EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Some fast projection methods based on Chan-Vese model for image segmentation

Jinming Duan*, Zhenkuan Pan, Xiangfeng Yin, Weibo Wei and Guodong Wang

Abstract

The Chan-Vese model is very popular for image segmentation. Technically, it combines the reduced Mumford-Shah model and level set method (LSM). This segmentation problem is solved interchangeably by computing a gradient descent flow and expensively and tediously re-initializing a level set function (LSF). Though many approaches have been proposed to overcome the re-initialization problem, the low efficiency for this segmentation problem is still not solved effectively. In this paper, we first investigate the relationship between the L^1-based total variation (TV) regularizer term of Chan-Vese model and the constraint on LSF and then propose a new technique to solve the re-initialization problem. In detail, four fast projection methods are proposed, i.e., split Bregman projection method (SBPM), augmented Lagrangian projection method (ALPM), dual split Bregman projection method (DSBPM), and dual augmented Lagrangian projection method (DALPM). These four methods without re-initialization are faster than the existing approaches. Finally, extensive numerical experiments on synthetic and real images are presented to validate the effectiveness and efficiency of these four proposed methods.

Keywords: Variational level set method; Chan-Vese model; Re-initialization; Projection method

1. Introduction

Image segmentation is a popular research topic in image processing, as it has a number of significant applications in object detection and moving object tracking, resources classification in SAR images, organs segmentation and 3D reconstruction in medical images, etc. Among the segmentation approaches, the variational models [1-4] are one of the influential and effective methods. In detail, the Snake model [5] and Mumford-Shah model [6] are two fundamental models for image segmentation using variational method. The first one is a typical parametric active contour model based on image edges and fast for segmentation. However, this parametric model is not very effective for images with weak edge and meanwhile fails to deal with adaptive topologies. The second one is a typical region-based model, which aims to replace the original image with a piecewise smooth image and a minimum contour for image segmentation by minimizing an energy functional. Theoretically, it is very difficult to optimize this Mumford-Shah functional as it includes two energy terms defined in two-dimensional image space and one-dimensional contour space respectively. In order to implement this model numerically, Aubert et al. [7-9] introduced the concept of shape derivative and transformed the two-dimensional energy term into one-dimensional one. Consequently, the original model becomes a parametric active contour model. Different from [5], authors in [7-9] developed a level set scheme [10] to achieve curve evolution for adaptive topologies. Another routine to optimize the Mumford-Shah model is to transform the term in contour space into the one in image space, which can be achieved via introducing a proper characteristic function for each different phase that represents different feature in an image. An equivalent energy functional of the original Mumford-Shah model was proposed in [11] via elliptic function approximation based on Gamma-convergence theory. Then, this new Gamma-convergence approximated Mumford-Shah model was extended to segment multiphase images [12-14], which forms the first Gamma-convergence family for variational image segmentation. The second family is variational level set method (VLSM) [15] that combines classical LSM and variational method. The most famous model of this family is Chan-Vese model [16], the first one making use of

* Correspondence: duanmujinming@126.com
College of Information Engineering, Qingdao University, Qingdao 266071, China

Heaviside function of LSF to design characteristic function and then realize two-phase piecewise constant image segmentation. Also, this model has been successfully extended for a great number of multiphase image segmentation [17-19]. The third family is variational label function method (VLFM) sometimes also called piecewise constant level set method [20-22] or fuzzy membership function method [23]. However, if the Heaviside function of LSF is considered as a label function, the third family is actually an extended version of the second one.

Technically, the energy functional minimization for image segmentation results in a set of partial differential equations (PDEs), which must be solved numerically. Compared with other traditional methods, the computational efficiency of variational image segmentation model is much slower, so developing its fast numerical algorithms is always a challenging task in this area. Traditionally, the models in the first two families are usually solved by gradient descent flow. Therefore, the resulting Euler equations always include complicated curvature term, which usually leads to slow computational efficiency. Previously, some fast algorithms for optimizing L^1-based total variation (TV) term have already been efficiently applied to the models of third family (VLFM). For example, novel split Bregman algorithm [24,25], dual method [26,27], and augmented Lagrangian method [21,22,28], and these fast algorithms all avoid computing complex curvature associated with TV regularizer term. Therefore, these proposed algorithms can improve the convergent rate to a great extent.

For the second family, the VLSM for image segmentation usually uses zero level set of a continuous sign distance function (SDF) to represent a contour and the geometric features (i.e., normal and curvature) can be calculated naturally via SDF. Along this way, the post-processing of curves and surfaces will be very convenient. However, the LSF is not preserved as a SDF anymore in the contour evolution and thus the geometric virtue on zero level set will be lost. There are two methods [29,30] to overcome this problem: the traditional one is periodically re-initializing the LSF as a SDF by solving a static eikonal equation or a dynamical Hamilton-Jacobi equation using upwind scheme [29,31-33]. However, this is very expensive and tedious and may make the zero level set moving to undesired positions. The novel one is by constraining LSF to remain a SDF during the contour evolution through adding penalty terms into the original energy functional [30,34]. However, the penalty parameter limits the time step for the LSF evolution due to Courant-Friedrichs-Lewy (CFL) condition [35] and thus the SDF cannot be preserved unless penalty parameter is very large, which cannot guarantee the stability of numerical computation. In order to avoid CFL condition, researchers in [36] proposed completely augmented Lagrangian method by introducing eight auxiliary variables and four penalty parameters, leading to numerous sub-minimization and sub-maximization problems for every introduced variable. Therefore, the resulting models are very complicated.

In this paper, we investigate the relationship between the TV regularizer term of Chan-Vese model and the constraint of LSF as a SDF and then propose a new model with fewer auxiliary variables in comparison with [36]. In this case, we can transform the constraint into a very simple algebra equation that can be explicitly implemented via direct projection approach without re-initialization. Based on this explicit model and novel technique, three algorithms in the third family (i.e., split Bregman algorithm, dual method, and augmented Lagrangian method) for optimizing the variational models can be conveniently extended to Chan-Vese model in second family, and thus four fast algorithms are developed (i.e., split Bregman projection method, augmented Lagrangian projection method, dual split Bregman projection method, and dual augmented Lagrangian projection method). Technically, the resulting equations in the proposed four algorithms include four components: (1): a simple Euler-Lagrange equation for LSF and this Euler-Lagrange equation can be solved via fast Gauss-Seidel iteration, (2): a generalized soft thresholding formula in analytical form, (3): a fast iterative formula for dual variable, and (4): a very simple projection formula. These four components can be used elegantly to avoid computing the complex curvature in [16,30,34]. In addition, all the four proposed fast projection methods can preserve full LSF as a SDF precisely without a very large penalty parameter due to the introduced Lagrangian multiplier and Bregman iterative parameter. So a relatively large time step is allowed to be employed to speed up LSF evaluation in comparison with [30,34]. Most importantly, even if the LSF is initialized as a piecewise constant function, it can be corrected automatically due to the iterative projection computation. Therefore, our proposed methods have both higher computational efficiency and better SDF fidelity than those reported in [30,34,36]. What is worth mentioning here is that our proposed algorithms are quite generic and can be easily extended to all models using VLSM for multiphase image segmentation, motion segmentation, 3D reconstruction etc. For example, the case [37] for multiphase image segmentation has been investigated by using augmented Lagrangian projection method in our recent work.

This paper is organized as follows: in Section 2, we first present the Chan-Vese model under VLSM framework and then review some previous approaches with constraint of LSF as a SDF. In Section 3, the fast split Bregman projection method (SBPM), augmented Lagrangian projection method (ALPM), dual split Bregman projection method (DSBPM), dual augmented Lagrangian projection method (DALPM) are presented. In Section 4, extensive numerical experiments have been conducted to compare our proposed fast methods with some existing approaches. Finally, concluding remarks and outlooks are given.

2. The Chan-Vese model and its traditional solution scheme

2.1 Mumford-Shah model

We first introduce the Mumford-Shah model that is the basic of this paper, and it can be discussed below. For a scalar image $f(x)$: $\Omega \to R$, the Mumford-Shah model can be stated as the following energy functional minimization problem

$$\underset{u,\Gamma}{\mathrm{Min}}\left\{E(u,\Gamma)=\alpha\int_{\Omega}(u-f)^2dx+\beta\int_{\Omega/\Gamma}|\nabla u|^2dx+\gamma\int_{\Gamma}ds\right\} \tag{1}$$

where f is the original input image. The objective of this model is to find a piecewise smooth image u and a minimum contour Γ to minimize (1). α, β, and γ are three positive penalty parameters. This problem is hard to solve due to inconsistent dimension μ and Γ. In order to solve Equation 1 approximately, Chan and Vese [16] first combined the reduced Mumford-Shah model [6] and VLSM [10] and proposed the following Chan-Vese model with an idea of dividing an image into two regions

$$\underset{u,\Gamma}{\mathrm{Min}}\left\{E(u,\Gamma)=\alpha_1\int_{\Omega_1}(u_1-f)^2dx+\alpha_2\int_{\Omega_2}(u_2-f)^2dx+\gamma\int_{\Gamma}ds\right\} \tag{2}$$

where $u = (u_1, u_2)$ stands for piecewise constant image mean value in regions Ω_1 and Ω_2, respectively, and $\Omega = \Omega_1 \cup \Omega_2$, $\Omega_1 \cap \Omega_2 = \varnothing$.

2.2 Traditional LSM

In order to understand the Chan-Vese model clearly, let us first recall some concepts of traditional LSM. $\Gamma(t)$ is defined as a closed contour that separates two regions $\Omega_1(t)$ and $\Omega_2(t)$, and a Lipschitz continuous LSF $\phi(x,t)$ is defined as

$$\begin{cases}\phi(x,t)>0 & x\in\Omega_1(t)\\ \phi(x,t)=0 & x\in\Gamma(t)\\ \phi(x,t)<0 & x\in\Omega_2(t)\end{cases} \tag{3}$$

where $\Gamma(t)$ corresponds to zero level set $\{x{:}\phi(x, t) = 0\}$ and its evolution equation can be transformed into zero level set of $\phi(x,t)$. Then, we differentiate $\phi(x,t) = 0$ with respect to t and obtain the following LSF evolution equation

$$\phi_t+\frac{dx}{dt}\cdot\nabla\phi=0 \tag{4}$$

As the normal on $\{x:\phi(x,t)=0\}$ is $\vec{N}=\nabla\phi/|\nabla\phi|$, Equation 4 can be rewritten as the following standard level set evolution equation:

$$\phi_t+v_N|\nabla\phi|=0 \tag{5}$$

where normal velocity v_{N} of $\Gamma(t)$ is $\frac{dx}{dt}\cdot\frac{\nabla\phi}{|\nabla\phi|}$.

Usually, $\phi(x,t)$ is defined as a SDF

$$\begin{cases}\phi(x,t)=d(x,\Gamma(t)) & x\in\Omega_1(t)\\ \phi(x,t)=0 & x\in\Gamma(t)\\ \phi(x,t)=-d(x,(t)) & x\in\Omega_2(t)\end{cases} \tag{6}$$

Where $d(x, \Gamma(t))$ denotes the Euclidean distance from x to $\Gamma(t)$. An equivalent constraint to Equation 6 is the eikonal equation

$$|\nabla\phi(x,t)|=1 \tag{7}$$

In order to satisfy Equation 7, an iterative re-initialization scheme [16] is used to solve the steady state of following equation:

$$\begin{cases}\phi_t+\mathrm{sign}(\phi_0)(|\nabla\phi|-1)=0 & \text{in } \Omega\times R\\ \phi(x,0)=\phi_0 & \text{in } \Omega\end{cases} \tag{8}$$

where ϕ_0 is the function to be reinitialized and sign(ϕ_0) denotes the sign function of ϕ_0.

2.3 The Chan-Vese model under VLSM framework and its solution

By using Heaviside function of LSF and its total variation form, Chan and Vese [16] transformed the model (2) into VLSM. In fact, a Heaviside function is defined as

$$H(x)=\begin{cases}1 & x\geq0\\ 0 & \text{otherwise}\end{cases} \tag{9}$$

Its derivative in the distributional sense is the Dirac function

$$\delta(x)=\frac{\partial H(x)}{\partial x} \tag{10a}$$

According to Equation 9, the characteristic function of Ω_1 and Ω_2 can be defined as

$$\chi_1(x)=H(\phi(x))=\begin{cases}1 & x\in\Omega_1\\ 0 & \text{otherwise}\end{cases} \tag{10b}$$

$$\chi_2(x)=1-H(\phi(x))=\begin{cases}1 & x\in\Omega_2\\ 0 & \text{otherwise}\end{cases} \tag{10c}$$

Based on the co-area formula [38] of characteristic functions, the length term in Equation 2 can be approximately defined in image space Ω as

$$\gamma\int_{\Gamma} ds = \gamma\int_{\Omega}|\nabla H(\phi)|dx = \gamma\int_{\Omega}|\nabla\phi|\delta(\phi)dx \tag{11}$$

Therefore, Equation 2 can be rewritten as the following VLSM:

$$\underset{\phi,u}{\mathrm{Min}}\left\{E(\phi,u_1,u_2) = \alpha_1\int_{\Omega}(u_1-f)^2H(\phi)dx + \alpha_2\int_{\Omega}(u_2-f)^2(1-H(\phi))dx + \gamma\int_{\Omega}|\nabla\phi|\delta(\phi)dx\right\} \tag{12}$$

Equation 12 is a multivariate minimization problem and usually solved via alternative optimization procedure. First fix ϕ to optimize u and then fix u for optimizing ϕ. In detail, when ϕ is fixed, we obtain

$$u_1 = \frac{\int_{\Omega} fH(\phi)dx}{\int_{\Omega} H(\phi)dx}, \quad u_2 = \frac{\int_{\Omega} f(1-H(\phi))dx}{\int_{\Omega}(1-H(\phi))dx} \tag{13}$$

On the other hand, when u is fixed, the sub-problem of optimization with respect to ϕ is as follows:

$$\underset{\phi}{\mathrm{Min}}\left\{E(\phi) = \int_{\Omega} Q_{12}(u_1,u_2)H(\phi)dx + \gamma\int_{\Omega}|\nabla H(\phi)|dx\right\} \tag{14}$$

where $Q_{12}(u_1,u_2) = \alpha_1(u_1-f)^2 - \alpha_2(u_2-f)^2$. In order to solve Equation 14, we need to compute the evolution equation of ϕ via gradient descent flow as

$$\begin{cases} \dfrac{\partial\phi}{\partial t} = \left(\gamma\nabla\cdot\left(\dfrac{\nabla\phi}{|\nabla\phi|}\right) - Q_{12}(u_1,u_2)\right)\delta(\phi) & \text{in } \Omega \\ \dfrac{\partial\phi}{\partial\vec{n}} = 0 & \text{on } \partial\Omega \end{cases} \tag{15}$$

In order to avoid singularity in numerical implementation for Equation 15, the Heaviside function and Dirac function are usually approximated by their regularized version with a small positive regularized parameter ε as

$$H_\varepsilon(\phi) = \frac{1}{2} + \frac{1}{\pi}\,\mathrm{arc}\,tan\left(\frac{\phi}{\varepsilon}\right) \tag{16a}$$

$$\delta_\varepsilon(\phi) = \frac{1}{\pi}\frac{\varepsilon}{\phi^2+\varepsilon^2} \tag{16b}$$

As both the energy functional (12) and the evolution Equation 15 do not include any exact definition of LSF ϕ as a SDF, the ϕ will not be preserved as a SDF during the contour evolution, which leads to accuracy loss in curve or surface expression.

The first correction approach to preserve the LSF as a SDF is solving Equation 8 using upwind scheme after some iterations of ϕ using Equation 15. However, this method is expensive and may cause the interface to shrink and move to undesirable positions. In order to make comparisons with other methods, we name this re-initialization approach as gradient descent equation with re-initialization method (GDEWRM).

The second correction approach, which was proposed by [30] as following, is to add the constraint Equation 7 as a penalty term into Equation 14 in order to avoid the tedious re-initialization process

$$\underset{\phi}{\mathrm{Min}}\left\{E(\phi) = \int_{\Omega} Q_{12}H_\varepsilon(\phi)dx + \gamma\int_{\Omega}|\nabla H_\varepsilon(\phi)|dx + \frac{\mu}{2}\int_{\Omega}(|\nabla\phi|-1)^2dx\right\} \tag{17}$$

Theoretically, μ should be a large penalty parameter in order to sufficiently penalize the constraint $|\nabla\phi| = 1$ as a SDF. However, under such circumstance, we cannot choose a relatively large time step to improve the computational efficiency due to the CFL stability condition [35]. Here, we name this method as gradient descent equation without re-initialization method (GDEWORM).

As an extension of (17), an augmented Lagrangian method (ALM) and a projection Lagrangian method (PLM) are proposed by [34] to remain the LSF as a SDF during the LSF evolution. These two extensions can be expressed as follows, respectively:

$$\underset{\phi}{\mathrm{Min}}\left\{E(\phi,\lambda) = \int_{\Omega} Q_{12}H_\varepsilon(\phi)dx + \gamma\int_{\Omega}|\nabla H_\varepsilon(\phi)|dx + \int_{\Omega}\lambda(|\nabla\phi|-1)dx + \frac{\mu}{2}\int_{\Omega}(|\nabla\phi|-1)^2dx\right\} \tag{18}$$

$$\underset{\phi,\vec{w}}{\mathrm{Min}}\left\{E\left(\phi,\vec{w}\right) = \int_{\Omega} Q_{12}H_\varepsilon(\phi)dx + \gamma\int_{\Omega}|\nabla H_\varepsilon(\phi)|dx + \int_{\Omega}\lambda\left(\left|\vec{w}\right|-1\right)dx + \frac{\mu}{2}\int_{\Omega}\left(\vec{w}-\nabla\phi\right)^2dx\right\} \tag{19}$$

Different from GDEWORM, ALM (18) enforces the constraint $|\nabla\phi| = 1$ via Lagrangian parameter λ. Therefore, a relatively small penalty parameter μ can be chosen to improve the stability of numerical calculation of Equation 18. The PLM (19) is actually proposed by combining variable splitting and penalty approach, so it is more efficient than GDEWORM due to the split technique. However, one drawback still exists: as μ becomes very large, the intermediate minimization process of PLM becomes increasingly ill-conditioned as happened for GDEWORM.

Using the similar idea, [36] introduced four auxiliary variables and four Lagrangian multipliers to deal with the same constrained optimization problem. The minimization problem is reformulated as following, and here, we name it as completely augmented Lagrangian method (CALM).

$$\begin{aligned} E\left(\phi,\varphi,s,\vec{v},\vec{w}\right) = & \int_\Omega Q_{12} s dx + \gamma \int_\Omega \left|\vec{v}\right| dx \\ & + \int_\Omega \lambda_2 (s - H_\varepsilon(\varphi)) dx + \frac{\mu_2}{2} \int_\Omega (s - H_\varepsilon(\varphi))^2 dx \\ & + \int_\Omega \vec{\lambda}_3 \cdot \left(\vec{v} - \nabla s\right) dx + \frac{\mu_3}{2} \int_\Omega \left(\vec{v} - \nabla s\right)^2 dx \\ & + \int_\Omega \lambda_1 (\varphi - \phi) dx + \frac{\mu_1}{2} \int_\Omega (\varphi - \phi)^2 dx \\ & + \int_\Omega \vec{\lambda}_4 \cdot \left(\vec{w} - \nabla \phi\right) dx + \frac{\mu_4}{2} \int_\Omega \left(\vec{w} - \nabla \phi\right)^2 dx \end{aligned} \tag{20}$$

$$s.t \left|\vec{w}\right| = 1. \tag{21}$$

Note that all of the above methods, GDEWRM (8, 14), GDEWORM (17), ALM (18), and PLM (19), only take efforts in how to add the constraint $|\nabla\phi| = 1$ into the original functional and ignore the TV regularizer term $\int_\Omega |\nabla H(\phi)| dx$. Therefore, their resulting evolution equations bring about complex curvature terms, and the computational efficiency will be very slow due to such complicated finite difference scheme for the curvature. Through introducing eight variables in CALM (20), each sub-minimization or sub-maximization problem of this model becomes very simple because there is no curvature term in these sub-problems. However, as we know, every variable including Lagrangian multiplier is defined in the domain of image space, which implies the more variables the model has, the less efficient it will become. Moreover, there are five penalty parameters setting up in the CALM, so the choices of these parameters are more difficult. In order to avoid computing curvature and meanwhile decrease the number of the introduced variables and parameters, we will design fast algorithms in the next section, by taking into full consideration the relationship between regularization term $\int_\Omega |\nabla H(\phi)| dx$ and constraint term $|\nabla\phi| = 1$.

3. Four fast projection methods

The fast split Bregman method [24,25], dual method [26], and augmented Lagrangian method [28] proposed for TV model for image restoration have been successfully extended to the Chan-Vese model under VLFM framework [20,38], but they cannot be directly applied to Chan-Vese model under VLSM framework due to the complex constraint $|\nabla\phi| = 1$. In this section, inspired by these fast algorithms, we aim to design some new fast algorithms for Chan-Vese model [16] without re-initialization under VLSM framework. Through introducing two or three auxiliary variables, the constraint is transformed into a very simple projection formula so that our proposed fast methods are able to avoid both expensive re-initialization process and complex curvature appearance in the evolution equations. Therefore, the proposed methods are faster than their counterparts with higher performance.

In order to state the problem clearly, we rewrite the traditional Chan-Vese model (14) and the constraint (7) as the following:

$$\underset{\phi}{\mathrm{Min}} \left\{ E(\phi) = \int_\Omega Q_{12}(u_1,u_2) H_\varepsilon(\phi) dx + \gamma \int_\Omega |\nabla\phi| \delta_\varepsilon(\phi) dx \right\} \tag{22a}$$

$$s.t. |\nabla\phi| = 1. \tag{22b}$$

Next, we will introduce each fast algorithm separately.

3.1 Split Bregman projection method

Unlike Equations 18 and 19, we do not put the constraint Equation 22b directly into functional Equation 22a. Instead, we introduce an auxiliary splitting variable $\vec{w}$ to replace the $\nabla\phi$ in the TV regularizer term $\int_\Omega |\nabla\phi| \delta_\varepsilon(\phi) dx$. Therefore, the constraint Equation 22b becomes constraint $\left|\vec{w}\right| = 1$ and another constraint $\vec{w} = \nabla\phi$ is produced. Then, we use the Bregman distance technique [25] by introducing Bregman iterative parameter $\vec{b}$ to satisfy the constraint $\vec{w} = \nabla\phi$, so we can transform Equation 22a, b into the following optimization problem:

$$\begin{aligned} \underset{\phi,\vec{w}}{Min} \left\{ E\left(\phi,\vec{w}\right) = \int_\Omega Q_{12}(u_1,u_2) H_\varepsilon(\phi) dx + \gamma \int_\Omega \left|\vec{w}\right| \delta_\varepsilon(\phi) dx \right. \\ \left. + \frac{\theta}{2} \int_\Omega \left(w - \nabla\phi - \vec{b}\right) dx \right\}, \end{aligned}$$

$$s.t. \left|\vec{w}\right| = 1,$$

In order optimize the above problem, we use the iterative technique as

$$\begin{aligned} \left(\phi^{k+1},\vec{w}^{k+1}\right) = & \arg\min_{\phi,\vec{w}} \left\{ E\left(\phi,\vec{w}\right) \right. \\ & = \int_\Omega Q_{12}(u_1,u_2) H_\varepsilon(\phi) dx \\ & + \gamma \int_\Omega \left|\vec{w}\right| \delta_\varepsilon(\phi) dx + \frac{\theta}{2} \int_\Omega \\ & \left. \times \left(w - \nabla\phi - \vec{b}^{k+1}\right) dx \right\}, \end{aligned} \tag{23a}$$

$$s.t. \left|\vec{w}\right| = 1, \tag{23b}$$

where $\theta > 0$ is a penalty parameter, $\vec{w}$ and $\vec{b}$ are vectors, $\vec{b}^{k+1} = \vec{b}^{k} + \nabla\phi^{k} - \vec{w}^{k}$, $\vec{b}^{0} = \vec{w}^{0} = \vec{0}$. The alternating minimization of $E\left(\phi, \vec{w}\right)$ with respect to ϕ and $\vec{w}$ leads to the Euler-Lagrange equations, respectively

$$\begin{cases} Q_{12}(u_1, u_2)\delta_\varepsilon(\phi) + \gamma\left|\vec{w}^{k}\right| \dfrac{\partial\delta_\varepsilon(\phi)}{\partial\phi} + \theta\nabla\cdot\left(\vec{w}^{k} - \nabla\phi - \vec{b}^{k+1}\right) = 0 & \text{in } \Omega \\ \left(\vec{w}^{k} - \nabla\phi - \vec{b}^{k+1}\right)\cdot\vec{n} = 0 & \text{on } \partial\Omega \end{cases}, \tag{24}$$

$$\begin{cases} \gamma\dfrac{\vec{w}}{\left|\vec{w}\right|}\delta_\varepsilon(\phi) + \theta\left(\vec{w} - \nabla\phi^{k+1} - \vec{b}^{k+1}\right) = 0 \\ \text{s. t.} \quad \left|\vec{w}\right| = 1 \end{cases}. \tag{25}$$

Equation 24 can be solved using semi-implicit difference scheme and Gauss-Seidel iterative method, and the first equation of Equation 25 can be expressed as a following generalized soft thresholding formula in analytical form

$$\vec{\tilde{w}}^{k+1} = \mathrm{Max}\left(\left|\nabla\phi^{k+1} + \vec{b}^{k+1}\right| - \frac{\gamma}{\theta}\delta_\varepsilon\left(\phi^{k+1}\right), 0\right)\frac{\nabla\phi^{k+1} + \vec{b}^{k+1}}{\left|\nabla\phi^{k+1} + \vec{b}^{k+1}\right|}. \tag{26}$$

Then, $\left|\vec{w}\right| = 1$ can be guaranteed via a simple projection technique as the following:

$$\vec{w}^{k+1} = \frac{\vec{\tilde{w}}^{k+1}}{\left|\vec{\tilde{w}}^{k+1}\right|}. \tag{27}$$

Note that after computing the projection (27), the constraint $\left|\vec{w}\right| = 1$ is precisely guaranteed so that the constraint $|\nabla\phi| = 1$ is indirectly adjusted by this projection technique when evolution Equation 24 for LSF reaches its steady state.

3.2 Augmented Lagrange projection method

The ALPM proposed in this part is different from previous ALM (18) and CALM (20). Here, we add the constraint $\vec{w} = \nabla\phi$ in energy functional through augmented Lagrangian method and let the constraint $|\nabla\phi| = 1$ as a simple projection of auxiliary variable $\vec{w}$. Compared with CALM (20) including eight variables and four parameters, our augmented Lagrangian projection method is introduced only by two auxiliary variables and one parameter θ. Similar to the Subsection 3.1, we introduce an auxiliary splitting variable $\vec{w}$ such that $\vec{w} \approx \nabla\phi$ when the following energy functional approaches reach minimum.

$$\left(\phi^{k+1}, \vec{w}^{k+1}, \vec{\lambda}^{k+1}\right) = \mathrm{Arg}\,\underset{\vec{\lambda}}{\mathrm{Max}}\,\underset{\phi,\vec{w}}{\mathrm{Min}} \left\{ \begin{aligned} E\left(\phi, \vec{w}, \vec{\lambda}\right) = &\int_\Omega Q_{12}(u_1, u_2)H_\varepsilon(\phi)dx + \gamma\int_\Omega\left|\vec{w}\right|\delta_\varepsilon(\phi)dx \\ &+ \int_\Omega \vec{\lambda}\cdot\left(\vec{w} - \nabla\phi\right)dx + \frac{\theta}{2}\int_\Omega\left(\vec{w} - \nabla\phi\right)^2 dx \end{aligned} \right\} \tag{28a}$$

$$s.t.\left|\vec{w}\right| = 1, \tag{28b}$$

where $\vec{\lambda}$ is the Lagrangian multiplier and θ is a positive penalty parameter. The augmented Lagrangian method reduces the possibility of ill-conditioning and makes the numerical computation stable through iterative Lagrangian multiplier during the process of the minimization. Therefore, different from the previous penalty methods (17, 19) which need a very large penalty parameter to penalize the constraint effectively, the constraint $\vec{w} = \nabla\phi$ of this method can be guaranteed without increasing θ to a very large value. Here, we minimize $E\left(\phi, \vec{w}, \vec{\lambda}\right)$ with respect to ϕ and $\vec{w}$ and maximize $E\left(\phi, \vec{w}, \vec{\lambda}\right)$ with respect to $\vec{\lambda}$. A saddle point of the min-max problems satisfies the following:

$$\begin{cases} Q_{12}(u_1, u_2)\delta_\varepsilon(\phi) + \gamma\left|\vec{w}^{k}\right| \dfrac{\partial\delta_\varepsilon(\phi)}{\partial\phi} + \nabla\cdot\vec{\lambda}^{k} + \theta\nabla\cdot\left(\vec{w}^{k} - \nabla\phi\right) = 0 & \text{in } \Omega \\ \left(\vec{\lambda}^{k} + \theta\left(\vec{w}^{k} - \nabla\phi\right)\right)\cdot\vec{n} = 0 & \text{on } \partial\Omega \end{cases} \tag{29}$$

$$\begin{cases} \gamma\dfrac{\vec{w}}{\left|\vec{w}\right|}\delta_\varepsilon\left(\phi^{k+1}\right) + \vec{\lambda}^{k} + \theta\left(\vec{w} - \nabla\phi^{k+1}\right) = 0 \\ \text{s. t.} \quad \left|\vec{w}\right| = 1 \end{cases} \tag{30}$$

$$\vec{\lambda}^{k+1} = \vec{\lambda}^{k} + \theta\left(\vec{w}^{k+1} - \nabla\phi^{k+1}\right), \vec{\lambda}^{0} = 0 \tag{31}$$

Equation 29 can be solved using the same method as Equation 24, and the first equation of Equation 30 can be solved using the following generalized soft thresholding formula in analytical form

$$\vec{\tilde{w}}^{k+1} = \mathrm{Max}\left(\left|\nabla\phi^{k+1} - \vec{\lambda}^{k}/\theta\right| - \frac{\gamma}{\theta}\delta_\varepsilon\left(\phi^{k+1}\right), 0\right)\frac{\nabla\phi^{k+1} - \vec{\lambda}^{k}/\theta}{\left|\nabla\phi^{k+1} - \vec{\lambda}^{k}/\theta\right|} \tag{32}$$

Then, the second equation of Equation 30 can be implemented as same as Equation 27.

3.3 Dual split Bregman projection method

The dual method [26] is another fast algorithm proposed in recent years for TV model for image restoration, and it has been extensively applied to variational image segmentation models [20] under VLFM framework. In Equation 22a, $\int_\Omega|\nabla\phi|\delta_\varepsilon(\phi)dx$ is not the total variation of ϕ, but its equivalent formula $\int_\Omega|\nabla H_\varepsilon(\phi)|dx$ is the total variation of $H_\varepsilon(\phi)$. Based on this observation, we can introduce a dual variable to replace $\int_\Omega|\nabla H_\varepsilon(\phi)|dx$ with its dual formula $\text{Sup}_{\vec{p}:\left|\vec{p}\right|\le 1}\int_\Omega H_\varepsilon(\phi)\nabla\cdot\vec{p}\,dx$. Thus, Equation 22a can be rewritten as following min-max functional:

$$\left(\phi^{k+1},\vec{p}^{k+1}\right)=\operatorname{Arg}\underset{\phi}{\text{Min}}\ \underset{\vec{p}:\left|\vec{p}\right|\le 1}{\text{Sup}}\left\{E\left(\phi,\vec{p}\right)=\int_\Omega Q_{12}(u_1,u_2)H_\varepsilon(\phi)dx+\gamma\int_\Omega H_\varepsilon(\phi)\nabla\cdot\vec{p}\ dx\right\} \tag{33}$$

For the constraint $|\nabla\phi|=1$ (22b), we first introduce an auxiliary variable $\vec{w}$ and add the new constraint $\vec{w}=\nabla\phi$ into (33) through Split Bregman iterative method, which is expressed as following

$$\left(\phi^{k+1},\overrightarrow{p}^{k+1},\overrightarrow{w}^{k+1}\right)=\operatorname{Arg}\underset{\phi,\vec{w}}{\text{Min}}\ \underset{\vec{p}:\left|\vec{p}\right|\le 1}{\text{Sup}}\left\{\begin{array}{l}E\left(\phi,\vec{p},\vec{w}\right)=\int_\Omega Q_{12}(u_1,u_2)H_\varepsilon(\phi)dx+\gamma\int_\Omega H_\varepsilon(\phi)\nabla\cdot\vec{p}\ dx\\ \qquad+\frac{\theta}{2}\int_\Omega\left(\vec{w}-\nabla\phi-\overrightarrow{b}^{k+1}\right)^2dx\end{array}\right\} \tag{34}$$

Then the constraint $|\nabla\phi|=1$ can be replaced by the constraint $\left|\vec{w}\right|=1$ so that we can conveniently use the projection formula in Equation 27. Actually, the effect of vector Bregman iterative parameter $\vec{b}$ is used to reduce the dependence on the penalty parameter θ, as the same role of Lagrangian multiplier λ in augmented Lagrangian projection method (28a). The Bregman iterative parameter $\vec{b}$ can be updated by $\overrightarrow{b}^{k+1}=\overrightarrow{b}^{k}+\nabla\phi^k-\overrightarrow{w}^{k}$, where $\overrightarrow{b}^0=\overrightarrow{w}^0=\vec{0}$. The Euler-Lagrange equation of ϕ in Equation 34 is derived as

$$\begin{cases}\left(Q_{12}(u_1,u_2)+\gamma\nabla\cdot\vec{p}^k\right)\delta_\varepsilon(\phi)+\theta\nabla\cdot\left(\vec{w}^k-\nabla\phi-\vec{b}^{k+1}\right)=0 & \text{in }\Omega\\ \left(\vec{w}^k-\nabla\phi-\vec{b}^{k+1}\right)\cdot\vec{n}=0 & \text{on }\partial\Omega\end{cases} \tag{35}$$

After ϕ^{k+1} is obtained, we can solve $\vec{p}\ k+1$ via the gradient descent method

$$\frac{\partial\vec{p}}{\partial t}=-\gamma\nabla H\left(\phi^{k+1}\right),\left|\vec{p}\right|\le 1 \tag{36}$$

By using semi-implicit difference scheme and the Karush-Kuhn-Tucker (KKT) conditions in [26], we can update $\vec{p}$, and get following fast iterative formula for this dual variable $\vec{p}\ k+1$

$$\vec{p}^{k+1}=\frac{\vec{p}^k-\tau\nabla H\left(\phi^{k+1}\right)}{1+\tau\left|\nabla H\left(\phi^{k+1}\right)\right|} \tag{37}$$

where $\tau\le 1/8$ is a time step as in [26].

Then, we can get a simple analytical form for auxiliary variable as the following:

$$\overrightarrow{\tilde{w}}^{k+1}=\nabla\phi^{k+1}+\vec{b}^{k+1} \tag{38}$$

Finally, we use projection formula of $\overrightarrow{\tilde{w}}^{k+1}$ as same as Equation 27 in order to satisfy the constraint $\left|\vec{w}\right|=1$.

3.4 Dual augmented Lagrangian projection method

The same idea in Subsection 3.3 can be extended to combine dual method and augmented Lagrangian projection method in Subsection 3.2, and this will lead to the dual augmented Lagrangian projection method. In detail, by introducing auxiliary variable $\vec{w}$ and putting the constraint $\vec{w}=\nabla\phi$, we can transform Equation 33 into following iterative minimization formulation:

$$\left(\phi^{k+1},\vec{p}^{k+1},\vec{w}^{k+1}\right)=\operatorname{Arg}\underset{\phi,\vec{w}}{\text{Min}}\ \underset{\vec{p}:\left|\vec{p}\right|\le 1}{\text{Sup}}\left\{\begin{array}{l}E\left(\phi,\vec{p},\vec{w}\right)=\int_\Omega Q_{12}(u_1,u_2)H_\varepsilon(\phi)dx+\gamma\int_\Omega H_\varepsilon(\phi)\nabla\cdot\vec{p}\ dx\\ \qquad+\int_\Omega\vec{\lambda}\cdot\left(\vec{w}-\nabla\phi\right)dx+\frac{\theta}{2}\int_\Omega\left(\vec{w}-\nabla\phi\right)^2dx\end{array}\right\} \tag{39}$$

The constraint $|\nabla\phi|=1$ can be also expressed as the constraint $\left|\vec{w}\right|=1$. By using the similar procedure, we can obtain the Euler-Lagrange equation of ϕ as the following;

$$\begin{cases}\left(Q_{12}(u_1,u_2)+\gamma\nabla\cdot\vec{p}^k\right)\delta_\varepsilon(\phi)+\nabla\cdot\lambda^k+\theta\nabla\cdot\left(\vec{w}^k-\nabla\phi\right)=0 & \text{in }\Omega\\ \left(\vec{\lambda}^k+\theta\left(\vec{w}^k-\nabla\phi\right)\right)\cdot\vec{n}=0 & \text{on }\partial\Omega\end{cases} \tag{40}$$

The $\vec{p}^{k+1}$ is updated as same as Equation 38, and $\overrightarrow{\tilde{w}}^{k+1}$ is the following analytical form

$$\overrightarrow{\tilde{w}}^{k+1}=\nabla\phi^{k+1}-\frac{\vec{\lambda}^k}{\theta} \tag{41}$$

Table 1 Abbreviations, full names, and their corresponding energy functionals of all methods for comparison

No.	Abbreviations	Full name	Energy functional
1	GDEWRM	Gradient descent equation with re-initialization [16]	$E(\phi)=\int_\Omega Q_{12}H(\phi)dx+\gamma\int_\Omega\lvert\nabla H(\phi)\rvert dx$ and $\begin{cases}\phi_t+\mathrm{Sign}(\phi_0)(\lvert\nabla\phi\rvert-1)=0\\ \phi(x,0)=\phi_0\end{cases}$
2	GDEWORM	Gradient descent equation without re-initialization [30]	$E(\phi)=\int_\Omega Q_{12}H_\varepsilon(\phi)dx+\gamma\int_\Omega\lvert\nabla H_\varepsilon(\phi)\rvert dx+\frac{\mu}{2}\int_\Omega(\lvert\nabla\phi\rvert-1)^2dx$
3	ALM	Augmented Lagrangian method [34]	$E(\phi,\lambda)=\int_\Omega Q_{12}H(\phi)dx+\gamma\int_\Omega\lvert\nabla H(\phi)\rvert dx+\int_\Omega\lambda(\lvert\nabla\phi\rvert-1)dx+\frac{\mu}{2}\int_\Omega(\lvert\nabla\phi\rvert-1)^2dx$
4	PLM	Projection Lagrangian method [34]	$E\left(\phi,\vec{w}\right)=\int_\Omega Q_{12}H(\phi)dx+\gamma\int_\Omega\lvert\nabla H(\phi)\rvert dx+\int_\Omega\lambda\left(\left\lvert\vec{w}\right\rvert-1\right)dx+\frac{\mu}{2}\int_\Omega\left(\vec{w}-\nabla\phi\right)^2dx$
5	CALM	Completely augmented Lagrangian method [36]	$E\left(\phi,\varphi,s,\vec{v},\vec{w}\right)=\int_\Omega Q_{12}sdx+\gamma\int_\Omega\left\lvert\vec{v}\right\rvert dx+\int_\Omega\lambda_2(s-H_\varepsilon(\varphi))dx$ $+\frac{\mu_2}{2}\int_\Omega(s-H_\varepsilon(\varphi))^2dx$ $+\int_\Omega\underset{3}{\vec{\lambda}}\cdot\left(\vec{v}-\nabla s\right)dx+\frac{\mu_3}{2}\int_\Omega\left(\vec{v}-\nabla s\right)^2dx\quad \text{s. t.}\quad\left\lvert\vec{w}\right\rvert=1$ $+\int_\Omega\lambda_1(\varphi-\phi)dx+\frac{\mu_1}{2}\int_\Omega(\varphi-\phi)^2dx$ $+\int_\Omega\underset{4}{\vec{\lambda}}\cdot\left(\vec{w}-\nabla\phi\right)dx+\frac{\mu_4}{2}\int_\Omega\left(\vec{w}-\nabla\phi\right)^2dx$
6	SBPM	Split Bregman projection method	$E\left(\phi,\vec{w}\right)=\int_\Omega Q_{12}H_\varepsilon(\phi)dx+\gamma\int_\Omega\left\lvert\vec{w}\right\rvert\delta_\varepsilon(\phi)dx+\frac{\theta}{2}\int_\Omega\left(w-\nabla\phi-\vec{b}^{k+1}\right)dx\ \text{s. t.}\ \left\lvert\vec{w}\right\rvert=1$
7	ALPM	Augmented Lagrangian projection method	$E\left(\phi,\vec{w},\vec{\lambda}\right)=\int_\Omega Q_{12}H_\varepsilon(\phi)dx+\gamma\int_\Omega\left\lvert\vec{w}\right\rvert\delta_\varepsilon(\phi)dx+\int_\Omega\vec{\lambda}\cdot\left(\vec{w}-\nabla\phi\right)dx+\frac{\theta}{2}\int_\Omega\left(\vec{w}-\nabla\phi\right)^2dx\ \text{s. t.}\ \left\lvert\vec{w}\right\rvert=1$
8	DSBPM	Dual Split Bregman projection method	$E\left(\phi,\vec{p},\vec{w}\right)=\int_\Omega Q_{12}H_\varepsilon(\phi)dx+\gamma\int_\Omega H_\varepsilon(\phi)\nabla\cdot\vec{p}\,dx+\frac{\theta}{2}\int_\Omega\left(\vec{w}-\nabla\phi-\vec{b}^{k+1}\right)^2dx\ \text{s. t.}\ \left\lvert\vec{w}\right\rvert=1$
9	DALPM	Dual augmented Lagrangian method	$E\left(\phi,\vec{p},\vec{w},\vec{\lambda}\right)=\int_\Omega Q_{12}H_\varepsilon(\phi)dx+\gamma\int_\Omega H_\varepsilon(\phi)\nabla\cdot\vec{p}\,dx+\int_\Omega\vec{\lambda}\cdot\left(\vec{w}-\nabla\phi\right)dx+\frac{\theta}{2}\int_\Omega\left(\vec{w}-\nabla\phi\right)^2dx\ \text{s. t.}\ \left\lvert\vec{w}\right\rvert=1$
10	FMM	Fuzzy membership method [38]	$E\left(\phi,\vec{w}\right)=\int_\Omega Q_{12}\phi dx+\gamma\int_\Omega\left\lvert\vec{w}\right\rvert dx+\frac{\theta}{2}\int_\Omega\left(w-\nabla\phi-\vec{b}^{k+1}\right)dx\qquad\text{s. t.}\ \ \phi\in[0,1]$

Then, we project $\vec{w}^{k+1}$ as in Equation 27. Finally, the Lagrangian multiplier $\vec{\lambda}$ can be updated as the following:

$$\vec{\lambda}^{k+1} = \vec{\lambda}^{k} + \theta\left(\vec{w}^{k+1} - \nabla\phi^{k+1}\right) \tag{42}$$

The advantages of the proposed four projection methods can be summarized as follows. (1): By introducing fewer auxiliary variables (i.e., two for SBPM, ALPM and three for DSBPM, DALPM) and considering the relationship between TV regularization term $\int_{\Omega}|\nabla H_{\varepsilon}(\phi)|dx$ or its equivalent form $\int_{\Omega}|\nabla\phi|\delta_{\varepsilon}(\phi)dx$ in Equation 22a and constraint term $|\nabla\phi| = 1$ in Equation 22b, we developed a very simple projection formula (27) in order to skillfully avoid expensive re-initialization process. (2): The proposed methods do not have many sub-minimization and sub-maximization problems and penalty parameters due to the fewer auxiliary variables, so it is very easy and efficient to implement. (3): The final Euler-equations of proposed fast projection algorithms only include a simple Euler-Lagrange equation (24, 29, 35, 40) that can be solved via fast Gauss-Seidel iteration, a generalized soft thresholding formula in analytical form (26, 32), a fast iterative formula for dual variable (37), and a very simple projection formula (27). This technique can elegantly avoid computing the complex curvature and thus improve the efficiency. (4): All the proposed methods can preserve full LSF as a SDF precisely without a very large penalty parameter. This is due to the introduced Bregman iterative parameters (23a, 34) and Lagrangian multipliers (28a, 39) and the projection computation, so a relatively large time step is allowed to be employed to speed up LSF evaluation as we will use semi-implicit gradient descent flow for (24, 29, 35, 40). (5): Even if the LSF is initialized as a piecewise constant function, it can be corrected automatically and precisely due to the projection computation. In conclusion, our proposed four projection methods will have both higher computational efficiency and better SDF fidelity, which can be validated in the next experimental Section.

4. Numerical experiments

In this section, we present some numerical experiments to compare the effectiveness and efficiency of our methods (i.e., SBPM, ALPM, DSBPM, and DALPM) with five previous ones (i.e., GDEWRM, GDEWORM, ALM, PLM, and CALM). In addition, we also compare the proposed four methods with the fast algorithm proposed in [38] for Chan-vese model under VLFM framework [20], which is named in this paper as fuzzy membership method (FMM). Therefore, there are totally ten algorithms involved in this paper. In order to make it easier to assess the exact differences between these models, we list the abbreviations of all methods, their full name, and corresponding energy functionals in Table 1.

In order to make the comparisons fair among different methods, we solve the PDEs in Equations 15, 17, 18, 19, 24, 29, 35, and 40 by semi-implicit difference scheme based on their gradient descent equations. As for FMM, we here adopt the method proposed in [38]. For CALM, we use the Gauss-Seidel fixed point iteration for solving

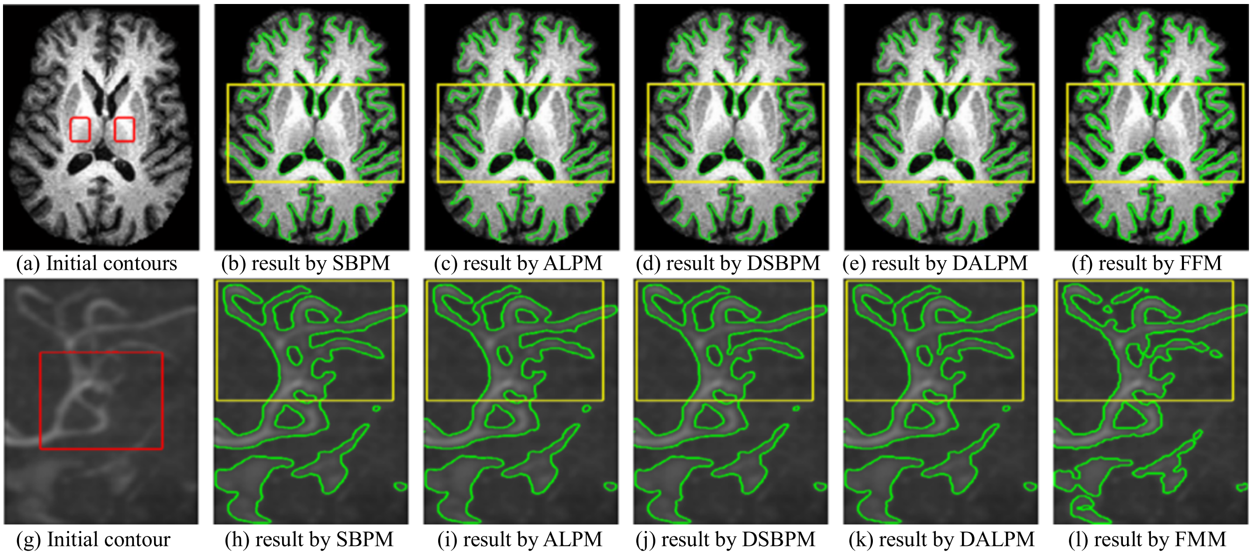


Figure 1 Comparison of our different projection methods with FMM. By their application to segment an MR image of brain and a CT image of vessel. **(a** and **g)** Original image with red initial contour. **(b-f** and **h-l)** The final segmentation results (i.e., green contours) by SBPM, ALPM, DSBPM, DALPM, and FMM, respectively.

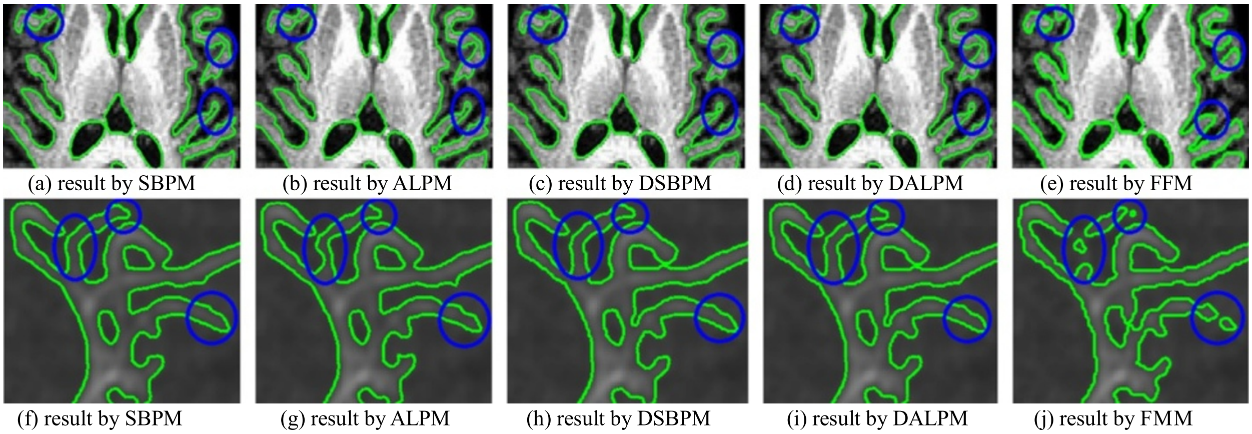


Figure 2 Zoom in small sub-regions of images in Figure 1 for detail comparisons. (a-j) The enlarged region of panels b to l of Figure 1, respectively.

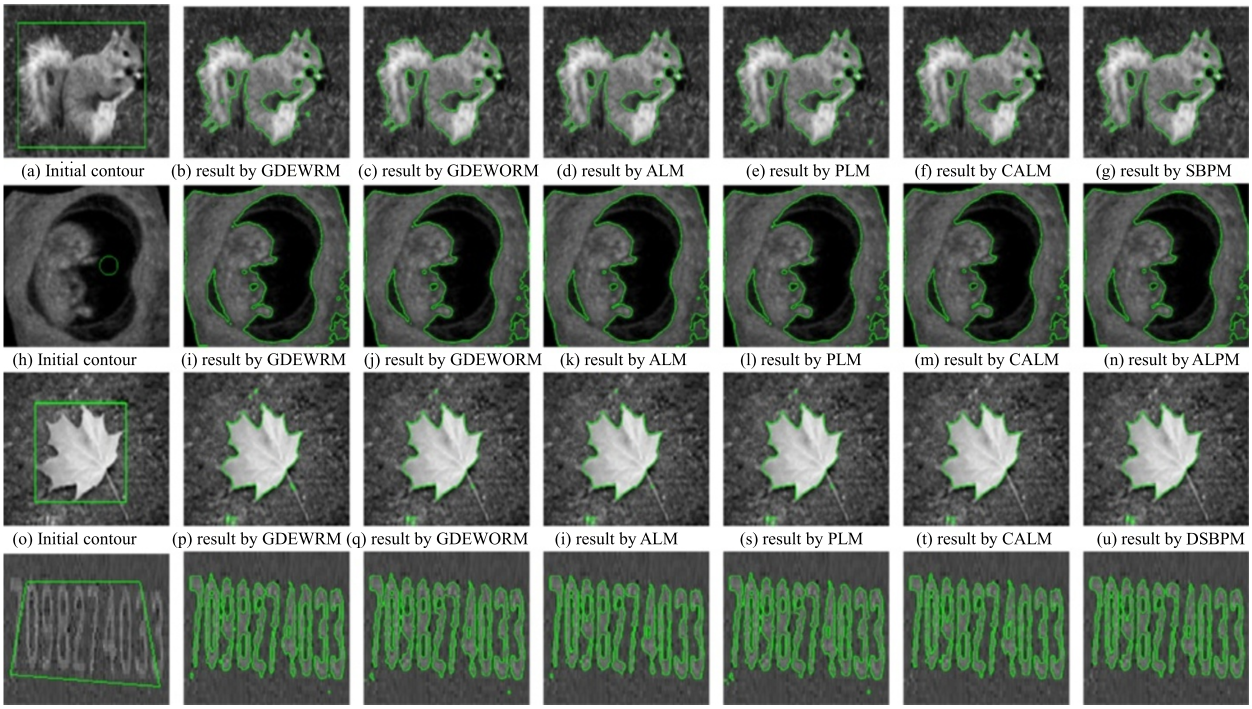
the LSF ϕ instead of fast Fourier transformation (FFT) for fair comparison with others. The initial LSF ϕ^0 is initialized as a same piecewise constant function for all the methods except initializing a SDF for GDEWRM. Equation 8 is solved by using the first order upwind scheme in every five iterations. In experiments 1 and 2, we set a one-step iteration for inside loop computation of ϕ for all the methods. However, ten-step iterations for ϕ in experiment 3 are required to achieve the final 3D SDFs fast. The parameter γ is usually formatted by $\gamma = \eta \times 255^2$, $\eta \in (0,1)$. We set the spatial step $h = 1$ and $\alpha_1 = \alpha_2 = 1$, $\tau = 0.125$, $\varepsilon = 3$. The stopping criterion is based on the relative energy error formula $|E^{k+1} - E^k|/E^k \leq \xi$, where ξ is a small prescribed tolerance and here we set 10^{-3} in all numerical experiments. All experiments are performed using Matlab 2010b on a Windows 7 platform with an Intel Core 2 Duo CPU at 2.33GHz and 2GB memory.

4.1 Experiment 1

In this experiment, we aim to compare the proposed four methods with the fast algorithm FMM. As FMM uses binary or label functions and continuous convex relaxation technique, it is very robust for initialization and fast and guaranteed to find a global minimizer. Our methods and FMM are applied to segment two medical images. One is MRI image of brain in the first row of Figure 1, and the other is CT image of vessel in the second row. The five methods are initialized with the same piecewise constant function (0 and 1). Here, we draw the red contours to represent their initial contours in the first column of Figure 1. Columns 2, 3, 4, 5, and 6 are the final segmentation results (i.e., green contours) by SBPM, ALPM, DSBPM, DALPM, and FMM, respectively. In order to make detailed comparisons, we crop a part of region indicated by the yellow rectangle in Figure 1 and enlarge them in Figure 2 where the first four columns are the results by the proposed four methods, respectively, and the last column is by FMM. One can observe from Figures 1 and 2 that the white matter in the brain and the vessel are extracted correctly and perfectly by the four methods. However, the results by FMM are less desirable. This can be clearly observed in column 5 of Figure 2, where some undesirable results of structure segmentation are marked with blue circles. However, we cannot tell easily some major differences among those segmentation results by all the four proposed methods. Further, fewer iterations and fast computational time shown in Table 2 demonstrate that the four methods are comparatively efficient as the fast FMM. In fact, the SBPM, ALPM, DSBPM, and DALPM are just different iterative schemes to solve the same system. The authors in [28] have proven their equivalence for the TV model. The segmentation results in Figure 1 and iterations and CPU times in Table 2 demonstrate consistency with their conclusion.

Table 2 Comparisons of iterations and computation time among our proposed fast methods

Image (size)	SBPM		ALPM		DSBPM		DALPM		FMM	
	Iterations	CPU time (s)	Iterations	CPU time (s)	Iterations	CPU time (s)	Iterations	CPU time (s)	Iterations	CPU time (s)
Brain (123 × 155)	19	0.327	19	0.331	20	0.308	20	0.325	22	0.326
Vessel (95 × 152)	23	0.313	22	0.306	22	0.309	22	0.303	24	0.329


(a) Initial contour (b) result by GDEWRM (c) result by GDEWORM (d) result by ALM (e) result by PLM (f) result by CALM (g) result by SBPM

(h) Initial contour (i) result by GDEWRM (j) result by GDEWORM (k) result by ALM (l) result by PLM (m) result by CALM (n) result by ALPM

(o) Initial contour (p) result by GDEWRM (q) result by GDEWORM (i) result by ALM (s) result by PLM (t) result by CALM (u) result by DSBPM

(v) Initial contour (w) result by GDEWRM (x) result by GDEWORM (y) result by ALM (z) result by PLM (I) result by CALM (II) result by DALPM

Figure 3 Comparisons of our different projection methods with previous five ones. To segment a squirrel image, an ultrasound baby image, a leaf image, and a synthetic noise number image. **(a, h, o, v)** Original image with initial green contour. **(b, i, p, w)** Results by GDEWRM. **(c, j, q, w)** Results by GDEWORM. **(d, k, i, y)** Results by ALM. **(e, l, s, z)** Results by PLM. **(f, m, t, I)** Results by CALM. **(g, n, u, II)** Results by our projection methods (from top to bottom is SBPM, ALPM, DSBPM, and DALPM, respectively).

4.2 Experiment 2

In this experiment, we will compare the efficiency of our methods with that of GDEWRM, GDEWORM, ALM, PLM, and CALM. All nine methods are run on four real and synthetic images including squirrel, ultrasound baby, leaf, and synthetic noise number images, respectively. In the first column of Figure 3, we initialize piecewise constant function (0 and 1) for all methods except GDEWRM, which is initialized with a SDF. Columns 2, 3, 4, 5, and 6 of Figure 3 are the results by GDEWRM, GDEWORM, ALM, PLM, and CALM respectively. In the last column of Figure 3, we only present the final segmentation result of squirrel, ultrasound baby, leaf, and number image by SBPM, ALPM, DSBPM, and DALPM respectively, because the visual effect and computational efficiency for all the four proposed methods

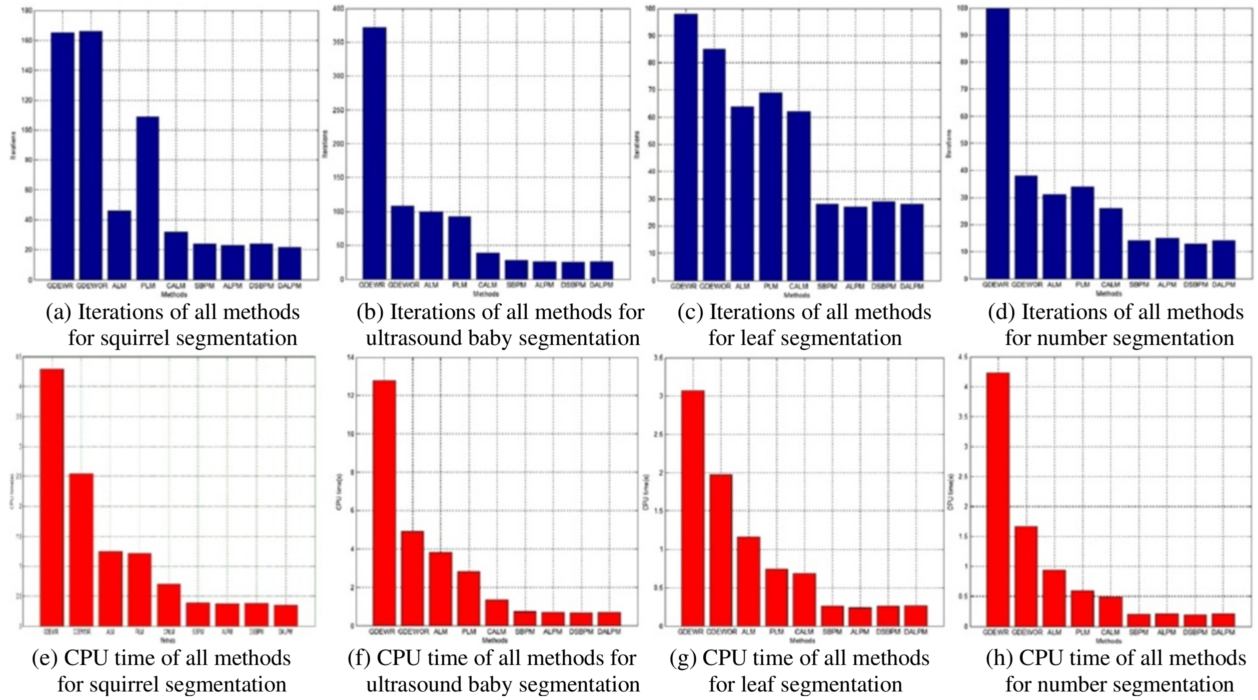
Table 3 Comparison of iterations and computation time using different segmentation methods

Methods	Iterations				CPU time (s)			
	Squirrel	Ultrasound baby	Leaf	Numbers	Squirrel	Ultrasound baby	Leaf	Number
	(155 × 122)	(180 × 175)	(128 × 87)	(128 × 127)	(155 × 112)	(180 × 175)	(128 × 87)	(128 × 127)
GDEWRM	165	372	98	100	4.296	12.769	3.070	4.230
GDEWORM	166	108	85	38	2.543	4.905	1.978	1.666
ALM	46	100	64	31	1.243	3.803	1.163	0.945
PLM	109	92	69	34	1.211	2.813	0.738	0.596
CALM	32	39	62	26	0.696	1.365	0.685	0.498
SBPM	24	28	28	14	0.389	0.723	0.256	0.202
ALPM	23	26	27	15	0.372	0.707	0.237	0.212
DSBPM	24	25	29	13	0.381	0.692	0.258	0.192
DALPM	22	26	28	14	0.351	0.701	0.262	0.208

are very similar on these images. From Figure 3, we can see that all the methods do a relatively good performance for segmenting both real and noise synthetic images. However, compared with other methods, all the four proposed methods perform better, which can be observed in the last column of Figure 3. In addition, we record total iterations and computation time of all nine methods for segmenting these images in Table 3. In order to make the experimental data in Table 3 meaningful, we draw Figure 4 to illustrate the differences regarding iterations and computation time. Figure 4a,b,c,d shows the total iterations with bar chart of all nine methods for segmenting squirrel, ultrasound baby, leaf, and number images, respectively, and Figure 4e,f,g,h draws the total CPU time for segmenting these images. According to Figure 4e,f,g,h, the computational time of the nine methods can be clearly ranked in the following order: SBPM ≈ ALPM ≈ DSBPM ≈ DALPM < CALM < PLM < ALM < GDEWORM < GDEWRM. The reason leading to this rank can be justified as follows. (1) All the methods compute faster than the GDEWRM due to its expensive re-initialization process. (2) Among these methods without re-initialization, ALM, PLM, and CALM are running faster than GDEWORM. For GDEWORM, the CFL condition limits its time step so that it cannot be fast, while ALM improves convergence rate by introducing Lagrange multiplier λ. PLM uses Lagrangian method and variable splitting technique to enhance the evolution speed, so PLM is faster than ALM. However, both ALM and PLM are limited by CFL condition and their speed is slowed. CALM introduces many scalar or vector auxiliary variables and Lagrangian multipliers to make each subproblem very simple as well as can avoid CFL condition, so it computes faster than ALM and PLM. (3) All the proposed methods can achieve the best efficiency and satisfactory segmentation results because the nonlinear curvature is replaced by the linear Laplace operator in Equations 24 and 29 or the dual divergence operator in Equations 35 and 40 as simple projection technique (27)

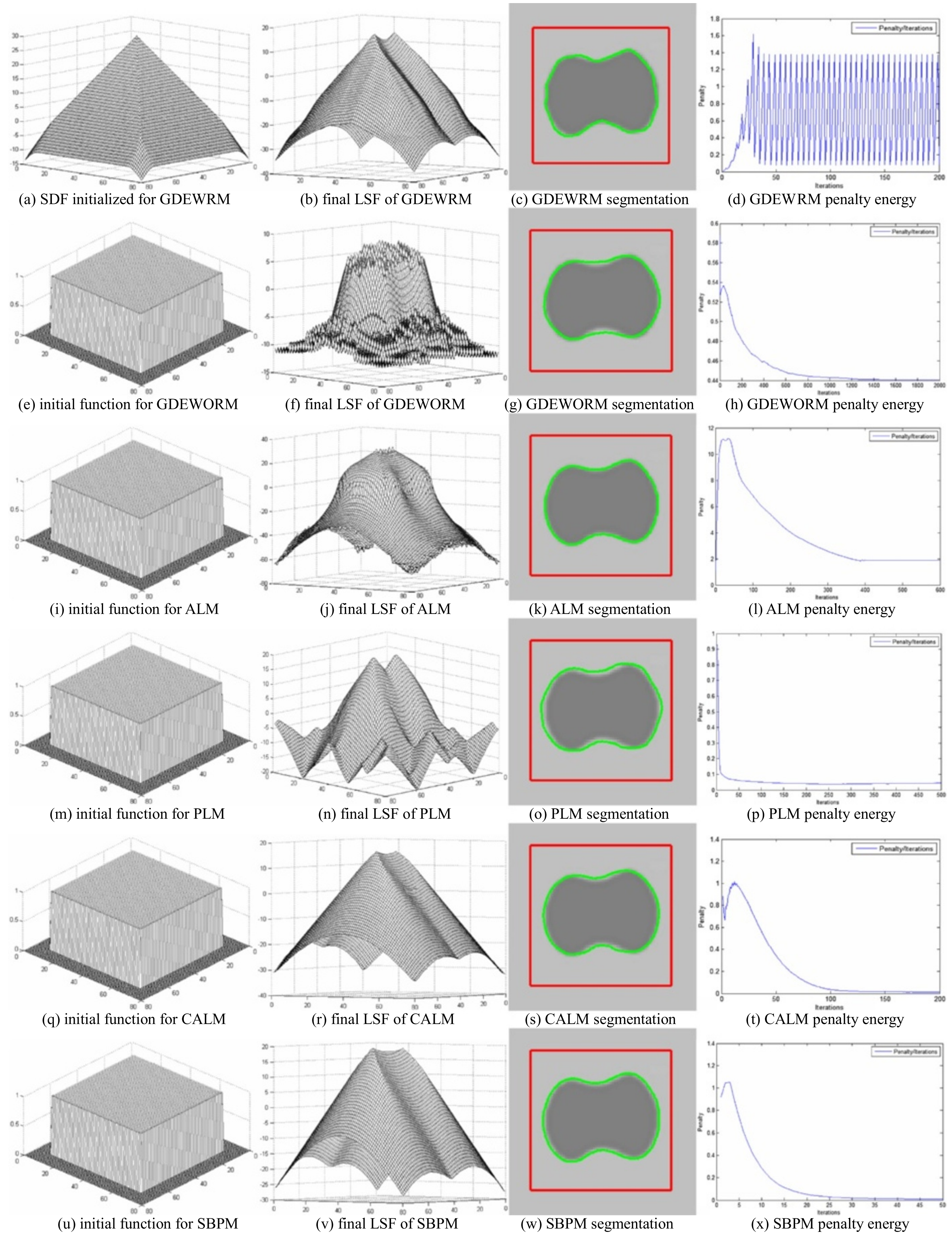
Table 4 Comparison of iterations, computation time, and SDF fidelity using different segmentation methods

Methods	Iterations	CUP time (s)	SDF fidelity value
GDEWRM	200	77.236	0.0385
GDEWORM	2000	224.094	0.4407
ALM	600	69.624	1.9005
PLM	500	46.644	0.0441
CALM	200	18.545	0.0259
SBPM	50	5.296	0.0149
ALPM	50	5.197	0.0149
DSBPM	50	4.924	0.0153
DALPM	50	5.213	0.0153



(a) Iterations of all methods for squirrel segmentation
(b) Iterations of all methods for ultrasound baby segmentation
(c) Iterations of all methods for leaf segmentation
(d) Iterations of all methods for number segmentation
(e) CPU time of all methods for squirrel segmentation
(f) CPU time of all methods for ultrasound baby segmentation
(g) CPU time of all methods for leaf segmentation
(h) CPU time of all methods for number segmentation

Figure 4 Graph Expression of iterations and CPU time of all the methods in Table 3. (a), **(b)**, **(c),** and **(d)** Iterations of all methods for segmenting squirrel, ultrasound baby, leaf and synthetic noise number images respectively. **(e)**, **(f)**, **(g),** and **(h)** Record of their CPU time.



(a) SDF initialized for GDEWRM (b) final LSF of GDEWRM (c) GDEWRM segmentation (d) GDEWRM penalty energy

(e) initial function for GDEWORM (f) final LSF of GDEWORM (g) GDEWORM segmentation (h) GDEWORM penalty energy

(i) initial function for ALM (j) final LSF of ALM (k) ALM segmentation (l) ALM penalty energy

(m) initial function for PLM (n) final LSF of PLM (o) PLM segmentation (p) PLM penalty energy

(q) initial function for CALM (r) final LSF of CALM (s) CALM segmentation (t) CALM penalty energy

(u) initial function for SBPM (v) final LSF of SBPM (w) SBPM segmentation (x) SBPM penalty energy

Figure 5 (See legend on next page.)

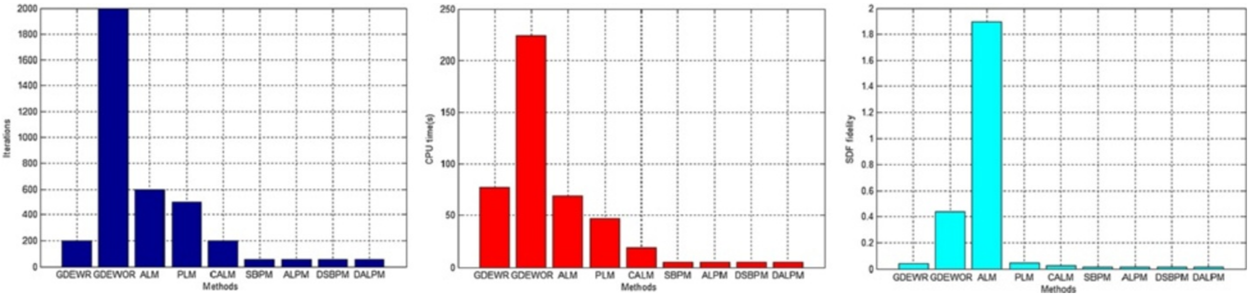
(See figure on previous page.)
Figure 5 SDF fidelity comparisons of our projection methods with previous methods. (a, e, i, m, q, u) Initial LSFs. **(b, f, j, n, r, v)** Final LSFs. **(c, g, k, o, s, w)** Initial contours marked by red rectangle and final segmentation results indicated by green contours. **(d, h, l, p, t, x)** Mean value of penalty energy plots $\int_{\Omega}(|\nabla\phi|-1)^2dx$ (closeness measure between LSF and SDF).

is used. In comparison with CALM, our projection methods have fewer sub-problems, so it is very efficient. In addition, by introducing the Bregman iterative parameters (23a, 34) and Lagrangian multipliers (28a, 39), a relatively large time step can be used to speed up LSF evaluation. Therefore, our methods compute faster than CALM and their efficiency ranks first. (4) The proposed four fast methods (i.e., SBPM, ALPM, DSBPM, and DALPM) are actually equivalent, which is validated in [25]. Therefore, these projection methods have very similar computation speed.

4.3 Experiment 3

In this experiment, we aim to compare SDF fidelity produced by our four methods and the other five. We segment a synthetic image (100×100) to obtain the SDF fidelity value in Table 4 as explained below. The first column of Figure 5 is the initial LSFs for all the methods. As there is no constraint of LSF as a SDF in GDEWRM, panel a is initialized as a SDF for this method. However, if it is initialized as a piecewise constant function, the LSF will be far away from SDF during the contour evaluation, even though the re-initialization process may be not able to pull LSF back to SDF. In this case, the comparisons of SDF preservation with other methods without re-initialization are not very fair. Based on the above observation, Figure 5e,i,m,q,u is initialized as the same piecewise constant function for GDEWORM, ALM, PLM, CALM, and SBPM, respectively. As all of our four projection methods achieve almost the same results, here, we only give the experimental data for SBPM in the last row of Figure 5. In the second column of Figure 5b,f,j,n,r,v are the final 3D LSFs of GDEWRM, GDEWORM, ALM, PLM, CALM, and SBPM, respectively. In the third column of Figure 5c,g,k,o,s,w are the same initial contours marked by red rectangle and the final segmentation results marked by green contours of above methods. In the last column, we draw the plots of mean value of penalty energy $\int_{\Omega}(|\nabla\phi|-1)^2dx$ by the above corresponding methods, which is used to measure the closeness between LSF and SDF. We denote SDF fidelity value as mean value in the last iteration for every method. The smaller this value is, the closer the LSF and SDF will be. We also put all these results in Table 4 for easy comparison.

Note that the final LSF of GDEWRM in Figure 5b is very close to SDF, which is validated by its very small SDF fidelity value (0.0385) in Table 4. However, the final green segmentation contour in Figure 5c shows that the zero level set of Figure 5b by this GDEWRM shrinks and cannot reach the exact location of the object. In fact, in order to obtain Figure 5b, this GDEWRM needs 300 re-initialization iterations after every five-step iteration of LSF evolution. So it is very expensive (total 77.236 s reported in Table 4), and this re-initialization leads to large jumps in its penalty energy plots in Figure 5d. For GDEWORM and PLM, their experimental results are displayed in the second and fourth row, respectively. Although their final SDF fidelity values are close to zero (i.e., 0.4407 for GDEWORM and 0.0441 for PLM), their final SDFs as shown in Figure 5f,n are not preserved nicely. Moreover, due to CFL condition, we need to choose a very small time step 10^{-4} and a very large penalty parameter 2×10^4 for GDEWORM and



(a) Iterations of all methods for a synthetic image segmentation in Fig 5

(b) CPU time of all methods for a synthetic image segmentation in Fig 5

(c) SDF fidelity value of all methods for a synthetic image segmentation in Fig 5

Figure 6 Graph expression of iterations and CPU time and SDF fidelity of all the methods in Table 4. (a), (b), and **(c)** Record iterations, CPU time, and SDF fidelity value of all methods, respectively.

PLM, respectively. This selection is aimed to guarantee the closeness between the LSF and SDF and stability of LSF evolution, but this leads to a large number of total iterations (i.e., respective 2,000 and 500). As we analyzed in experiment 2, PLM adopts variable splitting technique and therefore its total computation time and iterations are much less than GDEWORM. As large penalty parameters are employed in PLM and GDEWORM, we find that the final green contours by these two methods cannot segment the object precisely. For ALM, it improves convergence rate by introducing Lagrange multiplier λ so that we can choose a slightly larger time step 10^{-2} and a relatively smaller penalty parameter 10^{-1} to evolve the LSF. However, we conducted a great number of experiments for this method and note that it is very sensitive to parameters selection. Also, its final SDF fidelity value is always the largest among all the methods. This may be due to the fact that the introduced Lagrangian multiplier in ALM breaks the CFL condition. The fourth row of Figure 5 that demonstrates CALM is able to achieve a better 3D SDF (shown in Figure 5f) and a smaller SDF fidelity value (0.0259 shown in Table 4) than those by other methods except our methods. The computation speed of this method has been improved to a great extent as observed in Table 4 and Figure 5t. The last row of Figure 5 presents the experimental data by our proposed SBPM. Here, we emphasize that the other three proposed methods can achieve almost that same SDF and efficiency as SBPM. From Figure 5v, the final 3D LSF is perfectly preserved as SDF fidelity value is only 0.0149, the smallest one among all the methods in Table 4. Most impressively, we find that penalty parameter 10 can be large enough to penalize accurately the full LSF as SDF due to the introduced Lagrangian multiplier, Bregman iterative parameter, and the precise projection computation. In this case, a relatively large time 10^{-2} can be employed to speed up LSF evolution as shown in Figure 5x. In addition, even if the LSF is initialized as a piecewise constant function for SBPM, it can be corrected automatically and precisely due to the projection formula (27).

Lastly, we present Figure 6 that includes three bar graphs which correspond to iterations, CUP time, and SDF fidelity value in Table 4, respectively. However, Figure 6b shows that the slowest method is GDEWORM rather than GDEWRM, which is inconsistent with the conclusion in experiment 2. In fact, the time step in GDEWRM is set 10^{-2}, 100 times larger than that set for GDEWORM. We find that this time step together with 300 re-initialization iterations would not broke the stability of LSF evolution and simultaneously is able to achieve a very desirable SDF. In contrast, for the purpose of preserving distance feature, we should choose a very large penalty parameter for GDEWORM, which limits the speed of LSF evolution. Therefore, in this experiment, on the premise of preserving distance feature, the GDEWRM is faster than GDEWORM. From Figure 6c, the ability of SDF fidelity can be ranked as SBPM ≈ ALPM ≈ DSBPM ≈ DALPM > CALM > GDEWORM > PLM > GDEWRM > ALM. In conclusion, this experiment validates that the four projection methods perform excellently in both accuracy and speed of preserving SDF.

5. Conclusions

In this paper, by investigating the relationship between the L^1-based TV regularizer term of Chan-Vese model and the constraint on LSF and introducing some auxiliary variables, we design fast split Bregman projection method (SBPM), augmented Lagrangian projection method (ALPM), dual split Bregman projection method (DSBPM), and dual augmented Lagrangian projection method (DALPM). All these methods can skillfully avoid the expensive re-initialization process and simplify computation of curvatures. In our methods, there are fewer subproblems and penalty parameters, so they can be solved efficiently. Moreover, the full LSF can be preserved as a SDF precisely without a very large penalty parameter so that a relatively large time step can be used to speed up LSF evaluation. In addition, even if the LSF is initialized as a piecewise constant function, it can be corrected automatically and accurately due to analytical projection computation. Simulation experiments have validated the efficiency and performance of proposed methods in terms of computational cost and SDF fidelity.

Abbreviations

ALM: augmented Lagrangian method; ALPM: augmented Lagrangian projection method; CALM: completely augmented Lagrangian method; CFL: Courant-Friedrichs-Lewy; DALPM: dual augmented Lagrangian method; DSBPM: dual split Bregman projection method; FFT: fast Fourier transformation; FMM: fuzzy membership method; GDEWORM: gradient descent equation without re-initialization; GDEWRM: gradient descent equation with re-initialization; KKT: Karush-Kuhn-Tucker; LSF: level set function; LSM: level set method; PDEs: partial differential equations; PLM: projection Lagrangian method; SBPM: split Bregman projection method; SDF: sign distance function; TV: total variation; VLFM: variational label function method; VLSM: variational level set method.

Competing interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos.61305045, 61170106, and 61303079), National 'Twelfth Five-Year' Development Plan of Science and Technology (no.2013BAI01B03), and Qingdao Science and Technology Development Project (no. 13-1-4-190-jch).

Received: 10 April 2013 Accepted: 7 January 2014
Published: 27 January 2014

References

1. JM Morel, S Solimini, *Variational Methods in Image Segmentation* (Birkhauser, Boston, 1994)
2. TF Chan, M Moelich, B Sandberg, Some recent developments in variational image segmentation, in *Image Processing Based on Partial Differential Equations*, ed. by XC Tai, KA Lie, TF Chan, S Osher (Springer, Heidelberg, 2006), pp. 175–210

3. S Osher, N Paragios, *Geometric Level Set Methods in Imaging, Vision, and Graphics* (Springer, Heidelberg, 2003)
4. A Mitiche, IB Ayed, *Variational and Level Set Methods in Image Segmentation* (Springer, Heidelberg, 2010)
5. M Kass, A Witkin, D Terzopoulos, Snakes: active contour models. Int. J. Comput. Vis. **4**(1), 321–331 (1987)
6. D Mumford, J Shah, Optimal approximations by piecewise smooth functions and associated variational problems. Commun. Pure Appl. Math. **42**(5), 577–685 (1989)
7. G Aubert, M Barlaud, O Faugeras, S Jehan-Besson, Image segmentation using active contours: calculus of variations or shape gradient. SIAM J. Appl. Math. **63**(6), 2128–2154 (2003)
8. G Aubert, M Barlaud, S Duffner, A Herbulot, S Jehan-Besson, Segmentation of vectorial image features using shape gradients and information measures. J. Math. Imaging Vis. **25**(3), 365–386 (2006)
9. G Aubert, M Barlaud, E Debreuve, M Gastaud, Using the shape gradient for active contour segmentation: from the continuous to the discrete formulation. J. Math. Imaging Vis. **28**(1), 47–66 (2007)
10. S Osher, JA Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. **79**(1), 12–49 (1988)
11. L Ambrosio, VM Tortorelli, Approximation of functionals depending on jumps by elliptic functionals via gamma-convergence. Commun. Pur. Appl. Math. **43**(8), 999–1036 (1990)
12. S Esedoglu, YH Tsai, Threshold dynamics for the piecewise constant Mumford-Shah functional. J. Comput. Phys. **211**(1), 367–384 (2006)
13. YM Jung, SH Kang, JH Shen, Multiphase image segmentation by Modica-Mortola phase transition. SIAM J. Appl. Math. **67**(5), 1213–1232 (2007)
14. I Posirca, YM Chen, CZ Barcelos, A new stochastic variational PDE model for soft Mumford-Shah segmentation. J. Math. Anal. Appl. **384**(1), 104–114 (2011)
15. HK Zhao, TF Chan, B Merriman, S Osher, A variational level set approach to multiphase motion. J. Comput. Phys. **127**(1), 179–195 (1996)
16. TF Chan, LA Vese, Active contours without edges. IEEE Trans. image process. **10**(2), 266–277 (2001)
17. LA Vese, TF Chan, A multiphase level set framework for image segmentation using the Mumford and Shah model. Int. J. Comput. Vis. **50**(3), 271–293 (2002)
18. C Samson, L Blanc-Feraud, G Aubert, A level set model for image classification. Int. J. Comput. Vis. **40**(3), 187–197 (2000)
19. G Chung, LA Vese, Energy minimization based segmentation and denoising using a multilayer level set approach. LNCS, Springer-Verlang **3757**, 439–455 (2005)
20. X Bresson, S Esedoglu, P Vandergheynst, JP Thiran, S Osher, Fast global minimization of the active contour/snake model. J. Math. Imaging Vis. **28**(2), 151–167 (2007)
21. J Lie, M Lysaker, XC Tai, A binary level set model and some applications to Mumford-Shah image segmentation. IEEE Trans. Image process. **15**(5), 1171–1181 (2006)
22. J Lie, M Lysaker, XC Tai, A variant of the level set method and applications to image segmentation. Math. Comput. **75**(255), 1155–1174 (2006)
23. F Li, K Michael, T Zeng, C Shen, A multiphase image segmentation method based on fuzzy region competition. SIAM J. Imaging Sci. **3**(3), 277–299 (2010)
24. T Goldstein, S Osher, The split Bregman method for L1 regularized problems. SIAM J. Imaging Sci. **2**(2), 323–343 (2009)
25. T Goldstein, X Bresson, S Osher, Geometric applications of the split Bregman method: segmentation and surface reconstruction. J. Sci. Comput. **45**(1–3), 272–293 (2009)
26. A Chambolle, An algorithm for total variation minimization and applications. J. Math. Imaging Vis. **20**(1), 89–97 (2004)
27. ES Brown, TF Chan, X Bresson, Completely convex formulation of the Chan-Vese image segmentation model. Int. J. Comput. Vis. **98**(1), 103–121 (2012)
28. C Wu, XC Tai, Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models. SIAM J. Imaging Sci. **3**(3), 300–339 (2010)
29. YHR Tsai, LT Cheng, S Osher, HK Zhao, Fast sweeping algorithms for a class of Hamilton-Jacobi equations. SIAM J. Numer. Anal. **41**(2), 673–694 (2003)
30. C Li, C Xu, C Gui, MD Fox, Level set evolution without re-initialization: a new variational formulation, in *Proceedings of IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 1st edn. San Diego **20–25**, 430–436 (2005)
31. D Adalsteinsson, JA Sethian, The fast construction of extension velocities in level set methods. J. Comput. Phys. **148**(1), 2–22 (1999)
32. D Peng, B Merriman, S Osher, HK Zhao, M Kang, A PDE-based fast local level set method. J. Comput. Phys. **155**(2), 410–438 (1999)
33. M Sussman, E Fatemi, An efficient, interface preserving level set re-distancing algorithm and its application to interfacial incompressible fluid flow. SIAM J. Sci. Comput. **20**(4), 1165–1191 (1999)
34. C Liu, F Dong, S Zhu, D Kong, K Liu, New variational formulations for level set evolution without reinitialization with applications to image segmentation. J. Math. Imaging Vis **41**(3), 194–209 (2011)
35. R Courant, K Friedrichs, H Lewy, On the partial difference equations of mathematical physics. IBM J **11**(2), 215–234 (1967)
36. V Estellers, D Zosso, R Lai, S Osher, JP Thiran, X Bresson, An efficient algorithm for level set method preserving distance function. IEEE Trans. Image Process. **21**(12), 4722–4734 (2012)
37. C Liu, Z Pan, J Duan, New algorithm for level set evolution without re-initialization and its application to variational image segmentation. J Software **8**(9), 2305–2312 (2013)
38. X Bresson, A Short Guide on a Fast Global Minimization Algorithm for Active Contour Models (Online). https://googledrive.com/host/0B3BTLeCYLunCc1o4YzV1Ui1SeVE/codes_files/xbresson_2009_short_guide_global_active_contours.pdf. Accessed 22 Apr 2009

doi:10.1186/1687-5281-2014-7
Cite this article as: Duan *et al.*: Some fast projection methods based on Chan-Vese model for image segmentation. *EURASIP Journal on Image and Video Processing* 2014 **2014**:7.