

# Active Contours Under Topology Control Genus Preserving Level Sets

Florent Ségonne

Willow - Certis Laboratory - ENS / INRIA / ENPC  
Paris, France

## Abstract

*We present a novel framework to exert topology control over a level set evolution. Level set methods offer several advantages over parametric active contours, in particular automated topological changes. In some applications, where some a priori knowledge of the target topology is available, topological changes may not be desirable. This is typically the case in biomedical image segmentation, where the topology of the target shape is prescribed by anatomical knowledge. However, topologically constrained evolutions often generate topological barriers that lead to large geometric inconsistencies. We introduce a topologically controlled level set framework that greatly alleviates this problem. Unlike existing work, our method allows connected components to merge, split or vanish under some specific conditions that ensure that the genus of the initial active contour (i.e. its number of handles) is preserved. We demonstrate the strength of our method on a wide range of numerical experiments and illustrate its performance on the segmentation of cortical surfaces and blood vessels.*

## 1. Introduction

Active contours constitute a general technique of matching a deformable model onto an image by means of energy minimization. Curves, surfaces, or higher-dimensional geometric object, deform within an input image subject to both internal and external forces and external constraints. Since their introduction by Kass et al. in [20], active contours have benefited many computer vision research areas, such as image segmentation [5, 6, 7, 13, 23, 35], region tracking [25, 27], shape from stereo [9, 10, 14, 19, 28], shape from shading [12, 36], shape from point clouds [2, 34, 37], among others.

Geometric active contours, which represents the manifold of interest as level sets of functions defined on higher-dimensional manifolds [5, 26], offer many advantages over parametric approaches. In addition to their ease of implementation, level sets do not require any parameterization of the evolving contour. Self-intersections, which are costly to prevent in parametric deformable models, are naturally avoided and topological changes are automated. Also, the geometric properties of the contour, such as the normal or the curvature, are easily computed from the level set function. We refer the interested reader to the dissertation of Pons for an in-depth discussion of these concepts [29].

The ability to automatically change topology is often presented as an advantage of the level set method over explicit deformable models. However, this behavior is not desirable in some applications. This is

typically the case in biomedical image segmentation, where the topology of the target shape is prescribed by anatomical knowledge. In order to overcome this problem, a topology-preserving variant of the level set method has been proposed by Han, Xu, and Prince in [17]. The level set function is iteratively updated with a modified procedure based on the concept of *simple point* from digital topology [3]; the final mesh is obtained with a modified topology-consistent marching cubes algorithm. This method ensures that the resulting mesh has the same topology as the user-defined initial level set.

While such topological preservation is desired in some applications, it is often too restrictive. Because the different components of the contour are not allowed to merge or to split up, the number of connected components will remain constant throughout the evolution. This number must be known by the user *a priori* and the initial contour must be designed accordingly. Also, the sensitivity to initial conditions, which already limits the applicability and efficiency of active contour methods, is considerably increased. The initial contour must both have the same topology as the target shape and be close enough to the final configuration, otherwise the evolution is likely to be trapped in topological dead-ends including large geometric inconsistencies (*cf* Fig. 2-b, Fig. 3-b, and Fig. 7-a).

Although being able to control the topology of an active contour is an attractive feature, forcing it to remain identical through an evolution is a strong constraint. In this paper, we propose a method to exert a more subtle topological control on a level set evolution. Some *a priori* knowledge of the target topology can be integrated without requiring that the topology be known exactly. Our method greatly alleviates the sensitivity to initial conditions by allowing connected components to merge, split or vanish without changing the genus of the active contour, i.e. its number of handles. For example, an initial contour with a spherical topology may split into several pieces, go through one or several mergings, and finally produce a certain number of contours, all of which are topologically equivalent to a sphere. A subset of these components may then be selected by the user as the desired output (typically the largest component if one spherical contour is needed, the others being caused by noise).

Our approach is based on an extension of the concept of *simple point* to “multi-label” images, that we have called *multisimple point* [32]. This criterion ensures that no handles are generated or suppressed while splitting or merging the components of the object. The resulting algorithm fills the gap between the original level set framework and topology-preserving level set methods. We demonstrate the strength of our method on a wide range of numerical experiments and illustrate its performance on the segmentation of cortical surfaces and blood vessels. Part of this work was presented at a workshop of the *International Conference on Computer Vision* [31].

## 2. Background

### 2.1. Topology

Topology is a branch of mathematics that studies the properties of geometric figures that are preserved through deformations, twistings and stretchings, hence without regard to size, absolute position.

In this work, we focus on compact surfaces in three dimensions. Any compact connected orientable surface is homeomorphic to a sphere with some number of handles [8]. This number of handles is a topological invariant called the *genus*. For example, a sphere is of genus 0 and a torus is of genus 1. The genus  $g$  is directly related to another topological invariant called the *Euler characteristic*  $\chi$  by the formula  $\chi = 2 - 2g$ . The Euler characteristic is of great practical interest because it can be calculated from any polyhedral decomposition of the surface by  $\chi = V - E + F$ , where  $V$ ,  $E$  and  $F$  denote

respectively the number of vertices, edges and faces of the polyhedron<sup>1</sup>.

Homeomorphisms are used to define the *intrinsic* topology of an object, independently of the embedding space. For example, a knotted solid torus has the same genus as a simple torus, or a hollow sphere as two spheres. In order to topologically differentiate these surfaces, one needs a theory that considers the embedding space. Homotopy, which defines two surfaces to be homotopic if one can be continuously transformed into the other, is such a theory that provides a measure of an object’s topology (see [18] for an excellent course in algebraic topology).

## 2.2. Digital topology

Digital topology provides an elegant framework, which transposes the continuous concepts of topology to discrete images. In this theory, binary images inherit a precise topological meaning. In particular, the concept of homotopic deformation, necessary to assign a topological type to a digital object, is clearly defined through the notion of *simple point*. An extensive discussion of these concepts can be found in the work of Mangin et al [24]. In this section, some basic notions of digital topology are presented. All definitions are taken from the work of G. Bertrand, which we refer to for more details [3].

A 3D binary digital image  $I$  is composed of a foreground object  $X$  and its inverse, the complement  $\bar{X}$ . We first need the concept of *connectivity*, which specifies the condition of adjacency that two points must fulfill to be regarded as connected. Three types of connectivity are commonly used in 3D: 6-, 18- and 26-connectivity. Two voxels are 6-adjacent if they share a face, 18-adjacent if they share at least an edge and 26-adjacent if they share at least a corner. In order to avoid topological paradoxes, different connectivities,  $n$  and  $\bar{n}$ , must be used for  $X$  and  $\bar{X}$ . This leaves us with four pairs of compatible connectivities, (6,26), (6,18), (18,6) and (26,6). We note that digital topology does not provide a consistent framework for multi-label images, since no compatible connectivities could be chosen for neighboring components of the same object. Digital topology is strictly limited to binary images.

We then go to the definition of a *simple point*. This concept is central to the method of [17] and to our method. A point of a binary object is *simple* if it can be added or removed without changing the topology of both the object and its background, i.e. without changing the number of connected components, cavities and handles of both  $X$  and  $\bar{X}$ . A simple point is easily characterized by two *topological* numbers with respect to the digital object  $X$  and a consistent connectivity pair  $(n, \bar{n})$ . These numbers, denoted  $T_n(x, X)$  and  $T_{\bar{n}}(x, \bar{X})$  (or  $T_n$  and  $T_{\bar{n}}$  when no confusion is possible), have been introduced by G. Bertrand in [3] as an elegant way to classify the topology type of a given voxel. The values of  $T_n(x, X)$  and  $T_{\bar{n}}(x, \bar{X})$  characterize isolated, interior and border points as well as different kinds of junctions. In particular, a point is simple if and only if  $T_n(x, X) = T_{\bar{n}}(x, \bar{X}) = 1$ . Their efficient computation, which only involves the 26-neighborhood, is described in [4].

## 2.3. Topology preserving level sets

The level set method models the evolution of an active contour  $\Gamma : t \in \mathbb{R}^+ \rightarrow \Gamma(t)$ , where  $\Gamma(t)$  is a closed and embedded hypersurface in  $\mathbb{R}^n$ , by the level set of a function defined on  $\mathbb{R}^n$ . Although many functions could be chosen to represent the active contour  $\Gamma$ , the signed distance function is usually preferred for its stability in numerical computations. The moving contour  $\Gamma$  is represented by a level set

---

<sup>1</sup>In the case of multiple surfaces involving  $K$  connected components, the total genus is related to the total Euler-characteristic by the formula:  $\chi = 2(K - g)$ .

function  $\phi : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}$  such that:

$$\begin{cases} \phi(\mathbf{x}, t) < 0 & \text{if } \mathbf{x} \text{ is inside } \Gamma(t) \\ \phi(\mathbf{x}, t) = 0 & \text{if } \mathbf{x} \in \Gamma(t) \\ \phi(\mathbf{x}, t) > 0 & \text{if } \mathbf{x} \text{ is outside } \Gamma(t) \end{cases}$$

A deformation of  $\Gamma$  under the velocity field  $\mathbf{v}$ :

$$\frac{\partial \Gamma(\mathbf{x}, t)}{\partial t} = \mathbf{v}(\mathbf{x}, t)$$

corresponds to the level set formulation:

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} + \mathbf{v}(\mathbf{x}, t) \cdot \nabla \phi(\mathbf{x}, t) = 0.$$

The ability to automatically handle topology changes is a long-acknowledged advantage of the level set method over explicit deformable models, but may not be desirable in some applications where some prior knowledge of the target topology is available. Han et al. [17] have used the concept of simple point to design a topology-preserving variant of the level set framework. The binary object of interest is the interior of the contour, i.e. the domain where the level set function  $\Phi$  is strictly negative:  $X = \{\mathbf{x} \in I \mid \Phi(x) < 0\}$ . The digital topology of  $X$  is preserved during the evolution by the means of a modified update procedure detailed in Algorithm 1, below. This approach prevents digital non-simple grid points from changing sign, therefore retaining the initial digital topology throughout the level set evolution. Briefly, every time a point  $\mathbf{x}$  is about to change sign, the simple point condition (i.e.  $T_n(x, X) = T_{\bar{n}}(x, \bar{X}) = 1$ ) is verified: if the point is simple, the new value is accepted; in the other situation, a small value of the same sign as the current value is set, thereby preserving the digital topology of the binary object.

---

**Algorithm 1** Topology-preserving level sets. Han et al. [17]

---

```

for all iterations do
  for all grid points do
    Compute the new value of the level set function
    if the sign does not change then
      Accept the new value
    else {sign change}
      Compute the topological numbers
      if the point is simple then
        Accept the new value
      else
        Discard the new value
        Set instead a small value of the adequate sign
  
```

---

For this method to be useful, it must be complemented with a topology-consistent isocontour extraction algorithm. Standard marching squares or marching cubes algorithm [22] do not generate topologically consistent tessellations. In order to alleviate this problem, Han et al. have designed a modified

connectivity-consistent marching contour algorithm, by building a specialized case table for each type of digital topology. Using the topology-preserving level set algorithm and the topology-consistent marching contour algorithm in conjunction, with the same digital topology  $(n, \bar{n})$  throughout the process, guarantees that the output mesh is topologically equivalent to the user-defined initial level set.

### 3 Methods

The simple point condition is a very efficient way to detect topological changes during a level set evolution. However, in many applications, the topology-preserving level set method of Han et al. is too restrictive.

The primary concern is topological defects such as handles, which are difficult to retrospectively correct [16, 11, 15, 21, 30, 33]. On the other hand, changes in the number of connected components (including cavities) during the evolution are less problematic. Different connected components are easily identified using standard region growing algorithms. A subset of them may be selected by the user as the final output, typically the largest one if a single component is needed, the others being imputable to noise in the input data. Unexpected cavities, which can be interpreted as background  $\bar{n}$ -connected components, might generate large geometrical inaccuracies in level set evolutions that prevent their formation. This might be of particular importance when dealing with noisy data or in medical imaging, when unexpected medical structures, such as tumors, might exist in the volume to be segmented.

We extend the concept of simple point to allow distinct connected components to merge, split, appear, or disappear, while ensuring that no additional handle is generated in the object. For example, an initial contour with spherical topology may split into several pieces, generate cavities, go through one or several mergings, and finally produce a specific number of surfaces, all of which are topologically equivalent to a sphere.

#### 3.1 From simple points to multisimple points

The different values of  $T_n$  and  $T_{\bar{n}}$  characterize the topology type of a given point  $x$ , providing important information with regards to its connectivity to the object  $X$ . In particular, isolated points are characterized by  $T_n = 0$  and  $T_{\bar{n}} = 1$ , interior points by  $T_n = 1$  and  $T_{\bar{n}} = 0$ , different junctions by  $T_n > 1$  and  $T_{\bar{n}} = 1$ , etc. This additional information can be used to carefully design a multi-label digital framework, which allows different connected components to split, merge or vanish under topology control.

**Definition:** We say that a point is *multisimple* if it can be added or removed without changing the number of handles of the digital objects  $X$  and  $\bar{X}$ .

Contrary to the case of simple points, the addition of a multisimple point may merge several connected components, and its removal may split a component into several parts. We note  $C_n(x, X)$  the set of  $n$ -connected components of  $X \setminus \{x\}$  that are  $n$ -adjacent to  $x$ . If the cardinality of this set  $|C_n(x, X)|$  is strictly greater than one, the addition or removal of  $x$  involves a merge or a split respectively. By duality, the generation of one or several cavities in  $X$  can be interpreted as a split of connected components of  $\bar{X}$ . In [32], we introduced two *extended topological numbers*  $T_n^+(\mathbf{x}, X) = |C_n(x, X)|$  and  $T_{\bar{n}}^+(\mathbf{x}, \bar{X}) = |C_{\bar{n}}(x, \bar{X})|$ , which used in conjunction with the topological numbers  $T_n$  and  $T_{\bar{n}}$  provide an elementary

characterization of multisimple points.

### Characterization of multisimple points

$$\text{A point is said to be multisimple if and only if: } \begin{cases} T_n^+(x, X) = T_n(x, X) \\ T_{\bar{n}}^+(x, \bar{X}) = T_{\bar{n}}(x, \bar{X}) \end{cases} . \quad (1)$$

When merging or splitting connected components by adding or removing a multisimple point, the total genus (i.e. the total number of handles) of the different components is preserved. We note that, under this condition, isolated and interior points are multisimple points, which allows components to appear or disappear. In addition, the above criterion can be easily modified to allow more specific topological changes, such as preventing the formation of cavities or the generation of disconnected components during the evolution. For instance, no cavities will appear or disappear when  $T_n^+(x, X) = T_n(x, X)$  and  $T_{\bar{n}}(x, \bar{X}) = 1$ ; no components will split or merge when  $T_n(x, X) = 1$  and  $T_{\bar{n}}^+(x, \bar{X}) = T_{\bar{n}}(x, \bar{X})$ . Figure 1 illustrates on a simple example the concept of multisimple point.

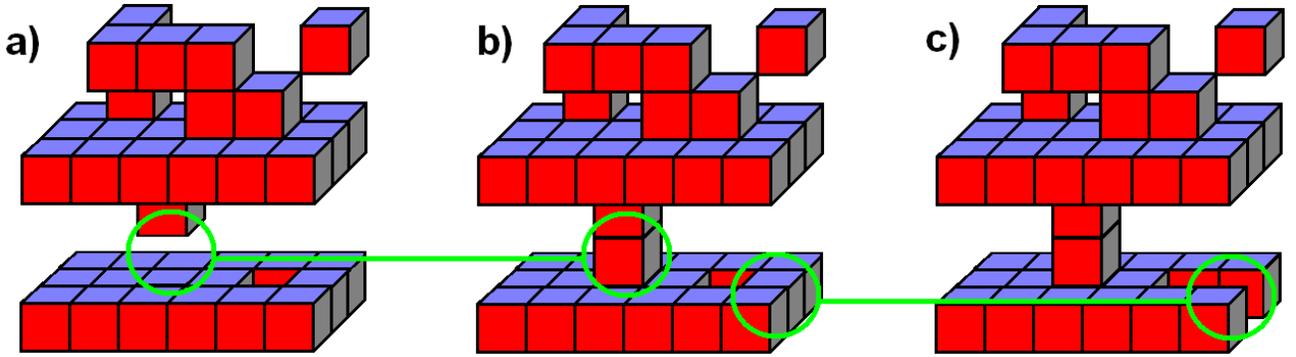


Figure 1: a-b) Addition of a multisimple point ( $T_n^+ = T_n = 2$  and  $T_{\bar{n}} = 1$ ). b-c) Deletion of a non-multisimple point ( $T_n^+ = 1 \neq T_n = 2$  and  $T_{\bar{n}} = 1$ ). The deletion of the circled voxel suppress a handle in the digital volume.

## 3.2 Level set evolution under topology control

With the concept of multisimple point in hand, we are now ready to describe our new level set framework. Similarly to the approach of Han et al. [17], we exploit the binary nature of the level set function  $\Phi$  that partitions the underlying digital image into strictly negative inside points and positive outside points. During the evolution, we maintain a map  $L$  of labels coding for the different connected components of  $X = \{\mathbf{x} \in I \mid \Phi(x) < 0\}$  and  $\bar{X} = \{\mathbf{x} \in I \mid \Phi(x) \leq 0\}$ . Each  $n$ -connected component of  $X$  is represented by a set of  $n$ -connected negative points, and is assigned a specific label in the label map  $L$ . Similarly,  $\bar{n}$ -connected components of  $\bar{X}$  constitute sets of  $\bar{n}$ -connected positive points, and are assigned distinct labels in  $L$ . The label map  $L$  is updated concurrently with the level set function  $\phi$ . For clarity, we first explain the main concepts. The update procedure for each grid point at each iteration is described in Alg. 2.

During the evolution, only voxels whose levelset value is about to change sign could modify the topology of the active contour. For those, the simple point condition, more restrictive, is checked first, because it is computationally cheaper. If the point is non-simple, then  $T_n^+(\mathbf{x}, X)$  and  $T_{\bar{n}}^+(\mathbf{x}, \bar{X})$  are computed in order to check the multisimple criterion. Two comments follow:

---

**Algorithm 2** Update Scheme for the Evolution of Level Sets Under Topology Control

---

Compute the new value of the level set function  
**if** the sign does not change **then**  
    Accept the new value  
**else** {sign change}  
    Compute the topological numbers  
    **if** the point is simple **then**  
        Accept the new value  
        **if** negative new value **then**  
             $L(\mathbf{x}) \leftarrow$  the only foreground label  $n$ -adjacent to  $\mathbf{x}$   
        **else**  
             $L(\mathbf{x}) \leftarrow$  the only background label  $\bar{n}$ -adjacent to  $\mathbf{x}$   
    **else** {non-simple point}  
        Compute the extended topological numbers  $T_n^+(\mathbf{x}, X)$  and  $T_{\bar{n}}^+(\mathbf{x}, \bar{X})$   
        **if** the point is multisimple  $\{T_n^+ = T_n$  and  $T_{\bar{n}}^+ = T_{\bar{n}}\}$  **then**  
            **if** negative new value **then**  
                **if**  $T_{\bar{n}} > 1$  **then**  
                    Split the  $\bar{n}$ -component of  $C_{\bar{n}}(\mathbf{x}, \bar{X})$   
                **if**  $T_n > 1$  **then**  
                    Merge the  $n$ -component that  $\mathbf{x}$  belongs to  
                     $L(\mathbf{x}) \leftarrow$  the only label  $n$ -adjacent to  $\mathbf{x}$   
            **else** {positive new value}  
                **if**  $T_{\bar{n}} > 1$  **then**  
                    Merge the  $\bar{n}$ -components of  $C_{\bar{n}}(\mathbf{x}, \bar{X})$   
                **if**  $T_n > 1$  **then**  
                    Split the  $n$ -component that  $\mathbf{x}$  belongs to  
                     $L(\mathbf{x}) \leftarrow$  the only label  $\bar{n}$ -adjacent to  $\mathbf{x}$   
            **else** {non-multisimple point}  
                Discard the new value  
                Set instead a small value of the adequate sign

---

- If  $\mathbf{x}$  is part of the background object  $\bar{X}$  and is a candidate for *addition* in  $X$ ,  $T_n^+(\mathbf{x}, X)$  can be deduced directly from the map  $L$ . In this case,  $C_n(\mathbf{x}, X)$  corresponds to the set of  $n$ -connected components of  $X$  whose label in the map  $L$  is  $n$ -adjacent to  $\mathbf{x}$ . Therefore,  $T_n^+(\mathbf{x}, X)$  is simply the number of distinct labels in  $L$  for points in  $X$  that are  $n$ -adjacent to  $\mathbf{x}$ . By duality, the same comment apply when  $\mathbf{x}$  is part of the foreground object  $X$  and is a candidate for addition in  $\bar{X}$ : in this situation  $T_{\bar{n}}^+(\mathbf{x}, \bar{X})$  can be deduced directly from the map  $L$ .
- On the other hand, when  $\mathbf{x}$  is a candidate for *removal* in  $X$ , the complete set  $C_n(\mathbf{x}, X)$  must be computed. In the most general case, this step is computationally expensive and standard region growing algorithms must be used to identify these components. However, when all the initial components have a spherical topology (i.e. do not possess any handles), which is the most common situation in practice, the computation of  $T_n^+(\mathbf{x}, X)$  involves local computations only: since the  $n$ -connected component of  $\mathbf{x}$  has a spherical topology, the condition  $T_n^+(\mathbf{x}, X) = T_n(\mathbf{x}, X)$  is always

verified. This implies that the first condition for multisimple point is always verified when all the initial components have a spherical topology (see Eq. 1). Obviously, the same condition applies when  $\mathbf{x}$  is a candidate for *removal* in  $\overline{X}$ .

Some care must be taken in order to ensure that the map of labels  $L$  is correctly updated. The more complex case is the removal of a multisimple point involving a split. In this case, some unused labels must be assigned to the new connected components. Algorithm 2 describes in detail the update procedure for the label map. Note that new components can only be generated through the splitting of an already existing component, as level set evolutions do not allow the spontaneous generation of new disconnected fronts.

The resulting procedure is an efficient level set method that prevents handles from being created during the evolution, allowing the number of connected components (including cavities) to vary. We insist on the fact that digital topology does not provide a consistent framework for multi-label images. However, by ensuring that no components of the same object  $X$  or  $\overline{X}$  are adjacent, topological inconsistencies are avoided.

### 3.3 Implementation Issues

The implementation of the level set method is computationally expensive. In order to increase the computational speed of geometric deformable models, a narrow band method is usually adopted [1]. Only the grid points that are in a small neighborhood of the active contour are updated. During the evolution, most points of the narrow-band do not change sign and do not imply a potential change of topology. The simple point condition, which only involves local calculations, is computationally cheap and leads to an efficient algorithm (see [4, 17]). Similarly, when the initial level set components do not possess any handles, multisimple points, which constitute a direct extension of the concept of simple points, only require local computations. This situation is the most common in practice, since one is most often interested in segmenting structures that have a spherical topology.

The merging of connected components into one single component can easily be done using the label map  $L$ . If a point  $\mathbf{x}$  is part of the background object and is a candidate for addition,  $C_n(\mathbf{x}, X)$  can be deduced directly from the label map and the voxels directly adjacent to  $\mathbf{x}$ . The merging step simply amounts to assigning the same label to each neighboring connected components.

One must be more careful when splitting components (i.e. split of a foreground component into several components or generation of cavities). During the evolution, some components may need to be split into several components, which requires the assignment of some unused labels to the new connected components that must be previously identified. This can be done efficiently using standard region growing algorithms.

### 3.4 Variations on Topologically-Controlled Level Sets

The proposed framework can be modified to allow more specific topological control during the level set evolution. The multisimple condition introduced in Sect 3.1 can be used to distinguish different topological changes. The splitting of a foreground component or the merging of several foreground

components correspond to the condition:

$$\begin{cases} T_n^+(\mathbf{x}, X) &= T_n(\mathbf{x}, X) \\ T_{\bar{n}}(\mathbf{x}, X) &= 1 \end{cases}, \quad (2)$$

while the generation of a cavity or the merging of several background components are characterized by:

$$\begin{cases} T_n(\mathbf{x}, X) &= 1 \\ T_{\bar{n}}^+(\mathbf{x}, X) &= T_{\bar{n}}(\mathbf{x}, X) \end{cases}. \quad (3)$$

We note that the condition  $T_n = 1$  implies  $T_n^+ = 1$ , which proves that the previous criteria characterize multisimple points.

Using these specific criteria, one can design more elaborate level set frameworks that allow some specific topological changes only. For instance, using the criterion 2, one can design a level set evolution that allow foreground components to split or merge only. On the other hand, cavities can be controlled using the criterion 3. Finally, note that the three criteria allows the user to control exactly the different topological changes applied to the digital object.

## 4 Experiments and Applications

In this section, we demonstrate the interest of using the genus-preserving level set method for image segmentation. We present some experiments illustrating the performance of our approach and introduce some potential applications. We first apply our level set framework to two synthetic examples to demonstrate its ability to handle disconnected components and cavities. Next, two real data segmentation tasks are presented: the generation of cortical surfaces from MRI images and the extraction of blood vessels from MRA images.

Our segmentation framework is simplistic. The deformation of the active contour  $\Gamma$  is driven by a velocity field that is a combination of an intensity-based term,  $(I - I_{thres})$ , and a mean curvature term,  $H$ :

$$\frac{\partial \Gamma(\mathbf{x}, t)}{\partial t} = \mathbf{v}(\mathbf{x}, t) \text{ with } \mathbf{v}(\mathbf{x}, t) = [\alpha(I(\mathbf{x}) - I_{thres}) - H(\mathbf{x}, t)]\mathbf{n}(\mathbf{x}, t),$$

where  $I$  denotes the scalar image to be segmented,  $I_{thres}$  is a suitable intensity threshold, which separates the object from the background,  $\mathbf{n}(\mathbf{x}, t)$  is the outward normal to the isosurface of the active contour at location  $\mathbf{x}$ , and  $\alpha$  is a weighting parameter. The corresponding level set evolution equation is:

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = [-\alpha(I - I_{thres}) + \frac{1}{n-1} \operatorname{div}(\frac{\nabla \phi}{|\nabla \phi|})] |\nabla \phi|.$$

More complex images would require more elaborate segmentation frameworks. However, the choice of a particular segmentation method is not the issue here. We rather focus on the improvements brought by our approach, as regards to the management of the topology, relative to the standard level set method and to the topology-preserving method of [17].

## 4.1 Synthetic data

### Experiment 1: Segmentation of a ‘C’ shape

First, we consider the segmentation of a simple ‘C’ shape under two different initializations (Fig. 2 and Fig. 3). Our method, columns c, is compared to the original level set formulation, columns a, and the topology-preserving model introduced by Han et al. [17], columns b. The differences of behavior are circled in the images. Two different initializations (a little sphere in Fig. 2 and a larger box in Fig. 3) were used to test the sensitivity of each method to initial conditions.

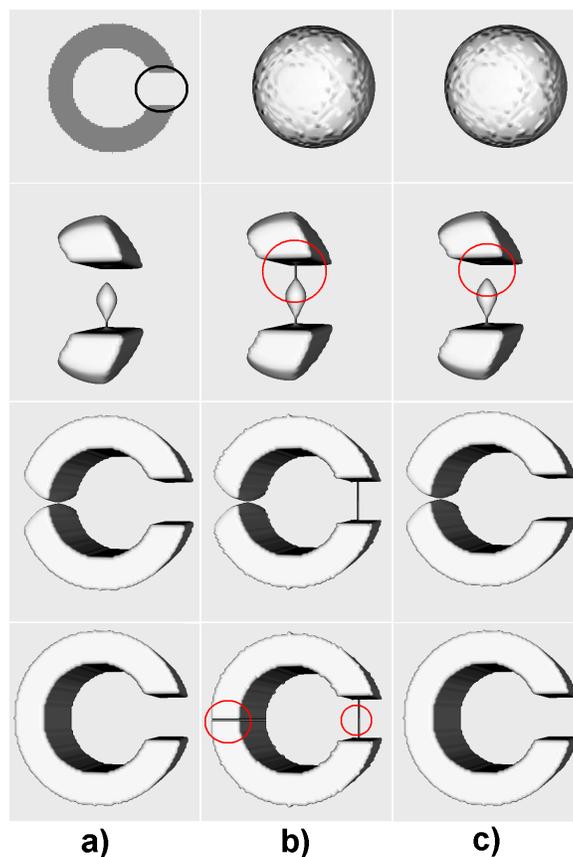


Figure 2: Segmentation of a ‘C’ shape using a spherical initialization. The top row shows cuts of the 3D shape locating the initial contour. a) Traditional level sets. b) Topology-preserving level sets. c) Genus-preserving level sets. Differences of behavior are circled in the images. In this case, our method behaves exactly like the standard level set method. Topology-preserving level sets are trapped in a deadlock.

In these simple examples, both standard level sets and genus-preserving level sets yield the expected result. With the first initialization (Fig. 2-a,c) the two methods behave in exactly the same way, because no handle is generated during the evolution. During the evolution, three distinct components are generated, one of which vanishes, while the two other components merge, closing the ‘C’ shape. With the second initialization (Fig. 3-a,c) they behave differently. Standard level sets temporarily generate a toroidal topology (row 3, column a), whereas our method prevents the formation of the handle (row 3,

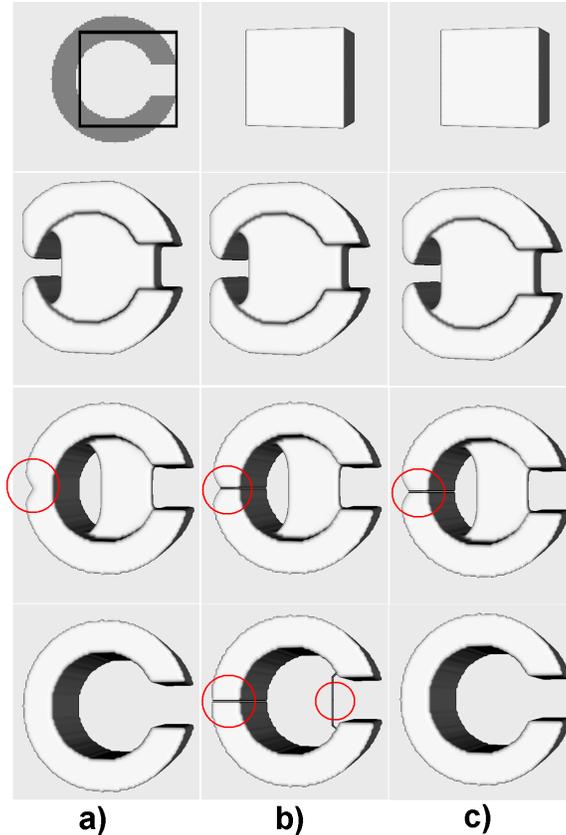


Figure 3: Segmentation of a ‘C’ shape using a rectangular initialization. The top row shows cuts of the 3D shape locating the initial component. a) Traditional level sets. b) Topology-preserving level sets. c) Genus-preserving level sets. Differences of behavior are circled in the images. Our method is able to achieve a correct segmentation without generating a toroidal topology during the evolution.

column c) by delaying a merging until a split in another part of the object occurs.

In contrast, topology-preserving level sets yield poor results. For the two different initializations, they get trapped in a topological deadlock. Although the final surface has the correct topology, it has large geometric errors (row 4, column b): a filament linking the two ends of the ‘C’ and a separating membrane at the middle of the ‘C’. These topological barriers, generated during the evolution, are difficult to correct retrospectively.

The behavior of our approach corresponds to a trade-off between standard level sets and topology-preserving level sets. Compared to the former, the formation or closing of handles is prevented. Compared to the latter, the ability to change topology under certain condition greatly alleviates the sensitivity to initial conditions.

## Experiment 2: Formation of cavities

The second experiment, shown in Fig.4, illustrates the ability of our approach to generate cavities during the evolution. The object to be segmented is a synthetic cube, containing 3 large cavities. 10 seed points, randomly selected inside and outside the volume, were used to initialize the level set evolution

reported in Fig.4. During the evolution, components merge, vanish and produce cavities, generating a final accurate representation constituted of 3 spherical cavities and the main object. We note that all the components are easily extracted, since they carry distinct labels that are iteratively updated during the evolution.

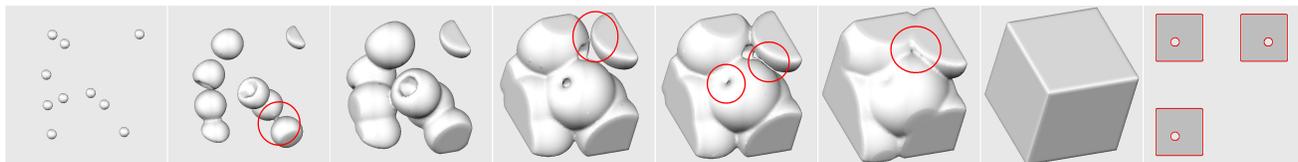


Figure 4: Segmentation of a cube containing 3 cavities. 10 initial seed points were randomly selected. Note how components split, merge and disappear during the evolution, and how the active contour encloses cavities.

## 4.2 Real data

Two segmentation tasks are presented that illustrate the potential benefits of our novel level set framework : the segmentation of cortical surfaces from MRI and the extraction of blood vessels from MRA data sets.

### Experiment 3: Cortical segmentation

Excluding pathological cases, the cortex, which corresponds to a highly-folded thin sheet of gray matter, has the topology of a sphere. The extraction of accurate and topologically-correct cortical representations is still an active research area. In this example, the cortical surface is initialized with 55 spherical components, automatically selected in a greedy manner, such that every selected point is located at a minimal distance of  $10mm$  from the previous ones (Fig.5). Topology-preserving level sets could not handle such an initialization, since the number of components would remain constant throughout the evolution. As a consequence, only one initial seed could be used, leading to a slower segmentation process and potentially to topological deadlocks. Standard level sets yield a final cortical surface with 18 handles. On the other hand, using our method, the components progressively merge together and enclose cavities, resulting in a final surface composed of 6 spherical components: the cortical surface and 5 small cavities.

### Experiment 4: Segmentation of blood vessels

Finally, we show how our method could be applied to the segmentation of blood vessels from Magnetic Resonance Angiography. Because these vessels do not split and merge, their topology is the one of several distinct components with no handles (i.e. each component has the topology of a sphere). While traditional level sets produce segmentations that could include topological defects, topologically constrained level sets would result in a slow and laborious segmentation. Since the simultaneous evolution of several components, which cannot be merged together, can easily be trapped in topological dead-ends, each component would need to be iteratively initialized, when the evolution of the previous one

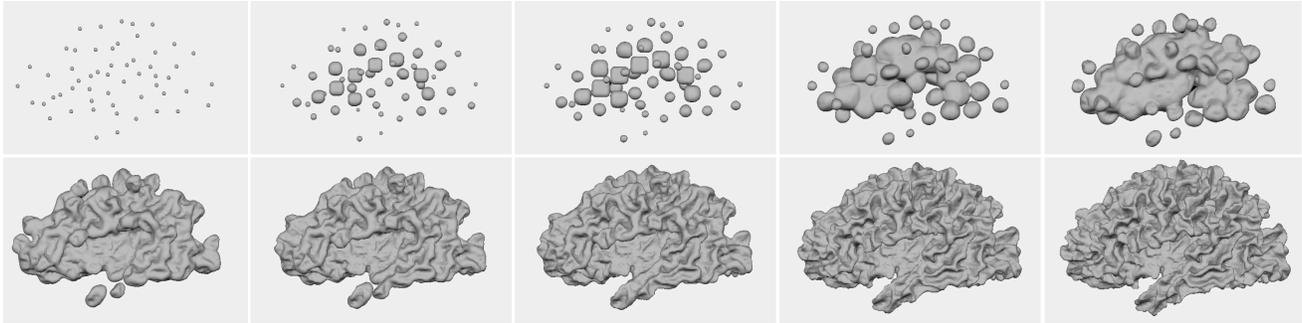


Figure 5: Segmentation of the cortex from an anatomical MRI. The initial level set was constituted of 55 connected components. The final surface has a spherical topology, corresponding to an Euler number of 2. The same level set evolution without topological control results in a final surface with 18 topological defects (Euler number of  $\chi = -34$ )

has terminated. Moreover, when using an initialization with a bounding box, topology-preserving level sets yield a final surface with many geometrical inconsistencies due to topological barriers, displayed in Fig. 7.

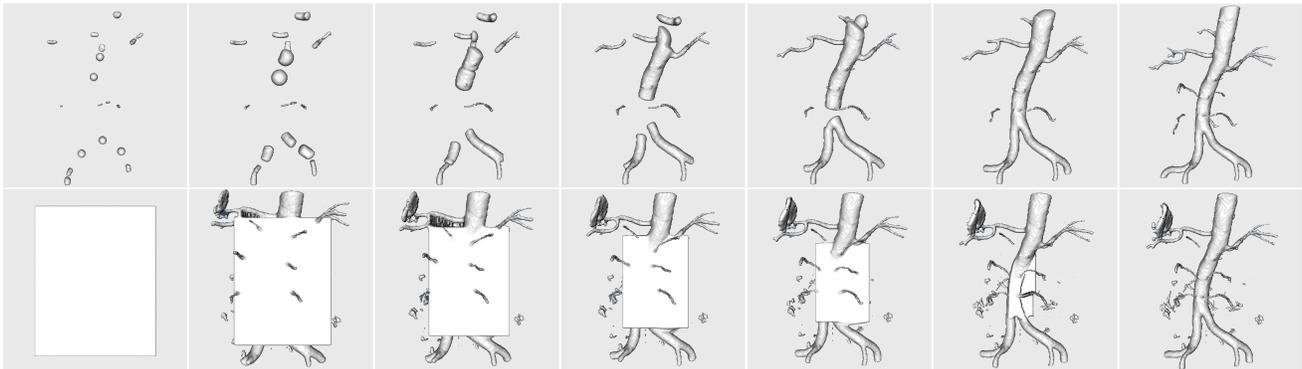


Figure 6: Segmentations of blood vessels in a 3D angiography under two different initializations. Top row: 20 seed points were selected to initialize the active contour, which generates 3 components. Bottom row: An enclosing contour is used to initialize the level set. After 9 merges and 99 splits, the final segmentation is constituted of 91 components, 53 of which were due to random noise.

On the other hand, our method offers the possibility to concurrently evolve multiple components that can merge, split and vanish. The initial contour can be initialized by a set of seed points, manually or automatically selected, or by a single enclosing component, without affecting much the final representation.

Figure 6 shows the segmentation of an angiography under two different initializations. In a first experiment (top row), 20 seed points were automatically selected at the brightest locations in the MRA. The level set evolution iteratively merged most of the components, generating a final segmentation with 3 spherical components. In the second experiment (bottom row), one single global component, enclosing

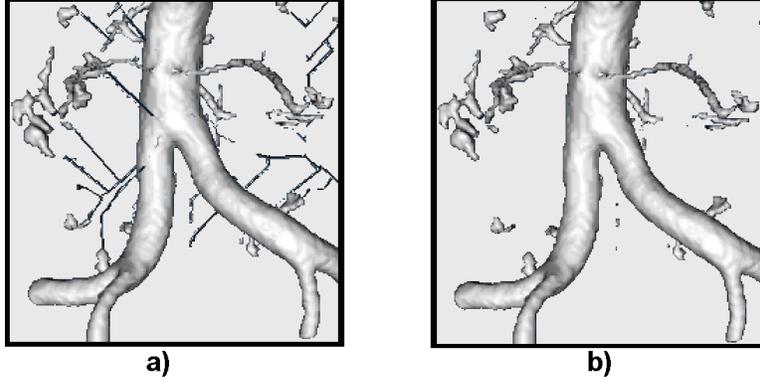


Figure 7: Segmentations of blood vessels from MRA produced by a) a topologically constrained evolution [17], starting from a bounding box b) our genus-preserving level set framework. The image contains several artifacts, which are mostly due to noise. As a consequence, several disconnected components are present in the final segmentation produced by our method. Topologically constrained segmentation fails to segment out these disconnected components and produce an incorrect segmentation.

most of the object of interest, was used to initialize the active contour. During the evolution, 9 components merged and 99 split producing a final segmentation composed of 91 components, 53 of which were single voxel components due to random noise in the imaging process.

## 5 Conclusion

We introduced a new level set framework that offers control over the topology of the level set components during the evolution. Contrary to previous approaches that either do not constrain the topology or enforce a hard topological constraint, our method exerts a subtle control over the topology of each component to prevent the formation of topological defects, such as handles (or cavities depending on the application and the choice of active contour model). Distinct components can merge, split or disappear during the evolution, but no handles are generated. In particular, a contour composed solely of spherical components will only produce spherical components throughout the evolution. In this case, the most common situation in practice, all computations are local and the multisimple point checking can be done efficiently. The only computational complexity comes from the generation of new components, as new labels need to be assigned to each.

While the original level set model does not provide any topological control, topology-preserving level sets impose too restrictive of a constraint. Our framework establishes a trade-off in between the two models. Compared to the former, the formation of new handles and the closing of existing handles are prevented. Compared to the latter, the ability to change topology under certain conditions greatly alleviates the sensitivity to initial conditions. Our framework offers a subtle topological control that alleviates most problems of topologically-constrained methods (i.e. sensitivity to initialization and noise, simultaneous evolution of multiple components and speed of convergence). The experiments presented in this paper illustrate some applications that could potentially benefit from our approach.

Finally, we also note that the proposed framework can be adapted to allow different levels of topological control during the level set evolution. Particularly, the two criteria 2 and 3 can be used to distinguish

different types of voxels, such as the ones leading to a split or a merge of components from the ones generating or destroying cavities, among others.

## References

- [1] D. Adalsteinsson and J.A. Sethian. A Fast Level Set Method for Propagating Interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995.
- [2] E. Bardinet, L.D. Cohen, and N. Ayache. A parametric deformable model to fit unstructured 3D data. *Computer Vision and Image Understanding*, 71(1):39–54, 1998.
- [3] G. Bertrand. Simple points, topological numbers and geodesic neighborhoods in cubic grids. *Pattern Recognition Letters*, 15(10):1003–1011, 1994.
- [4] G. Bertrand. A boolean characterization of three-dimensional simple points,” pattern recognition letters. *Pattern recognition letters*, 17:115–124, 1996.
- [5] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *The International Journal of Computer Vision*, 22(1):61–79, 1997.
- [6] A.D. Dale, B. Fischl, and Sereno M.I. Cortical surface-based analysis i: Segmentation and surface reconstruction. *NeuroImage*, 9:179–194, 1999.
- [7] C. Davatzikos and R.N. Bryan. Using a deformable surface model to obtain a shape representation of the cortex. *IEEE TMI*, 15:758–795, 1996.
- [8] M. P. DoCarmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.
- [9] Y. Duan, L. Yang, H. Qin, and D. Samaras. Shape reconstruction from 3D and 2D data using PDE-based deformable surfaces. In *European Conference on Computer Vision*, volume 3, pages 238–251, 2004.
- [10] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE’s, level set methods and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344, 1998.
- [11] B. Fischl, A. Liu, and A.M. Dale. Automated manifold surgery: Constructing geometrically accurate and topologically correct models of the human cerebral cortex. *IEEE Transaction on Medical Imaging*, 20:70–80, 2001.
- [12] Pascal Fua and Yves G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *The International Journal of Computer Vision*, 16(1):35–56, September 1995.
- [13] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Cortex segmentation: A fast variational geometric approach. *IEEE TMI*, 21(2):1544–1551, 2002.
- [14] B. Goldlücke and M. Magnor. Space-time isosurface evolution for temporally coherent 3D reconstruction. In *International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 350–355, 2004.
- [15] I. Guskov and Z. Wood. Topological noise removal. *Graphics I proceedings*, pages 19–26, 2001.
- [16] X. Han, C. Xu, U. Braga-Neto, and J.L. Prince. Topology correction in brain cortex segmentation using a multiscale, graph-based approach. *IEEE Transaction on Medical Imaging*, 21(2):109–121, 2002.

- [17] X. Han, C. Xu, and J.L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):755–768, 2003.
- [18] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.
- [19] H. Jin, S. Soatto, and A.J. Yezzi. Multi-view stereo beyond Lambert. In *International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 171–178, 2003.
- [20] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *The International Journal of Computer Vision*, 1(4):321–331, 1987.
- [21] N. Kriegeskorte and R. Goeble. An efficient algorithm for topologically segmentation of the cortical sheet in anatomical mr volumes. *NeuroImage*, 14:329–346, 2001.
- [22] W.E. Lorensen and H.E. Cline. Marching cubes: A high-resolution 3D surface reconstruction algorithm. *ACM Computer Graphics*, 21(4):163–170, 1987.
- [23] D. MacDonald, N. Kabani, D. Avis, and A.C. Evens. Automated 3d extraction of inner and outer surfaces of cerebral cortex from mri. *NeuroImage*, 12:340–356, 2000.
- [24] J.-F. Mangin, V. Frouin, I. Bloch, J. Regis, and J. Lopez-Krahe. From 3d magnetic resonance images to structural representations of the cortex topography using topology preserving deformations. *Journal of Mathematical Imaging and Vision*, 5:297–318, 1995.
- [25] D.N. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.
- [26] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.
- [27] N. Paragios and R. Deriche. Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding*, 97(3):259–282, 2005.
- [28] J.-P. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *The International Journal of Computer Vision*, 72(2):179–193, 2007.
- [29] Jean-Philippe Pons. *Methodological and Applied Contributions to the Deformable Models Framework*. PhD dissertation, Ecole Nationale des Ponts et Chaussées, November 18 2005.
- [30] F. Ségonne, J. Pacheco, and B. Fischl. Geometrically-accurate topology simplification of triangulated cortical surfaces using non-separating loops. *IEEE TMI*, 26(4):518–529, 2007.
- [31] F. Ségonne, J.-P. Pons, E. Grimson, and B. Fischl. A novel level set framework for the segmentation of medical images under topology control. In *Workshop on Computer Vision for Biomedical Image Applications: Current Techniques and Future Trends*, pages 135–145, 2005.
- [32] Florent Ségonne. *Segmentation of Medical Images under Topological Constraints*. PhD dissertation, Massachusetts Institute of Technology, December 12 2005.
- [33] D.W. Shattuck and R.M. Leahy. Automated graph based analysis and correction of cortical volume topology. *IEEE TMI*, 20(11):1167–1177, 2001.

- [34] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D.J. Kriegman. Parameterized families of polynomials for bounded algebraic curve and surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):287–303, March 1994.
- [35] C. Xu, D.L. Pham, M.E. Rettmann, D.N. Yu, and J.L. Prince. Reconstruction of the human cerebral cortex from magnetic resonance images. *IEEE TMI*, 18:467–480, 1999.
- [36] A.J. Yezzi and S. Soatto. Deformation: Deforming motion, shape average and the joint registration and approximation of structures in images. *The International Journal of Computer Vision*, 53(2):153–167, 2003.
- [37] H. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and non-parametric shape reconstruction from unorganized points using a variational level set method. *Computer Vision and Image Understanding*, 80(3):295–314, 2000.