

First Results for 3D Image Segmentation with Topological Map

Alexandre Dupas, Guillaume Damiand

▶ **To cite this version:**

Alexandre Dupas, Guillaume Damiand. First Results for 3D Image Segmentation with Topological Map. Discrete Geometry for Computer Imagery, Apr 2008, Lyon, France. pp.507-518, 10.1007/978-3-540-79126-3_45 . hal-00305485

HAL Id: hal-00305485
https://hal.science/hal-00305485

Submitted on 24 Jul 2008

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.

First Results for 3D Image Segmentation with Topological Map* **

Alexandre Dupas[1] and Guillaume Damiand[2]

[1] SIC-XLIM, 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 the first segmentation operation defined within the 3D topological map framework. Firstly we show how a traditional segmentation algorithm, found in the literature, can be transposed on a 3D image represented by a topological map. We show the consistency of the results despite of the modifications made to the segmentation algorithm and we study the complexity of the operation. Lastly, we present some experimental results made on 3D medical images. These results show the process duration of this method and validate the interest to use 3D topological map in the context of image processing.

Key words: Topological model, 3D Image segmentation, Intervoxel boundaries, Combinatorial maps.

1 Introduction

Segmentation of 3D images is a great challenge in many fields as for example in the analysis of medical images. The segmentation refers to the process of partitioning an image into regions which are homogeneous to a criterion. This kind of approach, called *region-based* segmentation, requires a representation of regions in the image.

There are many works that have studied the definition of such a structure to represent images. Topological data structures describe the image as a set of elements and their neighborhood relations. The most famous example is the Region Adjacency Graph (RAG) [1] 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 multiple adjacency or makes no difference between inclusion and adjacency relations. To solve these issues the RAG model has been extended, for instance in dual-graph structure to represent 2D images [2] or in topological maps [3–6]. These last have already been used in segmentation of 2D images [7, 8] and a previous work has defined two operations

* Partially supported by the ANR program ANR-06-MDCA-008-05/FOGRIMMI.
** Paper published in Proceedings of 14th International Conference DGCI, LNCS 4992, pp. 507-518, 2008. Thanks to Springer-Verlag Berlin Heidelberg. The original publication is available at http://www.springerlink.com/content/q783qhrp33n15v0v/

on 3D topological maps needed to achieve image segmentation [9] but without using it in a segmentation process.

Our general objective is to develop segmentation operations which profit from all the information stored by topological maps. As a first step toward this goal, we show in this paper how topological maps may be used as a 3D image representation model in a traditional segmentation process. In our knowledge, this is the first time that 3D image segmentation is achieved by using combinatorial maps.

In the literature, we found a region-based segmentation method, proposed by P. F. Felzenszwalb and D. P. Huttenlocher in [10], which seems to provide interesting results. This method merges neighboring homogeneous regions in order to produce another homogeneous region. This approach of the segmentation is called *bottom-up*. As a variant of the *split-and-merge* methods, it consists in the merging of small regions into bigger ones. In the original work, the segmentation using a local criterion is defined on 2D graph representation of the image, but was extended to different models and in particular to a hierarchical combinatorial map representation of 2D image in [8].

In this paper we present how this segmentation technique can be transposed to the topological map framework. The implementation of the homogeneity criterion, its computation and the algorithm used are also provided. We show that the results obtained with this method are similar to those that could be obtained with the original approach. Moreover the processing time is suitable for 3D image segmentation as it allows to segment real medical images. At this time, using topological map does not improve original methods. The next step of this work is to mix this approach with topological criteria to improve the segmentation results and show the contribution of topological maps to the 3D image processing.

We first present in Sect. 2 topological maps, which are combinatorial maps verifying specific properties, used to represent 3D images. We also introduce a topological map manipulation operation used in this work : the region merging. Then, Sect. 3 details the criterion and presents the segmentation algorithm. We also explain why despite the differences between this method and the classical one, the results are similar. In Sect. 4 we present the complexity analysis of the segmentation operation and we give some experimental results on 3D medical images. Lastly, we conclude and give some perspectives in Sect. 5.

2 Recalls on 3D Topological Maps

A 3D topological map is an extension of a combinatorial map used to represent a 3D image partition. Let us recall the notions of combinatorial maps, 3D images, intervoxel elements and topological maps that are used in this work.

2.1 Combinatorial Map

A combinatorial map is a mathematical model describing the subdivision of a space, based on planar maps. A combinatorial map encodes all the cells of the

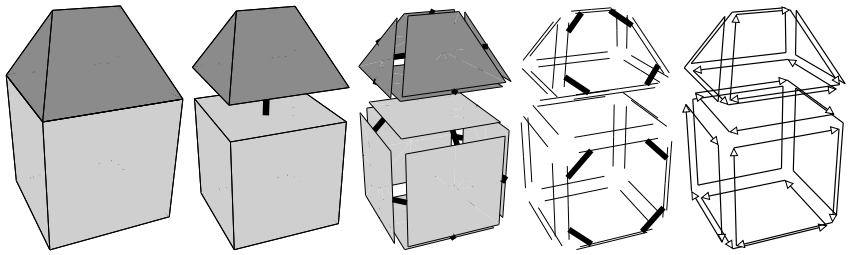


Fig. 1. The successive decompositions of an object to obtain the corresponding 3-map. (A) A 3D object. (B) Disjointed volumes. (C) Disjointed faces. (D) Disjointed edges. (E) Corresponding combinatorial map.

subdivision and all the incidence and adjacency relations between the different cells, and so describe the topology of this space.

A combinatorial map can be obtained intuitively by successive decompositions as we can see in Fig. 1. To describe the 3D object shown in Fig. 1 A, we first decompose the volumes of this object (Fig. 1 B) then the faces of these volumes (Fig. 1 C) and finally the edges of these faces (Fig. 1 D). At each step, we keep the adjacency relations between the decomposed cells (drawn by black segments but only partially drawn for the last step). The obtained elements after the last decomposition are called *darts* and are the only basic elements used in the definition of the combinatorial maps. In order to obtain the map, we report each adjacency relation onto darts. We call β_i the relation between two darts which describes an adjacency between two i-dimensional cells. Figure 1 E presents the combinatorial map corresponding to object shown in Fig. 1 A.

Let us see now the formal definition of 3D combinatorial maps:

Definition 1 (3D combinatorial map). *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[3] *on* D*;*
3. β_2 *and* β_3 *are two* involutions[4] *on* D*;*
4. $\beta_1 \circ \beta_3$ *is an involution*[5] *on* D*.*

The different constraints of the 3-map definition (β_1 is a *permutation*, other β_i are *involutions* and $\beta_1 \circ \beta_3$ is an involution) ensures the topological validity of described objects. For example, intuitively the last constraint says that two volumes can not be partially adjacent. If two volumes are adjacent for a face, they must be adjacent for all the edges of the face. See [11] for more details on maps and comparison with other combinatorial models.

[3] A *permutation* on a set S is a one to one mapping from S onto S.
[4] An *involution* f on a set S is a one to one mapping from S onto S such that $f = f^{-1}$.
[5] $\beta_1 \circ \beta_3$ is the composition of both permutations: $(\beta_1 \circ \beta_3)(x) = \beta_1(\beta_3(x))$.

2.2 3D Images and Intervoxel Elements

Let us now recall some usual notions about images and intervoxels elements. A voxel is a point of discrete space $\mathbb{Z}^3$ associated with 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 voxel sets having the same labeled value and which are 6-connected. The label of a voxel is given by a labeled function $l : \mathbb{Z}^3 \rightarrow L$ which gives for each voxel its label (a value in the finite set L). We speak about region for a maximal set of 6-connected voxel having the same label.

To avoid particular process for the image border voxels, we consider an infinite region R_0 that surrounds the image. If a region R_j is completely surrounded by a region R_i we say that R_j is *included* in R_i.

In the intervoxel framework [12], an image is considered as a subdivision of a 3-dimensional space in a set of cells: voxels are the 3-cells, surfels the 2-cells between two 3-cells, linels the 1-cells between two 2-cells and pointels the 0-cells between two 1-cells.

2.3 Topological Map

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 topological map. The 3D image, divided into three regions plus the infinite region R_0 (Fig. 2 A), is represented by the topological map which is divided in three parts labeled B, C and D. The minimal combinatorial map extracted from this image is displayed in Fig. 2 B. The embedding of the map is represented in Fig. 2 C and the inclusion tree of regions in Fig. 2 D.

The combinatorial map allows the representation of all the incidence and adjacency relations between cells of the subdivision. In the topological map framework, we use the combinatorial map as a topological representation of the partition of an image in regions. Each face of the topological map is separating two adjacent regions and two adjacent faces do not separate the same two regions. With these rules, we ensure the minimality (in number of cells) of the topological map (see [13, 9] for more details on topological maps).

The intervoxel matrix is the embedding of the combinatorial map. Each cell of the map is associated with intervoxel elements representing 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

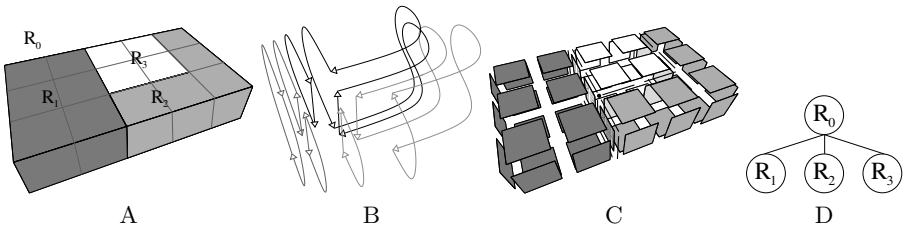


Fig. 2. The different parts of the topological map used to represent an image. (A) 3D image. (B) Minimal combinatorial map. (C) Intervoxel matrix (embedding). (D) Inclusion tree of regions.

border of edges, are embedded by pointels. Thus the intervoxel matrix allows to retrieve the geometry of the labeled image represented by the combinatorial map.

The inclusion tree of regions represents the inclusion relations. Each region in the topological map is associated to a node in the inclusion tree. The nodes are linked together by the inclusion relation defined in Sect. 2.2. To link this tree with the combinatorial map, each dart d of the map knows its belonging region (called $region(d)$). Each region R knows one of its dart called *representative dart* (called $rep(R)$).

The topological map can be modified by using an operation of regions merging [14]. This process, called *global merge*, allows to merge any number of sets of connected regions. In the resulting topological map, each set of connected regions is represented by only one region.

In order to handle such sets, we use a *disjoint-set forest* [15] of regions. We use the *union-find* trees to represent the disjoint-sets. The two possible operations on sets are the union of two sets, and find the corresponding set of a particular element. In [16], R. Tarjan shows that union and find on disjoint-set represented by trees can be considered as constant time operations.

The principle of the algorithm of global merging is to remove all existing faces between regions of a same set: we remove the faces in the combinatorial map and their corresponding embedding. The next step is to simplify incident cells to respect the minimality of the topological map. This process is also performed both in the combinatorial map and in the embedding. The last step of the global merging process is to rebuild the inclusion tree by using the list of the remaining regions and the combinatorial map already built.

The complexity of this operation is $O(|D|+|S|)$ where $|D|$ is the total number of darts in the combinatorial map and $|S|$ is the number of surfels of the removed faces in the embedding (see [14] for more details).

3 Operation of Segmentation

The segmentation is an image processing operation which leads to partition an image into multiple regions. The goal of segmentation is to simplify the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects in images.

The result of image segmentation is a set of regions that cover the entire image. All voxels in a region are similar with respect to some characteristics or computed properties, such as color or intensity. Adjacent regions are significantly different with respect to the same characteristics.

Let us present the characteristic used in this work to show the capacity of the topological map to be used in an image processing operation.

3.1 Criterion Used in the Segmentation Process

This method uses a local criterion proposed by P. F. Felzenszwalb and D. P. Huttenlocher in [10]. This segmentation criterion is based on intensity differences between neighboring pixels in 2D image. It is a region based segmentation using a characteristic onto regions and another characteristic between neighboring regions to produce a well-segmented (i.e. not over-segmented neither under-segmented) partition of the image.

We recall the principle of this segmentation. Each couple of adjacent regions is characterized by an external variation ($Ext(R_i, R_j)$). This value is the smallest intensity difference between neighboring voxels, one belonging to R_i and the other one belonging to R_j.

Each region is characterized by an internal variation ($Int(R)$): this value depends on voxels that are contained in the region. If a region contains voxels of the same intensity, its internal variation is 0. The authors in [10] prove that the new internal variation of two neighboring regions merged is equal to the external variation between the two regions. It allows the incremental computing of the internal variation of regions during the segmentation process.

We say that two regions are similar, and should be merged into one region, when the external variation between the regions is smaller than their minimum internal variation:

$$Ext(R_i, R_j) \leq MInt(R_i, R_j)$$

Where the minimum internal variation $MInt$ is

$$MInt(R_i, R_j) = min(Int(R_i) + \tau(R_i), Int(R_j) + \tau(R_j))$$

The threshold function τ controls the degree to which the external variation can actually be larger than the internal variations, and still have the regions be considered similar. We use, in this work, the same function as in the original work which depends on the size of the region, $\tau(R) = k/|R|$ where $|R|$ is the size (in number of voxels) of the region R and k is a constant defined by the user depending on the considered image.

To use this process with the topological map, the notion of external variation is transposed on faces. We define the external difference of a face F to be the smallest intensity difference between neighboring voxels across the face F (i.e. one voxel is on a side of the face, and its neighboring voxel is on the other side).

This value is related to the external variation of regions by the following formula. Let be two regions R_i and R_j witch are adjacent by k faces F_p, $p \in [1..k]$. The external variation $Ext(R_i, R_j)$ is equal to the minimum external variation of the faces F_p with $p \in [1..k]$. This gives the following formula:

$$Ext(R_i, R_j) = min(Ext(F_p), p \in [1..k])$$

For performance purposes, we store these values into the topological map. Each dart of the topological map knows both its belonging region and its belonging face. With the help of this structure, we store on each face its external variation and on each region its internal variation. These values are computed during the creation of the topological map and each operation applied to the map updates these values accordingly. So for each region and face of the topological map, we retrieve the corresponding variations in a constant time.

3.2 Algorithm

Algorithm 1 shows the segmentation process using a threshold function to produce the optimal[6] segmentation of the 3D image represented by the given topological map.

This process is divided in two steps. First, we build a symbolic merging of the regions. This step corresponds to the choice of the regions to merge in the topological map. Then, we use the merging operation recalled in Sect. 2.3 to convert the symbolic merging into an effective one.

The first step uses a disjoint-set forest of regions to efficiently represent the symbolic merging of regions. Before the segmentation, each region belongs to its own set. To merge symbolically two regions, we merge the two disjoint-set representing these regions. We compute the new characteristics of the resulting region and we set them to the head of the disjoint-set (which represents the resulting region).

To initialize the process, we build a list of faces sorted by increasing external variation. The first face in the list has the lowest external variation. In fact, each face his represented by one of its darts. To build the list, we run through each dart of the topological map. If its incident face is not marked, we add the current dart to the list L and we mark the corresponding face. When all the darts have been processed, the list contains one dart for each face of the image. Then the list is sorted by increasing order of the external variation of the incident face, by using a classical sort algorithm.

For each dart, we first check if the two incident regions around the incident face are not already merged. This is performed by checking if the belonging region

[6] according to P. F. Felzenszwalb and D. P. Huttenlocher in [10].

Algorithm 1: Segmentation

Input: Topological map M, Threshold function τ
Result: M is modified and represents the segmentation of the initial map

$L \leftarrow$ build the sorted list of faces;
while $L \neq \emptyset$ **do**
  $F \leftarrow L.pop()$;
  $R_1 \leftarrow region(F), R_2 \leftarrow region(\beta_3(F))$;
  if $R_1 \neq R_2$ **then**
    if $Ext(F) \leq MInt(R_1, R_2)$ **then**
      $R \leftarrow$ union of R_1 and R_2 in the disjoint-set forest;
      $Int(R) \leftarrow Ext(F)$;
Apply the global operation of region merging;

of the current dart d does not belong to the same disjoint-set as the belonging region of the dart $\beta_3(d)$ (i.e. the dart which belongs to the second region incident to the face). Then we compute the $MInt$ value for the two incident regions and if this value is lesser or equal to the external variation of the current face, we symbolically merge the two regions.

When all faces have been processed, the segmentation algorithm uses the merging operation to produce the topological map corresponding to the resulting regions.

The main difference between this algorithm and the graph-based process described in [10] is the usage of the external variation onto each face instead of the external variation between couple of regions. This modification is necessary since the topological map represents multi-adjacency while RAG only represents simple adjacency. We have to prove that both approaches are equivalent.

Property 1 ensures that two regions are merged if and only if they respect the original criterion equation. This property says that two regions R_i and R_j may be merged together if and only if they merge during the process of the first face which is incident to both regions in the increasing order of their external variation (this is due to the fact that the external variation of this first face is equal to the external variation between the two regions).

Property 1 (Validity). If two regions R_i and R_j are not merged when the algorithm process the first separating face, then they will never merge during the algorithm.

Proof. We consider two regions R_1 and R_2 in the topological map that are multi-adjacent. Let be F_p, $p \in [1..k]$, the different faces incident to both R_1 and R_2. We suppose that $Ext(F_1) \leq Ext(F_2) \leq \ldots \leq Ext(F_k)$. We know that the external variation of the two regions is equal to the external variation of the face F_1.

Algorithm 1 processes the faces in increasing external variation order. The first face considered to merge R_1 and R_2 is thus F_1.

Suppose $Ext(F_1) > MInt(R_1, R_2)$ then the two regions R_1 and R_2 are not merged according to the criterion. Given this hypothesis, we want to know if it

is possible for the two regions to merge when considering the other faces. There are two cases to consider.

If $MInt(R_1, R_2)$ does not change, the process of the other faces can not leads to the merging of the two regions since they have a greater external variation.

If $MInt(R_1, R_2)$ increases its value between the treatment of two faces F_i and F_j ($i \geq 1$ and $j > i$) we have to show that this modification will not alter the result of the algorithm.

Let suppose $Int(R_1) + \tau(R_1) < Int(R_2) + \tau(R_2)$ and thus $MInt(R_1, R_2) = Int(R_1) + \tau(R_1)$ (this supposition can be made without lost of generality, eventually by renaming the two regions). So $MInt(R_1, R_2)$ increases only if $Int(R_1) + \tau(R_1)$ increases. This is only possible if R_1 is merged with another neighboring region which will be called R_3.

To merge R_1 and R_3, we need a face F so that $Ext(F) \leq MInt(R_1, R_3)$. Since the list is sorted $Ext(F_1) \leq Ext(F)$, and by definition of $MInt$, $MInt(R_1, R_3) \leq Int(R_1) + \tau(R_1)$. Our starting hypothesis gives that $Ext(F_1) > Int(R_1) + \tau(R_1)$. When combining these assertions, we have: $Ext(F) \geq Ext(F_1) > Int(R_1) + \tau(R_1) \geq MInt(R_1, R_3)$. Thus $Ext(F_1) > MInt(R_1, R_3)$, the region R_1 and R_3 will not be merged and $Int(R_1) + \tau(R_1)$ will not increase. This property shows that R_1 will never be merged during the algorithm. So $MInt(R_1, R_2)$ does not increase and each face F_p, $p \in [2..k]$, $Ext(F_p) > MInt(R_1, R_2)$. The region R_1 and R_2 will never merge. □

Thus Prop. 1 is verified. Algorithm 1 gives the same result as the one proposed in [17].

4 Results and Analysis

Let us talk about the complexity of the segmentation. Let be $|F|$ the number of faces of the topological map and $|D|$ the number of darts of the topological map. We know that $|D| > |F|$ since there is at least one dart for each face of the map.

The initialization of the list of faces needs to run through all the darts. The marking operation as well as the checking of a mark are constant time operations. We perform $|F|$ insertions in the list and then we sort it which leads to a complexity for this operation in $O(|D| + |F| * \log(|F|))$. The union and find operations on disjoint-set are quasi-linear according to [16] and thus we consider it linear for our usage.

Then each face is considered in a loop executed $|F|$ times. The extraction of the list is a constant time operation as well as the retrieval of the belonging region of a dart and the β_3 operation. With the previous hypothesis on union-find trees, checking if the two regions are not merged is a constant time operation. The computation of the equation is also a constant time operation because we store the needed values on regions and faces. The access to these values is a constant time operation. The symbolic merging corresponds to the union operation on disjoint-sets and thus we consider it constant. So this part is performed in $O(|F|)$.

The last step of this algorithm is the application of the global merging operation on the topological map. We use the complexity given in Sect. 2.3 which is $O(|D| + |S|)$ where $|S|$ is the number of surfels of deleted faces.

This gives the complexity for the whole segmentation algorithm to $O(|D| + |F| * \log(|F|) + |S|)$. The operation depends on the number of darts of the whole topological map, the number of faces of the topological map and the number of surfels belonging to deleted faces.

We have written an application which allows to segment an image in the 3D topological map framework. We have measured several values during the segmentation of three images (which are medical brain images). In order to decrease the memory needed by the representation of the image, we pre-segment the image during the extraction of its topological map. Table 1 shows the initial and final number of regions for each of the 3 images so as it gives an idea of the effectiveness of the segmentation process. We have also collected the processing time of the three steps of the segmentation.

Table 1. Image information and processing time for the three steps of the segmentation algorithm (segmentation threshold $k = 5000$ and pre-segmentation $p = 3$).

Image	Img1	Img2	Img3
Size	256x256x44	256x256x111	256x256x124
Initial Regions	147924	431486	310421
Remaining Regions	10121	30179	22523
List initialization	2.93s	8.72s	6.25s
Symbolic merge	0.46s	1.30s	0.97s
Global merge	5.50s	14.99s	11.69s
Total	8.89s	25.10s	18.92s

The fastest step is the symbolic merging since it only manipulates regions by using union-find trees and the sorted list of faces. The list initialization is a slow step as it depends on the topological map size. The slowest part is, as expected, the region merging operation since it heavily depends on the number of darts in the topological map. The total processing time is nevertheless suitable for a use of the algorithm in the image analysis framework.

We present a segmentation result obtained on Img1. Figure 3 shows one slice of the medical image on left side. The image on the right represents the labeled view of the slice after the segmentation process. This view allows to identify all the resulting regions of the image since they have different colors. We could observe that most of the brain is represented by only one region whereas the skull is composed of many regions. The segmentation needs to be tuned in order to produce more accurate results depending on the problem. For example, a low segmentation threshold leads to an over-segmentation which could be used in an image processing chain as the initial input.

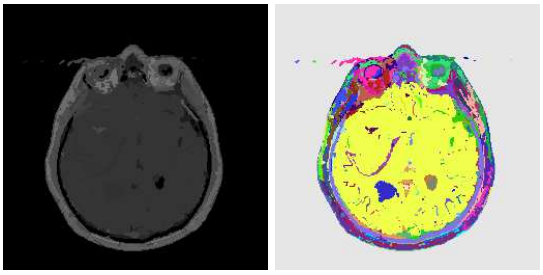


Fig. 3. A slice of a 3D brain image (256 x 256 x 44 gray image) and the segmentation result produced by the algorithm on the same slice ($k = 5000$, $p = 3$).

5 Conclusion

In this paper, we have presented the implementation of an existing bottom-up segmentation process in the 3D topological map. This work shows that the topological map is a suitable data structure to represent 3D images within the image processing framework.

We have detailed the homogeneity criterion used in the segmentation operation. It relies on intensity differences between neighboring voxels of the image. The aim of the segmentation process is to provide a set of homogeneous regions according to the criterion such as the merge of two regions produces another region which is not homogeneous.

The segmentation algorithm is divided in two main parts. Firstly we aim to produce a symbolic segmentation of the image, by handling its regions on a high level, and computing the criterion step by step during the merging process. Secondly, we use the existing operation of region merging in order to reflect the symbolic changes in the topological map.

We then proved that the modifications made to the initial segmentation algorithm do not modify its initial properties. We have analyzed the complexity of the whole segmentation process. It depends on the number of darts $|D|$, the number of faces $|F|$ and the number of surfels $|S|$ of the topological map. The complexity is given by the relation $O(|D| + |F| * \log(|F|) + |S|)$.

We have made some experiments on the segmentation process. Obtained processing times prove that the most expensive part of the algorithm is the region merging one and show that 3D topological maps can be used in real image processing applications.

The following step is to improve the segmentation process by taking advantage of the topological and geometrical information stored by the 3D topological map. We intend to use these information in addition with the existing criterion to obtain results that not only depend on the homogeneous criterion but also take into account the shape of regions as well as the adjacency relations between regions. For example one topological criterion may disallow the creation of dou-

ble torus region if no such object could exist in the image. This future work will show the advantage of using the topological map for 3D image segmentation.

References

1. Rosenfeld, A.: Adjacency in digital pictures. Information and Control **26** (1974) 24–33
2. Kropatsch, W., Macho, H.: Finding the structure of connected components using dual irregular pyramids. In: Discrete Geometry for Computer Imagery. (1995) 147–158, *invited lecture*
3. 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 (2001) 64–73
4. Braquelaire, J.P., Domenger, J.P.: Representation of segmented images with discrete geometric maps. Image and Vision Computing **17** (1999) 715–735
5. Damiand, G., Bertrand, Y., Fiorio, C.: Topological model for two-dimensional image representation: definition and optimal extraction algorithm. Computer Vision and Image Understanding **93** (2004) 111–154
6. 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 (1996) 151–162
7. Braquelaire, J.P., Brun, L.: Image segmentation with topological maps and interpixel representation. Journal of Visual Communication and Image Representation **9** (1998) 62–79
8. Haxhimusa, Y., Ion, A., Kropatsch, W., Brun, L.: Hierarchical image partitioning using combinatorial maps. In Hanbury, A., Bischof, H., eds.: 10th Computer Vision Winter Workshop. (February 2005) 43–52
9. Damiand, G., Resch, P.: Split and merge algorithms defined on topological maps for 3d image segmentation. Graphical Models **65** (2003) 149–167
10. Felzenszwalb, P.F., Huttenlocher, D.P.: Image segmentation using local variation. In: Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on. (1998) 98–104
11. Lienhardt, P.: Topological models for boundary representation: a comparison with n-dimensional generalized maps. Computer-Aided Design **23** (1991) 59–82
12. Khalimsky, E., Kopperman, R., Meyer, P.: Boundaries in digital planes. Journal of Applied Mathematics and Stochastic Analysis **3** (1990) 27–55
13. Damiand, G.: Définition et étude d'un modèle topologique minimal de représentation d'images 2d et 3d. Thèse de doctorat, Université Montpellier II (2001)
14. Dupas, A., Damiand, G.: Comparison of local and global region merging in the topological map. In: International Workshop on Combinatorial Image Analysis. Volume 4958 of Lecture Notes in Computer Science. (2008) 420–431
15. Cormen, T.H., Leiserson, C.E., Rivest, R.: Introduction to Algorithms. MIT Press (1990)
16. Tarjan, R.: Efficiency of a good but not linear set union algorithm. Journal of the ACM **22** (1975) 215–225
17. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International Journal of Computer Vision **59** (2004) 167–181