Threshold Dynamics for the Piecewise Constant Mumford-Shah Functional

Selim Esedoḡlu*
Mathematics Department
University of California Los Angeles

Yen-Hsi Richard Tsai†
Mathematics Department
University of Texas at Austin

October 22, 2004

Abstract

We propose an efficient algorithm for minimizing the piecewise constant Mumford-Shah functional of image segmentation. It is based on the threshold dynamics of Merriman, Bence, and Osher for evolving an interface by its mean curvature. We show that a very fast minimization can be achieved by alternating the solution of a linear parabolic partial differential equation and simple thresholding.

1 Introduction

Image segmentation is one of the fundamental tasks of computer vision. It forms a crucial preliminary step for subsequent object recognition and interpretation[21]. Its goal is to partition a given image into regions that contain distinct objects. The most common form of segmentation is based on the assumption that distinct objects in an image have different and approximately constant (or slowly varying) colors (or brightnesses in the case of monochrome imagery). A natural approach is therefore to try to decompose an image domain into approximately homogeneous regions that are separated by sharp changes in image features (such as colors or brightness). The boundaries of the homogeneous regions are called "edges"; they correspond to places in an image where different objects meet.

*Mathematics Department, UCLA. Los Angeles, CA 90095. **Email:** esedoglu@math.ucla.edu.
Research supported in part by NSF grant DMS-0410085, NSF grant DMS-9973341, NIH grant P20 MH65166, ONR grant N00014-03-1-0888, and ONR grant N00014-02-1-0720.

†Mathematics Department, University of Texas at Austin, TX 78712. **Email:** ytsai@math.utexas.edu.
Research supported in part by the National Science Foundation under agreement No. DMS-0111298

Variational models for image segmentation have had great success. In this approach, the correct segmentation for an image is exhibited as the minimizer of an appropriately chosen energy. One of the most successful and pioneering models that adopt this approach is the model of Mumford and Shah [22]. It poses the segmentation problem as that of a best approximation: It looks for the best "cartoon-like" (i.e. piecewise smooth) approximation of minimal complexity for a given image. Once such an approximation is constructed, the homogeneous regions and their boundaries become obvious.

In this paper we propose an extremely efficient PDE based algorithm for minimizing a version of the Mumford-Shah segmentation functional. The algorithm stems from the work of Merriman, Bence, and Osher (MBO) on diffusion generated motion by curvature [17, 18]. Similar to the MBO algorithm, our algorithm works by alternating the solution of a *linear* diffusion equation with thresholding. Thus, unlike other PDE based methods of minimizing the Mumford-Shah functional that involve complicated (sometimes degenerate) equations, all steps in our algorithm are amenable to fast numerical solution by well-established techniques.

This paper is organized as follows: In Section 2 we describe the Mumford-Shah model along with several of its variants, and introduce notation. Section 3 describes some recent previous work on fast solution of variational segmentation models; in particular, the work of Gibou and Fedkiw, and Song and Chan are recalled. Section 4 is devoted to recalling the Chan-Vese algorithm, which is a level set based method of minimizing certain variants of the Mumford-Shah model. Section 5 recalls the Merriman, Bence, Osher threshold dynamics for motion by mean curvature; this forms the basis on which we build our proposed algorithm for minimizing Mumford-Shah functional in Section 6. Section 7 describes some variants and extensions of the algorithms proposed in Section 6. Section 8 presents results of numerical experiments, and Section 9 provides a discussion indicating further improvements.

2 Mumford-Shah Model

The full Mumford-Shah model is a variational problem for approximating a given image by a piecewise smooth image of minimal complexity. Let $D \subset \mathbf{R}^N$ be a bounded domain with Lipschitz boundary, modeling the image domain (for example, the computer screen). Let $f(x) : D \to [0, 1]$ represent a grayscale image; we merely assume that $f(x)$ is a bounded measurable function. To find the segmentation of $f(x)$, Mumford and Shah proposed in [22] carrying out the following

minimization:

$$\min_{\substack{u:D\to\mathbf{R}\\K\subseteq D}} \left\{ MS(u,K) := \int_{D\setminus K} |\nabla u|^2\,dx + \mu\text{Length}(K) + \lambda \int_D (u-f)^2\,dx \right\} \tag{1}$$

Here, K is to be a closed subset of D given by the union of a finite number of curves. It represents the set of "edges" (i.e. boundaries of homogeneous regions) in the image f. The function u is the piecewise smooth approximation to f. Because of the Dirichlet integral taken over the set $D \setminus K$ in the energy, u is forced to be smooth in each connected component of $D \setminus K$. However, it is allowed have jumps across the curves that make up K.

The full Mumford-Shah functional (1) poses a formidable optimization problem. A number of algorithms have been proposed for its solution. For example, the work of Ambrosio and Tortorelli [1] shows how to approximate (1), in the sense of Gamma convergence [7], with a class of energies that are much more tractable numerically and that can be subsequently minimized via gradient descent; see [13] for numerical results. This is one of the best known ways to tackle (1) in its full generality.

In many vision applications, solving (1) in its full generality is an overkill (However, there are also many situations in which it is not general enough). For example, one might know that the objects in the scenes of interest are not only of smoothly varying, but actually approximately constant color (or grayscale intensity). Or, one might know in advance the maximum number of objects in the scene (and hence the maximum number of regions in the segmentation). An example of such an application is medical imaging, where one might for instance be interested in segmenting brain MRI images into background, gray matter, and white matter. Foreground – background segmentation is also a simple example in which one is interested in segmentations that only have two regions. In such scenarios it makes sense to work with a simplified version of (1) that is easier to minimize.

A very useful simplification of (1) is to restrict the minimization to functions (i.e. segmentations) that take a finite number of values. The resulting model is commonly referred to as the *piecewise constant Mumford-Shah model*. In this paper we will be concerned especially with the case where the solution takes only two (unknown) values, and hence is of the form $u(x) = c_1 \mathbf{1}_\Sigma(x) + c_2 \mathbf{1}_{D\setminus\Sigma}(x)$,

where $\Sigma \subseteq D$. In this case, (1) reduces to the following minimization problem:

$$\min_{\substack{\Sigma \subseteq D \\ c_1, c_2 \in \mathbf{R}}} \Big\{ E(\Sigma, c_1, c_2) := \text{Per}(\Sigma; D) + \lambda \int_{\Sigma} (c_1 - f)^2 \, dx + \lambda \int_{D \setminus \Sigma} (c_2 - f)^2 \, dx \Big\}. \quad (2)$$

Model (2) is known as the *two-phase, piecewise constant Mumford-Shah model*. Our algorithm for minimizing this functional efficiently is presented in Section 6. If (1) is restricted instead to segmentations that take at most four values, an approximate version of the resulting functional can be written as

$$\begin{aligned} \min_{\substack{\Sigma_1, \Sigma_2 \subset D \\ c_1, \ldots, c_4 \in \mathbf{R}}} \Big\{ & E(\Sigma_1, \Sigma_2; c_1, \ldots, c_2) := \text{Per}(\Sigma_1; D) + \text{Per}(\Sigma_2, D) \\ & + \lambda \int_{\Sigma_1 \cap \Sigma_2} (c_1 - f)^2 \, dx + \lambda \int_{\Sigma_1 \cap \Sigma_2^c} (c_2 - f)^2 \, dx \\ & + \lambda \int_{\Sigma_1^c \cap \Sigma_2} (c_3 - f)^2 \, dx + \lambda \int_{\Sigma_1^c \cap \Sigma_2^c} (c_4 - f)^2 \, dx. \Big\} \quad (3) \end{aligned}$$

In Section 7 we indicate how the proposed algorithm for the two-phase case (2) given in Section 6 can be extended to approximate multi-phase versions such as (3). The two-phase model (2) and the approximate multi-phase models such as (3) were previously considered by Chan and Vese using a level set formulation [5, 6]. Their technique is described in greater detail in Section 4.

3 Previous Work

There has been a lot of recent research activity on devising efficient algorithms for minimizing the Mumford-Shah and related segmentation energies. We have already mentioned the approach of Ambrosio and Tortorelli [1] which uses Gamma convergence to approximate the original functional with elliptic ones. It can be regarded as a diffused interface method, in which the "edges" in the segmentation are represented as thin transition layers. Another very successful PDE technique is via the level set method of Osher and Sethian [23, 24] to represent the unknown curves in the minimization of (1) or (2). This is the approach taken by Chan and Vese [5, 6], whose algorithms are among the most reliable to date. Section 4 is devoted to a detailed review of the Chan-Vese algorithm.

Recently, a number of new algorithms have been proposed for fast minimization of the piecewise constant Mumford-Shah functional. These papers are based on the observation that the PDEs resulting from the variational approaches mentioned above are hard to solve numerically. In particular, common solution techniques for these PDE do not scale very well with respect to the number of pixels in the given image. Therefore, several researchers proposed algorithms that try to avoid these difficult equations.

Gibou and Fedkiw [11] proposed an algorithm whose connections to the Mumford-Shah model are quite obscure, but which nevertheless yields segmentations that are similar to the ones obtained from Mumford-Shah based techniques. Their algorithm consists of several pre- and post-processing steps: First, the given original image is preprocessed by smoothing via the Perona-Malik non-linear diffusion scheme ([25]). Then, the k-means algorithm ([16]) is run (with two means) to separate the processed image into two phases. Finally, to introduce a way for the user to adjust the level of detail in the segmentation, the phase boundary thus found is evolved via motion by mean curvature for a certain amount of time. This algorithm has at least three parameters that need to be tuned in it (two for the Perona-Malik scheme, and one for the motion by mean curvature step) , whereas the piecewise constant Mumford-Shah model (2) has only one. A fundamental issue (among several) that we see with this algorithm is based on a rule of thumb we learned from Prof. J.-M. Morel: In machine vision applications, an algorithm with more than one parameter that needs to be tuned is actually very difficult to make completely automatic. Nevertheless, the approach of the present paper was partially motivated by this work of Gibou and Fedkiw, especially in regard to combining diffusion and thresholding for the purpose of segmentation.

Another recent proposal for fast segmentation came from Song and Chan [32]. Their approach, unlike that of Gibou and Fedkiw, is to stick with the piecewise constant model (2) – or its multiphase version – but to find a non-PDE technique of minimizing it. The algorithm they put forth commences with an initial partitioning of the pixels into two groups (phases), and then tests each pixel to see if switching it to the other phase decreases energy (2). If it does, the pixel is removed from one phase and added to the other. The algorithm sweeps through the pixels in the image with this procedure until no pixel can be updated. Numerical experiments with simple images that are close to being two valued show that this algorithm can find reasonable segmentations very quickly (in a matter of a few sweeps). However, it is also possible for it to terminate prematurely and get stuck; see [3] for examples where an improvement was also proposed.

4 Chan-Vese Algorithm

In [5] Chan and Vese proposed a level set method for numerical realization of the optimization problem (2). In this approach, the boundary of the unknown set Σ is represented as the 0-level set of a Lipschitz continuous function $\phi(x) : D \to \mathbf{R}$ so that $\Sigma = \{x \in D : \phi(x) \geq 0\}$. The idea is then to express the functional (2) in terms of the level set function ϕ. One ends up with:

$$\min_{\substack{\phi:D\to\mathbf{R}\\ c_1,c_2\in\mathbf{R}}} \underbrace{\int_D |\nabla H(\phi)| + \lambda\Big\{H(\phi)(c_1-f)^2 + (1-H(\phi))(c_2-f)^2\Big\}\,dx}_{:=CV(\phi,c_1,c_2)}. \quad (4)$$

where $H(\xi) : \mathbf{R} \to \mathbf{R}$ is the Heaviside function:

$$H(\xi) := \begin{cases} 0 & \text{if } \xi < 0, \\ 1 & \text{if } \xi \geq 0. \end{cases}$$

In practice, it is necessary to use a regularized version H_ε of (i.e. a smooth approximation to) $H(\xi)$. The resulting Euler-Lagrange equation for ϕ leads to the following gradient descent PDE:

$$\phi_t = H'_\varepsilon(\phi)\left\{\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda\Big\{(c_1-f)^2 - (c_2-f)^2\Big\}\right\}. \quad (5)$$

On the other hand, variation of (4) with respect to the two constants c_1, c_2 show that the optimal choice of these for a given ϕ is:

$$c_1 = \frac{\int_D H(\phi) f\,dx}{\int_D H(\phi)dx} \text{ and } c_2 = \frac{\int_D (1-H(\phi)) f\,dx}{\int_D (1-H(\phi))\,dx}. \quad (6)$$

The term $\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$ that appears in the right hand side of PDE (5) has a nice geometric interpretation: it is the curvature of the level sets of the function ϕ. It is also a degenerate elliptic term; clearly there are problems when $|\nabla\phi| = 0$. It makes (5) a costly PDE to solve.

We remark that the specific manner in which the Heaviside function in (5) is regularized plays an important role in the resulting flow. In the Chan-Vese paper [5], the Heaviside function is regularized as $H_\varepsilon(\xi) := \frac{1}{\pi}\arctan(x/\varepsilon) + \frac{1}{2}$ with $\varepsilon = 1$ regardless of the resolution. There, the authors point out that the non-compactly supported regularization allows their algorithm to detect interior contours in segmentations. This claim is verified in [4], where it is also shown that one could even replace the $H'_\varepsilon(\phi)$ term by 1. Another important point is that a casual

regularization of $H'_\varepsilon(\phi)$ may lead to inaccuracies due to anisotropy induced by the grid. It is possible but non-trivial to regularize H'_ε accurately; how to do so has been shown in detail in the recent work [8] of Engquist et. al.

Alternatively, one can replace the singular term $H'_\varepsilon(\phi)$ that appears in (5) with $|\nabla\phi|$:

$$\phi_t = |\nabla\phi| \left\{ \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right) - \lambda \Big\{ (c_1 - f)^2 - (c_2 - f)^2 \Big\} \right\}. \tag{7}$$

This is the approach of Marquina and Osher in [14] to solving the total variation flow – a closely related problem. It is easy to see that the evolution defined by (7) also decreases the energy (4). Furthermore, (7) can be interpreted in terms of the motion of the level sets of ϕ; each level set of ϕ moves with the following speed V in the outer normal direction:

$$\text{Normal velocity } V = \kappa - \lambda \Big((c_1 - f)^2 - (c_2 - f)^2 \Big). \tag{8}$$

where κ denotes the curvature of the level set, with the convention that it is negative when the curve is convex. However, due a maximum principle associated with (7), no new interior contour may appear during the evolution. This makes it less effective in certain applications.

The level set based algorithm of Chan and Vese can be extended to multi-phase piecewise constant models, such as the four-phase model (3). This is done in [6]. The idea is simply to introduce new level set functions to describe a greater number of regions using intersections between interiors and exteriors of the level sets. One can then use the Heaviside function to express characteristic functions of the various regions that appear in the integrals of (3) in terms of these level set functions. Subsequently, finding the Euler-Lagrange equations for the energy thus expressed in terms of level set functions yields a coupled system of nonlinear parabolic PDEs to be solved for gradient descent. The details can be found in [6]. The Chan and Vese algorithm was also extended to the piecewise smooth version of (1) where the discontinuity set is restricted to be a finite union of closed contours [6]. In this paper, we will not consider the piecewise smooth models.

5 MBO Scheme

Merriman, Bence, and Osher introduced a very interesting scheme to approximate the motion of an interface by its mean curvature [17, 18]. Their idea is to alternate the solution of the linear heat equation with thresholding. In this section we recall the original MBO algorithm by interpreting it in terms of another well known technique for approximating motion by mean curvature, namely the Allen-Cahn

equation (sometimes referred to as the phase field method):

$$u_t = 2\varepsilon\Delta u - \frac{1}{\varepsilon}W'(u). \tag{9}$$

Here, $W(\xi) : \mathbf{R} \to \mathbf{R}$ is a *double well potential* with equidepth wells at 0 and 1; for example a simple choice is $W(\xi) = \xi^2(1-\xi)^2$.

It is well known that in the limit $\varepsilon \to 0^+$ the rescaled solutions $u_\varepsilon(x, \frac{t}{\varepsilon})$ of (9) yield motion by mean curvature of the interface that separates the 0 and 1 phases of the solutions (A formal, matched asymptotic expansion that show this fact has been carried out in [26]; many researchers subsequently provided rigorous justification under various hypothesis – see [10] for example). Consider a "time splitting" scheme for the solution of (9): Choose a $\delta t > 0$ and alternate the following two steps to generate approximate solutions $u_n(x)$ at discrete times:

1. Let $v(x) = S(\delta t)u_n(x)$ where $S(\delta t)$ is the propagator (by time δt) of the following heat equation:
$$w_t = 2\varepsilon\Delta w$$
with appropriate boundary conditions.

2. Set $u_{n+1}(x) = T(\delta t)v(x)$, where $T(\delta t)$ is the propagator (by time δt) of the following ODE:
$$w_t = -\frac{1}{\varepsilon}W'(w)$$

The second step of the procedure given above is a stiff, nonlinear ODE. It has two stable, stationary solutions: $w = 0$ and $w = 1$ (and an unstable one at $w = \frac{1}{2}$). The essence of the original MBO scheme is the observation that for $\delta t > 0$ fixed and $\varepsilon \to 0^+$, solving of this ODE turns into *thresholding*: At every point x the value of $w(x,t)$ converges to one of the two stable equilibrium values (0 or 1), depending on whose basin of attraction it initially lies in:

$$\lim_{\varepsilon\to 0^+} T(\delta t)\xi = \begin{cases} 0 & \text{if } \xi \in (-\infty, \frac{1}{2}), \\ 1 & \text{if } \xi \in (\frac{1}{2}, \infty). \end{cases}$$

The MBO scheme replaces the second step (the solution of the ODE) with this thresholding:

<u>The MBO Scheme:</u>

1. Let $v(x) = S(\delta t)u_n(x)$ where $S(\delta t)$ is the propagator (by time δt) of the standard heat equation:
$$w_t = \Delta w$$
with appropriate boundary conditions.

2. Set

$$u_{n+1}(x) = \begin{cases} 0 & \text{if } v(x) \in (-\infty, \frac{1}{2}], \\ 1 & \text{if } v(x) \in (\frac{1}{2}, \infty). \end{cases}$$

The original MBO algorithm described above has been rigorously shown to approximate motion by mean curvature in [9] and [2]. Several generalizations of the basic algorithm have been given in [15, 29, 30, 31]. These works show how the original algorithm can be modified to approximate more general interfacial motions, such as motion by anisotropic curvature, motion by a constant plus curvature, and motion of multiple junctions. Moreover, an efficient numerical algorithm is proposed in [28] that is based on the fast Fourier transform on grids adapted to the evolving interface.

6 Proposed Dynamics for Mumford-Shah

Inspired by the MBO scheme that approximates motion by curvature via thresholding, we propose a similar thresholding scheme that approximates gradient descent for the two-phase piecewise constant Mumford-Shah functional. This constitutes the essential contribution of the present paper.

To discover a threshold dynamics for the gradient flow of (2), we first consider a phase-field approximation. The idea is to find a diffuse interface approximation of (2) and use it to motivate our proposed dynamics much as the Allen-Cahn equation (9) was used to motivate the original MBO scheme in Section 5.

A diffuse interface approximation for (2) is given by the following sequence of energies:

$$MS_\varepsilon(u, c_1, c_2) := \int_D \varepsilon|\nabla u|^2 + \frac{1}{\varepsilon}W(u) + \lambda\Big\{u^2(c_1 - f)^2 + (1-u)^2(c_2 - f)^2\Big\}\, dx. \tag{10}$$

where $\varepsilon > 0$ and the potential $W(\xi) = \xi^2(1-\xi)^2$ is exactly the same as in Section 5. It is possible to establish rigorously that $MS_\varepsilon \to MS$ as $\varepsilon \to 0^+$ in the sense of Gamma convergence based on the work [19, 20]. Variation of energy (10) with respect to u yields the following gradient descent equation:

$$u_t = 2\varepsilon\Delta u - \frac{1}{\varepsilon}W'(u) - 2\lambda\Big\{u(c_1 - f)^2 + (u-1)(c_2 - f)^2\Big\}. \tag{11}$$

We can find a candidate for the threshold dynamics of (2) by splitting equation (11). There are several ways in which (11) can be split. One possibility, which we shall pursue here, is to split it so that the thresholding step is the same as the one in the original MBO scheme. That leaves the solution of a linear heat equation *with a forcing term* as the first step of the new algorithm:

1. Let $v(x) = S(\delta t)u_n(x)$ where $S(\delta t)$ is the propagator (by time δt) of the linear parabolic equation:

$$w_t = \Delta w - 2\tilde{\lambda}\Big(w\big(c_1 - f\big)^2 + (w-1)\big(c_2 - f\big)^2\Big). \tag{12}$$

   with appropriate boundary conditions.

2. Set

$$u_{n+1}(x) = \begin{cases} 0 & \text{if } v(x) \in (-\infty, \frac{1}{2}), \\ 1 & \text{if } v(x) \in (\frac{1}{2}, \infty). \end{cases}$$

Notice that the proposed candidate dynamics above differs from the original MBO scheme only in the linear parabolic PDE involved. By using the maximum principle, it can be easily seen that just like the heat equation in the original MBO scheme, the PDE of the proposed scheme above preserves the condition $0 \leq w(x,t) \leq 1$ for all time. In fact, just like the original MBO algorithm, the dynamics proposed above preserves the order of solutions (i.e. it satisfies a comparison principle). Also notice that we have replaced the parameter λ that appears in the variational model (2) and in the diffuse interface flow (11) by $\tilde{\lambda}$ to indicate that the relation of $\tilde{\lambda}$ to the original scale parameter λ is yet to be determined.

To understand how $\tilde{\lambda}$ in the proposed dynamics should be related to the scale parameter λ in the model (2), we will consider the way a characteristic function evolves under the proposed PDE (12). Let $f(x)$ be a given continuous image, and let $u(x,0) = \mathbf{1}_\Sigma(x)$ be the initial condition for (12). Assume that Σ has smooth boundary $\partial\Sigma$, and let $p \in \partial\Sigma$. We will be looking at the solution of (12) in a neighborhood of the point p for a very short time; so we replace the term $f(x)$ in (12) by the constant $f(p)$, and for convenience define $A := (c_1 - f(p))^2$ and $B := (c_2 - f(p))^2$.
We first consider the one dimensional version of (12) on all $\mathbf{R}$:

$$\begin{aligned} \psi_t &= \psi_{xx} - 2\tilde{\lambda}(A+B)\psi + 2\tilde{\lambda}B \text{ on } (x,t) \in \mathbf{R} \times \mathbf{R}^+, \\ \psi(x,0) &= \mathbf{1}_{R^-}(x). \end{aligned} \tag{13}$$

The solution of (13) is simply:

$$\begin{aligned} \psi(x,t) &= e^{-2\tilde{\lambda}(A+B)t}(G_t * \mathbf{1}_{\mathbf{R}^-})(x) + 2\tilde{\lambda}B\int_0^t e^{-2\tilde{\lambda}(A+B)(t-s)}\,ds \\ &= e^{-2\tilde{\lambda}(A+B)t}\left\{(G_t * \mathbf{1}_{\mathbf{R}^-})(x) + \frac{B}{A+B}\Big(e^{2\tilde{\lambda}(A+B)t} - 1\Big)\right\}. \end{aligned} \tag{14}$$

where $G_t(x)$ is the Gaussian kernel in one dimension:

$$G_t(x) = \frac{1}{\sqrt{4\pi t}}e^{-\frac{x^2}{4t}}.$$

At $x = 0$ and $t = \delta t$, (14) gives

$$\begin{aligned}\psi(0,\delta t) &= e^{-2\tilde{\lambda}(A+B)\delta t}\left\{\frac{1}{2} + \frac{B}{A+B}\Big(e^{2\tilde{\lambda}(A+B)\delta t} - 1\Big)\right\} \\ &\approx \frac{1}{2} + \tilde{\lambda}\delta t(B-A).\end{aligned} \tag{15}$$

Differentiating (14) we get

$$\partial_x\psi(0,\delta t) = -e^{-2\tilde{\lambda}(A+B)\delta t}\frac{1}{\sqrt{4\pi\delta t}} \approx -\frac{1}{\sqrt{4\pi\delta t}}. \tag{16}$$

Let $p(t)$ be such that $\psi(p(t),t) = \frac{1}{2}$. Strict monotonicity of ψ for positive time implies that this defines $p(t)$ uniquely for $t > 0$. Formulas (15) and (16) allow us to estimate $p(t)$ as:

$$p(t) \approx -\frac{u(0,\delta t) - \frac{1}{2}}{\partial_x u(0,\delta t)} \approx 2\sqrt{\pi}\tilde{\lambda}(\delta t)^{\frac{3}{2}}(B-A). \tag{17}$$

We now move on to the 2-D problem

$$\begin{aligned} w_t &= \Delta w - 2\tilde{\lambda}(A+B)w + 2\tilde{\lambda}B \text{ for } (x,t) \in \mathbf{R}^2 \times \mathbf{R}^+, \\ w(x,0) &= \mathbf{1}_\Sigma(x).\end{aligned} \tag{18}$$

Again we consider the solution of (18) for a short interval of time and in a neighborhood of the point $p \in \partial\Sigma$. For $t \geq 0$, let $\Sigma(t) \subset \mathbf{R}^2$ be a bounded domain with smooth boundary $\partial\Sigma(t)$,and with $\Sigma(0) = \Sigma$. We assume that $\partial\Sigma(t)$ evolves smoothly in time. Let $d_{\Sigma(t)}(x)$ be the signed distance function to $\partial\Sigma(t)$.

Motivated by a change of variables used in [26], we consider the following ansatz as a candidate for the solution of (18) in a neighborhood of p:

$$w(x,t) = \psi(d_{\Sigma(t)}(x),t).$$

We have:

$$\begin{aligned} w_t(x,t) =& \psi_x\big(d_{\Sigma(t)}(x),t\big)\left(\frac{\partial}{\partial t}d_{\Sigma(t)}(x)\right) + \psi_t\big(d_{\Sigma(t)}(x),t\big) \\ =& \psi_x\big(d_{\Sigma(t)}(x),t\big)\left(\frac{\partial}{\partial t}d_{\Sigma(t)}(x)\right) + \psi_{xx}\big(d_{\Sigma(t)}(x),t\big) \\ &- 2\tilde{\lambda}(A+B)\psi + 2\tilde{\lambda}B.\end{aligned} \tag{19}$$

And,

$$\Delta w(x,t) = \psi_x\big(d_{\Sigma(t)}(x),t\big)\Delta d_{\Sigma(t)}(x) + \psi_{xx}\big(d_{\Sigma(t)}(x),t\big)|\nabla d_{\Sigma(t)}(x)|^2. \tag{20}$$

Combining formulas (19) and (20) and noting that $|\nabla d_{\Sigma(t)}(x)|^2 = 1$ we get

$$w_t(x,t) = \Delta w(x,t) - 2\tilde{\lambda}(A+B)w + 2\tilde{\lambda}B \\ + \psi\big(d_{\Sigma(t)}(x),t\big)\left\{\frac{\partial}{\partial t}d_{\Sigma(t)}(x) - \Delta d_{\Sigma(t)}(x)\right\}.$$

We see that if the curve $\partial\Sigma(t)$ evolves in such a way that the following holds

$$\frac{\partial}{\partial t}d_{\Sigma(t)}(x) = \Delta d_{\Sigma(t)}(x) \tag{21}$$

then the ansatz given by $u(x,t)$ satisfies equation (18). Along the curve given by $\partial\Sigma(t)$, we have that $\Delta d_{\Sigma(t)}(x)$ is the curvature, and $\partial_t d_{\Sigma(t)}(x)$ is the normal velocity. Thus, if $\gamma(\cdot,t)$ is a parametrization of $\partial\Sigma(t)$, then equation (21) will be satisfied close to the curve provided that $\gamma(t)$ evolves via motion by curvature:

$$\frac{\partial}{\partial t}\gamma(\cdot,t) = \kappa\big(\gamma(\cdot,t)\big)N\big(\gamma(\cdot,t)\big). \tag{22}$$

where κ denotes the curvature and N denotes the outward unit normal of the curve. Now let $\Gamma(\cdot,t)$ be a parametrized curve that evolves according to the motion law:

$$\begin{aligned}\frac{\partial}{\partial t}\Gamma(\cdot,t) &= \Big(\kappa\big(\Gamma(\cdot,t)\big) + 2\tilde{\lambda}\sqrt{\pi\delta t}(B-A)\Big)N\big(\Gamma(\cdot,t)\big) \\ \Gamma(\cdot,0) &= \gamma(\cdot,0).\end{aligned} \tag{23}$$

Then, formulas (22), and (17) imply that

$$\Gamma(\cdot,\delta t) \approx \left\{x \in \mathbf{R}^2 : w(x,\delta t) = \frac{1}{2}\right\}.$$

Proposed dynamics: We can now relate the parameter $\tilde{\lambda}$ in the candidate dynamics given above to the scale parameter λ that appears in the model (2). We simply choose $\tilde{\lambda}$ so that the normal velocity of the curve in (23) matches (8):

$$\tilde{\lambda} = \frac{\lambda}{2\sqrt{\pi\delta t}}.$$

With that, our threshold dynamics for the two-phase piecewise constant Mumford-Shah functional finally becomes:

1. Let $v(x) = S(\delta t)u_n(x)$ where $S(\delta t)$ is the propagator (by time δt) of the linear parabolic equation:

$$w_t = \Delta w - \frac{\lambda}{\sqrt{\pi\delta t}}\Big(w\big(c_1 - f\big)^2 + (w-1)\big(c_2 - f\big)^2\Big)$$

with appropriate boundary conditions.

2. Set

$$u_{n+1}(x) = \begin{cases} 0 & \text{if } v(x) \in (-\infty, \frac{1}{2}], \\ 1 & \text{if } v(x) \in (\frac{1}{2}, \infty). \end{cases}$$

The constants c_1, c_2 can be updated after each thresholding step according to the formulas:

$$c_1 = \frac{\int_D uf\,dx}{\int_D u\,dx} \text{ and } c_2 = \frac{\int_D (1-u)f\,dx}{\int_D (1-u)\,dx}.$$

However, our numerical experience suggests that it is also sufficient to update c_1, c_2 less frequently, for instance once every few iterations of the above algorithm.

The PDE that constitutes the first step of the iterative algorithm proposed above is a *linear* diffusion equation. As such, there are many efficient numerical techniques for its solution. One possibility is to discretize it implicitly and solve the resulting elliptic equation via a fast method such as plain vanilla multigrid. PDEs that result from many other approaches are much more troublesome. For example, the level set approach leads to PDEs that are singular and degenerate. For such equations, standard fast solution techniques such as multigrid do not always attain their "textbook" convergence rates. Also note that for small δt the PDE in the proposed algorithm actually becomes easier to solve, since it becomes, loosely speaking, more "diagonally dominant".

An important distinction of the threshold dynamics proposed above from the standard MBO algorithm is that the diffusion step involves the parameter δt explicitly in the linear parabolic equation that needs to be solved.

7 Extensions and Variants

In this section we consider several extensions of and variations on the thresholding algorithm proposed in the previous section.

Multi-phase segmentation: We start by indicating how the proposed threshold dynamics of Section 6, which minimizes the two-phase segmentation model (2), can be extended to multi-phase segmentation. This is easy to accomplish, and will be illustrated on the four-phase model (3): Once again we first write down a

phase-field approximation to (3), and then split the terms appropriately. The diffuse interface version (10) of (2) can be extended to four-phase segmentation as follows:

$$MS_\varepsilon(u_1,u_2,c_1,\dots,c_4) := \int_D \varepsilon\Big(|\nabla u_1|^2+|\nabla u_2|^2\Big)+\frac{1}{\varepsilon}\big(W(u_1)+W(u_2)\big) + \lambda\Big\{u_1^2u_2^2(c_1-f)^2+u_1^2(1-u_2)^2(c_2-f)^2 + (1-u_1)^2u_2^2(c_3-f)^2+(1-u_1)^2(1-u_2)^2(c_4-f)^2\Big\}\,dx. \quad (24)$$

Variations with respect to the functions u_1, u_2 yield the following gradient descent equations:

$$\partial_t u_1 = 2\varepsilon\Delta u_1-\frac{1}{\varepsilon}W'(u_1)-2\lambda\Big\{u_1\big[u_2^2(c_1-f)^2+(1-u_2)^2(c_2-f)^2\big] + (u_1-1)\big[u_2^2(c_3)^2+(1-u_2)^2(c_4-f)^2\big]\Big\}. \quad (25)$$

and

$$\partial_t u_2 = 2\varepsilon\Delta u_2-\frac{1}{\varepsilon}W'(u_2)-2\lambda\Big\{u_2\big[u_1^2(c_1-f)^2+(1-u_1)^2(c_3-f)^2\big] + (u_2-1)\big[u_1^2(c_2-f)^2+(1-u_1)^2(c_4-f)^2\big]\Big\}. \quad (26)$$

Splitting equations (25) and (26), in the same manner that (11) was split in Section 6, yields the following threshold dynamics:

1. Let $(v_1(x),v_2(x)) = S(\delta t)(u_{1,n}(x),u_{2,n}(x))$ where $S(\delta t)$ is the propagator (by time δt) of the following system of parabolic equations:

$$\partial_t w_1 = \Delta w_1-\frac{\lambda}{\sqrt{\pi\delta t}}\Big\{w_1\big[w_2^2(c_1-f)^2+(1-w_2)^2(c_2-f)^2\big] + (w_1-1)\big[w_2^2(c_3)^2+(1-w_2)^2(c_4-f)^2\big]\Big\}.$$

and

$$\partial_t w_2 = \Delta w_2-\frac{\lambda}{\sqrt{\pi\delta t}}\Big\{w_2\big[w_1^2(c_1-f)^2+(1-w_1)^2(c_3-f)^2\big] + (w_2-1)\big[w_1^2(c_2-f)^2+(1-w_1)^2(c_4-f)^2\big]\Big\}.$$

2. For $j=1,2$, set

$$u_{j,n+1}(x) = \begin{cases} 0 & \text{if } v_j(x)\in(-\infty,\frac{1}{2}],\\ 1 & \text{if } v_j(x)\in(\frac{1}{2},\infty).\end{cases}$$

As in the two-phase model, the constants of the segmentation can be updated after each thresholding step in the iteration. In the case of the four phase model, variations of the energy (24) with respect to $c_1, \dots, c_4$ yield the following optimal choices:

$$c_1 = \frac{\int_D u_1 u_2 f \, dx}{\int_D u_1 u_2 \, dx} \qquad c_2 = \frac{\int_D u_1 (1-u_2) f \, dx}{\int_D u_1 (1-u_2) \, dx}$$
$$c_3 = \frac{\int_D (1-u_1) u_2 f \, dx}{\int_D (1-u_1) u_2 \, dx} \qquad c_4 = \frac{\int_D (1-u_1)(1-u_2) f \, dx}{\int_D (1-u_1)(1-u_2) \, dx}.$$

Unlike the two-phase dynamics of Section 6, multi-phase dynamics in general involves a *non-linear* system of PDEs as one of its iterative steps, as can be seen above in the case of the four-phase model. However, the nonlinearity is only in the lowest order term, and we notice that the j-th equation in the system is in fact *linear* in w_j. Therefore, a natural solution approach would be a semi-implicit discretization where in the right hand side of the j-th equation w_j is treated implicitly, and w_i for $i \neq j$ are treated explicitly.

Alternatives to proposed dynamics: As mentioned before, the phase-field approximation (11) can be split in more than one way; the threshold dynamics we described in Section 6 is only one of them. Another way to split (11) is to group together all terms in the right hand side except the Laplacian; that leads to a thresholding scheme whose PDE step involves, just like the MBO scheme, the standard heat equation:

1. Let $v(x) = S(\delta t) u_n(x)$ where $S(\delta t)$ is the propagator (by time δt) of the standard heat equation:
$$w_t = \Delta w$$
with appropriate boundary conditions.

2. For every x, set
$$u_{n+1}(x) = \lim_{t \to \infty} y(t)$$
where $y(t)$ is the solution of the initial value problem:
$$\begin{cases} y' = -W'(y) - \tilde{\lambda}\Big(y(c_1 - f)^2 + (y-1)(c_2 - f)^2\Big), \\ y(0) = v(x). \end{cases}$$

The second step of the dynamics above can be interpreted as a form of thresholding. This would be done by finding the basins of attraction of stable equilibrium

solutions of the ODE. Then, the second step of the algorithm is equivalent to setting the value of $u_{n+1}(x)$ equal to the equilibrium solution of the ODE whose basin of attraction $w(x)$ happens to lie in. Note that unlike the one in the original MBO scheme, this thresholding would be space dependent: this is how the image information gets incorporated into the dynamics. Thus, unlike the original MBO scheme, the threshold value would be different from $\frac{1}{2}$ and depend on x. In this sense, the algorithm suggested above is akin to the extensions of MBO considered in [15, 30]

Another possibility is to split the dynamics into three steps. A convenient way of doing this is to do operator splitting in the linear parabolic equation that constitutes the first step of the algorithm proposed in Section 6. In other words, one can solve the PDE

$$w_t = \Delta w - \frac{\lambda}{\sqrt{\pi\delta t}}\Big(w\big(c_1 - f\big)^2 + (w-1)\big(c_2 - f\big)^2\Big)$$

up to time δt starting from the initial condition $w(x,0) = w_0(x)$ by repeating the following steps n times:

1.1 Solve the standard heat equation:

$$w_t = \Delta w \text{ with initial condition } w(x,0) = w_0(x).$$

1.2 Solve the pointwise (in x) ODE:

$$z_t = -\frac{\lambda}{\sqrt{\pi\delta t}}\Big(z\big(c_1 - f\big)^2 + (z-1)\big(c_2 - f\big)^2\Big)$$

with initial condition $z(x,0) = w(x, \frac{\delta t}{n})$.
1.3 Set $w_0(x) = z(x, \frac{\delta t}{n})$.

An advantage of the two alternatives presented above to the dynamics of Section 6 is that the PDE involved in these variants is the standard heat equation. The PDE of the algorithm of Section 6, although linear, has non-constant coefficients in its lowest order terms.

8 Numerical Examples

In this section we illustrate the algorithm of Section 6 and the multi-phase algorithm of Section 7 on a few test images.

As we mentioned before, the PDEs that appear in the proposed algorithms of Sections 6 and 7 can be solved efficiently in a number of different ways. For example, the PDE that appears in the proposed algorithm for two-phase segmentation

given in Section 6 has the form:

$$u_t = \Delta u - A(x)u + B(x)$$

To generate the numerical results shown in this section, we used the following discretization in time:

$$\frac{u^{n+1} - u^n}{\delta\tau} = \Delta u^{n+1} - Cu^{n+1} + (C - A(x))u^n + B(x) \tag{27}$$

where $\delta\tau$ is the time step size and C is a constant to be chosen large enough compared to $A(x)$. Once discretized also in space in some standard manner (using for instance the five point stencil for the Laplacian), the resulting linear system was solved for u^{n+1} using the fast Fourier transform.

The scheme given above is unconditionally stable and certainly fast; however, it must be pointed out that it is not necessarily the best way to solve the linear parabolic PDE in question; the scheme given above is intended merely as an example of what can be done. The more important point is the following: The PDE in question is simple enough that there are a variety of powerful and completely standard numerical techniques available for its solution. For instance, if desired one can easily apply the standard multigrid algorithm to solve the linear system in order to achieve faster results than what is attainable by the specific method quoted above. We leave the decision of the best choice to experts in fast algorithms for linear equations.

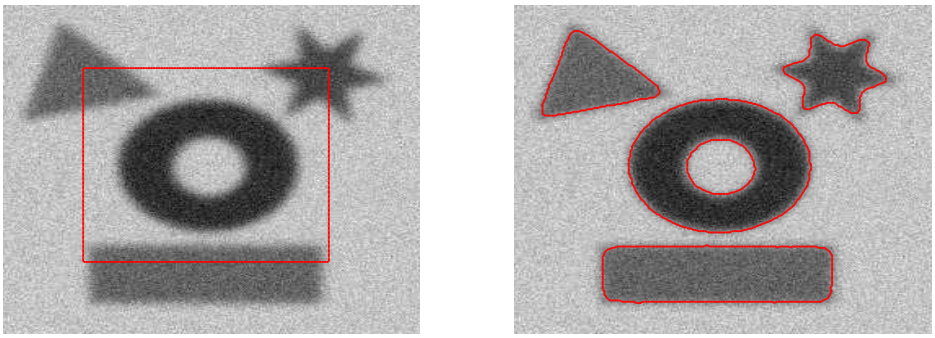

Figure 1: Test with a synthetic image of resolution 256×256. Left hand side figure has the initial contour superimposed on the image to be segmented. Right hand side figure shows the final contour found using the algorithm of Section 6. Only 3 iterations were necessary to reach an essentially steady state. Each iteration involved taking 3 steps with scheme (27), bringing the total number of steps taken for curve evolution to 9.

Calculations of Section 6 justify the proposed algorithms by considering their $\delta t \to 0^+$ limit. However, from a practical standpoint convergence in this limit is

not necessarily the most important aspect of the algorithms. Indeed, for applications the greatest strength of the proposed algorithms is that they converge very quickly to a fixed point especially when δt is chosen large.

Figure 1 shows results of an experiment with a synthetic image at 256×256 resolution. The image consists of objects with blurred boundaries corrupted with the addition of a significant amount of Gaussian noise. Nevertheless, the image is close to being piecewise constant; this makes it relatively easy to segment using piecewise constant models such as the ones considered in this paper. The proposed algorithm of Section 6 for two-phase segmentation required only 3 iterations to reach the steady state shown. Each one of these iterations requires the solution of the linear parabolic PDE for a short time interval of length δt; we accomplished this by taking three steps of size $\frac{\delta t}{3}$ with scheme (27). Using a larger number of (shorter) steps to solve the PDE at each iteration made no appreciable difference in the result.

Figure 2 shows the result of an experiment with the two-phase algorithm of Section 6 on a real test image at 256×256 resolution. This image is slightly more challenging than that of Figure 1, in that it is not very well approximated by just two constants. The algorithm required 15 iterations to reach an essentially steady state. Once again, at each iteration 3 steps of scheme (27) were used.

Figure 3 shows the result of an experiment with the four-phase algorithm presented in Section 7. The algorithm found only three regions with the initial conditions shown. We observed that multiphase segmentation algorithms, including the Chan-Vese algorithm and the one proposed in this paper, have more sensitive dependence on initial conditions than their two-phase versions. This is an aspect of multiphase segmentation algorithms that we did not intend to address in this work, our emphasis being instead on improving computational complexity.

Figure 4 shows the result of applying the proposed two-phase segmentation algorithm of Section 6 using different values of δt. From left to right, top to bottom, the algorithm was run on the test image shown in Figure 2 using $\delta t = \delta t_0$, $\delta t = 2\delta t_0$, $\delta t = 4\delta t_0$, and $\delta t = 8\delta t_0$, respectively. As can be seen, the solutions obtained are quite close; however with very large choices of δt there is noticeable oversmoothing.

Figure 5 shows the result of applying the proposed algorithm of Section 6 to the test image of Figure 2 at different resolutions. The resolutions range from 128×128 to 1024×1024. In each case the same parameters were used in the algorithm. In particular, in each case only 15 iterations of the algorithm were necessary, and only 3 steps of scheme (27) were taken per iteration. In contrast, the number of iterations in standard implementations of the two dimensional Chan-Vese algorithm grows at least linearly with respect to the number of pixels in the image.

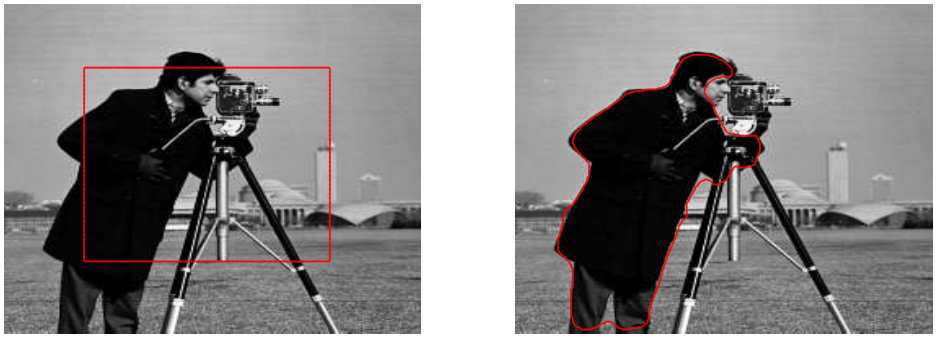

Figure 2: Test with a real image of resolution 256×256. Left hand side figure has the initial contour superimposed on the image to be segmented. Right hand side figure shows the final contour found using the algorithm of Section 6. Only 15 iterations were necessary to reach an essentially steady state. Each iteration involved taking 3 steps with scheme (27), bringing the total number of steps taken for curve evolution to 45.

9 Discussion

We mentioned in Section 8 that a major advantage of the algorithms proposed in this paper is that they can be applied with very large choices of the parameter δt, in which case they reach a fixed point very rapidly. However, when δt is too large, there is naturally a loss of accuracy in the recovered boundaries. More specifically, we observed that if δt is too large compared to the scale parameter λ^{-1}, then boundaries of the segmentation turn out to be slightly oversmoothed. It is possible to compensate for this lack of accuracy without giving up on the use of a very large δt. One can start the computation with a very large δt, and after a few step (once the segmentation is within a small neighborhood of the solution) switch to a more moderate choice. To keep things simple, in obtaining the numerical examples of the previous section we simply used a single moderate to small choice of δt; the convergence was still very fast.

Numerical solution of model (2), and also of the multi-phase versions such as (3), tend to be easier when the scale parameter λ is large, corresponding to a fine segmentation (i.e. a segmentation that allows high curvatures in the region boundaries). For then the segmentation models reduce to the well known *k-means* problem of data clustering (where data is one dimensional in the case of grayscale images); this fact was observed in [11]. Furthermore, when the geometric terms are dropped from the piecewise constant models, the algorithms of Chan and Vese become equivalent to the k-means algorithm, and converge rapidly. Another way of saying this is: When λ is large the PDE of the Chan-Vese algorithm becomes *di-*

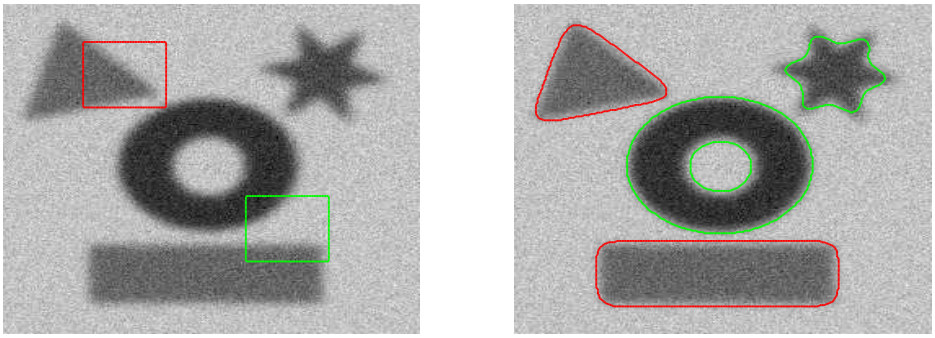

Figure 3: Segmenting the synthetic test image of Figure 1 using the multi-phase version of the algorithm in Section 6, which is explained in Section 7. Only 15 iterations were necessary to reach an essentially steady state. Each iteration involved taking 3 steps with scheme (27), bringing the total number of steps taken for curve evolution to 45.

agonally dominant (i.e. more and more local) so that even slow solution techniques converge rather quickly.

Most of the existing PDE based algorithms for minimizing segmentation functionals such as (2) and (3) suffer from a serious degradation in speed when the scale parameter λ is moderate or small, corresponding to a coarse segmentation in which the geometric penalty term plays a significant role. In this regime, our proposed algorithm exhibits dramatically improved efficiency.

It is possible to come up with a number of ad hoc tricks to further speed up the proposed segmentation process. A simple one of such involves starting the computation from a good initial guess: One can use k-means clustering (i.e. segmentation with no geometric regularization) to produce an initial segmentation to feed into the threshold dynamics, especially when a detailed final segmentation is desired (corresponding to a large λ value). This is equivalent to running the proposed algorithm first with a very large choice of λ, and then using the result found as initial guess for another run in which the user specified value of λ is used. A further option is to gradually decrease the parameter λ. Of course, these well-known "tricks" are not specific to the algorithm of this paper but can be used in combination with others as well.

Finally, it is worth discussing certain numerical difficulties of threshold dynamics due to finite discretization size. For the MBO algorithm, it is well-known that if δt is too small compared to the spatial resolution (i.e. the number of grid points), then the interface can get stuck and not move. This issue has been addressed by Ruuth in [28], where an adaptive version of the MBO algorithm is developed and solved using the fast Fourier transform. The same improvement can certainly be

Figure 4: Solutions (region boundaries) obtained by the algorithm of Section 6 on the test image of Figure 2 using different values of δt. Going left to right, top to bottom, the value of δt was double in each case. As implied by our calculations, there is little difference between the results, although for very large values of δt oversmoothing becomes noticeable.

developed for the proposed dynamics of this paper. Moreover, in our applications, accurate simulation of the dynamics is not important as long as we reach the correct steady state quickly; in practice, one therefore uses values of δt that are nowhere as small, in which case getting stuck becomes a non-issue.

Acknowledgment

The second author would like to thank the National Center for Theoretic Study, Taiwan for hosting his stay while parts of this research were being conducted.

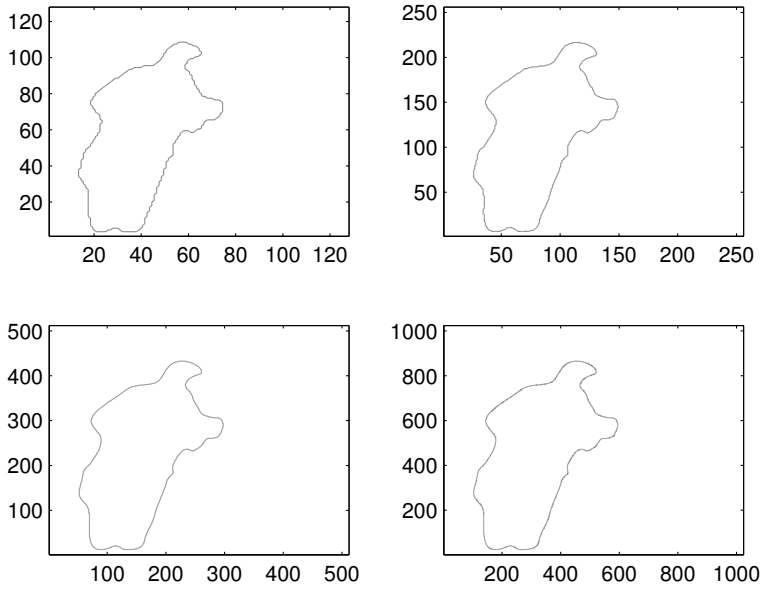

Figure 5: Solutions (region boundaries) obtained by the algorithm of Section 6 using the test image of Figure 2 at different resolutions. Going left to right, top to bottom, the resolutions were 128^2, 256^2, 512^2, and 1024^2, respectively. The same parameters were used throughout. In particular, the same number of iterations (namely 15) of the algorithm were taken in each case, regardless of the resolution. Also, the same number of steps of scheme (27) (namely 3) were used for each iteration, at every resolution.

References

[1] Ambrosio, L.; Tortorelli, V. M. *Approximation of functionals depending on jumps by elliptic functionals via Gamma convergence.* Comm. Pure Appl. Math. **43** (1990), pp. 999 – 1036.

[2] Barles, G.; Georgelin, C. *A simple proof of convergence for an approximation scheme for computing motions by mean curvature.* SIAM J. Numer. Anal. **32**:2 (1995), pp. 484 – 500.

[3] Chan, T. F.; Esedoglu, S. *A multiscale algorithm for Mumford-Shah image segmentation.* UCLA CAM Report 03-57 (December 2003).

[4] Chan, T. F.; Esedoglu, S.; Nikolova, M. *Algorithms for finding global minimizers of image segmentation and denoising models.* UCLA CAM Report 04-54 (September 2004).

[5] Chan, T. F.; Vese, L. A. *Active contours without edges*. IEEE Transactions on Image Processing. **10**:2 (2001), pp. 266 – 277.

[6] Chan, T. F.; Vese, L. A. *A multiphase level set framework for image segmentation using the Mumford and Shah model.* International Journal of Computer Vision. **50**:3 (2002), pp. 271 – 293.

[7] Dal Maso, G. *An introduction to* Γ*-convergence*. Progress in Nonlinear Differential Equations and their Applications. Birkhauser Boston Inc., Boston, MA 1993.

[8] Engquist, B.; Tornberg, A-K.; Tsai, Y-H. *Dirac-*δ *functions in level set methods.* CAM Report 04-16, UCLA, 2003. Under review.

[9] Evans, L. C. *Convergence of an algorithm for mean curvature motion.* Indiana University Mathematics Journal. **42** (1993), pp. 553 – 557.

[10] Evans, L. C.; Soner, H. M.; Souganidis, P. E. *Phase transitions and generalized motion by mean curvature.* Comm. Pure Appl. Math. **45**:9 (1992), pp. 1097 – 1123.

[11] Gibou F.; Fedkiw, R. *A fast hybrid k-means level set algorithm for segmentation.* Preprint.

[12] Kass, M.; Witkin, A.; Terzopoulos, D. *Snakes: Active contour models.* International Journal of Computer Vision. **1** (1988), pp. 321 – 331.

[13] March, R. *Visual reconstruction with discontinuities using variational methods.* Image and Vision Computing. **10** (1992), pp. 30 – 38.

[14] Marquina, A.; Osher, S. *Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal.* SIAM J. Sci. Comput., **22**(2), pp.387–405.

[15] Mascarenhas, P. *Diffusion generated motion by mean curvature.* UCLA CAM Report 92-33 (July 1992).

[16] MacQueen, J. *Some methods for classification and analysis of multivariate observations.* In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Vol. 1, pp. 281 – 297. Editors: Le Cam, L. M. and Neyman, J. University of California Press (1967).

[17] Merriman, B.; Bence, J. K.; Osher, S. J. *Diffusion generated motion by mean curvature.* UCLA CAM Report 92-18 (April 1992).

[18] Merriman, B.; Bence, J. K.; Osher, S. J. *Motion of multiple junctions: a level set approach.* J. Comput. Phys. **112**:2 (1994), pp. 334 – 363.

[19] Modica, L.; Mortola, S. *Un esempio di Gamma-convergenza*. Boll . Un. Mat. Ital. B (5). **14**:1 (1977), pp. 285 – 299.

[20] Modica, L. *The gradient theory of phase transitions and the minimal interface criterion.* Archive for Rational Mechanics and Analysis. **93** (1987), pp. 123 – 142.

[21] Morel, J.-M.; Solimini, S. *Variational Methods in Image Segmentation*. Progress in Nonlinear Differential Equations and their Applications. Birkhauser (1995).

[22] Mumford, D.; Shah, J. *Optimal approximations by piecewise smooth functions and associated variational problems.* Comm. Pure Appl. Math. **42** (1989), pp. 577 – 685.

[23] Osher, S.; Fedkiw, R. *Level set methods and dynamic implicit surfaces.* Applied Mathematical Sciences, 153. Springer-Verlag, New York, 2003.

[24] Osher, S.; Sethian, J. *Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations.* Journal of Computational Physics. **79**:1 (1988), pp. 12 – 49.

[25] Perona, P.; Malik, J. *Scale-space and edge detection using anisotropic diffusion.* IEEE PAMI **12**:1 (1990), pp. 629–639.

[26] Rubinstein, J.; Sternberg, P.; Keller, J. B. *Fast reaction, slow diffusion, and curve shortening.* SIAM J. Appl. Math. **49**:1 (1989), pp. 116 – 133.

[27] Rudin, L.; Osher, S.; Fatemi, E. *Nonlinear total variation based noise removal algorithms*. Physica D. **60** (1992), pp. 259 – 268.

[28] Ruuth, S. J. *Efficient algorithms for diffusion-generated motion by mean curvature.* J. Comput. Phys. **144**:2 (1998), pp. 603 – 625.

[29] Ruuth, S. J. *A diffusion generated approach to multiphase motion.* J. Comput. Phys. **145**:1 (1998), pp. 166 – 192.

[30] Ruuth, S. J.; Merriman, B. *Convolution generated motion and generalized Huygens' principles for interface motion.* SIAM J. Appl. Math. **60**:3 (2000), pp. 868 – 890.

[31] Ruuth, S. J.; Merriman, B. *Convolution-thresholding methods for interface motion.* J. Comput. Phys. **169**:2 (2001), pp. 678 – 707.

[32] Song, B.; Chan, T. *A fast algorithm for level set based optimization.* UCLA CAM Report 02-68 (December 2002).