On the homotopy test on surfaces with boundaries

Julien Rivaud* Francis Lazarus†

Abstract

Let G be a graph cellularly embedded in a surface $\mathcal{S}$ orientable or not, and with nonempty boundary. Given two closed walks c and d in G, we describe linear time algorithms to decide if c and d are homotopic in $\mathcal{S}$, either freely or with fixed basepoint. After $O(|G|)$ time preprocessing independent of c and d, our algorithms answer the homotopy test in $O(|c| + |d|)$ time, where $|G|$, $|c|$ and $|d|$ are the respective numbers of edges of G, c and d.

1 Introduction

Computational topology of surfaces has received much attention in the last two decades. Among the notable results we may mention the test of homotopy between two cycles on a surface [2], the computation of a shortest cycle homotopic to a given cycle [1], or the computation of optimal homotopy and homology bases [3]. In their 1999 paper, Dey and Guha announced a linear time algorithm for testing whether two curves are freely homotopic on a triangulated surface without boundary. In [4] we showed that their method is invalidated by subtle flaws and provided a new geometric approach that confirms the linear time bound on the free homotopy test. This technique can be extended to handle surfaces with boundaries by gluing a punctured torus to each boundary cycle. We nevertheless present a much simpler and self-contained method for surfaces with nonempty boundary which also answers the free homotopy test on non-orientable surfaces with boundary.

Let G be a graph cellularly embedded in a surface $\mathcal{S}$ with at least one boundary. Each face of G in $\mathcal{S}$ is thus a disk or an annulus. By extending its boundaries we can retract $\mathcal{S}$ onto a subgraph G' of G. Each homotopy class in G' has a canonical reduced representation obtained by removing spurs. To know whether two cycles of G are homotopic in $\mathcal{S}$ it is thus sufficient to compute their deformation retract on G', remove spurs until they are reduced and check them for equality (up to circular permutation).

An edge e of G does not necessarily retract on a single edge of G' but rather on a subwalk w_e of a boundary cycle of G' — that is a facial cycle of its embedding in $\mathcal{S}$. To achieve the claimed time bound we do not expand w_e down to edges of G' but keep it under the following abstract representation: a reference to a boundary walk along with start and end indices. In $\mathcal{O}(|G|)$ total time we can compute for all $e \in G$ the abstract subwalk which e retracts on. Retracting a cycle c of G then yields a sequence of such subwalks. The removal of a spur in the underlying expanded cycle can be expressed as an operation on these subwalks. These operations define a rewriting system that we run on the sequence of subwalks until its underlying cycle is reduced. This takes linear time in the initial number of subwalks, that is $\mathcal{O}(|c|)$ time.

The contractibility test easily reduces to the (free) homotopy test and we only consider this last test in this abstract. Given two cycles c and d in G, we first compute two reduced sequences s and t of abstract subwalks whose underlying cycles c' and d' are freely homotopic to c and d respectively. Even if c' and d' are cyclically equal, the sequences s and t may not be literal permutations of each other. If σ is a sequence of subwalks whose underlying reduced cycle is u, we define its ***canonical (cyclic) sequence*** $\mathrm{Can}(\sigma) = \mathrm{Can}(u)$ that only depends on u. We can now decide if c and d are homotopic by comparing $\mathrm{Can}(s)$ and $\mathrm{Can}(t)$ up to circular permutation of their subwalks. Since we can compute $\mathrm{Can}(s)$ and $\mathrm{Can}(t)$ in time proportional to their number of subwalks, we obtain:

Theorem 1 (Homotopy test) *Let G be a graph cellularly embedded in a surface $\mathcal{S}$ with at least one boundary. Let c and d be two cycles with a total of k edges in G. After a $\mathcal{O}(|G|)$ time preprocessing of G, independent of c and d, we can decide if c and d are freely homotopic in $O(k)$ time.*

2 Background

We provide some definitions and properties; see [5] or [6, chapter 3] for details on rotation systems.

Cellular embedding of graphs A graph is ***cellularly embedded*** in a surface $\mathcal{S}$ without boundary if every open face of its embedding is a disk. A cellular embedding can be encoded by a ***rotation system***, that is a set of ***half-edges*** with two unary operations: an involution exchanging the direction on edges, and a cyclic permutation around vertices. Each (half)-edge is associated a ***signature*** $\in \{-1, 1\}$ indicating whether

*GIPSA-Lab, CNRS, Grenoble, France; julien.rivaud@gipsa-lab.grenoble-inp.fr

†GIPSA-Lab, CNRS, Grenoble, France; francis.lazarus@gipsa-lab.grenoble-inp.fr

This is an extended abstract of a presentation given at EuroCG 2012. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

the orientation of the cyclic permutation is the same or not around its endpoints. The face traversal procedure described in [6] allows to traverse all the facial cycles in $O(|G|)$ time. In particular, we can determine whether each edge of G is incident to only one facial cycle or to two distinct facial cycles.

Surfaces with boundaries In order to handle surfaces with boundaries we allow every face of G in $\mathcal{S}$ to be either a disk or an annulus. In other words G is a cellular embedding in the closure $\hat{\mathcal{S}}$ of $\mathcal{S}$ obtained by attaching a disk to every boundary of $\mathcal{S}$. We record this information by storing a boolean for every facial cycle of G indicating whether the associated face is perforated or not. Assume that an edge e is incident to a perforated face f and a plain face f'. We can perform an elementary collapse of f' through its free edge e, thus extending the perforation in f. Equivalently, we can remove e from G and merge f and f' into a single (perforated) face. We obtain this way an embedding of $G - e$ into $\mathcal{S}$ that simulates a deformation retraction of $\mathcal{S}$ without actually modifying $\mathcal{S}$.

Homotopy in embedded graphs We consider homotopy of closed walks in G with respect to $\mathcal{S}$. Hence two cycles of G are homotopic if one can be continuously transformed to the other on $\mathcal{S}$. If all the facial cycles are tagged as boundaries, then $\mathcal{S}$ is deform retracts onto G and homotopy on $\mathcal{S}$ reduces to homotopy on G. In particular, every cycle of G has a canonical homotopic cycle obtained by removing spurs until the cycle is reduced. As usual, a ***spur*** is the concatenation of two opposite oriented edges and a cycle is ***reduced*** if it contains no spur.

3 Retracting $\mathcal{S}$ to a thick graph

We first reduce the number of vertices of G:

Lemma 2 *Let G be a graph embedded on a surface $\mathcal{S}$. We can contract the edges of a spanning tree of G in $\mathcal{O}(|G|)$ time. We obtain this way a graph G_1 embedded on $\mathcal{S}$ with a single vertex, fewer edges than G and as many faces. Cycles in G are homotopic if and only if their contractions in G_1 are homotopic.*

Proof. We assume that every edge of G points to its incident faces. Updating and contracting each edge of the spanning tree can be done in constant time per edge by updating the rotation system: no face disappears or changes of boundary status. Computing and contracting a spanning tree of G thus takes $\mathcal{O}(|G|)$ time and produces an embedded graph G_1. ☐

A retraction of $\mathcal{S}$ From now on we suppose that G has a single vertex. Let us call an edge ***free*** if it is incident to two distinct faces, exactly one of which is perforated. We will simulate a sequence of elementary collapses in order to retract $\mathcal{S}$ onto a subgraph G' of G. We will thus obtain an embedding of G' into $\mathcal{S}$ such that all faces are perforated. To this end we maintain a list L of free edges. We start by putting all the free edges of G into L. We then pick an edge $e \in L$ and simulate the collapse of its plain incident face by removing e from G and merging its two incident faces. In practice, we just mark e as a ***merging*** edge and tag e as well as its plain incident face f with the name of the incident perforated face. We next update L, removing e from L and adding in or removing from L the other edges of f according to the new status of their incident faces. We repeat this procedure until L is empty. This ensures that all the faces are perforated since otherwise the connectivity of $\mathcal{S}$ would imply the existence of a free edge. Note that the handling of a free edge always involves an incident plain face that was not merged before. It easily follows that the complexity of the whole retraction is bounded, up to a multiplicative constant, by the sum of the lengths of the facial cycles, hence to $|G|$.

We call G' the resulting embedded graph, *i.e.*, the graph G minus the merging edges. If b is a facial cycle of G' of length $|b|$ and $i \in \mathbb{Z}/|b|\mathbb{Z}$ we denote by $b[i]$ the $(i+1)$-th edge of b. An ***abstract subwalk*** of b — or just subwalk when there is no ambiguity — is a triplet $(i,j)_b$ where $i,j \in \mathbb{Z}/|b|\mathbb{Z}$. The ***underlying path*** of $(i,j)_b$ is the path $b[i]b[i+1]\cdots b[j]$. Call E_b the set of merging edges tagged with the facial cycle b. Those edges are incident to a tree T_b of faces (also tagged with b) of G whose union is bounded by b and only one among those faces is perforated. Any $e \in E_b$ cuts b into two subpaths b_p, b_e such that the concatenation $b_p \cdot e$ surrounds the perforated face. Clearly, e retracts onto b_e. We can express b_e as an abstract subwalk w_e of b as follows. When the merging edge e is removed during the retraction phase we keep two pointers from e to the previous and next edge in the incident plain face f to be collapsed. Those pointers delimitate the complementary subpath of f onto which e retracts. We can differentiate a `start` and an `end` between those pointers by taking into account the orientation of the incident perforated face and the signature of e. At the end of the whole retraction we can obtain w_e by following the `start` and `end` pointers respectively, until we hit a non-free edge. We summarize the discussion into the following

Proposition 3 *Let G be a graph embedded on a surface $\mathcal{S}$ with at least one boundary. In $\mathcal{O}(|G|)$ time we can compute:*

- *a subgraph G' of G on which $\mathcal{S}$ retracts,*
- *a set B of boundary cycles, one per boundary cycle of G',*
- *for each oriented edge $e \in G$, an abstract subwalk $(i,j)_b$ whose underlying path is the deformation retract of e onto G', where $b \in B \cup B^{-1}$.*

4 Reducing a sequence of subwalks

The ***length*** $|a|$ of an (abstract) subwalk a is the length of its underlying path. The ***underlying cycle*** of a sequence of subwalks is the cycle obtained by concatenation of the individual underlying paths. A sequence of subwalks is ***reduced*** if its underlying cycle is.

Our goal is to cyclically search and remove spurs, preserving the free homotopy class. We express these simplifications with the following set of rules: for all $(i,j)_b$ and $(k,l)_d$ such that $b[j] = d[k]^{-1}$:

$$(i,j)_b \cdot (k,l)_d \longrightarrow \begin{cases} \epsilon & \text{if } i=j \text{ and } k=l \\ (i,j-1)_b & \text{if } i\neq j \text{ and } k=l \\ (k+1,l)_d & \text{if } i=j \text{ and } k\neq l \\ (i,j-1)_b\cdot(k+1,l)_d & \text{otherwise} \end{cases} \quad (1)$$

The following lemma ensures the correctness of our simulation:

Lemma 4 *Let s be a sequence of subwalks. If s is not reduced then there exist two cyclically consecutive subwalks in s on which one of the above rules apply.*

Proof. Since G' has only one vertex no boundary cycle can contain a spur. ☐

Running the rewriting system until no rule can cyclically apply gives us a new sequence of subwalks whose underlying loop is cyclically reduced and remains in the same free homotopy class. To better control the number of rewrites needed to reach a reduced sequence, we add a special case to the previous rule set:

$$(i,j)_b \cdot (-j-1,l)_{b^{-1}} \longrightarrow \begin{cases} \epsilon & \text{if } |(i,j)_b| = |(-j-1,l)_{b^{-1}}| \\ (i,l-1)_b & \text{if } |(i,j)_b| > |(-j-1,l)_{b^{-1}}| \\ (i+1,l)_{b^{-1}} & \text{if } |(i,j)_b| < |(-j-1,l)_{b^{-1}}| \end{cases} \quad (2)$$

These new rules recognize right away when the second subwalk undoes a whole chunk of the first along the same boundary cycle, and compute in a single step the result of removing spurs until only one subwalk remains. In particular lemma 4 stays true. Rules of this second type take precedence over the rules of set (1); if both types apply then we use a rule of set (2).

Lemma 5 *A path of length 2 in G' appears at most once as a subwalk of boundary cycles.*

Proof. If y follows an oriented edge x in both facial walks containing x, then ρ is an involution around the common vertex v of x and y; in particular v has degree 2. Because G' has only one vertex, G' is a single loop; but then boundary cycles have length 1. ☐

Lemma 6 *Let s_1s_2 be a sequence of two subwalks on which some rule apply. Let s' be the resulting sequence, with precedence taken into account. Then no rule apply on s'.*

Proof. If s' has height 1, no rule can apply. Otherwise a rule of set (1) was used and $s' = (i,j-1)_b\cdot(k+1,l)_d$ where $s_1 = (i,j)_b$ and $s_2 = (k,l)_d$. In particular $b[j] = d[k]^{-1}$. If a rule of set (2) applies on s', then $d = b^{-1}$ and $k+1 = -(j-1)-1$. If a rule of type (1) applies on s', then $b[j-1] = d[k+1]^{-1} = (d^{-1})[-k-2]$ and the subpath $b[j-1]b[j]$ appears both in b at position $j-1$ and in d^{-1} at position $-k-2$. Using lemma 5 we again get $b = d^{-1}$ and $k = -j-1$. This cannot be because no rule of type (2) applied on s. ☐

Lemma 7 *Suppose no rule apply on the sequence s_1s_2. Let s_0 (resp. s_3) be a subwalk such that some rule apply on s_0s_1 (resp. s_2s_3), yielding with precedence a sequence of two subwalks $s'_0s'_1$ (resp. $s'_2s'_3$). Then no rule apply on s'_1s_2 (resp. $s_1s'_2$).*

Proof. The conditions on s'_1s_2 (resp. $s_1s'_2$) are exactly the same as on s_1s_2. ☐

The ***height*** of a sequence of subwalks s is the number $h(s)$ of subwalks composing it. The ***inertia*** of $s = s_1\cdots s_h$, denoted $i(s)$, is the maximum $k \le h$ such that for all $1 \le i \le k$, s_is_{i+1} triggers no rule.[1] If $i(s) = h(s)$ then s is ***cyclically inert***.

Lemma 8 *Let $s = s_1\cdots s_h$ be a sequence of subwalks of inertia $i < h$. Let r be the result of the rules applied on $s_{i+1}s_{i+2}$. If $i < h-1$ let $s' = s_1\cdots s_i\cdot r\cdot s_{i+3}\cdots s_h$ else let $s' = s_2\cdots s_{h-1}\cdot r$. Then $3h(s')-i(s') < 3h(s)-i(s)$.*

Proof. We first suppose $i < h-1$. If $h(s') = h(s)$ then $h(r) = 2$; lemmas 6 and 7 ensure $i(s') \ge i(s)+1$. Else $h(s') \le h(s)-1$ and of course $i(s') \ge i(s)-1$. We now handle the case $i = h-1$. We always have $i(s') \ge i(s)-2$; if $h(s') < h(s)$ the result follows. Otherwise $r = r_1r_2$. By lemma 6 r_1r_2 triggers no rule, and neither do $s_{h+1}r_1$ nor r_2s_2 by lemma 7. Hence $i(s') = h$. ☐

A direct consequence is:

Proposition 9 *Given a sequence s of subwalks we can compute in $\mathcal{O}(h(s))$ time a cyclically inert sequence s' of subwalks whose underlying cycle is freely homotopic to that of s. In particular s' is reduced.*

5 The free homotopy

Let c and d be two cycles on $\mathcal{S}$. Using propositions 3 and 9 we get two sequences s and t of subwalks whose underlying loops c' and d' are freely homotopic to c and d respectively. In particular c and d are freely homotopic if and only if $c' \equiv d'$ up to cyclic permutation. Explicitly comparing the underlying loops is too costly. We thus define a *canonical* representation of any reduced cycle as a sequence of subwalks, and show how to derive this canonical representation from s and t.

[1] By convention $s_{h+1} = s_1$.

A ***boundary mapping*** of an edge $e \in G'$ is any pair (b, i) where $b \in B \cup B^{-1}$ and $i \in \mathbb{Z}/|b|\mathbb{Z}$ such that $b[i] = e$. Every $e \in G'$ has exactly two boundary mappings. We choose an arbitrary total order on $B \cup B^{-1}$. We define as follows the ***canonical mapping*** $\mathrm{CM}(c, e)$ of $e \in c$ with respect to a reduced cycle c. Let p and n be the edges respectively preceding and following e in c. If en is a subpath of some boundary cycle then by lemma 5 there is a unique pair (b, i) such that en occurs at position i in b and we set $\mathrm{CM}(c, e) = (b, i)$. Else, if pe is a subpath of some boundary cycle then $\mathrm{CM}(c, e)$ is the unique (b, i) such that pe occurs at position $i - 1$ in b. Otherwise let $\mathrm{CM}(c, e)$ be the mapping of e with minimal $b \in B \cup B^{-1}$. Two consecutive edges e_1 and e_2 of c are said to ***agree*** with each other if $\mathrm{CM}(c, e_1) = (b, i)$ and $\mathrm{CM}(c, e_2) = (b, i + 1)$. Let c be a reduced cycle and $p = e_1 \cdots e_k \subset c$ a subpath of agreeing edges. Let $(b, i) = \mathrm{CM}(c, e_1)$ and $k = |b|\, q + r$ the Euclidean division of k by $|b|$. The ***leftmost*** sequence of p is $[(i, i-1)_b]^q$ if $r = 0$ and $[(i, i-1)_b]^q \cdot (i, i + r - 1)_b$ otherwise. If c is not the power of a boundary cycle then there is a unique decomposition $c = p_1 \cdots p_h$ into maximal subpaths of agreeing edges, that is where the last edge of p_i does not agree with the first edge of p_{i+1}. The ***canonical sequence*** of c is then the concatenation $\mathrm{Can}(c)$ of the leftmost sequences of $p_1, \ldots, p_h$. If c is the q-th power of a boundary cycle then $\mathrm{Can}(c) = [(0, -1)_b]^q$ is the leftmost sequence of the subpath of c following q times the corresponding boundary walk b. By definition $\mathrm{Can}(c)$ is unique up to circular permutation.

Lemma 10 *Two reduced cycles c and d are equal if and only if $\mathrm{Can}(c)$ and $\mathrm{Can}(d)$ are circular permutations of each other.*

If s is a sequence of subwalks with underlying cycle c then a subwalk $(i, j)_b \in s$ of underlying path $x \subset c$ is ***admissible*** if the l-th edge x_l of x has $\mathrm{CM}(c, w_l) = (b, i + l - 1)$. In particular, x_l agrees with x_{l+1}. A sequence of subwalks is admissible if all its abstract subwalks are. Of course $\mathrm{Can}(c)$ is admissible.

Lemma 11 *Let s is a sequence of subwalks with underlying cycle c. Let $w = (i, j)_b \in s$ with $i \neq j$. Then $w = (i, j-1)_b \cdot (j, j)_b$ where $(i, j-1)_b$ is admissible.*

Two consecutive subwalks $(i, j)_b$ and $(i', j')_{b'}$ in an admissible sequence ***agree*** with each other if $(b', i') = (b, j + 1)$ — in other words the last edge underlying $(i, j)_b$ agrees with the first underlying $(i', j')_{b'}$.

Lemma 12 *Let s is a sequence of subwalks with underlying cycle c. If $s_1 \cdots s_h \subset s$ is a sequence of agreeing subwalks of s then we can compute in $\mathcal{O}(h)$ time the corresponding leftmost sequence.*

Proof. Compute $|s_1| + \cdots + |s_h|$ and divide by $|b|$ ☐

Proposition 13 *Given a reduced sequence s of subwalks with underlying cycle c, we can compute $\mathrm{Can}(c)$ in $\mathcal{O}(|s|)$ time.*

Proof. By computing a single canonical mapping we can replace any subwalk of length 1 by an admissible one. Together with lemma 11 this ensures we can compute an admissible sequence $s' = s'_1 \cdots s'_h$ with underlying cycle c such that $h \leq 2\,|s|$. We then search for some k such that s_k disagrees with s_{k+1}. If there is none then c is the power of a boundary cycle b: return $\mathrm{Can}(c) = [(0, -1)_b]^q$ where $q = \frac{|c|}{|b|} = \mathcal{O}(|s|)$. Otherwise, cut s' into subsequences of agreeing subwalks, computing with lemma 12 and concatenating their respective leftmost sequences . ☐

Now we can prove theorem 1:

Proof. Use propositions 3, 9 and 13 to compute in $\mathcal{O}(h)$ time the canonical sequences of two reduced cycles c' and d' freely homotopic to c and d respectively. c and d are freely homotopic if and only if c' and d' are, which happens if and only if c' and d' are equal as cycles of G', or equivalently if and only if $\mathrm{Can}(c')$ and $\mathrm{Can}(d')$ are cyclic permutations of each other. That last test can be answered in $\mathcal{O}(h)$ time with a Knuth-Morris-Pratt string search of $\mathrm{Can}(c')$ in $\mathrm{Can}(d')\mathrm{Can}(d')$, with the added condition that $h(\mathrm{Can}(c')) = h(\mathrm{Can}(d'))$. Taking the initial preprocessing into account we have the claimed result. ☐

References

[1] É. Colin de Verdière and J. Erickson. Tightening non-simple paths and cycles on surfaces. *SIAM Journal on Computing*, 39(8):3784–3813, 2010.

[2] T. K. Dey and S. Guha. Transforming curves on surfaces. *Journal on Computer and System Sciences*, 58(2):297–325, 1999.

[3] J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. In *Proc. 16th ACM-SIAM Symp. on Discrete Algorithms. (SODA)*, pages 1038–1046, 2005.

[4] F. Lazarus and J. Rivaud. On the homotopy test on surfaces. arXiv:1110.4573v2 [cs.CG], 2011.

[5] W. S. Massey. *A Basic Course in Algebraic Topology*, volume 127 of *Graduate Texts in Mathematics*. Springer Verlag, 1991.

[6] B. Mohar and C. Thomassen. *Graphs on surfaces*. John Hopkins University Press, 2001.