Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Computers & Graphics 30 (2006) 62–69
COMPUTERS & GRAPHICS
www.elsevier.com/locate/cag

Computation of homology groups and generators

Samuel Peltier[a,*], Sylvie Alayrangues[b], Laurent Fuchs[a], Jacques-Olivier Lachaud[b]

[a]*SIC (FRE 2731 CNRS), Université de Poitiers, Boulevard Marie et Pierre Curie, 86962 Futuroscope Chasseneuil Cedex, France*
[b]*LaBRI, Université Bordeaux 1, 351 cours de la Libération, 33405 Talence Cedex, France*

Abstract

Topological invariants are extremely useful in many applications related to digital imaging and geometric modeling, and homology is a classical one, which has not yet been fully explored in image applications. We present an algorithm that computes the whole homology of an object of arbitrary dimension: Betti numbers, torsion coefficients and generators. Effective implementation of this algorithm has been realized in order to perform experimentations. Results on classical shapes in algebraic topology and on discrete objects are presented and discussed.

Keywords: Topological invariant; Shape invariant; Homology groups; Image analysis

1. Introduction

In digital image analysis, shape invariants are useful for classification, indexation, or, more recently, shape description [1]. They can be used in object simplification and object thinning. In solid modeling, shape invariants ensure the consistency of constructive operations. Computing topological invariants of objects has thus a significant impact in these domains. The fundamental group is an invariant that carries most of the topological information about an object. It has been studied by many authors [2–5] in the image analysis field. But the comparison of such groups is highly related to undecidable problems [4]. Many authors have proposed algorithms to compute the Euler characteristic (some of them summarized in [6]), but it is a simpler and less expressive topological invariant. Other approaches compute the Betti numbers [7] of embedded objects.

We focus here on homology groups, which are known to be computable in finite dimensions, and which have a good topological characterization power at least in low dimensions. For instance Euler characteristic and Betti numbers are straightforwardly deduced from homology groups. These groups are also the abelianized of homotopy groups [8].

Several approaches can be found into the literature; Kaczynski et al. [9] proposed to compute homology groups with a sequence of reductions. The idea is to derive a new object with less cells while preserving homology at each step of the transformation. During the computations, to ensure invertible coefficients, Kaczynski et al. choose them in a field. González-Díaz and Real [10] recently proposed an algorithm to compute cohomology information on digital objects that are subsets of the 3D body-centered cubic grid. They first construct a simplicial complex with identical topology. The cohomology is obtained by chain contraction in two passes, a thinning and an incremental algebraic thinning. All coefficients are in $\mathbb{Z}/2\mathbb{Z}$ (also a field).

The preceding approaches are interesting when dealing with embedded objects in 2 dimension or 3 dimension. Homology over a field is then enough to

*Corresponding author.
E-mail addresses: peltier@sic.univ-poitiers.fr (S. Peltier), alayrang@labri.fr (S. Alayrangues), fuchs@sic.univ-poitiers.fr (L. Fuchs), lachaud@labri.fr (J.-O. Lachaud).

0097-8493/$ - see front matter
doi:10.1016/j.cag.2005.10.011

characterize shapes, since objects have no torsion. On the contrary, we choose a more generic approach, valid for arbitrary dimensions and shapes. We address the problem of computing the whole homology over the coefficient domain $\mathbb{Z}$ (a ring, not a field). We not only compute the homology groups but also their generators, to delineate topological holes on shapes. For instance, the generators of the homology group of dimension 1 are connectivity lines of the shape: cutting along such lines does not divide the shape into two parts.

Even if elements of groups theory are usually known by most of the readers, their use in topology, which is the purpose of algebraic topology, is generally less known. Besides, if homology is a main classical tool in algebraic topology, computation of homology group generators is not so widely discussed in the literature. Even in recent specialized works [11,8], these computations are not explicitly considered.

So one contribution of this work is to report some recent works and bring these results to the imagery community. We also combine these works to classical results in homology theory to compute the homology groups (Betti numbers and torsion coefficients) and their generators, and finally we effectively implement these algorithms with numerous optimizations.

In the first part of the paper we recall classical definitions in homology theory. We choose here simplicial homology since it is widely used in geometric modeling and is straightforwardly applicable to digital objects. After that, we present our approach for computing homology groups: modified Smith Normal Form (mSNF) to compute generators and integer computations performed with a modulo.

Note that this approach is not only valid for simplicial structures but also for all combinatorial objects which realization is a CW-complex [12], e.g. cubical complexes or discrete objects. Lastly, we show some experiments, both on simplicial and discrete objects, and list some perspectives to this work.

2. Simplicial homology

Shapes are classically modeled with a cellular subdivision. Several combinatorial structures may represent such a subdivision. We choose here semi-simplicial sets, which can represent indifferently manifold or non-manifold objects. This structure is a subclass of simplicial sets, a structure studied in algebraic topology [13,14].

2.1. Semi-simplicial set

Definition (*May [13]*). A *semi-simplicial set* $S = (K, (d_i^q))$ is a graded family of sets $K = (K^q)_{q \in \mathbb{N}}$ together with maps $d_i^q : K^q \to K^{q-1}$ for $i = 0, \ldots, q$, which satisfy the following identity: $\forall \sigma \in K^q, d_j^{q-1}(d_i^q(\sigma)) = d_{i-1}^{q-1}(d_j^q(\sigma))$ *if* $j < i$.

The elements of K^q are called *q-simplices*. The d_i^q are called *boundary operators* (the subscripts q will generally be omitted later for clarity). Simplices are glued together consistently with these operators (see Figs. 1(a) and (b) for two examples).

Semi-simplicial sets are clearly adapted to the constructive operations of solid modeling [15]. They are also well suited to digital imagery [16]. To determine a semi-simplicial set that represents a given digital object, the first step is to construct a simplicial analog. One method is proposed in [10]. The second step is to number the vertexes of the simplicial analog; the boundary maps follow directly [17].

We can now introduce homology groups in an intuitive way. All objects are assumed to be finite. Note that the homology theory is applicable on most combinatorial structures. Some of our experiments have actually been conducted on cubical structures.

2.2. Chain, boundary homomorphism, chain complex

In a first step, we define group structures on semi-simplicial sets. A *p-chain* in K^p is a linear combination of p-simplices with integer coefficients. More formally, any p-chain is written uniquely as a finite sum $\sum_{i=1}^{n_p} \alpha_i^p \sigma_i^p$, where n_p is the cardinal of $K^p = \{\sigma_1^p, \ldots, \sigma_{n_p}^p\}$, and for all

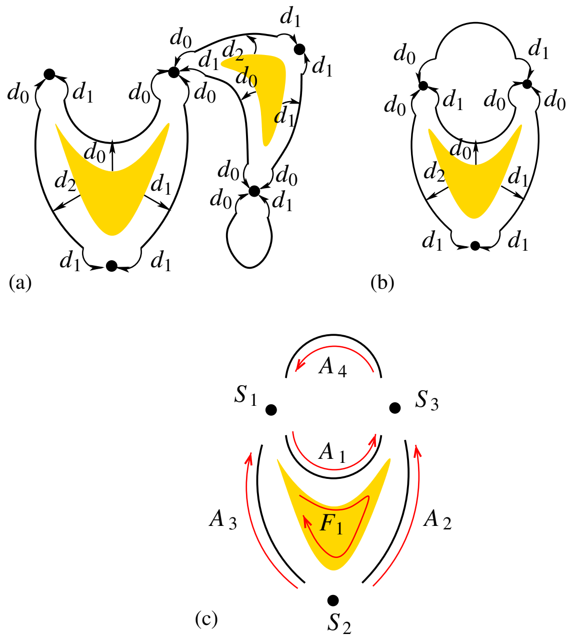


Fig. 1. In (a) and (b), examples of semi-simplicial sets. In (c), positive orientation of the simplices of (b).

i, α_i^p is an integer. The addition over p-chains is defined simply by adding coefficients simplex by simplex. The resulting groups are denoted by C_p. For all p, the set of p-simplicies K^p forms a basis for C_p (see [12, p. 28]).

A p-chain is a purely formal construction. The coefficients α_i have generally not a geometric interpretation, except for the coefficients 1 and -1. In this case $1 \cdot \sigma$ means that we consider the simplex σ with its orientation and $-1 \cdot \sigma$ means that we consider the simplex σ with its opposite orientation. This is consistent with the fact that simplices can be equipped with two orientations, one considered positive and the other negative. Fig. 1(c) displays the positive orientations of the simplices of Fig. 1(b). A formal definition of simplex orientation is available in classical algebraic topology books [12,8].

In a second step, we relate chain groups of successive dimensions with homomorphisms called boundary operators.

Definition. For all $p>0$, the *boundary* of a p-simplex σ^p, denoted by $\partial_p(\sigma^p)$, is the $(p-1)$-chain $\sum_{i=0}^{p}(-1)^i d_i(\sigma)$, a 0-simplex have an empty boundary. The boundary is extended as an homomorphism from C_p to C_{p-1}, meaning for any p-chain $c = \sum_{i=1}^{n_p} \alpha_i^p \sigma_i^p$, its boundary $\partial_p(c)$ is equal to $\sum_{i=1}^{n_p} \alpha_i^p \partial_p(\sigma_i^p)$.

Usually, when no confusion may arise, we simply write $\partial(c)$ for the boundary of a p-chain c. For example, in Fig. 1c, we have $\partial(F_1) = A_1 - A_2 + A_3$ and we can verify that $\partial(\partial(F_1)) = \partial(A_1 - A_2 + A_3) = 0$.

We have just constructed a sequence of chain groups C_p together with homomorphisms ∂_p, such that $C_n \xrightarrow{\partial_n} C_{n-1} \xrightarrow{\partial_{n-1}} \cdots \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0$. One can check that $\partial_{p-1}(\partial_p(c)) = 0$ for all p-chains. This sequence is called a *free chain complex*.

2.3. Cycle, boundary, hole

Homology groups of a combinatorial object are derived from specific subgroups of chains groups.

All p-chains whose boundary is empty are called *p-cycles*. For example, in Fig. 1(c), the 1-chains $A_1 - A_2 + A_3$ and $A_1 + A_4$ are 1-cycles. The set of p-cycles is a subgroup of C_p, denoted by Z_p.

Some p-chains are the boundary of a $(p+1)$-chain. They are called *p-boundaries*. For example, in Fig. 1(c), the 1-chain $A_1 - A_2 + A_3$ is the boundary of the 2-chain F. The set of p-boundaries form a subgroup of C_p, denoted by B_p. Since $\forall c \in C_p, \partial_{p-1}(\partial_p(c)) = 0$, we have $B_p \subset Z_p \subset C_p$.

A *p-dimensional hole* is a p-cycle which is not a p-boundary. For example, in Fig. 1(c), the 1-cycle $z_1 = A_1 + A_4$ is not a boundary. We define an equivalence relation in the group of p-cycles as follows: two p-cycles s and t are in the same equivalence class iff there exists a chain c with $s = t + \partial_{p+1}(c)$. They are then said to be *homologous*. In particular, when $s = \partial_{p+1}(c)$ then s is homologous to 0. The set of cycles is then partitioned by the homology relation, according to the hole they surround. Two cycles in the same equivalence class surround the same hole. The set of p-boundaries is the 0-equivalence class. For example, the cycle $z_2 = A_2 - A_3 + A_4$ is in the z_1 equivalence class because $z_1 = z_2 + \partial_2(F_1)$.

2.4. Homology groups

In any dimension p, the *homology group* H_p is defined as the group of the equivalent classes for the homology relation. It is exactly the quotient group of the p-cycles by the p-boundaries, $H_p = Z_p/B_p$. Homology groups are known to be topological invariants, meaning homeomorphic shapes have isomorphic homology groups.

For all p, there exists a finite number of elements of H_p from which we can deduce all H_p elements, thus H_p is called finitely generated. So, the group H_p verifies the fundamental theorem of finitely generated abelian groups [12], and H_p is isomorphic to a direct sum:

$$\underbrace{\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}}_{\beta_p} \oplus \mathbb{Z}/t_1^p\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/t_n^p\mathbb{Z}.$$

We denote by β_p the number of occurrences of $\mathbb{Z}$ in this direct sum: it is the number of elements of H_p with infinite order and is called the pth *Betti number*. The numbers $t_1^p, \ldots, t_n^p$ are called the *torsion coefficients* of H_p. To each group $\mathbb{Z}$ of H_p is associated a set of p-dimensional homologous cycles: they surround the same p-dimensional topological hole and are not the boundary of any $p+1$-chain. It is the same for each group $\mathbb{Z}/t_i^p\mathbb{Z}$: the associated homologous cycles are not the boundary of any $p+1$-chain. However, when taken t_i^p times, they become the boundary of some $p+1$-chain. An example is the 1-cycle A_2 in Fig. 2, which becomes a boundary only when taken two times: $2A_2 = \partial(F_1 + F_2)$.

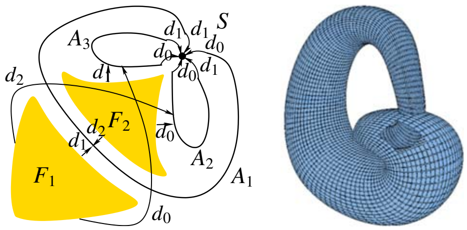
3. Computation of homology groups and generators

In this section, we first recall Agoston's algorithm principles, which allow to compute homology generators by reduction of incidence matrices to their mSNF. We then discuss about implementation problems linked to SNF computation. We thus propose some optimizations of mSNF which benefit both from a theoretical result obtained by Storjohann and from some improvements proposed by Dumas. The algorithm computes Betti numbers, torsion coefficients, and a set of "moduli generators".

3.1. Modified Smith normal of boundary homomorphism

Information on homology groups may be deduced from matrix representations of boundary homomorphisms. A natural basis of the p-chains group of a chain complex is the one made of all its p-simplices, i.e. K^p. In the following, the matrix E_{p+1}, called $(p+1)$th *incidence matrix*, represents the homomorphism ∂_{p+1} relatively to the canonical bases K^p (rows) and K^{p+1} (columns). Each column in E_{p+1} expresses the boundary of one $p+1$-simplex, decomposed on the base of p-simplices.

There exists bases in which any homomorphism has a very specific matrix form, the so-called *Smith Normal Form* (SNF). Cairns [18] proved that it is possible to simultaneously choose bases for each group of p-chains such that the matrix N_p representing each boundary operator relatively to these bases is in a normal form quite similar to SNF, we call it the mSNF. A SNF (resp. mSNF) is a matrix full of $0's$ except for an upper left (resp. right) square submatrix which is diagonal with increasing coefficients: $diag(\lambda_1, \ldots, \lambda_l)$ such that each λ_i is greater than 1 and divides each λ_j, for $j > i$. Moreover, Cairns explains how to deduce a set of generators of the homology group H_p directly from the matrix N_{p+1}. N_{p+1} is shown in Table 1.



(a) (b)

Fig. 2. Klein bottle. (a) Semi-simplicial and (b) geometric representation.

The set $\{b_1^p, \ldots, b_{\beta_p}^p\}$ generates the free part of H^p: they are p-cycles when read as a column in N_p and they have no boundary antecedent when read as a row in N_{p+1}. The set $\{a_1^p, \ldots, a_{\gamma_p}^p\}$ generates the torsion part of H^p: they are p-cycles when read as a column in N_p and they must be multiplied by the λ_i^p to have a boundary antecedent when read as a row in N_{p+1}.

Agoston [19] proposed an algorithm to compute all matrices N_p and keep tracks of changes of bases. The idea is to compute successively all matrices N_p from 0 to the maximal index of the desired homology groups. Each homomorphism is successively expressed in four pairs of bases as in Table 2.

At the end of the whole computation, each matrix N^p represents the homomorphisms ∂_p relatively to bases Γ^p and Γ^{p+1} such that $\Gamma^0 = V_0^{-1}K^0$, $\Gamma^1 = U_1 V_1^{-1} K^1, \ldots,$ $\Gamma^{n-1} = U_{n-1} V_{n-1}^{-1} K^{n-1}$, $\Gamma^n = U_n K^n$ where the matrices U_i and V_i are transfer matrices.

3.2. Implementation issues

Algorithms for computing the SNF or the presented modified version are well known (e.g. see [19,12]). But

Table 1
Modified SNF of boundary homomorphism ∂_{p+1}

	$(p+1)$-Cycles						Weak boundaries antecedents						
	a_1^{p+1}	$\cdots$	$a_{\gamma_{p+1}}^{p+1}$	b_1^{p+1}	$\cdots$	$b_{\beta_{p+1}}^{p+1}$	c_1^{p+1}	$\cdots$	$c_{\rho_p}^{p+1}$	$c_{\rho_p+1}^{p+1}$	$\cdots$	$c_{\gamma_p}^{p+1}$	
a_1^p							$\lambda_{\gamma_p}^p$		0				Weak boundaries
$\vdots$		0			0			$\ddots$			0		
$a_{\rho_p}^p$							0		$\lambda_{\gamma_p-\rho_p+1}^p$				
$a_{\rho_p+1}^p$										1		0	
$\vdots$		0			0			0			$\ddots$		
$a_{\gamma_p}^p$										0		1	
b_1^p													Cycles but not weak boundaries
$\vdots$		0			0			0			0		
$b_{\beta_p}^p$													
c_1^p													
$\vdots$		0			0			0			0		
$c_{\gamma_{p-1}}^p$													

Table 2
Expression of the homomorphisms

Step	Bases []\[] and matrix of ∂_p	Bases []\[] and matrix of ∂_{p+1}
0. Input from iteration p	$[(V_{p-1}U_{p-1}^{-1})^{-1}K^{p-1}]\backslash[U_pK^p]$ (mSNF) $N_p = V_{p-1}U_{p-1}^{-1}E_pU_p$	
1. Incidence matrix of ∂_{p+1}		$[K^p]\backslash[K^{p+1}]$ (incidence) E_{p+1}
2. Left-multiply E_{p+1} by U_p^{-1}		$[(U_p^{-1})^{-1}K^p]\backslash[K^{p+1}]$ $E'_{p+1} = U_p^{-1}E_{p+1}$
3. Compute the mSNF N_{p+1} of ∂_{p+1} from E'_{p+1}		$[(V_pU_p^{-1})^{-1}K^p]\backslash[U_{p+1}K^{p+1}]$ (mSNF) $N_{p+1} = V_pU_p^{-1}E_{p+1}U_{p+1}$
4. Right-multiply N_p by V_p^{-1}	$[(V_{p-1}U_{p-1}^{-1})^{-1}K^{p-1}]\backslash[U_pV_p^{-1}K^p]$ $N_pV_p^{-1}$ (same as N_p)	

major difficulties arise when trying to program them effectively. These problems are mainly linked to the high computational cost of the algorithms and to the possible appearance of very big integers during the process. The algorithm is namely valid as long as integer computations have an arbitrary precision. With standard 32 or 64 bit integers, the algorithm is no more accurate. In theory, this problem arises even in small chain complexes. Hafner et al. [20] have exhibited a 10×10 incidence matrix, with no value greater than 10, that induces huge *intermediate* integer numbers in SNF computation.

Deterministic and stochastic algorithms have been proposed to tackle these difficulties. A deterministic method proposed by Dumas et al. [21] consists mainly in ordering rows of incidence matrices by increasing pivot. His method also uses a result proved by Storjohann [22]. This result states that it is possible to choose an integer p such that the SNF is obtained when reducing incidence matrix with moduli p operations. More precisely, p is twice the matrix determinant value. As a particular case, when no torsion arises, the determinant value equals 1, which means that all operations can be done moduli 2. Stochastic algorithms have, for example, been proposed by Giesbrecht et al. [23]. They are generally more efficient than deterministic ones on sparse matrices, but are quite equivalent on dense matrices. They are, however, restricted to the SNF computation and do not extract generators.

3.3. *Effective implementation contributions*

It should be noted that the previously mentioned methods compute Betti numbers and torsion coefficients by transforming incidence matrices in a way that the homology groups remain isomorphic. But nothing ensures that generators are still valid. With our implementation we propose to explore this last issue. As far as we know, only Agoston [19] proposed an algorithm to compute all homology information (including generators), but its algorithm does not address the difficulties mentioned above.

The main steps of our implemented algorithm are described below. All operations made on the incidence matrix implies changes of bases that are stored in suitable matrices. Hence matrices multiplications are avoided in Agoston's method by applying each column operations of the incidence matrix E_p to corresponding rows of E_{p+1}, changes of bases are hence applied step by step.

(1) (*Prepare matrix for Dumas's algorithm.*) The rows of the incidence matrix are ordered by increasing pivot.
(2) (*Same as Dumas.*) The matrix is put in echelon form with as many pivots at 1 as possible by
   - first pass: only elementary row operations are applied,
   - second pass: all rows are reduced according to their gcd.
   - the matrix is now in triangular form: deduce submatrix determinant (which is also the product of the invariant factors).
   - All further integer operations are made modulo twice this determinant. It has indeed been proved (for example by Storjohann) that such a computation using an appropriate modulo preserves the homology information.
(3) (*Different from Dumas.*) Elementary rows and columns operations are performed to compute the modified SNF on the submatrix with non-zero rows. Changes of bases are traced. Agoston's algorithm is used to compute the generators, which are "moduli generators" in the sense they have been partly computed with a modulo.

Different versions are derived from this general algorithm. One version uses Agoston's algorithm (step 3) with a chosen modulo. Another version implement Dumas' method without using moduli. So we compute all homology information of semi-simplicial sets: Betti number and torsion coefficients of all homology groups, sets of generators and sets of "moduli generators".

Also, going one step further, based on theoretical result from cellular homology [8] we try our implementation on cubical sets, as shown in Fig. 4.

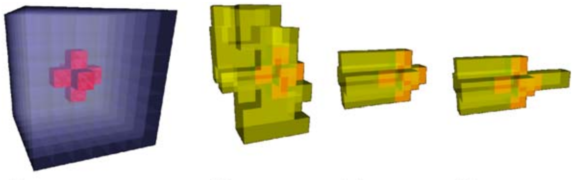


Fig. 4. Detecting cavities with homology generators of dimension 2 (see text).

4. Experimentations

We carry further our preliminary work presented in [24] with several new experimental validations of our work: (i) we not only test our homology computation technique on simplicial objects, but also on digital objects; (ii) we compare "modulo" generators with classical non-modulo generators; (iii) we are able to delineate holes on surfaces but also cavities within volumes; (iv) we study the sparseness of incidence matrix during reduction and derive a sparse implementation of our technique that decreases the time complexity by an order of magnitude.

Our approach is validated with object classically encountered when testing topological invariants (see Fig. 3) and also with discrete objects (see Fig. 4). For each shape, Betti numbers and torsion coefficients are extracted from the modified SNF. The generators are read in the matrices Γ^i. With this information, we are able to delineate each hole of the complex.

Fig. 3 shows the classical shapes and the corresponding generators. For the torus, we obtain two cycles, one for each 1-dimensional hole ($H_1(T) \cong \mathbb{Z} \oplus \mathbb{Z}$). According to the topological nature of the Moebius strip (homotopic to a circle), we found one cycle ($H_1(M) \cong \mathbb{Z}$). For the Klein bottle, two cycles are found, one for the free part of the homology and one for the torsion part ($H_1(K) \cong \mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z}$).

We have computed these generators using the previously described method with moduli operations. In Fig. 3 each object has approximately 2000 triangles. We observe that the "moduli" generators are homologous to those computed with arbitrary precision integers. We guess that this property can be justified in a strict mathematical way but as far as we know there is no indication to invalid or to confirm this property. Usual mathematical approaches are not really interested by the effective representation of the generators, which explains the lack of theoretical results on "moduli" generators.

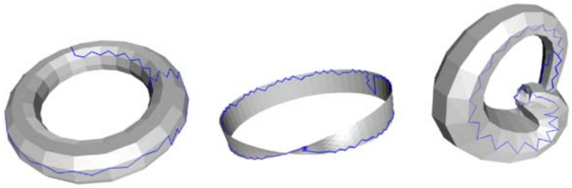


Fig. 3. Examples of homology generators on some classical surfaces.

We have also detected cavities on simple discrete objects. Fig. 4 shows some results using different approaches. Fig. 4(a) is the initial object on which we have computed homology generators. This object is a 3-dimensional topological ball, from which a small 3-ball has been removed. This object has one element for H_2, which corresponds to its cavity. We know that this object has no torsion part, so its generators could be computed using moduli 2 operations. Fig. 4(b) illustrates the moduli 2 generator computed by Agoston's method. Fig. 4(c) shows the generator obtained using Dumas' rows ordering of incidence matrix without moduli. One can see that the generator obtained with this pre-computation modification, is closer to the real cavity. We have observed this phenomena in many examples, and we think that it is mainly due to the rows ordering. The generator of Fig. 4(d) has been computed using moduli 2 operations and Dumas' row ordering. This seems to confirm that a preliminary ordering of rows contributes to obtain generators closer to the intuitive ones.

We have also studied the sparseness of the matrix during its reduction over many examples. In all these examples, we observed that the filling rate of the matrix does not change during its reduction to mSNF. Moreover, incidence matrices obtained from geometric objects are very sparse. This fact is surely due to the main structure of geometric objects we use, which are quite "regular" objects. As an example, for a cube made of 64 voxels and 240 surfels, the size of incidence matrix E_2 is 240×64, on which each column has only six non-zero elements (each voxel has exactly six surfels in its boundary); in other words only 2.5% of the incidence matrix are non-zero elements. As matrices operations have huge computation time on huge matrices, computing homology groups using sparse matrices contributes to optimize running time algorithm. Fig. 5 shows computation time of homology generators using

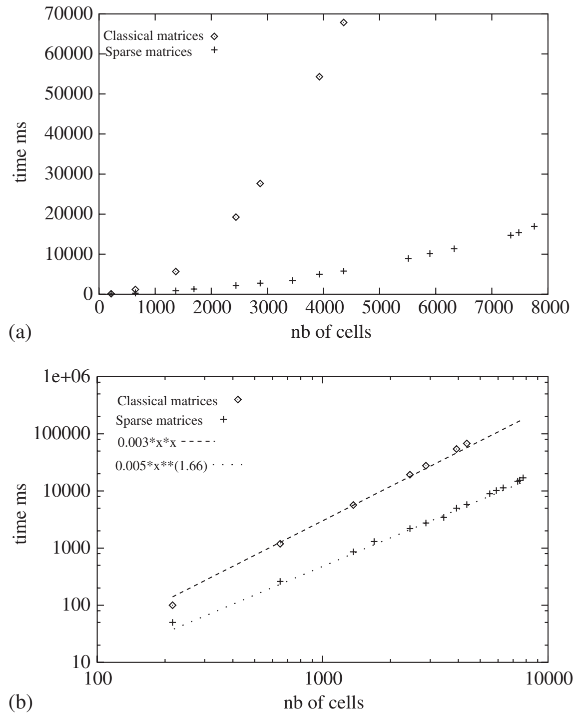


Fig. 5. Time complexity for computing homology generators with classical and sparse matrices implementation: (a) plot of the computation time in ms wrt the number of cells with a linear scale; (b) same plot in log-scale with a comparison with the curves x^2 and $x^{5/3}$ to estimate complexity.

classical matrices, and sparse matrices. For the objects we tested, the algorithm main complexity seems to be interestingly improved, dropping from $O(x^2)$ to $O(x^{5/3})$ if x is the number of cells.

5. Conclusion

To conclude, we have presented and implemented a technique to compute the whole homology of arbitrary finite shapes. We have addressed the problem of extracting generators of the homology groups with a modulo. We have also proposed some computing optimizations, using both Dumas and Agoston algorithms, together with moduli operations. Experiments have been presented, concerning sparseness, huge integer occurring, moduli generators and running time using sparse matrices. Future works will focus on exhibiting the theoretical link between moduli generators and $\mathbb{Z}$-generators. We also intend to study singular homology for hierarchical objects. Indeed hierarchical representations of discrete objects are thus more compact than simplicial or cubical ones. This would allow to compute more effectively homology generators on geometric objects, and at least on very huge objects.

References

[1] Allili M, Corriveau D, Ziou D. Morse homology descriptor for shape characterization. In: Proceedings of the ICPR 2004; 2004.

[2] Kong TY. A digital fundamental group. Computers and Graphics 1989;13(2):159–66.

[3] Boxer L. A classical construction for the digital fundamental group. Journal of Mathematical Imaging and Vision 1999;10:51–62.

[4] Malgouyres R. Computing the fundamental group in digital spaces. International Journal of Pattern Recognition and Artificial Intelligence 2001;15(7):1075–88.

[5] Ayala R, Dominguez E, Francès A, Quintero A. Homotopy in digital spaces. Discrete Applied Mathematics and Combinatorial Operational Research and Computing Science 2004;125.

[6] Kong TY, Rosenfeld A. Digital topology: introduction and survey. Computer Vision Graphics and Image Processing 1989;48(3):357–93.

[7] Delfinado CJA, Edelsbrunner H. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. Computer Aided Geometric Design 1995;12(7): 771–84.

[8] Hatcher A. Algebraic topology. Cambridge: Cambridge University Press; 2002. Available on http://www.math.cornell.edu/~hatcher/AT/ATpage.html.

[9] Kaczynski T, Mrozek M, Slusarek M. Homology computation by reduction of chain complexes. Computers & Mathematics with Applications 1998;34(4):59–70.

[10] González-Díaz R, Real P. Toward digital cohomology. In: Sanniti di Baja G, Svensson S, Nyström I, editors. Proceedings of DGCI'2003. Lecture notes in computer science, vol. 2886. Berlin: Springer; 2003. p. 92–101.

[11] Kaczynski T, Mischaikow K, Mrozek M. Computational homology. Berlin: Springer; 2004.

[12] Munkres JR. Elements of algebraic topology. Perseus Books; 1984.

[13] May JP. Simplicial objects in algebraic topology. Princeton, NJ: Van Nostrand; 1967.

[14] Curtis E. Simplicial homotopy theory. Advanced Mathematics 1971;6:107–209.

[15] Lang V, Lienhardt P. Geometric modeling with simplicial sets. In: Proceeding of pacific graphics'95. Seoul, Korea, 1995.

[16] Desbarats P, Gueorguieva S. Topological mainframe for numerical representations of objects. In: Proceeding of the international conference computer science and its applications. Lecture notes in computer science, vol. 2668. Berlin: Springer; 2003. p. 498–507.

[17] Lienhardt P, Elter H. Different combinatorial models based on the map concept for the representation of different types of cellular complexes. In: Proceedings of IFIP TC 5/WG II workshop and conference on geometric modeling in computer graphics. Berlin: Springer; 1993.

[18] Cairns SS. Introductory topology. Ronald Press Company; 1961.

[19] Agoston MK. Algebraic topology, a first course. New York: Marcel Dekker; 1976.

[20] Hafner JL, McCurley KS. Asymptotically fast triangularization of matrices over rings. In: Johnson D, editor.

SIAM, San Francisco, CA, USA Proceedings of the first annual ACM–SIAM symposium on discrete algorithms (SODA'90); 1990. p. 194–200.

[21] Dumas JG, Saunders B, Villard G. On efficient sparse integer matrix smith normal form computations, Journal of Symbolic Computation 2001.

[22] Storjohann A. Near optimal algorithms for computing Smith normal forms of integer matrices. In: Lakshman YN, editor. Proceedings of the 1996 international symposium on symbolic and algebraic computation. ACM; 1996. p. 267–74.

[23] Giesbrecht M. Nearly optimal-algorithms for canonical matrix-forms. SIAM Journal of Computing 1995;24(5): 948–69.

[24] Peltier S, Alayrangues S, Fuchs L, Lachaud J-O. Computation of homology groups and generators. In: Andrès E, Damiand G, Lienhardt P, editors. Proceedings of international conference on discrete geometry for computer imagery (DGCI'2005), Poitiers, France. Lecture notes in computer science, vol. 3429. Berlin: Springer; 2005. p. 195–205.