Research Article

Fast Image Segmentation Based on Efficient Implementation of the Chan-Vese Model with Discrete Gray Level Sets

Songsong Li and Qingpu Zhang

School of Management, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Songsong Li; lisongsong@hit.edu.cn

Received 3 December 2012; Accepted 28 January 2013

Academic Editor: Clara Ionescu

Copyright © 2013 S. Li and Q. Zhang. 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.

A new image segmentation based on fast implementation of the Chan-Vese model is proposed. This approach differs from previous methods in that we do not need to solve the Euler-Lagrange equation of the underlying variational problem. First, through experiments, we observe that for the smooth image segmentation, Chan-Vese model (CVM) can be simplified. Utilizing the Gaussian low pass filter, we pretreat the original image and regularize the level curves. Then, we calculate the energy directly on discrete gray level sets, find the minimizer of the energy, and obtain the segmentation results. We analyze the algorithm and prove that under discrete gray level sets, the global minimum of the energy is same as the one obtained by the previous methods. Another advantage of this method is that the reinitialization is not needed. Since there are at most 255 discrete gray level sets, the algorithm improves the computational speed dramatically. And the complexity of the algorithm is $O(N)$, where $N$ is the number of pixels in the image. So even for the large images, it is also very efficient. We apply our segmentation algorithm to synthetic and real world images to emphasize the performances of our model compared with other segmentation models.

1. Introduction

Images are the proper 2D projections of the 3D world containing various objects. To successfully reconstruct the 3D world, at least approximately, the first crucial step is to identify the regions in images that correspond to individual objects. This is the well-known problem of image segmentation. It has broad applications in variety of important fields such as computer vision and medical image processing.

Variational methods [1–16] have been extensively used and studied in image segmentation in the past decade because of their flexibility modeling and various advantages in the numerical implementation. These methods drive one or more initial curve(s), based on gradient and/or region information in the image, to the boundaries of objects in that image. The basic idea of variational methods is to minimize an energy. This functional generally depends on the features of the image. The classical way to solve the minimization problem is to solve the corresponding Euler-Lagrange equation. For instance, relatively early, Mumford and Shah [1] introduced a celebrated segmentation model by minimizing an energy functional that penalizes smoothness within regions and the length of their discontinuity contours. Recently, Chan and Vese [4] developed an active contour without edge model to deal with the problem of image segmentation by using the level-set framework introduced by Osher and Paragios [17]. Tsai et al. [5] also independently developed a segmentation method which is similar to it. The active contour methods [4–8, 10, 13] based on level-set framework have several advantages. First, they can deal with topological changes such as break and merger. Second, intrinsic geometric elements such as the normal vector and the curvature can be easily expressed with respect to the level-set function. Third, this level-set framework can be extended and applied in any dimension.

However, the active contour methods based on level-set framework have some limitations. First, these methods are usually implemented by solving the partial differential equations (PDEs) and thus computational efficiency sharply decline because of numerical stability constraints. Particularly, the signed distance reinitialization procedure is necessary. It severely limits the efficiency of Chan-Vese model. Second, most methods have the initialization problem [6]: different initial curves produce different segmentations because of
the nonconvexity of Chan-Vese model. More recently, some of the researchers develop fast algorithms [14–16, 18–20] for the Chan-Vese image segmentation model. In [14, 15, 18], the authors develop fast algorithms based on calculating the variational energy of the Chan-Vese model directly without the length term, that is, solving PDEs. In [14], the authors develop a fast method for image segmentation without solving the Euler-Lagrange equation of the underlying variational problem proposed by Chan and Vese [4]. Instead, they calculate the energy directly and check if the energy is decreased when they change a point inside the level-set to outside or vice versa. Later, various modifications of Chan-Vese model, related to different aspects of the image analysis, have been proposed, such as adaptive segmentation of vector images [7, 8, 19, 20] knowledge-based segmentation [13].

An efficient implementation method for the Chan-Vese model is proposed in this paper. Since the simplified Chan-Vese model is without the length term, the new method does not have to solve PDEs. Our method is the hybrid of the discrete simplified Chan-Vese model and the discrete gray level-set framework. First, we decouple the Chan-Vese model into two stages: in the first stage, utilizing Gaussian low pass filter to pretreat the original image and obtaining some appropriate smooth versions of the original image; in the second stage, segmenting the smooth image by Chan-Vese model without the length term. Second, we implement the previous algorithm based on the discrete gray level-set framework. Our segmentation method is also divided into two stages: in the first stage, we smooth the image into required scale; in the second stage, we calculate the energy directly on discrete gray level sets and find the minimizer of the energy. Each stage is independent and at each stage the method is flexible. The new method bears some similarities to [14, 15, 18], but we calculate the energy directly on discrete gray level sets, which is a new framework. More complicated issues which are not considered in [14], such as sensitivity to noise, are discussed. The segmentation method can also deal with complicated image, and the CPU time of large size/complicated images is not too long. First, we do not need to solve the Euler-Lagrange equation of the underlying variational problem, so the initial conditions and the procedure of reinitialization are not needed. Second, in the second stage of our algorithm, the main computation process is adding operators and logical operators, which costs little CPU time, so the complexity of the algorithm is O(N).

This paper is organized as follows. In Section 2, we present the Chan-Vese piecewise smooth active contours model [4] (also the algorithm of Tsai et al. [5]) for image segmentation. In Section 3, we give a full account of our model. Experimental results are presented in Section 4, and the final section is our conclusion.

2. The Main Idea of the Chan-Vese Active Contours Model

In this section, we present the main idea of the Chan-Vese active contours model in order to develop the idea of the new fast hybrid level-set algorithm.

The active contour model proposed by Chan and Vese [4] is a particular case of the Mumford-Shah model [1]. Consider an image \( u^0 \) with the domain \( \Omega \subset \mathbb{R}^2 \). Let the segmenting closed curve \( C \) (active contour) divide \( \Omega \) into two partitions \( \Omega_1 \) and \( \Omega_2 \), which is corresponding to the image subdomains inside and outside the curve \( C \), respectively. It minimizes the following energy functional:

\[
F(c_1, c_2, C) = v \cdot \text{Length}(C) + \sum_{i=1}^{2} \int_{\Omega_i} \left( u^0(x, y) - c_i \right)^2 dxdy, \quad i = 1, 2,
\]

where \( c_i \) is the average value of \( u^0(x, y) \) in each region \( \Omega_i \) and \( v \) is a positive constant.

Using the level-set method in [17], the authors replace the unknown curve \( C \) by the level-set function \( \phi(x, y) \) defined by

\[
C = \{(x, y) \mid \phi(x, y) = 0\},
\]

\[
\Omega_1 = \text{inside}(C) = \{(x, y) \mid \phi(x, y) > 0\},
\]

\[
\Omega_2 = \text{outside}(C) = \{(x, y) \mid \phi(x, y) < 0\}.
\]

Denote the Heaviside function \( H \)

\[
H(z) = \begin{cases} 
1, & \text{if } z \geq 0, \\
0, & \text{if } z < 0.
\end{cases}
\]

Then the energy functional (1) can be written in terms of the level-set formulation as

\[
F(c_1, c_2, \phi) = v \int_{\Omega} |\nabla H(\phi)| dxdy + \int_{\Omega} \left( u^0 - c_1 \right)^2 H(\phi) dxdy
\]

\[
+ \int_{\Omega} \left( u^0 - c_2 \right)^2 (1 - H(\phi)) dxdy.
\]

The final segmentation curve could be obtained from the minimizer of the energy functional (4) with respect to \( \phi \). Using a gradient descent method, minimizing (4) yields the following problem:

\[
\frac{\partial \phi}{\partial t} = \delta(\phi) \left[ v \cdot \nabla \left( \frac{\phi}{|\nabla \phi|} \right) - \left( u^0 - c_1 \right)^2 + \left( u^0 - c_2 \right)^2 \right],
\]

where \( \delta(\phi) = H'(\phi) \) in the distribution sense and

\[
c_1 = \frac{\int_{\Omega_1} u^0 H(\phi) dxdy}{\int_{\Omega} H(\phi) dxdy}, \quad c_2 = \frac{\int_{\Omega_2} u^0 (1 - H(\phi)) dxdy}{\int_{\Omega_2} (1 - H(\phi)) dxdy}.
\]

From the solution of the problems (5) and (6), we obtain the evolution of \( C(t) \), defined by \( \{(x, y) \mid \phi(x, y, t) = 0\} \), which is the boundary between the sets \( \{(x, y) \mid u^0 = c_1\} \) and \( \{(x, y) \mid u^0 = c_2\} \). Therefore, the original image is segmented \( u^0 \) into two parts.

It is noticed that (5) is a nonlinear parabolic partial differential equation. So it requires expensive computation
since the solution is on a large time domain (i.e., from the initial curve location to the finally achieved steady state). In particular, we need to reinitialize the level-set function $\phi$ to a signed distance function in each iteration. If we do not do it, when the level-set function $\phi$ becomes very sharp or flat during the evolution, it makes computation highly inaccurate. Although some semi-implicit methods in [21, 22] could be used to partially alleviate the computational burden, the process of reinitialization is very important and may not be avoided by using some level-set methods. All in all, the Chan and Vese algorithm works very well for image segmentation, but solving the Euler-Lagrange equations (5) and (6) and reinitializing the level-set function cost a lot of CPU time.

In Section 3, we will propose a fast hybrid gray level-set algorithm in order to overcome the previous shortcomings of the Chan-Vese model.

3. Our Method

In this section, we show a two-stage scheme for implementation of the piecewise constant segmentation model. More precisely, the smooth version of the original image is first obtained by some smooth filters, and then minimizing the modified Chan-Vese energy functional on the gray level sets, the image is divided into two subregions.

3.1. Segmentation for the Smooth Image. In the Chan-Vese segmentation model, the parameter $\nu$ in (4) is a weight of the regularizing term, which controls the length of the zero level-set. When the noise level is high, the parameter $\nu$ should be large and only large objects are detected. When the noise level is low, the parameter $\nu$ could be small and objects of small size are detected. Because the Chan-Vese model is independent of the gradient of the image, the input image can be smooth enough [4]. So the segmentation method for the smooth image should be different from the method for the noise image.

In Figure 1, we present results of the noise image and some smooth versions of the noise image by the Chan-Vese model with the same parameter $\nu = 0$, respectively. It is noticed that for the smooth image, the parameter $\nu$ could vanish in Chan-Vese model, and the good segmentation is also obtained.

Now, based on the previous facts, we assume that the input image is reasonably smooth, and then the parameter $\nu$ could vanish in (4). Hence the following simplified Chan-Vese model without the length term is considered:

$$ F(c_1, c_2, \phi) = \int_{\Omega} (u - c_1)^2 H(\phi) \, dx \, dy \tag{7} $$

where $\delta(\phi) = H'(\phi)$ in the distribution sense and

$$ c_1 = \frac{\int_{\Omega} u H(\phi) \, dx}{\int_{\Omega} H(\phi) \, dx}, \quad c_2 = \frac{\int_{\Omega} u (1 - H(\phi)) \, dx}{\int_{\Omega} (1 - H(\phi)) \, dx}. \tag{9} $$

The simplified Chan-Vese algorithm is sketched in the following.

**Algorithm 1.** The simplified Chan-Vese algorithm is the following two-stage scheme.

1. By the smooth filter obtain some appropriate smooth versions $u$ of the noise image $u^0$.

2. Calculate the minimum of the simplified Chan-Vese energy (7).

(i) Initialize $\phi^0, n = 0$.

(ii) Compute $c_1(\phi^n)$ and $c_2(\phi^n)$ by (9).

(iii) Solve the equation in $\phi$ from (8), to obtain $\phi^{n+1}$.

(iv) Reinitialize $\phi$ locally (this step is optional).

(v) Check whether the solution is stationary. If not, $n = n + 1$ and repeat (ii)–(v).

The main difference between Algorithm 1 and the original Chan-Vese model is whether the input image requires smooth pretreatment. In Algorithm 1, there are a lot of methods to smooth image, such as Gaussian low pass filter, averaging filter, Laplacian filter, and enhancing filter based on PDEs. In this paper, we choose the simplest and possibly one of the most popular ways, Gaussian low pass filter. Though all image information, such as the noise and edges, will be indistinguishably destroyed by this filter, it tampers with the segmentation result little, since the Chan-Vese model makes use of a stopping edge function based on the density of the image instead of the gradient. It is noticed that if the images are noisy, we do a preprocessing step to smooth or enhance the images firstly, or else this step is needless. In Figures 2 and 3, we present the other two experiments by Algorithm 1. In the two experiments, Figure 3(a) is smoother than Figure 2(a), so Figure 3(a) needs to be smoothed more. For the level-set method, it is inevitable to reinitialize the level-set function for the stability of algorithms. However, for Algorithm 1, we need not reinitialize the level-set function $\phi$ to a signed distance function in each iteration, which is different from the Chan-Vese model. In Section 4, from the numerical experiments, we show the interesting phenomenon.

Now, let us observe the following facts: for the noise image, the gray level lines are disorderly and irregular, while for the smooth image, the gray level lines are smooth and regular. These basic remarks are illustrated in Figure 4. In Figure 4(i), the contour line is very approximate to the real edges of the original image. Hence, we think the best segmentation result is one of the contour lines. How can the best segmentation result, that is, the most appropriate contour line, be obtained? In the next subsection, we will answer the question and then propose the new algorithm.

Let \( u_{i,j} \), for \((i, j) \in D \equiv \{1, \ldots, M\} \times \{1, \ldots, N\} \), be the gray level of a true \( M \)-by-\( N \) image \( u \) at pixel location \((i, j)\), and let \([s_{\text{min}}, s_{\text{max}}]\) be the range of \( u \), that is, \( s_{\text{min}} \leq u_{i,j} \leq s_{\text{max}} \). Let \( D_1 \subset D \) and \( D_2 = D \setminus D_1 \), and then the image \( u \) is divided into two regions by the pixel location. Instead \( u^0 \) by \( u \), \( H(\phi) \) by \( D_1 \), and \( (1 - H(\phi)) \) by \( D_2 \), respectively. Then minimizing the energy (7) is changed into minimizing

\[
F_1(c_1, c_2, D_1, D_2) = \sum_{(i,j) \in D_1} (u_{i,j} - c_1)^2 + \sum_{(i,j) \in D_2} (u_{i,j} - c_2)^2,
\]

with

\[
c_1 = \frac{\sum_{(i,j) \in D_1} u_{i,j}}{|D_1|}, \quad c_2 = \frac{\sum_{(i,j) \in D_2} u_{i,j}}{|D_2|},
\]

where \(|D_1| = \sum_{(i,j) \in D_1} 1\) is the number of pixels in \( D_1 \) and \(|D_2| = \sum_{(i,j) \in D_2} 1\) is the number of pixels in \( D_2 \). When the energy \( F_1 \) reaches a minimum, the best segmentation results are obtained, that is, the subregion \( D_1 \) and subregion \( D_2 \). It is noticed that since the selection of \( D_1 \) and \( D_2 \) is arbitrary, there are lots of combinations \((D_1, D_2)\), so minimizing the energy \( F_1 \) is difficult. Now, we try to narrow the scope of the probable combinations. For any \( K \in [s_{\text{min}}, s_{\text{max}}] \), the image \( u \) is divided into two subregions, that is, \( D^K_1 \equiv \{(k, l) : u_{k,l} < K\} \) and \( D^K_2 \equiv \{(k, l) : u_{k,l} \geq K\} \).
Figure 2: Segmentation results by Algorithm 1. (a) The noise image. (b) The smooth image by Gaussian low pass filter with \( \sigma = 0.8 \). (c) The segmentation contour line. (d) The segmentation result.

**Definition 2** (discrete gray level-set). The \( K \)-discrete gray level-set \( D^K \) which is the set of pixel location \((i, j)\) is defined as follows:

\[
D^K = \{(i, j) : u_{i,j} < K\},
\]

where \( u_{i,j} \) is the gray level of the image \( u \) at pixel location \((i, j)\).

Then \( D^K = D \setminus D_1 \), for \( D^K = \{(k, l) : u_{k,l} < K\} \). Let \( \mathcal{A} = \{(D^K, D_2^K), K \in [s_{\min}, s_{\max}]\} \), and then the element number of the set \( \mathcal{A} \) is \( s_{\max} - s_{\min} + 1 \). For every level \( K \), the image \( u \) is always divided into two disjointed subregions by \( D^K_1 \) and \( D^K_2 \). In Figure 4, the boundary of \( D^K_2(D^K_1) \) is displayed at the different levels \( K \), for the image \( u \).

**Theorem 3.** Minimizing the energy functional \( F_1 \) is equivalent to

\[
\min_{(D_1, D_2) \in \mathcal{A}} F_1(c_1, c_2, D_1, D_2) = \sum_{(i, j) \in D_1} (u_{i,j} - c_1)^2 + \sum_{(i, j) \in D_2} (u_{i,j} - c_2)^2,
\]

where \( c_1 \) and \( c_2 \) are defined as (11).

**Proof.** If \( (D_1, D_2) \in \mathcal{A} \), then for any \((i, j) \in (D_1, (i', j') \in D_2\), we have \( u_{i,j} \leq u_{i',j'} \) and \( c_1 \leq c_2 \), where \( c_1 \) and \( c_2 \) are defined as (11). It is obvious that \( \min F_1 \leq F_2 \). We only need to prove \( F_2 \leq F_1 \).

Assume there exist two subdomains \( D_1 \) and \( D_2 \) such that \( F_1 \) attains its minimizer. If there exist \((i_1, j_1) \in D_1 \) and \((i_2, j_2) \in D_2 \) such that \( u_{i_1,j_1} > u_{i_2,j_2} \), then we denote \( D'_1 = D_1 - (i_1, j_1) + (i_2, j_2) \) and \( D'_2 = D_2 - (i_2, j_2) + (i_1, j_1) \).

Comparing the energy \( F_1(c_1, c_2, D_1, D_2) \) and \( F'_1(c'_1, c'_2, D'_1, D'_2) \), we get

\[
F_1 - F'_1 = \sum_{(i,j) \in D_1} (u_{i,j} - c_1)^2 + \sum_{(i,j) \in D_2} (u_{i,j} - c_2)^2
- \sum_{(i,j) \in D'_1} (u_{i,j} - c'_1)^2 - \sum_{(i,j) \in D'_2} (u_{i,j} - c'_2)^2.
\]

Since \( c'_1 < c_1 \) and \( c'_2 > c_2 \), we have

\[
2c_1 - 2c_2 < 0.
\]

Hence we have the following.

**Theorem 4.** Minimizing the energy functional \( F_1 \) is equivalent to

\[
\max_{(D^K_1, D^K_2) \in \mathcal{A}} \mathcal{E}(K) \equiv \left\{ |D^K_1|c_1^2 + |D^K_2|c_2^2 \right\},
\]

where \( K \in [s_{\min}, s_{\max}] \) and \( c_1 \) and \( c_2 \) are defined as (11).
Now, if the energy functional $E$ reaches a maximum, the best segmentation results are obtained, that is, the subregion $D^K_1$ and subregion $D^K_2$. Since $K = s_{\min} \cdot s_{\min} + 1, \ldots, s_{\max}$, the energy functional $E$ has $s_{\max} - s_{\min} + 1$ cases, and then the maximum of $E$ is easily found. The algorithm is sketched here in the following.

**Algorithm 5.** The method of maximizing the following functional $E$.

1. Sweep the image $u$ once, record the number of all pixels at every gray level of the image $u$ which range from $s_{\min}$ to $s_{\max}$.

2. Calculate the energy $E(K)$ by (18), for $K \in [s_{\min}, s_{\max}]$, and find the maximizer $K$.

3. The image $u$ is divided into two subregions, that is, $D^K_1 = \{(i, j) : u < K, (i, j) \in D\}$ and $D^K_2 = \{(i, j) : u \geq K, (i, j) \in D\}$.

**Remark 6.** In fact, we restrict $D^K_1 = \{(i, j) : u < K, (i, j) \in D\}$, $D^K_2 = \{(i, j) : u \geq K, (i, j) \in D\}$, since if two pixels with the same value $K$ belong to both $D^K_1$ and $D^K_2$, it will be ambiguous to determinant. However, the pixel number of $u$ at $K$ gray level is a little and cannot influence the maximizer of the energy $E(K)$. The experiment result shows that our approximated method is still efficient.

3.3. **The Final Method.** Based on Algorithm 5, the following is the new two-phase scheme for image segmentation.

**Algorithm 7.** (i) **(Smooth the original image):** For the input noise image $u^0$, use the Gaussian smooth filter to obtain the smooth image $u_S$ (If the input image is noiseless, this step is optional).

(ii) **(Segmentation):** Use Algorithm 5 to obtain the segmentation results for the smooth image $u_S$, that is.

1. Sweep the image $u_S$ once, record the number of all pixels at every gray level of the image $u_S$ which range from $s_{\min}$ to $s_{\max}$.

2. Calculate the energy $E(K)$ by (18), for $K \in [s_{\min}, s_{\max}]$, and find the maximizer $K$.

3. The image $u^0$ is divided into two subregions, that is, $D^K_1 = \{(i, j) : u_S(i, j) < K, (i, j) \in \mathcal{H}\}$ and $D^K_2 = \{(i, j) : u_S(i, j) \geq K, (i, j) \in \mathcal{H}\}$, and this completes a segmentation.

**Remark 8 (segmentation for various types of noisy image).** There are lots of methods to obtain the smooth image in the first phase of the new method.

(i) If the type of noise is “salt and pepper,” for example, the AMF (adaptive median filter) can be selected.

(ii) If the noise is “addition gauss noise,” for example, the gaussian lower-pass filter, the TV method [23], the PM method [24], and the other anisotropic diffusion method can be used to smooth the original image.

(iii) If the noise is “Poisson noise,” for example, the Variational method [25] can be used to denoise the original image.

In the new algorithm, the segmentation is only dependent on the smooth image but not sensitive to the smoothing scale. In Figure 5, using the new algorithm, we deal with the image with different types of noise.

**Remark 9.** In the new algorithm, we select the Gauss low pass filter to obtain the smooth image. However the filter is not necessary. If the image is corrupted by the different way, we will select the different filters, such as average filter, Laplacian filter, and unsharp filter. For the addition Gaussian noise, the gaussian lower-pass filter may be the best tradeoff between time and segmentation.
Figure 4: The level lines at different gray levels. (a) The noise image with Gaussian white noise of mean 0 and variance 0.2. (b) The smooth image by Gaussian low pass filter with $\sigma = 1.5$. (c)–(g) The level lines of the noise image at the gray level, 60, 80, 120, 180, and 200. (h)–(l) The level lines of the smooth image at the gray level, 60, 80, 120, 180, and 200.

4. Simulations

In this section, numerical examples on some synthetic and real-world images are presented to illustrate the effectiveness of the new two-phase scheme in Section 4 by comparing it with Chan-Vese model.

4.1. Configuration. In the new algorithm, the first stage is smoothing the original image by low pass filter. The low pass filter is generally made by convolution with Gaussian of increasing variance, which is as follows:

$$u = G_\sigma \ast u^0,$$

(19)

where $G_\sigma = (1/\sigma) \exp[-|x|^2/4\sigma^2]$ and $\ast$ denotes the convolution operation. Koenderink [26] noticed that the convolution of signal with Gaussian at each scale is equivalent to the solution of the heat equation with the signal as initial datum $u^0$; that is,

$$\frac{\partial u}{\partial t} = \Delta u,$$

(20)

Assuming a time step size of $\Delta t$ and a space grid size of $h$, we quantize the time and space coordinates as follows:

$$t = n\Delta t, \quad n = 0, 1, 2, \ldots,$$

$$x = ih, \quad i = 0, 1, 2, \ldots, I,$$

$$y = jh, \quad j = 0, 1, 2, \ldots, J,$$

(21)
Figure 5: Segmentations for the Cameraman image with different types of noise. (a) The noise image corrupted by 50% salt and pepper noise. (b) The noise image corrupted by Poisson noise with the variance 4. (c)-(d) The segmentation results of the new algorithm. (e)-(f) The segmentation counter of the new algorithm.
where $Ih \times Jh$ is the size of image support. The classical five-point explicit numerical schemes of the heat equation (20) are as follows:

$$
\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \frac{u_{i+1,j}^{n} + u_{i-1,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n} - 4u_{i,j}^{n}}{h^2},
$$

or equivalently

$$
u_{i,j}^{n+1} = u_{i,j}^{n} + \lambda \left( u_{i+1,j}^{n} + u_{i-1,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n} - 4u_{i,j}^{n} \right),
$$

where $\lambda = \Delta t/h^2 < 1/4$ is for the stable of numerical scheme.

The simulations are performed in Matlab R2007b on a 2.8 GHz Pentium 4 processor. For comparison purpose, the Chan-Vese method (CVM) [4] is also tested. We utilize a locally one-dimensional (LOD) scheme for CVM, which is an unconditional scheme [10].

4.2. Segmentation Performance. In Figures 6 and 7, we illustrate the performance of CVM, Algorithms 1 and 7 on synthetic noise Test01 and Test02 images. Among the segmentations, all algorithms give similar and good performances. For CVM (the Chan-Vese method), utilizing the
unconditionally LOD scheme, the time step size can be sufficiently large to reduce the iteration steps (only 6/7 steps in Table 1). However, re-initializing the level-set function costs a lot of CPU time (in Table 1).

In Figure 8, on the synthetic noise Test03 image, we illustrate the results of Algorithm 1 with reinitialization and without reinitialization, respectively. One can see that in the segmentation process, reinitializing the level-set function or not influences the results of Algorithm 1 little (Figures 8(d)–8(g)). In CVM, reinitialization lets the level-set function $\phi$ not be very sharp, and $|\nabla \phi| = 1$ in mathematics. Algorithm 1 does not need $\nabla \phi$, so reinitialization is not necessary.

In Figures 9 and 10, we illustrate the results of Algorithm 1 without reinitialization and Algorithm 7 about a real CT and Stone image. Few differences between the segmented images are observed, but our method works much more faster than CVM (in Table 1).

4.3. Robustness with the Smooth Scale $\sigma$. Instead of the low pass filter, we use the scheme (23) to smooth the image. Hence, the smooth scale $\sigma$ depends on the parameter $\lambda$ and the iteration step. The smoother of the original image, the more regular of the optimal gray level line. But if we regularize
too much, we will remove details from the image (in figures, the corner information is lost). In Figure 11, we illustrate how our model works on a real nebula image with $\lambda = 0.1$ and different iteration steps. In Figure 11, we can see that in different smooth scales $\sigma$, the gray level lines are different and evolved into the more regular ones. Compared with CVM, Algorithm 1 need not the initial level-set function $\phi_0$ and reinitialization. In Algorithm 1, we just adjust the parameter $\lambda$ and the iteration step, and always obtain the global minimum of the energy functional.

4.4. Computational Complexity. We end this section by considering the complexity of our algorithm. Our algorithm requires two phases: smoothing the original image and segmenting. The first phase is done by the scheme (23). Similar to other low pass filters, it is fast. It is also worth mentioning that one of the efficient implementations of the scheme (23) is FFT (Fast Fourier Transform Algorithm) and the complexity of this stage is $O(N)$, where $N$ is the number of pixels in the image. In the second segmentation phase, our algorithm only sweeps the image once, so the complexity of
the stage is no more than $O(N)$. In Table 1, we compare the CPU time needed of all three algorithms. We summarize that the CPU time of our Algorithm 7 is about 0.01–0.08 seconds which is the fastest in three algorithms.

4.5. Comparison with Some Other Segmentation Methods.
There are some classical segmentation methods, such as the watershed algorithm [27], the Canny filter [28], and the Sobel filter. In Figure 12, the smooth versions of the images mentioned previously are processed by these algorithms. Watershed algorithm provides the advantages of stabilization and speediness, but is prone to oversegmentation in Figures 12(a)–12(c). In Figures 12(d)–12(f), the segmentation result of the Sobel operator is sensitive to the threshold: if the threshold is too small, the redundant edges will be detected; if the threshold is too big, the detected edges will be broken. The Canny filter is sensitive to noise in Figures 12(h)–12(i).

5. Summary and Further Research Directions
In this paper, we have proposed and implemented a new image segmentation algorithm based on the Chan-Vese
active contour model. The discrete gray level-set method is employed in our numerical implementation. This algorithm works in two steps, smoothing the noisy image by using the heat equation filter method and then using the new discrete gray level-set method to segment the region of the original image.

In the Chan-Vese segmentation algorithm, the initialization of the level-set functions is a difficult problem. In the proposed new segmentation algorithm, the initialization is not required. And each step is simple and easily achieved. In the first step, there are a lot of algorithms to get the smooth version of the original image, and in the second step, we sweep the image only once and calculate (18) at every gray level (in fact, only 256 gray level sets) and then find the optimal gray level. In Table 1, we show the CPU time of the Chan-Vese method and our proposed method. Obviously, our method is more efficient than the Chan-Vese method.

<table>
<thead>
<tr>
<th>Image</th>
<th>CVM CPU (s)</th>
<th>Steps</th>
<th>Algorithm 1 CPU (s)</th>
<th>Steps</th>
<th>Algorithm 7 CPU (s)</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test01</td>
<td>482.3674</td>
<td>7</td>
<td>130.0337</td>
<td>5</td>
<td>0.0554</td>
<td>4</td>
</tr>
<tr>
<td>Test02</td>
<td>277.7851</td>
<td>6</td>
<td>126.7503</td>
<td>6</td>
<td>0.0540</td>
<td>4</td>
</tr>
<tr>
<td>CT</td>
<td>437.8700</td>
<td>8</td>
<td>1.9336</td>
<td>36</td>
<td>0.0406</td>
<td>3</td>
</tr>
<tr>
<td>Stone</td>
<td>6.7793</td>
<td>81</td>
<td>0.0798</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 10: Segmentations for the noise Stone image. (a) The original image. (b) Smoothed Stone image with the numerical scheme (23) ($\lambda = 0.1$). (c) Algorithm 7. (d) Algorithm 1. (e)–(f) The segmentation counter of Algorithms 7 and 1, respectively.
Compared with the previous simultaneous segmentation methods [4, 5, 10], the proposed method is more simple, efficient, and flexible. First, we separate the segmentation processes into smoothing the original image and segmenting the smooth image into two regions, and in the second step, the mass of the computation process is adding operators and logical operators. Therefore, the CPU time of the second step is only a little (Tables 1 and 2). Second, the first step is just to get some smooth versions of the original image, so in some sense the second step is independent of the result of the first step, and the second step must have a result (of course, it is not always good).

While we have not pursued it in this paper, one of the potential advantages of our method is that we can also use it
Figure 12: Some classical segmentation methods for the smooth images. (a)–(c) The segmentations based on the watershed Algorithm. (d)–(f) The segmentations from the Sobel operator. (g)–(i) The segmentations from the Canny filter.

Table 2: The CPU time in seconds and iterative step.

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>Algorithm 1</th>
<th>Algorithm 5</th>
<th>Algorithm 1</th>
<th>Algorithm 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU (s)</td>
<td>Steps</td>
<td>CPU (s)</td>
<td>Steps</td>
</tr>
<tr>
<td>0.4154</td>
<td>0.0202</td>
<td>4</td>
<td>0.0323</td>
<td>3</td>
</tr>
<tr>
<td>0.1</td>
<td>1.7060</td>
<td>20</td>
<td>1.7703</td>
<td>26</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0640</td>
<td>10</td>
<td>0.0640</td>
<td>10</td>
</tr>
</tbody>
</table>

Acknowledgments

This work is supported by the Heilongjiang Postdoctoral Fund (Grant No. LBH-Z12102), the Fundamental Research Funds for the Central Universities (Grant No. HIT.HSS.201201), the National Natural Science Foundation of China (Grant No. 71173060, 70773028, and 71031003), and proposed by Chan et al. [8], and other processes are similar to this paper.
the Natural Science Foundation of Heilongjiang Province of China (Grant No. G201006).

References


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