

# Using 2D Topological Map Information in a Markovian Image Segmentation

Guillaume Damiand, Olivier Alata, and Camille Bihoreau

IRCOM-SIC, UMR-CNRS 6615 - bât. SP2MI, Bvd M. et P. Curie  
BP 30179, 86962 Futuroscope Chasseneuil Cedex, France  
{damiand,alata}@sic.univ-poitiers.fr

**Abstract.** Topological map is a mathematical model of labeled image representation which contains both topological and geometrical information. In this work, we use this model to improve a Markovian segmentation algorithm. Image segmentation methods based on Markovian assumption consist in optimizing a Gibbs energy function. This energy function can be given by a sum of potentials which could be based on the shape or the size of a region, the number of adjacencies, . . . and can be computed by using topological map. In this work we propose the integration of a new potential: the *global linearity of the boundaries*, and show how this potential can be extracted from the topological map. Moreover, to decrease the complexity of our algorithm, we propose a local modification of the topological map in order to avoid the reconstruction of the entire structure.

**Keywords:** Markovian segmentation, topological maps, region segmentation, boundaries linearity.

## 1 Introduction

Topological maps were studied since several years in 2D [1,2,3] and more recently in 3D [4,5,6,7]. Indeed, a topological map represents a labeled image with interesting properties: it is minimal in number of atomic elements (darts); it is complete, it represents both topology and geometry of the image; and it is unique. For these reasons, topological map allows to retrieve most of the information which may be required by an image processing algorithm with a low computational cost. So the topological map seems to be a good tool to define efficient image processing.

The main objective of image segmentation is to partition the pixels of an image. In such context, there are two main research axes: the boundary and the region based methods [8]. Fusions of both approaches have also been proposed. Image segmentation can be used in many applications like content-based image retrieval, computer-aided medical diagnostic, recovery of shape information from an image, . . .

In previous works [9,10,11], topological map was often used in split-and-merge algorithms, since it is well suited to implement efficiently such methods. In this work, we present a new utilization of the topological map to improve a Markovian segmentation algorithm.

The image partition contains areas of pixels considered as homogeneous following some properties. Using Markov models and Monte-Carlo Markov Chain (MCMC) implementation like Simulated Annealing (SA) [12], the only properties used for aggregating pixels are often statistical properties [13,14,15,16]. Nevertheless, many other geometrical or topological information on the segmented or label field could be used: the Markovian assumption for the representation of the hierarchical field composed by an observation field and a label field leads to an unnormalized Gibbs distribution; the energy of the Gibbs distribution can be written as a sum of potential functions which is a powerful tool for the fusion of information; geometrical or topological information on the label field could then be integrated in potential functions. Our aim is then to find some potential functions based on geometrical or topological properties and to compute them thanks to the topological map. In this paper, we show how to favor the creation of regions with linear boundaries during the segmentation process.

We first present in Sec. 2 a brief recall on topological maps that are combinatorial maps extended to represent images. Then in Sec. 3 we introduce the Markovian image segmentation and show how potentials are integrated in such a process. In Sec. 4 we define our new potential used to favor linear boundaries, and present how this new potential is integrated with topological maps in Sec. 5. We give experimental results in Sec. 6, then we conclude and present some perspectives in Sec. 7.

## 2 Topological Maps

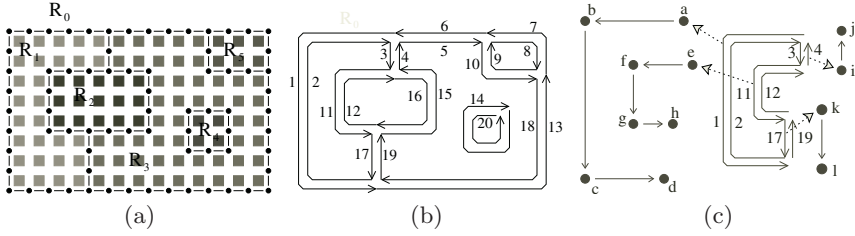
Topological maps are an extension of combinatorial maps [17,18] in order to represent in a unique and minimal way a labeled image. Indeed, combinatorial maps are a good model that allow to represent any orientable, quasi-manifold, closed subdivision in any dimension, but one object can be represented by different maps. We present here briefly the main notions of combinatorial maps and of topological maps (see [19,5] for more details).

Intuitively, a 2D combinatorial map is an extension of a planar graph that keeps the orientation of edges around each vertex. Each edge of the graph is divided in two parts. Basic elements obtained are called *darts* and are the unique atoms of the combinatorial map definition. A combinatorial map is an algebra composed by a set of darts that represent the elements of the subdivision, and 2 mappings defined of these darts that represent adjacency relations (this can be easily extended in  $n$ D, with  $n$  mappings).

We can see in Fig. 1 an image and the corresponding topological map. In this figure, there are 20 darts numbered from 1 to 20.  $\beta_1$  is a permutation<sup>1</sup> that connects a dart and the next dart of the same face. There are for example  $\beta_1(2) = 3$  and  $\beta_1(3) = 11$ .  $\beta_2$  is an involution<sup>2</sup> that connects the two darts belonging to the same edge. In our example, there are  $\beta_2(1) = 2$  (and since  $\beta_2$  is an involution also  $\beta_2(2) = 1$ ). When two darts  $d_1$  and  $d_2$  are such that

<sup>1</sup> A *permutation* on a set  $S$  is a one to one mapping from  $S$  onto  $S$ .

<sup>2</sup> An *involution*  $f$  on a set  $S$  is a one to one mapping from  $S$  onto  $S$  such that  $f = f^{-1}$ .



**Fig. 1.** (a) A 2D image drawn with its interpixel boundaries. (b) The corresponding topological map. Each dart is represented by a numbered arrow.  $\beta_1$  connects a dart and the next dart of the same face (drawn consecutively, orientation is represented with arrows).  $\beta_2$  connects two darts drawn parallel, close to each other, and with reverse orientations. (c) The embedding used in this work (partial representation). Each edge is linked with an 1D oriented curve. Only one dart of the two that composed an edge points to the 1D curve (liaisons are represented by dash arrows). The orientation of the curve is given by the orientation of this dart.

$\beta_i(d_1) = d_2$ , we say that  $d_1$  is  $i$ -sewn with  $d_2$ . We call  $i$ -sewing (resp.  $i$ -unsewing) the operation that connects two darts for  $\beta_i$  (resp. disconnects).

A topological map is a combinatorial map that represents a labeled image and that verifies particular properties. Indeed, this map is minimal, complete and unique. These properties lead to another characteristic of the topological map: each edge represents exactly an interpixel boundary between two regions of the image (this can be verified in Fig. 1). An interpixel boundary between two regions  $R_i$  and  $R_j$ , is the set of interpixel curves such that each line of these curves is incident to exactly one pixel of  $R_i$  and one pixel of  $R_j$ .

Combinatorial map represents the topological part of our model: all the cells of the space subdivision and all the adjacency and incidence relations. But it is also necessary to represent the geometry of the image. We call *embed* the operation that associates a geometrical model to a combinatorial map, and we speak about *embedding* to design this geometrical model.

There are many possibilities to embed a combinatorial map and the choice of one of them depends on the needs of each application. In this work, we link to each edge of the map an 1D oriented curve. This curve represents the geometry of the interpixel boundary associated with the edge. We can see in Fig. 1(c) this type of embedding for the map already presented in the previous figure (only a partial representation). Each 1D curve is described with a 1D combinatorial map (we have so a hierarchical model that facilitates its extension in higher dimension). Each vertex of these curves represents a point of an interpixel boundary, and each edge represents a maximal set of aligned lines.

### 3 Integration of Information in Markovian Image Segmentation

Let  $X = \{X_s, s \in S\}$  a family of random variables on a regular grid  $S$  which is a finite subset of  $\mathbb{Z}^2$  and an image  $x = \{x_s, s \in S\}$ , a realization of  $X$ . Each

$X_s$  has the same state space that can be  $\Omega_x = \{0, 1, \dots, 255\}$ , the set of gray pixel values, or  $\Omega_x = \mathbb{R}, \dots$  We now suppose that there is another random field  $L = \{L_s, s \in S\}$  called the label field with state space  $\Omega_l = \{c_1, c_2, \dots, c_K\}$  for which a realization will be  $l = \{l_s, s \in S\}$ . In our case, a label represents the region to which the pixel belongs.

From these definitions, we suppose that  $X$  and  $L$  are defined in a hierarchical way: for each  $l_s$ , we have a conditional model for  $X_s$  that can be, for example, an independently and identically distributed (i.i.d.) model or a 2D Gaussian Markovian model [13,15,16]. As the field  $X$  is supposed to be composed of  $K$  areas, we then have  $K$  probability laws, defined by the stochastic models chosen, describing the variations of gray pixel values in  $x$  for the different regions belonging to the  $K$  areas. Besides, such models allows the use of Maximum Likelihood-based algorithms. For example, in the case of the Gaussian i.i.d. mixture model,  $K$  and the parametric models,  $\theta_k = \{\mu_k, \sigma_k\}$ ,  $k = 1 \dots K$ , can be estimated with a Stochastic Expectation Maximization algorithm (SEM) [20].  $\mu_k$  and  $\sigma_k$  are respectively the mean and the standard deviation of the area  $k$ . In the following,  $K$  and  $\theta_k$ ,  $k = 1 \dots K$ , are then supposed to be known. At this step, we need now to estimate the segmented field,  $\hat{l}$ , or, in other words, the image partition.

The likelihood method defines the *a posteriori* law  $P(L = l/X = x)$  or  $P(l/x)$  as the probability to get one specific realization of the label field knowing the observation field. If the Markovian assumption is done about the  $(X, L)$  hierarchical field,  $P(l/x)$  can be written in a general form following the Gibbs distribution:

$$P(l/x) \propto \exp\left(-\frac{U(x, l)}{T}\right) \quad (1)$$

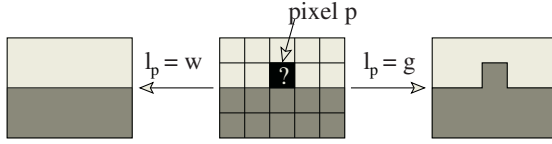
in which  $U$  is an “energy function”, depending on the observation and label fields, and  $T$ , the temperature.

Therefore, Maximum a Posteriori (MAP) estimation of  $l$ ,  $\hat{l}$ , consists in the minimization of  $U$  in order to maximize  $P(l/x)$ . SA methods have shown to be appropriate for such an optimization problem [12]. SA is an iterative process with decreasing  $T$ . For each  $T$ , a Gibbs sampler is done on the whole pixel of  $S$ : a label is sampled at each pixel following local probabilities of labels.

The energy function can be seen as a sum of weighted potentials, each one corresponding to a particular measure on the observation field or the label field. Potentials can be expressed locally, i.e. for each pixel, thanks to the Markovian assumption. Local probabilities of labels at each pixel can then be computed from these potentials. Therefore, the potential functions will allow us to integrate global information about the label field in a local probability.

## 4 The Global Linearity Potential (GLP)

In some regions, favoring geometric properties of boundaries during a segmentation algorithm can be of interest. As an example, aerial images of cities contain mostly regions with linear boundaries. In the following, we present a potential function that is able to take into account such property.



**Fig. 2.** An example of boundary pixel.

First of all, at a given iteration of the Gibbs sampler, the GLP (called  $\Phi_{GL}$ ) is calculated only on pixels belonging to a boundary, i.e. we exclude isolated pixels and pixels inside a region:

$$\Phi_{GL}(l_p) = \begin{cases} V(l_p), & \text{if } s \text{ belongs to a boundary} \\ 0, & \text{elsewhere} \end{cases} \tag{2}$$

$V(l_p)$  is the energy function associated with the value of  $l_p$ . For the example given in Fig. 2,  $V(l_p = w)$  is the energy when we consider the pixel  $p$  in the white region, and  $V(l_p = g)$  the energy when  $p$  is in the grey one. To favor the white case, we search for a function that gives:  $V(l_p = w) \leq V(l_p = g) \leq V(l_p = k) = 1$  with  $(w, g, k) \in \Omega_l^3$ ,  $w \neq g$ ,  $w \neq k$ ,  $g \neq k$ . To achieve this objective, we discretize each boundary in a succession of discrete segments. Therefore, we can choose  $V$  as follows, when the pixel  $s$  has the label  $l_p$ :

$$V(l_p) = \frac{n_b(l_p)}{\sum_{i=1}^{n_b(l_p)} l_i(l_p)} \tag{3}$$

where  $n_b(l_p)$  is the number of segments of the boundaries, and  $l_i(l_p)$ ,  $i = 1, \dots, n_b(l_p)$ , the length of the different segments. If we still consider the example in Fig. 2, it gives  $V(w) = \frac{1}{l_b} < V(g) = \frac{5}{l_b+2} < 1$ , with  $l_b$  the length in the case of a linear boundary;  $l_b > 1$  as we excluded isolated pixels.

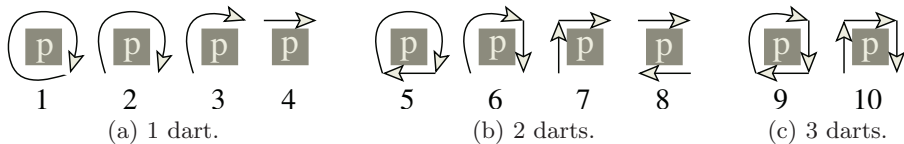
## 5 Integration of the GLP with Topological Maps

Since the SA is an iterative method, we compute at each iteration a topological map that corresponds to the current label field. Then, to compute the GLP in a pixel  $p$ , we proceed in four steps:

1. Test if  $p$  belongs to a boundary. Indeed, we compute the GLP only for boundary pixels, for other pixels GLP is equal to 0;
2. Compute in what regions  $p$  can be set. Indeed, since we do not consider the boundaries of isolated pixels, we can not set  $p$  is a region that leads to the creation of such pixel;
3. Modify locally the map to take into account the modification of the region of  $p$ . This optimization is necessary in order to avoid the entire reconstruction of the map for each pixel of the image;
4. Finally, compute GLP on this map by using a discretization algorithm.

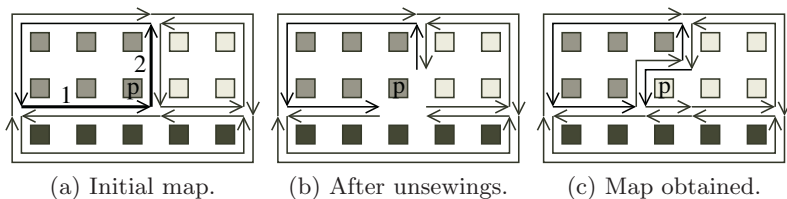
The first step can be easily achieved by testing the 4-neighbor pixels of  $p$ . When they are all in the same region as  $p$ ,  $p$  is not a boundary pixel and reciprocally.

For the step 2, we traverse the topological map in order to find all the darts that touch  $p$  (note that we can obtain the same result by looking at the regions of the 4-neighbor pixels of  $p$ , but darts computed here are going to be used in the following). We say that a dart touches a pixel when it represents an interpixel curve that contains a linel incident to the pixel. We can remark that only 0, 1, 2, 3 or 4 darts can touch a pixel. All possible configurations are given in Fig. 3 (We did not represent the cases when no dart and when 4 darts touch  $p$ ).



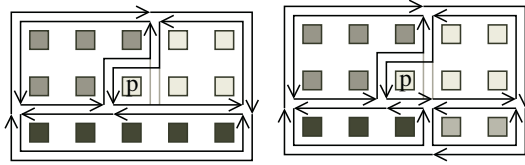
**Fig. 3.** The configurations of darts around a pixel, sorted in number of darts. Others configurations can be deduced from these ones by rotations or by changing the length of the darts.

Given  $p$ , retrieving all the incident darts can be done by traversing all the darts of the region of  $p$ , and for each dart by looking if the corresponding 1D curve contains a linel incident to  $p$  (by comparing coordinates). This can be performed in a linear complexity in number of linels that belong to the boundaries of the region of  $p$ . We present in Fig. 4(a) an image made of 3 regions (white, light grey and dark grey), and its topological map (in the same figure). The pixel  $p$  touches 2 darts (numbered 1 and 2), which corresponds to the case number 7 in Fig. 3.



**Fig. 4.** A configuration of darts around a boundary pixel, and local modifications performed to change the region of the pixel.

Then we can find all the regions around  $p$ , just by recovering the regions of the darts 2-sewn with darts that touch  $p$ . In Fig. 4(a),  $\beta_2(1)$  gives a dart that belongs to the dark grey region, and  $\beta_2(2)$  a dart that belongs to the white region.



**Fig. 5.** Two different maps with same initial configuration of darts around a boundary pixel when  $p$  belonged to the light grey region (initial darts drawn in light grey), and with different maps after modification.

The step 3 of the algorithm consists in affecting  $p$  successively into all these regions and modify locally the map in order to represent this modification. For that, we need to define for each case shown in Fig. 3 how to transform locally the map. First, we can remove cases numbered 1, 5 and 9 since we do not consider isolated pixels. For the same reason, we can remove the two cases not drawn in the figure where  $p$  touches 0 or 4 darts. We remove also the case numbered 8 because when we change the region of  $p$  this leads to topological modification of the map. With the GLP, we want to favor linear boundaries only by modifying locally the boundaries extract from the previous iteration. For this reason, we do not allow to change the region of  $p$  into a region when this leads to topological modification. So we still only have to consider cases numbered 2, 3, 4, 6, 7 and 10. We present in Fig. 4 modifications performed for the case 7, since we can not give here all the different algorithms and they are quite similar.

Starting from the map shown in Fig. 4(a), we change the region of  $p$  to the white region. This region touches the dart 2, so we first begin by cutting the face between the two darts 1 and 2 (see Fig. 4(b)). This is done by unsewing the two edges incident to the darts 1 and 2 and by decreasing their sizes. Then, we create two new edges, one which pass to the left of  $p$  and the second to the bottom. These edges are sewn with the darts that were previously sewn with the initial edges.

The map obtained is given in Fig. 4(c), but this is not the final result. Indeed, modifications performed here are done for the general case, and we do not obtain a topological map (the map in Fig. 4(c) is not minimal since there are some degree two vertices). We prefer to simplify the map after the local modification in order to propose a general algorithm and to decrease the number of different cases to consider. We can see in Fig. 5 two examples with the same initial configuration of darts around  $p$  when  $p$  belonged to the light grey region. For both configurations, putting  $p$  in the white region leads to the same map (those presented in Fig. 4(c)) since both local configurations of darts around  $p$  are the same. This is only during the simplification of the map (which merge edges around degree two vertices) that we perform different operation, and obtain so the two different maps shown in Fig. 5.

The last step of the GLP computation consists in retrieving, in the modified topological map, the two numbers used in the  $V(l_p)$  formula: the number of segments of the boundaries and the length of the different segments. For that, we use a discrete curve polygonalization algorithm of the pointels of each 1D

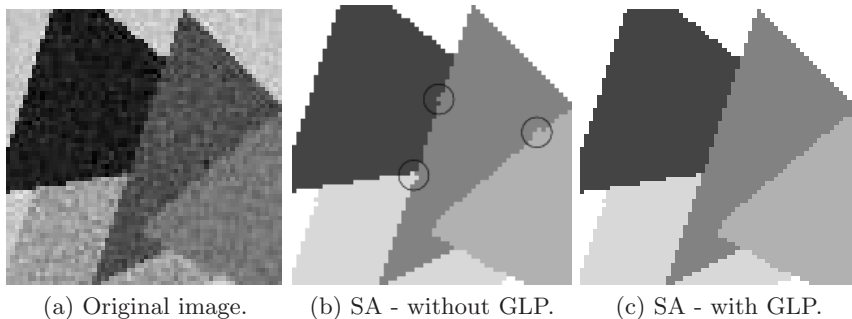


Fig. 6. GLP influence on a synthetic image.

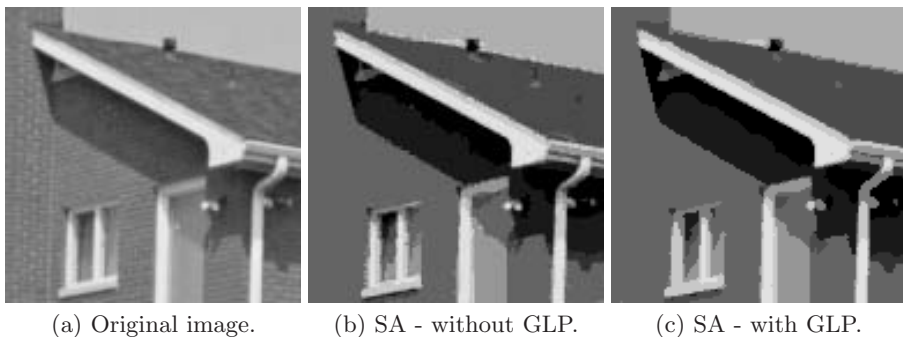


Fig. 7. GLP influence on a real image.

curves around  $p$  (with the algorithm presented in [21]). This gives the number of discrete segments of each boundaries around  $p$ . The length of each segment is simply computed with the Euclidean distance.

## 6 Experimental Results

The synthetic ( $64 \times 64$ ) image given in Fig. 6(a) has been used to test the method. This image is made up five Gaussian i.i.d. processes, with different means and variances, inside regions with linear boundaries. The weights between potentials<sup>3</sup> have been fixed in order to focus on the influence of the GLP. 200 iterations of the Gibbs sampler has been realized from  $T_0 = 2$  with the decreasing scheme:  $T_k = 0.99^k T_0$ . Figure 6(b) shows the result without using GLP and Fig. 6(c) with GLP. Both results are good but boundaries are better with the GLP. Moreover, the percentage of label errors is 0.3174 for Fig. 6(b) against 0.1709 for Fig. 6(c).

We can see in Fig. 7 one result obtained for a “real” image. We can observe the influence of the GLP by comparing Fig. 7(b) and Fig. 7(c). Boundaries are more straight with the GLP, even if the differences are not very visible. We need

<sup>3</sup> Besides the GLP, we used three potentials respectively based on the Gaussian probability law, the local label neighboring and the size of regions.



more experiments in order to fix the weights associated with each potential and so to show the interest of this potential in real applications.

## 7 Conclusion

In this work we have shown how the topological map can be used in order to improve an existing image processing: a Markovian segmentation algorithm. Indeed, topological map is a good model that allows to retrieve most of the information, both topological and geometrical, of the image in an efficient way. Moreover, this work shows that we can use this model in different algorithm and not only in split-and-merge approaches.

In other works, Markovian segmentation with MCMC implementation has been done essentially by using statistical properties. It was due to a lack of objects that allow to encode efficiently geometrical and topological properties of the label field. With topological map, we could propose new global potentials that allows to favor properties of label field. We show in this paper how to favor the global linearity of the boundaries.

It is possible now to extend this work in order to propose others potentials. Many others properties can be used, like the shape of a region or the number of adjacent regions. Our goal is to define a set of particular potentials that we can use or not, depending on the type of image to process. Moreover, the definition of algorithms to compute these potentials will probably lead questions on how to process particular operations on the topological map, like for the local modification. This is particularly interesting in order to improve our model and to propose new tools to deal with topological maps.

## References

1. Domenger, J.: Conception et implémentation du noyau graphique d'un environnement 2D1/2 d'édition d'images discrètes. Thèse de doctorat, Université Bordeaux I (1992)
2. 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
3. Paillancy, J., Jolion, J.: The frontier-region graph. In: *Workshop on Graph based representations*. Volume 12 of *Computing Supplementum.*, Springer (1997) 123–134
4. Braquelaire, J., Desbarats, P., Domenger, J., Wüthrich, C.: A topological structuring for aggregates of 3d discrete objects. In: *Workshop on Graph based representations*, Austria, IAPR-TC15 (1999) 193–202
5. Bertrand, Y., Damiand, G., Fiorio, C.: Topological encoding of 3d segmented images. In: *Discrete Geometry for Computer Imagery*. Number 1953 in *Lecture Notes in Computer Science*, Uppsala, Sweden (2000) 311–324
6. Braquelaire, J., Desbarats, P., Domenger, J.: 3d split and merge with 3-maps. In: *Workshop on Graph based representations*, Ischia, Italy, IAPR-TC15 (2001) 32–43
7. Damiand, G., Resch, P.: Topological map based algorithms for 3d image segmentation. In: *Discrete Geometry for Computer Imagery*. Number 2301 in *LNCS*, Bordeaux, France (2002) 220–231

8. Gonzales, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley (1993)
9. Brun, L., Domenger, J.: 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. (1997)
10. Brun, L., Domenger, J., Braquelaire, J.: Discrete maps : a framework for region segmentation algorithms. In: Workshop on Graph based representations, Lyon, IAPR-TC15 (1997) published in *Advances in Computing* (Springer).
11. Braquelaire, J., Brun, L.: Image segmentation with topological maps and inter-pixel representation. *Journal of Visual Communication and Image Representation* **9** (1998) 62–79
12. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **PAMI-6** (1984) 721–741
13. Bouman, C., Liu, B.: Multiple Resolutions Segmentation of Textured Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **13** (1991) 99–113
14. Kervrann, C., Heitz, F.: A Markov Random Field Model-based Approach to Un-supervised Texture Segmentation using Local and Global Spatial Statistics. *IEEE Trans. on Image Processing* **4** (1995) 856–862
15. Barker, S.A.: Image Segmentation using Markov Random Field Models. Phd thesis, University of Cambridge (1998)
16. Melas, D.E., Wilson, S.P.: Double markov random fields and bayesian image segmentation. *IEEE Trans. on Signal Processing* **50** (2002) 357–365
17. Jacques, A.: Constellations et graphes topologiques. In: *Combinatorial Theory and Applications*. Volume 2. (1970) 657–673
18. Cori, R.: Un code pour les graphes planaires et ses applications. In: *Astérisque*. Volume 27. Soc. Math. de France, Paris, France (1975)
19. Lienhardt, P.: Topological models for boundary representation: a comparison with n-dimensional generalized maps. *Computer Aided Design* **23** (1991) 59–82
20. Celeux, G., Diebolt, J.: The SEM Algorithm : a Probabilistic Teacher Algorithm Derived from the EM Algorithm for the Mixture Problem. *Computational statistics quarterly* **2** (1985) 73–82
21. Debled-Rennesson, I., Reveilles, J.P.: A linear algorithm for segmentation of digital curves. *International Journal of Pattern Recognition and Artificial Intelligence* **9** (1995) 635–662