Topologically Constrained Segmentation with Topological Maps

Alexandre Dupas, Guillaume Damiand

Topologically Constrained Segmentation with Topological Maps

Alexandre Dupas[1] and Guillaume Damiand[2]

[1] XLIM-SIC, Université de Poitiers, UMR CNRS 6172,
Bâtiment SP2MI, F-86962 Futuroscope Chasseneuil, France
dupas@sic.univ-poitiers.fr
[2] LaBRI, Université de Bordeaux 1, UMR CNRS 5800, F-33405 Talence, France
damiand@labri.fr

Abstract This paper presents incremental algorithms used to compute Betti numbers with topological maps. Their implementation, as topological criterion for an existing bottom-up segmentation, is explained and results on artificial images are shown to illustrate the process.

Keywords: Topological map, Topological constraint, Betti numbers, Segmentation.

1 Introduction

In the image processing context, image segmentation is one of the main steps. Currently, most segmentation algorithms take into account color and texture of regions. Some of them also introduce geometrical criteria, like taking the shape of the object into account, to achieve a good segmentation. Lesser works use topological information to enhance the segmentation of images. Topological maps have been developed with this objective in mind (see [1] for details). It aims to represent all the cells of the image and the incidence relations between these cells so we can easily compute topological features for each region of an image.

Our work intends to define operations and image processings that profit from topological features represented by topological maps to enhance the results. In this paper, we show how topological maps are used to integrate topological constraints within an existing bottom-up segmentation process, defined in [5]. In this work, Betti numbers are the topological invariants used. They count connected components, tunnels (also called handles) and cavities (sometimes called voids) for 3D objects. This interpretation gives a visual and comprehensive criterion on the desired objects in images. Betti numbers are wide studied topological features. For instance, Delfinado and Edelsbrunner gives a direct method [3] for computing the Betti numbers on simplicial complexes. Betti numbers are also used in 2D segmentation to control a level set approach of segmentation [6].

The paper is organized as follow. The topological map, used to represent 3D images is introduced in Sect. 2. Section 3 presents the computation of Betti numbers using topological maps. It also shows the incremental algorithm that is used during the segmentation process. In Sect. 4 the implementation of the constraint in the segmentation algorithm is discussed and some results on artificial images are shown. Section 5 concludes this paper and some perspectives of this work are given.

2 3D Topological Maps

3D topological maps are an extension of combinatorial maps used to represent partitions of 3D images into regions. This framework aims to represent all the cells belonging to the border of

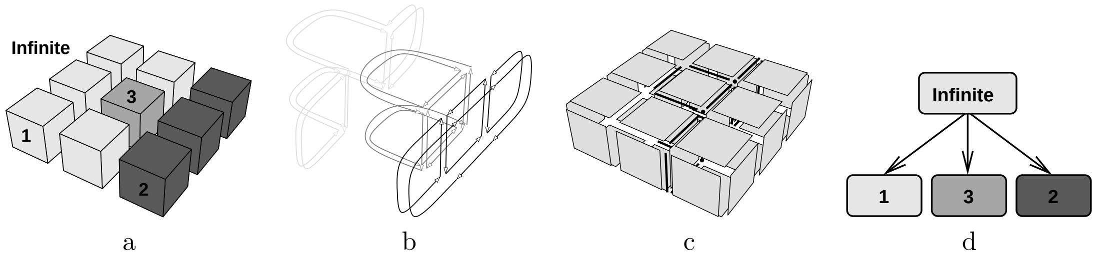

Figure 1: The different parts of the topological map used to represent an image. (a) 3D image. (b) Minimal combinatorial map. (c) Intervoxel matrix. (d) Inclusion tree of regions.

the regions and to preserve incidence relations between these cells. With these information, the topological map represents topological relations between regions.

A 3D image is a set of voxels having a grey level value representing the data. In 3D image, the notion of regions is defined by a 2-connected set of homogeneous voxels. Homogeneity is defined by a criterion which can be for instance a distance measure.

The 3D topological map uses a 3D combinatorial map to represent the borders of regions. For instance, a border between two adjacent regions is a face represented in the combinatorial map. To obtain a memory and time efficient structure, the number of cells in the combinatorial map is minimized such as no adjacency relation between regions is lost. The obtained combinatorial map representing the border of regions is called minimal.

The minimal combinatorial map does not represent the inclusion relation between regions. To preserve this information, an inclusion tree of regions is used. There is a link between regions in the tree and darts in the combinatorial map as each dart belongs to a region and each region is attached to a specific dart called representative dart that belongs to the external surface of the region.

The geometry of regions is given using the intervoxel framework to represent geometry of all cells of the minimal combinatorial map. This structure is the embedding of the map and thus allows to compute geometrical features for regions.

The topological map is formed by the association of these three structures representing a 3D image. Figure 1 shows an example of the topological map representing the image in Fig. 1(a). The minimal combinatorial map in Fig. 1(b) represents the three regions with 3 volumes, 8 faces, 6 edges and 4 vertices. This is the minimal representation of cells belonging to borders of regions. Figure 1(c) presents the intervoxel embedding of cells and Fig. 1(d) shows the inclusion tree of regions.

3 Computation of Topological Features

We are interested in constraining operations on topological maps with topology. Thus we aim to compute topological features of each region in order to define criteria using topological characteristics to constrain processes.

Betti numbers are topological invariants for homology groups that count holes in each dimension and thus represent the topological connectivity of objects. In a 3-dimensional space, as in 3D images, Betti numbers count connected components (b_0), tunnels (b_1) and cavities (b_2).

For instance the Betti numbers of the 3D object presented in Fig. 2(a) are $b_0 = 1$, $b_1 = 3$ and $b_3 = 3$.

The computation of Betti numbers is a classical problem in topology and we present here how they are computed for regions in the topological map. Due to the topological map definition, regions are sets of 2-connected voxels which means that each region r has only one connected component and thus $b_0(r) = 1$.

a b

Figure 2: (a) 3D object with one connected component, three tunnels and three cavities. (b) Union of two regions r_{min} and r_{max} with in dark gray internal marked faces on r_{max} and in light gray, the two other surfaces s_1 and s_2. In this case the merging of the two regions leads to the creation of a cavity associated to s_2. s_1 is incident to the external surface of the union and thus does not lead to the creation of a new cavity.

The number of cavities $b_2(r)$ of a region r is given by the number of included connected component of regions. The inclusion tree of regions is used to retrieve such information. An interesting property is that the number of surfaces of a region is equal to the number of cavities plus one. Indeed, for each cavity there is one internal surface and we have to count the external surface that is unique as there is only one connected component (in general the formula for the number of surfaces is $b_2(r) + b_0(r)$).

The number of tunnels $b_1(r)$ of the region r is related to genus of its surfaces. Indeed, genus $g(s)$ of surface s counts the number of tunnels of the surface. So, the sum of the genus for each surface of the region counts the number of tunnels for that region.

The genus g of a surface is, when considering closed orientable surfaces, linked to Euler characteristic χ of the surface by $g = \frac{2-\chi}{2}$. The Euler characteristic of a surface is computed using the polyhedral formula. Let be v, e and f the number of vertices, edges and faces of the considered surface thus the Euler characteristic of the surface is $\chi = v - e + f$.

As we are interested in regions, the modified Euler characteristic $\chi'(r)$ of region r is defined as the sum of $\chi(s), \forall s \in Surface(r)$. With this definition, $b_1(r)$ that is equal to the sum of genus is:

$$b_1(r) = \sum_{s \in Surface(r)} \frac{2 - \chi(s)}{2} = \frac{2(b_2(r) + 1) - \chi'(r)}{2}$$

We have shown how Betti numbers are computed using topological map. But this approach is not well suited for a segmentation process. Thus an incremental approach to compute b_1 and b_2 has been defined. To fit with segmentation parameters the incremental computation is defined when merging two connected regions called r_1 and r_2 in the following explanations.

Firstly, let define a useful property for regions. There is an order on regions based on the position of the first voxel of each region in the sweeping order of the image. We have $r_{min} = min(r_1, r_2)$ and $r_{max} = max(r_1, r_2)$. According to the definition of topological maps, there are only two possible configurations:

- r_{max} is included in r_{min};
- r_{max} and r_{min} belong to the same connected component of regions.

In the following parts, incremental algorithms used to compute b_2 and then b_1 are presented in that order as the incremental computation of b_1 needs the b_2 value.

3.1 Incremental Computation of b_2

The incremental computation of the number of cavities consists in finding changes in number of cavities when merging the two regions. The initialization part of this algorithm is to compute for each region the number of cavities. This is done using the classical algorithm that uses the inclusion tree of regions. The number of cavities of a region is stored and updated during the process.

When computing b_2 for $r = r_1 \cup r_2$, the first step is to run through the darts of the external surface of r_{max} and mark as internal each dart d such as $region(\beta_3(d)) = r_{min}$. Thus each face between the two regions has its darts belonging to r_{max} marked.

The second step is again to run trough darts of the external surface of r_{max} and compute the number k of connected components of darts ignoring internal marked darts. Each connected component of darts is representing a distinct surface for $r_1 \cup r_2$. The number of newly included regions can easily be computed from this value: this is $k - 1$.

Indeed, one of this connected component represent also a surface of r_{min}. If this surface is the external surface of r_{min} then it is also the external surface of r and not a cavity. If this surface is an internal surface of r_{min} the cavity is already counted by $b_2(r_{min})$. A special case is when no connected component of darts can be found ($k = 0$): r_{max} fills a cavity of r_{min}. The number of cavities for r has to decrease and this is taken into account by the formula.

Finally the number of cavities is obtained by adding $k-1$ to the sum of the number of cavities of the two regions.

$$b_2(r) = b_2(r_1) + b_2(r_2) + k - 1$$

Figure 2(b) presents an example of a classic configuration. The two regions, r_{min} and r_{max} have no inclusion. The external surface of r_{max} has been proceeded. The dark grey face is the only internal face. The two light grey ones, s_1 and s_2, represent two discovered connected components. s_1 is corresponding to the external surface of $r = r_{min} \cup r_{max}$ and is not the surface of a new cavity. s_2 on the contrary is an internal surface for r and bound a new cavity. In this case, $k = 2$ and $b_2(r) = 1$ as there is a new void.

3.2 Incremental Computation of b_1

The main idea behind the incremental computation of $b_1(r)$ is to look for changes in the modified Euler characteristic $\chi'(r)$. Using the general formula with $b_2(r)$ computed incrementally in Sect. 3.1, $b_1(r)$ is then computable. In the initialization part of the incremental algorithm, number of cells of each region is computed. This may be computed incrementally during the extraction of the topological map using the algorithm proposed in [2]. Thus χ' is stored and then updated for each region during the process.

When looking to $r = r_1 \cup r_2$, we are interested by the number of cells that will be removed when merging the two regions r_1 and r_2. These cells are the ones that belong to internal faces marked during the incremental computation of $b_2(r)$. So darts of the external surface of r_{max} are covered and each cell that fully belongs to internal marked faces is counted. A cell fully belongs to internal marked faces when at least one dart used to represent the cell is marked as internal and when each other dart belongs either to r_{min} or r_{max}. This step gives v_{int}, e_{int} and f_{int} which are respectively the number of vertices, edges and faces that fully belong to internal marked faces. These values are used to compute χ'_{int}.

The final step is to compute $\chi'(r) = \chi'(r_1) + \chi'(r_2) - 2\chi'_{int}$ and use this value to compute the number of tunnels $b_1(r)$ with general formula. Removed cells are counted twice in $\chi'(r_1)$ and $\chi'(r_2)$. This explain why χ'_{int} is subtracted twice.

$$b_1(r) = \frac{2(b_2(r) + 1) - \chi'(r)}{2}$$

4 Topologically Constrained Segmentation

In a previous work, a bottom-up segmentation process as been defined [5]. It uses an existing criterion based on the notion of contrast. In this work, the predicate that guides the segmentation process is extended using a topological constraint on Betti numbers.

4.1 Implementation during a bottom-up segmentation process

The bottom-up segmentation process uses a two steps algorithm. In a first part, regions are merged in a disjoint-set forest represented by union-find trees of regions (see [4] for more details). This is the high level view of the segmentation. The incremental computation algorithms used to obtain b_1 and b_2 have been adapted with a special darts coverage that uses union-find trees of regions to run through regions by taking into account symbolic merged regions.

In the initialization part of the segmentation χ' and b_2 is computed for each region as requested by the two incremental algorithms. Then algorithms are used to obtain the new number of tunnels and cavities. If both contrast and topological criteria allow the merge of regions, the stored values are updated in the high level representation. Finally, when all merging have been done, we use the last step of the segmentation algorithm to produce the corresponding topological map.

Several criteria on Betti numbers may be written such as a criterion that only allows the number of cavities to decrease. Another possibility may be to disallow changes of topology when merging two regions: when merging two regions the sum of their Betti numbers should be equal to the resulting Betti numbers (no tunnels and cavities created nor filled). A third option, used in this work, is to threshold Betti numbers: if resulting Betti numbers are greater than thresholds, then the merging is denied.

4.2 Results

To show results of segmentation, a constraint on Betti numbers is applied to artificial images. Figure 3 shows results of segmentation. For each figure, the maximal Betti numbers are given using the notation Betti(b_1,b_2) where b_i is the maximal allowed value for the i-th Betti number.

Let us discuss about the results shown in Fig. 3. The first image presented in Fig. 3(a) is an empty sphere segmented by constraining the number of tunnels and cavities to zero. Only one voxel is not merged with the other. This allows the dark region to have required topological properties according to the Betti numbers criterion. The three last figures represent the segmentation of a 2-torus part of an image according to different Betti numbers thresholds. For each of these images a 3D view of the main regions is given as well as a slice view of the image with all the regions having a different color. In Fig. 3(b) the Betti criterion allows the apparition of a region with 2 tunnels. The 2-torus has been segmented. In Fig. 3(c) tunnels and cavities are disallowed. This lead to 3 regions representing the 2-torus. On the slice view, as in the previous image, small regions are adjacent to the 2-torus. These regions cannot be merged with the 3 main regions due to the contrast criterion. They also cannot be merged with the surrounding region since it has a cavity which means that b_2 is greater than the threshold. Figure 3(d) shows the segmentation obtained without allowing tunnels but allowing a cavity. The 3 main regions are the same but the slice view shows less small adjacent regions since many of them have been allowed to merge with the surrounding region.

These results show a bit of the influence of the Betti criterion during the bottom-up segmentation.

5 Conclusion

In this paper a topological criterion is used to constrain the topology of regions during a segmentation process. Betti numbers, classical topological invariants, have been used. They count the number of connected components, in our case always one, the number of tunnels of a region as well as the number of cavities. We have shown how to compute these two last values both using information represented by topological maps and incrementally during a segmentation process. Some details about implementation of topological constraints have been discussed and finally some experiments show possibilities of topologically constrained segmentation in the image processing context.

In future works, we are looking into other uses of the Betti criterion. Currently, the thresholding criterion is used to deny some configurations during the segmentation process. As shown in the

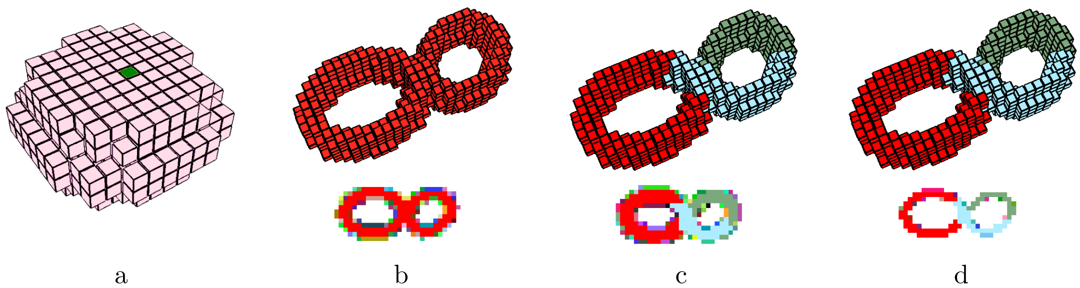

Figure 3: Experiments on 3D artificial images. The maximal value allowed on b_1 and b_2 is given for each images by Betti(b_1,b_2). On the last three images, a slice of the image is provided to show the amount of different regions around the segmented part of the 2-torus. (a) Sphere with Betti(0,0). (b) 2-torus with Betti(2,0). (c) 2-torus with Betti(0,0). (d) 2-torus with Betti(0,1).

results it leads to the impossibility to obtain the desired segmentation. Some other proposed criteria will be tried as well as new ones and experiments have to be achieved with real images. For instance, the criterion can be used as a segmentation checker. In this case, the segmentation process is non-controlled until the end. The result is checked and if the configuration is not correct using the incremental computation, we may know if a good segmentation has been obtained and thus allow to process the image again and stop when the good configuration occurs.

We are also interested in defining new operations on topological map like the splitting of regions and being able to guide the split according to topological criteria. For instance, splitting a 2-torus region into two 1-torus regions may be a useful task.

References

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

[2] G. Damiand, P. Peltier, L. Fuchs, and P. Lienhardt. Topological map: An efficient tool to compute incrementally topological features on 3d images. In *Proceedings of 11th International Workshop on Combinatorial Image Analysis*, volume 4040 of *Lecture Notes in Computer Science*, pages 1–15, Berlin, Germany, June 2006.

[3] Cecil Jose A. Delfinado and Herbert Edelsbrunner. An incremental algorithm for betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12:771–784, 1995.

[4] A. Dupas and G. Damiand. Comparison of local and global region merging in the topological map. In *Proceedings of 12th International Workshop on Combinatorial Image Analysis (IWCIA'08)*, volume 4958 of *Lecture Notes in Computer Science*, pages 420–431, Buffalo, NY, USA, April 2008. Springer-Verlag Berlin Heidelberg.

[5] A. Dupas and G. Damiand. First results for 3d image segmentation with topological map. In *Proceedings of 14th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI'08)*, volume 4992 of *Lecture Notes in Computer Science*, pages 507–518, Lyon, France, April 2008. Springer-Verlag.

[6] Chih-Yu Hsu, Chih-Hung Yang, and Hui-Ching Wang. Topological control of level set method depending on topology constraints. *Pattern Recognition Letters*, 29(4):537–546, March 2008.