Two Fast Alternating Direction Optimization Methods for
Multiphase Segmentation

Lu Tan 1, Weibo Wei 2, and Zhenkuan Pan 2

1,2 College of Information Engineering, Qingdao University, Qingdao, Shandong, China
luna086@163.com, njjustwwb@163.com

Abstract - In this paper, two new methods associated with alternating direction optimization, fast alternating direction method of multipliers(FlastADMM) and fast alternating minimization algorithm(FastAMA), are proposed for image segmentation using the multiphase Chan-Vese model, which is on the basis of piecewise constant optimal approximations. For these methods, we incorporate the variable splitting approach and a 'reset' condition in order to update the Lagrange multiplier and make sure the value of energy functional is always positive. The Osher and Stethian level set method, binary level set functions, thresholding method and projection formula are applied in the implementation. Finally, numerical results with rapid convergence are obtained by our methods, which are also compared with those of some other fast variational methods to demonstrate better effectiveness of our methods.

Keywords: Multiphase segmentation, fast alternating direction method of multipliers, fast alternating minimization algorithm, active contours, level sets.

1 Introduction

In image segmentation, major advances were made in two-phase image segmentation[1-3] in the early days. Mumford-Shah model proposed by Mumford D and Shah J [4] is regarded as the most significant region-based model. It has been extended to a great deal of applications. In 2001, Tony F. Chan and Luminita A. Vese proposed the Chan-Vese model [5] for active contours to detect objects in a given image. It is one of the simplified variants of Mumford-Shah model. Nevertheless, as the complexity of the images increases, 2-phase image segmentation is not able to meet the actual needs. Therefore, multiphase segmentation is applied to satisfy the demands. On the basis of Potts model [6][7] from statistical mechanics, Zhao et al. [8] started to study multiphase motion segmentation by using the level set method and proposed a model which can represent n different regions by n level set functions. In order to reduce the number of level set functions, Chan et al. continued their work and proposed multiphase segmentation model [9] which is a generalization of Chan-Vese model. Their scheme can naturally avoid "overlap" and "leakage" problem. But there are still some problems about solving the global optimization, accuracy, stability and speed.

Alternating direction method of multipliers(ADMM) was first described by Glowinski and Marocco [10] and alternating minimization algorithm(AMA) was presented by Tseng [11]. These techniques are commonly known as the Split Bregman Method [12], and are known to be an efficient solver for problems involving the total-variation norm [13]. These methods can be accelerated using optimal first order methods, of the type first proposed by Nesterov [14]. And accelerated variants of ADMM and AMA can be called FastADMM and FastAMA.

In our paper, we design methods (FastADMM and FastAMA) that can achieve computational efficiency and own even faster convergence to solve the functional of multiphase Chan-Vese model based on binary level sets framework [15,16]. The gradient descent method (GDM) [17], Chambolle’s dual method (DM) [18], alternating direction method of multipliers(ADMM) [19] i.e. the augmented Lagrangian method (ALM) [20], and alternating minimization algorithm(AMA) [11] are used to compare with our methods. But results of these methods which are used in multiphase segmentation model will be obtained in a slow convergence. Tom Goldstein et al. [19] introduced FastADMM and applied it to solve the TV model as an example and some other strongly convex problems. A predictor-corrector type acceleration step is used in this method.

The remaining of this paper is organized as follows. In Section 2, the binary level set formulation of the functional of multiphase Chan-Vese model used in our paper is reviewed along with its four traditional solution methods. Our proposed methods are discussed in Section 3 and its iterative discrete formulas for implementation will be presented in detail. In Section 4, some numerical experiments are given to illustrate the effectiveness of our method by comparing with other methods. Finally a conclusion is given in section 5.

2 The multiphase Chan-Vese model and its four traditional methods

2.1 The binary level set based formulation

In order to separate an image domain Ω into n subdomains with \( \Omega = \bigcup_{j=1}^{n} \Omega_i \) and \( \Omega_i \cap \Omega_j = \emptyset \). Vese and Chan defined up to \( n = 2^m \) phases and \( m \) level set functions. This way makes sure that each pixel \((x, y) \in \Omega\) will belong to one, and only one phase.
The level set method proposed by Osher and Sethian [21] is an effective representation for evolving curves and surfaces because of automatic change of topology. The main idea of the level set formulation is to implicitly represent a given interface $\Gamma(t)$ as the zero level set of a Lipschitz continuous function $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$). $\phi$ is defined as follows:

\[
\phi(x,t) > 0, \text{ if } x \text{ is inside } \Gamma(t) \\
\phi(x,t) = 0, \text{ if } x \text{ is at } \Gamma(t) \\
\phi(x,t) < 0, \text{ if } x \text{ is outside } \Gamma(t)
\]  

It is normal to define $\phi$ as a signed distance function in order to keep stability in numerical implementation. The distance function $\phi$ obeys the Eikonal equation,

\[ |\nabla \phi(x,t)| = 1 \]  

(2)

The variational level set method [13] gives a way to apply the level set formulation to the energy functional. For a given open region $\Omega$ with smooth boundary $\Gamma(t)$, simple facts can be got as follows:

\[
\text{length}(\Gamma) = \int_{\Gamma} |\nabla H(\phi)|dx = \int_{\Omega} \delta(\phi)|\nabla \phi|dx \\
\text{area}(\Omega) = \int_{\Omega} H(\phi)dx
\]  

(3)

where $H(x)$ and $\delta(x)$ are Heaviside function and Dirac delta function respectively. According to their work, for $i = 1,2,\ldots,n$, let $\left(b_i^1,b_i^2,\ldots,b_i^m\right)$ be the binary representation of $i-1$, where $b_i^k = 0 \vee 1$. The characteristic function $\chi_i(x)$ of $\Omega_i$ can be written as the following general expression:

\[
\chi_i(x) = \prod_{j=1}^{m} \left[b_i^j + (-1)^{b_i^{j+1}} H(\phi_j)\right].
\]  

(5)

Then energy functional for $n$ phases is obtained:

\[
E(\phi) = \sum_{j=1}^{m} \gamma \int_{\Omega} |\nabla H(\phi_j)|dx + \sum_{i=1}^{n} \alpha_i \int_{\Omega} Q_i \chi_i dx,
\]  

(6)

where $\gamma$ and $\left(\alpha_1,\alpha_2,\ldots,\alpha_n\right)$ is positive parameters. The function $Q_i$ is defined as $\left(c_i - f\right)^2$, $c_i$ is a constant vector which can be obtained by the mean intensity value of $f$ inside $\Omega_i$ as follows:

\[
c_i = \frac{\int_{\Omega} f \chi_i dx}{\int_{\Omega} \chi_i dx}.
\]  

(7)

On this basis, a new approach called a binary level set function is introduced by Johan Lie et al. [15] and Bresson et al. [16], which has a simpler definition about initialization of level set function. Firstly, assume that the interface is enclosing $\Omega_i \subset \Omega$. A discontinuous level set function $\phi$ is used instead. It is defined as follows:

\[
\phi(x) = \begin{cases} 
1, & \text{if } x \in \text{int}(\Omega_i) \\
0, & \text{if } x \in \text{ext}(\Omega_i)
\end{cases}
\]  

(8)

If the level set function $\phi(x)$ satisfy $\phi(x)^2 = 1$, then we can use the basis function $\phi(x)$ to calculate the length of the boundary of $\Omega_i$, and the area inside $\Omega_i$.

\[
\text{length}(\partial \Omega_i) = \int_{\Omega} |\nabla \phi(x)|dx
\]  

(9)

\[
\text{area}(\Omega_i) = \int_{\Omega} \phi(x)dx
\]  

(10)

Wang Qi et al. [22] proposed a multiphase Chan-Vese model based on a plurality of binary level set functions and alternating convex optimization in 2010. They rewrite the multiphase Chan-Vese model based on binary level set as:

\[
E(\phi) = \sum_{j=1}^{m} \gamma \int_{\Omega} |\nabla \phi_j|dx + \sum_{i=1}^{n} \alpha_i \int_{\Omega} Q_i \chi_i dx,
\]  

(11)

where $\phi_j$ is the binary level set function and the characteristic function $\chi_i(x)$ should be restated as follows:

\[
\chi_i(x) = \prod_{j=1}^{m} \left[b_i^j + (-1)^{b_i^{j+1}} \phi_j\right].
\]  

(12)

### 2.2 GDM, DM, ADMM or ALM and AMA for minimizing multiphase Chan-Vese model

#### 2.2.1 Gradient descent method (GDM)

The energy functional minimization problem associated with Equation (11) can be solved by computing the evolution equation of $\phi$ via gradient descent flow as:

\[
\begin{align*}
\frac{\partial \phi_j}{\partial t} &= \gamma \nabla \cdot \left(\nabla \phi_j\right) - \sum_{i=1}^{n} \alpha_i Q_i \frac{\partial \chi_i}{\partial \phi_j} & \text{in } \Omega \\
\frac{\partial \phi_j}{\partial n_j} &= 0 & \text{on } \partial \Omega
\end{align*}
\]  

(13)

Where the second formula of (13) is the boundary condition. But (13) includes fourth order derivatives need to be discretized using complex finite difference formulas and the integration steps of time marching depend on right hand terms heavily.

#### 2.2.2 Dual method (DM)

In order to speed up the calculation, Dual formula of TV norm $\int_{\Omega} |\nabla \phi|dx = \sup_{\|p\|} \int_{\Omega} \phi \nabla \cdot \hat{p}dx$ proposed by Chambolle [18] can be applied in the energy functional.
\[ E(\phi, \bar{p}) = \sum_{j=1}^{m} \int_{\Omega} \gamma \phi_j \nabla \cdot \bar{p}_j dx + \sum_{i=1}^{n} \alpha_i \int_{\Omega} Q_j \chi_i dx \]  

(14)

A way of alternating optimization is used to compute \( \phi_j \) and dual variable \( \bar{p}_j \):

\[
\frac{\partial \phi_j}{\partial t} = -\gamma \nabla \cdot \bar{p}_j - \sum_{i=1}^{n} \alpha_i Q_j \frac{\partial \chi_i}{\partial \phi_j} 
\]

(15)

\[
\frac{\partial \bar{p}_j}{\partial t} = -|\nabla \phi_j| \bar{p}_j - \nabla \phi_j 
\]

(16)

Though dual method can speed up the processing to some degree, it cannot obtain the ideal convergence rate.

### 2.2.3 Alternating direction method of multipliers (ADMM)

The basic idea of [23] is to use low-order variables instead of high-order variables and obtain an approximate result. The constraint \( \bar{w}_j = \nabla \phi_j \) is added in energy functional.

\[
E(\phi, \bar{w}) = \sum_{j=1}^{m} \int_{\Omega} \gamma \bar{w}_j dx + \sum_{i=1}^{n} \alpha_i \int_{\Omega} Q_j \chi_i dx 
+ \mu \sum_{j=1}^{m} \int_{\Omega} (\bar{w}_j - \nabla \phi_j)^2 dx
\]

(17)

where \( \bar{\lambda}_j \) is called the Lagrange multiplier and \( \mu \) is a penalization parameter. Then the variables are optimized as:

\[
\begin{align*}
\sum_{i=1}^{n} \alpha_i Q_i \frac{\partial \chi_i}{\partial \phi_j} + \nabla \cdot \bar{\lambda}_j + \mu \nabla \cdot (\bar{w}_j - \nabla \phi_j) &= 0 \quad \text{in } \Omega \\
(\mu (\nabla \phi - \bar{w}_j) - \bar{\lambda}_j) \cdot \bar{n} &= 0 \quad \text{on } \partial \Omega 
\end{align*}
\]

(18)

\[
\bar{w}_{j+1} = \max \left( \left| \nabla \phi_j^{k+1} - \frac{\bar{\lambda}_j}{\mu} \right| - \frac{\gamma}{\mu} \left| \nabla \phi_j^{k+1} - \frac{\bar{\lambda}_j}{\mu} \right| \right)
\]

(19)

### 2.2.4 Alternating minimization algorithm (AMA)

The computing framework of AMA is similar to ADMM. Both of them adopt the alternating direction method while AMA is simpler. The only significant difference is about the calculation of \( \phi_j \). Here Gradient descent method is used to ensure its convergence of iterative method. Then we can get \( \phi_{j+1} \) through the following iteration:

\[
\begin{align*}
\frac{\partial \phi_j}{\partial t} &= -\nabla \cdot \bar{\lambda}_j - \sum_{i=1}^{n} \alpha_i Q_i \frac{\partial \chi_i}{\partial \phi_j} \quad \text{in } \Omega \\
\frac{\partial \phi_j}{\partial n_j} &= 0 \quad \text{on } \partial \Omega 
\end{align*}
\]

(20)

### 3 Our proposed methods

Though the GDM, DM, ADMM or ALM and AMA have been successfully extended to the multiphase Chan-Vese model, they cannot get results with exact values as well as in rapid speed. Through introducing a ‘restart rule’- i.e. the acceleration parameters are reset when certain conditions are met. So both expensive computing process and complex item appearance in the evolution equations are able to be avoided by our proposed fast method.

#### 3.1 Fast alternating direction method of multipliers (FastADMM)

First the basic idea of using FastADMM is proposed. We introduce auxiliary variables \( \bar{v}_j (j = 1, 2, \ldots, m) \) to replace the \( \nabla \phi_j \), so the high-order variables can be simplified by low-order variables. Then the variables will be optimized respectively. Following conventions, the symbol ‘\( \rightarrow \)’ is going to be used to denote vector functions. The energy functional is rewritten as follows:

\[
E(c, \phi, \bar{w}, \tilde{\lambda}) = \sum_{j=1}^{m} \int_{\Omega} \gamma \bar{v}_j dx + \sum_{i=1}^{n} \alpha_i \int_{\Omega} Q_j \chi_i dx 
+ \frac{\mu}{2} \sum_{j=1}^{m} \int_{\Omega} (\bar{v}_j - \nabla \phi_j)^2 dx
\]

(21)

Where \( \tilde{\lambda}_j \) should be carefully calculated by the intermediate variable \( \bar{\lambda}_j \). Please notice that \( \phi_j \) should be updated by \( \bar{v}_j \).

If the value of \( \bar{w}_j \) is obtained by solving the Euler-Lagrange equation, \( \bar{v}_j \) will be updated by \( \bar{w}_j \). Detailed implementation of FastADMM is shown in Algorithm 1.

**Algorithm 1: FastADMM for multiphase Chan-Vese model**

1. Initialization: \( w_j = \bar{w}_j, \lambda_j = \bar{\lambda}_j, j = 1, 2, \ldots, m \), \( \alpha = 1, \mu > 0 \).
2. For \( k \geq 1 \), solve the following problems alternatively:
   1. Subproblem 1 about \( c_i^{k+1}, i = 1, 2, \ldots, n \):
      \[
      c_i^{k+1} = \arg \min \left\{ \psi_i \left( c_i \right) = E \left( c_i, \phi_j^k, \bar{w}_j^{k}; \tilde{\lambda}_j^{k} \right) \right\}. 
      \]
   2. Subproblem 2 about \( \phi_j^{k+1} \):
      \[
      \phi_j^{k+1} = \arg \min \left\{ \psi_i \left( \phi_j \right) = E \left( c_i^{k+1}, \phi_j, \bar{v}_j^{k}; \tilde{\lambda}_j^{k} \right) \right\}. 
      \]
   3. Subproblem 3 about \( \bar{w}_j^{k+1} \):
      \[
      \bar{w}_j^{k+1} = \arg \min \left\{ \psi_i \left( \bar{w}_j \right) = E \left( c_i^{k+1}, \phi_j^{k+1}, \bar{w}_j; \tilde{\lambda}_j^{k} \right) \right\}. 
      \]
\[ \lambda_j^{k+1} = \lambda_j^k + \mu (w_j^{k+1} - \nabla \phi_j^{k+1}) \]  

2.5. if \( E^k > 0 \), then a ‘restart rule’ and a relaxation factor are used to update \( \tilde{v}_j \) and \( \tilde{\lambda}_j^k \).

\[ \alpha^{k+1} = \frac{1 + \sqrt{1 + 4(\alpha^k)^2}}{2} \]  

\[ v_j^k = w_j^k + \frac{\alpha^k - 1}{\alpha^{k+1}} \left( w_j^k - w_j^k \right) \]  

\[ \tilde{\lambda}_j^{k+1} = \lambda_j^k + \frac{\alpha^k - 1}{\alpha^{k+1}} \left( \lambda_j^k - \tilde{\lambda}_j^k \right) \]  

else

\[ \alpha^{k+1} = 1, \quad v_j^k = w_j^k, \quad \tilde{\lambda}_j^{k+1} = \tilde{\lambda}_j^k \]

2.6. \( \phi_j^{k+1} \) need to be processed by threshold:

\[ \phi_j^{k+1} = \begin{cases} 1 & \phi_j^{k+1} > a \\ 0 & \text{otherwise} \end{cases} \]

3. The overall loop will be terminated if the stopping criterions (described in section 4) are satisfied.

The projection \( \prod_{i=1}^{n} \left( \cdot \right) \) in equation (24) is a simple truncation of \( \phi_j^{k+1} \) to the interval \([0,1]\). For \( k = 0,1,\ldots \), the minimizers of variables \( c_i, \phi_j, w_j \) in subproblems 1-3 can be obtained by minimizing the following energy functionals:

\[ e_i(c_i) = \sum_{i=1}^{n} \alpha_i \int_{\Omega} Q_i x_i dx \]  

\[ e_2(\phi_j) = \sum_{i=1}^{n} \alpha_i \int_{\Omega} Q_i x_i dx + \int_{\Omega} \tilde{\lambda}_j^k (\tilde{v}_j - \nabla \phi_j) dx \]  

\[ + \frac{\mu}{2} \int_{\Omega} (\tilde{v}_j - \nabla \phi_j)^2 dx \]

\[ e_3(\tilde{w}_j) = \int_{\Omega} \gamma (\tilde{w}_j) dx + \int_{\Omega} \tilde{\lambda}_j^k (\tilde{w}_j - \nabla \phi_j) dx \]  

\[ + \frac{\mu}{2} \int_{\Omega} (\tilde{w}_j - \nabla \phi_j)^2 dx \]

Thus the energy functional is simplified which can be computed by easier iterative algorithm and avoid the subproblem which occurs in \( \phi_j \) associating with no convergence successfully. Next, (31) to (33) will be solved respectively by different iterative methods.

3.1.1 Estimations of piecewise constant parameters

We can obtain \( \phi_j^{k+1}, i = 1,2,\ldots,n \) as in Equation (7).

3.1.2 Computing of the binary level set function

When \( \phi_j \) is being computed, semi-implicit Gauss-Seidel iterative scheme can be used because it can ensure its fast convergence. The \((k+1)\)th value of \( \phi_j^{k+1} \) and the \(k\)th auxiliary variable \( \tilde{v}_j^k \) should be fixed. This concept is also applied in the following paragraphs. The corresponding Euler-Lagrange equation of (32) is:

\[ \mu \nabla \cdot (\tilde{v}_j^k - \nabla \phi_j) + \nabla \cdot \tilde{\lambda}_j^k + \sum_{i=1}^{n} \alpha_i \int_{\Omega} Q_i \frac{\partial \chi}{\partial \phi_j} = 0. \]  

In experiments, we find that two or three iterative steps are enough to achieve a good minimizer of \( \phi_j \). It is a powerful guarantee of the energy functional minimization.

Now \( \tilde{\phi}_j^{k+1} \) becomes the nonstandard binary level set function. It must be projected to \([0,1]\) as in Equation (24):

\[ \tilde{\phi}_j^{k+1} = \max \left( \min \left( \tilde{\phi}_j^{k+1}, 1 \right), 0 \right). \]  

3.1.3 Calculation of the auxiliary variable

The soft thresholding formula [25-27] is used in this part to calculate the variable \( \tilde{w}_j^{k+1} \). It is one of the most classical algorithms which has been widely used to obtain minimizers of the variables in this kind of equation. The variables \( \phi_j^{k+1} \) and \( \tilde{\phi}_j^{k+1} \) are fixed. The calculation result is shown as:

\[ w_j^{k+1} = \max \left( \nabla \phi_j^{k+1} - \frac{\tilde{\lambda}_j^{k+1}}{\mu} \right) \left( 0, \frac{\nabla \phi_j^{k+1} - \tilde{\lambda}_j^{k+1}}{\mu} \right). \]  

After the minimizers of subproblems 1-3 are found, Lagrange multipliers should be updated according to (26). Now let us introduce the ‘restart rule’ which is able to guarantee convergence for weakly-convex problems. If the judging criteria \( E^k \) is greater than 0, the parameter sequence \( \{\alpha_j^k\} \) is used to over-relax the sequence of iteration and help update \( \tilde{v}_j \) and \( \tilde{\lambda}_j \). The definition of \( E^k \) is described in [19]. At the end of the implementation, we can work out the threshold \( a \) for binarization of \( \phi_j^{k+1} \) on the basis of the histogram of its result (as described in (30)). In section 4, the stopping criterions which can be used to terminate the overall loop are presented.

3.2 Fast alternating minimization algorithm (FastAMA)

The basic idea of using FastAMA is similar to FastADMM. In this algorithm, only one intermediate variable
$\lambda_j^\ast$ is used to accelerate the convergence. Detailed implementation of FastAMA is shown in Algorithm 2.

**Algorithm 2: FastAMA for multiphase Chan-Vese model**

1. Initialization: $w_j^0 = v_j, \lambda_j^0 = \lambda_j^0, \ (j=1,2,...,m)$, $\alpha^0 = 1, \mu > 0$.
2. For $k \geq 1$, solve the following problems alternatively:
   2.1. Subproblem 1 about $c_i^{k+1}, i=1,2,...n$:
   
   $$ c_i^{k+1} = \arg \min \left\{ e_1(c_i) = E(c_i, \phi_j^k, \tilde{w}_j^k, \lambda_j^k) \right\}. $$

   2.2. Subproblem 2 about $\phi_j^{k+1}$:
   
   $$ \phi_j^{k+1} = \arg \min \left\{ e_2(\phi_j) = E(c_i^{k+1}, \phi_j, \tilde{w}_j^k, \lambda_j^k) \right\}, $$

   where $\phi_j^{k+1}$ of (46) can be obtained as in equation (20), please notice that $\lambda_j^\ast$ should be replaced by the intermediate variable $\tilde{\lambda}_j^\ast$. And $\tilde{w}_j^{k+1}$ of (47) can be obtained as in (36). After the minimizers of subproblems 1-3 are found, Lagrange multipliers should be updated according to (41). The parameter sequence $\{\alpha^k\}$ is used to over-relax the sequence of iteration and help update $\lambda_j^\ast$. At the end of the implementation, binarization of $\phi^{k+1}$ is required as well. We use the same stopping criterions to terminate the overall loop as presented in section 4.

3. The overall loop will be terminated if the stopping criterions (described in section 4) are satisfied.

The minimizers of variables $c_i, \phi_j, \tilde{w}_j$ in subproblems 1-3 can be obtained by minimizing the following functionals:

$$ e_1(c_i) = \sum_{i=1}^{n} \alpha_i \int_{\Omega} Q_i c_i dx, \quad (45) $$

$$ e_2(\phi_j) = \sum_{i=1}^{n} \alpha_i \int_{\Omega} Q_i c_i dx + \int_{\Omega} \tilde{\lambda}_j^\ast (\tilde{v}_j - \nabla \phi_j) dx, \quad (46) $$

$$ e_3(\tilde{w}_j) = \int_{\Omega} \tilde{w}_j dx + \int_{\Omega} \tilde{\lambda}_j^\ast (\tilde{w}_j - \nabla \phi_j) dx + \frac{\mu}{2} \int_{\Omega} (\tilde{w}_j - \nabla \phi_j)^2 dx. \quad (47) $$

Where equation (45) can be minimized as presented in (7). $\phi_j^{k+1}$ of (46) can be obtained as in equation (20), please notice that $\lambda_j^\ast$ should be replaced by the intermediate variable $\tilde{\lambda}_j^\ast$. And $\tilde{w}_j^{k+1}$ of (47) can be obtained as in (36). After the minimizers of subproblems 1-3 are found, Lagrange multipliers should be updated according to (41). The parameter sequence $\{\alpha^k\}$ is used to over-relax the sequence of iteration and help update $\lambda_j^\ast$. At the end of the implementation, binarization of $\phi^{k+1}$ is required as well. We use the same stopping criterions to terminate the overall loop as presented in section 4.

4. **Numerical experiments**

In this section, the numerical results of our proposed methods are applied on some real cases and they will be compared with different methods (GDM, DM, ADMM or ALM, AMA) to demonstrate the effectiveness and efficiency of our methods. All the experiments are operated on the same platform (Matlab7.8) on a PC (Intel (R), CPU 2.60GHz). The same initial contours and initiations of variables for all the methods in each experiment are used in order to have a relatively neutral criterion for comparison. To clarify this, the initial values of variables are shown as follows:

GDM: $c_i^0 = 0, \phi_j^0 \in \{0,1\}$.

DM: $c_i^0 = 0, \phi_j^0 \in \{0,1\}, \tilde{p}_j^0 = 0$.

ADMM or ALM: $c_i^0 = 0, \phi_j^0 \in \{0,1\}, \tilde{w}_j^0 = 0, \tilde{\lambda}_j^0 = 0$.

AMA: $c_i^0 = 0, \phi_j^0 \in \{0,1\}, \tilde{w}_j^0 = 0, \tilde{\lambda}_j^0 = 0$.

FastADMM: $\alpha^0 = 1, c_i^0 = 0, \phi_j^0 \in \{0,1\}, \tilde{w}_j^0 = 0, \tilde{\lambda}_j^0 = 0$.

FastAMA: $\alpha^0 = 1, c_i^0 = 0, \phi_j^0 \in \{0,1\}, \tilde{w}_j^0 = 0, \tilde{\lambda}_j^0 = 0$.

As described in [28], the iterations need to be terminated when the following criterions are satisfied. In this paper, the same stopping criterions can be used in the proposed two methods.

1. We need to monitor the constraints errors in iterations:

   $$ R_{\tilde{w}_j}^k = \left\| \frac{\tilde{w}_j^k - \tilde{w}_j^{k-1}}{\tilde{w}_j^{k-1}} \right\|_{L^1}(j=1,2,...,m), $$

   with

   $$ R_{\tilde{w}_j}^k = \tilde{w}_j^k - \nabla \phi_j^k, $$

   where $\left\| \cdot \right\|_{L^1}$ denotes the $L^1$ norm on image domain $\Omega$. If $R_{\tilde{w}_j}^k < \varepsilon$ ($\varepsilon$ is a small enough parameter), iteration of outer repeat $k$ will be stopped. These good numerical indicators are also used to determine the values of $\mu$, which can be the basis of penalty parameter adjustment.
In iterations, the relative errors of Lagrange multipliers and the solution $\phi_j^k$ should be noticed. They should reduce to a sufficiently small level:

$$L_j^k = \left\| \phi_j^k - \phi_j^{k-1} \right\|_{L_j^k}, \quad (j = 1, 2, \ldots, m), \quad (50)$$

$$L_j^k = \left\| \phi_j^k - \phi_j^{k-1} \right\|_{L_j^k}. \quad (51)$$

The convergence of energy functional $E(\phi)$ need to be guaranteed. $\frac{|E(\phi^{k+1}) - E(\phi^k)|}{E(\phi^k)} \leq \varepsilon$ should be satisfied.

**Experiment 1.** Synthetic image of size 250x136 is used as the test image. In this experiment, two binary level set functions are used to detect three different subdomains ($m=2$). In Fig.1, some results of GDM, DM, ADMM or ALM, AMA and proposed two methods are firstly presented respectively so that we can make visual comparisons with the segmented images. Fig. 1(a) shows the original image. The initial contours are shown in Fig.1(b). Fig.1 (c)-(f) shows the segmentation results of GDM, DM, ADMM or ALM, AMA respectively. The segmented images shown in Fig. 1 (g) and (h) are from our proposed two methods. The parameters used in FastADMM for Fig.1 (g) are given as follows: $\gamma = 0.5$, $\mu = 0.4$, $\alpha_1 = 2$, $\alpha_2 = 1$, $\alpha_3 = 1$, $\alpha_4 = 2$ . And the parameters used in FastAMA for Fig.1 (h) are : $t = 0.1$, $\gamma = 5$, $\mu = 0.4$, $\alpha_1 = 3$, $\alpha_2 = 2$, $\alpha_3 = 1$, $\alpha_4 = 2$

Fig.1. The effects of GDM, DM, ADMM or ALM, AMA and proposed two methods. The first row: original image and the initial contours. The second and third row: segmentation results of GDM, DM, ADMM or ALM, AMA. The last row : results of our proposed two methods.

From left to right, we illustrate relative residuals (48), relative errors of Lagrange multipliers (50), relative error of $\phi_j^k$ (51) and energy curve along the outer repeat $k$ in Fig.2. The graphs come from Fig. 1 (g) and (h) respectively. It can be observed that the algorithm has converged long before 100 iterations. They also give important information about how to choose penalty parameter $\mu$ . In order to guarantee convergence as well as the speed of convergence, the constraint errors $R_{wR}^k$ should converge to zero with nearly the same speed. If $R_{wR}^k$ goes to zero quicker than others, then decrease $\mu . R_{wR}^k$ will converge to zero with the same speed as the iteration proceeds and the energy will decrease to a steady constant value when $\mu$ are chosen properly. This experiment points out that the selection of parameter $\gamma$ has no obvious effect on the results.
In the aspect of algorithm efficiency, iterations and computational time of methods presented in this experiment are given. It is easy to see that FastADMM and FastAMA have the faster convergence rate.

**TABLE 1. Comparisons of iterations and computational time**

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Iterations</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 1-(c): GDM</td>
<td>36</td>
<td>0.198</td>
</tr>
<tr>
<td>Fig. 1-(d): DM</td>
<td>20</td>
<td>0.163</td>
</tr>
<tr>
<td>Fig. 1-(e): ADMM</td>
<td>9</td>
<td>0.094</td>
</tr>
<tr>
<td>Fig. 1-(f): AMA</td>
<td>8</td>
<td>0.112</td>
</tr>
<tr>
<td>Fig. 1-(g): FastADMM</td>
<td>4</td>
<td>0.085</td>
</tr>
<tr>
<td>Fig. 1-(h): FastAMA</td>
<td>4</td>
<td>0.081</td>
</tr>
</tbody>
</table>

**Experiment 2.** In this experiment, our methods will be compared with GDM, DM, ADMM and AMA by using them on a image of size 256×256. The original image is presented in Fig. 3(a) (m=2) and the same initial contours are used in Fig. 2(b). Three parts contained by the methods mentioned above are given in Fig. 3(c)-(h). We can see all these methods can obtain almost the same segmentation effects. Parameters used in FastADMM and FastAMA are given: \( \alpha = 0.5, \mu = 0.4, \alpha_1 = 2, \alpha_2 = 1, \alpha_3 = 1, \alpha_4 = 2, \gamma = 0.04, \alpha = 3, \alpha_1 = 2, \alpha_2 = 1, \alpha_3 = 2. \)

Here a threshold method should be used to realize the binaryzation of \( \phi_j^{k+1} \). It is an important way to help find the accurate results. Non-threshold solutions of the proposed methods are shown as follows. It can be observed that non-threshold often results in fuzzy edges (red rectangles).

**Fig.3. The effects of GDM, DM, ADMM, AMA and proposed two methods. The first row: original image and the initial contours. The second and third row: segmentation results of these six methods.**

**Fig.4. Non-threshold solutions of the proposed methods. (i) comes from FastADMM. (j) comes from FastAMA.**

Next, the histograms of non-threshold solutions from FastADMM are given in Fig. 5. It gives us a good way to choose the threshold of \( \phi_j^{k+1} \). In this experiment, we find the threshold \( \alpha = 0.5 \) could be applicable.
From Table 2, it is obvious that the total computational cost required by our methods is much less than other four methods from the comparison.

**TABLE 2. Comparisons of iterations and computational time**

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Iterations</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 3-(c): GDM</td>
<td>29</td>
<td>0.298</td>
</tr>
<tr>
<td>Fig. 3- (d): DM</td>
<td>18</td>
<td>0.224</td>
</tr>
<tr>
<td>Fig. 3- (e): ADMM</td>
<td>7</td>
<td>0.162</td>
</tr>
<tr>
<td>Fig. 3- (f): AMA</td>
<td>6</td>
<td>0.158</td>
</tr>
<tr>
<td>Fig. 3- (g): FastADMM</td>
<td>4</td>
<td>0.126</td>
</tr>
<tr>
<td>Fig. 3- (h): FastAMA</td>
<td>3</td>
<td>0.132</td>
</tr>
</tbody>
</table>

Experiment 3. The results of all these methods are shown on a brain magnetic resonance image (MRI). From the original image of size 256×256 in Fig. 6(a), there are four parts need to be segmented. Fig. 6(b) shows the initial contours. The segmentation results from different methods are given in Figs. 6(c)-(h) and local enlarged results of (c)-(h) are shown in Figs. 6(i)-(n). Those subdomains separated from Fig. 6(g) and (h) with proposed methods are respectively presented in Figs. 6(o)-(p). The parameters used in FastADMM for Fig. 6 (g) are: $\gamma = 0.5$, $\mu = 0.4$, $\alpha_1 = 2$, $\alpha_2 = 3$, $\alpha_3 = 2$, $\alpha_4 = 1$. And the parameters used in FastAMA for Fig.1 (h) are : $t = 0.1$, $\gamma = 0.5$, $\mu = 0.4$, $\alpha_1 = 3$, $\alpha_2 = 2$, $\alpha_3 = 1$, $\alpha_4 = 2$.

Fig.6. The comparison between other methods and our methods on a MRI. The first row: original image and the initial contours. The second and Third row: results of other methods and our methods. The fourth and fifth row: zoomed small subregions (purple rectangles). The last two row: four different phases of (g) and (h) obtained by proposed methods.

In Table 3, comparisons of iterations and computational time using different methods are given.

**TABLE 3. Comparisons of iterations and computational time**

<table>
<thead>
<tr>
<th>Methods</th>
<th>Iterations</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 6-(c): GDM</td>
<td>25</td>
<td>2.58</td>
</tr>
<tr>
<td>Fig. 6-(d): DM</td>
<td>14</td>
<td>1.84</td>
</tr>
<tr>
<td>Fig. 6-(e): ADMM</td>
<td>9</td>
<td>0.69</td>
</tr>
<tr>
<td>Fig. 6-(f):AMA</td>
<td>9</td>
<td>0.86</td>
</tr>
<tr>
<td>Fig. 6-(g) and (o): FastADMM</td>
<td>5</td>
<td>0.41</td>
</tr>
<tr>
<td>Fig. 6-(h) and (p): FastAMA</td>
<td>4</td>
<td>0.33</td>
</tr>
</tbody>
</table>

5 Conclusions

In this paper, by using the relevant concepts of Nesterov’s accelerated algorithm, convex optimization and multiphase Chan-Vese model, we propose FastADMM and FastAMA for multiphase image segmentation. Our proposed accelerated methods have been validated by several numerical experiments. The comparison of results obtained by some
other approaches and our proposed approach indicate that our approach owns good enough effects and it is a good way to efficiently minimize the difficult functional. Our method can also be applied into surface segmentation, 3D reconstruction and image denoising models etc. in the future work. It is supposed to yield shorter runtime than the traditional methods, while the quality of results is identical.

Acknowledgements

The work has been partially supported by the National Natural Science Foundation of China (nos.61305045, 61170106, and 61303079).

References
