2014 16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing

Efficient Computation of Simplicial Homology Through Acyclic Matching

Ulderico Fugacci
Dept. of Computer Science, Bioengineering,
Robotics and Systems Engineering
University of Genova
Genova, Italy
Email: ulderico.fugacci@dibris.unige.it

Federico Iuricich
Dept. of Computer Science
and UMIACS
University of Maryland
College Park (MD), USA
Email: iuricich@umiacs.umd.edu

Leila De Floriani
Dept. of Computer Science, Bioengineering,
Robotics and Systems Engineering
University of Genova
Genova, Italy
Email: leila.defloriani@unige.it

***Abstract*—We consider the problem of efficiently computing homology with $\mathbb{Z}$ coefficients as well as homology generators for simplicial complexes of arbitrary dimension. We analyze, compare and discuss the equivalence of different methods based on combining reductions, coreductions and discrete Morse theory. We show that the combination of these methods produces theoretically sound approaches which are mutually equivalent. One of these methods has been implemented for simplicial complexes by using a compact data structure for representing the complex and a compact encoding of the discrete Morse gradient. We present experimental results and discuss further developments.**

I. Introduction

Recently, the problem of computing the topological features of a shape has drawn much attention because of its applications in several disciplines, including shape understanding, shape retrieval, and finite element analysis. Topological features provide global quantitative and qualitative information about a shape, such as the number of its connected components, and the number of holes and tunnels. These features are especially important in the analysis of high-dimensional data, where pure geometric tools are usually not feasible.

The existing literature about homology computation focuses mainly on the computation of simplicial homology with $\mathbb{Z}_2$ coefficients. This topological invariant is simpler to compute than the classical integer homology, but it fails in providing the torsion part of the homological information, present, for instance, in a Klein bottle. We are interested in investigating tools and strategies to efficiently compute simplicial homology for arbitrary simplicial complexes of any dimension, which provide the whole homological information, including Betti numbers, torsion coefficients and homology generators. For simplicial complexes in high dimensions, torsion coefficients and homology generators are definitely relevant to deeply understand the shape of any high-dimensional data. A significant example concerns an application to chemistry and biology [1]: the conformation space of a molecule has been analyzed from a dataset in $\mathbb{R}^{72}$ and it has been found through homological techniques that it has the geometry of a 2-dimensional surface composed of a sphere and a Klein bottle.

One way of facing the complexity of the homology computation problem is through simplification of the simplicial complex. In the literature, several techniques have been developed to reduce the size of the complex while preserving its homology. Such techniques are based on reductions or coreductions of pairs of simplices, or, equivalently, they construct a chain complex with a reduced number of cells and with the same homology as the original one, the *discrete Morse complex*. Our first contribution in this work is a comparison and an analysis of different methods combining discrete Morse theory and reductions and coreductions for homology computation. Our analysis has led us to consider new simplification methods interleaving reductions and coreductions and to show the theoretical equivalence of these approaches.

One of the main challenges in homology computation is dealing with complexes of large size and dimension. Currently, available implementations work on cubical complexes of large size, or on simplicial complexes of limited size. In this latter case the simplicial complex needs to be encoded through an incidence graph implementing the Hasse diagram which tends to be verbose and does not scale well with size and dimension. Here, we present a first implementation of one of the simplification methods by using a compact data structure encoding only the top simplices of the complex and a compact encoding of the discrete Morse gradient, and we present an experimental analysis and comparison.

The remainder of the paper is organized as follows. Section II contains some background notions, while Section III contains an overview of existing work and software tools. Section IV discusses the reduction-based and coreduction-based algorithms and shows the equivalence of these methods and of combinations of the two operators. Section V describes our implementation of the coreduction-based algorithm based on a compact encoding of the simplicial complex, while Section VI presents some experimental results. Finally, Section VII draws some concluding remarks and discusses future works.

978-1-4799-8448-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SYNASC.2014.84

II. Background Notions

In this section, we introduce some background notions which are at the basis of our work, namely simplicial complexes and simplicial homology, discrete Morse theory, reductions and coreductions.

A. Simplicial complexes and simplicial homology

A *simplex of dimension* k (briefly a k*-simplex*) is the convex hull of $k+1$ affinely independent points. Given a k-simplex σ, any simplex τ, which is the convex hull of a non-empty subset of the points generating σ, is called a *face* of σ. Conversely, σ is called a *coface* of τ. A *simplicial complex* Σ is a finite set of simplices such that each face of a simplex in Σ belongs to Σ, and each non-empty intersection of any two simplices in Σ is a face of both.

Given a simplicial complex Σ, it is possible to define the *chain complex* associated with Σ, denoted as $C_*(\Sigma) := (C_k(\Sigma), \partial_k)_{k\in\mathbb{Z}}$, where $C_k(\Sigma)$ is the free Abelian group generated by the k-simplices of Σ, and $\partial_k : C_k(\Sigma) \to C_{k-1}(\Sigma)$ is a homomorphism, called *boundary map*, which encodes the boundary relations between the k-simplices and the $(k-1)$-simplices of Σ such that $\partial_k\partial_{k+1} = 0$. We denote as $Z_k(\Sigma) := \ker \partial_k$ the group of the k-cycles of Σ and as $B_k(\Sigma) := \operatorname{Im} \partial_{k+1}$ the group of the k-boundaries of Σ. The k^{th} *homology group of* Σ *with coefficients in* $\mathbb{Z}$ is defined as

$$H_k(\Sigma) := H_k(C_*(\Sigma)) = \frac{Z_k(\Sigma)}{B_k(\Sigma)}$$

Given an arbitrary Abelian group A, we can define the k^{th} homology group with coefficients in A of Σ as $H_k(\Sigma; A) := H_k(C_*(\Sigma) \otimes_{\mathbb{Z}} A)$, where $\otimes_{\mathbb{Z}}$ is the tensor product of Abelian groups. If we consider $A = \mathbb{Z}_2$, $C_*(\Sigma) \otimes_{\mathbb{Z}} \mathbb{Z}_2 := (C_k(\Sigma) \otimes_{\mathbb{Z}} \mathbb{Z}_2, \partial_k \otimes_{\mathbb{Z}} \mathbb{Z}_2)_{k\in\mathbb{Z}}$ is the chain complex whose groups $C_k(\Sigma) \otimes_{\mathbb{Z}} \mathbb{Z}_2$ are just the $\mathbb{Z}_2$-vector spaces generated by the k-simplices of Σ and the homomorphisms $\partial_k \otimes_{\mathbb{Z}} \mathbb{Z}_2$ are the boundary maps ∂_k of Σ considered modulo 2.

By the theorem of structure for finitely generated Abelian groups [2], homology groups of a simplicial complex Σ can be expressed as

$$H_k(\Sigma) \cong \mathbb{Z}^{\beta_k}\langle c_1, \cdots, c_{\beta_k}\rangle \oplus \mathbb{Z}_{\lambda_1}\langle c'_1\rangle \oplus \cdots \oplus \mathbb{Z}_{\lambda_{p_k}}\langle c'_{p_k}\rangle$$

with $\lambda_{i+1} \mid \lambda_i$ and with λ_i non-invertible. We call β_k the k^{th} *Betti number* of Σ, $\bigoplus_{i=1}^{p_k} \mathbb{Z}_{\lambda_i}$ the *torsion part* of $H_k(\Sigma)$ and $c_1, \cdots, c_{\beta_k}, c'_1, \cdots, c'_{p_k}$ the *generators* of $H_k(\Sigma)$. For each k, the k^{th} Betti number β_k measures the number of independent, non-bounding, k-cycles in Σ. In dimension 0, β_0 counts the number of connected components of the complex, in dimension 1, its tunnels and its holes, in dimension 2, the shells surrounding voids or cavities, and so on.

It can be proven (see [3], Chapter X) that, for simplicial complexes embeddable in $\mathbb{R}^3$, each homology group is free, and, thus, its torsion part is trivial. For this reason, the $\mathbb{Z}$-homology groups of a simplicial complex Σ embeddable in $\mathbb{R}^3$ can be retrieved by just computing the homology of Σ with $\mathbb{Z}_2$ coefficients.

B. Discrete Morse theory

One important tool for simplifying the homology computation of a simplicial complex is *discrete Morse theory* [4], [5]. Given two simplices σ, τ of Σ, we write $\tau \prec \sigma$ if τ is a face of σ and $dim\sigma = dim\tau + 1$. A simplicial complex Σ is equipped with a function $f : \Sigma \to \mathbb{R}$, called a *discrete Morse function* if, for every simplex σ in Σ,

- $c^+(\sigma) := \#\{\rho \succ \sigma \mid f(\rho) \leq f(\sigma)\} \leq 1$,
- $c^-(\sigma) := \#\{\tau \prec \sigma \mid f(\tau) \geq f(\sigma)\} \leq 1$.

It is easy to show (see [4], Lemma 2.5) that, for a discrete Morse function, $c^+(\sigma)$ and $c^-(\sigma)$ cannot be simultaneously equal to 1. A simplex σ in Σ is *critical* if $c^+(\sigma) = c^-(\sigma) = 0$.

A *discrete vector field* V on Σ is a collection of pairs of simplices $(\tau, \sigma) \in \Sigma \times \Sigma$ such that $\tau \prec \sigma$ and each simplex of Σ appears in at most one pair in V.

A discrete Morse function $f : \Sigma \to \mathbb{R}$ induces a discrete vector field $V = \{(\tau, \sigma) \in \Sigma \times \Sigma \mid \tau \prec \sigma \text{ and } f(\tau) \geq f(\sigma)\}$ called the *gradient vector field* of f, on Σ. A pair $(\tau, \sigma) \in V$ can be viewed as an arrow from τ to σ.

Given a discrete vector field V, a *gradient path* from $\tilde{\tau} \in \Sigma$ to $\tau \in \Sigma$ is a sequence of simplices of Σ

$$\tilde{\tau} = \alpha_0, \beta_0, \alpha_1, \beta_1, \alpha_2, \cdots, \alpha_{r-1}, \beta_{r-1}, \alpha_r = \tau$$

where, for $i = 0, \cdots, r-1$, $\alpha_i \neq \alpha_{i+1}$, $(\alpha_i, \beta_i) \in V$ and $\alpha_{i+1} \prec \beta_i$. We say that such a path is a *non-trivial closed path* if $r > 0$ and $\alpha_0 = \alpha_r$.

Theorem 1 (Thm. 3.5 in [5]): A discrete vector field V is the gradient vector field of a discrete Morse function if and only if there are not non-trivial closed paths.

The *discrete Morse complex* associated with Σ is a chain complex $\mathcal{M}_* := (\mathcal{M}_k, \tilde{\partial}_k)_{k\in\mathbb{Z}}$, where groups $\mathcal{M}_k$ are generated by the critical k-simplices of the function f and the boundary maps $\tilde{\partial}_k$ are obtained by following the gradient vector field of f. For each Abelian group A, we have that $H_k(\Sigma; A) \cong H_k(\mathcal{M}_*; A)$ (see [4], Thm. 8.2).

A gradient vector field can be built without providing a discrete Morse function on a simplicial complex Σ by considering an acyclic matching of Σ.

Let Σ be a simplicial complex. A *matching of* Σ consists of a partition of Σ into three sets $\mathcal{A}$, $\mathcal{K}$, and $\mathcal{Q}$ along with a bijection $w : \mathcal{Q} \to \mathcal{K}$, such that, for each $\tau \in \mathcal{Q}$,

$$< \partial w(\tau), \tau >= \pm 1,$$

where, given two simplices σ and τ, $< \partial\sigma, \tau >$ denotes the coefficient of τ in $\partial\sigma$. We denote this decomposition as $(\mathcal{A}, w : \mathcal{Q} \to \mathcal{K})$.

Given a matching of Σ, we define a relation $\leq$ on $\mathcal{Q}$ by transitive closure as follows. Let us consider two distinct elements $\tau, \tau' \in \mathcal{Q}$;

$$\text{if } < \partial w(\tau), \tau' > \neq 0 \text{ then } \tau' < \tau.$$

A matching of Σ is *acyclic* if $\leq$ is antisymmetric, and, thus, it defines a partial order on $\mathcal{Q}$.

In order to formalize the relationship among these notions and discrete Morse theory, it is sufficient to note that, if we define $V := \{(\tau, w(\tau)) | \tau \in \mathcal{Q}\}$, the condition on the acyclicity of the matching is equivalent to the requirement that V does not contain non-trivial closed paths. Then, by Theorem 1, we are able to conclude that a matching is acyclic if and only if it generates a gradient vector field.

C. Reductions and coreductions

In this subsection, we present two classical homology-preserving operators introduced in [6], namely *reduction* and *coreduction*. Such operators enable reducing the size of a simplicial complex without affecting its homology. A *reduction* corresponds to a deformation retraction of a simplex, which is face of only another simplex onto the complex. The problem is that, in most situations, the available reductions are quickly exhausted. In order to overcome this issue, coreductions have been introduced, where a coredution can be viewed as the dual operation with respect to a reduction. A coreduction is not feasible on a simplicial complex, while it is available in the context of *S-complexes* [6]. Since we will work only with simplicial complexes, for the purpose of this work we consider an S-complex as a simplicial complex in which some simplices may be not present even if their cofaces are in the complex.

Let Σ be an S-complex and let σ be a simplex of Σ. We call the following sets of simplices respectively *(immediate) boundary* and *(immediate) coboundary of* σ with respect to Σ

$$\mathrm{bd}_\Sigma\, \sigma := \{\tau \in \Sigma \,|\, \tau \prec \sigma\},$$

$$\mathrm{cbd}_\Sigma\, \sigma := \{\rho \in \Sigma \,|\, \rho \succ \sigma\}.$$

Let Σ be an S-complex and let (σ, τ) be a pair of elements of Σ such that $< \partial\sigma, \tau >= \pm 1$. A pair (σ, τ) is a *reduction pair* if $\mathrm{cbd}_\Sigma\, \tau = \{\sigma\}$; it is a *coreduction pair* if $\mathrm{bd}_\Sigma\, \sigma = \{\tau\}$.

In a chain complex $C_*(\Sigma)$ associated with a simplicial complex Σ, the removal of a reduction, or of a coreduction, pair (σ, τ) produces a chain complex C'_* in which chain groups are generated by the simplices of Σ different from σ or τ, and the boundary maps are suitable restrictions of the boundary maps of $C_*(\Sigma)$.

It can be proven that the removal of a reduction, or of a coreduction, pair preserves the homology groups [6]. Furthermore, one can retrieve the homology generators of Σ from the homology generators of C'_* by computing their images under a suitable isomorphism map [7], [8].

III. Related work

The classical approach to the computation of the homology of a simplicial complex of finite dimension is based on the *Smith Normal Form (SNF) reduction* [9] applied to the boundary matrices describing the boundary maps ∂_k. Although this method is theoretically valid in any dimension, it has some inherent limitations due to the size of the boundary matrices and to the high complexity of the reduction algorithm, which is super-cubical in the number of the simplices of the complex. Another well-known problem is the appearance of large integers during reduction [10]. In the literature, several optimizations of the SNF algorithm have been developed. *Stochastic methods* [11] are efficient on sparse integer matrices, but they do not provide the homological generators. *Deterministic methods* [12], [13] perform the computations modulo an integer chosen by a determined criterion, but the information about torsion coefficients is lost with this strategy.

Another way to improve computation times is to reduce the size of the input complex without changing its homology, by applying iterative simplifications, thus computing the homology when no more simplifications are possible. Some of these approaches are based on *reductions* and *coreductions* [6]–[8], others simplify the simplicial complex via *acyclic subspaces* [14], [15]. A similar approach for reducing the size of a complex without affecting its homology is based on the notion of *tidy set* [16]. Another class of reduction approaches [17]–[19] is based on *discrete Morse theory* [4], [5], since a discrete Morse complex built from a given cell complex has the same homology as this latter but with fewer cells. Also, *decomposition-based* approaches have been proposed [20], [21], which relate the homology of a simplicial complex to the homologies of its sub-complexes, i.e., the sub-components of the input complex and their intersections. In [22], a *hierarchical representation* for a cell complex based on homology-preserving operators has been defined and implemented to efficiently compute the homology of the complex and extract its homology generators at different adaptive resolutions.

Several software tools for computing homology, mostly for cubical complexes and for persistence homology [23], [24], have been developed and distributed in the public domain. For persistent homology, we can mention *Dionysus* [25], *CAM* [26], and *SimpPers* [27]. These tools work on $\mathbb{Z}_2$ coefficient homology and do not provide homology generators. The software library [28] contains the implementation of simplification algorithms based on reduction and coreduction operations. *Perseus* [29] is an implementation based on discrete Morse theory, and performs both homology and persistent homology computations. Most of these tools compute the Betti numbers, but not the torsion part, and the homology generators; the majority of them have been developed for cubical complexes, while the ones that deal with simplicial complexes use verbose data structures for encoding them like the incidence graph. Recent work on computing persistence homology based on annotations uses also the incidence graph for efficiency [26], [27].

IV. Discrete Morse complexes through reductions and coreductions

A. Using coreduction sequences and reduction sequences

In this subsection, we describe two approaches which combine simplification operators and discrete Morse theory. Each algorithm for building a discrete Morse complex of a simplicial complex Σ consists of two steps. In the first step, a gradient vector field of a discrete Morse function is built by using simplification operators. The second step computes the discrete Morse complex by following the gradient vector field. The two algorithms described here perform either sequences of coreductions or of reductions to obtain an acyclic matching and, thus, a gradient vector field.

The first approach [18], that we call the *coreduction-based algorithm*, is based on the construction of an acyclic matching on a simplicial complex Σ by using coreduction pairs and removals of free simplices, where a *free simplex* is a simplex with an empty boundary. To obtain an acyclic matching $(\mathcal{A}, w : \mathcal{Q} \to \mathcal{K})$ of Σ, the approach in [18] initializes first sets $\mathcal{A}$, $\mathcal{Q}$, $\mathcal{K}$ and map w as null. Let us denote as Σ' the set of non-excised simplices of Σ and initialize it as Σ. While Σ' admits a coreduction pair, the algorithm excises a coreduction pair (σ, τ) from Σ': simplices σ and τ are added to sets $\mathcal{K}$ and $\mathcal{Q}$, respectively, and $w(\tau)$ is defined to be σ. When no more coreduction is feasible, a free simplex is excised from the complex and inserted in the set $\mathcal{A}$ of the critical simplices. The algorithm repeats these steps until no coreduction can be performed and set Σ' is empty.

The other approach [30], that we call the *reduction-based algorithm*, is based on the construction of an acyclic matching on a simplicial complex Σ by using reduction pairs and removals of top simplices, where a *top simplex* is simplex whose coboundary is empty. This method is dual with respect to the *coreduction-based algorithm*. To obtain an acyclic matching $(\mathcal{A}, w : \mathcal{Q} \to \mathcal{K})$ of Σ, the algorithm follows the same pattern as the previous one, but it removes reduction pairs and top simplices instead of coreduction pairs and free simplices. While the set of non-excised simplices Σ' admits a reduction pair, the algorithm excises a reduction pair from Σ' and suitably updates sets $\mathcal{K}$, $\mathcal{Q}$ and function w by adding σ to $\mathcal{K}$, τ to $\mathcal{Q}$ and setting $w(\tau) = \sigma$. When no more reduction is feasible, a top simplex is excised from the complex and inserted in the set $\mathcal{A}$ of the critical simplices. Then, the algorithm repeats this process until the set Σ' of the remaining simplices of Σ is empty.

In both cases, it has been proved that the matching $(\mathcal{A}, w : \mathcal{Q} \to \mathcal{K})$ produced on Σ is actually acyclic.

B. Equivalence of reduction, coreduction and interleaved sequences

In this section, we prove the equivalence between the coreduction-based and the reduction-based methods and we introduce another class of methods which could operate reductions and coreductions in an interleaved way.

Proposition 1: Given a simplicial complex Σ and the acyclic matching on it produced by a reduction-based algorithm, it is always possible to obtain the same matching (and, thus, the same discrete Morse complex) with a coreduction-based algorithm. The reverse is also true.

Proof: We just give a sketch of the proof. For brevity, we show that the acyclic matching produced by a reduction-based algorithm on Σ can be obtained with a coreduction-based algorithm. Given a simplicial complex Σ, consider the ordered sequence of reduction pairs and top simplices removed during the execution of a reduction-based algorithm. It can be proved that, considering the same sequence but applying it in the reverse order, it represents for Σ a sequence of coreduction pairs and free simplex removals. Furthermore, suitably reordering this reverse sequence of operators we can guarantee that the removal of a free simplex can be performed only if no coreduction pair is feasible. In this way, a sequence compatible with a coreduction-based algorithm is obtained. ■

Another method to build a matching on a simplicial complex is to execute removals of reduction and coreduction pairs in an interleaved way. Here, we prove that such class of approaches produces an acyclic matching and that all interleaved methods are equivalent.

Proposition 2: Given a simplicial complex Σ, the matching $(\mathcal{A}, w : \mathcal{Q} \to \mathcal{K})$ produced by any algorithm, which uses only reduction pairs, coreduction pairs, removals of top simplices and of free simplices, is acyclic.

Proof: We just give a sketch of the proof. By Definition of reduction and coreduction, it is immediate to prove that the proposed algorithm produces a matching of Σ. In order to conclude the proof, we have to show that the matching is acyclic i.e., that $\leq$ is a partial order on $\mathcal{Q}$. Consider set $\mathcal{Q}$ as built in any intermediate step of the proposed algorithm and let τ be the latest element inserted in $\mathcal{Q}$. The following properties, that can easily shown, allow us to conclude that the order defined on $\mathcal{Q}$ is partial:

1. τ is a maximal element with respect to the elements already inserted in $\mathcal{Q}$ originating from a coreduction pair;
2. τ is a minimal element with respect to the elements already inserted in $\mathcal{Q}$ originating from a reduction pair.

■

Having proved that any possible interleaved method leads to an acyclic matching, we are now interested in understanding if these different approaches could produce equivalent results or not. First at all, we claim the following useful result.

Remark 1: In an algorithm for constructing an acyclic matching, which uses just reduction pairs, coreduction pairs, removals of top simplices and removals of free simplices, each coreduction pair and free simplex removal cannot make a reduction pair feasible and also each reduction pair and top simplex removal cannot make a coreduction pair feasible.

Proposition 3: Any two algorithms which build a matching by using reduction pairs, coreduction pairs, removals of top simplices, only if no reduction is feasible, and of free simplices, only if no coreduction is feasible, are equivalent. More precisely, given a simplicial complex Σ and the acyclic matching on it produced by one of these algorithms, it is always possible to obtain the same matching (and, thus, the same discrete Morse complex) with the other algorithm and vice versa.

Proof: We prove that the acyclic matching produced by an algorithm by using just reduction pairs, coreduction pairs, removals of top simplices, only if no reduction is feasible, and of free simplices, only if no coreduction is feasible, on a simplicial complex can be also obtained with a coreduction-based algorithm. Let Σ be a simplicial complex and consider the ordered sequence of reduction pairs, coreduction pairs, top simplices and free simplices performed by the interleaved algorithm. By Remark 1, we can suitably order such sequence, moving all the coreduction pairs and the free simplices at the beginning, thus creating a new sequence equivalent to the previous one. We apply to the last part, composed only of reduction pairs and top simplices, of this new sequence the same sorting strategy proposed in Proposition 1 to transform a sequence originating from a reduction-based algorithm to a sequence compatible with a coreduction-based algorithm, and in this way we obtain the thesis. ■

From both an application and a theoretical point of view, it is interesting to find a method to build a gradient vector field which minimizes the number of critical simplices. The previous results show that all the proposed methods are equivalent from a theoretical point of view and the use of different matching operators (such as reduction and coreduction pairs), or the combination of more than one, does not actually affect the number of resulting critical simplices.

V. Coreduction-based algorithm for homology computation on a simplicial complex

A. A compact data structure for a simplicial complex

The data structure we used for encoding the simplicial complex is an adjacency-based data structure, called the *Generalized Indexed data structure with Adjacencies (IA*)* [31]. The IA* encodes all vertices and top simplices in a simplicial d-complex Σ. For each top simplex σ of dimension k, it encodes the vertices of σ and all the top simplices of dimension k $(k-1)$-adjacent to σ. For each vertex v, it encodes all the top 1-simplices incident in v and one top $(k+1)$-simplex for each k-connected component in the set of simplices incident in v. Moreover, the data structure encodes a partial coboundary relation for each $(k-1)$-simplex on the boundary of more than two top k-simplices. It has been shown in [32] that the IA* data structure is much more compact that the Incidence Graph (IG), which is the implementation of the Hasse diagram.

B. Compact encoding of the gradient vector field

The encoding of the gradient vector field is associated only with the top simplices. Each top k-simplex σ encodes a bitvector $grad$ of length $\sum_{i=1}^{k}\binom{k+1}{i+1}(i+1)$ representing all the possible pairings on its boundary. If two paired simplices ν and τ are both on the boundary of σ, the resulting pair will be encoded in the bitvector of σ.

Let j and l (with $j+1=l$) be the dimensions of ν and τ, respectively. Let $start = \sum_{i=l+1}^{k}\binom{k+1}{i+1}(i+1)$, be pos_ν the position of ν on the boundary of τ and pos_τ the position of τ on the boundary of σ. The bit, set for pair ν and τ, is in the position $start + (pos_\tau(l+1)) + pos_\nu$.

For efficiency, we store an additional bitvector, denoted as $paired(\sigma)$, for each top simplex σ. $paired(\sigma)$ encodes, for each simplex τ in the boundary of σ, whether τ is paired (or not). Using such bitvector, we are not forced to look outside σ for testing if τ can be paired or not.

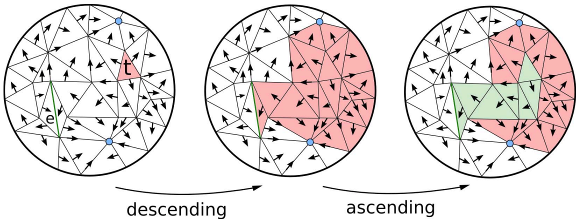

Fig. 1: Descending and ascending traversals used during the computation of the V-path connecting e and t.

C. Homology computation

1) Computing the gradient vector field: The coreduction-based algorithm is an implementation of the one introduced in [18] and described in IV-A on a simplicial complex encoded with an IA* data structure. Starting from the simplices of lower dimension, all possible coreductions are applied to build a gradient vector field V incrementally. In the first step, a vertex v is declared as critical in V. Working on the top simplices on the coboundary of v, all the 1-simplices eligible for a coreduction are paired in V. The algorithm then moves to the top simplices on the coboundary of the new paired simplices. Once all the possible 1-simplices have been paired, the working dimension is increased by one. Generally speaking, if k is the working dimension, all k-simplices are paired in V using valid coreductions. Once all available coreductions have been performed, if there are unpaired $(k-1)$-simplices in Σ, one of them is declared as critical in V and, if there are new coreductions available, they are performed. Once all the k-simplices have been paired, the working dimension is increased by one. When all the simplices have been paired or declared as critical, the algorithm ends.

2) Extracting boundary maps and homology generators: Here, we describe how we retrieve homology with $\mathbb{Z}$ coefficients and homology generators for a simplicial complex Σ of arbitrary dimension. Once a gradient vector field V on Σ has been computed, we can easily retrieve its homology groups.

In order to compute the homology of Σ with $\mathbb{Z}$ coefficients, we have to compute the boundary map $\tilde{\partial}_k : \mathcal{M}_k \to \mathcal{M}_{k-1}$ of the discrete Morse complex $\mathcal{M}_*$ associated with an acyclic matching built on Σ, i.e., we have to count the multiplicity of each gradient path between two critical simplices. The gradient vector field is visited with a descending traversal. Starting from a critical k-simplex σ, all the $(k-1)$-simplices in the boundary of σ are selected and, among them, only the $(k-1)$-simplices paired with a k-simplex different from σ are considered. From such k-simplices, a breadth-first traversal continues until all the V-paths starting from σ have been visited. Then, for each critical $(k-1)$-simplex τ, a dual ascending traversal is performed visiting only those simplices marked during the descending one. This results in extracting

Dataset	d	n	n_0	n_{top}
Buddha	2	3.2M	0.54M	1.08M
Elephant	2	9.2M	1.5M	3.07M
Fertility	2	1.4M	0.24M	0.48M
Skull	3	0.75M	37K	0.15M
Neghip	3	2.1M	93K	0.48M
7Klein	7	0.1M	0.11K	0.6K
9Sphere	9	0.22M	2.0K	911

TABLE I: Simplicial complexes used. Column d indicates the dimension of the complex, n indicates the total number of simplices, n_0 the number of vertices and n_{top} the number of top simplices.

the V-paths between two critical simplices. As an example, we show in Figure 1 the two steps performed for computing the V-path between the critical simplices t and e. Starting from t, the descending traversal is performed marking as visited all the triangles reached by a V-path starting at t. Then, the (trivial) ascending traversal is performed starting at e and navigating only the triangles previously visited until t is reached.

The homological equivalence between $\mathcal{M}_*$ and Σ implies that, by using the SNF reduction algorithm, we are able to obtain the simplicial homology of Σ. The proposed construction of the discrete Morse complex allows us to retrieve homology with respect to any coefficient. Specifically, we can compute homology with integer or $\mathbb{Z}_2$ coefficients. Homology generators are obtained by suitably navigating the gradient vector field, using the descending traversal, and storing the collections of visited simplices.

VI. Experimental results

We have implemented and tested a first prototype of our coreduction algorithm, based on the IA* data structure, and we have compared its performances with *Perseus* [29]. We have focused our comparisons on *Perseus* since its implementation is also based on reductions and coreductions. All our experiments have been performed on a desktop computer with a 3.2Ghz processor with 16GB of memory.

Table I describes the datasets used for the experiments. We have chosen datasets having most of their top simplices of maximal dimension. Such simplicial complexes well describe the trade-off between the two approaches, since they maximize the number of boundary and coboundary relations for each simplex. Since the *Perseus* implementation encodes all the incidence relations for each simplex explicitly, we expect such implementation to be much less compact compared to the IA* implementation. On the other hand, the computation based on the IA* data structure is expected to be more time consuming due to the retrieval of the incidence relations among simplices, since they are not encoded in the data structure. We used simplicial complexes with various maximal dimension d to evaluate the scalability of the two approaches.

For each simplicial complex, we have evaluated the performances for computing homology, and the discrete Morse gradient and for extracting the homology generators with both implementations. In *Perseus*, we have used the mixed approach using reductions and coreductions since it is faster when using an Incidence Graph (IG) to encode the complex, while we used our algorithm based on coreductions only for the IA* implementation. To evaluate the performances, we compute, for each complex, the maximum amount of memory required by the two algorithms and the timings for computing homology. For complex *Elephant*, we ran out of memory in our system when using *Perseus*.

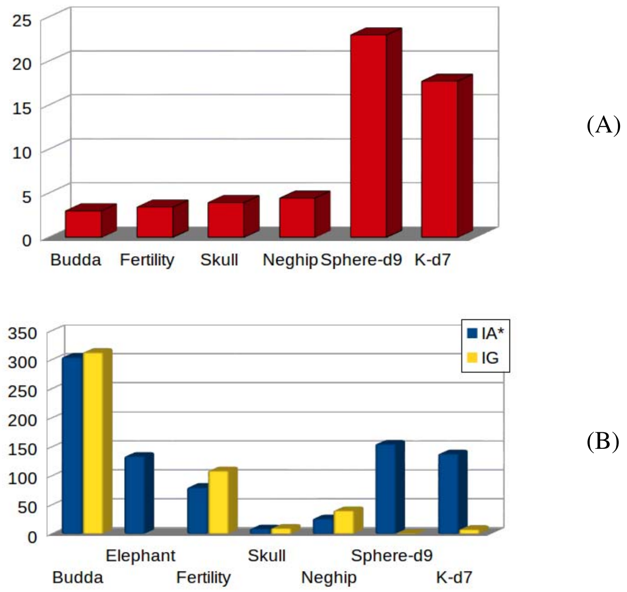

Fig. 2: (A) Comparison of the storage costs of the two implementations (IG/IA*) and (B) timings (in seconds) for the homology computation algorithms based on the IA* and IG data structures.

In Figure 2A, we show the ratios between the storage cost required by the two implementations (IG/IA*) considering the maximum amount of memory used at runtime. The IG-based implementation is about 3 times more expensive than the IA* implementation for 2-complexes. When working with 3-complexes the ratio becomes 4 and it grows to 17x and 24x for the 7- and 9-complexes, respectively. The storage cost for the IA* depends on the number of top simplices in the simplicial complex only. The IG representation, instead, limits the maximum size of the complex that can be handled in input when working in higher dimensions.

Considering the storage costs, the IA* data structure would be a natural choice when working with memory constraints. However, from the point of view of timing, it is not always the best choice. In Figure 2B, we compare the results obtained when computing homology with the two approaches. We notice that, for the 2- and 3-complexes, the IA* implementation is comparable to the IG one. Extraction times for boundary relations not explicitly encoded in the IA* structure are mainly affected by the dimension of the top simplices and, thus, they are not so relevant in lower dimensions. However, in higher dimensions, the lack of encoded relations in the IA* structure directly affects the computation times, thus resulting in a relevant overhead when compared to the IG implementation.

VII. Concluding Remarks

We have considered the problem of efficiently computing homology with $\mathbb{Z}$ coefficients and homology generators for simplicial complexes of arbitrary dimension. We have analyzed and proved the equivalence of different methods which allow reducing the complexity of the computation through reductions and coreductions, and we have developed a first implementation based on coreductions, on a space-efficient representation of the simplicial complex and on a compact encoding of the acyclic matching induced by the coreduction pairs.

This is a first step in the direction of a scalable and efficient computation of integer homology on simplicial complexes. In our future work, we plan to develop an efficient encoding for a simplicial complex in arbitrary dimensions based on the generalization of the PR-star octree [33], a compact topological data structure based on a spatial index for tetrahedral meshes, which stores only the vertices and the top simplices of the complex. This latter would not only reduce the storage cost further, but also allow an efficient localized homology computation, overcoming the limitation in this respect of the IA* data structure. We believe that the generalization of the PR-star octree could also speed up other existing approaches to persistent homology computation [26], [27].

Acknowledgements

This work has been partially supported by the US National Science Foundation under grant number IIS-1116747 and by the University of Genova through PRA 2013. The authors wish to thank Davide Bolognini, Emanuela De Negri and Maria Evelina Rossi for their helpful comments and suggestions.

References

[1] S. Martin, A. Thompson, E. A. Coutsias, and J.-P. Watson, “Topology of cyclo-octane energy landscape,” *Journal of Chemical Physics*, vol. 132, no. 23, p. 234115, 2010.

[2] M. Artin, *Algebra*. Prentice Hall, 1991.

[3] P. Alexandroff and H. Hopf, *Topologie i*. Berlin, 1935, vol. 1035.

[4] R. Forman, “Morse theory for cell complexes,” *Advances in mathematics*, vol. 134, no. 1, pp. 90–145, 1998.

[5] ——, “A user’s guide to discrete Morse theory,” *Sém. Lothar. Combin*, vol. 48, p. 35, 2002.

[6] M. Mrozek and B. Batko, “Coreduction homology algorithm,” *Discrete & Computational Geometry*, vol. 41, no. 1, pp. 96–118, 2009.

[7] M. Mrozek and T. Wanner, “Coreduction homology algorithm for inclusions and persistent homology,” *Comput. Math. Appl.*, vol. 60, no. 10, pp. 2812–2833, Nov. 2010.

[8] P. Dłotko, T. Kaczynski, M. Mrozek, and T. Wanner, “Coreduction homology algorithm for regular cw-complexes,” *Discrete & Computational Geometry*, vol. 46, no. 2, pp. 361–388, 2011.

[9] J. Munkres, *Elements of Algebraic Topology*, ser. Advanced book classics. Perseus Books, 1984.

[10] J. L. Hafner and K. S. McCurley, “Asymptotically fast triangularization of matrices over rings,” *SIAM Journal on Computing*, vol. 20, no. 6, pp. 1068–1083, 1991.

[11] M. Giesbrecht, “Probabilistic computation of the Smith Normal Form of a sparse integer matrix,” in *Algorithmic Number Theory*. Springer Verlag, 1996, pp. 173–186.

[12] R. Kannan and A. Bachem, “Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix,” *SIAM Journal on Computing*, vol. 8, no. 4, pp. 499–507, 1979.

[13] A. Storjohann, “Near optimal algorithms for computing Smith normal forms of integer matrices,” in *Proceedings of the 1996 international symposium on Symbolic and algebraic computation*, ser. ISSAC ’96. New York, NY, USA: ACM, 1996, pp. 267–274.

[14] M. Mrozek, P. Pilarczyk, and N. Żelazna, “Homology algorithm based on acyclic subspace,” *Comput. Math. Appl.*, vol. 55, no. 11, pp. 2395–2412, Jun. 2008.

[15] P. Brendel, P. Dłotko, M. Mrozek, and N. Żelazna, “Homology computations via acyclic subspace,” in *Proceedings of the 4th international conference on Computational Topology in Image Context*, ser. CTIC’12. Berlin, Heidelberg: Springer Verlag, 2012, pp. 117–127.

[16] A. Zomorodian, “The tidy set: a minimal simplicial set for computing homology of clique complexes,” in *Proceedings of the 2010 Annual Symposium on Computational Geometry*. ACM, 2010, pp. 257–266.

[17] V. Robins, P. J. Wood, and A. P. Sheppard, “Theory and algorithms for constructing discrete Morse complexes from grayscale digital images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1646–1658, 2011.

[18] S. Harker, K. Mischaikow, M. Mrozek, and V. Nanda, “Discrete Morse theoretic algorithms for computing homology of complexes and maps,” *Foundations of Computational Mathematics*, vol. 14, no. 1, pp. 151–184, 2014.

[19] S. Harker, K. Mischaikow, M. Mrozek, V. Nanda, H. Wagner, M. Juda, and P. Dłotko, “The efficiency of a homology algorithm based on discrete Morse theory and coreductions,” in *Proceedings 3rd International Workshop on Computational Topology in Image Context (CTIC 2010). Image A*, vol. 1, 2010, pp. 41–47.

[20] D. Boltcheva, D. Canino, S. M. Aceituno, J.-C. Léon, L. D. Floriani, and F. Hétroy, “An iterative algorithm for homology computation on simplicial shapes,” *Computer-Aided Design*, vol. 43, no. 11, pp. 1457 – 1467, 2011.

[21] R. H. Lewis and A. J. Zomorodian, “Multicore homology,” 2012.

[22] L. Čomić, L. De Floriani, F. Iuricich, and U. Fugacci, “Topological modifications and hierarchical representation of cell complexes in arbitrary dimensions,” *Computer Vision and Image Understanding*, vol. 121, pp. 2–12, 2014.

[23] H. Edelsbrunner and J. Harer, “Persistent homology-a survey,” *Contemporary mathematics*, vol. 453, pp. 257–282, 2008.

[24] A. Zomorodian and G. Carlsson, “Computing persistent homology,” *Discrete & Computational Geometry*, vol. 33, no. 2, pp. 249–274, 2005.

[25] D. Morozov, “Dionysus library for computing persistent homology,” 2012. [Online]. Available: http://www.mrzv.org/software/dionysus/

[26] J.-D. Boissonnat, T. K. Dey, and C. Maria, “The compressed annotation matrix: An efficient data structure for computing persistent cohomology,” in *Algorithms–ESA 2013*. Springer Verlag, 2013, pp. 695–706.

[27] T. K. Dey, F. Fan, and Y. Wang, “Computing topological persistence for simplicial maps,” *arXiv preprint arXiv:1208.5018*, 2012.

[28] B. Piotr, D. Pawel, J. Grzegorz, J. Mateusz, K. Andrzej, M. Marian, W. Hubert, W. Przemyslaw, and Z. Natalia, “Capd:: Redhom–homology software based on reduction algorithms,” 2014. [Online]. Available: http://capd.sourceforge.net

[29] V. Nanda, “Perseus: the persistent homology software.” Date, 2012. [Online]. Available: http://www.sas.upenn.edu/~vnanda/perseus/index.html

[30] B. Benedetti and F. H. Lutz, “Random discrete Morse theory and a new library of triangulations,” *Experimental Mathematics*, vol. 23, no. 1, pp. 66–94, 2014.

[31] D. Canino, L. De Floriani, and K. Weiss, “IA*: An adjacency-based representation for non-manifold simplicial shapes in arbitrary dimensions,” *Computers & Graphics*, vol. 35, no. 3, pp. 747–753, 2011.

[32] D. Canino and L. De Floriani, “Representing Simplicial Complexes with Mangroves,” *Proceedings of the 22nd Iinternational Meshing Roundtable*, pp. 465–483, 2013.

[33] K. Weiss, L. D. Floriani, R. Fellegara, and M. Velloso, “The PR-star octree: a spatio-topological data structure for tetrahedral meshes,” in *GIS*, I. F. Cruz, D. Agrawal, C. S. Jensen, E. Ofek, and E. Tanin, Eds. ACM, 2011, pp. 92–101.