DOMAIN DECOMPOSITION METHODS WITH GRAPH CUTS ALGORITHMS FOR IMAGE SEGMENTATION

XUE-CHENG TAI* AND YUPING DUAN†

Abstract. Recently, it is shown that graph cuts algorithms can be used to solve some variational image restoration problems, especially connected with noise removal and segmentation. For very large size problems, the cost for memory and computation increase dramatically. We propose a domain decomposition method with graph cuts algorithms. We show that the new approach is cost effective both for memory and computation. Experiments with large size 2D and 3D data are supplied to show the efficiency of the algorithms.

Key words. multiphase Mumford-Shah, graph cuts, image segmentation, domain decomposition

1. Introduction. Segmentation is one of the fundamental tasks for image processing. Mumford and Shah model [26] is an efficient tool for region based image segmentation. This model is robust to noise and can segment objects without edges. However, the minimization problem is difficult to solve numerically.

The level set method [11, 27] was first introduced to solve the Mumfod-Shah functional by Chan and Vese in [7, 34]. In [24, 25, 30], some variants of the level set method, so-called "Piecewise Constant Level Set Method" (PCLSM), was introduced. This method can identify several interfaces by one single level set function, which makes it easier to solve the Mumford-Shah model.

Traditionally, methods based on gradient descent are often used for solving the Mumford-Shah models, see [24, 25, 30]. These methods are normally slow and difficult to find global minimizers. Recently, a lot of work have been done on applying graph cuts algorithms for image segmentation [3, 4, 18, 12, 22]. It is more efficient for solving this kind of minimization problem. The connection of graph cuts and variational problems has been established in [2, 5, 10, 19]. For Mumford-Shah segmentation, some work using graph cuts optimization for two-phase Mumford-Shah model has been done in [9] and [13]. For multiphase problems, the method of [1, 10, 13, 23] can be adapted to image segmentation. In this work, we shall follow the approach given in [1]. In [1], the authors have extended the graph cuts idea of [10, 19, 20] to the multiphases Mumford-Shah segmentation and it is more suitable for practical applications. However, when the images become large and the number of phases increases, both computational cost and memory usage are greatly increased. In this work we try to find some remedies for these difficulties and show that we could get some algorithms which has quite high efficiency as well as low memory usage. We propose a method combining the domain decomposition method with graph cuts algorithms.

The paper is organized as follows. Section 2, we review the PCLSM and its applications to the Mumford-Shah model. Section 3, we review the graph cuts algorithm of [1] to the multiphase Mumford-Shah model. In Section 4, we combine the domain decomposition methods with this graph cuts idea to solve the Mumford-Shah model. Some implementation detailed are supplied in Section 5. Finally, in Section 6, we

*Division of Mathematical Science, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore and Department of Mathematics, University of Bergen, Johannes Brunsgate 12, N-5008 Bergen, Norway. `tai@mi.uib.no`

†Division of Mathematical Science, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. `DUAN0010@ntu.edu.sg.`

carry out some experiments by our method and compare the results with the original graph cuts algorithm.

2. Mumford-Shah model with PCLSM.

2.1. Mumford-Shah model. The Mumford-Shah model is a well known model for image segmentation problem [26]. In the model, Ω is a bounded domain and $u^0(x)$ is the input image. We search for a pair (u, Γ_i) through the following minimization problem:

$$E(u,\Gamma_i) = \int_\Omega (u-u^0)^2 dx + \mu \int_{\Omega\setminus\bigcup_i \Gamma_i} |\nabla u|^2 dx + \sum_{i=1}^n \gamma \int_{\Gamma_i} ds \tag{2.1}$$

where μ and γ are nonnegative constants and $\int_{\Gamma_i} ds$ is the length of the boundary of interfaces Γ_i. The most popular way to solve this minimization problem is applying the level set method [7], especially for the piecewise constant Mumford-Shah model. For such cases, the second term is vanished in the minimization functional.

2.2. Piecewise constant level set method. In [24, 25, 30], the piecewise constant level set method (PCLSM) was proposed and applied to the Mumford-Shah model. The main idea of PCLSM is to seek a partition of the domain Ω into n subdomains Ω_i, $i = 1, 2, \cdots, n$. The essential idea is to use a piecewise constant level set function ϕ to identify the subdomains.

$$\phi = i \quad in \quad \Omega_i. \tag{2.2}$$

Once the function ϕ is identified, we can construct the corresponding characteristic functions for each subdomains Ω_i as

$$\psi_i = \frac{1}{\alpha_i} \prod_{j=1, j\neq i}^n (\phi - j), \quad \alpha_i = \prod_{k=1,k\neq i}^n (i-k). \tag{2.3}$$

If ϕ is defined as in (2.2), we can have $\psi_i(x) = 1$ for $x \in \Omega_i$, otherwise we have $\psi_i(x) = 0$. Based on these characteristic functions, we can extract the geometrical information of the boundaries of the subdomains Ω_i. For example, the length of the interfaces surrounding each subdomains Ω_i should be

$$Length(\partial\Omega_i) = \int_\Omega |\nabla(\psi_i)|. \tag{2.4}$$

For some given values $c_i, i = 1, 2, \cdots n$, define

$$u = \sum_{i=1}^n c_i \psi_i. \tag{2.5}$$

We have $u = c_i$ in the corresponding subdomain Ω_i, if ϕ satisfies (2.2). In the next subsection, we shall use this idea for image segmentation with the Mumford-Shah model.

2.3. The minimization probelm. Assume u is a piecewise constant function as given in (2.5). The multiphases piecewise constant Mumford-Shah model is to solve the following minimization problem:

$$\min_{\mathbf{c}\in\mathbb{R}^n, \phi\in\{1,2,\cdots,n\}} E(\mathbf{c},\phi), \quad E(\mathbf{c},\phi) = \int_\Omega (u(\mathbf{c},\phi) - u^0)^2 dx + \frac{\gamma}{2}\sum_{i=1}^n \int_\Omega |\nabla\psi_i| dx \tag{2.6}$$

We use total variation (TV) of the characteristic function to replace the last term of the Mumford-Shah functional, measuring the length of the interfaces. Such an approach has also been used in other segmentation models in [8, 21]. It is easy to see that

$$\phi = \sum_{i=1}^{n} i\psi_i(\phi), \quad \nabla\psi_i = \psi_i'(\phi)\nabla\phi.$$

Thus, there exist two constants $\alpha_1(n) > 0, \alpha_2(n) > 0$, such that

$$\alpha_1(n)\int_\Omega |\nabla\phi|dx \le \sum_{i=1}^{n}\int_\Omega \psi_i(\phi)|dx \le \alpha_2(n)\int_\Omega |\nabla\phi|dx \tag{2.7}$$

Unless "symmetry" is a crucial issue for the segmentation problem, we replace the regularization term in $E(\mathbf{c}, \phi)$ by an equivalent functional and solve the following minimization problem:

$$\min_{\mathbf{c}\in\mathbb{R}^n, \phi\in\{1,2,\cdots,n\}} E(\mathbf{c},\phi), \quad E(\mathbf{c},\phi) = \int_\Omega (u-u^0)^2dx + \gamma\int_\Omega |\nabla\phi|dx. \tag{2.8}$$

This functional is the Mumford-Shah model we used in the paper. In [24, 25, 30], the constrained optimization problem (2.8) was solved by finding the saddle point of the corresponding augmented Lagrangian functional. In these methods, some iterative numerical methods are used to solve the corresponding Euler-Lagrange equations, such as gradient decent time marching scheme. In the next section, we shall construct a graph and solve minimization problem (2.8) by the graph cuts algorithms as in [1].

3. Graph cuts for multiphase Mumford-Shah Model. Instead of solving the Euler-Larange equation, graph cuts algorithms have been proposed in [1] to solve minimization problem (2.8). We give a review of this algorithm in the following.

3.1. Background on graph cuts. The graph cuts algorithm is an established powerful method to minimize certain kinds of energy functional. A directed capacitated graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a set of vertices $\mathcal{V}$ and directed edges $\mathcal{E}$. There are two special vertices in the graph, i.e. the source s and the sink t. A cut on graph $\mathcal{G}$ partitions the vertices into two disjoint groups S and T such that $s \in S$ and $t \in T$. The cost of the cut is the sum of capacities of all edges that go from S to T

$$c(S,T) = \sum_{u\in S, v\in T, (u,v)\in\mathcal{E}} c(u,v). \tag{3.1}$$

We focus on finding a cut with the smallest cost $c(S,T)$, namely the minimal cut. To solve the minimal cut problem, there are mainly two groups of algorithms: Goldberg-Tarjan style "push-relabel" methods [16] and Ford-Fulkerson style "augmenting paths" [15]. In our paper, we use the augmenting paths method [3].

3.2. Discretization of energy functional. Assume we want to segment a $M \times N$ image into $n (n \ge 2)$ phases. Let $\mathcal{P}$ denotes the index set of the pixels, i.e.

$$\mathcal{P} = \{(i,j)|i \in 1,\dots,M, j \in 1,\dots,N\}. \tag{3.2}$$

There are two different ways to discretize the TV term of the functional in (2.8), i.e. isotropic and anisotropic. Since the isotropic total variation is not graph representable, we consider anisotropic discretization of the TV term. The anisotropic discretization depends on the neighbor pixels adopted to represent the TV term. In this paper, we consider 4 and 8 neighbors for $2D$ images, c.f. [2, 10, 17, 13]

$$TV^4(\phi) = \sum_{i,j} |\phi_{i+1,j} - \phi_{i,j}| + |\phi_{i,j+1} - \phi_{i,j}|. \tag{3.3}$$

$$TV^8(\phi) = TV^4(\phi) + \frac{1}{\sqrt{2}} \sum_{i,j} (|\phi_{i+1,j+1} - \phi_{i,j}| + |\phi_{i+1,j-1} - \phi_{i,j}|). \tag{3.4}$$

The data fidelity term can be discretized directly. For a given $p = (i,j) \in \mathcal{P}$, define

$$\mathcal{N}_4(p) = \{(i \pm 1, j), (i, j \pm 1)\} \cap \mathcal{P}, \tag{3.5}$$

$$\mathcal{N}_8(p) = \{(i \pm 1, j), (i, j \pm 1), (i \pm 1, j \pm 1)\} \cap \mathcal{P}. \tag{3.6}$$

Using these notations, the discretization version of (2.8) can be written as

$$E_d(\mathbf{c}, \phi) = \sum_{p \in \mathcal{P}} |u_p - u_p^0|^2 + \gamma \sum_{p \in \mathcal{P}, q \in \mathcal{N}_k(p)} w_{p,q} |\phi_p - \phi_q|. \tag{3.7}$$

Above, $\mathcal{N}_k(p), k = 4, 8$, is defined as in (3.5)-(3.6) and w_{pq} is the corresponding weight for the discretized TV-term as in (3.3) and (3.4), see also [1]. u_p^0 is the intensity value of u_0 at $p \in \mathcal{P}$ and u_p is related to ϕ_p as in (2.5). We assume that the value of $c_i, i = 1, 2, \cdots n$ are known. For boundary points p, $\mathcal{N}_k(p)$ has less neighboring points.

By doing so, the minimization problem is transformed into discrete form which is graph representable. We can get the minimizer of (3.7) using the max-flow / min-cut algorithm using the algorithm of [1].

It is easy to extend the model to 3D problems. For example, we can use the neighborhood involving 6 neighbors for 3D and use the following term as the regularization term:

$$TV^{3D,6}(\phi) = \sum_{i,j,k} (|\phi_{i+1,j,k} - \phi_{i,j,k}| + |\phi_{i,j+1,k} - \phi_{i,j,k}| + |\phi_{i,j,k+1} - \phi_{i,j,k}|)$$

Later, we shall also test on 3D segmentation problems and this regularization term has been used there. We can also add more neighboring points to approximate the length better.

3.3. Graph construction. Recently, the graph of [10, 19, 20] has been extended in [1] to solve the multiphases Mumford-Shah model. In this subsection, we briefly review the essential ideas.

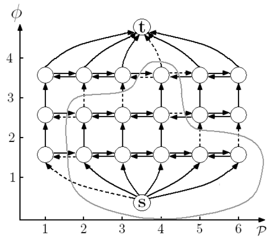
To use graph cuts algorithm for the multiphases segmentation problems, we have to introduce an extra dimension, i.e. we construct graph with one dimension higher than the original image. For a 2D image of size $M \times N$, we construct a graph in $3D$ containing $M \times N \times (n-1)$ vertices. More specifically, we have $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and

$$\mathcal{V} = \left\{ v_{p,l} \mid (p,l) \in \mathbb{R}^2 \times \mathbb{R} |\ p \in \mathcal{P},\ l \in \{1, \ldots, n-1\} \right\}. \tag{3.8}$$

The edges $\mathcal{E}$ are divided into two groups: $\mathcal{E}_D$ coresponds to the data fidelity term in (3.7) and $\mathcal{E}_R$ corresponds to the TV term in (3.7). They are defined, respectively, as

$$\mathcal{E}_D = \cup_{p \in \mathcal{P}} \left\{ (s, v_{p,1}) \cup_{l=1}^{n-2} (v_{p,l}, v_{p,l+1}) \cup (v_{p,n-1}, t) \right\}. \tag{3.9}$$

$$\mathcal{E}_R = \{ (v_{p,l}, v_{q,l}) \quad | \quad p \in \mathcal{P}, q \in \mathcal{N}_k(p), l \in 1, \ldots, n-1 \}. \tag{3.10}$$



(a)

1	3	3	4	2	2

(b)

FIG. 3.1. *(a) The graph corresponds to $1D$ signal of 6 grid points. We construct a 3 level grids for this 4 phase segmentation problem. The gray curve denotes the cut. (b) shows the values of the level set function ϕ at each grid point corresponding to the cut in (a).*

In Fig 3.1, the graph for a $1D$ signal with 4-phase segmentation is shown. The edges in $\mathcal{E}_D$ are illustrated as the vertical arrows while the edges in $\mathcal{E}_R$ are illustrated as the horizontal arrows in Fig 3.1. A cut is called admissible if it only serves one vertical edge for each $p \in \mathcal{P}$, c.f. [1]. In order to exclude non-adimissible cut, we introduce an artificial constant $\sigma > 0$ and define the capacity of the edges as:

$$c(s, v_{p,1}) = |u_p^0 - c_1|^2 + \frac{\sigma}{MN} \quad \forall p \in \mathcal{P}, \tag{3.11}$$

$$c(v_{p,l}, v_{p,l+1}) = |u_p^0 - c_l|^2 + \frac{\sigma}{MN} \quad \forall p \in \mathcal{P}, \quad \forall l \in 1, \dots n-2, \tag{3.12}$$

$$c(v_{p,n}, t) = |u_p^0 - c_n|^2 + \frac{\sigma}{MN} \quad \forall p \in \mathcal{P}. \tag{3.13}$$

$$c(v_{p,l}, v_{q,l}) = \gamma \cdot w_{pq}, \quad \forall p \in \mathcal{P}, \quad \forall q \in \mathcal{N}_k(p), \quad \forall l \in 1, \dots n-1. \tag{3.14}$$

In the above, γ is the regularization parameter, w_{pq} is the weight for the discretization of the TV-norm and $\mathcal{N}_k(p)$ is the set containing the neighbors of $p \in \mathcal{P}$ used in the discretization.

After adding all edges to the graph, we can solve the minimization by using the max-flow / min-cut algorithm. We emphasis that the segmentation problem is transfered from the size of $M \times N$ to the size of $M \times N \times (n-1)$.

3.4. An iterative segmentation scheme. In the last section, we show that graph cuts algorithms can be used to solve the Mumford-Shah minimization problem when the values of $\mathbf{c}$ are known. For minimization problem (2.8), we also need to estimate the $\mathbf{c}$ values and the following algorithms is rather robust and converges fast:

ALGORITHM 3.1. *(Graph cuts segmentation algorithm)*
Choose initial values for $\mathbf{c^0}$, set $l = 0$.
while ($\|\mathbf{c}^l - \mathbf{c}^{l-1}\| > tol$)
1. Use graph cuts to estimate ϕ^{l+1} from

$$\phi^{l+1} = argmin_{\tilde{\phi}} E_d(\mathbf{c}^l, \tilde{\phi}), \tag{3.15}$$

2. Compute the characteristic functions $\{\psi_k^{l+1}\}_{k=1}^n$ from ϕ^{l+1}, c.f. (2.5).

3. Update $\mathbf{c}^{l+1}$ *by*

$$c_k{}^{l+1} = \frac{\sum_{p\in\mathcal{P}} u_p^0 \psi_{k,p}}{\sum_{p\in\mathcal{P}} \psi_{k,p}}, \qquad k = 1, \dots, n. \tag{3.16}$$

4. Update $l + 1 \leftarrow l$.

The initial values $\mathbf{c}^0$ are computed very efficiently by the isodata algorithm, see [33]. For segmentation problems, the above iterative procedure normally converges in about 5-6 iterations. Compared with traditional gradient decent methods, it is normally 500 times faster for relatively large size 2D images we have tested, see [1]. When the size is very large, the memory and computational cost is becoming a challenge problem.

4. Graph cuts algorithms with domain decomposition. As we discussed, when the images become large, the computational and memory cost for the multiphase graph cuts algorithm increases greatly. This is causing problems for some data set with very large sizes, especially in 3D applications. We shall use a domain decomposition method to overcome these difficulties.

Domain decomposition methods is an efficient tool in large-scale computation and has been used to PDE problems [6, 14, 31, 28, 29, 32]. In [32], these techniques have been used for general convex minimization problems.

As was done in [32], the image domain can be decomposed into four regions and then use graph cuts algorithms to solve subproblems over the subdomains. We use Fig 4.1 to illustrate the decomposed subdomains.

Ω_1	Ω_2	Ω_1	Ω_2	Ω_1
Ω_3	Ω_4	Ω_3	Ω_4	Ω_3
Ω_1	Ω_2	Ω_1	Ω_2	Ω_1
Ω_3	Ω_4	Ω_3	Ω_4	Ω_3
Ω_1	Ω_2	Ω_1	Ω_2	Ω_1

FIG. 4.1. *An example of domain decomposition with 25 subdomains*

4.1. Non-overlapping domain decomposition. First, we consider the non-overlapping domain decomposition method. We assume Ω has been decomposed into 4 non-overlapping subdomains. The subdomains intersect only on their interfaces, see Fig. 4.1. We denote $\mathcal{P}_i \subset \mathcal{P}$, $i = 1, 2, 3, 4$, the index sets for the grid points of the subdomains, c.f. (3.2). Corresponding to each subdomain, we define energy functional

$$E_d^i(\mathbf{c}, \phi) = \sum_{p\in\mathcal{P}_i} |u_p - u_p^0|^2 + \gamma \sum_{p\in\mathcal{P}_i, q\in\mathcal{N}_k(p)\cap\mathcal{P}_i} w_{p,q}|\phi_p - \phi_q|. \tag{4.1}$$

The algorithm can be written as follows:

ALGORITHM 4.1. *(non-overlapping domain decomposition)*
Choose initial values for $\mathbf{c^0}$*, set* $l = 0$.
While ($\|\mathbf{c}^l - \mathbf{c}^{l-1}\| > tol$)
1. *For* $i = 1, 2, 3, 4$*, use a graph cuts algorithm to estimate* $\phi^{l+1}_{|\Omega_i}$ *from*

$$\phi^{l+1}_{|\Omega_i} = argmin_{\tilde{\phi}} E^i_d(\mathbf{c}^l, \tilde{\phi}). \tag{4.2}$$

2. *Compute the characteristic functions* $\{\psi^{l+1}_k\}^n_{k=1}$ *from* ϕ^{l+1}*, c.f. (2.5).*
3. *Update* $\mathbf{c}^{l+1}$ *according to the following discrete formula for* $\mathbf{c}$

$$c^{l+1}_k = \frac{\sum_{p\in\mathcal{P}} u^0_p \psi^{l+1}_{k,p}}{\sum_{p\in\mathcal{P}} \psi^{l+1}_{k,p}} \qquad k = 1, \ldots, n. \tag{4.3}$$

4. *Update* $l + 1 \leftarrow l$.

Here and later, we denote $\phi_{|\Omega_i}$ be the value of ϕ in Ω_i. For minimization problem (4.2), we only need to use graph cuts algorithms to find the values of ϕ^{l+1} in Ω_i. Each Ω_i contains many disjoint subdomains, i.e. $\Omega_i = \cap_j \Omega_{i,j}$. As the subprobnlems over $\Omega_{i,j}$ are independent of each other, we can use graph cuts algorithms to solve the subdomain problems simultaneously. If we have parallel computers, these subdomain problems can be solved in parallel. In our implementations, we just solve the problems one by one. Even so, the computational cost is reduced compared to solving the minimal cut problem of graph in the whole domain.

For a given $p \in \mathcal{P}_i$ on the boundary $\partial\Omega_i$ of Ω_i, the subdomain energy functional E^i_d only includes regularization terms related to $q \in \mathcal{N}_k(p) \cap \mathcal{P}_i$, i.e. the subdomain problems only regularize with point inside the subdomain. There is no regularization between the subdomains. Thus, this will cause some errors compared with Algorithm 3.1. Due to the reason that the $c_i, i = 1, 2, \cdots n$ are computed globally, it seems that the algorithm has always been able to find a good segmentation despite this error.

4.2. Overlapping domain decomposition. In the overlapping domain decomposition, the subdomains overlap with each other. Fig 4.2 dispatches the subdomains in our overlapping domain decomposition approach corresponding to the domain decomposition method presented in Fig 4.1. The dashed line denotes the boundary of the subdomains. In overlapping domain decomposition, we use the overlapping parts to influence the cuts of the interior parts in each subdomain. Therefore, the subdomains are no longer independent and have intimate relation with their neighbor subdomains in the segmentation. The overlapping size influence the convergence rate of the iterate process as analysed in [32]. Large overlapping size gives faster convergence for the iteration. However, it also leads to increased cost in solving the subdomain problems. A proper choice of the overlapping size is needed in order to get the best convergence.

As the subdomains have overlaps now, the corresponding index sets $\mathcal{P}_i$ also have overlaps. To explain the algorithm clearly, we need to introduce some notations.

We use Ω^0_i to denote the interior grid points of Ω_i and $\partial\Omega_i$ to denote the boundary grid points of Ω_i. Correspondingly, $\mathcal{P}^0_i$ is the index set for Ω^0_i and $\partial\mathcal{P}_i$ is the index set for $\partial\Omega_i$. Let

$$E^i_d(\mathbf{c}, \phi) = \sum_{p\in\mathcal{P}^0_i} |u_p - u^0_p|^2 + \gamma \sum_{p\in\mathcal{P}^0_i, q\in\mathcal{N}_k(p)} w_{p,q}|\phi_p - \phi_q|. \tag{4.4}$$

The overlapping domain decomposition algorithm is given in the following.

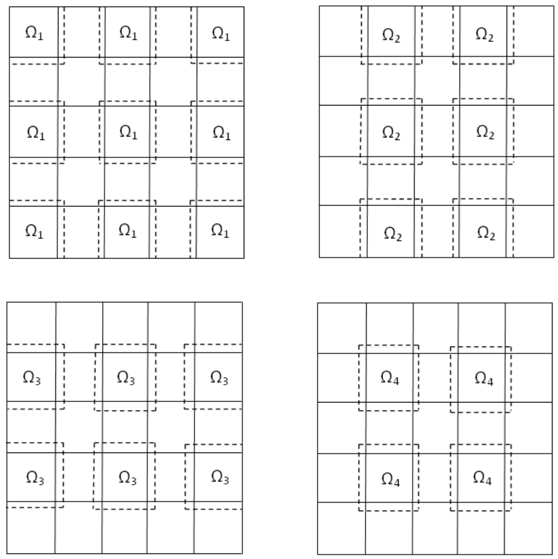


FIG. 4.2. *Four kinds of subdomains in the overlapping domain decomposition*

ALGORITHM 4.2. *(overlapping domain decomposition)*
Choose initial values for $\mathbf{c}^{\mathbf{0}}$ *and* ϕ^0. *Set* $l = 0$.
While ($\|\mathbf{c}^l - \mathbf{c}^{l-1}\| > tol$)

1. *For* $i = 1, 2, 3, 4$, *let* $\phi^{l+\frac{i}{4}} = \phi^{l+\frac{i-1}{4}}$ *in* $\Omega \backslash \Omega_i^0$ *and use a graph cuts algorithm to estimate* $\phi_{|\Omega_i^0}^{l+\frac{i}{4}}$ *from*
$$\phi_{|\Omega_i^0}^{l+\frac{i}{4}} = argmin_{\tilde{\phi}} E_d^i(\mathbf{c}^l, \tilde{\phi}). \tag{4.5}$$
2. *Compute the characteristic functions* $\{\psi_k^{l+1}\}_{k=1}^n$ *from* ϕ^{l+1}, *c.f. (2.5).*
3. *Update* $\mathbf{c}^{l+1}$ *according to the following discrete formula for* $\mathbf{c}$
$$c_k^{l+1} = \frac{\sum_{p\in\mathcal{P}} u_p^0 \psi_{k,p}^{l+1}}{\sum_{p\in\mathcal{P}} \psi_{k,p}^{l+1}} \qquad k = 1, \ldots, n. \tag{4.6}$$
4. *Update* $l + 1 \leftarrow l$.

However, as the subdomains overlap with each other, solving (4.5) is quite different from solving (4.2). The value of $\phi^{l+\frac{i}{4}}$ is equal to $\phi^{l+\frac{i-1}{4}}$ in $\Omega\backslash\Omega_i^0$ and thus have no need for computation. The value of $\phi^{l+\frac{i}{4}}$ in Ω_i^0 need to be solved through (4.5). For a point $p \in \mathcal{P}_i^0$, $\mathcal{N}_k(p)$ may be outside Ω_i^0. However, this does not cause any problem for solving (4.5) as the values outside Ω_i^0 is already known. This will take care of the regularization between the subdomains. We shall comment on the details for the implementation for (4.5) in Section 5

For this algorithm, we have, c.f. [32]

$$\begin{aligned} E_d(\mathbf{c}^{l+1}, \phi^{l+1}) \le E_d(\mathbf{c}^l, \phi^{l+1}) \le E_d(\mathbf{c}^l, \phi^{l+3/4}) &\le E_d(\mathbf{c}^l, \phi^{l+1/2}) \\ &\le E_d(\mathbf{c}^l, \phi^{l+1/4}) \le E_d(\mathbf{c}^l, \phi^l). \end{aligned}$$

This guarantees the monotonicity of the cost functional and thus gives a robust algorithm.

5. Implementation of the algorithms. For the implementation of Algorithm 3.1, we just need to construct the graph defined in (3.8) and (3.9)-(3.10) and then add the capacity (costs) as given in (3.11)-(3.14). Theoretically, any $\sigma > 0$ is enough to guarantee that any minimum cuts is admissible, see [1]. Once the graph is constructed, we use augmenting path algorithm to find the minimum cut.

The implementation of Algorithm 4.1 is also easy. For each subproblem, we construct the graph as we have done for Algorithm 3.1 and use the augmenting path algorithm to solve (4.2). Nearly the same codes used for Algorithm 3.1 can be used for Algorithm 4.1. The only difference is that we need to construct and solve the graph cuts problem over each subdomain instead of on the whole domain Ω.

For Algorithm 4.2, due to the overlapping of the subdomains, some extra care need to be given in solving subdomain problems (4.5). For a given $p \in \mathcal{P}_i^0$, $\mathcal{N}_k(p)$ may be outside Ω_i^0 and these values are known and needed for E_d^i in (4.4). If we take $k = 4$ or $k = 6$ for $\mathcal{N}_k$, then $\mathcal{N}_k(p)$ is alway within $\mathcal{P}_i = \mathcal{P}_i^0 \cup \partial\mathcal{P}_i$ for any $p \in \mathcal{P}_i^0$. Each Ω_i contains many disjoint subdomains, i.e. $\Omega_i = \cap_j \Omega_{i,j}$. As the subprobnlems over $\Omega_{i,j}$ are independent of each other, we can use graph cuts algorithms to solve the subdomain problems simultaneously or one by one. For each subdomain problem, we construct the graph for the subdomain $\Omega_{i,j}$ to include the interior and boundary grid points, i.e. the subdomain graph is

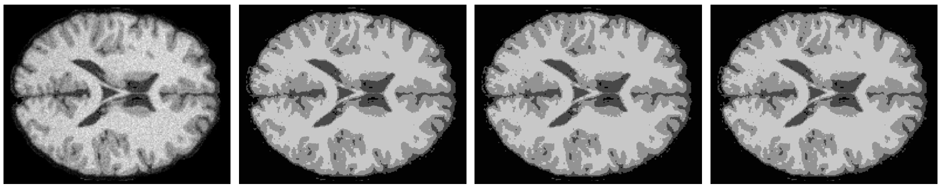
$$
\begin{aligned}
\mathcal{V}^{i,j} &= \left\{ v_{p,l} \mid (p,l) \in \mathbb{R}^2 \times \mathbb{R} |\ p \in \mathcal{P}_{i,j},\ l \in \{1, \dots, n-1\} \right\}. \\
\mathcal{E}^{i,j} &= \mathcal{E}_D^{i,j} \cup \mathcal{E}_R^{i,j}, \\
\mathcal{E}_D^{i,j} &= \cup_{p \in \mathcal{P}_{i,j}} \left\{ (s, v_{p,1}) \cup_{l=1}^{n-2} (v_{p,l}, v_{p,l+1}) \cup (v_{p,n-1}, t) \right\}. \\
\mathcal{E}_R^{i,j} &= \left\{ (v_{p,l}, v_{q,l}) \quad | \quad p \in \mathcal{P}_{i,j}^0, q \in \mathcal{N}_k(p), l \in 1, \dots, n-1 \right\}.
\end{aligned}
$$

In the above, notations $\mathcal{P}_{i,j}$ and $\mathcal{P}_{i,j}^0$ are self explainable. The capacity of the edges for the interior grid points are defined as in (3.11)-(3.14). The boundary value of $\phi^{l+\frac{i}{4}}$ is known as $\phi^{l+\frac{i}{4}} = \phi^{l+\frac{i-1}{4}}$ in $\Omega \backslash \Omega_i^0$. We only need to compute the value of $\phi^{l+\frac{i}{4}}$ in the interior of Ω_i which can be computed in parallel over the subdomains $\Omega_{i,j}$. To keep the boundary values unchanged, the capacity for the edges in $\mathcal{E}_D^{i,j}$ for any $p \in \partial\mathcal{P}_{i,j}$ should be defined as ∞ except one that indicates the value of the point p and the capacity for this edge should be given 0. Comparing with the implementation of Algorithms 3.1-4.1, we only need to set the capacity for the "vertical edges" to be ∞ or 0 for the grid points on the boundary of Ω_i. This is the only extra "care" that we need to take for the implementation of Algorithm 4.2.

In our implementations, we take $tol = 0.1$, $n = 4$ and $\sigma = 4(n-1)\gamma$. The values of γ varies with the examples. For Algorithm 4.2, we alway take $\phi^0 = 0$ and use ISODATA algorithm of [33] to get the initial values for $\mathbf{c}$. The size of overlapping is one pixel unless specified otherwise.

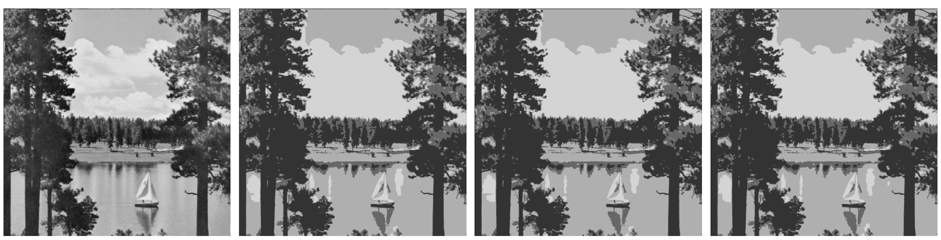
6. Numerical experiments. In the following, we implement our domain decomposition methods on synthetic and real data. We develop our codes in C++ using the augmenting path algorithm introduced in [3]. All numerical experiments were performed on a HP xw4600 Workstation with an Intel(R) Core(TM) 2 Duo CPU E6750 @ 2.66 GHz, 2.67 GHz and 2.00 GB of RAM.

In the first experiment, we implement our method to a real brain MR image of high resolution, shown in Fig 6.1(a). We use 4-phase image segmentation approach

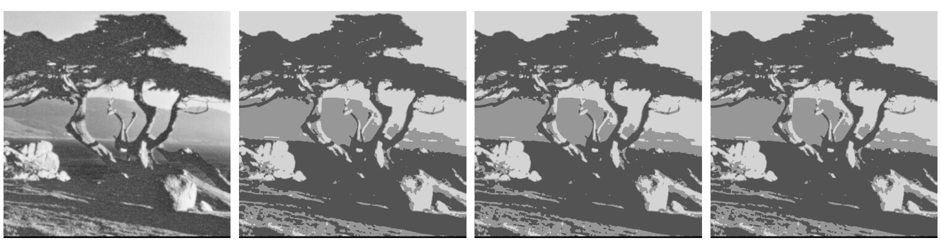


(a) Given image u^0 (b) Orignal alg. (c) Non-over alg. (d) Over alg.

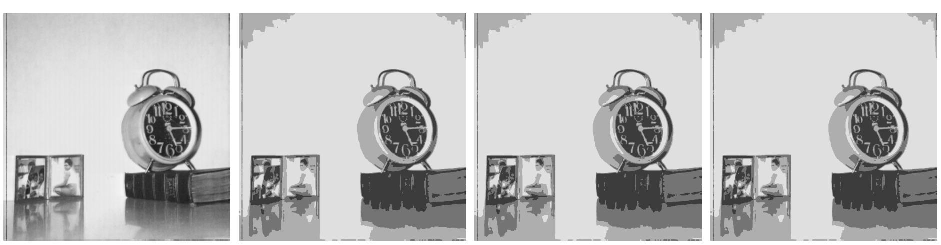
(e) Given image u^0 (f) Orignal alg. (g) Non-over alg. (h) Over alg.



(i) Given image u^0 (j) Orignal alg. (k) Non-over alg. (l) Over alg.



(m) Given imageu^0 (n) Orignal alg. (o) Non-over alg. (p) Over alg.



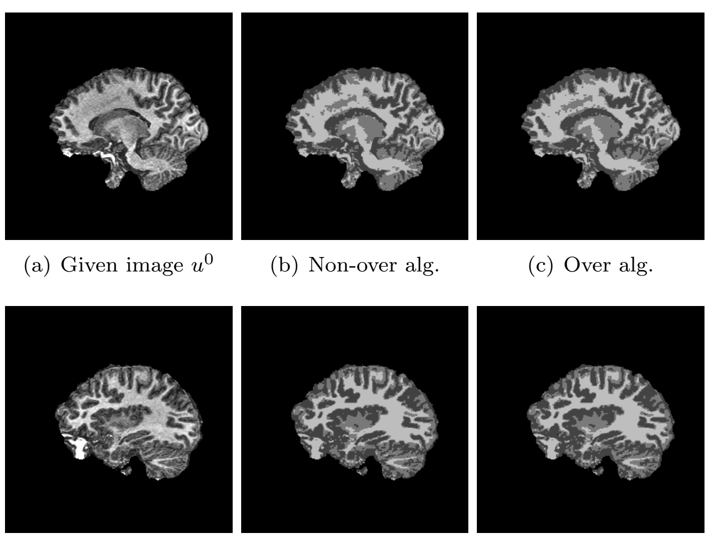
(q) Given image u^0 (r) Orignal alg. (s) Non-over alg. (t) Over alg.

Fig. 6.1. *The comparison result of MR, lena, lake, tree and clock*

to extract the 4 different classes of the brain image. This 4 different classes can be classified as: region 1: background, region 2: cerebrospinal fluid, region 3: gray matter and region 4: white matter. We segment the MR image with the TV^4 norm and the TV^8 norm respectively. The results of the image with the TV^4 norm are shown in the Fig 6.1(b)-(d) and the computation time is shown in the Table 6.1. We can see that our decomposition method can get almost the same result as using the original graph cuts algorithm (Algorithm 3.1) visually. In the meantime, the decomposition methods improve more than $\frac{1}{4}$ of the computation time. For this test, the size of the image is 670×530. We have used $25(5 \times 5)$ subdomains and $\gamma = 800$.

In the second experiment, we apply the domain decomposition methods to four different images. We choose four 1024×1024 images: Lena, Lake, Tree and Clock and use original graph cuts algorithm, non-overlapping decomposition method and overlapping decomposition method on these images respectively. The segmentation results are displayed in Fig 6.1 and the computation time is shown in Table 6.1. To see the superiority of domain decomposition methods, we enlarge the images to the size 2048×2048. For our computer, the original graph cuts cannot handle images of this size. However, our decomposition method can solve the segmentation problem well. Therefore, this decomposition method is useful for large scale images, especially for application to 3D problems. For this test, we have used $128(8 \times 16)$ subdomains and $\gamma = 500$.

In the third experiment, we implement our method on 3D MRI image. The original size of MRI image is $250 \times 250 \times 120$ and it is too large for the graph cuts algorithm (Algorithm 3.1) to handle. In this experiment, we use our method to MRI data and get the segmentation result. We choose two slices of the MRI data and show the comparison images in the Fig 6.2. We also give a comparison of the CPU time with different sizes of data extracted from the 3D MRI data to illustrate the scale of computation time with different sizes of image. In Fig 6.3, the table contains the CPU time of different sizes of images and the figure corresponds to the data in the table. For this test, we have used $1000(10 \times 10 \times 10)$ subdomains and $\gamma = 500$.

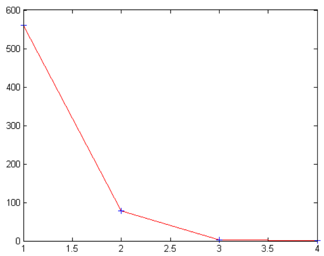


(a) Given image u^0 (b) Non-over alg. (c) Over alg.

(d) Given image u^0 (e) Non-over alg. (f) Over alg.

Fig. 6.2. *The comparison result of MRI. We show the slice nr.: 50 and 80.*

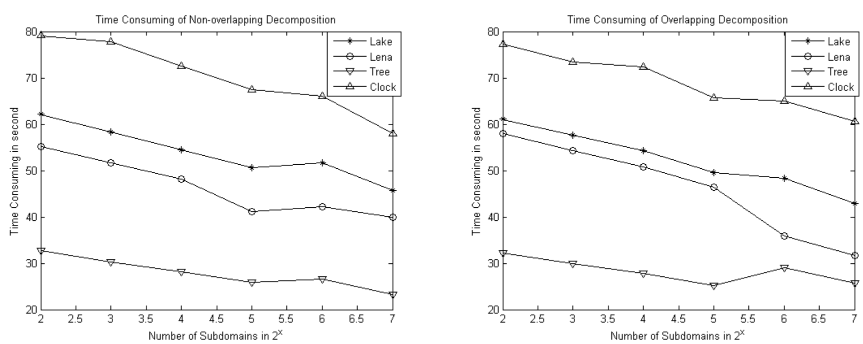
image size	CPU time
$250 \times 250 \times 120$	560.578
$120 \times 120 \times 60$	78.656
$60 \times 60 \times 30$	3.484
$30 \times 30 \times 15$	0.906



(a) (b)

FIG. 6.3. *Computation time in seconds for different resolution. In (b), 1 denotes the extracted image with size of* $250 \times 250 \times 120$*; 2 denotes the extracted image with size of* $120 \times 120 \times 60$*; 3 denotes the extracted image with size of* $60 \times 60 \times 30$*; 4 denotes the extracted image with size of* $30 \times 30 \times 15$.

Next, we compare the computational cost of the domain decomposition methods with different numbers of subdomains. We carry out this experiment on the same four images and decompose the images into 2^2, 2^3, 2^4, 2^5, 2^6, 2^7 subdomains using 2 elements of overlaps. The result of non-overlapping decomposition is shown in Fig 6.4(a) while the result of overlapping decomposition is shown in Fig 6.4(b). Through the results, it is obviously that the computing time decreases as the subdomain size decreases. This experiment illustrates that our domain decomposition methods can improve the speed of the segmentation problem.



(a) (b)

FIG. 6.4. *Time consuming of domain decompositiom methods.*

Besides, we try to illustrate that the energy obtained by domain decomposition methods approximates the energy of the original graph cuts Algorithm 3.1. We test the same four images of size 1024×1024 and decompose them into 128 subdomains. The size of the subdomains is 128×64. The energy of original graph cuts Algorithm 3.1 is marked with "$*$". The energies of non-overlapping and overlapping decomposition methods are calculated using the cut results in the entire domain. For non-overlapping case, we add all the weights of the cut edges on the image domain while we add all the weights of the cut edges of internal nodes in overlapping case. The energies of non-overlapping and overlapping decomposition are denoted by "$+$" and "o" respectively in the figure. The energy figure of each image is shown in Fig 6.5(a)-(d). Through the experiments, we see that the energy of our domain decomposition methods is

TABLE 6.1
Compution time in seconds for different experiments. – denotes problem can not handle by the method.

	Size	neighbor	Original alg	Non-over alg	Overlapping alg
Brain	670×530	4	10.25	7.421	7.531
Brain	670×530	8	22.844	15.562	14.813
Lena	1024×1024	4	51.641	38.797	31.688
Lake	1024×1024	4	57.45	44.657	42.875
Tree	1024×1024	4	29.859	22.594	25.86
Clock	1024×1024	4	74.203	56.438	63.344
MRI	$250 \times 250 \times 120$	6	–	353.282	507.875

convergent and the difference between the energy of domain decomposition methods and the energy of original graph cuts algorithm is acceptable to us.

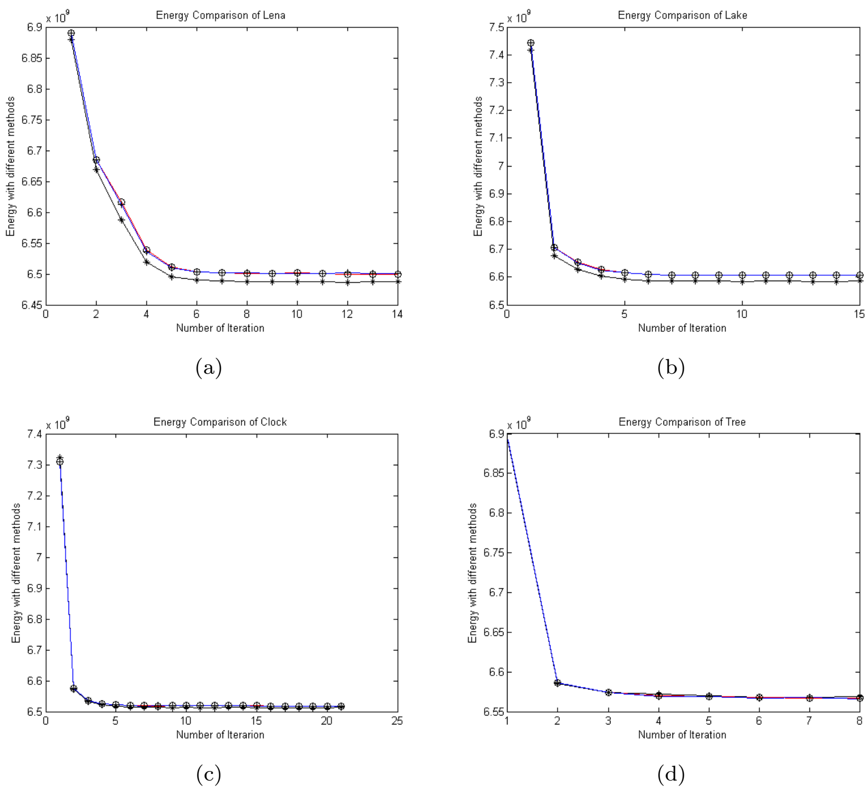


FIG. 6.5. *The energy comparision between original method (black curve), non-overlapping decomposition (blue curve) and overlapping decomposition method (red curve).*

7. Conclusion. In this work, we propose a new method to minimize the Mumford-Shah model with piecewise constant level set representation. We apply the domain decomposition methods to image segmentation and use graph cuts algorithm to minimize the energy functionals. The proposed method improves the computation efficiency. Even more, it greatly reduced the memory costs and enables us to solve very large size problems effectively. Due to the monotonicity property of the algorithms, its numerical performance is very robust. It is remarkable that the algorithm can segment a 3D MRI image with $7 \times 10^6 (250 \times 250 \times 120)$ voxels in just a few minutes and the quality is comparable with traditional variational methods.

Acknowledgments. The authors would like to thank Professor Wenbing Tao for valuable discussions and constructive suggestions. The research has been supported by MOE (Ministry of Education) Tier II project T207N2202 and IDM project NRF2007IDM-IDM002-010. In addition, support from SUG 20/07 is also gratefully acknowledged.

REFERENCES

[1] E. Bae and X.C. Tai. Graph cuts for the multiphase mumford-shah model using piecewise constant level set methods. *UCLA, Applied Mathematics, CAM-report 08-36*, 2008.

[2] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, pages 26–33, 2003.

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[5] A. Chambolle. Total variation minimization and a class of binary MRF models. *Lecture Notes in Computer Science*, 3757:136–152, 2005.

[6] T.F. Chan and T.P. Mathew. Domain decomposition algorithms. *Acta numerica*, 3:61–143, 2008.

[7] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.

[8] G. Chung and L.A. Vese. Energy minimization based segmentation and denoising using a multilayer level set approach. *Lecture notes in computer science*, 3757:439–455, 2005.

[9] J. Darbon. A note on the discrete binary mumford-shah model. *Lecture Notes in Computer Science*, 4418:283–294, 2007.

[10] J. Darbon and M. Sigelle. Image restoration with discrete constrained Total Variation part II: Levelable functions, convex priors and non-convex cases. *Journal of Mathematical Imaging and Vision*, 26(3):277–291, 2006.

[11] A. Dervieux and F. Thomasset. A finite element method for the simulation of Rayleigh-Taylor instability. *Lecture Notes in Mathematics*, 771:145–159, 1979.

[12] N. El-Zehiry and A. Elmaghraby. A graph cut based active contour for multiphase image segmentationl. In *IEEE International Conference on Image Processing (2008)*, pages 3188–3191.

[13] N. El-Zehiry, S. Xu, P. Sahoo, and A. Elmaghraby. Graph cut optimization for the Mumford-Shah model. In *Proc. of the Int. conf. Visualization, Imaging, and Image Processing, Palma de Mallorca, Spain (August 2007)*, pages 182–187.

[14] D. Firsov and S. H. Lui. Domain decomposition methods in image denoising using gaussian curvature. *J. Comput. Appl. Math.*, 193(2):460–473, 2006.

[15] L. Ford and D. Fulkerson. Flows in networks. 1962.

[16] A.V. Goldberg and R.E. Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921–940, 1988.

[17] D. Goldfarb and W. Yin. Parametric maximum flow algorithms for fast total variation minimization. Technical report, Technical report, Rice University, 2007.

[18] DM Greig, BT Porteous, and AH Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 271–279, 1989.

[19] H. Ishikawa. Exact optimization for Markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, 2003.

[20] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998. Proceedings*, pages 125–131, 1998.

[21] Y.M. Jung, S.H. Kang, and J. Shen. Multiphase image segmentation via modica-mortola phase transition. *SIAM Journal on Applied Mathematics*, 67(5):1213–1232, 2007.

[22] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[23] J. Lie, M. Lysaker, and X.C. Tai. Piecewise constant level set methods and image segmentation. In *Scale Space and PDE Methods in Computer Vision: 5th International Conference, Scale-Space*, volume 3459, pages 573–584. Springer, 2005.

[24] J. Lie, M. Lysaker, and X.C. Tai. A binary level set model and some applications to Mumford-Shah image segmentation. *IEEE Transactions on Image Processing*, 15(5):1171–1181, 2006.

[25] J. Lie, M. Lysaker, and X.C. Tai. A variant of the level set method and applications to image segmentation. *Mathematics of computation*, 75(255):1155–1174, 2006.

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

[27] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics*, pages 12–49, 1988.

[28] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*. Oxford University Press, 1999.

[29] B.F. Smith, P. Bjorstad, and W. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge Univ Pr, 2004.

[30] X.C. Tai, O. Christiansen, P. Lin, and I. Skjælaaen. Image segmentation using some piecewise constant level set methods with MBO type of projection. *International Journal of Computer Vision*, 73(1):61–76, 2007.

[31] X.C. Tai and M. Espedal. Rate of convergence of some space decomposition methods for linear and nonlinear problems. *SIAM Journal of Numerical Analysis*, pages 1558–1570, 1998.

[32] X.C. Tai and J. Xu. Global and uniform convergence of subspace correction methods for some convex optimization problems. *Mathematics of Computation*, 71(237):105–124, 2002.

[33] RD Velasco. Thresholding using the ISODATA clustering algorithm. *IEEE TRANS. SYS., MAN, AND CYBER.*, 10(11):771–774, 1980.

[34] 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.