IMAGE SEGMENTATION BY CONVEX APPROXIMATION OF THE MUMFORD-SHAH MODEL

XIAOHAO CAI*, RAYMOND CHAN†, AND TIEYONG ZENG‡

Abstract. The Mumford-Shah model is one of the most important image segmentation models, and has been studied extensively in the last twenty years. In this paper, we propose a convex segmentation model based on the Mumford-Shah model. Our model can be seen as finding a smooth approximation g to the piecewise smooth solution of the Mumford-Shah model. Once g is obtained, the two-phase or multiphase segmentation is done by thresholding g. The thresholds can be given by the users to reveal specific features in the image or they can be obtained automatically using a K-means method. Because of the convexity of the model, g can be solved efficiently by techniques like the split-Bregman algorithm or the Chambolle-Pock method. We prove that our model is convergent and the solution g is always unique. In our method, there is no need to specify the number of segments K ($K \geq 2$) before finding g. We can obtain any K-phase segmentations by choosing $(K-1)$ thresholds after g is found; and there is no need to recompute g if the thresholds are changed. Experimental results show that our method performs better than many standard 2-phase or multi-phase segmentation methods for very general images, including anti-mass, tubular, noisy, and blurry images.

Key words. Image segmentation, Mumford-Shah model, split-Bregman, total variation.

AMS subject classifications. 52A41, 65D15, 68W40, 90C25, 90C90

1. Introduction.

Let $\Omega \subset \mathbb{R}^2$ be a bounded open connected set, Γ be a compact curve in Ω, and $f : \Omega \to \mathbb{R}$ be a given image. Without loss of generality, we restrict the range of f in [0,1] and hence $f \in L^\infty(\Omega)$. In 1989, Mumford and Shah [34] proposed to solve the segmentation problem by minimizing the following energy:

$$E_{\mathrm{MS}}(g, \Gamma) = \frac{\lambda}{2} \int_\Omega (f-g)^2 dx + \frac{\mu}{2} \int_{\Omega \setminus \Gamma} |\nabla g|^2 dx + \mathrm{Length}(\Gamma), \tag{1.1}$$

where λ and μ are positive parameters, and $g : \Omega \to \mathbb{R}$ is continuous or even differentiable in $\Omega \setminus \Gamma$ but may be discontinuous across Γ. Here, the length of Γ can be written as $\mathcal{H}^1(\Gamma)$, the 1-dimensional Hausdorff measure in $\mathbb{R}^2$, see [3]. Because model (1.1) is nonconvex, it is very challenging to find or approximate its minimizer.

In [1], the Mumford-Shah energy (1.1) was approximated by a sequence of simper elliptic variational problems where the length of Γ was replaced by a phase field energy. Later, non-local approximation of (1.1) was proposed in [10, 22, 33]. By using a family of continuous and non-decreasing functions, they avoid computing Γ explicitly. In particular, their methods solve an anisotropic variant of the Mumford-Shah model (1.1). In [9], numerical approaches based on a discrete functional were considered for solving (1.1). Recently, a novel primal-dual algorithm based on a convex representation of (1.1) was proposed. It can solve (1.1) accurately. However, for a 128×128 image, it requires 600 seconds on a TEsla C1060 GPU machine. Until now, the bottleneck of solving (1.1) is still that the model itself is nonconvex.

* Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong. Email: xhcai@math.cuhk.edu.hk.

† Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong. Email: rchan@math.cuhk.edu.hk. This work is partially supported by RGC 400510 and DAG 2060408.

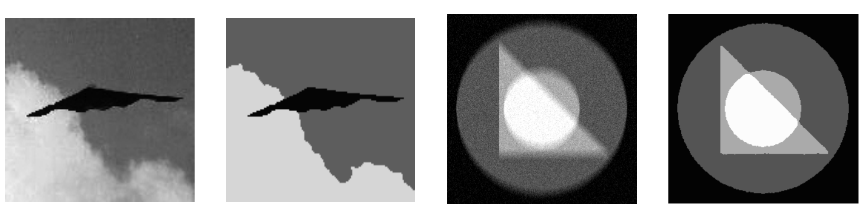
‡ Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong. Email: zeng@hkbu.edu.hk. This work is partially supported by RGC 211710, RGC 211911 and RFGs of HKBU.

Over the years, people have tried to simplify the model (1.1). For example, if we restrict $\nabla g \equiv 0$ on $\Omega \setminus \Gamma$, then this results in a piecewise constant Mumford-Shah model. Recently, in [36] a convex relaxation approach was proposed to solve it. In [14], the method of active contours without edges (Chan-Vese model) was introduced. It actually solves the piecewise constant Mumford-Shah model but restricting the solution to be a piecewise constant solution with only two constants. Multiphase Chan-Vese segmentation methods were proposed thereafter, see [39, 40]. These methods work well for certain image segmentation tasks, for example the cartoon image. However, the main drawback of these methods is that they can easily get stuck in local minima. In order to overcome this problem, convex relaxation approaches [6, 12] and the graph cut method [24] were proposed. There are also many other models based on the Chan-Vese model [14, 39], for example, two-phase segmentation algorithms in [17, 42, 43] and multiphase segmentation algorithms in [2, 7, 28, 29, 30, 38, 44]. Specifically, in [28], the multiphase Chan-Vese model was convexified by using fuzzy membership functions. In [38], a new regularization term was introduced which allows choosing the number of phases automatically. In [42, 43, 44], efficient methods based on the fast continuous max-flow method were proposed. In [17], the length term was replaced by a term involving framelets. The interested readers can read the references therein or [3] for more details.

In this paper, instead of tackling the challenging problem of finding an accurate piecewise smooth solution for the Mumford-Shah model (1.1), we propose to approximate the model by the model:

$$\inf_{g} \left\{ \frac{\lambda}{2} \int_{\Omega} (f - \mathcal{A}g)^2 dx + \frac{\mu}{2} \int_{\Omega} |\nabla g|^2 dx + \int_{\Omega} |\nabla g| dx \right\}, \tag{1.2}$$

where $\mathcal{A}$ can be the identity operator or a blurring operator. We will see that our model is closely related to (1.1). We remark that our model is convex and the solution g is a smooth function. Once g is found, then the segmentation is obtained by segmenting g using properly chosen threshold(s). To segment g into K segments, $K \geq 2$, we require $(K-1)$ thresholds which the users can define themselves or obtain automatically by the K-means method [25, 32]. Figure 1.1 shows two multiphase segmentation results from our method using thresholds from the K-means method.



(a) given image (b) three phases (a) given image (b) four phases

FIG. 1.1. *Multiphase segmentation results by our method.*

We will prove that under mild condition, our model has one and only one solution g which can be solved very fast by currently popular algorithms such as the split-Bregman algorithm [23] or the Chambolle-Pock method [11, 35]. One nice aspect of our method is that there is no need to recompute g when we change the thresholds to reveal different features in the image. Another nice aspect is that there is no need to specify K before finding g. We can obtain any K-phase segmentation by choosing $(K-1)$ thresholds after g is computed. In contrast, multiphase methods such as

[28, 44] require K to be given first; and, if K changes, the minimization problem has to be solved again.

Our tests in Section 4 show that our method can segment different kinds of images: anti-mass images, tubular images, images with very high noise, and images with blur and noise. For the last one, all the multiphase methods we tested [28, 38, 44] fail while our method can provide a very good result, see Figures 1.1 or 4.8. We will see that our method is fast comparing to popular two-phase segmentation methods [8, 14, 17, 43] and multiphase segmentation methods [28, 38, 44].

Note that once g is obtained and the thresholds are given, segmenting g into K segments require very little time. In fact, the complexity is proportional to the number of pixels in the image. Hence our method is quite suitable for users to play around with different thresholds to determine the number of segments they prefer and the different features within the image they like to reveal. However, we also provide a K-means method to compute the thresholds automatically for users who prefer an automated K-phase segmentation algorithm.

Our model provides a better understanding on the link between image segmentation and image restoration. Indeed, the effectiveness of our method suggests that for segmentation, a key idea is to extract the cartoon part in the image, i.e. g; and then cluster g into different phases. Based on this idea, it is likely that a more efficient segmentation method can be developed in the future along this line.

The rest of the paper is organized as follows. In Section 2, we derive our convex model (1.2) which is based on the Mumford-Shah model. We then show that our model has a unique solution. Then we discuss the relationship between our model with models in image restoration. In Section 3, we give the detail implementation of our method, and show that the resulting algorithm converges. In Section 4, we compare our method on various synthetic and real images with two-phase segmentation algorithms [17, 14, 43] and multiphase segmentation methods [28, 38, 44]. Conclusions are given in Section 5.

2. Our model.

Our model is motivated by the following simple but important observation about binary images: a binary image can be recovered quite well from its smoothed version by thresholding with a proper threshold. Figure 2.1 is an example to illustrate this point. Figure 2.1(a) is the true binary image and 2.1(b) is its smoothed version obtained by a Gaussian filter with size [5,5] and standard deviation 3. Obviously, pixels values near the boundary are smoothed. However, by using a threshold of 0.5 to threshold Figure 2.1(b) back to a binary image, we obtain Figure 2.1(c). We see that all the pixels of Figure 2.1(a) except some on the boundary are correctly recovered, see the difference image in Figure 2.1(d). Inspired by this idea, we will modify model (1.1) step by step to arrive at our model (1.2). Briefly speaking, we will use a smooth function to approximate the piecewise smooth solution of model (1.1), then apply a simple thresholding strategy to carry out the segmentation.

2.1. Derivation of our model.

Let $\Sigma = \overline{\text{Inside}(\Gamma)}$, then $\Gamma = \partial\Sigma$. We can rewrite model (1.1) as:

$$\begin{aligned} E^*_{\text{MS}}(\Sigma, g_1, g_2) := \frac{\lambda}{2}\int_{\Sigma\setminus\Gamma}(f-g_1)^2dx + \frac{\mu}{2}\int_{\Sigma\setminus\Gamma}|\nabla g_1|^2dx + \frac{\lambda}{2}\int_{\Omega\setminus\Sigma}(f-g_2)^2dx \\ + \frac{\mu}{2}\int_{\Omega\setminus\Sigma}|\nabla g_2|^2dx + \text{Per}(\Sigma), \end{aligned} \tag{2.1}$$

(a) true (b) given (c) recovered (d) difference

FIG. 2.1. *Segmentation from smooth image. (a): true 128×128 binary image; (b): given smoothed image of (a) by a Gaussian filter; (c): segmented binary result from (b) using threshold 0.5; (d): the difference image (a) and (c) where nonzero pixel values are scaled to 1 to reveal them clearly.*

where g_1 and g_2 are defined on $\Sigma \setminus \Gamma$ and $\Omega \setminus \Sigma$ respectively, and $\mathrm{Per}(\cdot)$ denotes the perimeter of Σ, i.e. $\mathrm{Per}(\Sigma) = \mathrm{Length}(\Gamma)$. Note that (2.1) is similar to Equation (9) in [12]. Observe that once Σ is fixed, then g_1 and g_2 are determined by the following two minimization problems:

$$\inf_{g_1 \in W^{1,2}(\Sigma\setminus\Gamma)} \left\{ \lambda \int_{\Sigma\setminus\Gamma} (f - g_1)^2 dx + \mu \int_{\Sigma\setminus\Gamma} |\nabla g_1|^2 dx \right\} \tag{2.2}$$

and

$$\inf_{g_2 \in W^{1,2}(\Omega\setminus\Sigma)} \left\{ \lambda \int_{\Omega\setminus\Sigma} (f - g_2)^2 dx + \mu \int_{\Omega\setminus\Sigma} |\nabla g_2|^2 dx \right\}. \tag{2.3}$$

For the definition of $W^{1,2}(\Omega)$, see [19, Chapter 5]. The existence and uniqueness of the solutions g_1 and g_2 are guaranteed by the following proposition.

PROPOSITION 2.1. *Let $f \in L^2(\Omega)$. Then the two minimization problems (2.2) and (2.3) have unique minimizers.*

Proof. Since Σ is closed, both the sets $\Omega \setminus \Sigma$ and $\Sigma \setminus \Gamma$ are open. Using the conclusions of Proposition 1 in [3] or Proposition 3 in [16], we conclude that problems (2.2) and (2.3) have unique minimizers. ∎

From the above analysis we can conclude that once the boundary Γ is fixed, i.e. Σ is fixed, then g_1 and g_2 are determined uniquely. Note that in [12], the Chan-Vese model is made convex once the mean values of f inside and outside Γ are fixed. Here, motivated by Theorem 2 of [12], we can derive and prove a similar theorem as follows for the Mumford-Shah model (2.1) once g_1 and g_2 are fixed and smoothly extended to the whole Ω.

THEOREM 2.2. *For any given fixed functions g_1 and $g_2 \in W^{1,2}(\Omega)$, a global minimizer for $E^*_{\mathrm{MS}}(\Sigma, g_1, g_2)$ in (2.1) can be found by carrying out the following convex minimization:*

$$\min_{0\le u\le 1} \left\{ \int_\Omega |\nabla u| + \frac{1}{2} \int_\Omega \left\{ \lambda(f - g_1)^2 + \mu|\nabla g_1|^2 - \lambda(f - g_2)^2 - \mu|\nabla g_2|^2 \right\} u(x) \right\}, \tag{2.4}$$

and setting $\Sigma = \{x : u(x) \ge \rho\}$ for a.e. $\rho \in [0, 1]$.

Proof. See Appendix I. ∎

From Theorem 2.2, we see that the term $\mathrm{Per}(\Sigma)$ of (2.1) is replaced by a convex integral term $\int_\Omega |\nabla u|$. In other words, the boundary information of Γ in (1.1) can be extracted from the TV-term $\int_\Omega |\nabla u|$. This motivates us to use $\int_\Omega |\nabla g|$ to extract the boundary information Length(Γ) in (1.1). Evidently, this approximation is also related to the fuzzy membership approach [6, 12, 28] to handle the Chan-Vese model. In the following, we therefore use $\int_\Omega |\nabla g|$ to approximate the boundary term (the last term) in the Mumford-Shah energy (1.1).

Next we consider simplifying the middle term in model (1.1). In (1.1), the solution is restricted to be a smooth function in $\Omega \setminus \Sigma$ and in $\Sigma \setminus \Gamma$. However, from the example given in Figure 1, we see that these smooth parts can be recovered quite well from a smooth function g in Ω by a proper thresholding. Therefore in the following, we look for solution $g \in W^{1,2}(\Omega)$. Then we have:

LEMMA 2.3. *If $g \in W^{1,2}(\Omega)$ and Γ is a closed curve with $m(\Gamma) = 0$, where $m(\cdot)$ is the Lebesgue measure, then $\int_\Gamma |\nabla g|^2 dx = 0$.*

Proof. Since $g \in W^{1,2}(\Omega)$, we have $\nabla g \in L^2(\Omega)$. Because of $m(\Gamma) = 0$, we get $\int_\Gamma |\nabla g|^2 dx = 0$ immediately. □

Thus the middle term of model (1.1) becomes:

$$\int_{\Omega\setminus\Gamma} |\nabla g|^2 dx = \int_\Omega |\nabla g|^2 dx - \int_\Gamma |\nabla g|^2 dx = \int_\Omega |\nabla g|^2 dx, \quad \forall g \in W^{1,2}(\Omega). \tag{2.5}$$

In view of Theorem 2.2 and (2.5), we propose our segmentation model as:

$$\inf_{g\in W^{1,2}(\Omega)} \left\{ \frac{\lambda}{2} \int_\Omega (f-g)^2 dx + \frac{\mu}{2} \int_\Omega |\nabla g|^2 dx + \int_\Omega |\nabla g| dx \right\},$$

where λ and μ are positive parameters. Since sometimes the given image is degraded by noise and/or blur, we extend this model to general cases by introducing a problem-related operator $\mathcal{A}$ in its fidelity term. Then finally our segmentation model is:

$$\inf_{g\in W^{1,2}(\Omega)} E(g) := \inf_{g\in W^{1,2}(\Omega)} \left\{ \frac{\lambda}{2} \int_\Omega (f-\mathcal{A}g)^2 dx + \frac{\mu}{2} \int_\Omega |\nabla g|^2 dx + \int_\Omega |\nabla g| dx \right\}, \tag{2.6}$$

where $\mathcal{A}$ may stand for the identity operator or a blurring operator. Obviously, if $\mu \neq 0$ in (2.6), g will be smooth. The following theorem shows the existence and uniqueness of g.

THEOREM 2.4. *Let Ω be a bounded connected open subset of $\mathbb{R}^2$ with a Lipschitz boundary. Let $f \in L^2(\Omega)$ and $\mathrm{Ker}(\mathcal{A}) \bigcap \mathrm{Ker}(\nabla) = \{0\}$[1], where $\mathcal{A}$ is a bounded linear operator from $L^2(\Omega)$ to itself and $\mathrm{Ker}(\mathcal{A})$ is the kernel of $\mathcal{A}$. Then (2.6) has a unique minimizer $g \in W^{1,2}(\Omega)$.*

Proof. See Appendix II. □

Lastly, we emphasize that model (2.6) can be minimized quickly by using currently available efficient algorithms such as the split-Bregman algorithm [23] or the Chambolle-Pock method [11, 35]. We leave the implementation to Section 3.

2.2. Relationship with image restoration. It is interesting to note that our model (2.6) itself can be regarded as an image restoration model to capture the cartoon part in the image and is closely related to the classical ROF image restoration model:

$$\inf_g \int_\Omega \left(\frac{\lambda}{2} (f-\mathcal{A}g)^2 dx + |\nabla g| \right), \tag{2.7}$$

[1] This condition actually restricts $\mathcal{A}1 \neq 0$. It means that $\mathcal{A}f \neq 0$ if f is a nonzero constant image. It is true for all blurring operators as they are convolution operators with positive kernels.

see [37]. The only difference is that we have an extra term $\int_\Omega |\nabla g|^2$. One of the important properties of the ROF model is that it can preserve important edge information but the staircase effect may be introduced. In order to avoid this, many works have been proposed, see [13, 31, 41, 5] for examples. In [13], Chan, Marquina and Mulet proposed to solve the following minimization problem:

$$\inf_g \int_\Omega \left(\frac{\lambda}{2}(f-g)^2 + |\nabla g|_{\epsilon_1} + \mu \frac{(\Delta g)^2}{|\nabla g|^3_{\epsilon_2}} \right), \tag{2.8}$$

where $|\nabla g|_{\epsilon_i} = \sqrt{|\nabla g|^2 + \epsilon_i}$, $i = 1, 2$ with ϵ_i being small positive parameters. The additional higher-order derivative term can remove the staircase effect. In [31, 41], the authors used second-order derivatives to replace the TV regularization term of model (2.7). Recently a novel regularization model, the total generalized variation, was proposed in [5] which also involves higher-order derivatives. Obviously, the cost and difficulty of solving the given models grow as the functionals became more and more complex.

In contrast, in our model (2.6), the staircase effect is reduced because of the middle term which contains the square of the first-order derivative and no other higher-order derivatives. Once the smooth solution g is found, a suitable thresholding then gives the image segmentation result. From the analysis of Section 2.1 and the numerical results in the coming Section 4, we see that the smoothness in g actually does not affect the segmentation significantly.

3. Numerical aspects. In this section, we first introduce the split-Bregman algorithm for solving our model (2.6). After that we give a strategy based on the K-means method to determine the thresholds automatically.

3.1. Solution of our segmentation model. The discrete setting of our model (2.6) is:

$$\min_g \left\{ \frac{\lambda}{2}\|f - \mathcal{A}g\|_2^2 + \frac{\mu}{2}\|\nabla g\|_2^2 + \|\nabla g\|_1 \right\}, \tag{3.1}$$

where $\|\nabla g\|_1 := \sum_{i\in\Omega} \sqrt{(\nabla_x g)_i^2 + (\nabla_y g)_i^2}$ is the classical discrete TV semi-norm. Here we adopt the backward difference with periodic boundary condition to approximate the discrete gradient operator ∇, i.e. for the first row of g, we define:

$$(\nabla_x g)_i = \begin{cases} g(1,1) - g(1,n), & i = 1, \\ g(1,i) - g(1,i-1), & i = 2, \cdots, n, \end{cases}$$

where n is the number of pixels of the first row of g and $g(1,i)$ represents the ith pixel of the first row of g . Similarly, we can define ∇_y. As (3.1) is convex, it can be solved by many methods such as the alternating direction method of multipliers which is convergent and is well-suited to distributed convex optimization, see [4, 20] and references therein. Specifically, its variant, the split-Bregman algorithm [23], is used widely to solve a very broad class of L_1 regularization problems. We can also use the Chambolle-Pock method [11, 35, 26] which provides convergence rate. In the following, we derive the split-Bregman algorithm of solving (3.1). Clearly the algorithm converges, since our model (3.1) is a kind of convex regularization problem, see [4, 20, 23] for more details of the convergence analysis.

Set $d_x = \nabla_x g$ and $d_y = \nabla_y g$ in (3.1), and this yields the constrained problem:

$$\min_g \left\{ \frac{\lambda}{2}\|f - \mathcal{A}g\|_2^2 + \frac{\mu}{2}\|\nabla g\|_2^2 + \|(d_x, d_y)\|_1 \right\}, \quad \text{s.t. } d_x = \nabla_x g \text{ and } d_y = \nabla_y g.$$

Using 2-norm to weakly enforce the above constraints, it becomes:

$$\min_{g,d_x,d_y} \left\{ \frac{\lambda}{2}\|f - \mathcal{A}g\|_2^2 + \frac{\mu}{2}\|\nabla g\|_2^2 + \|(d_x, d_y)\|_1 + \frac{\sigma}{2}\|d_x - \nabla_x g\|_2^2 + \frac{\sigma}{2}\|d_y - \nabla_y g\|_2^2 \right\}.$$

Applying the split-Bregman iteration to strictly enforce the constraints, we have at step $(k+1)$,

$$\begin{aligned}(g^{k+1}, d_x^{k+1}, d_y^{k+1}) = \arg\min_{g,d_x,d_y} \Big\{ &\frac{\lambda}{2}\|f - \mathcal{A}g\|_2^2 + \frac{\mu}{2}\|\nabla g\|_2^2 + \|(d_x, d_y)\|_1 \\ &+ \frac{\sigma}{2}\|d_x - \nabla_x g - b_x^k\|_2^2 + \frac{\sigma}{2}\|d_y - \nabla_y g - b_y^k\|_2^2 \Big\},\end{aligned} \tag{3.2}$$

$$b_x^{k+1} = b_x^k + (\nabla_x g^{k+1} - d_x^{k+1}), \quad b_y^{k+1} = b_y^k + (\nabla_y g^{k+1} - d_y^{k+1}). \tag{3.3}$$

The minimization (3.2) can be solved effectively by minimizing with respect to g and (d_x, d_y) alternatively. Hence we need to solve the following two minimization subproblems:

$$\begin{aligned} g^{k+1} = \arg\min_g \Big\{ &\frac{\lambda}{2}\|f - \mathcal{A}g\|_2^2 + \frac{\mu}{2}\|\nabla g\|_2^2 + \frac{\sigma}{2}\|d_x^k - \nabla_x g - b_x^k\|_2^2 \\ &+ \frac{\sigma}{2}\|d_y^k - \nabla_y g - b_y^k\|_2^2 \Big\}, \end{aligned} \tag{3.4}$$

$$\begin{aligned} (d_x^{k+1}, d_y^{k+1}) = \arg\min_{d_x,d_y} \Big\{ &\|(d_x, d_y)\|_1 + \frac{\sigma}{2}\|d_x - \nabla_x g^{k+1} - b_x^k\|_2^2 \\ &+ \frac{\sigma}{2}\|d_y - \nabla_y g^{k+1} - b_y^k\|_2^2 \Big\}. \end{aligned} \tag{3.5}$$

Since the right hand side of (3.4) is differentiable, g^{k+1} satisfies the following optimality condition:

$$(\lambda\mathcal{A}^*\mathcal{A} - (\mu + \sigma)\Delta)g = \lambda\mathcal{A}^* f + \sigma\nabla_x^T(d_x^k - b_x^k) + \sigma\nabla_y^T(d_y^k - b_y^k), \tag{3.6}$$

where $\mathcal{A}^*$ is the conjugate transpose of $\mathcal{A}$ and $\Delta = -(\nabla_x^T\nabla_x + \nabla_y^T\nabla_y)$. Since $\mathrm{Ker}(\mathcal{A}) \bigcap \mathrm{Ker}(\Delta) = \{0\}$, the matrix $[\lambda\mathcal{A}^*\mathcal{A} - (\mu + \sigma)\Delta]$ is positive definite and hence is invertible. Using the Gauss-Seidel method in [23] or the Fast Fourier Transforms to diagonalize the circulant matrices $\mathcal{A}$ and Δ (see [15]), equation (3.6) can be solved efficiently. For problem (3.5), it can be solved explicitly using a generalized shrinkage formula [23] as follows:

$$d_x^{k+1} = \max\left(s^k - \frac{1}{\sigma}, 0\right)\frac{s_x^k}{s^k}, \quad d_y^{k+1} = \max\left(s^k - \frac{1}{\sigma}, 0\right)\frac{s_y^k}{s^k}, \tag{3.7}$$

where $s_x^k = \nabla_x g^{k+1} + b_x^k$, $s_y^k = \nabla_y g^{k+1} + b_y^k$ and $s^k = \sqrt{(s_x^k)^2 + (s_y^k)^2}$. The following algorithm summarizes the procedure of solving our minimization problem (3.1).

Algorithm 1: Solving (3.1) by the split-Bregman algorithm

1. Initialize: $g^0 = f, d_x^0 = d_y^0 = b_x^0 = b_y^0 = 0$.
2. Do $k = 0, 1, \ldots,$ until $\frac{\|g^k - g^{k+1}\|_F}{\|g^{k+1}\|_F} < \epsilon$
   (a) Compute g^{k+1} by solving (3.6).
   (b) Compute d_x^{k+1} and d_y^{k+1} by the shrinkage formula (3.7).
   (c) Update b_x^{k+1} and b_y^{k+1} by the formula (3.3).
3. Output: g.

3.2. Determining the thresholds. As mentioned before, our segmentation result is obtained by thresholding the solution g of (3.1) with proper threshold(s) ρ. For example, for two-phase segmentation, one may choose ρ to be the mean value of g, and then use this ρ to threshold g into two phases. Or the user can try different values of ρ's to get the best result. Note that *there is no need to recompute the image g when we change ρ.* We just threshold the image g with the new ρ to get a new binary image and obtain the corresponding boundary using for example the CONTOUR command in MATLAB.

In case one wants to choose the thresholds automatically, here we discuss how to choose them using the K-means method [25, 27, 32]. To standardize the discussions, we begin by normalizing the pixel values of g to [0,1]. We do this by using the linear-stretch formula:

$$\bar{g} = \frac{g - g_{\min}}{g_{\max} - g_{\min}}, \tag{3.8}$$

where $g_{\max}$ and $g_{\min}$ represent maximum and minimum of g respectively.

The K-means method is a very efficient method to classify a given set into K clusters, with K specified in advance. Suppose we want to segment $\bar{g}$ into K segments, $K \geq 2$. We use the K-means method to classify the set of pixels values of $\bar{g}$ into K clusters. Let the mean value of each cluster be $\hat{\rho}_1, \hat{\rho}_2, \cdots, \hat{\rho}_K$, and without loss of generality, let $\hat{\rho}_1 \leq \hat{\rho}_2 \leq \cdots \leq \hat{\rho}_K$. Then we define the $(K-1)$ thresholds as:

$$\rho_i = \frac{\hat{\rho}_i + \hat{\rho}_{i+1}}{2}, \quad i = 1, 2, \ldots, K-1. \tag{3.9}$$

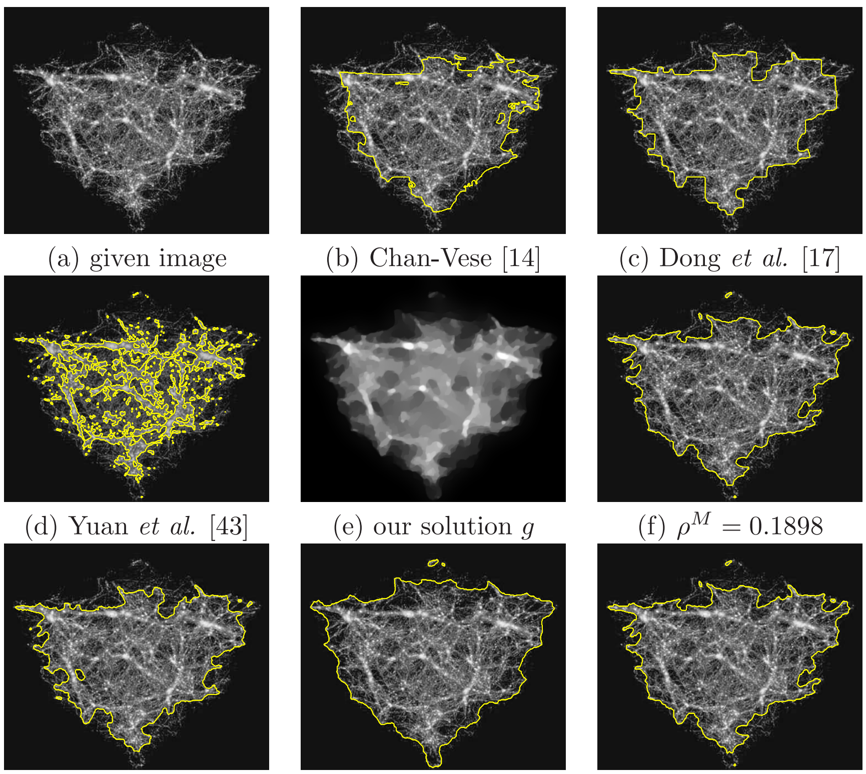
The ith phase of $\bar{g}$, $1 \leq i \leq K$, is then given by $\{x \in \Omega : \rho_{i-1} < \bar{g}(x) \leq \rho_i\}$. To obtain the boundary of the ith phase, we set pixels in the ith phase to 1 and all the other pixels to zero; then we invoke the command CONTOUR in MATLAB. Again we emphasize that if we change the thresholds, there is *no need to recompute g or $\bar{g}$.*

4. Experimental results. In this section, we compare our segmentation model (3.1) with the two-phase segmentation methods in [14, 17, 43] and the multiphase segmentation methods in [28, 38, 44]. Methods [14] and [17] use TV and framelets regularization terms respectively; therefore, we can compare the performance of these two different regularization approaches with ours. Methods [28, 38, 43, 44] are effective segmentation methods proposed recently. The codes we used are provided by the authors. Apart from some default settings, like the maximum number of iterations, the parameters in the codes are chosen by trial and error to give the best results of the respective methods.

For two-phase segmentation, we use ρ^M, ρ_1 and ρ^U to denote the thresholds we used in the test. They represent respectively the mean of the normalized image $\bar{g}$ given in (3.8), the threshold obtained by K-means given in (3.9), and a threshold chosen by us. For multiphase segmentation, we use the thresholds ρ_i's defined in (3.9). The tolerance ϵ and the step size σ used in the split-Bregman algorithm in (3.2) were fixed to be 10^{-4} and 2 respectively. The parameters λ, μ are chosen empirically. All the results were tested on a MacBook with 2.4 GHz processor and 4GB RAM. The boundaries of all the results are shown with color and superimposed on the given images.

4.1. Two-phase segmentation. *Example 1: Anti-mass image.* Figure 4.1(a) is the given image. Figures 4.1(b)–(d) are the results of methods [14, 17, 43] respectively.

Figure 4.1(e) is our smooth solution g from Algorithm 1 using parameters $\lambda = 3$ and $\mu = 1$, see (3.1). Figures 4.1(f)–(i) are the segmentation results on the normalized $\bar{g}$ (see (3.8)) with thresholds $\rho^M = 0.1898$, $\rho_1 = 0.2669$ and $\rho^U = 0.1, 0.2$ respectively. Note that ρ^M and ρ_1 are computed automatically. From the results, we see that our method can reveal different meaningful features in the image by choosing different ρ's; and this can be done without recomputing g. In contrast, for the methods [14, 17, 43], one will need to solve the minimization models again if one wants to adjust the parameters to obtain different features in the image.

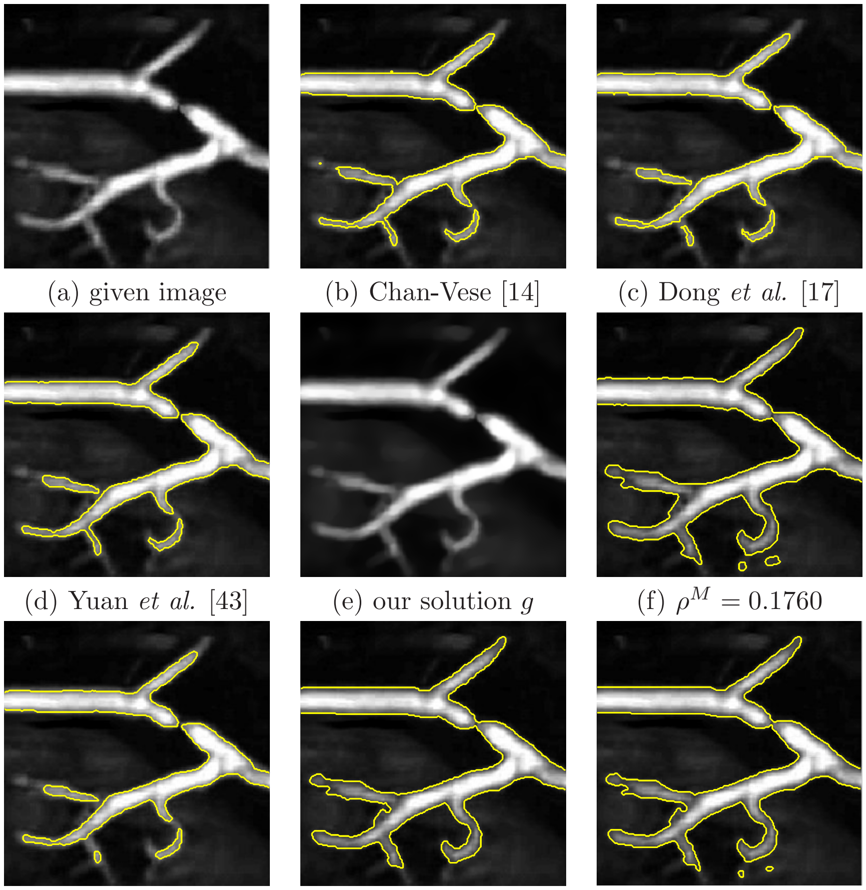


(a) given image; (b) Chan-Vese [14]; (c) Dong *et al.* [17]; (d) Yuan *et al.* [43]; (e) our solution g; (f) $\rho^M = 0.1898$; (g) $\rho_1 = 0.2669$; (h) $\rho^U = 0.1$; (i) $\rho^U = 0.2$

Fig. 4.1. *Anti-mass image segmentation. (a): given* 384×480 *image; (b)–(d): results of methods [14], [17] and [43] respectively; (e): our smooth solution g; (f)–(i): our segmentation results using thresholds* $\rho^M = 0.1898$, $\rho_1 = 0.2669$, $\rho^U = 0.1$ *and* 0.2 *respectively.*

Example 2: Tubular image. Figure 4.2(a) is a given Magnetic Resonance Angiography kidney image [21]. The boundaries of the vessels are blurry and vague so that they are hard to be detected. Figure 4.2(e) is the solution g from Algorithm 1 using $\lambda = 20$ and $\mu = 1$. Figures 4.2(f)–(h) are our segmentation results with thresholds $\rho^M = 0.1760, \rho_1 = 0.4019$ and $\rho^U = 0.2$ respectively. By comparing our results with the results from methods [14, 17, 43] in Figures 4.2(b)–(d), we see that our method can better detect and connect the blood vessels. Recently, we proposed a tight-frame method specifically for segmenting vessels [8]. Here we give the result of this method in Figure 4.2(i). We see that it is comparable to our method.

Example 3: Image with high noise. In Figures 4.3(a) and (b), we give the clean and the noisy images respectively. The noise we added is high: Gaussian noise with



(a) given image (b) Chan-Vese [14] (c) Dong *et al.* [17]

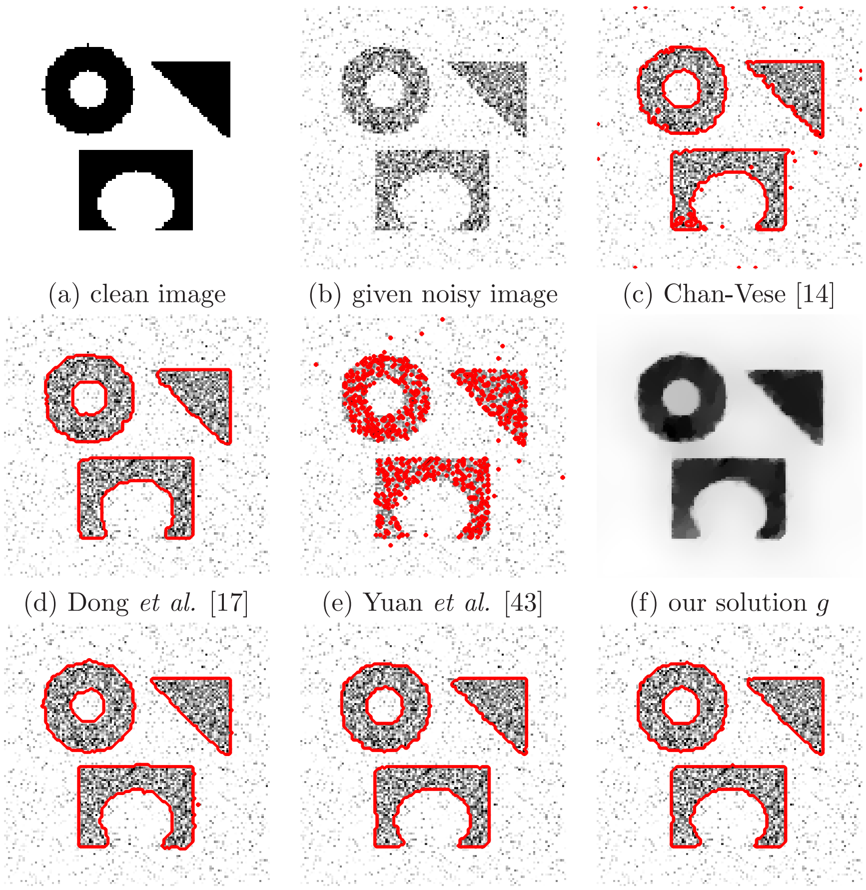
(d) Yuan *et al.* [43] (e) our solution g (f) $\rho^M = 0.1760$

(g) $\rho_1 = 0.4019$ (h) $\rho^U = 0.2$ (i) Cai *et al.* [8]

FIG. 4.2. *Kidney vascular system segmentation. (a): given* 256×256 *image; (b)–(d): results of methods [14], [17] and [43] respectively; (e): our solution g; (f)–(h): our segmentation of results using thresholds* $\rho^M = 0.1760$, $\rho_1 = 0.4019$ *and* $\rho^U = 0.2$ *respectively; (i) result of method [8].*

mean 0.6 and variance 0.25. Figures 4.3(c)–(e) give the results of methods [14, 17, 43] on the noisy image respectively. We see that method [17], which uses tight-frame regularization, recovers these objects better than method [14], which uses TV regularization; and that method [43] fails completely. Figure 4.3(f) is our solution g when $\lambda = 4$ and $\mu = 1$. Figures 4.3(g)–(i) are the segmentation results with thresholds $\rho^M = 0.8308, \rho_1 = 0.6371$ and $\rho^U = 0.7$ respectively. Clearly, our results are all good and comparable to Figure 4.3(d). However, our method is much faster (see Table 4.1 below). Notice that the differences between our results (g)–(i) are small, indicating that our method is robust with respect to the threshold.

Example 4: Blurry and noisy image. To illustrate the robustness of our method with respect to the threshold, we tested our method on two blurry images: Figure 4.4 with motion blur and Figure 4.5 with Gaussian blur respectively. For the motion blur, the motion is vertical and the filter size is 15. For the Gaussian blur, the filter used is of size $[15, 15]$ with standard deviation 15. For both images, we added a Gaussian noise with mean 10^{-3} and variance 2×10^{-3}. Figures 4.4 (f) and 4.5 (f) are our solutions g obtained by using $\lambda = 100$ and $\mu = 1$. From Figures 4.4(c)–(e) and



(a) clean image (b) given noisy image (c) Chan-Vese [14]

(d) Dong *et al.* [17] (e) Yuan *et al.* [43] (f) our solution g

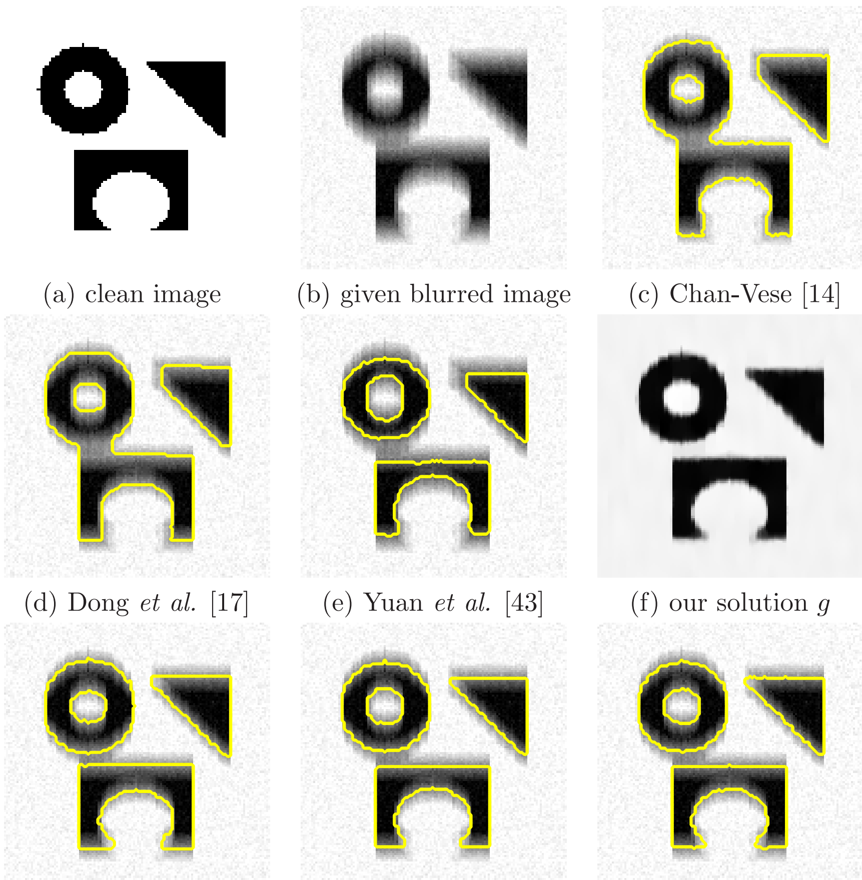
(g) $\rho^M = 0.8308$ (h) $\rho_1 = 0.6371$ (i) $\rho^U = 0.7$

FIG. 4.3. *Noisy image segmentation. (a): clean* 128×128 *image; (b): given noisy image; (c)–(e): results of methods [14], [17] and [43] respectively; (f): our solution g; (g)–(i): our segmentation results using thresholds* $\rho^M = 0.8308$, $\rho_1 = 0.6371$ *and* $\rho^U = 0.7$ *respectively.*

4.5(c)–(e), which are the results of methods [14, 17, 43], we see that all of them are not good. More precisely, methods [14, 17] give incorrect boundaries (linking the ring and the horseshoe objects together) while method [43] misses a large portion of the objects. In contrast, our boundary recovers the shapes of the objects very well, see Figures 4.4(g)–(i) and 4.5(g)–(i).

Table 4.1 gives the CPU time comparison of the methods. We see that our method is second only to the two-phase continuous max-flow method in [43]. But from Examples 1–4, we see that our method gives much better segmentation results than the method in [43]. We remark that for the examples we tested, the framelet method in [17] did not converge within the maximum number of iterations (300) using the given tolerance 10^{-3} specified in the code.

4.2. Multiphase segmentation. *Example 5: Three-phase image.* Figure 4.6(a) is the given image and Figures 4.6(b)–(d) are the three-phase segmentation results by methods [28, 38, 44]. Figure 4.6(e) is our solution g obtained with $\lambda = 30$ and $\mu = 0.1$. Figures 4.6(g)–(i) are the boundaries of the three phases obtained from $\bar{g}$



(a) clean image (b) given blurred image (c) Chan-Vese [14]

(d) Dong *et al.* [17] (e) Yuan *et al.* [43] (f) our solution g

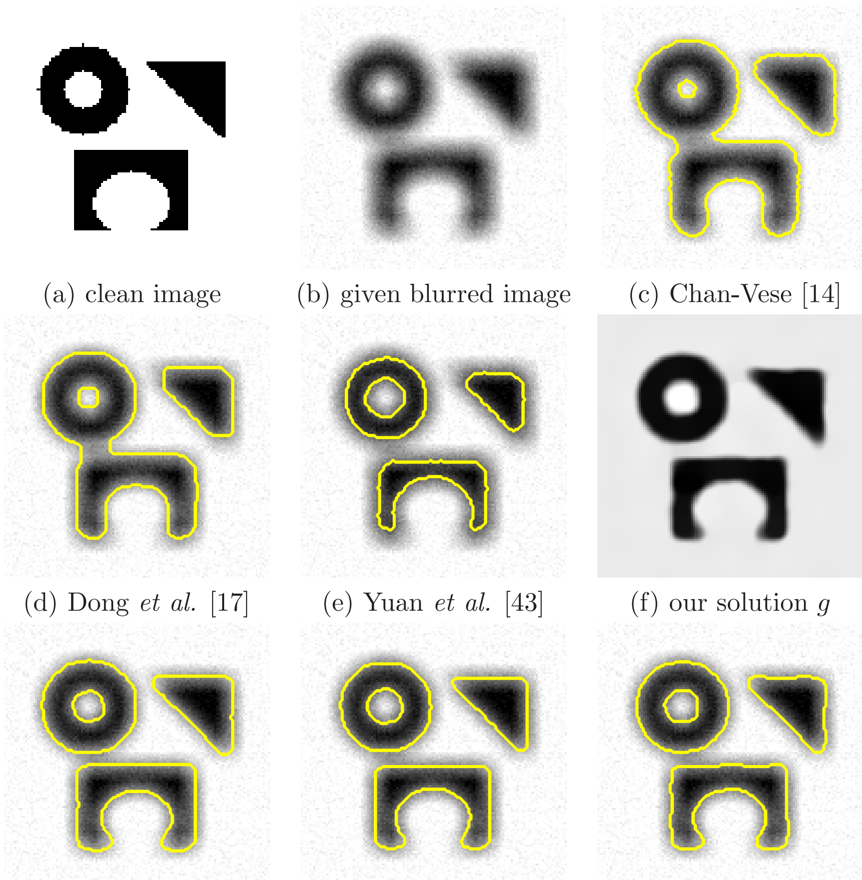
(g) $\rho^M = 0.7661$ (h) $\rho_1 = 0.5048$ (i) $\rho^U = 0.6$

FIG. 4.4. *Segmentation of motion blurred image. (a): clean* 128×128 *image; (b): given blurred and noisy image; (c)–(e): results of methods [14], [17] and [43] respectively; (f): our solution* g*; (g)–(i): our results using thresholds* $\rho^M = 0.7661$*,* $\rho_1 = 0.5048$ *and* $\rho^U = 0.6$ *respectively.*

TABLE 4.1
Iteration numbers and CPU time in second for two-phase segmentation.

	Chan-Vese [14]		Dong [17]		Yuan [43]		Our method	
Example	iter.	time	iter.	time	iter.	time	iter.	time
Figure 4.1	1000	263.73	300	83.82	64	6.01	172	18.38
Figure 4.2	1000	76.62	300	32.17	18	0.37	115	3.03
Figure 4.3	1000	23.42	300	10.18	108	0.42	63	0.49
Figure 4.4	1300	28.19	300	10.18	20	0.09	52	1.13
Figure 4.5	1500	31.78	300	10.18	24	0.10	65	1.21

(defined in (3.8)) using thresholds $\rho_1 = 0.1929$ and $\rho_2 = 0.6009$ which are computed automatically by the K-means method (3.9). Figure 4.6(f) is a trinary representation of the three phases by using the mean value of each phase to represent that phase. We see that all results are good except the result by method [44] (Figure 4.6(d)) which separates the cloud in the lower right corner into two parts. We emphasize that for



(a) clean image (b) given blurred image (c) Chan-Vese [14]

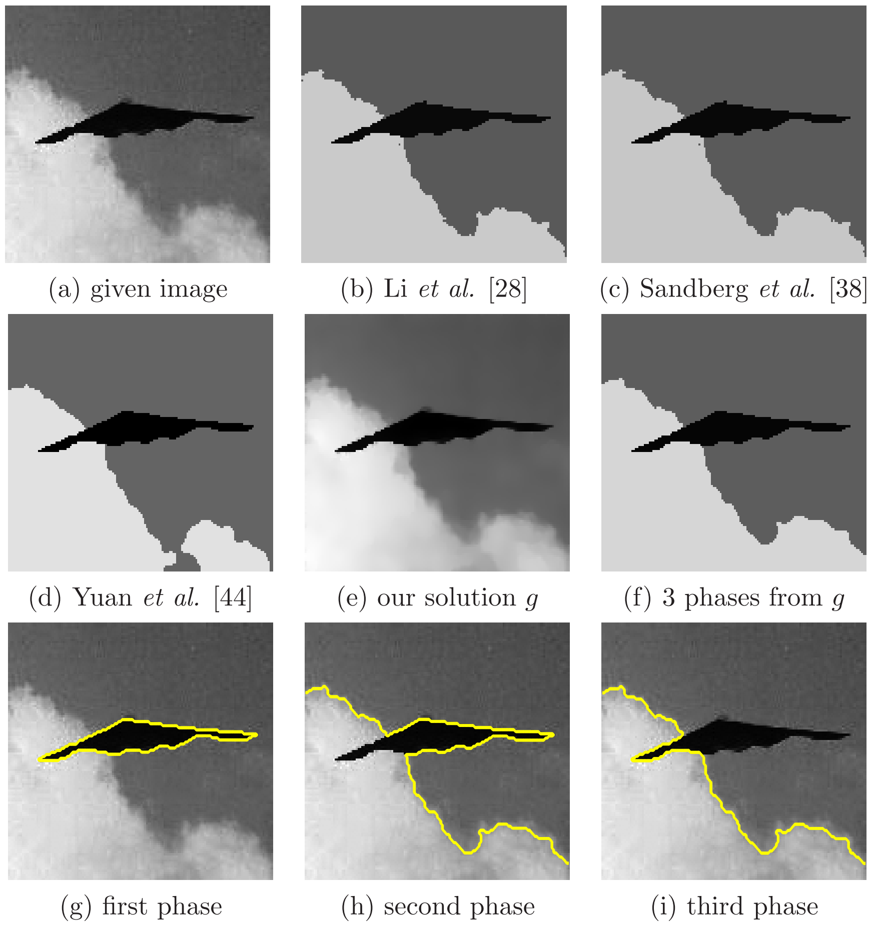
(d) Dong *et al.* [17] (e) Yuan *et al.* [43] (f) our solution g

(g) $\rho^M = 0.7324$ (h) $\rho_1 = 0.5033$ (i) $\rho^U = 0.6$

FIG. 4.5. *Segmentation of Gaussian blurred image. (a): clean* 128×128 *image; (b): given blurred and noisy image; (c)–(e): results of methods [14], [17] and [43] respectively; (f): our solution* g*; (g)–(i): our results using thresholds* $\rho^M = 0.7324$*,* $\rho_1 = 0.5033$ *and* $\rho^U = 0.6$ *respectively.*

our method, we do not need to determine the number of phases K at the beginning. We can modify K after obtaining g, and compute the thresholds $\{\rho_i\}_{i=1}^{K-1}$ by using (3.9) to segment g into K segments. This is not the case for methods [28, 44] where one has to specify K before minimizing their problems. Moreover, we found in our tests that method [28] is sensitive to initialization where different initializations may give quite different results.

Example 6: Four-phase noisy image. Figures 4.7(a) and (b) give the clean and the noisy images (Gaussian noise with zero mean and variance 0.03). Figure 4.7(f) is our solution g obtained by using $\lambda = 4$ and $\mu = 0.1$. The thresholds computed automatically by (3.9) are $\rho_1 = 0.1652, \rho_2 = 0.4978, \rho_3 = 0.8319$. The corresponding four phases segmentation is given in Figure 4.7(f). Figure 4.7(g) shows the four phases of g by using the mean values of each phase to represent the phase, and Figures 4.7(h)–(k) give the boundaries of the phases. We see that the four phases are almost recovered exactly by our method, see Figure 4.7(g). In contrast, method [28] (Figure 4.7(c)) segments one phase incorrectly; method [38] (Figure 4.7(d)) fails; and method [44] (Figure 4.7(e)) gives oscillation boundaries.



(a) given image (b) Li *et al.* [28] (c) Sandberg *et al.* [38]

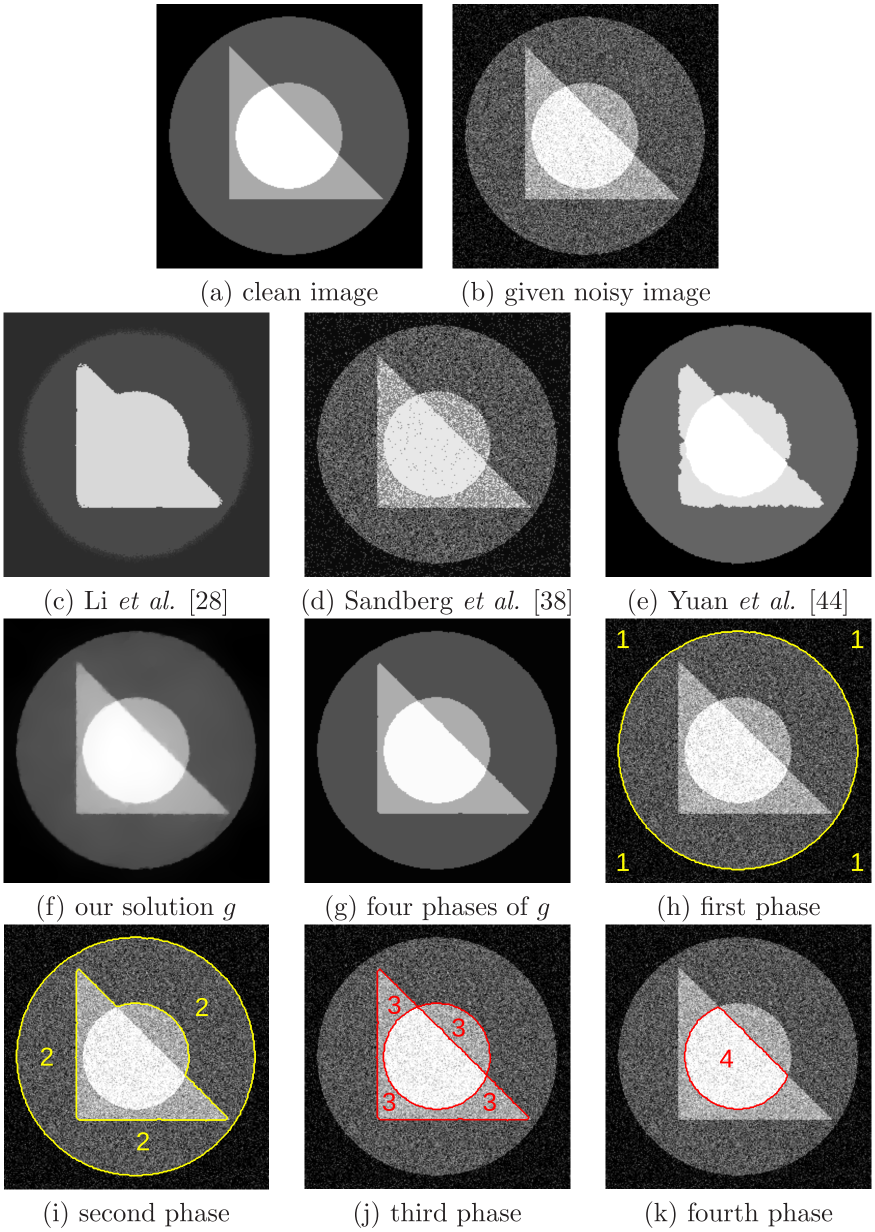
(d) Yuan *et al.* [44] (e) our solution g (f) 3 phases from g

(g) first phase (h) second phase (i) third phase

FIG. 4.6. *Three-phase segmentation. (a): given* 125×150 *image; (b)–(d): results of methods [28], [38] and [44] respectively; (e): our solution g; (f): three phases using thresholds* $\rho_1 = 0.1929, \rho_2 = 0.6009$*; (g)–(i): boundary of each phase of g.*

Example 7: Four-phase blurry and noisy image. The blurry and noisy image used is given in Figure 4.7(b). The blur is motion blur where the motion is vertical and the filter size is 15. The noise is Gaussian noise with mean 10^{-3} and variance 2×10^{-3}. Figure 4.8(f) is our solution g obtained by using $\lambda = 40$ and $\mu = 1$. The thresholds from the K-means method (3.9) are $\rho_1 = 0.1704, \rho_2 = 0.4971, \rho_3 = 0.8248$. Figure 4.8(g) gives the corresponding four phases, and Figures 4.8(h)–(k) give the boundaries of the phases. We see that the four phases of the image are almost recovered exactly by our method, see Figure 4.8(g). But from Figures 4.8(c)–(e), we see that the results of all the other multiphase methods [28, 38, 44] are not good.

Table 4.2 gives the CPU time comparison of the methods. We see that our method is the fastest. Note that method [44] is comparable to ours in time, but from Examples 5–7, we see that our method gives better segmentation. In fact, our model is based on the Mumford-Shah model (1.1) which admits more high order information. But methods [43, 44] are basically using constants to approximate regions. This may explain why they fail in Figures 4.3 (e) and 4.6 (d), and give poor results in others examples.



(a) clean image (b) given noisy image

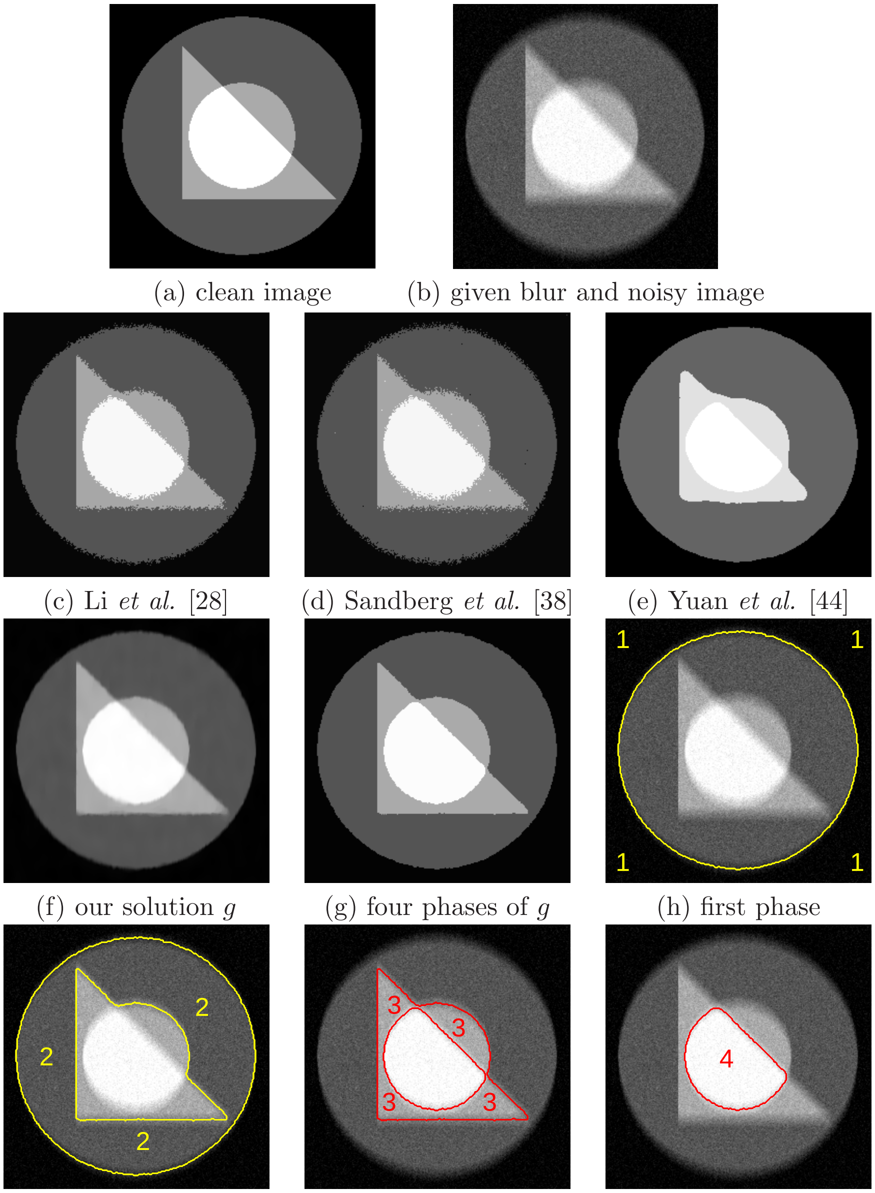
(c) Li *et al.* [28] (d) Sandberg *et al.* [38] (e) Yuan *et al.* [44]

(f) our solution g (g) four phases of g (h) first phase

(i) second phase (j) third phase (k) fourth phase

FIG. 4.7. *Four-phase segmentation for noisy image. (a): clean* 256×256 *image; (b): given noisy image; (c)–(e): results of methods [28], [38] and [44] respectively; (f): our solution g; (g): four phases using thresholds* $\rho_1 = 0.1652, \rho_2 = 0.4978, \rho_3 = 0.8319$*; (h)–(k) boundary of each phase of* g*.*

5. Conclusions. In this paper, we have proposed a *convex* segmentation model based on the Mumford-Shah model. Our method first finds the unique smooth minimizer by the split-Bregman algorithm [23], then uses thresholding strategy to segment the image. Since our model (3.1) can be regarded as an image restoration model, our method unifies the image processing works of image segmentation and image restora-



(a) clean image (b) given blur and noisy image

(c) Li *et al.* [28] (d) Sandberg *et al.* [38] (e) Yuan *et al.* [44]

(f) our solution g (g) four phases of g (h) first phase

(i) second phase (j) third phase (k) fourth phase

FIG. 4.8. *Four-phase segmentation for noisy and blurry image. (a): clean* 256×256 *image; (b): given blurred and noisy image; (c)–(e): results of methods [28], [38] and [44] respectively; (f): our solution* g*; (g): four phases using thresholds* $\rho_1 = 0.1704, \rho_2 = 0.4971, \rho_3 = 0.8248$*; (h)–(k) boundary of each phase of* g*.*

tion. Furthermore, our method combines the two-phase and multiphase segmentation into one single algorithm. In fact, one does not have to specify the phases before finding the solution to the model. One can segment the solution into different phases by choosing proper thresholds after the solution is obtained. We have introduced an effective way based on the K-means method to choose the thresholds automatically.

TABLE 4.2
Iteration numbers and CPU time in second for multiphase segmentation.

	Li [28]		Sandberg [38]		Yuan [44]		Our method	
Example	iter.	time	iter.	time	iter.	time	iter.	time
Figure 4.6	100	1.56	2	3.15	32	0.58	62	0.57
Figure 4.7	100	7.64	12	90.59	134	14.51	112	3.04
Figure 4.8	100	7.26	13	93.79	57	5.82	78	2.90

The experimental results show that our method is very effective and robust for many kinds of images, such as anti-mass, tubular, noisy, or blurry images.

Appendix I: Proof of Theorem 2.2. This proof basically follows the proof of Theorem 2 in [12]. Using the co-area formula and noting that $0 \le u \le 1$, we have $\int_\Omega |\nabla u| = \int_0^1 \mathrm{Per}(\{x : u(x) > \rho\})d\rho$. For the second term in (2.4), we proceed as follows:

$$\begin{aligned}
&\int_\Omega \{\lambda(f-g_1)^2 + \mu|\nabla g_1|^2 - \lambda(f-g_2)^2 - \mu|\nabla g_2|^2\}u(x)\\
=&\int_\Omega \{\lambda(f-g_1)^2 + \mu|\nabla g_1|^2 - \lambda(f-g_2)^2 - \mu|\nabla g_2|^2\}\int_0^1 \mathbf{1}_{[0,u(x)]}(\rho)d\rho dx\\
=&\int_0^1\int_\Omega \{\lambda(f-g_1)^2 + \mu|\nabla g_1|^2 - \lambda(f-g_2)^2 - \mu|\nabla g_2|^2\}\mathbf{1}_{[0,u(x)]}(\rho)dxd\rho\\
=&\int_0^1\int_{\Omega\cap\{x:u(x)>\rho\}} \{\lambda(f-g_1)^2 + \mu|\nabla g_1|^2 - \lambda(f-g_2)^2 - \mu|\nabla g_2|^2\}dxd\rho\\
=&\int_0^1\int_{\Omega\cap\{x:u(x)>\rho\}} \{\lambda(f-g_1)^2 + \mu|\nabla g_1|^2\}dxd\rho - C\\
&\quad+\int_0^1\int_{\Omega\cap\{x:u(x)>\rho\}^c} \{\lambda(f-g_2)^2 + \mu|\nabla g_2|^2\}dxd\rho,
\end{aligned}$$

where $C = \int_\Omega \{\lambda(f-g_2)^2+\mu|\nabla g_2|^2\}dx$ is independent of u. Set $\Sigma(\rho) = \overline{\{x : u(x) > \rho\}}$ and $\Gamma(\rho) = \partial\Sigma(\rho)$, we have

$$\begin{aligned}
&\int_\Omega |\nabla u| + \frac{1}{2}\int_\Omega \{\lambda(f-g_1)^2 + \mu|\nabla g_1|^2 - \lambda(f-g_2)^2 - \mu|\nabla g_2|^2\}u(x) \qquad (5.1)\\
=&\int_0^1 \mathrm{Per}(\Sigma(\rho))d\rho + \frac{1}{2}\int_0^1\int_{\Sigma(\rho)\setminus\Gamma(\rho)} \{\lambda(f-g_1)^2 + \mu|\nabla g_1|^2\}dxd\rho\\
&\quad+\frac{1}{2}\int_0^1\int_{\Omega\setminus\Sigma(\rho)} \{\lambda(f-g_2)^2 + \mu|\nabla g_2|^2\}dxd\rho - \frac{C}{2}\\
=&\int_0^1 E^*_{\mathrm{MS}}(\Sigma(\rho), g_1, g_2)d\rho - \frac{C}{2},
\end{aligned}$$

where $E^*_{\mathrm{MS}}(\Sigma(\rho), g_1, g_2)$ is given in (2.1). Hence, if $u(x)$ is a minimizer of the convex problem in (5.1), then the set $\Sigma(\rho)$ has to be the minimizer of the energy $E^*_{\mathrm{MS}}(\cdot, g_1, g_2)$ for a.e. $\rho \in [0, 1]$. $\square$

Appendix II: Proof of Theorem 2.4. Recall that $E(g)$ is defined in (2.6). First we prove that $0 \le \inf_g E(g) < \infty$. Indeed, the left side is obvious. Moreover, if we choose $g_0 = 0$, we get:

$$\inf_g E(g) \le E(g_0) = \frac{\lambda}{2}\int_\Omega f^2 dx < \infty.$$

Thus the minimal value of $E(g)$ must exist.

I) Existence: Note that $W^{1,2}(\Omega)$ is a reflective Banach space, and $E(g)$ is convex and lower semi-continuous. Using Proposition 1.2 in [18], we just need to prove that $E(g)$ is coercive over $W^{1,2}(\Omega)$. For any $g \in W^{1,2}(\Omega)$, obviously $\|\nabla g\|_{L^2(\Omega)} = (\int_\Omega |\nabla g|^2 dx)^{\frac{1}{2}}$ is bounded by $\sqrt{\frac{2}{\mu}E(g)}$. In order to prove that $E(g)$ is coercive over $W^{1,2}(\Omega)$, we just have to prove that $\|g\|_{L^2(\Omega)}$ can also be bounded by $\sqrt{E(g)}$. Using the Poincaré inequality on $W^{1,2}(\Omega)$, see [19], we have:

$$\|g - g_\Omega\|_{L^2(\Omega)} \le C_\Omega \|\nabla g\|_{L^2(\Omega)} \le C_\Omega \sqrt{\frac{2}{\mu}E(g)}, \tag{5.2}$$

where C_Ω is a positive constant and $g_\Omega = \frac{1}{|\Omega|}\int_\Omega g(x)dx$. Moreover,

$$\begin{aligned} g_\Omega \cdot \|\mathcal{A}1\|_{L^2(\Omega)} &\le \|f - \mathcal{A}g\|_{L^2(\Omega)} + \|f - \mathcal{A}(g - g_\Omega)\|_{L^2(\Omega)} \\ &\le \sqrt{\frac{2}{\lambda}E(g)} + \|f\|_{L^2(\Omega)} + \|\mathcal{A}\| \cdot \|g - g_\Omega\|_{L^2(\Omega)} \\ &\le \|f\|_{L^2(\Omega)} + \left(\sqrt{\frac{2}{\lambda}} + C_\Omega\|\mathcal{A}\|\sqrt{\frac{2}{\mu}}\right)\sqrt{E(g)}. \end{aligned} \tag{5.3}$$

By the assumption $\mathrm{Ker}(\mathcal{A})\bigcap\mathrm{Ker}(\nabla) = \{0\}$, we know that $\|\mathcal{A}1\|_{L^2(\Omega)}$ is nonzero. Thus g_Ω is bounded by a constant plus $\sqrt{E(g)}$ times a constant. Since

$$\|g\|_{L^2(\Omega)} \le \|g_\Omega\|_2 + \|g - g_\Omega\|_2,$$

using (5.2) and (5.3), $\|g\|_{L^2(\Omega)}$ can also be bounded by a constant plus $\sqrt{E(g)}$ times a constant. Hence $\|g\|_{W^{1,2}(\Omega)}$ is bounded by a constant plus $\sqrt{E(g)}$ times a constant. This means that $E(g)$ is coercive.

II) Uniqueness: We borrow the idea in [45]. Suppose g_1^* and g_2^* are both minimizers of $E(g)$. Since $E(g)$ is convex, for any $\theta \in (0,1)$ we have:

$$\theta E^*(g_1^*) + (1-\theta)E(g_2^*) = E(\theta g_1^* + (1-\theta)g_2^*). \tag{5.4}$$

Note that each term of $E(g)$ in (2.6) is convex; especially, the first two terms of $E(g)$ are strictly convex with respect to $\mathcal{A}g$ and ∇g respectively. Therefore (5.4) implies that the following two equalities hold:

$$\frac{\theta\lambda}{2}\int_\Omega (f - \mathcal{A}g_1^*)^2 dx + \frac{(1-\theta)\lambda}{2}\int_\Omega (f - \mathcal{A}g_2^*)^2 dx = \frac{\lambda}{2}\int_\Omega \left(f - \mathcal{A}(\theta g_1^* + (1-\theta)g_2^*)\right)^2 dx,$$

$$\frac{\theta\mu}{2}\int_\Omega |\nabla g_1^*|^2 dx + \frac{(1-\theta)\mu}{2}\int_\Omega |\nabla g_2^*|^2 dx = \frac{\mu}{2}\int_\Omega |\nabla(\theta g_1^* + (1-\theta)g_2^*)|^2 dx.$$

We thus have $\mathcal{A}g_1^* = \mathcal{A}g_2^*$ and $\nabla g_1^* = \nabla g_2^*$. By the assumption $\mathrm{Ker}(\mathcal{A})\bigcap\mathrm{Ker}(\nabla) = \{0\}$, we conclude that $g_1^* - g_2^* = 0$. □

REFERENCES

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

[2] E. Bae, J. Yuan, and X. Tai, *Simultaneous convex optimization of regions and region parameters in image segmentation models*, UCLA CAM Report 11–83, 2011.

[3] L. Bar, T. Chan, G. Chung, M. Jung, N. Kiryati, R. Mohieddine, N. Sochen, and L.A. Vese, *Mumford and Shah model and its applications to image segmentation and image restoration*, Handbook of Mathematical Imaging, Springer, 2011, pp. 1095–1157.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. and Trends Mach. Learning, 3(2010), pp. 1–122.

[5] K. Bredies, K. Kunisch, and T. Pock, *Total generalized variation*, SIAM J. Imaging Sciences, 3(2010), pp. 492–526 .

[6] X. Bresson, S. Esedoglu, P. Vandergheynst, J. Thiran, and S. Osher, *Fast global minimization of the active contour/snake model*, J. Math. Imaging Vision, 28(2007), pp. 151–167.

[7] E. Brown, T. Chan, and X. Bresson, *A convex relaxation method for a class of vector-valued minimization problems with applications to Mumford-Shah segmentation*, UCLA CAM Report 10–43, 2010.

[8] X. Cai, R. Chan, S. Morigi, and F. Sgallari, *Framelet-based algorithm for segmentation of tubular structures*, SSVM(2011), LNCS, 6667(2012), pp. 411–422.

[9] A. Chambolle, *Image segmentation by variational methods: Mumford and Shah functional and the discrete approximations*, SIAM J. Appl. Math., 55(1995), pp. 827–863.

[10] A. Chambolle, *Finite differences discretization of the Mumford-Shah functional*, Math. Modelling and Numerical Analysis, 33(1999), pp. 261–288.

[11] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, CMAP, Ecole Polytechnique, Tech. Rep. R.I. 685, 2010.

[12] T. Chan, S. Esedoglu, and M. Nikolova, *Algorithms for finding global minimizers of image segmentation and denoising models*, SIAM J. Appl. Math, 66(2006), pp. 1632–1648.

[13] T. Chan, A. Marquina, and P. Mulet, *High-order total variation-based image restoration*, SIAM Journal on Scientific Computing, 22(2000), pp. 503–516.

[14] T. Chan and L.A. Vese, *Active contours without edges*, IEEE Trans. Image Process., 10(2001), pp. 266–277.

[15] R. Chan and M. Ng, *Conjugate gradient method for Toeplitz systems*, SIAM Review, 38(1996), pp. 427–482.

[16] G. David, *Singular sets of minimizers for the Mumford-Shah functional (progress in mathematics)*, Birkhäuser Verlag, Basel, 2005.

[17] B. Dong, A. Chien, and Z. Shen, *Frame based segmentation for medical images*, Technical Report 22, UCLA, 2010.

[18] I. Ekeland, and R. Temam, *Convex analysis and variational problems*, SIAM, Philadelphia, 1999.

[19] L. C. Evans, *Partial differential equations*, American Mathematical Society, ISBN 0821807722, 1998.

[20] M. Figueiredo and J. Bioucas-Dias, *Restoration of Poissonian images using alternating direction optimization*, IEEE Trans. Image Process., 19(2010), pp. 3133–3145.

[21] E. Franchini, S. Morigi, and F. Sgallari, *Segmentation of 3D tubular structures by a PDE-based anisotropic diffusion model*, M. Dæhlen et al. (eds.): MMCS 2008, LNCS5862, 2010, pp. 224–241. Springer-Verlag Berlin Heidelberg, 2010.

[22] M. Gobbino, *Finite difference approximation of the Mumford-Shah functional*, Comm. Pure Appl. Math., 51(1998), pp. 197–228.

[23] T. Goldstein and S. Osher, *The split Bregman algorithm for L1 regularized problems*, SIAM J. Imaging Sciences, 2(2009), pp. 323–343.

[24] L. Grady and C. Alvino, *Reformulating and optimizing the mumford-shah functional on a graph – a faster, lower energy solution*, European Conference on Computer Vision (ECCV), 2008, pp. 248–261.

[25] J. Hartigan and M. Wang, *A K-means clustering algorithm*, Applied Statistics, 28(1979), pp. 100–108.

[26] B. He and X. Yuan, *Convergence Analysis of Primal-Dual Algorithms for a Saddle-Point Problem: From Contraction Perspective*, SIAM J. Imaging Sci., 5(2012), pp. 119–149.

[27] T. Kanungo, D. Mount, N. Netanyahu,C. Piatko,R. Silverman, and A. Wu, *An efficient k-means clustering algorithm: Analysis and implementation*, IEEE Trans. Pattern Analysis

and Machine Intelligence, 24(2002), pp. 881–892.

[28] F. Li, M. Ng, T. Zeng, and C. Shen, *A multiphase image segmentation method based on fuzzy region competition*, SIAM J. Imaging Sciences, 3(2010), pp. 277–299.

[29] F. Li, C. Shen, and C. Li, *Multiphase soft segmentation with total variation and H^1 regularization*, J. Math. Imaging Vision, 37(2010), pp. 98–111.

[30] J. Lie, M. Lysaker, and X. Tai, *A binary level set model and some applications to Mumford-Shah image segmentation*, IEEE Trans. on Image Processing, 15(2006), pp. 1171–1181.

[31] M. Lysaker, A. Lundervold, and X. Tai, *Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time*, IEEE Trans. on Image Processing, 12(2003), pp. 1579–1590.

[32] J. MacQueen, *Some methods for classification and analysis of multivariate observations*, Proc. 5th Berkeley Symposium, 1967, pp. 281–297.

[33] M. Morini and M. Negri, *Mumford-Shah functional as Γ-limit of discrete Perona-Malik energies*, Math. Models and Methods in Applied Sciences, 13(2003), pp. 785–805.

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

[35] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, *An algorithm for minimizing the Mumford-Shah functional*, Proc. 12th IEEE Int'l Conf. Computer Vision, 2009, pp. 1133–1140.

[36] T. Pock, D. Cremers, A. Chambolle, and H. Bischof, *A convex relaxation approach for computing minimal partitions*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 810–817.

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

[38] B. Sandberg, S. Kang, and T. Chan, *Unsupervised multiphase segmentation: a phase balancing model*, IEEE Trans. on Image Processing, 19(2010), pp. 119–130.

[39] 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 Processing, 10(2001), pp. 1169–1186.

[40] L. Vese and T. Chan, *A multiphase level set framework for image segmentation using the Mumford and Shah model*, Int. J. of Computer Vision, 50(2002), pp. 271–293.

[41] Y. You and M. Kaveh, *Fourth-order partial differential equation for noise removal*, IEEE Trans. on Image Processing, 9(2000), pp. 1723–1730.

[42] J. Yuan, E. Bae, and X. Tai, *A study on continuous max-flow and min-cut approaches*, CVPR, USA, San Francisco, 2010.

[43] J. Yuan, E. Bae, X. Tai, and Y. Boycov, *A study on continuous max-flow and min-cut approaches*, UCLA CAM Report 10–61, 2010.

[44] J. Yuan, E. Bae, X. Tai, and Y. Boycov, *A continuous max-flow approach to potts model*, ECCV, 2010.

[45] T. Zeng, X. Li, and M. Ng, *Alternating minimization method for total variation based wavelet shrinkage model*, Commun. Comput. Phys., 8(2010), pp. 976–994.