Optimal Algorithms to Detect Null-Homologous Cycles on 2-manifolds

Tamal K. Dey[1]

Abstract

Given a cycle of length k on a triangulated 2-manifold, we determine if it is null-homologous (bounds a surface) in $O(n+k)$ optimal time and space where n is the size of the triangulation. Further, with a preprocessing step of $O(n)$ time which is performed only once, we answer the same query for any cycle of length k in $O(g+k)$ time. This is optimal for $k > g$.

1 Introduction

In recent years a new focus has developed in studying the algorithmic aspects of topology [2, 5, 6, 7], a well developed branch of mathematics. This emergent field has been called "Computational Topology" [5, 7]. It is generally recognized that there is a vast repository of topological problems which have not been studied extensively from algorithmic point of view. In this paper, we address one such problem, namely, topologically distinguishing the curves(cycles) on 2-manifolds that bounds a surface (possibly empty and is not necessarily a disk). These cycles are called *null-homologous* cycles.

The importance of detecting null-homologous cycles comes from two facts. Homology groups of a topological space reveals its connectivity. To compute them efficiently Delfinado and Edelsbrunner [2] showed a geometric approach. Unfortunately this approach did not have any efficient implementation in dimensions higher than three since there is no known polynomial time algorithm to detect null-homologous cycles in higher dimensions. Although this paper does not provide a solution for this general problem, it gives a better understanding of null-homologous cycles on manifolds. Secondly, dectecting null-homologous cycles is related to a more difficult problem called contarctability problem, which asks if a given cycle is contractable to a single point. These cycles are called *null-homotopic* cycles. For 2-manifolds all null-homotopic cycles are null-homologous, though the reverse is not true. Recently, we provided an improved algorithm for detecting null-homotopic cycles on 2-manifolds. This algorithm runs in $O(n+gk)$ time where n is the size of the triangulation, k is the cycle length, and g is the genus of the given 2-manifold. We can use the algorithm for null-homologous cycles first, and declare those cycles not null-homotopic that are not null-homologous. Of course, we have to run the algorithm for null-homotopic cycles when a cycle is detected to be null-homologous. This approach saves time in those cases where the cycle is not null-homologous since null-homologous cycles can be detected in $O(n+k)$ time as shown here.

2 Preliminaries

A *2-manifold* is a topological space in which each of its points has a neighborhood homeomorphic to an open disk. A 2-manifold can be infinite or finite. Moreover, it can be closed or open depending

[1]Computer Science, Indiana University-Purdue University at Indianapolis, Indianapolis, IN 46202, USA.

on whether its closure coincides with itself or not. A closed and bounded 2-manifold is also called a compact 2-manifold. A Sphere and a klein bottle are two examples of compact 2-manifolds. A 2-manifold is called orientable if it has two distinct sides. Otherwise, it is non-orientable. For details see [4]. In this paper we consider only compact 2-manifolds.

A 2-manifold is triangulable in the sense that it can be represented as the union of a set of triangles, edges and vertices satisfying the following properties. Each pair of triangles either share a single vertex or a single edge, or are completely disjoint. Also, the triangles incident on a vertex can be ordered circularly so that two triangles share a common edge if and only if they are adjacent in this ordering.

2.1 Chains and Boundaries

A precise definition of null-homologous cycles can be given algebraically. We use the notations of [4] for this purpose. Although we will be working with the triangulations of 2-manifolds which are simplicial 2-complexes, we define all terms here in the most general setting. Let K be a simplicial complex. An oriented simplex $[v_0, v_1, ..., v_p]$ is the p-simplex with vertices $v_0, v_1, ..., v_p$ and the particular ordering on these vertices, $v_0v_1...v_p$. A p-chain on K is a function c from the set of oriented p-simplices of K to the integers where

1. $c(\sigma) = -c(\sigma')$ if σ and σ' are opposite orientations of the same simplex.
2. $c(\sigma) = 0$ for all but finitely many oriented p-simplices σ.

For an oriented simplex σ, the elementary chain c is the function defined as

$c(\sigma) = 1$
$c(\sigma') = -1$ if σ' is the opposite orientation of σ.
$c(\tau) = 0$ for all other oriented simplices τ.

We use the symbol σ to denote not only a simplex, or an oriented simplex, but also to denote the elementary p-chain c for the oriented simplex σ. With this notation we write a p-chain d as $d = a_1\sigma_1 + a_2\sigma_2 + ... + a_k\sigma_k$ where a_i's are integer coefficients.

If $\sigma = [v_0, v_1, ..., v_p]$ is an oriented simplex with $p > 0$, we define $\delta_p\sigma = \sum_{i=0}^{p}(-1)^i [v_0, ..., \hat{v}_i, ..., v_p]$, where the symbol $\hat{v}_i$ means that the vertex v_i is to be deleted from the array. The interpretation of δ_p is that it finds the oriented boundary of an oriented simplex. For example, an oriented triangle $[v_0v_1v_2]$ gives the oriented cycle $[v_0v_1]+[v_1v_2]+[v_2v_1]$ when operated by δ_2. For a p-chain $d = \sum a_i\sigma_i$, $z = \delta_p d = \sum a_i\delta_p\sigma_i$ is called the $(p-1)$-boundary of d. We also say z bounds d. A p-chain d is called a p-cycle if $\delta_p d = 0$.

With the addition p-chains form a group $C_p(K)$, called the group of p-chains of K. If $p < 0$ or $p > dimK$, we let $C_p(K)$ denote the trivial group. These groups are abelian [4]. The function $\delta_p : C_p(K) \rightarrow C_{p-1}(K)$ is an homomorphism. The set of p-chains that are mapped by δ_p to the identity element of $C_{p-1}(K)$ are called the group of p-cycles and denoted $Z_p(K)$. In other words, the kernel of δ_p is $Z_p(K)$. The image of $\delta_{p+1}: C_{p+1}(K) \rightarrow C_p(K)$ is called the group of p-boundaries and is denoted $B_p(K)$. A careful observation reveals that $B_p(K) \subset Z_p(K)$. The quotient group $H_p(K) = Z_p(K)/B_p(K)$ is called the pth homology group of K. A p-cycle z in K is null-homologous if it represents an identity in $H_p(K)$. Stated otherwise, a p-cycle z is null-homologous if there exists a $p+1$-chain d so that z bounds d. So a 1-cycle z on a triangulated 2-manifold is null-homologous if there is a 2-chain d bounded by z. Two p-cycles z_1, z_2 are called *homologous* if there exists a $p+1$-chain d so that $z_1 - z_2 = \delta_p d$.

2.2 Polygonal Schema

Any orientable or non-orientable 2-manifold can be represented by a simple polygon P with even number of edges which is also called a *polygonal schema* of S. Each edge of P has a signed label such that each unsigned label occurs twice. See [3] for details. Two edges with the same unsigned labels are called *partnered edges.* Partnered edges can have labels with the same or opposite signs. Two partnered edges with labels $+x$ and $-x$ represent the same edge on S but are oppositely directed on P. The labels of the edges that are directed in a clockwise direction around P are signed positively. In general, we use x^c to denote the complement of the label x. To reconstruct a surface homeomorphic to S from this polygonal representation, the oriented edges with the same labels are identified together in such a way that their orientations match. For simplicity, we say that S is obtained from P by identifying partnered edges appropriately.

An orientable 2-manifold with genus $g > 0$ can be represented *canonically* by a $4g$-gon where the labels on the edges around the polygon are of the form: $x_1 y_1 x_1^c y_1^c x_2 y_2 x_2^c y_2^c \ldots x_g y_g x_g^c y_g^c$. For $g = 0$, the 2-manifold is a sphere which can be represented canonically by two directed edges xx^c.

Similarly, a non-orientable 2-manifold can be represented *canonically* by a $2g$-gon where the labels on the edges around the polygon are of the form: $x_1 x_1 x_2 x_2 \ldots x_g x_g$.

2.3 Cycles and Null-homology

Let T denote a triangulation of an orientable 2-manifold S. Let C be any given cycle (oriented) on T. The cycle C is a sequence of oriented edges with the first and the last edge meeting at a vertex. Let P be a polygonal schema of S with a triangulation T' such that there is a one-to-one correspondence between triangles of T and T'. The following lemma proves that such a polygonal schema can be constructed.

LEMMA 2.1 A polygonal schema P with triangulation T' can be constructed from T where there is a one-to-one correspondence between triangles of T' and T.

PROOF. We construct a sequence of closed disks $D_1, D_2, \ldots, D_n$ in a plane incrementally such that $P = D_n$ at the end. Initially, $D_1 = \sigma'$, a triangle in the plane that corresponds to an arbitrarily chosen triangle σ_1 on M. Let $D_i = \sigma_1' \cup \sigma_2' \ldots \cup \sigma_i'$ after the *ith* step. At the $i+1$ *th* step we choose a triangle σ_{i+1} on T which has the following properties: (i) no triangle corresponding to σ_{i+1} has been included in D_i, and (ii) a triangle σ_j adjacent to σ_{i+1} by an edge has a corresponding triangle in D_i. These two conditions imply that there is an edge $e = \sigma_j \cap \sigma_{i+1}$ such that its corresponding edge e' on D_i appears on $bd(D_i)$[2]. We attach a triangle σ'_{i+1} to $bd(D_i)$ such that $\sigma'_{i+1} \cap bd(D_i) = e'$. This gives the new disk $D_{i+1} = D_i \cup \sigma'_{i+1}$. It is clear that if D_i is a closed disk, so is D_{i+1}. Finally, when we exhaust all triangles on M, we have D_n with the desired triangulation T'.

□

Let G be the graph on T obtained by identifying the partnered edges of $bd(P)$. If C intersects the edges of G, C can be written as a sequence of directed paths in T' with end points on $bd(P)$. Let $U = \{u_1, u_2, ..., u_r\}$ be this sequence of paths. A directed path u_i from v_1 to v_2 can be deformed to another directed path u'_i that lies entirely on $bd(P)$ and to the left of u_i. Let $U' = \{u'_1, u'_2, ..., u'_r\}$. The paths in U' correspond to a cycle C' on G. One can think of this process as deforming C to another cycle C' on T so that all edges of C' lie on the edges of G. We also say that C' is *carried by* $bd(P)$. In terms of homology, C and C' are homologous since $C - C'$ is a 2-chain consisting of the triangles between u_i and u'_i for $i = 1, ..., r$. If C does not intersect G, then it is trivially null-homologous since the corresponding cycle in T' lies in $int(P)$[3] which is an open disk. Henceforth, we consider only the difficult case where C intersects G.

Let $x_1, x_2, ..., x_t$ be the edges on G. Let $a_1, a_2, ..., a_{2t}$ be the sequence of signed labelled edges around (clockwise) $bd(P)$. This means $a_i = (+/-)x_j$ for some j. Consider the 1-chain $\beta = a_1 + a_2 + ... + a_{2t} = c_1x_1 + c_2x_2 + ... + c_tx_t$ where $c_i \in (-2, 0, 2)$. Let the weight w_i on the edge x_i be the number of times it is traversed in the clockwise direction minus the number of times it is traversed in the counter-clockwise direction on $bd(P)$ by the paths in U'. Consider the vectors $w = [w_1, w_2, ..., w_t]$ and $c = [c_1, c_2, ..., c_t]$. The following lemma serves as a main tool in our algorithm.

LEMMA 2.2 C is null-homologous if and only if $w = mc$ for some integer m.

PROOF. We prove that C' is null-homologous if and only if $w = mc$. Since C and C' are homologous the lemma follows.

Let γ denote the sum of the oriented triangles in T' where each triangle is oriented clockwise. If $w = mc$, the cycle C' represents a 1-chain that is a multiple of β. Since $\beta = \delta_2\gamma$, we have $C' = m\beta = m\delta_2\gamma = \delta_2(m\gamma)$ proving C' null-homologous. To prove the other direction, assume C' null-homologous. There must exist a 2-chain d such that $C' = \delta_2 d$. Also, C' is carried by $bd(P)$. Any 2-chain d with $\delta_2 d$ carried by $bd(P)$ must be a multiple of γ, see [4]. Thus $C' = \delta_2 d = \delta_2(m\gamma) = m\delta_2\gamma = m\beta$. This immediately implies $w = mc$. □

3 Algorithm

The polygon P is computed in $O(n)$ time. While computing P, we maintain pointers between corresponding triangles and edges of T and T'. We traverse the cycle C on T and detect the paths

[2] $bd(*)$ denotes the boundary of $*$

[3] $int(*)$ denotes the interior of $*$

$U = \{u_1, u_2, ..., u_r\}$ on T' in $O(k)$ time through pointers. The paths $U' = \{u'_1, u'_2, ..., u'_r\}$ on $bd(P)$ can easily be detected from the end points of the paths in U. Let j_a denote the number of times an edge a on $bd(P)$ is traversed in the clockwise direction minus the number of times it is traversed in the anti-clockwise direction. Then $w_i = j_a + j_{a'}$ where a and a' are the partnered edges with label x_i. We can count j_as for all edges on $bd(P)$ by traversing the paths in U'. However, the total length of the paths in U' can be $\Omega(kn)$ since each edge on C can contribute a path of length $\Omega(n)$ on $bd(P)$. To avoid this complexity increase, we count j_as as follows.

Let $v_1, v_2, ..., v_{2t}$ be the sequence of vertices around $bd(P)$ in the clockwise direction. In what follows all operations on subscripts are done *modulo* $2t$. Let $a = v_i v_{i+1}$ for some $i \in \{1, 2, ..., 2t\}$ which is traversed in clockwise direction from v_i to v_{i+1}. To count the number of paths in U' traversing a in the clockwise direction, we have to count the number of paths starting from v_i and the number of paths that have v_i inside. Let $\ell_{v_{i-1}}$ be the number of paths that traverse the edge $v_{i-1}v_i$ from v_{i-1} to v_i. Then the number of paths traversing a in the clockwise direction is $\ell_{v_i} = \ell_{v_{i-1}} + o_{v_i} - i_{v_i}$, where o_{v_i} is the number of paths starting at v_i and i_{v_i} is the number of paths ending at v_i. To compute ℓ_{v_i}, it is enough to know o_{v_i} and i_{v_i} for each vertex v_i and ℓ_{v_1} for the vertex v_1. Computing o_{v_i} and i_{v_i} is straightforward. For each end vertex of the paths in U (equivalently in U') we count the number of paths starting from that vertex and the number of paths ending at that vertex. For all other vertices o_{v_i} and i_{v_i} are zero. To compute ℓ_{v_1}, we number the vertices $v_1, v_2, ..., v_{2t}$ with integer indices $1, 2, ..., 2t$. Now it is simple to detect the paths in U' that include v_1 in between. For this, we check in constant time if 1 is in between the integer indices of the two end vertices of a path according to the circular sequence $1, 2, ..., t, 1$. The number of such paths added with o_{v_1} gives ℓ_{v_1}.

Once we have computed ℓ_{v_i} for each vertex v_i, it is straightforward to compute the vector w. We declare C to be null-homologous if and only if $w = mc$ for some integer m (Lemma 2.2). Computing o_{v_i}, i_{v_i} and ℓ_{v_1} while going around C (equivalently over the paths in U) takes $O(k)$ time if C has the length k. Computing ℓ_{v_i}'s while moving around $bd(P)$ takes $O(t)$ time. The vector c can be precomputed from P in $O(t)$ time. Checking if $w = mc$ takes at most $O(t)$ time. Since $t = O(n)$ we get an $O(n + k)$ time complexity algorithm.

THEOREM 3.1 Let T be a triangulation of a compact 2-manifold of genus g. Given a cycle C of length k on T, there exists an $O(n + k)$ optimal algorithm that detects if C is null-homologous.

4 On-line Queries

In an on-line setting, we are supposed to have a preprocessing step and then answer the null-homology of the query cycles which come on-line. For this we use the property of another invariant group of topological spaces called the *fundamental group*. Let P be the polygonal schema of S as constructed in Lemma 2.1 and G be the graph obtained by appropriately identifying edges of $bd(P)$. We construct a spanning tree Y of G. Let $b_1, b_2, ..., b_\ell$ be the edges of G not in Y and $b'_1, b'_2, ... b'_{2\ell}$ be the signed edges around $bd(P)$ corresponding to these edges. In [1], we proved that $b_1, b_2, ..., b_\ell$ represent the generators of the fundamental group of S with respect to the relation $b'_1, b'_2, ..., b'_{2\ell} = 1$. Also it is proved that $\ell = O(g)$. One dimensional homology group of a topological space can be obtained from its fundamental group by making the group operation commutative. Therefore $b_1, b_2, ..., b_\ell$ are also generators of $H_1(S)$ with the relation $b'_1 + b'_2 + ... + b'_{2\ell} = c_1 b_1 + c_2 b_2 + ... + c_\ell b_\ell = 1$ for $c_i \in \{-2, 0, 2\}$.

Let w_i be the weight of the edge b_i obtained from the paths $u'_1, u'_2, ..., u'_r$. The given cycle C is null-homologous if and only if $w = [w_1, w_2, ..., w_\ell] = m\,[c_1, c_2, ..., c_\ell]$ for some integer m. To compute w efficiently, we create another polygon Q from P in a preprocessing step as follows. The polygon Q has 2ℓ edges labeled $b'_1, b'_2, ..., b'_{2\ell}$ around it. We maintain pointers from the vertices of P to the vertices of Q as follows. Let $b'_i b'_{i+1}$ be any two consecutive edges in the sequence $b'_1 b'_2 ... b'_{2\ell}$ and $v_1 v_2 ... v_s$ be the vertices between b'_i and b'_{i+1} where v_1 is an endpoint of b'_i and v_s is an endpoint of b'_{i+1}. All these vertices point to the same vertex v between b'_i and b'_{i+1} in Q. The polygon Q can be thought of as the polygon P with all edges in the spanning tree Y shrunk to a single vertex. The path u'_i from v_1 to v_2 on $bd(P)$ is represented by the path u''_i from v'_1 to v'_2 on $bd(Q)$, where v'_1, v'_2 are the vertices on $bd(Q)$ corresponding to the vertices v_1, v_2 respectively. The vertices v'_1, v'_2 can be obtained from v_1, v_2 in $O(1)$ time through pointers. Applying similar technique of section 3 on $bd(Q)$ we can determine w_is in $O(g)$ time. The preprocessing step to create P and Q takes at most $O(n)$ time altogether. Given any query cycle C on line, the end vertices of u'_i and hence u''_i can be detected in $O(k)$ time. This is followed with an $O(\ell) = O(g)$ step to compute w_is and checking if $[w_1, w_2, ..., w_r] = m\,[c_1, c_2, ..., c_\ell]$. Hence on-line queries can be processed in $O(g + k)$ time once we have an $O(n)$ time preprocessing step.

5 Conclusions

We have presented an optimal algorithm to detect null-homologous cycles on triangulated 2-manifolds. We have also given an efficient solution for the on-line version of the problem. However, this algorithm is not optimal for $g > k$. Finding an optimal $O(k)$ algorithm for this problem remains open. A challenging problem is to dectect null-homologous cycles of arbitrary dimensions on arbitrary complexes. An efficient solution to this problem provides an efficient method to compute homology groups of simplicial complexes [2].

References

[1] T. K. Dey. A new technique to compute polygonal schema for 2-manifolds and null-homotopy detection. (1993), Manuscript.

[2] H. Edelsbrunner and C. Delfinado. An incremental algorithm for betti numbers of simplicial complexes. *Ninth Ann. Sympos. Comput. Geom.* (1993).

[3] W. S. Massey. Algebraic Topology: An Introduction. *Springer-Verlag.* New York (1967).

[4] J. R. Munkres. Elements of Algebraic Topology. *Addison-Wesley Publishing* (1984).

[5] H. Schipper. Determining contractibility of curves. *Proc. 8th ACM Sympos. on Comput. Geom.* (1992), 358–367.

[6] J. S. Snoeyink. Topological approaches in computational geometry. *Tech. Report STAN-CS-90-1331* (1990).

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