Region Merging with Topological Control

Alexandre Dupas, Guillaume Damiand

Region Merging with Topological Control

Alexandre Dupas [a] Guillaume Damiand [b]

[a] *Université de Poitiers, CNRS,*
XLIM-SIC, UMR6172, F-86962 Futuroscope, France

[b] *Université Lyon, CNRS,*
LIRIS, UMR5205, Université Lyon 1, F-69622, Villeurbanne, France

Abstract

This paper presents a region merging process controlled by topological features on regions in 3D images. Betti numbers, a well-known topological invariant, are used as criteria. Classical and incremental algorithms to compute Betti numbers using information represented by the topological map of an image are provided. The region merging algorithm, which merges of any number of connected components of regions together, is explained. A topological control of the merging process is implemented using Betti numbers to control the topology of an evolving 3D image partition. The interest of incremental approaches of Betti numbers computation is established by providing processing times comparison. A visual example showing the result of the algorithm and the impact of topological control is also given.

Key words: Topological constraint, Betti numbers, Region merging, Topological map, Image processing

1 Introduction

In this paper, a basic image processing operation is introduced: the region merging. This application uses a combinatorial model known as topological map that represents cells and relations between cells to describe the partition of the image into regions. A topological control of the merging process is allowed using criterion on Betti numbers for regions.

⋆ Paper published in Discrete Applied Mathematics Volume 157, Issue 16, 28 August 2009, Pages 3435-3446. Thanks to Elsevier. The original publication is available at http://dx.doi.org/10.1016/j.dam.2009.04.005.

Email addresses: `dupas@sic.univ-poitiers.fr` (Alexandre Dupas), `guillaume.damiand@liris.cnrs.fr` (Guillaume Damiand).

In image processing, many works have proposed methods to control the topology of a partition of the image. Specific constraints, known from the field of application, have to be enforced in order to provide realistic results. For instance, in medical application, several works are interested in the brain topological structure. One research subject is the extraction of a topologically correct cortical surface[1] (surface of the brain) or the correction of the topology of a previously extracted surface [2]. The cortical surface has the same topology as a sphere. Other applications involve the segmentation of blood vessels or the segmentation of trabecular bone. For this last application, [3] defines a classification of voxels based on the topology of the neighborhood of the voxel that allows to construct a topologically correct surface for the studied structure.

Most of the current approaches regarding topological control run into mainly two issues:

(1) they work on binary images to construct a binary partition of the space such as the surface between foreground and background objects has some particular topology. Some works are able to control topological features on multiple regions. For instance, [4] proposes a framework using a deformable model to segment brain structures but under a strong hypothesis allowing to convert the 4 class segmentation problem into a binary one. From the authors own words, the real topology of brain structures does not allow such hypothesis;
(2) they are interested in the control of topological features for surfaces and not volumes. The goal of either the topological control or the topological correction is often to obtain a 2D surface that has a particular topological feature. The main criterion on surface is the genus, a topological feature that allows to distinguish sphere surfaces from k-torus surfaces.

To overcome these issues, we propose to:

(1) work on sets of regions to represent any image partition;
(2) use Betti numbers: a topological feature allowing to characterize regions in 3D.

Betti numbers are a well-known topological feature [5]. They are invariant for homology groups and thus characterize some topological properties of the described object. For regions in 3D images, Betti numbers count connected components, tunnels and cavities. Several works have studied how to compute Betti numbers, in particular with an incremental approach using cellular complexes [6]. Betti numbers are also used in segmentation applications: in [7] the authors define a segmentation process on 2D images. In this work, the topological control uses Betti numbers to put constraint on the geometric active contour implemented by a traditional level set method. But in our knowledge, there is no work that proposes 3D split and merge segmentation method with

topological control.

Many works have studied models representing partitions of an image. Topological data structures describe images as a set of elements and their adjacency relations. The most famous example is the Region Adjacency Graph (RAG) [8] which represents each region by a vertex, and where neighboring regions are connected by an edge. But the RAG suffers from several drawbacks as it does not represent multi-adjacencies or makes no difference between inclusion and adjacency relations. To solve these issues, the RAG model has been extended, for instance, in dual-graph structures to represent 2D images [9] or in topological maps [10–15] used to represent 2D and 3D images. Topological maps represent all the cells of the subdivision contrary to graph-based approaches. Topological maps also have the advantage of being defined in 3D. Thus, topological maps are used in this work to represent the 3D image partition.

The aim of this work is to provide a topological control during modification operations on the topological maps. Region merging is one of the classical modification operation in image processing. It allows to merge at least two adjacent regions together in order to create the resulting region which is the union of the merged regions [16]. Region merging is a very common operation used for example in segmentation process. In this paper, the topological control of the partition produced during a merging process is developed using Betti numbers as a topological criterion.

The paper is organized as follow. Section 2 presents topological maps, the model used to represent 3D images in this work. In Section 3, Betti numbers are introduced and their computation using cellular models like topological maps is given. Section 4 explains the algorithm used to merge regions in topological maps. An experimental criterion using Betti numbers to control region merging is detailed in Section 5 and some processing times and one visual example are given. Finally, Section 6 concludes the paper and gives some perspectives for this work.

2 3D Topological Maps

3D topological maps are an extension of combinatorial maps used to represent 3D image partitions. Notions on combinatorial maps, 3D images, intervoxel elements and topological maps used in this work are presented in the following.

A combinatorial map [17] is a mathematical model describing the subdivision of a space, based on planar maps. A combinatorial map encodes all the cells of the subdivision and all the incidence and adjacency relations between the

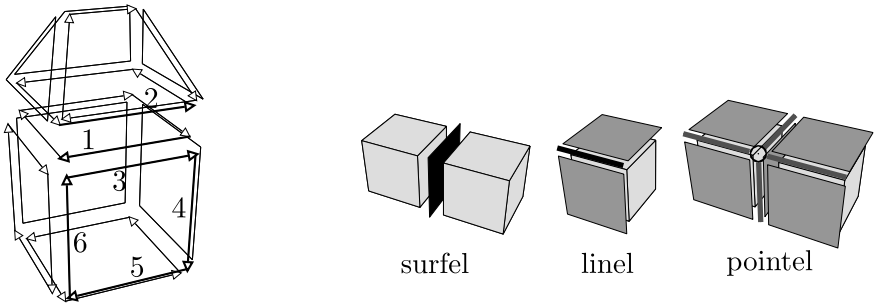


Fig. 1. (a) Example of combinatorial map representing two volumes. Arrows represent darts. Adjacency and incidence relations are deduced from the geometry. In this example, $\beta_3(1) = 2$, $\beta_2(1) = 3$ and $\beta_1(3) = 4$. (b) Intervoxel elements drawn in black are from left to right: one surfel, one linel and one pointel.

different cells, and so, describes the topology of the space. Definition 1 presents the formal definition of the combinatorial map in 3D.

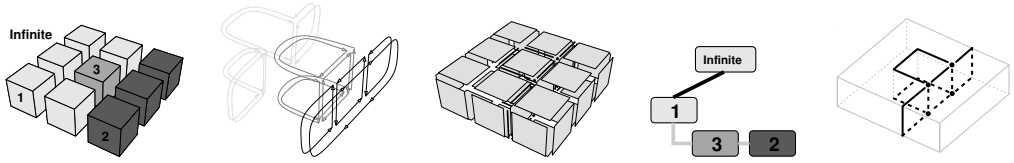
Definition 1 *A 3D combinatorial map, (or* 3*-map) is a* 4*-tuple* $M = (D, \beta_1, \beta_2, \beta_3)$ *where:*

(1) D *is a finite set of darts;*
(2) β_1 *is a permutation*[1] *on* D*;*
(3) β_2 *and* β_3 *are two involutions*[2] *on* D*;*
(4) $\beta_1 \circ \beta_3$ *is an involution.*

The single basic elements used in the definition of combinatorial maps are called *darts*, and adjacency relations are defined onto darts. β_i is a relation between two darts that describes an adjacency between two i-dimensional cells also called i-cells (see Figure 1a as an example of combinatorial map and [18] for more details on maps and comparisons with other combinatorial models). Intuitively, with this model, the notion of cells is represented by a set of darts linked by specific β_i relations. For example, a face incident to a dart d is represented by the set of darts accessible using any combination of β_1 and β_3 relations. Moreover, given a dart d, which belongs to the i-cell c, the i-cell adjacent to c along the $(i-1)$-cell which contains d is found using $\beta_i(d)$. For example, given a dart d that belongs to a face f and a volume v, the volume adjacent to v along f is the 3-cell containing $\beta_3(d)$: in Figure 1a, darts belonging to the same volume and the same face than 3 are 4, 5 and 6.

Few usual notions about images and intervoxel elements are now introduced. A voxel is a point of the discrete space $\mathbb{Z}^3$ associated to a value which could be a color or a gray level. A three dimensional image is a finite set of voxels. In this work, combinatorial maps are used to represent sets of voxels that have the

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



(a) (b) (c) (d) (e)

Fig. 2. The different parts of the topological map used to represent an image. (a) 3D image. (b) Minimal combinatorial map. (c) Intervoxel matrix (embedding). (d) Inclusion tree of regions. (e) Cellular representation of the same topological map.

same label and are 2-connected [3]. The label of a voxel is given by a labeling function $l : \mathbb{Z}^3 \rightarrow L$ that associates a label (a value in the finite set L) to each voxel. A maximal set of 2-connected voxels with the same label is called *region*. Let I be an image, the complement of a region r in I is denoted r^C. It is composed of voxels of I that do not belong to r. Since r is 2-connected, r^C is formed of 1-connected components of voxels. To avoid particular processes for the voxels that belong to the border of the image, an infinite region, usually called r_0, surrounds the image. Using the previous notation, $r_0^C = I$. A region r_j is *included* in r_i if each 1-connected set of voxels that contains at least one voxel from r_j and one voxel from r_0 contains at least one voxel of r_i (i.e. the paths from r_j to r_0 pass through r_1). Note that by using this definition, each region in I is included in the infinite region.

In the intervoxel framework [19], an image is considered as a subdivision of a 3-dimensional space in a set of unit elements: voxels are the cubes, surfels are the squares between two voxels, linels are the segments between surfels, and pointels are the points between segments (see example in Figure 1b).

The *topological map* is a data structure used to represent the subdivision of an image into regions. It is composed of three parts:

- a minimal combinatorial map representing the topology of the image;
- an intervoxel matrix used to retrieve geometrical information associated to the combinatorial map. The intervoxel matrix is called the *embedding* of the combinatorial map;
- an inclusion tree of regions.

Figure 2 presents an example of a topological map. The 3D image, composed of three regions plus the infinite region (Figure 2a), is represented by the topological map which is divided in three parts: the minimal combinatorial map Figure 2b, the embedding Figure 2c, and the inclusion tree of regions Figure 2d. Figure 2e represents the same topological map using a different formalism, called cellular decomposition. The geometry is drawn with light

[3] Two voxels are 2-connected if they are adjacent by a face. Two voxels adjacent by a face or by an edge are 1-connected. Two voxels adjacent by a face, an edge or a vertex are 0-connected.

gray lines. Vertices and edges of the model are drawn in black. Faces are implicit in this representation. Dashed lines represent elements usually hidden by the geometry. This representation is used to simplify figures in the following explanations.

In the topological map framework, the combinatorial map is used as a topological representation of the partition of an image in regions. Each face of the topological map separates two adjacent regions and two adjacent faces do not separate the same two regions. With these rules, the minimality in number of cells of the topological map is guaranteed (see [10,20] for more details on topological maps). The intervoxel matrix is the embedding of the combinatorial map. Each cell of the map is associated to intervoxel elements representing the geometrical information of the cell. A face, in the combinatorial map, is embedded by a set of surfels separating voxels of the two incident regions. The edges, which are the border of faces, are represented by a set of linels. The vertices, which are the border of edges, are embedded by pointels. The intervoxel matrix allows to retrieve the geometry of the labeled image represented by the combinatorial map. The inclusion tree of regions represents inclusion relations. Each region in the topological map is associated to a node in the inclusion tree. Nodes are linked together by the inclusion relation previously defined. In the inclusion tree, regions are grouped by connected components using one representative region of the component called *direct son*. Note that counting direct sons included in a region r gives the number of 1-connected components of regions included in r, i.e. gives the number of cavities of r.

External and *internal* surfaces are defined using the inclusion relation. For each cavity, there is an internal surface: the border between the including region and the included regions. For each region, there is one external surface: the border between the region and regions that are at the same level of inclusion (regions of the same connected component) or the including region.

Each dart d knows its belonging region (noted $region(d)$). Each region knows a *representative dart* (noted $rep(r)$). By definition, $rep(r)$ belongs to the external surface of r and its other incident region $r' = region(\beta_3(rep(r)))$, is a smaller region than r considering the sweeping order of the image voxels (i.e. r' is found before r when we run through the image with a scan line algorithm). It defines a full order on regions using the position of their first voxels. Let r_1 and r_2 be two adjacent regions such that $r_1 < r_2$: either r_2 is included in r_1, or r_1 and r_2 belong to the same connected component of regions.

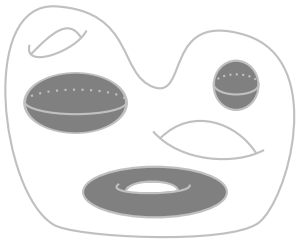


Fig. 3. Representation of a 3D object having one connected component, three tunnels and three cavities (drawn in gray). Note that one of the cavities is a torus and thus, is responsible for one of the tunnels.

3 Betti Numbers Computation

Betti numbers are very useful basic topological invariants. From a practical point of view for a 3D object, the Betti numbers represent the number of *holes* in each dimension. The first Betti number, noted b_0, counts connected components of the object. The second Betti number, b_1 counts tunnels, sometimes called handles. The third Betti number, b_2 counts the number of cavities also called voids. For closed oriented 3D objects, as regions in a 3D image, Betti numbers b_k with $k > 2$ are equal to zero. For instance, the three first Betti numbers of the 3D object presented in Figure 3 are $b_0 = 1$, $b_1 = 3$ and $b_2 = 3$.

Lemma 2 links the number of surfaces of a region to the Betti numbers of this region.

Lemma 2 *The number of surfaces $\#s(r)$ of a region r is given by the sum of the number of connected components and cavities of r: using Betti numbers, $\#s(r) = 1 + b_2(r)$.*

PROOF. The number of surfaces of a region is the sum of external surfaces and internal surfaces of that region. A region r in the topological map is a 2-connected set of voxels. Voxels not in r form 1-connected sets of voxels known as r^C. In these sets of voxels, one connected component of voxels includes r and the others are included in r. The surface splitting voxels of the region from voxels of the including connected component form the external surface. Since there is only one 2-connected component of voxels, there is only one external surface. For each 1-connected component of voxels included in a region r, there is one surface that surrounds the voxels and creates a cavity into r. There is one internal surface for each cavity. There is no other surface since if such a surface exists, the surface would at least surround a voxel and that means there would be a cavity in the region. □

3.1 Related definitions

The Betti numbers are related to the Euler characteristic, another topological invariant, of a region by Definition 3.

Definition 3 *The Euler characteristic $\chi(r)$ of a region r, considered as a cellular complex, is defined as the alternating sum of Betti numbers: $\chi(r) = b_0(r) - b_1(r) + b_2(r)$ (see [5]).*

Using the cellular decomposition of a region, Definition 4 gives another way to compute the Euler characteristic using the alternating sum of number of cells for each dimension.

Definition 4 *Using the cellular decomposition of a region, the Euler characteristic χ is also given by $\chi = k_0 - k_1 + k_2 - k_3$, where k_i denotes the number of cells of dimension i (see [5]).*

In [21] the authors define an incremental algorithm to compute the Euler characteristic of the region border (i.e. by considering only OD, 1D and 2D cells that belong to the border of the region). Let χ' be the Euler characteristic of the border of a region. It is obtained using Definition 4 with cells belonging to the border of the region. Definition 5 presents the way to compute χ'.

Definition 5 *Let $\chi'(r)$ be the Euler characteristic of the border of region r (the border of a region is composed of surfaces). Let $\#v(r)$, $\#e(r)$ and $\#f(r)$ be the number of vertices, edges and faces belonging to the border of r, then $\chi'(r) = \#v(r) - \#e(r) + \#f(r)$.*

3.2 Computation of Betti numbers using Topological Maps: First Algorithm

A first approach to the computation of the Betti numbers in the topological map framework is to use information provided by the model to obtain these values. The effective computation of homology group generators is avoided since only the rank of homology groups is required. The goal is to compute the Betti numbers by counting connected components, tunnels and cavities. In the following, formulas linking together the Betti numbers of a region and some features easy to compute using a topological map are presented. The algorithm is not given because it is straightforward: it consists in running through the map and counting the different numbers required in the formulas.

3.2.1 First and Third Betti Numbers

The first Betti number, b_0, counts connected components of a region. By definition in topological maps, a region is a 2-connected set of voxels and thus there is only one connected component for each region. The first Betti number $b_0(r)$ of a region r in a topological map M is always equal to one: $\forall r \in M, b_0(r) = 1$.

The third Betti number, b_2, counts the cavities of a region. In topological maps, the inclusion tree represents inclusion relations between regions: for each 1-connected component of included regions, there is a single cavity. As seen in Section 2, regions are grouped in the tree structure by connected components which are represented by direct sons. The number of cavities $b_2(r)$ of a region r is directly obtained by counting direct sons of r.

3.2.2 Second Betti Number

The second Betti number, b_1, counts the tunnels of a region. We have b_0 and b_2 easily and Definition 3 gives the relation between the Betti numbers and the Euler characteristic. From [21] we know how to compute the Euler characteristic $\chi'(r)$ of the border of r. In the following, the relation between these three elements and $b_1(r)$ is established.

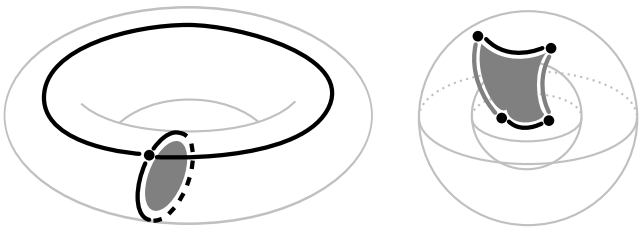
Firstly, implicit cells are defined in order to compute $\chi(r)$ from $\chi'(r)$. Definition 3 and Definition 4 suppose that the region is represented by a cellular complex only composed of n-cells homeomorphic to n-balls. Actually, topological maps represent cells that belong to the border of regions and thus regions are not cellular complexes. Some cells, required to obtain such property, are missing. To overcome this issue, implicit cells are introduced. Implicit cells depend on the number of tunnels and cavities as seen in Definition 6 and allow to obtain a cellular complex representation of a region (in which each i-cell is homeomorphic to an i-ball).

Definition 6 *Implicit cells are defined using topological information by the following two rules:*

- *for each tunnel, one implicit face is added to obtain a volume homeomorphic to a 3-ball.(Figure 4a);*
- *for each cavity, two implicit edges (being the new face border) and one implicit face are added to obtain a volume homeomorphic to a 3-ball.(Figure 4b).*

Proposition 7 gives the link between $\chi(r)$ and $\chi'(r)$.

Proposition 7 *For a region r represented in a topological map,* $\chi(r) = \chi'(r)/2$.



(a) (b)

Fig. 4. Implicit cells in topological maps. Vertices and edges represented by topological maps are drawn in *black*. For both cases, dark gray elements have been added: they are implicit cells required to have a connected polyhedral decomposition of the region and thus to be able to use the polyhedral formula to compute the Euler characteristic. (a) Tunnels: one face for each tunnel is added. (b) Cavities: two edges and one face for each cavity are added.

PROOF. Let k_i be the number of cells of dimension i in the cellular decomposition of r. By Definition 4, the Euler characteristic is given by $\chi(r) = k_0 - k_1 + k_2 - k_3$. Let $\#v(r)$, $\#e(r)$ and $\#f(r)$ be the number of vertices, edges and faces belonging to the border of r. By Definition 5 the Euler characteristic of the border of r is given by $\chi'(r) = \#v(r) - \#e(r) + \#f(r)$. By Definition 6, number k_i of i-cells is given by $k_0 = \#v(r)$, $k_1 = \#e(r) + 2b_2(r)$ (two edges are added for each cavity) and $k_2 = \#f(r) + b_1(r) + b_2(r)$ (one face is added for each tunnel and one for each cavity). k_3 is equal to $b_0(r)$ since there is a volume for each connected component: $k_3 = b_0(r) = 1$. The Euler characteristic is now given by:

$$\begin{aligned}
\chi(r) &= k_0 - k_1 + k_2 - k_3 \\
\chi(r) &= \#v(r) - (\#e(r) + 2b_2(r)) + (\#f(r) + b_1(r) + b_2(r)) - b_0(r) \\
\chi(r) &= \#v(r) - \#e(r) + \#f(r) - b_0(r) + b_1(r) - b_2(r) \\
\chi(r) &= \chi'(r) - \chi(r).
\end{aligned}$$

As an immediate consequence, $\chi(r) = \chi'(r)/2$. □

Now that Proposition 7 has been introduced, $b_1(r)$ is computed using the Euler characteristic of the border of r with respect to the formula presented in Proposition 8: it gives the number of tunnels of a region.

Proposition 8 *The second Betti number $b_1(r)$ of a region r is given by $b_1(r) = b_0(r) + b_2(r) - \chi'(r)/2$.*

PROOF. By Proposition 7 and Definition 3, b_1 is given by:

$$\begin{aligned}
\chi(r) &= \chi'(r)/2 \\
b_0(r) - b_1(r) + b_2(r) &= \chi'(r)/2 \\
b_1(r) &= b_0(r) + b_2(r) - \chi'(r)/2
\end{aligned}$$ □

3.3 Incremental Methods: Second Algorithm

The idea of this second approach is to compute the Betti numbers during a region merging process using values previously computed for the two regions merged. $b_0(r)$ being constant for each region r of a topological map, an incremental method is not needed.

3.3.1 Third Betti Number b_2

The incremental computation of the number of cavities consists in finding changes in number of cavities when merging two regions r_1 and r_2. Three configurations are possible. First, if r_2 fills one cavity of r_1 then merging the two regions leads to the removal of one cavity. Since regions have only one connected component, r_2 cannot fill more than one cavity. The second case occurs if the union of the two regions includes other regions. Depending on the configuration, one or more cavities are created: one by connected component of newly included regions. The last configuration happens if no cavity is filled nor created: there is no change in the number of cavities.

The idea of this algorithm lies on the link between the number of surfaces and the number of cavities presented in Lemma 2. Proposition 9 gives the relation between $b_2(r)$ to the number of surfaces of a region r. The incremental computation of b_2 computes the new number of surfaces when considering the union of two regions.

Proposition 9 *Let $\#s(r_1 \cup r_2)$ be the number of surface of the union of two regions r_1 and r_2. Number of cavities $b_2(r_1 \cup r_2)$ of this union is given by $b_2(r_1 \cup r_2) = \#s(r_1 \cup r_2) - 1$.*

PROOF. Direct by Lemma 2. □

The initialization part of the algorithm computes for each region the number of surfaces. Therefore, the number of cavities is obtained by using the non-incremental algorithm. The number of surfaces is computed using the relation given by Lemma 2. The number of surfaces is stored and updated during the merging process.

Suppose $r_1 < r_2$ in the order of regions defined in Section 3 when computing b_2 for $r_1 \cup r_2$. To compute $\#s(r_1 \cup r_2)$, the algorithm runs trough darts starting from the external surface of r_2. Let k be the number of connected components of darts of $r_1 \cup r_2$ ignoring the darts belonging to inner faces (i.e. darts such as d belongs to r_2 and $\beta_3(d)$ belongs to r_1). Each connected component of darts

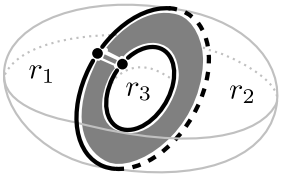


Fig. 5. Configuration with three regions r_1, r_2 and r_3 where r_1 and r_2 surround r_3. The merging of r_1 and r_2 leads to the inclusion of r_3. There is a cavity creation. $\#s(r_1) = \#s(r_2) = \#s(r_3) = 1$. $b_2(r_1) = b_2(r_2) = b_2(r_3) = 0$. $\#s(r_1 \cup r_2) = 2$. $b_2(r_1 \cup r_2) = 1$.

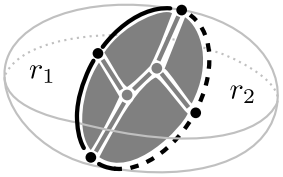


Fig. 6. Inner cells (drawn in dark gray) are cells that fully belong to the surface between r_1 and r_2. Black cells belonging to the border of the surface also belong to the border of the union of the two regions and thus, are not inner cells. In this example, there are 2 inner vertices, 5 inner edges and 4 inner faces.

represents a distinct surface of $r_1 \cup r_2$. The number of new surfaces is $k - 2$ since the k surfaces replace two previously counted surfaces: one surface for r_1 and the external surface of r_2. The number of surfaces of the union of r_1 and r_2 is given by $\#s(r_1 \cup r_2) = \#s(r_1) + \#s(r_2) + k - 2$. Using Proposition 9, $b_2(r_1 \cup r_2)$ is given by $b_2(r_1 \cup r_2) = \#s(r_1) + \#s(r_2) + k - 3$.

Figure 3.3.1 presents an example of a classic configuration creating a new inclusion. There are three regions in the figure: two regions r_1 and r_2 surround r_3. If r_1 and r_2 are merged, r_3 becomes included. The two inner cells (one face and one edge), drawn in dark gray, cannot be passed through when counting connected component of darts. Thus, the surface discovery process finds two surfaces on $r_1 \cup r_2$: k is equal to 2. One of the new surface is the external surface of $r_1 \cup r_2$ and the other one is the internal surface corresponding to the new cavity. This last surface also corresponds to the external surface of r_3. These two surfaces replace the external surface of r_1 and the external surface of r_2, and thus the number of surfaces of the union of r_1 and r_2 is given by $\#s(r_1 \cup r_2) = \#s(r_1) + \#s(r_2) + k - 2 = \#s(r_1) + \#s(r_2) + 2 - 2 = \#s(r_1) + \#s(r_2)$. Using the formula, the resulting number of cavities is given by $b_2(r_1 \cup r_2) = \#s(r_1 \cup r_2) - 1 = \#s(r_1) + \#s(r_2) - 1 = 1$.

3.3.2 Second Betti Number b_1

As seen in Proposition 8 the second Betti number $b_1(r)$ is computed using $b_0(r)$, $b_2(r)$ and the Euler characteristic $\chi'(r)$ of the border of r. To compute b_1 the idea is to compute each part of the formula using an incremental algorithm. Incremental algorithms for $b_0(r)$ and $b_2(r)$ have already been given. This section explains the incremental computation of χ' for the union of two regions r_1 and r_2.

For each couple (r_1, r_2) of adjacent regions, there is at least one surface be-

tween them (in case of multiple adjacencies, there are several surfaces). These surfaces might be composed of many cells. From these cells, let us qualify some of them as inner. Inner cells are cells that only belong to an inner surface. Cells that belong to the border of these surfaces are not considered (called outer cells as all the other cells). Note 1 details why these cells are not considered during the incremental computation. For instance, Figure 3.3.1 presents two regions, r_1 and r_2, simply adjacent. Only one surface lies between the two regions, but there are several cells that compose this surface. Actually, there are 2 inner vertices, 5 inner edges and 4 inner faces for $r_1 \cup r_2$. Let $in(r_1 \cup r_2)$ be the set of inner cells for $r_1 \cup r_2$ and $\chi'(in(r_1 \cup r_2))$ be the Euler characteristic of inner surfaces of $r_1 \cup r_2$. Proposition 10 gives an incremental definition of χ'.

Proposition 10 $\chi'(r_1 \cup r_2) = \chi'(r_1) + \chi'(r_2) - 2\chi'(in(r_1 \cup r_2))$

PROOF. Since inner and outer cells are defined for the union of two regions, the notation $\#c_{out}(r_1 \cup r_2 | r_1)$ is introduced where c is the type of counted cells (v for vertices, e for edges and f for faces). It stands for the number of outer cells for $r_1 \cup r_2$ restricted to cells that belong to r_1. With this notation, and by decomposition of the number of cells between inner cells (in) and outer cells (out) and using Definition 5, the value of $\chi'(r_1)$ is given by:

$$\begin{aligned}\chi'(r_1) =& \#v(r_1) - \#e(r_1) + \#f(r_1)\\ \chi'(r_1) =& + \#v_{in}(r_1 \cup r_2) + \#v_{out}(r_1 \cup r_2 | r_1) - \#e_{in}(r_1 \cup r_2)\\ & - \#e_{out}(r_1 \cup r_2 | r_1) + \#f_{in}(r_1 \cup r_2) + \#f_{out}(r_1 \cup r_2 | r_1).\end{aligned}$$

Using the same process for r_2, $\chi'(r_2)$ is defined by:

$$\begin{aligned}\chi'(r_2) =& \#v(r_2) - \#e(r_2) + \#f(r_2)\\ \chi'(r_2) =& + \#v_{in}(r_1 \cup r_2) + \#v_{out}(r_1 \cup r_2 | r_2) - \#e_{in}(r_1 \cup r_2)\\ & - \#e_{out}(r_1 \cup r_2 | r_2) + \#f_{in}(r_1 \cup r_2) + \#f_{out}(r_1 \cup r_2 | r_2).\end{aligned}$$

The merging process removes inner cells in order to obtain the resulting region: $\chi'(r_1 \cup r_2)$ is only expressed using outer cells:

$$\begin{aligned}\chi'(r_1 \cup r_2) =& + \#v_{out}(r_1 \cup r_2 | r_1) + \#v_{out}(r_1 \cup r_2 | r_2) - \#e_{out}(r_1 \cup r_2 | r_1)\\ & - \#e_{out}(r_1 \cup r_2 | r_2) + \#f_{out}(r_1 \cup r_2 | r_1) + \#f_{out}(r_1 \cup r_2 | r_2)\\ \chi'(r_1 \cup r_2) =& \chi'(r_1) + \chi'(r_2) - 2(\#v_{in}(r_1 \cup r_2) - \#e_{in}(r_1 \cup r_2) + \#f_{in}(r_1 \cup r_2))\\ \chi'(r_1 \cup r_2) =& \chi'(r_1) + \chi'(r_2) - 2\chi'(in(r_1 \cup r_2)). \qquad \square\end{aligned}$$

Note 1 *Outer cells that belong to $r_1 \cap r_2$ are not specifically considered in the previous proof. Actually, these cells are edges and vertices belonging to the border of inner surfaces. There are as many vertices as edges in the border*

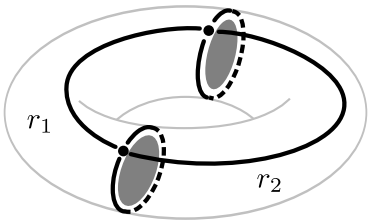


Fig. 7. Example of inner cells: the two faces drawn in *dark gray* are inner cells for $r_1 \cup r_2$. $b_1(r_1) = b_2(r_2) = 0$. $\chi'(r_1) = \chi'(r_2) = 2$. $\chi'(r_1 \cup r_2) = 0$. $b_1(r_1 \cup r_2) = 1$

of the surface, which means that its local Euler characteristic is equal to zero. Since the Euler characteristic is the value of interest and not the actual number of cells, vertices and edges that belong to $r_1 \cap r_2$ but are not inner have no effect on the Euler characteristic computation, even if they are counted twice.

In the initialization step, numbers of cells of the border of each region are computed. The number of cells is computed incrementally during the extraction of the topological map using the algorithm proposed in [21]. Thus, $\chi'(r)$ is stored and then updated for each region r during the merging process.

When looking at $r_1 \cup r_2$, the number of cells that fully belong to the intersection of the two regions is computed. These cells belong to inner faces of $r_1 \cup r_2$. Suppose $r_1 < r_2$ in the order of regions defined in Section 3. Each dart of the external surface of r_2 is traversed and each cell that fully belongs to an inner face is counted. A cell fully belongs to inner faces if each dart used to represent the cell belongs to either r_1, or r_2. This step gives $\#v_{in}(r_1 \cup r_2)$, $\#e_{in}(r_1 \cup r_2)$ and $\#f_{in}(r_1 \cup r_2)$ which are respectively the number of vertices, edges and faces that fully belong to inner faces. These values are used to compute $\chi'(in(r_1 \cup r_2))$ using Definition 4. The final step is to use Proposition 10 to obtain $\chi'(r_1 \cup r_2)$. Using the incremental value of b_2, Proposition 8 is applied to compute the number of tunnels $b_1(r_1 \cup r_2)$.

Figure 7 presents the incremental computation of $\chi'(r_1 \cup r_2)$. The Euler characteristic for the border of r_1 and for the border of r_2 is given by the alternated sum of two vertices, three edges and three faces: $\chi'(r_1) = \chi'(r_2) = 2$. The inner cells of $r_1 \cup r_2$ are the two faces drawn in dark gray: the Euler characteristic of inner cells is 2. Using the incremental formula, the Euler characteristic of the border of the union of the two regions is given by $\chi'(r_1 \cup r_2) = 0$. Thus $b_1(r_1 \cup r_2) = 1$: the resulting region has one tunnel. The first Betti number of $r_1 \cup r_2$ is $b_0(r_1 \cup r_2) = 1$ and the third Betti number $b_2(r_1 \cup r_2) = 0$ since $k = 1$, $\#s(r_1) = 1$ and $\#s(r_2) = 1$.

4 Region Merging in Topological Maps

Merging regions in the topological map can be useful for automated segmentation process like the bottom-up segmentation process proposed in [22] or for special operation like the removal of small regions. To provide time efficient algorithm for region merging, a global approach has been defined. Algorithm 1 presents the global approach of the region merging operation. More information about region merging operations are available in [16].

Algorithm 1: Global approach of the region merging operation

Data: Topological map M; `Oracle` function
Result: Merge all the regions by connected components according to `Oracle` in M.

```
1 foreach dart d of M do
2     if Oracle(region(d), region(β3(d))) then
3         Union of the disjoint-sets of region(d) and region(β3(d));
4 Remove inner faces for each disjoint-set;
5 Simplify the cells incident to the removed faces;
6 Build the new inclusion tree of regions;
```

The global region merging is divided in two main steps: the symbolic merging and the effective merging. The symbolic merging (line 1 of Algorithm 1) consists in merging regions into a disjoint-set data structure to create an high level partition of the regions. Two useful operations are defined onto disjoint-set data structures: $find$ that retrieves the belonging set of an element, and *union* that merges two sets together. The merging of the region is guided by an oracle which indicates if two regions should be merged together. The effective merging modifies the topological map to represent the resulting partition of the image. To obtain the resulting topological map, inner faces, between two regions belonging to a same disjoint-set, are first removed and the associated surfels are switched off in the embedding (line 4). Second, the topological map is simplified (line 5) to obtain the minimal representation of the partition. Third, the inclusion tree is build (line 6) from the resulting regions and the combinatorial map (see [10] for more details about the map simplification and the built of the inclusion tree).

The time complexity of Algorithm 1 is in $O(c \times \#d + \#s_{removed})$ where c is the complexity of the oracle, $\#d$ is the number of darts of the topological map and $\#s_{removed}$ is the number of removed surfel in the embedding.

To implement topological control of the region merging process, the Betti numbers are used as criteria. Computing Betti numbers on regions during the symbolic merging allows to use Betti numbers values in the oracle function. Before any merge, the process is initialized. Betti numbers are computed and stored for each region represented in the topological map. During the incremental process, stored values are updated.

The oracle function is divided into three steps. A first step allows the merge of regions according to a criterion (for instance, regions having the same label or regions having about the same color). Then, if the merge is allowed, incremental algorithms are used to compute Betti numbers values for the union of the two regions. The evaluation function of topological properties allows or denies the merge depending on the newly computed values. For example, we can deny the merging of two regions if it creates a new tunnel or if the number of cavities becomes greater than 5. If the merge is finally allowed, then the disjoint-sets of the two regions are merged and incremental features are updated for the resulting region. Using the final part of the global merge algorithm allows to obtain the desired topological map.

There is an issue to overcome. Since regions are handled using disjoint-sets during the symbolic merging, regions are not yet merged. A special algorithm is needed to traverse darts of the surface of a region ignoring inner faces. This allows to consider all the regions belonging to a same disjoint-set as only one region.

5 Experiments and Performance Analysis

In this section, experiments using region merging with and without topological control are shown. An application implementing the topological map model with operations allowing to extract the topological map from an image and merge regions has been developed. The computation of the Betti numbers has also been implemented using the classical approach and the incremental one.

Betti numbers provide an intuitive description of a topological object: they give tunnels and cavities count for regions. To provide an example of criterion using the Betti numbers, a predicate that makes the Betti numbers converge until reaching threshold values has been proposed. Two adjacent regions are allowed to merge if that does not create nor remove any tunnel or cavity or if their number evolve toward threshold values. Other example of criteria and different predicates should be defined depending on the application.

5.1 Processing Times Comparison

In this section, processing times of the region merging operation are studied. To compare results, a small (37x44x37) medical image has been used: it represents a region of interest (ROI) in a TEP image with 43198 regions. On this image a labeling function proposes a partition of this volume in 1784 regions. An algorithm that merges initial regions having the same label has been implemented. Experiments have been carried on a personal computer (AMD Athlon64, 2.0GHz, 512Mo RAM). The application is written in C++ and has been compiled using GCC 4.0.

The idea of the comparison is to measure processing times of the region merging with and without computing the Betti numbers. First, the classical approach of the computation of the Betti numbers is compared to the incremental one. In this comparison, the Betti numbers are only computed and not taken as a criterion for merges. Second, two examples of region merging with different values for the convergence criterion are used to illustrate the increase in number of resulting regions and the increase in processing time for the symbolic merging part of the algorithm.

Table 1 presents measured values during experiments. The first two rows describe the number of regions before and after the merging. The next three rows present the number of tested couple of adjacent regions, the number of computation of the Betti numbers, and the number of symbolic merging. The last four rows present processing times of different parts of the algorithm. Each column is described in the following paragraphs.

Column (1) presents results without computation of the Betti numbers. The final number of regions corresponds to the number given by the labeling function. The total processing time is about 2 seconds. The symbolic merging is, in this case, a fast processing since retrieving the label of a region is direct. In column (2), results with computation of Betti numbers by the classical approach are given. b_1 and b_2 are computed from scratch each time they are required during the oracle function. This process uses the classical algorithm that computes the Betti numbers from the topological map without taking previously computed values into account. The number of computation is equal to the number of symbolic merges since the Betti numbers are not considered as a criterion in this experiment. Dividing the overall processing time by the number of effective computation leads to an average value of 58 milliseconds per computation. Column (3) shows results of the same experiments but using the incremental approach instead of the classical one. Except for processing times results, the measured values are the same since the incremental approach intends to do the same as the classical computation algorithm. In this case, the symbolic merging part is about ten times faster than in the previous experi-

Table 1
Results from experiments: (1) without computing Betti numbers; (2) computing Betti numbers with a classical algorithm; (3) computing Betti numbers with an incremental algorithm; (4) using convergence thresholds ($b_1 \rightarrow 0$ and $b_2 \rightarrow 0$); (5) using convergence thresholds ($b_1 \rightarrow 5$ and $b_2 \rightarrow 1$).

Experiment		(1)	(2)	(3)	(4)	(5)
count	Initial regions	43198				
	Final regions	1784			1938	2158
	Possible merges	80398			100886	91535
	Computation	0	41414		62429	54057
	Symbolic merges	41414			41260	41040
proces. time (s)	Initialization	1.05	1.05	1.05	1.05	1.05
	Symbolic merging	0.07	2415.11	263.87	550.07	444.92
	Effective merging	0.94	0.93	0.95	0.93	0.95
	Total	2.06	2417.09	265.88	552.05	446.93

ment. This result shows the interest of the incremental approach as the average cost of one computation is 6.3 milliseconds. Column (4) presents results using convergence thresholds on both Betti numbers values. The aim of this threshold is to make the Betti numbers converge toward zero. This configuration leads to less effective merges. Thus, the more pair of regions are processed, the more regions remain in the final result. The average processing time of one computation is in this case of 8.8 milliseconds. The increase compared with the previous result is explained by the complexity of the regions not merged according to the Betti numbers constraint. If the merge of two regions is rejected, then other merging with adjacent regions are tried. The computation is performed more times. In column (5), results using a different threshold are given. In this experiment, the number of tunnels converges toward 5 and the number of cavity converges toward 1. The average processing time increases again because of the same problem. The final number of regions is greater since the convergence criterion is harder to satisfy.

Table 2 presents the distribution of regions by the Betti numbers. The first two rows present the distribution of the regions of the initial partition. Few regions have Betti numbers greater than 0. The next two rows present the distribution of the regions if a topological constraint is applied: Betti numbers converge toward 0. There is a smaller number of regions having Betti numbers greater than 0. The convergence threshold cannot be reached due to the labeling function: the merge of some regions is forbidden. The last two rows present the distribution of the regions if a topological constraint is applied: b_1 converges toward 5 and b_2 converges toward 1. In this case, the number of re-

Table 2
Distribution of regions by Betti numbers: (1-3) without topological constraint on Betti numbers; (4) using convergence thresholds ($b_1 \to 0$ and $b_2 \to 0$); (5) using convergence thresholds ($b_1 \to 5$ and $b_2 \to 1$).

	Regions	Value	0	1	2	3	4	5	6	7	> 7
1-3	1784	b_1	1768	6	1	2	1	1	0	0	5
		b_2	1781	2	0	0	0	0	0	0	1
4	1938	b_1	1932	2	2	0	0	0	1	1	0
		b_2	1934	1	1	0	0	0	0	0	2
5	2158	b_1	2099	24	6	9	11	5	0	2	2
		b_2	2147	4	2	0	1	0	1	0	3

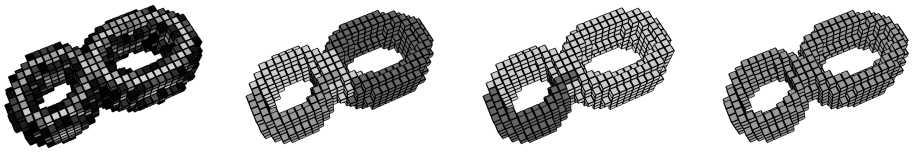
gions having more than 0 tunnel increases. The number of regions having more than 5 tunnels decreases. The number of regions that have cavities increases. However, as in the previous case, the convergence criteria of b_1 and b_2 conflict, which forbids the algorithm to reach the specified convergence threshold.

5.2 Example of Use on Artificial Image

To show the results of the topological criterion, a constraint on the second Betti number is applied to an artificial image in Figure 8. Figure 8a shows voxels of a gray-scale 3D image. In the original image, there is one region for each voxel. The application merges these regions in an order depending on voxel gray levels: the convergence criterion on Betti numbers controls the topology of the obtained regions. In Figure 8b, the criterion converges through the zero threshold: no tunnels are allowed. The 2-torus region that clearly appears is divided into three main regions that do not contain any tunnel. Figure 8c shows the result using a convergence threshold of 1. Two regions appear, one having a tunnel and not the other one. Different orders in the region merging lead to other results but no region has more than one tunnel according to the criterion. In Figure 8d, the threshold on b_1 is 5. Merges are allowed while the number of tunnel increases. In the example, the 2-torus is fully retrieved.

6 Conclusion

In this paper, two approaches of the Betti numbers computation for regions in topological maps have been presented. Firstly, an algorithm counting tun-



(a) (b) (c) (d)

Fig. 8. Experiments on 3D artificial images. The convergence threshold on the number of tunnels b_1 is given for each image. (a) voxels of a 2-torus object. Each voxel belongs to a different region. (b) 2-torus with $b_1 \rightarrow 0$. The object is divided into three main regions, none of them having any tunnel. (c) 2-torus with $b_1 \rightarrow 1$. One of the two main regions has a tunnel and the other one has none. Two 1-torus are not obtained due to the order in region merging. (d) 2-torus with $b_1 \rightarrow 5$. Actually, b_1 increases until reaching its maximal value (2) driven by voxels data.

nels and cavities by using information stored in topological maps is given to compute b_1 and b_2. Then, an incremental algorithm that allows the computation of the Betti numbers for the union of two regions is presented. It uses the Euler characteristic of the border and the number of surfaces of the two regions to retrieve b_1 and b_2. A region merging algorithm is presented to allow the modification of the represented partition of the image. This work proposes the Betti numbers as a new tool to control the topology of the regions during modification operations. Experiments show the results of the implementation of the Betti numbers criterion in the region merging operation.

The direct computation of the Betti numbers is presented. b_0 is always equals to one as there is only one single 2-connected component of voxels for each region. b_1 is computed using a formula that links the number of tunnels and the Euler characteristic of borders of a region. b_2 is computed using the inclusion tree of regions to count cavities. The incremental computation algorithm gives the Betti numbers for the union of two regions. b_1 depends on the change of the Euler characteristic of the border of the union of the two regions. The Euler characteristic computation counts cells of the border of regions represented by topological maps. The incremental computation of b_2 counts the number of newly created cavities using the number of surfaces of the two regions.

Global region merging operation has been developed. This operation allows the merging of any number of sets of connected regions. This approach of the region merging is well suited for automated processing like the bottom-up segmentation. The complexity of this operation is given and the integration of topological control to the region merging is explained. Processing times have been measured during experiments on real medical data. This shows the advantage of the incremental computation algorithm over the classical approach. Effects of the topological criterion are then shown by controlling the number of tunnels on an artificial image that represents a 2-torus region.

The results exposed in this paper show an encouraging progress toward solving

the problem of controlling the topology of regions in a 3D image. The fundamental point lies in the control of the topological information attached to each cell in the image. The topological map framework is well suited to compute complex topological features for each region in a 3D image as demonstrated by the computation of the Betti numbers. A direct perspective is to improve processing times of the Betti numbers computation algorithms. Currently, most of the processing time of the computation is consumed into map traversal. An idea would be to precompute some values in order to speed up computation.

To extend the work presented in this paper, other operations will be studied to allow different approaches to modify the partition of the image. For instance, the region splitting algorithm that divides a region into several ones is a key operation for split and merge segmentation. Implementing topological control during the split operation should allow to segment complex structures like brains. Further research will aim at the computation of other topological features like homology groups generators in order to provide more tools to develop segmentation within the topological map framework. Lastly, we want to use topological criterion in order to solve real world segmentation issues.

7 Acknowledgments

A preliminary version of this paper was presented at the 12th International Workshop on Combinatorial Image Analysis, Buffalo, NY, USA, April 2008 [16].

References

[1] D. W. Shattuck, R. M. Leahy, Automated graph based analysis and correction of cortical volume topology, IEEE Trans. Med. Imaging 20 (11) (2001) 1167–1177.

[2] F. Ségonne, W. E. L. Grimson, B. Fischl, Topological correction of subcortical segmentation, in: MICCAI (2), 2003, pp. 695–702.

[3] B. R. Gomberg, P. K. Saha, H. K. Song, S. N. Hwang, F. W. Wehrli, Topological analysis of trabecular bone mr images, IEEE Trans. Med. Imaging 19 (3) (2000) 166–174.

[4] S. Miri, N. Passat, J.-P. Armspach, Topology-preserving discrete deformable model: Application to multi-segmentation of brain mri, in: ICISP, 2008, pp. 67–75.

[5] J. R. Munkres, Elements of Algebraic Topology, Addison-Wesley, 1984.

[6] C. J. A. Delfinado, H. Edelsbrunner, An incremental algorithm for betti numbers of simplicial complexes on the 3-sphere, Computer Aided Geometric Design 12 (1995) 771–784.

[7] C.-Y. Hsu, C.-H. Yang, H.-C. Wang, Topological control of level set method depending on topology constraints, Pattern Recognition Letters 29 (4) (2008) 537–546.

[8] A. Rosenfeld, Adjacency in digital pictures, Information and Control 26 (1) (1974) 24–33.

[9] W. Kropatsch, H. Macho, Finding the structure of connected components using dual irregular pyramids, in: Proceedings of International Conference on Discrete Geometry for Computer Imagery, DGCI, 1995, pp. 147–158, *invited lecture*.

[10] G. Damiand, Topological model for 3d image representation: Definition and incremental extraction algorithm, Computer Vision and Image Understanding 109 (3) (2008) 260–289.

[11] J.-P. Braquelaire, J.-P. Domenger, Representation of segmented images with discrete geometric maps, Image and Vision Computing 17 (10) (1999) 715–735.

[12] G. Damiand, Y. Bertrand, C. Fiorio, Topological model for two-dimensional image representation: definition and optimal extraction algorithm, Computer Vision and Image Understanding 93 (2) (2004) 111–154.

[13] C. Fiorio, A topologically consistent representation for image analysis: the frontiers topological graph, in: Proceedings of International Conference on Discrete Geometry for Computer Imagery, DGCI, Lyon, France, 1996, pp. 151–162.

[14] U. Köthe, Xpmaps and topological segmentation - a unified approach to finite topologies in the plane, in: Proceedings of International Conference on Discrete Geometry for Computer Imagery, DGCI, Bordeaux, France, 2002, pp. 327–350.

[15] H. Meine, U. Köthe, The geomap: A unified representation for topology and geometry, in: 5th IAPR International Workshop, GbRPR 2005, Poitiers, France, 2005, pp. 132–141.

[16] A. Dupas, G. Damiand, Comparison of local and global region merging in the topological map, in: V. Brimkov, R. Barneva, H. Hauptman (Eds.), Proceedings of 12th International Workshop on Combinatorial Image Analysis (IWCIA'08), Vol. 4958 of Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, Buffalo, NY, USA, 2008, pp. 420–431.

[17] J. Edmonds, A combinatorial representation for polyhedral surfaces, Notices of the American Mathematical Society 7 (1960) 1.

[18] P. Lienhardt, Topological models for boundary representation: a comparison with n-dimensional generalized maps, Computer-Aided Design 23 (1991) 59–82.

[19] E. Khalimsky, R. Kopperman, P. Meyer, Boundaries in digital planes, Applied Mathematics and Stochastic Analysis 3 (1) (1990) 27–55.

[20] G. Damiand, P. Resch, Split and merge algorithms defined on topological maps for 3d image segmentation., Graphical Models 65 (1-3) (2003) 149–167.

[21] G. Damiand, P. Peltier, L. Fuchs, P. Lienhardt, Topological map: An efficient tool to compute incrementally topological features on 3d images, in: Proceedings of International Workshop on Combinatorial Image Analisys, IWCIA, Berlin, Germany, 2006, pp. 1–15.

[22] A. Dupas, G. Damiand, First results for 3d image segmentation with topological map, in: Proceedings of International Conference on Discrete Geometry for Computer Imagery, DGCI, Lyon, France, 2008, pp. 507–518.