

# The Piecewise Smooth Mumford-Shah Functional on an Arbitrary Graph

Leo Grady, *Member, IEEE*, Christopher Alvino

**Abstract**—The Mumford-Shah functional has had a major impact on a variety of image analysis problems including image segmentation and filtering and, despite being introduced over two decades ago, it is still in widespread use. Present day optimization of the Mumford-Shah functional is predominated by active contour methods. Until recently, these formulations necessitated optimization of the contour by evolving via gradient descent, which is known for its overdependence on initialization and the tendency to produce undesirable local minima. In order to reduce these problems, we reformulate the corresponding Mumford-Shah functional on an arbitrary graph and apply the techniques of combinatorial optimization to produce a fast, low-energy solution. In contrast to traditional optimization methods, use of these combinatorial techniques necessitates consideration of the reconstructed image outside of its usual boundary, requiring additionally the inclusion of regularization for generating these values. The energy of the solution provided by this graph formulation is compared with the energy of the solution computed via traditional gradient descent-based narrow-band level set methods. This comparison demonstrates that our graph formulation and optimization produces lower energy solutions than the traditional gradient descent based contour evolution methods in significantly less time. Finally, we demonstrate the usefulness of the graph formulation to apply the Mumford-Shah functional to new applications such as point clustering and filtering of non-uniformly sampled images.

**Index Terms**—Level sets, active contours, piecewise smooth Mumford-Shah, combinatorial optimization, graph reformulation

## I. INTRODUCTION

THE Mumford-Shah functional was devised to formulate the problem of finding piecewise smooth reconstructions of functions (e.g., images) as an optimization problem [43]. Optimizing the Mumford-Shah functional involves determining both a function and a contour across which 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 an 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 and speed that the contour should move to decrease the functional as quickly as possible. Thus, most methods for solving the Mumford-Shah functional involve alternating optimization of

the reconstruction function and the contour [19], [20], [55], [14]. The results of performing this style of optimization are well known and achieve satisfactory results that are usable for different imaging applications [55]. Unfortunately, this optimization of the Mumford-Shah functional using contour evolution techniques (typically implemented with gradient descent on level sets) is slow primarily due to the small steps that the contour must take at each iteration. This slowness is exacerbated by the fact that a very small perturbation of the contour can have a relatively large effect on the optimal reconstruction function. Additionally, these traditional methods often require many implementation choices (e.g., implementation parameters) and the result of these choices may cause differences in the final result.

Although new functionals for segmentation/filtering continue to be developed, the Mumford-Shah functional is still very widely used and optimized with level set methods (see [26], [35], [22], [59], [46], [25] for a variety of recent applications). In addition to applications, recent work in the computer vision community has also continued to address theoretical aspects of the Mumford-Shah functional and its optimization. Brox *et al.* have recently shown the Mumford-Shah functional to be interpretable as a first order approximation of an *a posteriori* maximization [14]. Efficient level set segmentations based on the Mumford-Shah functional have been presented by Piovano *et al.* in [45], who show that the global statistics can be approximated with Gaussian weighted local statistics. These methods perform their minimization with level sets based on the standard gradient descent approach. Also recent methods have appeared that present piecewise smooth segmentations without use of gradient descent. Bresson *et al.* has shown, based on the works of Chan *et al.* and Chambolle, that the fast global minimization of the boundary (with functions held fixed) in a modified Mumford-Shah functional is achievable without use of explicit curve evolution by taking advantage of a dual formulation of the TV-norm [13], [21], [18]. An *et al.* achieve state-of-the-art piecewise smooth segmentation without the explicit use of contour evolution by performing gradient descent minimization of phase fields [2]. Hintermüller *et al.* has reported inexact Newton-style optimization on the contour for the Mumford-Shah functional using conjugate gradients [36].

Practical energy minimization problems formulated on a finite series of variables can often be solved efficiently using combinatorial (graph-based) algorithms [34], [38], [53], [50]. 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 cell complex (graph). By reformulating the conventional, continuous, PDE on a graph it becomes straightforward to apply the arsenal of combinatorial optimization techniques to efficiently solve these variational problems. In this work, we reformulate the Mumford-Shah functional on a graph in order to reap the benefits of combinatorial optimization to reduce the difficulties of speed and local minima associated with the small contour improvements obtained via traditional contour evolution.

An additional benefit of reformulating an energy in a discrete, combinatorial setting is that such a generic formulation may be applied without modification to general data analysis problems, such as point clustering [23], mesh smoothing (segmentation) [54] or space-variant vision [47]. Even in the context of Cartesian image processing, recent work has examined modified neighborhood structures to generate improved segmentation results [39], [33]. Our combinatorial formulation allows for a straightforward application of the Mumford-Shah optimization in these scenarios. For example, the popular NL-Means algorithm of [16] may simply be interpreted as the addition of non-local neighborhoods (graph edges) to the graph with specific edge weights [28]. This perspective on the NL-Means algorithm allows us to directly apply our combinatorial Mumford-Shah algorithm to the graph defined by the extra edges generated by NL-Means. In this manner, the graph construction and energy optimization may be separated and addressed independently. In Section IV-B we demonstrate the use of the graph formulation to perform point clustering and to filter images defined on a space-variant data structure modeled after a biological sampling of image space.

Graph based optimization techniques have been previously used as components in optimization methods for functionals formulated in continuous space. Boykov and Cremers suggest using a max-flow/min-cut step to assist in level set updates [10]. Zeng *et al.* [60] and El-Zehiry *et al.* [27] employ a max-flow/min-cut operation as a component of their piecewise constant Mumford-Shah computations; we instead present a complete combinatorial reformulation and solution of the more general piecewise smooth Mumford-Shah functional. Likewise, graph methods have also been employed in the minimization of total variation for image filtering tasks [24], [6].

Traditional contour evolution optimizations pursue a contour update in the direction of steepest descent. Since this contour update represents a first variation of the Mumford-Shah functional, the calculation of this update does not require knowledge of the values of the idealized foreground and background functions (images) at locations distant from the contour. In contrast, the graph formulation that we pursue leads us to a combinatorial optimization approach that is capable of taking arbitrarily large steps of the contour location. In order to take these large steps, it is necessary for us to address the estimation of the foreground/background function values at locations (pixels) distant from the contour. The necessity for extending the foreground and background functions outside of their regions of evaluation, in the context of a global boundary optimization, have been noted in the literature [56], [13], [61], [15]. Our method of performing this extension is detailed in

Section II-C.

Graph formulations of the Mumford-Shah have previously appeared in the literature, but these variants employ inefficient optimization techniques and do not necessarily generalize to arbitrary graphs. Yu *et al.* formulated the Mumford-Shah energy on a lattice and used a greedy algorithm for performing the boundary update. Similarly, the MRF formulation of Geman and Geman [29] (and later addressed explicitly by Mumford [42]) is commonly viewed as equivalent to a discrete formulation of the Mumford-Shah functional in light of its convergence in the limit to the continuous Mumford-Shah model [17]. Our contributions in this paper are to: 1) Appeal to the well-established combinatorial analogues of differential operators to formulate our discrete version of the Mumford-Shah, yielding a different discrete instantiation of the Mumford-Shah functional than what has previously appeared in the literature (particularly the smoothness term), 2) Show that this discrete version of the Mumford-Shah can be optimized efficiently using combinatorial optimization tools only if values of the reconstructed foreground intensities are extended beyond the foreground region, 3) Propose a method for extending these intensity values to the entire graph (domain), 4) Show that the resulting algorithm strongly outperforms existing gradient descent-based level set optimizations of the Mumford-Shah functional, 5) Show how to apply the combinatorial formulation of the Mumford-Shah functional to nontraditional problems such as point clustering and nonuniform image processing.

In this work, we begin in Section II by reformulating the Mumford-Shah functional on a graph and then showing how to perform the optimization using known combinatorial methods. In Section III we perform several experiments to compare the combinatorial optimization associated with our graph formulation to the traditional gradient descent-based level set implementation. Our experiments focus on the relative speed of the two methods, as well as the convergence rate, robustness to initialization, robustness to parameter settings and the production of a lower-energy solution. In Section IV-A we show that the contour evolution produced via our combinatorial method is capable of non-local movement by taking arbitrarily large steps between iterations. Section IV-B shows how to use our general formulation of the Mumford-Shah functional to analyze data on a non-lattice graph. Finally, in Section V, we draw conclusions about the results of these experiments and suggest directions for future work.

## II. METHOD

In this section, we first define the continuous piecewise smooth Mumford-Shah model that we use. After this definition, we consider each of the three terms and formulate the combinatorial analogue of the piecewise smooth Mumford-Shah functional. With these combinatorial analogues, we proceed to show how to perform an optimization of the variables and then conclude this section with a summary of steps in the algorithm.

### A. Mumford-Shah formulation: Continuous and Combinatorial

We begin by fixing our notation. 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 here 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 relationship 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 [43], [55], formulated as

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

where  $\Omega$  represents the image domain,  $f$  represents the smooth foreground function,  $g$  is the smooth background function,  $R$  is the region of the image comprising the foreground,  $p$  is the pixel intensity,  $\Gamma(R)$  is the length of the contour of region  $R$ , and  $\alpha, \mu, \nu$  are free parameters. For ease of exposition, we assume that the image consists of grayscale values only, although the formulation could easily be extended to color or multispectral images. Although the piecewise-smooth Mumford-Shah model has proven useful images in which the objects are piecewise-smooth, other types of object appearance (e.g., textured objects) are not well-modeled by this form of the Mumford-Shah functional. To simplify the parameter space (and notation) we assume that all three free parameters are strictly positive and divided by the value of  $\mu$ . Consequently, we will omit the inclusion of  $\mu$  in the remaining exposition. Similarly motivated models were considered by Blake and Zisserman, who referred to the energy as the “weak membrane model” [5] and by the influential paper of Geman and Geman [29]. A comprehensive exposition on the properties of this model was provided by Ambrosio *et al.* [1].

Our formulation of (1) on a graph employs the combinatorial analogues of continuous differential operators (for justification and introduction to these combinatorial analogues, see [12], [40], [51]). 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 a 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

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

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 0-cochains (functions mapping nodes to real numbers). This designation 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. Putting together the above observations, the analogue of the first term on a graph is

$$E_1(f, g, r) = r^T (f - p)^2 + (1 - r)^T (g - p)^2. \quad (3)$$

In order to formulate the second term, recall that the combinatorial analogue of the gradient operator is given by the node-edge **incidence matrix** [40]

$$A_{e_{ij}v_k} = \begin{cases} +1 & \text{if } i = k, \\ -1 & \text{if } j = k, \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

for every vertex  $v_k$  and edge  $e_{ij}$ , where each  $e_{ij}$  has been arbitrarily assigned an orientation. 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 factor of one-half. Specifically, the second term may be formulated as

$$E_2(f, g, r) = \frac{1}{2} \left( r^T |A|^T (Af)^2 + (1 - r)^T |A|^T (Ag)^2 \right). \quad (5)$$

Finally, the contour length term may be formulated on a general graph by counting the number of edges spanning from  $R$  to  $\bar{R}$ , where  $\bar{R}$  indicates the set complement of  $R$ . Such a measure may be represented in matrix form as

$$E_3(f, g, r) = 1^T |Ar|. \quad (6)$$

If our graph is a standard 4-connected lattice (i.e., a 5-point stencil), then (6) produces the  $\ell_1$  measure of the contour of region  $R$ . If we view the graph as embedded in  $\mathbb{R}^N$  and wish to measure a Euclidean contour length, it was shown by Boykov and Kolmogorov [8] that a suitably weighted graph (and corresponding incidence matrix) could instead be used



in (6). However, since the Boykov-Kolmogorov construction was designed to produce a Euclidean contour length, we recommend only using this construction 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 recomposed to define the combinatorial analogue of the piecewise smooth Mumford-Shah model, i.e.,

$$E(f, g, R) = \alpha \left( r^T (f - p)^2 + (1 - r)^T (g - p)^2 \right) + \frac{1}{2} \left( r^T |A|^T (Af)^2 + (1 - r)^T |A|^T (Ag)^2 \right) + \nu 1^T |Ar|. \quad (7)$$

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  $r$ .

### B. Optimization

We adopt the alternating optimization procedure common to Mumford-Shah optimization [20], [55]. The alternating optimization procedure first treats the current contour,  $r$ , as fixed and then finds the optimal  $f$  and  $g$ . 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,  $r$ .

Before taking a partial derivative of (7) with respect to  $f$ , it is useful to rewrite the terms of (7) involving  $f$  in matrix form,

$$E(f, R) = \alpha (f - p)^T \text{diag}(r) (f - p) + \frac{1}{2} f^T A^T \text{diag}(|A|r) Af. \quad (8)$$

Taking a partial derivative of (8) with respect to  $f$  yields

$$\frac{\partial E}{\partial f} = 2\alpha \text{diag}(r) (f - p) + A^T \text{diag}(|A|r) Af. \quad (9)$$

The  $\text{diag}(\cdot)$  operator represents the diagonal matrix formed by placing the argument vector along the diagonal. Since both the first and second terms of (8) are positive semi-definite, the zero of (9) represents a minimum of (8). Consequently, the optimal  $f$  given a contour satisfies

$$(2\alpha \text{diag}(r) + A^T \text{diag}(|A|r) A) f = 2\alpha \text{diag}(r) p. \quad (10)$$

Consider the set  $R^* = \{v_i | \text{if } v_i \in R \text{ or } \exists e_{ij} \text{ s.t. } v_j \in R\}$  which includes all of the nodes in  $R$  as well as any nodes in  $\bar{R}$  that are connected with an edge to  $R$ . From (10) it is clear that the nodes in the complement of  $R^*$ ,  $\bar{R}^*$ , can take any value of  $f$  while satisfying (10). In other words, this part of  $f$  does not contribute to the energy in (7) and therefore may be ignored. Consequently, the optimum values of  $f$  in  $R^*$  may be found by solving

$$(2\alpha \text{diag}(r_{R^*}) + A_{R^*}^T \text{diag}(|A_{R^*}| r_{R^*}) A_{R^*}) f_{R^*} = 2\alpha \text{diag}(r_{R^*}) p_{R^*}. \quad (11)$$

It is important to note that since the energy computation in (7) reaches across the boundary by one pixel to produce the smoothness estimate, the optimum solution for  $f$  includes values at these pixels across the border. Since the system in (11) will generally force  $x_i = x_j$  across the border separating

nodes  $v_i$  and  $v_j$ , there will be no contribution to the energy in (7) from a gradient between these nodes. However, if any of the border nodes in  $\bar{R}$  are connected to more than one node inside  $R$  (e.g., at a corner) then the solution in (11) will cause the border node to take the average value of the nodes inside  $R$  and this node will contribute to the energy in (7). This one-pixel reach across the boundary is a result of the combinatorial analogue of the gradient norm and is necessary to ensure optimizability of the energy in (7), but does not empirically contribute in any significant way to the energy of the final solution.

We may follow the same steps as above to find the solution for  $g$ . Specifically, define set  $Q = \bar{R}$  and  $Q^* = \{v_i | \text{if } v_i \in Q \text{ or } \exists e_{ij} \text{ s.t. } v_j \in Q\}$ , leading to a solution of

$$(2\alpha \text{diag}(r_{Q^*}) + A_{Q^*}^T \text{diag}(|A_{Q^*}| r_{Q^*}) A_{Q^*}) g_{Q^*} = 2\alpha \text{diag}(r_{Q^*}) p_{Q^*}. \quad (12)$$

We can now address the optimization of  $r$ , given a fixed  $f$  and  $g$ . Noting that all three terms of (7) are submodular linear functions of  $r$ , we can solve for  $r$  as a max-flow/min-cut computation [38]. The reduction to a max-flow/min-cut computation may be more easily seen by writing the energy of (7) in traditional summation form as

$$E(f, g, r) = \sum_{v_i} \left( r_i \left( \alpha (f_i - I_i)^2 + \frac{1}{2} \sum_{e_{ij}} (f_i - f_j)^2 \right) \right) + \sum_{v_i} \left( (1 - r_i) \left( \alpha (g_i - I_i)^2 + \frac{1}{2} \sum_{e_{ij}} (g_i - g_j)^2 \right) \right) + \nu \sum_{e_{ij}} |r_i - r_j|. \quad (13)$$

In effect, the first and second terms describe unary terms penalizing data infidelity from the reconstructed image and nonsmoothness in the reconstructed image. The contour length (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 [9].

Unfortunately, use of a max-flow/min-cut computation to optimize (13) requires knowledge of  $f$  and  $g$  in the entire domain. However, since the values in  $f$  outside of  $R^*$  and the values in  $g$  outside of  $Q^*$  make no contribution to the energy in (7) we could use any solution for  $f$  and  $g$  in these regions without affecting the energy. Consequently, some regularization is necessary to assign these values by extending the known solutions for  $f_{R^*}$  and  $g_{Q^*}$ , so that the optimization in (13) can be performed.

### C. Extension of the reconstructed image

Outside of region  $R^*$ , any values of  $f$  will satisfy (10). Consequently, this part of  $f$  does not contribute to the computation of the energy in (7) and could simply be ignored if the values were not necessary to produce an optimal boundary

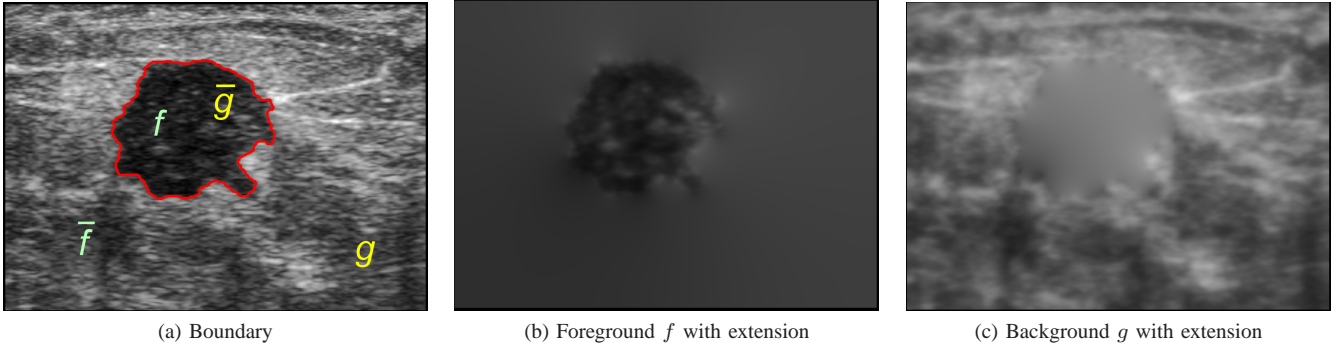


Fig. 1: Optimization of (13) using max-flow requires specification of values for the reconstructed foreground image function  $f$  in the region outside the foreground region  $R$ . Using a Laplace equation regularizer obtains the values for  $f$  and  $g$  for the entire image domain shown above.

location via minimization of (13). In fact, the existing level set literature that optimizes the contour via gradient descent, ignores the values of  $f$  outside region  $R$ , since the values of  $f$  distant from the contour are inconsequential. However, in our combinatorial formulation, we desire to take a contour step on the entire domain, regardless of the proximity of the new contour to the previous contour. Consequently, we will need to produce a meaningful  $f$  outside of region  $R^*$ . Although any choice of  $f$  outside of  $R^*$  will have no impact on the energy, the solution for  $f$  within  $R^*$  is uniquely determined relative to any boundary. Therefore, our approach will be to determine the optimal  $f$  within  $R^*$  by solving (11) and use this solution to generate an  $f$  outside of  $R^*$  that may be used as guidance for generating the next contour optimization.

An important assumption about  $f$  is that it is a continuous function at the contour. Therefore, in order to enforce maximum smoothness between  $f$  inside  $R^*$  and the extended  $f$  outside of  $R^*$ , we propose to regularize the unknown part of  $f$  by constructing the  $f$  outside of region  $R^*$  that satisfies the Laplace equation while treating the  $f$  inside of  $R^*$  (obtained from (10)) as Dirichlet boundary conditions. We apply the same procedure for extending  $g$  outside  $Q^*$ . Note, however, that other extensions of  $f$  and  $g$  are possible and may lead to improved performance.

We pause briefly to discuss the relationship of our method of extension to those done previously. The method mentioned here is most similar to that of Vese and Chan in [56] where they essentially discuss first solving the functions  $f$  and  $g$  inside their appropriate domains in order to minimize the functional, and then extending each of these functions outside of their respective domains by solving the Laplace equation with Dirichlet boundary conditions on the contour, and Neumann boundary conditions on the image boundary. A seemingly similar method of producing function extensions has been documented by Brox *et al.* in [15] where they compare various approximations to a single PDE containing an indicator function that can be solved to produce a meaningful function on the entire domain. While the PDE is still elliptic in this method, when the single PDE is solved the values of the function outside of the active domain have the potential to affect the values of the function inside the active domain, and

therefore the resulting function may not be optimal at each step inside the active domain. This is not true of the two-step approach both presented in [56] and also used presently. Zhao *et al.* discusses the method of extending functions via a Hamilton-Jacobi equation that is used in much of the previous level set work [61]. Bresson *et al.* discuss the necessity for extensions, but leave out the details for their specific extension method [13].

We may solve the Laplace equation on a general graph, given boundary conditions, by using the technique of [31], which we briefly review. Recall that the Laplacian matrix is defined as

$$L_{ij} = \begin{cases} d_i & \text{if } i = j, \\ -w_{ij} & \text{if } v_i \text{ and } v_j \text{ are adjacent nodes,} \\ 0 & \text{otherwise,} \end{cases} \quad (14)$$

and that  $L = A^T C A$ , for some diagonal matrix  $C$  taking the edge weights along the diagonal. If we treat the solution to (11),  $f_{R^*}$ , as fixed Dirichlet boundary conditions, we can decompose the Laplacian matrix into

$$\begin{bmatrix} L_{R^*} & B \\ B^T & L_{\bar{R}^*} \end{bmatrix}, \quad (15)$$

which allows us to solve the combinatorial Laplace equation by solving the system

$$L_{\bar{R}^*} f_{\bar{R}^*} = -B^T f_{R^*}. \quad (16)$$

In summary, the optimum for  $f_{R^*}$  and  $f_{\bar{R}^*}$  may be produced by solving sparse, positive definite systems of linear equations for which many fast methods exist (see [30]).

Following the same steps as above, the optimal  $g_{\bar{Q}^*}$  is given by solving the linear system

$$L_{\bar{Q}^*} g_{\bar{Q}^*} = -B^T g_{Q^*}. \quad (17)$$

An example of the  $f_R$ ,  $f_{\bar{R}}$ ,  $g_R$  and  $g_{\bar{R}}$  computed for a given boundary with this regularization method is shown in Figure 1.

We conclude the section with observations about how our graph formulation compares with discretized contour evolution approaches of the continuous energy. First, in contrast to the standard continuous methods, at each iteration we are

solving for a reconstructed image and contour that *optimally* minimize the Mumford-Shah energy, given a fixed contour (for the reconstruction computation) or a fixed reconstruction (for the contour computation). An important consequence of these globally optimal steps is that *all correct implementations will produce an equivalent answer*. Since the same answer is produced regardless of implementation, there is no need to choose any implementation parameters. For example, any linear system solver run to convergence will produce the same answer to (11). Therefore, Gaussian elimination might be faster or slower than conjugate gradients, but both methods will produce the same answer if run to convergence and therefore there is no need to be concerned that implementation choices will have an effect on the *quality* of the final solution. Second, because our contour optimization is not performed via gradient descent on the contour location, the contour update is capable of non-local movement to “snap” to the best contour, even if initialized far away. This non-local movement results in greater robustness to initialization, far fewer iterations (translating to faster speed) and greater robustness to choices of weightings for the three terms in the Mumford-Shah functional. Additionally, as shown in Section IV-A, this non-local movement capability allows our graph formulation to jump over intervening structures of arbitrary size to find a low-energy solution to the Mumford-Shah functional. We note that similar properties (non-local movement, decreased dependence on implementation parameters) have also been obtained by recent continuous formulations that produce global optimization of the contour at each step [13], [21].

Although our graph formulation produces optimal solutions for each alternating step of the Mumford-Shah minimization, it is important to stress that there is no guarantee that a global optimum will be obtained for the joint energy. In fact, it is entirely unclear whether or not alternating global energy minimization steps will produce a lower joint energy than a series of small alternating local energy minimization steps. Therefore it is an empirical question to determine which optimization strategy produces a lower joint energy for real images. Both optimization strategies (graph and level set) are empirically tested in Section III against a battery of natural images to better understand which optimization strategy works best on the joint energy in practice.

#### D. Relationship to Graph Cuts

The Graph Cuts algorithm for image segmentation and denoising was first introduced in [11], [7]. 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 the above definition of the term “Graph Cuts”, it is possible to observe that the contour optimization of (13) in

the combinatorial formulation of the Mumford-Shah shares much in common. Specifically, in (13), intensity priors are present (from the data term), the weights are submodular and the optimum of (13) is obtained via a max-flow/min-cut computation. However, by examining the above list of ingredients for Graph Cuts, one may also notice differences with the combinatorial Mumford-Shah. First, the edge weights are not modified to reflect likely contour gradients. Second, in addition to the intensity priors, (13) involves an additional unary term penalizing the estimate of the normalized gradient near the pixel (obtained via the smoothness term in the Mumford-Shah functional). Third, no hard constraints (seeds) are imposed to constrain the foreground/background assignment of any pixels. Fourth, there is no reconstructed image variable (i.e.,  $f$ ,  $g$ ) present in Graph Cuts. Finally, the solution of (13) is just one iteration in the overall optimization of the Mumford-Shah functional. Although the contour solution of (13) is optimal for each iteration, the overall energy minimization of the Mumford-Shah energy still produces a local minimum. It should be noted that certain extensions of the Graph Cuts work (e.g., GrabCuts [4], [48]) also utilize Graph Cuts as a subroutine while re-estimating the intensity priors at each iteration. However, unlike the Mumford-Shah formulation, this work does not include a specific smoothness penalty term or a reconstructed image, hard constraints are included and the edges are weighted by image gradients. Additionally, it should be noted that we are not arguing that the Mumford-Shah functional is better or worse than these other energy minimization approaches, but rather that the Mumford-Shah model is still heavily applied and that the optimization is currently done using gradient descent on level set functions. Given the traditional successes of applying a minimization of the Mumford-Shah energy and the recent successes of Graph Cuts (and its derivatives), it is not surprising that the algorithms bear a strong relationship.

#### E. Algorithm summary

The steps of the combinatorial Mumford-Shah algorithm are therefore:

- 1) Initialize the functions  $f$ ,  $g$  and the contour  $r$ .
- 2) Solve the linear system in (11) for the optimum  $f_{R^*}$ .
- 3) Solve the linear system in (16) for the extension of  $f_{R^*}$  to  $f_{\bar{R}^*}$ .
- 4) Solve the linear system in (12) for the optimum  $g_{Q^*}$ .
- 5) Solve the linear system in (17) for the extension of  $g_{Q^*}$  to  $g_{\bar{Q}^*}$ .
- 6) Solve the max-flow/min-cut problem in (13) for the optimum contour  $r$ .
- 7) Repeat steps 2–6 until the contour location remains constant.

Finally, we make a few observations about this procedure:

- 1) Traditional gradient descent-based level set implementations of the Mumford-Shah functional are not required to perform steps 3 and 5 on the entire image domain, since the extensions of  $f$  and  $g$  far from the contour are not pertinent to a gradient descent method. Although the inclusion of steps 3 and 5 are expected to double

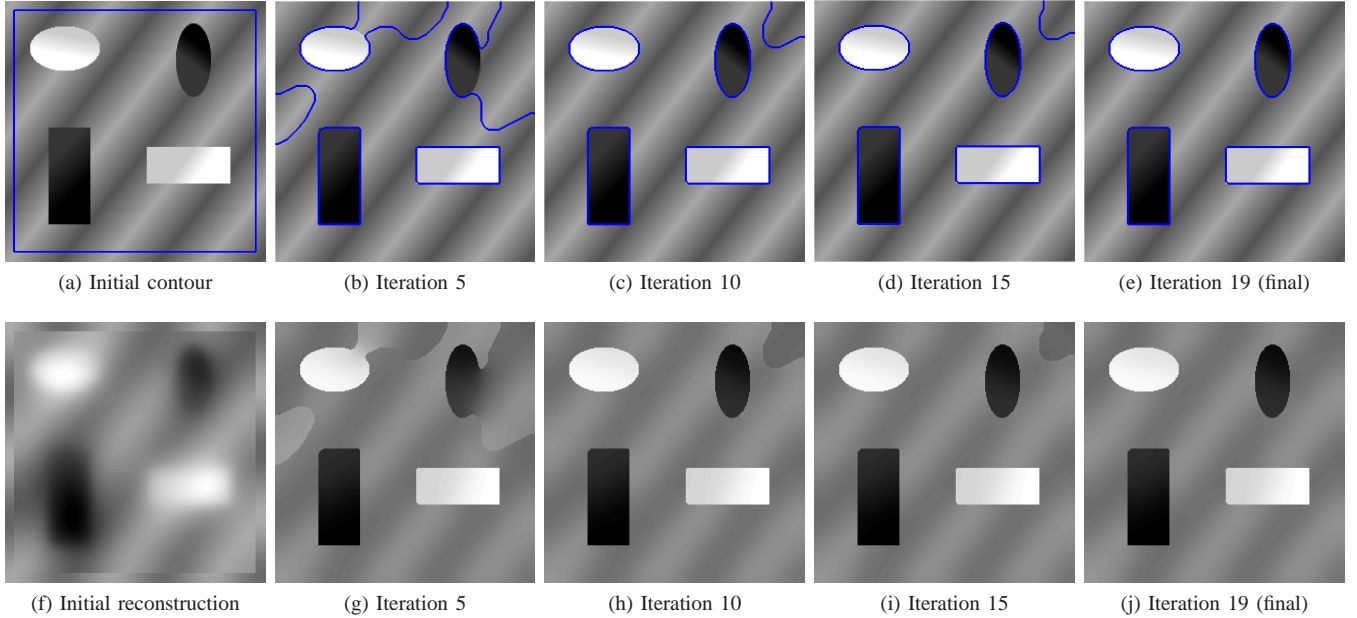


Fig. 2: Demonstration of correctness that our combinatorial formulation and optimization of the Mumford-Shah functional correctly recovers the contours of a piecewise smooth image and reconstructs the piecewise smooth approximation accordingly. Top: Contour evolution from initialization to stabilization. Blue contours indicate contour location. Bottom: Corresponding reconstruction of the piecewise smooth estimate of the image, given a contour.

the computation of the function estimation step, the drastic reduction in the number of iterations observed with our graph formulation (see Section III) more than compensates for these additional computations.

- 2) The same steps apply to solving the Mumford-Shah functional in any dimension — The only difference between dimensions is in the underlying lattice employed (e.g., a 6-connected lattice compared to a 4-connected lattice).
- 3) Although our primary focus is on the contour location (e.g., a segmentation), it is important to note that a piecewise smooth approximation to the image is also obtained via the solution to  $f$  and  $g$ . Consequently, our algorithm applies equally to image/signal denoising problems.
- 4) Due to the generality of the formulation, the same algorithm can be applied to the segmentation, clustering, denoising, etc. of any data defined on a graph. Examples of the application of this algorithm to the more general setting is given in Section IV-B.
- 5) Since global optima are obtained at each step, no implementation parameters (e.g., step size) are necessary in our graph formulation.
- 6) Since global optima are obtained at each step (relative to the choice of extension functions), the energy obtained after each minimization can never increase. In practice we have always observed rapid convergence.
- 7) In the context of image processing in which the graph is a lattice embedded into  $\mathbb{R}^2$  or  $\mathbb{R}^3$ , combinatorial optimization does have some drawbacks with respect to the level set methods. Specifically, since the regions are parameterized by binary sets of nodes, there is

no notion of sub-pixel accuracy available. Additionally, approximation of a Euclidean length of the contour via a large number of edges requires significant memory, especially in 3D.

As a demonstration of correctness, Figure 2 demonstrates that our combinatorial formulation and optimization appropriately segments a piecewise smooth image of size  $256 \times 256$  pixels.

### III. RESULTS

The positives and negatives of Mumford-Shah segmentation and reconstruction have been well-discussed in the literature. Our reformulation of the Mumford-Shah functional on a graph is intended to permit usage of the arsenal of combinatorial optimization methods to minimize the Mumford-Shah energy more quickly and to find lower-energy solutions. Consequently, our experiments focus on answering the following questions about the relative merits of traditional gradient descent-based level set implementations of the Mumford-Shah energy and the combinatorial optimization applied to our graph formulation of the Mumford-Shah functional:

- 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 answer the first three questions, we begin with a toy image of a black square on a white background. Such a trivial



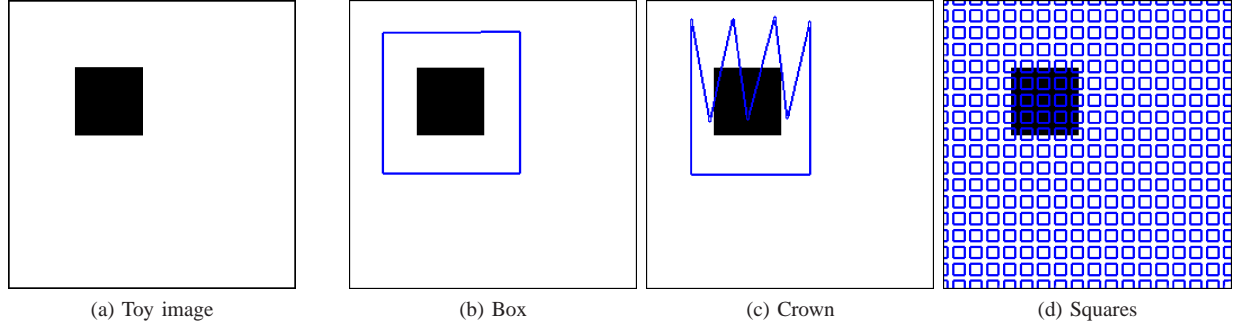


Fig. 3: Toy image to compare the speed of traditional contour evolution implementation with proposed combinatorial optimization of the Mumford-Shah functional presented in this paper. The contours (blue) indicate three different initializations used to generate results presented in Table I.

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 gradient descent on level sets) and our new graph formulation of the Mumford-Shah energy. The last question regarding relative energy of the solutions is addressed by running both the graph and level set optimizations on a series of 50 natural images, comparing their resulting energies.

We compared the combinatorial optimization of our graph formulation method with an efficient gradient descent-based narrow-band level set implementation of the continuous formulation similar to the one presented in [55] although the original piecewise-smooth level set implementation was presented in [20]. 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. In [55], the curve evolution is written as an initial value problem on a locally evolving geometric contour, and not as an initial value problem on a smoothed Heaviside function as in [20]. In our implementation of the level set method however, we do not initialize with the piecewise constant solution as done in [55]. The method employed alternating optimizations of the contour evolution and of the smooth functions as in the graph method and as has been used in all Mumford-Shah implementations 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 [49]. 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 [44] 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 (6) by using the weighted incidence matrix of the graph corresponding to the construction of Boykov and Kolmogorov [8] with an approximation to the Euclidean distance represented by a neighborhood connected with a distance of two pixels.

#### A. 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 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 3. For each of these initializations, we measured the number of iterations required to converge the level set via gradient descent 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. The time per iteration and time for a boundary update may not be multiplied directly to produce a total execution time since each method additionally requires an image reconstruction update. However, this function update is computed in roughly the same manner (assuming a finite differences discretization in the level set method) for both algorithms, although our combinatorial method requires twice as much time for the function update since the level set method does not require that the foreground/background functions be extended. If the function computations are much larger than the boundary updates then the relative execution times will be close to the ratio of iterations. However, if the function computations are much less than the boundary updates, then the relative execution times will be close to the ratio of the number of iterations for each method multiplied by the boundary update time for each method. Since the linear system



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

TABLE I: 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.

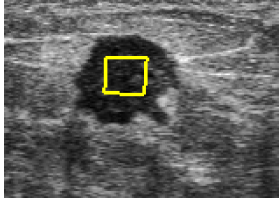

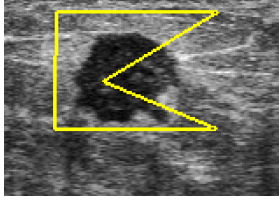
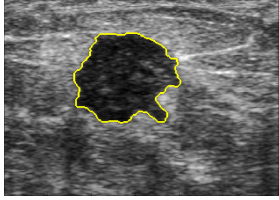
	Initialization 1				Initialization 2		
		iters.	sec./iter.			iters.	sec./iter.
	Level Set	312	0.0061		Level Set	1523	0.0073
	Graph	4	0.0616		Graph	7	0.1176
	Initialization 3				Final segmentation		
		iters.	sec./iter.				
	Level Set	1920	0.0101				
	Graph	5	0.1187				

Fig. 4: 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.

solver is not different for these two methods (on a Cartesian grid), we simply report both the total number of iterations and the boundary update time.

In this experiment, the parameters in the energy functional were chosen to favor the level set method as much as possible, as determined via manual adjustment. Table I 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 the initializations in these cases are very distant from the final contour.

These experiments suggest that the combinatorial optimization of the Mumford-Shah functional produces the correct solution much faster than the traditional level set optimization,

regardless of the resolution or contour initialization. We remind the reader that the parameters were chosen to favor the level set method. Choosing the parameters to favor our proposed graph method would have resulted in a stronger disparity in favor of the graph method. In the next section, we detail the results of an experiment intended to measure the robustness of the two approaches to parameter choice.

A third experiment was performed on a real ultrasound image in the same manner as the first. An initialization was 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 4 and correspond well with the results from our synthetic experiment. Once again, the parameters of the terms in the Mumford-Shah energy were chosen to best favor the level set method and both methods converged to roughly the same contour.

### B. Parameter robustness

The choices of the term parameters in (1) can make drastic differences in the optimal contour and reconstruction produced by minimizing the Mumford-Shah functional. 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

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

TABLE II: Comparison of robustness to the three term parameters in (1). Using the  $(128 \times 128)$  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 combinatorial optimizations of the MSF. A randomly generated set of parameters was rejected and re-run if the target square was not the minimum of the MSF. In all cases, combinatorial optimization produced the target square in two iterations, regardless of parameters. Note that the number of iterations reported for the level set method in Table I was much less than the averages reported here due to the fact that all of the results reported in I used parameters that were hand-selected to favor the level set convergence.

experiment, we examine the robustness of both the level set and graph formulations of the Mumford-Shah energy 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 3. For this experiment, we used the most simple, “box”, initialization of Figure 3 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 Mumford-Shah energy. 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.

The results of this experiment are displayed in Table II. 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 I was much less than the average reported in Table II due to the fact that all of the results reported in Table I used parameters that were hand-selected to favor the level set convergence.

### C. Energy minimization comparison

Beyond speed, our purpose in introducing combinatorial optimization techniques for solving the Mumford-Shah functional is to produce solutions with a lower energy than the solutions obtained by conventional gradient descent-based 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 to empirically compare the Mumford-Shah energies obtained by the final solutions.

This experiment was performed on 50 different natural images, of which 11 were from the Berkeley image database, 12 were miscellaneous photographs, and 27 were various medical images (12 ultrasound, 4 CT, 5 MRI, 3 digital

microscopy, 1 fMRI, and 2 angiography images). For each image, initializations and parameters were selected to produce a contour (for at least one algorithm) that was semantically meaningful. Cluster plots of the final graph MS energy vs. the final level set MS energy are shown in Figure 5 for these 50 images. In most cases, optimization of our graph formulation of the Mumford-Shah functional produced solutions with an equal or lower energy and sometimes produced solutions with dramatically lower energy. In a few cases, the level set formulation produced slightly lower energy (shown in Figure 6.)

The issue of computing the Mumford-Shah energy for a given solution is complicated by the Euclidean boundary length term. The measurement of Euclidean length for a pixelated segmentation was addressed separately by Boykov/Kolmogorov and Chan/Vese. Although it was shown that in the limit, both of these measures produce the true Euclidean boundary length, for a given finite segmentation they may differ in their estimates. Since our combinatorial formulation explicitly employs the Boykov/Kolmogorov measure and the level set implementation implicitly employs the Chan/Vese measure, we were concerned that this difference in Euclidean length measurements might bias the comparative solutions produced by each algorithm. Consequently, we computed the Mumford-Shah energy using both measures to see if any discrepancy would affect our valuation of which optimization method produced a lower energy. A scatter plot showing the relative Mumford-Shah energies of the solution using the Chan/Vese measurement is given in Figure 5a while the energies of the segmentations are recomputed using the Boykov/Kolmogorov measurement of boundary length and are displayed in Figure 5b. However, as can be seen from the figures, the difference between these two measures is minimal and does not change the conclusion that the combinatorial optimization produces consistently lower energies than the level set optimization.

In Fig. 6 we show select images whose segmentations differ in semantically meaningful ways. Note the more global nature of the segmentations produced by the combinatorial optimization method. This is a direct result of allowing the graph method to make globally optimal cuts at each iteration, while the level set method restricts the contour to progressing by gradient descent. We also show two of the four images where the level set optimization produced a segmentation with lower energy than the graph based optimization 6j and 6k. We believe that the lower-energy solution found by the level-set method in these instances could have also been obtained by the

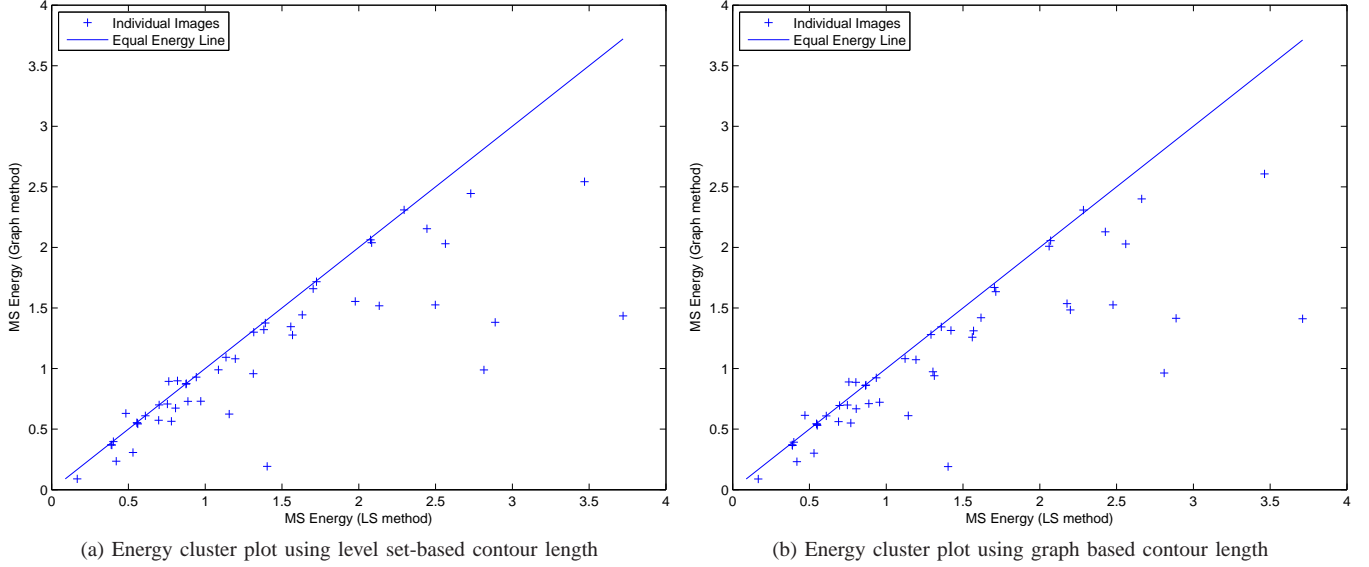


Fig. 5: Cluster plots of final Mumford-Shah function energies comparing energy minimization methods over 50 images. Each “+” represents an individual image showing the energy of graph solution vs. the energy of level set solution. Note that the only difference between the two plots is that **5a** is measured with the approximation to Euclidean boundary length provided by Chan and Vese [19] and **5b** is measured with the approximation to Euclidean boundary length provided by Boykov and Kolmogorov [8].

graph method in this case if we had made a different choice of extension functions.

#### IV. ADDITIONAL CAPABILITIES OF THE FORMULATION

In this section, we address two desirable aspects of our combinatorial formulation that are not available with a traditional formulation and level set optimization. First, we demonstrate the ability of a graph formulation to exhibit non-local movement, since the optimal contour is found at each iteration, there is no limitation that the contour move to a nearby location. Consequently, the segmentation may be found faster (by taking larger steps) but, more importantly, the formulation also permits the contour to “jump over” spatial obstacles that would prevent the contour from finding a low energy. Second, we give an example of how our combinatorial formulation permits the application of Mumford-Shah methods to data associated with an abstract graph. Specifically, we use the Mumford-Shah functional to perform filtering (piecewise reconstruction) of image data defined on a space-variant data structure intended to model the sampling scheme of the human retina and on a point clustering problem.

##### A. Non-local movement

A key advantage of the contour optimization in our graph reformulation of the Mumford-Shah energy is that it enables movement to the optimal location at each iteration. For this reason, our method is able to move to arbitrary image locations as predicted by the solution to (13) depending on the current estimate of the piecewise smooth reconstructions. The motion of the contour is thus *not limited to local movements* as are traditional optimizations of the contour by gradient descent.

Figure 7 illustrates three situations that are able to be segmented correctly by the proposed combinatorial optimization of the Mumford-Shah functional, 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 shown in Figure 7 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 [19] were able to determine inner boundaries of objects. The ability to segment this inner boundary was due to the mollified Heaviside function that was used to approximate a region indicator function. Indeed, one could argue that there always exists some heavy-tailed mollified Delta function that could achieve the segmentation of the inner ring for an annulus of fixed thickness. However, such heavy-tailed Delta functions ultimately sacrifice segmentation accuracy and thus, one must modify the mollification kernel during the optimization if high accuracy is desired. Such an implementation trick is completely obviated by using our proposed method which works for annuli of arbitrary thickness with no such temporary sacrifice in accuracy.

The work of [13] shows the ability to naturally attain such inner boundaries due to their method of total variation optimization using a modified Mumford-Shah functional. Some level of non-local movement in solving the Mumford-Shah functional with level set methods have been achieved in [57] using additive operator splitting [58], however they only illustrated the technique for the piecewise constant case. Specifically, they did not present the idea of extending the

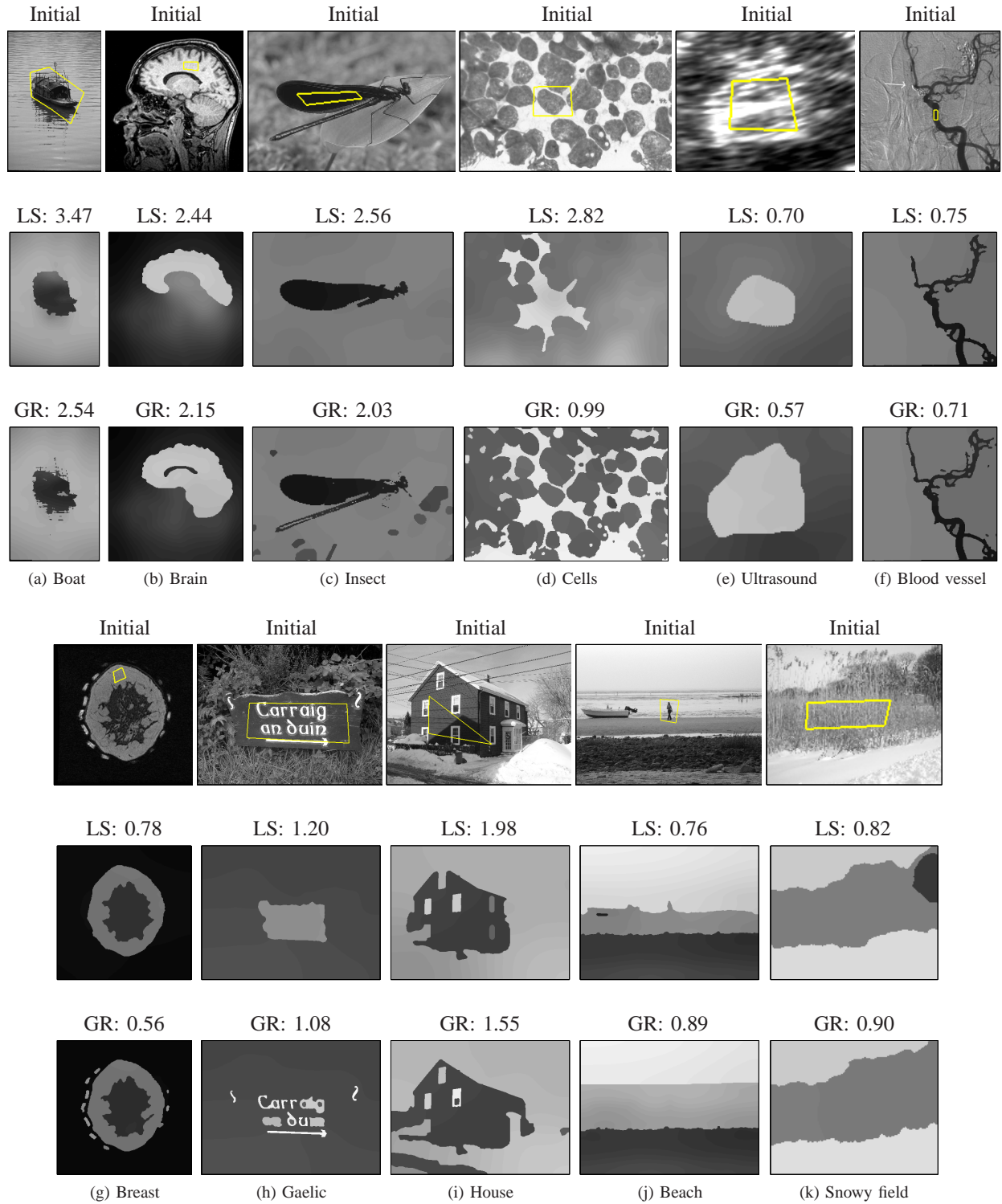


Fig. 6: Select images that show substantial or interesting differences between the traditional level set optimization and our proposed combinatorial optimization of the piecewise smooth Mumford-Shah functional. Top Row: Original image with contour initialization, Middle Row: Piecewise smooth reconstruction from the level set method is shown in middle row, Bottom Row: Piecewise smooth reconstruction from the graph method is shown on bottom row. The corresponding energies are displayed above each reconstruction. Most images yield lower energies with the graph method, but images 6j and 6k are two examples that exhibited slightly lower energy with the LS method.



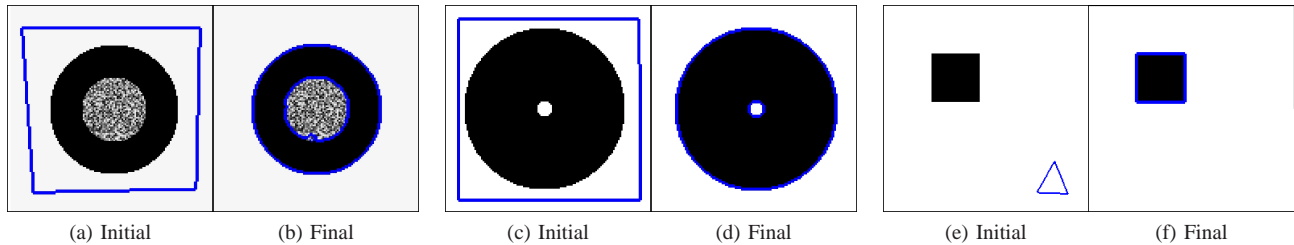


Fig. 7: Non-local movement: Since the contour optimization step in our graph formulation of the Mumford-Shah energy is not performed by gradient descent, the final contour is permitted to jump to a location distant from the initial contour location. While this effect is sometimes achieved in the literature 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.

approximation functions as in the proposed method. Additionally, the work of Hintermüller has the ability for non-local movement [36].

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

It should be pointed out that some applications may indeed prefer local evolution of a contour and that global minimization of the functional is not always desirable. In this work, we are interested only in the minimization of the energy. Due to the minimization procedure, the proposed combinatorial formulation progresses the contour in ways that are optimal at each iteration. If non-local movement is considered undesirable for some segmentation tasks (e.g., perhaps the outside of the annulus is the target segmentation), it is possible to restrict the movement of the contour to a fixed distance from the present contour. This restriction may be easily accomplished by fixing the  $r$  values in the contour optimization that are more than a predetermined distance,  $\rho$ , from the present contour. Using this device, the contour would not change its location past  $\rho$  pixels of its location at the previous iteration.

The term “non-local movement” is used in this paper to represent the ability of the contour to make arbitrarily large jumps in space. This goal is different from that in the recent work on Sobolev Active Contours [52], in which the authors desired to change the notion of what it means to be a “local” deformation on the space of curves. Thus, the authors in [52] still use gradient descent, only with a different notion of how the gradient is defined. The Sobolev Inner product as proposed in [52] does not allow for the non-local movement exhibited in Fig. 7. Simply stated, our method is not a gradient descent approach, and seeks to minimize the energy using an entirely different scheme than gradient descent.

### B. Mumford-Shah on a general graph

Image processing techniques are typically applied to Cartesian-sampled images in 2D or 3D with a local neighborhood structure. However, many other kinds of data exist in which the same tasks of clustering (i.e., segmentation) or

filtering are important. In order to apply our combinatorial formulation of the Mumford-Shah function, all that is necessary is to have a series of finite data points for which it is possible to define a neighborhood relationship (edge set) and such that each data point has an associated value (or vector of values). Therefore, by formulating the Mumford-Shah functional on a general graph, our algorithm has straightforward application to clustering points in a feature space or mesh smoothing and segmentation (see [31] for more examples from this point of view).

In this section, we choose two different application areas for demonstrating our general formulation of the Mumford-Shah energy. The first application is to a piecewise smooth reconstruction of a non-uniformly sampled image and the second application is to point clustering.

Non-uniformly sampled images may occur in applications due to the image acquisition device (e.g., ultrasound). Additionally, such sampling of images has also been heavily studied because of the connection to the sampling of biological retinas [47], [3], [31]. These biological retinas may exhibit a wide range of sampling [37], but the primate sampling structure may be described as consisting of a foveal pit of high density sampling in the center of visual space with an exponential reduction in sampling toward the periphery. Using the idealized sampling of the human retina provided in the free Graph Analysis Toolbox [32], we imported a standard Cartesian image to this foveal structure and optimized the Mumford-Shah functional to produce a piecewise-smooth reconstruction of the data. The results are displayed in Figure 8. Note that the only change necessary to employ this foveal structure was to change the node and edge set of the graph to match the non-uniformly sampled image (given by the Graph Analysis Toolbox, in this case), which then provides a new  $A$  operator. Given this  $A$  operator, the procedure in Section II-E may be applied exactly as described to produce the minimum of the combinatorial formulation of the Mumford-Shah energy.

Our combinatorial formulation of the Mumford-Shah energy applies not only to image data, but rather to any function assigning values to the graph nodes (a 0-cochain). Consequently, we may take the coordinates in feature space of a clustering problem and directly apply our algorithm to perform the clustering. In Figure 9 we generated two overlapping point clouds with 2D features ( $x$  and  $y$  coordinates) and connected the nodes with a Delaunay triangulation. We may treat

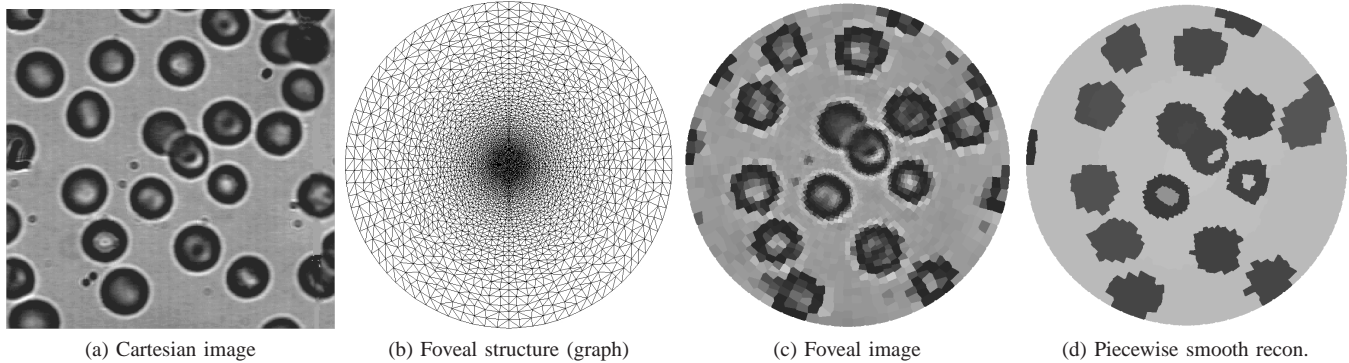


Fig. 8: Formulating the Mumford-Shah on a general graph permits usage on nonuniformly sampled images that was not available with previous formulations/optimizations. Here, we present the piecewise smooth reconstruction of an image that is resampled in accordance with the foveal sampling exhibited by the human retina [47], [31].

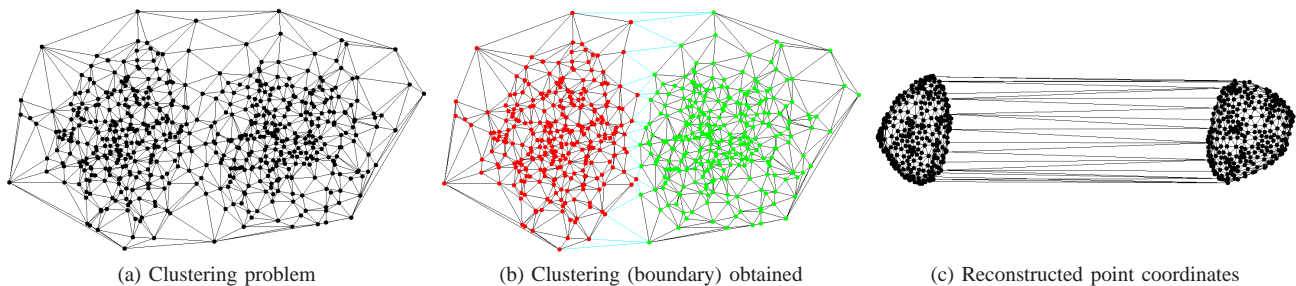


Fig. 9: Our graph formulation of the Mumford-Shah energy allows nonstandard applications such as clustering or mesh smoothing. In this example, the  $x$  and  $y$  coordinates of each point were used in place of the image intensities as the function to be reconstructed. The clustering obtained is shown in (b) where points in different clusters are colored red or green and the edges spanning the boundary are colored light blue. In (c) the reconstructed (smoothed) coordinates for each point are shown.

the foreground/background coordinates as independent scalar functions, which may be thought of as adding a second set of data/smoothness terms into (7). The clustering (boundary) and reconstructed coordinates for foreground and background are shown in Figure 9. Note that this problem is intended to demonstrate the generality of our formulation rather than the difficulty of this particular clustering problem.

## V. CONCLUSION

In this work, we began by reformulating the classical Mumford-Shah energy functional in terms of analogous differential operators on graphs. With this reformulation, we are able to apply the established arsenal of combinatorial optimization techniques for producing reconstruction and contour updates.

Our experiments indicate a dramatic improvement of our graph-formulated optimization over traditional gradient descent approaches for contour evolution. This improvement is in terms of speed, robustness to initialization, robustness to parameter settings and in the production of a solution representing a lower Mumford-Shah energy. Additionally, we employ a global combinatorial optimization technique that is not based on gradient descent to solve our graph formulation of the Mumford-Shah functional, which permits non-local movement of the contour to find low energy solutions.

Although our measured comparisons with the traditional level set method were performed in 2D, our formulation applies equally to 3D lattices. Additionally, in strong contrast to traditional methods of formulation and optimization of the Mumford-Shah functional, we demonstrated that our method permits application on arbitrary graph structures, including non-uniformly sampled images. Future work includes:

- 1) Introduction of multiple labels in the contour computation to address triple points and other junctions for which hierarchical methods and multiphase level sets have been employed [55], [41], [56].
- 2) Exploration of other choices for extending the  $f$  and  $g$  reconstructions beyond their region of calculation, other than the Laplace equation employed here. It is the view of the authors that the existence of global optimization methods for the contour location highlight the importance of the extension choice, which has not been addressed as an issue of fundamental importance in the literature.

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 that a graph-based formulation also permits application of the same techniques to more abstract domains, such as data clustering [23], mesh processing [54] and space-variant vision [47].

#### ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments and our colleagues Fabian Mörchen, Vladimir Kolmogorov, Mikael Rousson, and Gözde Ünal for useful discussions.

#### REFERENCES

- [1] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of Bounded Variation and Free Discontinuity Problems*. Oxford University Press, 2000. 3
- [2] J. An, M. Rousson, and C. Xu.  $\Gamma$ -convergence approximation to piecewise smooth medical image segmentation. In *Proc. of MICCAI*, pages 495–502. Springer LNCS 4792, 2007. 1
- [3] G. Baratoiff, C. Toepfer, and H. Neumann. Combined space-variant maps for optical flow based navigation. *Biological Cybernetics*, 83:199–209, 2000. 13
- [4] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *Proc. of ECCV*, pages 428–441, May 2004. 6
- [5] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987. 3
- [6] S. Bougleux, A. Elmoataz, and M. Melkemi. Discrete regularization on weighted graphs for image and mesh filtering. In *SSVM*, volume 4485 of LNCS, pages 128–139. Springer-Verlag, 2007. 2
- [7] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proc. of ICCV 2001*, pages 105–112, 2001. 6
- [8] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Proc. ICCV*, volume 1, pages 26–33, October 2003. 3, 8, 11
- [9] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on Pat. Anal. and Mat. Intel.*, 26(9):1124–1137, Sept. 2004. 4
- [10] Y. Boykov, V. Kolmogorov, D. Cremers, and A. Delong. An integral solution to surface evolution PDEs via geo-cuts. In *Proc. of ECCV*, volume 3953 of LNCS, pages 409–422, May 2006. 2
- [11] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pat. Anal. and Mat. Intel.*, 23(11):1222–1239, November 2001. 6
- [12] F. H. Branin, Jr. The algebraic-topological basis for network analogies and the vector calculus. In *Generalized Networks, Proceedings*, pages 453–491, Brooklyn, N.Y., April 1966. 3
- [13] X. Bresson, S. Esedoglu, P. Vanderghenst, J. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *J. Mathematical Imaging and Vision*, 28(2):151–167, 2007. 1, 2, 5, 6, 11
- [14] T. Brox and D. Cremers. On the statistical interpretation of the piecewise smooth Mumford-Shah functional. In F. Sgallari, editor, *Proc. of SSVN*, pages 203–213. Springer LNCS 4485, 2007. 1
- [15] T. Brox and D. Cremers. On local region models and a statistical interpretation of the piecewise smooth Mumford-Shah functional. *IJCV Online*, 2008. 2, 5
- [16] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation*, 4(2):490–530, 2005. 2
- [17] A. Chambolle. Image segmentation by variational methods: Mumford and Shah function and the discrete approximations. *SIAM J. of Appl. Math.*, 55(3):827–863, June 1995. 2
- [18] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1–2):89–97, 2004. 1
- [19] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. on Image Proc.*, 10(2):266–277, 2001. 1, 11
- [20] T. Chan and L. Vese. A level set algorithm for minimizing the Mumford-Shah functional in image processing. In *Workshop on VLSP*, pages 161–168. IEEE, 2001. 1, 4, 8
- [21] T. F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of denoising and segmentation models. *SIAM J. Appl. Math.*, 66:1632–1648, 2006. 1, 6
- [22] Y. Chen, W. Zhao, and Z. Wang. Segmentation of serial CT images based on an improved Mumford-Shah model. In T. Z. et al., editor, *MIPPR 2007: Automatic target recognition and image analysis*, volume 6786. SPIE, Nov. 2007. 1
- [23] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *PNAS*, 102(21):7426–7431, May 2005. 2, 15
- [24] J. Darbon. A note on the discrete binary Mumford-Shah model. In *Proc. of CV/CGCT*, volume 4418 of LNCS, pages 283–294. Springer-Verlag, 2007. 2
- [25] M. Droske and W. Ring. A Mumford-Shah level-set approach for geometric image registration. *SIAM J. of App. Math.*, 66(6):2127–2148, 2006. 1
- [26] M. Droske, W. Ring, and M. Rumpf. Mumford-Shah based registration: a comparison of a level set and a phase field approach. *Computing and Visualization in Science*, 2008. 1
- [27] N. El-Zehiry, S. Xu, P. Sahoo, and A. Elmaghraby. Graph cut optimization for the Mumford-Shah model. In *Proc. of VIIP*, 2007. 2
- [28] A. Elmoataz, O. Lzorzay, and S. Bougleux. Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. *IEEE Trans. on Image Processing*, 17(7):1047–1060, 2008. 2
- [29] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the Bayesian restoration of images. *IEEE Trans. on Pat. Anal. and Mat. Intel.*, 6(6):721–741, November 1984. 2, 3
- [30] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996. 5
- [31] L. Grady. *Space-Variant Computer Vision: A Graph-Theoretic Approach*. PhD thesis, Boston University, Boston, MA, 2004. 5, 13, 14
- [32] L. Grady and E. L. Schwartz. The Graph Analysis Toolbox: Image processing on arbitrary graphs. Technical Report TR-03-021, Boston University, Boston, MA, Aug. 2003. 13
- [33] L. Grady and E. L. Schwartz. Faster graph-theoretic image processing via small-world and quadtree topologies. In *Proc. of CVPR*, volume 2, pages 360–365, 2004. 2
- [34] D. Greig, B. Porteous, and A. Seheult. Exact maximum *a posteriori* estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989. 1
- [35] J. Han, B. Berkels, M. Droske, J. Hornegger, M. Rumpf, C. Schaller, J. Scorzin, and H. Urbach. Mumford-Shah model for one-to-one edge matching. *IEEE Trans. on Image Process.*, 16(11):2720–2732, Nov. 2007. 1
- [36] M. Hintermüller and W. Ring. An inexact Newton-CG-type active contour approach for the minimization of the Mumford-Shah functional. *J. Math. Imaging Vis.*, 20(1–2):19–42, 2004. 1, 13
- [37] A. Hughes. The topography of vision in mammals of contrasting life style: Comparative optics and retinal organization. In F. Crescitelli, editor, *The Visual system in vertebrates*, chapter 11, pages 613–756. Springer-Verlag, Berlin, New York, 1977. 13
- [38] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. on Pat. Anal. and Mat. Intel.*, 26(2):147–159, Feb. 2004. 1, 4
- [39] K. Li, X. Wu, D. Z. Chen, and M. Sonka. Optimal surface segmentation in volumetric images — A graph-theoretic approach. *IEEE Trans. on Pat. Anal. and Mat. Intel.*, 28(1):119–134, Jan. 2006. 2
- [40] C. Mattiussi. The finite volume, finite element and finite difference methods as numerical methods for physical field problems. In *AIEP*, pages 1–146. Academic Press Inc., April 2000. 3
- [41] B. Merriman, J. Bence, and S. Osher. Motion of multiple junctions: A level set approach. *J. Comput. Phys.*, 112(2):334–363, 1994. 14
- [42] D. Mumford. The statistical description of visual signals. In K. Kirshgassner, O. Mahrenholtz, and R. Mennicken, editors, *Proc. of ICIAM 95*. Akademie Verlag, 1995. 2
- [43] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure and Appl. Math.*, 42:577–685, 1989. 1, 3
- [44] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988. 8
- [45] J. Piovano, M. Rousson, and T. Papadopoulos. Efficient segmentation of piecewise smooth images. In F. Sgallari, editor, *Proc. of SSVN*, pages 709–720. Springer LNCS 4485, 2007. 1
- [46] R. Ramlau and W. Ring. A Mumford-Shah level-set approach for the inversion and segmentation of X-ray tomography data. *J. of Computational Physics*, 221(2):539–557, Feb. 2007. 1
- [47] A. S. Rojer and E. L. Schwartz. Design considerations for a space-variant visual sensor with complex-logarithmic geometry. In *Proc. ICPR*, volume 2, pages 278–285. IEEE Computer Society Press, 1990. 2, 13

- 14, 15
- [48] C. Rother, V. Kolmogorov, and A. Blake. “GrabCut” — Interactive foreground extraction using iterated graph cuts. In *Proc. of SIGGRAPH*, volume 23, pages 309–314. ACM, 2004. 6
  - [49] J. Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry*. Cambridge University Press, 1999. 8
  - [50] A. K. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *Proc. of ICCV*. IEEE, Oct. 2007. 1
  - [51] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, 1986. 3
  - [52] G. Sundaramoorthi, A. Yezzi, and A. Mennucci. Sobolev active contours. *Int. J. of Comp. Vis.*, pages 345–366, 2007. 13
  - [53] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields. In *Proc. of ECCV*, volume 2, pages 19–26, May 2006. 1
  - [54] G. Taubin. A signal processing approach to fair surface design. In R. Cook, editor, *Computer Graphics Proceedings. Special Interest Group in Computer Graphics (SIGGRAPH) 95*, pages 351–358, Los Angeles, CA, Aug. 1995. ACM, ACM. 2, 15
  - [55] A. Tsai, A. Yezzi, and A. Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Trans. on Image Proc.*, 10(8):1169–1186, 2001. 1, 3, 4, 8, 14
  - [56] L. Vese and T. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. of Comp. Vis.*, 50(3):271–293, 2002. 2, 5, 14
  - [57] Z. Wang, X. Yang, and P. Shi. Solving Mumford-Shah model equation by AOS algorithm. In *Int. Conf. on Sig. Proc.*, volume 1, pages 740–743. IEEE, 2002. 11
  - [58] J. Weickert, B. M. Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. on Image Proc.*, 7(3):398–410, March 1998. 11
  - [59] J. Xin and S. Nie. Urine sediment image segmentation based on level set and Mumford-Shah model. In *Bioinformatics and Biomedical Engineering*, pages 1028–1030, July 2007. 1
  - [60] X. Zeng, W. Chen, and Q. Peng. Efficiently solving the piecewise constant Mumford-Shah model using graph cuts. Technical report, Zhejiang University, 2006. 2
  - [61] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. of Comp. Phys.*, 127:179–195, 1996. 2, 5