Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2009.m9003s

Vol. **2**, No. 4, pp. 403-420
November 2009

3D Multiphase Piecewise Constant Level Set Method Based on Graph Cut Minimization

Tiril P. Gurholt[1,*] and Xuecheng Tai[1,2]

[1] *Centre for Integrated Petroleum Research (CIPR)/Department of Mathematics, University of Bergen, Allégaten 41, 5007 Bergen, Norway.*
[2] *Division of Mathematical Sciences, Nanyang Technological University, SPMS-04-01, 21 Nanyang Link, Singapore 637371.*

Received 6 March 2009; Accepted (in revised version) 30 April 2009

Abstract. Segmentation of three-dimensional (3D) complicated structures is of great importance for many real applications. In this work we combine *graph cut* minimization method with a variant of the *level set* idea for 3D segmentation based on the *Mumford-Shah* model. Compared with the traditional approach for solving the Euler-Lagrange equation we do not need to solve any partial differential equations. Instead, the *minimum cut* on a special designed graph need to be computed. The method is tested on data with complicated structures. It is rather stable with respect to initial value and the algorithm is nearly parameter free. Experiments show that it can solve large problems much faster than traditional approaches.

AMS subject classifications: 65K10, 49K20, 49K35

Key words: Piecewise constant level set method, energy minimization, graph cut, segmentation, three-dimensional.

1. Introduction

The object of this paper is to extend and test the two-dimensional (2D) graph cut algorithm proposed in [2] for 3D image segmentation. The algorithm is used for minimizing the *piecewise constant level set method* (PCLSM) [16], which is a region based segmentation approach in which object boundaries are detected both with and without gradient information. The PCLSM is minimized by finding the minimum cut on a special graph, thus we need not solve any partial differential equations. This yields an accurate solver, which detects complicated structures. The approach is fast, not very sensitive with respect to initial value and is almost parameter free.

The *level set method* [18] represents evolving interfaces by embedding them in a higher dimensional function ϕ, referred to as the level set function. Traditionally the evolving

*Corresponding author. *Email addresses:* tiril.gurholt@cipr.uib.no (T. P. Gurholt), tai@mi.uib.no, XCTai@ntu.edu.sg (X. Tai)

 ©2009 Global-Science Press

interfaces are represented by the zero level set of ϕ, which is usually given as a signed distance function, and evolves under the influence of forces in the normal direction of ϕ itself. This allows for representing complicated structures which do not depend on the discretization and automatic handling of topological changes in the evolving front, such as merging and splitting, naturally. The level set method is traditionally solved by first deriving an *Euler-Lagrange* equation and then solving the problem using standard numerical schemes. Since its introduction, the method has been widely studied within different fields. It has been applied for image processing problems such as noise removal and segmentation [6, 7, 16, 19, 24].

For region based image segmentation, the *Chan-Vese model* [7] used the level set framework to solve the *Mumford-Shah model* [17] for two-phase problems. It was later extended for segmenting multiphase problems [27] by representing the evolving fronts using several ϕ's. The PCLSM [16] was later introduced as an alternative region based image segmentation approach for multiphase problems. It segments images using a single piecewise constant level set function (PCLSF) ϕ. The method has previously been used for image segmentation by among others [8, 15, 16, 21, 23, 25, 26]. Traditionally the PCLSM is solved in a similar fashion as standard level set approaches, by deriving and solving an Euler-Lagrange equation. This is a time consuming procedure, in particular for larger data-sets. Recently a new method for solving the PCLSM was introduced in [2] for 2D segmentation problems. They employ *graph cut* for solving the problem, which is a graph based approach for fast and accurate minimizations of energy functions. Graph cut problems are solved by finding the *minimum cut* on a graph G, which is equivalent to finding the *maximum flow* on G due to the duality theorem of [11]. It was shown in [13] that the minimum cut/maximum flow algorithms can be used for minimizing certain energy functions in computer vision. Graph cut has previously been used for solving variational problems by among others [3, 5, 9, 10, 12]. In this article we extend the 2D approach of [2] for solving 3D multiphase problems, and demonstrate it for segmentation of real *computed tomography* (CT) data and on synthetic *magnetic resonance* (MR) images of the human brain.

An advantage using the level set method is that the evolving interface is implicitly represented by a higher dimensional function ϕ. Complicated structures are therefore naturally handled, as well as topological changes such as merging and breaking. The level set function ϕ is often a signed distance function, which has to be reinitialized to ensure a well-posed problem. Using the PCLSM no reinitialization is required, since ϕ is represented by a piecewise constant function [16]. Furthermore, multiphase problems are solved using a single ϕ as opposed to the approach of [27] which use several ϕ's. When the PCLSM is minimized by graph cut, the computation time is reduced significantly compared to the original formulation, which is shown in [2]. The minimization algorithm is robust and nearly parameter free. It remains stable for all test problems. The only user input required is an initial guess of the mean phase values $\mathbf{c}$. No initialization of ϕ is required. Numerical tests have shown that it is sufficient to provide a naive initial guess of $\mathbf{c}$, with distinct values that need not be in the vicinity of the true mean phase values. The original PCLSM formulation is sensitive to perturbations and ill-posed problems could

be encountered, and there are several parameters that needs to be chosen properly. It requires a relatively good initial guess of $\mathbf{c}$ [16]. As in the original formulation the number of phases can be overdetermined and some resulting phases can be empty. A disadvantage of the presented method is that the computation time increase significantly with the size of the data-set. More effort is therefore required to develop an even faster solver for very large data-sets. A second disadvantage of the method is that it requires more memory than the original PCLSM formulation. For large data-sets, and/or for problems including a large number of phases, memory problems might be encountered.

This paper is organized as follows. In Section 2 a brief review of related theoretical background is given, covering the Mumford-Shah model, the Chan-Vese model and the PCLSM. In Section 3 graph based minimization of the PCLSM for 3D multiphase problems is examined. In Section 4 we present results of the method used for solving two- and four-phase segmentation problems, and Section 5 concludes the paper.

2. Theoretical background

2.1. The Mumford-Shah model

The *Mumford-Shah model* [17] is an image segmentation model where image boundaries are detected based on regional information in the image. This enables the model to detect image boundaries both with and without gradient information. It seeks to approximate an image u^0 defined over domain Ω by a multiphase function $\tilde{u}$, which is either piecewise smooth or piecewise constant. Given n phases, there are n interfaces Γ_i separating the different phases. The energy functional minimized to find $\tilde{u}$ and Γ_i is given by

$$E(\tilde{u},\Gamma_i)=\int_{\Omega}(u^0-\tilde{u})^2\,d\bar{x}+\mu\int_{\Omega\setminus\Gamma_i}|\nabla\tilde{u}|^2\,d\bar{x}+\nu\sum_{i=1}^{n}\int_{\Gamma_i}ds. \tag{2.1}$$

For many applications, it is sufficient to consider approximation by piecewise constant function $\tilde{u}$. For such minimization problems the second term disappears. In the following only approximation by piecewise constant functions are considered.

2.2. The Chan-Vese model

The *Chan-Vese model* [18], also called *active contour without edges*, represents the *Mumford-Shah* energy functional (2.1) in the level set framework. It is a two-phase segmentation approach that minimize an energy functional using the level set framework, where an image $u^0\in\Omega$ is approximated by a piecewise constant function $\tilde{u}$. The discontinuities of $\tilde{u}$ represent image boundaries, modeled by the zero level set of ϕ. The level set function ϕ is usually given as a signed distance function. The energy functional minimized

is

$$E(\tilde{u},\phi) = \lambda_1 \int_\Omega (u^0-\tilde{u})^2 H(\phi)\, d\bar{x} + \lambda_2 \int_\Omega (u^0-\tilde{u})^2 (1-H(\phi))\, d\bar{x} + \nu \int_\Omega |\nabla H(\phi)| d\bar{x} + \mu \int_\Omega H(\phi) d\bar{x}. \tag{2.2}$$

The two first terms are minimized when the discontinuities of $\tilde{u}$ lies on the object boundary, which is when $\tilde{u}$ is given by

$$\tilde{u} = \begin{cases} c_1, & \text{if } \phi \geq 0, \\ c_2, & \text{if } \phi < 0, \end{cases} \quad \text{for } c_1 = \frac{\int_\Omega H(\phi) u^0\, d\bar{x}}{\int_\Omega H(\phi)\, d\bar{x}}, \;\; c_2 = \frac{\int_\Omega (1-H(\phi)) u^0\, d\bar{x}}{\int_\Omega (1-H(\phi))\, d\bar{x}}. \tag{2.3}$$

The last two terms impose regularization on the evolving interface. The parameters $\lambda_1 > 0$, $\lambda_2 > 0$, $\nu \geq 0$, $\mu \geq 0$ are fixed. H is the *Heaviside* function; $H(\phi) = 1$ for $\phi(\bar{x}) \geq 0$ and $H(\phi) = 0$ for $\phi(\bar{x}) < 0$. The constants c_1 and c_2 are iteratively updated at each time-step. A multiphase extension of the *Chan-Vese* model was presented in [27]. It approximates an image u^0 by a piecewise constant function $\tilde{u}$ using several level set functions ϕ. The zero level set of the various ϕ's represents the discontinuities of $\tilde{u}$.

2.3. The piecewise constant level set method

In [16] the PCLSM was presented as a new method for solving multiphase segmentation problems. It approximates an image u^0 by a piecewise constant function $\tilde{u}$ using a single level set function ϕ. The function ϕ is now given as a *step function*, referred to as the PCLSF, defined over domain Ω which has the same dimensionality as image u^0. For a n-phase segmentation, Ω is partitioned into n subdomains $\{\Omega_i\}_{i=1}^n$ and ϕ is defined by

$$\phi = i \in \Omega_i, \qquad \text{for } i = 1, \cdots, n. \tag{2.4}$$

The discontinuities of ϕ now represent the interfaces instead of the frequently used zero level set of ϕ. For each subdomain Ω_i, a characteristic function ψ_i can be defined as

$$\psi_i(\phi) = \frac{1}{\alpha_i} \prod_{j=1,\, j\neq i}^{n} (\phi - j) \quad \text{with} \quad \alpha_i = \prod_{k=1,\, k\neq i}^{n} (i-k). \tag{2.5}$$

Thus ψ_i is equal to 1 in domain Ω_i and otherwise 0. The boundary length of Ω_i can be found by computing the total variation of ψ_i, and the piecewise constant image function $\tilde{u}$ can be expressed in terms of the characteristic functions,

$$\tilde{u}(\phi) = \sum_{i=1}^{n} c_i \psi_i(\phi), \quad \text{for } c_1, \cdots, c_n \text{ constant.} \tag{2.6}$$

The *Mumford-Shah* energy functional (2.1) can now be rewritten as the PCLSM energy functional

$$E(\mathbf{c},\phi) = \int_\Omega (u^0 - \tilde{u}(\phi))^2 d\bar{x} + \frac{\nu}{2} \sum_{i=1}^{n} \int_\Omega |\nabla \psi_i(\phi)| d\bar{x}. \tag{2.7}$$

Division by 2 of the regularization term corrects for counting each interface twice. An expression for the constants $c_1, \cdots, c_n$ in (2.6) is obtained by minimizing the energy functional for fixed ϕ, which yields

$$c_i = \frac{\int_\Omega u^0 \psi_i(\phi)\, d\bar{x}}{\int_\Omega \psi_i(\phi)^2\, d\bar{x}}, \quad \forall\, i = 1, \cdots, n. \tag{2.8}$$

Thus, c_i is the mean phase value of u^0 for subdomain Ω_i and $\mathbf{c} = \{c_1, \cdots, c_n\}$. Similarly, ϕ is minimized for fixed $\mathbf{c}$.

Minimizing the regularization term for each ψ_i in (2.7) is a time consuming procedure that can significantly slow down the convergence rate of the numerical scheme. It is natural to explore the possibility of regularizing ϕ itself, as it can be expressed in terms of its characteristic functions as

$$\phi = \sum_{i=1}^{n} i\psi_i(\phi). \tag{2.9}$$

The derivative of ϕ and $\psi_i(\phi)$ are respectively

$$\nabla\phi = \sum_{i=1}^{n} i\nabla\psi_i(\phi) \quad \text{and} \quad \nabla\psi_i(\phi) = \psi_i'(\phi)\nabla\phi, \tag{2.10}$$

and resultingly we have

$$|\nabla\phi| \le \sum_{i=1}^{n} |i||\nabla\psi_i(\phi)| \quad \text{and} \quad \sum_{i=1}^{n} |\nabla\psi_i(\phi)| \le \sum_{i=1}^{n} |\psi_i'(\phi)||\nabla\phi|. \tag{2.11}$$

Therefore, the following relation exists

$$\beta_1(n)|\nabla\phi| \le \sum_{i=1}^{n} |\nabla\psi_i(\phi)| \le \beta_2(n)|\nabla\phi|, \tag{2.12}$$

where $\beta_1(n)$ and $\beta_2(n)$ are positive functions depending on n, as previously mentioned in [8]. Thus the level set function ϕ can be regularized directly to obtain a faster solver, and the simplified energy functional then becomes

$$E(\mathbf{c}, \phi) = \int_\Omega (u^0 - \tilde{u}(\phi))^2\, d\bar{x} + \nu \int_\Omega |\nabla\phi|\, d\bar{x}. \tag{2.13}$$

We solve the energy functional by

$$\min_{\mathbf{c},\phi} E(\mathbf{c}, \phi), \quad \text{subject to } \phi \in \{1, \cdots, n\}, \tag{2.14}$$

to segment u^0 into piecewise constant regions.

3. PCLSM using graph cut

Solving the Euler-Lagrange equation derived for the PCLSM is a quite time consuming operation [2, 25, 26]. This is a major limitation of the method. In [2] a novel graph based approach for fast solving of $2D$ multiphase segmentation problems using the PCLSM was presented. Here we extend their approach for handling of 3D problems and test the efficiency of the approach for solving such problems. The method is tested on both synthetic and real data.

The idea of [2] is to discretize the PCLSM energy functional (2.13) directly to obtain a minimization problem that is graph representable. The discretization yields

$$E_d(\mathbf{c},\phi) = \sum_{p\in D}(u_p^0 - \tilde{u}_p)^2\Delta + \nu TV_d(\phi), \tag{3.1}$$

where $p=(i,j,k)$ corresponds to voxels in input image u^0 and the approximated image function is $\tilde{u}_p = \tilde{u}(\phi(p))$. D is the discrete domain with the same dimensionality as u^0. Thus, if the number of voxels in u^0 is $(N_x \times N_y \times N_z)$, then

$$D = \{(i,j,k),\, i=1,\cdots,N_x,\, j=1,\cdots,N_y,\, k=1,\cdots,N_z\}. \tag{3.2}$$

The grid spacing is given by Δ_i for direction i and $\Delta = \prod_{i=1}^{3}\Delta_i$. The constant ν controls the influence of the regularization term $TV_d(\phi)$, which is the discrete *total variation* of ϕ. For different neighborhood systems, the value of ν should vary to obtain the same amount of smoothing. Following the ideas of [5, 20] the authors chose to approximate TV_d using a modified L_1-norm for an 8-neighborhood system. For 3D problems we apply either a 6- or 26-neighborhood system about a voxel p, defined as

$$N_6 = \{(i\pm1,j,k),(i,j\pm1,k),(i,j,k\pm1)\}, \tag{3.3a}$$

$$N_{26} = \{(i\pm1,j,k),(i,j\pm1,k),\{(i\pm1,j\pm1,k),$$
$$\{(i\pm1,j,k\pm1),(i,j\pm1,k\pm1),(i\pm1,j\pm1,k\pm1),(i,j,k\pm1)\}. \tag{3.3b}$$

The regularization term can then be represented on discrete form as

$$TV_d(\phi) = \sum_{p\in D}\sum_{q\in N_\kappa(p)}\frac{1}{2}\omega_{pq}|\phi_p - \phi_q|, \qquad \text{for } \omega_{pq} = \frac{\Delta}{\|p-q\|_2}. \tag{3.4}$$

ω_{pq} is the edge weight between voxel p and q, divided by 2 since $\omega_{pq} = \omega_{qp}$. Interestingly, the derived weight ω_{pq} is similar to that of the discrete *Cauchy-Crofton* formula derived in [3], which does not depend on level set. For solving this problem it is necessary to create a special graph, outlined in Section 3.1. The graph based minimization problem is further described in Section 3.2 and in Section 3.3 the numerical details are given.

3.1. Graph representation

A graph $G=(V,E)$ consists of a set of vertices V and a set of directed edges E that connect the vertices. To represent the discrete PCLSM (3.1) on graph G, a special graph

structure need to be constructed. The idea of [2] is to generate a graph with an extra dimension when compared to the input image u^0. Such a graph structure enables us to represent multiphase segmentation problems, and allows us to differentiate between the different phases. In 3D, their approach corresponds to creating a *multi-volume* graph. Similar graph structures have also been considered in [10, 14].

The set of vertices V in the multi-volume graph G can be defined as

$$V = \{v_{p,l} \,|\, \forall p \in D,\, l \in \{1, \cdots, n-1\}\} \cup \{v_s\} \cup \{v_t\} \tag{3.5}$$

for a n-phase problem. Index l specifies which volume vertex $v_{p,l}$ belongs to, and each vertex $v_{p,l}$ corresponds to a voxel $p = (i, j, k)$. The dimensionality of each volume should correspond to the dimensionality D of the input image u^0, as is defined in (3.2). Notice that the set V also contains two special vertices, the *source* v_s and the *sink* v_t. They are referred to as terminal nodes and are used for partitioning V into two disjoint sets. For two-phase segmentation problems it is sufficient to construct a single volume graph.

The set of edges E in graph G can be divided into two groups. The *first* group is the between volume edge set E_D, connecting adjacent volumes as well as the first and last volume to the terminal nodes. The data term edges can then be defined as

$$E_D = \left\{ \bigcup_{p \in D} E_p \right\}. \tag{3.6}$$

For each vertex $p \in D$, for all volumes, the vertex edge set E_p can be defined as

$$E_p = \{(v_s, v_{p,1}) \cup_{l=1}^{n-2} (v_{p,l}, v_{p,l+1}) \cup (v_{p,n-1}, v_t)\}. \tag{3.7}$$

The between volume edges are denoted by $(v_{p,l}, v_{p,l+1})$ and the edges connected to terminal nodes are denoted by $(v_s, v_{p,1})$ and $(v_{p,n-1}, v_t)$. Note that not all vertices in V are connected by an edge e to the terminal nodes. The *second* group of edges are the within volume directional edges E_R. These are the regularization edges, defined by

$$E_R = \{(v_{p,l}, v_{q,l}) \,|\, \forall p \in D,\, q \in N_\kappa(p),\, l \in \{1, \cdots, n-1\}\}. \tag{3.8}$$

$N_\kappa(p)$ is the neighborhood system about vertex $v_{p,l}$, usually defined as in (3.3) for 3D systems. Thus there are directed edges from $v_{p,l}$ to $v_{q,l}$ and from $v_{q,l}$ to $v_{p,l}$. The total set of edges E in graph G is now given by

$$E = E_R \cup E_D. \tag{3.9}$$

For a more detailed description of the graph construction, we refer to [2].

Each edge e in edge set E (3.9) is assigned an edge weight ω_e derived from the discrete PCLSM energy function (3.1). From the data term the edge weights of E_D (3.6) are derived and given by

$$\begin{aligned}
&\omega_e(v_s, v_{p,1}) = (u^0(p) - c_1)^2 \Delta, && \forall p \in D, \\
&\omega_e(v_{p,l}, v_{p,l+1}) = (u^0(p) - c_{l+1})^2 \Delta, && \forall p \in D, \quad l = \{1, \cdots, n-2\}. \\
&\omega_e(v_{p,n-1}, v_t) = (u^0(p) - c_n)^2 \Delta, && \forall p \in D,
\end{aligned} \tag{3.10}$$

For two-phase problems the expression for the edge weighs is reduced to

$$\begin{aligned} \omega_e(v_s, v_p) &= (u^0(p) - c_1)^2 \Delta, \quad \forall p \in D, \\ \omega_e(v_p, v_t) &= (u^0(p) - c_2)^2 \Delta, \quad \forall p \in D. \end{aligned} \tag{3.11}$$

The edge weights of E_R in (3.8) are derived from the regularization term and given by

$$\omega_e(v_{p,l}, v_{q,l}) = \omega_{pq}, \quad \forall p \in D, \;\; q \in N_{\kappa=6,26}, \;\; l = \{1, \cdots, n-1\}, \tag{3.12}$$

where ω_{pq} is as defined in (3.4). The weights in E_R do not depend on the level in graph G.

A cut C on graph G is a set of edges $C \subset E$ which separates the vertices into two disjoint sets V_s and V_t, by separating the terminal nodes so that $v_s \in V_s$ and $v_t \in V_t$. The cost of this cut C is the sum over all edges in C, given by

$$|C| = \sum_{e \in C} \omega_e. \tag{3.13}$$

The minimum cut is the cut with the smallest total cost. It is the cut which also minimize the energy function [4]. Therefore, to minimize the discrete PCLSM (3.1) the minimum cut need to be computed. For a cut to be an admissible cut, it should sever at least one, and not more than one, edge in the data term edge set E_D for each vertex $p \in D$. That is, for each vertex edge set E_p the cut should sever exactly one edge. If for a given $p \in D$, the cut does not sever an edge in E_p, it is not a valid cut since it does not separate the terminal nodes v_s and v_t. If it severs more than one edge in E_p, then the cut cannot be a minimum cut since severing a single edge will always lead to a smaller total cost [2].

Fig. 1 illustrates how a graph G is constructed for $2D$ segmentation problems. In (a) the ordinary graph used for two-phase problems is displayed and in (b) the extended graph used in the case of three-phase problems. The cut C partitions the vertices into two sets so that $v_s \in V_s$ and $v_t \in V_t$. The planar edges corresponds to E_R and the vertical edges to E_D. For illustration purposes only edges in the cut C are shown for E_D. For the extended graph G in (b) the vertical edges between the planes are used to further differentiate between phases. In both (a) and (b) the corresponding segmentation of the cut C is displayed below the graph.

3.2. Graph based minimization

For an admissible cut C on a graph G, as defined in the previous section for multiphase problems, the PCLSF ϕ can be constructed by taking into account the edges in the cut, $E \subset C$, and their level in graph G, as is illustrated in Fig. 1(c). Thus, for n-phase problems ϕ can be defined as

$$\phi(p) = \begin{cases} 1, & \text{if } (v_s, v_{p,1}) \in C, \\ l+1, & \text{if } (v_{p,l}, v_{p,l+1}) \in C, \\ n, & \text{if } (v_{p,n-1}, v_t) \in C, \end{cases} \qquad \forall\, p \in D, \;\; l = \{1, \cdots, n-2\}. \tag{3.14}$$

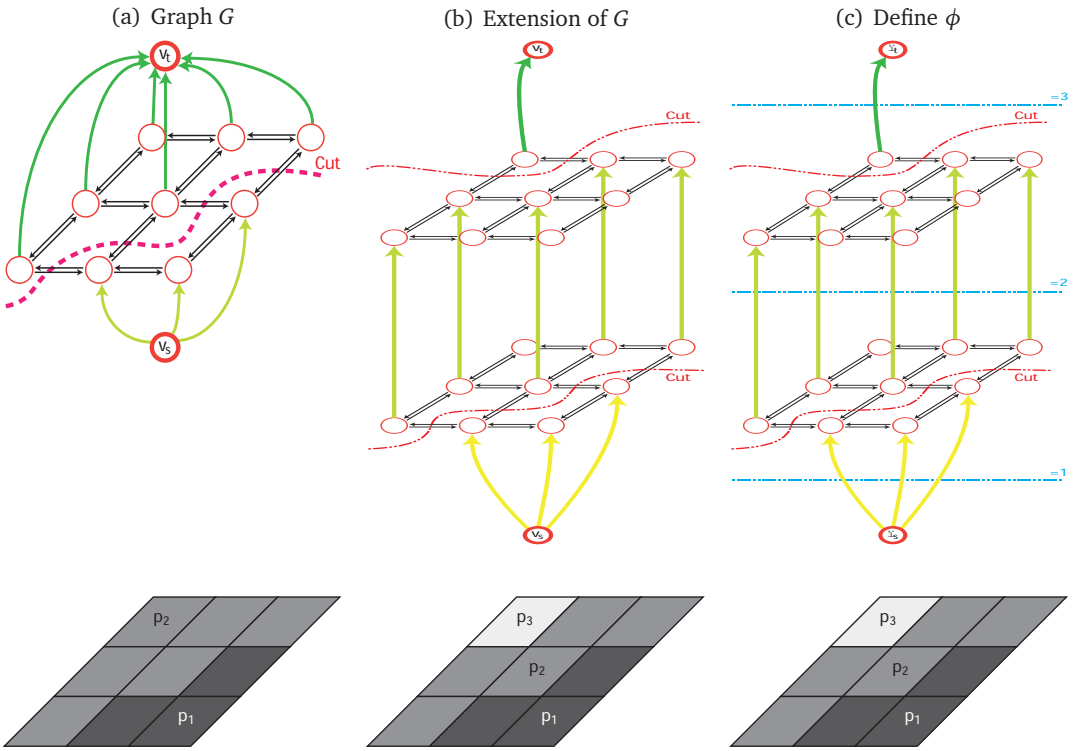


Figure 1: In (a) an illustration of an ordinary 2D graph G construction is shown, which can be used for representing two-phase problems. The cut C partitions V into two sets so that $v_s \in V_s$ and $v_t \in V_t$. The edge set E_R are the planar edges, and the vertical edges correspond to edge set E_D. For illustration purposes only edges in the cut C are shown for E_D, and the corresponding segmentation is displayed below. In (b) the graph has been extended to a multi-planar graph for handling three-phase problems. The between plane edges in set E_D are used to partition the graph further. The corresponding segmentation is also displayed. In (c) an illustration is given of how ϕ is defined by taking into account the edges in C and their level in G.

The discontinuities of ϕ corresponds to the current segmentation of the given problem. For two-phase segmentation problems where graph G only include a single volume, the definition of ϕ is reduced to

$$\phi(p) = \begin{cases} 1, & \text{if } (v_s, v_p) \in C, \\ 2, & \text{if } (v_p, v_t) \in C, \end{cases} \qquad \forall\, p \in D. \tag{3.15}$$

If the cut C is admissible, then ϕ is a single value function. For a given ϕ which fulfills (3.14), corresponding discrete characteristic functions ψ_i can be calculated. These are defined as

$$\psi_i(\phi) = \begin{cases} 1, & \text{for } \phi = i, \\ 0, & \text{otherwise}, \end{cases} \qquad \text{for } i = 1, \cdots, n, \tag{3.16}$$

which is a discrete version of (2.5). The approximate image function $\tilde{u}(\phi)$ can then, for a fixed ϕ, be specified as

$$\tilde{u}(\phi) = \sum_{i=1}^{n} c_i \psi_i(\phi), \qquad \text{for } c_i = \frac{\sum_{p\in D} u_p^0 \psi_i(\phi_p)}{\sum_{p\in D} \psi_i(\phi_p)^2}, \ \ \forall\, i = 1, \cdots, n, \tag{3.17}$$

which is the discrete version of (2.6) and (2.8).

To link the discrete PCLSM energy function $E_d(\mathbf{c}, \phi)$ in (3.1) to the graph G, edge weights are assigned to all edges E in G from $E_d(\mathbf{c}, \phi)$, as done for the data term edge weights $\omega_e(\cdot)$ in (3.10) and regularization term edge weights $\omega_e(v_{p,l}, v_{q,l})$ in (3.12). The cost of an arbitrary cut C on graph G is then equal to $|C| = E_d(\mathbf{c}, \phi)$. The energy function $E_d(\mathbf{c}, \phi)$ can therefore be minimized by solving the minimum cut problem,

$$\begin{aligned} \min_{C \,cut\, on\, G} |C| &= \min_{\phi} E_d(\mathbf{c}, \phi) \\ &= \min_{\phi} \sum_{p\in D} (u_p^0 - \tilde{u}_p)^2 \Delta + \nu \sum_{p\in D} \sum_{q\in N_\kappa(p)} \frac{1}{2} \omega_{pq} |\phi_p - \phi_q|. \end{aligned} \tag{3.18}$$

This is the cut which also minimize the continuous energy functional (2.14). For a more detailed exposition of the minimization approach in 2D, we refer to [2].

3.3. Numerical details

Numerically, the minimum cut problem in (3.18) is solved by iteratively minimizing

$$\min_{C(t)\, cut\, on\, G} |C(t)| = \min_{\phi^t} \sum_{p\in D} (u_p^0 - \tilde{u}_p^t)^2 \Delta + \nu \sum_{p\in D} \sum_{q\in N_\kappa(p)} \frac{1}{2} \omega_{pq} |\phi_p^t - \phi_q^t|. \tag{3.19}$$

For each time step t the minimum cut is found which depend on the time dependent value of $\tilde{u}^t$ in the data term, which again depend on the mean phase values $\mathbf{c}^t = (c_1^t, \cdots, c_n^t)$. After a single iteration, the minimum cut does not necessarily minimize functional (2.14). The optimal solution is first found when the mean phase values $\mathbf{c} = (c_1, \cdots, c_n)$ do not change significantly from one iteration to the next.

For each iteration, the edge weights $\omega_e(\cdot)$ of the data term are updated for the time dependent phase values $\mathbf{c}^t$,

$$\begin{aligned} &\omega_e^t(v_s, v_{p,1}) = (u^0(p) - c_1^t)^2 \Delta, && \forall p \in D, \\ &\omega_e^t(v_{p,l}, v_{p,l+1}) = (u^0(p) - c_{l+1}^t)^2 \Delta, && \forall p \in D, \quad l = \{1, \cdots, n-2\}. \\ &\omega_e^t(v_{p,n-1}, v_t) = (u^0(p) - c_n^t)^2 \Delta, && \forall p \in D, \end{aligned} \tag{3.20}$$

The index p corresponds to a voxel in u^0 and c_l^t is the mean value of *phase* l. The within volume edge weights $\omega_e(v_{p,l}, v_{q,l})$ (3.12) do not change. The only user interaction required for solving the problem is an initial guess of the mean phase vector $\mathbf{c}^0$, since ϕ need

not be initialized. After each iteration, the PCLSF ϕ^t can be defined as in (3.14) and the characteristic functions ψ_l^t can be defined as in (3.16). Furthermore, once ϕ^t and ψ_l^t has been specified, the approximate image function $\tilde{u}^t$ and the mean phase values $\mathbf{c}^t = (c_1^t, \cdots, c_n^t)$ can be updated by (3.17). An outline of the numerical approach is given in Algorithm 3.1.

Algorithm 3.1. Construct a graph $G = (V, E)$ and assign edge weights to all edges using $\omega_e(v_{p,l}, v_{q,l})$ (3.12) and $\omega_e(\cdot)$ (3.20), given an initial phase value guess $\mathbf{c}^0 = (c_1^0, \cdots, c_n^0)$. Then (2.14) can be minimized iteratively by solving

1) Compute minimum cut of G (3.19) for $\mathbf{c}^0$.
2) Update $\mathbf{c}^t$, $t = 1$.
3) *while*($\|\mathbf{c}^t - \mathbf{c}^{t-1}\| \geq \epsilon$), ϵ small
   i) Update weights $\omega_e(\cdot)$ as in (3.20).
   ii) Compute minimum cut of G (3.19) for $\mathbf{c}^t$.
   iii) Update $\mathbf{c}^t$, t++.

4. Results

For relatively large data-sets, it is nearly impossible to use the original PCLSM due to the expense of computation. Minimizing the PCLSM energy functional by graph cut, a very fast solver for relatively large data-set is obtained. Steady-state solution is now found in a matter of minutes using very few iterations, as shown in Table 1 for volumes of different sizes. This is in particular true for N_6 neighborhood systems, for N_{26} system the method is still a bit time consuming especially for larger data-set. Also for N_6 systems the computation time increase rapidly the larger the data-set is, but on an entirely different time scale. Thus, future effort is required to developing an even faster solver for the graph based minimization problem for very large data-sets. In [2] the graph cut approach for solving the PCLSM in 2D was compared to a traditional *Euler-Lagrange* approach. They showed that the graph cut approach solves the problems much faster than the traditional approach, obtaining very similar segmentation results. For volume segmentation we have previously experienced that it takes many hours to solve similar problems using an Euler-Lagrange approach.

A second factor that influence the computation time is the number of phases included in the segmentation problem. This is illustrated in Table 2 for three different volumes of varying sizes for 2-, 4-, 10- and 15-phase problems. Observe that the computation time increase significantly with both the size of the volume and for increasing number of phases. Although solving segmentation problems with a large number of phases is not very common, this illustrates the importance of developing an even faster solver. Due to memory problems, the computations has not been performed for 10- and 15-phase problems on the largest data-set.

Table 1: Compares computation time and number of iterations for minimizing (3.19) for varying volume sizes and neighborhood systems, N_6 and N_{26}. Observe that when N_6 systems are applied the solution is found very fast for all test cases. For N_{26} systems the computation time increase, in particular for the two largest volumes. For both N_6 and N_{26} systems, the solution is found using few iterations. The problem is solved by *maximum flow*, implemented in $C++$, based on the work of [4].

	N_6		N_{26}	
Volume Size	No. of It.	Time (min.)	No. of It.	Time (min.)
$(45 \times 45 \times 25)$	12	0.2000	8	0.3667
$(90 \times 90 \times 50)$	8	0.7167	17	10.5667
$(180 \times 180 \times 100)$	9	8.5833	17	142.2000

Table 2: The table compares the computation time for segmenting image volumes of different sizes for problems of varying number of phases. Observe that the computation time increase significantly both with the number of phases and with the size of the image volume. Due to memory problems, the computations could not be performed on the largest data-set for 10- and 15-phase problems. All computations are performed using a N_6 neighborhood system.

	Time (min.)			
Volume Size	2-Phase	4-Phase	10-Phase	15-Phase
$(45 \times 45 \times 25)$	0	0.0833	0.3333	0.5000
$(90 \times 90 \times 50)$	0.0167	0.6167	2.1167	3.7500
$(180 \times 180 \times 100)$	0.1500	5.1333	–	–

4.1. Real CT data

The method has been tested by segmenting a real 3D CT data-set of an unknown object. The object has a complicated topology with sharp edges, narrow ridges and a partly spiraling behavior with sharp twists and turns. Using a *two-phase* approach we are able to segment the object, as shown for different intermediate slices in Fig. 2. In column (a) the original data is displayed, in (b) the resulting segmentation and in (c) the segmentation boundary overlaid the original data. We see that the method is capable of capturing complicated structures within a volume, which has varying properties. By visual evaluation of these slices it is confirmed that our segmentation approach works satisfactory. In fact the segmentation perfectly match the conceptual boundary.

Scrolling through the volume slice-by-slice it is very difficult, if even possible, to imagine the structure of the object. It is first after segmenting the object and reconstructing it, that it is possible to comprehend its true structure, which is displayed in Fig. 3. In (a) the full reconstructed object is displayed, and in (b) and (c) different angles of a sub-volume is displayed. Observe that we are able to reconstruct the spiral interior shape of the object from the segmentation. In the presented example we have used a N_{26} neighborhood system in graph G. It is, however, interesting to observe that almost the same output is obtained using a N_6 system. Visually no difference is observed between the two approaches. This suggest that it is sufficient to use a N_6 neighborhood system, which is more computational efficient than a N_{26} system, to obtain a good segmentation of the data-set.

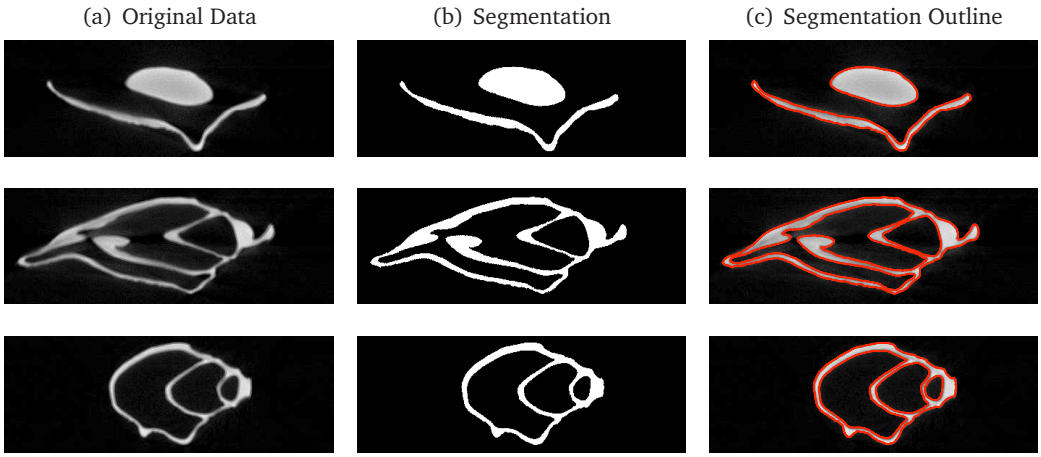
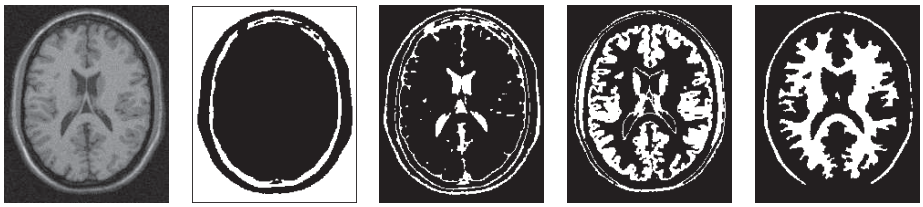


Figure 2: In column (a) three different slices of the original data are displayed. Column (b) displays the segmented data of the same slices and in (c) the contour of the segmented data overlaid the original data. The segmented data correspond to an output characteristic function ψ as defined in (3.16). Observe in (c) that good segmentation results are obtained, where the segmentation perfectly match the conceptual boundary. The segmentation is obtained using a N_{26} neighborhood system.

Figure 3: Displays different views of the reconstructed object based on the segmentation. In (a) the full object is displayed, and in (b) and (c) two different sub volume views. Observe that we are able to reconstruct the spiraling shape to the interior of the object.

4.2. Synthetic MR data

The method has also been tested on synthetic MR data (T1) of a normal human brain, obtained from [1]. It has 7% noise, 20% intensity non-uniformity and the voxel spacing is $1mm$ in all direction. In this paper, we work with a sub-volume u^0 of this data-set with $(217 \times 181 \times 106)$ voxels, which do not include e.g. empty slices or eye features. When segmenting u^0 using a *four-phase* PCLSM approach, the resulting characteristic functions ψ_i are prone to have *skull features* as shown in Fig. 4 for a given slice. The original data is displayed in (a) and in (b)-(e) the characteristic functions are displayed, ψ_1 through ψ_4.



(a) u^0 (b) ψ_1 (c) ψ_2 (d) ψ_3 (e) ψ_4

Figure 4: In (a) synthetic MR data u^0 obtained from [1] is displayed for a given slice. The resulting characteristic functions $\{\psi_1, \psi_2, \psi_3, \psi_4\}$ obtained when segmenting u^0 using a *four-phase* PCLSM approach are displayed in (b)-(e) for the same slice. Observe that *skull features* are present in all ψ's. To extract the CSF in (c), the gray matter in (d) and the white matter in (e) the removal of skull features is required. The segmentation is obtained using a N_6 neighborhood system.

The white regions of $\{\psi_1, \psi_2, \psi_3, \psi_4\}$ correspond to the segmented data for a given phase, while the black regions do not belong to the segmentation for the same phase. To extract the cerebrospinal fluid (CSF), gray and white matter given respectively in ψ_2, ψ_3 and ψ_4 the skull feature should be removed.

To remove skull features we have developed a simple skull stripping approach. It is based on the observation that the characteristic function ψ_1 from Fig. 4(b), which mostly segments out the background, resembles a *mask*. Therefore we derive a mask M from ψ_1 for the removal of skull features. It is created by employing the morphological operation of *closing* to create a single, closed, object where the gap which is observed in ψ_1 has been removed. Closing is defined as

$$\psi_1^C = \psi_1 \bullet S = (\psi_1 \oplus S) \ominus S, \tag{4.1}$$

where ψ_1^C is the closing of ψ_1 using a flat circular structuring element S. The computation of ψ_1^C is therefore done in a slice-by-slice manner. The closing operation performs first a *dilation* operation $\oplus$, then *erosion* $\ominus$. Dilation can be viewed as a region growing operation in which pixels are added to the object boundary. Erosion can on the contrary be viewed as a region shrinking operation in which pixels are removed from the boundary. For more information about morphological operations consider e.g. [22]. Once ψ_1^C is found, the mask M can be defined as

$$M = \psi_1^C < 1. \tag{4.2}$$

Skull features are then removed from $\{\psi_2, \psi_3, \psi_4\}$ by multiplying them with M. After removing the skull features, the largest group of connected components are extracted from the volume using labeling to eliminate random groups of voxels. A danger using this approach is that there is no way to guarantee that the mask M do not interfere with CSF, gray and white matter which lies within. By inspection, however, we have not found this to be a problem as long as an appropriate structuring element S is chosen. For real MR data, this skull stripping approach should be further developed.

Fig. 5 shows the original MR data in column (a) for selected *sagittal*, *coronal* and *axial* planes. In column (b)-(d) the *anatomical models* from [1] (underlay) are compared

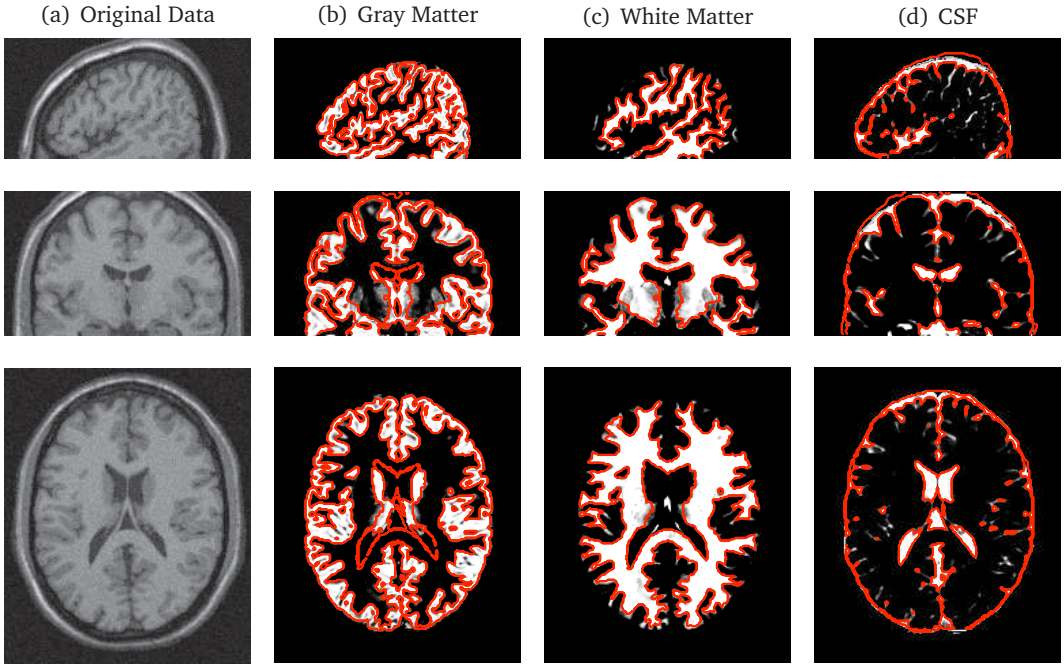
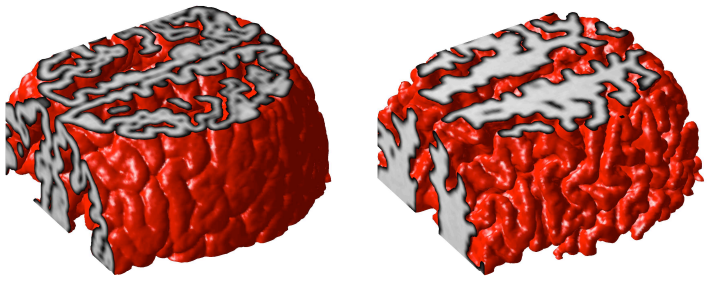


Figure 5: In *column* (a) the original data is displayed for a given *sagittal*, *coronal* and *axial* planes. In (b)-(d) the corresponding planes from the segmentation is compared with the *anatomical models* (underlay) from [1] for each tissue class. The red contour outlines the segmented gray matter in (b), white matter in (c) and CSF in (d). We observe that the segmentations detects the major trends in the anatomical model. The small deviations might come from e.g. the amount of smoothing in the regularization term. Another reason could be that more than these three anatomical models are used when creating the synthetic data volume. Our segmentation in (b) outline the internal ventricles filled with CSF, which is not due to gray mater but to a narrow ridge of voxels between white matter and CSF.



(a) Gray Matter (b) White Matter

Figure 6: Volume display of the reconstructed gray and white matter.

with the extracted gray matter, white matter and CSF using our approach for the same planes. Visual verification confirms that the segmented data corresponds well with the major trends in the original data, and it is verified that it detects the boundaries of the anatomical models. Small deviations comes from e.g. combining more than these three

anatomical models to create the full brain, the smoothing effect of the regularization term, and also from displaying only connected components. For gray matter we observe a deviation in the coronal plane. In the apex there is a small artifact that has not been removed. Also the internal ventricles which are filled by CSF are outlined in the gray matter segmentation. This is not due to actual gray matter, but to a narrow ridge of voxels in the input data which can be observed between CSF and white matter. For white matter we observe that the detected data corresponds well with the anatomical model. For CSF, in the apex of the sagittal and coronal plane we observe that the model detects slightly more data than the anatomical model indicates. By comparing the segmentation with the original data we do, however, observe that the voxel intensity in this region is very close to that of CSF. The segmentation model is therefore found to perform satisfactory for this data-set. In Fig. 6 the reconstructed subvolumes from the segmented data of the gray and white matter are displayed. These volume views confirms that the detected data has the expected shape.

Due to computation time requirement for larger data-sets, using a N_{26} neighborhood system to segment u^0 is too time consuming, as was shown in Table 1. Full volume segmentation has therefore only been performed using a N_6 system. For sub-volumes of u^0 similar results are, however, obtained using N_{26} systems. It requires less weight on constant value ν to obtain the same amount of smoothing.

5. Conclusion

We have demonstrated that the PCLSM can be minimized by graph cut for volume image segmentation on both synthetic and real data with complicated structures. This yields accurate segmentation for both N_6 and N_{26} neighborhood systems which converges swiftly to the optimal solution for relatively large data-sets. The computation time increase more rapidly for N_{26} systems with the size of the data-set. It is also influenced by the number of phases in the segmentation problem. For real time applications future work is required to derive an even faster graph cut solver for very large data-sets. The convergence rate of the presented solver is much faster than ordinary Euler-Lagrange approaches.

Acknowledgments Our courtesy goes to Capital Normal University, Department of Mathematics, Beijing, China, for providing the CT data. Financially, we would like to thank the support from the Centre for Integrated Petroleum Research (CIPR), University of Bergen, Norway, and Singapore MOE Grant T207B2202 and NRF2007IDMIDM002-010.

References

[1] BrainWeb: Simulated Brain Database. `http://www.bic.mni.mcgill.ca/brainweb/`, August 2008.

[2] E. Bae and X.-C. Tai. Graph cuts for the multiphase Mumford-Shah model using piecewise constant level set methods. *UCLA Computational and Applied Mathematics Reports* (08-36), 2008.

[3] Y. Boykov and V. Kolmogorov. Computing geodesic and minimal surfaces via graph cuts. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 1:22–33, 2003.

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

[5] A. Chambolle. Total variation minimization and a class of binary MRF models. In A. Rangarajan, B. C. Vemuri, and A. L. Yuille, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 3757 of *Lecture Notes in Computer Science*, pages 136–152. Springer Berlin/Heidelberg, 2005.

[6] T. F. Chan and J. Shen. *Image Processing And Analysis: Variational, Pde, Wavelet, And Stochastic Methods*. Society for Industrial and Applied Mathematics, 2005.

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

[8] O. Christiansen and X.-C. Tai. Fast implementation of piecewise constant level set methods. In X.-C. Tai, K.-A. Lie, T. F. Chan, and S. J. Osher, editors, *Image Processing Based on Partial Differential Equations*, Mathematics and Visualization, pages 289–308. Springer Berlin Heidelberg, 2007.

[9] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part I: Fast and exact optimization. *Journal of Mathematical Imaging and Vision*, 26(3):261–276, 2006.

[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):277Ð291, 2006.

[11] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[12] D. Goldfarb and W. Yin. Parametric maximum flow algorithms for fast total variation minimization. *RICE CAAM TR07-09*, 2007.

[13] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, 51(2):271–279, 1989.

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

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

[16] 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.

[17] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[18] S. J. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[19] S. J. Osher and R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag New Yourk, Inc., 2003.

[20] F. Ranchin, A. Chambolle, and F. Dibos. Total variation minimization and graph cuts for moving objects segmentation. In F. Sgallari, A. Murli, and N. Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 743–753. Springer Berlin/Heidelberg, 2007.

[21] A. Losnegård, O. Christiansen, and X.-C. Tai. Piecewise constant level set method for 3d image segmentation. In F. Sgallari, A. Murli, and N. Paragios, editors, *Scale space and variational methods in Computer vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 687–

696. Springer Berlin/Heidelberg, 2007.
[22] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag New York, Inc., 2 edition, 2003.
[23] 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.
[24] X.-C. Tai, K.-A. Lie, T. F. Chan, and S. J. Osher, editors. *Image Processing Based on Partial Differential Equations*. Mathematics and Visualization. Springer-Verlag New York, Inc., 2007.
[25] X.-C. Tai and C.-H. Yao. Image segmentation by piecewise constant Mumford-Shah model without estimating the constants. *Journal of Computational Mathematics*, 24(3):435–443, 2006.
[26] X.-C. Tai and C.-H. Yao. Fast PCLSM with Newton updating algorithm. In X.-C. Tai, K.-A. Lie, T. F. Chan, and S. J. Osher, editors, *Image Processing Based on Partial Differential Equations*, Mathematics and Visualization, pages 249–262. Springer Berlin Heidelberg, 2007.
[27] L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah mode. *International Journal of Computer Vision*, 50(3):271–293, 2002.