

Combining Topological Maps, Multi-Label Simple Points, and Minimum-Length Polygons for Efficient Digital Partition Model

Guillaume Damiand, Alexandre Dupas, Jacques-Olivier Lachaud

▶ **To cite this version:**

Guillaume Damiand, Alexandre Dupas, Jacques-Olivier Lachaud. Combining Topological Maps, Multi-Label Simple Points, and Minimum-Length Polygons for Efficient Digital Partition Model. International Workshop on Combinatorial Image Analysis, May 2011, Madrid, Spain. pp.56-69, 10.1007/978-3-642-21073-0_8 . hal-00597507

HAL Id: hal-00597507
https://hal.science/hal-00597507

Submitted on 1 Jun 2011

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.

Combining Topological Maps, Multi-Label Simple Points, and Minimum-Length Polygons for Efficient Digital Partition Model*

Guillaume Damiand[1], Alexandre Dupas[2], and Jacques-Olivier Lachaud[3]

[1] Université de Lyon, CNRS, LIRIS, UMR5205, F-69622, France
guillaume.damiand@liris.cnrs.fr
[2] Université de Poitiers, CNRS, SIC-XLIM, UMR6172, F-86962, France
Inserm, Unit 698, 75018 Paris, France
alexandre.dupas@gmail.com
[3] Université de Savoie, CNRS, LAMA, UMR5127, F-73376, France
jacques-olivier.lachaud@univ-savoie.fr

Abstract. Deformable models have shown great potential for image segmentation. They include discrete models whose combinatorial formulation leads to efficient and sometimes optimal minimization algorithms. In this paper, we propose a new discrete framework to deform any partition while preserving its topology. We show how to combine the use of multi-label simple points, topological maps and minimum-length polygons in order to implement an efficient digital deformable partition model. Our experimental results illustrate the potential of our framework for segmenting images, since it allows the mixing of region-based, contour-based and regularization energies, while keeping the overall image structure.
Keywords: Topological Map; ML-Simple Point; Minimum-Length Polygon; Deformable Model; Interpixel Boundaries; Multi-Label Image.

1 Introduction

Energy-minimizing techniques are now widely spread approaches for segmenting images into meaningful regions. They express the problem as a balance between several energies: (i) *fit-to-data*: energies that express the consistency of the regions with the data, (ii) *regularization*: energies that enforce spatial coherence. Fit-to-data is either region-based (squared error for homogeneity) or contour-based (strong gradient). Regularization is generally enforced through boundary length penalization, sometimes curvature penalization or some a priori knowledge on the probability distribution of the partition model.

Energy minimization can be solved in the continuous world with iterative solving of partial differential equations, leading to snakes and geometric or

* Paper published in Proceedings of 14th International Workshop on Combinatorial Image Analysis, LNCS 6636, pp. 56-69, 2011. Thanks to Springer Berlin Heidelberg. The original publication is available at http://dx.doi.org/10.1007/978-3-642-21073-0_8

geodesic active contours [13, 3, 4] and Mumford-Shah approximation [19, 25, 20]. Energy minimization can also be expressed in the discrete world, generally as a graph-cut problem [1, 5] or a Markov Random Field (e.g., see [24]). To sum up, continuous approaches achieve optimality for specific energies. Discrete approaches are close to the optimal for a wider set of energies, to the price of less pertinent regularization energies: for instance, the length of the boundary is approximated as its staircase length.

As noted by Meltzer *et al.* [17] or Szeliski *et al.* [24], appropriate energies are more important than optimality for good segmentation results. In this paper, we therefore focus on defining an energy-minimizing framework for image segmentation that is as versatile as possible to define energies and relations between regions. We propose a purely discrete framework, which encodes both the topology and the geometry of an image partition. This digital partition is "deformable" in the sense that pixels may change their label during the minimization. The deformation is controlled so that the partition may not change topology during the deformation. This property is desirable in many image analysis applications, like segmentation with a priori knowledge on the output topology (as often in biomedical image segmentation) or atlas matching. This aspect has even been recognized by the level-set community: although the level-set principle allows arbitrary topology changes, several authors have proposed methods to reintroduce topology control in the segmentation (e.g., see [22]). This paper extends the work of [9] because our framework incorporates a new length penalizer with better regularization properties. Moreover, it combines in a single balance the energies: on one side, length penalization and alignments of boundaries with strong gradients define one energy term (in the same spirit as geodesic active contours [4]), on the other side region homogeneity.

Our objective is to define a digital deformable partition model. To that effect, several bottlenecks must be solved. First, we need to describe digital contours as closed paths, eventually intersecting themselves. We use *interpixel* topology [14] which allows to have in digital spaces good topological properties similar to the ones in continuous spaces. Second, we need to control the topology of the deformable partition. This is achieved thanks to *multi-label simple points* [11, 9]. Last, during deformations, we need a data structure to manage various queries like access to image statistics in region, estimation of the length of boundaries, etc. We use *topological maps* [8] to describe all the cells composing the deformable partition, and all the incidence and adjacency relations between these cells. Then, we use the *minimum length polygon* [18, 23, 21] criteria in order to estimate the length or evolving regions.

The paper is organized as follows. We recall the main notions of these papers in Section 2 (for further details, interested readers must refer to the original works). Then, we present the deformable process and the energies in Section 3. Illustrative experiments are given in Section 4. Future works are discussed in Section 5.

2 Preliminaries Notions

2.1 Pixels, Image and Regions

A pixel is an element of the discrete space $\mathbb{N}^2$. Two pixels $p = (x, y)$ and $p' = (x', y')$ are *4-adjacent* if $|x - x'| + |y - y'| = 1$. A *4-path* between two pixels p and p' is a sequence of pixels $(p = p_1, \ldots, p_k = p')$ such that each couple of consecutive pixels are 4-adjacent. A set of pixels S is *4-connected* if there is a 4-path between any couple of pixels in S having all its pixels in S.

An image is a set of pixels. In this work, we consider *labeled images*, *i.e.* images in which each pixel is associated with a label (any value). In labeled images, a *region* is a maximal set of 4-connected pixels. Thus, regions form a partition of the image: two different regions are disjoint, and the union of all the regions is equal to the image. Moreover, we consider a specific region R_0, called the *infinite region*, which is the complementary of the image. Two regions are said *4-adjacent* if there is a pixel in the first region that is 4-adjacent to a pixel in the second region.

2.2 Interpixel Topology and Cubical Complexes

In interpixel topology, we consider the cellular decomposition of the euclidean space $\mathbb{R}^2$ into regular grids. *Pixels* are elements of dimension 2 (squares) of the decomposition, *linels* are elements of dimension 1 (segments) between pixels, and *pointels* are elements of dimension 0 (points) between linels. We call i-face an element of dimension i (see examples in Fig. 1).

For a pixel p, we note linels(p) the set of the four linels between p and its four 4-neighbors, and pointels(p) the set of the four pointels around p. For a linel l, we note pointels(l) the set of the two pointels around l. These notations are directly extended to set of elements. For example, we note pointels(L), L being a set of linels, the union of pointels(l) for all $l \in L$.

A *cubical complex* C is a set of pointels, linels, surfels, glued together by adjacency and incidence relations. A *free pair* in C is a couple (c, c') with $c \in C$ being an i-face ($0 \leq i \leq 1$), $c' \in C$ being an $(i+1)$-face incident to c', and such that there is no other $(i+1)$-face in C distinct from c' incident to c. In such a case, (c, c') can be removed from C without modifying its topology. This is the basic operation used to deform a cubical complex while preserving its topology called the *elementary collapse*.

A cubical complex C collapses onto a second cubical complex C' if there is a sequence of *elementary collapse* transforming C into C'. We say that a complex Y is *simple* for X if Y can be "removed" from X without modifying the topology of X, and symmetrically, a complex Y' is *add-simple* for X' if Y' can be "added" into X' without modifying the topology of X'. These two notions can be defined thanks to the collapse operation (see [6] for precise definitions, and some examples in Fig. 1).

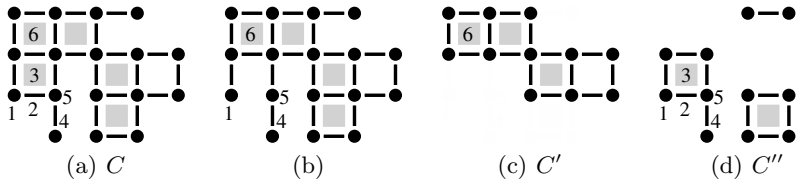


Fig. 1. Example of interpixel topology, cubical complex and collapse. (a) A cubical complex C. Pointels are black disks (for example 1 and 5), linels are black segments (for example 2 and 4) and pixels are gray squares (for example 3 and 6). Pointel 1 is incident to linel 2 and to pixel 3. Linel 2 is incident to pixel 3. Pointels 1 and 5 are adjacent. Linels 2 and 4 are adjacent. Couple $(2, 3)$ is a free pair. (b) Cubical complex obtained from C by the elementary collapse of free pair $(2, 3)$. (c) A cubical complex C' such that C collapses onto C'. (d) A cubical complex C'' such that C'' is simple for C. "Removing" C'' from C gives C'. Symmetrically, C'' is add-simple for C' and "adding" C'' in C' gives C.

2.3 Multi-label Simple Points

Given a pixel x belonging to region X, and a region R, we note $l(x, R)$ the linels around x that "touch" R: $l(x, R) = \{\text{linels } l | l \in \text{linels}(x)$ and the second pixel around l belongs to $R\}$. Similarly we note $p(x, R) = \{\text{pointels } p | p \in \text{pointels}(x)$ and the three pixels around p distinct from x belong to $R\}$. The *contour* of x touching R, called $c(x, R)$ is the composition of these two sets: $c(x, R) = (p(x, R), l(x, R))$. We note $c^-(x, R) = (p^-(x, R), l(x, R))$, with $p^-(x, R)$ the set of pointels touching a linel in $l(x, R)$, *i.e.* $p^-(x, R) = \text{pointels}(l(x, R))$.

The *frontier* between two regions X and R is noted $f(X, R)$. This is the set of all the linels between one pixel in X and one pixel in R, denoted $l(X, R)$, plus all the pointels touching these linels $p(X, R) = \text{pointels}(l(X, R))$.

Definition 1 of multi-label simple points is based on the interpixel topology and the simple and add-simple notions (see examples in Fig. 2). A pixel x belonging to region X is ML-simple for region R if the flip of x in R does not modify the topology of region R (first condition of the definition), does not modify the topology of region X (second condition of the definition), and if it does not modify the different frontiers around x (last condition of the definition). For the two first conditions, a cubical complex C is said homotopic to a 1-disk if it can be collapsed onto a complex that is composed only by one linel. For the last condition, intuitively it means that two regions adjacent along one frontier before the flip are still adjacent after the flip (*i.e.* the frontier still exists) and that two regions not adjacent before the flip are still not adjacent after.

Definition 1 (ML-simple points). *A pixel x, belonging to region X, is ML-simple for region R if:*

1. *$c(x, R)$ is homotopic to a 1-disk;*
2. *$c(x, X)$ is homotopic to a 1-disk;*

3. for each region O 4-adjacent to x, distinct from X and R: $c^-(x, O)$ is simple for $f(X, O)$; and add-simple for $f(R, O)$.

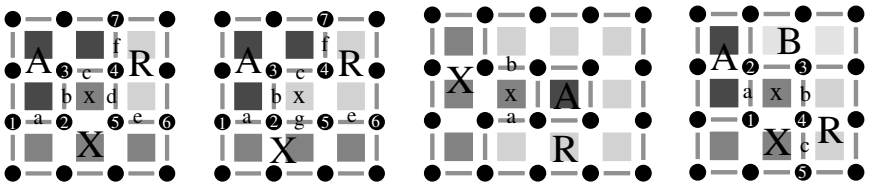


Fig. 2. Examples of ML-simple points and non ML-simple points. (a) x is ML-simple for R. $c(x, R) = (\{\}, \{d\})$ and $c(x, X) = (\{\}, \{g\})$ are homotopic to 1-disks. $c^-(x, A) = (\{2, 3, 4\}, \{b, c\})$ is simple for $f(X, A) = (\{1, 2, 3, 4\}, \{a, b, c\})$; and add-simple for $f(R, A) = (\{4, 7\}, \{f\})$. (b) Partition obtained from (a) after the flip of x in R. $f(X, R) = (\{2, 5, 6\}, \{e, g\})$, $f(X, A) = (\{1, 2\}, \{a\})$, $f(R, A) = (\{2, 3, 4, 7\}, \{b, c, f\})$. (c) x is not a ML-simple point for R because $c(x, R) = (\{\}, \{a, b\})$ is not homotopic to a 1-disk. Flipping x in R modifies the topology of R. (d) x is not a ML-simple point for R because $c^-(x, A) = (\{1, 2\}, \{a\})$ is not add-simple for $f(R, A) = (\{\}, \{\})$. Flipping x in R modifies the adjacency between regions.

The main operation used during the deformation algorithm is the *flip* of a pixel $x \in X$ into a given region R. This operation consists in removing x from its initial region X and adding it to region R. It has been proved in [11] that if the pixel x is ML-simple point for region R, it can be flipped into region R without modifying the topology of the partition. This operation modifies regions R and X, but also some frontiers. Since pointels are deduced from linels by definition of frontiers, modifications only concern the linels of the frontiers (see example in Fig. 2):

- $l(X, R) \leftarrow l(X, R) \setminus l(x, R) \cup l(x, X)$;
- For any region O with $O \neq X$, $O \neq R$: $l(X, O) \leftarrow l(X, O) \setminus l(x, O)$;
- For any region O with $O \neq X$, $O \neq R$: $l(R, O) \leftarrow l(R, O) \cup l(x, O)$;

2.4 Minimum Length Polygon

The Minimum Length Polygon (MLP) is a classical polygonalization method of digital contours [18, 23]. Assuming that the digital contour is the boundary of some digital object, it is the polygon with shortest perimeter, whose interior contains the centers of the pixels of the object and whose exterior contains the centers of the pixels of the background. One may also see it at the shortest closed polygonal line which stays within the strip formed by the minkowski sum of the digital contour and a centered unit square while making one loop inside it (see Fig. 3). It has been proved recently that the MLP is a good regularizer for digital

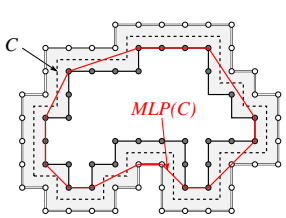


Fig. 3. Minimum Length Polygon of the digital contour C.

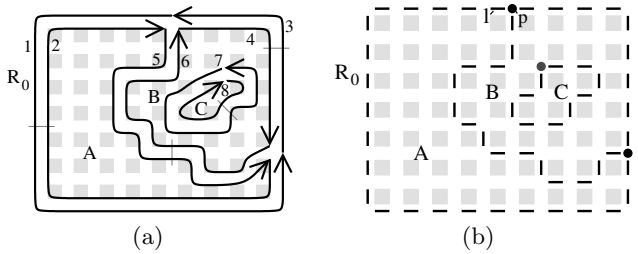


Fig. 4. Example of topological map. (a) Combinatorial map describing the topology of frontiers. (b) Interpixel matrix describing the geometry of frontiers.

active contours [10]: for instance, it is a kind of convex energy in a digital space. We use the combinatorial MLP in our experiments, since it is one of the fastest method for computing it [21].

2.5 Topological Map

Several works have proposed data structures to describe image partitions [12, 2]. The common goal is to represent regions of a given partition and all the incidence and adjacency relations between regions. Structures based on *combinatorial maps* [15] have several advantages justifying their use: they represent all the cells of the subdivision (vertices, edges and faces) and all the incidence and adjacency relations between these cells; they are defined based on a unique element called *dart*, simplifying their use; and they are defined in any dimension which allows extensions of this work in higher dimension.

Topological maps are an extension of combinatorial maps created to describe the whole topology of image partitions [8]. A topological map is composed of a combinatorial map that describes the cells of the partition and the incidence and adjacency relations between these cells (*e.g.* Fig. 4(a)), plus an interpixel matrix that describes the geometry of regions (*e.g.* Fig. 4(b)).

The combinatorial map describes each adjacency relation between two regions by an edge composed of two darts linked together. Thus, a dart can be seen as an half-edge (for example darts 1 and 2 in Fig. 4(a)). A cycle of successive darts describes a contour of a given region (for example $(1, 3)$ or $(2, 5)$). In the interpixel matrix, each linel belonging to one frontier between two regions is switched *on*, and each pointel touching more than two linels is switched *on*. Other elements are switched *off*. Each dart knows its region, and each region knows one dart of its external contour. Moreover, each dart d is associated with a pair $(pointel, linel)$ such that *pointel* corresponds to the origin of dart d, and *linel* is the first linel of the frontier associated with d (for example dart 1 is associated with pair (p, l)).

3 Deformable Model Process

To propose a new digital deformable partition model, we start by presenting the energies used and their computation algorithms. Then, we detail the main operations which are the test if a pixel is ML-simple, and the flip of a pixel. We study how to update the energies during the flip, trying to keep modifications as local as possible. Last, we give the global algorithm regrouping all these tools to minimize the energy of the digital deformable partition model.

3.1 Energies

We define the global energy of the partition as the weighted sum of two energies: E_d for the energy attached to data, and E_p for the energy attached to the deformable partition. In this work, we use the two following energies: $contour(C)$ is the MLP length of contour C weighted by the pixel gradient; $mse(R)$ is the minimum square error of region R.

These two energies can be directly computed from the topological map. For $contour(C)$, we run through all the darts of the given contour, and for each dart we run through all its linels, providing the geometry of the contour. The corresponding MLP can be computed from this geometry in time linear in the number of linels [21]. Then, we project each linel of the contour on the corresponding edge of the MLP. We compute the length of this projection, and multiply it by one minus the absolute difference between the two pixels around the current linel. This difference is divided by the maximal difference of the image, giving a number between 0 and 1. The value of $contour(C)$ is simply the sum of these values for each linel of the contour. If the image is uniform, $contour(C)$ is equal to the length of the MLP. Otherwise, $contour(C)$ becomes smaller when it separates more contrasted pixels.

For $mse(R)$, we run through each pixel of the image that belongs to the given region, and compute the number of pixels $M_0(R)$, the sum of the pixel values $M_1(R)$ and the sum of the pixel squared values $M_2(R)$. With these three values, we can directly estimate $mse(R) = \frac{1}{M_0(R)}(M_2(R) - \frac{M_1(R)}{M_0(R)})$.

To avoid several re-computation of these values, we store them in additional elements associated with the topological map. The value of $contour(C)$ is stored for each contour, and $M_0(R)$, $M_1(R)$ and $M_2(R)$ are stored for each region.

The two energies E_d and E_p are computed by summing up the basic energies on each element of the topological map: $E_d = w_R \times \sum_{\text{region } R} mse(R)$ and $E_p = w_C \times \sum_{\text{contour } C} contour(c)$. The weights w_R, w_C associated to both energies allow us to change the balance between the two energies.

3.2 Operations

The two main operations of the deformable process are the test if a pixel is ML-simple, and the flip of a pixel. Algorithm 1 allows to test if a given pixel is ML-simple. This algorithm follows directly Definition 1 (see [9] for more details).

Algorithm 1: `isSimple(x,R)`

Data: pixel $x \in X$; region R.
Result: *true* iff x is an ML-simple point for R.
if *not* isDisk$(c(x,R))$ **then return** $false$;
if *not* isDisk$(c(x,X))$ **then return** $false$;
foreach *region* O *4-adjacent to* x, $O \neq X$, $O \neq R$ **do**
  $P_1 \leftarrow \{l \in p(x,O) | l \in \text{pointels}(l(X,O) \setminus l(x,O))\}$;
  if *not* collapse$(c(x,O), P_1)$ **then return** $false$;
  $P_2 \leftarrow \{l \in p(x,O) | l \in f(R,O)\}$;
  if *not* collapse$(c(x,O)), P_2)$ **then return** $false$;
return *true*;

The two first lines of Algo. 1 correspond to the two first tests of Definition 1. Function isDisk, not given here, is simple to write since $c(x,R)$ is restricted to a small number of cases. Indeed, $c(x,R)$ can be empty, a cycle (*i.e.* a 2-disk), or made of several connected components: in these cases it is not a disk. The only other possible case is the case of the disk.

The "foreach" loop of Algo. 1 corresponds to the last condition of Definition 1; we only detail the two conditions add-simple and simple by using the definitions given in [6]. The main principle of these definitions is to test if the given set can be collapsed onto a specific set. This is the role of function collapse not given here. In our case, the first set is $c(x,O)$ which is bounded by the linels and pointels incident to the given pixel x. Thus, the collapse function is restricted to a small number of cases which can easily be tested.

The complexity of Algo. 1 is $O(1)$ since each test is restricted to pointels and linels around the considered pixel, and these numbers are both bounded by 4.

Algorithm 2: `flip(x,R)`

Data: a pixel $x \in X$ ML-simple for R.
Result: flip x into region R.
switch off linels in $l(x,R)$;
switch on linels in $l(x,X)$;
foreach *dart* d *touching* x **do**
  if *pointel*$(d) \in c^-(x,R)$ **then** update pair(d);

The second main operation is the flip which consists in removing the given pixel x from its original region X, and add it into its new region R. To be valid, this operation requires that x is a ML-simple pixel for region R. Algorithm 2 is made of three steps. First we switch off the linels in $l(x,R)$. Second we switch on the linels in $l(x,X)$. Since x is flip in region R, the linels in $l(x,R)$ become "inner linels", *i.e.* inside region R, while linels in $l(x,X)$ become frontier linels between region R and X. The last step consists to update associations between

darts and pairs. This updating is required when the pointel associated with a dart belongs to $c^-(x, R)$ (as in the case shown in Fig. 2(a)). In this case, the flip of x in R involves modifications of the extremities of some frontiers (in Fig. 2(a) $f(X, A) = (\{1, 2, 3, 4\}, \{a, b, c\})$ before the flip, and $f(X, A) = (\{1, 2\}, \{a\})$ after: one extremity has moved).

The complexity of Algo. 2 is also in $O(1)$. Indeed the number of linels in $l(x, R)$ and in $l(x, X)$ is at most 4, and the number of darts touching pixel x is at most 12 (3 darts for each pointel around x).

3.3 Update Energies for Flip

The main operation performed during the energy minimization process consists in flipping a pixel into a given region, and then to update the different energies accordingly. A first solution is to re-compute all the energies as explained in Sect. 3.1. But this involves a considerable computation time since the flip operation is called many times. A better solution consists in updating incrementally the energies.

For *mse*, we know that only region X and region R are modified. Thanks to the moments stored in each region, we can directly (*i.e.* in $O(1)$) update M_0, M_1 and M_2 for these two regions and thus the two *mse*. This modification can be made in constant time without additional cost for the `flip` algorithm. However, the problem is more complex for *contour* since there is no algorithm yet able to update a MLP after the modification of one of its pixels. Thus, *contour* is re-computed for the two contours around the flipped pixel by using the same algorithm than for the initialization step. The improvement of this particular point is one of our future goals.

The global energy of the partition can thus be updated by subtracting original values of each energy which will be modified, and by adding the updated energy values after the flip.

3.4 Energy Minimization Algorithm

We present in Algo. 3 the global minimization algorithm taking a topological map as input, and minimizing its energy by flipping ML-simple points. The algorithm starts by an initialization step, first to compute all the energies by using the principles given in Sect. 3.1. Then, each region X is considered to find a possible flip. This is the role of the `possibleFlip`(X) function, which is as follows. The function runs through each linel l describing a contour of X. Linel l separates two pixels: x in X, and a second pixel in another region R. If x is ML-simple for R, then flipping x into R is a possible candidate. Then, we flip x into R and compute the difference δ between the old energy (before the flip) and the new one (after the flip). Finally, we apply the reverse flip to retrieve the initial configuration and proceed to the next linel. If δ is negative, the flip of x into R decreases the global energy. In this case, this flip (x, R) is stored in region X into a variable called `candidate`(X), and function `possibleFlip`

returns true. Otherwise, we continue to run over the linels of the contours of X. If no flip decreasing the energy is found, `possibleFlip` returns false.

Algorithm 3: Digital partition minimization

Data: a topological map M; an image I.
Result: minimize the energy of the digital partition.
initialize all the energies of M; $S \leftarrow \emptyset$;
foreach *region* X **do**
  if *possibleFlip*(X) **then** $S \leftarrow S \cup \{X\}$;
while S *is not empty* **do**
  $X \leftarrow$ a region in S; $S \leftarrow S \setminus X$;
  $(x, R) \leftarrow$ `candidate`(X);
  `flip`(x, R);
  if *possibleFlip*(X) **then** $S \leftarrow S \cup \{X\}$;
  foreach *region* O *4-adjacent to* X **do**
    $S \leftarrow S \setminus O$;
    if *possibleFlip*(O) **then** $S \leftarrow S \cup \{O\}$;

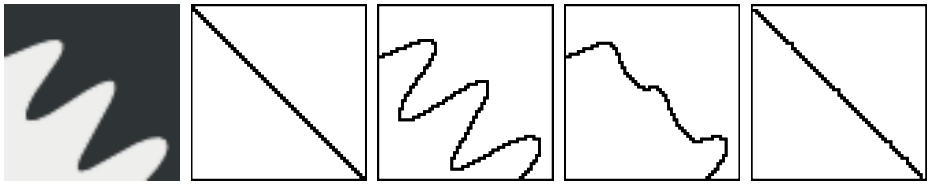
At the end of the initialization loop, S contains all regions having a flip that decreases the energy. In the main loop of the algorithm, we choose a region in the set, and flip the corresponding pixel. This operation changes the geometry and the energy values of the regions X and R. Stored flips for X, R, and their adjacent regions can be invalid (for example the considered pixel is not a simple point anymore, or the flip increases the energy) and should be updated using the `possibleFlip` algorithm. We can then proceed with an other region of S.

The main loop finishes when the set of candidate becomes empty. In this case, the minimization process is terminated: there is no possible flip decreasing the global energy.

4 Experiments

In this section, we first give some deformation results on artificial image using various weights w_R and w_C. Then, we show how the digital deformable partition model is used to fit an initial partition with two pictures from the Berkeley Segmentation Dataset [16]. In the following, weight values w_R and w_C are written as the ratio w_R/w_C.

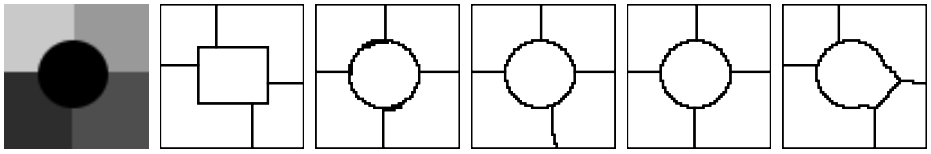
In Fig. 5, we deform the initial partition (Fig. 5(b)) of an image containing two regions (Fig. 5(a)). Different weights are used to illustrate how their values modify the resulting partition. If the weight of the contour energy is low ($w_R/w_C < 1/10^3$ as for example in Fig. 5(c)), the value of the region energy is greater than the value of the contour energy: the contour energy does not change the resulting partition. As the weight of the contour energy increases, the curve separating the two regions becomes straighter (Fig. 5(d)). If the weight of



(a) Image (b) Partition (c) 1 / 1 (d) 1 / 10^5 (e) 0 / 1

Fig. 5. Experimentation on an artificial image with two simple regions.

the region energy is set to zero, the obtained contour is nearly a straight line (Fig. 5(e)). It does not exactly match the frontiers of the initial partition: where the border crosses the curve, the gradient part of the contour energy slightly alter the shape of the border and produces some irregularities.

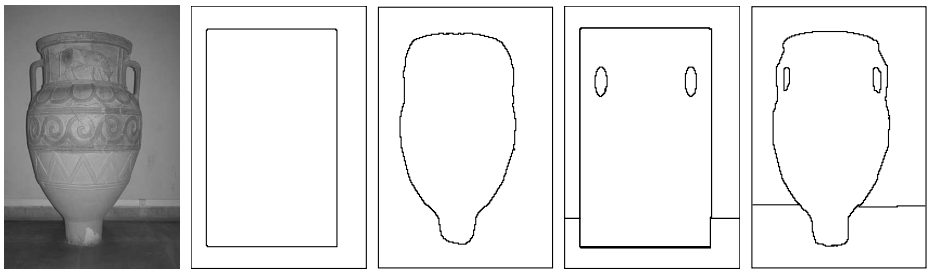


(a) Image (b) Partition (c) 1 / 1 (d) 1 / 10^4 (e) Two steps (f) 0 / 1

Fig. 6. Deformation of an artificial image with five regions.

In Fig. 6, we use the deformation process on an artificial image with five regions (Fig. 6(a)). The frontiers of the initial partition, presented in Fig. 6(b), are not aligned with the border of the different shape in the original image. The deformation process is applied with two different ratio between the energy weights. When the ratio between w_R and w_C is high (Fig. 6(c)), the region energy is privileged and borders follow correctly the image data. But we see that there is some artifact around the disk region: they are caused by the starting point of the different edges which do not move during the deformation process. When the ratio is lower, for instance $1/10^4$, some of the borders move correctly. However the edge separating the two darker gray region is not positioned correctly (Fig. 6(d)): the energy cost of having a curved border outweighs the gain in the region *mse*. Thus, the border is not able to move toward the separation of the two regions. To solve this issue, we applied a two step deformation with first a 1/1 ratio and then a $1/10^4$ ratio: the produced segmentation fits correctly the image data without artifact (Fig. 6(e)). Figure 6(f) presents the partition obtained using the contour energy alone: some parts of the disk are retrieved thanks to the gradient measure used in the energy. The bottom right side of the disk is straighten to minimize the length of the contour. If we use a 2D energy based on the number of linels, similar to the one used in [9], the initial partition has already a minimal

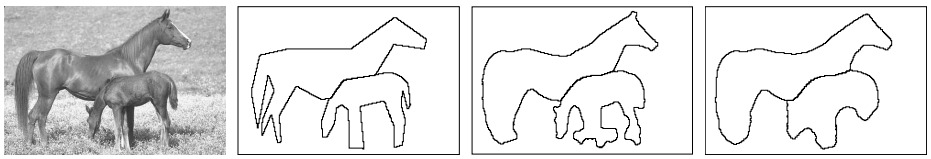
contour energy. This shows the improvement of the regularization power of the new contour energy based on the MLP and the gradient.



(a) Image (b) Partition A (c) A $1/10^4$ (d) Partition B (e) B $1/10^4$

Fig. 7. Deformation of different initial partitions of a picture.

The first real image is presented Fig. 7. Using the partition presented in Fig. 7(b), the deformation produced the partition shown in Fig. 7(c) with a $1/10^4$ weight ratio. We note that the shape of the object approximate the amphora in the picture but the contours are not perfectly matched. Using the partition presented in Fig. 7(d), the result (Fig. 7(e)) also fits correctly the amphora but for instance the curved shape just up from the handles is not correctly matched due to a high MLP value.



(a) Image (b) Partition (c) $1/10^4$ (d) $1/10^6$

Fig. 8. Deformation of an initial partition of a picture with different weight ratio.

In Fig. 8, we show deformation processes applied on a picture containing two horses (Fig. 8(a)). With the initial partition, Fig. 8(b), the first deformation guided by a weight ratio of $1/10^4$ fit the horses shapes (Fig. 8(c)): the legs of the small horse are segmented with the dark area of grass around. This is caused by the region energy which decreases when dark area are grouped. In Fig. 8(d), the weight ratio is $1/10^6$: the smaller horse is poorly segmented and the contour of the taller horse are approximated. We show that the contours tend to be as straight as possible: this remove thin parts of the regions.

5 Conclusion

In this paper, we have presented a new method of deformable digital partition. Several works have shown the interest of using deformable models for image segmentation, but to our knowledge, no work has yet be done on deforming any partition while preserving its topology. To achieve this objective, we used: (1) topological maps to efficiently compute and store features associated with the image and the partition; (2) minimum length polygons as estimator tools; and (3) ML-simple points to guarantee the preservation of the topology.

Our experiments show the interest of this approach to deform an initial partition according to image data. We can, for instance, propose a guided segmentation tool where a user creates an initial partition that is fitted to image data using the deformation process. We can also foresee the use of the deformable partition to improve the result of segmentation algorithms according to our energies.

In future works, we will focus on improving the update of the MLP after a flip operation. The MLP is currently completely re-computed after flips, involving important computation time. This can be improved by defining dynamic MLP capable of incremental update. Another goal is to use other energies to improve our results. We can, for example, use edge-detector filters (as Canny or Marr-Hildreth detectors) instead of a simple gradient energy, or add geometrical energies on regions to take their shapes into account (for example stretching or compaction). Finally, we could consider the extension of this work in 3D. Since topological maps and ML-simple points have already been defined in 3D [7, 9], we need to study the extension of MLP in higher dimension.

References

1. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.
2. L. Brun, J.-P. Domenger, and J.-P. Braquelaire. Discrete maps : a framework for region segmentation algorithms. In *Proc. GBR*, pages 83–92, 1997.
3. V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66:1–31, 1993.
4. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. J. Comput. Vision*, 22(1):61–79, 1997.
5. C Couprie, L. Grady, L. Najman, and H. Talbot. Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In *Proc. ICCV*, pages 731–738. IEEE, 2009.
6. M. Couprie and G. Bertrand. New characterizations of simple points, minimal non-simple sets and p-simple points in 2d, 3d and 4d discrete spaces. In *Proc. DGCI*, volume 4992 of *LNCS*, pages 105–116, 2008.
7. G. Damiand. Topological model for 3d image representation: Definition and incremental extraction algorithm. *Computer Vision and Image Understanding*, 109(3):260–289, 2008.
8. G. Damiand, Y. Bertrand, and C. Fiorio. Topological model for two-dimensional image representation: Definition and optimal extraction algorithm. *Computer Vision and Image Understanding*, 93(2):111–154, 2004.

9. G. Damiand, A. Dupas, and J.-O. Lachaud. Fully deformable 3d digital partition model with topological control. *Pattern Recognition Letters*, 2011, to appear.
10. F. de Vieilleville and J.-O. Lachaud. Digital Deformable Model Simulating Active Contours. In *Proc. DGCI*, volume 5810 of *LNCS*, pages 203–216, 2009.
11. A. Dupas, G. Damiand, and J.-O. Lachaud. Multi-label simple points definition for 3d images digital deformable model. In *Proc. DGCI*, volume 5810 of *LNCS*, pages 156–167, 2009.
12. C. Fiorio. A topologically consistent representation for image analysis: the frontiers topological graph. In *Proc. DGCI*, volume 1176 of *LNCS*, pages 151–162, 1996.
13. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Comput. Vision*, 1(4):321–331, 1988.
14. E. Khalimsky, R. Kopperman, and P.R. Meyer. Computer graphics and connected topologies on finite ordered sets. *Topology and its Applications*, 36:1–17, 1990.
15. P. Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *Int. J. Comput. Geom. Ap.*, 4(3):275–324, 1994.
16. D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. ICCV*, volume 2, pages 416–423, 2001.
17. T. Meltzer, C. Yanover, and Y. Weiss. Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In *Proc. ICCV*, volume 1, pages 428–435, 2005.
18. U. Montanari. A note on minimal length polygonal approximation to a digitized contour. *Communications of the ACM*, 13(1):41–47, 1970.
19. D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–684, 1989.
20. T. Pock, A. Chambolle, D. Cremers, and H. Bischof. A convex relaxation approach for computing minimal partitions. In *Proc. CVPR*, pages 810–817, 2009.
21. X. Provençal and J.-O. Lachaud. Two linear-time algorithms for computing the minimum length polygon of a digital contour. In *Proc. DGCI*, volume 5810 of *LNCS*, pages 104–117, 2009.
22. F. Ségonne. Active contours under topology control - genus preserving level sets. *Int. Journal of Computer Vision*, 79:107–117, 2008.
23. J. Sklansky, R. L. Chazin, and B. J. Hansen. Minimum perimeter polygons of digitized silhouettes. *IEEE Trans. Computers*, 21(3):260–268, 1972.
24. R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(6):1068–1080, 2008.
25. L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. Comput. Vision*, 50(3):271–293, 2002.