Journal of Computational Physics 350 (2017) 657–667

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An efficient iterative thresholding method for image segmentation

Dong Wang, Haohan Li, Xiaoyu Wei, Xiao-Ping Wang *

Department of Mathematics, The Hong Kong University of Science and Technology, Hong Kong

ARTICLE INFO

Article history:
Received 19 February 2017
Received in revised form 18 May 2017
Accepted 9 August 2017
Available online 8 September 2017

Keywords:
Iterative thresholding
Image segmentation
Piecewise constant Mumford–Shah functional
Convolution
Fast Fourier transform

ABSTRACT

We proposed an efficient iterative thresholding method for multi-phase image segmentation. The algorithm is based on minimizing piecewise constant Mumford–Shah functional in which the contour length (or perimeter) is approximated by a non-local multi-phase energy. The minimization problem is solved by an iterative method. Each iteration consists of computing simple convolutions followed by a thresholding step. The algorithm is easy to implement and has the optimal complexity $O(N \log N)$ per iteration. We also show that the iterative algorithm has the total energy decaying property. We present some numerical results to show the efficiency of our method.

1. Introduction

Image segmentation is one of the fundamental tasks in image processing. In broad terms, it is the process of partitioning a digital image into many segments according to a characterization of the image. The motivation behind this is to auto-determine which part of an image is meaningful for analysis, which also makes it one of the fundamental problems in computer vision. Many practical applications require image segmentation, like content-based image retrieval, machine vision, medical imaging, object detection and traffic control systems [17].

Variational methods have enjoyed tremendous success in image segmentation. A typical variational method for image segmentation starts by choosing an energy functional over the space of all legal segmentations, minimizing which gives a segmentation with desired properties. For instance, the celebrated Mumford–Shah model [18] uses the following formulation of energy:

$$E_{MS}(u,\Gamma) = \lambda \int_D (u-f)^2 dx + \mu \int_{D\setminus\Gamma} |\nabla u|^2 dx + Length(\Gamma), \tag{1}$$

where Γ is a closed subset of D given by the union of a finite number of curves representing the set of edges (i.e. boundaries of homogeneous regions) in the image f, u is a piecewise smooth approximation to f, and μ, λ are positive constants. Despite its descriptiveness, non-convexity of (1) makes the minimization problem difficult to analyze and solve numerically [1].

* Corresponding author.
E-mail addresses: dwangaf@connect.ust.hk (D. Wang), hlibb@connect.ust.hk (H. Li), xweiaf@connect.ust.hk (X. Wei), mawang@ust.hk (X.-P. Wang).

http://dx.doi.org/10.1016/j.jcp.2017.08.020
0021-9991/

To address this issue, a useful simplification of (1) is to restrict the minimization to functions (i.e. segmentations) that take a finite number of values. The resulting model is commonly referred to as the piecewise constant Mumford–Shah model. In particular, there is the following two-phase Chan–Vese functional [7,20]:

$$E_{CV}(\Sigma, C_1, C_2) = \lambda Per(\Sigma; D) + \int_{\Sigma} (C_1 - f)^2 dx + \int_{D\setminus\Sigma} (C_2 - f)^2 dx \tag{2}$$

where Σ is the interior of a closed curve and $Per(.)$ denotes the perimeter. C_1 and C_2 are averages of f within Σ and $D \setminus \Sigma$ respectively:

$$C_1 = \frac{\int_{\Sigma} f dx}{\int_{\Sigma} dx} \quad \text{and} \quad C_2 = \frac{\int_{D\setminus\Sigma} f dx}{\int_{D\setminus\Sigma} dx}$$

The level set method was used here to solve the minimization problem: Let $\phi(x) : D \to R$ be a Lipschitz continuous function with $\Sigma = \{x \in D : \phi(x) > 0\}$ and $D \setminus \Sigma = \{x \in D : \phi(x) < 0\}$. We can rewrite (2) as

$$E_{CV}(\phi, C_1, C_2) = \int_{D} \{\lambda|\nabla H(\phi)| + H(\phi)(C_1 - f)^2 + (1 - H(\phi))(C_2 - f)^2\} dx \tag{3}$$

where $H(\cdot) : R \to R$ is the Heaviside function

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

In practice, a regularized version of H denoted by H_ε is used. Then the Euler–Lagrange equation of (3) with respect to ϕ is given by

$$\frac{\partial \phi}{\partial t} = -H'_\varepsilon(\phi)\{-\{(C_1 - f)^2 - (C_2 - f)^2\} + \lambda \nabla \cdot (\frac{\nabla \phi}{|\nabla \phi|})\} \tag{4}$$

Equation (4) is nonlinear and requires regularization when $|\nabla \phi| = 0$.

Over the years, various modifications [1,4,13,19,20] are used in order to solve the equations more efficiently. For example, in [2], the authors instead of solving the optimal problem directly. They solved a dual formulation of the continuous Potts model based on its convex relaxation. In [5], a two-stage segmentation method is proposed. In the first stage, the authors apply the split Bregman method [12] to find the minimizer of a convex variant of the Mumford–Shah functional. In the second stage, a K-means clustering algorithm is used to choose $k - 1$ thresholds automatically to segment the image into k segments. One of the advantages of this method is that there is no need to specify the number of segments before finding the minimizer. Any k-phase segmentation can be obtained by choosing $k - 1$ thresholds after the minimizer is found. In [8], a frame-based model was introduced in which the perimeter term was approximated by a term involving framelets. The framelets were used to capture key features of biological structures. The model can also be fast implemented using split Bregman method [12].

Chan et al. [6] considered a convex reformulation to part of the Chan–Vese model. Given fixed values of C_1 and C_2, a global minimizer can be found. It is then demonstrated in [23] that this convex variant can be regarded as a continuous min-cut (primal) problem, and a corresponding continuous max-flow problem can be formulated as its dual. Efficient algorithms are developed by taking advantage of the strong duality between the primal and the dual problem, using the augmented Lagrangian method or the primal–dual method (see [21,23] and references therein).

Esedoglu et al. [10] proposed a phase-field approximation of (2) in which the Ginzburg–Landau functional is used to approximate the perimeter:

$$\begin{aligned} &E^{\varepsilon}_{MS}(u, C_1, C_2) \\ &= \int_{D} \left\{ \lambda \left(\varepsilon |\nabla u|^2 + \frac{1}{\varepsilon} W(u) \right) + u^2 (C_1 - f)^2 + (1 - u)^2 (C_2 - f)^2 \right\} dx \end{aligned} \tag{5}$$

where $\varepsilon > 0$ is the approximate interface thickness and $W(\cdot)$ is a double-well potential. Variation of (5) with respect to u yields the following gradient descent equation:

$$u_t = \lambda \left(2\epsilon \Delta u - \frac{1}{\epsilon} W'(u) \right) - 2\{u(C_1 - f)^2 + (u - 1)(C_2 - f)^2\}$$

which can be solved efficiently by an MBO [16] based threshold dynamic method that works by alternating the solution of a linear (but non-constant coefficient) diffusion equation with thresholding.

Recently, in a series of papers [3,11,14,15], Bertozzi et al. introduced a binary and multi-class version of the Ginzburg–Landau energy functional on graphs using graph Laplacian, and derived MBO based threshold dynamics methods for the semi-supervised learning problems on high dimensional data classification, using a similar approach as in [10].

The idea of approximating the perimeter of a set by a non-local energy (using heat kernel) is introduced in [9] to design an efficient threshold dynamics method for multi-phase problems with arbitrary surface tensions. The method is also generalized to wetting on rough surfaces in [22]. In this paper, we propose an efficient iterative thresholding method for minimizing the piecewise constant Mumford–Shah functional based on the relaxation and linearization procedure introduced in [9]. The perimeter term in (2) is approximated by a non-local multi-phase energy constructed based on convolution of the heat kernel with the characteristic functions of regions. An iterative algorithm is then derived to minimize the approximate energy. The procedure works by alternating the convolution step with the thresholding step. The convolution can be implemented efficiently on a uniform mesh using the fast Fourier transform (FFT) with the optimal complexity of $O(N \log N)$ per iteration. We also show that the algorithm has the total energy decaying property.

The rest of the paper proceeds as follows. In Section 2, we first give the approximate piecewise constant Mumford–Shah functional. We then derive the iterative thresholding scheme based on the linearization of the approximate functional. The monotone decrease of energy at each iteration is proved (with details given in the appendix). In Section 3, we present some numerical examples to show the efficiency of the method.

2. An efficient iterative thresholding method for image segmentation

In this section, we introduce an iterative thresholding method for multi-phase image segmentation based on the Chan–Vese model [20]. The perimeter terms will be approximated by a non-local multi-phase energy constructed based on convolution of the heat kernel with the characteristic functions of regions. The iterative algorithm is then derived as an optimization procedure for the approximate energy.

2.1. The approximate Chan–Vese functional

Let Ω denote the domain of an input image f given by a d-dimensional vector. Our task is to find an n-phase partition $\{\Omega_i\}_{i=1}^n$ of Ω and $C = (C_1, C_2, \cdots, C_n) \in \mathbb{R}^n$ which minimize

$$\mathcal{E}(\{\Omega_i\}_{i=1}^n, C) = \sum_{i=1}^{n} \left[\int_{\Omega_i} g_i d\Omega_i + \lambda |\partial \Omega_i| \right] \tag{6}$$

where Ω_i represents the region of the ith phase; $|\partial\Omega_i|$ is the length of a boundary curve of the region Ω_i; $g_i = ||C_i - f||_2^2$ ($||.||_2$ denotes the l^2 vector norm).

Let $u = (u_1(x), u_2(x), \cdots, u_n(x))$ where $\{u_i(x)\}_{i=1}^n$ are the characteristic functions of the regions $\{\Omega_i\}_{i=1}^n$. We then look for u^* and C^* such that

$$(u^*, C^*) = \underset{u \in \mathcal{S}, C \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{i=1}^{n} \left[\int_{\Omega} u_i(x) g_i(x) d\Omega + \lambda |\partial \Omega_i| \right], \tag{7}$$

where $\mathcal{S} = \left\{ u = (u_1, u_2, \cdots, u_n) \in BV(\Omega) : u_i(x) = 0, 1, \text{and} \sum_{i=1}^{n} u_i = 1 \right\}$. As pointed out in [9], when $\delta t \ll 1$, the length of $\partial\Omega_i \cap \partial\Omega_j$ can be approximated by

$$|\partial \Omega_i \cap \partial \Omega_j| \approx \sqrt{\frac{\pi}{\delta t}} \int_{\Omega} u_i G_{\delta t} * u_j d\Omega, \tag{8}$$

where $*$ represents convolution and

$$G_{\delta t}(x) = \frac{1}{4\pi \delta t} \exp\left(-\frac{|x|^2}{4\delta t}\right)$$

is the heat kernel.

The above integral measures the amount of heat that escapes from Ω_j to Ω_i. That can estimate the size of the boundary between Ω_i and Ω_j after normalization. Therefore,

$$|\partial \Omega_i| \approx \sum_{j=1, j \neq i}^{n} \sqrt{\frac{\pi}{\delta t}} \int_{\Omega} u_i G_{\delta t} * u_j d\Omega. \tag{9}$$

Hence the total energy can be approximated by

$$\mathcal{E}^{\delta t}(u,C)=\sum_{i=1}^{n}\int_{\Omega}\left(u_i g_i+\lambda\sum_{j=1,j\neq i}^{n}\frac{\sqrt{\pi}}{\sqrt{\delta t}}u_i G_{\delta t}*u_j\right)d\Omega. \tag{10}$$

Now, the solution of (7) can be approximated by finding (u^*,C^*) such that

$$(u^*,C^*)=\underset{u\in\mathcal{S},C\in\mathbb{R}^n}{\operatorname{argmin}}\ \mathcal{E}^{\delta t}(u,C). \tag{11}$$

2.2. Derivation of the iterative thresholding method

In this section, we present the derivation of an iterative thresholding method for the minimization problem (11). The derivation is based on the relaxation and linearization procedure introduced in [9].

We now use an iterative scheme to solve (11) as follows. Given an initial guess $u^0=(u_1^0,u_2^0,\cdots,u_n^0)$ and $C^0=(C_1^0,C_2^0,\cdots,C_n^0)$, we compute a series of minimizers

$$u^1,C^1,u^2,C^2,\cdots,u^{k+1},C^{k+1},\cdots$$

such that

$$u^{k+1}=\underset{u\in\mathcal{S}}{\operatorname{argmin}}\,\mathcal{E}^{\delta t}(u,C^k), \tag{12}$$

$$C^{k+1}=\underset{C\in\mathbb{R}^n}{\operatorname{argmin}}\,\mathcal{E}^{\delta t}(u^{k+1},C) \tag{13}$$

for $k=0,1,2,\cdots$.

Now, the original problem (11) is split into solving two optimization problems (12) and (13) alternatively. For the problem (13), when u^{k+1} is computed through (12), it is easy to show that the variation of (13) with respect to C_i gives the optimal choice of C_i^{k+1} as

$$C_i^{k+1}=\frac{\int_\Omega u_i^{k+1}f\,d\Omega}{\int_\Omega u_i^{k+1}d\Omega} \tag{14}$$

for $i=1,2,\cdots,n$.

The problem (12) is to minimize a non-convex energy functional defined on a non-convex admissible set. However, we can relax (12) to a problem defined on a convex admissible set by finding u^{k+1} such that

$$u^{k+1}=\underset{u\in\mathcal{K}}{\operatorname{argmin}}\,\mathcal{E}^{\delta t}(u,C^k) \tag{15}$$

where $\mathcal{K}$ is the convex hull of $\mathcal{S}$:

$$\mathcal{K}=\left\{u=(u_1,u_2,\cdots,u_n)\in BV(\Omega):0\le u_i(x)\le 1,\text{ and }\sum_{i=1}^{n}u_i=1\right\}. \tag{16}$$

The following lemma shows that the relaxed problem (15) is equivalent to the original problem (12). Therefore we can solve the relaxed problem (15) instead.

Lemma 2.1. *Let $u=(u_1,u_2,\cdots,u_n)$ and denote $C=(C_1,C_2,\cdots,C_n)$ as a constant vector. Then*

$$\underset{u\in\mathcal{S}}{\operatorname{argmin}}\,\mathcal{E}^{\delta t}(u,C)=\underset{u\in\mathcal{K}}{\operatorname{argmin}}\,\mathcal{E}^{\delta t}(u,C). \tag{17}$$

Proof. See Appendix A. □

In the following, we show that the minimization problem (15) can be solved by a simple thresholding method. Suppose that we have the kth iteration $(u_1^k,u_2^k,\cdots,u_n^k)\subset\mathcal{S}$ and thus we have $g_i^k=||C_i^k-f||_2^2$ with

$$C_i^k=\frac{\int_\Omega u_i^k f\,d\Omega}{\int_\Omega u_i^k d\Omega}$$

for $i=1,2,\cdots,n$. Then the energy functional $\mathcal{E}^{\delta t}(u,C^k)$ can be linearized near the point $u^k=(u_1^k,u_2^k,\cdots,u_n^k)$ by

$$\begin{aligned}\mathcal{E}^{\delta t}(u,C^k)&\approx\mathcal{E}^{\delta t}(u^k,C^k)\\&+\mathcal{L}(u_1-u_1^k,u_2-u_2^k,\cdots,u_n-u_n^k,u_1^k,u_2^k,\cdots,u_n^k)+h.o.t.\end{aligned} \tag{18}$$

where

$$\mathcal{L}(u_1, u_2, \cdots, u_n, u_1^k, u_2^k, \cdots, u_n^k) = \sum_{i=1}^{n} \int_\Omega \left(u_i g_i^k + \sum_{j=1, j\neq i}^{n} \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} u_i G_{\delta t} * u_j^k \right) d\Omega$$
$$= \sum_{i=1}^{n} \int_\Omega u_i \left(g_i^k + \sum_{j=1, j\neq i}^{n} \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} G_{\delta t} * u_j^k \right) d\Omega. \tag{19}$$

We can now determine the next iteration $(u_1^{k+1}, u_2^{k+1}, \cdots, u_n^{k+1})$ by minimizing the linearized functional

$$\min_{(u_1, u_2, \cdots, u_n)\in\mathcal{K}} \mathcal{L}(u_1, u_2, \cdots, u_n, u_1^k, u_2^k, \cdots, u_n^k). \tag{20}$$

Denote

$$\phi_i^k := g_i^k + \sum_{j=1, j\neq i}^{n} \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} G_{\delta t} * u_j^k \tag{21}$$
$$= g_i^k + \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} (1 - G_{\delta t} * u_i^k). \tag{22}$$

We have

$$\mathcal{L}(u_1, u_2, \cdots, u_n, u_1^k, u_2^k, \cdots, u_n^k) = \sum_{i=1}^{n} \int_\Omega u_i \phi_i^k d\Omega = \int_\Omega \sum_{i=1}^{n} u_i \phi_i^k d\Omega. \tag{23}$$

The optimization problem (20) becomes minimizing a linear functional over a convex set. It can be carried out at each $x \in \Omega$ independently. By comparing the coefficients $\phi_i^k(x)$ (non-negative) of $u_i(x)$ in the integrand of (23), it is easy to see that the minimum is attained at

$$u_i^{k+1}(x) = \begin{cases} 1 & \text{if } \phi_i^k(x) = \min_l \phi_l^k(x), \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

The following theorem shows that the total energy $\mathcal{E}^{\delta t}$ decreases in the iteration for any $\delta t > 0$. Therefore, our iteration algorithm always converges to a minimum for any initial partition.

Theorem 2.1. *Let $(u_1^{k+1}, u_2^{k+1}, \cdots, u_n^{k+1})$ and $(C_1^{k+1}, C_2^{k+1}, \cdots, C_n^{k+1})$ be the $k+1$th iteration derived above, we have*

$$\mathcal{E}^{\delta t}(u^{k+1}, C^{k+1}) \leq \mathcal{E}^{\delta t}(u^k, C^k) \tag{25}$$

for all $\delta t > 0$.

Proof. See Appendix B. □

We are then led to the following iterative thresholding algorithm:

Algorithm: I

Step 0. Given an initial partition $\Omega_1^0, ..., \Omega_n^0 \subset \Omega$ and the corresponding $u_1^0 = \chi_{\Omega_1^0}, ..., u_n^0 = \chi_{\Omega_n^0}$. Set a tolerance parameter $\tau > 0$.

Step 1. Given kth iteration $(u_1^k, \cdots, u_n^k) \subset \mathcal{S}$, we compute g_i^k and the following convolutions for $i = 1, \cdots, n$:

$$\phi_i^k := g_i^k + \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} (1 - G_{\delta t} * u_i^k) \tag{26}$$

Step 2. Thresholding: Let

$$\Omega_i^{k+1} = \left\{ x : \phi_i^k(x) < \min_{j\neq i} \phi_j^k(x) \right\} \tag{27}$$

and define $u_i^{k+1} = \chi_{\Omega_i^{k+1}}$ where $\chi_{\Omega_i^{k+1}}$ represents the characteristic function of region Ω_i^{k+1}.

(a) Given Image. (b) Initial Contour. (c) Final Contour.

Fig. 1. Segmentation results for the classic cameraman image with $\delta t = 0.03$ and $\lambda = 0.01$. The algorithm converges in 15 iterations with a computational time of 0.1188 seconds.

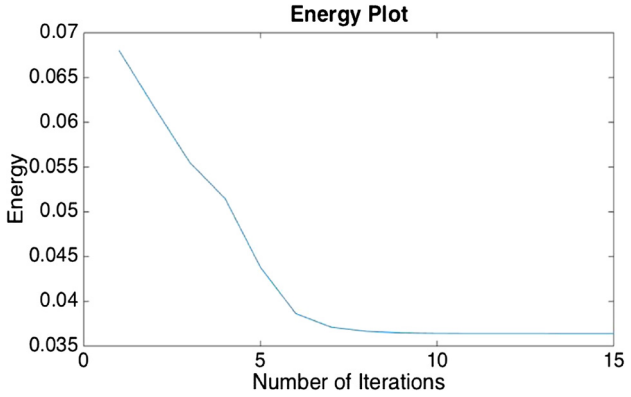


Fig. 2. Energy curve for the iteration algorithm with $\delta t = 0.03$ and $\lambda = 0.01$.

Step 3. Let the normalized L^2 difference between successive iterations be

$$e^{k+1} = \frac{1}{|\Omega|} \int_{\Omega} \sum_{i=1}^{n} |u_i^{k+1} - u_i^k|^2 d\Omega.$$

If $e^{k+1} \leq \tau$, stop. Otherwise, go back to step 1.

Remark 2.1. The convolutions in Step 1 are computed efficiently using FFT with a computational complexity of $O(N \log(N))$, where N is the total number of pixels. Therefore the total computational cost at each iteration is also $O(N \log(N))$.

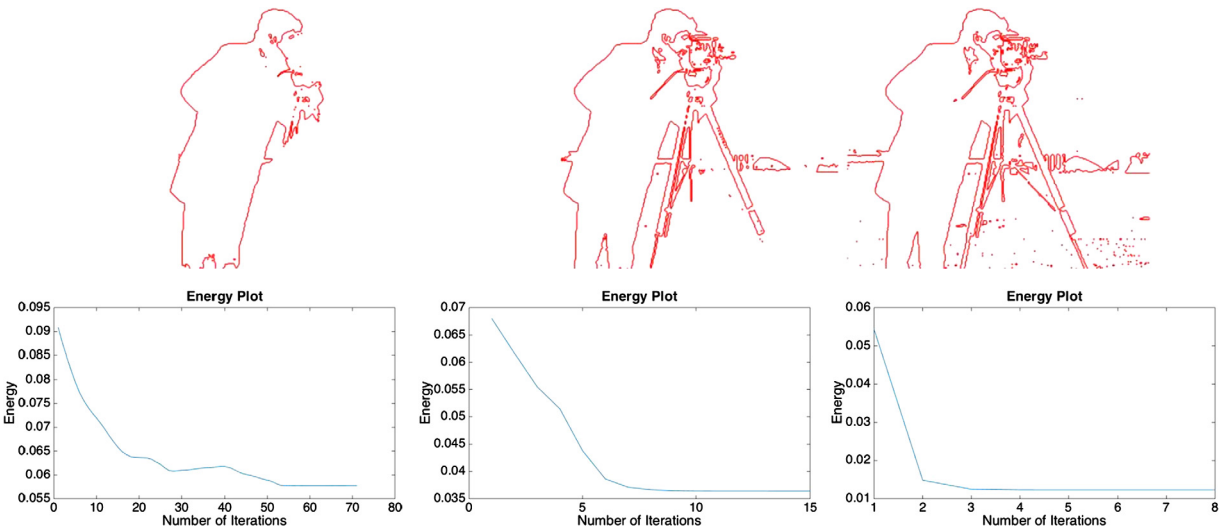
Remark 2.2. In Step 3, e^k measures the percentage of pixels on which $u_i^{k+1} \neq u_i^k$. Therefore the tolerance τ specifies the threshold of the percentage of pixels changing during the iteration below which the iteration stops.

3. Numerical results

We now present numerical examples to illustrate the performance of our algorithm. We implement the algorithm in MATLAB. All the computations are carried out on a MacBook Pro laptop with a 3.0 GHz Intel(R) Core(TM) i7 processor and 8 GB of RAM.

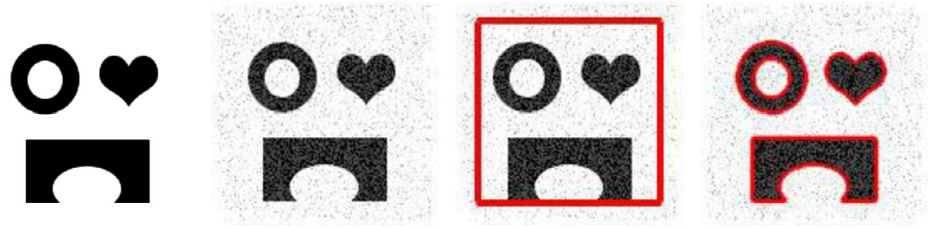
3.1. Example 1: cameraman

We first test our algorithm on the standard cameraman image using two-phase segmentation. Fig. 1(a) is the original image. We start with the initial contour given in Fig. 1(b). We choose $\delta t = 0.03$ and $\lambda = 0.01$. Our algorithm takes only 15 iterations to converge to a complete steady state, i.e. $e^k = 0$ (for $k = 15$) with a total computation time of only 0.1188 seconds. Fig. 1(c) gives the final segmentation contour. We also plot the normalized energy $\mathcal{E}^{\delta t}/|\Omega|$ as a function of the iteration number k in Fig. 2, which verifies the monotone decay of the energy. In fact, the energy decays quickly in the first few iterations and almost reaches steady state in less than 10 iterations.



(d) $\lambda = 0.025$. (e) $\lambda = 0.01$. (f) $\lambda = 0.001$.

Fig. 3. Segmentation contours and energy curves for $\delta t = 0.03$ and different λ values.



(a) Given Image. (b) Noised Image. (c) Initial Contour. (d) Final Contour.

Fig. 4. Noisy image segmentation with $\delta t = 0.03$, $\lambda = 0.1$.

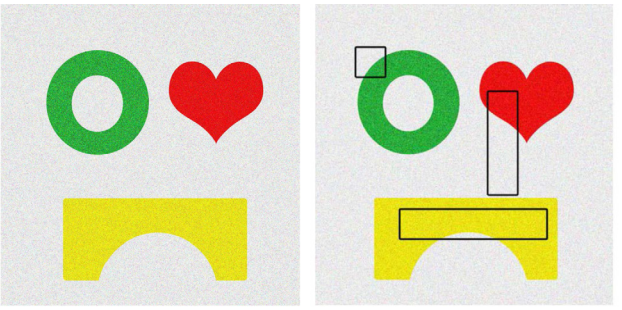
To study the effect of the parameter λ in the energy (10), we run our algorithm on the same test image for three different values of $\lambda = 0.001, 0.01$ and 0.025 but with a fixed $\delta t = 0.03$. The final segmentation contours together with the energy curves are shown in Fig. 3. As the figure shows, larger $\lambda = 0.025$ turns to smooth out the small-scale structures while smaller $\lambda = 0.001$ would pick up more noisy regions. This is easy to understand since λ measures the relative importance of the contour length and the data term in the Chan–Vese functional to be minimized. A larger λ tends to shorten the total contour length and therefore does not favor small-scale structures. On the other hand, convergence is much faster for a smaller λ while a larger λ would require more iterations to converge as shown by the energy curves.

3.2. Example 2: image with heavy noise

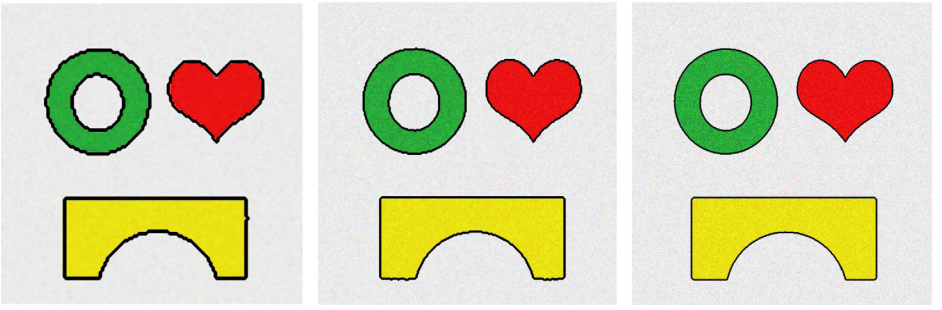
Now, we apply our algorithm to a heavy noised image (Fig. 4(a)). The original image was a clear synthetic one. Gaussian noise with mean 0.6 and variance 0.5 was added to the image to give Fig. 4(b). The initial contours are given in Fig. 4(c). We apply our two-phase algorithm to the image with $\delta t = 0.03$ and $\lambda = 0.1$. The algorithm converges in 11 iterations with runtimes of 0.019 seconds. Fig. 4(d) shows the final segmentation result.

3.3. Example 3: a synthetic four-phase image

We next use a synthetic color image given in Fig. 5(a). The image f is a vector-valued function. Gaussian noise is added with mean 0 and variance 0.04 to each component of image f. The initial contours are given in Fig. 5(b). We apply our four-phase algorithm to the image with three different resolutions from 128×128 to 512×512. In each case, $\delta t = 0.01$ and $\lambda = 0.003$. The algorithm converges in $7 \sim 8$ iterations for all resolutions with runtimes of $0.0444, 0.1333, 0.6706$ seconds respectively, which demonstrates good stability of and robustness of our method. Figs. 5(c)–5(e) show the final segmentation result.



(a) Image with Noise. (b) Initial Contour.



(c) 128×128. (d) 256×256. (e) 512×512.

Fig. 5. Segmentation for images with different resolutions and with the parameters $\delta t = 0.01$ and $\lambda = 0.003$.

(a) Given Color Image. (b) Initial Contour. (c) Final Contour.

Fig. 6. Two-phase segmentation for a 375×500 RGB image and with parameters $\delta t = 0.01$ and $\lambda = 0.005$.

(a) Initial Contour. (b) Final Contour. (c) Four Segments.

Fig. 7. Four phase segmentation for a 375×500 RGB image with $\delta t = 0.01$ and $\lambda = 0.003$.

3.4. Example 4: flower color image

We now consider an image containing flowers of different colors in Fig. 6(a). We first use a two-phase segmentation algorithm with $\delta t = 0.01$ and $\lambda = 0.005$ and the initial contour in Fig. 6(b). The algorithm converges in 20 iterations with a runtime of 0.6751 seconds. The final segmentation result is given in Fig. 6(c). We also use a four-phase segmentation algorithm with $\delta t = 0.01$ and $\lambda = 0.003$ and the initial contour in Fig. 7(a). The algorithm converges in 18 iterations with a runtime of 1.1007 seconds. The final segmentation result is given in Fig. 7(b) and 7(c).

4. Conclusions

We have proposed an efficient iterative thresholding algorithm for the Chan–Vese model for multi-phase image segmentation. The algorithm works by alternating the convolution step with the thresholding step and has the optimal computational complexity of $O(N \log N)$ per iteration. We prove that the iterative algorithm has the property of total energy decay. The numerical results show that the method is stable and the number of iterations before convergence is independent of the spacial resolution (for a given image). The relative importance of the different effects in the energy functional is studied by tuning the parameter λ.

Acknowledgements

We thank Prof. Tony Chan, Zuowei Shen, Xuecheng Tai and Xiaoqun Zhang for helpful discussions and suggestions. This research was supported in part by the Hong Kong Research Grants Council (GRF grants 605513, 16302715 and 16324416, CRF grant C6004-14G, and NSFC-RGC joint research grant N-HKUST620/15).

Appendix A. Proof of Lemma 2.1

Let $v = (v_1, v_2, \cdots, v_n) \in \mathcal{K}$ be a minimizer of $\mathcal{E}^{\delta t}(u, C)$ on $\mathcal{K}$. Since $\mathcal{S} \subset \mathcal{K}$, we have

$$\begin{aligned}\mathcal{E}^{\delta t}(v, C) &= \min_{u\in\mathcal{K}} \mathcal{E}^{\delta t}(u, C)\\ &\le \min_{u\in\mathcal{S}} \mathcal{E}^{\delta t}(u, C).\end{aligned}$$

Therefore, we only need to prove that $v \in \mathcal{S}$.

We prove it by contradiction. If $v \notin \mathcal{S}$, then there exists a set $A \subseteq \Omega$ ($|A| > 0$) and a constant $0 < \epsilon < \frac{1}{2}$ such that for some $k, l \in \{1, \cdots, n\}$ with $k \neq l$,

$$v_k(x), v_l(x) \in (\epsilon, 1-\epsilon), \quad \forall x \in A.$$

Denote

$$u_m^t(x, t) = v_m(x) + t(\delta_{m,l} - \delta_{m,k})\chi_A(x)$$

for $m = 1, 2, \cdots, n$ where $\chi_A(x)$ represents the characteristic function of region A and

$$\delta_{m,l} = \begin{cases} 1 & m = l \\ 0 & m \neq l. \end{cases}$$

When $-\epsilon \le t \le \epsilon$, we have $u_m^t(x, t) \ge 0$ and $\sum_{m=1}^{n} u_m^t(x, t) = 1$ so that $u^t(x, t) = (u_1^t(x, t), \cdots, u_n^t(x, t)) \in \mathcal{K}$ and we have

$$\frac{du_m^t}{dt} = (\delta_{m,l} - \delta_{m,k})\chi_A(x), \qquad \frac{d^2u_m^t}{dt^2} = 0. \tag{A.1}$$

A direct computation gives

$$\begin{aligned}\frac{d^2\mathcal{E}^{\delta t}}{dt^2} &= \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} \sum_{m,n,m\neq n} \int_\Omega \frac{du_m^t}{dt}\left(G_{\delta t} * \frac{du_n^t}{dt}\right) d\Omega\\ &= \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} \sum_{m,n,m\neq n} (\delta_{m,l} - \delta_{m,k})(\delta_{n,l} - \delta_{n,k}) \int_\Omega \chi_A G_{\delta t} * \chi_A d\Omega \qquad (A.2)\\ &= -\frac{4\lambda\sqrt{\pi}}{\sqrt{\delta t}} \int_\Omega \chi_A G_{\delta t} * \chi_A d\Omega\\ &< 0.\end{aligned}$$

Thus, $v(x) = u(x, 0)$ cannot be a minimizer. This contradicts the assumption.

Appendix B. Proof of Theorem 2.1

From (19), we have

$$\begin{aligned}
&\mathcal{E}^{\delta t}(u^k, C^k) + \sum_{i=1}^{n}\int_{\Omega}\sum_{j\neq i,\, j=1}^{n}\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}}u_i^k G_{\delta t} * u_j^k d\Omega = \mathcal{L}(u_1^k, \cdots, u_n^k, u_1^k, \cdots, u_n^k)\\
&\geq \mathcal{L}(u_1^{k+1}, \cdots, u_n^{k+1}, u_1^k, \cdots, u_n^k) = \mathcal{E}^{\delta t}(u^{k+1}, C^{k+1})\\
&+\sum_{i=1}^{n}\int_{\Omega}\left(u_i^{k+1}(g_i^k - g_i^{k+1}) + \sum_{j=1,\, j\neq i}^{n}\frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}}u_i^{k+1}G_{\delta t} * u_j^k\right)d\Omega\\
&-\sum_{i=1}^{n}\int_{\Omega}\sum_{j\neq i,\, j=1}^{n}\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}}u_i^{k+1}G_{\delta t} * u_j^{k+1}d\Omega.
\end{aligned}$$

That leads to

$$\mathcal{E}^{\delta t}(u^k, C^k) \geq \mathcal{E}^{\delta t}(u^{k+1}, C^{k+1}) + I \tag{B.1}$$

with

$$\begin{aligned}
I =& \sum_{i=1}^{n}\int_{\Omega}\left(u_i^{k+1}(g_i^k - g_i^{k+1}) + \sum_{j=1,\, j\neq i}^{n}\frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}}u_i^{k+1}G_{\delta t} * u_j^k\right)d\Omega\\
&-\sum_{i=1}^{n}\int_{\Omega}\sum_{j\neq i,\, j=1}^{n}\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}}u_i^{k+1}G_{\delta t} * u_j^{k+1}d\Omega\\
&-\sum_{i=1}^{n}\int_{\Omega}\sum_{j\neq i,\, j=1}^{n}\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}}u_i^{k}G_{\delta t} * u_j^{k}d\Omega\\
=& I_1 + I_2
\end{aligned}$$

where

$$\begin{aligned}
I_1 =& \sum_{i=1}^{n}\int_{\Omega}u_i^{k+1}(g_i^k - g_i^{k+1})d\Omega\\
I_2 =& \sum_{i=1}^{n}\sum_{j=1,\, j\neq i}^{n}\int_{\Omega}\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}}u_i^{k+1}G_{\delta t} * (u_j^k - u_j^{k+1})d\Omega\\
&-\sum_{i=1}^{n}\sum_{j=1,\, j\neq i}^{n}\int_{\Omega}\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}}(u_i^k - u_i^{k+1})G_{\delta t} * u_j^k d\Omega.
\end{aligned}$$

Now, we only need to prove that $I_1 \geq 0$ and $I_2 \geq 0$. From the definition of C_i^{k+1} and using the fact that $\int_\Omega u_i^{k+1} f d\Omega = \int_\Omega u_i^{k+1} d\Omega C_i^{k+1}$, we have

$$\begin{aligned}
I_1 &= \sum_{i=1}^{n}\int_{\Omega}u_i^{k+1}(||C_i^k - f||_2^2 - ||C_i^{k+1} - f||_2^2)d\Omega\\
&= \sum_{i=1}^{n}\int_{\Omega}u_i^{k+1}(||C_i^k||_2^2 - ||C_i^{k+1}||_2^2 - 2\langle C_i^k - C_i^{k+1}, f\rangle)d\Omega\\
&= \sum_{i=1}^{n}\left\{\int_{\Omega}u_i^{k+1}d\Omega(||C_i^k||_2^2 - ||C_i^{k+1}||_2^2 - 2\langle C_i^k - C_i^{k+1}, C_i^{k+1}\rangle)\right\} \qquad \text{(B.2)}\\
&= \sum_{i=1}^{n}\left\{\int_{\Omega}u_i^{k+1}d\Omega||C_i^k - C_i^{k+1}||_2^2\right\} \geq 0.
\end{aligned}$$

By changing the order of the two summations in the second part of I_2 and using the fact that $\sum_{i=1}^{n} u_i^k = 1$ for any k, we obtain

$$\begin{aligned} I_2 &= \sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\int_{\Omega}\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}}(u_i^{k+1}-u_i^k)G_{\delta t} * (u_j^k - u_j^{k+1})d\Omega \\ &= \sum_{i=1}^{n}\int_{\Omega}\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}}(u_i^{k+1}-u_i^k)G_{\delta t} * \left(\sum_{j=1,j\neq i}^{n}(u_j^k - u_j^{k+1})\right)d\Omega \\ &= \sum_{i=1}^{n}\int_{\Omega}\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}}(u_i^{k+1}-u_i^k)G_{\delta t} * (1-u_i^k-(1-u_i^{k+1}))d\Omega \\ &= \sum_{i=1}^{n}\int_{\Omega}\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}}(u_i^{k+1}-u_i^k)G_{\delta t} * (u_i^{k+1}-u_i^k)d\Omega \geq 0. \end{aligned} \tag{B.3}$$

Combining (B.1), (B.2) and (B.3) gives (25).

References

[1] L. Ambrosio, V.M. Tortorelli, Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence, Commun. Pure Appl. Math. 43 (8) (1990) 999–1036.
[2] E. Bae, J. Yuan, X.-C. Tai, Global minimization for continuous multiphase partitioning problems using a dual approach, Int. J. Comput. Vis. 92 (1) (2011) 112–129.
[3] A.L. Bertozzi, A. Flenner, Diffuse interface models on graphs for classification of high dimensional data, Multiscale Model. Simul. 10 (3) (2012) 1090–1118.
[4] A. Braides, Approximation of Free-Discontinuity Problems, vol. 1694, Springer Science & Business Media, 1998.
[5] X. Cai, R. Chan, T. Zeng, A two-stage image segmentation method using a convex variant of the Mumford–Shah model and thresholding, SIAM J. Imaging Sci. 6 (1) (2013) 368–390.
[6] T.F. Chan, S. Esedoğlu, M. Nikolova, Algorithms for finding global minimizers of image segmentation and denoising models, SIAM J. Appl. Math. 66 (5) (2006) 1632–1648.
[7] T.F. Chan, L.A. Vese, Active contours without edges, IEEE Trans. Image Process. 10 (2) (2001) 266–277.
[8] B. Dong, A. Chien, Z. Shen, Frame based segmentation for medical images, Commun. Math. Sci. 9 (2) (2010) 551–559.
[9] S. Esedoğlu, F. Otto, Threshold dynamics for networks with arbitrary surface tensions, Commun. Pure Appl. Math. 68 (5) (2015) 808–864.
[10] S. Esedoğlu, Y.-H.R. Tsai, Threshold dynamics for the piecewise constant Mumford–Shah functional, J. Comput. Phys. 211 (1) (2006) 367–384.
[11] C. Garcia-Cardona, E. Merkurjev, A.L. Bertozzi, A. Flenner, A.G. Percus, Multiclass data segmentation using diffuse interface methods on graphs, IEEE Trans. Pattern Anal. Mach. Intell. 36 (8) (2014) 1600–1613.
[12] T. Goldstein, S. Osher, The split Bregman method for L1-regularized problems, SIAM J. Imaging Sci. 2 (2) (2009) 323–343.
[13] J. Liu, X. Tai, H. Huang, Z. Huan, A fast segmentation method based on constraint optimization and its applications: intensity inhomogeneity and texture segmentation, Pattern Recognit. 44 (9) (2011) 2093–2108.
[14] E. Merkurjev, A.L. Bertozzi, F. Chung, A Semi-Supervised Heat Kernel Pagerank MBO Algorithm for Data Classification, Technical Report, 2016.
[15] E. Merkurjev, J. Sunu, A.L. Bertozzi, Graph MBO method for multiclass segmentation of hyperspectral stand-off detection video, in: 2014 IEEE International Conference on Image Processing, ICIP, IEEE, 2014, pp. 689–693.
[16] B. Merriman, J.K. Bence, S. Osher, Diffusion Generated Motion by Mean Curvature, Department of Mathematics, University of California, Los Angeles, 1992.
[17] A. Mitiche, I.B. Ayed, Variational and Level Set Methods in Image Segmentation, vol. 5, Springer Science & Business Media, 2010.
[18] D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, Commun. Pure Appl. Math. 42 (5) (1989) 577–685.
[19] A. Tsai, A. Yezzi, A.S. Willsky, Curve evolution implementation of the Mumford–Shah functional for image segmentation, denoising, interpolation, and magnification, IEEE Trans. Image Process. 10 (8) (2001) 1169–1186.
[20] L.A. Vese, T.F. Chan, A multiphase level set framework for image segmentation using the Mumford and Shah model, Int. J. Comput. Vis. 50 (3) (2002) 271–293.
[21] K. Wei, X.-C. Tai, T.F. Chan, S. Leung, Primal–dual method for continuous max-flow approaches, in: Computational Vision and Medical Image Processing V: Proceedings of the 5th Eccomas Thematic Conference on Computational Vision and Medical Image Processing, VipIMAGE 2015, Tenerife, Spain, October 19–21, 2015, CRC Press, 2015, p. 17.
[22] X. Xu, D. Wang, X.-P. Wang, An efficient threshold dynamics method for wetting on rough surfaces, J. Comput. Phys. 330 (2017) 510–528.
[23] J. Yuan, E. Bae, X.-C. Tai, A study on continuous max-flow and min-cut approaches, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2010, pp. 2217–2224.