A remark on the MBO scheme and some piecewise constant level set methods *

Xue-Cheng Tai† Oddvar Christiansen‡ Ping Lin§
Inge Skjælaaen¶

April 1, 2005

Abstract

The MBO scheme was originally proposed for solving equations from motion by mean curvature. Recently, the MBO scheme was interpreted as a splitting scheme for the phase field model and extended to image segmentation problems. In this work, we combine the MBO scheme with some piecewise constant level set methods (PCLSM) proposed recently. The combined scheme is much more efficient compared to the schemes used in the original PCLSM. Numerical experiments are given to demonstrate this. Advantages and disadvantages are discussed for the combined scheme.

1 Introduction

Recently, some piecewise constant level set methods (PCLSM) were proposed in Lie-Lysaker-Tai [9, 10] for image segmentation and other interface problems. These methods are closely related to the phase field models [5, 20, 21, 7, 1, 17, 18]. A scheme related to the phase field model for motion by mean curvature is the MBO scheme of Merriman-Bence-Osher [15]. The MBO scheme [15] is in fact a splitting scheme for the phase field models, see Esedouglu and Tsai [6]. Using the relationship between MBO and the splitting scheme, the MBO scheme was also extended to image segmentation in [15].

*We acknowledge support from the Norwegian Research Council and IMS of the National University of Singapore.

†Department of Mathematics, University of Bergen, Johannes Brunsgate 12, N-5007 Bergen, Norway. Email: Tai@mi.uib.no. URL: http://www.mi.uib.no/˜tai.

‡Department of Mathematics, University of Bergen, Johannes Brunsgate 12, N-5007 Bergen, Norway.

§Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543.

¶Department of Mathematics, University of Bergen, Johannes Brunsgate 12, N-5007 Bergen, Norway.

The phase field models [5, 20, 21, 7, 1, 17, 18] and the piecewise constant level set approaches [9, 10] are essentially trying to minimize some energy under some constraints, c.f. (28) for the exact forms of the constraints. If a function satisfies these constraints, the function must be a piecewise constant function. For the phase field models, these constraints are handled by a penalization method [5, 7, 17, 18]. For a given penalization parameter, the solution of the phase field models can be a smooth function and thus the energy for the phase field models is energy for smooth functions. It was verified theoretically that the smooth energy converges to the energy of a piecewise constant function when the penalization parameter goes to zero [5, 7, 17, 18]. In the piecewise constant level set approaches [9, 10], the constraints are handled by the augmented Lagrangian method. The advantage for such an approach is that we do not need to use a penalization parameter which must be taken to be very small in order to approximate the constrained minimization problem. Because the solution in our approach is a piecewise constant function, energies for smooth functions cannot be used. Instead, energies involving total variation of piecewise smooth functions are used for the length term.

The purpose of this work is to clarify a relationship between the methods proposed in [9, 10] and [6]. Similar to [6], we can combine the MBO scheme with the PCSLM of [9, 10]. The combined schemes offer efficient implementations for the PCLSM of [9, 10]. The efficiency may further be improved by the operator splitting schemes of [12, 13, 24]. A number of algorithms are summarized during the description of our work. Algorithm 1 is a recall of the MBO scheme. Algorithms 2 and 3 are an explanation of MBO as a splitting scheme for a phase field model. Algorithms 5 and 6 are then the applications of Algorithms 2 and 3 incorporated with dimensional splitting to a piecewise constant level set function for multi-phase image segmentation. Our numerical experiments indicate that Algorithm 6 is mostly recommended.

2 Sequential and parallel splitting algorithms

For a given function space V and an operator (linear or nonlinear) defined in V, we often need to solve the following time dependent equation:

$$\frac{\partial \phi}{\partial t} + A(\phi) = f(t), \quad t \in [0, T], \quad \phi(0) = \hat{\phi} \in V. \tag{1}$$

In case that the operator A and the function f can be split in the following way:

$$A = A_1 + A_2 + \cdots A_m, \quad f = f_1 + f_2 + \cdots f_m, \tag{2}$$

then some splitting schemes can be used to approximate the solution of (1). Normally, the operators A_i are simpler and easier to solve. The first scheme is called the parallel splitting scheme or additive operator splitting (AOS) scheme. First we choose a time step τ and set $\phi^0 = \hat{\phi}$. At each time level $t_j = j\tau$, we compute $\phi^{j+\frac{i}{2m}}$ in parallel for $i = 1, 2, \cdots, m$ from:

$$\frac{\phi^{j+\frac{i}{2m}} - \phi^j}{m\tau} + A_i(\phi^{j+\frac{i}{2m}}) = f_i(t_j), \text{ and then set } \phi^{j+1} = \frac{1}{m}\sum_{i=1}^{m} \phi^{j+\frac{i}{2m}}. \quad (3)$$

Note that all the subproblems for the operators A_i use the same initial value ϕ^j. This algorithm was first proposed in Lu, Neittaanmaki and Tai [12, 13]. It was discovered independently later in [24] and used in a different context for image processing [25, 22, 3, 2]. This scheme is locally second order of accuracy and globally first order of accuracy, i.e.

$$e^j = \phi^j - \phi(t_j) = O(\tau). \quad (4)$$

See [13] for a proof of this error estimate. The advantage of the above scheme is that all the subproblems can be computed in parallel. Another advantage of the scheme is that it treats all the operators A_i in the same way. For image processing problems, the operators A_i are differential operators in the x_i directions. Thus this scheme will treat all the spatial variables in a symmetrical way and avoid the artifacts produced by treating the spatial variables in nonsymmetrical ways.

The following sequential scheme, sometimes also called the multiplicative operator splitting (MOS) scheme can also be used to approximate the solution of (1):

$$\frac{\phi^{j+\frac{i}{m}} - \phi^{j+\frac{i-1}{m}}}{\tau} + A_i(\phi^{j+\frac{i}{m}}) = f_i(t_j), \quad i = 1, 2, \cdots, m. \quad (5)$$

The above scheme uses different initial values for the A_i operators and thus must be computed sequentially for $i = 1, 2, \cdots, m$. This scheme also has the first order convergence as stated in (4). Both schemes (3) and (5) are absolutely stable for some differential operators, see [14, 11].

In case the equation (1) has a steady state, then the steady state satisfies

$$A(\phi) = f. \quad (6)$$

Both schemes (3) and (5) can be used to compute the solution of (6). However, the parameter τ should not be regarded as a time step, but as a relaxation parameter which can be taken large. For the stability and convergence analysis for using (3) and (5) for solving equation (6), we refer to [12, 13]. Some other algorithms which have the same advantages as scheme (3) for solving equation (6) were also proposed in [12, 13].

In order to be consistent with the notations of [15], the schemes (3) and (5) can also be written in a more general form. For given $v \in V$ and operators A_i (linear or nonlinear), let w_i be the solutions of

$$\frac{\partial w_i}{\partial t} + A_i(w_i) = f_i(t_j), \quad w_i(0) = v, \quad i = 1, 2, \cdots, m. \quad (7)$$

Denote the mapping from v to $w_i(t)$ as $T_{i,j}(t)$, i.e. $w_i(t) = T_{i,j}(t)v$. Then the parallel scheme (3) is a discretized version of the following iteration:

$$\phi^{j+1} = \frac{1}{m}\sum_{i=1}^{m} T_{i,j}(m\tau)\phi^j, \quad \phi^0 = \hat{\phi}. \tag{8}$$

The schemes (5) is a discretized version of the following iteration:

$$\phi^{j+1} = T_{m,j}(\tau) \circ \cdots \circ T_{2,j}(\tau) \circ T_{1,j}(\tau)\phi^j, \quad \phi^0 = \hat{\phi}. \tag{9}$$

3 The MBO scheme

Merriman, Bence, and Osher introduced a very interesting scheme to approximate the motion of an interface by its mean curvature [15]. Suppose we wish to follow an interface moving with a normal velocity equal to its mean curvature. The MBO scheme for the case of two regions is given as an algorithm below:

Algorithm 1 *(MBO scheme for two regions)*
Choose initial value $\phi(0) = \pm 1$ *and the time step* τ. *For* $n = 0, 1, 2, \cdots$ *and* $t_n = n\tau$,

- *Solve* $\tilde{\phi}(t), t \in [t_n, t_{n+1}]$ *from*

$$\tilde{\phi}_t = \Delta\tilde{\phi}, \qquad \tilde{\phi}(t_n) = \phi(t_n) \text{ in } \Omega, \quad \frac{\partial\tilde{\phi}}{\partial n} = 0 \text{ on } \partial\Omega. \tag{10}$$

- *Set*

$$\phi(t_{n+1}) = \begin{cases} -1 & \text{if } \tilde{\phi}(t_{n+1}) < 0, \\ 1 & \text{if } \tilde{\phi}(t_{n+1}) \geq 0. \end{cases} \tag{11}$$

In the original paper [15], the phase function is taking values 0 or 1. Here we use ± 1 to be consistent with our notation. To apply the above scheme for mean curvature motion of multiphase symmetric junctions, one just needs to use multiple phase functions $\phi_i, i = 1, 2, \cdots r$ and project the largest value of ϕ_i to 1 and the others to -1 (See [19]).

The connection between the MBO scheme and the splitting algorithm is revealed in [6, 7, 17, 18, 8] by interpreting it as a phase field method. Let u be the solution of

$$u_t = \epsilon\Delta u - \frac{1}{\epsilon}W'(u), \tag{12}$$

with $W(s) = (s^2 - 1)^2/2$. It is known that the rescaled solution $u(x, \frac{t}{\epsilon})$ is the solution of the mean curvature motion in the limit when $\epsilon \to 0^+$, c.f. [7, 17, 18].

If we use the splitting scheme (9) to solve (12), we would need to solve the following two equations on $[t_n, t_{n+1}]$:

$$a)\ \phi_t = \epsilon\Delta\phi, \qquad b)\ \phi_t = -\frac{1}{\epsilon}W'(\phi). \tag{13}$$

The rescaled solution $\phi(x, t_n/\epsilon)$ of (13.a) is exactly the solution of (10). When $\epsilon \to 0^+$, the rescaled solution $\phi(x, t_n/\epsilon)$ of (13.b) has three values, i.e. 1, 0, -1. We drop the nonstable solution 0 and get (11). This splitting scheme bears some of the natures of the algorithm that have been analysed in [11].

4 The binary levelset method

The binary level set method was originally introduced in [9]. To introduce the main idea, let us first assume that the interface is enclosing $\Omega_1 \subset \Omega \subset R^d$. By standard level set methods the interior of Ω_1 is represented by points $\vec{x} : \phi(\vec{x}) > 0$, and the exterior of Ω_1 is represented by points $\vec{x} : \phi(\vec{x}) < 0$. We instead use a discontinuous level set function ϕ, with $\phi(\vec{x}) = 1$ if $\vec{x}$ is an interior point of Ω_1 and $\phi(\vec{x}) = -1$ if $\vec{x}$ is an exterior point of Ω_1, i.e.

$$\phi(\vec{x}) = \begin{cases} 1 & \text{if } \vec{x} \in \text{int}(\Omega_1), \\ -1 & \text{if } \vec{x} \in \text{ext}(\Omega_1). \end{cases} \tag{14}$$

Thus Γ is implicitly defined as the discontinuity of ϕ. This representation can be used for various applications where subdomains need to be identified. We shall use this idea for image segmentation. Let us assume that u_0 is an image consisting of two distinct regions Ω_1 and Ω_2, and that we want to construct a piecewise constant approximation u to u_0. Let $u(\vec{x}) = c_1$ in Ω_1, and $u(\vec{x}) = c_2$ in Ω_2. If $\phi(\vec{x}) = 1$ in Ω_1, and $\phi(\vec{x}) = -1$ in Ω_2, u can be written as the sum

$$u = \frac{c_1}{2}(\phi + 1) - \frac{c_2}{2}(\phi - 1). \tag{15}$$

The formula (15) can be generalized to represent functions with more than two constant values by using multiple functions $\{\phi_i\}$ following the essential ideas of the level set formulation used in [4, 23]. A function having four constant values can be associated with two level set functions $\{\phi_i\}_{i=1}^2$ satisfying $\phi_i^2 = 1$. More precisely, a function given as

$$\begin{aligned} u = & \tfrac{c_1}{4}(\phi_1 + 1)(\phi_2 + 1) - \tfrac{c_2}{4}(\phi_1 + 1)(\phi_2 - 1) \\ & - \tfrac{c_3}{4}(\phi_1 - 1)(\phi_2 + 1) + \tfrac{c_4}{4}(\phi_1 - 1)(\phi_2 - 1), \end{aligned} \tag{16}$$

is a piecewise constant function of the form

$$u(\vec{x}) = \begin{cases} c_1, & \text{if } \phi_1(\vec{x}) = 1, \quad \phi_2(\vec{x}) = 1, \\ c_2, & \text{if } \phi_1(\vec{x}) = 1, \quad \phi_2(\vec{x}) = -1, \\ c_3, & \text{if } \phi_1(\vec{x}) = -1, \quad \phi_2(\vec{x}) = 1, \\ c_4, & \text{if } \phi_1(\vec{x}) = -1, \quad \phi_2(\vec{x}) = -1. \end{cases}$$

Introducing basis functions ψ_i as in the following

$$\begin{aligned} u =& c_1 \underbrace{\frac{1}{4}(\phi_1+1)(\phi_2+1)}_{\psi_1} \\ &+ c_2 \underbrace{(-1)\frac{1}{4}(\phi_1+1)(\phi_2-1)}_{\psi_2} + \cdots , \end{aligned} \tag{17}$$

we see that u can be written as

$$u = \sum_{i=1}^{4} c_i \psi_i. \tag{18}$$

For more general cases, we can use N level set functions to represent 2^N phases. To simplify notation, we define the vectors $\vec{\phi} = \{\phi_1, \phi_2, \ldots, \phi_N\}$ and $\vec{c} = \{c_1, c_2, \ldots, c_{2^N}\}$. For $i = 1, 2, \ldots, 2^N$, let $(b_1^{i-1}, b_2^{i-1}, \ldots, b_N^{i-1})$ be the binary representation of $i-1$, where $b_j^{i-1} = 0 \vee 1$. Furthermore, set

$$s(i) = \sum_{j=1}^{N} b_j^{i-1}, \tag{19}$$

and write ψ_i as the product

$$\psi_i = \frac{(-1)^{s(i)}}{2^N} \prod_{j=1}^{N} (\phi_j + 1 - 2b_j^{i-1}). \tag{20}$$

Then a function u having 2^N constant values can be written as the weighted sum

$$u = \sum_{i=1}^{2^N} c_i \psi_i. \tag{21}$$

If the level set functions ϕ_i satisfy $\phi_i^2 = 1$ and ψ_i are defined as in (18) or (20), then $\mathrm{supp}(\psi_i) = \Omega_i$, $\psi_i = 1$ in Ω_i, and $\mathrm{supp}(\psi_i) \cap \mathrm{supp}(\psi_j) = \emptyset$ when $j \neq i$. This ensures non-overlapping phases, and in addition $\bigcup_i \mathrm{supp}(\psi_i) = \Omega$, which prevents vacuums. It is clear that ψ_i is the characteristic function of the set Ω_i.

If the level set functions satisfy $\phi_i^2 = 1$, then we can use the basis functions ψ_i to calculate the length of the boundary of Ω_i and the area inside Ω_i, i.e.

$$|\partial\Omega_i| = \int_\Omega |\nabla \psi_i| dx, \qquad \text{and} \qquad |\Omega_i| = \int_\Omega \psi_i dx. \tag{22}$$

The first equality of (22) shows that the length of the boundary of Ω_i equals the total variation of ψ_i. See [27] for more explanations about the total variations of functions that might have discontinuities. In numerical computations, we use the approximation

$$\int_\Omega |\nabla\psi_i| dx \doteq \int_\Omega \sqrt{|\nabla\psi_i|^2 + \epsilon}\, dx, \tag{23}$$

for a small ϵ and the gradient $\nabla\psi$ is approximated by forward finite differences except at some part of the boundary where a backward finite difference is used.

We have now introduced a way to represent a piecewise constant function u by using the binary level set functions. Based on this we propose to minimize the following Mumford-Shah functional to find a segmentation of a given image u_0 [16]:

$$F_{ms}(\vec{\phi}, \vec{c}) = \frac{1}{2}\int_\Omega | u - u_0 |^2 dx + \beta \sum_{i=1}^{2^N} \int_\Omega |\nabla\psi_i| dx. \tag{24}$$

In the above, β is a nonnegative parameter controlling the regularizing, u is a piecewise constant function depending on $\vec{\phi}$ and $\vec{c}$, as in (21). The first term of (24) is a least square functional, measuring how well the piecewise constant image u approximates u_0. The second term is a regularizer measuring the length of the edges in the image u_0. It is easy to see that

$$c_1(N) \sum_{i=1}^{N} \int_\Omega |\nabla\phi| dx \le \sum_{i=1}^{2^N} \int_\Omega |\nabla\psi_i| dx \le c_2(N) \sum_{i=1}^{N} \int_\Omega |\nabla\phi| dx, \tag{25}$$

where $c_1(N)$ and $c_2(N)$ only depend on N. Thus, we can replace the regularization term by an equivalent one and get the following simplified cost functional:

$$F(\vec{\phi}, \vec{c}) = \frac{1}{2}\int_\Omega | u - u_0 |^2 dx + \beta \sum_{i=1}^{N} \int_\Omega |\nabla\phi_i| dx. \tag{26}$$

Considering the constraints imposed on the level set functions, we find that the segmentation problem is the following constrained minimization problem

$$\boxed{\min_{\vec{\phi}, \vec{c}} F(\vec{\phi}, \vec{c}), \quad \text{subject to} \quad \phi_i^2 = 1, \forall\, i.} \tag{27}$$

Recall that $\vec{\phi}$ is a vector having N elements ϕ_i. For notational simplicity, we introduce a vector $\vec{K}(\vec{\phi})$ of the same dimension as $\vec{\phi}$ with $K_i(\vec{\phi}) = \phi_i^2 - 1$. It is easy to see that

$$\phi_i^2 = 1, \forall i \Leftrightarrow \vec{K}(\vec{\phi}) = \vec{0}. \tag{28}$$

In the next section, we try to use the MBO scheme to solve this minimization problem and point out a relationship between our scheme and the scheme of [6].

5 Relation between the MBO and the binary levelset method

In order to make the relation clear, we shall consider the two dimensional two-phase model here, that is, we need to solve

$$\min_{\phi,\,\vec{c}} F(\phi,\vec{c}), \quad \text{subject to} \quad \phi^2 = 1 \tag{29}$$

for $N = 1$ and $m = 2$. The minimization functional in this case is:

$$F(\phi,\vec{c}) = \frac{1}{2}\int_{\Omega} |\, u(\phi,\vec{c}) - u_0 \,|^2 dx + \beta \int_{\Omega} |\nabla\phi| dx. \tag{30}$$

For the two-phase problem, we have

$$\psi_1 = \frac{1}{2}(1-\phi), \qquad \psi_2 = \frac{1}{2}(1+\phi). \tag{31}$$

In case that $\phi^2 = 1$, we can use relation (31) to show that the minimization functional of (33) is exactly

$$F_\mu(\phi,\vec{c}) = \frac{1}{2}\int_{\Omega(\phi=1)} |c_1 - u_0\,|^2 dx + \frac{1}{2}\int_{\Omega(\phi=-1)} |\, c_2 - u_0\,|^2 dx + \beta\int_{\Omega} |\nabla\phi| dx. \tag{32}$$

If we use a penalization term to tackle the constraint $\phi^2 = 1$, we need to choose a small μ and solve:

$$\min_{\phi,\,\vec{c}} F_\mu(\phi,\vec{c}), \tag{33}$$

where

$$F_\mu(\phi,\,\vec{c}) = \frac{1}{2}\int_{\Omega} |\, u(\phi,\,\vec{c}) - u_0\,|^2 dx + \beta\int_{\Omega} |\nabla\phi| dx + \frac{1}{\mu}\int_{\Omega} (\phi^2-1)^2 dx, \tag{34}$$

where $u(\phi,\vec{c}) = c_1\psi_1(\phi) + c_2\psi_2(\phi)$. In order to find a minimizer for (33), we shall find $\vec{c}$ and ϕ that satisfy

$$a)\ \frac{\partial F_\mu}{\partial \vec{c}} = 0, \qquad b)\ \frac{\partial F_\mu}{\partial \phi} = 0. \tag{35}$$

As u is linear with respect to $\vec{c}$, we see that F_μ is quadratic with respect $\vec{c}$. For a given ϕ^n, the minimizer of F_μ with respect to $\vec{c}$ satisfies

$$\sum_{j=1}^{2}\int_{\Omega} \psi_i(\phi^n)\psi_j(\phi^n) c_i^n = \int_{\Omega} u_0\psi_i(\phi^n), \quad \forall i, i = 1, 2. \tag{36}$$

For a fixed $\vec{c}$, the steepest decent method in ϕ for the energy functional (34) gives the following equation for the level set function ϕ:

$$\phi_t = \beta\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - (u(\phi,\,\vec{c}) - u_0)\frac{\partial u}{\partial\phi} - \frac{1}{\mu}W'(\phi), \tag{37}$$

with boundary condition

$$\frac{\nabla\phi}{|\nabla\phi|}\cdot\vec{n}=0 \text{ on } \partial\Omega.$$

To use the sequential splitting scheme (5) to compute ϕ for a given $\vec{c}^n$, we choose a $\tau>0$ and an initial value ϕ^0 and then solve

$$\frac{\phi^{n+1/2}-\phi^n}{\tau}=\frac{\partial F}{\partial\phi}(\phi^{n+1/2},\,\vec{c}^n). \tag{38}$$

Afterwards, we need to solve

$$\frac{\phi^{n+1}-\phi^{n+1/2}}{\tau}=-\frac{1}{\mu}W'(\phi^{n+1}). \tag{39}$$

For the parallel splitting scheme (3), we will need to solve

$$\frac{\phi^{n+1/4}-\phi^n}{2\tau}=-\frac{1}{\mu}W'(\phi^{n+1/4}). \tag{40}$$

and

$$\frac{\phi^{n+1/2}-\phi^n}{2\tau}=\frac{\partial F}{\partial\phi}(\phi^{n+1/2},\,\vec{c}^n). \tag{41}$$

In the end, we set

$$\phi^{n+1}=\frac{1}{2}(\phi^{n+1/4}+\phi^{n+1/2}).$$

For simplicity, we define the function for the MBO projection to be

$$\mathcal{P}(x)=\begin{cases}1 & \text{if } x\geq 0\\ -1 & \text{if } x<0\end{cases}. \tag{42}$$

If we replace the solving of (39) and (40) by the MBO projection, we get the following two algorithms.

Algorithm 2 *(Sequential MBO scheme) For* $n=0,1,2,\cdots$

- *Solve* $\phi^{n+1/2}$ *from*

$$\frac{\phi^{n+1/2}-\phi^n}{\tau}=\beta\nabla\cdot\left(\frac{\nabla\phi^{n+1/2}}{|\nabla\phi^{n+1/2}|}\right)-(u(\phi^{n+1/2},\vec{c}^n)-u_0)\frac{\partial u}{\partial\phi}(\phi^{n+1/2},\vec{c}^n). \tag{43}$$

- *Set*

$$\phi^{n+1}=\mathcal{P}(\phi^{n+1/2}). \tag{44}$$

- *Compute* $\vec{c}^n$ *from (36).*

Algorithm 3 *(AOS MBO scheme) For* $n=0,1,2,\cdots$

- *Solve* $\phi^{n+1/2}$ *from*

$$\frac{\phi^{n+1/2}-\phi^n}{2\tau} = \beta\nabla\cdot\left(\frac{\nabla\phi^{n+1/2}}{|\nabla\phi^{n+1/2}|}\right) - (u(\phi^{n+1/2},\vec{c}^n)-u_0)\frac{\partial u}{\partial\phi}(\phi^{n+1/2},\vec{c}^n). \tag{45}$$

- *Set*

$$\phi^{n+1} = (\mathcal{P}(\phi^n)+\phi^{n+1/2})/2. \tag{46}$$

- *Compute* $\vec{c}^n$ *from (36).*

For the above two algorithms, we may not need to update the values for c^n at each iteration.

If we replace the total variation regularization term $\int_\Omega |\nabla\phi|dx$ by $\int_\Omega |\nabla\phi|^2 dx$ (which is not suggested though), we need to replace the curvature term $\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$ by $\Delta\phi$ (i.e the Laplacian of ϕ) in all the algorithms. If we do this for Algorithm 2, we will get essentially the same algorithms of [6]. For clarity, we write this scheme in the following:

Algorithm 4 *(The scheme of [6]) For* $n = 0, 1, 2, \cdots$

- *Solve* $\phi^{n+1/2}$ *from*

$$\frac{\phi^{n+1/2}-\phi^n}{\tau} = \beta\Delta\phi^{n+1/2} - (u(\phi^{n+1/2},\vec{c}^n)-u_0)\frac{\partial u}{\partial\phi}(\phi^{n+1/2},\vec{c}^n). \tag{47}$$

- *Set*

$$\phi^{n+1} = \mathcal{P}(\phi^n). \tag{48}$$

When we need to identify more than two subdomains, we need to use multiple level set functions ϕ_i. The iterations for the multiple level set functions are essentially the same as for the one level set function case. The interplay between the different level set functions are through the values of $u(\vec{\phi})$ which depends on all the level set functions.

6 A Piecewise Constant Level Set Method

The binary level set method presented in §4 needs to use more than one level set function ϕ when more than two phases are needed in the segmentation. Here we shall introduce a method that just needs one level set function to represent multiphase segmentation. This idea was originally introduced in [10]. Assume that we need to find N regions $\{\Omega_i\}_{i=1}^N$ which form a partition of Ω. In order to find the regions, we want to find a piecewise constant function which takes values

$$\phi = i \text{ in } \Omega_i, \quad i = 1, 2, \ldots, N. \tag{49}$$

With this approach we just need one function to identify all the phases in Ω. The basis functions ψ_i associated with ϕ are defined in the following form:

$$\psi_i = \frac{1}{\alpha_i} \prod_{\substack{j=1 \\ j \neq i}}^{N} (\phi - j) \text{ and } \alpha_i = \prod_{\substack{k=1 \\ k \neq i}}^{N} (i - k). \tag{50}$$

It is clear that the function u given by $u = \sum c_i \psi_i$ is a piecewise constant function and $u = c_i$ in Ω_i if ϕ is as in (49). The function u is a polynomial of order $N-1$ in ϕ. Each ψ_i is expressed as a product of linear factors of the form $(\phi - j)$, with the ith factor omitted. Thereupon $\psi_i(\mathbf{x})=1$ for $\mathbf{x} \in \Omega_i$, and $\psi_i(\mathbf{x})$ equals zero elsewhere as long as (49) holds.

To ensure that different values of ϕ should correspond to different function values of $u(\phi, \vec{c})$ at convergence, we introduce

$$K(\phi) = (\phi - 1)(\phi - 2) \cdots (\phi - N) = \prod_{i=1}^{N} (\phi - i). \tag{51}$$

If a given function $\phi : \Omega \mapsto R$ satisfies

$$K(\phi) = 0, \tag{52}$$

there exists a unique $i \in \{1, 2, \ldots, N\}$ for every $x \in \Omega$ such that $\phi(x) = i$. Thus, each point $x \in \Omega$ can belong to one and only one phase if $K(\phi) = 0$. The constraint (52) is used to guarantee that there is no vacuum and overlap between the different phases. In [26] some other constraints for the classical level set methods were used to avoid vacuum and overlap.

In order to segment a given image, we shall solve the following constrained minimization problem:

$$\min_{K(\phi)=0} \frac{1}{2} \int_{\Omega} |\, u(\phi, \vec{c}) - u_0 \,|^2 dx + \beta \int_{\Omega} |\nabla \phi| dx. \tag{53}$$

Above, $u(\phi, \vec{c}) = \sum c_i \psi_i$ and ψ_i are given as in (50). The minimization variables are ϕ and $\vec{c}$. In [10], the length of the subdomain boundaries were used as the regularization term. Here we replace the regularization term by the total variation of ϕ which is equivalent to the regularization term up to a constant, c.f. (25).

For the new level set method, the function $W(\phi)$ is defined as $W(\phi) = |K(\phi)|^2$. If we use a penalization method to deal with the constraint $K(\phi) = 0$, then the penalization functional for this case will be:

$$F_\mu(\phi, \vec{c}) = \frac{1}{2} \int_{\Omega} |\, u(\phi, \vec{c}) - u_0 \,|^2 dx + \beta \int_{\Omega} |\nabla \phi| dx + \frac{1}{\mu} \int_{\Omega} |K(\phi)|^2 dx. \tag{54}$$

If we split $A(\phi) = \frac{\partial F_\mu}{\partial \phi}$ into a sum of $B(\phi) = \frac{\partial F}{\partial \phi}$, $C(\phi) = \frac{1}{\mu} W'(\phi)$ and use the splitting schemes and MBO projections for such a splitting, we will get two algorithms for this piecewise constant level set method. We will omit the details of these algorithms as they look rather similar to Algorithms 2 and 3. The only difference is the MBO projection. For the level set method presented in this section, the MBO projection is given by:

$$\mathcal{P}(x) = \begin{cases} 1 & \text{if } x \leq 1.5 \\ i & \text{if } x \in (i-0.5, i+0.5] \\ N & \text{if } x > N-0.5 \end{cases} . \quad (55)$$

In order to further simplify the computations, we shall split B into a sum of

$$B_i(\phi) = D_i\left(\frac{D_i\phi}{|\nabla\phi|}\right) + \frac{1}{d}(u(\phi,\vec{c}) - u_0)\frac{\partial u}{\partial \phi}(\phi,\vec{c}), \quad i = 1, 2, \cdots d.$$

Above D_i denotes the partial derivative with respect to x_i and d is the dimension of Ω, i.e. $\Omega \subset R^d$. We see that

$$A = B_1 + B_2 + \cdots B_d + C.$$

If we use scheme (3) for such a splitting, we will get the following algorithm if the penalization is replaced by the MBO projection (55).

Algorithm 5 *(Dimensional AOS MBO scheme) For* $n = 0, 1, 2, \cdots$

- *Solve* $\phi^{n+i/2d}$ *in parallel for* $i = 1, 2, \cdots d$ *from*

$$\frac{\phi^{n+i/2d} - \phi^n}{d\tau} = \beta D_i \cdot \left(\frac{D_i\phi^{n+i/2d}}{|\nabla\phi^{n+i/2d}|}\right) - \frac{1}{d}(u(\phi^{n+i/2d},\vec{c}^n) - u_0)\frac{\partial u}{\partial \phi}(\phi^{n+i/2d},\vec{c}^n). \quad (56)$$

- *Set*

$$\phi^{n+1} = \frac{1}{d+1}\left(\mathcal{P}(\phi^n) + \sum_{i=1}^{d} \phi^{n+i/2d}\right). \quad (57)$$

- *Compute* $\vec{c}^n$ *from (36).*

However, if we solve the subproblems associated with the operators B_i by the parallel splitting scheme (3) and do the MBO projection in a sequential fashion, then the algorithm will look like:

Algorithm 6 *(Dimensional sequential MBO scheme) All the other steps are the same as for Algorithm 5, only replace the MBO projection by:*

- *Set*

$$\phi^{n+1} = \mathcal{P}\left(\frac{1}{d}\sum_{i=1}^{d} \phi^{n+i/2d}\right). \quad (58)$$

Normally, Algorithm 6 is faster than Algorithm 5. Due to the fact that all the dimensional variables are treated in a symmetrical manner, it avoids the artifacts of the dimensional variables. Each subproblem is a one dimensional problem on the lines parallel to the axes and they can be solved efficiently using exact solver for tri-diagonal matrices [12, 13, 24]. We have tested all the proposed algorithms, and it was found that Algorithm 6 combined with the level set method of §6 is the favorable due to it efficiency and simplicity to implement.

7 Numerical experiments

It should be noted that the basis functions ψ_i and the MBO projection operator $\mathcal{P}$ are different for the binary level set method of section §4 and the piecewise constant level set method of section §6. Thus one should use the correct forms in the different algorithms.

For Algorithms 2 and 3, the subproblems (43) and (45) are nonlinear. We use the following Picard iteration to solve these nonlinear equations:

$$\frac{\phi^{new} - \phi^n}{m\tau} = \beta\nabla \cdot \left(\frac{\nabla\phi^{new}}{|\nabla\phi^{old}|}\right) - (u(\phi^{old}, \vec{c}^n) - u_0)\frac{\partial u}{\partial \phi}(\phi^{old}, \vec{c}^n). \tag{59}$$

Normally, we start with the initial value $\phi^{old} = \phi^n$. A CG method can be used to get a ϕ^{new} through the above linear equation. We use this ϕ^{new} as the initial values again and get another ϕ^{new} to be used as initial value. We do a fixed number of iterations for the Picard process. In all the experiments shown later, this iteration number is even set to be 1.

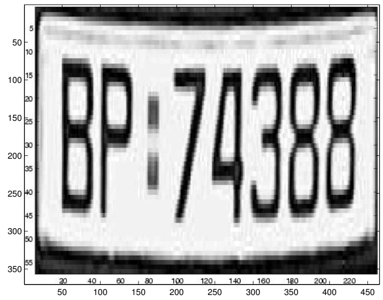
We use the same strategy to solve the nonlinear equation (56), i.e.

$$\frac{\phi_i^{new} - \phi^n}{m\tau} = \beta D_i \cdot \left(\frac{D_i\phi_i^{new}}{|\nabla\phi_i^{old}|}\right) - \frac{1}{d}(u(\phi_i^{old}, \vec{c}^n) - u_0)\frac{\partial u}{\partial \phi}(\phi_i^{old}, \vec{c}^n). \tag{60}$$

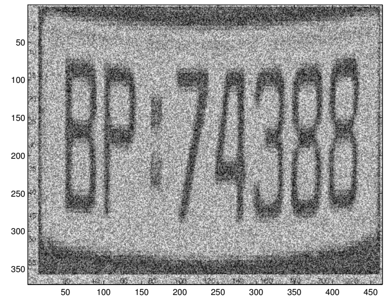
Similar to the solving of (59), we use ϕ^n as the initial value to get a ϕ_i^{new} through the above equation, and then use this ϕ_i^{new} again as the initial value to get another ϕ_i^{new} to be used as the initial value again. As for (59), we just do one such iteration in all the experiments shown later. Note that the equation (60) reduces to some one dimensional equations in the x_i-direction and thus can be efficiently solved by direct solvers for tri-diagonal matrices [12, 13, 24]. Moreover, all these one dimensional problems can be solved in parallel.

We validate and compare the (dimensional) *Sequential MBO scheme*, Alg. 6, and the (dimensional) *AOS MBO scheme*, Alg. 5. We consider only two-dimensional cases and restrict ourself to gray-scale images, but the schemes can handle any dimension and can be extended to vector-valued images as well. Synthesized images, natural images and an MR image are evaluated. The original image is known for some cases which we evaluate her. Thereupon it is trivial to find the perfect segmentation result. To complicate such a segmentation process we typically expose the original image with Gaussian distributed noise and use the polluted image as the observation data u_0. In all examples the iteration is terminated when the relative change in the levelset function ϕ in L^2-norm is less then 10^{-8}. All tests are run on a 2.8GHz Pentium 4 processor.

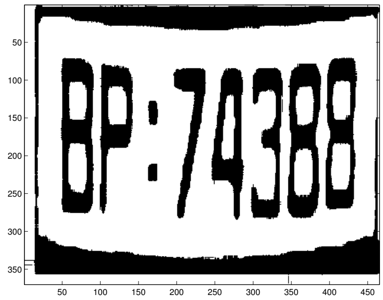
In the first example we illustrate a 2-phase segmentation on a real car plate image. Locating and reading car plates is a well known problem, and there are a number of commercial software available. Below we demonstrate that the *Sequential MBO scheme* and the *AOS MBO scheme* can be used for this segmentation. We challenge these two segmentation techniques by adding Gaussian distributed noise to the real image and use the polluted image in Fig. 1(b) as the observation data. The difficult part is to find the optimal choice for τ and



(a) Real image of a car plate.



(b) Input noise image (SNR$\approx$ 1.7).

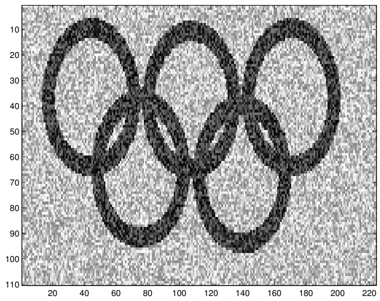


(c) Segmented using sequential MBO scheme. $\tau = 0.5$ and $\beta = 6$.

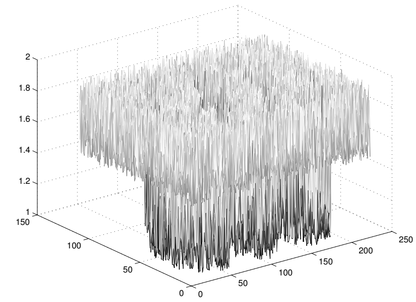
(d) Segmented using AOS MBO scheme. $\tau = 0.5$ and $\beta = 3$.

Figure 1: Character and number segmentation from a car plate.

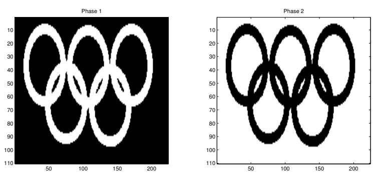
β, and we observe that the choice differs for the two methods. Both methods perform well, see Fig. 1(c) and Fig. 1(d). However, with this amount of noise we miss some details along the edges for the characters and numbers, even though we have large regularization parameters. The values we have used are $\beta = 6$ for the *Sequential MBO* scheme and $\beta = 3$ for the *AOS MBO* scheme. For the *Sequential MBO* scheme the number of iterations is 15, and the CPU time is 67 seconds. For the *AOS MBO* scheme the number of iterations is 27, and the CPU time is 116 seconds. The *Sequential MBO* is the faster one in all our results.



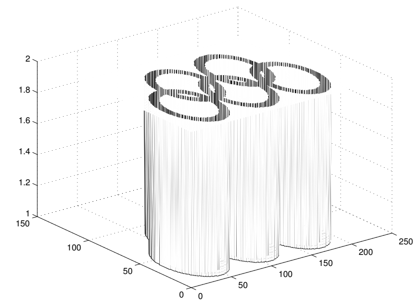
(a) Observed image u_0 (SNR$\approx$ 2.3).



(b) Initial level set function ϕ.



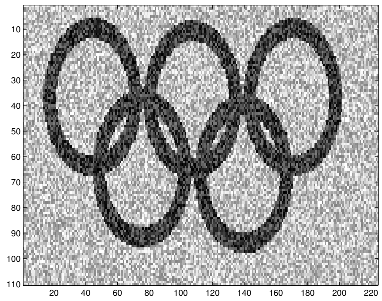
(c) Different phases using sequential scheme.



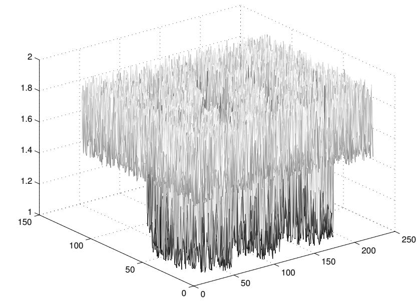
(d) At convergence ϕ approaches 2 constant values.

Figure 2: Segmentation of the Olympic rings using the sequential MBO scheme. $\tau = 0.5, \beta = 0.08$

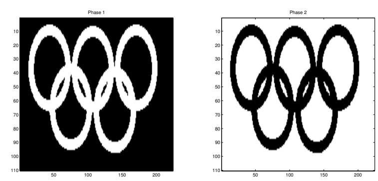
In the next example we illustrate a 2-phase segmentation on a noisy synthetic image containing 5 rings as shown in Fig. 2(a). The image is segmented using both the *Sequential MBO scheme* and the *AOS MBO scheme*. The results are shown in Fig. 2 and Fig. 3 respectively. For the *Sequential MBO* scheme the number of iterations is 2, and the CPU time is 1 second. For the *AOS MBO* scheme the number of iterations is 9, and the CPU time is 4.2 seconds. The input data u_0 is given in Fig. 2(a) and Fig. 3(a). In Fig. 2(d) and Fig. 3(d) the ϕ functions are depicted at convergence. ϕ approaches the predetermined constants $\phi = 1 \vee 2$. Each of these constants represents one unique phase as seen in Fig. 2(c) and Fig. 3(c).



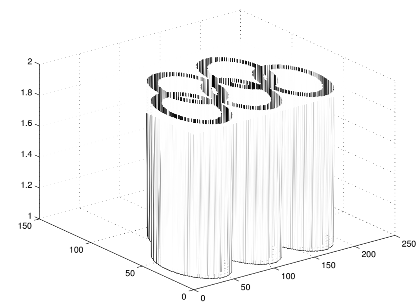
(a) Observed image u_0 (SNR$\approx$ 2.3).



(b) Initial level set function ϕ.



(c) Different phases using sequential scheme.



(d) At convergence ϕ approaches 2 constant values.

Figure 3: Segmentation of the Olympic rings using the AOS MBO scheme. $\tau = 0.5, \beta = 0.08$

The sequential MBO scheme has some artifacts caused by dimensional splitting. However, the artifacts is nearly noticeable by human eyes. Fig. 4 shows the difference between the image segmented by the *Sequential MBO scheme* and the image segmented by the *AOS MBO scheme*.

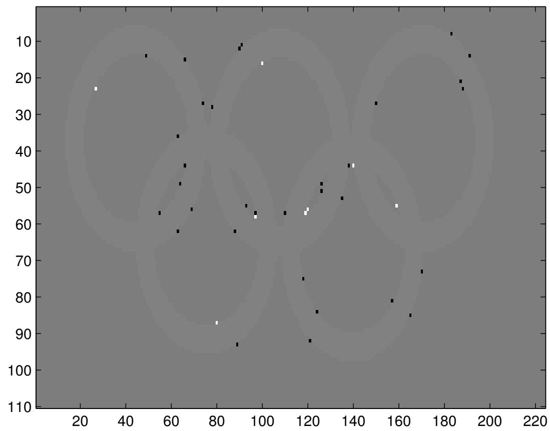


Figure 4: Difference image between the image segmented by the *Sequential MBO scheme* and the image segmented by the *AOS MBO scheme*.

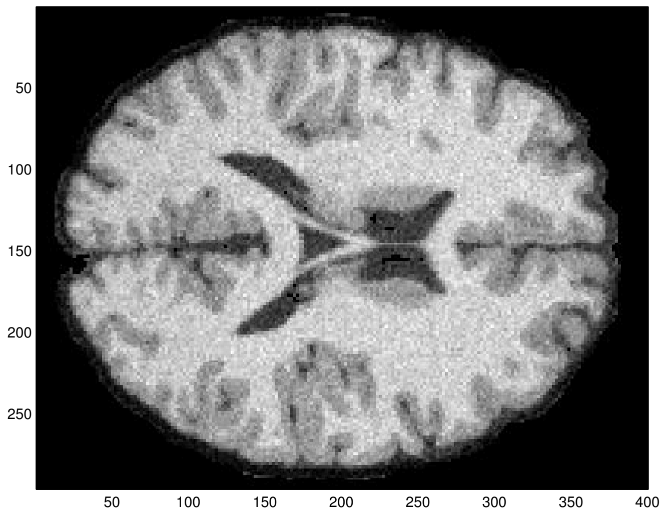
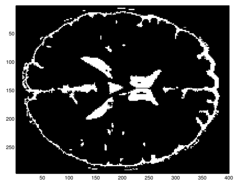
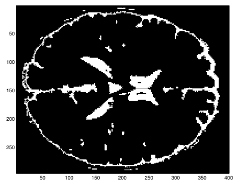


Figure 5: MRI image with a change in the intensity values going from left to right caused by the non-uniform RF-puls.

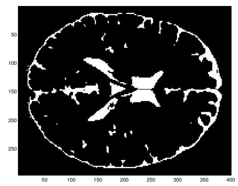
In our next example segmentation of an MR image is demonstrated. The image in Fig. 5 is available to the public at *http://www.bic.mni.mcgill.ca/brainweb/*. These realistic MRI data are used by the neuroimaging community to evaluate the performance of various image analysis methods in a setting where 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 %, for details concerning the noise level percent and the intensity level see *http://www.bic.mni.mcgill.ca/brainweb/*. Both the *Sequential MBO scheme* and the *AOS MBO scheme* are used to segment the MRI phantom and the results are depicted in Fig. 6. There are three tissue classes that should be identified; phase 1: cerebrospinal fluid, phase 2: gray matter, phase 3: white matter. Because of this, 4-phase segmentation was used, but we do not depict the background phase here. We have used $\beta = 0.52, \tau = 0.7$ in the *Sequential* and $\beta = 0.19, \tau = 0.7$ in the *AOS* scheme. For the *Sequential MBO* scheme the number of iterations is 13, and the CPU time is 42 seconds. For the *AOS MBO* scheme the number of iterations is 26, and the CPU time is 82 seconds. The time step has large impact on the regularization.



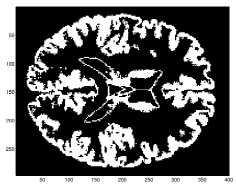
(a) Sequential: phase 1.



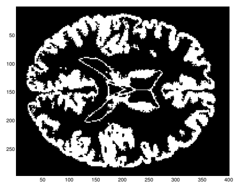
(b) AOS: phase 1.



(c) Exact: phase 1.

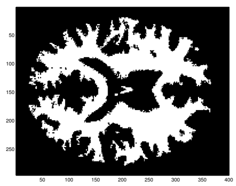


(d) Sequential: phase 2.

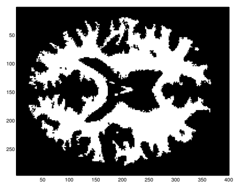


(e) AOS: phase 2.

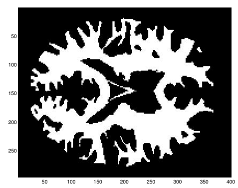
(f) Exact: phase 2.



(g) Sequential: phase 3.



(h) AOS: phase 3.



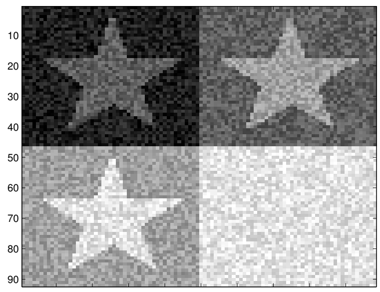
(i) Exact: phase 3.

Figure 6: Segmentation of MRI phantom using *Sequential MBO scheme* and *AOS MBO scheme*. Last column shows exact segmentation.

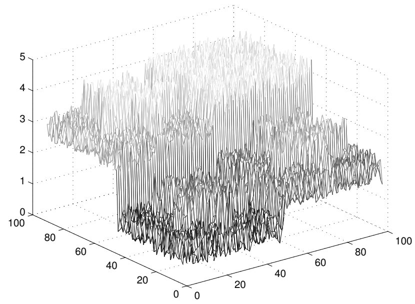
In Fig. 7 we show the results from a 4-phase segmentation of a star image, using the *Sequential MBO scheme* and the *AOS MBO scheme*, respectively. In both cases $\beta = 0.1$, and $\tau = 1$.

For the *Sequential MBO* scheme the number of iterations is 2, and the CPU time is 0.32 seconds. For the *AOS MBO* scheme the number of iterations is 4, and the CPU time is 0.49 seconds.

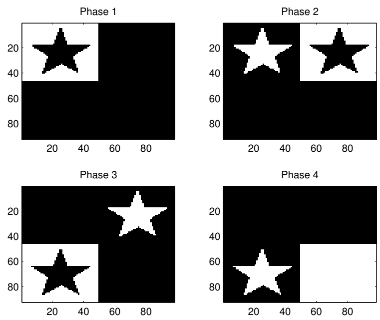
In this example, both the τ and β that give the best result were much easier to find than for the MR image.



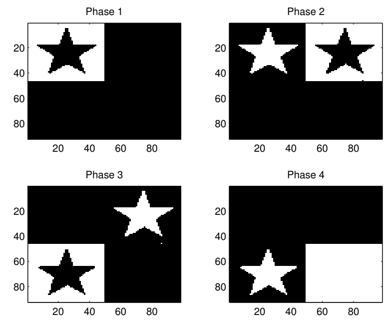
(a) Observed image u_0 (SNR$\approx$ 14.4).



(b) Initial levelset function.



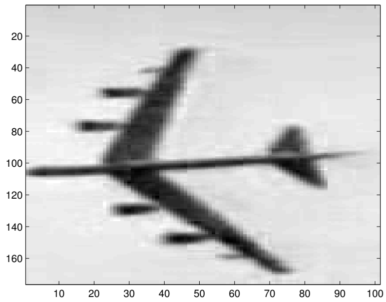
(c) The four phases from Sequential MBO. CPU time: 0.32 sec.



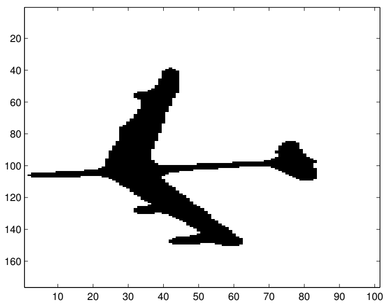
(d) The four phases rom AOS MBO. CPU time: 0.49 sec.

Figure 7: 4-phase segmentation using Sequential and AOS MBO scheme.

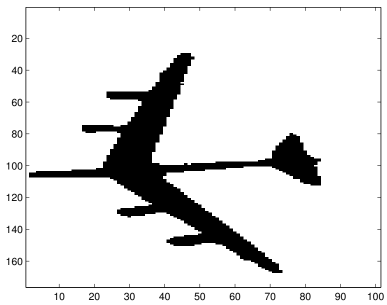
As mentioned, the MBO scheme is very sensitive to the regularization parameter β and the time step τ. In some cases a large β is needed in order to keep the boundary of a phase smooth, and Fig. 8 shows the effect from a small change in τ. Here $\beta = 30$. The difference in τ in Fig. 8(b) and (c) is very small, yet it leads to quite different results. Because of the sensitivity to the time step illustrated here, finding the good parameters sometimes requires quite an effort. In these cases we reach convergence after 7 iterations taking 3.2 seconds resulting in Fig. 8(b) and 10 iterations taking 4.6 seconds, Fig. 8 (c). Fig. 9 illustrates the same, but here we have added noise to the input image. Convergence is reached after 9 iterations taking 4.1 seconds, Fig. 9(b), and 15 iterations taking 6.9 seconds, Fig. 9(c).



(a) Input picture

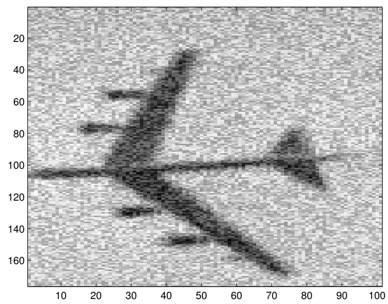


(b) Segmented image u, $\tau = 0.03$.

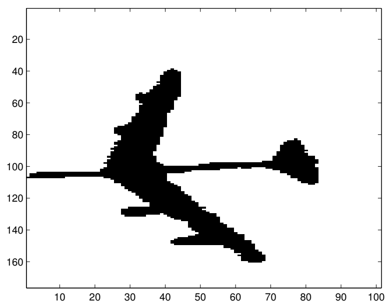


(c) Segmented image u, $\tau = 0.01$.

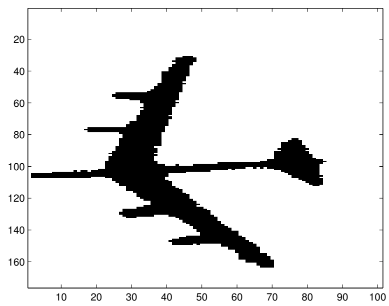
Figure 8: A small difference in the timestep τ results in very different segmentations.



(a) Input image (SNR≈ 3.4).



(b) Segmented image u, $\tau = 0.03$.



(c) Segmented image u, $\tau = 0.01$.

Figure 9: A small difference in the timestep τ results in very different segmentations.

8 Conclusions and remarks

In this work, we propose to combine the MBO scheme of [15] with the piecewise constant level set methods of [9, 10]. Numerical experiments show the success of these schemes. The scheme combining the binary level set method of [9] with the MBO scheme [15] is rather similar to the scheme proposed in [6], see §4. The schemes using the piecewise constant level set method of [10] and the MBO scheme of [15] in a fashion with the AOS or MOS seem to be new compared with other proposed schemes. The numerical experiments show that these schemes are fast and give good results. Note that only one level set function is needed to segment any number of phases. The schemes are rather sensitive to the choice of the time step τ, some further researches need to be done in order to find a systematical strategy to choose the time step for different applications.

References

[1] Gilles Aubert and Pierre Kornprobst. *Mathematical problems in image processing*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2002. Partial differential equations and the calculus of variations, With a foreword by Olivier Faugeras.

[2] Danny Barash. Nonlinear diffusion filtering on extended neighborhood. *Appl. Numer. Math.*, 52(1):1–11, 2005.

[3] Danny Barash, Tamar Schlick, Moshe Israeli, and Ron Kimmel. Multiplicative operator splittings in nonlinear diffusion: from spatial splitting to multiple timesteps. *J. Math. Imaging Vision*, 19(1):33–48, 2003.

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

[5] Qiang Du, Chun Liu, and Xiaoqiang Wang. A phase field approach in the numerical study of the elastic bending energy for vesicle membranes. *J. Comput. Phys.*, 198(2):450–468, 2004.

[6] Selim Esedoglu and Yen-Hsi Richard Tsai. Threshold dynamics for the piecewise constant mumford-shah functional. Tech. Rep. CAM04-63, UCLA Dep. Math, 2004.

[7] L. C. Evans, H. M. Soner, and P. E. Souganidis. Phase transitions and generalized motion by mean curvature. *Comm. Pure Appl. Math.*, 45(9):1097–1123, 1992.

[8] Roland Glowinski, Ping Lin, and Xingbin Pan. An operator-splitting method for a liquid crystal model. *Comput. Physics Comm.*, 152:242–252, 2003.

[9] J. Lie, M. Lysaker, and X.-C. Tai. A binary level set model and some applications to image processing. *UCLA CAM 04-31*, 2004.

[10] Johan Lie, Marius Lysaker, and Xue.-Cheng. Tai. A variant of the levelset method and applications to image segmentation. *UCLA CAM03-50*, 2003.

[11] P.-L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16(6):964–979, 1979.

[12] T. Lu, P. Neittaanmaki, and X-C Tai. A parallel splitting up method and its application to navier-stoke equations. *Applied Mathematics Letters*, 4:25–29, 1991.

[13] T. Lu, P. Neittaanmaki, and X-C Tai. A parallel splitting up method for partial differential equations and its application to navier-stokes equations. *RAIRO Math. Model. and Numer. Anal.*, 26:673–708, 1992.

[14] G. I. Marchuk. Splitting and alternating direction methods. In *Handbook of numerical analysis, Vol. I*, Handb. Numer. Anal., I, pages 197–462. North-Holland, Amsterdam, 1990.

[15] Barry Merriman, James K. Bence, and Stanley J. Osher. Motion of multiple functions: a level set approach. *J. Comput. Phys.*, 112(2):334–363, 1994.

[16] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42(5):577–685, 1989.

[17] Jacob Rubinstein, Peter Sternberg, and Joseph B. Keller. Fast reaction, slow diffusion, and curve shortening. *SIAM J. Appl. Math.*, 49(1):116–133, 1989.

[18] Jacob Rubinstein, Peter Sternberg, and Joseph B. Keller. Front interaction and nonhomogeneous equilibria for tristable reaction-diffusion equations. *SIAM J. Appl. Math.*, 53(6):1669–1685, 1993.

[19] Steven J. Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. *J. Comput. Phys.*, 144(2):603–625, 1998.

[20] C. Samson, L. Blanc-Feraud, G. Aubert, and J. Zerubia. A level set model for image classification. *IJCV*, 40(3):187–198, 2000.

[21] C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia. A variational model for image classification and restoration. *TPAMI*, 22(5):460–472, may 2000.

[22] Gabriele Steidl, Joachim Weickert, Thomas Brox, Pavel Mrázek, and Martin Welk. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides. *SIAM J. Numer. Anal.*, 42(2):686–713 (electronic), 2004.

[23] Luminita A. Vese and Tony F. Chan. A new multiphase level set framework for image segmentation via the Mumford and Shah model. *International Journal of Computer Vision*, 50:271–293, 2002.

[24] J. Weickert, B. H. Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE TRans. Image Process.*, 7:398–409, 1998.

[25] Joachim Weickert and Gerald Kühne. Fast methods for implicit active contour models. In *Geometric level set methods in imaging, vision, and graphics*, pages 43–57. Springer, New York, 2003.

[26] Hong-Kai Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127(1):179–195, 1996.

[27] W. P. Ziemer. *Weakly differentiable functions.* Springer-Verlag, 1989.