Pattern Recognition Letters xxx (2011) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A fast algorithm to compute cohomology group generators of orientable 2-manifolds

Paweł Dłotko [1]

Institute of Computer Science, Jagiellonian University, Łojasiewicza 6, 30-348 Kraków, Poland

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Computational cohomology
Cohomology generators
Combinatorial 2-manifold

ABSTRACT

In this paper a fast algorithm to compute cohomology group generators of cellular decomposition of any orientable open or closed 2-manifold is described. The presented algorithm is a dual version of algorithm to compute homology generators presented by David Eppstein in "Dynamic generators of topologically embedded graphs" and developed by Jeff Erickson and Kim Whittlesey in "Greedy optimal homotopy and homology generators". Some parts of the paper bases on ideas presented in "Optimal discrete Morse functions for 2-manifolds" by Thomas Lewiner, Helio Lopes and Geovan Tavares. Extension of the algorithm to some non-manifold cases is provided.

1. Introduction

In a past 20 years computational homology theory has gained a considerable attention in the computer science community. The Betti numbers and homology generators has been used in many areas of mathematics like dynamical systems (Mischaikow and Mrozek, 2002) and outside for instance in the material sciences (Day et al., 2009). A number of programming libraries (CAPD, 2002–2010; CHomP, 2002–2010) and algorithms has been developed for Betti numbers and homology group generators computations. Among others, a big progress has been done on computing the minimal (with respect to the considered metric) homology and homotopy generators of cellular decomposition of orientable closed 2-manifolds, see Eppstein (2004), Erickson and Whittlesey (2005). There are also known algorithms to compute homology group generators of 2 and 3 dimensional complexes based on graph pyramids (Peltier et al., 2009) and simplification operations (Damiand et al., 2006, 2008).

Until recently the computational aspect of the cohomology theory has been left apart. In our opinion the reason was the conceptual difficulty of the cohomology theory. The first algorithm to compute cohomology group generators together with the cup product, which provides a ring structure of cohomology has been presented in (González-Diáz and Real, 2003). This algorithm is based on a concept of chain contraction. Another algorithm to compute cohomology generators of 2d images by using irregular graph pyramids has been discussed in (González-Diáz et al., 2009).

Recently the first cohomology group generators of 3 dimensional manifolds with boundary has been shown to be useful in discrete geometrical approach to Maxwell's equations (Dłotko and Specogna, 2010a). Also cohomology generators are useful in image context like global mesh parametrization, texture mapping, shape matching and shape morphing (Gu and Yau, 2002; Gu et al., 1996; Guo et al., 2006; Desbrun et al., 2008) just to name some of areas.

The algorithm to compute cohomology generators presented in (Dłotko and Specogna, 2010a) is based on a number of reduction techniques originally designed for homology computations (Mrozek et al., 2008; Mrozek and Batko, 2009) called shavings, algebraic reduction (Kaczynski et al., 1998) and finally standard Smith Normal From computations. Consequently it can be applied to any finite simplicial, cubical or general CW complex. Generality however has its computational cost. Therefore one may wonder if in case of 3-dimensional computer graphics where 2-dimensional orientable manifolds are considered there exist quicker way of obtaining cohomology generators? In this paper we are to present a dual version of Eppstein (2004) algorithm which provide cohomology group generators for combinatorial surfaces (i.e. cell decompositions of closed orientable 2-manifolds). Also a modification inspired by Erickson and Whittlesey (2005) to obtain minimal length cohomology generators intersecting in a chosen 2-face is discussed. Moreover the extension of the presented methods to open 2-manifolds and some non manifold cases is provided. The intuition how homology and cohomology generators differ is presented in Fig. 2.

In this paper a main focus is given to computing first cohomology group generators of cellular decomposition of 2-manifolds. The reason is because for 2-manifolds zeroth and second cohomology group are known a priori. Later in this paper, without loose of generality, we will work on connected manifolds. In that case the rank

E-mail address: pawel.dlotko@uj.edu.pl

[1] The author is supported by MNiSW Grant N N206 625439 and partially funded by the Google Research Awards Program.

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.10.005

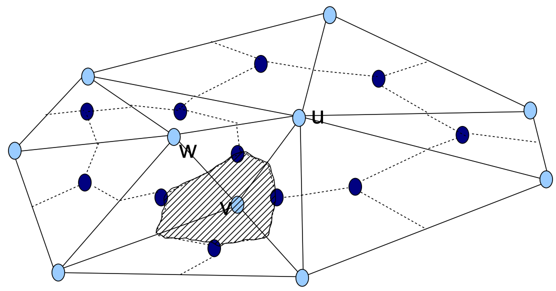

Fig. 1. Dual cell structure on a piece of triangulation of 2-manifold. Black dots denote the dual vertices, broken lines – dual edges. With a hatched 2-cell a cell dual to the primal vertex v is presented.

of both zeroth and second cohomology group is one when closed manifold is considered. For open manifolds the rank of zeroth cohomology group is one and the rank of second cohomology group is zero. The zeroth cohomology group generator can be represented by a map sending every vertex of the manifold to 1. The second cohomology group generator, for a closed orientable manifold can be represented by a map sending single 2-cell to 1 and all remaining 2-cells to 0. Also testing if a complex is open or closed manifold can be done in linear time. Therefore zeroth and second cohomology group generators can be easily obtained for orientable manifolds.

The paper is structured as follows: In Section 2 some preliminaries are provided. In Section 3 the Eppstein algorithm (Eppstein, 2004) is reviewed. In Section 4 an algorithm to compute cohomology generators is presented. In Section 5 necessary modifications to obtain optimal cohomology generators are discussed (optimality in a sense, that the length of cut is minimal, see Fig. 2). In Section 6 the case of open manifold is reviewed. In Section 7 some more general non-manifold cases of sets and various coefficient fields for which the presented algorithm can return cohomology generators are presented. In Section 8 a short complexity analysis of the presented algorithms is given. Finally in Section 9 the conclusions are drawn.

2. Preliminaries

A closed *2-manifold* is a topological space in which every point has a neighborhood homeomorphic to $\mathbb{R}^2$. Manifold is said to be orientable if one can assign an orientation to all its 2-cells in a consistent way, see Hatcher (2002). As in (Colin de Verdiere and Lazarus, 2003) we will work on *combinatorial surfaces* $M = (\mathcal{M}, G)$, where $\mathcal{M}$ is a 2-manifold and G is a weighted graph embedded on $\mathcal{M}$ such that every face of an embedding is a topological disk. The weight of an edge in G is equal to the length of a corresponding path in $\mathcal{M}$.

For a combinatorial surface $M = (\mathcal{M}, G)$ let us define a dual graph $G^* = (V,E)$, where V is the set of barycenters of faces of G and a dual edge e^* is placed between two dual vertices if their corresponding primal faces share an edge in G.

It is clear, that classical concepts of simplicial or cubical complexes on 2-manifold are in fact combinatorial surfaces with a graph G being their 1-skeleton.

Below informal introduction to (co)homology theory is given. Let M_i denote a set of i-dimensional elements in M. A group of formal sums $\sum_{e\in M_i}\alpha_e e$ for α_e being elements of coefficient group is called the ith chain group of M and denoted by $C_i(M)$. A cochain group $C^i(M)$ is formally defined as a group dual to $C_i(M)$ i.e. group of maps from $C_i(M)$ to a coefficients group (integers for sake of simplicity). Note that a cochain $c^* \in C^i(M)$ can be represented by a chain. More specifically, one may think of a cochain $c^* \in C^i(M)$ as a chain $\sum_{s\in M_i}\langle c^*, s\rangle s$. In this case computing the value of cochain

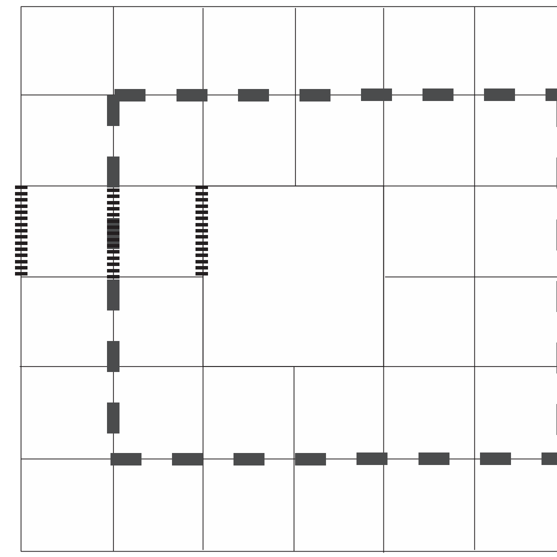

Fig. 2. An intuition of a first homology and cohomology group and their generators. Let us consider an annulus in the plane. The rank of first (co)homology group is equal to the number of holes in the considered set, one in this case. The homology generator is any cycle that surround the hole one time (for example the one depicted by dotted cycle). An example of cohomology generator is depicted with three black, densely dotted edges. One may think of a cohomology generator as a minimal set of edges that block any cycle surrounding a hole. In this case the cohomology generator is dual to the homology generator as it will be explained later in the paper.

c^* on a chain $c(\langle c^*,c\rangle)$ is equivalent to computing scalar product $\left\langle \sum_{s\in M_i}\langle c^*, s\rangle s, c\right\rangle$. Due to this correspondence in this paper we will notation $\langle,\rangle$ both to denote evaluation of a cocycle on a cycle and a scalar product of two cycles.

Let us take any $a,b \in M$ and define $\kappa^T(a,b) = \kappa(b,a)$ (where $\kappa(a,b)$ denotes the given incidence between cells a and b. Note the roles of a and b are "swapped" in the lhs and rhs of the formula). Once the incidence is provided we can give a definition of boundary and coboundary map with $\partial(b) = \sum_{a\in M}\kappa(b,a)a$ and $\delta(a) = \sum_{b\in M}\kappa^T(b,a)b$ for $dim(b) = dim(a)+1$ on a basis of group of (co)chains and extend lineally to (co)chains.

With (co)boundary operator we are able now to define (co)cycles and (co)boundaries. A group $Z_i(M) = \{e \in C_i(M) | \partial e = 0\}$ is called a group of i-cycles. A group $B_i(M) = \{e \in C_i(M)|$ there exist $d \in C_{i+1}(M) | \partial d = e\}$ is called a group of i-boundaries. A quotient group $H_i(M) = Z_i(M)/B_i(M)$ is a ith homology group. Dually a group $Z^i(M) = \{e \in C^i(M) | \delta e = 0\}$ is called a group of i-cocycles. A group $B^i(M) = \{e \in C^i(M)|$ there exist $d \in C^{i-1}(M) | \delta d = e\}$ is called a group of i-coboundaries. A quotient group $H^i(M) = Z^i(M)/\ B^i(M)$ is a ith cohomology group. A basic intuition of the first homology and cohomology groups and generators is presented in Fig. 2. As one can see (co)cycles defined as formal sums with coefficients may be considered on various coefficients fields. In this way (co)homology groups over various coefficient groups are defined. For further details consult (Hatcher, 2002).

In this paper a classical version of Poincaré duality for manifolds is used. This presentation follows the line in (Bredon, 1993). Let us define a dual cell structure M^* on 2-manifold $\mathcal{M}$ with a cell structure M. The dual cell structure is defined based on a dualization operator D: $M \to M^*$. As in case of dual graph the barycenters of faces $f \in M$ corresponds to the vertices $D(f)$ of M^*. For faces $f_1, f_2 \in M$ intersecting in an edge $e = f_1 \cap f_2 \in M$ two segments of line joining $D(f_1)$ with a barycenter of e and $D(f_2)$ with barycenter of e form together a dual edge $D(e)$. For every vertex $v \in M$ let $e_1, \ldots, e_n \in M$ be all the edges having $e_i \cap v \neq \emptyset$ for every $i \in \{1, \ldots, n\}$. The polygon bounded by $D(e_1), \ldots, D(e_n)$ is

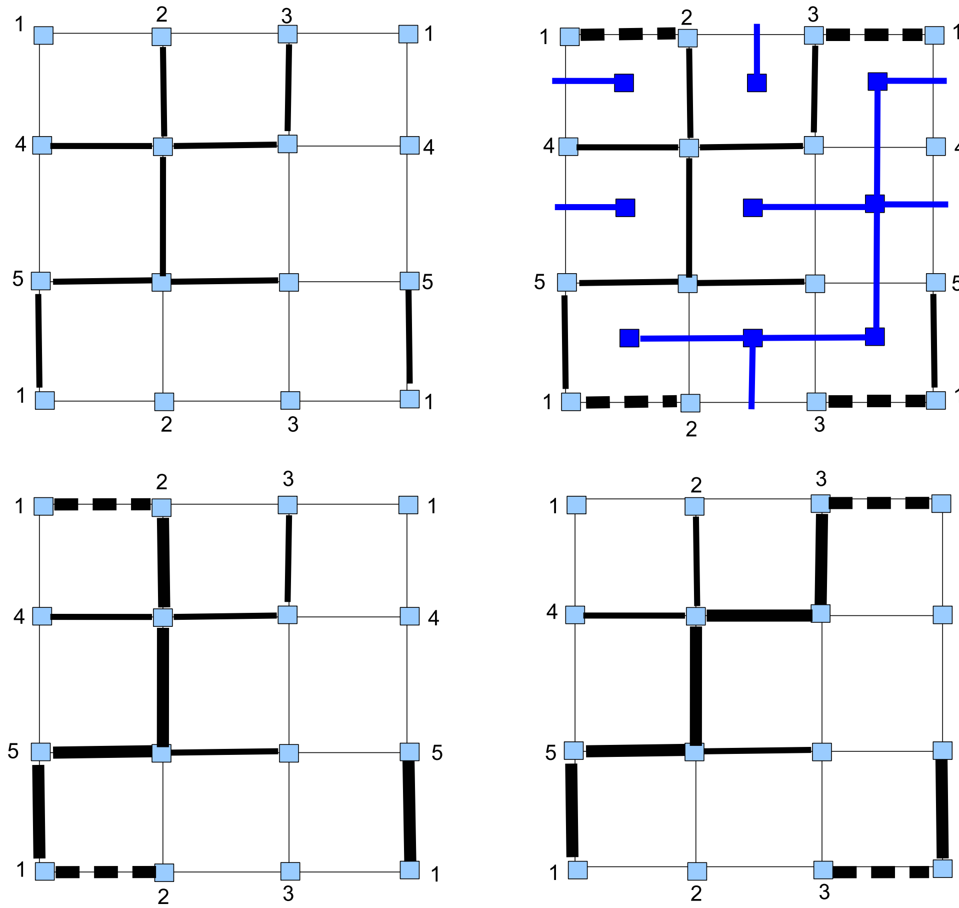

Fig. 3. The Eppstein algorithm. On the upper left the spanning tree on a primal complex of a standard cubical representation of a torus. On the upper right the spanning tree on a dual graph is added (in dotted line the edges which are neither in primal nor dual graph). On the bottom row two first homology group generators closed by the dotted edges and the tree of the primal complex are indicated with the double bold continuous and dotted edges.

by definition $D(v)$ (see Fig. 1 where $D(v)$ is depicted). Further let us assume, that the cell structure M is provided with a fixed orientation of every cell (provided as an incidence index κ). By a skeleton of dual cell structure we mean all dual vertices and edges. It is straightforward, that 1-skeleton of dual cell structure and the dual graph are homeomorphic (although not equal). Therefore further those concepts will be used interchangeably.

Note the bijective correspondence between M and M^* provided by the map D.

Note that for each p-chain $c \in C_p(M)$ a $(n-p)$-chain $D(c) \in C_{n-p}(M^*)$ (where n is a dimension of manifold) is assigned. Moreover $\delta D(c) = D(\partial c)$ due to the definition of the incidence index in the dual complex. Therefore the δ operator on the primal cell structure is mapped through D to ∂ operator on the dual cell structure. Therefore we have $H^{n-p}(M) \approx H_p(M^*)$. It is also clear that $H_p(M) \approx H_p(M^*)$ since M and M^* have the same based manifold. By putting those two results together the *Poincaré duality* is obtained:

$$H_p(M) \approx H^{n-p}(M)$$

Similar argument guiding to the Poincaré duality has been also presented in (Edelsbrunner and Harer, 2010).

3. Eppstein algorithm

In this section the Eppstein algorithm (Eppstein, 2004) is recalled. For the theoretical background and motivation consult (Eppstein, 2004). In this section a pure algorithm and its illustration is provided.

Let M be a cell structure on a orientable 2-manifold $\mathcal{M}$ and $M^* = D(M)$. The steps of Eppstein algorithm are as follows:

1. Let $T \subset M$ be a spanning tree of 1-skeleton of M.
2. Let $T^* \subset (M^* \setminus \cup_{e \in T} D(e))$ be a spanning tree of 1-skeleton of $(M^* \setminus \cup_{e \in T} D(e))$.[2]
3. Let $E = \{e \in M$ such that e is an edge and $e \notin T$ and $D(e) \notin T^*\}$.
4. For a edge $e \in E$ let $C(e)$ denote the unique cycle in $T \cup e$. The set of cycles $\{C(e) | e \in E\}$ forms a basis of the first homotopy group of M (see Erickson and Whittlesey, 2005).[3]

The presented algorithm returns in fact the first homotopy group generators of the considered manifold. However when one construct a cycle based on the loops representing fundamental group basis, one obtain a first homology group basis of the surface. This is a consequence of Hurewicz theorem (Greenberg, 1967). Let us indicate, that the Hurewicz theorem is applicable only for homology computed over $\mathbb{Z}$ coefficients. However the Universal Coefficients Theorem for Homology stays that the following exact sequence exists (see Th. 3A.3 Hatcher, 2002):

$$0 \to H_1(M, \mathbb{Z}) \otimes G \to H_1(M, G) \to Tor(H_0(M), G) \to 0$$

Since $H_0(M)$ is a free group we have, that $Tor(H_0(M), G) = 0$. Consequently one have an isomorphism $H_1(M, \mathbb{Z}) \otimes G \approx H_1(M, G)$. Due

[2] A dual tree will be also referred to as cotree. The presented technique will be also referred to as tree–cotree decomposition.

[3] Formally one should require, that all the cycles share a common base point. However since our aim are (co)homology generators, we can remove this requirement.

Please cite this article in press as: Dłotko, P. A fast algorithm to compute cohomology group generators of orientable 2-manifolds. Pattern Recognition Lett. (2011), doi:10.1016/j.patrec.2011.10.005

to the classification theorem of orientable 2-manifolds $H_1(M,\mathbb{Z})$ is torsion free. Due to the property of tensor product $\mathbb{Z}\otimes A\approx A$, for A being a abelian group. In this paper we will restrict to the case of A being $\mathbb{Z}_p,\mathbb{Q},\mathbb{R}$ or $\mathbb{C}$. Therefore we can compute homology over any group $\mathbb{Z}_p,\mathbb{Q},\mathbb{R},\mathbb{C}$ by simply treating the integers −1, 0, +1 being the coefficients of the representatives of generators as elements of the suitable group. The supports of generators remains unchanged. Therefore we have a straightforward isomorphism between homology over $\mathbb{Z}$ and over $\mathbb{Z}_p,\mathbb{Q},\mathbb{R},\mathbb{C}$ group of coefficients. Therefore from the Hurewicz theorem the presented algorithm, apart from homotopy group generators, give (co)homology generators over presented coefficient groups.

To obtain homology generators one need to construct set of chains the support of which are the cycles $\{C(e)\}_{e\in E}$. This can be easily done by "walking around" a cycle $C(e)$ and assigning to an edge e a value 1 if it is crossed according to its fixed orientation and −1 if it is crossed in the direction opposite to its fixed orientation (note that the cycles do not have self intersections). The straightforward details of the procedure are left for a reader.

4. Algorithm to compute cohomology generators

In this section we show that, surprisingly, from Eppstein algorithm one is able to obtain also the cohomology generators of the considered manifold.

First let us indicate, that the procedure presented in Section 3 can be run also when the primal and dual cell structures are exchanged provided the map D^{-1} is used instead of the map D (since dual complex M^* is just another cellular decomposition of the considered orientable manifold, the Eppstein algorithm can be used for the dual complex). In this case, the Eppstein algorithm provide first homology group generators $\{c_i\}_{i=1}^n$ of the dual complex M^*.

Let us go back to the Poincaré duality presented in Preliminaries. It implies, that D^{-1}: $C_*(M^*)\to C_*(M)$ induces an isomorphism $D_*^{-1}:H_p(M^*)\to H^{n-p}(M)$. In our case $n=2$. Let us use this fact for $p=1$. We get, that $D_*^{-1}:H_1(M^*)\to H^1(M)$ is an isomorphism induced by the map D^{-1}. Therefore $\{D^{-1}(c_1),\dots,D^{-1}(c_n)\}$ are the cohomology generators of $H^1(M)$.

Moreover, since in the Eppstein algorithm all spanning trees T of M and dual spanning trees T^* of $(M^*\setminus\bigcup_{t\in T}D(t))$ are acceptable it is easy to see, that the points (1), (2) and (3) of the algorithm presented in Section 3 can be left without any change. Only the point (4) of the algorithm should be replaced by the following set of instructions:

4. For an edge $e\in E$ let $C(D(e))$ denote the unique dual cycle in $T^*\cup D(e)$.
5. For the set of dual cycles $\{C(D(e))|e\in E\}$ let $h_1,\dots,h_n\in Z_1(M^*)$ be the cycles supported in $C(D(e))$ with coefficients obtained from orienting the dual cycle (see Fig. 5). Then $\{h_1,\dots,h_n\}$ represents a basis of $H_1(M^*)$ (see Section 3).
6. As the direct consequence of the Poincaré duality the set of cocycles $D^{-1}(h_1),\dots,D^{-1}(h_n)\in Z^1(M)$ forms basis of $H^1(M)$.

In Fig. 4 the supports of the cocycles $D^{-1}(h_1),\dots,D^{-1}(h_n)$ are indicated.

5. In search of optimality, Erickson and Whittlesey modification

In this section an analogue of Erickson and Whittlesey modification (Erickson and Whittlesey, 2005) to Eppstein algorithm is discussed. Let us focus here only on minimal generators that intersect on a fixed base point. First let us present a algorithm to obtain minimal length homotopy generators of orientable 2-manifold taken from Erickson and Whittlesey (2005):

1. Let T be a tree of shortest paths from a chosen vertex v (obtained from Dijkstra algorithm).
2. For each edge $e\in M\setminus T$ let $\sigma(e)$ be the length of the unique cycle in $T\cup e$ containing vertex v.
3. Let T^* be a maximal spanning tree of $M^*\setminus\bigcup_{t\in T}D(t)$ where the weight of an edge $D(e)\in M^*\setminus\bigcup_{t\in T}D(t)$ is $\sigma(e)$.
4. Let $E=\{e\in M$ such that e is an edge and $e\notin T$ and $D(e)\notin T^*\}$.
5. For a edge $e\in E$ let $C(e)$ denote the unique cycle in $T\cup e$. The set of cycles $\{C(e)|e\in E\}$ forms shortest basis of the first homotopy group of $\mathcal{M}$ with the base point v (see Erickson and Whittlesey, 2005).

Similarly as in case of Eppstein algorithm to obtain cohomology group generators it suffices to exchange primal and dual complexes (and use map D^{-1} instead of D) in the algorithm presented above (with the weight of dual edges equal to the length of the corresponding edges in dual graph). The obvious details of obtaining cocycles from its supports are skipped here. To complete the section let us present, in Fig. 6, a illustration of this procedure for a standard representation of a torus.

6. Manifolds with boundary

The presented algorithms can be extended, without extensive modifications, to manifolds with boundary. However the motivation behind the algorithm for open manifold case is more general than behind the algorithms for closed manifold case presented in Sections 4 and 5. Even though algorithms in Sections 4 and 5 can be concluded from theory presented in this Section, their motivation has been left in order to show algorithmic beauty of Poincaré duality. For example of manifold with boundary consult Fig. 7.

Let us remind, that topological 2 dimensional manifold with boundary $\mathcal{M}$ is a set such that every point of $x\in\mathcal{M}$ has a neighborhood $x\in U\subset\mathcal{M}$ homeomorphic to $\mathbb{R}^2$ or a half plane $\{(a,b)\in\mathbb{R}^2|a\leqslant 0\}$. As in Section 2 let us consider a combinatorial surface $M=(\mathcal{M},G)$, this time with boundary, where G is a graph as in Section 2. The definition of dual graph and dual cell structure remains unchanged (i.e. for edges and vertices in the boundary of the manifold there are no dual elements in the dual complex).

As in the case of closed manifolds, let us first think how to obtain homology generators in case of 2 dimensional manifolds with boundary. To achieve this aim let us remind the method presented in (Lewiner et al., 2003) where a tree–cotree decomposition is used to obtain optimal discrete Morse function on open or closed 2-manifold (idea of optimal and not optimal discrete Morse function is presented in Fig. 8). The idea of the method[4] is explained in Fig. 9 (note that the presented algorithm is exactly a modification of the Eppstein algorithm). For an introduction to discrete Morse theory consult (Forman, 2001). A discrete Morse function is said to be optimal if the number of critical cells in every dimension is equal to the rank of homology group. For an example of optimal and not optimal discrete Morse function see Fig. 8. For a optimal discrete Morse function, in the considered case when there are no torsions, each critical cell correspond to Betti number and homology generator, see Forman (2001). It is also clear that the projection of the cycle closed in the primal tree by the critical cell is a homology generator (see Lewiner, 2005 for explanation and Fig. 8 for general idea).

By a relative cocycle let us denote a path that ends up in the boundary of the manifold. As in the previous case we will show, that the cohomology generators are either a cocycle or a relative cocycle closed by a critical edge in the dual tree (see Fig. 9).

[4] In this case the values of the Morse function are skipped unlike the case presented in (Lewiner et al., 2003). The reason is, because they are not needed to obtain homology generators.

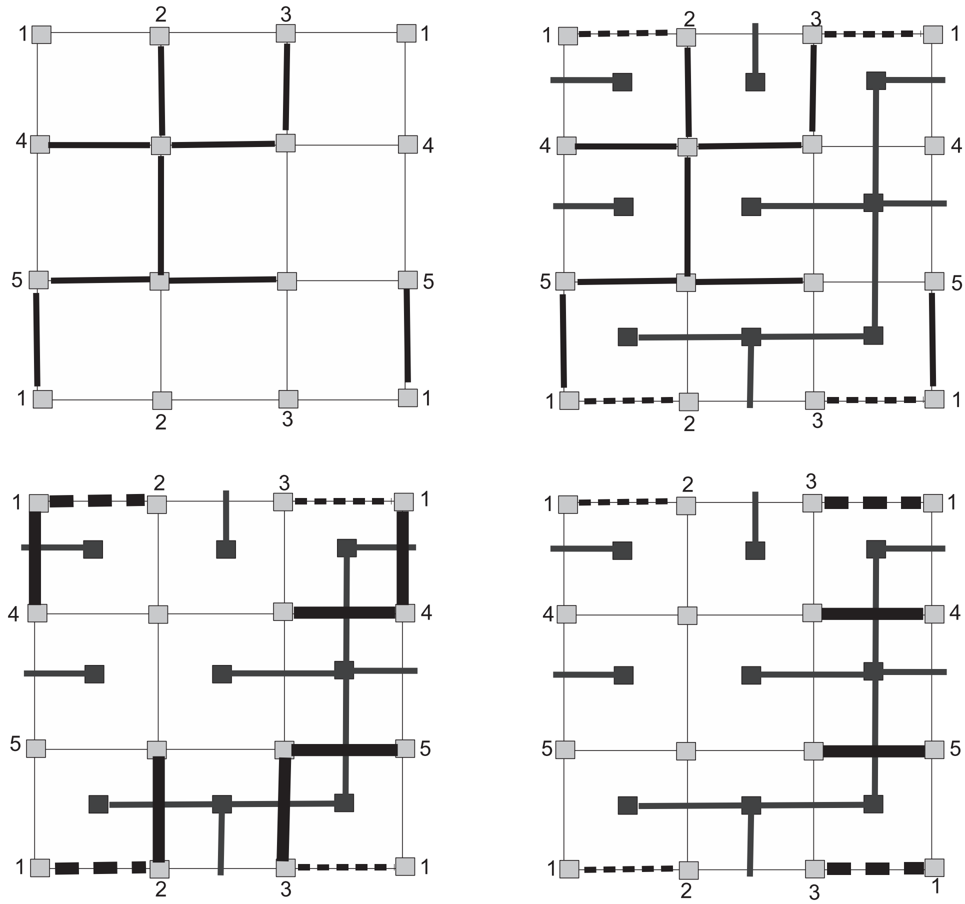

Fig. 4. Eppstein algorithm modification presented on a torus. In the upper left primal spanning tree is depicted with bold edges. On the upper right a dual spanning tree is depicted with blue dual edges. With the dotted lines the two edges [1,2] and [1,3] that are neither in primal nor in dual tree. On the lower left with the double bold edges the support of the dual cycle closed by edge [1,2] is presented. On the lower right with the double bold edges the support of the dual cycle closed by edge [1,3] is presented. Both of the dual cycles with the coefficients obtained as in Fig. 5 are first cohomology group generators of a torus.

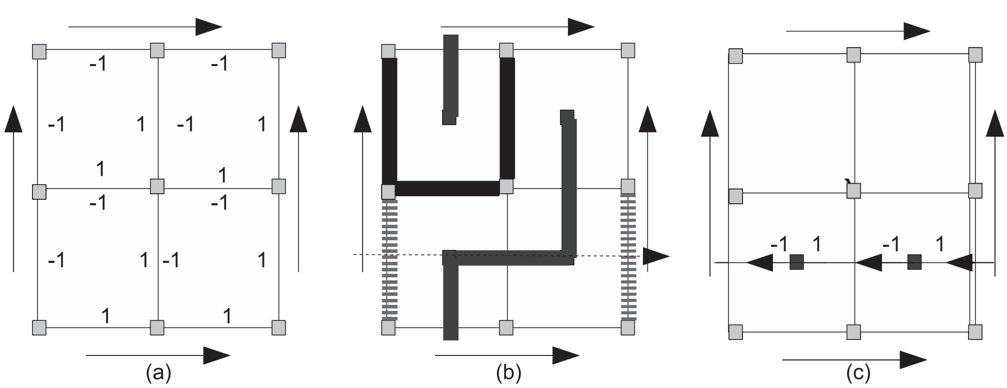

Fig. 5. Idea how to orient a dual cycle. On (a) the integers denote standard cubical incidence index between 2 dimensional cells of torus with the edges in their boundary. On (b) primal and dual spanning tree together with dotted red edge that close horizontal dual cycle. With the horizontal long arrow an orientation of the cycle is chosen. In the (c) the incidence induced in the dual cycle from (a) primal complex is depicted with integers by the dual vertices. Note that the induced orientation of dual edges is exactly the opposite to the orientation of dual cycle fixed on the figure (b). Therefore coefficients of all the edges crossed by the dual cycle are -1.

The key observation that can be derived from Fig. 9 is that the homology generator on the bottom left and the dual cocycle (which, as it will be shown, represent cohomology generator) on the bottom right intersect in a single edge, namely the critical edge that is neither in tree nor in the cotree (depicted as dotted edge at the top right picture). This is a result of the fact, that tree and cotree have empty intersection. This is obviously a general phenomenon which does not depend on the number of critical edges. This phenomenon gives in fact a kind of duality between homology and cohomology which was previously explored by the author in case of so called STT and GSTT algorithms (Dłotko and Specogna, 2010b). The presented reasoning is analogous to the one in Theorem 4.8 in (Dłotko and Specogna, 2010b). Let us remind the Universal Coefficient Theorem for cohomology. For further details consult (Hatcher, 2002).

ARTICLE IN PRESS

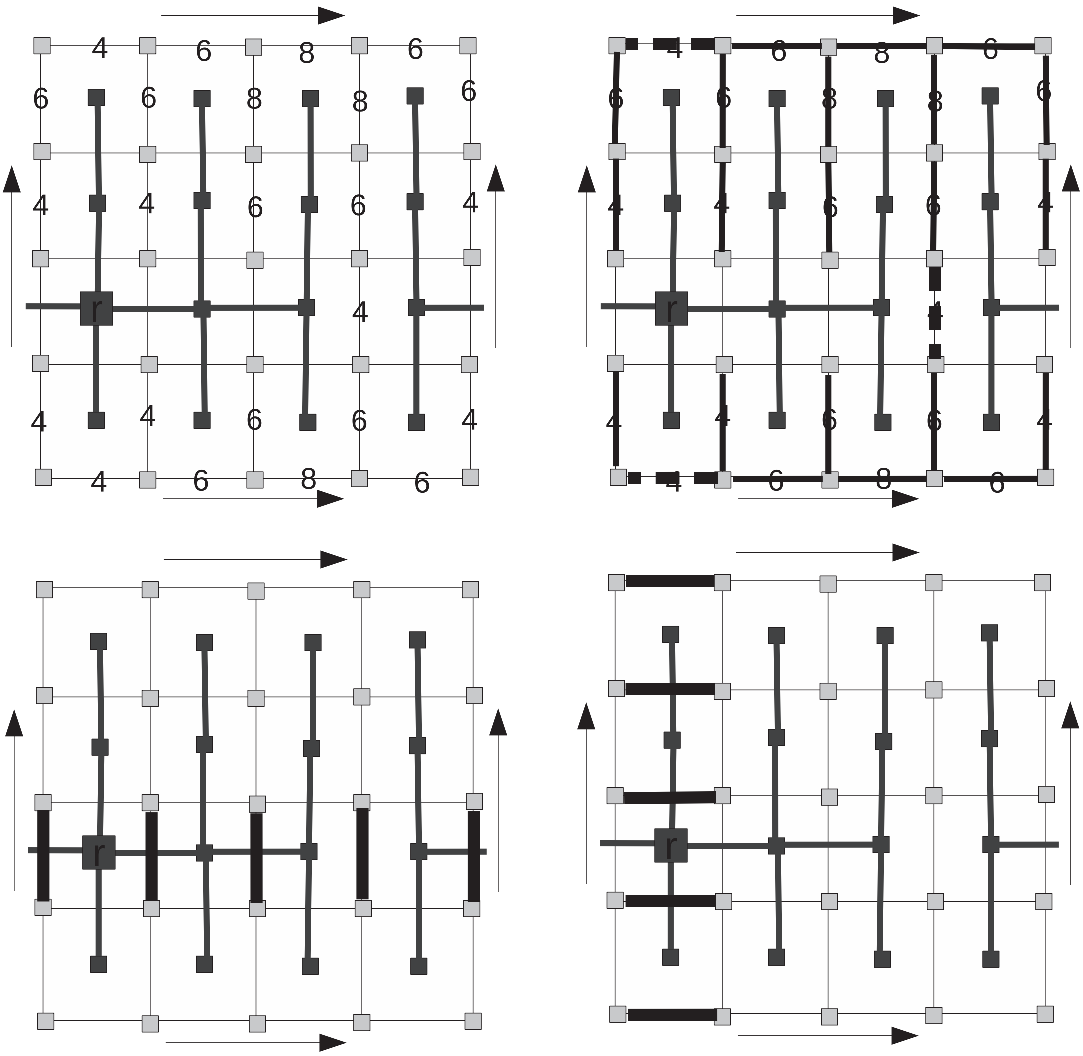

Fig. 6. Erickson and Whittlesey algorithm modification to obtain minimal cohomology basis. In this picture we do not use the convention of naming the boundary vertices in order not to cause a confusion (arrows indicate the way in which faces are glued together). The general assumption of this example is that all the edges have length 1. On the top left the shortest path dual tree rooted at the 2-cell with the biggest black square. The numbers next to the primal edges are their weights. The weight of an edge is a length of a dual cycle closed by it. Moreover the dual cycle need to contain a dual vertex r being a root of the dual tree. On the top right the maximal spanning tree of the $M^{*}\setminus\bigcup_{t\in T}D(t)$. By the dotted edges the edges closing cycles are depicted. On the bottom row the supports of two minimal length cohomology generators are depicted with bold edges.

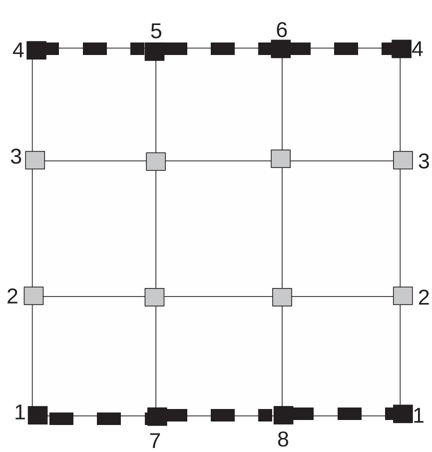

Fig. 7. A cylinder – manifold with boundary. Note only the vertical edges being glued together. The horizontal edges, being the boundary of the manifold, are depicted by dotted edges.

Theorem 1 Hatcher, 2002, Th. 3.2. *If a chain complex C of free abelian groups has homology groups $H_n(C)$, then the cohomology groups $H^n(C,G)$ of the cochain complex $Hom(C,G)$ are determined by split exact sequences:*

$$0 \to Ext(H_{n-1}(C),G) \to H^n(C,G) \to Hom(H_n(C),G) \to 0$$

As it is shown in (Dłotko and Specogna, 2010b) in the considered case $Ext(H_0(C),G) = 0$ provides $H^1(C,G) \approx Hom(H_1(C),G)$. Let is consider $G = \mathbb{Z}$. Since $H_1(C)$ is in the considered case a free group we have a natural correspondence between $H^1(C,\mathbb{Z})$ and $H_1(C)$. Namely for every $c^* \in Z^1(C)$ being one of fixed generators of $H^1(C,\mathbb{Z})$ there exist a corresponding cycle c_* being a generator of dual $H_1(C)$ basis.

Let us denote a cocycle or a relative cocycle by c^* and the corresponding homology generator by c_*. It is clear, that $|\langle c^*,c_*\rangle| = 1$, where $\langle c^*,c_*\rangle$ denotes the evaluation of c^* on the cycle c_* (since the supports of c^* and c_* intersect only in the

Please cite this article in press as: Dłotko, P. A fast algorithm to compute cohomology group generators of orientable 2-manifolds. Pattern Recognition Lett. (2011), doi:10.1016/j.patrec.2011.10.005

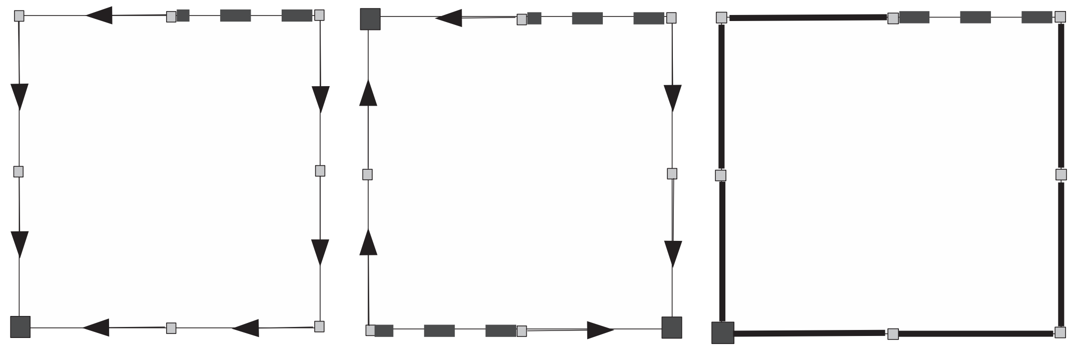

Fig. 8. On the left, an example of optimal discrete Morse function on a circle. Red bold dotted edges denote critical points, arrows between vertices and edges denote pairing of cells. In the middle an example of not optimal discrete Morse function on a circle. On the right primal tree corresponding to the gradient flow of the optimal discrete Morse function is illustrated. A cycle closed by unique critical (dashed) edge is clearly a support of first homology group generator of the cycle.

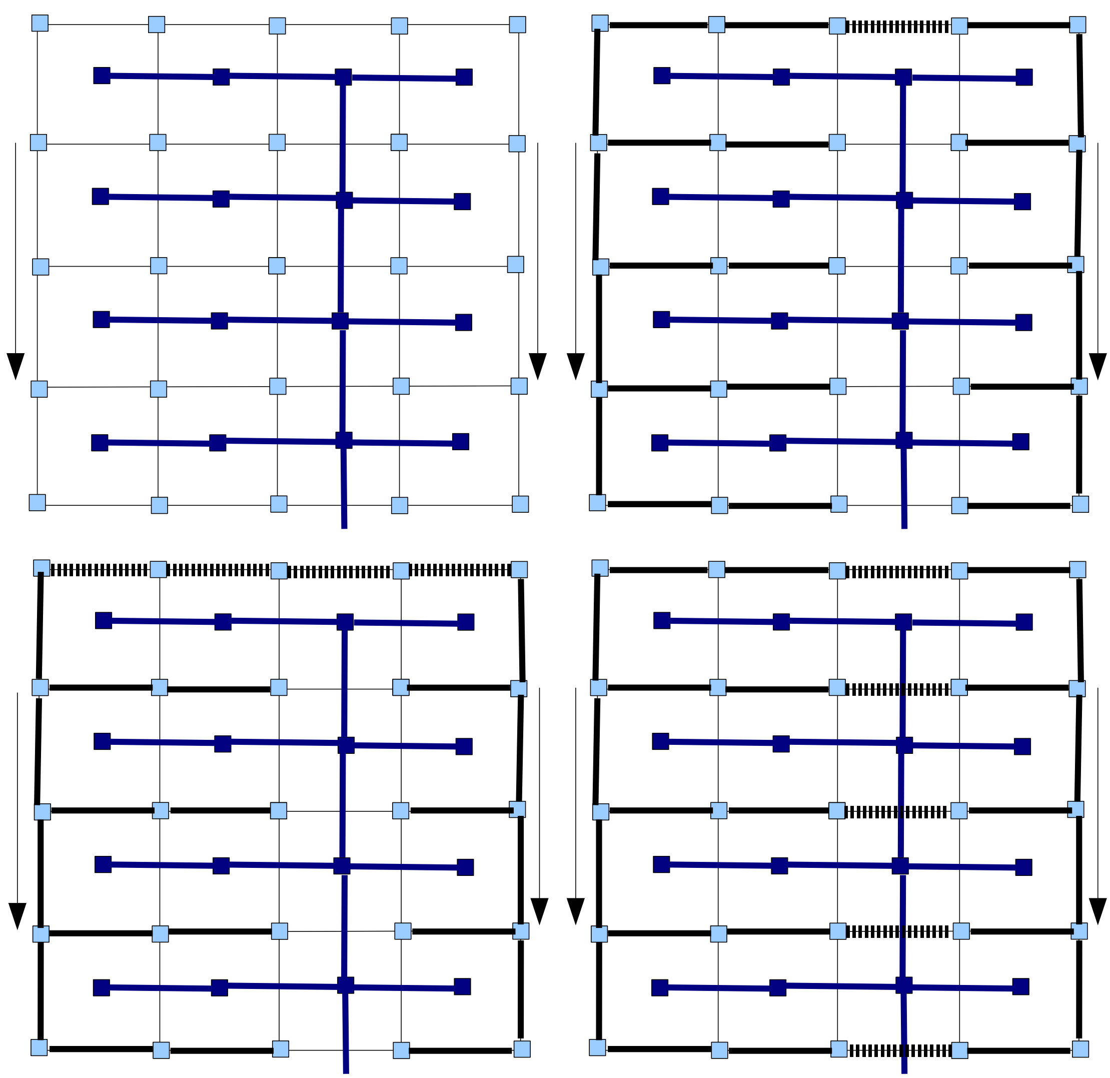

Fig. 9. An illustration of Lewiner–Lopes–Tavares algorithm (Lewiner et al., 2003) based on a cellular decomposition of cylinder. Note only the vertical edges of the boundary being glued together as it is indicated by the arrows. On the upper left the dual tree (cotree) with a single boundary edge added as in (Lewiner et al., 2003) in order to obtain optimal discrete Morse function for manifold with boundary. On the upper right the primal tree is depicted. The unique edge not in tree nor in cotree is marked with dotted line. On the bottom left the projection of the homology generator to the initial cell structure is depicted with dotted line. Note however, that a dotted edge, together with the cotree close a relative cycle with respect to the boundary of the manifold. This relative cycle, marked with dotted edges, is a cohomology generator of this manifold.

critical edge). See the idea of dual homology and cohomology generators in Fig. 2. The presented evaluation is in fact the map which induce isomorphism in Theorem 1 in case when $Ext(H_0(M),G) = 0$.

Let us first show, that c^* obtained from the algorithm is cohomologically nontrivial. Suppose by contrary, that there exist $d^* \in C^0(M)$ such that $\delta d^* = c^*$. It implies, that $1 = \langle c^*, c_* \rangle = \langle \delta d^*, c_* \rangle = \langle d^*, \partial c_* \rangle = \langle d^*, 0 \rangle = 0$, a contradiction. Therefore c^* is nontrivial

Please cite this article in press as: Dłotko, P. A fast algorithm to compute cohomology group generators of orientable 2-manifolds. Pattern Recognition Lett. (2011), doi:10.1016/j.patrec.2011.10.005

in $H^1(M)$. It is clear that for other homology generator obtained by the algorithm $d_* \neq c_*$ one have $\langle c^*, d_* \rangle = 0$ since the intersection of supports of c^* and d_* is empty. From Theorem 1 it follows, that the class of c_* is mapped through the isomorphism to the class of c^* (i.e. c^* is dual to c_*). Since in the considered case $H_1(M)$ is torsion-free we have a straightforward correspondence between $H_1(M)$ and $H^1(M)$ generators. Therefore once the procedure to obtain cohomology basis explained in Section 4 and illustrated for the considered case in Fig. 9 is applied for all critical edges we get a corresponding cohomology basis.

Let us therefore formalize the algorithm, inspired by Lewiner et al. (2003), to obtain cohomology generators of two dimensional manifold with boundary:

1. Let T^* be a spanning tree of M^*.
2. For single $b \in M$ being a boundary edge of M such that $D(b) \notin T^*$ do $T^* = T^* \cup \{D(b)\}$.
3. Let T be a spanning tree of $M \backslash D^{-1}(T^*)$.
4. Let $E = \{e \in M$ such that e is an edge and $e \notin T$ and $D(e) \notin T^*\}$.
5. Let $\{C(e) | e \in E\}$ be the set of oriented (as in Fig. 5) cocycles or relative cocycles in dual tree. Those cocycles are $H^1(M)$ generators.

The proof of correctness of the presented algorithm is straightforward from Theorem 1 and the discussion after it. As a result of the algorithm a set of cocycles is obtained. Each cocycle is either a closed path in the dual tree, or its both ends lie on the boundary of manifold. In both cases the cocycle is cohomologically nontrivial. Tree and a dual cotree are distinct (i.e. build from a different set of edges), therefore intersection of the cocyle closed in dual tree by a critical edge $e \in E$ with its dual homology generator being a cycle in primal tree closed by e contains only the edge e, see Fig. 3. Therefore, from the discussion after Theorem 1, $\{C(e)\}_{e \in E}$ forms a basis of the first cohomology group of the considered manifold.

At the end of the section let us discuss the case of zeroth and second cohomology group generators of manifold with boundary. For the zeroth cohomology group the situation is analogous to the closed manifold case. However due to the presence of boundary of manifold, the second cohomology group vanishes. Therefore in the case of open manifolds zeroth and second cohomology group generator are also known a priori.

7. Non-manifold case and non-orientable manifolds

It is a straightforward consequence of Sections 4–6, that the presented algorithms can be applied in every case, when the algorithm in (Lewiner et al., 2003) return a optimal discrete Morse function. As it is indicated in (Lewiner et al., 2003) a optimal discrete Morse function is obtained also in the following cases:

1. Dangling edge – 2 manifolds joined with an edge (or sequence of edges) – see Fig. 10(a).
2. Singular vertex – 2 manifolds sharing common vertex – see Fig. 10(b).

In (Lewiner et al., 2003) it is shown, that Lewiner–Lopes–Tavares algorithm provide optimal discrete Morse function in those cases. In fact this case can be extended to any set of dangling edges and singular vertices which, after collapsing every manifold to a point and extending singular vertex to an edge, form acyclic graph. In this case the suitable algorithm should be run for each "manifold" component of the set separately. The 1-cocycles obtained in this way are first cohomology group basis of the considered set. The zeroth cohomology group generator in this case is again the map sending every vertex of the considered set to 1. Second cohomology group generators should be considered separately for each "manifold" component of the set.

The algorithms presented in this paper are applicable for orientable two dimensional manifolds. One may wonder – why not to make them also for non orientable manifolds? The reason was, that in case of non orientable closed manifolds a generator of the torsion is also obtained when integer homology are considered (what can be easily seen in a canonical scheme of Klein bottle). It is not clear for the author how differentiate this generator from any other generator to be able to say which is the torsion one. However if one intend to use $\mathbb{Z}_2$ coefficients, then the algorithm presented in Sections 4–6 can be used for any manifold orientable or not. The algorithm works

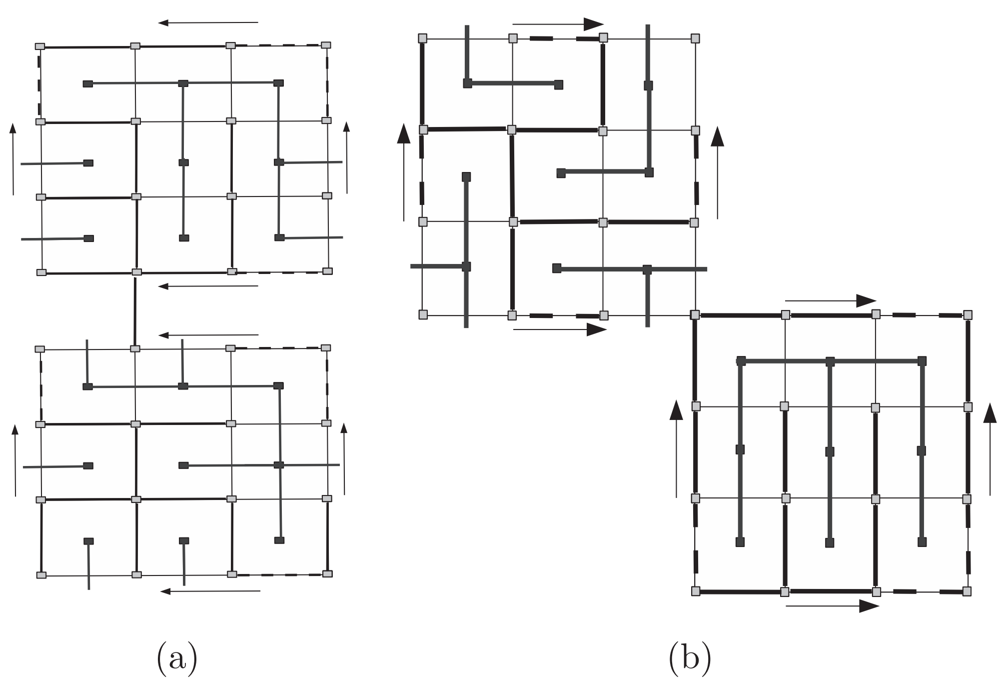

Fig. 10. (a) Example of dangling edge scenario – two toruses joined by an edge. Clearly it is still possible to obtain cohomology generators with the algorithms presented in this paper. (b) Example of singular vertex scenario for two toruses sharing a common vertex. Clearly it is still possible to obtain cohomology generators with the algorithms presented in this paper. We want to remark, that in this case the dual tree is a forest, not a connected tree.

also for a number of non orientable open manifolds provided lack of torsion in homology (like in case of Möbius band). Detailed classification of the manifolds for which the algorithm works is a matter of further studies.

8. Complexity

In this section the complexity aspects of the presented algorithms are discussed. To let those algorithms run it suffices to have for instance a pointer-based representation of a primal complex. Each cell is a data structure that stores pointers to its boundary and coboundary cells. Moreover it has an extra integer field to store additional information which indicate if a simplex is a part of primal or dual tree or non of them. The extra integer is used to store primal and dual tree directly in the data structure.

General assumption in this paper is that all the considered combinatorial surfaces are finite structures. For simplicity let us assume, that the degree of every vertex both in primal and dual graph are bounded by some constant for all considered inputs (if no, this maximal degree should be taken into account in the analysis).

In the case of algorithm presented in Section 4 the spanning trees and the edges in the set E can be found in $O(card(M))$ time by simple BFS strategy. Then the set of cycles can be oriented in a time $O(card(E)card(M))$, since every cycle may have a length proportional to $card(M)$. Consequently the whole algorithm has $O(card(M) + card(E)card(M))$ complexity.

In the case of algorithm presented in Section 5 the shortest paths tree can be found by Dijkstra algorithm in $O(card(M)log(card(M)))$ time. With the same time the dual maximal spanning tree can be found by using for example Kruskal's algorithm. As in the previous case the set of dual cycles can be found in $O(card(E)card(M))$ time. Consequently the algorithm works in $O(card(M) log(card(M)) + card(E)card(M))$ time.

Testing if the manifold admits a boundary can be made in linear time by looking at the coboundary elements of each edge. Therefore an algorithm in Section 6 work in linear time. Finding the "manifold" parts of the considered set, the test described in Section 7 can be made in linear time with a BFS-like algorithm. Therefore the cohomology generators can be obtained also in linear time.

9. Conclusions and further work

In this paper in Section 4 a fast algorithm to compute cohomology group generators of a orientable 2-manifolds without boundary is provided. In order to obtain the set of shortest generators intersecting in a fixed dual vertex in Section 5 a modification of the algorithm working in $O(card(M)log(card(M)) + card(E)card(M))$ may be used. Later in the Section 6 the algorithm is generalized to the case of orientable 2-manifolds with boundary. In fact in Section 7 this condition is relaxed to a set which can be decomposed to orientable manifolds such that two of them are separated or joined with an edge or share a common vertex. In Section 8 short discussion of complexity of the algorithm is presented.

Time complexity and general simplicity of the algorithm can make it very useful for the computer graphic community. We would like to point out that, with some additional work,[5] it is also possible to obtain the globally minimal set of cohomology generators over $\mathbb{Z}_2$ analogously as it is described in (Erickson and Whittlesey, 2005).

The technique presented in this paper seems do not have an analogue for higher dimensional manifolds. The reason is, that the 1-skeleton of primal and dual complex gives a complete description of 2-manifold, but are not enough for higher dimensional manifolds.

Acknowledgements

I want to thank Marian Mrozek, Ruben Specogna and Roman Srzednicki for all the suggestions which help me to complete this paper. I would also like to thank anonymous referees for all the valuable comments and suggestions.

References

Bredon, G.E., 1993. Topology and Geometry. Springer-Verlag.
CAPD, Computer Assisted Proofs in Dynamic, 2002–2010. Software Available From <http://www.capd.ii.uj.edu.pl/>.
CHomP, Computational Homology Project, 2002–2010. Software Available From <http://www.chomp.rutgers.edu/>.
Colin de Verdiere, E., Lazarus, F., 2003. Optimal pants decompositions and shortest homotopic cycles on an orientable surface. Lect. Notes Comput. Sci. 51 (4), 478–490.
Damiand, G., Peltier, S., Fuchs, L., 2006. Computing homology for surfaces with generalized maps: Application to 3D images. Lect. Notes Comput. Sci. (4292), 235–244.
Damiand, G., Peltier, S., Fuchs, L., 2008. Computing homology generators for volumes using minimal generalized maps. Lect. Notes Comput. Sci. (4958), 63–74.
Day, S., Kalies, W., Wanner, T., 2009. Verified homology computations for nodal domains. SIAM J. Multiscale Model. Simul. 7 (4), 1695–1726.
Desbrun, M., Kanso, E., Tong, Y., 2008. Discrete Differential Forms for Computational Modeling. Oberwolfach Seminars, Discrete Differential Geometry.
Dłotko, P., Specogna, R., 2010a. Efficient cohomology computation for electromagnetic modeling. CMES: Comput. Model. Eng. Sci. 48 (4), 247–278.
Dłotko, P., Specogna, R., 2010b. Critical analysis of the spanning tree techniques. SIAM J. Numer. Anal. (48), 1601–1624.
Edelsbrunner, H., Harer, J., 2010. Computational Topology, An Introduction. American Mathematical Society, ISBN 0-8218-4925-5.
Eppstein, D., 2004. Dynamic generators of topologically embedded graphs. In: SODA '03 Proc. of 14th Annual ACM–SIAM Symposium on Discrete Algorithms, pp. 599–608.
Erickson, Jeff, Whittlesey, Kim, 2005. Greedy optimal homotopy and homology generators. In: Proc. 16th Annual ACM–SIAM Symposium Discrete Algorithms, pp. 1038–1046.
Forman, R., 2001. A user's guide to discrete Morse theory. In: Proc. of 2001 Internat. Conf. on Formal Power Series and Algebraic Combinatorics. A Special Volume of Advances in Applied Mathematics.
González-Diáz, R., Real, P., 2003. Computation of cohomology operations on finite simplicial complexes. Homol. Homotopy Appl. (5), 83–93.
González-Diáz, R., Mabel, A., Ham, I., Kropatsch, W.G., 2009. Irregular graph pyramids and representative cocycles of cohomology generators. In: Proc. of 7th IAPR-TC-15 Internat. Workshop on Graph-Based Representations in Pattern Recognition. Lecture Notes in Computer Science.
Greenberg, M.J., 1967. Lectures on Algebraic Topology. Mathematics Lecture Note Series. W.A. Benjamin.
Gu, X., Yau, S., 2002. Computing conformal structures of surfaces. Commun. Inform. Systems (2), 121–146.
Gu, X., Wang, Y., Yau, S., 1996. Multiresolution computation of conformal structures of surfaces. J. Systemics Cybernet. Inform. (1), 2004.
Guo, X., Li, X., Bao, Y., Gu, X., Qin, H., 2006. Meshless thin-shell simulation based on global conformal parameterization. IEEE Trans. Visualiz. Comput. Graph. 12 (3), 375–385.
Hatcher, A., 2002. Algebraic Topology. Cambridge University Press (Available Online).
Kaczynski, T., Mrozek, M., Ŝlusarek, M., 1998. Homology computation by reduction of chain complexes. Comput. Math. 41, 59–70.
Lewiner, T., 2005. Geometric Discrete Morse Complexes. Ph.D. Thesis (Available Online).
Lewiner, Th., Lopes, H., Tavares, G., 2003. Optimal discrete Morse functions for 2-manifolds. Comput. Geom. Theory Appl. 26 (3), 221–233.
Mischaikow, K., Mrozek, M., 2002. In: Fiedler, B., Iooss, G., Kopell, N. (Eds.), The Conley Index Theory in Handbook of Dynamical Systems III: Towards Applications. Elsevier Science B.V., North Holland, Singapore, pp. 393–460.
Mrozek, Marian, Batko, Bogdan, 2009. Coreduction homology algorithm. Discrete Comput. Geom. 41, 96–118.
Mrozek, M., Pilarczyk, P., Żelazna, N., 2008. Homology algorithm based on acyclic subspace. Comput. Math. 55, 2395–2412.
Peltier, S., Ion, A., Kropatsch, W.G., Damiand, G., Haxhimusa, Y., 2009. Directly computing the generators of image homology using graph pyramids. Image Vision Comput. 27 (7), 846–853.

[5] This additional work is computing the set of shortest generators for every base point (root of primal tree) and, by solving linear systems of equation, finding set of globally minimal length generators. We want to indicate that this additional work is big comparable to work needed to run the algorithm presented in this paper.