HAL
open science

Topological Map: An Efficient Tool to Compute Incrementally Topological Features on 3D Images

Guillaume Damiand, Samuel Peltier, Laurent Fuchs, Pascal Lienhardt

To cite this version:

Guillaume Damiand, Samuel Peltier, Laurent Fuchs, Pascal Lienhardt. Topological Map: An Efficient Tool to Compute Incrementally Topological Features on 3D Images. 11th International Workshop on Combinatorial Image Analysis (IWCIA 2006), Jun 2006, Berlin, Germany. pp.1-15, 10.1007/11774938_1 . hal-00366101

HAL Id: hal-00366101

https://hal.science/hal-00366101v1

Submitted on 25 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Topological Map: An Efficient Tool to Compute Incrementally Topological Features on 3D Images

Guillaume Damiand, Samuel Peltier, Laurent Fuchs, and Pascal Lienhardt

SIC - bât. SP2MI, Bvd M. et P. Curie
BP 30179, 86962 Futuroscope Chasseneuil Cedex - France
{damiand,peltier,fuchs,lienhardt}@sic.univ-poitiers.fr

Abstract. In this paper, we show how to use the three dimensional *topological map* in order to compute efficiently topological features on objects contained in a 3D image. These features are useful for example in image processing to control operations or in computer vision to characterize objects. Topological map is a combinatorial model which represents both topological and geometrical information of a three dimensional labeled image. This model can be computed incrementally by using only two basic operations: the removal and the fictive edge shifting. In this work, we show that Euler characteristic can be computed incrementally during the topological map construction. This involves an efficient algorithm and open interesting perspectives for other features.
Keywords. topological features, model for image representation, inter-voxel boundaries, combinatorial map.

1 Introduction

In this paper, we show how to use the three dimensional *topological map* [1, 2] in order to compute efficiently topological features on objects contained in a 3D image. Topological map is a combinatorial model which represents both topological and geometrical information of a three dimensional labeled image with particular properties that makes it a good model for features extraction. Indeed, it represents the topology of 3D labeled images with a minimal number of cells, while conserving all the region adjacencies and incidences.

More precisely, the topological map is incrementally built from a 3D image by using simple removal operations of subdivision cells that verify particular properties. Moreover, removal operations are controlled in order to preserve topological information. After all removals, the topological map represents the regions of a 3D labeled image by their boundaries, which are closed orientable subdivided surfaces.

The main idea of this work is to incrementally compute topological features on regions of a 3D image during the topological map construction. We present here the case of Euler characteristic computation; this is a first example and our approach can be extended to several topological features (as canonical polygonal schema or homology classes computation [3]).

Euler characteristic χ of a subdivided object is the alternating sum of numbers of cells (vertices, edges, faces, etc). Let S be a closed orientable subdivided surface and let $\#V$ (resp. $\#E$, $\#F$ and g) be its number of vertices (resp. edges, faces and tunnels[1]). In this case, it is well known that $\chi(S) = \#V - \#E + \#F = 2(1-g)$ [4] gives the complete classification of surfaces.

Euler characteristic and its variants have several applications to image analysis and digital geometry [5]. For example, it can be used to prevent topological alterations in a transformation process or to validate a given segmentation.

Usually, Euler characteristic is computed from a given subdivision, see [6–9] and references therein. Indeed, it is difficult to analyze the consequences of local changes (adding or removing cells) for topological features. However in our approach, thanks to image scanning and to topological map, consequences of adding cells to the subdivision can be translated into local cases analysis and allows us to obtain the variation of the topological features. Hence the Euler characteristic is computed during the topological map construction with only a small additional cost.

To the authors knowledge such incremental approach had not been yet proposed. In the general context of pavings, an incremental algorithm can be deduced from some results of [10] but this general approach is not well suited for 3D digital imagery.

The paper is organized as follows: Section 2 gives some recalls on topological map. Section 3 presents our incremental method to compute incrementally Euler Characteristic and Section 4 concludes and gives some perspectives.

2 Recalls on Topological Maps

2.1 Combinatorial Maps

A subdivision of a 3D topological space is a partition of the space into 4 subsets whose elements are 0D, 1D, 2D and 3D *cells* (respectively called vertices, edges, faces and volumes, and noted i-cell, $i = 0 \ldots 3$). Border relations are defined between these cells, where the border of an i-cell is a set of *(j<i)*-cells. Two cells are *incident* when one belongs to the border of the second, and two i-cells are *adjacent* if they are both incident to a common *(j<i)*-cell.

The topology of nD subdivision of orientable spaces without boundary can be represented by n-dimensional combinatorial maps, or *n-maps* [11–15]. Intuitively, a 3D combinatorial map can be obtained by successive decompositions of an orientable 3D object. We first distinguish the volumes of this object, then the faces of these volumes, and then the edges of these faces. The elements resulting from the last decomposition are called *darts* and are the basic elements of the combinatorial map definition. To obtain the map, adjacency relations between i-cells are reported onto darts (denoted β_i). These β_i have to verify some particular properties in order to ensure the validity of the represented subdivision (for

[1] or holes in more general topological context

example β_1 is a permutation and other β_i are involutions, see for example [16] for the formal definition).

We present an example of combinatorial map in Fig. 1B, and the corresponding represented object in Fig. 1A. β_1 connects an oriented edge and the following oriented edge incident to the same face and the same volume, β_2 connects the two faces incident to the same edge and the same volume, and β_3 connects the two volumes incident to the same edge and the same face. In order to simplify the figures, β_i are not explicitly drawn but can be (generally) deduced from the shape of objects.

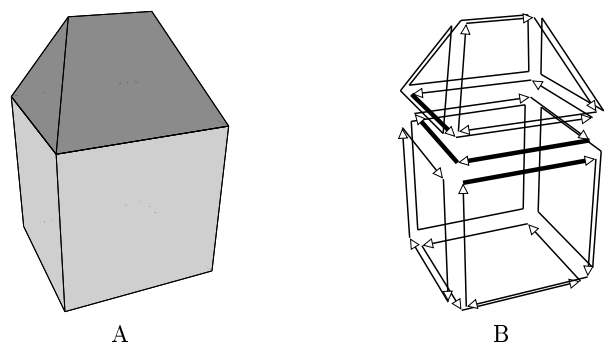


Fig. 1. Usual representation of a 3D combinatorial map. (A) A 3D object. (B) Implicit representation of the corresponding combinatorial map, where β_i applications are not explicitly drawn.

Within the combinatorial map framework, all cells are implicitly represented through the notion of *orbit*. Intuitively, an orbit $< \beta_{i_1}, \ldots, \beta_{i_j} > (d)$ is the set of darts that can be reached with a breadth-first search algorithm, starting with d, and using all combinations of all β_{i_k} or $\beta_{i_k}^{-1}$ permutations $\forall k, 1 \leq k \leq j$. With this notion, each cell is defined as a particular orbit. Based on the cells definition, we can retrieve the classical *cell degree* notion. The degree of an i-cell c is the number of distinct *(i+1)*-cells incident to c. Note that in a n-dimensional space, the degree is not defined for n-cells, since *(n+1)*-cells do not exists in such a space.

2.2 Removal Operations

Topological maps are constructed mainly by using removal operations. The i-dimensional removal operation (denoted i-removal) consists in removing an i-cell. This leads to the merging of the two *(i+1)*-cells incident to the removed cell. For 3D subdivisions, we can remove a face (2-removal, e.g. Fig. 2), an edge (1-removal) or a vertex (0-removal). We only present here the main notions about these operations. A more complete description can be found in [17] where

general definitions of removal and contraction operations are provided for any dimension.

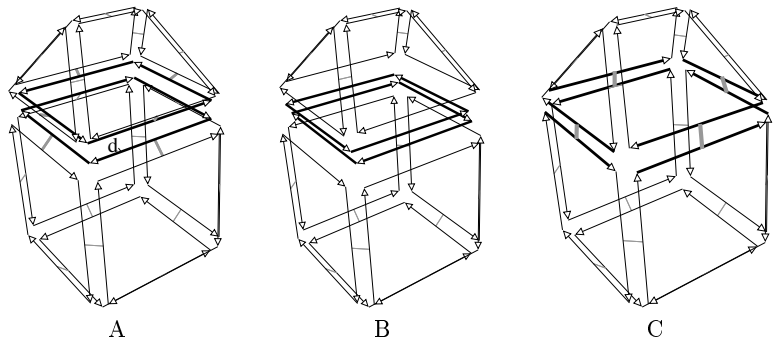


Fig. 2. 2-removal of the face incident to dart d. (A) Initial configuration with two adjacent volumes. (B) The removed face is isolated. (C) Adjacent faces of the initial removed face are joined by modifying β_2 relations.

Any face of a 3-map can be removed without any constraint (e.g. Fig. 2), since the degree of a face, in a 3D subdivision, is always equal to one or two. The face removal operation consists mainly to locally modify the β_2 relation for each dart that belongs to the neighborhood of the removed face (all removal operations are based on similar principle).

The 1-removal (removal of an edge) can be applied only for edges whose degree is one or two. Otherwise it is not possible to automatically decide how to connect the faces incident to the removed edge. This operation is achieved in a similar way than for face removal, but here by modifying β_1 relation. Vertex removal can only be applied for vertices whose degree is one or two. This operation is performed in a similar way than for edge removal, but with different cases to take into account, due to the un-homogeneous definition of combinatorial maps (β_1 is a permutation while others β_i are involutions).

Validity of removal operations can be proved whatever the initial configuration and the cell to remove (even for degenerated cases, as for example removal of a dangling face adjacent to an unique volume, see [17]).

2.3 Topological Map

Combinatorial maps can be used to represent labeled images [18–23, 2, 24], where cells correspond to interpixel or intervoxel elements (pointels, linels, surfels or voxels). For representing 3D labeled images, the main idea of our approach is first to build a complete combinatorial map, that represents all the intervoxel cells of the image, and then to progressively simplify it with removal operations, as long as no topological information is lost. The minimal map obtained by this

construction scheme, called *topological map*, represents all the adjacency and incidence relations between regions of the image.

This is the main property of topological map: to be minimal according to the number of cells, while conserving all the adjacency and incidence relations. To avoid losses of information, we control the operations used during the construction. There are two cases to consider:

- the first case is volume disconnection, when a region is completely included into another one. In this case, we obtain in our model two connected components, one which represents the external surface, and a second which represents the inner surface. We add an inclusion tree on the regions of the image, that allows us to keep relations between these two surfaces[2];
- the second case is face disconnection, when a face has different borders. Here, we add a constraint on the 1-removal operation in order to avoid this type of disconnection. Indeed, this case only occurs when we remove a degree one edge, which is not a dangling edge. By avoiding to remove such an edge, we keep each face connected, and thus homeomorphic to a topological disk. We call *fictive edges* the particular edges kept by this additional constraint, since they do not represent an adjacency relation between regions. By opposition, other edges are called *real edges*. We introduce the notion of *real degree* of a vertex, which is the vertex degree but without considering incident fictive edges.

The topological map construction is made through 5 steps, each one being a simplification of the map obtained by the previous step:

Step 1: Initialization. Given a 3D labeled image, build a 3-map representing a 3D grid made of cubic volumes, plus an enclosing volume which represents the infinite region.

Step 2: Remove each face shared by two voxels having the same label. This step merge volumes that belong to the same region. After this step, each boundary between two regions is represented by a unique surface made of square faces (corresponding to surfels).

Step 3: Remove each degree two edge, and each dangling edge, except isolated edges. This step simplifies the boundaries of each region by merging its faces. We can classify each edge e depending on its degree d:
- $d > 2$: e is not removed due to the precondition of the 1-removal operation. This type of edge belongs to a junction of different boundaries;
- $d = 2$: e is removed because the two incident faces belong to the same boundary and can thus be merged into a unique face. Moreover, this removal can not involves a disconnection since the two faces are different;
- $d = 1$ and e is an isolated edge: this case corresponds to the minimal representation of a sphere with two vertices, one isolated edge and one face.

[2] Nevertheless the problem of interlaced rings is not take into account by inclusion tree but this is a main drawback of all topological structures. This could be eventually avoided by adding fictive faces to keep cells homeomorphic to balls

Thus, e is not removed otherwise we remove a surface that represents an adjacency relation;
- $d = 1$ and e is a dangling edge: e is removed because it not represents an adjacency relation and its removal can not involves a disconnection;
- $d = 1$ and e is not a dangling edge: e is not removed to avoid the disconnection of the face. This is the unique case which involves the creation of a fictive edge.

Step 4: Remove each real degree two vertex incident to two non-loop edges, after shifting all fictive edges incident to this vertex. This step simplifies the boundaries of each region by merging real edges. Since this step and the following are both concerned by the real degree vertex, explanations are merged and presented after the last step.

Step 5: Remove each real degree zero vertex incident to at least two edges, including one non-loop edge, after shifting all fictive edges incident to this vertex, except one non-loop edge. This step simplifies the boundaries of each region by grouping fictive edges on same vertices.

We can classify each vertex v depending on its real degree d. We consider the real degree and not the degree since fictive edges are not take into account during this simplification of boundaries. But they are necessary to keep each face connected and for that, they are shifted (pushed along one incident edge) before the vertex removal. If the real degree d is:
- $d > 2$: v is not removed since it belongs to a junction of different boundaries with more than two real edges;
- $d = 2$: if at least one real edge is a loop, v is not removed since the loop corresponds to a face and thus represents an adjacency information. Otherwise, v is removed since the two incident real edges belong to the same boundary;
- $d = 1$: v is not removed because the real edge is a loop (same reason than the previous case);
- $d = 0$: if v is incident to an unique edge, it is not removed since this is the case of the sphere. If v is only incident to $2k$ loops, v is not removed since this case corresponds to the minimal representation of a torus with k holes. Otherwise, there are at least 2 edges and at least one non-loop edge. In this case, v is removed in order to regroup fictive edges on a same vertex.

We can see a first example in Fig. 3 which shows a 3D image and the corresponding topological map. The second example given in Fig. 4 shows a case where a region R_1 is totally included into another one R_2 without other adjacency regions. In such a case, the representation obtained in topological map corresponds to classical canonical representation of surfaces (in our example we obtain the torus canonical representation with 1 vertex, 2 edges and 1 face).

We have presented here the construction of topological map by successive steps. But in practice, we use an incremental extraction algorithm (presented for example in [25, 2]) which extract topological map in a single scan of the image. The image is scanned from left to right, from behind to front and from

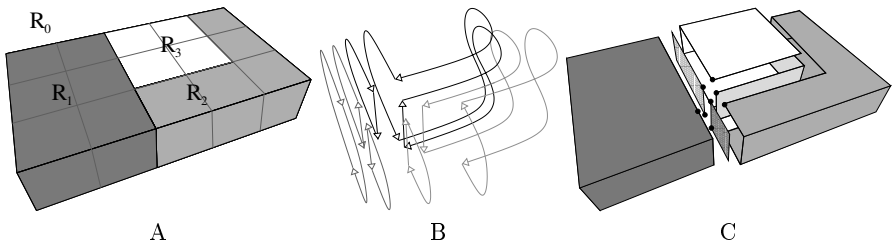


Fig. 3. (A) A 3D image. (B) The corresponding topological map (partial representation without the infinite volume). (C) The represented subdivision in intervoxel elements.

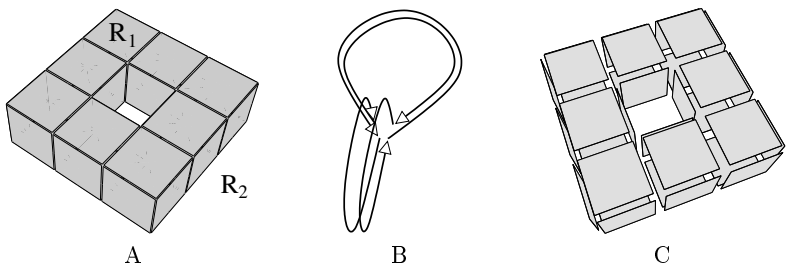


Fig. 4. (A) A region R_1 totally included into another one R_2. This is a partial representation of a 3D image with other regions around R_2 but that are not adjacent to R_1. (B) The corresponding topological map (partial representation without the infinite volume) with one face, two edges and one vertex. (C) The surfels that compose the boundary surface between R_1 and R_2.

up to bottom. For each voxel, a cube is added to the combinatorial map already built. Then, we remove some faces, edges and vertices, depending on the local current configuration.

3 Incremental Euler Characteristic Computation

Euler characteristic χ of each surface of topological map can be computed using the alternated sum of numbers of i-cells, for each $i = 0 \ldots 3$. We propose here to compute incrementally the Euler characteristic during the topological map construction, with only a small additional cost. This can be achieved just by studying the different removal operations and their effect on the number of cells.

The incremental Euler computation given here is only valid for regions represented by a unique surface in topological map. An orientable surface without boundary is completely characterized by χ (this also can be done with its genus). Euler characteristic of a set of surfaces does not give any characteristic information on surfaces. To extend this work in order to obtain topological features for region made of many surfaces, we need to study other characteristics. For

example, we are currently interesting on homology groups and the way they can be computed incrementally by using topological map.

In the following, we note $\#F$ the number of faces, $\#E$ the number of edges, and $\#V$ the number of vertices of a region, and χ the Euler characteristic before each operation, and we use the same notation with the prefix n ($\#nF$, $\#nE$, $\#nV$ and $n\chi$) for the same numbers after the operation.

3.1 Cube Creation

The first step of the incremental extraction algorithm consists in creating a cube and add it to the combinatorial map already built. Thus, we just increase the number of cells of the region that contains this voxel $\#nV = \#V + 8$, $\#nE = \#E + 12$ and $\#nF = \#F + 6$ in order to count all the cells of the new cube. After this step, some cells are eventually counted twice. Moreover we can have temporally an invalid Euler characteristic since it corresponds to several surfaces. But the following simplifications are going to eventually decrease these numbers, depending on the current configuration, and finally, one connected component is re-obtained and thus the valid Euler characteristic.

3.2 2-removal

The face removal is used directly after the cube creation, and uniquely on faces of the new cube in the case of the incremental algorithm. For this reason, faces considered here are only square faces, made of 4 edges. Faces are only removed between volumes that belong to the same region, and thus there are only a region which is concerned by this removal and for which we need to update its topological characteristics.

$\#nF = \#F - 2$, since faces are counted twice in the initial subdivision and the both half faces are removed during the 2-removal (see example in Fig. 5).

$\#nE = \#E - 4$. There are two cases depending on the degree of the edges incident to the removed face. If the degree of each edge is greater than two (e.g. Fig. 5), each edge is counted twice in the initial subdivision and grouped after the removal. Thus, there are 8 edges before, and 4 after which gives the difference -4. The second case is when some edges incident to the removed face are degree two edges (e.g. Fig. 6). In this case, degree one edges are counted only once in the initial subdivision, but are completely removed after the 2-removal operation. For this reason, both cases involve exactly the same evolution on the number of edges.

$\#nV =?$. Concerning the number of vertices, the problem is more complicated. Indeed, there are many different cases, depending on the number of vertices counted twice in the initial subdivision, and depending also on the number vertices that are grouped or not after the face removal. Since the number of cases seems to be too much important, we use the topological map in order to update the number of vertices.

We just count the number of vertices incident to the removed face before its removal, and count again the same number after this removal. The difference

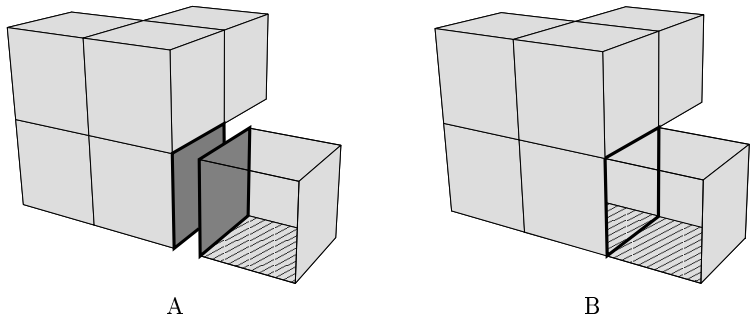


Fig. 5. Face removal where no edge incident to the face is a degree 2 edge. (A) Before the 2-removal The new cube is drawn on the right of the current region (we do not have represented two faces to see the interior of the volume). (B) After the removal: the 2 dark grey faces are removed, and the 8 bold edges are merged into 4 edges.

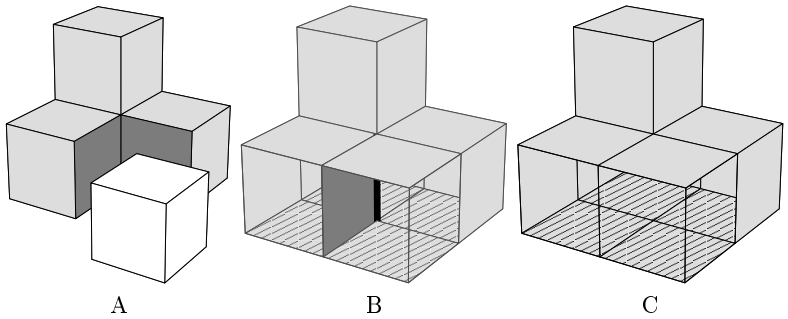


Fig. 6. Face removal where some edges incident to the face are degree 2 edges. (A) Before added the new cube (drawn in white) in a region made of 4 voxels. This addition will be done by 2-removing both dark grey faces. (B) Subdivision obtained after the first 2-removal. 3 faces are not drawn in order to see the interior of the volume. The black thick edge incident to the second face to remove is a degree one edge (i.e. incident twice to the same face). (C) After the second face removal, the degree one edge has completely disappeared.

gives immediately the new number of vertices depending on the old one. Of course, this solution involves a small additional cost. But this cost is very small since we are in a 3D discrete grid and thus we are sure that at most 6 edges are incident to a given vertex.

3.3 1-removal

The third step of the construction of topological map consists in removing each degree two edge, and each dangling edge (except isolated edges). Now, the initial combinatorial map can have different kind of volumes, since we have already

merged some of them during the first step. But the map is already closed and when we process an edge, we are sure that this edge is incident to two volumes. For this reason, number of cells need to be updated in a same way for both regions incident to the removed edge.

When we remove an edge, two possible different cases can be obtained:

- when a degree two edge is removed;
- when a degree one dangling edge is removed.

Other cases are avoided by definition of topological map.

First case: Degree two edge This case is shown in Fig. 7. The new characteristics are: $\#nF = \#F - 1$: two faces are merged into one; $\#nE = \#E - 1$: one edge is removed; $\#nV = \#V$: the number of vertices is still unchanged; and thus $n\chi = \chi$: there are no topological modification.

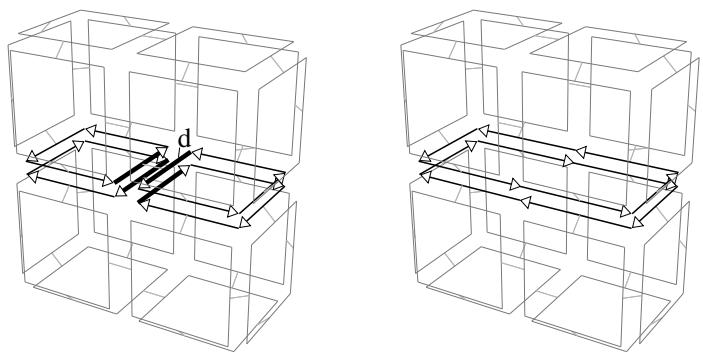


Fig. 7. 1-removal of the degree two edge incident to dart *d*. (A) Before the 1-removal. (B) After.

Second case: Degree one dangling edge This case is shown in Fig. 8. The new characteristics are: $\#nF = \#F$: since the removed edge is inside a face, no face are merged; $\#nE = \#E - 1$: the edge is removed; $\#nV = \#V - 1$: due to the removal of the edge, the degree one vertex incident to the removed edge vanishes; and thus $n\chi = \chi$: the Euler characteristic remains unchanged.

Thus we can conclude that the 1-removal does not change Euler characteristic of concerned regions, whatever the configuration of the removed edge.

3.4 0-removal

The last step of the construction of topological map consists in removing each vertex which is either:

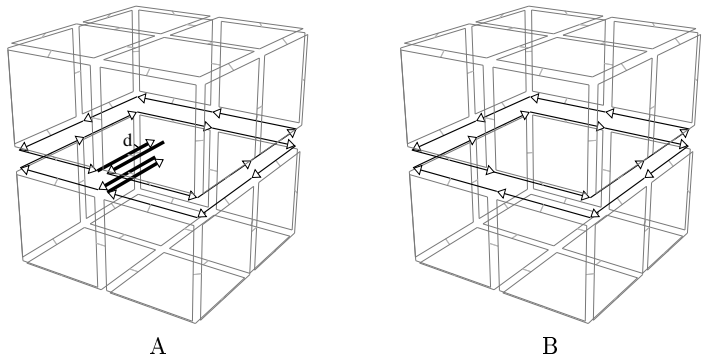


Fig. 8. 1-removal of the degree one dangling edge incident to dart d. (A) Before the 1-removal. (B) After. The removal of the edge involves the disappearance of the degree one vertex.

- a real degree two vertex incident to two non-loop edges, after shifting all fictive edges incident to this vertex;
- a real degree zero vertex incident to at least two edges, including one non-loop edge, after shifting all fictive edges incident to this vertex, except one non-loop edge.

The additional conditions (concerning the non-loop edges) ensure that the 0-removal does not involves the disappearance of a face. Thus, there are only two cases to consider:

- when the degree of the vertex is two;
- when the degree of the vertex is one and the edge is dangling.

Indeed, there are the two unique possible configurations obtained after the fictive edges shifting starting from both cases (given above) of the topological map construction.

Actually, these two cases involve the same modifications: the disappearance of one edge and one vertex. Thus, the new characteristics are: $\#nF = \#F$, $\#nE = \#E - 1$, $\#nV = \#V - 1$ and thus $n\chi = \chi$. We can conclude as for the 1-removal: the 0-removal does not change Euler characteristic of concerned regions.

Note that for 0-removal, there are many regions that are concerned by these modifications: each region which is incident to the removed vertex. Thus, topological characteristics need to be updated for each such regions.

3.5 Fictive Edge Shifting

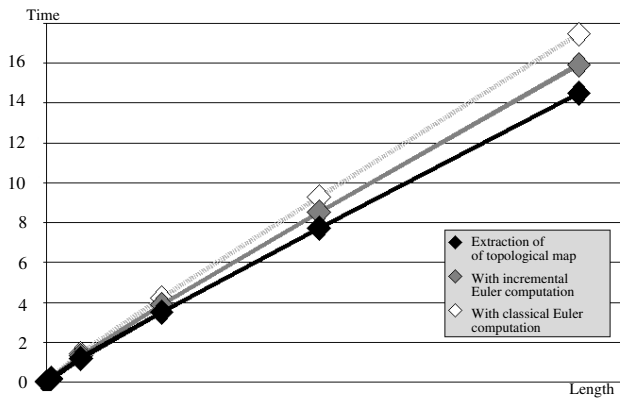
We also need to study the possible evolutions during the fictive edge shifting. This can be done immediately since there is no modification, neither for the number of volumes, nor for faces, edges and vertices. Obviously, the Euler characteristic remains unchanged after this operation.

3.6 Experimentations

We have implemented the incremental Euler characteristic computation in our computer software which computes the topological map incrementally. This program is developed in C++ without particular optimization. All our experiments were made on a classical personal computer with a Athlon 2000MHz CPU and 512Mb of memory and a Linux Debian System.

Our experiments are made on random artificial images in order to be able to test easily many different images. We have generated images of size range $4 \times 4 \times 4$ to $160 \times 160 \times 160$, and for each size we have generated 10 random images in order to compute an average of the obtained results. For each image, the number of generated regions is a random number between 1 and the size of the image.

We compared the time needed to compute Euler characteristic with the classical method (counting the number of cells and compute the altering sum) and with our incremental method. We can see in Fig. 9 the results obtained by our experiments. Moreover, we have also verified that both methods give the same result.



Length	4	8	16	32	64	96	128	160
Extraction	0	0	0,02	0,17	1,2	3,51	7,72	14,49
Incremental	0	0	0,03	0,19	1,33	3,88	8,53	15,93
Classical	0	0	0,02	0,19	1,44	4,22	9,27	17,46

Fig. 9. Time (in seconds) necessary to extract topological map alone (in black in the figure and the first line of the array), with the incremental Euler computation (in dark grey in the figure and the second line of the array), and with the classical algorithm (in white and dash line in the figure and the last line of the array). Each time is the average of 10 extractions for image of size $Length \times Length \times Length$.

We can first observe that the additional cost taken by our incremental Euler characteristic computation is small compared to the time necessary to extract

the topological map alone. Since we do not have optimized our software, this additional time can be reduced by using some programming techniques. Second, we can observe that our incremental algorithm is faster than the classical algorithm (about 10%), what shows the interest to use the incremental solution.

4 Conclusion and Perspectives

In this paper, we show how to compute Euler Characteristic "on the fly" during the topological map construction. This computation is efficient since topological map construction is efficient and only a small additional cost is needed to compute Euler Characteristic.

The proposed algorithm is incremental as it uses computations from one step to determine the result for the next step. Our experiments show that the additional time necessary to compute incrementally Euler characteristic is very small. Moreover, this solution is more efficient than the classical algorithm which consists in counting the number of cells of the final subdivision and using the altering sum.

This first result is interesting since we are able to compute efficiently during the topological map construction, a topological feature. Moreover, this computation can be optimized, either by using some programming techniques, or by studying the different cases that can occur during face removal in order to remove the counting of vertices. For that, it could be possible to study directly the possible cases by considering directly the added voxel and not only face removal.

Now, we are working on the computation, in a similar way, of other topological features. We have first results for the canonical polygonal schema computation. Intuitively, a canonical polygonal schema of a given surface is the minimal polygon such that when each edges are correctly identified two by two, we obtain the initial surface. We have shown that this notion can be directly obtained from topological map (for regions that are composed by a unique face since the polygonal schema is not defined for other cases).

The next step is to extend computations of topological features for higher dimensions in order to be able to characterize any type of regions and not only those made of an unique surface (which is the case for Euler characteristic). For that, we are interesting on the calculation of homology groups and generators of these groups.

Homology groups are topological invariants that deal with holes in a topological spaces. These invariants can be computed into each dimension and concrete interpretation can be given for low dimensions. In dimension 0, homology groups characterizes connected components, in dimension 1, homology groups characterizes holes, and cavities are described by dimension 2 homology groups[3].

[3] roughly speaking for each dimension p, the p-th homology group H_p is isomorphic to a direct sum $\underbrace{\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}}_{\beta_p} \oplus \mathbb{Z}/t_1^p\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/t_n^p\mathbb{Z}$. where β_p is called the *p-th Betti number*, the integers $t_1^p, \ldots, t_n^p$ are called the *torsion coefficients* of H_p.

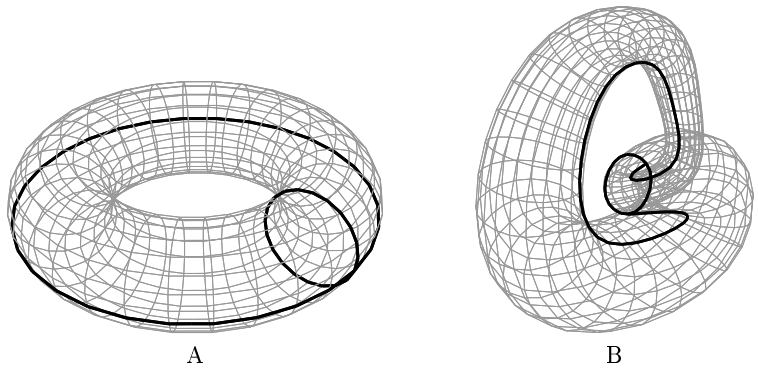


Fig. 10. Illustration of homology generators. (A) For a torus, two paths that surround the two holes are highlighted. (B) For a Klein bottle, two paths are highlighted, one surround the hole and the other indicates the torsion part of the Klein bottle.

In order to represent homological informations directly on the image, an algorithm that computes generators of homology groups can be used [26]. Such generators are cycles (i.e. closed paths) that surround holes, see examples given on figure 10. Our goal is now to compute these generators incrementally during topological map construction.

References

1. Bertrand, Y., Damiand, G., Fiorio, C.: Topological map: Minimal encoding of 3d segmented images. In: Workshop on Graph-Based Representations in Pattern Recognition, Ischia, Italy, IAPR-TC15 (may 2001) 64–73
2. Damiand, G.: Définition et étude d'un modèle topologique minimal de représentation d'images 2D et 3D. Thèse de doctorat, Université Montpellier II (décembre 2001)
3. Hatcher, A.: Algebraic Topology. Cambridge University Press (2002) available on http://www.math.cornell.edu/∼hatcher/AT/ATpage.html.
4. Agoston, M.K.: Algebraic Topology, a first course. Pure and applied mathematics. Marcel Dekker Ed. (1976)
5. Klette, R., Rosenfeld, A.: Digital Geometry - Geometrics Methods for Digital Pictures Analysis. Morgan Kaufmann, San Francisco (2004)
6. Lee, C., Poston, T., Rosenfeld, A.: Holes and genus of 2D and 3D digital images. CVGIP: Graphical Models and Image Processing **55**(1) (January 1993) 20–47
7. Imiya, A., Eckhardt, U.: The Euler Characteristics of discrete objects and discrete quasi-objects. Computer Vision and Image Understanding **75**(3) (September 1999) 307–318
8. Desbarats, P., Domenger, J.P.: Retrieving and using topological characteristics from 3d discrete images. In: Proceedings of the Computer Vision Winter Workshop. (February 2002) 130–139
9. Brimkov, V., Maimone, A., Nordo, G.: An explicit formula for the number of tunnels in digital objects. ArXiv Computer Science e-prints (May 2005)

10. Spehner, J.: Merging in maps and in pavings. Theoretical Computer Science **86**(2) (September 1991) 205–232
11. Edmonds, J.: A combinatorial representation for polyhedral surfaces. Notices of the American Mathematical Society **7** (1960)
12. Tutte, W.: A census of planar maps. Canad. J. Math. **15** (1963) 249–271
13. Jacques, A.: Constellations et graphes topologiques. In: Combinatorial Theory and Applications. Volume 2. (1970) 657–673
14. Cori, R.: Un code pour les graphes planaires et ses applications. PhD thesis, UniversitÃľ Paris VII (1973)
15. Cori, R.: Un code pour les graphes planaires et ses applications. In: Astérisque. Volume 27. Soc. Math. de France, Paris, France (1975)
16. Lienhardt, P.: Topological models for boundary representation: a comparison with n-dimensional generalized maps. Commputer Aided Design **23**(1) (1991)
17. 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 (november 2003) 408–419
18. Fiorio, C.: A topologically consistent representation for image analysis: the frontiers topological graph. In: Discrete Geometry for Computer Imagery. Number 1176 in Lecture Notes in Computer Science, Lyon, France (november 1996) 151–162
19. Brun, L., Domenger, J.P., Braquelaire, J.P.: Discrete maps : a framework for region segmentation algorithms. In: Workshop on Graph-Based Representations in Pattern Recognition, Lyon, IAPR-TC15 (april 1997) published in Advances in Computing (Springer).
20. Paillonc y, J.G., Jolion, J.M.: The frontier-region graph. In: Workshop on Graph-Based Representations in Pattern Recognition. Volume 12 of Computing Supplementum., Springer (april 1997) 123–134
21. Braquelaire, J.P., Brun, L.: Image segmentation with topological maps and inter-pixel representation. Journal of Visual Communication and Image Representation **9**(1) (march 1998) 62–79
22. Braquelaire, J.P., Desbarats, P., Domenger, J.P., Wüthrich, C.: A topological structuring for aggregates of 3d discrete objects. In: Workshop on Graph-Based Representations in Pattern Recognition, Austria, IAPR-TC15 (may 1999) 193–202
23. Bertrand, Y., Fiorio, C., Pennaneach, Y.: Border map: a topological representation for nd image analysis. In: Discrete Geometry for Computer Imagery. Number 1568 in Lecture Notes in Computer Science, Marne-la-Vallée, France (1999) 242–257
24. Damiand, G., Bertrand, Y., Fiorio, C.: Topological model for two-dimensional image representation: definition and optimal extraction algorithm. Computer Vision and Image Understanding **93**(2) (february 2004) 111–154
25. Bertrand, Y., Damiand, G., Fiorio, C.: Topological encoding of 3d segmented images. In: Discrete Geometry for Computer Imagery. Number 1953 in Lecture Notes in Computer Science, Uppsala, Sweden (december 2000) 311–324
26. Peltier, S., Alayrangues, S., Fuchs, L., Lachaud, J.O.: Computation of homology groups and generators. In: Discrete Geometry for Computer Imagery. Volume 3429 of LNCS., Springer (2005) 195–205