Optimally Cutting a Surface into a Disk*

Jeff Erickson† Sariel Har-Peled‡

University of Illinois at Urbana-Champaign
{jeffe,sariel}@cs.uiuc.edu
http://www.cs.uiuc.edu/~{jeffe,sariel}

Submitted to *Discrete & Computational Geometry*: July 2, 2002
Revised and resubmitted June 3, 2003

Abstract

We consider the problem of cutting a subset of the edges of a polyhedral manifold surface, possibly with boundary, to obtain a single topological disk, minimizing either the total number of cut edges or their total length. We show that this problem is NP-hard in general, even for manifolds without boundary and for punctured spheres. We also describe an algorithm with running time $n^{O(g+k)}$, where n is the combinatorial complexity, g is the genus, and k is the number of boundary components of the input surface. Finally, we describe a greedy algorithm that outputs a $O(\log^2 g)$-approximation of the minimum cut graph in $O(g^2 n \log n)$ time.

*A preliminary version of this paper was presented at the 18th Annual ACM Symposium on Computational Geometry [20]. See http://www.cs.uiuc.edu/~jeffe/pubs/schema.html for the most recent version of this paper.

†Partially supported by a Sloan Fellowship, NSF CAREER award CCR-0093348, and NSF ITR grant DMR-0121695.

‡Partially supported by NSF CAREER award CCR-0132901.

1 Introduction

Several applications of three-dimensional surfaces, such as parameterization [21, 50] and texture mapping [3, 47], require information about the underlying topological structure in addition to the geometry. In some cases, we wish to simplify the surface topology, to facilitate algorithms that can be performed only if the surface is a topological disk.

In the texture mapping problem, we wish to find a continuous and invertible mapping from the texture, usually a two-dimensional rectangular image, to the surface. Unfortunately, if the surface is not a topological disk, no such map exists. In such a case, the only feasible solution is to cut the surface so that it becomes a topological disk. (Haker *et al.* [28] present an algorithm for directly texture mapping models with the topology of a sphere, where the texture is also embedded on a sphere.) Of course, when cutting the surface, one would like to find the best possible cut under various considerations. For example, one might want to cut the surface so that the resulting surface can be texture mapped with minimum distortion [21, 27, 50]. To our knowledge, all previous approaches for the texture-mapping problem either rely on cutting heuristics with no quality guarantees [27, 49, 52] or require the user to cut the surface beforehand [21, 47].

One method for reducing a manifold to a topological disk is to cut along the boundary of a so-called *canonical polygonal schema*. This boundary is a set of $2g$ cycles, all passing through a common base point in $\mathcal{M}$, such that cutting along those cycles breaks $\mathcal{M}$ into a topological disk. (We give more formal definitions in Section 2.) Lazarus *et al.* [40] presented and implemented two algorithms for computing a canonical polygonal schema of an orientable surface of complexity n and with genus g, in time $O(gn)$, simplifying an earlier algorithm of Vegter and Yap [60]. Since those cycles must share a common point, it is easy to find examples where the overall size of those cycles is $\Omega(gn)$. Furthermore, those cycles share several edges and are visually unsatisfying.

For most applications, however, canonical schemata are overkill. It is usually sufficient to find an arbitrary collection of edges whose removal transforms the surface into a topological disk. We call such a set of edges a *cut graph*; see Figure 1 for an example. Cut graphs have several advantages. First, they are compact. Trivially, any cut graph contains at most n edges of the surface mesh, much less than any canonical schema in the worst case, and we expect it to be even smaller in practice. Second, it is quite easy to construct a cut graph for an arbitrary polyhedral surface in $O(n)$ time, using a breadth-first search of the dual graph [14], or simply taking a maximal set of edges whose complement is connected [40]. Finally, the cut graph has an extremely simple structure: a tree with $O(g)$ additional edges. As such, it should be easier to manipulate algorithmically than other representations. For example, Dey and Schipper [14] describe fast algorithms to determine whether a curve is contractible, or two curves are homotopic, using an arbitrary cut graph instead of a canonical schema.

In this paper, we investigate the question of how find the "best" such cutting of a surface, restricting ourselves to cuts along the edges of the given mesh. Specifically, we want to find the smallest subset of edges of a polyhedral manifold surface $\mathcal{M}$, possibly with boundary, such that cutting along those edges transforms $\mathcal{M}$ into a topological disk. We also consider the weighted version of this problem, where each edge has an arbitrary non-negative weight and we want to minimize the total weight of the cut graph. The most natural weight of an edge is its Euclidean length, but we could also assign weights to take problem-specific

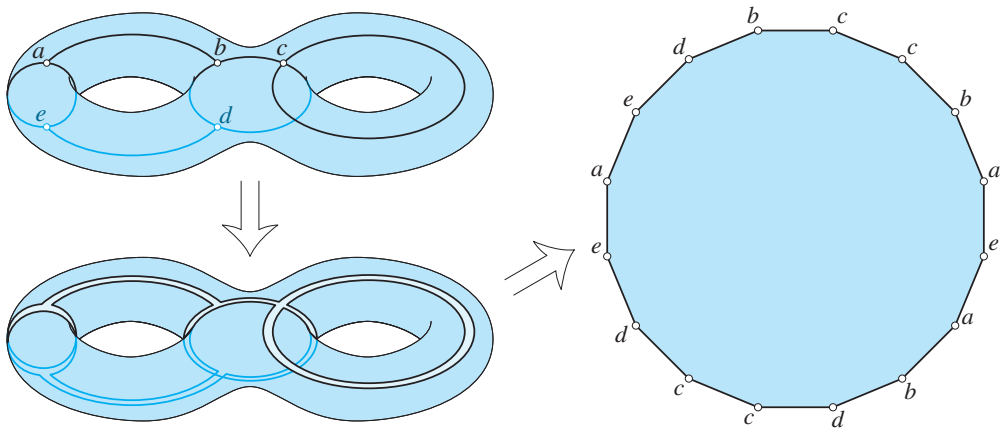

Figure 1. A cut graph for a two-holed torus and its induced (non-canonical) polygonal schema.

considerations into account. For example, if we want to compute a texture mapping for a specific viewpoint, we could make visible edges more expensive, so that the minimum cut graph would minimize the number of visible edges used in the cuts. Our algorithms do not require the edge weights to satisfy the triangle inequality.

We show that the minimum cut graph of any polyhedral manifold $\mathcal{M}$ with genus g and k boundary components can be computed in $n^{O(g+k)}$ time. This is exponential in the worst case, since there are manifolds with both $g = \Omega(n)$ and $k = \Omega(n)$. We also show that computing a minimum cut graph NP-hard, even if g or k is fixed (but not both). Finally, we present a simple and efficient greedy approximation algorithm for this problem. Our algorithm outputs a cut graph whose weight is a factor $O(\log^2 g)$ larger than optimal, in $O(g^2 n \log n)$ time.[1] If $g = 0$, the approximation factor is exactly 2. As a tool in our approximation algorithm, we also describe efficient algorithms to compute shortest and nearly-shortest nontrivial cycles in a manifold; we believe these algorithms are of independent interest.

2 Background

Before presenting our new results, we review several useful notions from topology and describe related results in more detail. We refer the interested reader to Hatcher [31], Munkres [46], or Stillwell [53] for further topological background and more formal definitions. For related computational results, see the recent surveys by Dey, Edelsbrunner, and Guha [12] and Vegter [59].

2.1 Topology

A 2-*manifold with boundary* is a set $\mathcal{M}$ such that every point $x \in \mathcal{M}$ lies in a neighborhood homeomorphic to either the plane $\mathbb{R}^2$ or a closed halfplane. The points with only halfplane neighborhoods constitute the *boundary* of $\mathcal{M}$; the boundary consists of zero or more disjoint circles. This paper will consider only *compact* manifolds, where every infinite sequence of points has a convergent subsequence.

[1] To simplify notation, we define $\log x = \max\{1, \lceil \log_2 x \rceil\}$.

The *genus* of a 2-manifold $\mathcal{M}$ is the maximum number of disjoint non-separating cycles $\gamma_1, \gamma_2, \ldots, \gamma_g$ in $\mathcal{M}$; that is, $\gamma_i \cap \gamma_j = \varnothing$ for all i and j, and $\mathcal{M} \setminus (\gamma_1 \cup \cdots \cup \gamma_g)$ is connected. For example, a sphere and a disc each have genus 0, a torus and a Möbius strip each have genus 1, and a Klein bottle has genus 2.

A manifold is *orientable* if it has two distinct sides, and *non-orientable* if it has only one side. Although many geometric applications use only orientable 2-manifolds (primarily because non-orientable manifolds without boundary cannot be embedded in $\mathbb{R}^3$ without self-intersections) our results will apply to non-orientable manifolds as well. Every (compact, connected) 2-manifold with boundary is characterized by its orientability, its genus g, and the number k of boundary components [22]. For example, the disk is the only oriented 2-manifold with genus zero and one boundary component.

A *polyhedral* 2-manifold is constructed by gluing closed simple polygons edge-to-edge into a *cell complex*: the intersection of any two polygons is either empty, a vertex of both, or an edge of both. We refer to the component polygons as *facets*. (Since the facets are closed, every polyhedral manifold is compact.) For any polyhedral manifold $\mathcal{M}$, the number of vertices and facets, minus the number of edges, is the *Euler characteristic* χ of $\mathcal{M}$. Euler's formula [19] implies that χ is an invariant of the underlying manifold, independent of any particular polyhedral representation: $\chi = 2 - 2g - k$ if the manifold is orientable, and $\chi = 2 - g - k$ if the manifold is non-orientable. Euler's formula implies that if $\mathcal{M}$ has v vertices, then $\mathcal{M}$ has at most $3v - 6 + 6g$ edges and at most $2v - 4 + 4g - k$ facets, with equality for orientable manifolds where every facet and boundary circle is a triangle. We let $n \leq 6v - 10 + 10g - k$ denote the total number of facets, edges, and vertices in $\mathcal{M}$.

The 1-*skeleton* $\mathcal{M}_1$ of a polyhedral manifold $\mathcal{M}$ is the graph consisting of its vertices and edges. We define a *cut graph* G of $\mathcal{M}$ as a subgraph of $\mathcal{M}_1$ such that $\mathcal{M} \setminus G$ is homeomorphic to a disk. (Cut graphs are generalizations of the *cut locus* of a manifold $\mathcal{M}$, which is essentially the geodesic medial axis of a single point.) The disk $\mathcal{M} \setminus G$ is known as a *polygonal schema* of $\mathcal{M}$. Each edge of G appears twice on the boundary of polygonal schema $\mathcal{M} \setminus G$, and we can obtain $\mathcal{M}$ by gluing together these corresponding boundary edges. Finding a cut graph of $\mathcal{M}$ with minimum total length is clearly equivalent to to finding a polygonal schema of $\mathcal{M}$ with minimum perimeter.

Any 2-manifold has a so-called *canonical* polygonal schema, whose combinatorial structure depends only on the genus g, the number of boundary components k, and whether the manifold is orientable.[2] The canonical schema of an orientable manifold is a $(4g + 3k)$-gon with successive edges labeled

$$x_1, y_1, \bar{x}_1, \bar{y}_1, \ \ldots, x_g, y_g, \bar{x}_g, \bar{y}_g, \ z_1, e_1, \bar{z}_1, \ \ldots, z_k, e_k, \bar{z}_k;$$

for a non-orientable manifold, the canonical schema is a $(2g + 3k)$-gon with edge labels

$$x_1, x_1, \ \ldots, x_g, x_g, \ z_1, e_1, \bar{z}_1, \ \ldots, z_k, e_k, \bar{z}_k.$$

Every pair of corresponding edges x and $\bar{x}$ is oriented in opposite directions. Gluing together corresponding pairs in the indicated directions recovers the original manifold, with

[2]There are actually several different ways to define canonical schemata; the one described here is merely the most common. For example, the canonical schema for an oriented surface without boundary could also be labeled $x_1, x_2, \ldots, x_{2g}, \bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{2g}$.

the unmatched edges e_i forming the boundary circles. For a manifold $\mathcal{M}$ without boundary, a *reduced* polygonal schema is one where all the vertices are glued into a single point in $\mathcal{M}$; canonical schemata of manifolds without boundary are reduced. We emphasize that the polygonal schemata constructed by our algorithms are neither necessarily canonical nor necessarily reduced.

2.2 Previous and Related Results

Dey and Schipper [14] describe a linear-time algorithm to construct a reduced, but not necessarily canonical, polygonal schema for any triangulated orientable manifold without boundary. Essentially, their algorithm constructs an arbitrary cut graph G by depth-first search, and and then shrinks a spanning tree of G to a single point. (See also Dey and Guha [13].)

Vegter and Yap [60] developed an algorithm to construct a canonical schema in optimal $O(gn)$ time and space. Two simpler algorithms with the same running time were later developed by Lazarus *et al.* [40]. The "edges" of the polygonal schemata produced by all these algorithms are (possibly overlapping) paths in the 1-skeleton of the input manifold. Since there are polyhedral manifolds where every canonical schema contains $\Omega(gn)$ edges, these algorithms are optimal. We will modify one of the algorithms of Lazarus *et al.* to construct short nontrivial cycles and cut graphs.

Very recently, Colin de Verdiére and Lazarus consider the problem of optimizing canonical polygonal schemata [11]. Given a canonical polygonal schema for a triangulated oriented manifold $\mathcal{M}$, their algorithm constructs the shortest canonical schema in the same homotopy class. Surprisingly (in light of our Theorem 3.1) their algorithm runs in polynomial time under some mild assumptions about the input. As a byproduct, they also obtain a polynomial-time algorithm to construct the minimum-length simple loop homotopic to a given path.

Surface parameterization is an extremely active area of research, thanks to numerous applications such as texture mapping, remeshing, compression, and morphing. For a sample of recent results, see [1, 18, 21, 52, 27, 42, 43, 49, 51, 50, 63] and references therein. In most of these works, surfaces of high genus are parameterized by cutting them into several (possibly overlapping) patches, each homeomorphic to a disk, each with a separate parameterization. A recent exception is the work of Gu *et al.* [27], which computes an initial cut graph in $O(n \log n)$ time by running a shortest path algorithm on the dual of the manifold mesh, starting from an arbitrary seed triangle. Essentially the same algorithm was independently proposed by Steiner and Fischer [52]. Once a surface has been cut into a disk (or several disks), further (topologically trivial) cuts are usually necessary to reduce distortion [27, 49, 51]. Many of these algorithms include heuristics to minimize the lengths of the cuts in addition to the distortion of the parameterization [27, 43, 49], but none with theoretical guarantees.

All of our algorithms are ultimately based on Dijkstra's single-source shortest path algorithm [15, 55]. Many previous results have used Dijkstra's algorithm or one of its continuous generalizations [35, 44, 58] to discover interesting topological structures in 2-manifolds, such as cut graphs [27, 52], small handles ('topological noise') [26], texture atlases [43], contour trees [2, 41], and Reeb graphs [32, 52].

3 Computing Minimum Cut Graphs is NP-Hard

In this section, we prove that finding a minimum cut graph of a triangulated manifold is NP-hard. We consider two versions of the problem. In the *weighted* case, the manifold is assumed to be a polyhedral surface in $\mathbb{R}^3$ and we want to compute the cut graph whose total Euclidean length is as small as possible. In the *unweighted* case, we consider the cut graph with the minimum number of edges; the geometry of the manifold is ignored entirely.

Both reductions are from the *rectilinear Steiner tree* problem: Given a set P of n points from a $m \times m$ square grid in the plane, find the shortest connected set of horizontal and vertical line segments that contains every point in P. This problem is NP-hard, even if m is bounded by a polynomial in n [24]. Our reduction uses the *Hanan grid* of the points, which is obtained by drawing horizontal and vertical lines through each point, clipped to the bounding box of the points. At least one rectilinear Steiner tree of the points is a subset of the Hanan grid [30].

We begin by considering the spacial case $g = 0$.

Theorem 3.1. *Computing the length of the minimum (weighted or unweighted) cut graph of a triangulated punctured sphere, is NP-hard.*

Proof: First consider the weighted case, where the weight assigned to each edge of the manifold is its Euclidean length. Let P be a set of n points in the plane, with integer coordinates between 1 and m. We construct, in $O(n^2)$ time, a punctured sphere as follows. Assume that P lies on the xy-plane in $\mathbb{R}^3$. We modify the Hanan grid of P by replacing each terminal with a square of width $1/2n$, rotated 45 degrees so that its vertices lie on the neighboring edges. These squares will form the punctures. We then attach a *basin* under each face f of the modified Hanan grid, by joining the boundary of f to a slightly scaled copy of f on the plane $z = -n^2$. We also attach a basin of depth $n^2 + 1$ to the boundary of the entire modified Hanan grid. The side facets of each basin are trapezoids. The basins are tapered so that adjacent basins intersect only on the modified Hanan grid. Triangulating this surface arbitrarily, we obtain a polyhedral sphere $\mathcal{M}$ with n punctures and overall complexity $O(n^2)$. See Figure 2.

Let G^* be a minimum weighted cut graph of $\mathcal{M}$. We easily observe that G^* contains only edges from the original Hanan grid and contains at least one vertex of every puncture. Thus, the edges of G^* are in one-to-one correspondence with the edge of a rectilinear Steiner tree of P.

For the unweighted case, we modify the original $m \times m$ integer grid instead of the Hanan grid. To create a punctured sphere, we replace each terminal point with a small diamond as above. We then fill in each modified grid cell with a triangulation, chosen so that the shortest path between any two points on the boundary of any cell stays on the boundary of that cell; this requires a constant number of triangles per cell. The resulting manifold $\mathcal{M}'$ has complexity $O(m^2)$. By induction, the shortest path between any two points on the modified grid lies entirely on the grid. Thus, any minimal unweighted cut graph of $\mathcal{M}'$ contains only edges from the modified grid. It follows that if the minimum unweighted cut graph of $\mathcal{M}'$ has r edges, the length of any rectilinear Steiner tree of P is exactly r. □

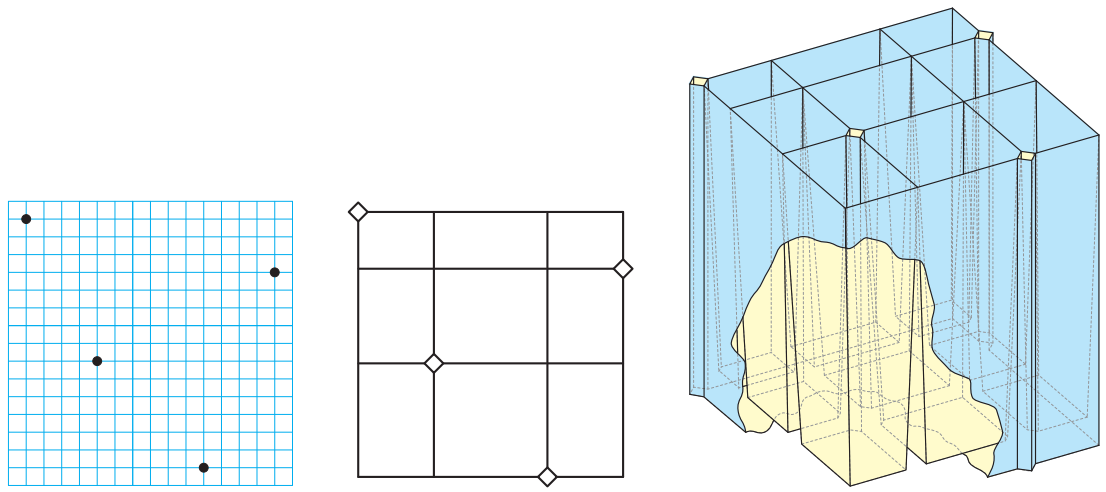

Figure 2. (a) A set of integer points. (b) The modified Hanan grid. (c) A cut-away view of the resulting punctured sphere.

We can easily generalize the previous proof to manifolds with higher genus, with or without boundary, oriented or not, by attaching small triangulated tori or cross-caps to any subset of punctures.

Theorem 3.2. *Computing the length of the minimum (weighted or unweighted) cut graph of a triangulated manifold with boundary is NP-hard, even if the genus or the number of boundary components (but not both) is fixed.*

4 Computing Minimum Cut Graphs Anyway

We now describe an algorithm to compute the minimum cut graph of a polyhedral manifold in $n^{O(g+k)}$ time. For manifolds with constant Euler characteristic, our algorithm runs in polynomial time.

Our algorithm is based on the following characterization of the minimum cut graph as the union of shortest paths. A *branch point* of a cut graph is any vertex with degree greater than 2. A simple path in a cut graph from one branch point or boundary point to another, with no branch points in its interior, is called a *cut path*.

Our minimum cut graph algorithm exploits Lemma 4.1 by composing potential minimum cut graphs out of $O(g+k)$ shortest paths. Unfortunately, a single pair of nodes in $\mathcal{M}$ could be joined by $2^{\Omega(n)}$ shortest paths, in $2^{\Omega(g+k)}$ different isotopy classes, in the worst case. To avoid this combinatorial explosion, we can add a random infinitesimal weight $\varepsilon \cdot w(e)$ to each edge e. The Isolation Lemma of Mulmuley, Vazirani, and Vazirani [45] implies that if the weights $w(e)$ are chosen independently and uniformly from the integer set $\{1, 2, \ldots, n^2\}$, all shortest paths are unique with probability at least $1 - 1/n$; see also [10, 37].[3] *In the*

[3]Alternately, if we choose $w(e)$ uniformly from the *real* interval $[0, 1]$, shortest paths are unique with probability 1. This may sound unreasonable, but recall that no polynomial-time algorithm is known to compare sums of square roots of integers in any model of computation that does not include square root

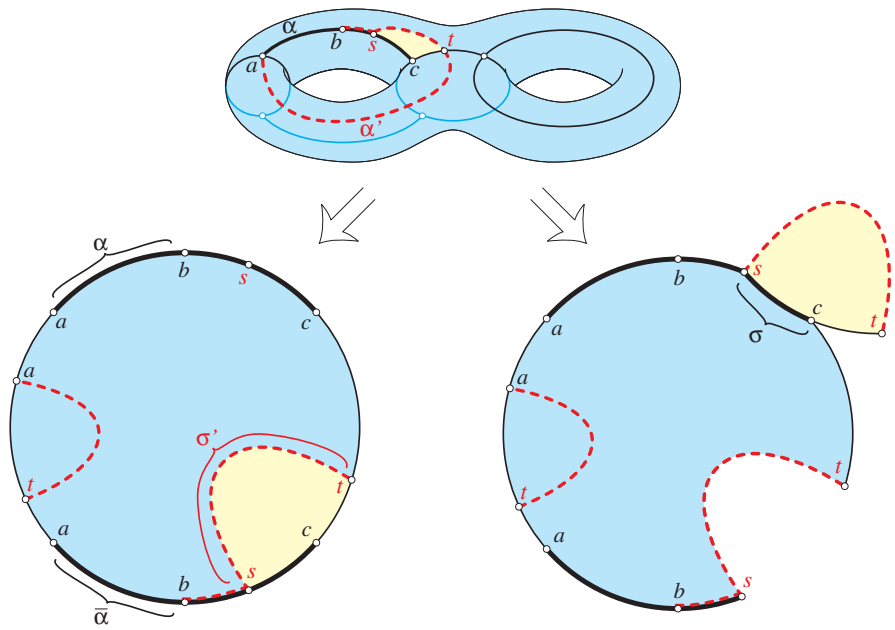

Figure 3. If the dashed path α' from a to b is shorter than the bold path α from a to b, then the cut graph can be shortened by cutting along σ' and regluing along σ.

remainder of the paper, we assume that shortest paths in the 1*-skeleton of any polyhedral manifold are unique.*

Lemma 4.1. *Let $\mathcal{M}$ be a polyhedral* 2*-manifold, possibly with boundary, and let G^* be a minimum cut graph of $\mathcal{M}$. Any cut path in G^* can be decomposed into two equal-length shortest paths in $\mathcal{M}_1$ whose common endpoint may lie in the middle of an edge.*

Proof: Let G be an arbitrary cut graph of $\mathcal{M}$, and consider a cut path between two (not necessarily distinct) branch points a and c of G. Let b be the midpoint of this path, and let α and β denote the subpaths from b to a and from b to c, respectively. Note that b may lie in the interior of an edge of $\mathcal{M}_1$. Finally, suppose α is *not* the shortest path from b to a in $\mathcal{M}_1$. To prove the lemma, it suffices to show that G is not the shortest cut graph of $\mathcal{M}$.

Let α' be the true shortest path from b to a. Clearly, α' is not contained in G. Walking along α' from b to a, let s be the first vertex whose following edge is not in G, and let t be the first vertex in G whose preceding edge is not in G. (Note that s and t may be joined by a single edge in $\mathcal{M} \setminus G$.) Finally, let $\sigma' \subset \alpha'$ be the true shortest path from s to t. Equivalently, σ' is the first maximal subpath of α' whose interior lies in $\mathcal{M} \setminus G$. See Figure 3.

The subpath σ' cuts $\mathcal{M} \setminus G$ into two smaller disks. We claim that some subpath σ of either α or β appears on the boundary of both disks and is longer than σ'. Our claim implies that cutting $\mathcal{M} \setminus G$ along σ' and regluing the pieces along σ gives us a new polygonal schema with smaller perimeter, and thus a new cut graph shorter than G. See Figure 3 for an example.

as a primitive operation [8]. Thus, to compute Euclidean shortest paths in a geometric graph with integer vertex coordinates, we must either assume exact real arithmetic or (grudgingly) accept some approximation error [23].

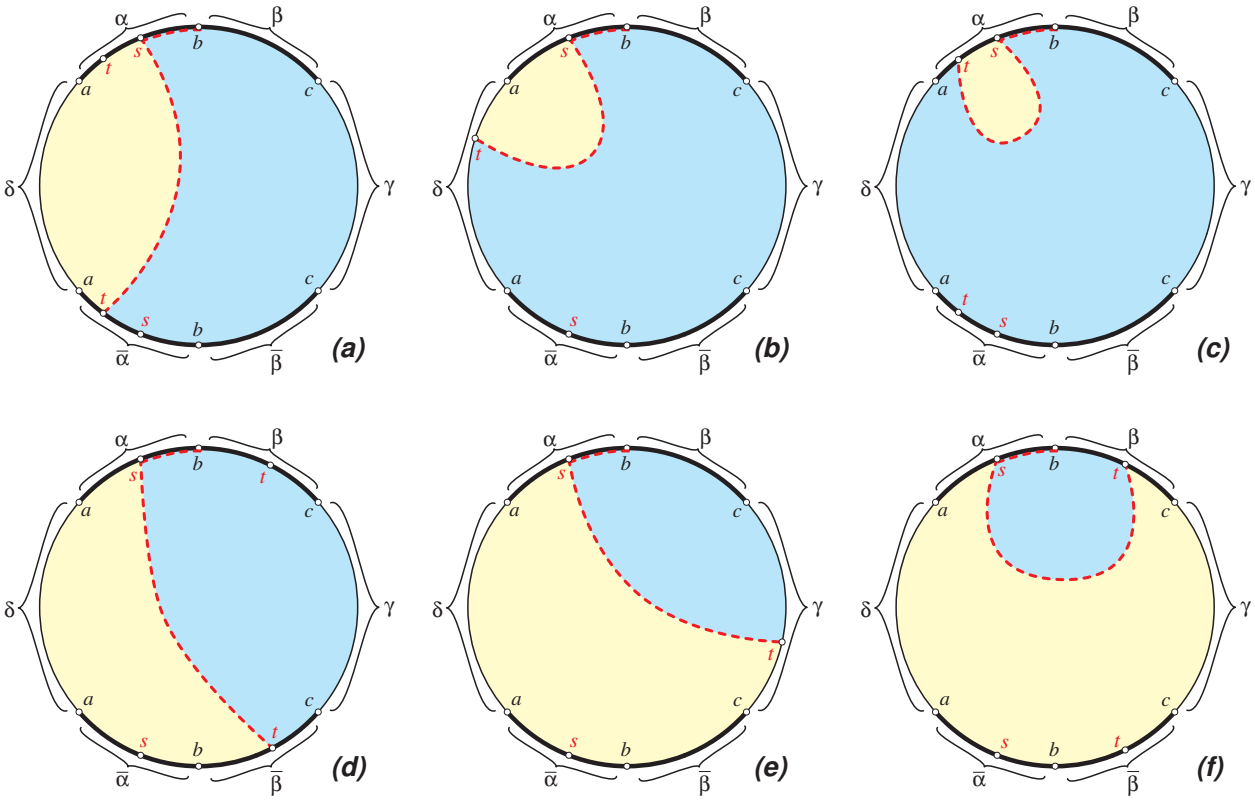

Figure 4. Six cases for the proof of Lemma 4.1 for orientable manifolds; all other cases are reflections of these. The dashed path is σ'. In each case, some subpath σ of α or β appears on the boundary of both sub-disks.

We prove our claim by exhaustive case analysis. First consider the case where the manifold $\mathcal{M}$ is orientable. We can subdivide the entire boundary of the disk $\mathcal{M} \setminus G$ into six paths labeled consecutively $\alpha, \beta, \gamma, \bar{\beta}, \bar{\alpha}, \delta$. Here, $\bar{\alpha}$ and $\bar{\beta}$ are the corresponding copies of α and β in the polygonal schema. Because $\mathcal{M}$ is orientable, α and $\bar{\alpha}$ have opposite orientations, as do β and $\bar{\beta}$. Either or both of γ and δ could be empty. See the lower left part of Figure 3. The subpath σ' can enter the interior of the disk $\mathcal{M} \setminus G$ from four of these six paths (α, β, $\bar{\alpha}$, and $\bar{\beta}$) and leave the interior of the disk through any of the six paths.

Suppose σ' enters the interior of $\mathcal{M} \setminus G$ from α; the other three cases are symmetric. Figure 4 shows the six essentially different ways for σ' to leave the interior of $\mathcal{M} \setminus G$. In each case, we easily verify that after cutting along σ', some subpath σ of either α or β is on the boundary of both disks. Let $\cdot$ denotes path concatenation, and let $\alpha[x, y]$ denote the subpath of path α from point x to point y.

(a) If σ' leaves through $\bar{\alpha}$, we can take $\sigma = \alpha[s, t]$. Since σ and σ' have the same endpoints and σ' is a shortest path, σ' must be shorter than σ.

(b) If σ' leaves through δ, we can take $\sigma = \alpha[s, a]$. Since $|\alpha[b, s] \cdot \sigma'| < |\alpha|$, we have $|\sigma'| < |\alpha[s, a]| = |\sigma|$.

(c) If σ' leaves through α, we can take $\sigma = \alpha[s, t]$. We argue that $|\sigma'| < |\sigma|$ as in case (a).

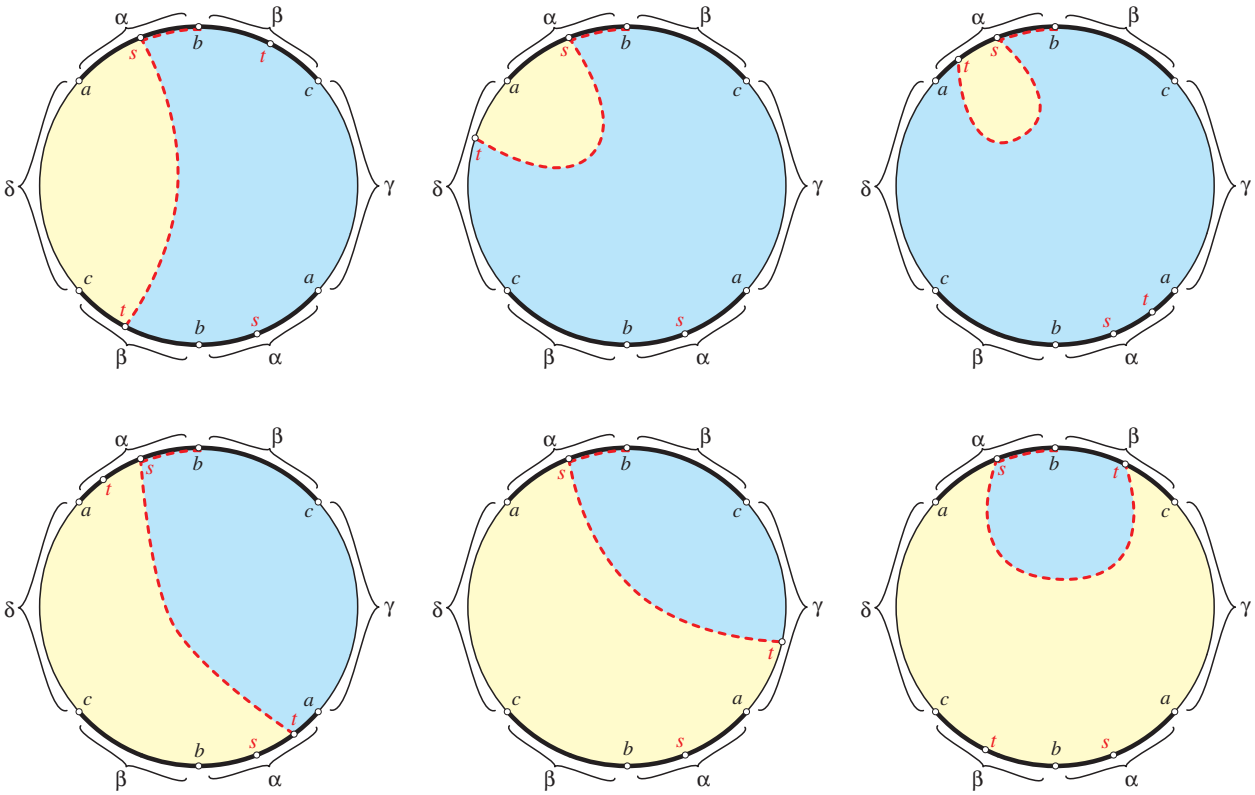

Figure 5. Six additional cases for the proof of Lemma 4.1 for non-orientable manifolds; all other cases are reflections or rotations of these. In each case, some subpath of α or β appears on the boundary of both sub-disks.

(d) If σ' leaves through $\bar{\beta}$, we can take $\sigma = \beta[b, t]$. In this case, $\rho = \alpha[b, s] \cdot \sigma'$ is the shortest path from b to t, which implies that $|\sigma'| \leq |\rho| < |\beta[b, t]| = |\sigma|$.

(e) If σ' leaves through γ, we can take $\sigma = \beta$. Clearly, $|\sigma'| < |\alpha| = |\beta| = |\sigma|$.

(f) Finally, if σ' leaves through β, we can take $\sigma = \beta[b, t]$. We argue that $|\sigma'| < |\sigma|$ as in case (d).

If $\mathcal{M}$ is non-orientable, the path $\alpha\beta$ could appear either with the same orientation or with opposite orientations on the boundary of the disk $\mathcal{M} \setminus G$. If the orientations are opposite, the previous case analysis applies immediately. Otherwise, the boundary can be subdivided into six paths labeled consecutively $\alpha, \beta, \gamma, \alpha, \beta, \delta$. Without loss of generality, σ' enters the interior of $\mathcal{M} \setminus G$ from α and leaves through any of these six paths. The six cases are illustrated in Figure 5. Again, we easily verify that in each case, some subpath σ of either α or β is on the boundary of both disks. We omit further details. □

For any cut graph G of a manifold $\mathcal{M}$, we define the corresponding *reduced* cut graph $\hat{G}$ as follows. First we remove any topologically trivial cuts; that is, we repeatedly remove any edge with a vertex of degree 1 that is not on the boundary of $\mathcal{M}$. We then augment the cut graph by adding all the boundary edges of $\mathcal{M}$. Finally, we contract each maximal path through degree-2 vertices into a single edge. The resulting reduced cut graph $\hat{G}$ is 2-edge connected, and each of its vertices has degree at least 3. Every vertex of $\hat{G}$ is either a branch

point or a boundary point of G, and every edge of $\hat{G}$ corresponds to either a cut path or a boundary path in G. However, in general, not all branch points and cut paths in G are represented in $\hat{G}$.

Lemma 4.2. *Let $\mathcal{M}$ be a polyhedral 2-manifold with genus g and k boundary components. Any reduced cut graph $\hat{G}$ of $\mathcal{M}$ is connected, has between $\max\{1, k\}$ and $4g+2k-2$ vertices, and has between $g + \max\{0, 2k-1\}$ and $6g + 3k - 3$ edges.*

Proof: Let G' be the cut graph corresponding to $\hat{G}$, after all the trivial cuts have been removed. The boundary of the polygonal schema $\mathcal{M} \setminus G'$ can be partitioned into cut paths and boundary paths, each corresponding to an edge in $\hat{G}$. Thus, $\hat{G}$ is connected.

Let v and e denote the number of vertices and edges in $\hat{G}$, respectively. If any vertex in $\hat{G}$ has degree $d \geq 4$, we can replace it with $d-2$ trivalent vertices and $d-3$ new edges of length zero. Thus, in the worst case, every vertex in $\hat{G}$ has degree exactly 3, which implies that $3v = 2e$. Since $\hat{G}$ is embedded in $\mathcal{M}$ with a single face, Euler's formula implies that $v - e + 1 = \chi = 2 - 2g - k$ if $\mathcal{M}$ is orientable, and $v - e + 1 = \chi = 2 - g - k$ if $\mathcal{M}$ is non-orientable. It follows that $v \leq 4g + 2k - 2$ and $e \leq 6g + 3k - 3$, as claimed.

On the other hand, $\hat{G}$ has at least one vertex on each boundary component of $\mathcal{M}$, and at least one vertex even if $\mathcal{M}$ has no boundary, so $v \geq \max\{1, k\}$. Thus, Euler's formula implies that $2 - 2g - k = v - e + 1 \geq \max\{1, k\} - e + 1$ if $\mathcal{M}$ is orientable, or equivalently, $e \geq 2g + \max\{0, 2k-1\}$. Similarly, if $\mathcal{M}$ is non-orientable, Euler's formula implies that $e \geq g + \max\{0, 2k-1\}$. □

We are now finally ready to describe our minimum cut graph algorithm.

Theorem 4.3. *The minimum cut graph of a polyhedral 2-manifold $\mathcal{M}$ with genus g and k boundary components can be computed in time $n^{O(g+k)}$.*

Proof: We begin by computing the shortest path between every pair of vertices in $\mathcal{M}$ in $O(n^2 \log n)$ time by running Dijkstra's single-source shortest path algorithm for each vertex [15, 34], breaking ties using random infinitesimal weights as described above. Once these shortest paths and midpoints (i.e., a midpoint is the point on the edge where the wavefront collide from both sides) have been computed, our algorithm enumerates by brute force every possible cut graph that satisfies Lemmas 4.1 and 4.2, and returns the smallest such graph.

Each cut graph is specified by a set V of up to $4g + 2k - 2$ vertices of $\mathcal{M}$, a set E of up to $6g + 3k - 3$ edges of $\mathcal{M}$, a trivalent multigraph $\hat{G}$ with vertices V, and an assignment of edges in E to edges in $\hat{G}$. E is the subset of edges of the manifold that contain midpoints of cut paths in the cut graph; each edge (v, w) of $\hat{G}$ is assigned a unique edge $e \in E$ to define the corresponding cut path in $\mathcal{M}$. This cut path is the concatenation of the shortest path from v to e, e itself, and the shortest path from e to w. If the midpoint of this cut path is not in the interior of e, we declare the cut path invalid, since it violates Lemma 4.1. (Because shortest paths between vertices are unique, the midpoint of any cut path in the minimal cut graph must lie in the interior of an edge.) If all the cut paths are valid, we then check that every pair of cut paths is disjoint, except possibly at their endpoints, and that removing all the cut paths from $\mathcal{M}$ leaves a topological disk.

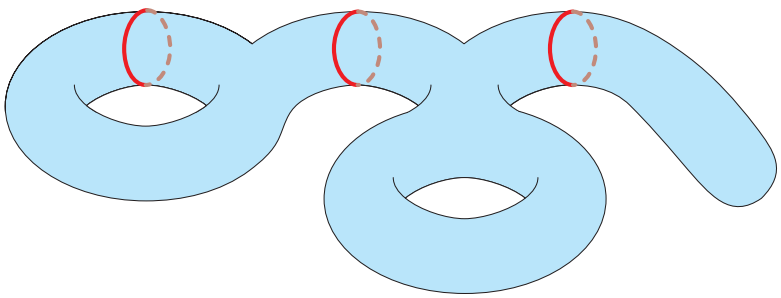

Figure 6. From left to right: non-separating, essential but separating, and trivial cycles on a 2-manifold.

Our brute-force algorithm considers $O(n^{4g+2k-2})$ different vertex sets V, $O(n^{6g+3k-2})$ different edge sets E, at most

$$\binom{(4g+2k-2)^2}{6g+3k-2} < (4g+2k-2)^{2(6g+3k-2)} < (6n)^{2(6g+3k-2)} = n^{O(g+k)}$$

different graphs $\hat{G}$ for each vertex set, and at most $(6g+3k-2)!$ different assignments of edges in each graph $\hat{G}$ to edges in each edge set E. Thus, $n^{O(g+k)}$ potential cut graphs are considered altogether. The validity of each potential cut graph can be checked in $O(n)$ time. □

5 Finding Short Nontrivial Cycles

As a step toward efficiently computing approximate minimum cut graphs, we develop algorithms to compute shortest and nearly-shortest nontrivial cycles in arbitrary 2-manifolds, possibly with boundary. Although our most efficient approximation algorithm for cut graphs requires only approximate shortest nontrivial cycles on manifolds without boundary, we believe these algorithms are of independent interest.

We distinguish between two type of nontrivial simple cycles. A simple cycle γ in $\mathcal{M}$ is *non-separating* if $\mathcal{M} \setminus \gamma$ has only one connected component. A simple cycle γ in $\mathcal{M}$ is *essential* if it is not contractible to a point or a single boundary cycle of $\mathcal{M}$. Every non-separating cycle is essential, but the converse is not true. Formally, non-separating cycles are *homologically* nontrivial, and essential cycles are *homotopically* nontrivial. See Figure 6.

As can be seen immediately from Figure 6, it is not possible to determine whether a cycle is non-separating, essential, or trivial by examining only a local neighborhood. Dey and Schipper [14] describe an algorithm to determine whether an arbitrary circuit is contractible in $O(n)$ time; their algorithm begins by computing an arbitrary cut graph of the manifold. Fortunately, as we shall see shortly, we can simplify their algorithm considerably when the given cycle is simple.

5.1 Shortest Cycles

We begin by describing how to find the shortest nontrivial cycle through a given vertex. Our algorithm uses a combination of Dijkstra's single-source shortest path algorithm [15]

and a modification of the canonical polygonal schema algorithm of Lazarus *et al.* [40]. Our algorithm is similar to the approach taken by Guskov and Wood [26] to find and remove small handles from geometric models reconstructed from noisy data; see also [62].

The algorithm of Lazarus *et al.* builds a connected subset S of a triangulated manifold without boundary, starting with a single triangle and adding new triangles on the boundary of S one at a time. If a new triangle intersects the boundary of S in more than one component, the algorithm checks which of the following three cases holds: (1) $\mathcal{M} \setminus S$ is connected; (2) neither component of $\mathcal{M} \setminus S$ is a disk; or (3) one component of $\mathcal{M} \setminus S$ is a disk. In the final case, the algorithm adds the disk component to S and continues searching the other component of $\mathcal{M} \setminus S$. If we run this algorithm until either case (1) or case (2) holds, the total running time is $O(n)$. See Lazarus *et al.* [40] for further details. Similar approaches have been used for mesh compression by Gumhold and Straßer [25], Taubin and Rossignac [56, 48], Isenburg and Snoeyink [33], and others.

First we describe a straightforward generalization of the algorithm used by Lazarus to determine the structure of $\mathcal{M} \setminus S$.

Lemma 5.1. *Let $\mathcal{M}$ be a connected polyhedral 2-manifold $\mathcal{M}$ whose genus g and number of boundary components k are known. Any simple cycle γ in $\mathcal{M}_1$ can be classified as non-separating, essential but separating, or trivial in $O(m)$ time, where m is the complexity of the smaller component of $\mathcal{M} \setminus \gamma$. In particular, if γ is non-separating, the running time is $O(n)$.*

Proof: We perform two simultaneous depth-first searches, starting from either side of any vertex of γ. If either search meets a vertex already visited by the other, γ is a non-separating cycle. The running time in this case is trivially $O(n)$.

Conversely, if one search halts without meeting the other, γ is a separating cycle. Let C be the smaller component of $\mathcal{M} \setminus \gamma$. To determine whether γ is essential, we compute the Euler characteristic and the number of boundary components of C, which we denote by $\chi(C)$ and $k(C)$, respectively. (This can be done on the fly during the depth-first search phase.) Let χ denote the Euler characteristic of $\mathcal{M}$. The cycle γ is contractible if and only if one of the following conditions holds; otherwise, γ is essential.

- C is a disk, or equivalently, $\chi(C) = 1$.
- C is an annulus, or equivalently, $\chi(C) = 0$ and $k(C) = 2$.
- $\mathcal{M} \setminus C$ is a disk, or equivalently, $\chi(C) = \chi - 1$. The equivalence follows from the inclusion-exclusion formula $\chi(A \cup B) = \chi(A) + \chi(B) - \chi(A \cap B)$.
- $\mathcal{M} \setminus C$ is an annulus, or equivalently, $\chi(C) = \chi(\mathcal{M})$ and $k(C) = k(M)$.

Let m denote the number of features in C. The depth-first search of C examines edges of γ that lie on the boundary of C, but not technically part of C itself. We easily observe that C has at most $3m$ boundary edges. Thus, if γ is a separating cycle, the total running time of our algorithm is $O(m)$. □

Note that we can simplify this algorithm slightly if the manifold has no boundary, since in that case neither component of $\mathcal{M} \setminus \gamma$ is an annulus. We will use this simplification in our approximate cut graph algorithm.

Lemma 5.2. *Let u be a vertex of a polyhedral 2-manifold $\mathcal{M}$, possibly with boundary. The shortest essential cycle in $\mathcal{M}_1$ that contains u can be computed in $O(n \log n)$ time.*

Proof: We find the shortest essential cycle through u by simulating a circular wave expanding from u. Whenever the wave touches itself, either we have the shortest essential cycle through u, or one component of the wave bounds a disk in $\mathcal{M}$ and we can continue expanding the other component.

We modify the algorithm of Lazarus *et al.* [40] as follows. First, S is no longer a set of triangles but a more general connected subset of vertices, edges, and facets of $\mathcal{M}$. Initially, S contains only the source vertex u. Second, we use Dijkstra's algorithm to determine the order for edges to be added. We add a facet to S only when all its vertices have been added to S, either directly or as part of another facet. We run the algorithm from Lemma 5.1 whenever the topology of S changes, that is, when we add a new edge vw with both endpoints on the boundary of S. If $\mathcal{M} \setminus S$ is disconnected, we continue only if one component of $\mathcal{M} \setminus S$ is a disk *or an annulus*, as can be checked using Lemma 5.1. In that case, we add the disk or annulus component of $\mathcal{M} \setminus S$ to S, discard the vertices of that component from the Dijkstra priority queue, and continue searching in the other component. If $\mathcal{M} \setminus S$ is connected, or if neither component is a disk or an annulus, we have found the shortest essential cycle through u, consisting of the shortest path from u to v, the edge vw, and the shortest path from w to u.

Dijkstra's algorithm requires $O(n \log n)$ time. Each time we find a trivial cycle, we spend $O(m)$ time and discard a disk with complexity at least m. Thus, the total time spent performing cycle classification and maintaining the wavefront set S is $O(n)$. Thus, the total running time of our algorithm is $O(n \log n)$. □

Running this algorithm once for every vertex of $\mathcal{M}$ immediately gives us the shortest essential cycle.

Corollary 5.3. *Let $\mathcal{M}$ be a polyhedral 2-manifold, possibly with boundary. The shortest essential cycle in $\mathcal{M}_1$ can be computed in $O(n^2 \log n)$ time.*

A simple modification of our algorithm allows us to find shortest *non-separating* cycles in the same asymptotic time.

Lemma 5.4. *Let u be a vertex of a polyhedral 2-manifold $\mathcal{M}$, possibly with boundary. The shortest non-separating cycle in $\mathcal{M}_1$ that contains u can be computed in $O(n \log n)$ time.*

Proof: The only change from the previous algorithm is that if we discover an essential separating cycle, we continue recursively in both components of $\mathcal{M} \setminus S$. The cost of Dijkstra's algorithm is still $O(n \log n)$, but we now must spend extra time in the cycle-classification algorithm of Lemma 5.1. As before, the total time spent finding trivial cycles is $O(n)$, since we can charge the search time to the discarded components.

Let $T(n, g)$ denote the total time spent finding separating essential cycles in a manifold of complexity n, excluding features that have already been added to S, and genus g. This function satisfies the recurrence

$$T(n, g) \leq T(m, h) + T(n - m, g - h) + O(m),$$

where $m \leq n/2$ is the complexity of the smaller component of $\mathcal{M} \setminus S$ and h is its genus. The base case of the recurrence is $T(n, 1) = 0$, since every essential cycle on a genus-1 surface is non-separating.

Similar recurrences appear in the analysis of output-sensitive planar convex hull algorithms [6, 9, 36, 61], suggesting that the solution to our recurrence is $T(n, g) = O(n \log g)$. Indeed, we can prove this by induction as follows. Suppose

$$T(n, g) \leq T(m, h) + T(n - m, g - h) + cm$$

for some constant c. We claim that $T(n, g) \leq cn \lg g$. The inductive hypothesis implies that

$$\begin{aligned} T(n, g) &\leq cm \lg h + c(n - m) \lg(g - h) + cm \\ &\leq \max_{1 \leq h \leq g-1} (cm \lg h + c(n - m) \lg(g - h)) + cm. \end{aligned}$$

A simple application of derivatives implies that the right hand side of this inequality is maximized when $h = mg/n$. Thus,

$$\begin{aligned} T(n, g) &\leq cm \lg \frac{mg}{n} + c(n - m) \lg \frac{(n - m)g}{n} + cm \\ &= cn \lg g + cm \lg m + c(n - m) \lg(n - m) - cn \lg n + cm. \end{aligned}$$

Since $m \leq n/2$ and $n - m \leq n$, we can simplify this inequality to

$$T(n, g) \leq cn \ln g + cm \lg(n/2) + c(n - m) \lg n - cn \lg n + cm = cn \lg g,$$

completing the proof.

Thus, the total time spent in the cycle-classification phase of our algorithm is $O(n \log g)$. Since $g \leq n$, this is dominated by the cost of maintaining the Dijkstra priority queue. □

Corollary 5.5. *Let $\mathcal{M}$ be an polyhedral 2-manifold, possibly with boundary. The shortest non-separating cycle in $\mathcal{M}_1$ can be computed in $O(n^2 \log n)$ time.*

5.2 Nearly-Shortest Cycles

As we will argue in the next section, computing short nontrivial cycles is the bottleneck in our approximate cut graph algorithm. Fortunately, exact minimum cycles are not necessary for our results. We can speed up our cut graph algorithm, without significantly increasing the approximation factor, by searching for a nontrivial cycle at most twice as long as the shortest. Our approximation algorithm assumes that the manifold $\mathcal{M}$ has no boundary; fortunately, as we shall see in the next section, this is sufficient for our purposes.

Our approximation algorithm works as follows. First, we compute a set of shortest paths (in fact, a cut graph) that intersects every non-separating cycle in the manifold $\mathcal{M}$. Then we contract each shortest path π in this set to a point and find the shortest nontrivial cycle through that point, as described by Lemmas 5.2 and 5.4.

Lemma 5.6. *Let π be a shortest path between two vertices in a polyhedral 2-manifold $\mathcal{M}$, and let γ^* be the shortest essential (resp. non-separating) cycle in $\mathcal{M}_1$ that intersects π. In $O(n \log n)$ time, one can compute an essential (resp. non-separating) cycle γ in $\mathcal{M}$ such that $|\gamma| \leq 2|\gamma^*|$.*

Proof: Let $\mathcal{M}'$ be the manifold obtained by contracting the shortest path π to a single vertex v. Because π has no cycles, $\mathcal{M}'$ has the same topological type as $\mathcal{M}$. Let γ' be the shortest essential (resp. non-separating) cycle in $\mathcal{M}'$ that passes through v. Clearly, $|\gamma'| \leq |\gamma^*|$. We can compute this cycle in $O(n \log n)$ time by Lemma 5.2 (resp. Lemma 5.4).

We construct a cycle γ in $\mathcal{M}$ by concatenating two paths α and β, where α contains the edges of γ' and β is the subpath of π between the endpoints of α. The sequence of edge contractions that transforms $\mathcal{M}$ to $\mathcal{M}'$ also transforms γ' to γ. Hence, γ' is an essential cycle of $\mathcal{M}$. Because β is a subpath of a shortest path, β is actually the shortest path between the endpoints of α, so $|\beta| \leq |\alpha| = |\gamma'|$. It follows that $|\gamma| = |\alpha| + |\beta| \leq 2|\gamma'| \leq 2|\gamma^*|$. □

This lemma suggests a natural algorithm for finding a short nontrivial cycle: Compute a set of shortest paths that intersect every essential cycle (and thus every non-separating cycle), and then run the algorithm from Lemma 5.6 for each path in this set.

Lemma 5.7. *Let $\mathcal{M}$ be a polyhedral 2-manifold without boundary. In $O(n \log n)$ time, one can compute a set Π of $O(g)$ shortest paths on $\mathcal{M}_1$ such that every non-separating cycle (and thus every essential cycle) in $\mathcal{M}_1$ intersects at least one path in Π.*

Proof: We compute a cut graph G as follows. First we compute a shortest-path tree T from an arbitrary initial vertex v using Dijkstra's algorithm. We then compute an arbitrary spanning tree T^* of the dual of $\mathcal{M} \setminus T$, that is, the graph whose vertices are facets of $\mathcal{M}$ and whose edges join pairs of facets that share a common edge *not* in T. Analysis similar to Lemma 4.2 implies that there are $O(g)$ edges that do not appear in T and whose dual edges do not appear in T^*. Call this set of unclaimed edges E. Let Π be the set of $O(g)$ shortest paths from v to the endpoints of E; these paths are all in T. Finally, let $G = \Pi \cup E$.

We easily observe that $\mathcal{M} \setminus G$ is a topological disk, so G is a cut graph. It follows that every non-separating cycle in $\mathcal{M}_1$ intersects G. Since every vertex of G is also a vertex of some path in Π, every non-separating cycle in $\mathcal{M}_1$ intersects at least one path in Π. □

Notice that this algorithm does not work if $\mathcal{M}$ has a boundary, since the dual graph of $\mathcal{M} \setminus T$ could be disconnected.

Corollary 5.8. *Let $\mathcal{M}$ be a polyhedral 2-manifold with genus g and no boundary, and let γ^* be its shortest essential (resp. non-separating) cycle. In $O(gn \log n)$ time, one can compute an essential (resp. non-separating) cycle γ in $\mathcal{M}_1$ such that $|\gamma| \leq 2|\gamma^*|$.*

Proof: We construct a set Π of $O(g)$ shortest paths, at least one of which is guaranteed to intersect γ^*, as described in the previous lemma. Then for each path $\pi \in \Pi$, we contract π to a point and find the shortest nontrivial cycle through that point in $O(n \log n)$ time, as described by Lemma 5.6. □

6 Approximate Minimum Cut Graphs

We now describe a simple polynomial-time greedy algorithm to construct an approximate minimum cut graph for any polyhedral manifold $\mathcal{M}$.

To handle manifolds with boundary, it will be convenient to consider the following simplified form. Given a manifold $\mathcal{M}$ with genus g and k boundary components, the corresponding *punctured* manifold $(\overline{\mathcal{M}}, P)$ consists of a manifold $\overline{\mathcal{M}}$ with the same genus as $\mathcal{M}$ but without boundary, and a set P of k points in $\overline{\mathcal{M}}$, called *punctures*. To construct $\overline{\mathcal{M}}$, we contract every boundary component of $\mathcal{M}$ to a single point, which becomes one of the punctures in P. (We could simulate this contraction by artificially assigning every boundary edge of $\mathcal{M}$ a weight of zero, although this would require a few simple changes in our algorithms.) If any vertex of $\mathcal{M}$ has multiple edges to the same boundary component, $\overline{\mathcal{M}}$ contains only the edge with smallest weight, breaking ties using the Isolation Lemma as above. If $\mathcal{M}$ has no boundary, then $\overline{\mathcal{M}} = \mathcal{M}$ and $P = \varnothing$.

Our goal now is to compute the minimum cut graph of $\overline{\mathcal{M}}$ that touches every puncture in P; henceforth, we call this simply the minimum cut graph of $(\overline{\mathcal{M}}, P)$. This reduction is motivated by the following trivial observation.

Lemma 6.1. *The minimum cut graph of any polyhedral 2-manifold $\mathcal{M}$ has the same length as the minimum cut graph of $(\overline{\mathcal{M}}, P)$.*

Our approximation algorithm works as follows. We repeatedly cut along short nontrivial cycles until our surface becomes a collection of punctured spheres, connect the punctures on each component by cutting along a minimum spanning tree, and finally (if necessary) re-glue some previously cut edges to obtain a single disk. The resulting cut graph is composed of a subset of the edges of the short nontrivial cycles and all the edges of the minimum spanning forest.

6.1 Using Short Nontrivial Cycles

The first component of our algorithm is a subroutine to compute approximately shortest nontrivial cycles, described by Corollary 5.8. As required by that algorithm, the input manifold $\overline{\mathcal{M}}$ has no boundary; the punctures are completely ignored.

The following argument relates the length of the shortest nontrivial cycles to the length of the minimum cut graph.

Lemma 6.2. *Let G be any cut graph of a polyhedral 2-manifold $\overline{\mathcal{M}}$ with genus g and no boundary. The shortest cycle in G contains $O((\log g)/g)$ of the total length of G.*

Proof: First consider the reduced cut graph $\hat{G}$, constructed by repeatedly contracting any edge with a vertex of degree less than three, as in Section 4. Every vertex in $\hat{G}$ has degree at least 3. Without loss of generality, assume that every vertex in $\hat{G}$ has degree exactly 3, splitting each high-degree vertex into a tree of degree-3 vertices if necessary, as in the proof of Lemma 4.2. A straightforward counting argument implies that any trivalent graph whose girth (minimum cycle length) is c must have at least $3\sqrt{2} \cdot 2^{c/2} - 2$ vertices if c is odd, and

at least $2 \cdot 2^{c/2} - 2$ vertices if c is even [7]. By Lemma 4.2, $\hat{G}$ has at most $4g - 2$ vertices, so $\hat{G}$ must have a cycle $\hat{\gamma}$ with most $2(\lg g + 1) = O(\log g)$ edges.

Starting with $\hat{G}_0 = \hat{G}$, we inductively define a sequence of reduced graphs $\hat{G}_1, \hat{G}_2, \ldots$ as follows. For each $i > 0$, let $\hat{\gamma}_i$ denote the shortest cycle in $\hat{G}_{i-1}$. We obtain $\hat{G}_i$ by reducing the graph $\hat{G}_{i-1} \setminus \hat{\gamma}_i$, or equivalently, removing the vertices of $\hat{\gamma}_i$ and all their edges, and then contracting $|\hat{\gamma}_i|$ nearby length-2 paths to single edges. Our earlier argument implies that each cycle $\hat{\gamma}_i$ has at most $2(\lg g + 1)$ edges. Thus, for each i, we have $|E(\hat{G}_i)| = |E(\hat{G}_{i-1})| - 6(\lg g + 1)$. Lemma 4.2 implies that the original reduced cut graph $\hat{G}$ has at least g edges, so we can repeat this process at least $g/6(\lg g + 1)$ times.

Let γ_i denote the cycle in the original cut graph G corresponding to $\hat{G}_i$. By our construction, γ_i and γ_j are disjoint for all $i \neq j$, so we have a set of at least $g/6(\lg g + 1)$ disjoint cycles in G. At least one of these cycles has length at most $6(\lg g + 1)/g = O((\log g)/g)$ times the total length of G. □

Since every cycle in the minimum cut graph is non-separating, and therefore essential, we immediately have the following corollary.

Corollary 6.3. *For any polyhedral 2-manifold $\overline{\mathcal{M}}$ with genus g and no boundary, both the length of the shortest non-separating cycle and the length of the shortest essential cycle are at most $O((\log g)/g)$ times the length of the minimum cut graph of $\overline{\mathcal{M}}$.*

6.2 Puncture-Spanning Trees

The second component of our cut graph algorithm is a subroutine to compute the *minimum puncture-spanning tree* of a punctured manifold $(\overline{\mathcal{M}}, P)$, that is, the minimum spanning tree of the punctures P in the shortest-path metric of $\overline{\mathcal{M}}_1$.

Lemma 6.4. *The minimum puncture-spanning tree of any punctured polyhedral 2-manifold $(\overline{\mathcal{M}}, P)$ can be computed in $O(n \log n)$ time.*

Proof: We simulate Kruskal's minimum spanning tree algorithm [39, 55] by adding shortest puncture-to-puncture paths one at a time, in increasing order of length. To compute the shortest paths, we simultaneously propagate wavefronts from all k punctures using Dijkstra's algorithm. Whenever two wavefronts (*i.e.*, two growing shortest-path trees) collide, we add a new edge to the evolving minimum spanning tree and merge those two wavefronts. To implement this algorithm efficiently, we maintain the wavefronts in a union-find data structure. The resulting running time is $O(n \log n)$. □

This is essentially the algorithm proposed by Takahashi and Matsuyama [54] to compute approximate Steiner trees in arbitrary graphs. The same algorithm was also recently used by Sheffer [49] and by Lévy *et al.* [43] to compute cut graphs, where surface features with high discrete curvature play the role of punctures.

Lemma 6.5. *The length of the minimum puncture-spanning tree of any punctured polyhedral 2-manifold $(\overline{\mathcal{M}}, P)$ is at most twice the length of any cut graph of $(\overline{\mathcal{M}}, P)$.*

Proof: The *minimum Steiner tree* of P is the subgraph of $\overline{\mathcal{M}}_1$ of minimum total weight that includes every point in P. Since any cut graph of $(\overline{\mathcal{M}}, P)$ must touch every puncture, no cut graph is shorter than this minimum Steiner tree. On the other hand, the minimum spanning tree of P has at most twice the length of the minimum Steiner tree [38, 54]. □

6.3 Analysis

We now have all the components of our greedy cut graph algorithm. At any stage of the algorithm, we have a punctured manifold $(\overline{\mathcal{M}}, P)$. Our algorithm repeatedly cuts along a short non-separating cycle of $\overline{\mathcal{M}}$, using Corollary 5.8. This cut creates one or two new boundary circles, which we collapse to new punctures. When the manifold is reduced to a collection of punctured spheres, we cut along the minimum puncture-spanning tree of each component using the algorithm in Lemma 6.4.

Each non-separating cycle cut reduces the genus of $\mathcal{M}$ by 1. This immediately implies that our algorithm performs exactly g cycle cuts, so the overall running time is

$$g \cdot O(gn \log n) + O(n \log n) = O(g^2 n \log n).$$

For any graph X, let $|X|$ denote its total length. Let G^* denote the minimum cut graph of $(\overline{\mathcal{M}}, P)$. Let $(\overline{\mathcal{M}}_i, P_i)$ denote the punctured manifold after $g - i$ cycle cuts have been performed, so $\mathcal{M}_i$ has genus i, and let G_i^* denote the minimum cut graph of $\overline{\mathcal{M}}_i$, *ignoring* the punctures P_i. Since collapsing edges cannot increase the minimum cut graph length, we have $|G_i^*| \le |G_g^*| \le |G^*|$ for all i.

Let γ_i denote the short non-separating cycle of $\mathcal{M}_i$ found by Corollary 5.8. (We easily observe that *any* cut graph of $\overline{\mathcal{M}}_i$ must intersect this cycle.) Lemma 6.2 and Corollary 5.8 imply that $|\gamma_i| \le O((\log i)/i) \cdot |G_i^*|$ for all i and j. Summing over all g cuts, we conclude that the total length of all cycle cuts is at most

$$\sum_{i=1}^{g} O((\log i)/i) \cdot |G_i^*| = O(\log^2 g) \cdot |G^*|.$$

Similarly, Lemma 6.5 implies that the minimum puncture-spanning forest has length at most $2|G^*|$. Finally, regluing previously cut edges to obtain a single disk only reduces the length of the final cut graph. Thus, the final cut graph computed by our algorithm has length at most $O(\log^2 g) \cdot |G^*|$.

Theorem 6.6. *Given a polyhedral 2-manifold $\mathcal{M}$ with genus g and k boundary components, an $O(\log^2 g)$-approximation of its minimum cut graph can be constructed in $O(g^2 n \log n)$ time.*

Cutting along short essential cycles instead of short non-separating cycles leads to exactly the same asymptotic running time and approximation bounds, although the algorithm and its analysis are slightly more complicated. For purposes of analysis, we can divide the algorithm into phases, where in the ith phase, we cut along a short essential cycle of every component of the manifold that has genus i. Essential cycle cuts can separate the manifold

into multiple components, but since each component must have nontrivial topology, the algorithm performs at most $g-1$ separating cuts. At the end of the algorithm, if necessary, we reglue along some previously cut edges to obtain a single topological disk. We refer to the earlier version of this paper for further details [20].

7 Open Problems

We have developed new algorithms to compute exact and approximate minimal cut graphs for manifold surfaces with arbitrary genus and arbitrary boundary complexity. Our approximation algorithm is particularly simple.

Our results suggest several open problems, the most obvious of which is to improve the running times and approximation factors of our algorithms. Is the minimum cut graph problem *fixed-parameter tractable* [16]? That is, can we compute exact minimum cut graphs in time $f(g,k) \cdot n^{O(1)}$ for some function f? The similarity to the Steiner problem offers some hope here, since the minimum Steiner tree of k nodes in an n-node graph can be computed in $O(3^k n + 2^k n^2 + n^3)$ time [17, 29]. See also Bern [4] for related results on computing Steiner trees of graphs embedded in surfaces of small genus.

The approximation algorithm of Theorem 6.6 is somewhat indirect. It computes a short cut graph by repeatedly computing a 'reasonable' cut graph and then extracting a short nontrivial cycle that interacts with this cut graph. It is natural to conjecture that one can compute such a short cut graph directly, resulting in a faster algorithm. In particular, we conjecture that an approximately minimum cut graph can be computed in $O(gn \log n)$ time.

How well can we approximate the minimum cut graph in nearly-linear time? There are several simple heuristics to compute 'good' cut graphs in $O(n \log n)$ time, such as the dual shortest-path algorithm used by Gu *et al.* [27] and by Steiner and Fischer [52], and the algorithm described in the proof of Lemma 5.7. How well do these algorithms approximate the minimum cut graph?

More generally, is there a simple, practical, $O(1)$-approximation algorithm, like the minimum spanning tree approximation of Steiner trees? In fact, it might be that our algorithm provides such an approximation, as our current analysis seems to be far from tight. Unfortunately, the general Steiner tree problem is MAXSNP-hard [5], so an efficient $(1+\varepsilon)$-approximation algorithm for arbitrary $\varepsilon > 0$ seems unlikely.

Several authors have pointed out apparent tradeoffs between the quality of parameterizations and the length of the required cut graph; see, for example, Sorkine *et al.* [51]. How hard is it to compute the (approximately) shortest cut graph required for a parameterization whose distortion is less than some given limit? Conversely, how hard is it to (approximately) minimize the distortion of a parameterization, given an upper bound on the permitted length of the surface cuts? The complexity of these problems clearly depends on which distortion measure is used, but we expect almost any variant of this problem to be NP-hard.

Finally, can our ideas be applied to other useful families of curves on manifolds, such as homology generators (families of $2g$ cycles that intersect in g pairs) and pants decompositions (maximal sets of pairwise disjoint essential cycles [57])?

Acknowledgments. We would like to thank Herbert Edelsbrunner for an enlightening initial conversation. We are also grateful to Noga Alon, Steven Gortler, John Hart, Benjamin Sudakov, Kim Whittlesey, and Zoë Wood for helpful discussions, and to the anonymous referees for their suggestions.

References

[1] P. Alliez, M. Meyer, and M. Desbrun. Interactive geometry remeshing. *Proc. SIGGRAPH '02*, 347–354, 2002.

[2] U. Axen and H. Edelsbrunner. Auditory Morse analysis of triangulated manifolds. *Mathematical Visualization*, 223–236, 1998. Springer-Verlag.

[3] C. Bennis, J.-M. Vézien, G. Iglésias, and A. Gagalowicz. Piecewise surface flattening for non-distorted texture mapping. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 237–246, 1991. vol. 25.

[4] M. Bern. Faster exact algorithms for Steiner trees in planar networks. *Networks* 20(1):109–120, 1990.

[5] M. Bern and P. Plassman. The Steiner problem with edge lengths 1 and 2. *Inform. Proc. Lett.* 32(4):171–176, 1989.

[6] B. K. Bhattacharya and S. Sen. On a simple, practical, optimal, output-sensitive randomized planar convex hull algorithm. *J. Algorithms* 25:177–193, 1997.

[7] N. Biggs. Constructions for cubic graphs with large girth. *Elec. J. Combin.* 5:A1, 1998.

[8] J. Blömer. Computing sums of radicals in polynomial time. *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, 670–677, 1991.

[9] T. M. Chan, J. Snoeyink, and C. K. Yap. Primal dividing and dual pruning: Output-sensitive construction of 4-d polytopes and 3-d Voronoi diagrams. *Discrete Comput. Geom.* 18:433–454, 1997.

[10] S. Chari, P. Rohatgi, and A. Srinivasan. Randomness-optimal unique element isolation with applications to perfect matching and related problems. *SIAM J. Comput.* 24(5):1036–1050, 1995.

[11] È. Colin de Verdiére and F. Lazarus. Optimal systems of loops on an orientable surface. *Proc. 43rd Annu. IEEE Sympos. Found. Comput. Sci.*, 627–636, 2002.

[12] T. Dey, H. Edelsbrunner, and S. Guha. Computational topology. *Advances in Discrete and Computational Geometry*, 109–143, 1999. Contemporary Mathematics 223, American Mathematical Society.

[13] T. K. Dey and S. Guha. Transforming curves on surfaces. *J. Comput. Sys. Sci.* 58:297–325, 1999.

[14] T. K. Dey and H. Schipper. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. *Discrete Comput. Geom.* 14:93–110, 1995.

[15] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik* 1:269–271, 1959.

[16] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Monographs in Computer Science. Springer-Verlag, 1999.

[17] S. Dreyfus and R. Wagner. The Steiner problem in graphs. *Networks* 1:195–207, 1971.

[18] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. *Proc. SIGGRAPH '95*, 173–182, 1995.

[19] D. Eppstein. Seventeen proofs of Euler's formula: $V - E + F = 2$. *The Geometry Junkyard*, May 2001. ⟨http://www.ics.uci.edu/~eppstein/junkyard/euler/⟩.

[20] J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, 244–253, 2002.

[21] M. S. Floater. Parameterization and smooth approximation of surface triangulations. *Comput. Aided Geom. Design* 14(4):231–250, 1997.

[22] G. K. Francis and J. R. Weeks. Conway's ZIP proof. *Amer. Math. Monthly* 106:393–399, 1999. ⟨http://new.math.uiuc.edu/zipproof/⟩.

[23] M. R. Garey, R. L. Graham, and D. S. Johnson. The complexity of computing Steiner minimal trees. *SIAM J. Appl. Math.* 32:835–859, 1977.

[24] M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM J. Appl. Math.* 32:826–834, 1977.

[25] S. Gumhold and W. Straßer. Real time compression of triangle mesh connectivity. *Proc. SIGGRAPH '98*, 133–140, 1998.

[26] I. Guskov and Z. Wood. Topological noise removal. *Proc. Graphics Interface*, 19–26, 2001.

[27] X. Gu, S. J. Gortler, and H. Hoppe. Geometry images. *Proc. SIGGRAPH '02*, 355–361, 2002.

[28] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle. Conformal surface parameterization for texture mapping. *IEEE Trans. Visualizat. Comput. Graph.* 6(2):181–187, 2000.

[29] M. Hallett and T. Wareham. A compendium of parameterized complexity results. *SIGACT News* 25(3):122–123, 1994. ⟨http://web.cs.mun.ca/~harold/W_hier/compendium.html⟩.

[30] M. Hanan. On Steiner's problem with rectilinear distance. *SIAM J. Appl. Math.* 14:255–265, 1966.

[31] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2001. ⟨http://www.math.cornell.edu/~hatcher/⟩.

[32] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. *Proc. SIGGRAPH '01*, 203–212, 2001.

[33] M. Isenburg and J. Snoeyink. Face Fixer: Compressing polygon meshes with properties. *Proc. SIGGRAPH 2000*, 263–270, 2000.

[34] D. B. Johnson. Efficient algorithms for shortest paths in sparse networks. *J. Assoc. Comput. Mach.* 24(1):1–13, 1977.

[35] R. Kimmel, A. Amir, and A. M. Bruckstein. Finding shortest paths on surfaces using level sets propogation. *IEEE Trans. PAMI* 17:635–640, 1995.

[36] D. G. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm? *SIAM J. Comput.* 15:287–299, 1986.

[37] A. Klivans and D. A. Spielman. Randomness efficient identity testing of multivariate polynomials. *Proc. 33rd Annu. ACM Sympos. Theory Comput.*, 216–223, 2001.

[38] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for Steiner trees. *Acta Inform.* 15:141–145, 1981.

[39] J. B. Kruskal, Jr. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.* 7:48–50, 1956.

[40] F. Lazarus, M. Pocchiola, G. Vegter, and A. Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, 80–89, 2001.

[41] F. Lazarus and A. Verroust. Level set diagrams of polyhedral objects. *Proc. 5th ACM Sympos. Solid Modeling Appl.*, 130–140, 1999.

[42] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. MAPS: Multiresolution adaptive parameterization of surfaces. *Proc. SIGGRAPH '98*, 95–104, 1998.

[43] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *Proc. SIGGRAPH '02*, 362–371, 2002.

[44] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou. The discrete geodesic problem. *SIAM J. Comput.* 16:647–668, 1987.

[45] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica* 7:105–113, 1987.

[46] J. R. Munkres. *Topology*, 2nd edition. Prentice-Hall, 2000.

[47] D. Piponi and G. Borshukov. Seamless texture mapping of subdivision surfaces by model pelting and texture blending. *Proc. SIGGRAPH '00*, 471–478, 2000.

[48] J. Rossignac. Edgebreaker: Connectivity compression for triangle meshes. *IEEE Trans. Visualiz. Comput. Graph.* 5:47–61, 1999.

[49] A. Sheffer. Spanning tree seams for reducing parameterization distortion of triangulated surfaces. *Proc. Shape Modeling International*, 2002.

[50] A. Sheffer and E. de Sturler. Surface parameterization for meshing by triangulation flattening. *Proc. 9th International Meshing Roundtable*, 161–172, 2000. ⟨http://www.andrew.cmu.edu/user/sowen/abstracts/Sh742.html⟩.

[51] O. Sorkine, D. Cohen-Or, R. Goldenthal, and D. Lischinski. Bounded-distortion piecewise mesh parameterization. *Proc. 12th IEEE Visualization Conf.*, 355–362, 2002.

[52] D. Steiner and A. Fischer. Cutting 3D freeform objects with genus-n into single boundary surfaces using topological graphs. *Proc. 7th ACM Sympos. Solid Modeling Appl.*, 336–343, 2002.

[53] J. Stillwell. *Classical Topology and Combinatorial Group Theory*, 2nd edition. Graduate Texts in Mathematics 72. Springer-Verlag, 1993.

[54] H. Takahashi and A. Matsuyama. An approximate solution for the network Steiner tree problem. *Math. Japonica* 24:573–577, 1980.

[55] R. E. Tarjan. *Data Structures and Network Algorithms.* CBMS-NSF Regional Conference Series in Applied Mathematics 44. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983.

[56] G. Taubin and J. Rossignac. Geometric compression through topological surgery. *ACM Trans. Graph.* 17(2):84–115, 1998.

[57] W. Thurston. *Three-Dimensional Geometry and Topology, Volume 1.* Princeton University Press, New Jersey, 1997.

[58] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Trans. Automatic Control* 40:1528–1538, 1995.

[59] G. Vegter. Computational topology. *Handbook of Discrete and Computational Geometry*, chapter 28, 517–536, 1997. CRC Press LLC.

[60] G. Vegter and C. K. Yap. Computational complexity of combinatorial surfaces. *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, 102–111, 1990.

[61] R. Wenger. Randomized quickhull. *Algorithmica* 17:322–329, 1997.

[62] Z. Wood, H. Hoppe, M. Desbrun, and P. Schröder. Isosurface topology simplification. Tech. Rep. MSR-TR-2002-28, Microsoft Research, 2002. ⟨http://www.multires.caltech.edu/pubs/⟩.

[63] G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multi-dimensional scaling. *IEEE Trans. Visual. Comput. Graph.* 8:198–207, 2001.