

# A flexible and efficient numerical framework for image segmentation by energy minimization

Günay Doğan<sup>1,2</sup>, Pedro Morin<sup>3</sup>, Ricardo H. Nochetto<sup>4</sup>

<sup>1</sup> Theiss Research, San Diego, CA, USA, [gunay.dogan@nist.gov](mailto:gunay.dogan@nist.gov)

<sup>2</sup> Mathematical Sciences Division, NIST, Gaithersburg, MD, 20899, USA

<sup>3</sup> Univ. of Maryland, College Park, MD, 20754, USA, [rhn@math.umd.edu](mailto:rhn@math.umd.edu)

<sup>4</sup> Univ. Nacional del Litoral, Sante Fe, Argentina, [pmorin@santafe-conicet.gov.ar](mailto:pmorin@santafe-conicet.gov.ar)

**Abstract** A widely used approach to image segmentation is to define corresponding segmentation energies and to compute shapes that are minimizers of these energies. In this work, we introduce a flexible and efficient numerical framework for minimization of such energies. The framework enables use of various gradient descent flows, including  $H^1$  flows that are fast and stable. For this, we model the geometry explicitly and make use of shape differential calculus. We discretize the resulting partial differential equations using finite elements and obtain linear systems that can be solved efficiently. Incorporating spatial adaptivity, time step controls, topological changes results in a robust practical method.

**Keywords** Image segmentation, gradient descent flows, finite elements, adaptivity.

## 1 Introduction

Image segmentation is the problem of finding distinct regions that are uniform with respect to certain image features in given images. It is a fundamental problem in image processing and a large body of work addressing this problem exists. After the seminal work of Kass et al [10], variational approaches to image segmentation have gained popularity and are now widely used.

Typically the key to a variational image segmentation algorithm is the definition of a suitable segmentation energy over a set of shape candidates, such as curves in 2d or surfaces in 3d. Then an optimization method, such as gradient descent, can be used to compute the minimum of the segmentation energy. In this approach, one starts with a non-optimal initial shape and deforms it in a way that decreases its energy. Ideally the optimal shape obtained at the end of this procedure should coincide with the boundaries of the regions sought in the image.

There are a few critical components to an effective implementation of a variational segmentation method. One is the numerical representation of the shape and ensuring the accuracy and efficiency of the representation. Options are Lagrangian representations (parametric methods, triangulations) and Eulerian representations (level sets); each have their own weaknesses and strengths. Another critical component is the computation of energy-decreasing updates or descent direction at each iteration of shape evolution. In this work, we address both aspects of variational image segmentation and introduce an efficient and flexible numerical framework that enables us to compute a large class of advantageous descent directions. It has the following key features:

- *Parameter-free Lagrangian representation:* The shape representation is Lagrangian, namely curves are represented as polygons, surfaces as triangulations. The shapes are not parametrized; only a list of their simplices is needed for computation.
- *Choice of good descent directions:* The  $L^2$  gradient descent commonly used in the minimization procedure is typically slow and not very stable. We provide the ability to compute various  $H^1$  gradient descent flows, which are more stable and result in smoother flows. Moreover, using second order derivatives, we can speed up the convergence of the iterations.

- *Spatial adaptivity*: We maintain a coarse representation of the shape, where resolution is not needed, and a fine representation with higher node density, where the shape and the image vary more. This is very beneficial for both accuracy and efficiency.
- *Discretization with finite elements*: The geometric relationships and the resulting partial differential equations (PDE) are discretized using the finite element method. This results in well-defined linear systems, which can be solved efficiently. The discretization is intrinsic, it does not depend on a parametrization of the shape.

We also incorporate appropriate time step selection procedures and topological changes in 2d.

## 2 Segmentation Energies and Shape Calculus

The shape energies used in image processing typically have the following form

$$J(\Gamma) = \int_{\Gamma} g(x, \Gamma) dS + \int_{\Omega_1} f_1(x, \Omega_1) dx + \int_{\Omega_2} f_2(x, \Omega_2) dx, \quad (1)$$

where  $\Gamma$  denotes a surface in  $\mathbb{R}^d$ ,  $\Omega_1$  is the domain enclosed by  $\Gamma$  and  $\Omega_2$  is the domain outside  $\Gamma$ . The functions  $g(x, \Gamma)$ ,  $f_1(x, \Omega_1)$ ,  $f_2(x, \Omega_2)$  are image-dependent weight functions and are crucial in defining a successful segmentation energy. They may depend on the surface  $\Gamma$  and/or the domains  $\Omega_1, \Omega_2$ . For example, we may have  $g = g(x, \nu)$ , where  $\nu$  is the outer unit normal of  $\Gamma$ , or we may have  $f_1 = f_1(x, c_{\Omega_1})$ , where  $c_{\Omega_1} = \frac{1}{|\Omega_1|} \int_{\Omega_1} I(x) dx$  is average of image intensity  $I(x)$  in  $\Omega_1$ .

Starting with an initial surface  $\Gamma$ , we would like to compute a gradient descent velocity  $\vec{V}$  to deform  $\Gamma$  in a way that decreases its energy (1). For this, we need to quantify the effect of a candidate velocity  $\vec{V}$  on the energy (1). A given velocity  $\vec{V}$  would evolve the surface  $\Gamma$  through a set of ordinary differential equations:  $\Gamma_t = \{x(t, x_0) : \frac{dx}{dt} = \vec{V}(x(t)), x(0) = x_0, x_0 \in \Gamma\}$ .

Using this, we define the first shape derivative of  $J(\Gamma)$  at  $\Gamma$  with respect to given velocity  $\vec{V}$  [4]:  $dJ(\Gamma; \vec{V}) = \lim_{t \downarrow 0} \frac{1}{t} (J(\Gamma_t) - J(\Gamma))$ . Similarly we can define the second shape derivative:  $d^2J(\Gamma; \vec{V}, \vec{W}) = d(dJ(\Gamma; \vec{V}))(\Gamma; \vec{W})$ . The second shape derivative provides second order variation information and can be used to implement faster Newton-type minimization algorithms.

Now we introduce two examples of shape energies used for image segmentation. The first is the geodesic active contour model [1] and is given by weighted surface integral  $J_{GAC}(\Gamma)$ . The second example is the Mumford-Shah functional  $J_{MS}(\Gamma)$  [2], which incorporates domain information and can be used to segment noisy images into piecewise smooth regions.

$$J_{GAC}(\Gamma) = \int_{\Gamma} g(x) dS, \quad J_{MS}(\Gamma) = \sum_{i=1}^2 \frac{1}{2} \left( \int_{\Omega_i} (u_i - I)^2 + \mu |\nabla u_i|^2 \right) dx + \gamma \int_{\Gamma} dS.$$

In  $J_{GAC}(\Gamma)$ ,  $g(x) = 1/(1 + \frac{|\nabla I(x)|^2}{\lambda^2})$  is edge indicator function defined by image gradient  $\nabla I(x)$ . Functions  $u_i$  in  $J_{MS}(\Gamma)$  are obtained from  $-\mu \Delta u_i + u_i = I$  in  $\Omega_i$ ,  $\frac{\partial u_i}{\partial \nu_i} = 0$  on  $\partial \Omega_i$ .

The first shape derivatives of  $J_{GAC}, J_{MS}$  are given by [8, 9]

$$dJ_{GAC}(\Gamma; \vec{V}) = \int_{\Gamma} \left( g\kappa + \frac{\partial g}{\partial \nu} \right) V dS, \quad dJ_{MS}(\Gamma; \vec{V}) = \int_{\Gamma} \left( \frac{1}{2} [|u - I|^2] + \frac{\mu}{2} [|\nabla u|^2] + \gamma \kappa \right) V dS,$$

where  $\nu$  is the outer unit normal to  $\Gamma$ ,  $\kappa$  is the mean curvature of  $\Gamma$ ,  $[f] = f_1 - f_2$  denotes the jump of  $f$  across  $\Gamma$  and  $V = \vec{V} \cdot \nu$  is the normal component of  $\vec{V}$ . We can see that the first shape derivatives have the following form:  $dJ(\Gamma; \vec{V}) = \int_{\Gamma} (w_1(x, \Gamma)\kappa + w_2(x, \Gamma)) dS$ .

The second shape derivatives of  $J_{GAC}, J_{MS}$  have the form

$$d^2J(\Gamma; \vec{V}, \vec{W}) = \int_{\Gamma} \left( \alpha(x, \Gamma) \nabla V \cdot \nabla W + \beta(x, \Gamma) VW \right) dS + \int_{\Gamma} R(x, \Gamma) dS, \quad (2)$$

where  $\alpha(x, \Gamma), \beta(x, \Gamma)$  are weight functions,  $R(x, \gamma)$  contains some remaining terms and  $V = \vec{V} \cdot \nu, W = \vec{W} \cdot \nu$ . The specific expressions for  $J_{GAC}, J_{MS}$  can be found in [8], [9] respectively.

Since the shape derivatives depend only on the normal components  $V, W$  of  $\vec{V}, \vec{W}$ , we will work with the scalar velocity fields  $V, W$ , thereby write  $dJ(\Gamma; V), d^2J(\Gamma; V, W)$ .

### 3 Gradient Descent Flows

The shape derivatives introduced in the previous section allow us to evaluate the effect of given velocities on the shape energies. Our goal, however, is to compute a velocity that is a good descent direction for a given surface  $\Gamma$ . For this, we introduce a scalar product  $b(\cdot, \cdot)$  with the associated Hilbert space  $H(\Gamma)$  on  $\Gamma$ . Then we can solve the following equation for  $V$

$$b(V, \phi) = -dJ(\Gamma; \phi), \quad \forall \phi \in H(\Gamma). \quad (3)$$

It is easy to see that the velocity computed this way makes the shape derivative negative, thus decreases the energy:  $dJ(\Gamma; V) = -b(V, V) \leq 0$ . Possible choices for the scalar product  $b(\cdot, \cdot)$  are the  $L^2$  scalar product  $\langle V, W \rangle_{L^2} = \langle V, W \rangle = \int_{\Gamma} VW dS$  or the weighted  $H^1$  scalar product  $\langle V, W \rangle_{H^1} = \langle \alpha \nabla V, \nabla W \rangle + \langle \beta VW \rangle$ , where  $\alpha = \alpha(x, \Gamma), \beta = \beta(x, \Gamma)$  are positive weight functions. The gradient descent flow resulting from the  $L^2$  scalar product is known to exhibit slow convergence and may not be stable. The  $H^1$  scalar product on the other hand results in smooth evolutions and is very stable. Constant coefficient versions have been implemented and examined in [3, 11]. Moreover, if the second shape derivatives are used as the basis of the  $H^1$  scalar products, one can achieve faster convergence with fewer iterations [8, 9, 5].

In order to compute a gradient descent velocity, we need to put together the information from geometry, the shape derivative and the gradient descent equation (3), namely, three more equations of basic differential geometry:  $\vec{\kappa} = -\Delta_{\Gamma} \vec{X}, \quad \kappa = \vec{\kappa} \cdot \nu, \quad \vec{V} = V\nu$ . These can be imposed weakly by multiplying with test functions  $\phi \in H(\Gamma)$  and  $\vec{\phi} \in [H(\Gamma)]^d$ , and integrating by parts:  $\langle \vec{\kappa}, \vec{\phi} \rangle = \langle \nabla_{\Gamma} \vec{X}, \nabla_{\Gamma} \vec{\phi} \rangle, \quad \langle \kappa, \phi \rangle = \langle \vec{\kappa} \cdot \nu, \phi \rangle \quad \langle \vec{V}, \vec{\phi} \rangle = \langle V\nu, \vec{\phi} \rangle$ . Now we can expand the quantities  $\vec{X}, \kappa, \vec{\kappa}, V, \vec{V}$  in terms of finite element basis functions  $\{\phi_i\}_{i=1}^N, \{\vec{\phi}_i\}_{i=1}^N$  (piecewise linear in our implementation) and obtain the fully discretized scheme. The complete set of equations for the weak form and the corresponding linear system is given by

$$\begin{aligned} \langle \vec{\kappa}, \vec{\phi} \rangle &= \langle \nabla_{\Gamma} \vec{X}, \nabla_{\Gamma} \vec{\phi} \rangle, & \langle \kappa, \phi \rangle &= \langle \vec{\kappa} \cdot \nu, \phi \rangle, & b(V, \phi) &= -dJ(\Gamma, \phi), & \langle \vec{V}, \vec{\phi} \rangle &= \langle V\nu, \vec{\phi} \rangle, \\ \vec{M}\vec{K} &= \vec{A}\vec{X}, & M\mathbf{K} &= \vec{N}^T \vec{K}, & B\mathbf{V} &= -M_{w_1} \mathbf{K} - \mathbf{w}_2, & \vec{M}\vec{V} &= \vec{N}\mathbf{V}. \end{aligned}$$

The vectors  $\vec{X}, \mathbf{K}, \vec{K}, \mathbf{V}, \vec{V}$  store the finite element coefficients for the quantities  $\vec{X}, \kappa, \vec{\kappa}, V, \vec{V}$  respectively. Details of the finite element discretization and the linear system can be found in [6], where a semi-implicit scheme was proposed. The current scheme is explicit and velocity vector computed by this can be used to update the surface  $\Gamma^n$  to obtain  $\Gamma^{n+1}$  by  $\vec{X}^{n+1} = \vec{X}^n + \tau \vec{V}$ .

### 4 Computation

In order to realize a flexible implementation that is effective and robust in practice, some additional computational procedures are needed [5, 6, 7]:

- *Spatial adaptivity*: Geometric adaptivity and data-based adaptivity, i.e. we maintain a fine mesh where the local geometry or the image varies more and a coarse mesh elsewhere.
- *Time step selection*: As the goal is energy minimization, the time steps are taken to ensure energy decrease. Moreover, an extra check is imposed to guard against mesh distortions.
- *Topological changes in 2d*: The number of regions to be segmented are not known in advance. This requires capabilities to merge or split shapes, currently implemented for 2d curves.

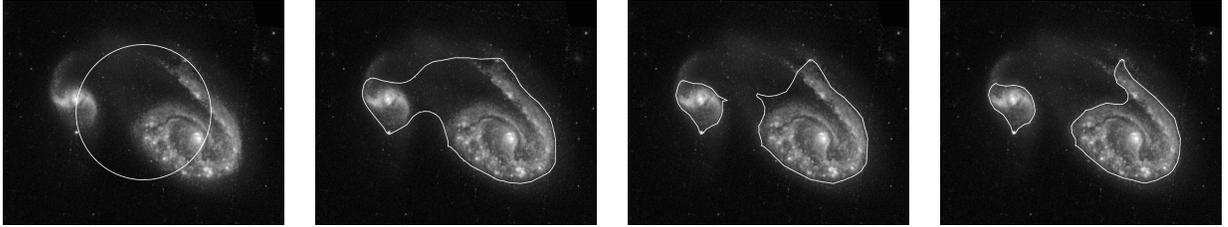


Figure 1: Segmentation of a galaxy image using the Mumford-Shah functional and  $H^1$  gradient descent flow based on the second shape derivative. Mumford-Shah functional is well-suited for images with no distinct edges, because it incorporates global information during segmentation. This example also illustrates topological changes in 2d. The initial curve splits into two during the evolution to capture the two galaxies in the image.

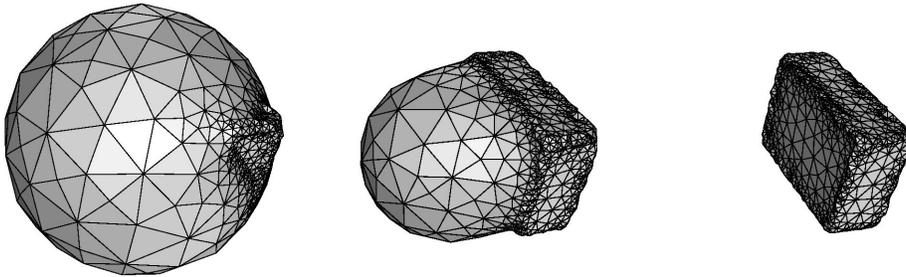


Figure 2: Segmentation of a synthetic 3d image using the geodesic active contour model and the  $H^1$  gradient descent flow based on the second shape derivative. This example also illustrates spatial adaptivity. We start with a coarse initial surface. As the surface starts capturing the object, it is refined to resolve the object better. The final model has fine resolution only where it is needed (edges, corners), and a coarse representation elsewhere (faces).

We demonstrate the framework with two examples. First we use the Mumford-Shah functional to segment a galaxy image with no edges (see Figure 1). The galaxy image is hard to segment for traditional methods because it does not have well-defined boundaries or edges. But it is a good example demonstrating the power of the Mumford-Shah functional, which incorporates global information. We use the  $H^1$  flow to segment image. We can see the role of topological changes in this example. We start with a single closed curve. It splits into two curves to capture both galaxies in the image.

In the second example, we use the geodesic active contour model on a synthetic 3d example again using the corresponding  $H^1$  flow (see Figure 2). This example illustrates the benefits of incorporating spatial adaptivity in the method. We start with a coarse initial mesh and use modest computational resources at the initial iterations. As the surface starts capturing the target object, it refines to resolve the object better. At the termination of the evolution, the surface has fine resolution only where needed (edges, corners) and maintains a coarse representation elsewhere. Performing the same experiment with a fine surface mesh from the start would require an order of magnitude more computation for the same result.

## 5 Conclusion

We have introduced a flexible and efficient numerical framework for image segmentation by energy minimization. Our work is aimed at the variational segmentation approaches in literature. In these approaches, initial shapes (curves, surfaces) specified by the users are deformed to minimize

associated segmentation energies, so that the optimal shapes computed correspond to the sought segmentation. The key feature of our framework is that it enables practitioners to define and compute a large class of descent updates, which yield stable evolutions and often converge with fewer iterations. This is in contrast with the  $L^2$  gradient descent traditionally used in this field. We also use spatial adaptivity to tune the resolution of the mesh with respect to the variation in the image and in the geometry. This significantly improves computational efficiency without compromising accuracy. Finally, incorporating time step controls and topological changes results in a robust practical method. We have demonstrated the effectiveness of our method with the geodesic active contour model and the Mumford-Shah functional.

## References

- [1] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *IJCV*, 22(1):61–79, 1997.
- [2] T. Chan and L. Vese. A level set algorithm for minimizing the Mumford-Shah functional in image processing. In *Proc. of 1st IEEE VLSM Workshop*, pages 161–168, 2001.
- [3] G. Charpiat, P. Maurel, J.-P. Pons, R. Keriven, and O. Faugeras. Generalized gradients: Priors on minimization flows. *IJCV*, 73(3):325–344, 2007.
- [4] M. C. Delfour and J.-P. Zolésio. *Shapes and Geometries*. SIAM, Philadelphia, PA, 2001.
- [5] G. Doğan, P. Morin, and R. H. Nochetto. A variational shape optimization approach for image segmentation with a Mumford-Shah functional. *SIAM J. Sci. Comput.*, 30(6), 2008.
- [6] G. Doğan, P. Morin, R. H. Nochetto, and M. Verani. Discrete gradient flows for shape optimization and applications. *Comput. Methods Appl. Mech. Engrg.*, 196(37-40), 2007.
- [7] G. Doğan. *A variational shape optimization framework for image segmentation*. PhD thesis, Department of Mathematics, University of Maryland, College Park, 2006.
- [8] M. Hintermüller and W. Ring. A second order shape optimization approach for image segmentation. *SIAM J. Appl. Math.*, 64(2):442–467, 2003/04.
- [9] M. Hintermüller and W. Ring. An inexact Newton-CG-type active contour approach for the minimization of the Mumford-Shah functional. *J. Math. Imag. Vision*, 20(1-2):19–42, 2004.
- [10] M. Kass, A. Witkin, and D. Terzopoulos. Active contour models. *IJCV*, 1(4):321–331, 1988.
- [11] G. Sundar, A. Yezzi, and A. C. Mennucci. Sobolev active contours. *IJCV*, 73:345–366, 2007.