Computing Homology Generators for Volumes Using Minimal Generalized Maps

Guillaume Damiand, Samuel Peltier, Laurent Fuchs

Computing Homology Generators for Volumes using Minimal Generalized Maps* **

Guillaume Damiand[1], Samuel Peltier[2], and Laurent Fuchs[3]

[1] LaBRI, Université Bordeaux 1, UMR CNRS 5800, 33405 Talence cedex, France
damiand@labri.fr
[2] IFP, 92852 Rueil-Malmaison Cedex, France
samuel.peltier@ifp.fr
[3] SIC, Université de Poitiers, 86962 Futuroscope Chasseneuil Cedex, France
fuchs@sic.univ-poitiers.fr

Abstract. In this paper, we present an algorithm for computing efficiently homology generators of 3D subdivided orientable objects which can contain tunnels and cavities. Starting with an initial subdivision, represented with a generalized map where every cell is a topological ball, the number of cells is reduced using simplification operations (removal of cells), while preserving homology. We obtain a minimal representation which is homologous to the initial object. A set of homology generators is then directly deduced on the simplified 3D object.

Key words. topological features, homology generators, generalized maps.

1 Introduction

In this paper, we present an algorithm for computing efficiently the three dimensional *minimal generalized map* homologous to a given 3D object. Then we show how cells that belong to homology group generators can be directly characterized onto this minimal object.

Homology is a topological invariant, classically studied in algebraic topology [1], which characterizes an object by its "holes" in each dimension. This corresponds to connected components in dimension 0, tunnels in dimension 1, cavities in dimension 2; this notion of hole can be generalized in any dimension. For each dimension d, the number of $d-$dimensional holes of a given object is called its d^{th} Betti number. Homology group generators are d-dimensional paths (edges, faces) that surround the d-dimensional holes.

Generalized maps [2] are a combinatorial cellular structure which can be used to represent both topological and geometrical information of a three dimensional subdivision, with particular properties that makes it a good model for features extraction. In this work, generalized maps are used to compute a minimal cell decomposition (called *minimal map*) of a 3-manifold in $\mathbb{R}^3$ with the same homology as the initial 3D object. For that, we extend in 3D the work of [3]. Starting from the initial subdivision, where every cell is equivalent to a topological ball,

* Partially supported by the ANR program ANR-06-MDCA-008-05/FOGRIMMI.
** Paper published in Proceedings of 12th IWCIA 2008, LNCS 4958, pp. 63-74, 2008. Thanks to Springer-Verlag Berlin Heidelberg. The original publication is available at http://www.springerlink.com/content/8274885tl8720800/

the number of cells is progressively reduced using removal operations [4]. At the end of the simplification, we show that the minimal obtained object is homologous to the initial subdivision. Moreover, this minimal map allows us to directly characterize cells of the subdivision that belong to homology generators.

This paper is organized as follows. In Section 2, basic notions related to generalized maps and homology groups are recalled. The removal operations, which are used to compute a minimal map are introduced. In Section 3, the algorithm for computing a minimal map is detailed, and its complexity is discussed. We give the arguments to show that our algorithm provides a minimal object with the same homology as the initial one. Finally, Section 4 concludes and gives some perspectives.

2 Preliminaries

In this section some basic notions are presented. Our algorithm deals with subdivisions of 3D topological spaces. A subdivision is a partition into 4 subsets whose elements are $\{0,1,2,3\}$-*cells* of dimension 0, 1, 2 and 3 (respectively called vertices, edges, faces and volumes). The border of an i-cell is a set of *(j<i)*-cells. Two cells are *incident* if one belongs to the border of the other, and two i-cells are *adjacent* if they are both incident to a common *(j<i)*-cell. The *cell degree* of an i-cell c is the number of distinct *(i+1)*-cells incident to c. We only consider quasi-manifolds[4].

Generalized Maps For 3D quasi-manifolds, incidence and adjacency relations can be represented using 3-dimensional generalized maps (3-*G-maps*) [5]. Intuitively, a 3D generalized map can be obtained by successive (from volumes to vertices) decompositions of a 3D object into elementary elements called *darts*. Then, adjacency relations between i-cells are reported onto darts (denoted α_i). *Involution*[5] α_i connects the two darts incident to the two adjacent i-cells incident to the darts (see [5] for a formal definition).

Within the generalized map framework, all cells are implicitly represented through the notion of *orbit*. Given, $\{p_1,\ldots,p_j\}$, a set of involutions, and a dart d, an orbit $< p_1,\ldots,p_j > (d)$ is the set of darts that can be reached with a breadth-first search algorithm, starting with d, and using all combinations of p_i $\forall k, 1 \leq k \leq j$.

Removal Operations Removal operations are the basic operations used during our algorithm. The removal of an i-cell c (called i-removal of c) leads to the merging of the two *(i+1)*-cells incident to c. For 3D subdivisions, i-removal

[4] A n-dimensional quasi-manifold is an nD space subdivision which can be obtained by gluing together n-dimensional cells along *(n-1)*-dimensional cells. In such subdivision, an *(n-1)*-cell cannot belong to the boundary of more than two n-cells. This notion is weaker that the manifold property, see [5].

[5] An involution f on S is a one to one mapping from S onto S such that $f = f^{-1}$.

operations are defined for $i = 0, 1, 2$ (see [4] for the definition of removal operations). The i-removal operation consists mainly in locally modify the α_i relation for each dart that belongs to the neighborhood of the removed cell.

Homology In this part, basic homology notions are recalled; interested readers can find more details in [1] for algebraic approach and [6] for more computational approach. The notion of homology is defined in an algebraic way using the sets of $i-$cells used to describe the 3D manifold. Within this context, a *p-chain* (i.e. a chain of dimension p) is a formal sum of p-cells. From this, the group C_p of p-chains is defined. The boundary of a p-chain is defined as the sum of the boundaries of its p-cells. Note that the boundary of a p-chain must be a $(p-1)$-chain. The set of p-chains which have a null boundary (i.e. *p-cycles*) is a subgroup of C_p (denoted Z_p). The set of p-chains which are boundaries of a $(p+1)$-chain (i.e. *p-boundaries*) form a subgroup of C_p (denoted B_p).

An essential property is that the boundary of any boundary is null. Hence, every boundary is a cycle and B_p is contained in Z_p. Two p-cycles z_1 and z_2 are *homologous* if their difference is a boundary, i.e. there is a $p+1$-chain f such that $z_1 = z_2 + \partial f$. From this, an equivalence relation can be defined and the homology class of z is the set $\{z + b \mid b \in B_p\}$. The *homology group of dimension* p, denoted H_p, is defined as the quotient group Z_p/B_p, and its elements are the homology classes. For a group G, a *set of generators* is a maximal subset S of elements of G, such that every element of G can uniquely be defined as a linear combination of elements of S.

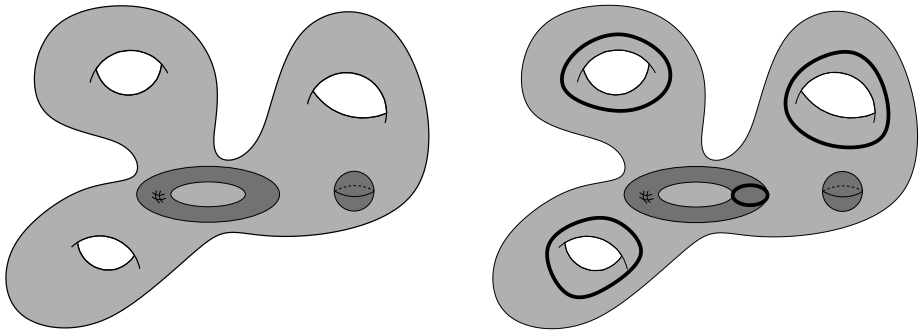
3 Computation of Homology Generators

In this section, we present an algorithm for computing the generators of homology groups of 3D orientable objects with cavities, i.e. an object bounded by one or more orientable surfaces.

3.1 Related Works

In [3], the authors propose an algorithm for computing a minimal representation of a 2D surface. It is shown that the homology generators H_1 can be directly deduced from this minimal representation.

In [7], the authors study the homology of 3D manifolds object X bounded by several surfaces. Indeed the considered objects are 3D balls with tunnels and cavities. For example, Fig. 1(a) illustrates a 3D object which contains three tunnels and two cavities. The authors then show how to compute homology generators H_1 and H_2 of such objects. Moreover, it is shown that if X is bounded by $j+1$ surfaces $s_0, ..., s_j$, then the set $\{s_1, ..., s_j\}$ is a basis of $H_2(X)$. They also introduce the notion of longitudinal and latitudinal generators of a surface and show that if s_0 denotes the external boundary of X, then the set of latitudinal generators of $\{s_1, ..., s_j\}$ together with the longitudinal generators of s_0 forms a basis of $H_1(X)$ (s_0 is called external surface, others s_i are called internal



(a) (b)

Fig. 1. (a): a 3D object with 3 tunnels and 2 cavities, (b): the latitudinal generator of the torus cavity together with the longitudinal generators of the external boundary forms a basis of the first homology group of the object.

surfaces). For example, on Fig. 1(b), the homology generators of the object represented in (a) is made of the 3 longitudinal generators of the external surface together with the latitudinal generator of the torus cavity. Note that the first homology groups of a sphere is trivial.

3.2 Simplification Algorithm

In this section, we extend the algorithm presented in [3] to 3D regions with cavities and tunnels. Our algorithm, given in Algorithm 1, provides the same result as in [7] but by working only with the initial subdivision of the object, and use basic simplification operations and combinatorial characteristics of cells.

Starting from a subdivision of a 3D object, where each cell is homeomorphic to a topological balls, we simplify progressively the subdivision, by decreasing cell dimension. First we remove faces (i.e. 2-cells) while keeping the volume homeomorphic to a topological ball. For that, we keep *fictive faces*, i.e. faces that are "inside" the volume, and whose removal involves map disconnection. We obtain a representation made of only one volume. To compute the minimal representation for other cells, we use the algorithm of [3] on each surface of the map. But for that, it is necessary to remove all the fictive faces in order to obtain 2D objects. After having computed the minimal representation of each surface, we need to reconstruct the minimal representation of the 3D object. This is achieved by adding the minimal number of fictive faces in order to obtain a connected volume homeomorphic to a topological ball.

This minimal subdivision is homologous to the initial object, and allows to directly compute the homology generators of the initial object by simple cell characterization. Moreover, this principle gives some perspectives to generalize our approach to n-dimensional objects.

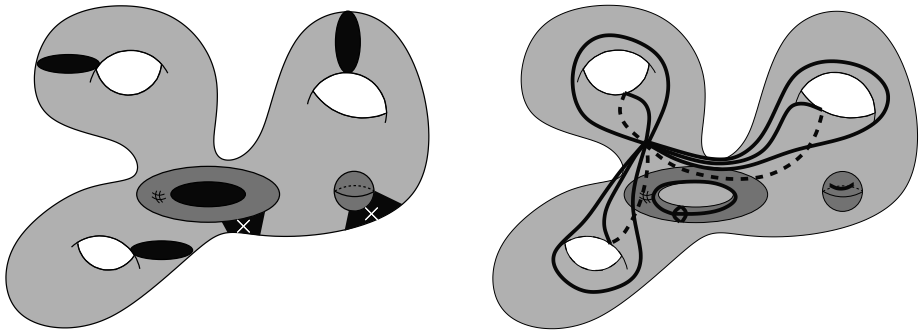
Algorithm 1: Simplification of a 3D subdivision in its minimal homologous form.

Input: A generalized map M representing an orientable subdivision of a 3D object such that each cell is homeomorphic to a topological balls
Output: The minimal subdivision homologous to M

```
1  foreach face f of the map do
       if the degree of f is 2 then
           Remove f;
       else if f is a dangling face then
           push(P, f);
           repeat
               f ← pop(P);
               push in P all the dangling faces adjacent to f;
               Remove f;
           until empty(P) ;
       else Mark f as fictive face;
2  Mark external surface, without considering fictive faces;
3  Remove all fictive faces;
4  Compute the H_1 generators of each surface;
5  if the external surface is a sphere then
       ext ← the only edge of the external surface;
   else for one edge out of two e of the external surface do
       Add a fictive face along e;
       ext ← e;
   foreach internal surface s do
       if s is a sphere then
           int ← the only edge of s;
       else for one edge out of two e of s do
           Add a fictive face along e;
           int ← e;
       Add a fictive face between int and ext;
```

Now we detail more precisely each step of our algorithm. The first simplification step (line 1 of Algorithm 1) is similar to the algorithm described in [3], which provides a minimal representation (in term of cells) in the case of 3D objects bounded by only one surface. The difference concerns the dimension of processed cells, since we need here to consider in first volumes (3-cells) by removing faces (2-cells), while in [3] faces (2-cells) are processed by removing edges (1-cells).

During this step, we remove either degree two faces (i.e. faces between two different volumes), or dangling faces (i.e. faces inside a volume). Indeed, both faces can be removed without modifying the homology of the subdivision. When we remove a dangling face f, we need to reconsider dangling faces adjacent to f. Indeed, some faces adjacent to f can be non-dangling before f was removed and become dangling after its removal, hence non minimal subdivision can be obtained. To reconsider these faces, we use a stack of dart which contains one



(a) (b)

Fig. 2. (a) After the first simplification step, the 3D object contains some fictive faces. Black faces are fictive faces which stop up the tunnels, and black faces marked with crosses are fictive faces that link the different surfaces. (b) Result obtained after having disconnected surfaces and compute H_1 generators for each surface. External surface is represented by 6 edges and 1 vertex, the torus cavity is represented by 2 edges and 1 vertex and the sphere by 1 edge and 2 vertices. The obtained map is disconnected in three connected components.

dart of each reconsidered face, and when we remove a dangling face, we put in this stack one dart for each dangling adjacent faces.

The subdivision obtained after applying the first step is composed by only one volume homeomorphic to a 3D ball, some real faces (faces with each dart 3-free) and some fictive faces (faces with each dart not 3-free). Fig. 2(a) illustrates a possible subdivision that can be obtained after the first simplification step. In this subdivision, there is exactly one volume, and thus the non 3-free darts belong necessarily to fictive faces (i.e. degree one faces incident twice to the volume). Note that there are two types of fictive faces: the ones which link different surfaces in order to keep only one connected component, and the ones which stop up the tunnels, and that allow to keep the volume homeomorphic to a topological ball.

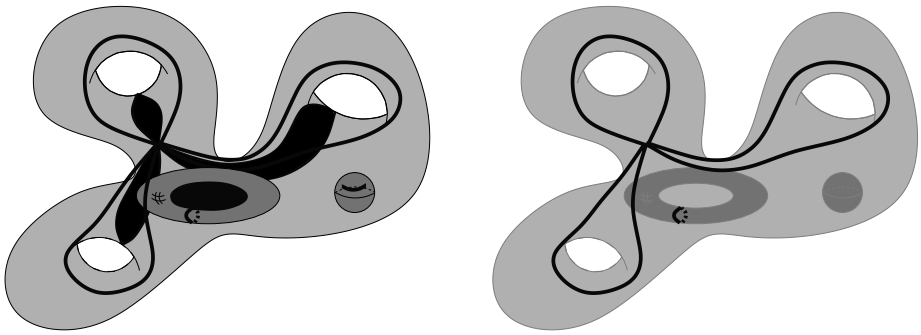
In the second step (line 2 of Algorithm 1), we work on each surface. Firstly, we mark the external surface. This step is necessary since H_1 generators are longitudinal for external surface whereas they are latitudinal for internal surfaces. Finding a dart of the external surface can be achieved directly by searching among all the darts the one associated with the smaller 3D coordinates. Starting from this dart, we can run through all the darts of the external surface by using a breadth first search algorithm which uses involutions α_0, α_1, and α_2, and which jumps over darts of fictive faces. Darts not 3-free belong to fictive faces, and darts 3-free and non-marked belong to internal surfaces.

Then, all the fictive faces are removed in order to continue the simplifications in smaller dimension (line 3 of Algorithm 1). The next step (line 4 of Algorithm 1)

consists in computing independently the H_1 generators for each surface. This step is not detailed here since it is achieved by using the work of [3].

After this step, we obtain the minimal representation of each surface, composed with one face, one edge and two vertex if the corresponding surface is a sphere, and composed with one face, $2k$ edges and one vertex if the corresponding surface is a torus with k holes. Each edge of the minimal representation belong to an H_1 generator of the 2D object, except if the corresponding surface is a sphere. Indeed, in such a case, there is no H_1 generator (see Fig. 2(b)).

Now, we have all necessary information to reconstruct the minimal representation of the initial 3D subdivision. This is the goal of the last step of our algorithm (line 5 of Algorithm 1). We firstly add fictive faces along one edge out of two of each surface, except for surfaces which represent spheres. This step is necessary to stop up the tunnels, and then obtain a volume homeomorphic to a topological ball. There are two cases to consider: if the surface is a sphere, there is no fictive face to add, otherwise the surface is a torus with k holes, and in this case there are k fictive faces to add in order to cut all the tunnels. Since the surface is composed with $2k$ edges in its minimal representation, to add k fictive faces, we just need to add one fictive face along one edge out of two. Then, fictive faces are added in order to connect each internal face with the external face (Fig. 3(a)).



(a) (b)

Fig. 3. (a) The minimal map obtained at the end of our algorithm (partial representation, fictive faces which link internal surfaces and the external surface are not represented.). This map is homologous to the initial subdivision. (b) H_1 generators are composed with all the edges non-incident to fictive faces, and non-incident to spheres.

After these two steps, we obtain a map where each cell is homeomorphic to a topological ball. This is necessary in order to ensure that this map is homologous to the initial subdivision. Moreover, this map is composed with $j+1$ real faces, one for each surface (external and internal surfaces) of the initial subdivision.

This minimal map M gives directly the homology generators:

1. H_2 generator is composed with all the real faces that belong to internal surfaces of M;
2. H_1 generator is composed with all the edges of M not incident to fictive faces, and which are not incident to a sphere (see Fig. 2);
3. H_0 generator is always isomorphic to $\mathbb{Z}$ since we only consider one connected object.

3.3 Validity of the Method

In this section, we use the works of [8] and [7] to show that the resulting object is *minimal*; homology is preserved; and homology generators can be directly characterized.

The Object is Minimal and Homology is Preserved The first step of the algorithm (the simplification process) preserves the homology. In [8], the authors use *interior face reduction* to simplify a simplicial complex. These reductions are equivalent to the removal operations used in [3] which consist in taking a common i-face c of exactly two $(i+1)$-simplices a and b, and delete c and replace b and c by a cell which represents their union. It is proven in [8] that interior face reductions preserve homology and thus we can conclude that this is also the case for removal operations.

After the removal of fictive faces, each surface is simplified in its minimal form while preserving homology (step 3 and 4 which use [3]). The last step build a minimal representation of the initial 3D object. This is done into 3 steps:

1. each internal generator of the external surface is filled;
2. each longitudinal generator of internal surfaces is filled;
3. each internal surface is connected to the external surface.

It is shown in [9] that adding faces into each latitudinal generator *cut* the volume corresponding to the external surface into a topological ball. For each internal surface homeomorphic to a torus with g holes, each latitudinal generator (tunnels of internal surfaces) is filled by a face as these generators are no longer generators when the internal surface is considered as a cavity [7]. Lastly, each cavity is connected to the external surface and the obtained volume is minimal as we have added the minimal number of fictive faces. Moreover, each cavity is homeomorphic to a topological ball as each tunnel has been filled.

Direct Characterization of Homology Generators As mentioned before, if a 3-manifold X is bounded by $j+1$ surfaces $s_0, ..., s_j$, then the set $\{s_1, ..., s_j\}$ is a basis of $H_2(X)$ (see [7]). This set corresponds to all non fictive internal faces.

Moreover, the set of longitudinal generators of s_0 together with all the latitudinal generators of all internal surfaces forms a basis of H_1. Once the minimal form of each surface has been computed (step 4 of the algorithm), all the edges are either a longitudinal or a latitudinal generator. As seen before, all the latitudinal generators of s_0 and all the longitudinal generators of the internal surfaces

are incident to a fictive face. Thus homology generators of the 3D object are all the edges that are not incident to a fictive face. Note that detecting an edge incident to a fictive face is done in a combinatorial way, thus we do not need the linking numbers or perturb the generators as it is done in [7].

3.4 Complexity

The complexity of Algorithm 1 is equal to $O((3-\chi)\times n)$ with $\chi = \sum_{i=1}^{k} \chi_i$, χ_i being the Euler characteristic of surface s_i, and n is the number of darts of the subdivision.

The first step is linear in number of faces of the map. Firstly, each face is consider at most twice, a first time during the loop around all the faces of the map, and a second time during the second loop which remove dangling faces. When the face is reconsidered, it is removed and thus it will be never reconsidered later.

To test the face degree, we use union-find trees [10] allowing to represent efficiently disjointed sets. This structure is handled by two operations: *find* which returns, given an element, the representative of the set, and *union* which allows to merge two sets. The amortized cost of a series of m union-find operations on n elements can be done in time $O(n.\alpha(m,n))$ with $\alpha(m,n)$ being the inverse Ackermann function which grows extremely slowly, and which is less than 5 in practical cases (see [10] for the demonstration about the complexities).

We link each dart of a volume of the initial subdivision with an union-find tree representing the volume. When we remove a face, we merge both corresponding trees by using the *union* operation. The test if d and $\alpha_3(d)$ belong to the same volume is simply achieved by testing if *find(d)* is equal to *find*($\alpha_3(d)$). Since we only consider one 3D object, subdivided in several volumes in the initial subdivision, we are sure that if d and $\alpha_3(d)$ belong to the same volume, the corresponding face is a degree one face (i.e. incident twice to the volume) and otherwise the face is a degree two face.

The face removal is achieved locally, by running through each edge incident to the face to remove and by modifying locally α_2 involutions. Moreover, to test if a face f is dangling or not, we have to run through each edge incident to f, and test if the edge is only incident to f, i.e. if d a dart of the edge is such that $\alpha_{23}(d) = d$.

To summarize the first step, the cost of the test on the face degree can be bounded by 5, the cost of the dangling face test is linear in number of edges of the face, and the face removal is also achieved linearly in number of edges of the face. This shows that the first step of Algorithm 1 is linear in number of darts of the map (indeed, the number of darts is always greater than the number of cells).

The second step (mark external surfaces) is also achieved linearly in number of darts of the map. Indeed, find the smaller dart of the map need to run through all the darts. Then, mark the surface is achieved by using for example a breadth first search algorithm by using involutions α_0, α_1, and α_2, and jumping over darts of fictive faces.

The fictive faces removal is achieved linearly in number of darts of the map, since we need to consider each dart, and modify locally involution α_2 for those that belong to fictive faces. Testing if a dart belongs to a fictive face is achieved in constant time (if d is 3-free or not).

Compute H_1 generators of a surface s_i is achieved in $O((3-\chi_i) \times n_i)$ where n_i is the number of darts of surface s_i, and χ_i is the Euler characteristic of the surface (see [3]). Since we compute H_1 generators for each surface, we obtain the final complexity by adding the complexity of each surface, which gives $O((3-\chi) \times n)$ with $\chi = \sum_{i=1}^{k} \chi_i$ and n is the number of darts of the map.

The last step is achieved linearly in number of darts of the map since we just run through all the edges of the map, using the mark on darts to distinguish longitudinal and latitudinal generators, and distinguish external and internal surfaces. Moreover, adding a face along a loop is achieved in constant time, and adding a face between two edges that belong to two distinct surfaces is also achieved in constant time.

This gives the global complexity of our method: $O((3-\chi) \times n)$. Indeed, all steps are linear in number of darts, except the step which allows to compute H_1 generators of each surface which depends on the number of darts multiply by the sum of the Euler characteristics of all surfaces.

3.5 Geometry of Generators

Algorithm 1 gives a method to compute the minimal representation homologous to a given 3D subdivision. This minimal representation allows to characterize directly the homology generators of the object. Depending on the need of applications, it is sometime necessary to embed the generators onto the original subdivision, for example to draw the H_1 generators onto the surfaces.

This is directly possible for H_2 generators since they are composed with all the non-fictive faces of the minimal subdivision, and each non-fictive face corresponds exactly to one surface into the original subdivision. This surface can be retrieved easily by running through the original subdivision and following the boundary (i.e. darts 3-free) and jumping over darts not 3-free.

But the problem is more complex for H_1 generators. Indeed, in the last step of Algorithm 1, fictive faces are added along one edge out of two, without particular properties on chosen edges. This is possible because all the configurations obtained by adding fictive faces along one edge out of two are homologous. However, configurations are not equivalent if we take into account the links with the initial subdivision. Indeed, in such a case, we need to distinguish latitudinal and longitudinal generators since they do not have the same role for the homology of the 3D object.

Algorithm 1 can easily be modified in order to compute a minimal subdivision which is homologous to the original subdivision, and which take into account these two kinds of generators. It is only necessary to make two modifications:

1. after having computed H_1 generators, we determine the class (latitudinal or longitudinal) of these generators by using the algorithm given in [7];

2. when we add fictive faces along edges, we need to chose edges which are not H_1 generators. These edges are either latitudinal generators that belong to the external surface, or longitudinal generators that belong to an internal surface. Since edges are distinguish by the previous modification, this modification can be directly added in the last step of Algorithm 1, by replacing the loop "For one edge out of two e of the external surface" with "ForEach latitudinal edge e of the external surface", and replacing the second loop "For one edge out of two e of s" with "ForEach longitudinal edge e of s".

With these two basic modifications, the computed minimal subdivision is not only homologous to the initial subdivision but H_1 generators can also be embedded onto the initial surfaces. Note that compute this embedding is not so straightforward than for H_2 generators since we need to keep links between edges of the minimal representation and edges of the subdivision during the whole simplification process.

However, these modifications involve complexity modifications. Indeed, to distinguish longitudinal and latitudinal generators, we use the method given in [7] which is in $O(n^2\bar{g})$ with $\bar{g} = max_{1\leq i\leq k} g_i$, and g_i is the genus of the surface s_i. This step is thus the more expensive part of the modified method, and so gives the global complexity of the method. One perspective of this work is to either improve this part, or remove it by computing generators by using only the combinatorial structure.

4 Conclusion

In this paper, we have presented an algorithm that computes the minimal generalized map homologous to a given 3D object, orientable and with or without cavities. Thanks to this minimal form, we can characterize easily and directly the cells that belong to homology generators. This gives a new method to compute efficiently the generators of a 3D object.

The main interest of our approach is to use a simple method which simplify the given subdivision. Moreover, the method is efficient since the complexity of our algorithm is in $O((3-\chi)\times n)$ with $\chi = \sum_{i=1}^{k} \chi_i$, where each χ_i is the Euler characteristic of surface s_i, and where n is the number of darts of the subdivision.

We have proposed a modified version of our algorithm which allows to embed the H_1 generators onto the initial subdivision. To do that, the method given in [7] to distinguish longitudinal and latitudinal generators is used. However, with this additional step, the complexity of the method become in $O(n^2\bar{g})$ with $\bar{g} = max_{1\leq i\leq k} g_i$, and g_i is the genus of the surface s_i.

Another main interest of our approach is that the simplification is made by decreasing cell dimension. This allows to reuse previous work in 2D [3]. We disconnect the 3D object into several 2D surfaces by removing fictive faces. After having 2D minimal representations, we insert back fictive faces in order to obtain the 3D minimal representation. This principle can be generalized in nD, where we simplify an nD object into a representation with only one n-cell, then $(n-1)$ fictive cells are removed to process independently $(n-1)$D objects. Lastly, $(n-1)$

fictive cells are inserted back to obtain the minimal representation of the initial object. However, there is a lot of work to do to validate the method. We need to show that the homology is preserved, and show the link between the minimal representation and the homology generators. This is one perspective of this work: extend our method to deal with nD objects. Moreover, we can also study how to consider orientable or non-orientable objects, with or without boundaries.

Another perspective is to improve the step which allows to distinguish longitudinal and latitudinal generators. Indeed, this step is necessary if we need to keep a link between the minimal representation and the original object, but using the method of [7] leads to increase the complexity of our algorithm. We want to study the possibility to compute this information directly onto the subdivision, by using the fictive faces to characterize the different edges of the original object, and propagate these information during the simplification steps.

References

1. Hatcher, A.: Algebraic Topology. Cambridge University Press (2002) available on http://www.math.cornell.edu/∼hatcher/AT/ATpage.html.
2. Lienhardt, P.: Topological models for boundary representation: a comparison with n-dimensional generalized maps. Commputer Aided Design **23** (1991)
3. Damiand, G., Peltier, S., Fuchs, L.: Computing homology for surfaces with generalized maps: Application to 3d images. In: Procedings of 2nd International Symposium on Visual Computing. Volume 4292 of LNCS., Lake Tahoe, Nevada, USA, Springer-verlag (2006) 235–244
4. Damiand, G., Lienhardt, P.: Removal and contraction for n-dimensional generalized maps. In: Discrete Geometry for Computer Imagery. Number 2886 in Lecture Notes in Computer Science, Naples, Italy (2003) 408–419
5. Lienhardt, P.: N-dimensional generalized combinatorial maps and cellular quasi-manifolds. International Journal of Computational Geometry and Applications **4** (1994) 275–324
6. Kaczynski, T., Mischaikow, K., Mrozek, M.: Computational Homology. Volume 157 of Applied Mathematical Sciences. Springer (2004)
7. Tamal, K.D., Sumanta, G.: Computing homology groups of simplicial complexes in $\boldsymbol{R^3}$. Journal of the ACM **45** (1998) 266–287
8. Kaczynski, T., Mrozek, M., Slusarek, M.: Homology computation by reduction of chain complexes. Computers & Math. Appl. **34** (1998) 59–70
9. Kartasheva, E., Gasilov, V.: Computer aided analysis of the polyhedrons topology. In: GraphiCon97. (1997) 60–66
10. Tarjan, R.: Efficiency of a good but not linear set union algorithm. Journal of the ACM **22** (1975) 215–225