q_a and contain a minimum number of edges. Note that some polygons in P_a may collapse to a chain or even a vertex. Let Q_a^{view} be the boundary of the union of all facets on the surface of $\overline{P}$ which is

visible to a vertex of Q_a. In order to extend the portion of Q which appears on the current surface, we remove some S-bigcap(v_i) ($v_i \in V(P_a)$) in the order $i = 1, 2, ..., m$ such that the vertices of P_a are removed and the vertices of Q_a^{view} are removed or on the boundary of view($v_1, v_2, ..., v_m$) (this boundary is denoted as P_b). Update Q_a and P_a. If Q_a^{view} is removed, then repeat the above process to extend the portion of Q which appears on the surface of $\overline{P}$. If some vertices of Q_a^{view} exist on P_b and are not adjacent to any vertex of $V(Q_a)$, we remove all these vertices by removing some S-bigcaps. Then Q_a^{view} will be removed after we remove the S-bigcaps with tip vertices in the updated P_a. We now find the new Q_a^{view} and repeat the above process until $\overline{P} = Q$. From Lemma 4.1, we can tetrahedralize S-bigcap(v) by adding edges between v and each other vertex in S-bigcap(v). It is easy to see that after an S-bigcap is removed, the remainder $\overline{P} - Q$ is still a region between two nested convex polyhedra.

Algorithm BIGCAP-REMOVE

1. Remove the caps with the tip vertices of degree less than 6 until a portion of Q appears on the current surface of $\overline{P}$ when we remove the cap with tip $p \in V(P)$. Let P_a be a set of polygons which only contain the boundary polygon of bottom(p) at this step. Compute

$$Q_a = \{q \mid q \in V(Q), \text{ there exists a vertex } v \in V(P) \text{ such that } vq \in E(\overline{P})\}.$$

2. Compute

$$V_a^P = \{v \mid v \in V(P_a) \text{ and there exists at least one edge } q_1q_2 \in E(Q) \text{ such that } vq_1q_2 \text{ is a facet on the surface of } \overline{P}\}.$$

The vertices in V_a^P partition the polygons of P_a into a set of chains C_a (see Figure 4.5). Let $S_a(v)$, $v \in V(P_a)$, be the union of the chains in C_a except the chain containing v.

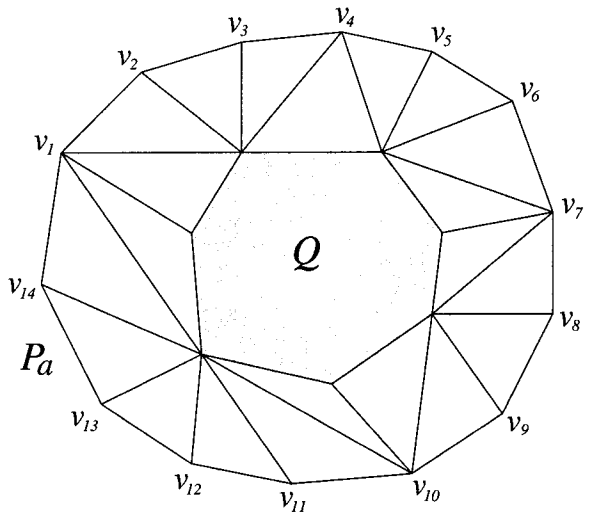

Figure 4.5: The outer boundary is a polygon in P_a and the shaded area is a portion of Q. $V_a^P = \{v_1, v_4, v_7, v_{10}\}$ partitions the polygon into four chains: $v_1v_2v_3v_4$, $v_4v_5v_6v_7$, $v_7v_8v_9v_{10}$ and $v_{10}v_{11}v_{12}v_{13}v_{14}v_1$.

3. Let Q_a^{view} be the boundary of the union of all facets on the surface of $\overline{P}$ which is visible to a vertex of Q_a.

   (a) Find the minimum number of vertices $v_1, v_2, ..., v_{m'}, v_{m'+1}, ..., v_m \in V(P_a)$ such that $v_1, ..., v_{m'} \in V(P_a) - V_a^P$, $v_{m'+1}, ..., v_m \in V_a^P$ and after remov-

ing $S_a(v_i)$-bigcap(v_i) in the order $i = 1, 2, ..., m$, the vertices of P_a are removed and the vertices of Q_a^{view} are removed or on the boundary of view$(v_1, v_2, ..., v_m)$. If there exist more than one optimal solutions, we break ties by selecting the solution with the minimum number of edges added (refer to Remark 1). For each $i = 1, 2, ..., m$, after $S_a(v_i)$-bigcap(v_i) is removed, update chains in C_a which contain v_i such that each edge in the new chain forms a triangle with a vertex of $V(Q)$.

(b) Remove $S_a(v_i)$-bigcap(v_i) in the order $i = 1, 2, ..., m$. The boundary of view$(v_1, v_2, ..., v_m)$ is denoted as P_b.

(c) For each $S_a(v_i)$-bigcap(v_i) $(i = 1, 2, ..., m)$, if there exist facets in bottom(v_i) which are not triangles, then triangulate these facets arbitrarily. Tetrahedralize $S_a(v_i)$-bigcap(v_i) by adding edges between v_i and each other vertex in $S_a(v_i)$-bigcap(v_i).

4. If $\overline{P} = Q$, then stop. Output the tetrahedra produced in Steps 3c and 6b.

5. Compute

$$Q'_a = \{q \mid q \in V(Q), \text{ there exists a vertex } v \in V(P) \text{ such that } vq \in E(\overline{P})\}.$$

Since $\overline{P}$ has changed after the S-bigcaps are removed in Step 3b, we know that Q'_a and Q_a may be different. Let P'_a be a set of the smallest polygons in the area bounded by P_b on the current surface of $\overline{P}$ such that the polygons of P'_a

surround Q'_a and do not intersect Q'_a. Here, the smallest polygon is a polygon with minimum number of edges (see Figure 4.6). If Q_a^{view} is removed, then set

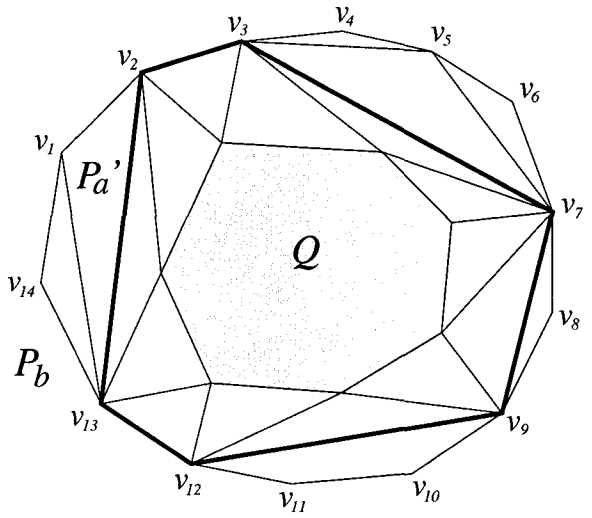

Figure 4.6: The outer boundary is P_b and the shaded area is a portion of Q. Q'_a consists of the seven vertices of the shaded area. $v_2v_3v_7v_9v_{12}v_{13}$ is the smallest polygon surrounding Q'_a and not intersecting Q'_a.

$Q_a \leftarrow Q'_a$ and $P_a \leftarrow P'_a$. Go to Step 2.

6. For each vertex $q \in V(Q_a) \cap V(Q'_a)$, consider the triangle $qy'_1y'_2$, where $y'_1y'_2$ is an edge in P'_a. If q is visible to some vertices of the chain in P_b whose two endpoints are y'_1 and y'_2 and whose interior vertices are not adjacent to any vertex in $V(Q'_a)$, then let $y_1y_2...y_k$ denote this chain with $y_1 = y'_1$ and $y_k = y'_2$. (Refer to Remark 2.)

   (a) The area on $\overline{P}$ whose boundary is $y_1y_2...y_k$ can be considered as a triangulation surface of the polygon $y_1y_2...y_k$. All chords of the chain $y_1y_2...y_k$ which belong to $E(\overline{P})$ form a partial order according to the subset-relation

of their corresponding chain (see Figure 4.7). It is easy to see that y_1y_k is a maximal element of this partial order.

(b) For each chord y_iy_j $(1 \leq i < j \leq k)$ in the decreasing partial order, if there exists a vertex y_l $(i < l < j)$ such that $\{y_{i+1}, ..., y_{j-1}\}$ is contained in the interior of dome(y_l), then remove S_b-bigcap(y_l), where $S_b = P_b - y_{i+1}y_{i+2}...y_{j-1}$. Tetrahedralize S_b-bigcap(y_l) by using the method described in Step 3c.

(c) Update the chain $y_1y_2...y_k$ and P_b by replacing the chain $y_iy_{i+1}...y_j$ with edge y_iy_j. Then repeat Steps 6b and 6c until vertices $y_2, y_3, ..., y_{k-1}$ are removed.

7. Set $Q_a \leftarrow Q'_a$ and $P_a \leftarrow P'_a$. Go to Step 2.

Remark 1: In Step 3a, we find the minimum number of vertices $v_1, v_2, ..., v_{m'}, v_{m'+1}, ..., v_m \in V(P_a)$ such that $v_1, ..., v_{m'} \in V(P_a) - V_a^P$, $v_{m'+1}, ..., v_m \in V_a^P$ and after removing $S_a(v_i)$-bigcap(v_i) in the order $i = 1, 2, ..., m$, the vertices of P_a are removed and the vertices of Q_a^{view} are removed or on P_b. For each vertex v of V_a^P, there exists a chain on Q such that v is adjacent to each vertex of the chain. These chains form the boundary of the portion of Q which appears on the surface of $\overline{P}$ (this boundary is denoted as $\overline{Q}_a$). Thus, for each edge $q_1q_2 \in \overline{Q}_a$, there exists a vertex $v \in V_a^P$ such that vq_1q_2 is a triangle facet on the surface of $\overline{P}$. If we do not remove the vertex v,

the triangle facet vq_1q_2 will always stay on the surface of $\overline{P}$. Let

$$\text{bridge}(\overline{Q}_a) = \{vq_1q_2 \mid v \in V(P_a), q_1q_2 \in \overline{Q}_a, vq_1q_2 \text{ is a facet on the surface of } \overline{P}\}.$$

Note that bridge$(\overline{Q}_a)$ spans the connection structure between Q_a and P_a. We want to hold this structure as long as possible in the removal process of Step 3b. So we remove the vertices of $V(P_a) - V_a^P$ first, and then remove the vertices of V_a^P. In order to control the number of edges added when removing S-bigcaps in Step 3b, we find the minimum number of vertices among $V(P_a)$ and treat these vertices as the tip vertices of the removed S-bigcaps, and we break ties by selecting the solution with the minimum number of edges added.

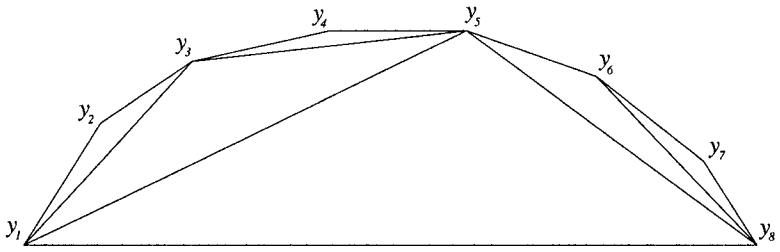

Figure 4.7: All chords on the triangulation surface of the polygon $y_1y_2...y_8$ form a partial order. For example, $y_1y_3 \prec y_1y_5 \prec y_1y_8$ (since chain(y_1y_3) is a subchain of chain(y_1y_5) and chain(y_1y_5) is a subchain of chain(y_1y_8)).

Remark 2: In Step 6a, the area enclosed by the polygon $y_1y_2...y_k$ on $\overline{P}$ is a triangulation surface without an interior vertex (refer to Lemma 4.7). For each chord y_iy_j, there is a corresponding subchain in $y_1y_2...y_k$ with endpoints y_i and y_j, denoted as chain(y_iy_j). Define $y_iy_j \prec y_{i'}y_{j'}$ if and only if chain(y_iy_j) is a subchain of chain$(y_{i'}y_{j'})$. Thus, all the chords form a partial order (see Figure 4.7). Since any triangulation

has at least two "ears", there always exists at least one chord y_iy_{i+2} $(1 \leq i \leq k-2)$ on the triangulation surface of $y_1y_2...y_k$. This guarantees that there exists a vertex y_{i+1} such that $\{y_{i+1}\}$ is contained in the interior of dome(y_{i+1}). Thus, Step 6b can always find a chord y_iy_j with a vertex y_l such that $\{y_{i+1}, ..., y_{j-1}\}$ is contained in the interior of dome(y_l). Suppose there exist vertices y_l and $y_{l'}$ in chords y_iy_j and $y_{i'}y_{j'}$ such that $\{y_{i+1}, ..., y_{j-1}\}$ and $\{y_{i'+1}, ..., y_{j'-1}\}$ are contained in the interiors of dome(y_l) and dome$(y_{l'})$ respectively. If $y_iy_j \prec y_{i'}y_{j'}$, then the boundary of bottom(y_l) may be enclosed by the boundary of bottom$(y_{l'})$ on the surface of $\overline{P}$. In order to avoid this case, we consider each chord in the decreasing partial order.

Now let us consider the correctness of the BIGCAP-REMOVE algorithm. we first prove that $S_a(v)$-bigcap(v) and S_b-bigcap(v) are actually S-bigcaps.

Lemma 4.3 *$S_a(v)$-bigcap(v) in Step 3a and S_b-bigcap(v) in Step 6b are S-bigcaps.*

Proof Consider an $S_a(v)$-bigcap(v). From the definition of S-bigcap, we only need to prove the closed region between $\overline{P}$ and $CH((V(\overline{P}) - U_v^{S_a(v)}) \cup V(Q))$ is a simple polyhedron. Let R denote this closed region, R_1 be the intersection of the surfaces of R and $\overline{P}$, and R_2 be the intersection of the surfaces of R and $CH((V(\overline{P}) - U_v^{S_a(v)}) \cup V(Q))$. Notice that $\overline{P}$ and $CH((V(\overline{P}) - U_v^{S_a(v)}) \cup V(Q))$ are two nested convex polyhedra. Since $U_v^{S_a(v)}$ contains at least one vertex, i.e., v, R is a solid with positive volume and the intersection of the surfaces of $\overline{P}$ and $CH((V(\overline{P}) - U_v^{S_a(v)}) \cup V(Q))$ is a polygon, denoted as R_{12}. From the definition of $U_v^{S_a(v)}$, we know that R_1 does not

contain a vertex of Q in its interior. We now prove R_2 does not contain a vertex of P in its interior. Suppose R_2 contains a vertex $p \in V(P)$ in its interior. Then p and its incident facets are visible to v. Recall that $S_a(v)$ is the union of the chains in C_a except the chain containing v. From the structure of these chains we know that each vertex of $S_a(v)$ cannot be the interior vertex of R_2. Thus $p \notin V(S_a(v))$. It follows that $p \in U_v^{S_a(v)}$. Thus, p cannot lie on the surface of $CH((V(\overline{P}) - U_v^{S_a(v)}) \cup V(Q))$. This is a contradiction. Hence, R_2 does not contain a vertex of P in its interior. It follows that the intersection of R_1 and R_2 is the polygon R_{12}. Therefore, R is a simple polyhedron. Similarly, we can prove that S_b-bigcap(v) is an S-bigcap. □

From Lemmas 4.1 and 4.3, we have the following result.

Corollary 4.2 *$S_a(v)$-bigcap(v) and S_b-bigcap(v) are star-shaped polyhedra with v in their kernels.*

From Corollary 4.2, we know that Step 3c is correct. We now prove the correctness of Step 3a.

Lemma 4.4 *In Step 3a, if we select all the vertices of $V(P_a)$ in any order, such as $p_1, p_2, ..., p_k \in V(P_a)$ (where $k = |V(P_a)|$), then, after removing $S_a(p_i)$-bigcap(p_i) $i = 1, 2, ..., k$, the vertices of Q_a^{view} are removed or on the boundary of* view$(p_1, p_2, ..., p_k)$.

Proof Suppose there is an order of vertices $p_1, p_2, ..., p_k \in V(P_a)$ (where $k = |V(P_a)|$) such that after $S_a(p_i)$-bigcap(p_i) are removed in the order $i = 1, 2, ..., k$, there still

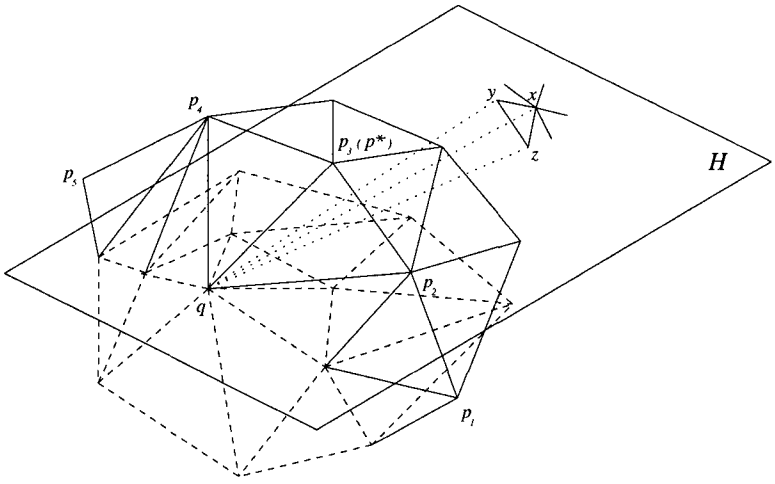

Figure 4.8: $q \in V(Q_a)$ is visible to $x \in V(Q_a^{\text{view}})$. p_i $(i = 1, 2, 3...)$ lies on P_a. The solid lines lie on P and the dashed lines lie on Q. The plane H passes through q and separates Q and tetrahedron $qxyz$.

exists a vertex $x \in V(Q_a^{\text{view}})$ on the resulting surface of $\overline{P}$. Let P' be the polyhedron before $S_a(p_i)$-bigcap(p_i) are removed, and xyz be a facet on the surface of P' which is visible to a vertex $q \in V(Q_a)$. Since Q and tetrahedron $qxyz$ are convex, there must exist a plane which passes through q and separates Q and tetrahedron $qxyz$. This plane cuts a polygon of P_a which surrounds Q_a into two parts (see Figure 4.8). On the side of the plane which does not contain Q, there exists at least one vertex of P_a, say p^*. It is easy to see that p^* can see facet xyz. Thus xyz lies on view(p^*). If x is an interior vertex of a view(p_j) $(1 \leq j \leq k)$, then x will be removed when we remove $S_a(p_j)$-bigcap(p_j). Hence, x must lie on the boundary of view(p^*) and lie on the boundary or exterior of view(p), $p \in V(P_a) - \{p^*\}$. Therefore, x lies on the boundary of view$(p_1, p_2, ..., p_k)$. □

From this lemma, we can derive the following result.

Lemma 4.5 *In Step 3a, there always exists the minimum number of vertices* $v_1, v_2, ..., v_{m'}, v_{m'+1}, ..., v_m \in V(P_a)$ *such that* $v_1, ..., v_{m'} \in V(P_a) - V_a^P$, $v_{m'+1}, ..., v_m \in V_a^P$ *and after removing* $S_a(v_i)$*-bigcap(*v_i*), i* $= 1, 2, ..., m$, *the vertices of* P_a *are removed and the vertices of* Q_a^{view} *are removed or on* P_b.

Proof Note that we can select the vertices in any order in Lemma 4.4. Thus, there exists a sequence of vertices $p_1, p_2, ..., p_k \in V(P_a)$ (where $k = |V(P_a)|$) such that $V_a^P = \{p_i \mid k - |V_a^P| + 1 \leq i \leq k\}$ and after removing $S_a(p_i)$-bigcap(p_i) in the order $i = 1, 2, ..., k$, the vertices of P_a are removed and the vertices of Q_a^{view} are removed or on P_b. Thus, the solution space of the optimization problem is not empty. Hence, there always exists an optimal solution. □

Lemma 4.6 P_b *does not contain any vertices of* P *in its interior.*

Proof Since P_b is the boundary of view($v_1, v_2, ..., v_m$), from Corollary 4.1 we know that P_b does not contain any vertices of $V(P) - \bigcup_{i=1}^{m} V(S_a(v_i))$ in its interior. From the structure of $S_a(v_i)$, we know that each vertex of $S_a(v_i)$ cannot be an interior vertex of view($v_1, v_2, ..., v_m$). Thus, P_b does not contain any vertices of P in its interior. □

Now let us consider the property of the chains in Step 6.

Lemma 4.7 *Let* $y_1y_2...y_k$ *be a chain in* P_b *such that* y_1y_k *is an edge in* P'_a *and each* y_i $(1 < i < k)$ *is not adjacent to any vertex of* Q. *Let* A *be the area on the surface of* $\overline{P}$ *which is bounded by the polygon* $y_1y_2...y_k$ *and does not contain any vertex in*

$V(P_b) - \{y_1, y_2, ..., y_k\}$. *The interior of* A *does not contain any vertex of* P *and* Q.

Proof It follows from Lemma 4.6 that the interior of A does not contain any vertex of P. Suppose $Q_A \subset V(Q)$ is a set of interior vertices of A. From Step 3c we know that view$(v_1, v_2, ..., v_m)$ is a triangulation surface. Since y_i $(1 < i < k)$ is not adjacent to any vertex of Q_A, there must exist some vertices of P which are interior vertices of A and adjacent to some vertices of Q_A. This is a contradiction. □

From Lemma 4.7 and Remark 2 we know that Step 6b is correct.

In the BIGCAP-REMOVE algorithm, after an S-bigcap is removed in Steps 3 or 6, the remainder $\overline{P} - Q$ can be considered as two nested convex polyhedra $\overline{P}$ and Q sharing some vertices, edges or facets (refer to Figure 4.3). Thus, the algorithm can iteratively remove S-bigcaps until $\overline{P} = Q$. From the above lemmas and analysis, we know that Algorithm BIGCAP-REMOVE is correct.

Theorem 4.1 *Let* P *and* Q *be two nested convex polyhedra such that* Q *is contained in* P. *Algorithm* BIGCAP-REMOVE *outputs a tetrahedralization of* $P - Q$.

4.4 The number of tetrahedra

In this section, we analyze the number of tetrahedra produced by the BIGCAP-REMOVE algorithm. When we remove an S-bigcap(v), bottom(v) may contain some vertices and edges of Q. An upper bound on the number of the interior edges in bottom(v) and not in $E(Q)$ is given in the following lemma.

Lemma 4.8 *For an S-bigcap(v), let P_v be the boundary of* bottom(v) *and Q_v be the portion of Q which appears in* bottom(v). *Let $E_i(v)$ be the set of the interior edges of* bottom(v) *which do not lie on Q_v. Then $|E_i(v)| \leq |V(P_v)| + 3|V(Q_v)| - 3$.*

Proof Since the total number of edges in a triangulation of bottom(v) is $3(|V(P_v)| + |V(Q_v)|) - |E(P_v)| - 3$, we know that the total number of interior edges of bottom(v) is $3(|V(P_v)| + |V(Q_v)|) - 2|E(P_v)| - 3$. Thus, $|E_i(v)| \leq |V(P_v)| + 3|V(Q_v)| - 3$, where equality holds if and only if Q_v does not contain an interior edge of bottom(v) on the surface of Q. □

The following lemma shows the relationship between the number of tetrahedra and the number of interior edges in a tetrahedralization of $P - Q$.

Lemma 4.9 *Let P and Q be two convex simplicial polyhedra such that P contains Q and P does not intersect Q. Let t and e_i be the number of tetrahedra and interior edges in a tetrahedralization of $P - Q$ respectively. Then $t = e_i + n - 6$*

Proof Let n_P, e_P, f_P (resp. n_Q, e_Q, f_Q) denote the number of vertices, edges and facets of P (resp. Q), respectively. Let e and f denote the number of edges and facets in a tetrahedralization of $P - Q$ respectively. Since P and Q are simplicial polyhedra, each facet on P and Q is a triangle. From Euler's formula we know that $n_P - e_P + f_P = 2$. Since the edges on the surface of P can be counted in two ways, that is, $3f_P = 2e_P$, we have $f_P = 2n_P - 4$. Similarly, we have $f_Q = 2n_Q - 4$. Since the facets in a tetrahedralization of $P - Q$ can be counted in two ways, that is, $4t = 2f - f_P - f_Q$, we have $f = 2t + (f_P + f_Q)/2 = 2t + n - 4$. From Euler's formula in three-dimensional space we know that $n - e + f - t = 2$. Thus, $t = e - 2n + 6$. Since $e_P = 3n_P - 6$ and $e_Q = 3n_Q - 6$, we have $t = e_i + n - 6$. □

From Lemma 4.9 we know that the number of tetrahedra in a tetrahedralization of $P - Q$ is linear if and only if the number of interior edges is linear. We will distribute the edges added in the algorithm to the vertices of P and Q. The number of edges assigned to a vertex v is called the *weight* of v, denoted as $w(v)$. Initially, let $w(v) = 0$ for $v \in V(P) \cup V(Q)$. We will assign a weight to each vertex such that the summation of $w(v)$ for $v \in V(P) \cup V(Q)$ is equal to the total number of edges added in the algorithm.

The process of assigning a weight to each vertex is called the *weighting process.* The following lemmas describe the weighting process such that the weight of each vertex involved is bounded by a constant.

Lemma 4.10 *For Step 1 of the* BIGCAP-REMOVE *algorithm, there exists a weighting process such that the weight of the tip of each cap is less than or equal to 2.*

Proof In Step 1 of the algorithm, we have the following three cases:

1. If a vertex v of degree 3 is removed, no edge is added to $P - Q$. Thus $w(v) = 0$.

2. If a vertex v of degree 4 is removed, one edge is added to $P - Q$ to construct $\overline{P}$. We set $w(v) \leftarrow w(v) + 1$. Thus $w(v) = 1$.

3. If a vertex v of degree 5 is removed, two edges are added to $P - Q$ to construct $\overline{P}$. We set $w(v) \leftarrow w(v) + 2$. Thus $w(v) = 2$.

Therefore, the weight of the tip of each cap removed in Step 1 is less than or equal to 2. □

Lemma 4.11 *Let S_3 be a set of vertices involved in Step 3 of the* BIGCAP-REMOVE *algorithm. There exists a weighting process such that $w(v)$, $v \in S_3$, increases by at most 4.*

Proof In Step 2, the vertices of V_a^P partition the polygons of P_a into a set of chains C_a. Consider a chain $x_1x_2...x_k$ in C_a. Since the polygon in P_a which contains $x_1x_2...x_k$ is the smallest polygon surrounding a portion of Q_a, denoted as $Q(x_1...x_k)$, there are two patterns regarding the connection between $Q(x_1...x_k)$ and this chain: (1) no vertex of x_i $(2 \leq i \leq k-1)$ is adjacent to any vertex of $Q(x_1...x_k)$ (called

pattern-I, see Figure 4.9(a)), and (2) each vertex x_i $(1 \leq i \leq k)$ is adjacent to the same vertex $q^* \in V(Q(x_1...x_k))$ (called pattern-II, see Figure 4.9(b)). For each vertex x_i $(2 \leq i \leq k-1)$ of the chain which is selected as the tip of $S_a(x_i)$-bigcap(x_i) in Step 3b, let polygon P_{x_i} be the boundary of bottom(x_i) and V_{x_i} be the vertex set of $S_a(x_i)$-bigcap(x_i). Because we add an edge $x_i v$ for each vertex $v \in V_{x_i} - \{x_i\}$ in Step 3c, we set $w(v) \leftarrow w(v) + 1$ for $v \in V_{x_i} - \{x_i\}$. Consider the following two cases regarding bottom(x_i).

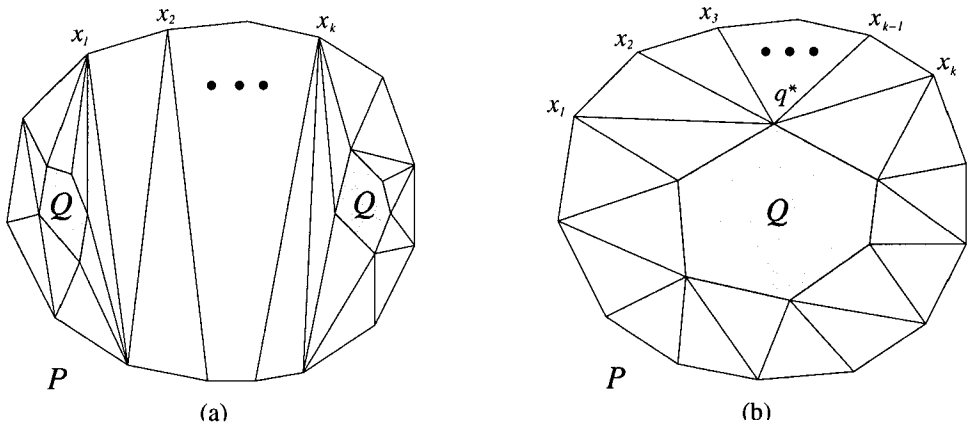

Figure 4.9: Pattern-I is illustrated in (a) and pattern-II is illustrated in (b). The outer boundary lies on the surface of P and the shaded areas lie on the surface of Q.

- If bottom(x_i) does not contain any vertex of Q in its interior, then the triangulation surface of polygon P_{x_i} has $|V(P_{x_i})| - 3$ interior edges. So in pattern-I, for each vertex $p \in V(P_{x_i})$ except three vertices, set $w(p) \leftarrow w(p) + 1$; in pattern-II, for each vertex $p \in V(P_{x_i})$ except vertex q^* and two other vertices, set $w(p) \leftarrow w(p) + 1$.

- If there is a set of vertices and edges Q_{x_i} on the surface of Q which appear

in the interior of bottom(x_i), from Lemma 4.8 we know that the number of the interior edges on the triangulation surface between P_{x_i} and Q_{x_i} is less than or equal to $|V(P_{x_i})| + 3|V(Q_{x_i})| - 3$. So for each vertex y of $V(P_{x_i})$ (except three vertices in pattern-I, or vertex q^* and two other vertices in pattern-II), set $w(y) \leftarrow w(y) + 1$; and for each vertex y of $V(Q_{x_i})$, set $w(y) \leftarrow w(y) + 3$.

Now let us consider removing each S-bigcap with tip in V_a^P. For each vertex $x \in V_a^P$, there is a chain $q_1 q_2 ... q_k$, $q_i \in V(Q_a)$ $(1 \leq i \leq k)$, such that each q_i is adjacent to x (refer to Figure 4.5). Let polygon P_x be the boundary of bottom(x) and V_x be the vertex set of $S_a(x)$-bigcap(x). Because we add an edge xv for each vertex $v \in V_x - \{x\}$ in Step 3c, we set $w(v) \leftarrow w(v) + 1$ for $v \in V_x - \{x\}$. Consider the following two cases regarding bottom(x).

- If bottom(x) does not contain any vertex of Q in its interior, then the triangulation surface of polygon P_x has $|V(P_x)| - 3$ interior edges. So for each vertex $y \in V(P_x)$ except three vertices, set $w(y) \leftarrow w(y) + 1$.

- If there is a set of vertices and edges of Q which appear in the interior of bottom(x), similarly to the above case, set $w(y) \leftarrow w(y) + 1$ for each vertex $y \in V(P_x)$ except three vertices, and set $w(y) \leftarrow w(y) + 3$ for each vertex y of Q in the interior of bottom(x).

We now analyze how $w(v)$, $v \in S_3$ increases. There are five cases depending upon the locations of vertices

1. $v \in V(P)$ is an interior vertex of a dome(x), $x \in V(P_a)$. We add an edge between x and v if $xv \notin E(\overline{P})$. So $w(v) = 1$.

2. $v \in V(P)$ lies on the boundary of bottom(v_i), $v_i \in V(P_a)$, $i = 1, 2, ..., m$ (refer to Step 3). Since $v_1, v_2, ..., v_m$ is an optimal solution in Step 3a, vertex v cannot be shared by more than 2 boundaries of the bottoms of v_i $(1 \leq i \leq m)$. Thus, $w(v)$ increases by at most 4.

3. $v \in V(P_a)$ is the tip of an $S_a(v)$-bigcap(v) removed in Step 3b. $w(v)$ does not increase when $S_a(v)$-bigcap(v) is removed.

4. $v \in V(Q)$ does not lie on Q_a. v must be in the interior of a bottom(v_i) $(1 \leq i \leq m)$. So $w(v)$ increases by at most 4.

5. v belongs to $V(Q_a)$. From the above weighting process, we know that $w(v)$ increases by at most 2.

Therefore, the weight of $w(v)$, $v \in S_3$, increases by at most 4 in Step 3. □

Lemma 4.12 *Let S_6 be a set of vertices involved in Step 6b of the* BIGCAP-REMOVE *algorithm. There exists a weighting process such that $w(v)$, $v \in S_6$, increases by at most 4.*

Proof In Step 6, consider the chain $y_1y_2...y_k$ in P_b. Let y_l $(i < l < j)$ be the vertex in Step 6b such that $\{y_{i+1}, ..., y_{j-1}\}$ is contained in the interior of dome(y_l). Let polygon P_{y_l} be the boundary of bottom(y_l) and V_{y_l} be the vertex set of S_b-bigcap(y_l). Similar to the proof of Lemma 4.11, we set $w(v) \leftarrow w(v) + 1$ for $v \in V_{y_l} - \{y_l\}$. Consider the following two cases regarding bottom(y_l).

- If bottom(y_l) does not contain any vertex of Q in its interior, then the triangulation surface of polygon P_{y_l} has $|V(P_{y_l})| - 3$ interior edges. So for each vertex $y \in V(P_{y_l})$ except y_i, y_j and one other vertex, set $w(y) \leftarrow w(y) + 1$.

- If there is a set of vertices and edges of Q which appear in the interior of bottom(y_l), similarly to the proof of Lemma 4.11, set $w(y) \leftarrow w(y) + 1$ for each vertex $y \in V(P_{y_l})$ except y_i, y_j and one other vertex, and set $w(y) \leftarrow w(y) + 3$ for each vertex y of Q in the interior of bottom(y_l).

Now let us analyze how $w(v)$, $v \in S_6$ increases. There are four cases depending upon the locations of vertices.

1. $v \in V(P)$ is an interior vertex of a dome(y), $y \in V(P_b)$. We add an edge between y and v if $yv \notin E(\overline{P})$. So $w(v) = 1$.

2. $v \in V(P)$ lies on the boundary of bottom(y), $y \in V(P_b) - V(P'_a)$ (refer to Step 6b). Since we select chord y_iy_j $(1 \leq i < j \leq k)$ in the decreasing partial order, the bottom of the removed S-bigcap is as "big" as possible. So vertex v

cannot be shared by more than 2 boundaries of the bottoms of the tip vertices. Thus, $w(v)$ increases by at most 4.

3. $v \in V(P_b)$ is the tip of an S-bigcap removed in Step 6b. $w(v)$ does not increase when the S-bigcap is removed.

4. v is in $V(Q)$. v must be in the interior of the bottom of a vertex in P_b. So $w(v)$ increases by at most 4.

□

From the above Lemmas, we can derive the main result of this chapter.

Theorem 4.2 *Let P and Q be two nested convex polyhedra such that Q is contained in P. The* BIGCAP-REMOVE *algorithm produces at most $9n - 6$ tetrahedra for the region $P - Q$.*

Proof For any $v \in V(P)$ removed in Step 1, from Lemma 4.10 we know that $w(v) \leq 2$. For any remaining $v \in V(P)$, there are two cases:

- If v never appears in $V(P_a)$, then from Lemma 4.11 or Lemma 4.12 we know that $w(v) \leq 4$.

- If v appears in $V(P_a)$ or $V(P_b)$, then from Lemma 4.11 or Lemma 4.12 we know that $w(v)$ increases by at most 4 when v becomes a vertex in P_a or P_b. Since v

will be removed after P_a or P_b are removed, we know that $w(v)$ increases by at most 4 when v is removed. Thus, we have $w(v) \leq 8$.

For any $v \in V(Q)$, there are also two cases:

- If v never appears in $V(Q_a)$, then from Lemma 4.11 or Lemma 4.12 we know that $w(v) \leq 4$.
- If v appears in $V(Q_a)$, then from Lemma 4.11 or Lemma 4.12 we know that $w(v)$ increases by at most 4 when v becomes a vertex in Q_a. From the algorithm we know that v cannot stay in $V(Q_a)$ after P_a and P_b are removed. Thus we have $w(v) \leq 8$.

In summary, for any vertex of P and Q, the upper bound of its weight is 8. Since the number of the edges added in the algorithm is equal to the summation of $w(v)$ for $v \in V(P) \cup V(Q)$, we know that the number of the edges added in the algorithm is less than $8n$. Notice that the number of the interior edges in our tetrahedralization of $P - Q$ is less than or equal to that of the edges added in the algorithm (equality holds if and only if P and Q are simplicial polyhedra). It follows from Lemma 4.9 that the number of tetrahedra in our tetrahedralization of $P - Q$ is at most $9n - 6$.

□

4.5 Concluding remarks

In this chapter, we have described an algorithm, called BIGCAP-REMOVE, to tetrahedralize the region between two nested convex polyhedra P and Q. This algorithm avoids using Steiner points and can also work for the degenerate case in which more than three vertices of P and Q are coplanar. This algorithm also produces at most $9n - 6$ tetrahedra. Thus we have solved the open problem proposed by Bern [16].

In Step 3a of the algorithm, in order to control the number of edges added when removing S-bigcaps, we find the minimum number of vertices $v_1, v_2, ..., v_m \in V(P_a)$ such that after removing $S_a(v_i)$-bigcap(v_i) in the order $i = 1, 2, ..., m$, the vertices of P_a are removed and the vertices of Q_a^{view} are removed or on P_b. From Lemma 4.5, we know that Step 3a is correct. However, we do not know how to solve this optimization problem in polynomial time. Hence BIGCAP-REMOVE is not a polynomial time algorithm. We conjecture that the region between two nested convex polyhedra can be tetrahedralized with a linear number of tetrahedra in polynomial time.

Chapter 5

A Lower Bound on the value of β for β-Skeletons Belonging to MWTs

The minimum weight triangulation problem is one of the most longstanding open problems in computational geometry. Computing subgraphs of the minimum weight triangulation is one approach to attacking this problem. Keil [62] conjectured that the β-skeleton is a subgraph of the MWT when $\beta = \frac{2}{3}\sqrt{3}$. In Section 5.1, we survey related results in this area. In Section 5.2, we construct a counterexample to Keil's conjecture. In Section 5.3, we prove a new lower bound for the β-skeleton. Finally, we conclude our work in Section 5.4.

5.1 Introduction

Let S be a finite set of points in the Euclidean plane. A *triangulation* of S is a maximum set of non-crossing edges with their endpoints in S. It partitions the interior of the convex hull of S into non-overlapping triangular faces. The weight of a triangulation is the sum of the lengths of its edges. A *minimum weight triangulation*, denoted by MWT, of S is a triangulation that minimizes the weight among all triangulations of S.

Computing an MWT is one of the outstanding open problems listed by Garey and Johnson [51]. The complexity status of this problem has been unknown since it was proposed in 1975 [92]. Works on this problem can be classified into several types: determining the complexity of the MWT problem [74], designing approximation algorithms [56, 72, 73, 81], using the integer programming method [67, 99], designing efficient algorithms for restricted classes of point sets [2, 4, 52, 65, 75, 103], and computing subgraphs of the MWT [11, 45, 62, 100, 111]. Computing subgraphs of the MWT seems to be a promising approach, as the edges which are always in the MWT play a very important role in both exact and approximation algorithms for computing the MWT. If we can compute a subgraph of the MWT that is connected (actually it is sufficient that the subgraph has a constant number of connected components [28]), then the remaining edges could be added by triangulating the resulting polygonal regions using dynamic programming.

Basically, there are two approaches to computing subgraphs of the MWT. The first approach is based on the LMT-skeleton, and was presented independently by Dickerson *et al.* [45] and Belleville *et al.* [11]. Several variants of the LMT-skeleton have been considered recently [3, 10, 29, 44, 55]. In particular, the improved LMT-skeleton heuristic proposed by Beirouti and Snoeyink [10] can compute the exact MWT of tens of thousands of points in minutes.

The second approach was first studied by Gilbert [52], who showed that the shortest edge in S is in MWT(S). Yang *et al.* [111] showed that all mutual nearest-neighbor edges are also in MWT(S). Keil [62] proved that the β-skeleton of S for $\beta = \sqrt{2}$ is a subgraph of MWT(S), where the β-skeleton was introduced by Kirkpatrick and Radke [63] and defined as follows: For $\beta > 1$, the *β-skeleton* of S is a set of edges with endpoints in S and each edge e in the set satisfies the empty-disks condition, i.e., no element in S lies inside the two disks of diameter $\beta|e|$ that pass through both endpoints of e. Yang [109] extended Keil's result to $\beta \approx 1.27905$. Cheng and Xu [30] improved further to $\beta \approx 1.17682$.

The key to the proof that the β-skeleton belongs to the MWT in Keil's pioneering paper is the validity of the following two lemmas, namely the length lemma and the remote length lemma.

Lemma 5.1 *(***Length Lemma [62]** *) Let x and y be the endpoints of an edge in the $\sqrt{2}$-skeleton of a set S of points in the plane. Let p and q be two points in S such that*

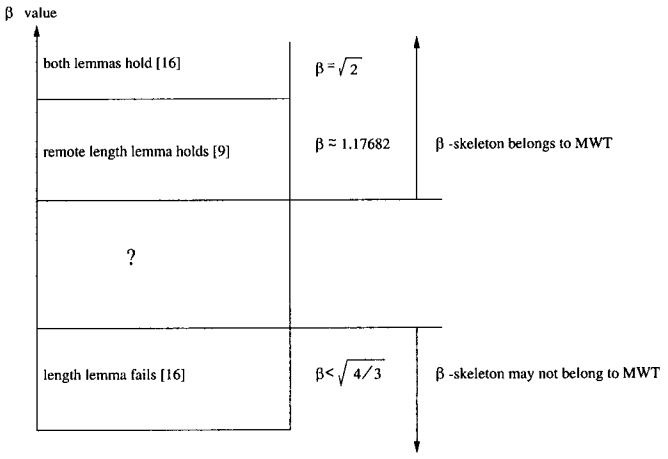

Figure 5.1: An illustration for Keil's conjecture.

the line segment pq intersects the segment xy. Then $|pq|$ *is greater than* $|xy|$, $|xp|$, $|xq|$, $|yp|$, *and* $|yq|$.

Lemma 5.2 *(***Remote Length Lemma [62]** *) Let x and y be the endpoints of an edge in the* $\sqrt{2}$*-skeleton of S, and* p, q, r, *and s be four other distinct points in S with p and r lying on one side and q and s on the other side of the line through xy. Assume pq and rs intersect xy, and pq does not intersect rs. Then, either* $|\ pq\ | > |\ pr\ |$ *or* $|\ rs\ | > |\ pr\ |$.

Keil mentioned that the length lemma holds for $\beta \geq \frac{2}{3}\sqrt{3}$. For $\beta < \frac{2}{3}\sqrt{3}$, there exists a four-point counterexample for the length lemma. Thus, for $\beta < \frac{2}{3}\sqrt{3}$, the β-skeleton may not belong to the MWT. Keil conjectured that the β-skeleton is a subgraph of MWT for $\beta = \frac{2}{3}\sqrt{3}$ (≈ 1.15470). Cheng and Xu [30] proved that the remote length lemma is still true for $\beta = \frac{1}{3}\sqrt{2\sqrt{3}+9}$ (≈ 1.17682). This is very close to the bound conjectured by Keil. Recently, Aichholzer *et al.* [3] proved

that, for $\beta > \frac{2}{3}\sqrt{3}$, the β-skeleton of convex polygons and a certain class of star-shaped polygons is a subgraph of the MWT. The situation to date is summarized in Figure 5.1.

In this chapter, we disprove Keil's conjecture by presenting a new lower bound on the value of β (i.e., $\beta = \frac{1}{6}\sqrt{2\sqrt{3}+45} \approx 1.16027$) such that if $\beta < \frac{1}{6}\sqrt{2\sqrt{3}+45}$, the β-skeleton is not always a subgraph of the MWT.

5.2 A counterexample of Keil's conjecture

In this section, we construct a set of points whose β-skeleton does not belong to its MWT when $\frac{2}{3}\sqrt{3} \leq \beta < \frac{1}{6}\sqrt{2\sqrt{3}+45}$. Thus, this is not only a counterexample of Keil's conjecture ($\beta = \frac{2}{3}\sqrt{3}$), but can also be used to prove the new lower bound ($\beta = \frac{1}{6}\sqrt{2\sqrt{3}+45}$).

Throughout this chapter, let α be the angle that the chord xy ($x, y \in S$) subtends at one of the circles (refer to Figure 5.2). It follows from the definition of the β-skeleton that $\beta = 1/\sin\alpha$. Thus, the definition of a β-skeleton can be rewritten as follows: xy ($x, y \in S$) is an edge in the β-skeleton ($\beta > 1$) of S if and only if there does not exist a point $z \in S$ such that $\angle xzy > \arcsin(1/\beta)$. Notice that the β-skeleton defined in this thesis is a superset of that used in [62] (refer to Lemma 1 of [62]). When $\beta = \frac{2}{3}\sqrt{3}$, we have $\alpha = \pi/3$. Thus, the length lemma is violated for $\alpha > \pi/3$

[62].

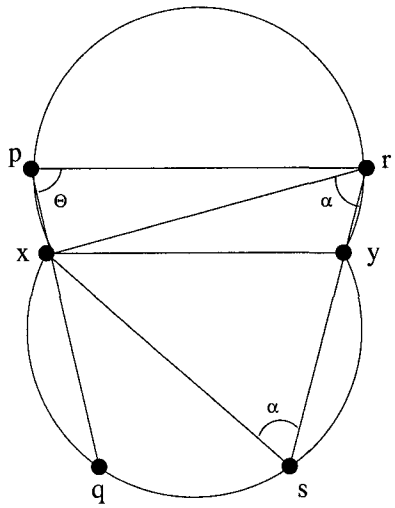

Figure 5.2: Some relationships among the edges and angles.

In [30], Cheng and Xu proved the following property that describes a critical structure. From this structure, they derived an improved version of the remote length lemma for $\beta = \frac{1}{3}\sqrt{2\sqrt{3}+9}$. Thus, by using the same proof strategy in [62], they proved that the β-skeleton is a subgraph of the MWT for any $\beta > \frac{1}{3}\sqrt{2\sqrt{3}+9}$.

Lemma 5.3 *([30]) When $\beta = \frac{1}{3}\sqrt{2\sqrt{3}+9}$, there exists a point set S such that xy, $x, y \in S$, is an edge in the β-skeleton of S and $p, q, r, s \in S$ are other distinct points satisfying the following conditions (refer to Figure 5.2):*

(i) $\angle xpy = \angle xqy = \angle xry = \angle xsy = \arcsin(1/\beta)$, $x \in pq$, $y \in rs$, $pq \cap rs = \emptyset$, and p and r lie on the same side of the line through xy;

(ii) $|pq| = |pr| = |rs|$; and

(iii) $pqsr$ is a regular trapezoid such that $\angle xpr$ and $\angle yrp$ are acute.

In the remainder of this chapter, let $\alpha = \angle xpy$ and $\theta = \angle xpr$. From Lemma 5.3, we derive the following equations that can be used in our proofs:

$$\sin\alpha = \frac{3}{\sqrt{2\sqrt{3}+9}}, \quad \cos\alpha = \frac{\sqrt{2\sqrt{3}}}{\sqrt{2\sqrt{3}+9}},$$

$$\sin\theta = \frac{\sqrt{2\sqrt{3}}}{2}, \quad \cos\theta = \frac{\sqrt{3}-1}{2},$$

$$\sin 2\theta = \frac{\sqrt{2\sqrt{3}}(\sqrt{3}-1)}{2}, \quad \cos 2\theta = 1-\sqrt{3}.$$

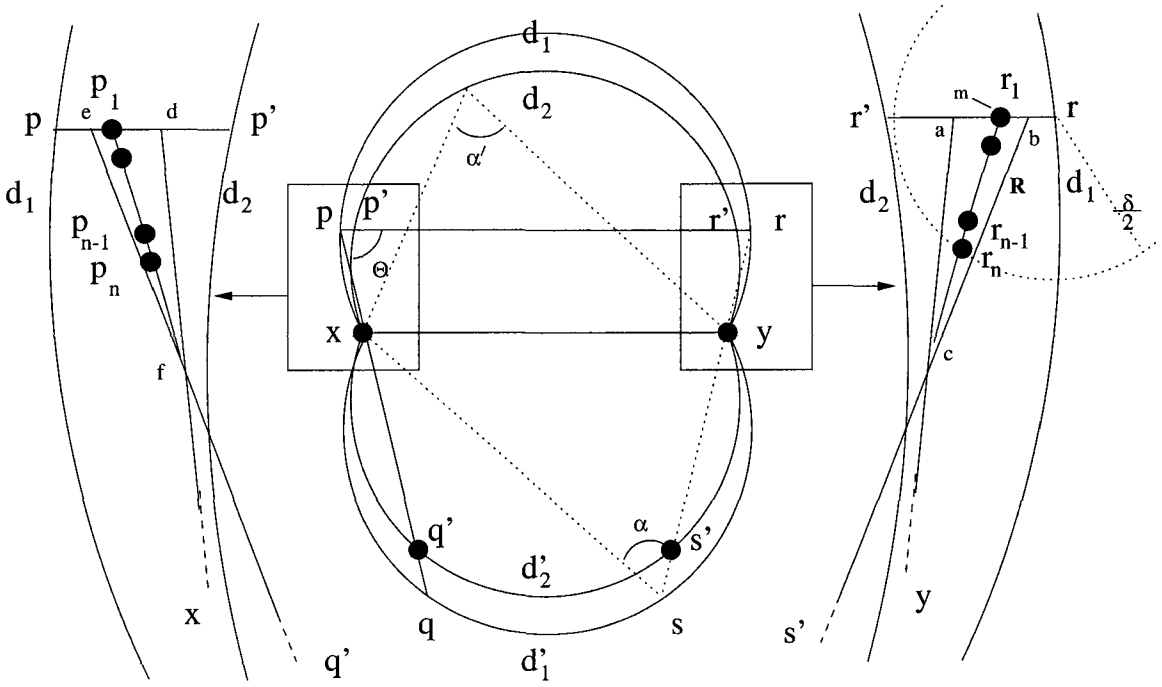

Figure 5.3: The arrangement of sites (darkened dots).

Now, we describe the arrangement of a set of $2n+4$ sites (the value of n will be determined later) such that the β-skeleton is not included in the MWT for $\beta \in$

$[\ \frac{2}{3}\sqrt{3},\ \frac{1}{6}\sqrt{2\sqrt{3}+45}\ )$.

Let x and y be two different points in the plane and draw two circles, d_1 and d_1', with the same diameter $\frac{1}{3}\sqrt{2\sqrt{3}+9}|xy|$, such that both circles pass through x and y (refer to Figure 5.3). The union of the regions bounded by d_1 and d_1' is the forbidden neighbourhood with $\beta = \frac{1}{3}\sqrt{2\sqrt{3}+9}$ for points x and y [62]. Denote the two big arcs with endpoints x and y in d_1 and d_1' by $arc(xd_1y)$ and $arc(xd_1'y)$, respectively. Similarly, for any fixed $\beta \in [\ \frac{2}{3}\sqrt{3},\ \frac{1}{6}\sqrt{2\sqrt{3}+45}\ )$, draw two circles, d_2 and d_2', with the same diameter $\beta|xy|$ that pass through x and y. Denote the two big arcs with endpoints x and y in d_2 and d_2' by $arc(xd_2y)$ and $arc(xd_2'y)$, and let the centers of d_1 and d_2 lie on the same side of the line through xy. Since $\frac{1}{6}\sqrt{2\sqrt{3}+45} < \frac{1}{3}\sqrt{2\sqrt{3}+9}$, the region bounded by $arc(xd_2y) \cup arc(xd_2'y)$ is included in that bounded by $arc(xd_1y) \cup arc(xd_1'y)$. Let points $p, r \in arc(xd_1y)$, $q, s \in arc(xd_1'y)$ such that the four points satisfy the three conditions of Lemma 5.3. Let pq intersect $arc(xd_2'y)$ at q', rs intersect $arc(xd_2'y)$ at s', and pr intersect $arc(xd_2y)$ at p' and r', where p' is nearer to p than to r.

From Lemma 5.3, we have

$$|\ pq'\ | = |\ rs'\ | < |\ pr\ |\ .$$

Now, we choose a pair of points a and b on $r'r$ which are very close to point r such

that the triangle $\triangle abc$ lies outside d_2 and inside d_1, where c is the intersection point of $s'b$ and ya. Let m be a point on ab very near to b, and let $\widehat{cm}$ be an arc of a circle whose center is at the far right-hand side such that any line tangential to the circle at the arc $\widehat{cm}$ will intersect ray $\vec{cs}$ and the arc $\widehat{cm}$ cuts the acute angle between the line segments $s'c$ and yc (this ensures that $\widehat{cm}$ will be 'convex' facing s', q', x, p_i and 'concave' facing y). We now arrange a site on points x, y, s', and q', respectively. We also arrange n sites ($R = \{r_1, r_2, ..., r_n\}$) evenly on the arc $\widehat{cm}$ such that the maximum distance from $r_i \in R$ to r is $\frac{\delta}{2}$. Symmetrically, we arrange n sites ($P = \{p_1, p_2, ..., p_n\}$) near p in the same manner as those near r. The value of δ is determined as follows: let $\epsilon = |ss'|$, ß= $|pr| - \max\{|p_i y| \mid 1 \leq i \leq n\}$, æ= $|pr| + |py| - 2|pq'|$, and then, we set $\delta = \frac{1}{2}\min\{\epsilon, ß, æ\}$. We need to guarantee that δ is positive so that the above arrangement is significant. We show how this is achieved in the following lemma:

Lemma 5.4 *For* $\beta \in [\ \frac{2}{3}\sqrt{3},\ \frac{1}{6}\sqrt{2\sqrt{3}+45}\ )$, $\delta > 0$.

Proof Since $\frac{1}{6}\sqrt{2\sqrt{3}+45} < \frac{1}{3}\sqrt{2\sqrt{3}+9}$, d_2' is strictly included in d_1'. Thus, $\epsilon = |ss'| > 0$.

Now we prove that ß > 0. Consider the triangle $\triangle ypq$. Since $\angle xpy = \angle xqy < \pi/3$, we have $|py| < |pq|$. Thus, $|py| < |pr|$. It follows from the arrangement that $\angle p_1 p_i y > \pi/2$, $i = 2, ..., n$. So $|p_i y| < |p_1 y| < |py|$. Hence, $|p_i y| < |pr|$, $i = 1, ..., n$. It follows that ß > 0.

Finally, we prove that æ > 0. Refer to Figure 5.4. Let $\alpha' = \angle xq'y$. In triangles

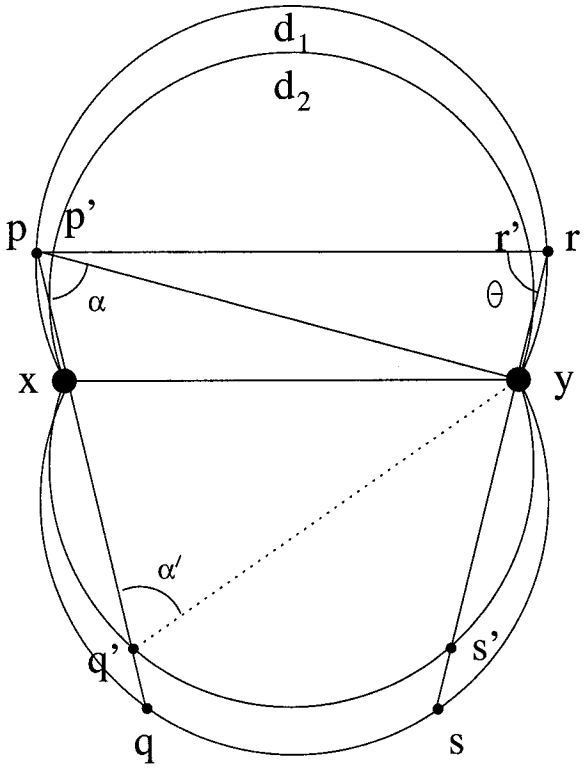

Figure 5.4: For the proof of Lemma 5.4.

$\triangle pry$ and $\triangle pq'y$, by the law of sines we have

$$\frac{|\, pr \,|}{\sin(\pi - \theta - (\theta - \alpha))} = \frac{|\, py \,|}{\sin\theta}, \tag{5.1}$$

$$\frac{|\, pq' \,|}{\sin(\pi - \alpha' - \alpha)} = \frac{|\, py \,|}{\sin\alpha'}. \tag{5.2}$$

Then, by (5.1) we have

$$| pr | = \frac{\sin(2\theta - \alpha)}{\sin\theta} | py | = \frac{2\sqrt{2\sqrt{3}}}{\sqrt{2\sqrt{3}+9}} | py | .$$

By (5.2), we have

$$\begin{aligned}
2 | pq' | &= 2\frac{\sin(\alpha' + \alpha)}{\sin\alpha'} | py | \\
&= (2\cos\alpha + 2\frac{\cos\alpha'}{\sin\alpha'}\sin\alpha) | py | \\
&= (2\cos\alpha + 2\sqrt{(\frac{1}{\sin\alpha'})^2 - 1} * \sin\alpha) | py | \\
&= (2\cos\alpha + 2\sqrt{(\beta)^2 - 1} * \sin\alpha) | py | \\
&< (\frac{2\sqrt{2\sqrt{3}}}{\sqrt{2\sqrt{3}+9}} + 2\sqrt{(\frac{\sqrt{2\sqrt{3}+45}}{6})^2 - 1} * \frac{3}{\sqrt{2\sqrt{3}+9}}) | py | \\
&= (\frac{2\sqrt{2\sqrt{3}}}{\sqrt{2\sqrt{3}+9}} + 2\sqrt{\frac{2\sqrt{3}+9}{36}} * \frac{3}{\sqrt{2\sqrt{3}+9}}) | py | \\
&= (\frac{2\sqrt{2\sqrt{3}}}{\sqrt{2\sqrt{3}+9}} + 1) | py | \\
&= | pr | + | py | .
\end{aligned}$$

Thus, æ=$| pr | + | py | - 2 | pq' | > 0$. Therefore, δ is positive. □

By the above arrangement, we obtain a point set $\mathcal{S} = \{q', x, p_n, p_{n-1}, ..., p_1, r_1, r_2, ..., r_n, y, s'\}$, which can be used to prove the new lower bound. Before we prove this in the next section, we will summarize some properties of the set $\mathcal{S}$ in the following

lemma.

Lemma 5.5 *Let* $\mathcal{S} = \{q', x, p_n, p_{n-1}, ..., p_1, r_1, r_2, ..., r_n, y, s'\}$ *be the point set described above. Then,*

(i) the line segments $s'r_i$ $(i = 1, ..., n)$ *and* yr_i $(i = 1, ..., n-1)$ *do not cross each other in their interiors, and neither do the line segments* $q'p_i$ $(i = 1, ..., n)$ *and* xp_i $(i = 1, ..., n-1)$.

(ii)

$$| rs' | > | r_1s' | > | r_2s' | > \cdots > | r_ns' |;$$

$$| pq' | > | p_1q' | > | p_2q' | > \cdots > | p_nq' | .$$

(iii)

$$| r_is' | = | p_iq' | < | p_ir_i |, i = 1, ..., n.$$

(iv)

$$| p_iy | < | p_ir_i |, i = 1, ..., n.$$

(v)

$$| p_1r_i | > | p_nr_n |, | p_iy | \geq | p_ny |, i = 1, ..., n.$$

Proof (i). Since the arc $\widehat{cm}$ is 'convex' with respect to s' and 'concave' with respect to y, the line segments $s'r_i$ $(i = 1, ..., n)$ and yr_i $(i = 1, ..., n-1)$ do not cross each

other in their interiors. Symmetrically, $q'p_i$ and xp_i also do not cross.

(ii). Since all the sites of R lie on arc $\widehat{cm}$ which is almost a straight line and $\angle r_i y s' > \pi/2$, we have $\angle s' r_1 r > \pi/2$, and $\angle s' r_i r_{i-1} > \pi/2$, $i = 2, ..., n$. Thus, $| rs' | > | r_1 s' | > | r_2 s' | > \cdots > | r_n s' |$. Symmetrically, the other inequalities also hold.

(iii). Because the maximum distance from p_i to p and from r_i to r is $\delta/2$, we have $| pr | < | p_i r_i | + \delta$. From (ii), we know that $| r_i s' | < | rs' |$. Since $| rs' | = | rs | - \epsilon = | pr | - \epsilon$, we have

$$\begin{aligned} | r_i s' | &< | pr | - \epsilon \\ &< | p_i r_i | + \delta - \epsilon \\ &< | p_i r_i | \quad (\text{since } \epsilon \geq 2\delta). \end{aligned}$$

(iv). Since $| pr | - | p_i y | \geq ß$, we have

$$\begin{aligned} | p_i y | &\leq | pr | - ß \\ &< | p_i r_i | + \delta - ß \\ &< | p_i r_i | \quad (\text{since } ß \geq 2\delta). \end{aligned}$$

(v). In triangle $\triangle p_1 p_n r_n$, $\angle p_1 p_n r_n$ is obtuse. Then, $\mid p_1 r_n \mid > \mid p_n r_n \mid$. In triangle $\triangle p_1 p_i r_i$, $i = 2, ..., n$, $\angle p_1 p_i r_i$ is obtuse. Thus, $\mid p_1 r_i \mid > \mid p_i r_i \mid$. Since $p_i p_n r_n r_i$ is a regular trapezoid and $\angle p_i p_n r_n$ is obtuse, we have $\mid p_i r_i \mid > \mid p_n r_n \mid$. Therefore, $\mid p_1 r_i \mid > \mid p_n r_n \mid$. Similarly, in triangle $\triangle p_i p_{i+1} y$, $i = 1, ..., n-1$, $\angle p_i p_{i+1} y$ is obtuse. Then, $\mid p_i y \mid > \mid p_{i+1} y \mid$, $i = 1, ..., n-1$. Hence, $\mid p_i y \mid > \mid p_n y \mid$. □

5.3 The proof of the lower bound

Let $\mathcal{S} = \{q', x, p_n, p_{n-1}, ..., p_1, r_1, r_2, ..., r_n, y, s'\}$ be the point set constructed in the previous section. In this section, we prove a new lower bound by showing that, for $\beta \in [\ \frac{2}{3}\sqrt{3},\ \frac{1}{6}\sqrt{2\sqrt{3}+45}\ )$, the line segment xy belongs to the β-skeleton of $\mathcal{S}$, but does not belong to MWT($\mathcal{S}$).

All the triangulations of $\mathcal{S}$ can be divided into two groups related to xy: if a triangulation contains edge xy, then it belongs to the first group; otherwise, it belongs to the second group. Let T_{xy} denote the MWT over the first group and T denote the MWT over the second group. The relationship between T_{xy} and T can be stated as follows:

Lemma 5.6 *T_{xy} and T differ only in the edges in the interior of the simple polygon $L = q' x p_n p_{n-1} ... p_1 r_1 r_2 ... r_n y s'$.*

Proof The convex hull of S includes sites q', x, p_1, r_1, y, s'. Thus, edges $q'x, xp_1, p_1 r_1$,

r_1y, ys', $s'q'$ belong to any triangulations of $\mathcal{S}$. Lemma 5.5 (i) implies that no edge with endpoints in $\mathcal{S}$ intersects the interior of the edges xp_2, xp_3, ..., xp_n, yr_2, yr_3, ..., yr_n, p_np_{n-1}, $p_{n-1}p_{n-2}$, ..., p_2p_1, r_nr_{n-1}, $r_{n-1}r_{n-2}$, ..., r_2r_1. Thus, they belong to any triangulations of $\mathcal{S}$. Therefore, the difference between T_{xy} and T is the internal edges bounded by the polygon Ł= $q'xp_np_{n-1}...p_1r_1r_2...r_nys'$. □

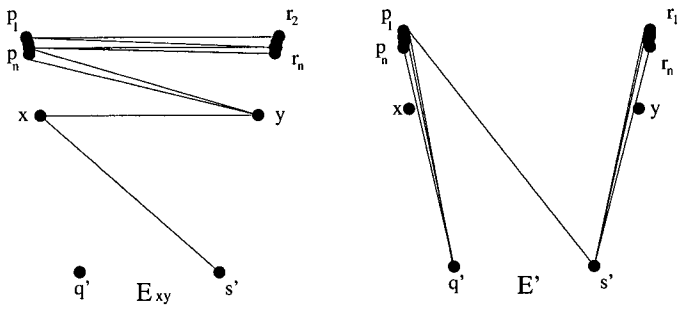

Figure 5.5: An illustration for the edge sets E_{xy} (left) and E' (right).

For an edge set F, let $\omega(F)$ denote the sum of the lengths of all the edges in F. Let E_{xy} be a subset of T_{xy}, which consists of all the edges of T_{xy} in the interior of the polygon Ł. Then we have the following property:

Lemma 5.7 $\omega(E_{xy}) > n(|\ p_nr_n\ | + |\ p_ny\ |) - |\ p_1r_1\ | + |\ xy\ | + |\ xs'\ |$.

Proof In quadrilateral $xys'q'$, since $xs' = yq'$, xs' or yq' belongs to E_{xy}. Without loss of generality, we can suppose that $xs' \in E_{xy}$ (refer to Figure 5.5(left)). Let $E'_{xy} = (E_{xy} - \{xy, xs'\}) \cup \{p_1r_1\}$. Since $E'_{xy} - \{p_1r_1\}$ consists of the $2n-1$ diagonals in the triangulation of the polygon $p_1...p_nxyr_n...r_1$, there are $2n$ edges in E'_{xy}. Since there are at most n edges with endpoint y, we can group these $2n$ edges in E'_{xy} into n pairs of edges such that for each pair of edges, there exists at most one edge with

endpoint y. Thus, the n pairs of edges can be classified into two sets, denoted by $C_1 = \{(p_u y, p_v r_w) \mid 1 \leq u, v, w \leq n\}$, and $C_2 = \{(p_u r_w, p_v r_x) \mid 1 \leq u, v, w, x \leq n\}$. Consider each edge $p_u y$. We have $\mid p_u y \mid \geq \mid p_n y \mid$ by Lemma 5.5(v). Consider each edge $p_u r_w$. If $u < w$, then $\angle p_u p_w r_w$ is obtuse. Thus, $\mid p_u r_w \mid > \mid p_w r_w \mid$. Furthermore, we have $\mid p_u r_w \mid > \mid p_w y \mid$ by Lemma 5.5(iv). Since $p_w p_n r_n r_w$ is a regular trapezoid and $\angle p_w p_n r_n$ is obtuse, we have $\mid p_w r_w \mid > \mid p_n r_n \mid$. Thus $\mid p_u r_w \mid > \mid p_n r_n \mid$. If $u > w$, we can prove that $\mid p_u r_w \mid > \mid p_u y \mid$ and $\mid p_u r_w \mid > \mid p_n r_n \mid$ by using the similar argument. If $u = w$, it is obvious that $\mid p_u r_w \mid \geq \mid p_n r_n \mid$. Therefore, for each pair $(p_u y, p_v r_w) \in C_1$, we have

$$\mid p_v r_w \mid + \mid p_u y \mid \geq \mid p_n r_n \mid + \mid p_n y \mid,$$

where the equality holds if and only if $u = v = w = n$. For each pair $(p_u r_w, p_v r_x) \in C_2$, we have

$$\mid p_u r_w \mid + \mid p_v r_x \mid > \mid p_n r_n \mid + \mid p_n y \mid .$$

Since there are n pairs in C_1 and C_2, we have

$$\omega(E'_{xy}) > n(\mid p_n r_n \mid + \mid p_n y \mid).$$

Hence,

$$\omega(E_{xy}) > n(\mid p_n r_n \mid + \mid p_n y \mid) - \mid p_1 r_1 \mid + \mid xy \mid + \mid xs' \mid .$$

□

Lemma 5.8 *Suppose* $\beta \in [\ \frac{2}{3}\sqrt{3},\ \frac{1}{6}\sqrt{2\sqrt{3}+45}\ )$. *If* $n > \frac{2}{2æ-3\delta}(\mid p_1r_1 \mid - \mid xy \mid + \mid p_1s' \mid - \mid xs' \mid)$ *then* $\omega(T_{xy}) > \omega(T)$.

Proof Let E be a subset of T, which consists of all the edges of T in the interior of the polygon Ł. Let $E' = \{q'p_1, q'p_2, ..., q'p_n, s'r_1, s'r_2, ..., s'r_n, p_1s'\}$ be a triangulation of Ł(see Figure 5.5(right)). Since $xy \notin E'$, $\omega(E) \leq \omega(E')$ by the definition of T. Thus,

$$\omega(E) \leq \omega(E') = 2\sum_{i=1}^{n} \mid p_iq' \mid + \mid p_1s' \mid .$$

From the structure of $\mathcal{S}$, we know that $\mid p_nr_n \mid > \mid pr \mid -\delta$ and $\mid p_ny \mid > \mid py \mid -\frac{\delta}{2}$. Since æ$= \mid pr \mid + \mid py \mid -2 \mid pq' \mid$, we have

$$\mid p_nr_n \mid + \mid p_ny \mid > \mid pr \mid + \mid py \mid -\frac{3\delta}{2} = 2 \mid pq' \mid + æ - \frac{3\delta}{2}.$$

By the definition of δ, we know that æ $\geq 2\delta$. Thus, æ$-\frac{3\delta}{2} > 0$. Let $\epsilon' =$ æ $- \frac{3\delta}{2}$. Hence,

$$\mid p_nr_n \mid + \mid p_ny \mid > 2 \mid pq' \mid + \epsilon',$$

By Lemma 5.7, we have

$$\begin{aligned}
\omega(E_{xy}) &> n(\mid p_n r_n \mid + \mid p_n y \mid) + \mid xy \mid - \mid p_1 r_1 \mid + \mid xs' \mid \\
&> n(2 \mid pq' \mid + \epsilon') + \mid xy \mid - \mid p_1 r_1 \mid + \mid xs' \mid \\
&> 2\sum_{i=1}^{n} \mid p_i q' \mid + n\epsilon' + \mid xy \mid - \mid p_1 r_1 \mid + \mid xs' \mid .
\end{aligned}$$

Thus, from the condition of the lemma, we have

$$\omega(E_{xy}) - \omega(E) > n\epsilon' + \mid xy \mid - \mid p_1 r_1 \mid + \mid xs' \mid - \mid p_1 s' \mid > 0.$$

Therefore, $\omega(T_{xy}) > \omega(T)$. □

From the above lemmas, we can prove the main result:

Theorem 5.1 $\beta = \frac{1}{6}\sqrt{2\sqrt{3}+45}$ *is a lower bound for the* β*-skeleton belonging to MWTs.*

5.4 Concluding remarks

In [62], Keil conjectured that the β-skeleton is a subgraph of the MWT when $\beta = \frac{2}{3}\sqrt{3}$. In [30], Cheng and Xu proved that $\beta = \frac{1}{3}\sqrt{2\sqrt{3}+9}$ is an upper bound on the value of β of a β-skeleton belonging to MWTs of a planar point set. In this chapter, we proved that $\beta = \frac{1}{6}\sqrt{2\sqrt{3}+45}$ is a lower bound. Therefore, we have settled Keil's conjecture. However, closing the gap between the upper and lower bounds is still an open problem.

In related work, Aichholer *et al.* [2] discussed light edges. For an edge e joining two points of a planar point set S, if $|e|$ is shorter than all other edges with endpoints in S which intersect e, then e is called *light*; otherwise, e is called *non-light*. In [30, 62, 100, 111], all the edges which have been identified to always be in the MWTs are light edges. Unfortunately, the example presented by Bose *et al.* [21] showed that the graph consisting of all these identified edges is not connected. In order to find a connected subgraph of MWTs, it seems that a future study should concentrate on nontrivial non-light edge identification.

From Lemma 5.8, we know that some $q'p_i$ and $s'r_i$ $(1 \leq i \leq n)$ must be in MWT($\mathcal{S}$). These are non-light edges because their lengths are longer than $|xy|$. It is the author's opinion that the method used in the proofs of this chapter may be useful for finding non-light edges that must be in the MWTs of a planar point set.

Chapter 6

Conclusions and Future Research

Tetrahedralizations and triangulations are fundamental problems in computational geometry [37, 17, 46, 47, 53, 82]. In this thesis, we have investigated four problems regarding tetrahedralizations and triangulations. There are still many issues which need to be studied in this research area, such as Steiner triangulations [18, 19, 49], robust tetrahedralizations [7, 43, 94], graph drawing of optimal triangulations [69, 70, 101, 102, 104], and mesh refinement [31, 85, 87, 96].

The open questions concerning specific aspects of our research have been given at the end of each chapter. We now close this thesis by listing some interesting open problems in the area of tetrahedralization [76, 78, 79].

Flip graph connectivity

In R^3, a strictly convex hexahedron formed from five vertices can be tetrahedralized in two ways: either as a pair of tetrahedra separated by a face, or as three tetrahedra surrounding an interior diagonal (see Figure 1.3). If two (three) adjacent tetrahedra of the tetrahedralization form a strictly convex hexahedron, then a *flip* replaces the tetrahedra by the other possible tetrahedralization of the hexahedron containing three (two) tetrahedra. The flip can be considered to be a face "flip", where one interior face is "flipped" for three interior faces or vice versa.

Let S be a set of points in R^3. The *flip graph* of the tetrahedralizations of S is the graph whose vertices are all the tetrahedralizations of S and whose edges represent flips between them. That is, two vertices (tetrahedralizations) are connected by an edge if they differ by a flip.

Question [48, 60]: *Is the flip graph connected for a set of points in the general position in R^3?*

As usual, the general position means that no three points are collinear and no four points are coplanar. It is even unknown if the flip graph of tetrahedralizations contains an isolated vertex, i.e., whether there exists a tetrahedralization in which no three tetrahedra surround an edge, nor two tetrahedra share a face and form a convex hexahedron.

The following are some partial and related results on this problem. In R^2, the flips correspond to convex quadrilateral diagonal switches. Since every triangulation

of a set of points can be transformed into the Delaunay triangulation by a sequence of edge flips [68], the flip graph of triangulations in R^2 is connected. de Loera *et al.* [40] proved that in R^2 all triangulations of n points have at least $n-3$ flip neighbours and in R^3 all tetrahedralization of n points in the convex position and with no three points collinear have at least $n-4$ flip neighbours. The flip operation can be generalized in higher dimension. In R^6, Santos [89] constructed a triangulation of 324 points (not in general position) which admit no flip operations. In R^{234}, he also constructed a triangulation of 552 points in the convex position but not in the general position, which admit no flip operations. Thus, in R^6 and R^{234} there exist point sets (not in general position) whose flip graphs of triangulations can have isolated vertices.

Hamiltonian tetrahedralizations

Given a tetrahedralization T, the *dual graph* of T has a vertex for each tetrahedron and an edge for each pair of tetrahedron that share a triangle facet.

Question [5]: *Can every convex polyhedron be tetrahedralized such that the dual graph of the tetrahedralization has a Hamiltonian path?*

This is important because a tetrahedralization that has a Hamiltonian path can be pipelined by some graphics rendering engines so that it can be displayed quickly.

In R^2, the dual graph of a triangulation has a vertex for each triangle and an edge for each pair of triangles that share an edge. Arkin *et al.* [5] proved that every

convex polygon can be triangulated such that the dual graph of the triangulation has a Hamiltonian path and that this property does not necessarily hold for nonconvex polygon.

Tetrahedralizations of non-strictly convex polyhedra

A *non-strictly convex polyhedron* is a convex polyhedron with facets of more than three edges. For such polyhedra in some geometric modeling, the non-triangle facets may be triangulated in several different ways. We then need to tetrahedralize this polyhedron such that all the tetrahedra are compatible with the triangulated surface, in the sense that each triangle on the surface is a facet of a tetrahedron.

Question [16]: *How hard is it to decide whether a non-strictly convex polyhedron with a triangulated surface can be tetrahedralized without Steiner points?*

The following are some related results. If a polyhedron is strictly convex, it can be easily tetrahedralized by the "starring" method [16]: selecting a vertex v of the polyhedron, and then form tetrahedra from v and each triangle facet that is not adjacent to v. If a polyhedron is not convex, Ruppert and Seidel [88] proved that finding a tetrahedralization of a nonconvex polyhedron without Steiner points is NP-complete; this problem remains NP-complete even for star-shaped polyhedra.

Bibliography

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, MA, 1974.

[2] O. Aichholzer, F. Aurenhammer, S.-W. Chen, N. Katoh, M. Taschwer, G. Rote, and Y.-F. Xu. Triangulations intersect nicely. *Discrete and Computational Geometry*, 16:339–359, 1996.

[3] O. Aichholzer, F. Aurenhammer, and R. Hainz. New results on MWT subgraphs. *Information Processing Letters*, 69:215–219, 1999.

[4] E. Anagnostou and D. Corneil. Polynomial time instances of the minimum weight triangulation problem. *Computational Geometry: Theory and Applications*, 3:247–259, 1993.

[5] E. M. Arkin, M. Held, J. S. B. Mitchell, and S. S. Skiena. Hamiltonian triangulations for fast rendering. *Visual Comput.*, 12(9):429–444, 1996.

[6] D. Avis and H. ElGindy. Triangulating point sets in space. *Discrete and Computational Geometry*, 2:99–111, 1987.

[7] C. Bajaj and T. K. Dey. Convex decomposition of polyhedra and robustness. *SIAM Journal on Computing*, 21:339–364, 1992.

[8] T. J. Baker. Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation. *Engineering with Computers (USA)*, 5:161–175, 1989.

[9] I. J. Balaban. An optimal algorithm for finding segment intersections. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 211–219, 1995.

[10] P. Beirouti and J. Snoeyink. Implementations of the LMT heuristic for minimum weight triangulation. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 96–105, 1998.

[11] P. Belleville, M. Keil, M. McAllister, and J. Snoeyink. On computing edges that are in all minimum-weight triangulations. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages V7–V8, 1996.

[12] A. Below, U. Brehm, J. A. De Loera, and J. Richter-Gebert. Minimal simplicial dissections and triangulations of convex 3-polytopes. *Discrete and computational geometry*, 24:35–48, 2000.

[13] A. Below, J. A. De Loera, and J. Richter-Gebert. The complexity of finding small triangulations of convex 3-polytopes. unpublished manuscript, 2000.

[14] A. Below, J. A. De Loera, and J. Richter-Gebert. Finding minimal triangulations of convex 3-polytopes is NP-hard. In *Proc. 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 9–11, 2000.

[15] J. L. Bentley and T. A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, C-28(9):643–647, September 1979.

[16] M. Bern. Compatible tetrahedralizations. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 281–288, 1993.

[17] M. Bern and D. Eppstein. Mesh generation and optimal triangulation. In D.-Z. Du and F. K. Hwang, editors, *Computing in Euclidean Geometry*, volume 1 of *Lecture Notes Series on Computing*, pages 23–90. World Scientific, Singapore, 1992.

[18] M. Bern and D. Eppstein. Polynomial-size nonobtuse triangulation of polygons. *Int. J. Comp. Geometry and Applications*, 2:241–255, 1992.

[19] M. Bern, D. Eppstein, and J. Gilbert. Provably good mesh generation. *J. Comput. Syst. Sci.*, 48:384–409, 1994.

[20] J.-D. Boissonnat, A. Cérézo, O. Devillers, and M. Teillaud. Output-sensitive construction of the 3-d Delaunay triangulation of constrained sets of points. In *Proc. 3rd Canad. Conf. Comput. Geom.*, pages 110–113, 1991.

[21] P. Bose, L. Devroye, and W. Evans. Diamonds are not a minimum weight triangulation's best friend. In *Proc. 8th Canad. Conf. Comput. Geom.*, pages 68–73, 1996.

[22] J. Bramble and M. Zlanak. Triangular elements in the finite element method. *Math. Comput.*, 24:809–820, 1970.

[23] K. Q. Brown. Comments on "Algorithms for reporting and counting geometric intersections". *IEEE Trans. Comput.*, C-30:147–148, 1981.

[24] B. Chazelle. Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm. *SIAM Journal on Computing*, 13:488–507, 1984.

[25] B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *Journal of ACM*, 39(1):1–54, 1992.

[26] B. Chazelle and L. Palios. Triangulating a non-convex polytope. *Discrete and computational geometry*, 5:505–526, 1990.

[27] B. Chazelle and N. Shouraboura. Bounds on the size of tetrahedralizations. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 231–239, 1994.

[28] S.-W. Cheng, M. J. Golin, and J. C. F. Tsang. Expected case analysis of β-skeletons with applications to the construction of minimum-weight triangulations. In *Proc. 7th Canad. Conf. Comput. Geom.*, pages 279–284, 1995.

[29] S.-W. Cheng, N. Katoh, and M. Sugai. A study of the LMT-skeleton. In *Proc. 7th Annu. Internat. Sympos. Algorithms Comput.*, volume 1178 of *Lecture Notes in Computer Science*, pages 256–265, 1996.

[30] S.-W. Cheng and Y.-F. Xu. Approaching the largest β-skeleton within a minimum weight triangulation. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 196–203, 1996.

[31] L. P. Chew. Constrained delaunay triangulations. *Algorithmica*, 4:97–108, 1989.

[32] F. Chin and S. Fung. Approximation of minimum triangulation for polyhedron with bounded degrees. In *Proc. 12th Annu. Internat. Sympos. Algorithms Comput.*, volume 2223 of *Lecture Notes in Computer Science*, pages 172–184, 2001.

[33] F. Chin, S. Fung, and C.-A. Wang. Approximation for minimum triangulations of simplicial convex 3-polytopes. *Discrete and computational geometry*, 26:499–511, 2001.

[34] B. K. Choi, H. Y. Shin, Y. I. Yoon, and J. W. Lee. Triangulation of scattered data in 3d space. *Computer Aided Design*, 20(5):239–248, 1988.

[35] A. K. Cline and R. J. Renka. A constrained 2-dimensional triangulation and the solution of closest node problems in the presence of barriers. *SIAM Journal on Numerical Analysis*, 27(5):1305–1321, 1990.

[36] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* MIT Press, Cambridge, MA, 1990.

[37] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer-Verlag, Berlin, 1997.

[38] L. De Floriani and E. Puppo. A survey of constrained Delaunay triangulation algorithms for surface representaion. In G. G. Pieroni, editor, *Issues on Machine Vision.* Springer-Verlag, New York, NY, 1989.

[39] J. de Loera. Computing minimal and maximal triangulations of convex polytopes. unpublished manuscript, 1999.

[40] J. de Loera, F. Santos, and J. Urrutia. The number of geometric bistellar neighbors of a triangulation. *Discrete and computational geometry*, 21:131–142, 1999.

[41] T. K. Dey. Triangulation and CSG representation of polyhedra with arbitrary genus. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 364–372, 1991.

[42] T. K. Dey, C. L. Bajaj, and K. Sugihara. On good triangulations in three dimensions. *International Journal of Computational Geometry & Applications*, 2(1):75–95, 1992.

[43] T. K. Dey, K. Sugihara, and C. L. Bajaj. Delaunay triangulations in three dimensions with finite precision arithmetic. *Computer Aided Geometric Design*, 9:457–470, 1992.

[44] M. T. Dickerson, J. M. Keil, and M. H. Montague. A large subgraph of the minimum weight triangulation. *Discrete and computational geometry*, 18:289–304, 1997.

[45] M. T. Dickerson and M. H. Montague. A (usually?) connected subgraph of the minimum weight triangulation. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 204–213, 1996.

[46] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Heidelberg, West Germany, 1987.

[47] H. Edelsbrunner. *Geometry and Topology for Mesh Generation*, volume 7 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, UK, 2001.

[48] H. Edelsbrunner, F. P. Preparata, and D. B. West. Tetrahedrizing point sets in three dimensions. *Journal of Symbolic Computation*, 10:335–347, 1990.

[49] D. Eppstein. Approximating the minimum weight triangulation. *Discrete and Computational Geometry*, 11:163–191, 1994.

[50] T.-P. Fang and L. A. Piegl. Algorithm for constrained Delaunay triangulation. *Visual Comput.*, 10(5):255–265, 1994.

[51] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, NY, 1979.

[52] P. D. Gilbert. New results on planar triangulations. M.Sc. thesis, Report R-850, University of Illinois, Urbana, IL, 1979.

[53] J. E. Goodman and J. O'Rourke. *Handbook of Discrete and Computational Geometry.* CRC Press, New York, NY, 1997.

[54] J. E. Goodman and J. Pach. Cell decomposition of polytopes by bending. *Israel J. of Math.*, 64(2):129–138, 1988.

[55] R. Hainz, O. Aichholzer, and F. Aurenhammer. New results on minimum weight triangulations and the LMT-skeleton. In *Abstracts 13th European Workshop Comput. Geom.*, pages 4–6, 1997.

[56] L. Heath and S. Pemmaraju. New results for the minimum weight triangulation problem. *Algorithmica*, 12:533–552, 1994.

[57] J. Hershberger and J. Snoeyink. Convex polygons made from few lines and convex decompositions of polyhedra. In *Proc. 3rd Scand. Workshop Algorithm*

Theory, volume 621 of *Lecture Notes in Computer Science*, pages 376–387. Springer-Verlag, 1992.

[58] J. E. Hopcroft and R. E. Tarjan. Efficient planarity testing. *Journal of ACM*, 21:549–568, 1974.

[59] B. Joe. 3-dimensional triangulations from local transformations. *SIAM Journal on Scientific and Statistical Computing*, 10(4):718–741, 1989.

[60] B. Joe. Construction of three-dimensional Delaunay triangulations using local transformations. *Computer Aided Geometric Design*, 8(2):123–142, 1991.

[61] B. Joe. Delaunay versus max-min solid angle triangulations for three-dimensional mesh generation. *International Journal of Numerical Methods in Engineering*, 31(5):987–997, 1991.

[62] M. Keil. Computing a subgraph of the minimum weight triangulation. *Computational Geometry: Theory and Applications*, 4:13–26, 1994.

[63] D. G. Kirkpatrick and J. D. Radke. A framework for computational morphology. In G. T. Toussaint, editor, *Computational Geometry*, pages 217–248. North-Holland, Amsterdam, Netherlands, 1985.

[64] D. G. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm? *SIAM Journal on Computing*, 15:287–299, 1986.

[65] G. T. Klincsek. Minimal triangulations of polygonal domains. *Discrete Mathematics*, 9:121–123, 1980.

[66] D. E. Knuth. Big omicron and big omega and big theta. *SIGACT News*, 8(2):18–24, 1976.

[67] Y. Kyoda, K. Imai, F. Takeuchi, and A. Tajima. A branch-and-cut approach for minimum weight triangulation. In *Proc. 8th Annu. Internat. Sympos. Algorithms Comput.*, volume 1350 of *Lecture Notes in Computer Science*, pages 384–393, 1997.

[68] C. L. Lawson. Software for C^1 surface interpolation. In J. Rice, editor, *Mathematical Software III*, pages 161–194. Academic Press, 1977.

[69] W. Lenhart and G. Liotta. Drawing outerplanar minimum weight triangulations. *Information Processing Letters*, 57(5):253–260, 1996.

[70] W. Lenhart and G. Liotta. Drawable and forbidden minimum weight triangulations. In *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 1–12, 1998.

[71] N. J. Lennes. Theorems on the simple finite polygon and polyhedron. *American Journal of Mathematics*, 33:37–62, 1911.

[72] C. Levcopoulos and D. Krznaric. Quasi-greedy triangulations approximating the minimum weight triangulation. *Journal of Algorithms*, 27:303–338, 1998.

[73] A. Lingas. A new heuristic for minimum weight triangulation. *SIAM Journal on Algebraic Discrete Methods*, 8(4):646–658, 1987.

[74] E. L. Lloyd. On triangulations of a set of points in the plane. In *Proc. 18th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 228–240, 1977.

[75] H. Meijeri and D. Rappaport. Computing the minimum weight triangulation of a set of linearly ordered points. *Information Processing Letters*, 42:35–38, 1993.

[76] J. S. B. Mitchell and J. O'Rourke. Computational geometry column 42. *International Journal of Computational Geometry & Applications.* to appear.

[77] J.-M. Moreau and P. Volino. Constrained Delaunay triangulation revisited. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 340–345, 1993.

[78] J. O'Rourke. Computational geometry column 23. *International Journal of Computational Geometry & Applications*, 4(2):239–242, 1994.

[79] J. O'Rourke. Computational geometry column 40. *International Journal of Computational Geometry & Applications*, 10(6):649–651, 2000.

[80] J. Pach and M. Sharir. On vertical visibility in arrangements of segments and the queue size in the Bentley-Ottmann line sweeping algorithm. *SIAM Journal on Computing*, 20:460–470, 1991.

[81] D. A. Plaisted and J. Hong. A heuristic triangulation algorithm. *Journal of Algorithms*, 8:405–437, 1987.

[82] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, NY, 1985.

[83] V. T. Rajan. Optimality of the Delaunay triangulation in R^d. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 357–363, 1991.

[84] J. Richter-Gebert. Finding small triangulations of polytope boundaries is hard. *Discrete and Computational Geometry*, 24:503–517, 2000.

[85] M.-C. Rivara. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *International Journal of Numerical Methods in Engineering*, 20:745–756, 1984.

[86] B. L. Rothschild and E. G. Straus. On triangulations of the convex hull of n points. *Combinatorica*, 5:167–179, 1985.

[87] J. Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 83–92, 1993.

[88] J. Ruppert and R. Seidel. On the difficulty of triangulating three-dimensional non-convex polyhedra. *Discrete and Computational Geometry*, 7:227–253, 1992.

[89] F. Santos. A point configuration whose space of triangulations is disconnected. *J. Amer. Math. Soc.*, 13:611–637, 2000.

[90] E. Schönhardt. Uber die zerlegung von dreieckspolyedern in tetraeder. *Math. Annalen*, 98:309–312, 1928.

[91] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete and Computational Geometry*, 19:1–17, 1998.

[92] M. I. Shamos and D. Hoey. Closest-point problems. In *Proc. 16th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 151–162, 1975.

[93] M. I. Shamos and D. Hoey. Geometric intersection problems. In *Proc. 17th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 208–215, 1976.

[94] J. R. Shewchuk. Robust adaptive floating-point geometric predicates. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 141–150, 1996.

[95] J. R. Shewchuk. A condition guaranteeing the existence of higher-dimensional constrained delaunay triangulations. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 76–85, 1998.

[96] J. R. Shewchuk. Tetrahedral mesh generation by delaunay refinement. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 86–95, 1998.

[97] J. R. Shewchuk. Mesh generation for domains with small angles. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 1–10, 2000.

[98] J. R. Shewchuk. Sweep algorithms for constructing higher-dimensional constrained delaunay triangulations. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 350–359, 2000.

[99] A. Tajima and H. Imai. Computational investigations of the optimality of two and three dimensional triangulations under several criteria. In *Proc. 10th Canad. Conf. Comput. Geom.*, 1998.

[100] C. A. Wang, F. Chin, and Y.-F. Xu. A new subgraph of minimum weight triangulations. *Journal of Combinatorial Optimization*, 1:115–127, 1997.

[101] C. A. Wang, F. Chin, and B. Yang. Maximum weight triangulation and graph drawing. *Information Processing Letters*, 70:17–22, 1999.

[102] C. A. Wang, F. Chin, and B. Yang. Triangulations without minimum-weight drawing. *Information Processing Letters*, 74:183–189, 2000.

[103] C. A. Wang and Y.-F. Xu. Minimum weight triangulations with convex layers constraint. *Journal of Global Optimization*, 15:73–83, 1999.

[104] C. A. Wang and B. Yang. On non-proximity drawability of maximal planar graphs. In *3rd CGC Workshop on Computational Geometry*, 1998.

[105] C. A. Wang and B. Yang. A tight bound for β-skeleton of minimum weight triangulations. In *Proc. 6th International Workshop on Algorithms and Data*

Structures, volume 1663 of *Lecture Notes in Computer Science*, pages 265–275, 1999.

[106] C. A. Wang and B. Yang. Optimal tetrahedralizations of some convex polyhedra. In *Abstracts 16th European Workshop Comput. Geom.*, pages 5–9, 2000.

[107] C. A. Wang and B. Yang. Tetrahedralization of two nested convex polyhedra. In *Proc. 6th Annu. Internat. Conf. Computing and Combinatorics*, volume 1858 of *Lecture Notes in Computer Science*, pages 291–298, 2000.

[108] C. A. Wang and B. Yang. A lower bound for beta-skeleton belonging to minimum weight triangulations. *Computational Geometry: Theory and Applications*, 19:35–46, 2001.

[109] B. Yang. A better subgraph of the minimum weight triangulation. *Information Processing Letters*, 56:255–258, 1995.

[110] B. Yang and C. A. Wang. Minimal tetrahedralizations of a class of polyhedra. In *Proc. 12th Canad. Conf. Comput. Geom.*, pages 81–90, 2000.

[111] B. Yang, Y.-F. Xu, and Z.-Y. You. A chain decomposition algorithm for the proof of a property on minimum weight triangulations. In *Proc. 5th Annu. Internat. Sympos. Algorithms Comput.*, volume 834 of *Lecture Notes in Computer Science*, pages 423–427, 1994.

Appendix A

The proof of Theorem 2.4

Proof Refer to [37, Theorem 9.1] for a proof of necessity. Now we prove the sufficiency. Suppose $(V(\mathsf{E}), \mathsf{E})$ is a connected plane graph. After iteratively deleting the leaves and their adjacent edges from $(V(\mathsf{E}), \mathsf{E})$, we obtain a new graph denoted as $(V(E'), E')$. Note that $(V(E'), E')$ is a connected plane graph without degree 1 vertices. For $(V(E'), E')$, let $F(E')$ be the set of bounded faces and $\mathrm{bd}(E')$ be the set of the boundary edges of the unbounded face. From Euler's formula we know that $|V(E')| - |E'| + |F(E')| = 1$. Since each bounded face has at least three edges and each edge is shared by two faces, we have that $3|F(E')| + |\mathrm{bd}(E')| \leq 2|E'|$, where equality holds if and only if each bounded face is a triangle. Substituting Euler's formula into this inequality, we obtain

$$|E'| \leq 3|V(E')| - |\mathrm{bd}(E')| - 3. \tag{A.1}$$

In order to maintain connectivity, we recover E from E' by adding edges one at a time in the inverse order of the deletion for constructing E' from E. When we add an edge in a bounded face, the left side of (A.1) increases by 1 and the right side of (A.1) increases by 3. So the inequality (A.1) still holds, where the equality holds if and only if each bounded face is an E-empty triangle. When we add an edge in the unbounded face, there are two cases that occur. Let ab be the edge need to add and b be a new vertex in the current graph, and B be an empty set at the beginning of the adding process. From (A.1) we know the following inequality holds before adding edges.

$$|E'| \leq 3|V(E')| - (|B| + |\mathrm{bd}(E')|) - 3. \tag{A.2}$$

If $b \notin V(CH(\mathsf{E}))$, then the left side of (A.2) increases by 1 and the right side of (A.2) increases by 3. If $b \in V(CH(\mathsf{E}))$, then put b into B. So the left side of (A.2) increases by 1 and the right side of (A.2) increases by 2. Thus, after adding all the edges, the inequality (A.2) still holds. Since $V(CH(\mathsf{E})) \subseteq B \cup V(\mathrm{bd}(E'))$ (refer to Figure A.1), we have that $|E(CH(\mathsf{E}))| = |V(CH(\mathsf{E}))| \leq |B| + |V(\mathrm{bd}(E'))| = |B| + |\mathrm{bd}(E')|$, where equality holds if and only if $E(CH(\mathsf{E})) = \mathrm{bd}(E')$. Hence, from (A.2) we obtain

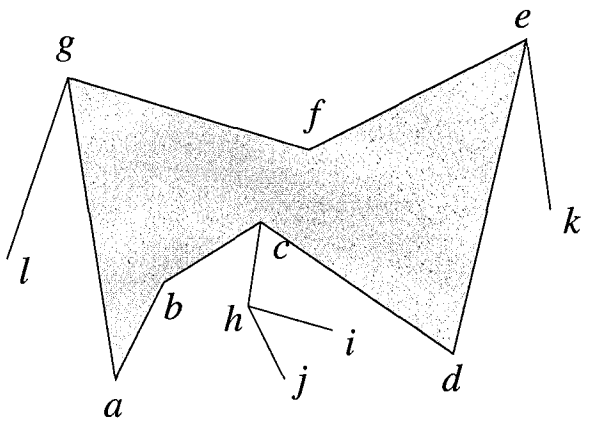

Figure A.1: $V(\mathrm{bd}(E')) = \{a, b, c, d, e, f, g\}, B = \{j, k, l\}$ and $V(CH(\mathsf{E}))$= {a, j, d, k, e, g, l}.

$$|\mathsf{E}| \leq 3|V(\mathsf{E})| - |V(CH(\mathsf{E}))| - 3, \tag{A.3}$$

where equality holds if and only if each bounded face is an E-empty triangle and the unbounded face is the exterior of $CH(\mathsf{E})$. Therefore, it follows from the condition of the theorem that E is a triangulation. □

Appendix B

The proof of Lemma 3.3

Proof Refer to Figure 3.17. We only prove the lemma for $m = 3$; it can be proven in a similar fashion for $m > 3$. Consider the facet $z_0z_1v_2$. $v_3, z_3 \notin \text{tetra}(z_0z_1v_2)$ since $v_2v_3, z_0z_3 \notin T$. Thus, $\text{tetra}(z_0z_1v_2)$ may contain z_2, or v_0, or v_1, or the vertex c which does not belong to $V(\overline{B}_2)$. Therefore, we have the following cases:

1. $z_2 \in \text{tetra}(z_0z_1v_2)$. Then $z_0z_2 \in T$, which is a diag-I (see Figure B.1(left)). Consider the interior facet $z_0z_2v_2$. $z_1 \in \text{tetra}(z_0z_2v_2)$, and the other vertex in $\text{tetra}(z_0z_2v_2)$ may be v_0, or v_1, or the vertex a which does not belong to $V(\overline{B}_2)$.

   (a) $v_0 \in \text{tetra}(z_0z_2v_2)$. Then $z_2v_0, v_0v_2 \in T$, which are diag-Is. In total, there are at least three diag-Is on $\overline{B}_2$.

   (b) $v_1 \in \text{tetra}(z_0z_2v_2)$. Then $z_2v_1 \in T$, which is a diag-I. Considering $z_0z_2v_1$,

at least one diag-I or diag-II is incident on z_0, or z_2, or v_1. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_2$.

(c) $a \in \text{tetra}(z_0 z_2 v_2)$. Then $v_2 a, z_2 a \in T$, which are diag-IIs. Considering $z_0 v_1 v_2$, at least one diag-I or diag-II is incident on z_0, or v_1, or v_2. In total, there are at least two diag-Is and two diag-IIs, or one diag-I and three diag-IIs on $\overline{B}_2$.

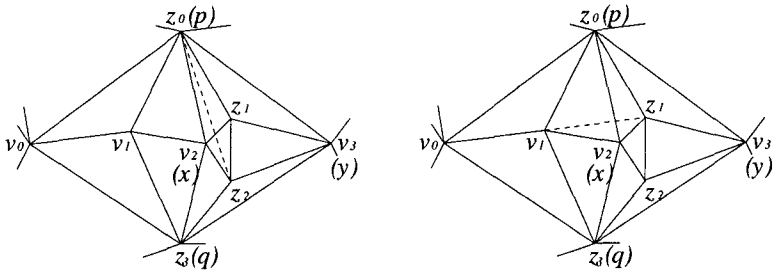

Figure B.1: Case 1 (left) and Case 2 (right) of the proof of Lemma 3.3.

2. $v_1 \in \text{tetra}(z_0 z_1 v_2)$. Then $z_1 v_1 \in T$, which is a diag-I (see Figure B.1(right)). Consider the interior facet $z_1 v_1 v_2$. $z_0 \in \text{tetra}(z_1 v_1 v_2)$, and the other vertex in $\text{tetra}(z_1 v_1 v_2)$ may be z_2, or z_3, or v_0, or the vertex b which does not belong to $V(\overline{B}_2)$.

(a) $z_2 \in \text{tetra}(z_1 v_1 v_2)$. Then $z_2 v_1 \in T$, which is a diag-I. Considering $z_1 z_2 v_1$, at least one diag-I or diag-II is incident on z_1, or z_2, or v_1. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_2$.

(b) $z_3 \in \text{tetra}(z_1 v_1 v_2)$. Then $z_1 z_3 \in T$, which is a diag-I. Considering $z_1 z_3 v_1$, at least one diag-I or one diag-II is incident on z_1, or z_3, or v_1. In total,

there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_2$.

(c) $v_0 \in \text{tetra}(z_1 v_1 v_2)$. Then $z_1 v_0, v_0 v_2 \in T$, which are diag-Is. In total, there are at least three diag-Is on $\overline{B}_2$.

(d) $b \in \text{tetra}(z_1 v_1 v_2)$. Then $z_1 b, v_1 b, v_2 b \in T$, which are diag-IIs. In total, there are at least one diag-I and three diag-IIs on $\overline{B}_2$.

3. $v_0 \in \text{tetra}(z_0 z_1 v_2)$. Then $z_1 v_0, v_0 v_2 \in T$, which are diag-Is (see Figure B.2(left)). Considering the interior facet $z_1 v_0 v_2$, at least one diag-I or diag-II is incident on z_1, or v_0, or v_2. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_2$.

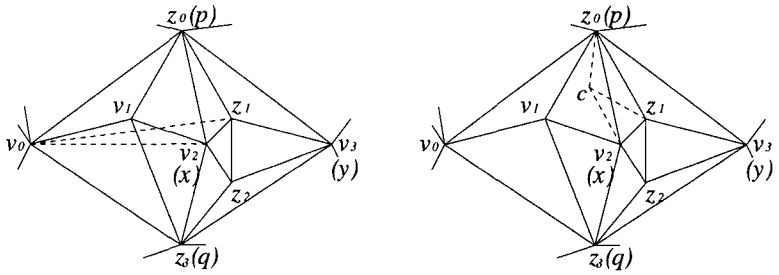

Figure B.2: Case 3 (left) and Case 4 (right) of the proof of Lemma 3.3.

4. $c \in \text{tetra}(z_0 z_1 v_2)$ and $c \notin V(\overline{B}_2)$. Then $z_1 c, v_2 c \in T$, which are diag-IIs (see Figure B.2(right)). Consider $z_1 z_2 v_2$.

(a) $z_3 \in \text{tetra}(z_1 z_2 v_2)$. Then $z_1 z_3 \in T$, which is a diag-I. Considering $z_3 v_1 v_2$, at least one diag-I or diag-II is incident on z_3, or v_1, or v_2. In total, there are at least two diag-Is and two diag-IIs, or one diag-I and three diag-IIs on $\overline{B}_2$.

(b) $v_0 \in \text{tetra}(z_1 z_2 v_2)$. Then $z_1 v_0, z_2 v_0, v_0 v_2 \in T$, which are diag-Is. In total, there are at least three diag-Is and two diag-IIs on $\overline{B}_2$.

(c) $v_1 \in \text{tetra}(z_1 z_2 v_2)$. Then $z_1 v_1, z_2 v_1 \in T$, which are diag-Is. In total, there are at least two diag-Is and two diag-IIs on $\overline{B}_2$.

(d) $c' \in \text{tetra}(z_1 z_2 v_2)$. Then $z_1 c', z_2 c', v_2 c' \in T$, which are diag-IIs. In total, there are at least five diag-IIs on $\overline{B}_2$.

(e) $c \in \text{tetra}(z_1 z_2 v_2)$. Then $z_2 c \in T$, which is a diag-II. Consider $z_0 v_1 v_2$.

i. z_2 or v_0 belong to $\text{tetra}(z_0 v_1 v_2)$. At least one diag-I is incident on z_0, or v_1, or v_2. In total, there are at least one diag-I and three diag-IIs on $\overline{B}_2$.

ii. $c'' \in \text{tetra}(z_0 v_1 v_2)$ and $c'' \notin V(\overline{B}_2)$. Then $v_1 c'', v_2 c'' \in T$. In total, there are at least five diag-IIs on $\overline{B}_2$.

iii. $c \in \text{tetra}(z_0 v_1 v_2)$. Then $v_1 c \in T$. Consider the other triangle facets in $\overline{B}_2$ (except $z_0 z_1 v_2, z_1 z_2 v_2, z_0 v_1 v_2$). If there is at least one triangle facet whose tetra-vertex is $d \neq c$, then $\overline{B}_2$ associates with at least one diag-I and four diag-IIs, or five diag-IIs; otherwise, if all the vertices in $\overline{B}_2$ are adjacent to c, then there are four diag-IIs on $\overline{B}_2$ with a common endpoint c. This completes the proof of Lemma 3.3.

□

Appendix C

The proof of Lemma 3.4

Proof Refer to Figure 3.18. We only prove the lemma for $m = 3$; it can be proven in a similar fashion for $m > 3$. Consider the facet pz_1z_2. $v_2, q \notin \text{tetra}(pz_1z_2)$ since $pv_2, pq \notin T$. Thus, $\text{tetra}(pz_1z_2)$ may contain v_0, or v_1, or v_3, or the vertex c which does not belong to $V(\overline{B}_3)$. Therefore, we have the following cases:

1. $v_0 \in \text{tetra}(pz_1z_2)$. Then $z_1v_0, z_2v_0 \in T$, which are diag-Is (see Figure C.1(left)). Considering $z_1z_2v_2$, at least one diag-I or diag-II is incident on z_1, or z_2, or v_2. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_3$.

2. $v_1 \in \text{tetra}(pz_1z_2)$. Then $z_2v_1 \in T$, which is a diag-I (see Figure C.1(right)). Consider the interior facet pz_2v_1. $z_1 \in \text{tetra}(pz_2v_1)$, and the other vertex in $\text{tetra}(pz_2v_1)$ may be v_0, or v_3, or the vertex a which does not belong to $V(\overline{B}_3)$.

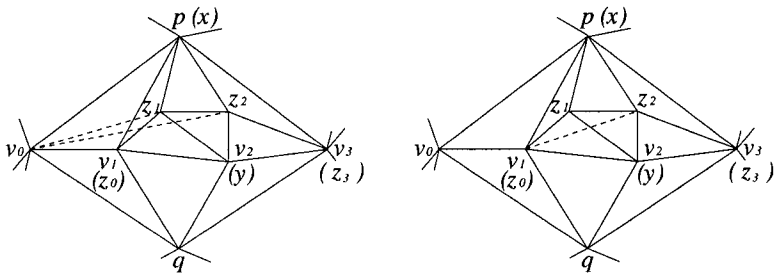

Figure C.1: Case 1 (left) and Case 2 (right) of the proof of Lemma 3.4.

(a) $v_0 \in \text{tetra}(pz_2v_1)$. Then $z_2v_0 \in T$, which is a diag-I. Considering $z_2v_0v_1$, at least one diag-I or diag-II is incident on z_2, or v_0, or v_1. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_3$.

(b) $v_3 \in \text{tetra}(pz_2v_1)$. Then $v_1v_3 \in T$, which is a diag-I. Considering $z_2v_1v_3$, at least one diag-I or one diag-II is incident on z_2, or v_1, or v_3. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_3$.

(c) $a \in \text{tetra}(pz_2v_1)$. Then $z_2a, v_1a \in T$, which are diag-IIs. Considering v_1v_2q, at least one diag-I or one diag-II is incident on v_1, or v_2, or q. In total, there are at least two diag-Is and two diag-IIs, or one diag-I and three diag-IIs on $\overline{B}_3$.

3. $v_3 \in \text{tetra}(pz_1z_2)$. Then $z_1v_3 \in T$, which is a diag-I (see Figure C.2(left)). Consider the interior facet $z_1v_2v_3$. $z_2 \in \text{tetra}(z_1v_2v_3)$, and the other vertex in $\text{tetra}(z_1v_2v_3)$ may be v_0, or v_1, or q, or the vertex b which does not belong to $V(\overline{B}_3)$.

(a) $v_0 \in \text{tetra}(z_1v_2v_3)$. Then $z_1v_0, v_0v_2, v_0v_3 \in T$, which are diag-Is. In total,

there are at least four diag-Is on $\overline{B}_3$

(b) $v_1 \in \text{tetra}(z_1 v_2 v_3)$. Then $v_1 v_3 \in T$, which is a diag-I. Considering $v_1 v_2 q$, at least one diag-I or one diag-II is incident on v_1, or v_2, or q. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_3$.

(c) $q \in \text{tetra}(z_1 v_2 v_3)$. Then $z_1 q \in T$, which is a diag-I. Considering $z_1 v_1 q$, at least one diag-I or one diag-II is incident on z_1, or v_1, or q. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_3$.

(d) $b \in \text{tetra}(z_1 v_2 v_3)$. Then $z_1 b, v_2 b \in T$, which are diag-IIs. Considering $v_1 v_2 q$, at least one diag-I or one diag-II is incident on v_1, or v_2, or q. In total, there are at least two diag-Is and two diag-IIs, or one diag-I and three diag-IIs on $\overline{B}_3$.

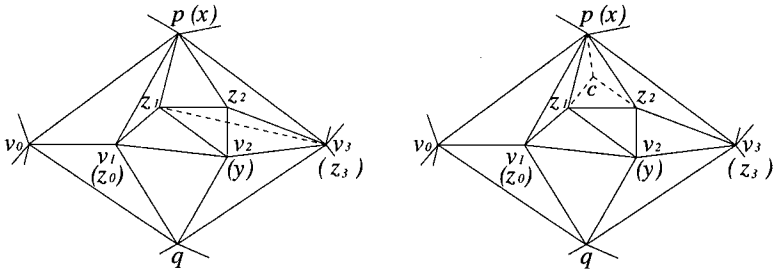

Figure C.2: Case 3 (left) and Case 4 (right) of the proof of Lemma 3.4.

4. $c \in \text{tetra}(p z_1 z_2)$ and $c \notin V(\overline{B}_3)$. Then $z_1 c, z_2 c \in T$, which are diag-IIs (see Figure C.2(right)). Consider $z_1 z_2 v_2$.

(a) $v_0 \in \text{tetra}(z_1 z_2 v_2)$. Then $z_1 v_0, z_2 v_0, v_0 v_2 \in T$, which are diag-Is. In total, there are three diag-Is and two diag-IIs on $\overline{B}_3$.

(b) $v_1 \in \text{tetra}(z_1 z_2 v_2)$. Then $z_2 v_1 \in T$, which is a diag-I. Considering $v_1 v_2 q$, at least one diag-I or one diag-II is incident on v_1, or v_2, or q. In total, there are at least two diag-Is and two diag-IIs, or one diag-I and three diag-IIs on $\overline{B}_3$.

(c) $v_3 \in \text{tetra}(z_1 z_2 v_2)$. Then $z_1 v_3 \in T$, which is a diag-I. Considering $v_1 v_2 q$, at least one diag-I or one diag-II is incident on v_1, or v_2, or q. In total, there are at least two diag-Is and two diag-IIs, or one diag-I and three diag-IIs on $\overline{B}_3$.

(d) $q \in \text{tetra}(z_1 z_2 v_2)$. Then $z_1 q, z_2 q \in T$, which are diag-Is. In total, there are at least two diag-Is and two diag-IIs on $\overline{B}_3$.

(e) $c' \in \text{tetra}(z_1 z_2 v_2)$. Then $z_1 c', z_2 c', v_2 c' \in T$, which are diag-IIs. In total, there are at least five diag-IIs on $\overline{B}_3$.

(f) $c \in \text{tetra}(z_1 z_2 v_2)$. Then $z_2 c \in T$, which is a diag-II. Consider $z_1 v_1 v_2$.

i. q or v_0 belongs to $\text{tetra}(z_1 v_1 v_2)$. At least one diag-I is incident on z_1, or v_1, or v_2. In total, there are at least one diag-I and three diag-IIs on $\overline{B}_3$.

ii. $c'' \in \text{tetra}(z_1 v_1 v_2)$ and $c'' \notin V(\overline{B}_3)$. Then $z_1 c'', v_1 c'', v_2 c'' \in T$. In total, there are at least six diag-IIs on $\overline{B}_3$.

iii. $c \in \text{tetra}(z_1 v_1 v_2)$. Then $v_1 c \in T$. Consider the other triangle facets in $\overline{B}_3$ (except $p z_1 z_2, z_1 z_2 v_2, z_1 v_1 v_2$). If there is at least one triangle facet whose tetra-vertex is $d \neq c$, then $\overline{B}_3$ associates with at least one diag-I

and four diag-IIs, or five diag-IIs; otherwise, if all the vertices in $\overline{B}_3$ are adjacent to c, then there are four diag-IIs on $\overline{B}_3$ with a common endpoint c. This completes the proof of Lemma 3.4.

□

Appendix D

The proof of Lemma 3.5

Proof Refer to Figure 3.19. We only prove the lemma for $m = 3$; it can be proven in a similar fashion for $m > 3$. Consider the facet pz_1z_2. $v_3, q \notin \text{tetra}(pz_1z_2)$ since $pv_3, pq \notin T$. Thus, $\text{tetra}(pz_1z_2)$ may contain z_3, or v_0, or v_1, or v_2, or the vertex c which does not belong to $V(\overline{B}_4)$. Therefore, we have the following cases:

1. $z_3 \in \text{tetra}(pz_1z_2)$. Then $z_1z_3 \in T$, which is a diag-I (see Figure D.1(left)). Consider the interior facet pz_1z_3. $z_2 \in \text{tetra}(pz_1z_3)$, and the other vertex in $\text{tetra}(pz_1z_3)$ may be v_0, or v_1, or v_2, or the vertex a which does not belong to $V(\overline{B}_4)$.

   (a) $v_0 \in \text{tetra}(pz_1z_3)$. Then $z_1v_0, z_3v_0 \in T$, which are diag-Is. In total, there are at least three diag-Is on $\overline{B}_4$.

(b) $v_1 \in \text{tetra}(pz_1z_3)$. Then $z_1v_1, z_3v_1 \in T$, which are diag-Is. In total, there are at least three diag-Is on $\overline{B}_4$.

(c) $v_2 \in \text{tetra}(pz_1z_3)$. Then $z_3v_2 \in T$, which is a diag-I. Considering pv_1v_2, at least one diag-I or one diag-II is incident on p, or v_1 or v_2. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_4$.

(d) $a \in \text{tetra}(pz_1z_3)$. Then $z_1a \in T$, which is a diag-II. Consider pz_1v_2.

i. $v_0 \in \text{tetra}(pz_1v_2)$. Then $z_1v_0, v_0v_2 \in T$, which are diag-Is. In total, there are at least three diag-Is and one diag-II on $\overline{B}_4$.

ii. $v_1 \in \text{tetra}(pz_1v_2)$. Then $z_1v_1 \in T$, which is a diag-I. In total, there are at least two diag-Is and one diag-II on $\overline{B}_4$.

iii. $a' \in \text{tetra}(pz_1v_2)$ and $a' \notin V(\overline{B}_4)$. Then $z_1a', v_2a' \in T$. In total, there are at least one diag-I and three diag-IIs on $\overline{B}_4$.

iv. $a \in \text{tetra}(pz_1v_2)$. Then $v_2a \in T$. Considering pv_1v_2, at least one diag-I or one diag-II is incident on p, or v_1, or v_2. In total, there are at least two diag-Is and two diag-IIs, or one diag-I and three diag-IIs on $\overline{B}_4$.

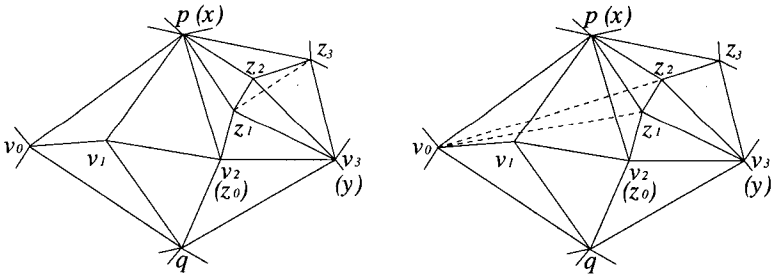

Figure D.1: Case 1 (left) and Case 2 (right) of the proof of Lemma 3.5.

2. $v_0 \in \text{tetra}(pz_1z_2)$. Then $z_1v_0, z_2v_0 \in T$, which are diag-Is (see Figure D.1(right)). Considering pz_2v_0, at least one diag-I or diag-II is incident on p, or z_2, or v_0. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_4$.

3. $v_1 \in \text{tetra}(pz_1z_2)$. Then $z_1v_1, z_2v_1 \in T$, which are diag-Is. Considering pz_2v_1, at least one diag-I or diag-II is incident on p, or z_2, or v_1. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_4$.

4. $v_2 \in \text{tetra}(pz_1z_2)$. Then $z_2v_2 \in T$, which is a diag-I (see Figure D.2(left)). Consider the interior facet pz_2v_2. $z_1 \in \text{tetra}(pz_2v_2)$, and the other vertex in $\text{tetra}(pz_2v_2)$ may be z_3, or v_0, or v_1, or the vertex b which does not belong to $V(\overline{B}_4)$.

   (a) $z_3 \in \text{tetra}(pz_2v_2)$. Then $z_3v_2 \in T$, which is a diag-I. Considering pz_3v_2, at least one diag-I or diag-II is incident on p, or z_3, or v_2. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_4$.

   (b) $v_0 \in \text{tetra}(pz_2v_2)$. Then $z_2v_0, v_2v_0 \in T$, which are diag-Is. In total, there are at least three diag-Is on $\overline{B}_4$.

   (c) $v_1 \in \text{tetra}(pz_2v_2)$. Then $z_2v_1 \in T$, which is a diag-I. Considering pz_2v_1, at least one diag-I or one diag-II is incident on p, or z_2, or v_1. In total, there are at least three diag-Is, or two diag-Is and one diag-II on $\overline{B}_4$.

   (d) $b \in \text{tetra}(pz_2v_2)$. Then $z_2b, v_2b \in T$, which are diag-IIs. Considering pv_1v_2, at least one diag-I or one diag-II is incident on v_1, or v_2, or p. In total,

there are at least two diag-Is and two diag-IIs, or one diag-I and three diag-IIs on $\overline{B}_4$.

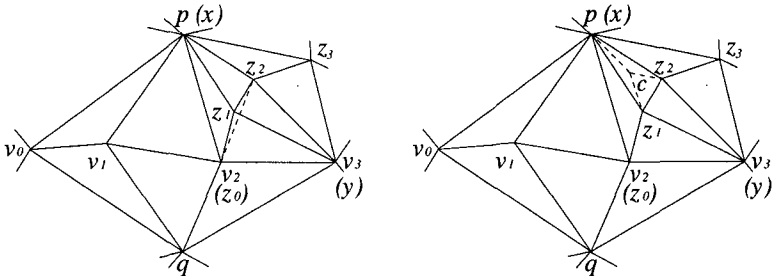

Figure D.2: Case 3 (left) and Case 4 (right) of the proof of Lemma 3.5.

5. $c \in \text{tetra}(pz_1z_2)$ and $c \notin V(\overline{B}_4)$. Then $z_1c, z_2c \in T$, which are diag-IIs (see Figure D.2(right)). Consider pz_1v_2. z_3 cannot be the tetra-vertex of pz_1v_2 (otherwise, $z_3 \in \text{tetra}(pz_1z_2)$, which contradicts $c \in \text{tetra}(pz_1z_2)$).

   (a) $v_0 \in \text{tetra}(pz_1v_2)$. Then $z_1v_0, v_0v_2 \in T$, which are diag-Is. In total, there are two diag-Is and two diag-IIs on $\overline{B}_4$.

   (b) $v_1 \in \text{tetra}(pz_1v_2)$. Then $z_1v_1 \in T$, which is a diag-I. Considering $z_1v_1v_2$, at least one diag-I or one diag-II is incident on z_1, or v_1, or v_2. In total, there are at least two diag-Is and two diag-IIs, or one diag-I and three diag-IIs on $\overline{B}_4$.

   (c) $c' \in \text{tetra}(pz_1v_2)$. Then $z_1c', v_2c' \in T$, which are diag-II. Considering pv_1v_2, at least one diag-I or one diag-II is incident on v_1, or v_2, or p. In total, there are at least one diag-I and four diag-IIs, or five diag-IIs on $\overline{B}_4$.

   (d) $c \in \text{tetra}(pz_1v_2)$. Then $v_2c \in T$, which is a diag-II. Consider pv_1v_2.

i. $v_0 \in \text{tetra}(pv_1v_2)$. Then $v_0v_2 \in T$, which is a diag-I. In total, there are at least one diag-I and three diag-IIs on $\overline{B}_4$.

ii. $v_3 \in \text{tetra}(pv_1v_2)$. Then $v_1v_3 \in T$, which is a diag-I. In total, there are at least one diag-I and three diag-IIs on $\overline{B}_4$.

iii. $c'' \in \text{tetra}(pv_1v_2)$ and $c'' \notin V(\overline{B}_4)$. Then $v_1c'', v_2c'' \in T$. In total, there are at least five diag-IIs on $\overline{B}_4$.

iv. $c \in \text{tetra}(pv_1v_2)$. Then $v_1c \in T$. Consider the other triangle facets in $\overline{B}_4$ (except $pz_1z_2, pz_1v_2, pv_1v_2$). If there is at least one triangle facet whose tetra-vertex is $d \neq c$, then $\overline{B}_4$ associates with at least one diag-I and four diag-IIs, or five diag-IIs; otherwise, if all the vertices in $\overline{B}_4$ are adjacent to c, then there are four diag-IIs on $\overline{B}_4$ with a common endpoint c. This completes the proof of Lemma 3.5.

□