Reformulating and Optimizing the Mumford-Shah Functional on a Graph — A Faster, Lower Energy Solution

Leo Grady and Christopher Alvino

Siemens Corporate Research — Department of Imaging and Visualization
755 College Road East, Princeton, NJ 08540

Abstract. Active contour formulations predominate current minimization of the Mumford-Shah functional (MSF) for image segmentation and filtering. Unfortunately, these formulations necessitate optimization of the contour by evolving via gradient descent, which is known for its sensitivity to initialization and the tendency to produce undesirable local minima. In order to reduce these problems, we reformulate the corresponding MSF on an arbitrary graph and apply combinatorial optimization to produce a fast, low-energy solution. The solution provided by this graph formulation is compared with the solution computed via traditional narrow-band level set methods. This comparison demonstrates that our graph formulation and optimization produces lower energy solutions than gradient descent based contour evolution methods in significantly less time. Finally, by avoiding evolution of the contour via gradient descent, we demonstrate that our optimization of the MSF is capable of evolving the contour with non-local movement.

1 Introduction

The Mumford-Shah functional (MSF) formulates the problem of finding piecewise smooth reconstructions of functions (e.g., images) as an optimization problem [1]. Optimizing the MSF involves determining both a function and a contour across which function smoothness is not penalized. Unfortunately, since smoothness of the reconstruction is not enforced across the contour and since the contour is variable in the optimization, the functional is not easily minimized using classical calculus of variations.

Given a fixed contour it is possible to solve for the optimal reconstruction function by solving a straightforward elliptic PDE with Neumann boundary conditions. Additionally, given a fixed piecewise smooth reconstruction function, it is possible to determine, at each point on the contour, the direction that the contour would move to decrease the functional as quickly as possible. Thus, most methods for solving the MSF involve alternating optimization of the reconstruction function and the contour [2,3,4]. The results of performing this type of optimization are well known and achieve satisfactory results that are used for different imaging applications [4]. Unfortunately, this optimization of the MSF using contour evolution techniques (typically implemented with level sets) is slow primarily due to the small steps taken by the contour at each iteration. This slowness is exacerbated by the fact that a small perturbation of the contour can have a relatively large effect on the optimal reconstruction function. Additionally, these

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 248–261, 2008.

traditional methods often require many implementation choices (e.g., implementation parameters) and these choices may produce differences in the final result.

Practical energy minimization problems formulated on a finite set of variables can be solved efficiently using combinatorial algorithms [5,6,7]. Furthermore, because of the well-established equivalence between the standard operators of multidimensional calculus and certain combinatorial operators, it is possible to rewrite many PDEs formulated in $\mathbb{R}^N$ equivalently on a complex (graph). Reformulating the conventional, continuous, PDE on a graph permits straightforward application of the arsenal of combinatorial optimization techniques to efficiently solve these variational problems. An alternate view of our approach is to consider rewriting the continuous energy functional in terms of the precise discrete operations that would be performed on a computer to evaluate the energy of a particular solution. By writing this energy in discrete terms, we can design our optimization method to optimize the energy value that would actually be measured by the computer. In this work, we reformulate the difficult MSF on a graph so that we may apply a combinatorial optimization to reduce the difficulties of speed and local minima associated with the small contour improvements obtained via traditional contour evolution. An added benefit of reformulating an energy in a combinatorial setting is that such a generic formulation may be applied without modification to higher dimensional data or general data analysis problems, such as point clustering, mesh smoothing/segmentation or space-variant vision.

Graph based optimization techniques have previously been used as components in optimization methods for functionals formulated in continuous space. Boykov *et al.* suggest using a max-flow step to assist in level set updates [8]. Zeng *et al.* [9] and El-Zehiry *et al.* [10] employ a max-flow operation as a component of their minimization of the piecewise constant MSF; we instead present a complete combinatorial reformulation and solution of the more general piecewise smooth MSF. Likewise, graph techniques have also been employed in the minimization of total variation methods [11].

Traditional contour evolution optimizations pursue a contour update in the direction of the highest gradient. Since this contour update represents a first variation of the MSF, calculation of this update does not require knowledge of the idealized foreground and background functions (images) at locations distant from the contour. In contrast, our graph formulation leads us to a combinatorial optimization approach that is capable of taking arbitrarily large steps of the contour location. Taking these large steps requires us to address the estimation of the foreground/background function values at locations (pixels) distant from the contour. To the knowledge of the authors, this work represents the first proposal of extending these foreground and background functions outside their region of evaluation.

2 Method

In this section, we define the continuous piecewise smooth Mumford-Shah model that we use, and go through each energy term to formulate the combinatorial analogue of the piecewise smooth MSF. With these combinatorial analogues, we then proceed to show how to optimize the foreground/background reconstruction and contour location.

2.1 Mumford-Shah Formulation: Continuous and Combinatorial

A **graph** consists of a pair $G = (V, E)$ with **vertices (nodes)** $v \in V$ and **edges** $e \in E \subseteq V \times V$, with $N = |V|$ and $M = |E|$. An edge, e, spanning two vertices, v_i and v_j, is denoted by e_{ij}. A **weighted graph** assigns a value to each edge called a **weight**. The weight of an edge, e_{ij}, is denoted by $w(e_{ij})$ or w_{ij} and is assumed to be nonnegative. The **degree** of a vertex is $d_i = \sum w(e_{ij})$ for all edges e_{ij} incident on v_i. The following will also assume that our graph is connected and undirected (i.e., $w_{ij} = w_{ji}$). An image may be associated with a graph by identifying each pixel with a node and defining an edge set to represent the local neighborhood of the pixels (e.g., a 4-connected lattice).

Since the inception of the Mumford-Shah functional, there have been several related notions of what constitutes *the* Mumford-Shah functional. In this work, we follow the level set literature to consider the piecewise smooth model [1,4], formulated as

$$E(f, g, R) = \alpha \left(\int_R (f - p)^2 + \int_{\Omega \backslash R} (g - p)^2 \right) + \mu \left(\int_R ||\nabla f||^2 + \int_{\Omega \backslash R} ||\nabla g||^2 \right) + \nu \Gamma(R) \,, \quad (1)$$

where Ω represents the image domain, f is the smoothed foreground function, g is the smoothed background function, R is the region of the image comprising the foreground, p is the pixel intensity, $\Gamma(R)$ is a function returning the length of the contour of region R, and α, μ, ν are free parameters. We assume that the image consists of grayscale values only, although the formulation can be extended to color or multispectral images. To simplify the parameter space, we assume that all three free parameters are strictly positive and divided by the value of μ. Thus, we will omit the inclusion of μ in the remaining exposition. Similar models were considered by Blake and Zisserman, who referred to the energy as the “weak membrane model” [12] and by the influential paper of Geman and Geman [13].

Formulation of (1) on a graph requires the use of combinatorial analogues of the continuous vector calculus operators (for an introduction to these combinatorial analogues, see [14]). Although combinatorial representations of differential operations are fairly well established, the challenge in the graph reformulation of any particular energy (or PDE) is to associate variables in the continuous formulation with representative combinatorial structures (pixels, edges, cycles, etc.) and, as in the continuous case, to produce a useful representation of a “contour”. Specifically, each integral may be considered as a pairing between a chain (domain of integration) and cochain (function to be integrated). Associating each pixel in our image with a node in the graph, the integration over a collection of pixels (in set $S_R \subseteq V$) may be represented by the $N \times 1$ chain vector r, where

$$r_i = \begin{cases} 1 & \text{if } v_i \in S_R, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The other two variables in E are cochains taking real values, i.e., $f_i \in \mathbb{R}, g_i \in \mathbb{R}$. Note also that the image I is treated as a vectorized, real-valued cochain existing on the nodes (pixels). Both chains and cochains will be treated as column vectors.

The first (data) term in (1) concerns quantities associated with pixels (i.e., intensities). We chose above to associate nodes with pixels, so p, f, and g must represent a 0-cochain (a function mapping nodes to real numbers). This matches the continuous conception of these quantities as scalar fields. Since the data term in (1) integrates over a set of the domain for which p, f and g are defined, r must represent a 0-chain indicating a region of the domain. Thus, the analogue of the first term on a graph is

$$E_1(f,g,r) = r^T\left(f-p\right)^2 + \left(1-r\right)^T\left(g-p\right)^2 . \tag{3}$$

In order to formulate the second term, recall that the combinatorial analogue of the gradient operator is given by the node-edge incidence matrix, A, [14]. Consequently, we may write the gradient of f as the product Af. However, since gradients are *vector functions* (corresponding to cochains on edges in the combinatorial setting) and the integral in the second term is performed over a *scalar* function (i.e., the norm of the gradient at each point), we have to transfer the gradient cochain associated with edges back to a scalar cochain associated with nodes. Such an operator may be represented by the absolute value of the incidence matrix, although each edge is now double counted, requiring a normalizing factor of one-half. Specifically, the second term may be formulated as

$$E_2(f,g,r) = \frac{1}{2}\left(r^T|A|^T\left(Af\right)^2 + \left(1-r\right)^T|A|^T\left(Ag\right)^2\right). \tag{4}$$

Finally, the contour length term may be formulated on a graph by counting the edges spanning from R to $\overline{R}$. Such a measure may be represented in matrix form as

$$E_3(f,g,r) = 1^T|Ar|. \tag{5}$$

If our graph is a standard 4-connected lattice, then (5) produces the ℓ_1 measure of the contour of region R. If we view the graph as embedded in $\mathbb{R}^N$ and wish to measure Euclidean contour length, it was shown [15] that a suitably weighted graph and corresponding incidence matrix could instead be used in (5). However, since this construction was designed to produce a Euclidean contour length, we use this construction only in term E_3. For purposes of generality and clarity here, we will continue to use the same A in all terms.

All three terms may now be put back together to define the combinatorial analogue of the piecewise smooth Mumford-Shah model, i.e.,

$$\begin{aligned} E(f,g,R) = \alpha\left(r^T\left(f-p\right)^2 + \left(1-r\right)^T\left(g-p\right)^2\right) + \\ \frac{1}{2}\left(r^T|A|^T\left(Af\right)^2 + \left(1-r\right)^T|A|^T\left(Ag\right)^2\right) + \nu 1^T|Ar|. \end{aligned} \tag{6}$$

Given the above definition of the combinatorial analogue of the Mumford-Shah functional, we now proceed to show how to optimize the variables f, g and r.

2.2 Optimization

We adopt the alternating optimization procedure common to optimization of the MSF [3,4]. The alternating optimization procedure first treats the current contour, r, as fixed and then finds the optimal f, g. Given an f and g, the optimal r may then be found. We begin by considering the production of an optimal f and g from a fixed contour, r.

Taking a partial derivative of (6) with respect to f yields

$$\frac{\partial E}{\partial f} = 2\alpha\,\mathrm{diag}(r)\,(f-p) + A^T\mathrm{diag}\left(|A|r\right)Af. \tag{7}$$

The $\mathrm{diag}(\cdot)$ operator represents the diagonal matrix formed by placing the argument vector on the diagonal. Since both the first and second terms of (6) are positive semi-definite, the zero of (7) represents a minimum of (6). Therefore, the optimal f given a contour satisfies

$$\left(2\alpha\,\mathrm{diag}(r) + A^T\mathrm{diag}\left(|A|r\right)A\right)f = 2\alpha\,\mathrm{diag}(r)p. \tag{8}$$

We pause to consider the interpretation of this optimum for f. The construction A^TCA, for diagonal C, produces the Laplacian matrix with edge weights given by the diagonal of C [14]. Note that in the standard conception of the MSF, these weights all equal unity — the domain is homogeneous, except at the contour. Consequently, (8) can be interpreted within region R as solving for the f that would be produced from initializing f within R to the image and then running linear isotropic diffusion for time equal to $\frac{1}{\alpha}$.

Outside of region R, any values of f will satisfy (8). In the computation of the energy in (6) this part of f does not contribute and may be ignored. In fact, in the existing literature, the values of f outside region R are never considered, since an infinitesimal gradient step is being taken by the contour of the level set function and values of f distant from the contour are inconsequential. However, in our combinatorial formulation, we desire to take an *optimal* contour step, regardless of the proximity of the new contour to the previous contour. Consequently, we will need to produce a meaningful f outside of region R. An important assumption about f is that it is a continuous function as it approaches the contour. Therefore, in order to enforce maximum smoothness between f inside R and the extended f outside of R, we propose to construct f outside of region R to satisfy the Laplace equation while treating the f inside of R (from (8)) as Dirichlet boundary conditions. We extend g inside of R similarly. Note, however, that other extensions of f and g are possible and may lead to improved performance. Using this construction, we may produce the optimal f inside the region as

$$\left(\alpha I + L_R\right)f_R = \alpha p_R, \tag{9}$$

where I is the identity matrix and L_R indicates the portion of the Laplacian matrix corresponding to the region R. Recall that the Laplacian matrix is defined $L = A^TCA$, for some diagonal matrix C taking the edge weights along the diagonal. We may solve the Laplace equation on a general graph, given boundary conditions, by using the technique of [16], which requires the solution to a linear system of equations with a subset of the Laplacian. Using the same procedure, the optimal g_R is given by solving the system

$$\left(\alpha I + L_{\overline{R}}\right)g_{\overline{R}} = \alpha p_{\overline{R}}, \tag{10}$$

and g_R may also be found by solving the combinatorial Laplace equation as in [16].

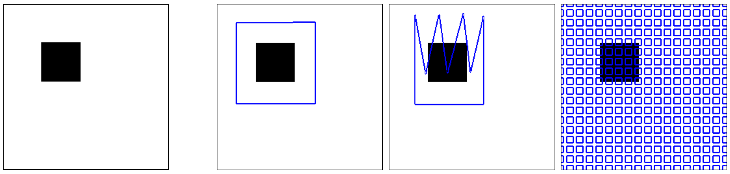
We can now address the optimization of r, given a fixed f and g. Noting that all three terms of (6) are submodular linear functions of r, we can solve for r as a max-flow/min-cut computation [6]. In effect, the first and second terms describe unary terms penalizing data infidelity from the reconstructed image and nonsmoothness in the reconstructed image. The third term penalizes contour length and is written in terms of strictly positive weights, producing a submodular energy that may be optimized effectively with a max-flow/min-cut computation. Minimum cut computations on graphs representing images are very fast using the algorithm of Boykov and Kolmogorov [17].

We conclude the section with observations about our graph formulation compared with discretized contour evolution approaches of the continuous energy. First, in contrast with standard continuous methods, at each iteration we are solving for a reconstructed function and contour that *optimally* minimize the MSF given a fixed contour or a fixed reconstruction, respectively. Due to these globally optimal steps, *all correct implementations will produce an equivalent answer*. Thus, there is no need for any implementation parameters. For example, any linear system solver will produce the same answer to (9). One method may be faster than another, but both methods will produce the same answer if run to convergence; thus, there is no need to be concerned that implementation choices will affect the *quality* of the solution. Second, because our contour optimization is not performed via gradient descent, the contour can move non-locally, and "snap" to the lowest energy contour, even for distant initializations. This non-local movement results in greater robustness to initialization, far fewer iterations and greater robustness to choice of the three term weightings in the MSF. Additionally, as shown in Section 3.3, this non-local movement capability allows our formulation to jump over intervening structures of arbitrary size to find a low-energy solution to the MSF.

2.3 Relationship to Graph Cuts

The Graph Cuts algorithm for image segmentation/denoising was first introduced in [18,19]. This algorithm has been greatly extended since inception to where it is somewhat unclear what comprises "Graph Cuts". However, all algorithms under the title "Graph Cuts" seem to have the following qualities: 1) Defined on a (possibly directed) graph, 2) Using submodular edge weights to reflect likely contour locations, 3) Possibly including an intensity prior assigning each pixel to foreground/background, 4) Possibly including hard constraints (seeds) to force pixels to be foreground or background, 5) Optimization via a max-flow/min-cut computation, 6) Produces a global optimum of the desired energy.

With this definition of "Graph Cuts", we observe that the contour optimization in the combinatorial formulation of the MSF, (6), shares much in common. Specifically, intensity priors are present (from the data term), the weights are submodular and the optimum of (6) is obtained via a max-flow/min-cut computation. However, by examining the above list, one may also notice differences with the combinatorial MSF. First, the edge weights are not modified to reflect image gradients. Second, in addition to the intensity priors, (6) involves an additional unary term penalizing the estimate of the normalized gradient near the pixel (from the smoothness term in the MSF). Third, no hard constraints (seeds) are imposed to constrain the labeling of any pixels. Fourth, there is no reconstructed image variable (i.e., f, g) present in Graph Cuts. Finally, the solution of r (contour) is just one part of one iteration in the overall optimization of the MSF. Although the solution of r is optimal for each iteration, the overall energy minimization



(a) Toy image (b) Box (c) Crown (d) Squares

Fig. 1. Toy image to compare the speed of traditional contour evolution implementation with proposed combinatorial optimization of the Mumford-Shah functional presented in this paper. The contours (blue) indicate different initializations used to generate the results in Table 1.

of the MSF still produces a local minimum. It should be noted that certain extensions of the Graph Cuts work also use Graph Cuts as a subroutine while re-estimating the intensity priors at each iteration [20]. However, unlike the MSF, this work does not include a specific smoothness penalty or a reconstructed image, but hard constraints are included and the edges are weighted by image gradients.

3 Results

The positives and negatives of MSF segmentation and reconstruction have been well-discussed in the literature. Our reformulation of the MSF on a graph is intended to permit the usage of combinatorial optimization methods to minimize the MSF more quickly and to find lower-energy solutions. Consequently, our experiments are dedicated to answering the following questions about the merits of traditional contour evolution optimizations of the MSF with the proposed combinatorial optimization applied to our graph formulation:

1. *Speed*: Which procedure finds a solution with fewer iterations? What is the relative cost per iteration? What is the dependence of performance on resolution?
2. *Initialization*: Which procedure is more robust to initialization of the contour?
3. *Parameters*: Which procedure is more robust to the choice of parameter settings?
4. *Energy minimization*: Which procedure produces solutions with lower energy?

To address the first three questions, we begin with a toy image of a black square on a white background. Such a trivial image was chosen since 1) There is a clear energy minimum, 2) A relatively smooth energy landscape, 3) The same answer for a wide range of parameters, 4) A clear stopping criterion (i.e., when the contour matches the square). For these reasons, we can perform controlled experiments to probe the answers to the questions posed above about the relative performance of traditional contour evolution implementations (via level sets) and our new graph formulation of the MSF.

We compared the combinatorial optimization of our graph formulation method with an efficient narrow-band level set implementation of the continuous formulation similar to the one presented in [4], although the original piecewise-smooth level set implementation was presented in [3]. Great care was taken to ensure the correctness and efficiency of the level set implementation so that a fair and accurate comparison could be made between the two methods. The method employed alternating optimizations of the contour

Table 1. Results of experiment comparing speed of convergence for level set (LS) solver and our graph (GR) formulation. Note: 1) The parameter settings were chosen to *best favor the level set method* in every experiment, 2) Exactly the same initializations were given to both algorithms, 3) The size and spacing of the squares initialization was chosen to favor the LS method. Time reported "per iteration" refers to update of the contour location, since computation of the reconstructed image is the same in both methods (although this computation is effectively doubled for GR since the inside/outside functions are extended beyond their respective region). Note that while the displayed number of level set iterations may seem particularly high, it is important to note that the initializations in these cases are very distant from the desired contour.

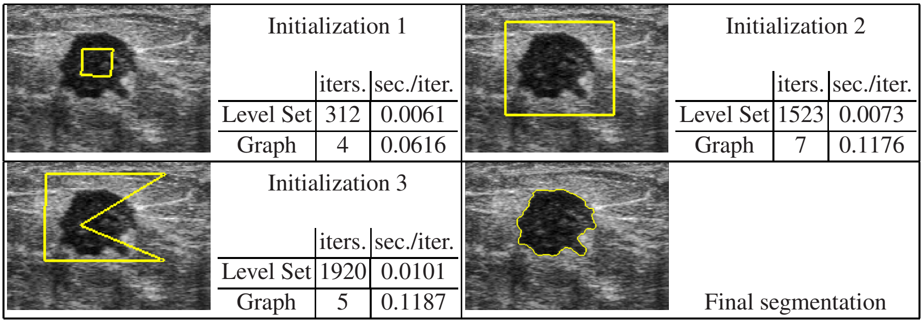
Initialization/Resolution	LS iterations	LS mean iter. time	GR iterations	GR mean iter. time
Box (64×64)	41	0.002s	2	0.0064s
Box (128×128)	126	0.0057s	2	0.0211s
Box (256×256)	140	0.0199s	2	0.0838s
Crown (64×64)	262	0.0023s	4	0.0091s
Crown (128×128)	1393	0.0061s	3	0.0239s
Crown (256×256)	110	0.0245s	4	0.1019s
Squares (64×64)	294	0.0072s	3	0.0094s
Squares (128×128)	940	0.0112s	3	0.0295s
Squares (256×256)	540	0.0624s	3	0.1177s

evolution and of the smooth functions as in the graph method and as has been used in all MSF minimizations of which we are aware. For efficiency, the level set function was computed and stored only in a narrow band around the contour, in which we maintained the sub-pixel position of the contour. Force extensions were computed on pixels which neighbored the contour as illustrated in [21]. When computing the level set function update, the spatial derivatives associated with the curvature term were computed with central differences, and the spatial derivatives associated with the data terms were computed with the numerical scheme detailed in [22] to ensure that the viscosity solution was obtained for the portion of the level set evolution that is a Hamilton-Jacobi equation. At each contour evolution step, we updated with an explicit forward-Euler scheme in which the maximally stable time step was taken to ensure both stability and speed of the level set function evolution.

Our implementation of max-flow/min-cut was taken directly from the online code of Vladimir Kolmogorov. In order to produce a comparable comparison between the level set optimization and our graph framework in these 2D experiments, we choose to calculate contour length of the cut with respect to a Euclidean measure in (5) by using the weighted incidence matrix of the graph corresponding to the construction of Boykov and Kolmogorov [15] with an approximation to the Euclidean distance represented by a neighborhood connected with a distance of two pixels.

3.1 Speed and Initialization

Our first experiment examines the relative speed of traditional level set implementations and our new graph formulation for the box image using various image resolutions and contour initializations. In this experiment, we created three initializations — A larger



Initialization 1

	iters.	sec./iter.
Level Set	312	0.0061
Graph	4	0.0616

Initialization 2

	iters.	sec./iter.
Level Set	1523	0.0073
Graph	7	0.1176

Initialization 3

	iters.	sec./iter.
Level Set	1920	0.0101
Graph	5	0.1187

Final segmentation

Fig. 2. Comparison of number of iterations and speed of iteration for different initializations on ultrasound image. Parameters were chosen to best benefit the level set method.

square surrounding the target square, an erratic "crown"-shaped initialization centered on the target square and small squares tiled throughout the image. These three initializations are displayed in Figure 1. For each of these initializations, we measured the number of iterations required to converge the level set and graph methods to the known optimum solution and the average time taken to produce one contour update for each method when run on an Intel Xeon 2.40GHz processor with 1GB of RAM. In this experiment, the parameters in the energy functional were chosen to favor the level set method as much as possible.

Table 1 displays the results of this experiment. The time reported "per iteration" in this table refers to the update of the contour location, since the computation of the reconstructed image is the same in both methods (although this computation is effectively doubled in our graph method since the inside/outside functions need to be extended beyond their region). Therefore, even though each iteration of our graph method is slightly more expensive than an iteration of the level set method, the improvement of 1–3 orders of magnitude in the number of iterations causes the total runtime of the graph method to be much less than that of the level set method. Additionally, the graph method converges within 2–4 iterations regardless of the resolution, initialization or parameters. Note that while the displayed number of level set iterations may seem particularly high, it is important to note that these initializations are very distant from the contour.

These experiments suggest that the proposed combinatorial optimization of the MSF produces a solution much faster than the traditional level set optimization, regardless of the resolution or contour initialization. We remind the reader that the energy term parameters were chosen to favor the level set method. Choosing the parameters to favor the proposed graph method would have resulted in an even stronger disparity in favor of our method.

A third experiment was performed on a real ultrasound image in the same manner as the first. Initializations were introduced inside the target object, outside the object and then erratically inside and outside the object. The results in terms of number of iterations and speed of each iteration are shown in Figure 2 and correspond well with the results from our synthetic experiment. Once again, the parameters of the terms in the MSF were chosen to favor the level set method and both methods converged to roughly the same contour.

3.2 Parameter Robustness

The choices of the term parameters in (1) can make drastic differences in the optimal contour and reconstruction produced by minimizing the MSF. Even if the optimal contour and reconstruction are the same for different choices of parameters, the parameter choices could affect the speed of convergence for a given initialization. In this experiment, we examine the robustness of both the contour evolution and graph formulations of the MSF to the choice of parameters in terms of the number of iterations needed to reach the optimum solution. Once again, we employ the toy example of Figure 1. For this experiment, we used the most simple, “box”, initialization of Figure 1 since we expect that both algorithms will reach the target contour for all parameter choices. We ran fifty iterations in which the parameters for each of the three terms of (1) were chosen independently from a uniform distribution within the interval of zero to one and then both the level set and graph algorithms were applied to minimize the MSF. If the target square was not the optimum solution for the randomly generated parameters, this parameter set was rejected and the trial re-run. After each parameter set, the number of iterations and average time per iteration were recorded.

Table 2. Comparison of robustness to the three term parameters in (1). Using the (128×128) toy image above with the “box” initialization, for 50 trials we randomly chose the three term parameters from independent uniform distributions on the interval $(0, 1)$ and ran both the level set and graph optimizations of the MSF. A set of parameters was rejected and re-run if the target square was not the minimum of the MSF. For all parameters, the graph optimization produced the target square in 2 iterations. Note that the number of iterations reported for the level set method in Table 1 was much less than the averages reported here because all of the results reported in Table 1 used parameters that were hand-selected to favor the level set convergence.

Optimization algorithm	Mean iterations	Median iterations	Iteration number standard deviation
Level set	1614.40	1520	391.80
Graph	2	2	0

The results of this experiment are displayed in Table 2. We see that the rate of convergence of the level set method is highly dependent on the parameters, while the rate of convergence for the graph method is completely independent of the parameter set. Both algorithms exhibited independence of the per iteration time on the parameter set. Empirically, the results of this experiment concur with our experience that the convergence rate, and solution achieved, of the graph method is much less sensitive to the parameter settings than the level set method. Note that the number of iterations reported for the level set method in Table 1 was much less than the averages reported in Table 2 due to the fact that all of the results reported in Table 1 used parameters that were hand-selected to favor the level set convergence.

3.3 Non-local Movement

A key advantage of the contour optimization in our graph reformulation of the MSF is that it enables movement to the optimal location at each iteration. For this reason,

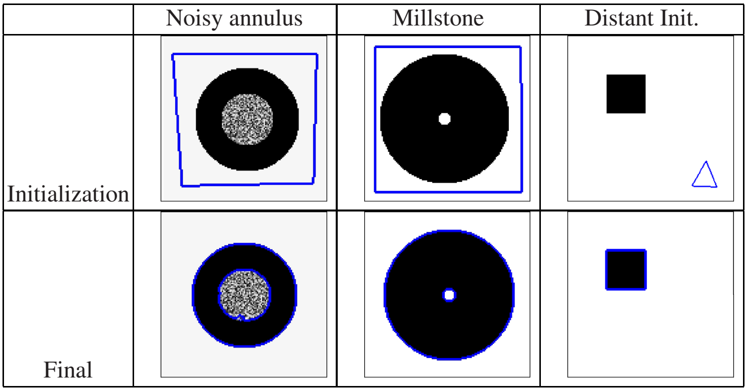


Fig. 3. Non-local movement: Since the contour optimization step in our graph formulation of the Mumford-Shah energy is not performed by gradient descent, the final contour is permitted to jump to a location distant from the initial contour location. While this effect is sometimes achieved in the level set literature by the use of mollified region indicator functions, we note that the mollifier support must be wider than the width of the annulus for such an approach to succeed.

our method is able to move to arbitrary image locations as predicted by the solution to (6) depending on the current estimate of the reconstruction functions. The motion of the contour is thus *not limited to local movements* as are traditional optimizations of the contour by gradient descent. Figure 3 illustrates three situations that are segmented correctly by the combinatorial optimization of the MSF, but where standard gradient descent methods fail.

The piecewise smooth MSF may drive non-local movement via insufficient smoothness, permitting the penetration of an annulus with a center comprised of pure noise. The final segmentation in Figure 3 is not achievable by gradient descent of the contour.

In the millstone image, we are able to achieve correct segmentation of the inner ring instantly. We would like to draw attention to the method by which Chan and Vese [2] were able to determine inner boundaries of objects. The ability to segment this inner boundary was only due to the mollified Heaviside function that was used to approximate a region indicator function. Indeed, the ability to achieve segmentation of an inner boundary in this manner is limited by the width of the resulting mollified Delta function, a quantity which should be kept low to maintain accuracy. We should also point out the work of [23] who indicate the ability to naturally attain such inner boundaries due to their method of total variation optimization of a modified MSF. Some level of non-local movement in solving the MSF with level set methods have been achieved in [24] using additive operator splitting [25], however they only illustrated the technique for the piecewise constant case.

Finally, we illustrate that distant (non-overlapping) initializations are not a problem to the combinatorial method as they are in gradient descent methods. Such a poor initialization could occur via automatic initialization of outlier image data. Regardless of the distance of the initialization from the object, our optimization is able to quickly ascertain such salient object boundaries.

3.4 Energy Minimization

Beyond speed, our purpose in introducing combinatorial optimization techniques for solving the MSF is to produce solutions with a lower energy than the solutions obtained by conventional level set techniques. In order to compare solutions in terms of minimal energy, we must address natural images for which the energy landscape is nontrivial. In this section we apply both the graph-based and level set algorithms to natural images using the same initialization/parameters and then compare the MSF energy obtained by the final solutions. The energy value is measured in exactly the same way for both algorithms — By evaluating (6). Using (6) to evaluate the energy might at first seem to be biased toward producing lower energies for the graph-based technique. However, it is important to realize that (6) details *in finite matrix form, the operations actually employed on a computer for estimating gradients and contour length*, with gradients computed by finite differences.

Our next experiment empirically compares the energy obtained for the solution of both optimizations for a variety of natural images, given the same initialization and parameters. For each image, initializations and parameters were selected to produce a contour (for at least one algorithm) that was semantically meaningful. Results of this experiment are shown in Figure 4. In every case, optimization of our graph formulation of the MSF produced solutions with an equal or lower energy and in less time.

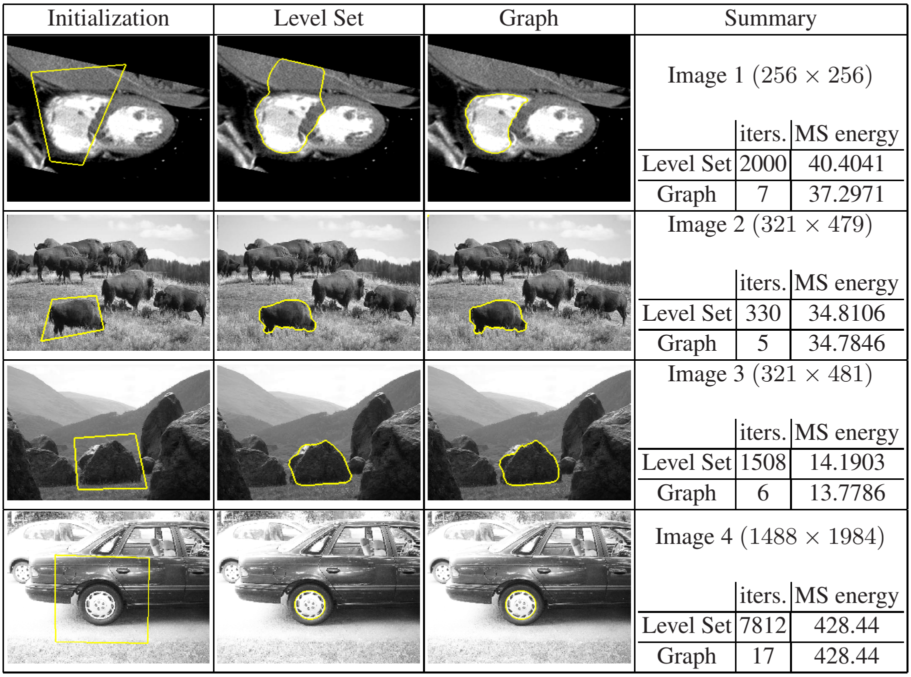


Image 1 (256 × 256)

	iters.	MS energy
Level Set	2000	40.4041
Graph	7	37.2971

Image 2 (321 × 479)

	iters.	MS energy
Level Set	330	34.8106
Graph	5	34.7846

Image 3 (321 × 481)

	iters.	MS energy
Level Set	1508	14.1903
Graph	6	13.7786

Image 4 (1488 × 1984)

	iters.	MS energy
Level Set	7812	428.44
Graph	17	428.44

Fig. 4. Comparison of the energy obtained for a solution produced by our graph formulation/optimization method with the traditional level set approach. For both algorithms, the initialization, parameters and energy calculation method were identical.

4 Conclusion

In this work, we began by reformulating the classical Mumford-Shah energy functional in terms of analogous differential operators on graphs. An equivalent, alternate way of looking at this reformulation is to think of it as writing the MSF explicitly in terms of the specific finite operations that would be applied on a computer to estimate gradients and contour length. With this new reformulation, we may apply well-established combinatorial optimization techniques for producing reconstruction and contour updates.

Our experiments indicate a dramatic improvement of our graph-formulated optimization over a traditional contour evolution approaches. This improvement is in terms of speed, robustness to initialization, robustness to parameter settings and in production of a solution representing a lower MSF energy. Additionally, we employ combinatorial optimization that is not based on gradient descent to solve our graph formulation of the MSF, which permits non-local movement of the contour to find low energy solutions. It could be argued that several modifications to the level set method have been suggested for improving speed (e.g., multiresolution [4]) that were not employed in our experiments. Although we did not employ these modifications to the level set method, it seems unlikely that they would converge on real images as quickly as the proposed method. Even if it were possible to modify the contour evolution approach to converge after a few iterations to a good, low-energy solution, such an achievement would only serve to make the modified level set approach roughly equivalent in performance to the proposed approach, at the cost of additional complexity and optimization parameters. The contour evolution approach would still not permit non-local movement of the contour. Additionally, some modifications (e.g., multiresolution) could equally be applied to improve the performance of both approaches.

Finally, we hope that this work has illustrated the idea that a reformulation of traditional (continuous) PDE approaches in terms of their analogous differential operators on graphs (combinatorial operators) can permit the use of powerful combinatorial optimization techniques that may more quickly find lower energy solutions when compared to their standard level set counterparts. Although our primary motivation for reformulating traditionally continuous energies in terms of combinatorial operators is to provide faster, simpler, lower energy solutions capable of non-local movement of contours, it is important to note that a graph-based formulation permits application of the same techniques to more abstract domains, such as data clustering, mesh processing and space-variant vision.

References

1. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. Pure and Appl. Math. 42, 577–685 (1989)
2. Chan, T., Vese, L.: Active contours without edges. IEEE TIP 10, 266–277 (2001)
3. Chan, T., Vese, L.: A level set algorithm for minimizing the Mumford-Shah functional in image processing. In: Workshop on VLSM, pp. 161–168. IEEE, Los Alamitos (2001)
4. Tsai, A., Yezzi, A., Willsky, A.: Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. IEEE TIP 10, 1169–1186 (2001)

5. Greig, D., Porteous, B., Seheult, A.: Exact maximum *a posteriori* estimation for binary images. Journal of the Royal Statistical Society, Series B 51, 271–279 (1989)
6. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE PAMI 26, 147–159 (2004)
7. Sinop, A.K., Grady, L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In: Proc. of ICCV. IEEE, Los Alamitos (2007)
8. Boykov, Y., Kolmogorov, V., Cremers, D., Delong, A.: An integral solution to surface evolution PDEs via geo-cuts. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 409–422. Springer, Heidelberg (2006)
9. Zeng, X., Chen, W., Peng, Q.: Efficiently solving the piecewise constant Mumford-Shah model using graph cuts. Technical report, Zhejiang University (2006)
10. El-Zehiry, N., Xu, S., Sahoo, P., Elmaghraby, A.: Graph cut optimization for the Mumford-Shah model. In: Proc. of VIIP (2007)
11. Bougleux, S., Elmoataz, A., Melkemi, M.: Discrete regularization on weighted graphs for image and mesh filtering. In: SSVM, pp. 128–139. Springer, Heidelberg (2007)
12. Blake, A., Zisserman, A.: Visual Reconstruction. MIT Press, Cambridge (1987)
13. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE PAMI 6, 721–741 (1984)
14. Mattiussi, C.: The finite volume, finite element and finite difference methods as numerical methods for physical field problems. In: AIEP, pp. 1–146. Academic Press Inc., London (2000)
15. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: Proc. ICCV, vol. 1, pp. 26–33 (2003)
16. Grady, L., Schwartz, E.: Anisotropic interpolation on graphs: The combinatorial Dirichlet problem. Technical Report CAS/CNS-TR-03-014, Boston University (2003)
17. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE PAMI 26, 1124–1137 (2004)
18. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE PAMI 23, 1222–1239 (2001)
19. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: Proc. of ICCV 2001, pp. 105–112 (2001)
20. Rother, C., Kolmogorov, V., Blake, A.: “GrabCut” — Interactive foreground extraction using iterated graph cuts. In: Proc. SIGGRAPH, vol. 23, pp. 309–314. ACM, New York (2004)
21. Sethian, J.: Level set methods and fast marching methods: evolving interfaces in computational geometry. Cambridge University Press, Cambridge (1999)
22. Osher, S., Sethian, J.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics 79, 12–49 (1988)
23. Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J., Osher, S.: Fast global minimization of the active contour/snake model. J. Mathematical Imaging and Vision 28, 151–167 (2007)
24. Wang, Z., Yang, X., Shi, P.: Solving Mumford-Shah model equation by AOS algorithm. In: Int. Conf. on Sig. Proc., vol. 1, pp. 740–743. IEEE Computer Society Press, Los Alamitos (2002)
25. Weickert, J., Romeny, B.M., Viergever, M.A.: Efficient and reliable schemes for nonliner diffusion filtering. IEEE Trans. on Image Proc. 7, 398–410 (1998)