Computing a Canonical Polygonal Schema of an Orientable Triangulated Surface

Francis Lazarus[*] Michel Pocchiola[†] Gert Vegter[‡] Anne Verroust[§]

Abstract

A closed orientable surface of genus g can be obtained by appropriate identification of pairs of edges of a $4g$-gon (the polygonal schema). The identified edges form $2g$ loops on the surface, that are disjoint except for their common end-point. These loops are generators of both the fundamental group and the homology group of the surface. The inverse problem is concerned with finding a set of $2g$ loops on a triangulated surface, such that cutting the surface along these loops yields a (canonical) polygonal schema. We present two optimal algorithms for this inverse problem. Both algorithms have been implemented using the CGAL polyhedron data structure.

1 Introduction

Let M_g be a regular $4g$–gon, whose successive edges are labeled $a_1, b_1, \overline{a}_1, \overline{b}_1, \cdots, a_g, b_g, \overline{a}_g, \overline{b}_g$. Edge x is directed counterclockwise, edge $\overline{x}$ clockwise. The space obtained by identifying edges x and $\overline{x}$, as indicated by their direction, is a closed oriented surface; See e.g. [6, Chapter 1.4]. This surface, called orientable surface of genus g, is homeomorphic to a 2–sphere with g handles. E.g., $\mathcal{M}_1$ is the torus; See Figure 1. The labeled polygon M_g is called the *canonical polygonal schema* of $\mathcal{M}_g$.

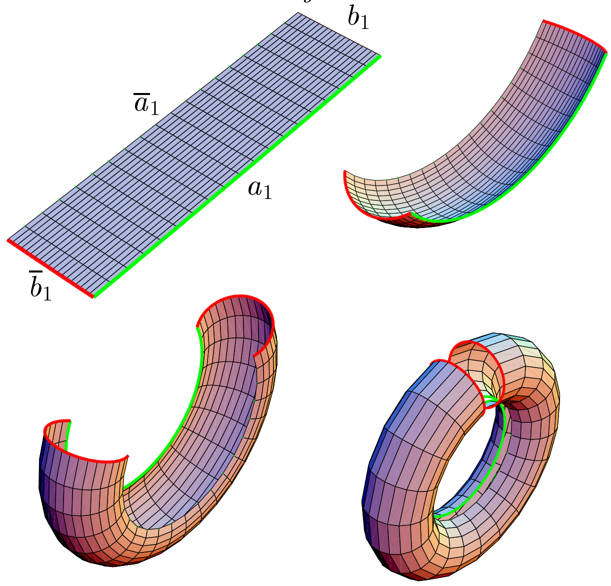

Figure 1: From polygonal schema to orientable surface: the torus.

It is easy to see that all vertices are identified to a single point p_0 of the surface. After identification in pairs, the edges of the polygonal schema form $2g$ curves on $\mathcal{M}_g$, which are disjoint, except for their common endpoint p_0. These $2g$ loops are generators of the fundamental group of $\mathcal{M}_g$ (and of the first homology group). In the sequel we drop the dependence on the genus from our notation, i.e., $\mathcal{M}$ denotes a closed orientable surface of genus g.

In this paper we consider the inverse problem: Given a combinatorial (triangulated) sur-

[*]CNRS and University of Poitiers, France. *E-mail*: `lazarus@sic.sp2mi.univ-poitiers.fr`

[†]Dépt d'Informatique, Ecole Normale Supérieure, Paris, France. *E-mail*: `Michel.Pocchiola@ens.fr`

[‡]Dept. of Math. and CS, University of Groningen, The Netherlands. *E-mail*: `gert@cs.rug.nl`

[§]INRIA Rocquencourt, France. *E-mail*: `Anne.Verroust@inria.fr`

face, find a *canonical* set of PL-curves (generators) such that, after cutting the surface along these generators, we obtain a canonical polygonal schema for the surface. A PL-curve is an alternating sequence of edges and vertices, where edges connect two successive vertices that lie in the same face, either in its interior or on the interior of one of its boundary edges.

In [8] an algorithm is sketched that constructs a canonical set of generators in optimal time and space. In this paper, we present in detail a simple optimal algorithm; we call this the *incremental method*, since we construct the generators while traversing all triangles of the surface. Our main result is

Theorem 1 *A canonical set of PL-generators for an orientable closed surface of genus g, with a total of n vertices, edges and faces, can be computed in $O(gn)$ time and space, which is worst-case optimal. Each PL-generator consists of $O(n)$ edges and vertices.*

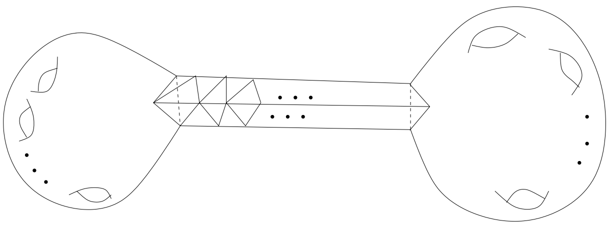

Figure 2: A surface with two groups of $\lceil g/2 \rceil$ and $\lfloor g/2 \rfloor$ handles, separated by a thin tunnel of size $\Omega(n)$. Regardless of the position of the base-point p_0, at least half of this tunnel must be crossed by at least $\lfloor g/2 \rfloor$ generators.

Optimality is easy to establish; See Figure 2.

Furthermore, we show how to turn Brahana's method [1] into a second algorithm computing a canonical set of generators in optimal time and space. We have implemented both methods using the `C++` library CGAL, and have compared the quality of the output of both algorithms. Although both algorithms are optimal, our implementation of Brahana's method seems to produce better (less complex) generators than the incremental method; cf Figure 7.

There are several reasons for presenting these algorithms here: (i) our algorithms greatly simplify the method of [8], (ii) full details are presented for the first time, (iii) the algorithms have been implemented, and (iv) the algorithms can be used to solve several other problems in computational topology. Among the applications are the construction of PL-homeomorphisms between surfaces, the construction of (a part of) the universal covering space of the surface, solving the contractibility problem of PL-curves on surfaces, cf [5], deciding whether two PL-curves on a surface are homotopic, and, if so, constructing a homotopy, cf [4]. Other applications are conceivable in connection with morphing, where a suitable parametrization of 2-manifolds is provided by the disk obtained by cutting along the canonical generators.

For general background material on computational topology, also in connection with applications, we refer to the surveys [3] and [7].

2 Surfaces with collars

Triangulated surfaces will be represented by Doubly-Connected Edge List, a data structure for representing subdivisions of surfaces. We refer to [2, Chapter 2] for details on this data structure. Note that every undirected edge of the triangulation corresponds to exactly two half-edges. The incremental algorithm starts with the open surface $\mathcal{S} = \mathcal{M} \setminus \{t_0\}$, where t_0 is an arbitrary (closed) triangle, eventually containing the common base point of the constructed generators. Initially, the topological boundary $\mathcal{B}$ of $\mathcal{S}$ is the boundary of t_0. The algorithm proceeds by visiting triangles incident to $\mathcal{B}$ along at least one edge, and cutting these (closed) triangles from $\mathcal{S}$. Note that the non-visited part of $\mathcal{M}$ is an open subset of $\mathcal{M}$. The topological boundary $\mathcal{B}$ is adjusted accordingly. It is represented as a circular sequence of half-edges, oriented in such a way that the triangle to the left of a half-edge belongs to $\mathcal{S}$. We say that a vertex *occurs* in $\mathcal{B}$ if it is the origin of a half-edge in $\mathcal{B}$.

As we will explain in more detail, the boundary $\mathcal{B}$ may become non-regular during this process, in the sense that a vertex occurs multiply in $\mathcal{B}$, or it contains both a half-edge and its opposite partner (called its *Twin* in [2]). See Figure 3 (Bottom). Yet, the irregularity of $\mathcal{B}$, and hence of the surface $\mathcal{S}$, is restricted. This is made more precise by introducing the notion of a *collar* of an open surface.

Definition 2 *A surface with collar in $\mathcal{M}$ is a pair $(\mathcal{S}, c)$, where $\mathcal{S}$ is an open submanifold of $\mathcal{M}$, and $c : \mathbb{S}^1 \times [0,1] \to \mathcal{M}$ is a continuous map, such that*
1. $c(\mathbb{S}^1 \times (0,1]) \subset \mathcal{S}$, and the restriction $c\,|_{\mathbb{S}^1\times(0,1]} : \mathbb{S}^1 \times (0,1] \to \mathcal{S}$ is an embedding;
2. $c(\mathbb{S}^1 \times \{0\}) \subset \mathcal{M} \setminus \mathcal{S}$;
3. The topological boundary of $\mathcal{S}$ (viz $\overline{\mathcal{S}} \setminus \mathcal{S}$) is the image of the closed curve $c : \mathbb{S}^1 \times \{0\} \to \mathcal{M}$.

Observe that the curve $c : \mathbb{S}^1 \times \{0\} \to \mathcal{M}$ is in general *not* an embedding. The curve $c : \mathbb{S}^1 \times \{1\} \to \mathcal{M}$, which is an embedding, may be considered as a 'regularization' of the – perhaps non-regular – boundary of $\mathcal{S}$. We refer to the half-open strip $c(\mathbb{S}^1 \times (0,1])$ as the *collar of* $\mathcal{S}$. This collar has *attachment curve* $c(\mathbb{S}^1 \times \{0\})$, and *free boundary* $c(\mathbb{S}^1 \times \{1\})$. Note that every continuous curve connecting a point in $\mathcal{S}$ with a point in $\mathcal{M} \setminus \mathcal{S}$ intersects the collar of $\mathcal{S}$.

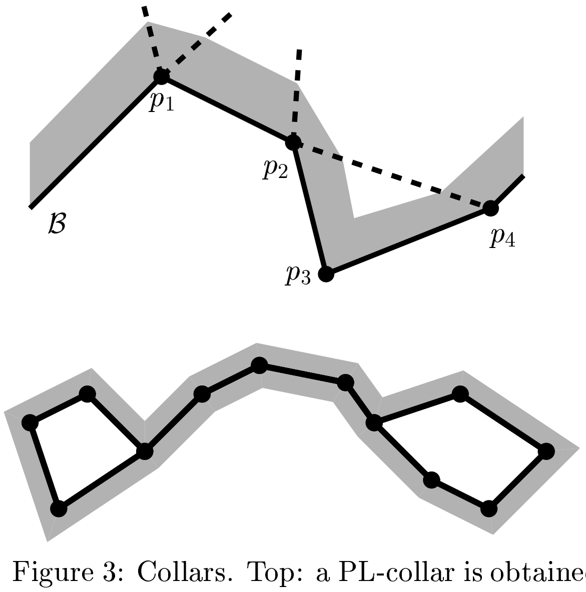

Figure 3: Collars. Top: a PL-collar is obtained by inserting vertices near the tail of half-edges incident to $\mathcal{B}$, or in a corner of a triangle. Bottom: a collar on a singular curve $\mathcal{B}$.

A collar $\mathcal{S}$ has a straightforward representation in the PL-setting. To this end, we insert a vertex near the tail of each half-edge in $\mathcal{S}$ emanating from a vertex of $\mathcal{B}$. Note that in this way an edge with both endpoints on $\mathcal{B}$ obtains two vertices. Furthermore, if two successive half-edges of $\mathcal{B}$, sharing a common vertex v, are incident to the same triangle t of $\mathcal{S}$, there is no half-edge of $\mathcal{S}$ emanating from v. In this case, we insert a vertex in the interior of t (e.g., on the bisector of the angle of t at v). Connecting the sequence of inserted vertices by edges we obtain a PL-collar of $\mathcal{S}$; See Figure 3. This type of collar will be used in Section 4.

As usual, the Euler characteristic $\chi(\mathcal{S})$ of $\mathcal{S}$ is the alternating sum of the numbers of vertices, edges and faces of $\mathcal{S}$. Cutting the surface along $\mathcal{B}$ we obtain a boundary of $\mathcal{S}$ consisting of a cyclic sequence of half-edges (where some pairs of half-edges may correspond to the same undirected edge of $\mathcal{M}$). Gluing a disk along this cyclic sequence of half-edges yields a closed orientable surface. By definition, the genus g of $\mathcal{S}$ is the genus of the latter surface. It is straightforward to check that $\chi(\mathcal{S}) = 1 - 2g$.

3 Surface traversal

We now describe the algorithm that visits all triangles of $\mathcal{M}$, starting from a single triangle. This algorithm is the backbone for the construction of a canonical system of generators, to be described in Section 4. Globally speaking the algorithm proceeds as follows. The main procedure MP, which is called on the complement $\mathcal{S}$ of the initial triangle, visits a triangle t incident upon the topological boundary $\mathcal{B}$ of $\mathcal{S}$, updates $\mathcal{S}$ and $\mathcal{B}$, and calls itself recursively on the updated version of $\mathcal{S}$. During this recursive process, $\mathcal{S}$ may become disconnected, in which case MP is called recursively on each connected component. It may also happen that $\mathcal{S}$ is not disconnected, but is not a surface with collar either (it will turn out that in the latter case the collar is split). Furthermore, in view of our ultimate goal of constructing generators, it is inefficient to visit connected components that are homeomorphic to a disk. Therefore we also

require $\mathcal{S}$ has positive genus g.
To fill in the details, we first specify the input of the main procedure.

Precondition of MP. The main procedure MP takes as input a pair $(\mathcal{S}, g)$, where $\mathcal{S}$ is a surface with collar, which has *positive* genus g.

In particular, the condition $g > 0$ guarantees that MP will not be called on disks, which is crucial in the analysis of the time complexity. The process of visiting triangle t, incident upon the topological boundary $\mathcal{B}$, is called an *extension.* We distinguish two types of extensions.

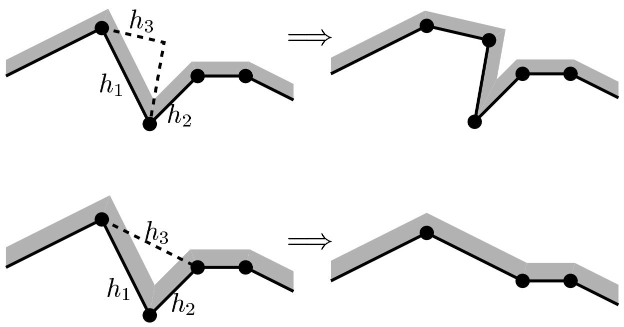

Figure 4: A regular extension.

Regular Extension: Triangle t shares either two vertices and one half-edge h_1 (Figure 4, top), or three vertices and two half-edges h_1, h_2 (Figure 4, bottom), with $\mathcal{B}$.

We update $\mathcal{B}$ in the former case by replacing the half-edge h_1 with the two-chain h_2, h_3, in the latter case by replacing the two-chain h_1, h_2 with the half-edge h_3. Note that the topological types of $\mathcal{B}$ and the collar do not change upon a regular extension. In particular, $\mathcal{S}' = \mathcal{S} \setminus \{t\}$ is a surface with collar. Therefore, the main procedure MP is called recursively on $\mathcal{S}'$. It is obvious that the Euler characteristic, and hence the genus, does not change under regular extension.

Splitting Extension: Triangle t shares three vertices and one half-edge with $\mathcal{B}$ (Figure 5, upper part).

The vertex of t, not adjacent to the common half-edge of $\mathcal{B}$ and t, is called the *split vertex*, and is denoted by v_s. Let the vertices of t be v_1, v_2 and v_3, such that v_1v_2 is a half-edge of $\mathcal{B}$, and hence $v_3 = v_s$. Let L be the part of $\mathcal{B}$ between v_3 and v_1, and let R be the part between v_2 and v_3. Then $\mathcal{B}$ is split into $\mathcal{B}_l = v_1v_3L$ and $\mathcal{B}_r = v_3v_2R$. We distinguish two sub-cases:

$S \setminus \{t\}$ *is not connected.* In this case $S \setminus \{t\}$ consists of two connected components, $\mathcal{S}_l$ and $\mathcal{S}_r$ say, with topological boundary $\mathcal{B}_l$ and $\mathcal{B}_r$, respectively. Both $\mathcal{S}_l$ and $\mathcal{S}_r$ are surfaces with collars, with attachment curves $\mathcal{B}_l$ and $\mathcal{B}_r$, respectively.

$S \setminus \{t\}$ *is connected.* In this case the topological boundary of $S \setminus \{t\}$ is $\mathcal{B}_l \cup \mathcal{B}_r$, so $S \setminus \{t\}$ is not a surface with collar. In particular, MP does not accept $S \setminus \{t\}$ as input. To remedy this situation, let γ be a shortest edge-path in $S \setminus \{t\}$ connecting $\mathcal{B}_l$ and $\mathcal{B}_r$, called the *join-path* (of $\mathcal{B}_l$ and $\mathcal{B}_r$). Let $v_l \in \mathcal{B}_l$ and $v_r \in \mathcal{B}_r$ be the extremal vertices of γ. See Figure 5

Lemma 3 *1. If $\mathcal{S} \setminus \{t\}$ is connected, and γ is a join-path, then $\mathcal{S} \setminus (\{t\} \cup \gamma)$ is a surface with collar, having genus $g - 1$.*
2. If $\mathcal{S} \setminus \{t\}$ is not connected, its connected components $\mathcal{S}_l$ and $\mathcal{S}_r$ are surfaces with collar. Moreover, if their genuses are g_l and g_r, respectively, then $g = g_l + g_r$.

PROOF. We only prove the first part, the proof of the second part being similar. First observe that $\mathcal{S}' = \mathcal{S} \setminus (\{t\} \cup \gamma)$ has one triangle (viz t) less than $\mathcal{S}$. Furthermore, edge v_1v_2 does not occur in $\mathcal{S}'$, but the edges of γ occur once in $\mathcal{S}$ and twice in $\mathcal{S}'$. Similarly, if the split vertex v_s is not a vertex of γ, it occurs twice in $\mathcal{S}$, as do the vertices of γ. Therefore, the Euler characteristics χ of $\mathcal{S}$ and χ' of $\mathcal{S}'$ are related by $\chi' = \chi + 1 + \#V(\gamma) - \#E(\gamma) = \chi + 2$. Here $V(\gamma)$ and $E(\gamma)$ denote the numbers of vertices and edges of γ. The same identity holds if v_s occurs (once or twice) in γ. The last identity yields $g' = g - 1$. Moreover, $\mathcal{S}'$ is a surface with collar, where the collar has attachment curve $\mathcal{B}'$: $v_1 \to v_s \overset{*}{\to} v_l \overset{*}{\to} v_r \overset{*}{\to} v_2 \to v_s \overset{*}{\to} v_r \overset{*}{\to} v_l \overset{*}{\to} v_1$. Here $v \to v'$ denotes a half-edge from v to v', and $v \overset{*}{\to} v'$ denotes a path from v to v' consisting of zero or more half-edges. □

In the notation of Lemma 3, if $\mathcal{S} \setminus \{t\}$ is not connected, the main procedure MP is recursively called on the connected components

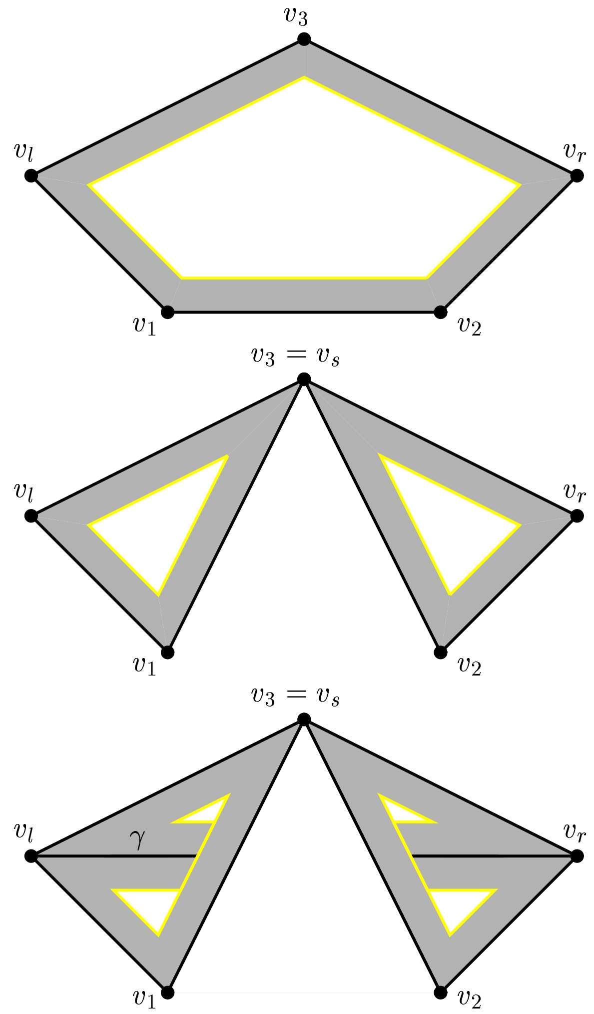

Figure 5: A splitting extension.

$\mathcal{S}_l$, if $g_l > 0$, and $\mathcal{S}_r$, if $g_r > 0$. Furthermore, if $\mathcal{S} \setminus \{t\}$ is connected, the main procedure is recursively called on $\mathcal{S}'$ if $g' > 0$.

Lemma 4 *1. If $\mathcal{S} \setminus \{t\}$ is connected, the join-path γ can be determined in time proportional to the size of $\mathcal{S}$.*
2. If $\mathcal{S} \setminus \{t\}$ has two connected components, establishing non-connectedness and computing the genuses of the connected components can be performed in time proportional to the size of the smaller connected component.

Proof. When a split occurs, we try to construct the join-path γ by means of a *tandem search* traversing the edges of the surface in parallel from the sources $\mathcal{B}_l$ and $\mathcal{B}_r$. More precisely, assuming the edges in $\mathcal{B}$ are already colored, we start coloring the half-edges of $\mathcal{B}_l$ and $\mathcal{B}_r$ with different colors, until the search exhausts the smaller of $\mathcal{B}_l$ and $\mathcal{B}_r$. After that, we reset the color of the larger part of the boundary. Then we visit the open surface $\mathcal{S} \setminus \{t\}$ by two parallel traversals, one starting from the curve $\mathcal{B}_l$ and the other one from $\mathcal{B}_r$. We visit and color new edges and their vertices with the same color as the extended boundary from which the traversal started. Then either the tandem search succeeds in connecting $\mathcal{B}_l$ and $\mathcal{B}_r$ by the join-path γ, or it detects that $\mathcal{S} \setminus \{t\}$ has two connected components $\mathcal{S}_l$ and $\mathcal{S}_r$ by exhausting the smaller of these two components. In the latter case we compute the genus of the smaller component by determining the number of vertices, edges and faces. Lemma 4, part 2, gives the genus of the other connected component. □

Lemma's 3 and 4 allow us to analyze the time complexity of the traversal of the initial surface $\mathcal{M}$. To this end, let t_0 be an arbitrary triangle of $\mathcal{M}$.

Corollary 5 *The call of the main procedure on the surface with collar $\mathcal{S}_0 = \mathcal{M} \setminus \{t_0\}$ is executed in time $O(gn)$, where g is the genus of $\mathcal{M}$ and n is the total number of vertices, edges and triangles in $\mathcal{M}$.*

Proof. The proof goes by induction with respect to the lexicographic order on the set of pairs (g,n). Our inductive hypothesis is:
IH(g,n): A call of MP on a surface with collar of genus g and complexity n, takes $O(gn)$ time.

Suppose the claim has been proven for collared surfaces of genus g' and complexity n' such that (g',n') lexicographically precedes (g,n). Consider a surface $\mathcal{S}$ of genus g and complexity n. Let t be the first triangle visited in the call of MP on $\mathcal{S}$, and let $\mathcal{S}' = \mathcal{S} \setminus \{t\}$.

Detecting whether an extension is regular or splitting can be easily implemented by coloring the vertices of the attachment curve $\mathcal{B}$. If visiting t corresponds to a regular extension, the call subsequent call of MP on $\mathcal{S}'$ takes $O(g(n-1))$ time. Since a regular extension is performed in constant time, IH(g,n) holds in this case.

So suppose visiting t corresponds to a splitting extension. If $\mathcal{S}'$ is connected, we construct a join-path γ; according to Lemma 4, this takes $O(n)$ time. If the genus $g-1$ of $\mathcal{S} \setminus (\{t\} \cup \gamma)$ is zero, the recursion terminates, so the hypothesis holds in this case. If $g > 1$, the main procedure recurs on $\mathcal{S}'$, which, according to the inductive hypothesis, takes $O((g-1)n)$ time. The call on $\mathcal{S}$ takes $O(n)$ additional time dedicated to the construction of γ, so the inductive hypothesis holds in this case.

If $\mathcal{S}'$ is not connected, it has two connected components $\mathcal{S}_l$ and $\mathcal{S}_r$, with genus g_l and g_r, and complexitiy n_l and n_r, respectively. Note that $n_l + n_r = n - 1$, and recall that $g_l + g_r = g$. Splitting the boundary (viz locating the split vertex v_s) and recoloring its smaller connected component again goes in $O(\min(n_l, n_r))$ time, using a tandem search like we did in the proof of Lemma 4.

If $g_l > 0$ and $g_r > 0$, the main procedure is recursively called on both $\mathcal{S}_l$ and $\mathcal{S}_r$, where it spends $O(g_l n_l)$ and $O(g_r n_r)$ time, respectively. Since detecting disconnectedness, and computing g_l and g_r, takes $O(\min(n_l, n_r))$ time, we see that the overall time complexity is $O(gn)$.

If $g_l = 0$, the recursive call on $\mathcal{S}_r$ takes $O(gn_r)$ time (the topological disk $\mathcal{S}_l$ is discarded). Therefore, the overall time complexity in this case is $O(gn_r + \min(n_l, n_r))$, hence again $O(gn)$. If $g_r = 0$, we argue similarly. □

4 Constructing generators

The backbone algorithm from Section 2 will now be extended by constructing a canonical set of generators from a base-point in the initial triangle. These generators will be routed along an *approach path* γ_{AP}, which connects the base point with the boundary of the non-visited part of the surface. As the algorithm proceeds, we should take care that generators we are about to complete do not intersect already constructed generators. Yet, we allow already constructed generators to intersect the non-visited part of the surface, although possible intersections should be confined to the collar of the non-visited part.

More precisely, let t_0 be the first triangle visited, and let the base-point p_0 be an interior point of t_0. We first extend the precondition, introduced in Section 2 for calling the main procedure MP on a non-visited surface $\mathcal{S}$ with collar. To this end, we assume from now on that a collar is piecewise linear, as described in Section 2 (See also Figure 3). In particular, a collar of $\mathcal{S}$ only intersects edges and faces of $\mathcal{S}$ incident upon the attachment curve $\mathcal{B}$, and such edges are intersected in interior points. Furthermore, we require that the attachment curve $\mathcal{B}$ of $\mathcal{S}$ has a distinguished half-edge h_{APA}, satisfying the following conditions:

(AP1) The base-point p_0 is connected by a PL-curve γ_{AP} to h_{APA}; apart from p_0, this *approach path* is disjoint from $\mathcal{S}$, and it does not share any point with already constructed generators and approach paths;

(AP2) The terminal point of γ_{AP} on h_{APA} can be connected to the free boundary of the collar of $\mathcal{S}$ by a line segment inside the face of $\mathcal{S}$ incident upon h_{APA}, which does not intersect any of the generators constructed so far;

(AP3) No already constructed generator intersects the free boundary of the collar of $\mathcal{S}$. No already constructed approach path intersects $\mathcal{S}$.

The distinguished edge h_{APA} is called the *approach path aperture* of $\mathcal{S}$. The existence of the line segment, refered to in condition AP2, will allow us to extend the approach path when visiting new triangles.

Lemma 6 *The main procedure* MP *can be enhanced in such a way that:*
1. It maintains the invariants (AP1), (AP2) and (AP3)
2. When called on the initial surface with boundary $\mathcal{M} \setminus \{t_0\}$, it constructs a canonical set of g generator pairs, in time $O(gn)$.

PROOF. Before describing the actual enhancement of MP, we impose some restrictions on the traversal and the approach paths, and introduce some primitive operations that facilitate the description of the algorithm.

We require that, during the traversal of the surface, the next triangle visited in a call of MP on $\mathcal{S}$ is incident upon the approach path aperture h_{APA}, contained in the boundary $\mathcal{B}$ of $\mathcal{S}$. Furthermore, we require that approach paths do not intersect vertices of $\mathcal{M}$.

A basic operation is that of *cloning an approach path*. Cloning an approach path γ_{AP}, directed from p_0 to its terminal vertex on the approach path aperture h_{APA}, amounts to constructing a PL-path from p_0 to h_{APA}, with the same combinatorial structure as γ_{AP} (i.e., intersecting the same sequence of edges and faces of $\mathcal{M}$). This clone should not share any point with already constructed approach paths or generators, apart from p_0. To avoid ambiguities, we assume that a clone runs to the left of its original. In view of condition (AP1), any approach path can be cloned, and cloning can even be repeated on clones.

Furthermore, we employ the notion of *routing a PL-curve* along (part of) the free boundary of a PL-collar. This operation is similar to cloning, in that we construct a PL-curve inside the PL-collar, which has the same combinatorial structure as (a sub-path of) the free boundary of the collar. We require this curve to be disjoint from already constructed generators and approach paths, which is possible in view of conditions (AP1) and (AP3).

Now consider a *regular extension*. Set the approach path aperture h'_{APA} of $\mathcal{S}' = \mathcal{S} \setminus \{t\}$ to one of the half-edges in the boundary of t, not incident upon $\mathcal{B}$ (e.g. h_3 in Figure 4). According to (AP2), there is a line segment $s = pp'$ connecting the terminal vertex p of γ_{AP} with a point p' inside t and on the free boundary of the collar of $\mathcal{S}$. Let q be a point on h'_{APA} not belonging to the collar of $\mathcal{S}$. Such a point exists, since h'_{APA} does not belong to $\mathcal{B}$, and since the PL-collar of $\mathcal{S}$ only intersects faces incident upon $\mathcal{B}$. Extending γ_{AP} with pp' and $p'q$, we obtain an approach path γ'_{AP} for $\mathcal{S}'$ satisfying

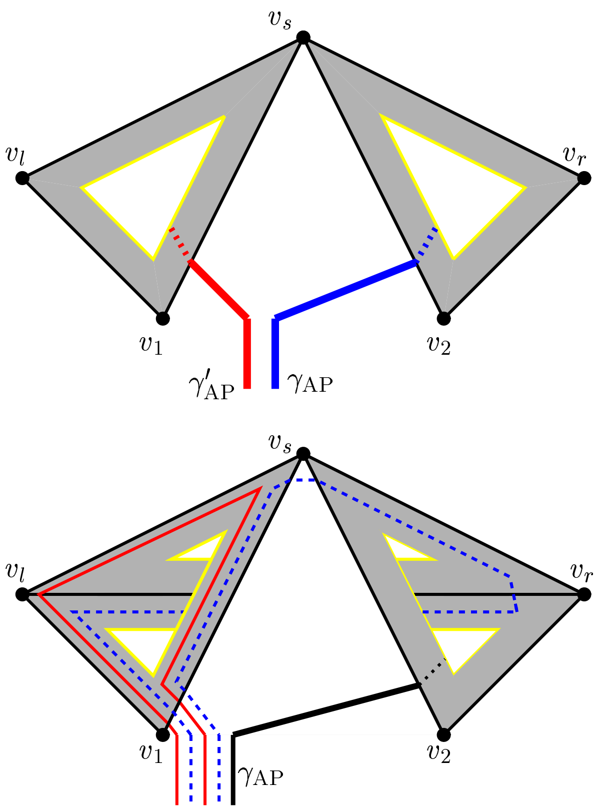

Figure 6: Splitting extensions upon visit of triangle $t = v_1v_2v_3$. Top: $\mathcal{S} \setminus \{t\}$ is not connected. Bottom: $\mathcal{S} \setminus \{t\}$ is connected, so a pair of generators is constructed.

(AP1). Furthermore, since q does not belong to the collar of $\mathcal{S}$ there is a line segment qq', with q' on the free boundary of the collar of $\mathcal{S}'$, that is disjoint from the collar of $\mathcal{S}$. In other words, (AP2) holds for $\mathcal{S}'$. Since we do not complete any generators, (AP3) also holds for $\mathcal{S}'$. The enhanced version of a regular extension obviously takes $O(1)$ time. It remains to consider a *splitting extension*. If $\mathcal{S}' = \mathcal{S} \setminus \{t\}$ is disconnected, and both g_l and g_r are positive, we construct a clone γ'_{AP} of the approach path γ_{AP}. Now we extend γ'_{AP} to the half-edge v_1v_s of $\mathcal{B}_l$, and we extend γ_{AP} to the half-edge v_sv_2 of $\mathcal{B}_r$ (the notation is as in Section 2); See Figure 6, Top. Arguing as in the case of a regular extension, we conclude that conditions (AP1), (AP2) and (AP3) hold for the connected components $\mathcal{S}_l$ and $\mathcal{S}_r$ of $\mathcal{S}'$, with approach path apertures v_1v_s and v_sv_2, respectively. If g_l or g_r is zero, we just extend the approach path to the non-visited part of positive genus in $O(1)$ time. Cloning only needs to be done in case the genus of both non-visited parts is less than the genus of $\mathcal{S}$, which happens at most $g-1$ times. Therefore, the overall complexity of all splitting extensions of this type is $O(gn)$.

Finally, consider a splitting extension in which $\mathcal{S}' = \mathcal{S} \setminus \{t\}$ is connected. Now we construct *four disjoint clones* γ_1, γ_2, $\overline{\gamma}_1$ and $\overline{\gamma}_2$ of γ_{AP}, whose respective end-points p_1, p_2, $\overline{p}_1$ and $\overline{p}_2$, occur in this order on the approach path aperture h_{APA} between v_1 and the end-point of γ_{AP}. The approach path γ_{AP} is now extended to the half-edge $v_s v_2$, see Figure 6, Bottom. As before, we can do this in such a way that (AP1), (AP2) and (AP3) holds for $\mathcal{S}'$. Finally, we complete a pair of generators by connecting the end-points of γ_1 and γ_2 with the end-points of $\overline{\gamma_1}$ and $\overline{\gamma_2}$, respectively, by two curves σ_1 and σ_2; See Figure 6, Bottom. More precisely, let $\mathcal{F}$ and $\mathcal{F}'$ be the free parts of the collars of $\mathcal{S}$ and $\mathcal{S}'$. Then σ_1 is a PL-curve obtained by connecting p_1 to a point near v_1 on $v_1 v_s$ by a curve inside t, and subsequently routing it along the part of $\mathcal{F}$ near $v_1 \overset{*}{\to} v_l \overset{*}{\to} v_s$ and along the part of $\mathcal{F}'$ near $v_s \overset{*}{\to} v_1$, and, finally, connecting it to $\overline{p_1}$. Furthermore, σ_2 is a PL-curve obtained by connecting p_2 to a point near v_1 on $v_1 v_s$, and subsequently routing it along the part of $\mathcal{F}'$ near p$v_1 \overset{*}{\to} v_l \overset{*}{\to} v_r$, then along the part of $\mathcal{F}$ near $v_l \overset{*}{\to} v_s$, letting it traverse t near v_s, then routing it along the part of $\mathcal{F}'$ near $v_s \overset{*}{\to} v_1$, and, finally, connecting it to $\overline{p}_2$ by a curve inside t. Obviously, σ_1 and σ_2 do not intersect the free boundary of the collar of $\mathcal{S}'$. Furthermore, it is easy to see that these curves can be constructed in such a way that they are disjoint from any generators or approach paths already constructed.

The time complexity of this splitting operation is $O(n)$, since the generators share only a constant number of edges and vertices with each edge and face of $\mathcal{M}$. Since there are exactly g splitting extensions of this type, the overall time complexity is $O(gn)$. □

The main theorem is a straightforward consequence of Lemma 6.

5 Brahana's algorithm

The inverse of a path p is denoted by $\imath(p)$ or p^{-1}, and for a set of paths S we denote the set $S \cup \imath(S)$ by $\hat{S}$.

Let G be a maximal subgraph of the vertex-edge graph of $\mathcal{M}$ such that $\mathcal{M} \setminus G$ is connected, and let T_G be a tubular neighbourhood of G in $\mathcal{M}$. By construction, $\mathcal{M} \setminus T_G$ is a topological disk and G is a deformation retract of T_G. Therefore a set of generators of the fundamental group $\pi(G,x)$ of G at x is also a set of generators of the fundamental group $\pi(\mathcal{M},x)$ of $\mathcal{M}$ at x. We can decompose our method into three steps:

1. First we construct a set $\mathcal{G}$ of $(2g)$ generators of $\pi(G,x)$, associated with a set E of $(2g)$ directed edges of $\mathcal{M}$ under a bijection $\ell : E \to \mathcal{G}$, and a cycle ϕ of $\hat{E}$ such that for $e \in \hat{E}$:

$$\ell(\phi(e))\ell(\phi^2(e))\ldots\ell(\phi^{4g}(e)) \sim \epsilon_x \qquad (*)$$

in $\pi(\mathcal{M},x)$. Here ϵ_x is the trivial path at x.

2. Secondly, we transform in $O(gn)$ time the set $\mathcal{G}$ into a set $\mathcal{H}$ of generators x_i, y_i of $\pi(G,x)$, each of linear complexity, such that a loop in $\mathcal{H}$ is homotopic (in G) to the concatenation of $O(g)$ loops in $\hat{\mathcal{G}}$, and the relation satisfied by the x_i, y_i in $\pi(\mathcal{M},x)$ is in 'canonical form', i.e.

$$[x_1,y_1]\cdots[x_g,y_g] \sim \epsilon_x. \qquad (**)$$

As usual, $[x_i,y_i]$ is the commutator $x_i y_i x_i^{-1} y_i^{-1}$, and $\sim$ denotes path-homotopy.

3. Finally, we show how to construct in $O(gn)$ time a canonical set of generators x_i^*, y_i^* of $\pi(\mathcal{M},x)$ such that $x_i \sim x_i^*$ and $y_i \sim y_i^*$ in T_G.

Step 1. We construct a spanning tree T of G rooted at x. Let E denote the set of non-tree edges in G; Each edge in E is oriented arbitrarily and each edge in T is oriented towards the root. Without loss of generality we assume for convenience that there is only one edge e_{sink} of the tree incident upon x. For each (directed) edge $e \in \hat{E}$ we consider the shortest edge-path $\gamma_e = e e_1 e_2 \cdots e_{\mathrm{sink}}$ from e to x in T. By construction, for $e \neq e'$ the paths γ_e and $\gamma_{e'}$ coincide only on a proper suffix sub-path, i.e., both paths can be decomposed as $\gamma_e = \tau_{e,e'}\gamma_{e,e'}$ and $\gamma_{e'} = \tau_{e',e}\gamma_{e',e}$, where $\gamma_{e,e'} = \gamma_{e',e}$ and $\tau_{e,e'}$ and $\tau_{e',e}$ are disjoint except at their sink $v(e,e')$. One can check that the relation on the edges in $\hat{E}$ defined by $e \prec e'$ if the sink edges of $\tau_{e,e'}$, $\tau_{e',e}$ and the source edge of $\gamma_{e,e'}$ are in counterclockwise order around their common endpoint $v(e,e')$ — with respect to the choice of an orientation of the surface $\mathcal{M}$ — is a transitive relation.

Let now $\ell(e)$ be the loop with basepoint x obtained by concatenation of the loops $\imath(\gamma_e)$ and $\gamma_{\imath(e)}$, removing one of the two occurrences of e^{-1}, i.e., $\ell(e) = \imath(\gamma_e)\gamma_{\imath(e)}$. Note that $\ell(\imath(e)) = \imath(\ell(e))$. The set $\mathcal{G} := \ell(E)$ is a set of $(2g)$ generators of $\pi(G,x)$, and consequently of $\pi(\mathcal{M},x)$. Furthermore, the unique relation in $\pi(\mathcal{M},x)$ satisfied by these generators is $(*)$, where the operator ϕ is defined by $\phi(e) = \psi \circ \imath(e)$. Here $\psi(e)$ is the successor of e with respect to the circular order on $\hat{E}$, induced by the linear order $\prec$.

Step 2. We use a sequence of Brahana transformations, cf [8]. Let $\ell_i = \ell(\phi^i(e))$ for some $e \in \hat{E}$, and let M be the loop $\ell_1 \cdots \ell_{4g}$. The loop M can be decomposed into $aX_1bX_2a^{-1}X_3b^{-1}X_4$, where a and b are loops in $\hat{\mathcal{G}}$, and X_4 is nonempty (unless X_1, X_2 and X_3 are empty, in which case we are done). If X_1, X_2, X_3 are not all empty we replace the loops a and b by the loops $x = aX_1bX_2a^{-1}$ (consequently $b^{-1} = X_2a^{-1}x^{-1}aX_1$) and $y = X_3X_2a^{-1}$ ($a = y^{-1}X_3X_2$) to obtain successively $M \sim \overbrace{aX_1bX_2a^{-1}}^{x}X_3b^{-1}X_4 \sim x\,\overbrace{X_3X_2a^{-1}}^{y}x^{-1}aX_1X_4 \sim [x,y]X_3X_2X_1X_4 \sim X_3X_2X_1X_4[x,y]$.
If X_1, X_2, X_3 are all empty, then we simply set $x = a$, $y = b$. In both cases $M \sim M'[x,y]$ where M' is the concatenation in some order of the loops in $\hat{\mathcal{G}} \setminus \{a, a^{-1}, b, b^{-1}\}$, and where x and y are loops composed of $O(g)$ generators in $\hat{\mathcal{G}}$. The loops a and b and their corresponding edges in $\hat{E}$ are said to be *converted.* After j such transformations we have converted a set $\mathcal{G}_j$ of $2j$ generators in $\mathcal{G}$ into a set $\mathcal{H}_j$ of $2j$ generators $x_1, y_1, x_2, y_2, \ldots, x_j, y_j$, such that $M \sim M_j[x_1,y_1]\cdots[x_j,y_j]$. Here M_j is the concatenation in some order of the loops in $\hat{\mathcal{G}} \setminus \hat{\mathcal{G}}_j$. For $j = g$ we obtain generators which satisfy $(**)$, but whose total complexity is only in $O(g^2n)$.

We now explain how to reduce the complexity of these loops by homotopy to $O(gn)$. First we examine how the relation $\phi = \psi \circ \imath$ is tranformed. For $j \geq 0$ and for $e \in \hat{E} \setminus \hat{E}_j$ we define $\psi_j(e)$ to be the $\prec$-successor of e in $\hat{E} \setminus \hat{E}_j$, and $\phi_j(e)$ to be the edge e' such that the successor of $\ell(e)$ in M_j is $\ell(e')$.

Lemma 7 *$\phi_j(e) = \psi_j \circ \imath(e)$.*

Proof. We prove the result by induction. The case $j = 0$ follows from the definition of ϕ. Let a and b be the loops converted at step $j+1$. One has $M_j = aX_1bX_2a^{-1}X_3b^{-1}X_4$ and $M_{j+1} = X_3X_2X_1X_4$. Let $e' = \phi_{j+1}(e)$. If $e' = \phi_j(e)$, then $e' = \psi_j(\imath(e)) = \psi_{j+1}(\imath(e))$, since e and e' are not converted at step $j+1$. Assume now that $e' \neq \phi_j(e)$, and let e_i^k for $k = 1,2$ and $i = 1,2,3,4$ be defined by $X_i = \ell(e_i^1)X_i'\ell(e_i^2)$ if X_i is non empty. The pair (e,e') coincides with one of the pairs $(e_k^2, e_{k'}^1)$ where k precedes k' in the order 3,2,1,4. For example if $e = e_3^2$ and $e' = e_2^1$ then $\psi_j(\imath(e_3^2)) = \phi_j(e_3^2) = \imath(b)$ and $\psi_j(\imath(b)) = \phi_j(b) = e_2^1 = e'$. Therefore, $\psi_{j+1}(\imath(e)) = e'$. The other cases are similar. □

We are now ready to decrease in optimal time the complexity of the loops x_i, y_i. Assume that $x_j = \ell(a)\ell(b)\cdots\ell(z)$ and let $sc(x_j)$ be the loop defined by $\imath(\gamma_a)\ell(\imath(a),b)\ell(\imath(b),c)\cdots\ell(\imath(y),z)\gamma_{\imath(z)}$, where $\ell(e,e')$ is the concatenation of the two paths $\tau_{e,e'}$ and $\imath(\tau_{e',e})$. Clearly $x \sim sc(x)$, and the size of $sc(x)$, i.e., its number of edges in $\mathcal{M}$, is in $O(n)$. Starting from its source e, we can visit the edges of $\ell(e,e')$ in time proportional to its size if we can determine efficiently the vertex $v(e,e')$. In view of Lemma 7 this can easily be done in $O(1)$ time, provided we maintain for each node v of the tree T the $\prec$-ordered list $L_j(v)$ of edges $e \in \hat{E}$ whose corresponding loops have non yet been converted, and whose associated paths γ_e lie along v. The lists $L_0(v)$ are easily created in $O(gn)$ time, and updated in $O(n)$ time, each time an edge is converted by a traversal of the corresponding loop.

Step 3. Omitted from this version.

6 Implementation

We have implemented both the incremental and Brahana's algorithm in `C++`, using the CGAL polyhedron data structure. Being purely combinatorial, the implementation does not present particular difficulties. It can be seen that the canonical set of PL-loops can be drawn without vertices interior to faces. In practice, a PL-loop is specified by the list of edges it crosses. Also, each edge of the combinatorial surface points to the list of loops it is crossed by. In order to visualize the PL-loops, we uniformly insert in each edge a number of points equals to the size of its list. We then link these points according to each loop list.

In Section 4 we always visit a triangle incident to the approach path aperture. In practice, we can choose any triangle incident to the boundary and keep the same complexity. In our implementation we use a 'potato peeling' traversal. This heuristic produces nicer loops. Experimentation shows that Brahana's algorithm generally produces curves with lower complexity (total number of segments). However the incremental method may be competitive when the initial set of generators in Brahana's method satisfies a relation 'close' to the other canonical form: $a_1b_1a_2b_2\dots a_gb_g\overline{a_1}\overline{b_1}\dots\overline{a_g}\overline{b_g}$. In this case, the final generators are indeed expressed as $\Omega(g^2)$ initial generators.

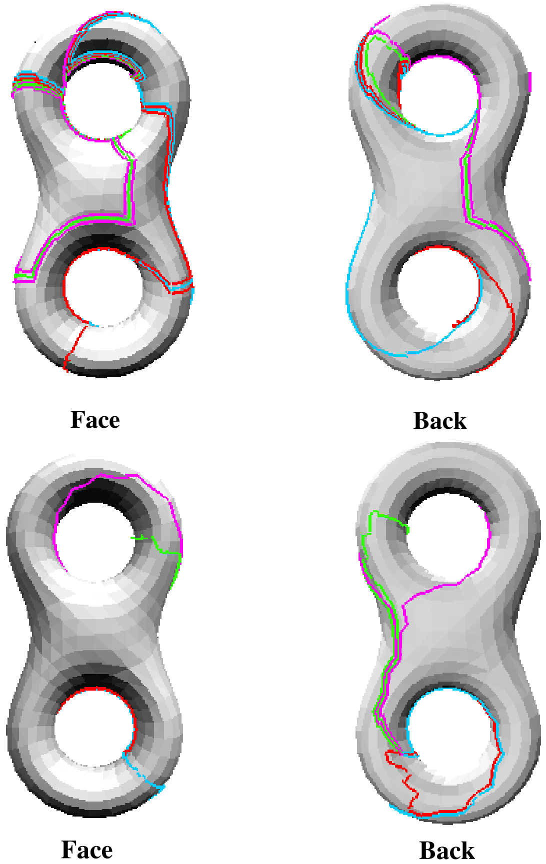

Figure 7: Output of the incremental algorithm (top) and Brahana's algorithm (bottom).

References

[1] T. Brahana. Systems of circuits on 2-dimensional manifolds. *Ann. Math.*, 23:144–168, 1921.

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

[3] T. Dey, H. Edelsbrunner, and S. Guha. Computational topology. In *Adv. in Discr. and Comput. Geom.*, volume 223 of *Contemp. Math.*, p. 109–143, 1999.

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

[5] H. Schipper. Determining contractibility of curves. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, p. 358–367, 1992.

[6] J. Stillwell. *Classical Topology and Combinatorial Group Theory*. Springer-Verlag, New York, 1993.

[7] G. Vegter. Computational topology. In J. E. Goodman and J. O'Rourke, editors, *Handbook Discr. Comput. Geom.*, ch. 28, p. 517–536. CRC Press LLC, 1997.

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

Addendum[1]

Here we comment on both our algorithms, applied to the pelvis bone data set. When computing a canonical polygonal schema with Brahana's algorithm (See Section 5), only one Brahana-transformation is needed. The total size of the computed loops is 513 edges.
The initial loop M (beginning of step 2 in Brahana's algorithm) turns out to be:

$$(+0)(-5)(-0)(+1)(-2)(-1)(+2)(+3)(+5)(-4)(-3)(+4)$$

In the notation of Section 5, the algorithm takes $a = (+3)$ and $b = (+5)$. The first part of the Brahana transformation is:

$$x = aX_1bX_2a^{-1} = (+3)(+5)(-4)(-3),$$

so $X_1 = \epsilon$, $X_2 = (-4)$, $X_3 = (+4)(+0)$, and $X_4 = (-0)(+1)(-2)(-1)(+2)$. Thus M becomes:

$$(+0)(+1)(+2)(-1)(-2)(+x)(+3)(-x)(+4)(-3)(-0)(-4).$$

The second part of this Brahana transformation corresponds to the choice $y = X_3X_2a^{-1} = (+4)(+0)(-4)(-3)$, and M becomes (See Figure 8):

$$(+0)(-4)(-0)(+1)(-2)(-1)(+2)(+x)(+y)(-x)(-y)(+4)$$

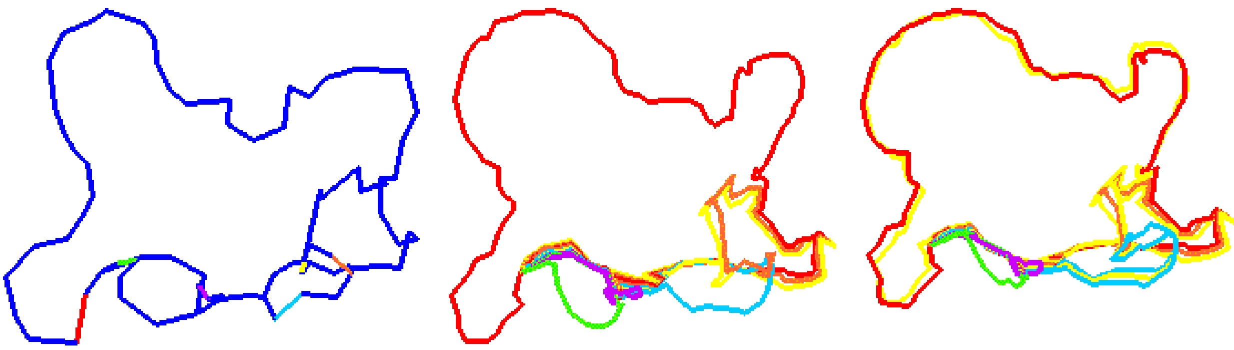

Figure 8: Brahana's algorithm. Left: G and its spanning tree T. Middle and right: the corresponding loops and the canonical set of PL-generators during and after one Brahana transformation. The color coding is: $0 \leftrightarrow$ red; $1 \leftrightarrow$ green; $2 \leftrightarrow$ purple; $3 \leftrightarrow$ blue; $4 \leftrightarrow$ yellow; $5 \leftrightarrow$ orange.

The iterative method, on the other hand, computes a canonical polygonal schema consisting of a total of 1126 edges. Figure 9 shows the first splitting extension, and the first couple of associated loops.
Finally, we show the output of both algorithms on the pelvis data set; See Figure 10.

[1]Not part of the submission for the proceedings

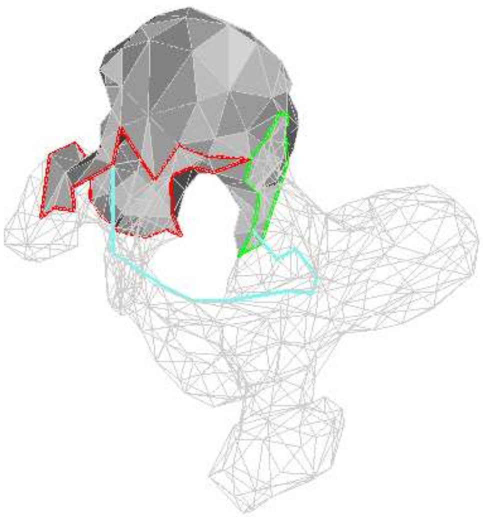

Figure 9: Iterative method. Left: the first splitting extension and the join-path. Right: the corresponding loops.

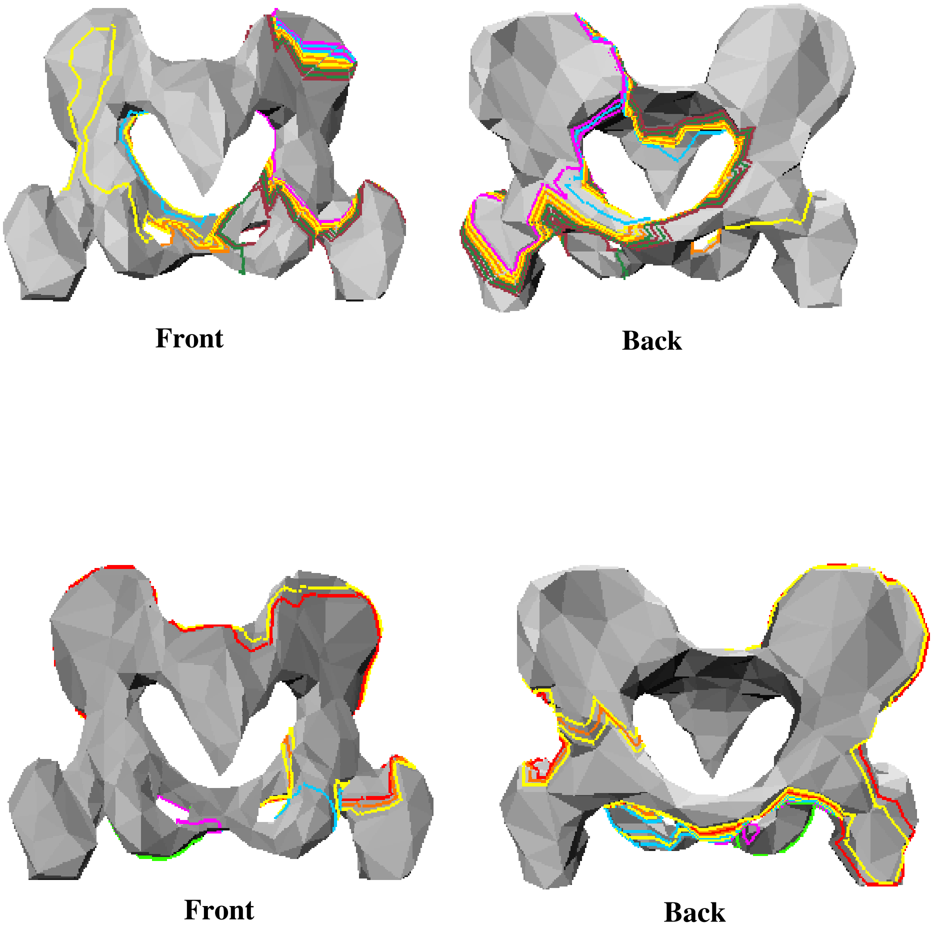

Figure 10: Output of our algorithms on the pelvis data set. Top: The incremental method. Bottom: Brahana's method