

# Topological Map: Minimal Encoding of 3d Segmented Images

Yves Bertrand<sup>1</sup>, Guillaume Damiand<sup>2</sup> and Christophe Fiorio<sup>2</sup>

<sup>1</sup>IRCOM-SIC, SP2MI, BP30179, F-86962 Futuroscope Cedex

<sup>2</sup>LIRMM, 161 rue Ada, F-34392 Montpellier Cedex 5

bertrand@sic.sp2mi.univ-poitiers.fr, {damiand, fiorio}@lirmm.fr

## *Abstract*

*In this paper we define formally and completely the 3d topological map introduced in [5]: a model which encodes totally and minimally all the topological information of a 3d image. In particular, we focus on the problem of disconnections induced by the constructive definition based on several levels of maps. This multilevel representation is more or less a graph pyramid in sense that each level can be compute from the previous one in term of merge operations. Furthermore, algorithms extracting these maps from a segmented image have been given in [5] and have been implemented and tested in practical applications.*

## **1. Introduction**

Before image analysis or processing, it's necessary to group all information contained in images: this is regions segmentation which is the first step of any image processing. After this first step, analysis process can be run in order to try to recognize objects in the image. Some interactive or semi-interactive process can also be applied, for instance user can select some regions to be merged. But all these process need one data structure which encodes all the information resulting of regions segmentation, and which allows efficient implementation of such algorithms. Such a data structure should characterize the objects it encodes, should be single, minimal and invariant for rigid transformations (rotation, translation, scale, local deformation). Moreover, one other important constraint is the data size of the structure. Indeed, we work in 3d, and a too large data size can prevent using the model.

We present in this paper a model which fulfil all these properties. This model encodes all topological information of images and allows efficient algorithms for image processing. Moreover, an embedding which encodes objects geometry can be easily added. It characterizes objects in respect of the constraints presented above. The minimality of our model ensure the important property that two topological equivalent objects have the same and single representation.

Geometrical modelling is used to represent 3d objects and several data structures have been defined. For example, [4] use a boundary representation based on planar or combinatorial map (first described in [12, 19]). [17] has shown that combinatorial maps encode all topological information of any subdivision of a  $n$ -d space. For this reason, combinatorial maps derived structures have often been used to solve the problem of encoding two dimensional images, for instance the frontier topological graph [14], the topological maps [8, 6] or dual graphs [15]. But either these structures are hardly extendable in 3d, as the frontier topological graph and dual graphs, or aren't minimal and depend on the objects geometry, as the border map [5] and the extension of the topological map proposed in [7].

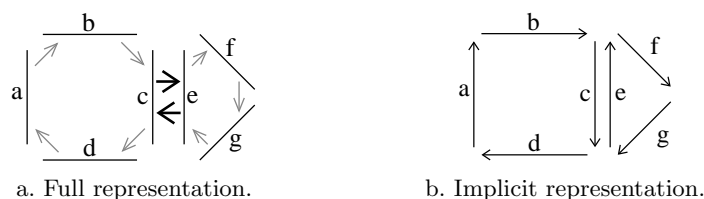
In order to present our model, we introduce a new notion of simplification level. These levels allow us to progressively simplify our structure, and finally obtain the minimal and single structure we look for. This special construction gives in a straight way an algorithmic definition of the topological map, which is the minimal map encoding an image. This definition can be easily extended in  $n$ -d by generalizing the construction process. Moreover, some intermediary levels of map introduced are interesting. Low levels are simpler to extract from image but are more consuming in memory space. This is the reverse for the higher levels. Moreover, algorithms are simpler to write for low levels, but their running time are often more long due to the greater number of basic elements to look at and to update. These different levels of simplification allow to choose in between different sorts of maps according to the specification of our application. We give Section 2. a short recall of combinatorial maps. Then, Section 3. we show how they will be used to encode segmented images. Section 4. presents the 3d topological map and our notion of simplification levels. In Section 5. we show briefly one of our implementation of this model and present some results. Then Section 6. we conclude the paper.

## 2. Combinatorial maps recall

Combinatorial maps are a mathematical model of space subdivision representation. A subdivision is a partition of a  $n$  dimensional topological space into  $(n + 1)$  subsets whose elements are *cells* of dimension  $0, 1, 2, 3, \dots n$ . Border relations are defined in between these cells. We say that two cells are *incident* when one belongs to the border of the second, and that two  $i$ -cells are *adjacent* if they are both incident to the same cell. Combinatorial maps encode all the subdivisions and the incident relations, and so represent all the topology. They are defined formally for any dimension, and we say  $n$ -map for a  $n$  dimensional

combinatorial map. The  $n$ -map can encode all orientable manifold subdivisions of a  $n$  dimensional space. They were generalized in [16, 17] in order to encode all  $n$  dimensional, orientable or not, subdivisions of  $n$  space.

The definition of  $n$ -map is based onto a single basic type of abstract elements, called *darts*, on which are defined a permutation  $\beta_1$  (a bijection of  $S$  on  $S$ ) and  $n - 1$  involutions  $\beta_2 \dots \beta_n$  (permutations  $f$  such that  $f = f^{-1}$ ).



**Fig. 1.** Two representations of the same combinatorial map.

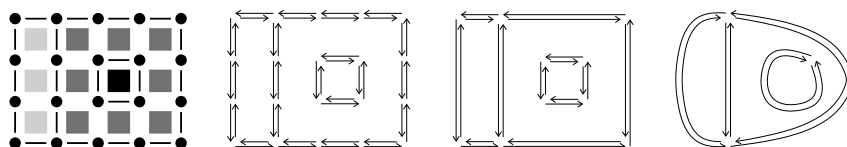
Each permutation  $\beta_i$  connects two  $i$ -cells. Furthermore, all topological cells of the space are represented implicitly, and we can easily find them all by a simple and efficient cover algorithm. We can see an example of a 2-map Figure 1.a. On this figure, darts are represented by straight line segments,  $\beta_1$  by grey arrows and  $\beta_2$  by thick black arrows. We can see on this figure, that for instance,  $\beta_1(a) = b$  and  $\beta_2(c) = e$ .  $\beta_1$  connects an edge and the following edge on the same face, and  $\beta_2$  connects two faces which are incident to the same edge. We can draw the same map without  $\beta_i$  explicit representation, as we can see Figure 1.b where each  $\beta_i$  can be deduced by the drawing configuration.

Combinatorial maps encode only objects topology, and not their geometry. But it's very easy to add some geometry elements to some (even all) topological cells of the combinatorial map: this operation is call *embedding*. The simplest embedding of a combinatorial map, and also the most used, consist at linking each topological vertex of the map with the coordinates of an Euclidean space point. This embedding is of course valid only when all the edges of the map are straight line segments, which is not already true.

### 3. Images, borders and map

We are only concerned with modelling of segmented images. As the combinatorial maps model objects by their borders, we must look at the borders of the regions of our segmented images. For that, we use interpixel topology [18, 1]

which define totally and formally borders in segmented images. Moreover, combinatorial maps encode the same space subdivisions as those used by interpixel topology.



**Fig. 2.** A regions segmented image, and some maps which it models.

We can see Figure 2, a two dimensional regions segmented image, represented with its interpixels boundaries, and three combinatorial maps which encodes the boundaries of this image. A map encodes the boundaries of an image if each edge of the map represent a part of an image boundary. Each of these maps encode the same image. This can be easily proved by computing all the topological characteristics of these maps, and verify that they are equivalent. But to have a complete definition of a structure encoding all information contained in an image, we have to define, first, a topological model, secondly, join it with an embedding model, and then give algorithms extracting those models from a region segmented image. We only present in this paper, the topological aspect of our work, due to a lack space. We can see [3] for an optimal extraction algorithm of our structure, and [13] for topological proof of objects validity.

#### 4. Topological Map in dimension three

We have seen previously that several different maps can encode the same object. We want to define the map which will characterize this object in a minimal way. Our approach consists to start with a complete map, made up of cubes sewn together. We then progressively simplify this map by merging two elements while the topology of the represented objects doesn't change. The main advantage of this method is its simplicity which allow us to easily extend it in dimension  $n$ . We just have to find what mergings we have to do. We call the starting map the level 0 map, and define each level from the previous one by a particular sort of merging.

In the following we will say  $i$  dimensional merging for the operation which takes two different  $i$  dimensional adjacent objects and make a simple object which is

the union of the two. These merging operations can be easily described for all dimension in the framework of combinatorial maps. [13] can be seen for more explanation about merging operations in combinatorial map.

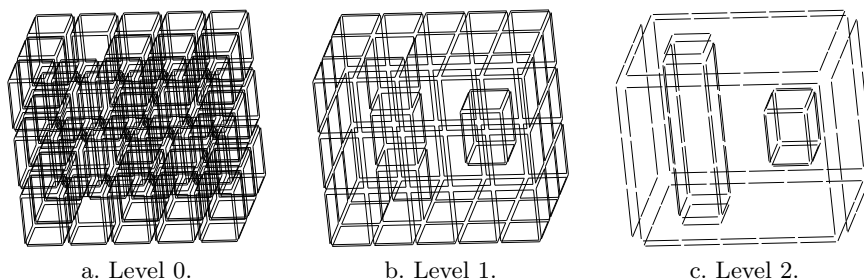
In dimension three, we have three different types of merging: the volume merging (merging of dimension three), the face merging (merging of dimension two) and the edge merging (merging of dimension one). The starting map for our merging operations is the level 0 map.

**Definition 1.** The level 0 map corresponding to an  $n \times m \times k$  voxels image, is the map having  $n \times m \times k$  cubes  $\beta_3$  sewn in between them, each cube corresponding to a voxel.

Of course, each cube is made up 6 faces  $\beta_2$  sewn between them and each face being made up 4 darts  $\beta_1$  sewn in between them. We begin by merging volumes belonging to the same region.

**Definition 2.** The level 1 map is the map obtained from the level 0 map by merging each couple of adjacent volumes belonging to the same region.

For the level 1 map, as we can see Figure 3.b, the volume merging remove all faces being inside a same region. We have now only faces incident to two different regions: those which encode the borders of the image. Note that in



**Fig. 3.** Levels 0, 1, and 2 for a three dimensional image.

the first figures we don't represent face orientation in order to keep the figure readable. For the same reason, we just show half of darts of the map. Indeed, each face should be duplicated by another face sewn by  $\beta_3$  and representing the same border seen by the other region.

We can see on this example that this map is single and totally defined by its construction algorithm. We can also prove that the order of the merging doesn't

change the map obtained, which ensure the unicity of this map. Each region of this level 1 map is now encoded by at least one map. Indeed, a region always has one exterior border (except the infinite region which has only one interior border) but can have several interior borders. The paving has an interior border due to the fact that there is a region (the cube) totally included into it. The important information that these two borders belong to the same volume is lost. This is an important topological information and it must be kept. To solve this problem, we add to our structure an inclusion tree which recall this information. With this tree, we can retrieve easily the inclusion topological property. Now, we have to look for face merging in order to obtain the next simplification level.

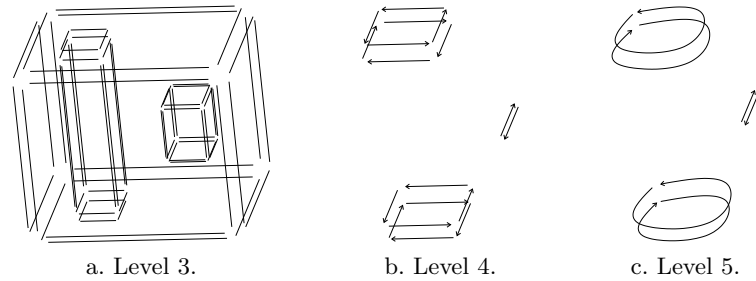
**Definition 3.** The level 2 map is the map obtained from the level 1 map by merging each couple of adjacent and coplanar faces incident to a degree two edge.

We can see Figure 3.c the level 2 map of our example. We merge only the face incident to a degree two edge, because there are the only ones which can be merged without losing topological information. Moreover, we chose, initially, to merge only coplanar faces in order to be able to embed edges by a straight line segment. This allow us to define a simple embedding by associating with each topological vertex of the map the coordinates of an Euclidean point, which is not true if we merge non-coplanar faces. But this face merging can lead to face disconnections, as we can see on our example for the bottom and up faces. Two borders of the same face have been disconnected. But this information is a topological information and must be kept. For that, we add another structure which, for each face of the 3-map, keep the list of borders of this face, each border being represented by one dart. For our example, the upper face have two darts in its list of borders, one for exterior border and one for interior. If each dart of the map know the face it belongs to, we can reconstruct each face correctly and compute topological characteristics for each face, even those who have disconnected borders.

We can remark on our example that each edge of the map have the same length, which correspond to the length of a voxel. We are going to do some edge merging in order to remove non-necessary darts.

**Definition 4.** The level 3 map is the map obtained from the level 2 map by merging each couple of adjacent and lined up edges incident to a degree two vertex.

We can see the level 3 map of our example Figure 4.a. The merging of the lined



**Fig. 4.** Levels 3, 4 and 5.

up edges incident to degree two vertex can't lead to disconnection. In fact, this merging can be considered as a simple extension of lined up edges of the map, when we can. Until this level, we have only merge lined up cells, and so we have all edges of this level 3 map which can be embedded by a straight line segment. The embedding of this map can be simply defined by its vertices. This is this level map which is called *border map* in [5], and which is fluently used by the geometrical modelisation world, due to its simplicity of implantation and using. But this level isn't really satisfying. It isn't stable for rigid transformations (two equivalent images by these transformations don't have the same map) and the number of darts of this map depends on the geometry of the modeled objects. This level can't characterize in a single way objects its encode. To solve this problem, we should do the same merging for the other cells we have already done for lined up cells.

**Definition 5.** The level 4 map is the map obtained from the level 3 map by merging each couple of adjacent and non coplanar faces incident to a degree two edge.

As for the coplanar faces merging, the merge of non coplanar faces can also involves face disconnections, but this problem is solve with the same solution already used for level 2, with the face structure and its list of borders. We can see Figure 4.b the level 4 map of our example. We can remark that it remains just a few darts. But this few number of darts contains, with the additional structure, all the same topological information than the 3 previous levels. For instance, the cube included into the paving is now represented just by two darts  $\beta_2$  sewn between them. The information associated is: "there is only one face between this two objects, this face doesn't have border and doesn't have adjacency face". There is not lost of topological information. Now we just have to look at the edge merging to define the last level: the topological map.

**Definition 6.** The level 5 map is the map obtained from the level 4 map by merging each couple of adjacent and non lined up edges incident to a degree two vertex.

This merging can't lead to disconnection and modification of topological characteristics. We can see the level 5 map for our example image Figure 4.c. It is just made of six darts. Two for the cube, and two for the different faces with their two distinct borders. One face represents the exterior face of the paving, and the other the interior face which crosses this paving. This map contains all the topological information of the corresponding image.

We can prove that this level 5 map is topologically equivalent to the preceding ones by computing and comparing all topological characteristics. Moreover, this map is minimal for the merging operations: we can't merge two adjacent cells without lost topological information. This is why we call this last level the *topological map*. This map verify all our needs: it's single, minimal, invariant for rigid transformation and encode all topological information of the image. All these properties can easily be proven. We can't give here these proofs due to lack of available place.

## 5. Experimentation and analysis

We have developed a complete software which takes a three dimensional image and computes directly any level of map encoding this image. This software is based upon the linear and in one scan of the image algorithm present in [3]. Moreover, we have implemented the topological model presented in this paper as well as an embedding model which encode geometry of objects in the image. We have tried several different embeddings to compare the different results in term of memory space consuming and in term of computation time. One interesting embedding is a hierarchical structure, where each dimension  $i$  cell of the topological map was embedded by a dimension  $i$  map, which is of course embedded itself with this hierarchical principle. This give an interesting recursive definition of the maps. With this embedding, we extract topological map of an  $512 \times 512 \times 180$  image in about 10 minutes<sup>1</sup> of computation, and the final model need about 300 Mb which stays reasonable in regard of actual PC.

Another advantage of our different simplification levels is that the memory space necessary to our structure decrease more and more for each higher level of maps we computed. This is a very important advantage, because the space

---

<sup>1</sup> With the input/output necessary time.



memory needed by the first level is, for an  $512 \times 512 \times 180$  image, about one giga-bytes and can't be compute on standard computer. We can look at [3] for more details on the need of memory of the different levels of map.

## 6. Conclusion

In this paper, we have presented a model which encodes all topological information of three dimensional objects: the topological map. This structure is minimal, stable for rigid transformations, and give a single characterization of three dimensional objects.

To give a formal definition of this map, we have introduced a notion of simplification level which allows us to define progressively and easily each level of map from the previous one. Moreover, some levels are interesting and actually used by some particular applications. Indeed, if we haven't memory space constraints and if we want to develop immediately a model for little image, low level should be used. On the other hand, for big image, we must work with a higher level which is less costly in memory space, and more fast in computing time for all algorithms. But higher levels are more difficult to implement. These different levels of map are interesting also by the large choice they give.

Our simplification levels can be look as map pyramids [11] where the contraction operation would be replaced by our merging. This is one of our present interest for future works. Moreover, the topological map can be the starting point of several simplifications, but which would be with loss of information. This would give a multi-scale representation.

At last, these levels were successfully tested in a software which extracts any level from a three dimension image. We now must study, as done in two dimension in [9, 10, 2], operations acting on our topological map. When this operations will be defined, we will be able to make some high level treatments. Our main goal is to be able, with the help of the maps, to refine automatically the segmentation of an image, and to have an interactive software for manipulation of three dimensional segmented images.

## References

- [1] E. Ahronovitz, J.-P. Aubert, and C. Fiorio, The star-topology: a topology for image analysis, In *5<sup>th</sup> Discrete Geometry for Computer Imagery*, pp. 107–116. Groupe GDR PRC/AMI du CNRS, september 1995.
- [2] E. Ahronovitz, C. Fiorio, and S. Glaize, Topological operator on the topological graph of frontiers, In *Discrete Geometry for Computer Imagery*, number 1568 in LNCS, pp. 207–217, Marne-la-Vallée, France, 1999.

- [3] Y. Bertrand, G. Damiand, and C. Fiorio, Topological encoding of 3d segmented images, In *Discrete Geometry for Computer Imagery, DGCI'2000*, number 1953 in LNCS, pp. 311–324, Uppsala, Sweden, december 2000.
- [4] Y. Bertrand and J.-F. Dufourd, Algebraic specification of a 3d-modeler based on hypermaps, *CVGIP: Graphical Models and Image Processing*, 56(1), pp. 29–60, january 1994.
- [5] Y. Bertrand, C. Fiorio, and Y. Pennaneach, Border map: a topological representation for nd image analysis, In *Discrete Geometry for Computer Imagery*, number 1568 in LNCS, pp. 242–257, Marne-la-Vallée, France, 1999.
- [6] J.-P. Braquelaire and L. Brun, Image segmentation with topological maps and inter-pixel representation, *Journal of Visual Communication and Image Representation*, 9(1), pp. 62–79, march 1998.
- [7] J.-P. Braquelaire, P. Desbarats, J.-P. Domenger, and C. Wüthrich, A topological structuring for aggregates of 3d discrete objects, In *Workshop on Graph based representations*, Austria, may 1999. IAPR-TC15.
- [8] L. Brun, *Segmentation d'images couleur à base Topologique*, Thèse de doctorat, Université Bordeaux I, décembre 1996.
- [9] L. Brun and J.-P. Domenger, A new split and merge algorithm with topological maps and inter-pixel boundaries, In *The fifth International Conference in Central Europe on Computer Graphics and Visualization*, february 1997.
- [10] L. Brun, J.-P. Domenger, and J.-P. Braquelaire, Discrete maps : a framework for region segmentation algorithms, In *Workshop on Graph based representations*, Lyon, April 1997. IAPR-TC15, published in *Advances in Computing* (Springer).
- [11] L. Brun and W. Kropatsch, Pyramids with combinatorial maps, Technical Report 57, Institute of Computer Aided Design, Vienna University of Technology, lstr. 3/1832,A-1040 Vienna AUSTRIA, december 1999.
- [12] R. Cori, Un code pour les graphes planaires et ses applications, In *Astérisque*, volume 27. Soc. Math. de France, Paris, France, 1975.
- [13] G. Damiand and P. Lienhardt, Fusions dans les cartes généralisées de dimension 3 : évolution des caractéristiques topologiques, Rapport de recherche, LIRMM, 161 rue Ada – 34392 Montpellier Cedex 5 – France, 2001.
- [14] C. Fiorio, A topologically consistent representation for image analysis: the frontiers topological graph, In *6th International Workshop, DGCI'96*, number 1176 in LNCS, pp. 151–162, Lyon, France, November 1996.
- [15] W. Kropatsch, Building irregular pyramids by dual-graph contraction, *Vision, Image and Signal Processing*, 142(6), pp. 366–374, december 1995.
- [16] P. Lienhardt, Subdivision of n-dimensional spaces and n-dimensional generalized maps, In *5<sup>th</sup> Annual ACM Symposium on Computational Geometry*, pp. 228–236, Saarbrücken, Germany, 1989.
- [17] P. Lienhardt, Topological models for boundary representation: a comparison with n-dimensional generalized maps, *Computer Aided Design*, 23(1), 1991.
- [18] A. Rosenfeld, T. Kong, and A. Wu, Digital surfaces, *CVGIP: Graphical Models and Image Processing*, 53(4), pp. 305–312, july 1991.
- [19] W. Tutte, *Graph Theory*, Cambridge University Press, 1984.