

Research Article

A New and Fast Multiphase Image Segmentation Model for Color Images

Yunyun Yang and Boying Wu

Department of Mathematics, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Yunyun Yang, yangyunyun1986@gmail.com

Received 10 October 2011; Accepted 27 December 2011

Academic Editor: Pedro Ribeiro

Copyright © 2012 Y. Yang and B. Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a new and fast multiphase image segmentation model for color images. We propose our model by incorporating the globally convex image segmentation method and the split Bregman method into the piecewise constant multiphase Vese-Chan model for color images. We have applied our model to many synthetic and real color images. Numerical results show that our model can segment color images with multiple regions and represent boundaries with complex topologies, including triple junctions. Comparison with the Vese-Chan model demonstrates the efficiency of our model. Besides, our model does not require a priori denoising step and is robust with respect to noise.

1. Introduction

Image segmentation [1–8] is an important technique for detecting objects and analyzing images in computer vision and image processing. Image segmentation is concerned with the task of partitioning a given image into different classes or regions corresponding to different objects and the background in the image.

Mumford and Shah [9] formulated the image segmentation problem and proposed the famous Mumford-Shah model. For a special case of the Mumford-Shah model, Chan and Vese [1] proposed the famous Chan-Vese model without using the image gradient. Chan and Vese gave the two-phase level set formulation for gray images in [1]. Then, the authors extended their scalar Chan-Vese algorithm to the vector-valued case in [2]. There are many other image segmentation models for color images, and we mention the works in [10, 11].

Either the scalar Chan-Vese algorithm in [1] or the Chan-Vese model for vector-valued images in [2] is mainly for images with two phases. To deal with images with multiple regions, Vese and Chan extended their original two-phase Chan-Vese model [1, 2] to a multiphase model by using a multiphase level set formulation in [3]. In their multiphase Vese-Chan model [3], multiple regions can be represented by multiple level set functions. Vese and

Chan described their model in two cases: piecewise constant case and piecewise smooth case. In this paper, we mainly focus on the piecewise constant multiphase Vese-Chan model.

In fact, Zhao et al. [12], Samson et al. [13], and Paragios and Deriche [14] have already proposed several multiphase image segmentation models to segment images with multiple regions before the multiphase Vese-Chan model. However, these models have the natural problems of vacuum and overlap. Compared with these models, the multiphase Vese-Chan model has several advantages. Firstly, it automatically avoids the problems of vacuum and overlap by construction. Secondly, it needs fewer level set functions to represent the same number of phases in the piecewise constant case. Finally it can represent boundaries with complex topologies, including triple junctions.

Recently, the split Bregman method has been applied to solve image segmentation problem more efficiently. The efficiency of the split Bregman method for image segmentation has been demonstrated in [15–18].

In this paper, we propose a fast multiphase image segmentation model in a variational level set formulation for color images. Our model is an extension of our previous model for gray images proposed in [19]. Our model has been applied to many synthetic and real color images. Numerical results show that our model has the advantages of the original Vese-Chan model [3] for multiphase image segmentation, but our model is much more efficient. The robustness of our model to noise has also been demonstrated by the numerical results.

The outline of the paper is as follows. We review our improved multiphase image segmentation model for gray images in Section 2.1. In Section 2.2, we extend our model to work for color images. Then, we apply the split Bregman method to solve our proposed minimization problem more efficiently in Section 2.3. Section 3 gives the algorithm and numerical results of our model. We conclude this paper in Section 4.

2. Our Model

2.1. Our Model for Gray Images

In our previous work [19], we have proposed an improved active contour model for multiphase image segmentation based on the piecewise constant multiphase Vese-Chan model [3], the globally convex image segmentation method [20] and the split Bregman method [15–17]. The model we proposed in [19] is mainly for gray images. For simplicity, we only review the four-phase model in this section.

Let $\Omega \subset \mathbb{R}^2$ be the image domain and $u_0 : \Omega \rightarrow \mathbb{R}$ be a given gray level image. Our proposed model is given in the following level set formulation:

$$\min_{0 \leq \phi_1, \phi_2 \leq 1} E(\phi_1, \phi_2) = \min_{0 \leq \phi_1, \phi_2 \leq 1} \left(|\nabla \phi_1|_g + |\nabla \phi_2|_g + \mu \langle \phi_1, r_1 \rangle + \mu \langle \phi_2, r_2 \rangle \right), \quad (2.1)$$

where ϕ_1 and ϕ_2 are two level set functions, and μ is a positive constant. $|\nabla \phi_i|_g$ and $\langle \phi_i, r_i \rangle$ are defined as

$$\begin{aligned} |\nabla \phi_i|_g &= \int g(|\nabla u_0(\mathbf{x})|) |\nabla \phi_i(\mathbf{x})| d\mathbf{x}, \\ \langle \phi_i, r_i \rangle &= \int \phi_i(\mathbf{x}) r_i(\mathbf{x}) d\mathbf{x}, \end{aligned} \quad (2.2)$$

where

$$g(\xi) = \frac{1}{1 + \beta|\xi|^2} \quad (2.3)$$

is an edge detector function [17, 18, 21], and β is a parameter that determines the detail level of the segmentation. r_1 and r_2 are defined as

$$\begin{aligned} r_1 &= \left((u_0 - c_{11})^2 - (u_0 - c_{01})^2 \right) \phi_2 + \left((u_0 - c_{10})^2 - (u_0 - c_{00})^2 \right) (1 - \phi_2), \\ r_2 &= \left((u_0 - c_{11})^2 - (u_0 - c_{10})^2 \right) \phi_1 + \left((u_0 - c_{01})^2 - (u_0 - c_{00})^2 \right) (1 - \phi_1), \end{aligned} \quad (2.4)$$

where $c = (c_{11}, c_{10}, c_{01}, c_{00})$ is a constant vector of averages in the four phases.

Given $\phi_1(0, \mathbf{x}) = \phi_{1,0}(\mathbf{x})$, $\phi_2(0, \mathbf{x}) = \phi_{2,0}(\mathbf{x})$, once the minimization problem (2.1) is solved, the image domain Ω can be partitioned into four regions Ω_i ($i = 1, 2, 3, 4$) by thresholding the level set functions as follows:

$$\begin{aligned} \Omega_1 &= \{ \mathbf{x} : \phi_1(\mathbf{x}) > \alpha, \phi_2(\mathbf{x}) > \alpha \}, \\ \Omega_2 &= \{ \mathbf{x} : \phi_1(\mathbf{x}) > \alpha, \phi_2(\mathbf{x}) < \alpha \}, \\ \Omega_3 &= \{ \mathbf{x} : \phi_1(\mathbf{x}) < \alpha, \phi_2(\mathbf{x}) > \alpha \}, \\ \Omega_4 &= \{ \mathbf{x} : \phi_1(\mathbf{x}) < \alpha, \phi_2(\mathbf{x}) < \alpha \}, \end{aligned} \quad (2.5)$$

for some $\alpha \in (0, 1)$, we choose $\alpha = 0.5$ in this paper. This is different from [3] where the zero level set is used to identify the boundary. We use the $\alpha = 0.5$ level set to identify the boundary because in our paper we have restricted $0 \leq \phi_i \leq 1$. The average vector $c = (c_{11}, c_{10}, c_{01}, c_{00})$ can be obtained by

$$\begin{aligned} c_{11} &= \text{mean}(u_0) \quad \text{in } \Omega_1, \\ c_{10} &= \text{mean}(u_0) \quad \text{in } \Omega_2, \\ c_{01} &= \text{mean}(u_0) \quad \text{in } \Omega_3, \\ c_{00} &= \text{mean}(u_0) \quad \text{in } \Omega_4. \end{aligned} \quad (2.6)$$

2.2. Our Model for Color Images

In this section, we extend our model in Section 2.1 from gray images to color images. Let $u_0 : \Omega \rightarrow \mathbb{R}^3$ be a given color image and u_{0i} ($i = 1, 2, 3$) the three channels of u_0 . Let $c_i = (c_{1i}, c_{2i}, c_{3i}, c_{4i})$ ($i = 1, 2, 3$) be the average vector of the i th channel.

The extension of our model to color images is still in the same formulation as (2.1):

$$\min_{0 \leq \phi_1, \phi_2 \leq 1} E(\phi_1, \phi_2) = \min_{0 \leq \phi_1, \phi_2 \leq 1} \left(|\nabla \phi_1|_g + |\nabla \phi_2|_g + \mu \langle \phi_1, r_1 \rangle + \mu \langle \phi_2, r_2 \rangle \right). \quad (2.7)$$

The difference between (2.7) and (2.1) lies in the definition of r_1 and r_2 . In our model for color images (2.7), r_1 and r_2 are defined differently from (2.4):

$$\begin{aligned} r_1 &= \left(\sum_{i=1}^3 (u_{0i} - c_{1i})^2 - \sum_{i=1}^3 (u_{0i} - c_{3i})^2 \right) \phi_2 + \left(\sum_{i=1}^3 (u_{0i} - c_{2i})^2 - \sum_{i=1}^3 (u_{0i} - c_{4i})^2 \right) (1 - \phi_2), \\ r_2 &= \left(\sum_{i=1}^3 (u_{0i} - c_{1i})^2 - \sum_{i=1}^3 (u_{0i} - c_{2i})^2 \right) \phi_1 + \left(\sum_{i=1}^3 (u_{0i} - c_{3i})^2 - \sum_{i=1}^3 (u_{0i} - c_{4i})^2 \right) (1 - \phi_1). \end{aligned} \quad (2.8)$$

Besides, another difference is the computing of the edge detector function $g(|\nabla u_0(\mathbf{x})|)$. According to the definition of g in (2.3), we need to compute $|\nabla u_0(\mathbf{x})|^2$ when computing g .

When u_0 is a gray image, for each pixel $\mathbf{x} = (x, y)$, $|\nabla u_0(\mathbf{x})|^2$ is defined as

$$|\nabla u_0(\mathbf{x})|^2 = \left[\left(\frac{\partial u_0}{\partial x} \right)^2 + \left(\frac{\partial u_0}{\partial y} \right)^2 \right] (\mathbf{x}). \quad (2.9)$$

When u_0 is a color image, $|\nabla u_0(\mathbf{x})|^2$ is defined as

$$|\nabla u_0(\mathbf{x})|^2 = \left\{ \left[\left(\frac{\partial u_{01}}{\partial x} \right)^2 + \left(\frac{\partial u_{02}}{\partial x} \right)^2 + \left(\frac{\partial u_{03}}{\partial x} \right)^2 \right] + \left[\left(\frac{\partial u_{01}}{\partial y} \right)^2 + \left(\frac{\partial u_{02}}{\partial y} \right)^2 + \left(\frac{\partial u_{03}}{\partial y} \right)^2 \right] \right\} (\mathbf{x}). \quad (2.10)$$

Similarly to the case of gray images, once the minimization problem (2.7) is solved, the image domain Ω can also be partitioned into four regions Ω_i ($i = 1, 2, 3, 4$) by (2.5). Then, the average vector $c_i = (c_{1i}, c_{2i}, c_{3i}, c_{4i})$ ($i = 1, 2, 3$) for each channel can be updated by

$$\begin{aligned} c_{1i} &= \text{mean}(u_{0i}) \quad \text{in } \Omega_{1i}, \\ c_{2i} &= \text{mean}(u_{0i}) \quad \text{in } \Omega_{2i}, \\ c_{3i} &= \text{mean}(u_{0i}) \quad \text{in } \Omega_{3i}, \\ c_{4i} &= \text{mean}(u_{0i}) \quad \text{in } \Omega_{4i}, \end{aligned} \quad (i = 1, 2, 3). \quad (2.11)$$

Let u be the fitting image of the given color image u_0 and u_i ($i = 1, 2, 3$) the three channels of u . Then, each channel u_i of the fitting image u can be expressed as

$$u_i = c_{1i} \phi_1 \phi_2 + c_{2i} \phi_1 (1 - \phi_2) + c_{3i} (1 - \phi_1) \phi_2 + c_{4i} (1 - \phi_1) (1 - \phi_2). \quad (2.12)$$

Thus, we can obtain the fitting image u by

$$\begin{aligned} u(:, :, 1) &= u_1, \\ u(:, :, 2) &= u_2, \\ u(:, :, 3) &= u_3. \end{aligned} \quad (2.13)$$

We denote the four segmented phases of u as phase_1 , phase_2 , phase_3 , and phase_4 , then the four phases can be given as

$$\begin{aligned}\text{phase}_1(:, :, i) &= c_{1i}\phi_1\phi_2, \\ \text{phase}_2(:, :, i) &= c_{2i}\phi_1(1 - \phi_2), \\ \text{phase}_3(:, :, i) &= c_{3i}(1 - \phi_1)\phi_2, \\ \text{phase}_4(:, :, i) &= c_{4i}(1 - \phi_1)(1 - \phi_2),\end{aligned}\tag{2.14}$$

where $i = 1, 2, 3$ denotes the three channels.

In fact, u can also be obtained by

$$u = \text{phase}_1 + \text{phase}_2 + \text{phase}_3 + \text{phase}_4.\tag{2.15}$$

Remark 2.1. The multiphase image segmentation model we proposed is different from the piecewise constant multiphase Vese-Chan model in [3] for the following reasons. First, our energy functional $E(\phi_1, \phi_2)$ in (2.1) or (2.7) is different from [3]. Second, we incorporate information from the edge into the energy functional by using a nonnegative edge detector function g to detect boundaries more easily. Third, we apply the split Bregman method, and our model is much more efficient than the model in [3]. Last, we do not need the procedure to reinitialize ϕ_i to the signed distance function to its zero-level curve as [1-3], we simply initialize ϕ_i as a binary step function which takes 1 inside a region and 0 outside. We have restricted ϕ_i to $[0, 1]$, thus we do not need the Heaviside function H or its smooth versions H_ε as [3].

2.3. Application of the Split Bregman Method to Update ϕ_1 and ϕ_2

To minimize the energy functional $E(\phi_1, \phi_2)$ in (2.1) or (2.7) with respect to ϕ_1 and ϕ_2 , one traditional method is using the standard gradient descent method directly [1-3]. In this section, we apply the split Bregman method to solve our proposed minimization problem (2.7) more efficiently. First, we introduce two auxiliary variables: $\vec{d}_1 \leftarrow \nabla\phi_1$ and $\vec{d}_2 \leftarrow \nabla\phi_2$. Then, we add two quadratic penalty functions to weakly enforce the resulting equality constraints and get the unconstrained problem as follows:

$$\begin{aligned}& (\phi_1^*, \phi_2^*, \vec{d}_1^*, \vec{d}_2^*) \\ &= \arg \min_{\substack{0 \leq \phi_1, \phi_2 \leq 1 \\ \vec{d}_1, \vec{d}_2}} \left(|\vec{d}_1|_g + |\vec{d}_2|_g + \mu \langle \phi_1, r_1 \rangle + \mu \langle \phi_2, r_2 \rangle + \frac{\lambda}{2} \|\vec{d}_1 - \nabla\phi_1\|^2 + \frac{\lambda}{2} \|\vec{d}_2 - \nabla\phi_2\|^2 \right),\end{aligned}\tag{2.16}$$

where λ is a positive constant.

We then strictly enforce the constraints by applying the Bregman iteration, and this results in the following optimization problem:

$$\begin{aligned}
& \left(\phi_1^{k+1}, \phi_2^{k+1}, \vec{d}_1^{k+1}, \vec{d}_2^{k+1} \right) \\
& = \arg \min_{\substack{0 \leq \phi_1, \phi_2 \leq 1 \\ \vec{d}_1, \vec{d}_2}} \left(\left| \vec{d}_1 \right|_g + \left| \vec{d}_2 \right|_g + \mu \langle \phi_1, r_1 \rangle + \mu \langle \phi_2, r_2 \rangle + \frac{\lambda}{2} \left\| \vec{d}_1 - \nabla \phi_1 - \vec{b}_1^{k+1} \right\|^2 + \frac{\lambda}{2} \left\| \vec{d}_2 - \nabla \phi_2 - \vec{b}_2^{k+1} \right\|^2 \right), \\
& \vec{b}_1^{k+1} = \vec{b}_1^k + \nabla \phi_1^{k+1} - \vec{d}_1^{k+1}, \\
& \vec{b}_2^{k+1} = \vec{b}_2^k + \nabla \phi_2^{k+1} - \vec{d}_2^{k+1}.
\end{aligned} \tag{2.17}$$

Keep \vec{d}_1, \vec{d}_2 , and ϕ_2 fixed, the Euler-Lagrange equation of the optimization problem (2.17) with respect to ϕ_1 is

$$\Delta \phi_1 = \frac{\mu}{\lambda} r_1 + \nabla \cdot (\vec{d}_1 - \vec{b}_1), \quad \text{whenever } 0 \leq \phi_1 \leq 1. \tag{2.18}$$

Note here that from the definition of r_2 in (2.8), r_2 is a function of ϕ_1 . Here, when we minimize (2.17) with respect to ϕ_1 , we just consider r_2 as a constant. From the numerical results in the next section, we can see that this consideration does not affect the performance our model. Similarly, when we minimize (2.17) with respect to ϕ_2 for fixed \vec{d}_1, \vec{d}_2 , and ϕ_1 , we also consider r_1 as a constant and get the following Euler-Lagrange equation:

$$\Delta \phi_2 = \frac{\mu}{\lambda} r_2 + \nabla \cdot (\vec{d}_2 - \vec{b}_2), \quad \text{whenever } 0 \leq \phi_2 \leq 1. \tag{2.19}$$

For (2.18) and (2.19), the central difference and the backward difference are used for the Laplace operator and the divergence operator, respectively. Then the resulting numerical scheme is

$$\begin{aligned}
\alpha_{l,i,j} &= d_{l,i-1,j}^x - d_{l,i,j}^x + d_{l,i,j-1}^y - d_{l,i,j}^y - \left(b_{l,i-1,j}^x - b_{l,i,j}^x + b_{l,i,j-1}^y - b_{l,i,j}^y \right), \\
\beta_{l,i,j} &= \frac{1}{4} \left(\phi_{l,i-1,j} + \phi_{l,i+1,j} + \phi_{l,i,j-1} + \phi_{l,i,j+1} - \frac{\mu}{\lambda} r_l + \alpha_{l,i,j} \right), \\
\phi_{l,i,j} &= \max \{ \min \{ \beta_{l,i,j}, 1 \}, 0 \},
\end{aligned} \tag{2.20}$$

where $l = 1$ or 2 denotes the two schemes for (2.18) and (2.19).

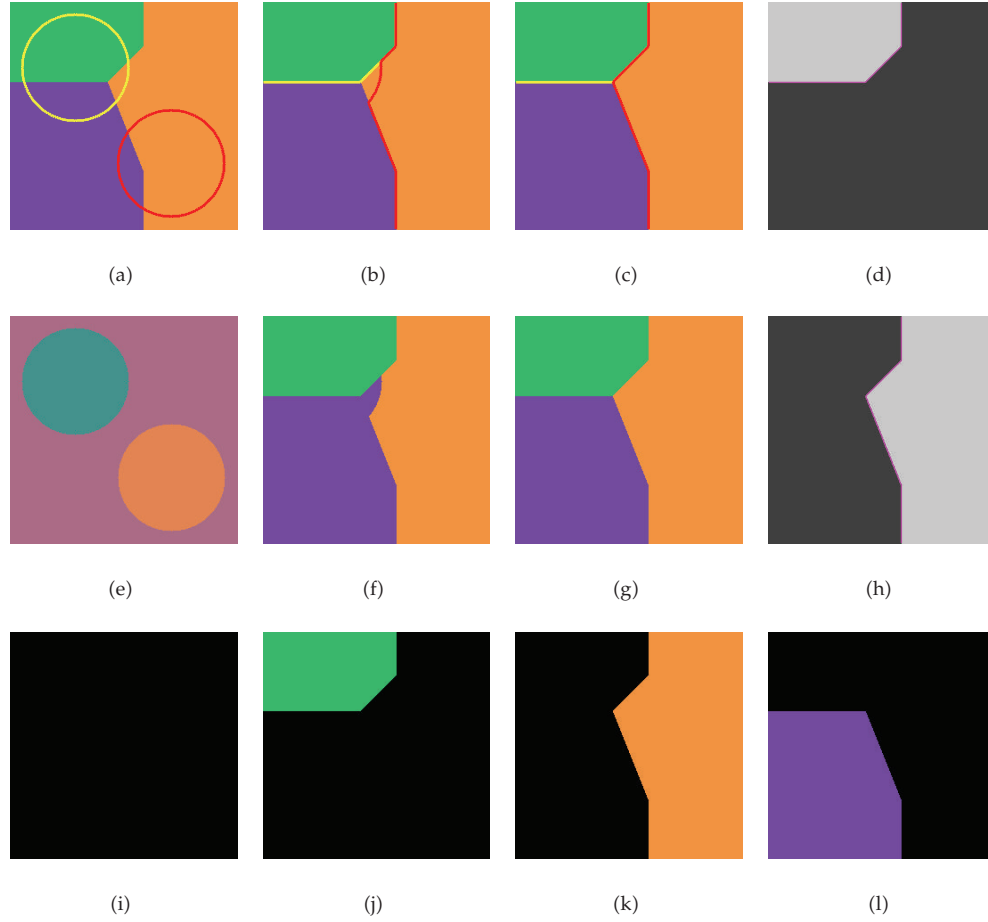


Figure 1: Results for a synthetic image with our model. (a)–(c): The active contour evolving process from the initial contour to the final contour; (e)–(g): The corresponding fitting images u at different iterations; (d) and (h): The α level sets of the final ϕ_1 and ϕ_2 ; (i)–(l): The final four phases; size = 256×256 .

Then we minimize (2.17) with respect to \vec{d}_1 (\vec{d}_2) for fixed ϕ_1 , ϕ_2 , and \vec{d}_2 (ϕ_1 , ϕ_2 and \vec{d}_1) and obtain

$$\begin{aligned}\vec{d}_1^{k+1} &= \text{shrink}_g\left(\vec{b}_1^k + \nabla\phi_1^{k+1}, \frac{1}{\lambda}\right) = \text{shrink}\left(\vec{b}_1^k + \nabla\phi_1^{k+1}, \frac{g}{\lambda}\right), \\ \vec{d}_2^{k+1} &= \text{shrink}_g\left(\vec{b}_2^k + \nabla\phi_2^{k+1}, \frac{1}{\lambda}\right) = \text{shrink}\left(\vec{b}_2^k + \nabla\phi_2^{k+1}, \frac{g}{\lambda}\right),\end{aligned}\quad (2.21)$$

where

$$\text{shrink}(\mathbf{x}, \gamma) = \begin{cases} \frac{\mathbf{x}}{|\mathbf{x}|} \max(|\mathbf{x}| - \gamma, 0), & \mathbf{x} \neq 0, \\ 0, & \mathbf{x} = 0. \end{cases}\quad (2.22)$$

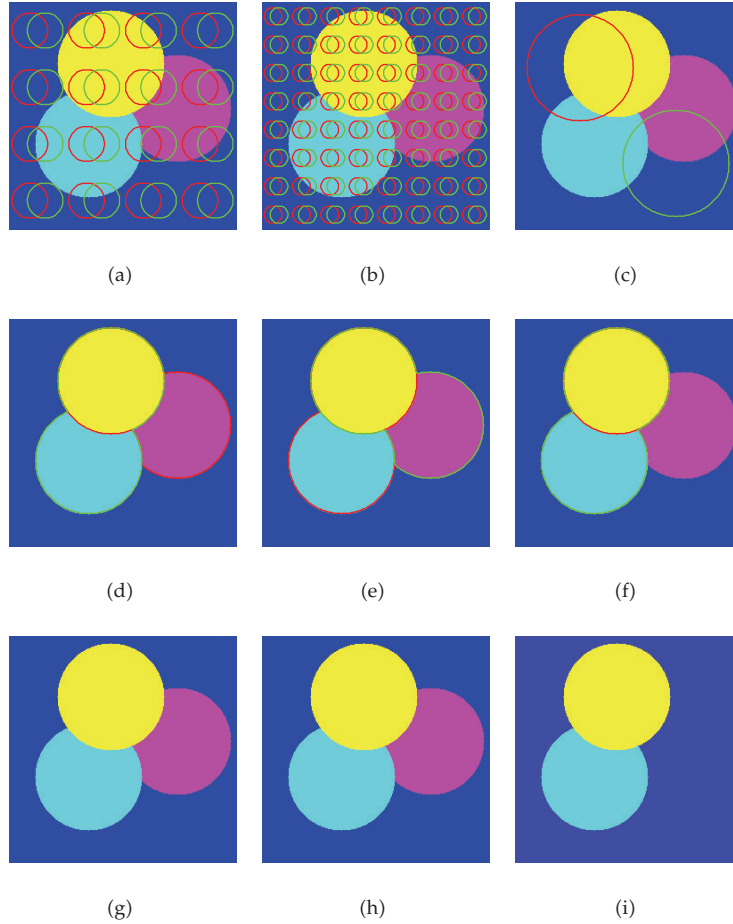


Figure 2: Results for a synthetic image with different initial conditions; (a)–(c): The original image with three different initial contours; (d)–(f): The corresponding final contours; (g)–(i): The corresponding final fitting images; size = 256×256 .

3. The Algorithm and Numerical Results

3.1. The Algorithm

The split Bregman algorithm for the proposed minimization problem (2.7) in Section 2.3 can be summarized as in Algorithm 1.

3.2. Numerical Results

Our model has been applied to synthetic and real color images with multiple regions in this section. When working with the level set function ϕ_i , we simply initialize the level set function ϕ_i as a binary step function which takes 1 inside a region and 0 outside. The advantage of using binary step functions as the initial level set functions is that new contours can emerge easily and the curve evolution is significantly faster than the evolution from initial functions as signed distance maps. In this paper, $\lambda = 1$ and $\beta = 0$ are used for all

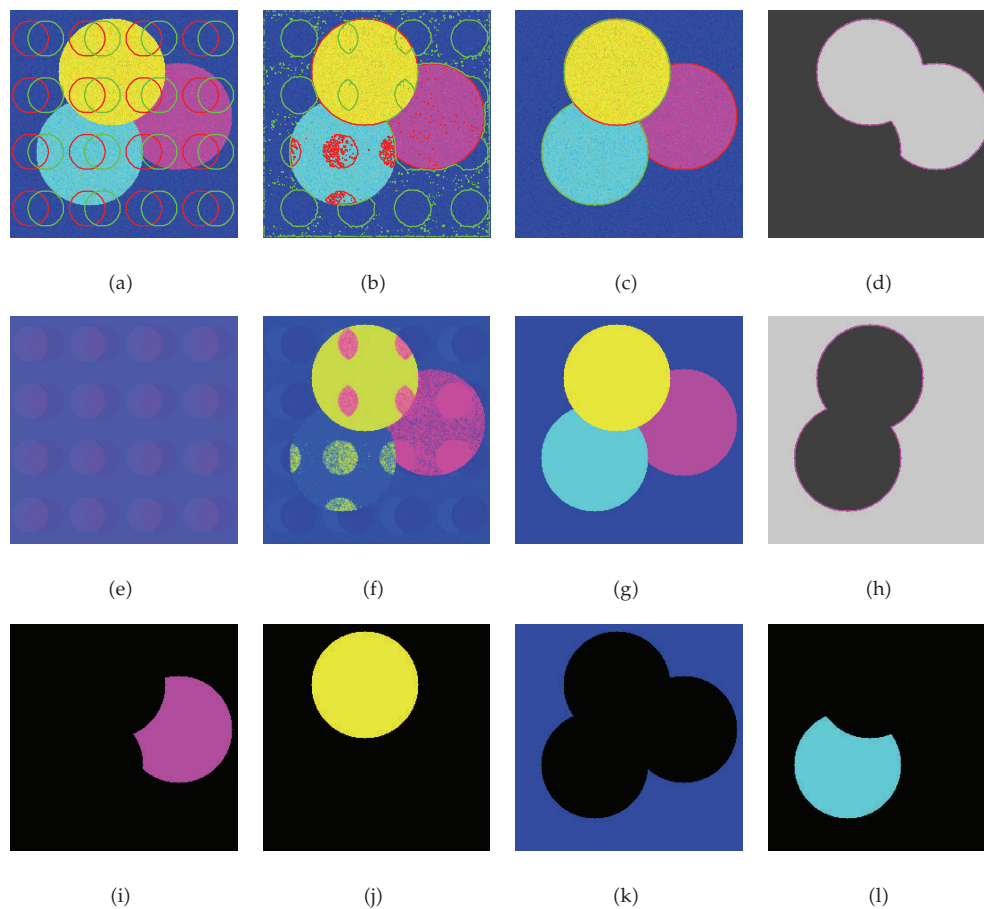


Figure 3: Application of our model to the noisy version of the image from Figure 2. (a)–(c): The curve evolution process; (e)–(g): The corresponding fitting images u ; (d) and (h): The α level sets of the final ϕ_1 and ϕ_2 ; (i)–(l): The final four phases.

```

1: Input  $u_0$  and  $\phi_1^0, \phi_2^0$ 
2: Compute initial  $\Omega_i$  and  $c$  by (2.5) and (2.11)
3: while  $\|\phi_1^{k+1} - \phi_1^k\| > \epsilon$  or  $\|\phi_2^{k+1} - \phi_2^k\| > \epsilon$  do
4: Define  $r_1^k, r_2^k$  by (2.8)
5: Update  $\phi_1^{k+1}, \phi_2^{k+1}$  and  $\vec{d}_1^{\rightarrow k+1}, \vec{d}_2^{\rightarrow k+1}$  by (2.20) and (2.21)
6: Update  $\vec{b}_l^{\rightarrow k+1} = \vec{b}_l^{\rightarrow k} + \nabla \phi_l^{k+1} - \vec{d}_l^{\rightarrow k+1}$  ( $l = 1, 2$ )
7: Update  $\Omega_i$  and  $c$  by (2.5) and (2.11)
8: end while

```

Algorithm 1

images. Unless otherwise specified, we choose $\mu = 50/255^2$ for all synthetic color images and $\mu = 30/255^2$ for all real images. The size of each image is specified in each figure.

Figure 1 shows the application of our model to a synthetic color image with a triple junction. Most of the models [12, 13] need three level set functions to represent the triple

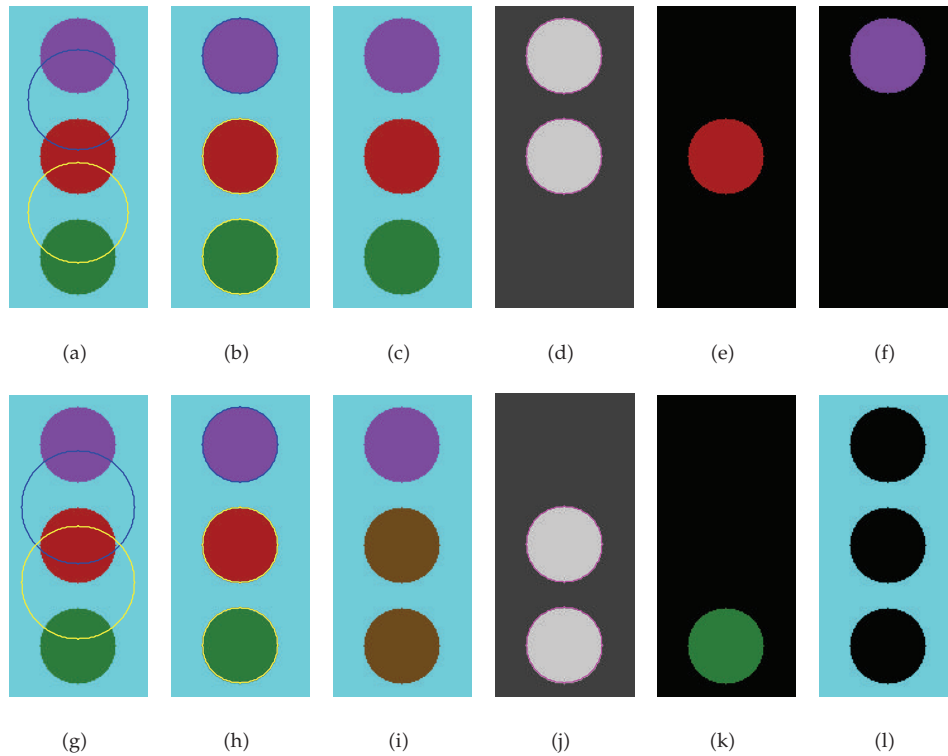


Figure 4: Results for a synthetic image with two different initial conditions. (a) and (g): The two different initial contours; (b) and (h): The corresponding final contours; (c) and (i): The corresponding final fitting images; (d) and (j): The α level sets of the final ϕ_1 and ϕ_2 with the initial condition (a); (i)–(l): The final four phases with the initial condition (a); size = 240×110 .

junction. Here, we need only two level set functions. The $\alpha = 0.5$ level sets of the final ϕ_1 and ϕ_2 are shown in (d) and (h), which have to overlap on a segment of the triple junction.

We then apply our model to another synthetic image in Figures 2 and 3. In Figure 2, we show the results for the clean image with three different initial conditions. When we seed with small initials in (a) and (b), our model can work well, both the final contours and the final fitting images are correct. However, when we use two disjoint circles as the initial curves, our model will be trapped in a local minimum, and the pink object will be missed as shown in (f) and (i). Then we add random noises with standard deviation 30.0 to the clean image. Figure 3 shows the results for the noisy version. We can observe that our model works well even if the noise level is high. In the following experiments, the noises we added are all random noise with standard deviation 30.0.

The results for another synthetic color image without or with noise are shown in Figures 4 and 5. The results for the clean image with two different initial conditions are shown in Figure 4. When we use two disjoint initial curves as shown in (a), both the final contour and the final fitting image are good. When using two joint initial curves as shown in (g), we can get the right final contour shown in (h), but the final fitting image is incorrect shown in (i). The two bottom objects are considered to have the same average value. The results for the noisy version are shown in Figure 5.

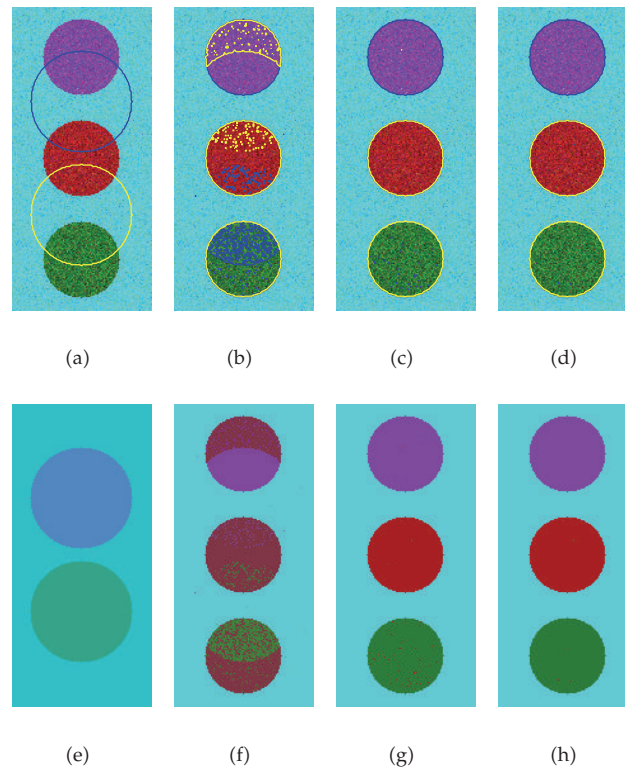


Figure 5: Application of our model to the noisy version of the image from Figure 4. (a)–(d): The curve evolution process; (e)–(h): The corresponding fitting images u .

Figures 6, 7, and 8 show the applications of our model to two synthetic images and their noisy versions. Both the two images have three objects with different shapes, which may be difficult to detect. However, our model can identify the objects correctly for both clean images and noisy versions.

We have declared that the standard deviation of the random noise added to the above synthetic images is 30.0. Now, we increase the noise level and give the results with our model in Figures 9 and 10. For the above five synthetic images, when we increase the noise level to 50.0, our model can give the correct final contours and final fittings, which can be clearly seen from Figure 9. Even the noise is increased to a much higher level 80.0, our model can still detect the object boundaries of the synthetic images and give good fitting images. The corresponding initial contours, final contours and final fitting images are shown in Figure 10. Here, we only need to choose $\mu = 1/255^2$ to avoid detecting the high-level noise for the noisy versions in Figures 9 and 10. This demonstrates the robustness of our model to noise.

We then apply our model to three real color images, and the results are shown in Figures 11, 12, 13, 14, and 15. We seed with small initial curves for all of these real images. Figures 11 and 12 show the results for an image of flowers, while Figures 13 and 14 give the results for a 4-colors image. We can observe that our model can segment these color images well. In Figure 15 we apply our model to an image with contours without gradient (cognitive contours). From the final fitting image shown in (d), it can be seen that our model can also detect “contours without edges”.

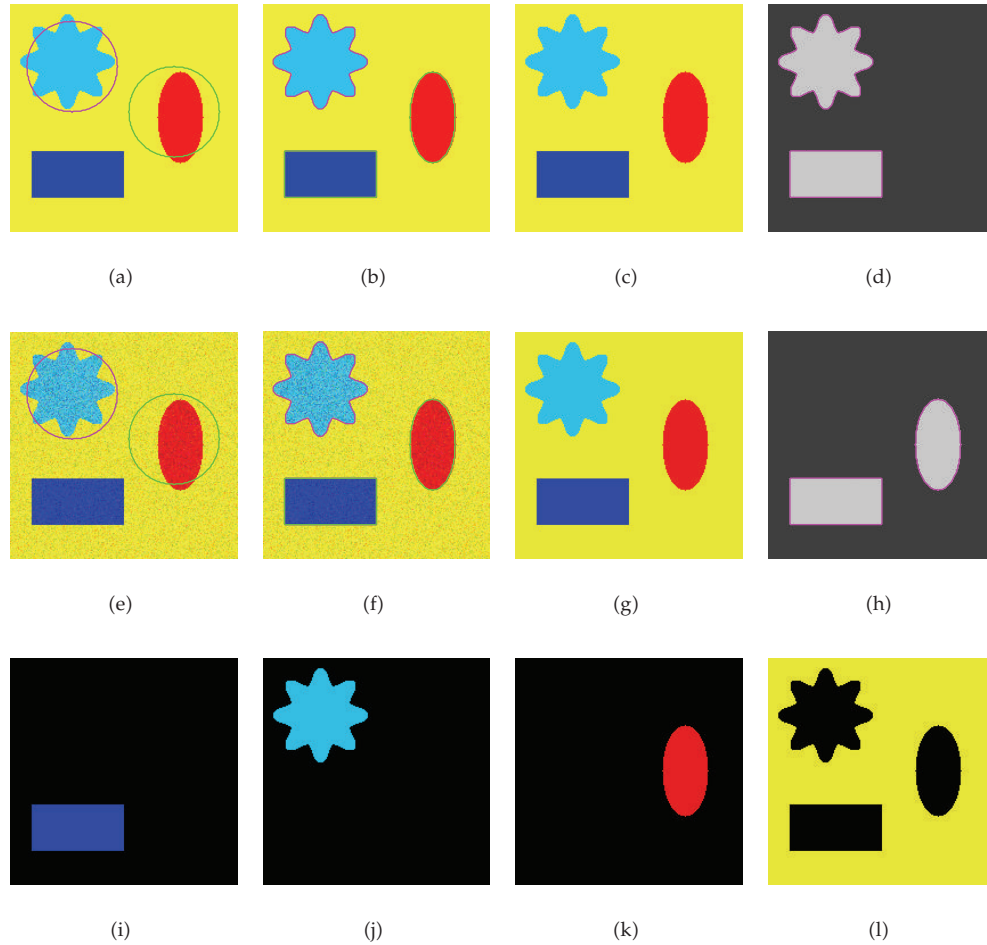


Figure 6: Results for a synthetic image without noise and with noise. (a)–(c) and (e)–(g): The initial contour, final contour and final fitting image for the clean and noisy images; (d), (h) and (i)–(l): The α level sets of the final ϕ_1 , ϕ_2 and the final four phases for the noisy version; size = 200×200 .

Figures 16 and 17 show the results of our model for two real color images from the Berkeley image set. In Figures 16 and 17, we give the curve evolution process in Row 1 and the corresponding fitting images in Row 2, while the final four phases are shown in Row 3. It can be observed that our model can handle these two real color images very well.

Our model can be applied to segment real medical MRI images. We apply our model to two real brain MRI images in Figures 18 and 19. Similarly, the curve evolution process, corresponding fitting images and final four phases are shown in Row 1, Row 2 and Row 3, respectively. We can see that our model can identify quite well the gray matter, the white matter, and so forth.

To demonstrate the efficiency of our model, we compare the computation (CPU) time for the above five synthetic images with the Vese-Chan model from [3] and our model in Tables 1 and 2. In Table 1, we test the CPU time for all the clean synthetic images used in this paper. Except the image from Figure 1, we use the noisy versions for other four synthetic images for Table 2. From Tables 1 and 2, we can see that our model is much more efficient than

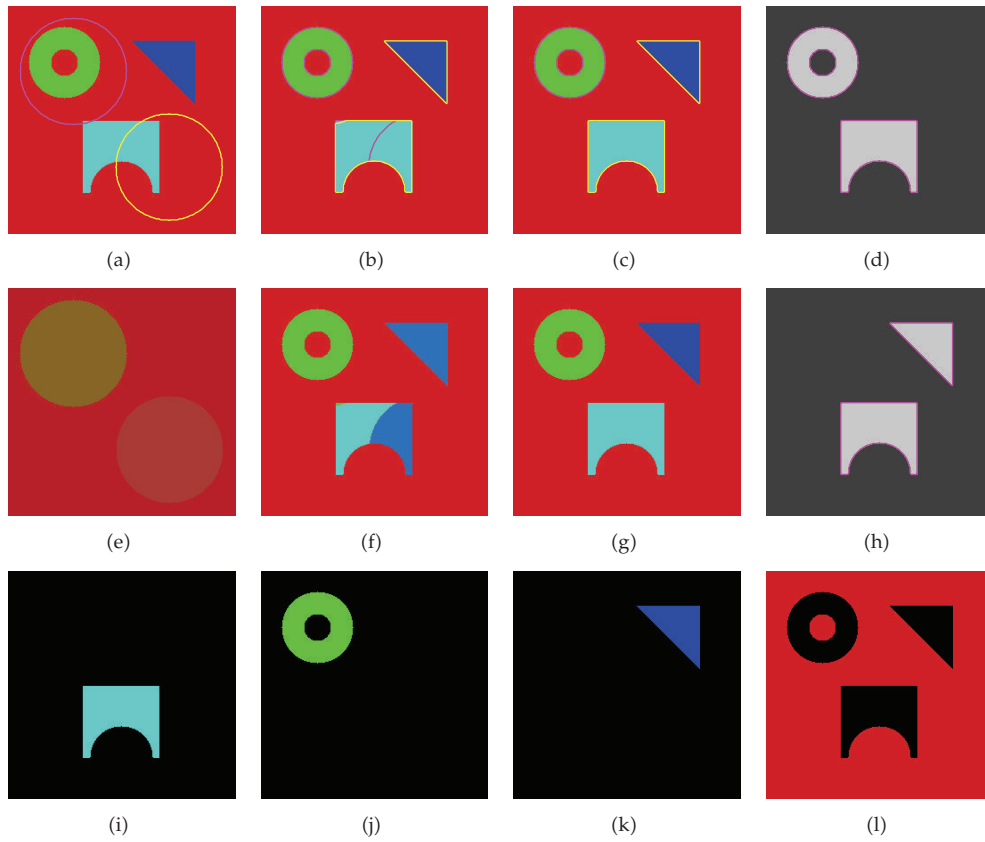


Figure 7: Results for a clean synthetic image with our model. (a)–(c): The curve evolution process; (e)–(g): The corresponding fitting images u ; (d) and (h): The α level sets of the final ϕ_1 and ϕ_2 ; (i)–(l): The final four phases; size = 256×256 .

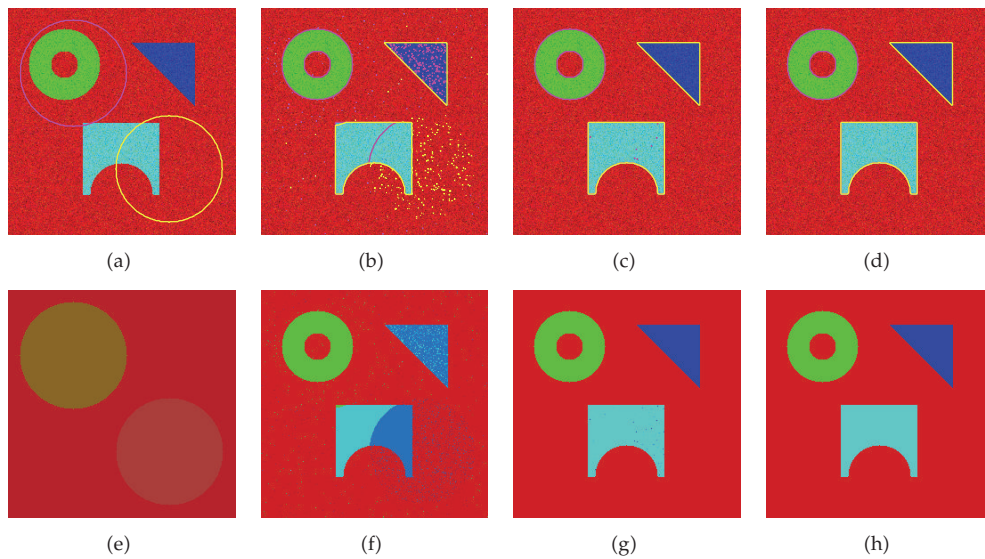


Figure 8: Application of our model to the noisy version of the image from Figure 7. (a)–(d): The curve evolution process; (e)–(h): The corresponding fitting images u .

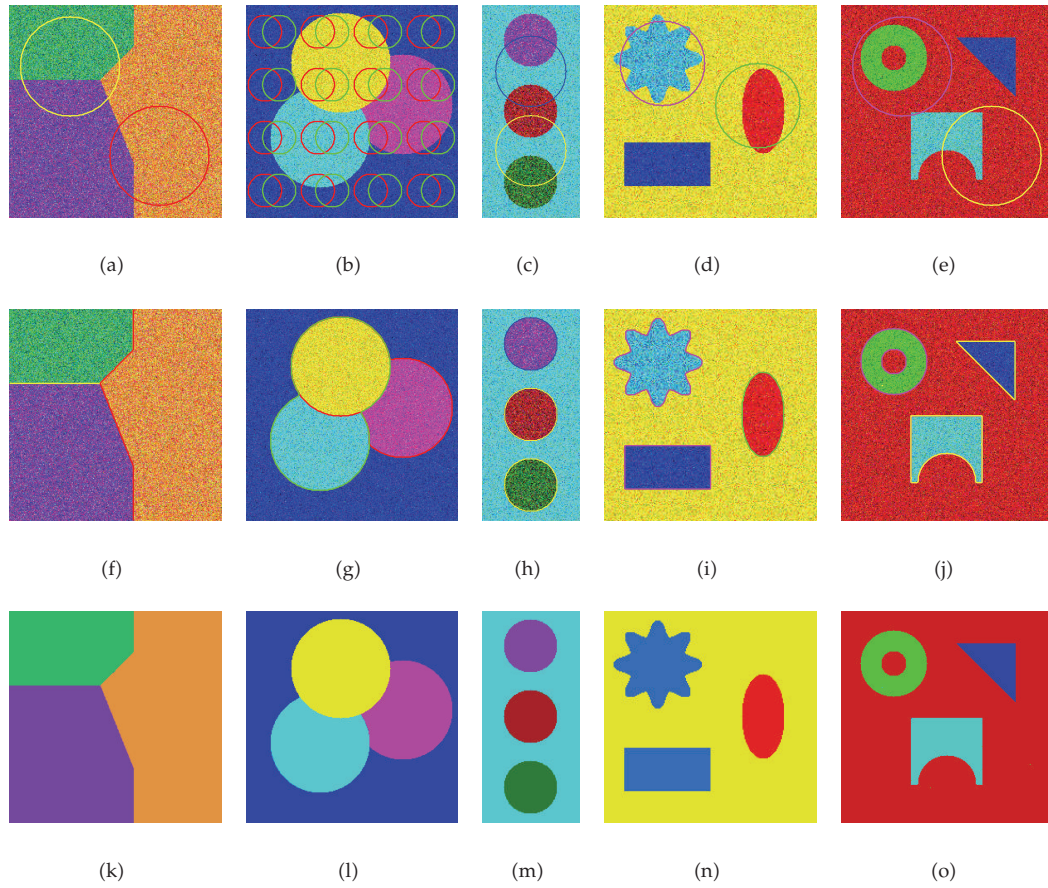


Figure 9: Results for the synthetic images added by random noise with standard deviation 50.0. (a)–(e): The initial contours; (f)–(j): The final contours; (k)–(o): The final fitting images.

Table 1: The CPU time (in second) for the Vese-Chan model and our model for the clean synthetic images from Figures 1, 2, 4, 6, and 7. The sizes of images are 256×256 , 256×256 , 240×110 , 200×200 , and 256×256 , respectively.

	Figure 1	Figure 2	Figure 4	Figure 6	Figure 7
The Vese-Chan model	107.0692	106.8941	60.4858	61.9038	108.2529
Our model	0.488706	0.487907	0.198382	0.203033	0.494109

Table 2: The CPU time (in second) for the Vese-Chan model and our model for the noisy synthetic images from Figures 3, 5, 6, and 8. The sizes of images are 256×256 , 240×110 , 200×200 , and 256×256 , respectively.

	Figure 3	Figure 5	Figure 6	Figure 8
The Vese-Chan model	228.9331	131.9350	74.3890	198.9657
Our model	1.044941	0.432722	0.243982	0.908158

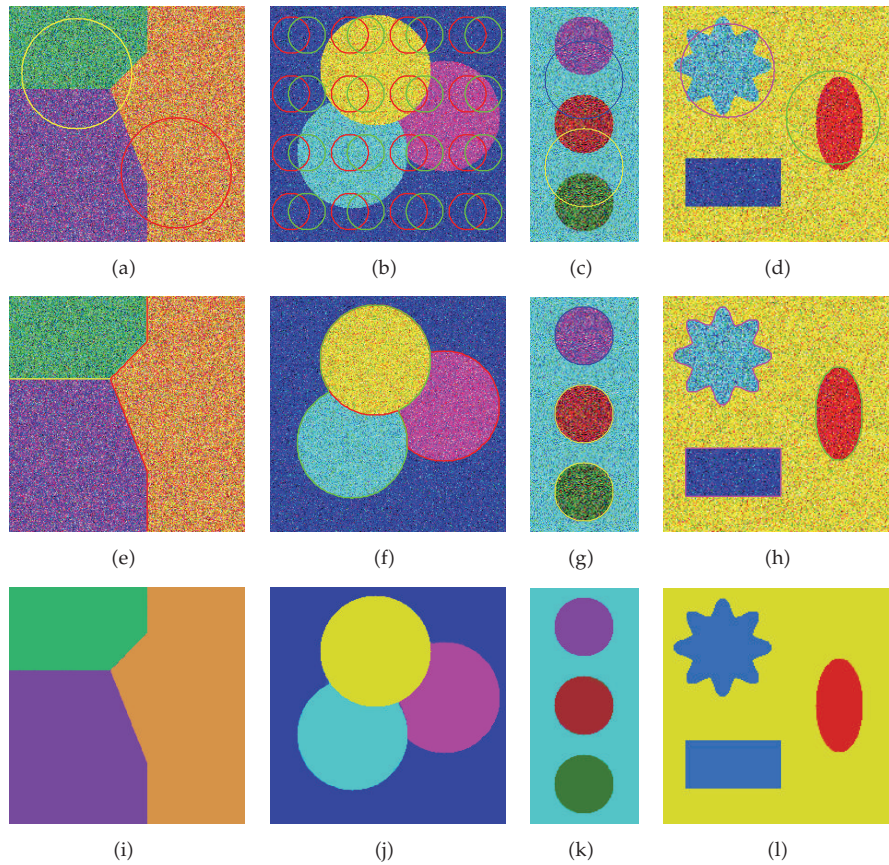


Figure 10: Results for the synthetic images added by random noise with standard deviation 80.0. (a)–(d): The initial contours; (e)–(h): The final contours; (i)–(l): The final fitting images.

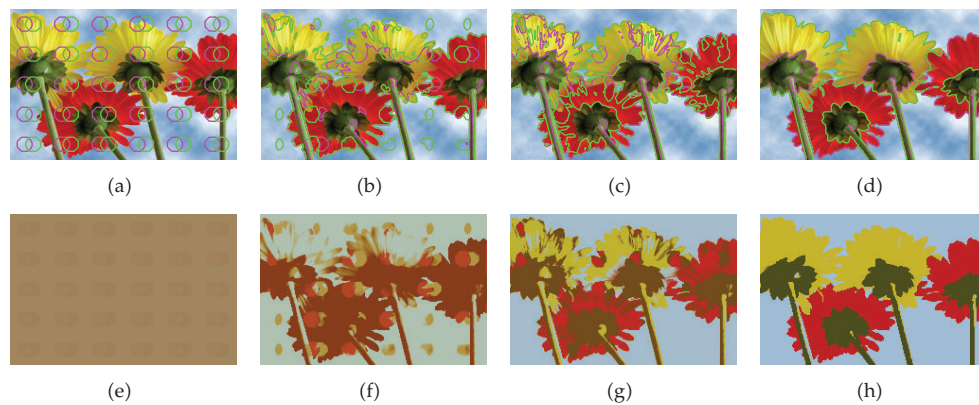


Figure 11: Application of our model to a real color image of flowers. (a)–(d): The curve evolution process. (e)–(h): The corresponding fitting images u ; size = 160×240 .



Figure 12: Final four phases of the flowers image.

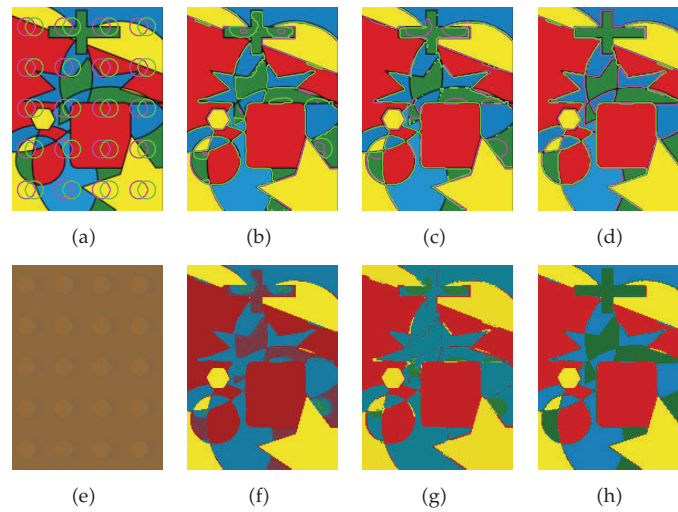


Figure 13: Application of our model to a 4-color image. (a)–(d): The curve evolution process; (e)–(h): The corresponding fitting images u ; size = 185×136 .

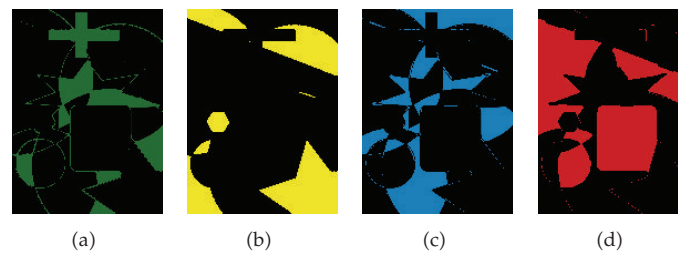


Figure 14: Final four phases of the 4-colors image.

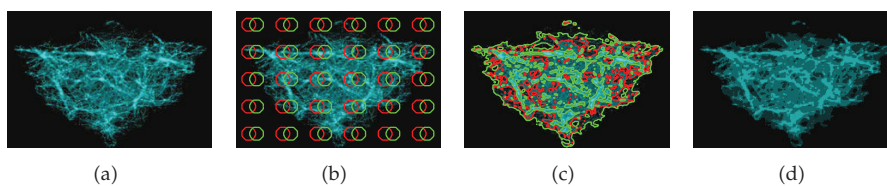


Figure 15: Application of our model to a real color image. (a): The original image; (b): The initial contour; (c): The final contour; (d): The final fitting image; size = 160×240 .

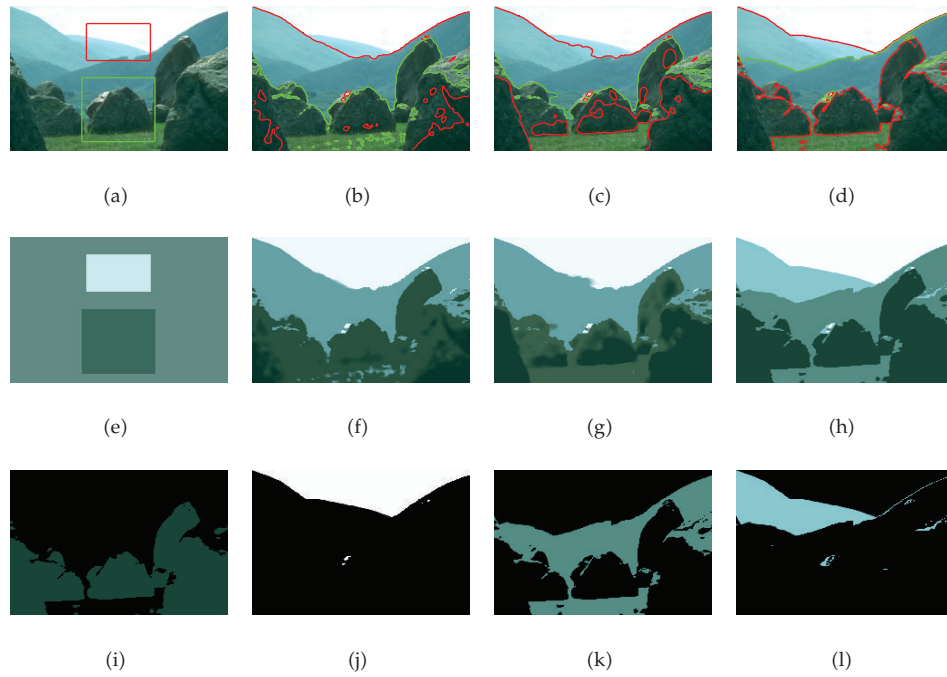


Figure 16: Results for a real color image with our model. (a)–(d): The curve evolution process; (e)–(h): The corresponding fitting images u ; (i)–(l): The final four phases; size = 160×240 .

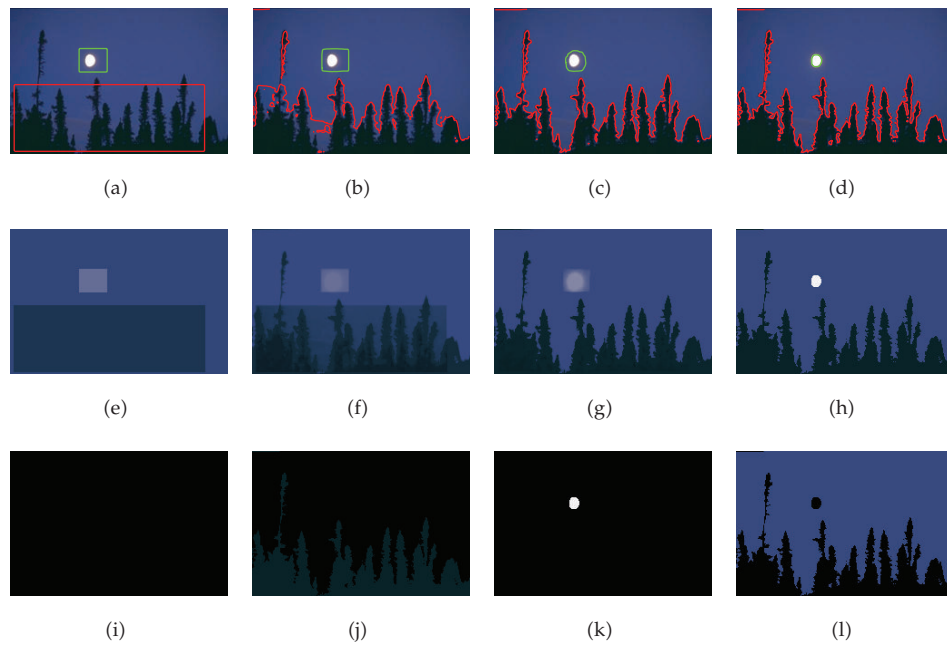


Figure 17: Results for a real color image with our model. (a)–(d): The curve evolution process; (e)–(h): The corresponding fitting images u ; (i)–(l): The final four phases; size = 160×240 .

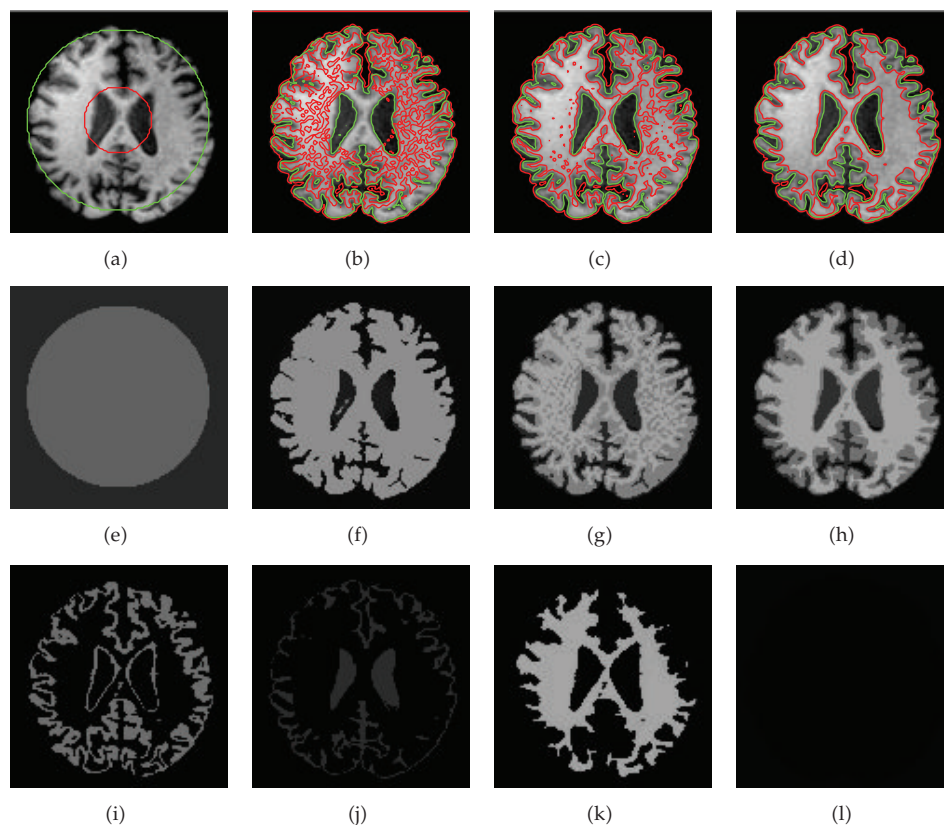


Figure 18: Results for a brain MRI image with our model. (a)–(d): The curve evolution process; (e)–(h): The corresponding fitting images u ; (i)–(l): The final four phases; size = 120×117 .

the Vese-Chan model by applying the split Bregman method. We record the CPU time from our experiments with Matlab codes run on an ACPI Multiprocessor PC, Intel(R) Core(TM)2 Quad CPU Q8200, 2.33 GHz, 2 GB RAM, with Matlab R2010a on Windows XP.

4. Conclusion

In this paper, we propose a new and fast multiphase image segmentation model for color images. Our model is an extension of the model for gray images we proposed in [19]. We have tested our model with many synthetic and real color images. Numerical results show that our model has all the benefits of the Vese-Chan model [3], including robustness even with noisy data and automatic detection of interior contours. However, our model is much more efficient by applying the split Bregman method. Of course, our model has its limitation. Our model is a piecewise constant multiphase segmentation model, and it mainly focuses on homogeneous multiphase images. It only considers the global information of the given image, thus it cannot deal with images with inhomogeneity. In future work, we will extend our model to handle inhomogeneous multiphase images by taking both local and global information into consideration.

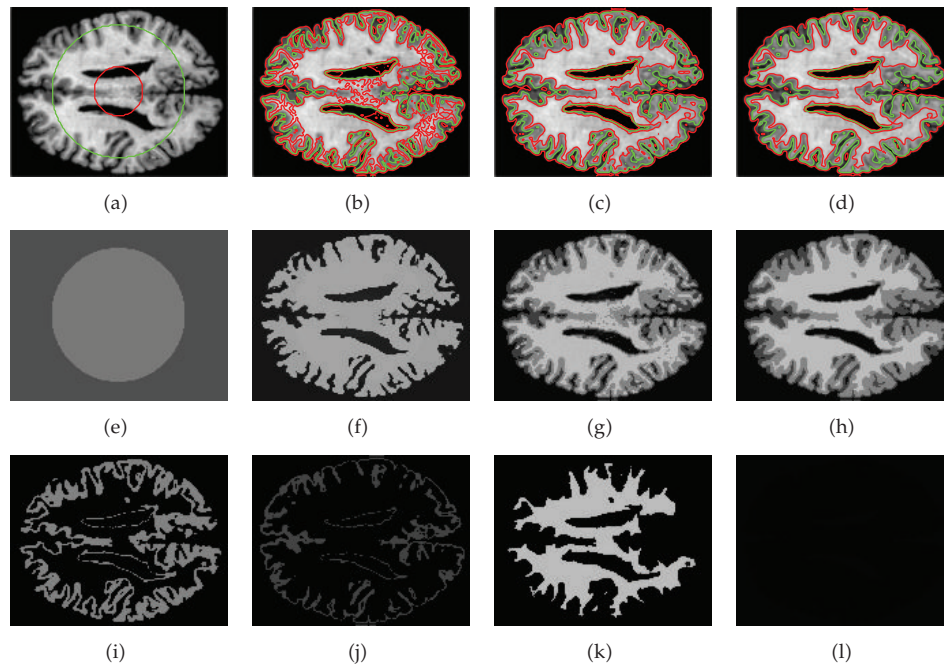


Figure 19: Results for a brain MRI image with our model. (a)–(d): The curve evolution process; (e)–(h): The corresponding fitting images u ; (i)–(l): The final four phases; size = 125×160 .

Acknowledgment

This paper is supported by Natural Sciences Foundation of Heilongjiang Province (A200909).

References

- [1] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [2] T. F. Chan, B. Y. Sandberg, and L. A. Vese, "Active contours without edges for vector-valued images," *Journal of Visual Communication and Image Representation*, vol. 11, no. 2, pp. 130–141, 2000.
- [3] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, 2002.
- [4] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [5] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.
- [6] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.
- [7] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: a level set approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158–175, 1995.
- [8] H.-C. Hsin, "Texture segmentation in the joint photographic expert group 2000 domain," *IET Image Processing*, vol. 5, no. 6, pp. 554–559, 2011.
- [9] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

- [10] G. Sapiro, "Color snakes," *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 247–253, 1997.
- [11] J. Shah, "Curve evolution and segmentation functionals: application to color images," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '96)*, vol. 1, pp. 461–464, Lausanne, Switzerland, 1996.
- [12] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, "A variational level set approach to multiphase motion," *Journal of Computational Physics*, vol. 127, no. 1, pp. 179–195, 1996.
- [13] C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia, "A variational model for image classification and restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 5, pp. 460–472, 2000.
- [14] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for supervised texture segmentation," *International Journal of Computer Vision*, vol. 46, no. 3, pp. 223–247, 2002.
- [15] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.
- [16] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for L1-minimization with applications to compressed sensing," *SIAM Journal on Imaging Sciences*, vol. 1, no. 1, pp. 143–168, 2008.
- [17] T. Goldstein, X. Bresson, and S. Osher, "Geometric applications of the split Bregman method: segmentation and surface reconstruction," *Journal of Scientific Computing*, vol. 45, no. 1–3, pp. 272–293, 2010.
- [18] Y. Yang, C. Li, C.-Y. Kao, and S. Osher, "Split Bregman method for minimization of region-scalable fitting energy for image segmentation," in *Proceedings of the 6th International Symposium on Visual Computing*, vol. 6454 of *Lecture Notes in Computer Science*, pp. 117–128, Springer, Berlin, Germany, 2010.
- [19] Y. Yang and B. Wu, "Improved active contour model for multiphase image segmentation based on the Vese-Chan model," submitted to: *Journal of Applied Mathematics*.
- [20] T. F. Chan, S. Esedoglu, and M. Nikolova, "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM Journal on Applied Mathematics*, vol. 66, no. 5, pp. 1632–1648, 2006.
- [21] X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran, and S. Osher, "Fast global minimization of the active contour/snake model," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 2, pp. 151–167, 2007.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

