Fast Piecewise Constant Level Set methods (PCLSM) with Newton Updating

Xue-Cheng Tai Changhui Yao

Abstract

In this work, we try to develop a fast method for piecewise constant level set methods (PCLSM) when they are used for Mumford-Shah image segmentation. Just one level set function is needed to identify arbitrary number of phases for the segmentation problem. For the Mumford-Shah image segmentation model with PCLSM, one needs to minimize a smooth energy functional under some constrains. In order to solve the minimization problem, fast Newton updating algorithm is used to solve the Euler-Lagrangian equations. Due to the special structure of the segmentation functional, the cost for the Newton updating is nearly the same as for gradient decent method. However, the convergence is much faster with a good initial guess. Numerical experiments are given to show the efficiency and other advantages of the methods.

Keywords

PCLSM, Level set method, image segmentation, fast algorithm, Newton method.

I. introduction

The level set method proposed by Osher and Sethian [18] is a versatile tool for tracing interfaces separating a domain Ω into subregions. Interfaces are treated as the zero level set of some functions. Moving the interfaces can implicitly be done by evolving the level set functions instead of moving the interfaces directly. For a recent survey on the level set methods see [20], [13], [19].

In [7], [8], [12] some variants of the level set method of [18], the so called "piecewise constant level set method (PCLSM)", was proposed to identify arbitrary number of subregions using just one level set function. The method can be used for different applications. In [7], [8], [12], the ideas have been used for image segmentation. In [15], [22], applications to inverse shape identification problems involving elliptic and reservoir equations are shown. Different efforts have been tried to accelerate the convergence of the algorithms. In this work, we shall try to propose a quasi-Newton method which needs nearly the same cost as the gradient decent method, but has a much faster convergence. Let us note that Newton-type of methods have been used for the traditonal level set method in [2] using shape derivatives. In our approach, no derivatives with respect to shapes are needed.

Before we go any further, we want to mention some recent related approaches that have been used in the literature for image segmentation, [8], [6], [17], [16], [5], [4]. The so called "Binary Level Set" method as in [8], [6], [17], [16], [5] is more related to the phase field models. The model of [4] use multilayers, instead the constant values, and multiple level set functions to represent the phases.

This paper is organized in the following way. In §II, we review the piecewise constant level set method. In §III a faster quasi-Newton updating scheme is proposed. Details are supplied to show that the cost for this algorithm is nearly the same as for the simple gradient updating scheme. In §IV, numerical experiments are given to show the efficiency of the proposed algorithm.

II. Piecewise constant level set methods for image segmentation

We shall first recall the PCLSM of [7]. The essential idea of the PCLSM of [7] is to use a piecewise constant level set function to identify the subdomains. Assume that we need to partition the domain

CIPR,Department of Mathematics, University of Bergen, Norway. Email: Tai@mi.uib.no. Web page http://www.mi.uib.no/ ˜ tai.

CIPR,Department of Mathematics,University of Bergen, Norway. (e-mail:changhui.yao@cipr.uib.no)

Ω into subdomains Ω_i, $i = 1, 2, \ldots, n$ and the number of subdomains is a priori known. In order to identify the subdomains, we try to identify a piecewise constant level set function ϕ such that

$$\phi = i \quad \text{in } \Omega_i, \quad i = 1, 2, \ldots, n. \tag{1}$$

Thus, for any given partion $\{\Omega_i\}_{i=1}^n$ of the domain Ω, it corresponds to a unique PCLS function ϕ which takes the values $1, 2, \cdots, n$. Associated with such a level set function ϕ, the characteristic functions of the subdomains are given as

$$\psi_i = \frac{1}{\alpha_i} \prod_{j=1, j\neq i}^{n} (\phi - j), \quad \alpha_i = \prod_{k=1, k\neq i}^{n} (i - k). \tag{2}$$

If ϕ is given as in (1), then we have $\psi_i(x) = 1$ for $x \in \Omega_i$, and $\psi_i(x) = 0$ elsewhere. We can use the characteristic functions to extract geometrical information for the subdomains and the interfaces between the subdomains. For example,

$$\text{Length}(\partial\Omega_i) = \int_\Omega |\nabla\psi_i| dx, \quad \text{Area}(\Omega_i) = \int_\Omega \psi_i dx. \tag{3}$$

In fact, the level set function also satisfies the relation $\phi = \sum i\psi_i$. Define

$$K(\phi) = (\phi - 1)(\phi - 2)\cdots(\phi - n) = \prod_{i=1}^{n}(\phi - i). \tag{4}$$

At every point in Ω, the level set function ϕ satisfies

$$K(\phi) = 0. \tag{5}$$

This level set idea has been used for Mumford-Shah image segmentation in [7]. For a given digital image $u_0 : \Omega \mapsto R$ which may be corrupted by noise and blurred, the piecewise constant Mumford-Shah segmentation model is to find curves Γ and constant values c_i to minimize:

$$\sum_i \int_{\Omega_i} |c_i - u_0|^2 dx + \beta|\Gamma|. \tag{6}$$

The curves Γ separate the domain Ω into subdomains Ω_i and $\Omega = \cup_i\Omega_i \cup \Gamma$. In Chan-Vese [23], the traditional level set idea of [18] was used to repsent the curves Γ and to solve the problem (6). In [7], PCLSM was used for the Mumford-Shah model (6). Note that a function u given by:

$$u = \sum_{i=1}^{n} c_i\psi_i \tag{7}$$

is a piecewise constant function and $u = c_i$ in Ω_i if ϕ is as given in (1). The sum in u involves characteristic functions of polynomial functions of order $n-1$ in ϕ and the unknown coefficient c_i. Each ψ_i is expressed as a product of linear factors of the form $(\phi - j)$, with the ith factor omitted.

Based on the above observations, we propose to solve the following constrained minimization problem for segmenting an image u_0:

$$\min_{\substack{\mathbf{c},\, \phi \\ K(\phi)=0}} \Big\{ F(\mathbf{c}, \phi) = \frac{1}{2}\int_\Omega |u - u_0|^2 dx + \beta \sum_{i=1}^{n} \int_\Omega |\nabla\psi_i| dx + \nu \int_\Omega |\nabla\phi| dx \Big\}. \tag{8}$$

We see that large approximation errors will be regularized by the fidelity term $\frac{1}{2}\int_\Omega |u-u_0|^2$. From (3), it is clear that the latter two terms as the regularization terms suppress oscillation. The regularization parameter $\beta > 0, \nu > 0$ control the effect of the two terms. If the image u_0 is a piecewise constant function and we take $\beta = 0, \nu = 0$, then any minimizers of (8) will give a function u such that $u = u_0$ where u is related to the minimizers $\mathbf{c}$ and ϕ in (7).

In [7], the augmented Lagrangian method was used to solve the constrained minimization problem (8). The augmented Lagrangian functional for this minimization problem is defined as

$$L(\mathbf{c}, \phi, \lambda) = F(\mathbf{c}, \phi) + \int_\Omega \lambda K(\phi)\, dx + \frac{r}{2}\int_\Omega |K(\phi)|^2 dx, \tag{9}$$

where $\lambda \in L^2(\Omega)$ is the multiplier and $r > 0$ is a penalty parameter. For the augmented Lagrangian method, it is necessary to choose the penalization parameter r very large. To find a minimizer for (8), we need to find the saddle points for L. The following Uzawa gradient algorithm was used in [7] to find a saddle point for $L(\mathbf{c}, \phi, \lambda)$.

***Algorithm** 1: Choose initial values for ϕ^0 and λ^0. For $k = 1, 2, \cdots,$ do:*

1. *Find $\mathbf{c}^k$ from*
$$L(\mathbf{c}^k, \phi^k, \lambda^{k-1}) = \min_{\mathbf{c}} L(\mathbf{c}, \phi^k, \lambda^{k-1}). \tag{10}$$
2. *Use (7) to update $u = \sum_{i=1}^n c_i^k \psi_i(\phi^{k-1})$.*
3. *Find ϕ^k from*
$$L(\mathbf{c}^k, \phi^k, \lambda^{k-1}) = \min_{\phi} L(\mathbf{c}^k, \phi, \lambda^{k-1}). \tag{11}$$
4. *Use (7) to update $u = \sum_{i=1}^n c_i^k \psi_i(\phi^k)$.*
5. *Update the Lagrange-multiplier by*
$$\lambda^k = \lambda^{k-1} + rK(\phi^k). \tag{12}$$

This algorithm has a linear convergence and its convergence has been analyzed by Kunisch and Tai in [9] under a slightly different context. The algorithm has also been used by Chan and Tai in [24], [3] for elliptic inverse problems.

The minimizer c^k for (10) can be obtained by solving a small $n \times n$ linear algebraic system. The minimizer for (11) is normally solved by the gradient decent method, i.e.

$$\phi^{new} = \phi^{old} - \Delta t \frac{\partial L}{\partial \phi}(\mathbf{c}^k, \phi^{old}, \lambda^{k-1}). \tag{13}$$

The step size Δt is chosen by a trial and error approach and it is fixed during the whole iterative procedure. It is not necessary to solve the minimization problem (11) exactly. The gradient iteration (13) is terminated when

$$\left\| \frac{\partial L}{\partial \phi}(\mathbf{c}^k, \phi^{new}, \lambda^{k-1}) \right\|_{L^2} \leq \frac{1}{10} \left\| \frac{\partial L}{\partial \phi}(\mathbf{c}^k, \phi^{k-1}, \lambda^{k-1}) \right\|_{L^2} \tag{14}$$

is reached or else after a fixed number of iterations. To compute $\frac{dL}{d\phi}$, it is easy to see that

$$\frac{\partial L}{\partial \phi} = (u-u_0)\frac{\partial u}{\partial \phi} - \beta \sum_{i=1}^n \nabla \cdot \Big(\frac{\nabla \psi_i}{|\nabla \psi_i|}\Big) \frac{\partial \psi_i}{\partial \phi} - \nabla \cdot \Big(\frac{\nabla \phi}{|\nabla \phi|}\Big) + \lambda K'(\phi) + rK(\phi)K'(\phi). \tag{15}$$

It is easy to get $\partial u/\partial \phi$, $\partial \psi_i/\partial \phi$ and $K'(\phi)$ from (7),(2) and (4).

III. Quasi-Newton updating

Different approaches have been used to accelerate the convergence of the PCLSM. Motivated by [5], the MBO projection of [14] has been applied in [21] to deal with the constraint $K(\phi) = 0$. In [21], [22], [11], some kind of "soft" MBO projection was used. In this work, we try to use a quasi-Newton method to deal with the constraint.

Given $c^k, \phi^{k-1}, \lambda^{k-1}$, the following Newton method can be used to update ϕ and λ to get ϕ^k and λ^k, c.f [1]:

$$\begin{pmatrix} \frac{\partial^2 L}{\partial \phi^2} & \frac{\partial^2 L}{\partial \phi \partial \lambda} \\ \frac{\partial^2 L}{\partial \phi \partial \lambda} & 0 \end{pmatrix} \begin{pmatrix} \phi^k - \phi^{k-1} \\ \lambda^k - \lambda^{k-1} \end{pmatrix} = - \begin{pmatrix} \frac{\partial L}{\partial \phi} \\ \frac{\partial L}{\partial \lambda} \end{pmatrix}. \tag{16}$$

In order to solve the above system, we need to invert a huge linear algebraic system due to the regularization terms in (8). In many practical applications, it is often useful to replace the Newton updating by quasi-Newton updating, i.e. replace the Hessian matrix by some approximate Hessian matrix. Our numerical experiments indicate that the following approach is rather efficient. In order to describe the approach, we define

$$Q(\mathbf{c}, \phi, \lambda) = \frac{1}{2} \int_\Omega |u(\mathbf{c}, \phi) - u_0|^2 dx + \int_\Omega \lambda K(\phi)\, dx. \tag{17}$$

Our numerical experience also reveals that it is not necessary to use the penalization term with the Newton updatings. Thus, we also define

$$L_0(\mathbf{c}, \phi, \lambda) = F(\mathbf{c}, \phi) + \int_\Omega \lambda K(\phi)\, dx. \tag{18}$$

It is easy to see that L_0 is equal to L if we take $r = 0$ in (9). In addition, the functional L_0 given in (18) reduces to Q if we take $\beta = \nu = 0$. Thus the Hessian matrix for Q is a good approximation for the Hessian matrix of L_0 using the fatc that β and ν are normally very small. The new algorithm using the quasi-Newton updating and the Lagrangian functional L_0 is given in the following:

***Algorithm** 2:* **(Quasi-Newton algorithm)** *Choose initial values ϕ^0, λ^0. For $k = 1, 2, \cdots$, do:*

1. *Find $\mathbf{c}^k$ from*

$$L_0(\mathbf{c}^k, \phi^{k-1}, \lambda^k) = \min_{\mathbf{c}} L_0(\mathbf{c}, \phi^{k-1}, \lambda^k). \tag{19}$$

2. *Update $u = \sum_{j=1}^n c_j^k \psi_j(\phi^{k-1})$.*
3. *Find ϕ^k, λ^k from*

$$\begin{pmatrix} \frac{\partial^2 Q}{\partial \phi^2} & \frac{\partial^2 Q}{\partial \phi \partial \lambda} \\ \frac{\partial^2 Q}{\partial \phi \partial \lambda} & 0 \end{pmatrix} \begin{pmatrix} \phi^k - \phi^{k-1} \\ \lambda^k - \lambda^{k-1} \end{pmatrix} = - \begin{pmatrix} \frac{\partial L_0}{\partial \phi} \\ \frac{\partial L_0}{\partial \lambda} \end{pmatrix}. \tag{20}$$

4. *Update $u = \sum_{j=1}^n c_j^k \psi_j(\phi^k)$.*
5. *If converged, end the loop. Otherwise go to step 1.*

In order to solve (20), we need to invert the approximate Hessian matrix

$$\tilde{H} = \begin{pmatrix} \frac{\partial^2 Q}{\partial \phi^2} & \frac{\partial^2 Q}{\partial \phi \partial \lambda} \\ \frac{\partial^2 Q}{\partial \phi \partial \lambda} & 0 \end{pmatrix} \Bigg|_{(\mathbf{c}^k, \phi^{k-1}, \lambda^{k-1})}.$$

It is easy to see that $\partial L_0 / \partial \lambda = K(\phi^{k-1})$ and $\partial L_0 / \partial \phi$ can be obtained from (15) by setting $r = 0$. Using the chain rule, it is true that

$$\frac{\partial^2 Q}{\partial \phi^2} = \Big(\frac{\partial u}{\partial \phi}\Big)^2 + (u - u_0)\frac{\partial^2 u}{\partial \phi^2} + \lambda K^{''}(\phi), \quad \frac{\partial^2 Q}{\partial \phi \partial \lambda} = \frac{\partial^2 Q}{\partial \lambda \partial \phi} = K^{'}(\phi). \tag{21}$$

To solve this algebraic system, it is equivalent to solve a 2×2 system at each grid point. Thus, the cost for Algorithm 2 is nearly the same as for Algorithm 1. The solving of (19) is the same as in [7]. For clarity, we briefly outline it here. As u is linear with respect to the c_i values, we see that L_0 is quadratic with respect to c_i. Thus the minimization problem (19) can be solved exactly. Note that

$$\frac{\partial L_0}{\partial c_i} = \int_\Omega \frac{\partial L_0}{\partial u}\frac{\partial u}{\partial c_i} = \int_\Omega (u-u_0)\psi_i dx \quad \text{for } \; i = 1, 2, \cdots n. \tag{22}$$

Therefore, the minimizer of (19) satisfies a linear system of equations $A\mathbf{c}^k = b$:

$$\sum_{j=1}^{n} \int_\Omega (\psi_j \psi_i) c_j^k \, dx = \int_\Omega u_0 \psi_i \, dx, \quad \text{for } \; i = 1, 2, \cdots n. \tag{23}$$

In the above $\psi_j = \psi_j(\phi^{k-1})$, $\psi_i = \psi_i(\phi^{k-1})$ and thus, $\mathbf{c}^k = \{c_i^k\}_{i=1}^n$ depends on ϕ^{k-1}. The matrix A and vector b are assembled at each iteration and the equation (23) is solved by an exact solver.

Some remarks about the above algorithm are given in the following.

***Remark** 1:* In order to have convergence for the quasi-Newton algorithm, we need relative good initial values. There are different ways to get initial values. In our simulations, we have use Algorithm 1 for getting them. For many of the test examples, the simple scaling procedure outlines in §IV is good enough to get Algorithm 2 to converge.

***Remark** 2:* Similar to Algorithm 1, it is not necessary to update the $\mathbf{c}$ values too earlier and too often during the iteration procedure, c.f. [7]. For image segmentation problems, it is rather easy to get good initial guesses for the $\mathbf{c}$ values.

***Remark** 3:* Generally, we set $\nu = 0$ and take a small value for β. If the interfaces are oscillatory, we increase the value of β. When the noise is large, we take a nonzero value for ν which gives faster convergence for the algorithms.

IV. NUMERICAL EXAMPLES

In this section, we will present some numerical examples with images that have been tested on other realted algorithms. We have used the following scaling procedure to get initial values for ϕ and $\mathbf{c}$.

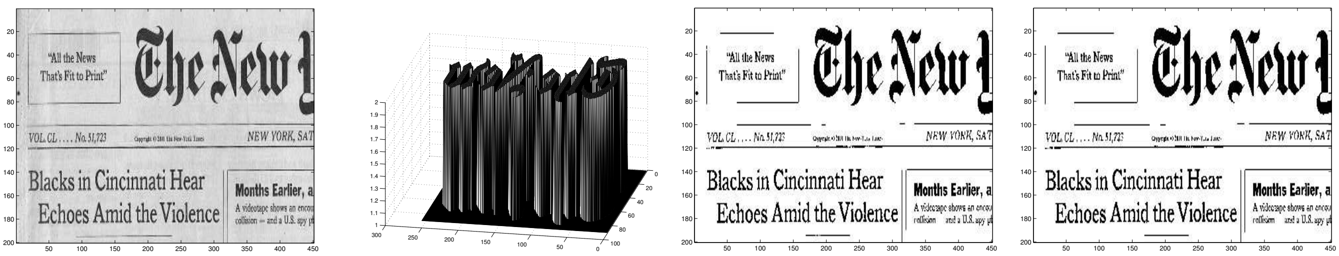
First, we need to determine the phase number n before we start. Once the value of n has been fixed, we scale u_0 to a function between 1 and n and take this as the initial value for ϕ, i.e.

$$\phi^0(x) = 1 + \frac{u_0(x) - \min_{x\in\Omega} u_0}{\max_{x\in\Omega} u_0 - \min_{x\in\Omega} u_0} \times (n-1). \tag{24}$$

For Algorithm 2, we also need an initial value for $\mathbf{c}$ and it is obtained by the following technique. ¿From ϕ^0, we define $\tilde{\phi}^0 = 1$ if $\phi^0 \le 1.5$, $\tilde{\phi}^0 = i$ if $\phi^0 \in (i - 1/2, i + 1/2], i = 2, 3, \cdots, n-1$, and $\tilde{\phi}^0 = n$ if $\phi^0 > n - 1/2$. Use this $\tilde{\phi}^0$ as ϕ^k in (23) to get a $\mathbf{c}^k$ and use it as an initial value for $\mathbf{c}$. The initial values obtain by this procedure are often good enough to get convergence for Algorithm 2. If it is not, we use them as initial values for Algorithm 1. We do a fixed number of iterations and then use the obtained image of Algorithm 1 as the initial value for Algorithm 2. In the following, we shall refer to Algorithm 1 as the **gradient updating** algorithm and refer to Algorithm 2 as the **Newton updating** algorithm.

We consider only two-dimensional grey scale images. To complicate the segmentation process we typically expose the original image with Gaussian distributed noise and use the polluted image as observation data u_0. To indicate the amount of noise that appears in the observation data, we report the signal-to-noise-ratio: SNR=$\frac{\text{variance of data}}{\text{variance of noise}}$.

First, we use two examples to demonstrate that the Newton updating is an efficient alternative to the multiphase algorithm of [10] where standard level set formulation is utilized and of [7] where standard PCLSM was used with the augmented Lagrangian method. We begin with an image of an old newspaper where only two phases are needed. One phase represents the characters and the other phase represents the background of the newspaper. In this test, we do not need to use the gradient method to generate the initial values for Algorithm 2. The Newton updating algorithm only uses 12 iterations to obtain an image that is as good as the image produced by the gradient updating algorithm at 300 iterations, where $\beta = 0.5, \nu = 0, r = 10^4, \triangle t = 5e - 6$. The segmentation has been done on the whole newspaper. In order to show the results clearly, we have just plotted a small portion of the images. The results achieved with Newton updating and Gradient updating are shown in Fig.1(c) and Fig.1(d) respectively. The image obtained by the Newton method looks the same as the one obtained by the gradient method.

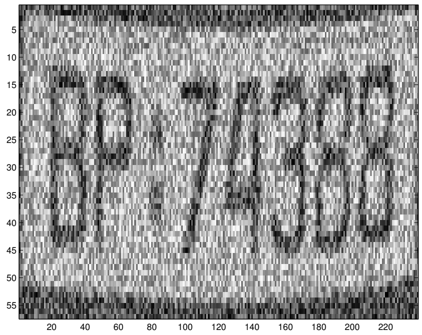


(a) An old newspaper (b) A small partition of ϕ (c) 12 Newton iterations (d) 300 Gradient iterations

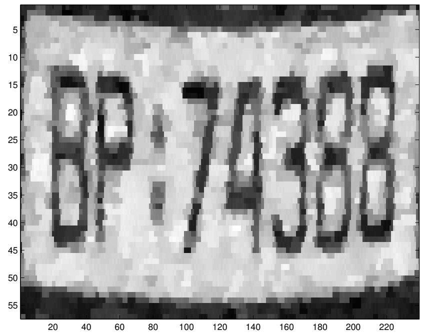
Fig. 1. Segmented images by Newton updating and Gradient updating. Only a small portion of the image is plotted here. (a) An old real newspaper. (b) A small partition of the convergent ϕ, it is a piecewise constant function. (c) Segmented image using Newton updating at 12 iterations, $\beta = 0.1$. (d) Segmented image using Gradient updating at 300 iterations.

The next example is a 2-phase segmentation on a real car plate image. The purpose of this test is to compare the performance of different algorithms that have been used in the literature. Like in [7], we challenge the segmentation techniques by adding a large amount Gaussian distributed noise to the real image and use the polluted image in Fig.2(b) as the observation data. We shall compare three algorithms, i.e the Newton updating algorithm, the gradient updating algorithm and the Algorithm of Chan-Vese [10]. As the noise is large, the simple scaling procedure is not good enough to get convergence for the Newton updating. Thus, we use the scaling procedure to get the initial values for Algorithm 1, where $\beta = 25, \nu = 0.3, r = 0.25 \times 10^4, \triangle t = 1e - 7$, and use the obtained image produced by Algorithm 1 at 137 iterations as the initial image for the Newton updating algorithm. It was observed that 5 Newton iterations is able to produce an image that is as good as the image produced by Algorithm 1 using 1059 iterations. The segmented images are displayed in Fig.2. Fig.2(f) is the image produced by Chan/Vese method (CVM), c.f. [10]. This example and the other examples clearly demonstrate the efficiency of the Newton updating algorithm.

(a) Original observation car plate.



(b) Noisy image



(c) Initial value for Newton updating

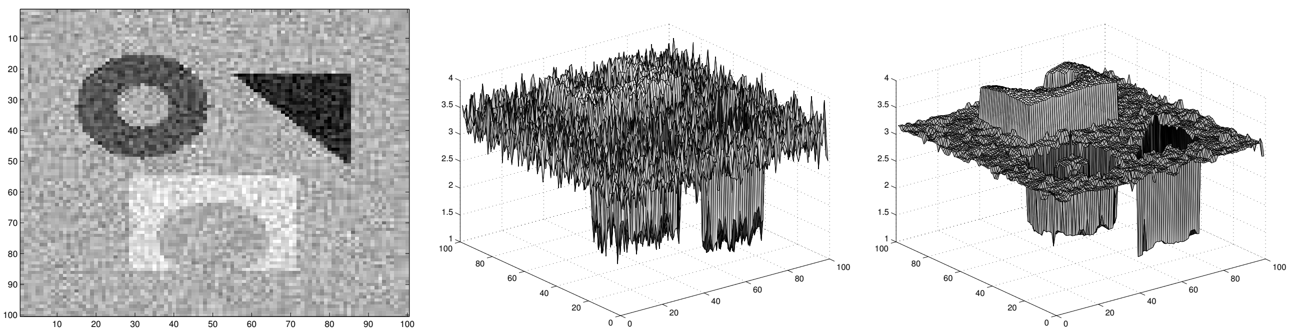
(d) 5 Newton iterations

(e) 1059 Gradient iterations

(f) Segmented with CVM .

Fig. 2. A comparison between Newton, gradient updatings and the CV method. (a) An original observation car plate. (b) A noisy car plate with SNR $\approx$1.7. (c) The initial value of ϕ for Newton updating. (d) Segmented image using Newton updating at 5 iterations with $\beta = 0.5$. (e) Segmented image by gradient updating at 1059 iterations. (f) Segmented image with CVM.

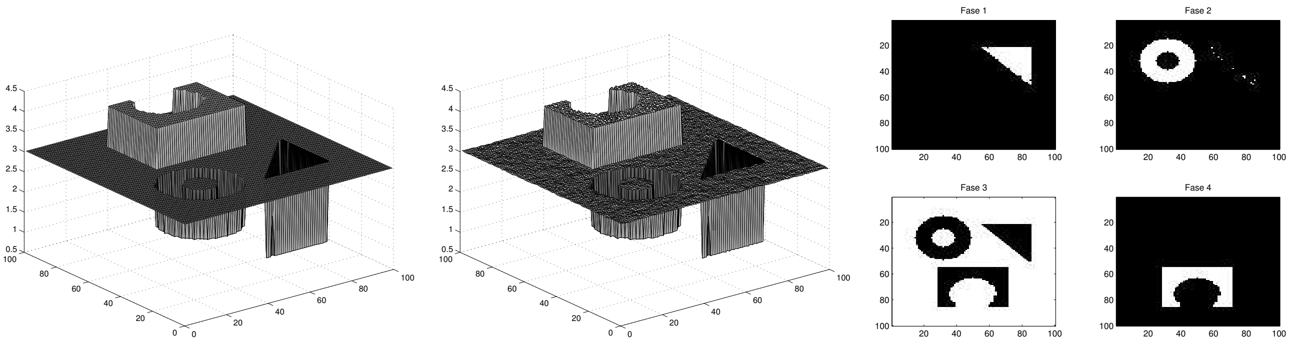
In order to show that the Newton updating algorithm can also just use one level set function to identify arbitrary number of phases, we have test it on a 4-phase segmentation problem. We begin with a noisy synthetic image containing 3 objects (and a background) as show in Fig.3(a). This is the same image as in [7], [10]. We take $\beta = 1.7, \nu = 0.075, r = 10^2, \triangle t = 5e - 6$ and use 147 iterations of Algorithm 1 to produce the initial image for Algorithm 2, see Fig.3(c). A careful evaluation of our algorithm is reported below. The observation data u_0 is given in Fig.3(a) and the only assumption we make is that a 4-phase model should be utilized to find the segmentation. The results of Fig.3(d) is produced by 6 Newton iterations starting from the initial image given in Fig.3(c). The gradient updating scheme needs 800 iterations to come to a similar segmentation. In the end, ϕ approaches the predetermined constants $\phi = 1 \vee 2 \vee 3 \vee 4$. Each of these constants represents one unique phase as seen in Fig.3(f). Our result is in accordance with what were reported in [7], [10]. For some applications, we may not know the exact number of phases. As was demonstrated in [7], some of the phases will be empty if we take n to be bigger than 4. Some of the phases will be merged into one phase if we take n to be less than 4.



(a) Observed image u_0 .

(b) Initial ϕ^0 for Algorithm 1

(c) Initial ϕ^0 for Algorithm 2



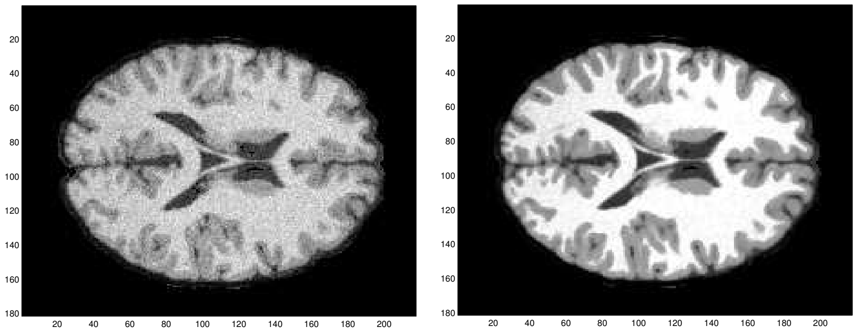
(d) 6 Newton iterations

(e) 800 Gradient iterations

(f) At converges ϕ approaches 4 constant values with (d).

Fig. 3. A four-phase segmentation are used to test the Newton updating algorithm. (a) An observed image u_0 (SNR$\approx$ 5.2). (b) The initial image used for Algorithm 1. (c) Initial ϕ^0 for the Newton updating produced by 147 iterations of Algorithm 1 using the initial image of (b). (d) Segmented with Newton updating at 6 iterations with $\beta = 0.1$. (e) The segmented image by Algorithm 1 at 800 iterations. (f) Each segmented phase $\phi = 1 \vee 2 \vee 3 \vee 4$.

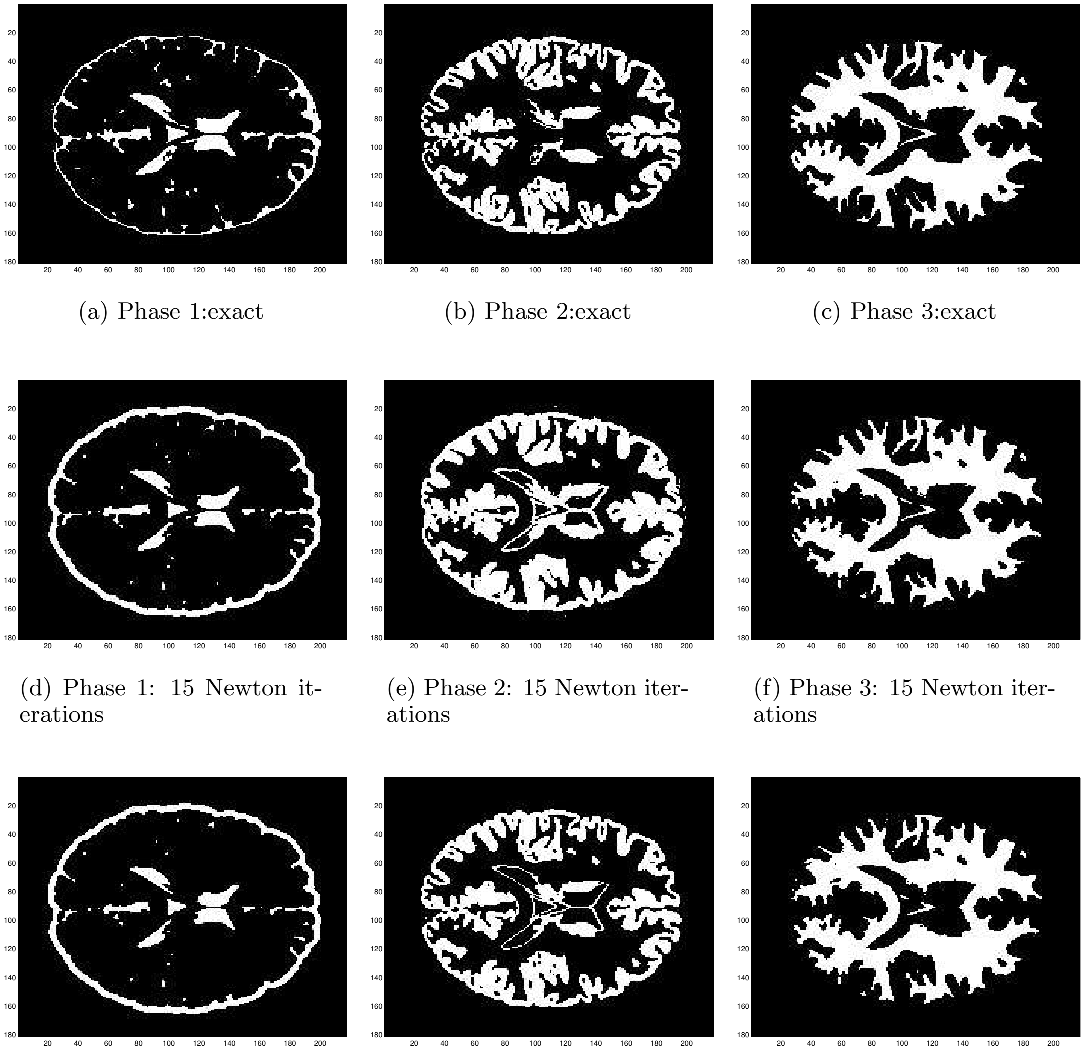
In the last example segmentation of a MR image is demonstrated. The image in Fig.5(a) is available to the public at *http : //www.bic.mni.mcgill.ca/brainweb/*. These realistic MRI data are used by the neuro imaging community to evaluate the performance of various image analysis methods in a setting where the truth is known. For the image used in this test the noise level is 7% and the non-uniformity intensity level of the RF-puls is 20%, see *http : //www.bic.mni.mcgll.ca /brainweb/* for details concerning the noise level percent and the intensity level of the RF-puls. We take $\beta = 0.04, \nu = 0, r = 0.25 \times 10^4, \triangle t = 5e - 6$, and use 29 iterations of Algorithm 1 to produce the initial image for Algorithm 2, see Fig.4(b). ¿From the initial image given in Fig.4(b), only 15 Newton iterations are needed to produce the segmented image shown in Fig.5(d)(e)(f). Compared with Fig.5(g) (h)(i) which are produced by Algorithm 1 with 250 gradient iterations, it takes less time for Algorithm 2 to get the results of Fig.5(d)(e)(f). In Fig.5 there are three tissue classes that should be identified; phase 1: cerebrospinal fluid, phase 2: gray matter, phase 3: white matter.



(a) MRI brain image

(b) Initial image for Newton updating

Fig. 4. (a) MRI brain image with a change in the intensity values going from left to right caused by the non-uniformity RF-puls. (b) Initial image for Algorithm 2 using 29 iterations of Algorithm 1.



(a) Phase 1:exact

(b) Phase 2:exact

(c) Phase 3:exact

(d) Phase 1: 15 Newton iterations

(e) Phase 2: 15 Newton iterations

(f) Phase 3: 15 Newton iterations

(g) Phase 1: 250 gradient iterations

(h) Phase 2: 250 gradient iterations

(i) Phase 3: 250 gradient iterations

Fig. 5. Three phases from the exact, Newton updating and gradient updating. (a) (b) (c) are the exact phases. (d)(e)(f) are the segmented phases with Newton updating at 15 iterations with $\beta = 0.1$. (g)(h)(i) are the segmented phases with gradient updating at 250 iterations.

V. Conclusion

We have also done many other tests for Algorithm 2 with the Newton updating. It is confirmed that the Newton updating algorithm is very fast. We can use the gradient updating scheme to produce the initial image for the Newton updating. There are also many other methods that can be used to get the initial images.

Another PCLSM was proposed in [8] and it was called the Binary Level Set Method. The binary level set method extends the ideas of [6], [17] and phase field models [16]. It is clear that there is no problem to extend the Newton updating algorithm to the binary level set method to accelerate the convergence.

The algorithms proposed here are able to identify arbitrary number of phase by just one level set function. Moreover, the method is easy to be extended to higher dimensional problems to segment color and video images.

References

[1] Dimitri P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Computer Science and Applied Mathematics. Academic Press Inc. Harcourt Brace Jovanovich Publishers, New York, 1982.

[2] Martin Burger, Benjamin Hackl, and Wolfgang Ring. Incorporating topological derivatives into level set methods. *J. Comput. Phys.*, 194(1):344–362, 2004.

[3] Tony F. Chan and Xue-Cheng Tai. Identification of discontinuous coefficients in elliptic problems using total variation regularization. *SIAM J. Sci. Comput.*, 25(3):881–904 (electronic), 2003.

[4] Jason T. Chung and Luminita A. Vese. Image segmentation using a multilayer level-set approach. *UCLA-CAM 03-53*, 2003.

[5] Selim Esedoglu and Yen-Hsi Richard Tsai. Threshold dynamics for the piecewise constant mumford-shah functional. Cam-report-04-63, UCLA, Applied Mathematics, 2004.

[6] Frédéric Gibou and Ronald Fedkiw. A fast hybrid k-means level set algorithm for segmentation. *Stanford Technical Report*, 2002.

[7] J.Lie, M. Lysaker, and X.-C. Tai. A variant of the level set method and applications to image segmentation. Cam-report-03-50, UCLA, Applied Mathematics, 2003.

[8] J.Lie, M. Lysaker, and X.-C. Tai. A binary level set model and some applications for mumford-shah image segmentation. Cam-report-04-31, UCLA, Applied Mathematics, 2004.

[9] K.Kunisch and X.-C. Tai. Sequential and parallel splitting methods for bilinear control problem in hilbert space. *SIAM J.Numer.Anal.*, 34:91–118, 1997.

[10] L.A.Vese and T.F.Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.

[11] Hongwei Li and Xue-Cheng Tai. Piecewise constant level set methods (pclsm) for multiphase motion. Technical report, UCLA, Applied Mathematics, 2005.

[12] J. Lie, M. Lysaker, and X.-C. Tai. Piecewise constant level set methods and image segmentation. In Ron Kimmel, Nir Sochen, and Joachim Weickert, editors, *Scale Space and PDE Methods in Computer Vision: 5th International Conference, Scale-Space 2005*, volume 3459, pages 573–584. Springer-Verlag, Heidelberg, April 2005.

[13] S.Osher M.Burger. A survey on level set methods for inverse problems and optimal design. Cam-report-04-02, UCLA, Applied Mathematics, 2004.

[14] B. Merriman, J. Bence, and S. Osher. Motion of multiple junctions: A level set approach. *J. Comput. Phys.*, 112(2):334, 1994.

[15] Lars K. Nielsen, Xue-Cheng Tai, Sigurd Aannosen, and Magne Espedal. A binary level set model for elliptic inverse problems with discontinuous coefficients. Technical report, UCLA, Applied Mathematics, 2005.

[16] J. Shen. Gamma-convergence approximation to piecewise constant Mumford-Shah segmentation. Cam-report-05-16, UCLA, Applied Mathematics, 2005.

[17] B. Song and T. Chan. A fast algorithm for level set based optimization. Cam-report-02-68, UCLA, Applied Mathematics, 2002.

[18] S.Osher and J.A.Sethian. Fronts propagating with curvature dependent speed: algorithms based on hamilton-jacobi formulations. *J.Comput.Phys.*, 79:12–49, 1988.

[19] S.Osher and R.Fedkiw. An overview and some recent rsults. *J.Comput.Phys*, 169 No. 2:463–502, 2001.

[20] Xue-Cheng Tai and Tony F. Chan. A survey on multiple level set methods with applications for identifying piecewise constant functions. *Int. J. Numer. Anal. Model.*, 1(1):25–47, 2004.

[21] Xue-Cheng Tai, Oddvar Christiansen, Ping Lin, and Inge Skjaelaaen. A remark on the mbo scheme and some piecewise constant level set methods. Cam-report-05-24, UCLA, Applied Mathematics, 2005.

[22] Xue-Cheng Tai and Hongwei Li. Piecewise constant level set methods (pclsm) for elliptic inverse problems. Technical report, UCLA, Applied Mathematics, 2005.

[23] T.Chan and L.A.Vese. Active contours without edges. *IEEE image Proc.*, 10:266–277, 2001.

[24] T.F.Chan and X-C.Tai. Level set and total variant regularization for elliptic inverse problems with discontinuous coefficients. *J.Comput.Physics*, 193:40–66, 2003.