Topology Noise Removal for Curve and Surface Evolution

Chao Chen and Daniel Freedman

Abstract. In cortex surface segmentation, the extracted surface is required to have a particular topology, namely, a two-sphere. We present a new method for removing topology noise of a curve or surface within the level set framework, and thus produce a cortal surface with correct topology. We define a new energy term which quantifies topology noise. We then show how to minimize this term by computing its functional derivative with respect to the level set function. This method differs from existing methods in that it is inherently continuous and not digital; and in the way that our energy directly relates to the topology of the underlying curve or surface, versus existing knot-based measures which are related in a more indirect fashion. The proposed flow is validated empirically.

1 Introduction

Active contour model, first introduced by Kass *et al.* [KWT88], is a well-known tool to perform the task of image segmentation, namely, computing a *contour* which separates the image (the *domain of interest*) into two parts, inside and outside. In such model, one evolves an initial contour according to the prescribed differential equations, until a steady-state is reached. The steady-state is then taken to be the desired contour.

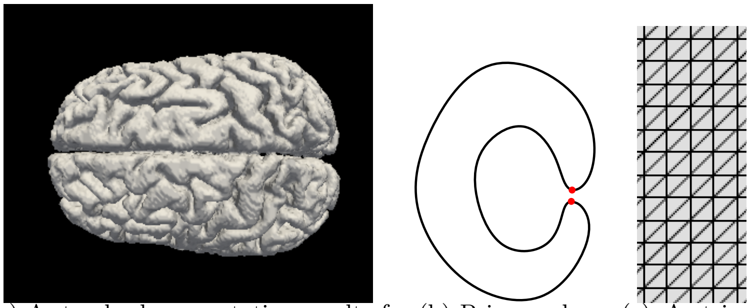
In the level set framework, the contour is implicitly represented as the zero level set of a *level set function*, $\phi : \Omega \to \mathbb{R}$, where Ω is the domain of interest. ϕ is often taken as the *signed distance function*, whose absolute value is the distance to the contour, and whose sign is negative for points inside the contour and positive for points outside the contour. [1] Instead of evolving the contour, one then evolves ϕ and takes the zero level set of its steady-state as the final contour. In this paper, we take ϕ as the signed distance function for ease of illustration. However, our method could be applied to any level set function.

The level set method brings various advantages: parameterization free, extendable to any dimension, numerically stable. More importantly, it can handle changes in contour topology for free, due to the implicit representation.

However, in certain applications, the topological flexibility is not desirable. The framework cannot distinguish meaningful topological features from noise, even if we have prior knowledge of the contour's topology. For example, in segmentation of a human cortex (Figure 1(a)), it would be beneficial if the extracted

[1] In some instances in the literature, the sign convention is the opposite of what we have described.

surface is homeomorphic to a two-sphere [HXP03]. However, due to the complex geometry of a cortex surface, standard level set method (i.e. geodesic active contour [CKS97,KKO^{+}95], Chan-Vese [CV01]) would not achieve the correct topology. The narrow seam between the two hemispheres is especially challenging. In Figure 4(e), we show part of a standard segmentation result, namely, the part between the two hemispheres. Many holes appear, corresponding to small handles between the two hemispheres.



(a) A standard segmentation result of a human cortex, with topology noise.

(b) Prior works.

(c) A triangulation

Fig. 1.

Prior Works. Han *et al.* [HXP03] proposed a digital algorithm which prevent topology changes during the evolution. Whenever a pixel's level set function value is updated and the sign of the function value changes, a check is applied to this pixel to make sure that it does not change the (digital) topology of the contour. If it does lead to a topological change, the pixel's function value is only updated to almost zero yet with the same sign as its old value. However, this topology control is detached from the energy minimization which drives the rest of the level set evolution. As a result, the method leads to undesirable artifacts, such as a contour which is a single pixel away from the wrong topology. Various digital topology methods have been developed, to achieve different topology constraints [BP07,Ség08].

In order to produce more natural results, other methods [SY07,GV08] incorporate topology control through the introduction of an extra term in the energy functional to be minimized. The energy minimization framework then naturally encourages the contour to have the correct topology. These methods use a common intuition, based on knot theory: penalize the energy when two different parts of the contour meet (the two red points in Figure 1(b)).

All these methods prevent topology changes and maintain a same topology through the evolution. However, they cannot fix incorrect topology that already exists. This topological rigidity is inconsistent with the levelset framework (Please see the end of this section for more discussion). In addition, the energy-based methods only prevent topology changes which arise from the merging of components. However, there are other types of topology change which may occur, such as the creation of new components which are far away from other components of the contour.

Contributions. We propose a new method for topology control within the level set framework. In this paper, the goal is to ensure that the contour (d-manifold) is homeomorphic to a d-sphere; however, the method can be extended to more general topological priors (Section B). For the remainder of this paper, "correct topology" will mean the topology of the d-sphere. We have two major contributions.

1, A measure of the contour's topology noise. Based on a recent theoretical result in computational topology [EMP09], we define the *total robustness* of a contour, which, in a sensible way, indicates how close a contour is to having the correct topology. This measure is the sum of the robustness of individual topology features, where the robustness of a class measures how easy it is to "shake off" this feature by perturbing the contour.

2, A flow which drives the contour to the correct topology. Using the concept of total robustness, we compute a flow which pushes the level set function and thus the contour towards the correct topology. Specifically, we compute the functional derivative of the total robustness with respect to ϕ, and evolve ϕ according to gradient descent. On its own, this flow leads to a global minimum of the total robustness, and thus a contour with correct topology. Note that in practice, this flow will be combined with other flows, such as those designed for image segmentation; in this scenario, the global minimum property may be lost.

Unlike previous works, our method allows flexibility in the topology of the contour *during the evolution*, while ensuring that the topology of the *final* contour is correct. This is beneficial for two reasons: (1) the initial contour could have a different topology than the desired output contour (i.e. results from other segmentation model); (2) we do not need to worry about potential topology changes due to the discretization of the time step. Furthermore, compared with energy-based methods, our method addresses all types of "incorrect" topology, not just those which arise from merging of components as in [SY07,GV08].

Our flow changes the evolving level set function as locally as possible, and thus, will not change the geometry of the contour much. This flow could be used to postprocess a standard segmentation result, namely, correct the topology of the result. In such scenario, it plays a similar role to topology correcting methods [SPF07,YDG09], which locate topology defects of a given surface and correct them locally. Unlike these methods, our method retains the level set framework due to the advantages described above (numerical stability, lack of need for parameterization, etc.), as well as the fact that it is the "native language" for many applications in vision and graphics. Furthermore, our flow could be easily combined with other flows (i.e. geodesic active contour), so that the correction result is natural (See Section 4).

In this paper, for ease of exposition, we focus our discussion on the case when the contour C is a one-manifold and the domain is a 2D image ($\Omega \subset \mathbb{R}^2$). However, our algorithm can be naturally extended to d-manifolds in $(d+1)$-

dimensional domain. In specific, we implemented and experimentally verified our method on 3D images.

2 Background

Topology Features. In this paper, instead of talking about topology of the contour, C, we focus on the topology of the object enclosed by the contour, $O = \phi^{-1}(0, \infty] = \{x \in \Omega \mid \phi(x) \leq 0\}$. It is provable that if two objects have different topology, then the corresponding contours have different topology. [2] The "correct topology" constraint, namely, C is homeomorphic to a d-sphere, is equivalent to the constraint that O is homeomorphic to a $(d+1)$-dimensional ball.

The topology feature we discuss are homology classes over $\mathbb{Z}_2$ field, which are well studied in algebraic topology (see [Mun84] for a formal introduction). In 2D images, 0D and 1D classes of an object are the components and the holes, respectively. For example, in Figure 3(a), the object has four components and three holes. In 3D image, 0D, 1D and 2D classes are the components, handles, and voids of the object. Formally, the modulo-2 sum of any set of homology classes is also a homology class. The group of classes form a vector space. In this paper, however, we only focus on a canonical basis of such vector space, and call this basis the set of topology features. For example, in Figure 3(a), we only consider the set of four components and the set of three holes, denoted as $\mathsf{H}_0(O)$ and $\mathsf{H}_1(O)$ respectively. Their union is denoted as $\mathsf{H}(O)$ for convenience.

Critical Points. Given a domain Ω and a function $\phi : \Omega \to \mathbb{R}$, as we increase a threshold t from $-\infty$ to $+\infty$, the sublevel set, $\phi^{-1}(-\infty, t]$, grows from the empty set to the entire domain. During this process, topology of the sublevel set come into existence ("birth") and disappear ("death"). The points in the domain at which such topology changes happen are called *critical points*. Their function values are *critical values*. This notation, defined by Cohen-Steiner *et al.* [CSEH07], is an extension of Morse theory. Please note that we assume zero is a *regular value*, namely, a value which is not critical. In other words, we assume no topology change happens at $t = 0$.

In Figure 2(Left), on the top, we show a contour as well as the object enclosed, which has 3 components. Below it, we draw the graph of the corresponding signed distance function ϕ (using ϕ as the z coordinate). For ease of understanding, we also draw the contour in the graph, which is the intersection of the graph and the $\{z = 0\}$ plane. There are seven critical points, highlighted with red marks. Four *minimal* points, $m_0, \cdots, m_3$, are the birth places of four components, born at $\phi(m_0), \cdots, \phi(m_3)$ respectively. Three *saddles* s_1, s_2, s_3 are the the points at which components are merged, corresponding to the death of 3 components. There are also *maximal* points, at which holes disappear during the growth (there are three maximal points in Figure 3(a)).

In application, we compute these critical points and their characteristic (minimal, saddle, or maximal) in a discretized framework. We triangulate the image,

[2] The converse, however, is not true. For example, given a contour homeomorphic to the disjoint union of two one-spheres, the object enclosed could be either an annulus or the disjoint union of two disks.

Ω, using the set of pixels as vertices (Figure 1(c)). We approximate the growth of the sublevel set, $\phi^{-1}(-\infty, t]$, by a growing subcomplex, namely, a subset of simplices (vertices, edges and triangles). Each simplex, σ, has a distinct function value and enters the growing subcomplex as soon as the threshold $t \geq \phi(\sigma)$. Critical points then correspond to simplices which changes the topology of the growing subcomplex. Although there are other ways to detect critical points (i.e. [BP07]), this method, known as the *persistence homology algorithm* [EH09], is the foundation of the algorithm to compute robustness, which will be used later.

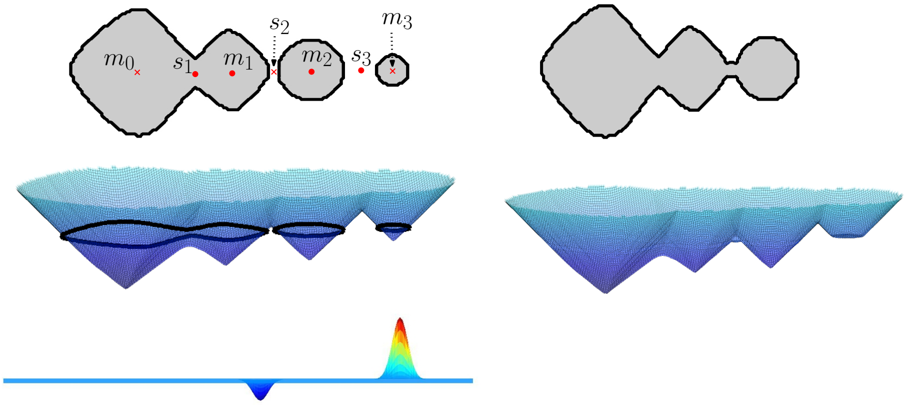


Fig. 2. Left: a contour with topology noise, the graph of its signed distance function, ϕ, and the graph of the flow, $\partial\phi/\partial t = -\delta E/\delta\phi$.
Right: a perturbation of ϕ and its contour. Topology noise is removed.

Robustness. Edelsbrunner *et al.* [EMP09] defined a measure of homology classes of the object O, $\rho_\phi : \mathsf{H}(O) \to \mathbb{R}$. Intuitively, given a homology class $\alpha \in \mathsf{H}(O)$, its *robustness*, $\rho_\phi(\alpha)$ measures how easily it is to kill α by perturbing the function. In other words, $\rho_\phi(\alpha)$ is the minimal error we can tolerate to get a function which approximates ϕ and kills α. Different classes of an object has different robustness. The ones with small robustness are considered noise, and will be removed by our method.

Given $r \geq 0$, we call $h : \Omega \to \mathbb{R}$ an *r-perturbation* of ϕ if the L_∞ distance between the two functions is upperbounded by r, formally, $\|h - \phi\|_\infty = \max_{x\in\Omega} |h(x) - \phi(x)| \leq r$. Define the robustness, $\rho_\phi(\alpha)$, as the minimal r such that there exists an r-perturbation of ϕ, h, such that α is dead in the perturbed object, $h^{-1}(\infty, 0]$. By dead, we mean that α either disappears or is merged with some other class which is more robust.

For example, in Figure 2(Left), to kill the middle component of O, we have two options: (1) Increase function values of points near the minimal point m_2. In the function graph, this is equal to pushing the cone tip up so that $h(m_2) = 0 + \epsilon$, ϵ positive and arbitrarily close to zero. The component would then disappear in $h^{-1}(-\infty, 0]$. (2) Decrease function values of points near the saddle point s_2.

In the function graph, we drag the saddle down so that $h(s_2) = 0 - \epsilon$. The component would then merge with the middle component, which is more robust.

The two options lead to two new functions, h_m and h_s, which are $|\phi(m_2)|$- and $|\phi(s_2)|$-perturbations of ϕ, respectively. Since here $|\phi(m_2)| > |\phi(s_2)|$, we choose the the second option as the best perturbation, whose L_∞ distance from ϕ, $|\phi(s_2)|$, is thus the robustness of the corresponding component. Similarly, we determine that the robustness of the right component and the left component are $|\phi(m_3)|$ and $|\phi(m_0)|$ respectively (note that $|\phi(m_3)| < |\phi(s_3)|$). In Figure 2(Right), we show a $|\phi(m_3)|$-perturbation which kills the right and the middle component, but not the left one.

It is noticeable that the robustness has a close relationship with critical values and critical points. In fact, each class $\alpha \in \mathsf{H}(O)$ corresponds to a specific critical point, c_α. The absolute value of the corresponding critical value is the robustness, namely, $\rho_\phi(\alpha) = |\phi(c_\alpha)|$. Bendich *et al.* [BEMP09] provided an *robustness algorithm* to compute the robustness of each homology class of the object, as well as the associated critical point. The algorithm is based on the persistent homology algorithm, and takes cubic time in the worst case and almost linear time in practice.

In general, given a function ϕ, there are a lot of critical points, each is relevant to the topology of some sublevel set. Using robustness algorithm, we could identify those which are the most relevant to the topology of the object O. Changing their neighborhood would kill topology noise. And furthermore, this changing is local and thus will not harm the geometry of the object much. In Figure 3(a), we show a contour and its object, with topology noise. The level set function ϕ has many critical points (red marks). The robustness algorithm finds the 6 critical points corresponding to the 6 undesired homology classes of O (Figure 3(b)). Changing their neighborhood leads to a perturbation of ϕ, and thus a new object with no topology noise (Figure 3(c)).

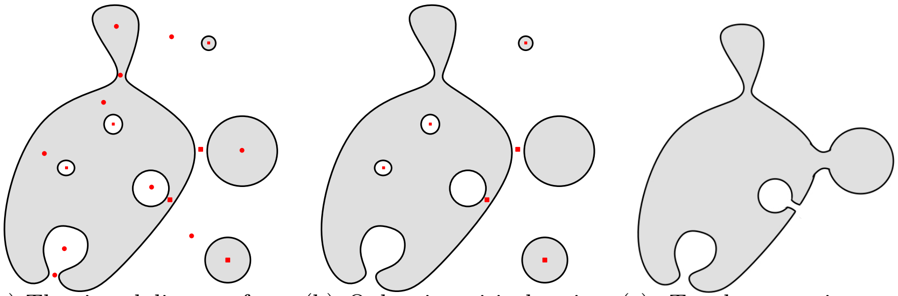
Although the algorithm could identify relevant critical points, it is not clear how to change the local neighborhood of them to get the correct topology while keeping the geometry of O as intact as possible. In this paper, we define an energy term defined on ϕ. This energy term is minimized if and only if the contour/object has the correct topology. We then use gradient descent to iteratively evolve the level set function until the contour/object has the correct topology. This gradient only changes a small neighborhood of these relevant critical points and thus ensure the geometry is less influenced.

3 Method

The Energy Term: Total Robustness. Given a contour C, and the level set function ϕ, we define an energy term as the sum of the k-th power of robustness of topology noise. Formally, the *degree-k total robustness* of O is

$$Rob_k(O) = \sum_{\alpha \in \mathsf{H}(O)} \rho_\phi(\alpha)^k - (\max_{\alpha \in \mathsf{H}(O)} \rho_\phi(\alpha))^k. \tag{1}$$

This energy sums over all classes of O except for the most robust one, which is the component we want to keep. This component is born at the global minimal



(a) The signed distance function ϕ has many critical points (red marks).

(b) Only six critical points are relevant to topology noise of O.

(c) Topology noise removed.

Fig. 3. A contour with topology noise.

point, c_{min}. Its robustness is $\phi(c_{min})$. Thus the last term of Equation (1) is $-|\phi(c_{min})|^k$.

Assuming O has at least one component, the total robustness is non-negative. It reaches its global minimum, zero, if and only if the contour/object has the correct topology.

We are now ready to compute the flow which drives the contour C, and its signed distance function ϕ, towards the minimum of the total robustness. Denoting the energy $E(\phi) = Rob_k(\phi)$, the goal is to compute the functional derivative $\delta E/\delta\phi$, and thus the desired flow $\partial\phi/\partial t = -\delta E/\delta\phi$. By gradient descent, the flow minimizes E.

Basing on the robustness theory, we could prove that the difference between two level set functions upperbounds the difference between their total robustness. Therefore, the energy term E is continuous. Unfortunately, E is not differentiable everywhere. We will now compute the derivative for cases when E is differentiable, and explain the intuition. The non-differentiable cases will be discussed in Appendix A. We prove that the set of functions at which E is non-differentiable is measure zero in the space of functions (Theorem 1). This theorem leads to the construction of a smoothed approximation of E, whose corresponding flow can be efficiently computed.

For the rest of the section, we use degree-3 total robustness as the energy, $E(\phi) = Rob_3(\phi)$.

The Functional Derivative. We first assume that all critical points have distinct values. In this case, E is differentiable. Recall that c_α is the critical point of ϕ associated to a homology class $\alpha \in \mathsf{H}(O)$, such that $\phi(c_\alpha) = \rho_\phi(\alpha)$. We can rewrite E as a function depending on ϕ and relevant critical points,

$$E(\phi) = \sum_{\alpha\in\mathsf{H}(O), c_\alpha\neq c_{min}} sign\left(\phi(c_\alpha)\right)\phi^3(c_\alpha)$$

The set of relevant critical points $\{c_\alpha|\alpha \in \mathsf{H}(O)\}$ depends on the function ϕ and can be computed by running the robustness algorithm once. Recall that the algorithm discretizes the image into a triangulation and approximate the growing sublevel set by a growing subcomplex. In this case, a sufficiently small perturbation of ϕ will preserve the order in which simplices entering the growing subcomplex. This would lead to the fact that the output of the algorithm remains almost the same. In specific, the critical point associated to each homology class remains the same. The sign of its critical value remains the same. Only its critical *value* will change (according to the perturbation).

Finally, we can rewrite $E(\phi)$ as a functional and compute the functional derivative.

$$E(\phi) = \int\limits_{\Omega} \left(\sum_{\alpha \in \mathsf{H}(O), c_\alpha \neq c_{min}} \delta(x - c_\alpha) \right) sign\left(\phi(x)\right) \phi^3(x) dx$$

$$\frac{\delta E}{\delta \phi} = 3 \left(\sum_{\alpha \in \mathsf{H}(O), c_\alpha \neq c_{min}} \delta(x - c_\alpha) \right) sign\left(\phi(x)\right) \phi^2(x) \tag{2}$$

where $\delta(x - c_\alpha)$ is a Dirac delta function. The second equation is due to the Euler-Lagrange equation and the fact that c_α's and $sign(\phi(c_\alpha))$'s are constant for a small perturbation of ϕ. In our implementation, we use a standard trick from the level set literature and relax $\delta(x - c_i)$ to a bounded C^1 approximation with width σ, $\delta_\sigma(x - c_i)$, e.g. a Gaussian with variance σ^2.

Next, we illustrate the intuition of the functional derivative in Equation (2). The derivative $\delta E/\delta\phi$ is the weighted sum of a set of Dirac delta functions (or Gaussians in the relaxed case), centered at relevant critical points, except for c_{min}. The weights are proportional to the square of the corresponding critical values multiplied by its sign. During the evolution, we update ϕ according to $\partial\phi/\partial t = -\delta E/\delta\phi$. This pushes function values near the relevant critical point towards zero, and thus shrinks the corresponding robustness. In Figure 2(Left), we draw the graph of the flow, $\partial\phi/\partial t$, at the bottom. The flow lifts the minimal point, m_3, corresponding to the right component. For the middle component, the flow suppresses the saddle point, s_2. The rate of lifting or suppression is proportional to the square of the corresponding robustness, so that classes with large robustness have their robustness decreased faster than those with small robustness.

Although the energy is not everywhere differentiable, we could construct a smoothed approximation E_λ. The functional derivative $\delta E_\lambda/\delta\phi$ is the same as Equation (2) at ϕ further away from the non-differentiable set, and can be computed by Monte Carlo method otherwise. Please see Appendix A for detailes.

Global Minimum. Typically gradient descent leads only to a local minimum of the energy. In our case, we in fact achieve the global minimum, i.e. zero energy. The reason is that Equation (2) is zero if and only if one class left, when $E(\phi) = 0$. Therefore, the output of our flow has the "correct topology".

4 Implementation and Experimental Results

For implementation of the level set method, we use the freely available implementation by Zhang [Zha]. We compute robustness using our own implementation of the robustness algorithm [BEMP09]. The flow is then computed according to Equation (2).

We use our method as a postprocessing step. We first apply a standard segmentation algorithm (GAC or Chan-Vese). Using this output as an initial contour, we evole with our flow to correct the topology. Please note that this scheme will not fit other topological control methods, as they require that the topology be held constant throughout the evolution.

We verify the method on medical images. In Figure 4(a) and 4(b), we segment the image of a slice of brain white matter. Our method is able to correct various types of topological noise, and recovers a contour homeomorphic to a one-sphere. Please note that these images are only a proof of concept. The boundary of a slice of brain white matter is not necessarily homeomorphic to a one-sphere.

In a 3D MR image, however, a cortex surface would be homeomorphic to a two-sphere. We verify our method with a 3D MR image (from the example data of FreeSurfer [Fre]). We compare the initial surface (the result of a standard segmentation) and the surface after applying our flow. In Figure 4(c) and 4(d), we show a handle that is removed after applying our flow. In Figure 4(e) and 4(f), we show the part of the cortex surface between the left and right hemispheres. Note the holes in the surface corresponds to handles between the two hemispheres. Our flow will fix the topology by either remove small handles, or merging handles nearby. Therefore, after applying the flow, the surface has only one big hole left.

We note the choice of the width of the Gaussian kernel, σ, is essential in the converging speed. The bigger σ is, the faster the result converges. On the other hand, σ desides how natural the fixing result could be. In our experiments, we use a Gaussian kernel with $\sigma = 0.5$. Further study of this parameter will be done in the future.

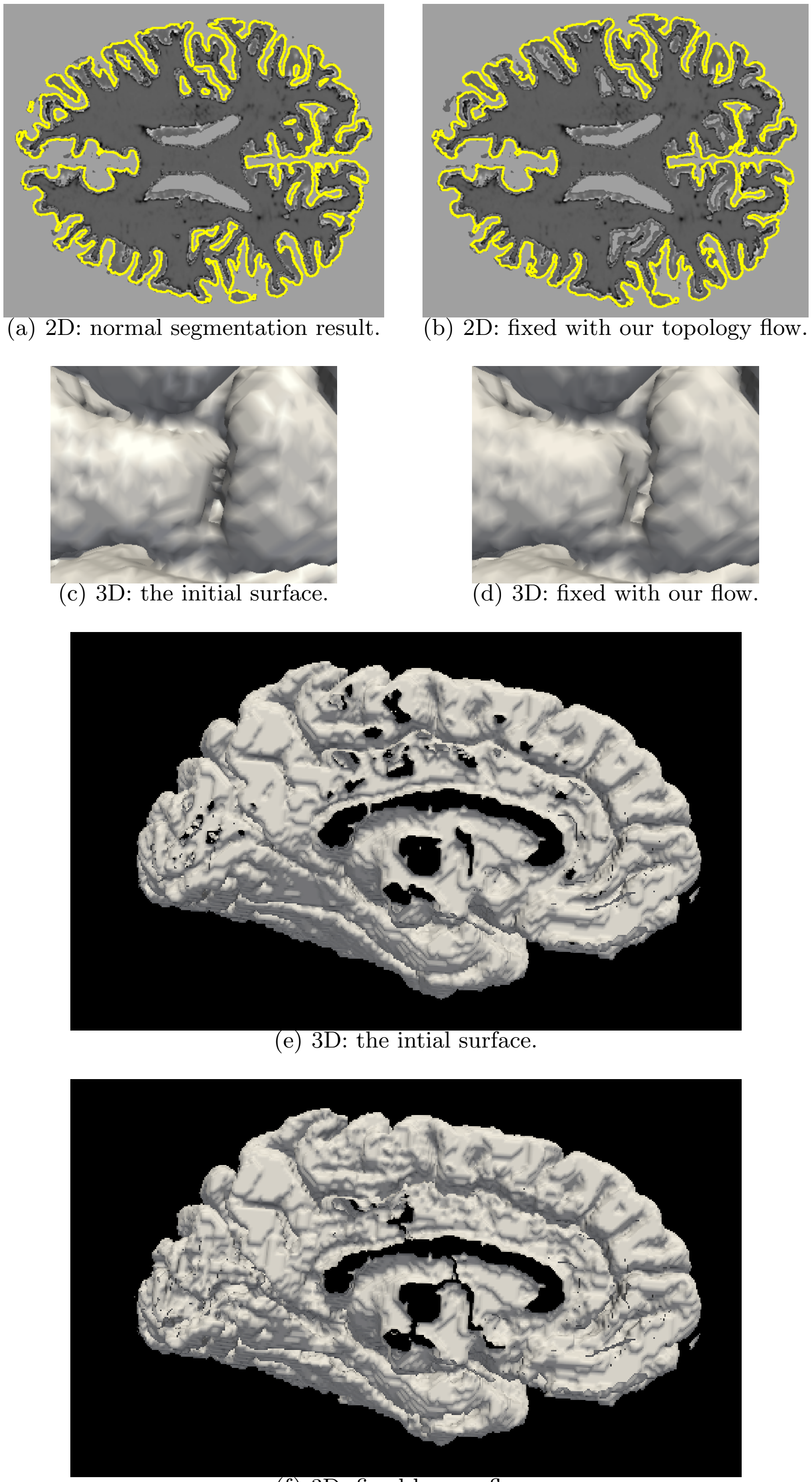
We run experiments on a computer with a 2.53GHz Xeon Processor and 96GB RAM. For a 2D image with 200×200 pixels, the evolution takes 15 iterations. Each iteration costs 2 to 3 minutes. For a 3D image with 256×256×256 voxels, the evolution takes 40 iterations. Each iteration costs 15 to 18 minutes. One bottleneck of our method is the memory cost. For a 200×200 2D image, the memory cost is 500MB. For a 256×256×256 3D image, the memory cost is 70GB. This has close relationship with the huge number of simplices in a triangulation. For example, for a 256×256×256 3D image, the number of simplices is 26×16.5 million. We are currently developing another robustness algorithm, which would only rely on the total number of voxels, which is 16.5 million.

References

[BEMP09] Paul Bendich, Herbert Edelsbrunner, Dmitriy Morozov, and Amit Patel. Robustness of level sets and interval preimages. preprint, Dec. 2009.

[BP07] Pierre-Louis Bazin and Dzung L. Pham. Topology correction of segmented medical images using a fast marching algorithm. *Computer Methods and Programs in Biomedicine*, 88(2):182–190, 2007.

[CKS97] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.



(a) 2D: normal segmentation result.

(b) 2D: fixed with our topology flow.

(c) 3D: the initial surface.

(d) 3D: fixed with our flow.

(e) 3D: the intial surface.

(f) 3D: fixed by our flow.

Fig. 4.

[CSEH07] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37:103–120, 2007.

[CV01] Tony F. Chan and Luminita A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[EH09] Herbert Edelsbrunner and John Harer. *Computational Topology. An Introduction.* Amer. Math. Soc., 2009.

[EMP09] Herbert Edelsbrunner, Dmitriy Morozov, and Amit Patel. Quantifying transversality by measuring the robustness of intersections. preprint, 2009.

[Fre] Freesurfer. `http://surfer.nmr.mgh.harvard.edu/`.

[GV08] Carole Le Guyader and Luminita A. Vese. Self-repelling snakes for topology-preserving segmentation models. *IEEE Transactions on Image Processing*, 17(5):767–779, 2008.

[HXP03] Xiao Han, Chenyang Xu, and Jerry L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(6):755–768, 2003.

[KKO$^+$95] Satyanad Kichenassamy, Arun Kumar, Peter J. Olver, Allen Tannenbaum, and Anthony J. Yezzi. Gradient flows and geometric active contour models. In *ICCV*, pages 810–815, 1995.

[KWT88] Michael Kass, Andrew P. Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[Mun84] J. R. Munkres. *Elements of Algebraic Topology.* Addison-Wesley, Redwook City, California, 1984.

[Ség08] Florent Ségonne. Active contours under topology control - genus preserving level sets. *International Journal of Computer Vision*, 79(2):107–117, 2008.

[SPF07] Florent Ségonne, J. Pacheco, and Bruce Fischl. Geometrically accurate topology-correction of cortical surfaces using nonseparating loops. *IEEE Trans. Med. Imaging*, 26(4):518–529, 2007.

[SY07] Ganesh Sundaramoorthi and Anthony J. Yezzi. Global regularizing flows with topology preservation for active contours and polygons. *IEEE Transactions on Image Processing*, 16(3):803–812, 2007.

[YDG09] Rachel Aine Yotter, Robert Dahnke, and Christian Gaser. Topological correction of brain surface meshes using spherical harmonics. In *MICCAI (1)*, pages 125–132, 2009.

[Zha] Yan Zhang. The matlab toolbox for 2d/3d image segmentation using level-set based active contour/surface with aos scheme. `http://ecson.org/resources/active_contour_segmentation.html`.

A Non-differentiable Cases.

In this section, we discuss the non-differentiable cases of the energy E.

Previously, we assumed that all critical points have distinct values. If we relax such constraint, there are cases at which E is non-differentiable. For example, in Figure 2(Left), if we let $|\phi(s_3)| = |\phi(m_3)|$, the critical point associated to the right component could be either m_3 or s_3. Thus the functional derivative is not well defined. In general, non-differentiable cases happen when critical points share a same value, causing ambiguity of associating c_α to each class α.

On the other hand, we are able to prove the following theorem. The proof is omitted.

Theorem 1. *The set of points at which $E(\phi)$ is non-differentiable, denoted as ND, is measure 0.*

This theorem means that E is differentiable almost everywhere, leading to the possibility of approximating E with a smoothed version, E_λ, whose derivative can be efficiently computed.

Denote Φ as the space of functions ϕ, we define the smoothed energy term as

$$E_\lambda(\phi) = \int_\Phi E(\xi) G_\lambda(\phi - \xi) d\xi$$

where $G_\lambda : \Phi \to \mathbb{R}$ is a C^1 function defined on function space with a compact support $B_\lambda(0) = \{\xi \in \Phi \mid \|\xi\|_2 \leq \lambda\}$, and which integrates to 1. [3]

Due to Theorem 1 and a change of variables, the derivative is

$$\frac{\delta E_\lambda}{\delta \phi} = \int_{B_\lambda(\phi) - ND} \frac{\delta E}{\delta \phi}(\phi - \xi) G_\lambda(\xi) d\xi$$

For sufficiently small λ, the λ-ball $B_\lambda(\phi)$ around most points ϕ will not intersect the set of non-differentiable functions; as a result, $\partial E_\lambda / \partial \phi \approx \partial E / \partial \phi$, where the latter can be evaluated directly from Equation (2) without an integral. In cases where $B_\lambda(\phi)$ does indeed intersect the set of non-differentiable functions, the integral may be approximated by Monte Carlo methods.

B Discussions

In this paper, we have restricted our discussion to 1D contours in 2D domains. However, our framework can be extended, with very little extra effort, to general d-dimensional contours in $(d+1)$-dimensional domains. The algorithm is implemented for 3D MR image, the original target of our method. Furthermore, our model is not restricted to Euclidean space.

In this paper, our definition of the "correct topology" is when the contour is homeomorphic to a d-sphere (or say, the object is homeomorphic to a $(d+1)$-ball). This topology constraint could profitably be extended. Recall in the definition of total robustness, we ignore the class with the largest robustness. In general, we can use a set of integers $\{N_0, N_1, \dots\}$ such that for each dimension, k, we choose to ignore the N_k largest robustnesses contributed by k-dimensional homology classes. The output contour would then be guaranteed to enclose an object whose number of k dimensional classes is upperbounded by N_k. In specific, in 3D image, we could specify the upperbound of number of components, handles, and voids of the result object.

[3] This is analogous to smoothing a piecewise smooth function defined on Euclidean space by the convolution with a smooth Gaussian kernel.