

# Image Segmentation with a Sobolev Gradient Method

R. J. Renka

*Department of Computer Science & Engineering, University of North Texas,  
Denton, TX 76203-1366*

---

## Abstract

The most effective methods for finding object boundaries in a digital image involve minimizing a functional over a set of curves or surfaces, where the functional includes internal energy terms for regularization and external energy terms that align the curves or surfaces with object boundaries. Current practice is to seek critical points of the energy functional by what amounts to a steepest descent iteration with the discretized  $L_2$  gradient. Since the functional involves derivatives, a descent method with a discretized Sobolev gradient is likely to be much more efficient. We demonstrate this with test results for an implementation of a variational level set method for edge-based segmentation with active contours in two dimensions.

Keywords: Image segmentation; Sobolev gradient; Variational level set method; Active contours; Snakes; Gradient descent

---

## 1 Introduction

Given a digital image defined by an array of pixels (graylevel intensity values on a uniform rectangular grid), the segmentation problem is to find a set of boundary curves that partition the image into a set of regions such that either the pixels in each region are similar with respect to some property such as intensity or texture, or the boundary curves are aligned with edges, defined by abrupt changes of intensity. The method is referred to as region-based in the first case, or edge-based in the second. The problem is readily generalized to color images and to three-dimensional arrays of pixels associated with a sequence of parallel MRI slices. There are numerous applications such as delineating the boundary of a tumor in a brain scan or an object in a satellite image.

---

*Email address:* [renka@cs.unt.edu](mailto:renka@cs.unt.edu) (R. J. Renka).

Richardson [1] discussed the application of the Sobolev gradient method to a variety of partial differential equations (PDEs) that arise in image processing. His test results demonstrate substantial benefits in computational efficiency. Sundaramoorthi, Yezzi, and Mennucci [2] coined the term ‘Sobolev active contours’ for active contour models with a Sobolev gradient in a space of smooth curves. Rather than using a Sobolev inner product on the space of level set functions as we do here, they use a Sobolev inner product in the tangent space at each point on a manifold whose elements are smooth curves. This complicates the numerical methods because, at each descent step, rather than simply updating the level set function with a multiple of the negative Sobolev gradient vector, the set of parametric (or discrete) curves must be extracted from the level set function, updated with the negative gradient vectors, and then used to update the level set function. Sundaramoorthi et al list several advantages of their method, such as regularity of flow and the favoring of global translations, which are particularly useful in tracking applications. It is difficult to quantify, but we conjecture that the same benefits accrue to our simpler method.

The segmentation problem is formulated as an energy minimization problem in Section 2, numerical solution methods are discussed in Section 3, and test results are presented in Section 4.

## 2 Variational problem

Before displaying the energy functional used in our experiments, we motivate its choice by discussing the important ideas that led to its development. PDE-based methods for treating image processing problems lend themselves naturally to finite differencing on a uniform rectangular grid with unit mesh widths. We thus represent the image by a function  $I : \Omega \rightarrow \mathbf{R}$ , where  $\Omega$  is a rectangular subset of  $\mathbf{R}^2$ . The first application of active contours or snakes [3] consisted of finding a parametric curve  $C : [0, 1] \rightarrow \Omega$  that minimizes

$$E_1(C) = \frac{\alpha}{2} \int_0^1 |C'(t)|^2 dt + \frac{\beta}{2} \int_0^1 |C''(t)|^2 dt - \lambda \int_0^1 |\nabla I(C(t))| dt \quad (1)$$

for positive constants  $\alpha, \beta$ , and  $\lambda$ . The first two terms constitute internal energy — regularizing terms that enforce smoothness. The first term, referred to as elasticity, is associated with total curve length, while the second term, rigidity, corresponds to bending energy. The last term, the external energy, is a potential energy term that aligns the curve with edges where  $I$  has steep gradient.

A more general edge detector is defined by a positive decreasing function  $g$  such that  $g(z) \rightarrow 0$  as  $z \rightarrow \infty$ ; for example,  $g(z) = \frac{1}{1+z^2}$ . Then  $g(|\nabla I(x, y)|)$  is small near edges, and bounded above by 1 where  $I$  is constant. In practice we use

$$g(|\nabla G_\sigma * I|) = \frac{1}{1 + |\nabla(G_\sigma * I)|^2},$$

where  $G_\sigma * I$  denotes convolution of  $I$  with a Gaussian smoothing kernel  $G_\sigma(x, y) = \frac{1}{\sqrt{\sigma}} e^{-(x^2+y^2)/4\sigma}$ . The standard deviation is chosen to remove noise without smearing out edges too much — typically one or two pixels;  $\sigma = 1.5$ , for example.

An undesirable property of the functional in (1) is its dependence on the parameterization of the curve. A change of parameter can produce an arbitrary change in the energy. This led to the development of geometric active contours using parameterization by arc length which is intrinsic to the curve. One such method was introduced in [4]:

$$E_2(C) = \int_0^1 g(|\nabla I(C(t))|) |C'(t)| dt.$$

Since this functional represents total curve length in a Riemannian metric induced by  $I$ , its minimizer is referred to as a geodesic active contour.

In order to treat the case of more than one object and/or multiply-connected objects, it is necessary to allow for more than one curve. This implies that the number of curves must be allowed to change as the energy functional is decreased. With a parametric or discrete (particle) curve representation, a procedure for splitting and merging curves would be inconvenient and unwieldy, but with an implicit curve representation, the changes in topology are automatic and natural. To this end, define  $C$  as the zero level set of a Lipschitz continuous function  $u : \Omega \rightarrow \mathbf{R}$ , and define  $\omega \subset \Omega$ , not necessarily connected, as the set of points on which  $u$  is negative, so that

$$\begin{aligned} u(\mathbf{x}) &< 0 \text{ for } \mathbf{x} \in \omega, \\ u(\mathbf{x}) &= 0 \text{ for } \mathbf{x} \in C, \\ u(\mathbf{x}) &> 0 \text{ for } \mathbf{x} \in \Omega \setminus \omega. \end{aligned}$$

Then  $C$  is the boundary of  $\omega$ , and the geometric quantities associated with  $C$  are easily computed as follows, where we use the Heaviside function defined

by

$$H(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$

and its distributional derivative  $\delta$ .

- Area of  $\omega$ :  $|\omega| = \int_{\Omega} H(-u(\mathbf{x}))$ .
- Curve length:  $|C| = \int_{\Omega} |\nabla H(u(\mathbf{x}))| = \int_{\Omega} \delta(u(\mathbf{x})) |\nabla u(\mathbf{x})|$ .
- Outward unit normal to  $C$ :  $n = \frac{\nabla u}{|\nabla u|}$ .
- Curvature of level set at  $\mathbf{x}$ :  $\kappa = \nabla \cdot \frac{\nabla u}{|\nabla u|}$ .

The level set method [5] consists of parameterizing  $u(\mathbf{x})$  by time  $t \in \mathbf{R}^+$  to obtain a moving front (or active contour)  $C(t) = \{\mathbf{x} : u(\mathbf{x}, t) = 0\}$  associated with an evolving function  $u(\mathbf{x}, t)$ . Conversely, given a velocity  $\mathbf{x}'(t)$  at each point on a moving front  $C(t)$ , a function  $u(\mathbf{x}, t)$  with  $C(t)$  as zero level set at every  $t$  is a solution to the initial value problem obtained by differentiating  $u(\mathbf{x}(t), t) = 0$ :  $u_t + \langle \nabla u(\mathbf{x}(t), t), \mathbf{x}'(t) \rangle = 0$  for specified  $u(\mathbf{x}, 0)$ . Denoting the speed of evolution in the outward normal direction by  $F$ , we obtain the level set equation

$$u_t + F|\nabla(u)| = 0.$$

A gradient descent method for minimizing an energy functional  $E(u)$  has this form with  $F$  dependent on the image  $I$  as well as the level set function  $u$ .

In addition to automatically handling changes in topology, the level set method allows for corners and cusps which cannot be represented with a regular parameterization. Another advantage is that the formulation is the same in any number of space dimensions. A disadvantage of the level set formulation is the cost of updating the entire set of grid point function values, representing all of the level sets, when only the zero level set is of interest. The Narrow Band level set method [6] restricts attention to a narrow band of grid points centered on the zero level set, updating function values only at those grid points. At each time step during the evolution, the narrow band pixels are updated, thus moving the zero level set curves, and the array indices defining the new narrow band are computed. For an  $N$  by  $N$  grid, the zero level set has  $O(N)$  points. The disadvantage is a more complex algorithm and the need for frequent re-initialization.

The level set function is usually initialized to the signed distance function associated with an initial front. In order to maintain accurate and stable evolution it may be necessary to periodically re-initialize the level set function to

signed distance from the evolving front, typically by computing the steady-state solution to

$$u_t = \text{sign}(u)(1 - |\nabla u|).$$

Re-initialization is necessary if the speed function  $F$  does not preserve signed distance to the zero level set. In the case of the Narrow Band method, it is necessary whenever the front approaches the boundary of the band — every time step with a very narrow band. Also, many methods perturb the front, possibly causing it to pass through weak boundaries in the case of edge-based segmentation. A better method is to add a distance regularizing term to the energy functional [7]:

$$E_3(u) = \frac{1}{2} \int (|\nabla u| - 1)^2.$$

This eliminates the need for expensive ad-hoc re-initialization. Furthermore, it allows for a very narrow band, it enables stable flow with larger time steps, and it enables stable flow without upwind differencing.

A very effective edge-based segmentation method is based on the following energy functional [7,8]:

$$E(u) = \mu \int_{\Omega} \frac{1}{2} (|\nabla u| - 1)^2 + \lambda \int_{\Omega} g \delta(u) |\nabla u| + \nu \int_{\Omega} g H(-u), \quad (2)$$

where  $g = g(|\nabla(G_{\sigma} * I)|)$  is the edge indicator function, and  $\mu$ ,  $\lambda$ , and  $\nu$  are constants with  $\mu > 0$  and  $\lambda \geq 0$ . Increasing  $\lambda$  penalizes curve length except near edges. The last term, except near edges, is the weighted area interior to the zero level set curve (where  $u < 0$ ). The curve shrinks if  $\nu > 0$ , or expands if  $\nu < 0$ . We used this functional in our experiments with  $\mu = 0.04$ ,  $\lambda = 5.0$ , and  $\nu = 1.5$ . For computation we replace  $\delta(u)$  by a regularized Dirac function in  $C^1(\mathbf{R})$ :

$$\delta_{\epsilon}(u) = \begin{cases} \frac{1}{2\epsilon} \left[ 1 + \cos\left(\frac{\pi u}{\epsilon}\right) \right] & \text{if } |u| \leq \epsilon \\ 0 & \text{if } |u| > \epsilon \end{cases}$$

with  $\epsilon = 1.5$ .

### 3 Numerical methods for curve evolution

The problem is now to minimize or at least find critical points of the energy functional. The following observations apply to the more general problem of finding critical points of any functional  $E(u)$  for which the Euler-Lagrange equation is a nonlinear PDE. Consider the following three approaches.

- (1) Solve  $\nabla E(u) = 0$  directly by a Newton iteration:

$$u^{n+1} = u^n - H(u^n)^{-1} \nabla E(u^n)$$

for Hessian  $H$ . Alternatively, a quasi-Newton method may be more effective.

- (2) Evolve the parabolic PDE  $u_t = -\nabla E(u)$  to steady state with, for example, a first-order explicit method using constant time-step size:

$$u^{n+1} = u^n - \Delta t \nabla E(u^n),$$

where  $\Delta t$  is limited only by the CFL condition since accuracy of transients is not important.

- (3) Use a gradient descent method such as a steepest descent iteration:

$$u^{n+1} = u^n - \tau_n \nabla E(u^n),$$

where the step size  $\tau_n$  is either constant or is computed by a line search. Alternatively, a nonlinear conjugate gradient method, in which the search direction is a linear combination of the negative gradient direction and the search direction from the previous iteration, is usually faster.

The second option seems to be the method of choice, at least for the segmentation problem. The first option, Newton's method, is very fast when it works, but may not be feasible. The Hessian might be positive definite only in the immediate vicinity of a critical point, requiring a very accurate initial estimate, or it might be too expensive to evaluate or require too much storage. The third option appears to have been overlooked, perhaps because the method of steepest descent is reputedly slow. This reputation is well-deserved when the method is implemented with the ordinary gradient obtained by discretizing the  $L_2$  gradient. With the Sobolev gradient, however, steepest descent is often very effective [9].

Define a differential operator  $D : H^{1,2}(\Omega) \rightarrow L_2(\Omega)^3$  by

$$D(u) = \begin{pmatrix} u \\ \nabla u \end{pmatrix},$$

and let

$$\langle g, h \rangle_{H^{1,2}(\Omega)} = \int_{\Omega} gh + \langle \nabla g, \nabla h \rangle = \langle Dg, Dh \rangle_{L_2(\Omega)^3}.$$

From (2) the Fréchet derivative of  $E$  is

$$\begin{aligned} E'(u)h &= \lim_{\alpha \rightarrow 0} \frac{1}{\alpha} [E(u + \alpha h) - E(u)] \\ &= \int_{\Omega} \mu \left\langle \nabla u - \frac{\nabla u}{|\nabla u|}, \nabla h \right\rangle + \lambda \left\langle \frac{g\delta_{\epsilon}(u)\nabla(u)}{|\nabla u|}, \nabla h \right\rangle + \\ &\quad \lambda g\delta'_{\epsilon}(u)|\nabla u|h - \nu g\delta_{\epsilon}(u)h. \end{aligned} \tag{3}$$

For  $u \in H^{1,2}(\Omega)$ ,  $E'(u)$  is a bounded linear functional on  $H^{1,2}(\Omega)$  and hence, by the Riesz representation theorem, there exists  $\nabla_S E(u) \in H^{1,2}(\Omega)$ , termed the Sobolev gradient, such that

$$\begin{aligned} E'(u)h &= \langle \nabla_S E(u), h \rangle_{H^{1,2}(\Omega)} \\ &= \langle D\nabla_S E(u), Dh \rangle_{L_2(\Omega)^3} = \langle D^* D\nabla_S E(u), h \rangle_{L_2(\Omega)} \end{aligned}$$

for all  $h \in H^{1,2}(\Omega)$ , where  $D^*$  denotes the adjoint of  $D$ . For  $u \in H^{2,2}(\Omega)$ ,  $E'(u)$  is also bounded on  $L_2(\Omega)$  and is represented by an  $L_2$  gradient:

$$E'(u)h = \langle \nabla E(u), h \rangle_{L_2(\Omega)}$$

for all  $h \in H^{1,2}(\Omega)$ . The two gradients are thus related by

$$\nabla_S E(u) = (D^* D)^{-1} \nabla E(u), \tag{4}$$

where

$$D^* D = (I \ -\nabla) \begin{pmatrix} I \\ \nabla \end{pmatrix} = I - \Delta.$$

Since  $H^{2,2}(\Omega)$  is a dense subspace of  $H^{1,2}(\Omega)$ , equation (4) can be extended to  $u \in H^{1,2}(\Omega)$  by continuity. In order to obtain an expression for  $\nabla E(u)$ , we integrate by parts in (3):

$$E'(u)h = \int_{\Omega} \left[ -\mu \operatorname{div} \left( \nabla u - \frac{\nabla u}{|\nabla u|} \right) - \lambda \operatorname{div} \left( \frac{g\delta_{\epsilon}(u)\nabla u}{|\nabla u|} \right) \right] h$$

$$+\lambda g\delta'_\epsilon(u)|\nabla u| - \nu g\delta_\epsilon(u)] h,$$

with  $\nabla u = 0$  on the boundary of  $\Omega$ . Hence

$$\begin{aligned} \nabla E(u) &= -\mu \operatorname{div} \left( \nabla u - \frac{\nabla u}{|\nabla u|} \right) - \lambda \operatorname{div} \left( \frac{g\delta_\epsilon u \nabla u}{|\nabla u|} \right) + \\ &\quad \lambda g\delta'_\epsilon(u)|\nabla u| - \nu g\delta_\epsilon(u) \\ &= -\mu \left[ \Delta u - \operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right) \right] - \lambda \delta_\epsilon(u) \operatorname{div} \left( \frac{g\nabla u}{|\nabla u|} \right) - \nu g\delta_\epsilon(u). \end{aligned} \quad (5)$$

We discretize the problem by replacing integrals by sums and using second-order difference approximations to derivatives. At each descent iteration we compute a discretized  $L_2$  gradient from (5), compute a Sobolev gradient by solving Poisson's equation (4), and then update the level set function  $u$  with a scalar multiple of the negative Sobolev gradient. We have implemented both a Narrow Band level set method as described in [8] and a method that updates the level set function on the entire rectangular grid. In the case of the narrow band the linear systems are solved by a conjugate gradient method with diagonal preconditioner and Dirichlet boundary conditions outside the narrow-band grid points. In the case of the full grid we use a fast Poisson solver based on cosine transforms which enforce zero Neumann boundary conditions.

In general a line search is beneficial when evaluations of the energy functional are cheap relative to evaluations of its gradient. For the problem treated here, we found it slightly more efficient to use a constant step size.

## 4 Test results

All our computation was done in Matlab which is extremely well-suited to image processing applications. It is not only easy and convenient, but we suspect that, with careful attention to vectorization, the code is at least as efficient as that produced by the best compilers. Our experiments were run under Matlab 7.5 on a MacBook Air with 1.8 GHz processor.

We tested our method on four images, displayed in Figures 1 to 4, with the computed level set curves depicted in white in Figure 3, and black in the other three figures. The figure captions specify the image name, resolution, and number of descent iterations required for convergence with the Narrow Band level set method and Sobolev gradient, where convergence was defined by an upper bound on the size of the gradient.

In order to assess relative computational efficiency, Table 1 displays iteration

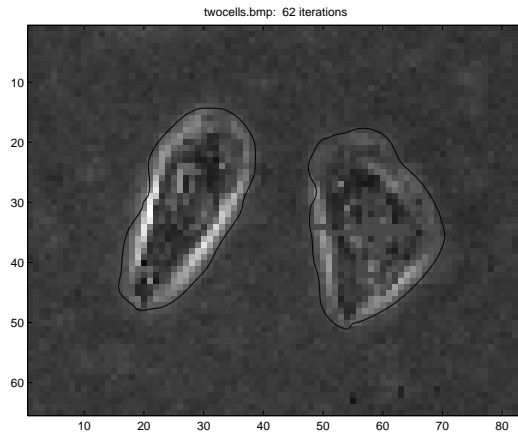


Fig. 1. TwoCells: 83 X 65, 62 iterations

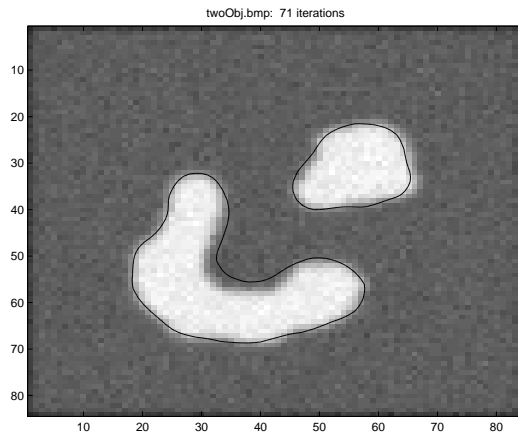


Fig. 2. TwoObjects: 84 X 84, 71 iterations

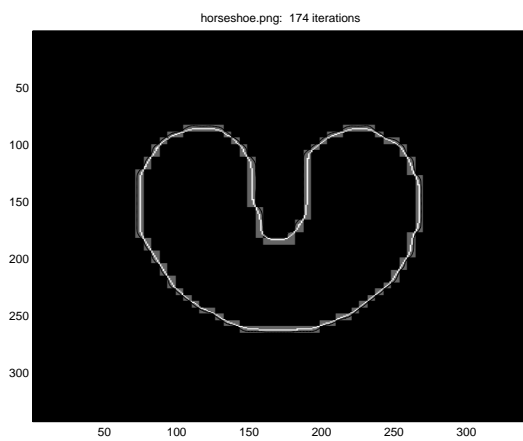


Fig. 3. Horseshoe: 342 X 342, 174 iterations

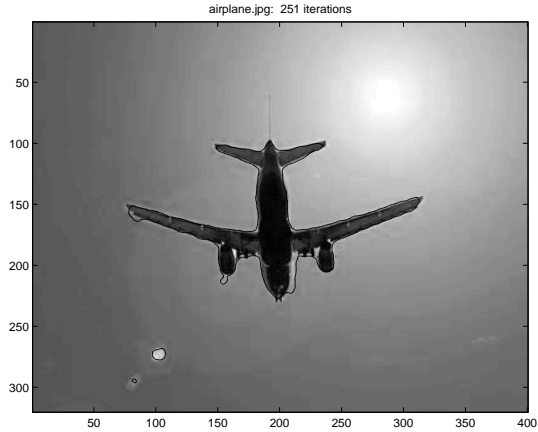


Fig. 4. Airplane: 400 X 320, 251 iterations

Image	Full grid		Narrow band	
	$L_2$ gradient	Sobolev gradient	$L_2$ gradient	Sobolev gradient
TwoCells	210	70	160	62
	8.7	5.6	5.9	4.2
TwoObjects	310	110	260	71
	12.4	7.1	7.9	4.7
Horseshoe	820	190	640	174
	288.1	109.2	54.4	37.3
Airplane	1020	240	810	251
	403.2	158.2	80.8	56.4

Table 1

Iteration counts and execution times in seconds.

counts and execution times for the four images and each of four methods: 1) the original method of [7] using the  $L_2$  gradient on the entire grid, 2) the original method with the Sobolev gradient in place of the  $L_2$  gradient, 3) the Narrow Band level set method of [8], and 4) the Narrow Band level set method with the Sobolev gradient. In all cases the level set function was initialized to a piecewise constant function with zero values on a large rectangle surrounding the objects. In the case of the (nonsmooth)  $L_2$  gradient, we were unable to find an effective test for convergence, and we therefore interactively terminated each test when the contour stopped changing noticeably. The execution times

include the time associated with displaying the evolving curves (which reduces the advantage of the narrow band). The appearance of the final contours for all four methods are nearly identical but, in the case of the Sobolev gradient, the contours are smoother and evolve in large steps. The time step size was 5.0 with the  $L_2$  gradient and 25.0 with the Sobolev gradient.

## References

- [1] W. B. Richardson Jr., Sobolev gradient preconditioning for image-processing PDEs, *Commun. Numer. Meth. Engng* 24 (2008) 493–504.
- [2] G. Sundaramoorthi, A. Yezzi, A. Mennucci, Sobolev active contours, *Int. J. Comp. Vis.* 73 (2007) 345–366.
- [3] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, *Int. J. Comp. Vis.* 1 (1987) 321–331.
- [4] V. Caselles, R. Kimmel, G. Shapiro, Geodesic active contours, *Int. J. Comp. Vis.* 22 (1997) 61–79.
- [5] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, *J. Comp. Phys.* 79 (1988) 12–49.
- [6] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-based fast local level set method, *J. Comp. Phys.* 155 (1999) 410–438.
- [7] C. Li, C. Xu, C. Gui, M. D. Fox, Level set formulation without re-initialization: a new variational formulation, *Proc. 2005 IEEE CVPR, San Diego, 2005*, 430–436.
- [8] C. Li, C. Xu, K. M. Konwar, M. D. Fox, Fast distance preserving level set evolution for medical image segmentation, *Proc. ICARCV06, 2006*, 1–7.
- [9] J. W. Neuberger, Sobolev gradients and differential equations, *Springer Lecture Notes in Mathematics #1670, 1997*.