Computational Geometry 46 (2013) 148–153

Contents lists available at SciVerse ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Bounded-degree polyhedronization of point sets

Gill Barequet [a,b,1], Nadia Benbernou [c], David Charlton [c], Erik D. Demaine [c], Martin L. Demaine [c], Mashhood Ishaque [a,2], Anna Lubiw [d], André Schulz [e,3], Diane L. Souvaine [a,2], Godfried T. Toussaint [f,g,a,4], Andrew Winslow [a,*,2]

[a] Department of Computer Science, Tufts University, MA, United States
[b] Department of Computer Science, The Technion—Israel Institute of Technology, Israel
[c] Computer Science and Artificial Intelligence Laboratory, MIT, MA, United States
[d] David R. Cheriton School of Computer Science, University of Waterloo, Canada
[e] Institut für Mathematische Logik und Grundlagenforschung, Westfälische Wilhelms-Universität, Münster, Germany
[f] Department of Music, Harvard University, MA, United States
[g] School of Computer Science, McGill University, Canada

ARTICLE INFO

Article history:
Received 31 December 2010
Accepted 28 February 2012
Available online 7 March 2012
Communicated by Stephane Durocher

Keywords:
Serpentine
Tetrahedralization
Convex hull
Gift wrapping

ABSTRACT

In 1994 Grünbaum showed that, given a point set S in $\mathbb{R}^3$, it is always possible to construct a polyhedron whose vertices are exactly S. Such a polyhedron is called a polyhedronization of S. Agarwal et al. extended this work in 2008 by showing that there always exists a polyhedronization that can be decomposed into a union of tetrahedra (tetrahedralizable). In the same work they introduced the notion of a serpentine polyhedronization for which the dual of its tetrahedralization is a chain. In this work we present a randomized algorithm running in $O(n\log^6 n)$ expected time which constructs a serpentine polyhedronization that has vertices with degree at most 7, answering an open question by Agarwal et al.

1. Introduction

Any set S of points in the plane (not all of which are collinear) admits a *polygonization*, that is, there is a simple polygon whose vertex set is exactly S. Similarly, a point set $S \subset \mathbb{R}^3$ admits a *polyhedronization* if there exists a simple polyhedron that has exactly S as its vertices. In 1994, Grünbaum proved that every point set in $\mathbb{R}^3$ (not all of which are coplanar) admits a polyhedronization. Unfortunately, the polyhedronizations generated by Grünbaum's method can be impossible to tetrahedralize. This is because they may contain *Schönhardt polyhedra*, a class of nontetrahedralizable polyhedra [5].

In 2008, Agarwal, Hurtado, Toussaint, and Trias described a variety of methods for producing polyhedronizations with various properties [1]. One of these methods, called hinge polyhedronization, produces *serpentine polyhedronizations*, meaning that the polyhedron admits a tetrahedralization whose dual (a graph where each tetrahedron is a node and each edge connects a pair of nodes whose primal entities are tetrahedra sharing a face) is a chain. Serpentine polyhedronizations

* Corresponding author.
E-mail addresses: barequet@cs.tufts.edu (G. Barequet), nbenbern@mit.edu (N. Benbernou), dchar@mit.edu (D. Charlton), edemaine@mit.edu (E.D. Demaine), mdemaine@mit.edu (M.L. Demaine), mishaq01@cs.tufts.edu (M. Ishaque), alubiw@uwaterloo.ca (A. Lubiw), andre.schulz@uni-muenster.de (A. Schulz), dls@cs.tufts.edu (D.L. Souvaine), godfried@cs.mcgill.ca (G.T. Toussaint), awinslow@cs.tufts.edu (A. Winslow).
1 Research supported in part by a grant from Intel Corporation.
2 Research supported in part by NSF grants CCF-0830734 and CBET-0941538.
3 Partially supported by the German Research Foundation (DFG) under grant SCHU 2458/1-1.
4 Research supported in part by NSERC (Canada) and the Radcliffe Institute for Advanced Study at Harvard University.

0925-7721/$ – see front matter
doi:10.1016/j.comgeo.2012.02.008

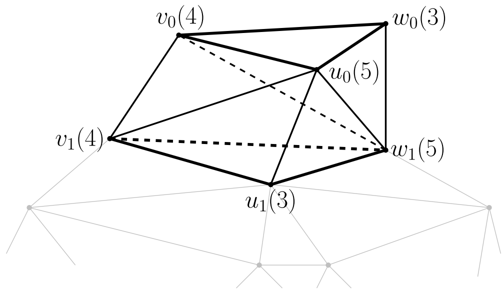

Fig. 1. Constructing a tunnel between T_0, T_1. The vertices u_0 and v_0 have degree 5 and 4, while w_0 has degree 3. The other end of the tunnel, T_1, has three vertices that will be labeled u_1, v_1, w_1 with degree 3, 4, and 5 (shown in parentheses), respectively.

produced by the hinge polyhedronization method are guaranteed to have two vertices with edges to every other vertex in the set. As a result, two vertices in these constructions have degree $n-1$, where n is the number of points in the set. It should be noted that some tetrahedra produced by this method may be degenerate if the point set is not in general position, that is, it contains four coplanar points. A natural question, and one posed by Agarwal et al., is whether it is always possible to create serpentine polyhedronizations with bounded degree.

In this work we describe a randomized algorithm for point sets in general position that constructs a serpentine polyhedronization with constant bounded degree. This algorithm runs in $O(n \log^6 n)$ expected time, and the expectation is independent of the input point set and output polyhedronization. The bound on the degree of the produced polyhedronizations is 7, which we show is nearly optimal for all sets of more than 12 points. Such bounded-degree serpentine polyhedronizations are useful in applications of modeling and graphics, where low local complexity is desirable for engineering and computational efficiency.

2. Setting

The convex hull of S, written $\mathcal{CH}(S)$, is the intersection of all half-spaces containing S. The boundary of each face of $\mathcal{CH}(S)$ is a polygon with coplanar vertices. In the next five sections we assume that S has no four coplanar points, and in the conclusion briefly discuss relaxing this assumption. so each of the faces of $\mathcal{CH}(S)$ is triangular. We call the three vertices composing a face of $\mathcal{CH}(S)$ a *face triplet*.

We will make reference to points and faces that *see* each other. We say that a pair of points p, q can see each other if the segment pq does not intersect a portion of any polyhedron present (either the convex hull of a set of points or a portion of the partially constructed polyhedronization). Similarly, two segments s_1 and s_2 can see each other if every pair of points $p \in s_1$ and $q \in s_2$ can see each other. A face f is the planar region bounded by a triangle formed by three points of S. A point p can see a face f if p can see every point in f (strong visibility). Similarly, a point p can see a segment s if p can see every point on s.

3. Algorithm

In this section we present a high-level description of the algorithm. Begin with a point set $S \subset \mathbb{R}^3$. Select a face triplet of $\mathcal{CH}(S)$ arbitrarily. Call this face triplet T_0. Let $S_0 = S \setminus T_0$. Assign the labels u_0, v_0, w_0 arbitrarily to the vertices of T_0 and connect the three vertices to form a triangle.

Next we search for a face triplet T_1 of $\mathcal{CH}(S_0)$ that we can attach to the triangle T_0 via a polyhedron *tunnel* (see Fig. 1). The tunnel is tetrahedralizable and has the face triplet T_0 at one end, the face triplet T_1 at the other end, and it is disjoint from the interior of $\mathcal{CH}(S_0)$. The method for selecting T_1 is described in the next section. Once the tunnel is construction, we will require that vertex w_0 has degree 3 and vertices u_0 and v_0 have degrees 5 and 4 (not necessarily respectively). Moreover, the vertices of the face triplet T_1 which we will call u_1, v_1, w_1 should have degree 3, 4, and 5, respectively. Note that the vertex degree only counts edges in the tunnel and that the constructed tunnel must meet the degree requirements for the vertices of T_0 while it determines the labeling of the vertices u_1, v_1, w_1 in T_1.

After finding a face triplet T_1 that meets these requirements, the process is repeated for T_1 and S_1, T_2 and S_2, where $S_i = S_{i-1} \setminus T_i$, until S_i contains fewer than three points. At each step T_i has three vertices: u_i connected to 3 vertices of S_i, and v_i, w_i that are connected to 1 and 2 vertices of S_i (not necessarily respectively). Once S_i contains fewer than three points, a degenerate tunnel is built out of the remaining points and the algorithm stops. In Sections 4 and 5 we prove that such a construction is always possible, producing a valid serpentine polyhedronization with vertex degrees bounded by 7. In Section 6 we provide details regarding the data structures used and provide an analysis of the algorithm's running time.

4. Tunnel construction

Here we prove that given T_i, it is always possible to find a face triplet T_{i+1} such that a three-tetrahedron tunnel $(\Delta_1 \Delta_2 \Delta_3)$ can be constructed between them.