



The iterative convolution–thresholding method (ICTM) for image segmentation



Dong Wang^{a,b,*}, Xiao-Ping Wang^{a,b,c}

^a School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Guangdong 518172, China

^b Shenzhen International Center for Industrial and Applied Mathematics, Shenzhen Research Institute of Big Data, Guangdong, 518172, China

^c Department of Mathematics, Hong Kong University of Science and Technology, Hong Kong, China

ARTICLE INFO

Article history:

Received 14 April 2020

Revised 17 April 2022

Accepted 13 May 2022

Available online 14 May 2022

Keywords:

Convolution

Thresholding

Image segmentation

Heat kernel

ABSTRACT

Variational methods, which have been tremendously successful in image segmentation, work by minimizing a given objective functional. The objective functional usually consists of a fidelity term and a regularization term. Because objective functionals may vary from different types of images, developing an efficient, simple, and general numerical method to minimize them has become increasingly vital. However, many existing methods are model-based, converge relatively slowly, or involve complicated techniques. In this paper, we develop a novel iterative convolution–thresholding method (ICTM) that is simple, efficient, and applicable to a wide range of variational models for image segmentation. In ICTM, the interface between two different segment domains is implicitly represented by the characteristic functions of domains. The fidelity term is usually written into a linear functional of the characteristic functions, and the regularization term is approximated by a functional of characteristic functions in terms of heat kernel convolution. This allows us to design an iterative convolution–thresholding method to minimize the approximate energy. The method has the energy-decaying property, and thus the unconditional stability is theoretically guaranteed. Numerical experiments show that the method is simple, easy to implement, robust, and applicable to various image segmentation models.

© 2022 Elsevier Ltd. All rights reserved.

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. Many practical applications require image segmentation, such as content-based image retrieval, machine vision, medical imaging, object detection, and traffic control systems [1].

Many methods for image segmentation have been developed along directions including direct thresholding, k-means clustering, edge detection, region growing, compression-based methods, histogram-based methods, graph partitioning, partial-differential-equation-based methods, and variational methods (e.g., [2]). Among them, variational methods have been tremendously successful in image segmentation. In variational methods, one usually minimizes an objective functional in order to obtain some desired properties. The objective functional usually consists of a fidelity term and a regularization term. The fidelity term keeps

the consistency of the segmentation with the original image, and the regularization term aims to make the final contour more regularized. The first successful variational model for image segmentation was the Mumford–Shah (MS) model [3], which minimizes an objective functional over the space of piecewise smooth functions to obtain a segmentation with desired properties. Despite the descriptiveness of the energy functional, its non-convexity makes the minimization problem difficult to analyze and solve numerically [4]. A useful simplification is to restrict the minimization to functions (i.e., segmentations) that are piecewise constant functions. The resulting model is commonly referred to as the Chan–Vese (CV) model [5,6].

When images are intensity inhomogeneous, a local intensity fitting (LIF) model [7,8] was proposed to overcome the segmentation difficulty caused by intensity inhomogeneity. Wang et al. [9] proposed a local global intensity fitting (LGIF) model combining the advantages of the CV model and the LIF model by taking into account the local and global intensity information. Several locally statistical active contour (LSAC) models have also been proposed for image segmentation with intensity inhomogeneity. For example, Zhang et al. [10] proposed a statistical energy functional based on the distribution of each local region in the transformed domain,

* Corresponding author.

E-mail addresses: wangdong@cuhk.edu.cn (D. Wang), mawang@ust.hk (X.-P. Wang).

which combines the bias field, level set function, and piecewise constant function approximating the true image signal.

Over the years, various numerical methods have been developed to solve the above segmentation models (e.g., CV, LIF, LGIF, and LSAC). For example, instead of solving the optimization problem directly, Bae et al. [11] solved a dual formulation of the continuous Potts model based on its convex relaxation. Moreover, Cai et al. [12] proposed a two-stage segmentation method combining the split Bregman method and K-means clustering algorithm.

Among these, the level-set method has been widely used to successfully implement image segmentation models, and allows automatic detection of interior contours (e.g., [13,14]). It can be easily applied to general variational models on image segmentation. However, the level-set method involves tracking a level-set function, with its zero-level set representing the interface. To avoid numerical deterioration of the interface, the level-set function needs to be periodically reinitialized as a distance function from the interface. In addition, the method introduces an artificial time step, which must be relatively small for stability reasons, restricting the evolution of contours. Also, it is difficult to generalize the method to multi-segment cases with an arbitrary number of segments.

In this paper, we consider a general energy functional consisting of a fidelity term and a regularized term for the n -segment image segmentation problem:

$$\mathcal{E} = \sum_{i=1}^n \int_{\Omega_i} F_i(f, \Theta_1, \dots, \Theta_n) dx + \lambda \sum_{i=1}^n |\partial \Omega_i|, \quad (1)$$

where $f : \Omega \rightarrow [0, 1]^d$ is the image, d is the number of channels, $\Theta_i = (\Theta_{i,1}, \Theta_{i,2}, \dots, \Theta_{i,m})$ contains all of the possible variables or functions in fidelity terms, λ is a parameter, Ω_i denotes the domain of each segment, and $|\partial \Omega_i|$ denotes the perimeter of the boundary $\partial \Omega_i$. F_i can be general and can include many existing models as special cases (e.g., CV, LIF, LGIF, and LSAC models, and many others).

Motivated by the threshold dynamics method for simulating multiphase flow [15], we propose to use a characteristic function to denote the region of each segment. The edge of each segment can then be implicitly represented by the corresponding characteristic function. Under such a representation, we then approximate the perimeter term (*i.e.*, regularization term) by a functional of characteristic functions in terms of heat kernel convolution and propose a novel iterative convolution–thresholding method (ICTM) to minimize (1), which can be applied to a range of models for image segmentation. ICTM alternatively updates Θ_i and the characteristic functions in a simple way, with an optimal computational complexity $O(N \log(N))$ (N : number of pixels) at each iteration. In addition, we theoretically prove that each iteration decreases the value of the objective functional, implying the convergence of the method. Extensive numerical results show that ICTM is efficient, robust, and applicable to a wide range of models for image segmentation.

The rest of this paper is organized as follows. Section 2 gives a review on the related work to ICTM. Section 3 gives the derivation of ICTM and shows the unconditional stability property. Section 4 presents a variety of numerical experiments to show the performance of the ICTM. Section 5 provides a discussion and conclusion.

2. Related work

This section reviews works related to ICTM. ICTM is motivated by a threshold dynamics method developed by Merriman, Bence, and Osher (MBO) [16] to simulate the motion of an interface driven by the mean curvature. More precisely, at each time step, let $D \subset$

\mathbb{R}^n be a domain where its boundary $\Gamma = \partial D$ is to be evolved via mean curvature flow. The MBO method then generates a new interface, Γ_{new} (or equivalently, D_{new}) via the following two steps:

Step 1. Solve the initial value problem for the heat diffusion equation until time $t = \tau$,

$$u_t = \Delta u,$$

$$u(t = 0, \cdot) = \chi_D,$$

where χ_D is the characteristic function of the domain D . Let $\tilde{u}(x) = u(\tau, x)$.

Step 2. Obtain a new domain D_{new} with boundary $\Gamma_{\text{new}} = \partial D_{\text{new}}$ by

$$D_{\text{new}} = \left\{ x : \tilde{u}(x) \geq \frac{1}{2} \right\}.$$

The MBO method has been shown to converge to mean curvature flow [17,18]. Esedoglu and Otto [15] generalized this type of method to multiphase flow with arbitrary surface tensions. The method has attracted much attention owing to its simplicity and unconditional stability. It has subsequently been extended to applications including wetting dynamics [19–22], image processing [23–25], data classification [26,27], shape reconstruction [28,29], graph partitioning [30], topology optimization [31,32], foam bubbles [33], and general target-valued harmonic maps [34–36]. Some connections can also be made to central voronoi tessellation (CVT)-based methods (see e.g., [37]).

Other related techniques are the level-set-based methods and phase field methods. In the level-set method, the regularization term is usually approximated by the total variation (TV) of the Heaviside function of the level-set function (see e.g., [5,6]). There are also many other modifications on the regularization term. For instance, Chien et al. [38] introduced a frame-based model in which the perimeter term is approximated by a term involving framelets. A phase-field approximation of the energy was proposed in [23] for the two-phase CV model, in which the Ginzburg–Landau functional is used to approximate the perimeter of the domain. Recently, in [39,40], the Allen–Cahn model was studied for segmentation of intensity-inhomogeneous images.

ICTM also uses a convex relaxation idea developed in [41]. Usually, the binary segmentation problem consists of estimating a binary mask $u : \Omega \rightarrow \{0, 1\}$ that separates the image into two different areas. Chan et al. [41] relaxed the binary problem so that $u(x)$ takes its values in the interval $[0, 1]$, and the relaxed energy is usually convex in u , which is defined over the convex function set. Then, a globally optimal solution u^* may be computed using a gradient descent scheme. Pock et al. [42] proposed a convexification technique to write the nonlinearities of the functional in a convex way by introducing an auxiliary variable. Many applications and extensions have been introduced to refine and extend convexification methods for image segmentation (see e.g., [43]).

3. Derivation of the method

In this section, we derive ICTM to minimize (1). For simplicity, we first derive ICTM for the binary segmentation in Section 3.1. The generalization of the method to the multi-segment case is straightforward, as we show in Section 3.2.

3.1. Derivation of ICTM for the two-segment case

For simplicity, we describe ICTM in the case of two-phase segmentation. ICTM is a region-based method. In our method, the first segment Ω_1 is denoted by its characteristic function $u(x)$; that is,

$$u(x) := \begin{cases} 1 & \text{if } x \in \Omega_1, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Then, the characteristic function of the second segment Ω_2 is $u_2(x) = 1 - u(x)$. Note that the interface between two segments is now implicitly represented by $u(x)$.

As pointed out in [15], when $\tau \ll 1$, the length of $\partial\Omega_1$ can be approximated by

$$|\partial\Omega_1| \approx \sqrt{\frac{\pi}{\tau}} \int_{\Omega} u G_{\tau} * (1-u) dx, \quad (3)$$

where $*$ represents convolution, and

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

The rigorous proof of the convergence as $\tau \rightarrow 0$ can be found in Miranda et al. [44].

The fidelity terms in (1) can be written into an integral on the whole domain Ω by multiplying the integrand by u or $1-u$. That is,

$$\int_{\Omega_1} F_1 dx = \int_{\Omega} u F_1 dx, \quad \int_{\Omega_2} F_2 dx = \int_{\Omega} (1-u) F_2 dx.$$

The total energy (1) can thus be approximated by

$$\mathcal{E} \approx \mathcal{E}^{\tau}(u, \Theta) := \mathcal{E}_f(u, \Theta) + \mathcal{E}_r^{\tau}(u, \Theta), \quad (5)$$

where

$$\mathcal{E}_f(u, \Theta) = \int_{\Omega} (u F_1(f, \Theta) + (1-u) F_2(f, \Theta)) dx,$$

and

$$\mathcal{E}_r^{\tau}(u, \Theta) = \lambda \sqrt{\frac{\pi}{\tau}} \int_{\Omega} u G_{\tau} * (1-u) dx.$$

The convergence of \mathcal{E}^{τ} to \mathcal{E} when $\tau \rightarrow 0$ can be referred to [44]. The solution for the segmentation can thus be approximated by finding $u^{\tau,*}$ such that

$$(u^{\tau,*}, \Theta^{\tau,*}) = \arg \min_{u \in \mathcal{B}, \Theta \in \mathcal{S}} \mathcal{E}^{\tau}(u, \Theta), \quad (6)$$

where

$$\mathcal{B} := \{u(x) \in BV(\Omega, \mathbb{R}) \mid u(x) \in \{0, 1\}, \forall x \in \Omega\},$$

$BV(\Omega, \mathbb{R})$ denotes the bounded-variation functional space, and $\mathcal{S} = S_1 \times S_2 \times \dots \times S_n$, with S_i being the admissible sets for Θ_i .

Now, we apply the coordinate descent method [45] to minimize $\mathcal{E}^{\tau}(u, \Theta)$; that is, starting from an initial guess (u^0) , we find the minimizers iteratively in the following order:

$$\Theta^0, u^1, \Theta^1, \dots, u^k, \Theta^k, \dots$$

Without loss of generality, assuming that u^k is calculated, we fix u^k and find the minimizer of $\mathcal{E}^{\tau}(u^k, \Theta)$ to obtain Θ^k . That is,

$$\Theta^k = \arg \min_{\Theta \in \mathcal{S}} \mathcal{E}^{\tau}(u^k, \Theta). \quad (7)$$

Here and in the subsequent sections, we generally assume that for the n -segment case, the global minimizer of

$$\sum_{i=1}^n \int_{\Omega_i} F_i(f, \Theta_1, \dots, \Theta_n) dx,$$

exists and is unique on \mathcal{S} .

Remark 3.1. This assumption is reasonable for models for image processing because most models use strictly convex fidelity terms, such as those in CV, LIF, LGIF, and LSAC.

Because \mathcal{E}_r^{τ} is independent of Θ , one only needs to find the global minimizers of \mathcal{E}_f with respect to Θ to obtain Θ^k . That is,

$$\begin{aligned} \Theta^k &= \arg \min_{\Theta \in \mathcal{S}} \mathcal{E}_f(u^k, \Theta) \\ &= \arg \min_{\Theta \in \mathcal{S}} \int_{\Omega} (u^k F_1 + (1-u^k) F_2) dx. \end{aligned} \quad (8)$$

This optimization problem can be solved in different ways for different types of functionals. For example, if \mathcal{E}_f is strictly convex and differentiable with respect to each element in Θ , then each element $\Theta_{i,j}$ ($i = 1, 2, j \in [m]$) in Θ^k can be obtained by solving the following system of equations:

$$\begin{cases} \frac{\partial \mathcal{E}_f}{\partial \Theta_{1,1}} = 0, \dots, \frac{\partial \mathcal{E}_f}{\partial \Theta_{1,m}} = 0, \\ \frac{\partial \mathcal{E}_f}{\partial \Theta_{2,1}} = 0, \dots, \frac{\partial \mathcal{E}_f}{\partial \Theta_{2,m}} = 0. \end{cases} \quad (9)$$

Remark 3.2. We use the notation $\frac{\partial(\cdot)}{\partial \cdot}$ to denote either variation (when $\Theta_{i,j}$ represents scalar functions) or derivative (when $\Theta_{i,j}$ represents scalar variables). Then, (9) can be either solved exactly or numerically using the Gauss–Seidel strategy, similar to that used in [46] (see examples in Section 4).

After solving Θ^k , we then solve u^{k+1} by

$$u^{k+1} = \arg \min_{u \in \mathcal{B}} \mathcal{E}^{\tau}(u, \Theta^k). \quad (10)$$

Note that the set \mathcal{B} contains the boundary points of the convex set \mathcal{K} :

$$\mathcal{K} := \{u(x) \in BV(\Omega, \mathbb{R}) \mid u(x) \in [0, 1], \forall x \in \Omega\}.$$

In other words, \mathcal{K} is the convex hull of \mathcal{B} .

When Θ^k is fixed, it is easy to check that $\mathcal{E}^{\tau}(u, \Theta^k)$ is a concave functional, because $\mathcal{E}_f(u, \Theta^k)$ is linear and $\mathcal{E}_r^{\tau}(u, \Theta^k)$ is concave. Given the fact that the minimizer of a concave functional on a convex set can always be attained at the boundary points of the convex set and by finding a minimizer on a convex set \mathcal{K} , we relax the original problem (10), resulting in the following equivalent problem:

$$u^{k+1} = \arg \min_{u \in \mathcal{K}} \mathcal{E}^{\tau}(u, \Theta^k). \quad (11)$$

Note that

$$\begin{aligned} \mathcal{E}^{\tau}(u, \Theta^k) &= \int_{\Omega} u F_1(f, \Theta^k) + (1-u) F_2(f, \Theta^k) dx \\ &\quad + \lambda \sqrt{\frac{\pi}{\tau}} \int_{\Omega} u G_{\tau} * (1-u) dx, \end{aligned}$$

is only dependent on u . One can compute the variation of $\mathcal{E}^{\tau}(u, \Theta^k)$ with respect to u as follows:

$$\begin{aligned} \left\langle \frac{\delta \mathcal{E}^{\tau}(u, \Theta^k)}{\delta u}, v \right\rangle &= \lim_{\varepsilon \rightarrow 0} \frac{\mathcal{E}^{\tau}(u + \varepsilon v, \Theta^k) - \mathcal{E}^{\tau}(u, \Theta^k)}{\varepsilon} \\ &= \int_{\Omega} (F_1(f, \Theta^k) - F_2(f, \Theta^k)) v dx \\ &\quad + \lambda \sqrt{\frac{\pi}{\tau}} \int_{\Omega} v G_{\tau} * (1-u) dx \\ &\quad + \lambda \sqrt{\frac{\pi}{\tau}} \int_{\Omega} u G_{\tau} * (-v) dx \\ &= \int_{\Omega} (F_1(f, \Theta^k) - F_2(f, \Theta^k)) v dx \\ &\quad + \lambda \sqrt{\frac{\pi}{\tau}} \int_{\Omega} v G_{\tau} * (1-u) dx \\ &\quad + \lambda \sqrt{\frac{\pi}{\tau}} \int_{\Omega} v G_{\tau} * (-u) dx \end{aligned}$$

$$= \int_{\Omega} v(F_1(f, \Theta^k) - F_2(f, \Theta^k) + \lambda \sqrt{\frac{\pi}{\tau}} G_{\tau} * (1 - 2u)) dx,$$

where the third equality is due to the fact that $\int_{\Omega} vG_{\tau} * (-u) dx = \int_{\Omega} uG_{\tau} * (-v) dx$ and $\langle \cdot, \cdot \rangle$ denotes the L^2 inner product. The linearization of $\mathcal{E}^{\tau}(u, \Theta^k)$ at u^k is then written by

$$\begin{aligned} \mathcal{L}^{\tau}(f, \Theta^k, u^k, u) &:= \mathcal{E}^{\tau}(u^k, \Theta^k) + \left\langle u - u^k, \frac{\delta \mathcal{E}^{\tau}(u, \Theta^k)}{\delta u} \Big|_{u=u^k} \right\rangle \\ &= \int_{\Omega} u\phi^k dx + \mathcal{N}^k, \end{aligned}$$

where \mathcal{N}^k is the collection of all terms depending on u^k , and

$$\phi^k = F_1^k - F_2^k + \lambda \sqrt{\frac{\pi}{\tau}} G_{\tau} * (1 - 2u^k),$$

with $F_i^k = F_i(f, \Theta^k)$. The sequential linear programming then leads us to the following linearized problem:

$$u^{k+1} = \arg \min_{u \in \mathcal{K}} \mathcal{L}^{\tau}(f, \Theta^k, u^k, u) = \arg \min_{u \in \mathcal{K}} \int_{\Omega} u\phi^k dx. \quad (12)$$

After the above relaxation and linearization, the optimization problem (10) is approximated by minimizing a linear functional over a convex set. Because $u(x) \geq 0$, it can be carried out in a pointwise manner by checking whether $\phi^k(x) > 0$ or not. That is, the minimum can be attained at

$$u^{k+1}(x) = \begin{cases} 1 & \text{if } \phi^k(x) \leq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

Now, combining (8) and (13) yields [Algorithm 1](#).

Algorithm 1: The ICTM for approximating minimizers of (1).

Input: Ω : image domain; f : image, $\tau > 0$; $tol > 0$: tolerance; and $u^0 \in \mathcal{B}$.

Output: $u^* \in \mathcal{B}$.

while $\|u^{k+1} - u^k\|_2 > tol$ **do**

1. For the fixed u^k , find

$$\Theta^k = \arg \min_{\Theta \in \mathcal{S}} \int_{\Omega} u^k F_1 + (1 - u^k) F_2 dx.$$

2. Use Θ^k and evaluate

$$\phi^k(x) = F_1^k - F_2^k + \lambda \sqrt{\frac{\pi}{\tau}} G_{\tau} * (1 - 2u^k).$$

3. Set

$$u^{k+1}(x) = \begin{cases} 1 & \text{if } \phi^k(x) \leq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Remark 3.3. Let N be the number of pixels. In [Algorithm 1](#), Step 1 is usually a pointwise assignment of values that can be computed in $O(N)$ operations for most variational models. In Step 2, the convolution is computed by fast Fourier transform (FFT), and thus the computational complexity is $O(N \log(N))$. Thus, the total computational complexity at each iteration for [Algorithm 1](#) is $O(N \log(N))$.

[Theorem 3.4](#) below shows that the total energy $\mathcal{E}^{\tau}(u, \Theta)$ decreases in the iteration for any $\tau > 0$. Therefore, our iteration algorithm always converges to a stationary partition for any initial partition.

Theorem 3.4 (Stability). Let (u^k, Θ^k) be the k th iteration derived in [Algorithm 1](#). We have

$$\mathcal{E}^{\tau}(u^{k+1}, \Theta^{k+1}) \leq \mathcal{E}^{\tau}(u^k, \Theta^k),$$

for any τ .

Proof. The proof has two parts:

$$\mathcal{E}^{\tau}(u^{k+1}, \Theta^k) \leq \mathcal{E}^{\tau}(u^k, \Theta^k),$$

and

$$\mathcal{E}^{\tau}(u^{k+1}, \Theta^{k+1}) \leq \mathcal{E}^{\tau}(u^{k+1}, \Theta^k).$$

The second is a direct consequence of (8). Therefore, we only need to prove the first one.

Write

$$\mathcal{L}^{\tau}(f, \Theta^k, u^k, u^k) = \mathcal{E}^{\tau}(u^k, \Theta^k) - \frac{\lambda \sqrt{\pi}}{\sqrt{\tau}} \int_{\Omega} u^k G_{\tau} * u^k dx,$$

and

$$\begin{aligned} \mathcal{L}^{\tau}(f, \Theta^k, u^k, u^{k+1}) &= \mathcal{E}^{\tau}(u^{k+1}, \Theta^k) \\ &\quad + \frac{\lambda \sqrt{\pi}}{\sqrt{\tau}} \int_{\Omega} u^{k+1} G_{\tau} * (u^{k+1} - 2u^k) dx. \end{aligned}$$

From (13), we have

$$\mathcal{L}^{\tau}(f, \Theta^k, u^k, u^{k+1}) \leq \mathcal{L}^{\tau}(f, \Theta^k, u^k, u^k).$$

That is,

$$\begin{aligned} \mathcal{E}^{\tau}(u^{k+1}, \Theta^k) &\leq \mathcal{E}^{\tau}(u^k, \Theta^k) - \frac{\lambda \sqrt{\pi}}{\sqrt{\tau}} \int_{\Omega} (u^k - u^{k+1}) G_{\tau} * (u^k - u^{k+1}) dx \\ &= \mathcal{E}^{\tau}(u^k, \Theta^k) - \frac{\lambda \sqrt{\pi}}{\sqrt{\tau}} \int_{\Omega} [G_{\tau/2} * (u^k - u^{k+1})]^2 dx \\ &\leq \mathcal{E}^{\tau}(u^k, \Theta^k). \end{aligned}$$

□

As we will show by numerical examples in [Section 4](#), ICTM converges fast, and the number of iterations for convergence is greatly reduced. One can understand this advantage of the ICTM as follows: The approximate energy functional (5) is the summation of a strictly convex functional (or more generally, a functional with a global minimizer) with respect to Θ (i.e., \mathcal{E}_f) and a concave functional only dependent on u (i.e., \mathcal{E}_r^{τ}). In the first step, Θ^k is the optimal choice to decrease the energy. In the second and the third steps, we find the minimizer of the linear approximation, which is also the optimal choice to minimize the linearized functional. Moreover, the minimizer can give a smaller value in (5) because the graph of the functional \mathcal{E}_r^{τ} is always below its linear approximation. This accelerates the convergence of ICTM.

3.2. Derivation of ICTM for the multi-segment case

To derive ICTM for the multi-segment case with n segments, we use n characteristic functions and define

$$u_i(x) = \chi_{\Omega_i}(x) := \begin{cases} 1 & \text{if } x \in \Omega_i, \\ 0 & \text{otherwise,} \end{cases} \quad i \in [n] \quad (14)$$

where $[n]$ denotes $\{1, 2, \dots, n\}$. Then, we denote $u = (u_1, u_2, \dots, u_n)$ and define

$$\tilde{\mathcal{B}} = \{u \in BV(\Omega, \mathbb{R}^n) \mid u_i \in \{0, 1\}, i \in [n], \text{ and } \sum_{i=1}^n u_i(x) = 1\}.$$

In the n -segment case, similar to (3), the perimeter of Ω_i is approximated by

$$|\partial \Omega_i| \approx \sqrt{\frac{\pi}{\tau}} \int_{\Omega} u_i G_{\tau} * (1 - u_i) dx.$$

Then, the total energy (1) can be approximated by

$$\mathcal{E}^{\tau}(u, \Theta) = \mathcal{E}_f(u, \Theta) + \mathcal{E}_r^{\tau}(u, \Theta), \quad (15)$$

where $\mathcal{E}_f(u, \Theta) = \sum_{i=1}^n \int_{\Omega} u_i F_i(f, \Theta) dx$ and

$$\mathcal{E}_r^\tau(u, \Theta) = \lambda \sqrt{\frac{\pi}{\tau}} \sum_{i=1}^n \int_{\Omega} u_i G_\tau * (1 - u_i) dx.$$

Again, we apply the coordinate descent method to minimize $\mathcal{E}^\tau(u, \Theta)$. When u^k is fixed, Θ^k can be obtained via

$$\Theta^k = \arg \min_{\Theta \in \mathcal{S}} \mathcal{E}_f(u^k, \Theta). \quad (16)$$

After Θ^k is computed, using the same relaxation and linearization procedure as in [Section 3.1](#), we arrive at

$$u^{k+1} = \arg \min_{u \in \tilde{\mathcal{K}}} \mathcal{L}^\tau(f, \Theta^k, u^k, u), \quad (17)$$

where

$$\mathcal{L}^\tau(f, \Theta^k, u^k, u) = \sum_{i=1}^n \int_{\Omega} u_i \phi_i^k dx,$$

$$\phi_i^k = F_i(f, \Theta^k) + \lambda \sqrt{\frac{\pi}{\tau}} G_\tau * (1 - 2u_i^k),$$

and $\tilde{\mathcal{K}}$ denotes the convex hull of $\tilde{\mathcal{B}}$.

Then, the minimum is attained at

$$u_i^{k+1}(x) = \begin{cases} 1 & \text{if } i = \arg \min_{\ell \in [n]} \phi_\ell^k(x), \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

Remark 3.5. Note that in [\(18\)](#), $\arg \min_{\ell \in [n]} \phi_\ell^k$ may have more than one solution. In this case, we simply set $i = \min\{\arg \min_{\ell \in [n]} \phi_\ell^k\}$.

The algorithm is almost identical to [Algorithm 1](#), except it replaces step 3 with [\(18\)](#). In addition, we have [Theorem 3.6](#), which is same as [Theorem 3.4](#), to guarantee that the total energy $\mathcal{E}^\tau(u, \Theta)$ decreases in the iteration for any $\tau > 0$. Therefore, ICTM always converges to a stationary solution for any initial guess and an arbitrary number of segments.

Theorem 3.6 (Stability). Let (u^k, Θ^k) be the k th iteration as computed using [\(16\)](#) and [\(18\)](#). We have

$$\mathcal{E}^\tau(u^{k+1}, \Theta^{k+1}) \leq \mathcal{E}^\tau(u^k, \Theta^k),$$

for any τ .

The proof is similar to that in [Theorem 3.4](#).

4. Numerical experiments

In this section, we demonstrate the efficiency of the proposed algorithms by numerical examples. We implement the algorithms in MATLAB installed on a laptop with a 2.7-GHz Intel Core i5 processor and 8 GB of RAM. We apply our methods to different models and also compare our results with those obtained from the level-set methods in Li et al. [\[8\]](#) and Zhang et al. [\[10\]](#). Our results show the clear advantage of ICTM in terms of simplicity and efficiency. The software for ICTM and all images used in all experiments are available at <https://github.com/dongwang91/ICTM>. Most images in the experiments are from [\[5,8,10\]](#) and corresponding software code.

We first apply ICTM to the CV model to discuss several properties (e.g., performance, efficiency, and tuning parameters) on ICTM in [Section 4.1](#). In [Sections 4.2](#) and [4.3](#), we apply ICTM to the LSAC and LIF models to show comparisons between ICTM and level-set methods (e.g., quantitative comparison, visual comparison, and efficiency comparison). The results and parameters of the level-set methods are obtained from the software code from <https://www4.comp.polyu.edu.hk/~cslzhang/LSACM/LSACM.htm> for the LSAC model and <http://www.imagecomputing.org/~cmli/code/> for the LIF model. In all numerical experiments, we set the tolerance $tol = 10^{-6}$ in [Algorithm 1](#).

4.1. Applications to the CV

The first application of the proposed ICTM is to the CV model [\[24\]](#). Specifically, in the CV model, the objective functional for the n -segment case is

$$\mathcal{E}_{CV}(\Omega_1, \dots, \Omega_n, C_1, \dots, C_n) = \lambda \sum_{i=1}^n |\partial \Omega_i| + \sum_{i=1}^n \int_{\Omega_i} |C_i - f|^2 dx, \quad (19)$$

where $\partial \Omega_i$ is the boundary of the i th segment Ω_i , $|\partial \Omega_i|$ denotes the perimeter of the domain Ω_i , and λ is a positive parameter.

In the framework of ICTM, we can write

$$F_i(f, \Theta_1, \Theta_2, \dots, \Theta_n) = |C_i - f|^2,$$

where $\Theta_i = C_i$, and $S_i = \mathbb{R}$ for $i \in [n]$.

In Step 1 of [Algorithm 1](#), when u^k is fixed,

$$\int_{\Omega} u |C_1 - f|^2 + (1 - u) |C_2 - f|^2 dx,$$

which is strictly convex with respect to C_1 and C_2 . Hence, direct calculation of the stationary points yields

$$C_1^k = \frac{\int_{\Omega} u^k f dx}{\int_{\Omega} u^k dx}, \quad C_2^k = \frac{\int_{\Omega} (1 - u^k) f dx}{\int_{\Omega} 1 - u^k dx}, \quad (20)$$

which are the average intensities of image f in Ω_1 and Ω_2 , respectively.

For the n -segment case, in Step 1, when u^k is fixed, $\sum_{i=1}^n \int_{\Omega} u_i^k |C_i - f|^2 dx$ is strictly convex with respect to C_i , $i \in [n]$. Hence, the minimizer is given by

$$C_i^k = \frac{\int_{\Omega} u_i^k f dx}{\int_{\Omega} u_i^k dx}, \quad (21)$$

which are the average intensities of the image f in Ω_i . They are all consistent with the definition of C_i in the CV model [\(19\)](#). Then, using C_i^k and u_i^k , one can calculate ϕ^k (or ϕ_i^k for n -segment case) with heat kernel convolution using the FFT followed by the thresholding step (*i.e.*, step 3) to obtain u^{k+1} (or u_i^{k+1}).

Remark 4.1. The update formula of C_i^k in [\(20\)](#) or [\(21\)](#) is general for arbitrary numbers of channels. For instance, if f is a gray image, then C_i^k is a scalar function. If f is a color image with three channels, then the corresponding C_i^k is a vector-valued function.

In the following, we use several experiments to show the performance of the ICTM. In all of the experiments in this section, to make the consistency on the choice of τ , we interpret the domain of the image ($n \times m$ pixels) as a uniform discretization of a continuous computational domain $[-a, a] \times [-b, b]$. Here,

$$\begin{cases} a = \pi, b = \frac{m\pi}{n} & \text{if } n \geq m, \\ a = \frac{n\pi}{m}, b = \pi & \text{if } n < m. \end{cases}$$

If there is no other statement, convolutions are efficiently evaluated using the FFT.

4.1.1. Feasibility of the ICTM

In [Fig. 1](#), we show the results of the ICTM applied to a flower image. The figure shows the initial contour, final contour, and final segments, from left to right. In the sequence, all of the initial guesses u^0 are set to be the characteristic functions of domains enclosed by the corresponding initial contours. In the first row, we use the two-segment case to obtain the two segments of the image; in the second row, we use the n -segment case to obtain four segments of the image. The parameters (τ, λ) are $(0.02, 0.05)$ and $(0.02, 0.02)$, and the numbers of iterations are 15 and 14. The results show that ICTM converges to the stationary solutions in few steps.



Fig. 1. (First row): Two-phase segmentation with $(\tau, \lambda) = (0.02, 0.05)$. (Second row): Four-phase segmentation with $(\tau, \lambda) = (0.02, 0.02)$. The numbers of iterations is 15 and 14. Left to right: Initial contour, final contour, and final segments. The size of the image is 375×500 . See Section 4.1.1.



Fig. 2. Left to right: initial contour, final contour with $\tau = 0.05$ and $\lambda = 0.01, 0.05, 0.07$, and 0.09 . The size of the image is 256×256 . See Section 4.1.2.



Fig. 3. Left to right: final contour with $\lambda = 0.04$ and $\tau = 0.01, 0.03, 0.05, 0.07$, and 0.09 . The size of the image is 256×256 . See Section 4.1.2.

4.1.2. Tuning parameters

To further investigate the dependence of results on the parameters (τ, λ) , we use a cameraman image as an example. We start with the same initial contours and compute the results by tuning the parameters. In Fig. 2, we list the final result with fixed $\tau = 0.05$ and different $\lambda = 0.01, 0.05, 0.07$ and 0.09 . Because λ controls the weight of the regularization term in the objective functional, the total length of the contour decreases as the weight of the regularization term increases. In Fig. 2, we observe that as λ increases, the length of the final contour becomes smaller, and the minimizer tries to drop small noisy parts. The whole contour tends to become more regularized.

In Fig. 3, we compute the final results with fixed $\lambda = 0.04$ and varying $\tau = 0.01, 0.03, 0.05, 0.07$, and 0.09 with the same initial contour as that in Fig. 2. From Fig. 3, one can observe that as τ increases, the final contour becomes relatively smoother and tends to capture more details.

4.1.3. Sensitivity to image noise

In this experiment, we check the sensitivity of the ICTM to the noise of the image. In Fig. 4, we test the ICTM on a color image with different Gaussian noise $\mathcal{N}(0, \Sigma)$ with $\Sigma = 0.025I, 0.05I, 0.1I$, and $0.2I$, where I is the 3×3 identity matrix. Starting with the

same initial contour and same parameter $(\tau, \lambda) = (0.01, 0.1)$, we obtain similar results with four segments after 20, 23, 29, and 48 iterations, respectively. This implies that the ICTM is relatively insensitive to the noise on the image in terms of visual results, parameter settings, and the number of iterations.

4.1.4. Sensitivity to the image resolution

In the last experiment on the CV model, we check the sensitivity to the resolution of images. Figure 5 shows four results considering the same image with different resolutions. Starting from the same initial contour and same parameter $(\tau, \lambda) = (0.01, 0.1)$, we again obtain similar results after 25, 23, 22, and 22 iterations. This indicates the robustness of the ICTM to the resolution of images in the sense of the number of iterations.

We describe the study of the sensitivity to the initial guess in Section 4.2.2.

4.2. Applications to the LSAC model

In this section, we now apply ICTM to the LSAC model where the objective functional is

$$\mathcal{E}_{LSAC}(\Omega_1, \dots, \Omega_n, C_1, \dots, C_n, v_i, \dots, v_n, \rho, b)$$

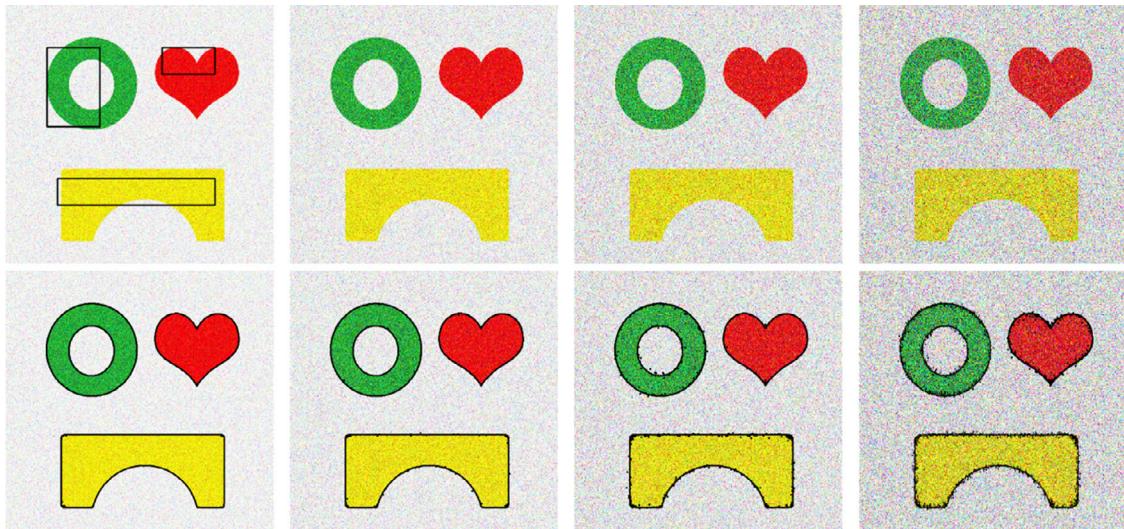


Fig. 4. **Left to right:** Images with Gaussian noise $\mathcal{N}(0, \Sigma)$ where $\Sigma = 0.025I, 0.05I, 0.1I$, and $0.2I$ where I is the 3×3 identity matrix. The initializations are performed using the contours in the first image. The size of the image is 512×512 . See Section 4.1.3.

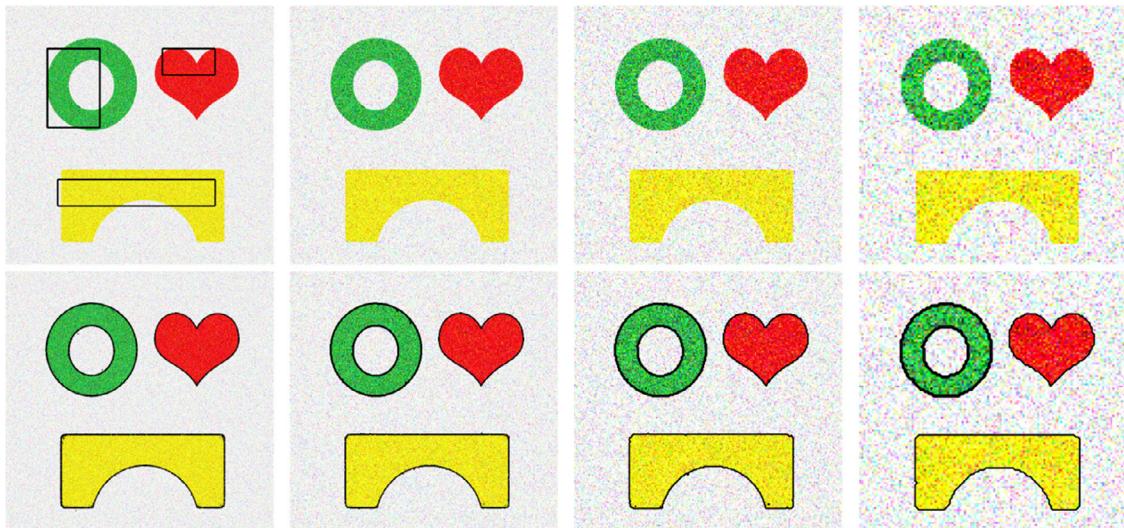


Fig. 5. **Left to right:** Images with 1024×1024 , 512×512 , 256×256 , and 128×128 pixels. The initializations are performed using the contours in the first image. See Section 4.1.4.

$$= \lambda \sum_{i=1}^n |\partial\Omega_i| + \sum_i^n \int_{\Omega_i} \int_{\Omega} I_\rho(x-y) (\log(v_i) + |f(x) - b(y)C_i|^2 / 2v_i^2) dy dx,$$
(22)

where

$$I_\rho(x) = \begin{cases} 1 & \text{if } |x| < \rho, \\ 0 & \text{otherwise,} \end{cases}$$

and ρ is a parameter in the kernel I_ρ . Without loss of generality, we use two-segment examples to demonstrate the efficiency of the ICTM. The n -segment case can be implemented in a similar way.

In the framework of the ICTM, we choose

$$F_i(f, \Theta_1, \Theta_2, \dots, \Theta_n) = \int_{\Omega} I_\rho(x-y) (\log(v_i) + |f(x) - b(y)C_i|^2 / 2v_i^2) dy,$$

and $\Theta_i = (v_i, b(x), C_i)$ for any $i \in [2]$. Direct calculation shows that the global minimizer of

$$\int_{\Omega} u^k F_1(f, \Theta_1, \Theta_2) + (1-u^k) F_2(f, \Theta_1, \Theta_2) dx,$$

occurs at its unique stationary point. By computing the variation and using the Gauss-Seidel strategy, we can obtain an approximation to

Step 1 of Algorithm 1 as

$$\begin{aligned} C_1^k &= \frac{\int_{\Omega} (I_\rho * b^{k-1}) f u^k dx}{\int_{\Omega} (I_\rho * b^{k-1}) u^k dx}, \\ C_2^k &= \frac{\int_{\Omega} (I_\rho * b^{k-1}) f (1-u^k) dx}{\int_{\Omega} (I_\rho * b^{k-1}) (1-u^k) dx}, \\ v_1^k &= \sqrt{\frac{\int_{\Omega} \int_{\Omega} I_\rho(x-y) u^k(x) (f(x) - b^{k-1}(y)C_1^k)^2 dy dx}{\int_{\Omega} \int_{\Omega} I_\rho(x-y) u^k(y) dy dx}}, \\ v_2^k &= \sqrt{\frac{\int_{\Omega} \int_{\Omega} I_\rho(x-y) (1-u^k(x)) (f(x) - b^{k-1}(y)C_2^k)^2 dy dx}{\int_{\Omega} \int_{\Omega} I_\rho(x-y) (1-u^k(y)) dy dx}}, \\ b^k(x) &= \frac{[C_1^k/(v_1^k)^2] I_\rho * (f u^k) + [C_2^k/(v_2^k)^2] I_\rho * (f (1-u^k))}{[(C_1^k/v_1^k)^2] I_\rho * u^k + [(C_2^k/v_2^k)^2] I_\rho * (1-u^k)}. \end{aligned} \quad (23)$$

In Step 2, ϕ^k can then be directly computed based on (23), which is followed by the thresholding step (i.e. Step 3) to obtain u^{k+1} .

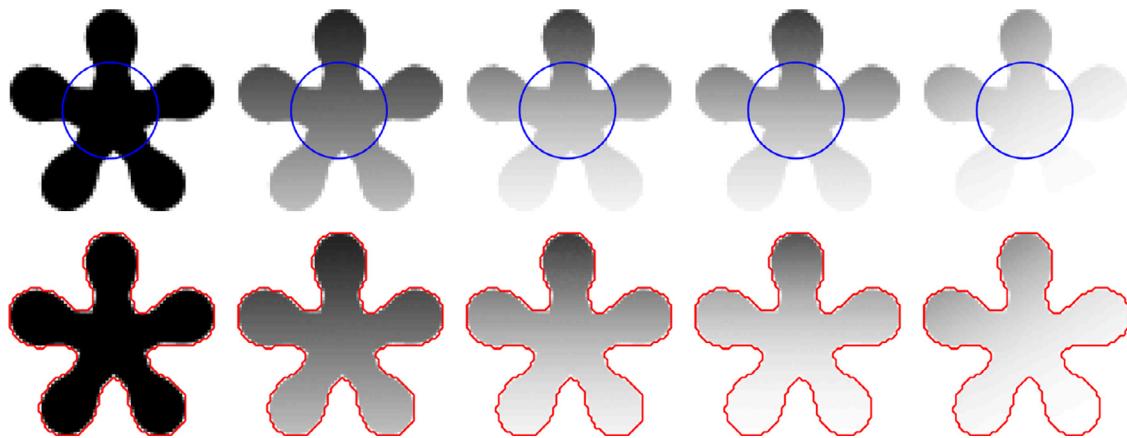


Fig. 6. (First row): Initial contour of the same image with a different intensity inhomogeneity. (Second row): The segmented region. In all five experiments in the ICTM, $\rho = 15$, $\lambda = 0.1$, and $\tau = 0.01$. The image size is 100×100 . See Section 4.2.1.

Table 1

Comparison of the number of iterations and the CPU time for each case from left to right in Fig. 6 between the ICTM and the level-set method in [10]. In all five of the experiments in the ICTM, $\rho = 15$, $\lambda = 0.1$, and $\tau = 0.01$. The image size is 100×100 . See Section 4.2.1.

# of Iterations	ICTM Method in [10]	8	7	7	7	7
CPU time (seconds)	ICTM Method in [10]	0.31	0.30	0.29	0.29	0.29
		0.31	0.33	0.87	2.69	4.00

Table 2

JS index of ICTM and the level-set method in [10]. See Section 4.2.1.

JS Index	ICTM Method in [10]	1	1	0.999	0.994	0.994
		0.985	0.990	0.987	0.981	0.980

In the following, we show several numerical experiments and compare our results with those of Zhang et al. [10] where the level-set approach was used.

4.2.1. Quantitative and efficiency comparison

In this section, we provide a detailed quantitative comparison with the level-set methods applied on the LSAC model. We consider a star-shaped image with the ground truth. Figure 6 shows the segmentation results for five images with different levels of intensity inhomogeneity computed from ICTM. Table 1 shows the efficiency and robustness of the proposed ICTM when compared with the level-set method [10]. The number of iterations needed for the ICTM to converge remains almost the same, at 7 for different intensity inhomogeneities, while the number of iterations increases from 7 to about 240 for the level-set method in Zhang et al. [10]. The CPU time for the ICTM remains almost at 0.30 s, while the CPU time for the level-set method in Zhang et al. [10] changes from 0.31 to 4.00 s for different intensity inhomogeneities. The segmentation results obtained from Zhang et al. [10] are the same as those listed in the second row in Fig. 6.

We also use the Jaccard similarity (JS) as an index to quantitatively measure the accuracy of our segmentation. The JS index between two regions, S_1 and S_2 , is calculated as $JS(S_1, S_2) = |S_1 \cap S_2| / |S_1 \cup S_2|$, which describes the ratio between the intersection areas of S_1 and S_2 . In the five experiments in Fig. 6, we have $JS(S_1, S_2) = 1, 1, 0.999, 0.994$, and 0.994, respectively, when we set S_1 as the numerical result and S_2 as the ground truth. The parameters in the five experiments are all fixed as $(\rho, \lambda, \tau) = (15, 0.1, 0.01)$. In Table 2, we list the comparison of the JS index between ICTM and the level-set method in [10], implying that the

ICTM also outperforms the level-set method in terms of the accuracy.

4.2.2. Sensitivity to the initial guess

To further check the robustness of our results to the initial guess, we choose 10 different initial guesses as displayed in Fig. 7. The final segmentation results are all identical to those in Fig. 6. In all cases, we set $\rho = 20$, $\tau = 0.01$, and $\lambda = 0.1$. In Table 3, we observe that, for any initial guess in Fig. 7, the ICTM converges in about 7 steps, with a CPU time of around 0.3 s. However, in the level-set method in [10], the number of iterations and the CPU time are sensitive to the initial guess, ranging from 8 s to 60 s. Also, the JS index of ICTM, shown in Table 3, remains at 0.994, which is insensitive to the initial guesses; in contrast, those of the level-set method are oscillating.

4.2.3. Efficiency comparison on noisy intensity-inhomogeneous images

We then apply the ICTM to five different, noisy intensity-inhomogeneous images. The results in Fig. 8 again show that our ICTM is efficient and accurate. The parameters for the five figures, from left to right, are $(\rho, \lambda, \tau) = (15, \frac{0.1}{\sqrt{\pi}}, 0.02)$, $(5, \frac{0.26}{\sqrt{\pi}}, 0.03)$, $(10, \frac{0.002}{\sqrt{\pi}}, 0.01)$, $(10, \frac{0.7}{\sqrt{\pi}}, 0.03)$, and $(10, \frac{0.035}{\sqrt{\pi}}, 0.002)$. As presented in Table 4, the number of iterations in ICTM is 5, 48, 15, 29, and 18. However, the number of iterations in the level-set method is 57, 219, 670, 290, and 230. Table 4 shows that ICTM is faster than the level-set method.

4.3. Applications to the LIF model

Finally, we apply ICTM to the LIF model for the two-phase case with the objective functional

$$\begin{aligned} \mathcal{E}_{LIF}(\Omega_1, \dots, \Omega_n, C_1, \dots, C_n) &= \lambda \sum_{i=1}^n |\partial \Omega_i| \\ &+ \sum_{i=1}^n \mu_i \int_{\Omega} \int_{\Omega_i} G_{\sigma}(x-y) |C_i(x) - f(y)|^2 dy dx, \end{aligned} \quad (24)$$

In this case, we choose $F_i(f, \Theta_1, \Theta_2, \dots, \Theta_n) = \mu_i \int_{\Omega} G_{\sigma}(x-y) |C_i(x) - f(y)|^2 dx$ and $\Theta_i = C_i(x)$ for any $i \in [2]$. When (u_1^k, \dots, u_n^k) are fixed,

$$\mathcal{E}_f = \int_{\Omega} u^k F_1(f, C_1, C_2) + (1-u^k) F_2(f, C_1, C_2) dy,$$

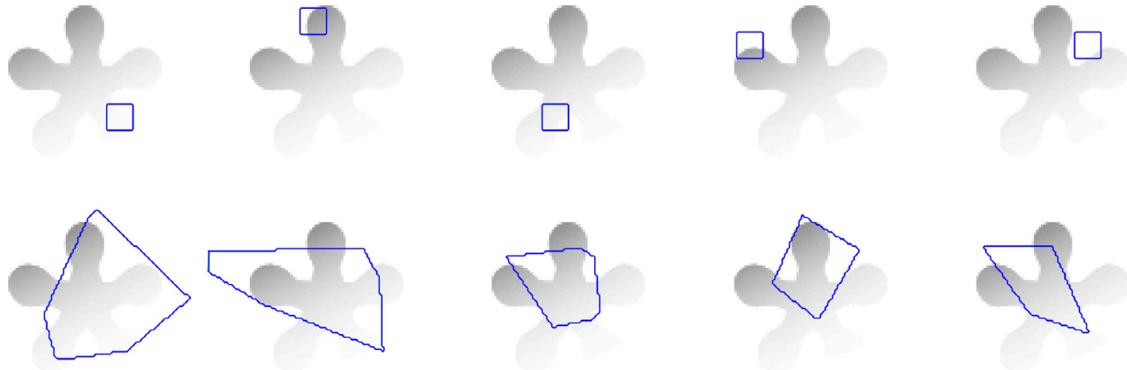


Fig. 7. **Left to right:** 10 different initial guesses for the segmentation. See [Section 4.2.2](#).

Table 3

The comparison between the CPU time and JS indices of the 10 initial guesses in [Fig. 7](#). In both methods, we set $\rho = 20$. In ICTM, $\tau = 0.01$ and $\lambda = 0.1$. See [Section 4.2.2](#).

CPU time (seconds)	ICTM Method in [10]	0.24	0.31	0.31	0.33	0.28	0.29	0.28	0.31	0.29	0.30
JS Index	ICTM Method in [10]	0.994	0.994	0.994	0.994	0.994	0.994	0.994	0.994	0.994	0.994

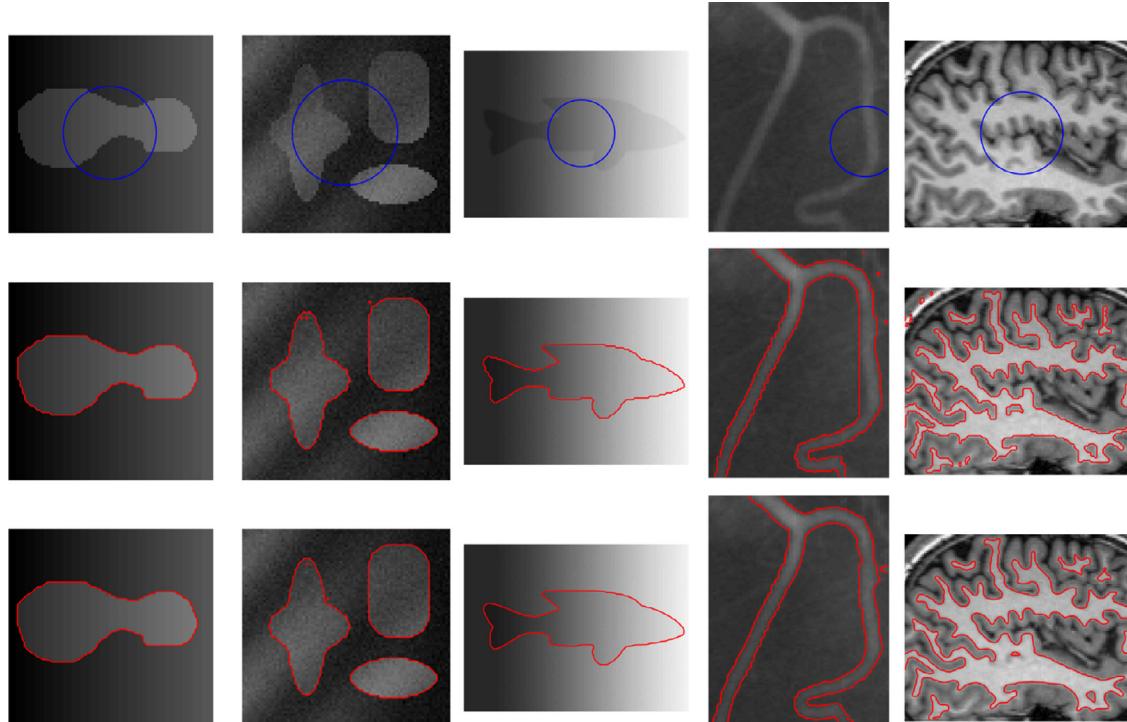


Fig. 8. **(First row):** Initial contour. **(Second row):** Results obtained using the ICTM. **(Third row):** Results obtained from the level-set method in [10]. **Parameters from left to right:** $(\rho, \lambda, \tau) = (15, \frac{0.1}{\sqrt{\pi}}, 0.02)$, $(5, \frac{0.26}{\sqrt{\pi}}, 0.03)$, $(10, \frac{0.002}{\sqrt{\pi}}, 0.01)$, $(10, \frac{0.7}{\sqrt{\pi}}, 0.03)$, and $(10, \frac{0.035}{\sqrt{\pi}}, 0.002)$. **Image sizes from left to right:** 85 × 88, 75 × 79, 100 × 150, 131 × 103, and 91 × 122. See [Section 4.2.3](#).

Table 4

Comparison of the number of iterations and CPU time between ICTM and the level-set method in [10]. See [Section 4.2.3](#).

# of iterations	ICTM Method in [10]	5	48	15	29	18
CPU time (seconds)	ICTM Method in [10]	0.15	0.23	0.31	0.52	0.26

is strictly convex with respect to $C_i(x)$, $i \in [2]$. Then, Step 1 in [Algorithm 1](#) gives

$$\begin{aligned} \iint_{\Omega} u^k(y) G_{\sigma}(x-y) [C_1(x) - f(y)] dy dx &= 0, \\ \iint_{\Omega} (1-u^k(y)) G_{\sigma}(x-y) [C_2(x) - f(y)] dy dx &= 0, \end{aligned}$$

whose solutions are given by

$$C_1^k(x) = \frac{G_{\sigma} * (u^k f)}{G_{\sigma} * u^k}, \quad C_2^k(x) = \frac{G_{\sigma} * ((1-u^k) f)}{G_{\sigma} * (1-u^k)}. \quad (25)$$

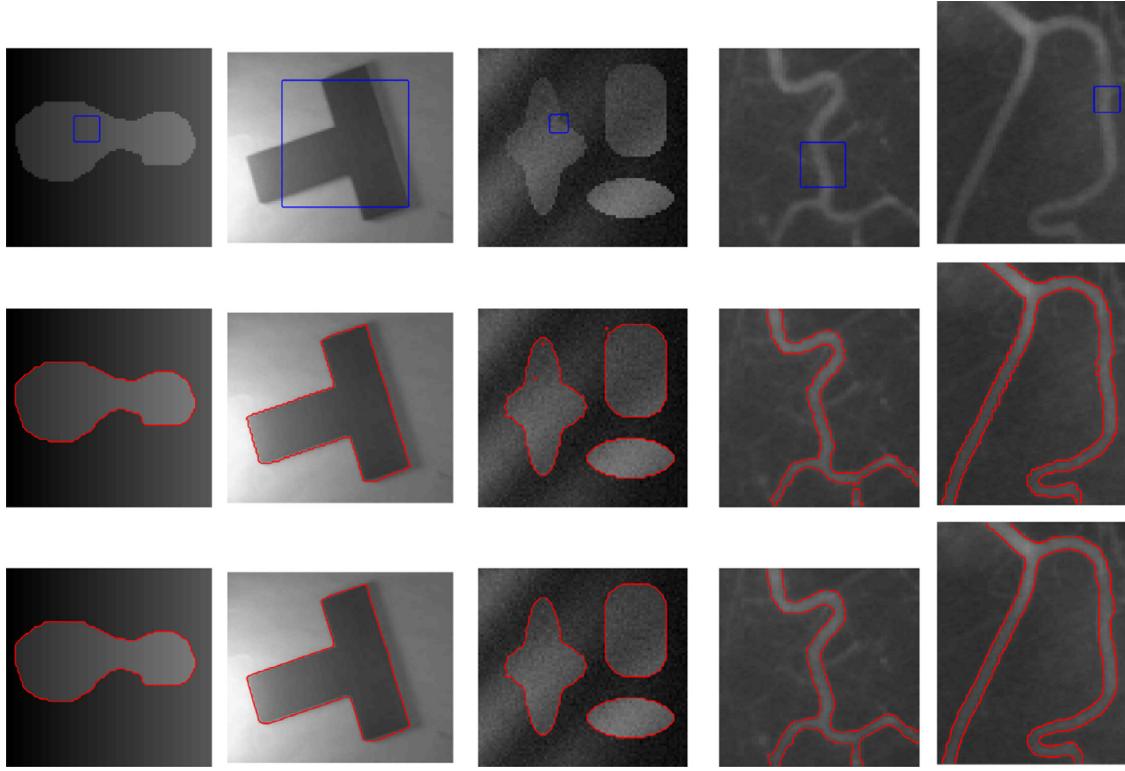


Fig. 9. (First row): Initial contour. (Second row): Results obtained using ICTM. (Third row): Results obtained using the level-set method in [8]. In all five of the experiments, $\mu_1 = \mu_2 = 1$. **Parameters from left to right:** $(\sigma, \tau, \lambda) = (3, 5, 50), (3, 5, 150), (3, 3, 245), (3, 10, 110)$, and $(3, 2, 90)$. **Image sizes from left to right:** $85 \times 88, 96 \times 127, 75 \times 79, 110 \times 111$, and 131×103 . See Section 4.3.1.

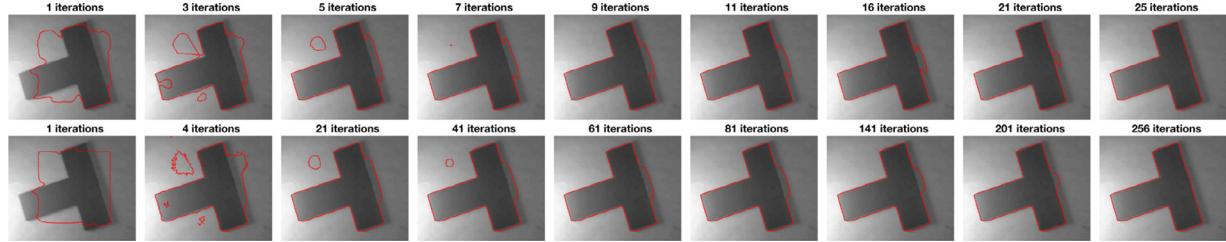


Fig. 10. Comparison between ICTM and the level-set method in [8] on the evolution process from the initial contour to the final contour. First row: results generated from ICTM, Second row: results generated from the level-set method in [8]. See Section 4.3.2.

Remark 4.2. In (25), $C_i^k(x)$ may not be defined at some $x \in \Omega$, as $G_\sigma * u^k$ or $G_\sigma * (1 - u^k)$ can be zero (at least numerically). Because $G_\sigma * u^k \geq 0$ and $G_\sigma * (1 - u^k) \geq 0$, we add a small number ($\varepsilon > 0$) to regularize C_1 and C_2 as follows:

$$C_1^k(x) = \frac{G_\sigma * (u^k f) + \varepsilon}{G_\sigma * u^k + \varepsilon},$$

$$C_2^k(x) = \frac{G_\sigma * ((1 - u^k) f) + \varepsilon}{G_\sigma * (1 - u^k) + \varepsilon}.$$

In the subsequent experiments, we set $\varepsilon = 10^{-6}$.

Again, the evaluation of ϕ^k in step 2 of Algorithm 1 from (23) is followed by the thresholding step (i.e. step 3) to determine u^{k+1} .

In the following, we show numerical examples and compare our results with those of Li et al. [8] using the level-set method. To be consistent with the parameter choice in Li et al. [8] and the implementation in the software code of [8] from <http://www.imagecomputing.org/~cmli/code/>, we generate G_σ by the MATLAB function `fspecial` and compute $G_\sigma * (u^k f)$, $G_\sigma * u^k$, $G_\sigma * ((1 - u^k) f)$, and $G_\sigma * (1 - u^k)$ by the MATLAB function `conv2`.

Table 5

Comparison of the number of iterations and CPU time between ICTM and the level-set method in [8] for the results, presented from left to right in Fig. 9. See Section 4.3.1.

# of iterations	ICTM Method in [8]	13	25	43	28	47
CPU time (second)	ICTM Method in [8]	0.027	0.08	0.06	0.13	0.10

4.3.1. Efficiency comparison

Figure 9 displays several numerical experiments on different intensity-inhomogeneous images. In all five of the experiments, we set $\mu_1 = \mu_2 = 1$. In Fig. 9, from left to right, we set $(\sigma, \tau, \lambda) = (3, 5, 50), (3, 5, 150), (3, 3, 245), (3, 10, 110)$, and $(3, 2, 90)$. In Table 5, we compare ICTM and the level-set method used by Li et al. [8] in terms of the number of iterations for convergence and the CPU time. In all of the examples, ICTM converges in significantly fewer iterations and a shorter CPU time, demonstrating its higher efficiency.

4.3.2. Evolution of the contour

We further use the second example in Fig. 9 to study the curve evolution process from the initial contour to the final contour. In

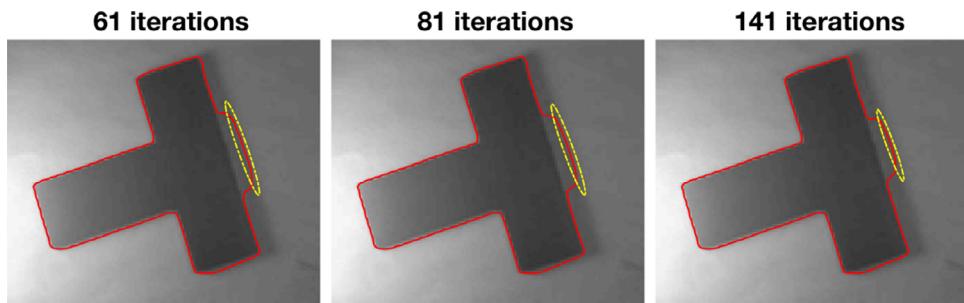


Fig. 11. Selected snapshots of the curve evolution in the level-set method in [8]. See Section 4.3.2.

Fig. 10, we provide several snapshots of the curve evolution of ICTM and the level-set method used in [8]. In the first several steps, the evolution process of the level-set method is similar to that of ICTM. However, when there is a flat curve (highlighted using yellow dashed lines in Fig. 11), ICTM evolves much faster than the level-set method. We note that the evolution is driven by the fidelity term and mean curvature from the variational model. Because we have a theoretic guarantee that ICTM is unconditionally stable in any time step, we can use a relatively large step size τ to make the curve evolve faster when the curvature is relative small (i.e., flat part). However, in the level-set method, because of the stability issue, one can only choose a small time step size to gradually evolve the curve, which requires many more steps to converge.

5. Conclusion and discussion

In this paper, we have proposed the ICTM method, which is applicable to a range of models for image segmentation. We have considered image segmentation as the minimization of a general energy functional consisting of a fidelity term of the image and a regularized term. The interfaces between different segments are implicitly determined by the characteristic functions of the segments. In the objective functional, the fidelity part is then written into a linear functional of characteristic functions (i.e., $\int_{\Omega} uF_1 + (1-u)F_2 dx$), and the regularized term is approximated by a concave functional of characteristic functions (i.e., $\lambda\sqrt{\frac{\pi}{\tau}} \int_{\Omega} uG_{\tau} * (1-u) dx$). We have proved the energy-decaying property of the method. Numerical experiments have shown that the method is simple, efficient, unconditionally stable, and insensitive to the number of segments. ICTM converges in significantly fewer iterations than the level-set method for all the examples we have tested.

Numerically, we have observed the fast convergence of the ICTM. However, theoretical analysis on the convergence rate is still required. For noisy images, the current ICTM still requires good initial guesses. We expect that this can be improved by incorporating some efficient initialization techniques into ICTM. This paper has presented a few applications of ICTM to several variational models for image segmentation. However, we believe that ICTM can be applied to a much larger class of image segmentation models. In the era of big data and deep learning, combining the ICTM framework and deep neural networks is a new direction to explore.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

D. Wang acknowledges support from National Natural Science Foundation of China (grant No. 12101524) and the University De-

velopment Fund from The Chinese University of Hong Kong, Shenzhen (UDF01001803). X.-P. Wang acknowledges support from the Hong Kong Research Grants Council GRF (GRF grants 16308421, 16305819, 16303318) and the University Development Fund from The Chinese University of Hong Kong, Shenzhen (UDF01002028). We also thank the referees for their helpful comments.

References

- [1] A. Mitiche, I.B. Ayed, *Variational and Level Set Methods in Image Segmentation*, Springer Berlin Heidelberg, 2011, doi:[10.1007/978-3-642-15352-5](https://doi.org/10.1007/978-3-642-15352-5).
- [2] L.G. Shapiro, G.C. Stockman, *Computer Vision*, Pearson, 2001.
- [3] D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, *Commun. Pure Appl. Math.* 42 (5) (1989) 577–685, doi:[10.1002/cpa.3160420503](https://doi.org/10.1002/cpa.3160420503).
- [4] L. Ambrosio, V.M. Tortorelli, Approximation of functional depending on jumps by elliptic functional via t -convergence, *Commun. Pure Appl. Math.* 43 (8) (1990) 999–1036, doi:[10.1002/cpa.3160430805](https://doi.org/10.1002/cpa.3160430805).
- [5] T.F. Chan, L.A. Vese, Active contours without edges, *IEEE-IP* 10 (2) (2001) 266–277, doi:[10.1109/83.902291](https://doi.org/10.1109/83.902291).
- [6] 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, doi:[10.1023/A:1020874308076](https://doi.org/10.1023/A:1020874308076).
- [7] C. Li, C.-Y. Kao, J.C. Gore, Z. Ding, Implicit active contours driven by local binary fitting energy, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, doi:[10.1109/cvpr.2007.383014](https://doi.org/10.1109/cvpr.2007.383014).
- [8] C. Li, C.-Y. Kao, J. Gore, Z. Ding, Minimization of region-scalable fitting energy for image segmentation, *IEEE Trans. Image Process.* 17 (10) (2008) 1940–1949, doi:[10.1109/tip.2008.2002304](https://doi.org/10.1109/tip.2008.2002304).
- [9] L. Wang, C. Li, Q. Sun, D. Xia, C.-Y. Kao, Active contours driven by local and global intensity fitting energy with application to brain MR image segmentation, *Comput. Med. Imaging Graph.* 33 (7) (2009) 520–531, doi:[10.1016/j.compmedimag.2009.04.010](https://doi.org/10.1016/j.compmedimag.2009.04.010).
- [10] K. Zhang, L. Zhang, K.-M. Lam, D. Zhang, A level set approach to image segmentation with intensity inhomogeneity, *IEEE Trans. Cybern.* 46 (2) (2016) 546–557, doi:[10.1109/tyb.2015.2409119](https://doi.org/10.1109/tyb.2015.2409119).
- [11] E. Bae, J. Yuan, X.-C. Tai, Global minimization for continuous multiphase partitioning problems using a dual approach, *Int. J. Comput. Vis.* 92 (1) (2010) 112–129, doi:[10.1007/s11263-010-0406-y](https://doi.org/10.1007/s11263-010-0406-y).
- [12] 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, doi:[10.1137/120867068](https://doi.org/10.1137/120867068).
- [13] K. Zhang, L. Zhang, H. Song, W. Zhou, Active contours with selective local or global segmentation: a new formulation and level set method, *Image Vis. Comput.* 28 (4) (2010) 668–676, doi:[10.1016/j.imavis.2009.10.009](https://doi.org/10.1016/j.imavis.2009.10.009).
- [14] K. Zhang, H. Song, L. Zhang, Active contours driven by local image fitting energy, *Pattern Recognit.* 43 (4) (2010) 1199–1206, doi:[10.1016/j.patcog.2009.10.010](https://doi.org/10.1016/j.patcog.2009.10.010).
- [15] S. Esedoglu, F. Otto, Threshold dynamics for networks with arbitrary surface tensions, *Commun. Pure Appl. Math.* 68 (5) (2014) 808–864, doi:[10.1002/cpa.21527](https://doi.org/10.1002/cpa.21527).
- [16] B. Merriman, J.K. Bence, S. Osher, Diffusion Generated Motion by Mean Curvature, Department of Mathematics, University of California, Los Angeles, 1992. <http://ftp.math.ucla.edu/pub/camreport/cam92-18.pdf>
- [17] L.C. Evans, Convergence of an algorithm for mean curvature motion, *Indiana Math. J.* 42 (2) (1993) 533–557, doi:[10.1215/0022258X-42-2-533](https://doi.org/10.1215/0022258X-42-2-533).
- [18] D. Swartz, N.K. Yip, Convergence of diffusion generated motion to motion by mean curvature, *Commun. Partial Differ. Eqs.* 42 (10) (2017) 1598–1643, doi:[10.1080/03605302.2017.1383418](https://doi.org/10.1080/03605302.2017.1383418).
- [19] X. Xu, D. Wang, X.-P. Wang, An efficient threshold dynamics method for wetting on rough surfaces, *J. Comput. Phys.* 330 (2017) 510–528, doi:[10.1016/j.jcp.2016.11.008](https://doi.org/10.1016/j.jcp.2016.11.008).
- [20] D. Wang, X.-P. Wang, X. Xu, An improved threshold dynamics method for wetting dynamics, *J. Comput. Phys.* 392 (2019) 291–310, doi:[10.1016/j.jcp.2019.04.037](https://doi.org/10.1016/j.jcp.2019.04.037).

- [21] S. Jiang, D. Wang, X.-P. Wang, An efficient boundary integral scheme for the MBO threshold dynamics method via the NUFFT, *J. Sci. Comput.* 74 (1) (2017) 474–490, doi:[10.1007/s10915-017-0448-1](https://doi.org/10.1007/s10915-017-0448-1).
- [22] D. Wang, S. Jiang, X.-P. Wang, An efficient boundary integral scheme for the threshold dynamics method II: applications to wetting dynamics, *J. Sci. Comput.* 81 (3) (2019) 1860–1881.
- [23] S. Esedoglu, R. Tsai, Threshold dynamics for the piecewise constant Mumford–Shah functional, *J. Comput. Phys.* 211 (1) (2006) 367–384, doi:[10.1016/j.jcp.2005.05.027](https://doi.org/10.1016/j.jcp.2005.05.027).
- [24] D. Wang, H. Li, X. Wei, X.-P. Wang, An efficient iterative thresholding method for image segmentation, *J. Comput. Phys.* 350 (2017) 657–667, doi:[10.1016/j.jcp.2017.08.020](https://doi.org/10.1016/j.jcp.2017.08.020).
- [25] J. Ma, D. Wang, X.-P. Wang, X. Yang, A characteristic function-based algorithm for geodesic active contours, *SIAM J. Imaging Sci.* 14 (3) (2021) 1184–1205, doi:[10.1137/20m1382817](https://doi.org/10.1137/20m1382817).
- [26] E. Merkurjev, T. Kostić, A.L. Bertozzi, An MBO scheme on graphs for classification and image processing, *SIAM J. Imaging Sci.* 6 (4) (2013) 1903–1930, doi:[10.1137/120886935](https://doi.org/10.1137/120886935).
- [27] M. Jacobs, A fast MBO scheme for multiclass data classification, in: *Lecture Notes in Computer Science*, Springer International Publishing, 2017, pp. 335–347, doi:[10.1007/978-3-319-58771-4_27](https://doi.org/10.1007/978-3-319-58771-4_27).
- [28] S. Esedoglu, S. Ruuth, R. Tsai, Threshold dynamics for shape reconstruction and disocclusion, in: *IEEE International Conference on Image Processing 2005*, IEEE, 2005, doi:[10.1109/icip.2005.1530102](https://doi.org/10.1109/icip.2005.1530102).
- [29] D. Wang, An efficient iterative method for reconstructing surface from point clouds, *J. Sci. Comput.* 87 (38) (2021) 1–24, doi:[10.1007/s10915-021-01457-4](https://doi.org/10.1007/s10915-021-01457-4).
- [30] Y. van Gennip, N. Guillen, B. Osting, A.L. Bertozzi, Mean curvature, threshold dynamics, and phase field theory on finite graphs, *Milan J. Math.* 82 (1) (2014) 3–65, doi:[10.1007/s00032-014-0216-8](https://doi.org/10.1007/s00032-014-0216-8).
- [31] H. Chen, H. Leng, D. Wang, X.-P. Wang, An efficient threshold dynamics method for topology optimization for fluids, *SIAM Trans. Appl. Math.* 3 (1) (2022) 26–56, doi:[10.4208/csiam-am.SO-2021-0007](https://doi.org/10.4208/csiam-am.SO-2021-0007).
- [32] W. Hu, D. Wang, X.-P. Wang, An efficient iterative method for the formulation of flow networks, *Commun. Comput. Phys.* (2022) 1–24, to appear.
- [33] D. Wang, A. Cherkaev, B. Osting, Dynamics and stationary configurations of heterogeneous foams, *PLoS ONE* 14 (4) (2019) e0215836, doi:[10.1371/journal.pone.0215836](https://doi.org/10.1371/journal.pone.0215836).
- [34] D. Wang, B. Osting, A diffusion generated method for computing Dirichlet partitions, *J. Comput. Appl. Math.* 351 (2019) 302–316, doi:[10.1016/j.cam.2018.11.015](https://doi.org/10.1016/j.cam.2018.11.015).
- [35] B. Osting, D. Wang, A diffusion generated method for orthogonal matrix-valued fields, *Math. Comput.* 89 (322) (2019) 515–550, doi:[10.1090/mcom/3473](https://doi.org/10.1090/mcom/3473).
- [36] D. Wang, An efficient unconditionally stable method for Dirichlet partitions in arbitrary domains, *arXiv preprint arXiv:2108.11552* (2021) 1–27.
- [37] 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, doi:[10.1016/j.patcog.2011.02.022](https://doi.org/10.1016/j.patcog.2011.02.022).
- [38] A. Chien, B. Dong, Z. Shen, Frame-based segmentation for medical images, *Commun. Math. Sci.* 9 (2) (2011) 551–559, doi:[10.4310/cms.2011.v9.n2.a10](https://doi.org/10.4310/cms.2011.v9.n2.a10).
- [39] C. Liu, Z. Qiao, Q. Zhang, Two-phase segmentation for intensity inhomogeneous images by the Allen–Cahn local binary fitting model, *SIAM J. Sci. Comput.* 44 (1) (2022) B177–B196, doi:[10.1137/21m1421830](https://doi.org/10.1137/21m1421830).
- [40] Z. Qiao, Q. Zhang, Two-phase image segmentation by the Allen–Cahn equation and a nonlocal edge detection operator, *arXiv preprint arXiv:2104.08992* (2021) 1–27.
- [41] T.F. Chan, S. Esedoglu, M. Nikolova, Algorithms for finding global minimizers of image segmentation and denoising models, *SIAM J. Appl. Math.* 66 (5) (2006) 1632–1648, doi:[10.1137/040615286](https://doi.org/10.1137/040615286).
- [42] T. Pock, T. Schoenemann, G. Graber, H. Bischof, D. Cremers, A convex formulation of continuous multi-label problems, in: *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2008, pp. 792–805, doi:[10.1007/978-3-540-88690-7_59](https://doi.org/10.1007/978-3-540-88690-7_59).
- [43] T. Pock, D. Cremers, H. Bischof, A. Chambolle, Global solutions of variational models with convex regularization, *SIAM J. Imaging Sci.* 3 (4) (2010) 1122–1145, doi:[10.1137/090757617](https://doi.org/10.1137/090757617).
- [44] M. Miranda, D. Pallara, F. Paronetto, M. Preunkert, Short-time heat flow and functions of bounded variation in \mathbb{R}^n , *Annales de la faculté des sciences de Toulouse Mathématiques* 16 (1) (2007) 125–145, doi:[10.5802/afst.1142](https://doi.org/10.5802/afst.1142).
- [45] S.J. Wright, Coordinate descent algorithms, *Math. Program.* 151 (1) (2015) 3–34, doi:[10.1007/s10107-015-0892-3](https://doi.org/10.1007/s10107-015-0892-3).
- [46] X.-P. Wang, C.J. Garcia-Cervera, W. E, A Gauss–Seidel projection method for micromagnetics simulations, *J. Comput. Phys.* 171 (1) (2001) 357–372, doi:[10.1006/jcph.2001.6793](https://doi.org/10.1006/jcph.2001.6793).

Dr. D. Wang is an Assistant Professor in the School of Science and Engineering at the Chinese University of Hong Kong, Shenzhen. He has broad interests in analytical and computational methods for problems in applied mathematics, especially in computational fluid dynamics, computational material science, image processing, optimization and machine learning. After receiving his Bachelor degree in Mathematics from Sichuan University in 2013, Dong earned his Ph.D. in Applied Mathematics at the Hong Kong University of Science and Technology in 2017. Before moving to CUHKSZ, he was an Assistant Professor Lecturer in the Department of Mathematics at the University of Utah.

Prof. X.-P. Wang received his BSc Degree in mathematics from Peking University in 1984 and his PhD Degree in mathematics from Courant Institute (NYU) in 1990. He was a postdoctoral at MSRI in Berkeley, University of Colorado in Boulder before he moved to the Hong Kong University of Science and Technology (HKUST) in 1994. He was the Head and Chair Professor of Mathematics Department at HKUST. Prof. Wang is currently a Presidential Chair Professor at SSE/CUHKSZ. Prof. Wang's current research interests are: Modeling and Simulations of Interface Problem and Multi-phase Flow; Topology Optimization and Numerical Methods for Micro-magnetics Simulations. He received Feng Kang Prize of Scientific Computing in 2007 and was a plenary speaker at the SIAM conference on mathematical aspects of materials science (2016) and a plenary speaker at the International Congress of Industrial and Applied Mathematics (2019).