



Multiphase Image Segmentation Using the Deformable Simplicial Complex Method

Dahl, Vedrana Andersen; Christiansen, Asger Nyman; Bærentzen, Jakob Andreas

Published in:

Proceedings of 22nd International Conference on Pattern Recognition (ICPR 2014)

Link to article, DOI:

[10.1109/ICPR.2014.182](https://doi.org/10.1109/ICPR.2014.182)

Publication date:

2014

Document Version

Early version, also known as pre-print

[Link back to DTU Orbit](#)

Citation (APA):

Dahl, V. A., Christiansen, A. N., & Bærentzen, J. A. (2014). Multiphase Image Segmentation Using the Deformable Simplicial Complex Method. In L. O'Conner (Ed.), *Proceedings of 22nd International Conference on Pattern Recognition (ICPR 2014)* (pp. 1002-1007). IEEE. <https://doi.org/10.1109/ICPR.2014.182>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Multiphase Image Segmentation Using the Deformable Simplicial Complex Method

Vedrana Andersen Dahl
Asger Nyman Christiansen
Jakob Andreas Bærentzen
DTU Compute
Technical University of Denmark
{vand, asny, janba}@dtu.dk

Abstract—The deformable simplicial complex method is a generic method for tracking deformable interfaces providing explicit interface representation, topological adaptivity, and multiphase support. As such, the deformable simplicial complex method can readily be used for representing active contours in image segmentation. We show the benefits of using the deformable simplicial complex method, by presenting an approach for segmenting an image into a known number of segments characterized by distinct mean pixel intensities.

I. INTRODUCTION

Segmentation is one of the basic operations in 2D as well as 3D image processing. In some cases, segmentation is performed by a labeling of pixels. However, often we wish to impose some geometric knowledge about the region being segmented, and this can be done via a *deformable model*. In 2D, a deformable model is a curve that perform the segmentation by evolving under forces derived from the image. Such models are generally classified as either explicit or implicit, depending on the method used for representing the curve. Explicit methods utilize a parametric representation of the curve in a Lagrangian formulation, while implicit methods represent the curve as a level set of two dimensional function which evolves according to an Eulerian formulation. Despite this fundamental difference on how the curve is represented, the underlying principles of both methods are the same [1].

Still, the choice of the curve representation is crucial for the implementation of the deformable model. The explicit methods come with a curve which is accurately represented using a desired resolution. Furthermore, deforming forces are applied directly to the curve. As a drawback, the explicit methods are prone to self-intersections and do not handle topology changes. For implicit methods the biggest benefit is a trivial support of topology changes during evolution. As a drawback, the implicit curve representation is bound to a regular grid. Moreover, it might be difficult to incorporate desired forces in implicit formulation.

If we had a method for representing a curve which is explicit and yet has a natural support for topology changes, we could develop an image segmenting framework, while focusing on modeling the forces governing segmentation and without having to consider curve representation. Deformable simplicial complex (DSC) method developed by Misztal and Bærentzen [2] comes with those two important attributes, and even more. Apart from allowing for explicit curve representation and

robust topological adaptivity the DSC method has a natural multi-phase support.

DSC has been implemented in 2D or 3D and has applications in fluid simulation [3, 4], topology optimization [5] and computing cut loci on Riemannian manifolds [6]. In this paper we present our initial study of using 2D DSC for image segmentation. Our aim was to investigate how DSC method behaves when coupled with image data, with the focus on the way DSC handles topological adaptivity and multiple phases. Therefore we do not utilize image features and we leave a development of such a segmentation for the future work. We chose to consider an image which is to be segmented into a known number of phases with a distinctive mean pixel intensity. This choice is made because we aim at test the performance of the DSC based image segmentation in a well known setting.

For the same reason, we decided to seek inspiration for the forces governing segmentation among the popular and well tested methods, which formulate image segmentation problem in the formalism of the deformable models, using either the explicit or the implicit representation of the curve. We bring an overview of such methods in the following section.

II. BACKGROUND

The basic principle of deformable models is to perform image segmentation by evolving a curve in an image. The curve moves under the influence of *external forces*, which are computed from the image data, and *internal forces* which have to do with the curve itself. The popularity of deformable models is largely due to the well known snakes method by Kass et al. [7]. Snakes utilize an explicit curve representation $\mathbf{X}(s, t) = (X(s, t), Y(s, t))$ where $s \in [0, 1]$ is arclength and $t \in \mathbf{R}^+$ is time, which in a discrete setting reduces to a sequence of points. Such a curve evolves with external and internal forces

$$\frac{\partial \mathbf{X}}{\partial t} = \mathbf{F}_{\text{ext}}(\mathbf{X}) + \mathbf{F}_{\text{int}}(\mathbf{X}). \quad (1)$$

In a classical snakes formulation, the external forces are edge based and they pull the curve towards the image locations with a large gradient magnitude. The internal forces discourage stretching and bending of the curve segments,

$$\mathbf{F}_{\text{int}} = \frac{\partial}{\partial s} \left(\alpha \frac{\partial \mathbf{X}}{\partial s} \right) + \frac{\partial^2}{\partial s^2} \left(\beta \frac{\partial^2 \mathbf{X}}{\partial s^2} \right), \quad (2)$$

with weights α and β controlling the elasticity and rigidity term.

Following the success of snakes, an abundance of external forces has been proposed (see [8] for a comprehensive overview), most aiming at increasing the attraction range of external forces, as in e.g. balloons [9]. For challenging segmentation tasks, the forces allowing for user interaction have also been suggested as e.g. volcano forces [7].

The deformable models utilizing an explicit curve representation have difficulties dealing with topology changes, such as merging or splitting. Whenever such a change occurs a costly reparametrization scheme is needed [10].

As the solution to the problem of topological adaptivity, the approaches utilizing implicit curve representation quickly followed [11, 12]. The curve is here represented as a zero level set of a 2D scalar function $\phi(x, y, t)$ defined on the image. The curve is evolved in accordance with a level set [13] equation

$$\frac{\partial \phi}{\partial t} = v(\kappa) \nabla \phi, \quad (3)$$

where $\nabla \phi$ denotes the gradient of ϕ (normal direction of the curve), κ is the curvature at the zero level set and $v(\kappa)$ is a given speed function coupled with the image data. In original formulations [11, 12] a speed function is constructed such that it vanishes in image locations of the large gradient magnitude. The evolution with therefore stop when the curve it meets an edges of the segmentation object. To ensure that the curve does not leak through, additional stopping terms to a speed function have been proposed e.g. in [14].

A common drawback of the methods described so far is that they are edge based, which provides only a local support and is unsuitable for noisy images or for segmenting objects with weak edges. Region based methods [15, 16] combine the information obtained from the image segmentation to improve the robustness of the model. Those methods use level set formulation of Mumford-Shah [17] minimization problem seeking a piecewise smooth (or, as a reduced case, a piecewise constant) approximation of an image. The term stopping the evolution of the curve depends on the obtained segmentation of the image. The level set equation for two-phase case (an object and a background) and without the term regularizing the length of the curve is

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= [(I - m_{\text{in}})^2 - (I - m_{\text{out}})^2] \nabla \phi \\ &= (m_{\text{out}} - m_{\text{in}})(I - m_{\text{out}} + I - m_{\text{in}}) \nabla \phi. \end{aligned} \quad (4)$$

Here m_{in} and m_{out} are current means of the area inside the curve and outside the curve and I is used to denote the image intensity. This speed function can be viewed as a signed pressure force, such that the curve shrinks where it is outside the object and expands where it is inside the object, yielding a flow that pulling two mean estimates apart.

A general idea of representing more than two phases is to use more than one level set function. The straightforward solution is to represent each phase with one level set as in [18] where a additional energy is introduced to minimize (but not eliminate) overlap and vacuum. The regional based multiphase framework proposed in [19] uses a hierarchical approach where

they subsequently segment previously obtained segments, allowing for a segmentation of an unknown number of phases, but having the limitation of segmenting all triple points as T-junctions. In [20] 2^n phases can be represented using n level sets by assigning an unique combination of n positive or negative signs to each phase, with the drawback of a possible bias where more than one phases meet.

Segmentation images using DSC method complies fully with the formal framework of the deformable models. All forces previously suggested in either explicit or implicit formulation can be applied on the curves represented with DSC. Furthermore, DSC comes with a natural multiphase support, allowing us to define multiphase segmentation without the tedious level set coupling due to the nature of the representation of the curve.

III. METHOD

The DSC method is a key ingredient of our approach. We therefore include a description of the principles and the features of the DSC method with a focus on the 2D variant. The readers interested in implementational details or the 3D version are referred to work by Misztal and Bærentzen [2]. Even though DSC has a natural multiphase support, we start by considering a two-phase case, which is later generalized to a multiphase case.

A. The Deformable Simplicial Complex Method

Like the level set method, the DSC method is a method for dealing with deformable interfaces. While level sets are defined on a regular grid in either 2D or 3D, the DSC is defined on a simplicial 2-complex or 3-complex, corresponding to the triangularization in 2D or tetrahedralization in 3D. Just as the sign of the level set function determines whether a grid point is inside or outside the curve, the simplices of the largest dimension in DSC (triangles in 2D and tetrahedra in 3D) are labeled either *inside* or *outside*. As a result, a DSC interface (curve in 2D and surface in 3D) is represented as a set of lower dimension simplices (line segments in 2D and triangles in 3D) dividing interior from exterior.

The deformation of the interface is performed by moving the interface points. This means that the DSC method is explicit in nature and it preserves the advantages of Lagrangian methods, despite the described analogy with level sets. Still, the approach shares one of the biggest advantage of the Eulerian methods – topological adaptivity. Whenever the interface moves, the triangulation (tetrahedralization) is updated accordingly to accommodate the change. If, for example, the two interfaces components collide this change will cause them to merge.

The key to this topological adaptivity lies in a series of mesh operations performed with each deformation of the interface. Loosely described, the procedure in the 2D case is as follows. The interface points are not moved to their destination in a single step. Instead, each points is moved in the direction of the destination as far as possible *without* inverting triangles. When all points have moved, a mesh improvement routine is applied. This includes removing degenerate triangles, improving the mesh quality (via Laplacian smoothing and edge flips) and inserting points. The interface points are now

moved further towards their destination. This is repeated until all points have reached their destination. Mesh improvement routine relies on some chosen threshold values, defining e.g. when a triangle is degenerate. Those thresholds are a *part of* DSC and a single parameter required from a user is an average edge length of the triangles. It should be noted that DSC is customizable. For example, topology changes may be prohibited and the resolution or the required mesh quality may be changed or adaptive.

To summarize, the DSC method represents interface explicitly and yet it handles topology changes. Change of the interface are obtained by defining a displacement of vertices, possibly supplemented by actively changing labels of some triangles. Another advantage of the DSC method is that the discretization of the space may be exploited. For example, the area or perimeter of the inside region can easily be calculated. And lastly, DSC comes with a natural multiphase support. One can use an arbitrary number of triangle labels rather than just *inside* and *outside*.

B. Two-Phase Segmentation

Our two-phase segmentation approach evolves a curve with external and internal forces. External forces are region based and are directly adopted from (4), but this time explicitly formulated as forces acting on the curve points $\mathbf{X}$ as

$$\mathbf{F}_{\text{ext}}(\mathbf{X}) = (m_{\text{out}} - m_{\text{in}})(I - m_{\text{out}} + I - m_{\text{in}})\mathbf{N}, \quad (5)$$

where $\mathbf{N}$ is an outwards pointing normal.

Since we have an explicit curve representation, we use internal forces as defined in (2). It should be noted however that those regulatory terms have a different role here than in classical snakes, where elasticity is used to hold the curve points together at the expense of shrinking the curve. Our DSC curve representation automatically handles reparametrization and maintains a stable length of line segments. This fully removes the need for the first regulator term, and fairing of the curve can be performed using rigidity alone.

These forces will suffice if we e.g. want to segment an contiguous region from an initial seed point. If we want to segment a phase containing discontinuous regions or we want to introduce holes, we need to supplement curve evolution with the possibility of a triangle changing a label. We refer to this as *label flip*. In our initial two-phase model this happens when a triangle labeled as *inside* has a mean pixel intensity m_t such that $|m_t - m_{\text{out}}| < |m_t - m_{\text{in}}|$, or vice versa.

When label flips are allowed we initialize by bisecting the dynamic range of the image and labeling the triangles according to their mean intensity. When testing the curve evolution without the label flips we initialize with a spherical region marked *inside*.

Note that the singular *crossing* points might occur even when using only two phases, consider fx. checkerboard pattern. In our present two-phase implementation we do not apply any forces to crossing points, and those are resolved automatically through displacement of the neighbouring points on the interface.

C. Multiphase Segmentation

In multiphase segmentation, triangles are labeled as belonging to a phase $i \in \{1, 2, \dots, N\}$. Interfaces are now line segments dividing two phases and *triple junction* points generally occur. We still need to define only two actions in order to perform segmentation: the forces on the interface points and the label flip procedure.

Let us first consider interface points $\mathbf{X}_{ij}$ shared by just two phases i and j . Region forces (5) are generalized by using m_i and m_j , which are current mean intensities of the phases i and j . When considering interface points between any two phases (any combination of i and j), the external forces will be weaker between phases with mean intensities closer to each other than for two phases with intensity means closer together. As an example, consider the case illustrated in Fig. 1 *left* where the pixel intensities range from 0 to 255. Expanding region forces applied to the curve in order to include pixels with the intensity m_3 into a phase 3 have a magnitude which is a few times larger between phases 1 and 3 (denoted f_{13} in the illustration) than between phases 2 and 3 (denoted f_{23}). The fact that the force magnitudes differ substantially depending on the interface may cause convergence problems, and to avoid these we normalize the region forces. We require the resulting force to take a value of 1 or -1 when the image intensity I exactly equals one of the two means in the expression for the force. A normalized formulation of the region forces is

$$\mathbf{F}_{\text{ext}}(\mathbf{X}_{ij}) = \frac{I - m_i + I - m_j}{m_i - m_j} \mathbf{N}_{ij}. \quad (6)$$

Here $\mathbf{N}_{ij}$ is a curve normal pointing from the phase i towards the phase j , and with $\mathbf{X}_{ij}$ we indicate that the force is applied only to interface points shared by phases i and j . See on Fig. 1 *middle* how this affects forces in our example, and note that both a force magnitude f_{13} and f_{23} take a value 1 in m_3 .

In our current implementation we apply the external forces only to some crossing points, namely those crossing points which are shared by an set of regions which all belong to an unique phase. In particular, we apply a force for triple junctions and do not apply a force to crossings which occur in two-phase case. The reasoning behind this is as follows. If only distinct phases meets at crossing point, the resulting force on the point is resolved by considering pixel intensity I . It should be a phase with a mean closest to I that expands most. This can be formulated as a weighed combination the normal contributions of each phase. On the other hand, if there is phase represented twice around a crossing point the normals of those two regions will conflict most probably point in opposite direction. Normal movement in this case should not only only a position of the crossing point but should result in a phase merge in case of expansion, and a split when shrinking the phase. For now we refrained from include those active topology changes in our model.

As described above, for the first type of crossings which include triple junctions, the force on a point is defined as a combination of the contributions from each phase. We decided to retain only the contributions of the two phases which have the mean intensity directly below and directly above the image intensity I and to weight the two contributions using a linear function of I . To exemplify the approach let us consider a

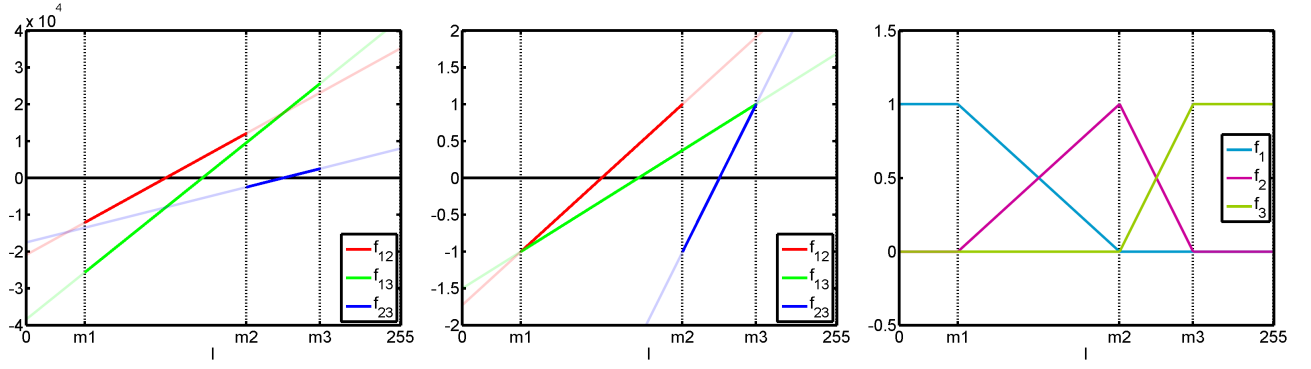


Fig. 1. An illustration showing magnitudes of forces used in three-phase segmentation, $\mathbf{F}_{\text{ext}}(\mathbf{X}_{ij}) = f_{ij}\mathbf{N}_{ij}$. On the *left* the magnitudes obtained by directly generalizing (5) to three labels, in *middle* the magnitudes after applying our proposed normalization. On the *right* a force applied to a triple junction expanded to contributions of each phase $\mathbf{F}_{\text{ext}}(\mathbf{X}_{123}) = f_1\mathbf{N}_1 + f_2\mathbf{N}_2 + f_3\mathbf{N}_3$.

triple junction point between regions 1, 2 and 3, and the pixel intensity which falls I between m_1 and m_2 . The resulting force would be

$$\mathbf{F}_{\text{ext}}(\mathbf{X}_{123}) = \frac{m_2 - I}{m_2 - m_1}\mathbf{N}_1 + \frac{I - m_1}{m_2 - m_1}\mathbf{N}_2, \quad (7)$$

where $\mathbf{N}_i$ is a curve normal pointing away from the region i . On Fig. 1 *right* you can see the weights of the three contributions for any pixel intensity I . Note that in case of $I = m_i$ the triple junction point moves merely due to the expansion of the region i .

The internal need also be generalized to include crossings. This has been done by tracing the boundaries of each phase and, in case of the crossing point, averaging the phase contributions. As a result, the rigidity term will try to move the crossing point towards the configuration where the angles of all the regions around the point are equal.

Finally, we also need to consider a label flip. In a multiphase case we need a more robust similarity measure than mean triangle intensity which might place a triangle containing e.g. half of pixels with intensity m_1 and half of pixels of intensity m_3 into a phase 2 provided that means are sorted and somewhat equidistant. Instead, a following scheme has been implemented. Each pixel of a triangle is assigned to a phase based on the distances $|I - m_i|$. The label of the dominant phase is then assigned to a whole triangle.

IV. RESULTS

Fig. 2 shows an outcome of our two-phase segmentation which matches the results obtained by other implementations of region based forces. Arguably, the initialization places a curve close to the final solution, but notice automatic topology changes underway and the correctly resolved crossing points. To demonstrate the robustness to initialization and to illustrate how the curve evolves subject to region forces we show in Fig. 3 a segmentation without label flips. Notice the fully automatic change in topology where the two branches merge.

An example of four-phase segmentation is shown in Fig. 4. Topology changes and triple junctions are handled fully automatically. In Fig. 5 we show the effect of changing the average triangle edge length, which is the only parameter relating to DSC in our current implementation. We can see that the detail

on a different *scale* has been extracted from the image. A thin gray band appears where the dark and the light intensities blend. Notice also a small boundary artifact present in our current implementation – a triangle edge may not be shorter than a given threshold value, so an object may not come arbitrarily close to the boundary of the image. Finally, in Fig. 6 we show a three-phase segmentation without label flips. Note again how topology automatically changes when two regions meet.

V. CONCLUSION

Our results confirm that the DSC method can be an interesting addition to an existing toolbox of deformable models used for image processing. We demonstrate how the two important qualities of the DSC method, adaptive topology and multiphase support, can be exploited in an image segmentation application. We believe that the results of our initial investigation are promising, and we are determined to use the DSC method in other segmentation applications.

Firstly, we plan to formulate our current approach as a Mumfor Shah [17] model for piecewise constant and piecewise smooth approximation. We also plan segmenting images into an *unknown* number of phases, given some quality threshold for the resulting segmentation/approximation.

We are already working on a texture based image segmentation framework which will utilize the DSC method for the curve representation. And finally, we wish to develop a volumetric segmentation based on the DSC method. In that regard the results shown here are important, since all region based forces easily generalize to 3D.

REFERENCES

- [1] C. Xu, A. Yezzi Jr, and J. L. Prince, “On the relationship between parametric and geometric active contours,” in *Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar Conference on*, vol. 1. IEEE, 2000, pp. 483–489.
- [2] M. K. Misztal and J. A. Bærentzen, “Topology-adaptive interface tracking using the deformable simplicial complex,” *ACM Transactions on Graphics (TOG)*, vol. 31, no. 3, p. 24, 2012.
- [3] M. K. Misztal, R. Bridson, K. Erleben, J. A. Bærentzen, and F. Anton, “Optimization-based fluid simulation on unstructured meshes,” in *VRIPHYS*, 2010, pp. 11–20.
- [4] M. K. Misztal, K. Erleben, A. Bargteil, J. Fursund, B. Christensen, J. A. Bærentzen, and R. Bridson, “Multiphase flow of immiscible

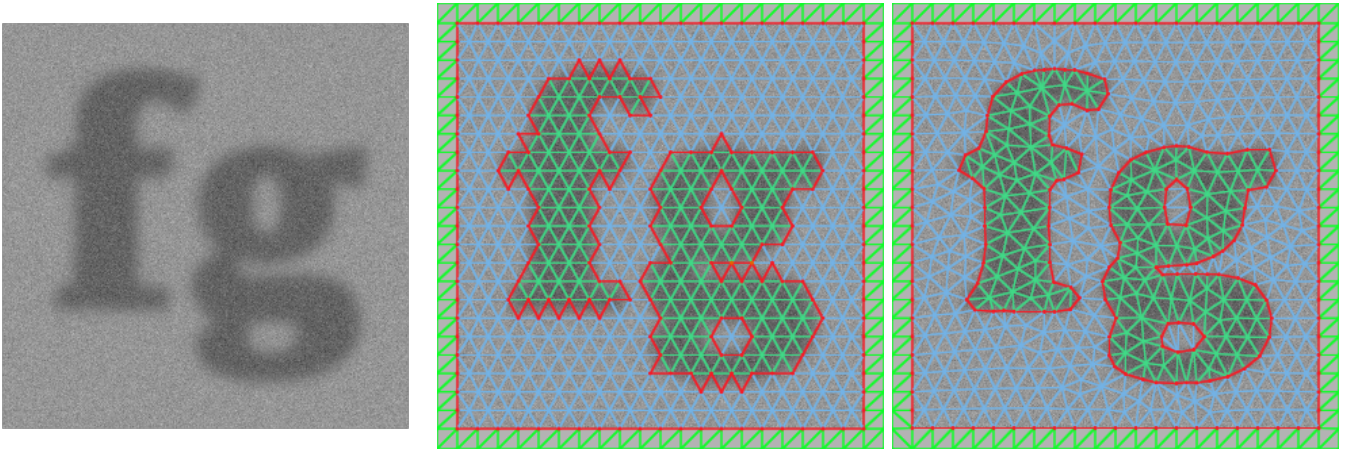


Fig. 2. Two-phase segmentation. On the *left* an original image featuring Gaussian noise and smoothed edges. In the *middle* the configuration after the initial face labeling but before evolving the curve. On the *right* a configuration after 25 iterations.

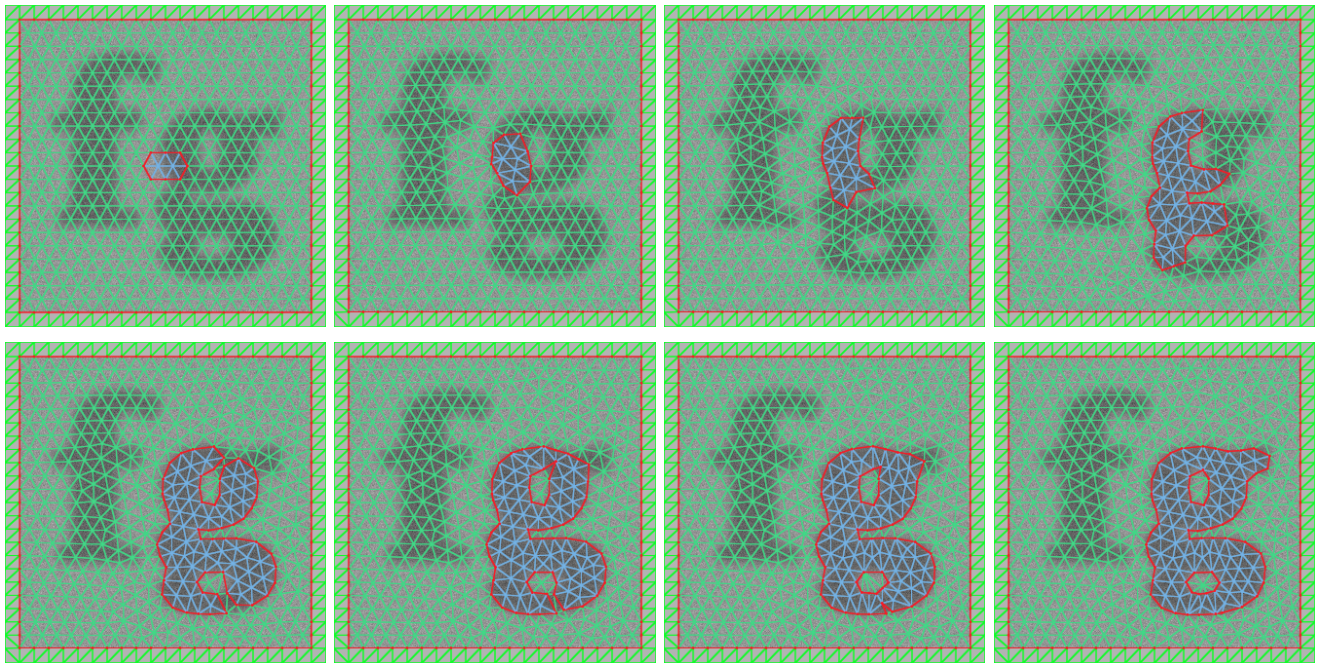


Fig. 3. Segmenting only a single contiguous region. In the *upper left* corner an initial configuration with a few faces assigned an *inside* label, followed with iterations 3, 9, 28, 43, 45, 47 and 82 of curve evolution.

- fluids on unstructured moving meshes,” in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 2012, pp. 97–106.
- [5] A. N. Christiansen, M. Nobel-Jørgensen, N. Aage, O. Sigmund, and J. A. Bærentzen, “Topology optimization using an explicit interface representation,” *Structural and Multidisciplinary Optimization*, pp. 1–13, 2013.
 - [6] M. K. Misztal, J. A. Bærentzen, F. Anton, and S. Markvorsen, “Cut locus construction using deformable simplicial complexes,” in *Voronoi Diagrams in Science and Engineering (ISVD), 2011 Eighth International Symposium on*. IEEE, 2011, pp. 134–141.
 - [7] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.
 - [8] C. Xu, D. L. Pham, and J. L. Prince, “Image segmentation using deformable models,” in *Handbook of Medical Imaging, Volume 2: Medical image processing and analysis*, M. Sonka, J. M. Fitzpatrick, and B. R. Masters, Eds., 2002.
 - [9] L. D. Cohen, “On active contour models and balloons,” *CVGIP: Image understanding*, vol. 53, no. 2, pp. 211–218, 1991.
 - [10] T. McInerney and D. Terzopoulos, “T-snakes: Topology adaptive snakes,” *Medical image analysis*, vol. 4, no. 2, pp. 73–91, 2000.
 - [11] V. Caselles, F. Catté, T. Coll, and F. Dibos, “A geometric model for active contours in image processing,” *Numerische mathematik*, vol. 66, no. 1, pp. 1–31, 1993.
 - [12] R. Malladi, J. A. Sethian, and B. C. Vemuri, “Shape modeling with front propagation: A level set approach,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 2, pp. 158–175, 1995.
 - [13] S. Osher and J. A. Sethian, “Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations,” *Journal of computational physics*, vol. 79, no. 1, pp. 12–49, 1988.
 - [14] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” *International journal of computer vision*, vol. 22, no. 1, pp. 61–79, 1997.
 - [15] A. Yezzi Jr, A. Tsai, and A. Willsky, “A statistical approach to snakes for bimodal and trimodal imagery,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 898–903.
 - [16] T. F. Chan and L. A. Vese, “Active contours without edges,” *IEEE*

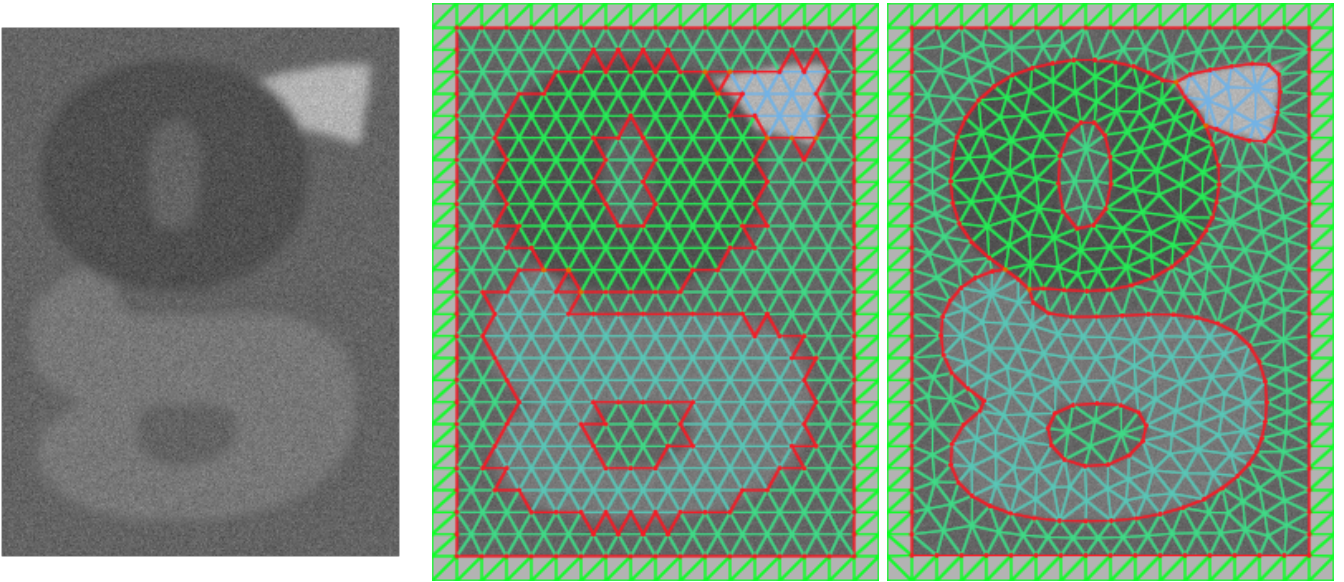


Fig. 4. Segmenting four phases. On the *left* an original image featuring Gaussian noise and smoothed edges. In the *middle* the configuration after the initial face labeling but before evolving the curve. On the *right* a configuration after 30 iterations.

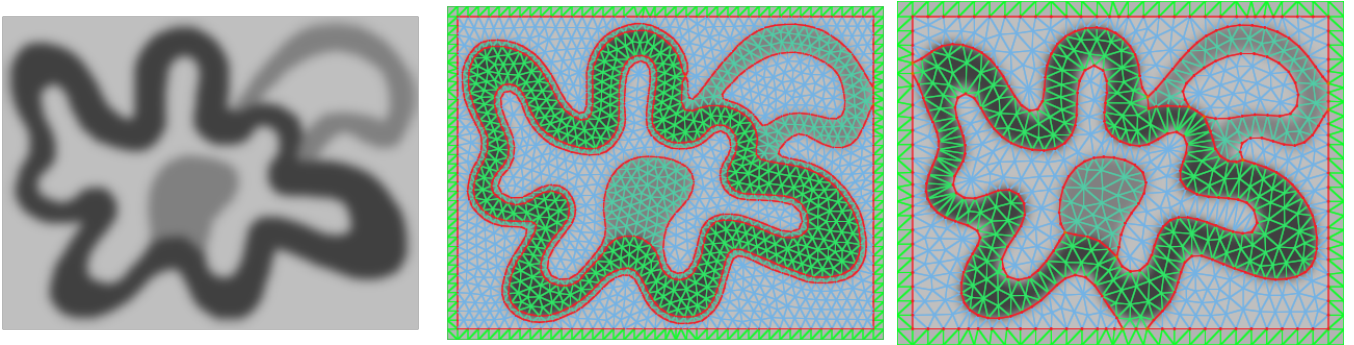


Fig. 5. The effect of the average edge length parameter. On the *left* an original image featuring Gaussian noise and smoothed edges. In the *middle* a segmentation with the average edge length set to 10. For the segmentation on the *right* the average edge length was set to 15.

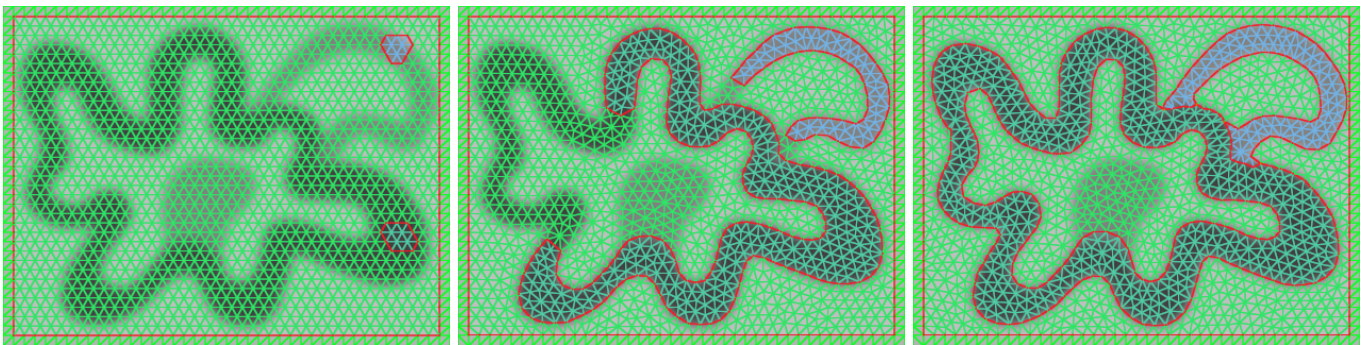


Fig. 6. Segmenting two contiguous regions in a three-phase case. On the *left* a user provided initialization. In the *middle* and on the *right* iterations 110 and 170 of the curve evolution.

- Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [17] D. Mumford and J. Shah, “Optimal approximations by piecewise smooth functions and associated variational problems,” *Communications on pure and applied mathematics*, vol. 42, no. 5, pp. 577–685, 1989.
- [18] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, “A variational level set approach to multiphase motion,” *Journal of computational physics*, vol. 127, no. 1, pp. 179–195, 1996.
- [19] A. Tsai, A. Yezzi Jr, and A. S. Willsky, “Curve evolution implementation

- of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification,” *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1169–1186, 2001.
- [20] L. A. Vese and T. F. Chan, “A multiphase level set framework for image segmentation using the mumford and shah model,” *International journal of computer vision*, vol. 50, no. 3, pp. 271–293, 2002.