Image and Vision Computing 29 (2011) 365–381

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

# Combinatorial Optimization of the piecewise constant Mumford-Shah functional with application to scalar/vector valued and volumetric image segmentation[☆]

Noha El-Zehiry [a,*,1], Prasanna Sahoo [b], Adel Elmaghraby [c]

[a] Siemens Corporate Research, Department of Imaging and Visualization Princeton, NJ 08540, United States
[b] Department of Mathematics, University of Louisville, Louisville, KY 40292, United States
[c] Computer Science Department, University of Louisville, Louisville, KY 40292, United States

ARTICLE INFO

*Article history:*
Received 21 January 2010
Received in revised form 22 July 2010
Accepted 16 September 2010

*Keywords:*
Active contours
Graph cuts
Image segmentation

ABSTRACT

Front propagation models represent an important category of image segmentation techniques in the current literature. These models are normally formulated in a continuous level sets framework and optimized using gradient descent methods. Such formulations result in very slow algorithms that get easily stuck in local solutions and are highly sensitive to initialization.

In this paper, we reformulate one of the most influential front propagation models, the Chan–Vese model, in the discrete domain. The graph representability and submodularity of the discrete energy function is established and then max-flow/min-cut approach is applied to perform the optimization of the discrete energy function. Our results show that this formulation is much more robust than the level sets formulation. Our approach is not sensitive to initialization and provides much faster solutions than level sets. The results also depict that our segmentation approach is robust to topology changes, noise and ill-defined edges, i.e., it preserves all the advantages associated with level sets methods.

## 1. Introduction

Image segmentation is one of the most important and critical tasks in the field of computer vision. Snakes and deformable models have been introduced by Kass, Witkin and Terzouplos [1]. Following the seminal work of Kass, Witkin and Terzouplos [1] several edge based models [2–6] and region based models [7,8] have been introduced to solve the image segmentation problem. All the aforementioned models aim at capturing or extracting the contours of the different distinct objects in the image, and hence provide a pixel labeling that assigns a specific class to each pixel.

These techniques made a breakthrough and tremendously contributed to the problem of pixel labeling. Nevertheless, they suffer from three major disadvantages. First, they find the nearest local minimum to the initialized contour. In other words, they are very sensitive to initialization. Secondly, the mathematical formulation of most of them depends, in one way or another, on the gradient which highly affects the robustness to noise. Finally, these techniques cannot handle topology changes.

The image segmentation problem has been formulated (independent of the edge) by Mumford and Shah [9]. The problem can be described as follows:

- Given an image of interest $u_o : \Omega \rightarrow \mathbb{R}$, where $\Omega$ is an open bounded subset in $\mathbb{R}^2$ that consists of several connected components $\Omega_i$ and bounded by a closed boundary $C = \partial\Omega$, find an approximation $u$ of $u_o$ such that $u$ is smooth within the components $\Omega_i$ and sharp in the transition across the boundary $C$.

Mumford and Shah [9] proposed to solve this problem via minimizing the energy function

$$E^{MS}(u,c) = \int_{\Omega}(u-u_o)^2 dxdy + \mu\int_{\Omega\backslash C}|\nabla u|^2 dxdy + \nu|C| \tag{1}$$

where $\mu$ and $\nu$ are positive weights to control the effect of each term in 1). Theoretically, the existence of a minimizer that minimizes the energy function in (1) has been proved [9]. However, a reduced case that can be practically implemented is obtained by assuming a piecewise constant model, in other words, $u = c_i$ (constant) inside each connected component $\Omega_i$. Then the energy function is simplified to

$$E^{MS}(u,C) = \sum_i \int_{\Omega_i}(u_o-c_i)^2 dxdy + \nu|C|. \tag{2}$$

From the theoretical point of view, the Mumford–Shah model was one of the most important formulations of the image segmentation problem

[☆] This paper has been recommended for acceptance by Sinisa Todorovic.

[*] Corresponding author.

*E-mail addresses:* noha.el-zehiry@siemens.com (N. El-Zehiry), sahoo@louisville.edu (P. Sahoo), adel@louisville.edu (A. Elmaghraby).

[1] This work has been completely developed at the University of Louisville.

0262-8856/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2010.09.002

in the field of computer vision. However, this model was not very useful until numerical approximations and implementations were introduced.

Several research groups introduced numerical implementations to the Mumford–Shah model (see [10] for a review). On the top of these contributions comes the "Active Contour without Edges" [11] a.k.a Chan–Vese model. Chan and Vese introduced a level set formulation for the Mumford–Shah functional that converted the problem into a mean curvature flow problem just like the active contours but the results outperformed the classical active contours because the stopping term did not depend on the gradient of the image which reduces the dependence on clear edges. Segmentation results in [11] illustrate the robustness to noise and topology changes due to the implicit representation of the active contour in a level set framework.

The model introduced in [11] has been extended by the same authors in [10] to handle multi-phase evolution and segment complex topological structures. The papers [10,11] served as a solid infrastructure for several segmentation techniques that have been introduced later. The presentation of the level set formulation of the Mumford–Shah functional (see [11]) provided a major breakthrough in image segmentation and triggered the development of many subsequent techniques. These techniques used level set formulation of the Mumford–Shah functional as the main building block for novel segmentation algorithms and then handled other aspects of the problem, such as incorporating shape priors [12,13,15–17]. Interested reader should refer to [12,13] and [14] for an account on these new techniques.

However, the Chan–Vese model and level sets formulation in general suffer from the following drawbacks: 1) Level set formulation is very slow as it requires the evolution of the contour with infinitesimal steps around the initialized contour. 2) The optimization is generally performed using gradient descent which is prone to getting stuck in local solutions which makes the level set formulation very sensitive to initialization. 3) Level set methods require the tuning of many parameters such as the step size, for example.

Motivated by these drawbacks, we explore the discrete formulation of the piecewise constant Mumford–Shah model and the combinatorial optimization of the discrete energy function. In this paper, we present a novel attempt to combine the advantages of two standard frameworks of image segmentation: graph cuts and level sets. Due to the major importance and impact of the Chan–Vese model "Active Contour without Edges" on the image segmentation problem, we chose it to be the underlying level set model of our formulation. A discrete representation of the Chan–Vese model is presented. The graph representability of the new discrete model is proved and the optimization is performed using graph cuts. The dynamic labeling associated with the graph cut minimization will improve the speed of the implementation, and the fact that graph cuts solve for global minimum rather than a local one will improve the accuracy of the algorithm and make it much less sensitive to initialization.

It is worth mentioning that our model is not a new implementation of the Chan–Vese model but a different numerical technique to perform front propagation. Before introducing the details of our formulation, we review in the next section the most related previous work to our contribution.

### 1.1. Previous work

Xu et al. [18] presented a heuristic approach to combine the benefits of graph cuts and active contours. The authors assumed that that "the desired segmentation contour is a global within its a priori known size (width) contour neighborhood (CN, which is defined as a belt-shaped neighborhood region around the contour)". The initial contour as well as the contour neighborhood (CN), for a certain segment or image, will be defined by the user. The objective of the algorithm was to find the closest contour that is a global minimum within the initial contour neighborhood. Due to the assumption that the authors made, their segmentation algorithm requires a well trained expert that can initialize the contour properly, otherwise the segmentation would fail. On the contrary to their approach, the global minimum is obtained, in our formulation, regardless of the location of the initial contour. Moreover, in [18], the authors outlined that their algorithm cannot segment multiple objects or handle topology changes.

We also would like to point out the noticeable contributions of Darbon [19] and Grady and Alvino [20] in this problem. The former has investigated the minimization of the binary Mumford–Shah model in a different perspective than ours in which the minimization problem is looked at as a contrast invariant filter. Darbon's formulation is limited to $L-1$ fidelity terms. The work of Grady and Aalvino [20] is the most related to our work in this paper. They have presented combinatorial reformulation of differential operators in order to present solutions for PDE models on discrete lattices. Neither of the previous studies nor ours in [21] introduced the formulation for higher dimensions or vector valued images. But, in this paper, we will present extensions of the formulation to the surface evolution in 3D and to the segmentation of vector valued images.

The paper will be organized as follows: Section 2 will introduce the most related work that serve as preliminary material to our approach and will be used later in our formulation. Section 3 will present the proposed formulation including the discrete formulation of the Chan–Vese model and the optimization using graph cuts. Section 4 introduces a natural extension of the proposed method for the vector valued images and Section 5 provides the formulation for the 3D segmentation. Section 6 will explain the experimental results including quantitative and qualitative comparisons with the active contour without edges model and finally, Section 7 will discuss the conclusion and the future work.

## 2. Related work and background

### 2.1. Active contour without edges

Active contour without edges [11], introduced by Chan and Vese, is one of the leading papers that introduced numerical implementation for the Mumford–Shah functional through a level set framework. This section will review the Chan–Vese formulation as a preliminary step to introduce the discrete formulation of it and the graph cut minimization of the corresponding discrete energy function.

Recall that the problem of interest is to evolve a curve $C$ in $\Omega$ until it captures the boundaries of the region of interest $\omega$ in $\Omega$. In other words, a contour $C$ is initialized in $\Omega$, and the objective is to obtain a final contour $C_o = \partial\omega$.

Chan and Vese introduced the "fitting energy term"

$$F_1(C) + F_2(C) = \int_{inside(C)} |u(x,y) - c_1|^2 dxdy + \int_{outside(C)} |u(x,y) - c_2|^2 dxdy \tag{3}$$

where $u$ is the intensity level of the pixel $(x,y)$, $c_1$ and $c_2$ represent the mean intensity values inside and outside $C$, respectively. This fitting energy is minimum if and only if $C = C_o$ that is

$$\inf_{C} \{F_1(C) + F_2(C)\} \approx 0 \approx F_1(C_o) + F_2(C_o).$$

Furthermore, to maintain the smoothness of the contour, the length of the contour has been added as a regularization term and the final energy function has been introduced as

$$\begin{aligned} F(c_1, c_2, C) = {} & \mu \; Length(C) + \nu \; Area(inside(C)) \\ & + \int_{inside(C)} |u(x,y) - c_1|^2 dxdy \\ & + \int_{outside(C)} |u(x,y) - c_2|^2 dxdy. \end{aligned} \tag{4}$$

Using the level set calculus [22,23], Chan and Vese reformulated in a level set framework as

$$\begin{aligned} F(c_1,c_2,\phi) = {} & \mu\int_\Omega \delta(\phi(x,y))|\nabla\phi(x,y)|dxdy + \nu\int_\Omega H(\phi(x,y))dxdy \\ & + \lambda_1\int_\Omega |u(x,y)-c_1|^2 H(\phi(x,y))dxdy \\ & + \lambda_2\int_\Omega |u(x,y)-c_2|^2(1-H(\phi(x,y)))dxdy, \end{aligned} \tag{5}$$

where $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ is the level set function satisfying

$$\phi(x,t)\begin{cases} >0 & \text{for } x\in\omega\ (\textit{inside}) \\ =0 & \text{for } x\in\partial\omega = C(t)\ (\textit{boundary}) \\ <0 & \text{for } x\in\Omega-\omega\ (\textit{outside}). \end{cases} \tag{6}$$

$H(\phi)$ is the Heaviside step function defined as

$$H(\phi) = \begin{cases} 1 & \text{if } \phi\geq 0 \\ 0 & \text{if } \phi<0, \end{cases} \tag{7}$$

$\delta(\phi) = H'(\phi)$ is the Dirac delta function, and $\lambda_1$, $\lambda_2$, $\mu$ and $\nu$ are fixed parameters such that $\lambda_1$, $\lambda_2>0$ and $\mu,\nu\geq 0$.

The minimization of $F$ in Eq. (5) is achieved by taking the Euler–Lagrange equations, and hence the problem is reduced to iterative solution of

$$\frac{\partial\phi}{\partial t} = \delta_\epsilon\left[\mu\, div\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \nu - \lambda_1(u_o-c_1)^2 + \lambda_2(u_o-c_2)^2\right] = 0 \tag{8}$$

with initial condition $\phi(0,x,y) = \phi_o(x,y)$ in $\Omega$. Eq. (8) has been solved in an iterative manner using the gradient descent numerical implementation. The constants $c_1$ and $c_2$ can be reformulated using the level set function $\phi$ as

$$c_1(\phi) = \frac{\int_\Omega u(x,y)H(\phi(x,y))dxdy}{\int_\Omega H(\phi(x,y))dxdy} \tag{9}$$

and

$$c_2(\phi) = \frac{\int_\Omega u(x,y)(1-H(\phi(x,y)))dxdy}{\int_\Omega(1-H(\phi(x,y)))dxdy}. \tag{10}$$

It results in the piecewise smooth approximation for the image $u(x,y)$ as

$$u(x,y) = c_1H(\phi(x,y)) + c_2(1-H(\phi(x,y))).$$

The model suffers from two drawbacks (which we consider common drawbacks in the algorithms that depend mainly on the level set method). First, the use of the level set method is computationally very expensive which makes the model slow and hence the speed is an issue that should be looked at. Second, the minimization of the Chan–Vese energy function depends mainly on the gradient descent method which can minimize the energy function perfectly in a local sense. In other words, the gradient descent method returns the closest local minimum to the initialization. Hence, the model is very sensitive to initialization and it would be better if a global minimum can be obtained regardless of the initialization.

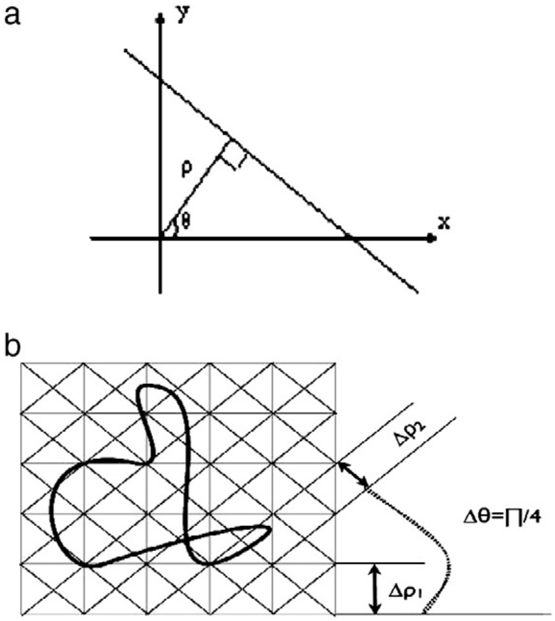

**Fig. 1.** (a) Polar representation of the straight line in two dimensional space. (b) Four families of parallel lines at angles 0, 45, 90 and 135. In our numerical implementation we set $\Delta\rho$ to one in all directions.

These two factors motivated us to work on introducing an integrated graph cut framework for the curve evolution problem to resolve the previous issues. The dynamic labeling associated with the graph cut minimization will improve the speed of the implementation. The fact that graph cuts solve for global minimum rather than a local one will also improve the accuracy of the algorithm and will make it much less sensitive to initialization.

### 2.2. Cut metrics

Approximating the length of the contour using graph representation was an essential step in our proposed framework. This problem has been addressed before by Kolmogorov and Boykov [24,25]. The authors construct a grid graph and assign the weights to the edges of the graph such that the cost of the cut approximates the length of the contour. It has been done using Cauchy–Crofton formula. Cauchy–Crofton formula implies that drawing sufficiently large number of straight lines in all directions from 0 to $2\pi$ and counting the number of points of intersections of the lines and the contour of interest approximates the length of the contour.

Mathematically, a line $L$ can be uniquely represented (See Fig. 1(a)) by an ordered pair $(\theta,\rho)$. Hence the lines in $R^2$ can be described as

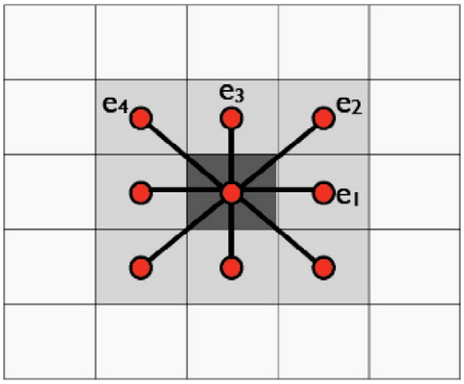

**Fig. 2.** Neighborhood system of size 8. $\Delta\theta = \frac{\pi}{4}$, $|e_1|=|e_3|=1$ and $|e_2| = |e_4| = \sqrt{2}$.

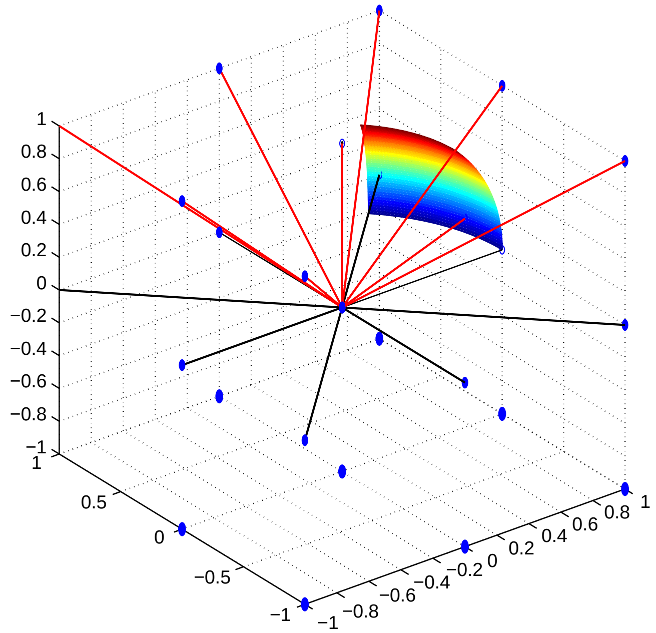

**Fig. 3.** Neighborhood system used to approximate the surface area of the surface $S$.

$\{(\theta,\rho)|0\leq\theta\leq 2\pi,\rho\geq 0\}$. The mathematical formulation of the Euclidean contour length using the Cauchy–Crofton formula is given by

$$\|C\|_E = \frac{1}{2}\int_0^{\pi}\int_{-\infty}^{\infty} n(\theta,\rho)d\rho d\theta \qquad (11)$$

where $\|C\|_E$ is the Euclidean length and $n(\theta,\rho)$ is the number of times that the line $L$ represented by $(\theta,\rho)$ intersects the contour C. By reasonably partitioning the set $[0,\pi] \times \mathbb{R}$, Boykov and Kolmogorov derived the following discrete formula to approximate the length of the contour

$$\|C\|_E = \frac{1}{2}\sum_k n_k \frac{\delta^2\Delta\theta_k}{|e_k|}$$

$n_k$ is the number of points in the neighborhood system, and $|e_k|$ is the length of the edge. Letting $w_k = \frac{\delta^2\Delta\theta_k}{2|e_k|}$, the last equation reduces to

$$\|C\|_E = \sum_k n_k w_k. \qquad (12)$$

Fig. 1(b) shows an example for four families (horizontal, vertical, principal and non principal diagonal ) of parallel lines.

## 3. The proposed method

This section will present the details of the discrete formulation of the Chan–Vese energy functional. But, first, we would like to introduce the graph notation that will be used throughout the paper. A graph $\mathcal{G}=\{\mathcal{V},\varepsilon\}$ consists of a vertex set $v\in\mathcal{V}$ and an edge set $e\in\varepsilon \quad \mathcal{V}\times\mathcal{V}$. Every pixel $p=(x,y)$ will have a corresponding vertex $v_p\in\mathcal{V}$. An edge that is incident to $v_p$ and $v_q$ is denoted $e_{pq}$. In a weighted graph, each edge $e_{pq}$ is assigned a weight $w_{pq}$. A cut $\mathcal{C}$ is a set of edges whose removal partitions the graph into two disjoint sets $S$ and $\overline{S}$. In max-flow applications, we distinguish two vertices; the source of the flow $\mathcal{S}$ and the target of the flow $\mathcal{T}$.

### 3.1. Discrete formulation

Define a binary variable $x_p$ for each pixel $p=(x,y)\in\Omega$ such that

$$x_p = \begin{cases} 1 & \text{if } \phi(p) > 0; \\ 0 & \text{otherwise.} \end{cases} \qquad (13)$$

Observing the analogy between the proposed binary variable $x_p$ in (13) and the Heaviside step function in Eq. (7), the last three terms in Eq. (5) can be easily discretized as follows:

$$\begin{aligned} F(c_1,c_2,\phi,x_1,\ \ldots,x_n) = {} & \mu\int_\Omega \delta(\phi(x,y))|\nabla\phi(x,y)|dxdy \\ & + \nu\textstyle\sum_p x_p + \lambda_1\sum_p |u(x,y)-c_1|^2 x_p \\ & + \lambda_2\textstyle\sum_p |u(x,y)-c_2|^2\left(1-x_p\right). \end{aligned} \qquad (14)$$

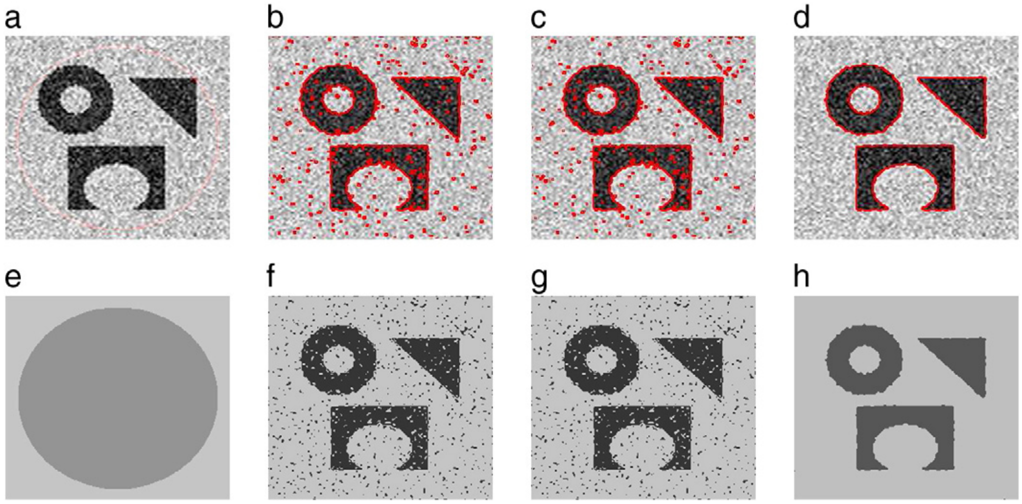

**Fig. 4.** Noise and topology — illustration of robustness to topology changes and noise. (a) Initialization $C$: $\sqrt{(x-128)^2+(y-128)^2}=114$. (b, c)Two intermediate iterations of the evolution (d) Final result of the curve evolution. (e) The piecewise constant approximation for the initialization. (f, g) The approximation model of the intermediate steps. (h) The piecewise constant approximation model for the image. Final result is obtained after 9 iterations, cpu = 2203 μs with $\mu=0.1\times 255^2$.

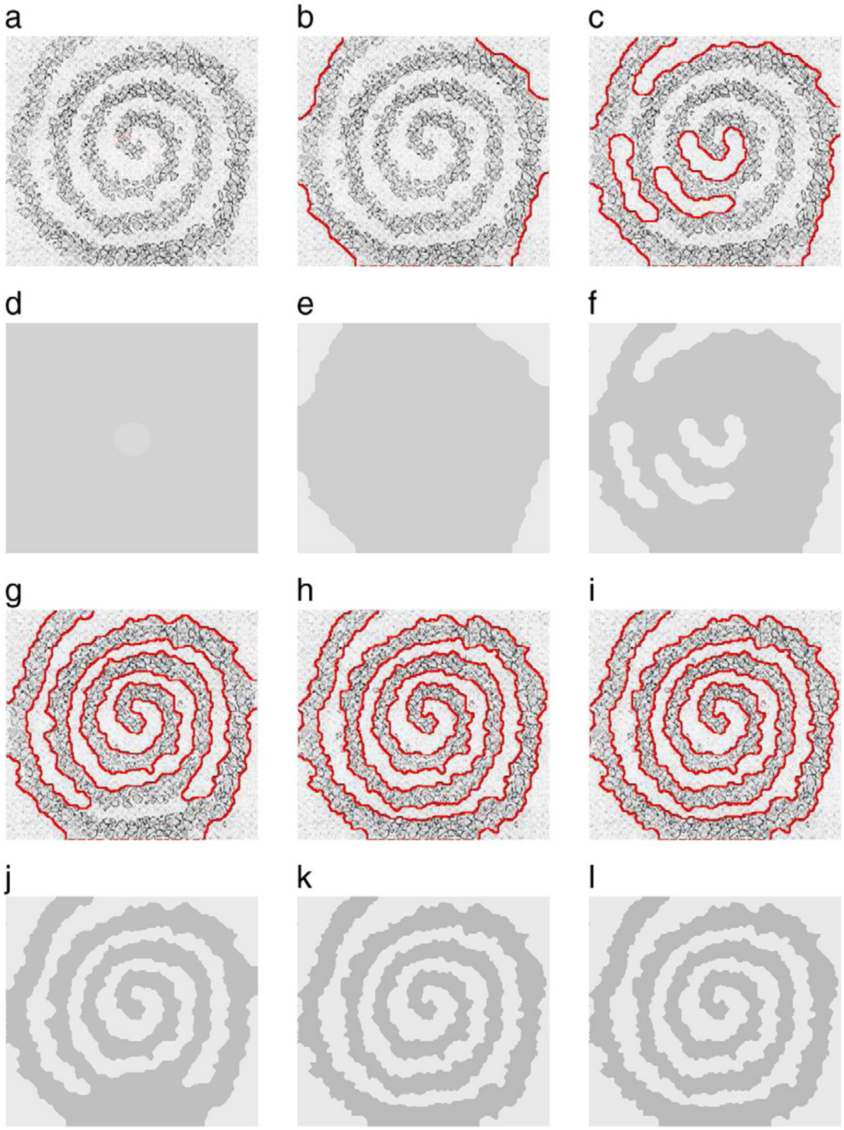

**Fig. 5.** Spiral — illustration of robustness to undefined edges, size = 234×191. (a, b) Initialization $C:\sqrt{(x-117)^2+(y-95)^2}=25$, and the initialization model, respectively. (c, e, g, i) All the intermediate iterations of the evolution. (d, f, h, j) The approximate model after each of the intermediate iterations. (k, l) Final result of the evolution and the image model. Final result is obtained after 9 iterations, cpu = 2944 μs.

The constants $c_1$ and $c_2$ can be represented in discrete form in the same manner:

$$c_1=\frac{\sum_p u(p)x_p}{\sum_p x_p}, \tag{15}$$

$$c_2=\frac{\sum_p u(p)\left(1-x_p\right)}{\sum_p\left(1-x_p\right)}. \tag{16}$$

To represent the length of the contour (the first term in Eq. (5)) on a graph, the discrete representation introduced by Kolmogorov and Boykov [24,25] reviewed in the Section 2.2 is used. Eq. (12) has been used with 8-point neighborhood system shown in Fig. 2 and $\Delta\rho$ was set to one in all direction. Let $\varepsilon_k=\{e_1,e_2,e_3,e_4\}$ be the set of edges in the 8-point neighborhood system and $w_1,w_2,w_3$ and $w_4$ be the edge costs assigned to $e_1,e_2,e_3$ and $e_4$ respectively. Notice that $|e_1|=|e_3|=1$ and $|e_2|=|e_4|=\sqrt{2}$, $w_1=\frac{\pi}{8}$, $w_2=\frac{\pi}{8\sqrt{2}}$, $w_3=\frac{\pi}{8}$ and $w_4=\frac{\pi}{8\sqrt{2}}$. To calculate $n_k$, we introduce a detector function $d(p,q)$ that detects whether the line connecting the two pixels p and q intersects the contour or not. It is obvious that the line intersects the contour *if and only if* $x_p$ and $x_q$ have different labels, hence the detector function can be expressed as:

$$d(p,q)=x_p\left(1-x_q\right)+x_q\left(1-x_p\right). \tag{17}$$

Combining this with Eq. (12), the contour length can be rewritten as

$$\|C\|_E=\sum_{e_{pq}\in\varepsilon_k} w_{pq}d(p,q). \tag{18}$$

The final discrete form of the energy function is expressed as

$$F(x_1,\ldots x_n)=\mu\sum_{e_{pq}\in\varepsilon_k} w_{pq}\left(x_p\left(1-x_q\right)+x_q\left(1-x_p\right)\right)+\nu\sum_p x_p \\ +\lambda_1\sum_p|u(p)-c_1|^2x_p+\lambda_2\sum_p|u(p)-c_2|^2\left(1-x_p\right). \tag{19}$$

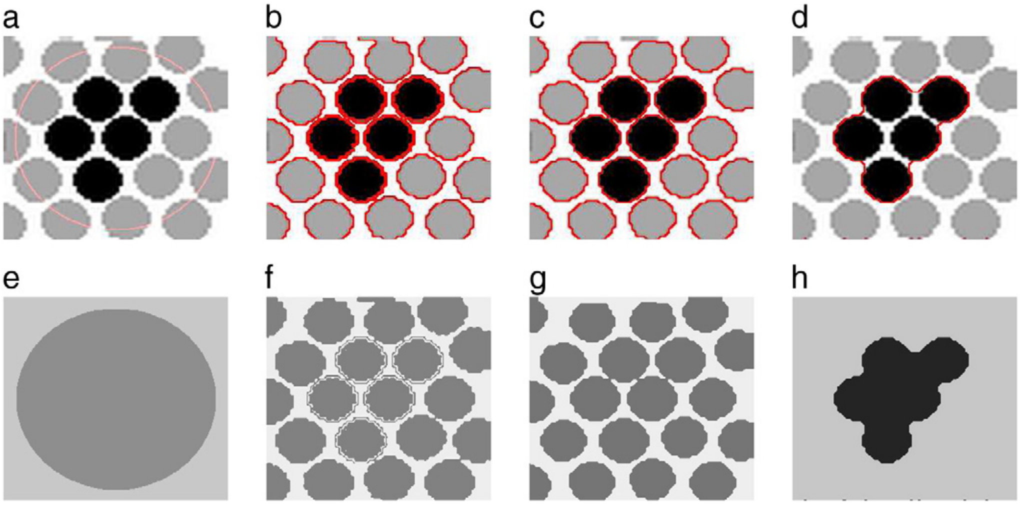

**Fig. 6.** Grouping — illustration of robustness to topology changes and effect of the regularization term on grouping objects. (a) Initialization $C$: $\sqrt{(x-128)^2+(y-128)^2}=114$. (b, c) Two intermediate iterations of the evolution (d) Final result of the curve evolution. (e) The piecewise constant approximation for the initialization. (f, g) The approximation model of the intermediate steps. (h) The piecewise constant approximation model for the image. Final segmentation result is obtained after 4 iterations, cpu = 1121 μs with $\mu = 0.1 \times 255^2$.

To compare our segmentation results to the results in [11], we will use the same parameters values that they used. Hence, throughout this paper, we will fix $\lambda_1 = \lambda_2 = 1$, and $\nu = 0$. Meanwhile, the choice of these values for the weighting coefficient is intuitive. Since there is no prior information about the foreground or the background, then weighting them equally is the most natural choice. In fact, $\lambda_1$ and $\lambda_2$ can be totally dropped from the formulation as the relative importance of the regularization to the data smoothness will not be affected, however, we would like to leave the formulation in its most general form and highlight that $\lambda_1$ and $\lambda_2$ can be calculated from prior information depending on the application of interest.

### 3.2. Graph representation

The major challenge in using graph cuts as an optimization tool in classification problems is to prove that the energy function being optimized can be represented using a discrete lattice in which the cost of the minimum cut reflects the value of the optimal solution for the energy minimization problem. The classification problem is then solved by assigning all the vertices attached to the source a label that corresponds to the first class and the rest of the vertices will be assigned another label that corresponds to the second class.

In this subsection, we will prove that the energy function in Eq. (19) can be minimized using graph cuts. The results in [26], suggested that any $F^2$ class function of $n$ binary variables represented as

$$E(x_1, \ldots, x_n) = \sum_p E^p\left(x_p\right) + \sum_{p<q} E^{p,q}\left(x_p, x_q\right) \tag{20}$$

is graph representable if and only if it is a submodular function, i.e. it satisfies the condition

$$E^{p,q}(0,0) + E^{p,q}(1,1) \le E^{p,q}(0,1) + E^{p,q}(1,0). \tag{21}$$

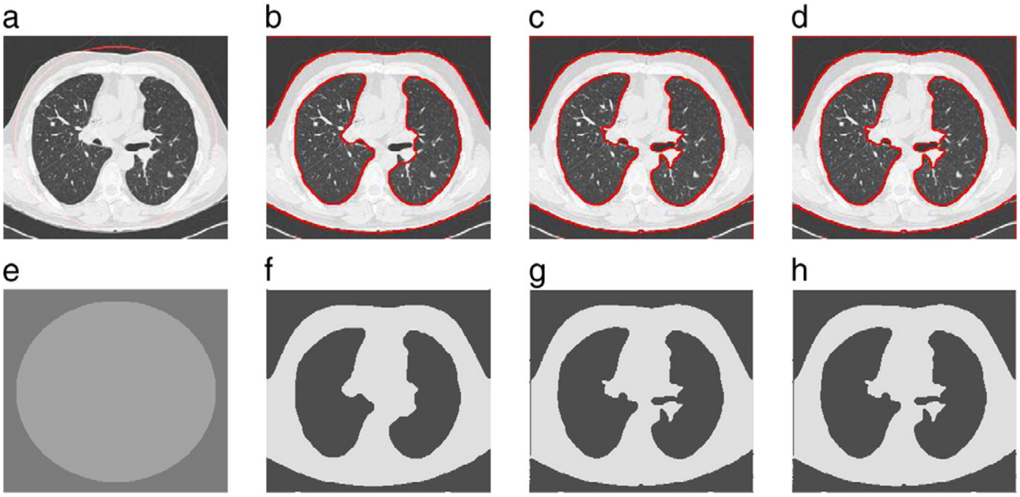

**Fig. 7.** Lung MRI — lung segmentation from the chest MRI. (a) Initialization $C$: $\sqrt{(x-128)^2+(y-128)^2}=114$. (b, c) Two intermediate iterations of the evolution (d) Final result of the curve evolution. (e) The piecewise constant approximation for the initialization. (f, g) The approximation model of the intermediate steps. (h) The piecewise constant approximation model for the image. cpu = 1041 μs with $\mu = 0.5 \times 255^2$.

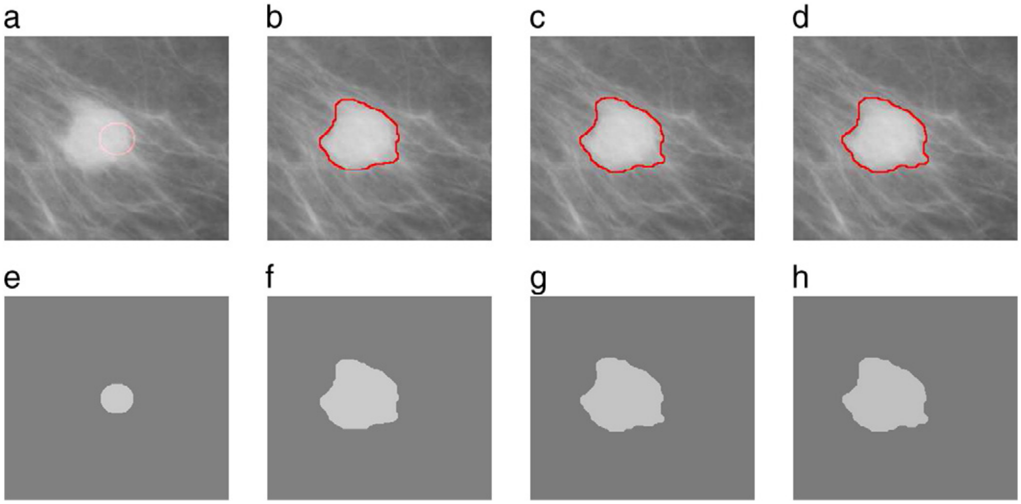

**Fig. 8.** Mammogram/Initialization 1 — detecting the edge of a mass in a mammogram. (a) Initialization $C$: $\sqrt{(x-128)^2+(y-128)^2}=20$. (b, c) Two intermediate iterations of the evolution (d) Final result of the curve evolution. (e) The piecewise constant approximation for the initialization. (f, g) The approximation model of the intermediate steps. (h) The piecewise constant approximation model for the image. Final segmentation result is obtained after 5 iterations, cpu = 1041 μs with $\mu = 0.1\times255^2$.

The correspondence between Eq. (20) and our discrete formulation (Eq.(19)) shows that our energy function is an $F^2$ class function with

$$E^p\left(x_p\right) = \lambda_1|u(p)-c_1|^2 x_p + \lambda_2|u(p)-c_2|^2\left(1-x_p\right) \tag{22}$$

$$\begin{aligned} E^{p,q}\left(x_p,x_q\right) &= \left(x_p\left(1-x_q\right)+x_q\left(1-x_p\right)\right)w_{pq} \\ &= \left(x_p+x_q-2x_px_q\right)w_{pq}, \end{aligned} \tag{23}$$

where $w_{pq}$ is the edge weight of the edge joining pixels $p$ and $q$ (Notice that each pixel has a corresponding vertex on the graph). The verification of the submodularity is straightforward and follows immediately from the definition of $E^{p,q}$. Since $E^{p,q}$ has a nonzero value *if and only if* $x_p \neq x_q$, we have $E^{p,q}(0,0) = E^{p,q}(1,1) = 0$ and $E^{p,q}(0,1)$ and $E^{p,q}(1,0)$ are always positive because $w_{pq}$ is always positive. Hence, the submodularity constraint described in Eq. (21) is always satisfied.

### 3.3. Graph construction and pixel labeling

Having proved the submodularity of the proposed energy function, we can construct the graph and solve for the minimum cut. We adopted the graph construction procedure introduced in [26]. We construct a graph $\mathcal{G} = (\mathcal{V}, \varepsilon)$. The set of vertices $\mathcal{V}$ consists on $n+2$ vertices, $\mathcal{V}_o = \{v_1, v_2, ..., v_n\}$ vertices correspond to the binary variables $x_p$ for all $p \in \{1,2,...,n\}$ that represent the $n$ pixels of the image of interest. Two auxiliary vertices $\mathcal{S}$ and $\mathcal{T}$ that will correspond, later, to the classification labels 0 and 1 are added.

The edge weights are defined using the energy function formulation in Eqs. (22) and (23). For each pixel $p = (x,y)$ and using the initial values for $c_1$ and $c_2$, we calculate $(u(p)-c_1)^2$ and $(u(p)-c_2)^2$. If $(u(p)-c_1)^2 > (u(p)-c_2)^2$, we add an edge $\mathcal{S}v_p$ with weight $(u(p)-c_1)^2$. Otherwise, we add an edge $v_p\mathcal{T}$ with weight $(u(p)-c_2)^2$. The previous weighted edges represent the external energy in the curve evolution problem. The internal energy is represented using the pixel interaction that represents the length of the contour. The interaction between two neighboring pixels $p$ and $q$ (in the 8 neighborhood system described earlier) is calculated using Eq. (23), since $E^{p,q}(0,0)$ and $E^{p,q}(1,1)$ are always zero so we add an edge $v_pv_q$ with weight $w_{pq}$. The value of $w_{pq}$ depends on the on the relative position of $q$ with respect to $p$ as discussed in the Section 3.1.

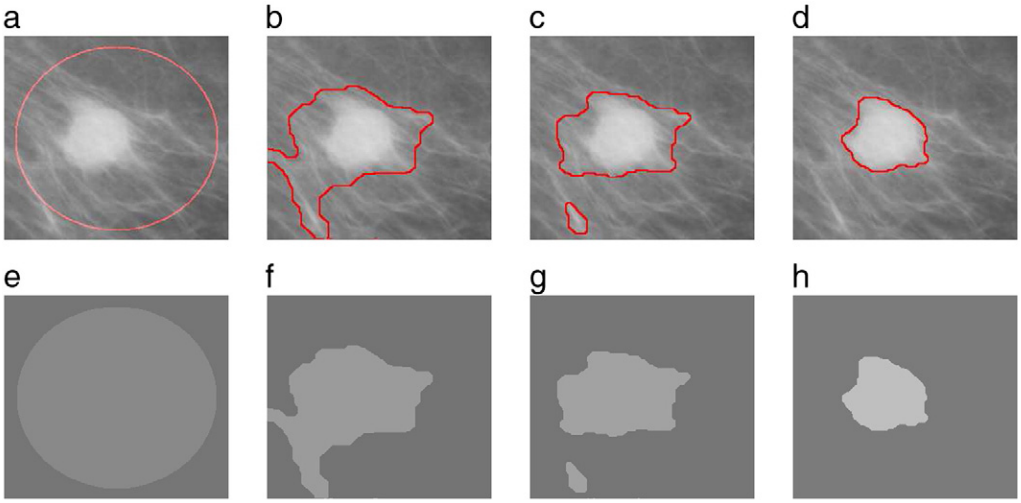

**Fig. 9.** Mammogram/initialization 2 — detecting the edge of a mass in a mammogram. (a) Initialization $C$: $\sqrt{(x-128)^2+(y-128)^2}=114$. (b, c) Two intermediate iterations of the evolution (d) Final result of the curve evolution. (e) The piecewise constant approximation for the initialization. (f, g) The approximation model of the intermediate steps. (h) The piecewise constant approximation model for the image. Final segmentation result is obtained after 13 iteration, cpu = 2724 μs with $\mu = 0.1\times255^2$.

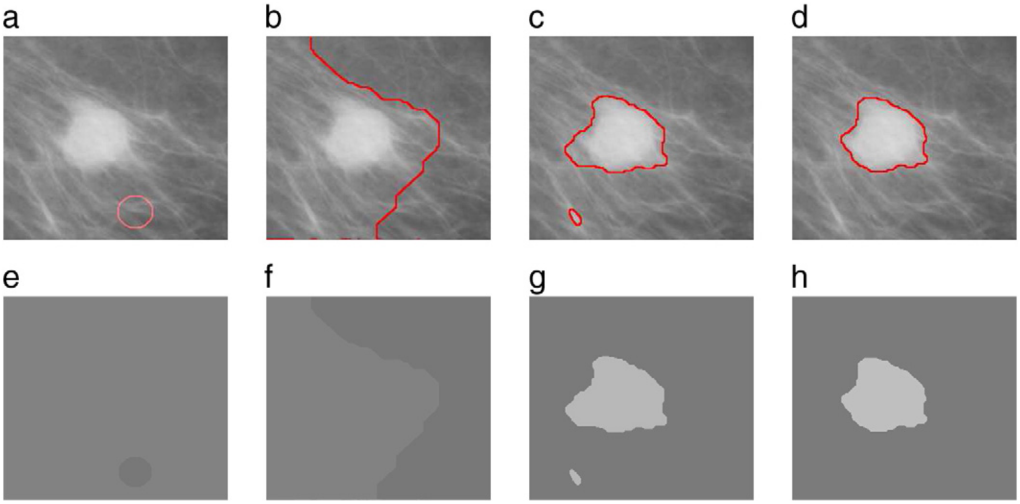

**Fig. 10.** Mammogram/Initialization 3 — Detecting the edge of a mass in a mammogram. ((a) Initialization $C$: $\sqrt{(x-150)^2+(y-220)^2}=20$. (b, c) Two intermediate iterations of the evolution (d) Final result of the curve evolution. (e) The piecewise constant approximation for the initialization. (f, g) The approximation model of the intermediate steps. (h) The piecewise constant approximation model for the image. Final segmentation result is obtained after 13 iterations, cpu = 34705 µs with $\mu=0.1\times255^2$.

The algorithm can be summarized as follows:

1. Initialize the contour C anywhere in the image. For each $p=(x,y)\in\Omega$, the binary variable $x_p$ is initialized by assigning 0 if $p$ is outside $C$ and 1 if $p$ is inside $C$.
2. Calculate $c_1$ and $c_2$ using Eqs. (15) and (16).
3. For each pixel $p=(x,y)\in\Omega$
   - Calculate $E^1(x_p)=(u(p)-c_1)^2$ and $E^0(x_p)=(u(p)-c_2)^2$.
   - if $E^1(x_p)>E^0(x_p)$ add an edge $\mathcal{S}v_p$ with weight $(u(p)-c_1)^2$, otherwise add an edge $v_p\mathcal{T}$ with weight $(u(p)-c_2)^2$.
   - For each pixel $q$ in the 8-neighborhood system of $p$ calculate $w_{pq}$ using Eq. (12) and add an edge $v_pv_q$ with weight $w_{pq}$.
4. Solve the graph $\mathcal{G}$ for the minimum $\mathcal{ST}$ cut, $\mathcal{C}$ that partitions the vertex set into two disjoint sets $S$ and $\overline{S}$. Obtain the new labels $x_p$ for all $p$
   - If $v_p\in S$, then $x_p$ is given a label 0.
   - If $v_p\in\overline{S}$, then $x_p$ is given a label 1.
5. Using the new labels obtained from the previous step, update $c_1$ and $c_2$ using Eqs. (18) and (19).
6. Repeat steps 2–5 until $c_1$ and $c_2$ become fixed and the energy is minimized.
7. The piecewise constant approximation can be obtained as $u=c_1x_p+c_2(1-x_p)$.

### 3.4. Computational complexity

The combinatorial optimization literature provides several min-cut/max-flow algorithms with polynomial time complexity. The graph construction and min-cut/max-flow introduced by Kolmogorov and Zabih [26] has been adopted here. The algorithm is practically 2 to 5 times faster than the state of the art min-cut/max-flow algorithm such as Ford–Fulkerson method and Goldberg–Tarjan method. We refer the reader to [27] for a detailed comparison. The proposed method in this paper has the same polynomial time complexity as the method in [27]. The only concern in analyzing the complexity of our algorithm is to decide relationship between the number of iterations and the size of the image. To address this concern, the mammogram image has been resampled to 64×64, 128×128, 256×256, 512×512 and 1024×1024. Each of the resized images has been segmented using $\lambda_1=1$, $\lambda_2=1$, $\mu=0.1\times255^2$, and the initialized contour is $C:\sqrt{(x-x_o)^2+(y-y_o)^2}=25$ where $(x_o,y_o)$ is the center pixel of each of the resized images; (32, 32), (64, 64), (128, 128), (256, 256) and (512, 512), respectively. In all of the images (except the first one), the energy is minimized and the contour of the object of interest has been reached after 10 iterations. The first image took 3 more iterations to converge and this is because of the relative location of the initial contour with respect to the object of interest but not really relevant to the size of the image. Hence, this experiment shows that the number of iterations necessary for convergence is independent upon the size of the data set and hence our iterative algorithms still preserves the polynomial time complexity.

## 4. Vector valued image segmentation

This section introduces a natural extension of the proposed segmentation model in Section 3 to handle vector valued images. Let $u(x,y,i)$ be the intensity value of the spatial coordinates $(x,y)$ in the $i$th channel with $i\in\{1,2,...,N\}$ where $N$ is the number of channel in the case of interest. The different channels contain the same image with different information, for example, the R-G-B components of a color image or different wavelength at which the image is captured. $\overline{C}_1=(c_{11},c_{21},...,c_{N1})$ is a vector of the mean intensity value inside the contour $C$ for the different channels. Similarly, $\overline{C}_2=(c_{11},c_{22},...,c_{N2})$ is a vector that represents the mean intensity values outside the contour $C$ for the different channels. Then, the extension to the vector valued case can be represented by minimizing the following energy function

$$\begin{aligned}F\left(x_1,x_2,\ ....,x_n,\overline{C}_1,\overline{C}_2\right) &= \mu\ length(C)\\ &+\sum_{p\in\Omega}\left(\frac{1}{N}\sum_{i=1}^{N}\lambda_{i1}|u(x,y,i)-c_{i1}|^2\right)x_p\\ &+\sum_{p\in\Omega}\left(\frac{1}{N}\sum_{i=1}^{N}\lambda_{i2}|u(x,y,i)-c_{i2}|^2\right)\left(1-x_p\right)\end{aligned}\tag{24}$$

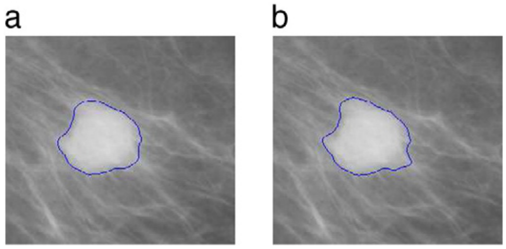

**Fig. 11.** Detecting the edge of a mass in a mammogram. (a) Initial Contour is entirely inside the mass, cpu = 73735 ms. (b) Initial contour is outside the mass, cpu = 541072 ms.

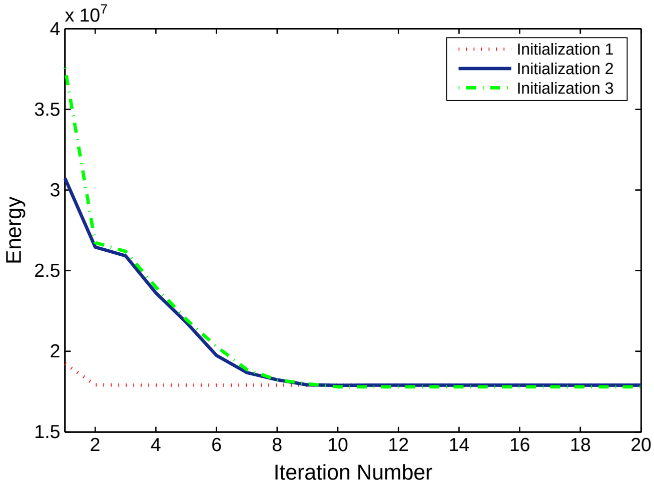

**Fig. 12.** Illustration of the energy minimization in the mammogram image for three different initializations. Initialization 1 $C:\sqrt{(x-128)^2+(y-128)^2}=20$, Initialization 2 $C:\sqrt{(x-128)^2+(y-128)^2}=114$ and Initialization 3 $C:\sqrt{(x-150)^2+(y-220)^2}=20$.

where $\lambda_{i1}$ and $\lambda_{i2}$ are constant parameters associated with each channel. The graph construction to minimize the fitting energy function in Eq. (24) is performed in the same way presented in the Section 3.3. The only difference is in the t-links edge weights because the t-links edge weights will be derived from $N$ channels rather than one in the scalar value. More precisely, for each pixel $p=(x,y)\in\Omega$, $E_1$ and $E_2$ will be calculated as: $E_1(x_p)=\frac{1}{N}\sum_{i=1}^{N}(u(x,y,i)-c_{i1})^2$ and $E_0(x_p)=\frac{1}{N}\sum_{i=1}^{N}(u(x,y,i)-c_{i2})^2$.

Then if $E^1(x_p)>E^0(x_p)$, an edge $\mathcal{S}v_p$ with weight $E^1(x_p)$ is added to the graph, otherwise an edge $v_p\mathcal{T}$ with weight $E^0(x_p)$ is added. The min $\mathcal{ST}$ cut is found and the values of $x_p$ for all $p$ are updated. The values of $\overline{C}_1$ and $\overline{C}_2$ are updated according to the new labels using the following formulas:

$$c_{i1}=\frac{\sum_{p\in\Omega}u(x,y,i)x_p}{\sum_{p\in\Omega}x_p} \tag{25}$$

$$c_{i2}=\frac{\sum_{p\in\Omega}u(x,y,i)\left(1-x_p\right)}{\sum_{p\in\Omega}\left(1-x_p\right)}. \tag{26}$$

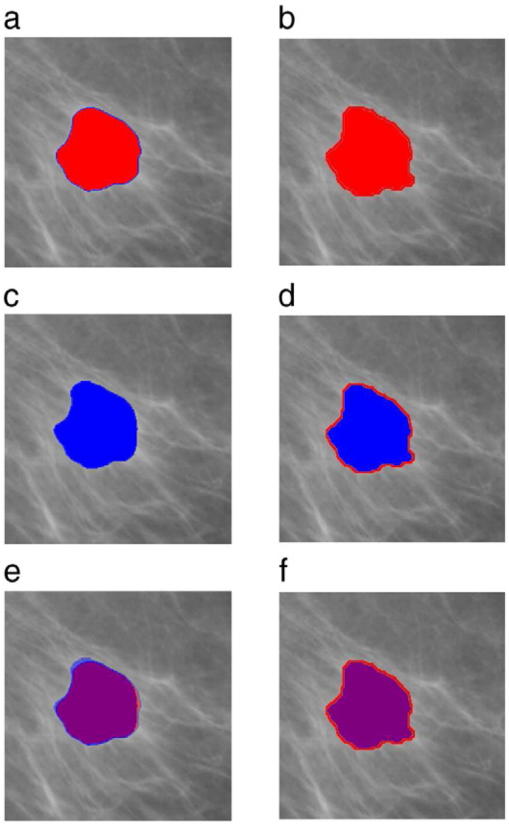

**Fig. 14.** Similarity between the segmentation results when different initializations are used in both level set framework and graph cut framework. Column 1 shows the result of the level set. Column 2 shows the result of our approach. (a, b) Segmentation results when the contour is initialized inside the mass. (c, d) Segmentation results when the contour is initialized outside the mass. (e, f) The overlap between the segmentation results given the two different initializations.

After minimizing the energy, the piecewise constant model approximation is estimated by calculating the norm of the mean values inside and outside the contour resulting in the following image model:

$$u(x,y)=\sqrt{\frac{\sum_{i=1}^{N}c_{i1}^2}{N}}x_p+\sqrt{\frac{\sum_{i=1}^{N}c_{i2}^2}{N}}\left(1-x_p\right). \tag{27}$$

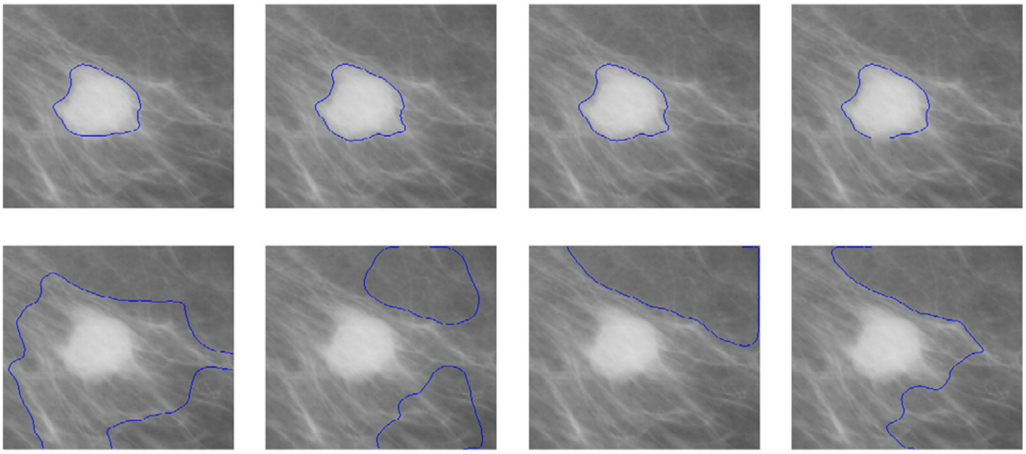

**Fig. 13.** Segmentation results obtained using Chan–Vese when 8 different initializations were used.

**Table 1**
Quantitative assessment of the robustness of our approach to initialization.

| | Graph cuts | Level sets |
|---|---|---|
| Number of cases the algorithm returned trivial solution | 0 | 10 |
| Number of cases the algorithm diverges from the correct solution | 0 | 18 |
| Number of cases the algorithm returned an acceptable solution | 50 | 32 |
| Average SI of the successful cases | 100% | 98.22% |
| Standard deviation of SI of the successful cases | 0 | 1.47% |

## 5. Volumetric image segmentation

This section presents another natural extension for the proposed approach in Section 3 to handle the segmentation of 3D images. The extension is straight forward because we will follow almost the exact same algorithm, however, the problem description is slightly different because of the dimensionality of the domain of interest. Thereby, we will introduce the formalization of the problem in 3D and highlight the differences from the 2D case. The problem can be described as follows:

Given a 3D volume of interest $u:\Omega\rightarrow\mathbb{R}$, where $\Omega$ is an open bounded subset in $\mathbb{R}^3$ that consists of several connected components $\omega_i$, the problem of interest is to evolve a surface $S$ in $\Omega$ until it stops at the boundaries of interest $\omega$ in $\Omega$, i.e. the objective is to obtain the surface $S_o=\partial\omega$. Hence, the variational formulation in a level set framework can be described as follows:

$$\begin{aligned} F(c_1,c_2,\phi) = {} & \mu\int_\Omega \delta(\phi(x,y,z))|\nabla\phi(x,y,z)|dxdydz \\ & + \nu\int_\Omega H(\phi(x,y,z))dxdydz \\ & + \lambda_1\int_\Omega |u(x,y,z)-c_1|^2 H(\phi(x,y,z))dxdydz \\ & + \lambda_2\int_\Omega |u(x,y,z)-c_2|^2(1-H(\phi(x,y,z)))dxdydz, \end{aligned} \tag{28}$$

The discrete formulation for the energy function in Eq. (28) can also be described by Eq. (29)

$$\begin{aligned} F(x_1,\ \ldots x_n) = {} & \mu\sum_{p,q\in e_k} w_k\left(x_p\left(1-x_q\right)+x_q\left(1-x_p\right)\right)+\nu\sum_p x_p \\ & +\lambda_1\sum_p|u(p)-c_1|^2x_p+\lambda_2\sum_p|u(p)-c_2|^2\left(1-x_p\right). \end{aligned} \tag{29}$$

with two fundamental differences: First, the open set $\Omega$ is a three dimensional set which changes the context of using the equation. Second, the weights $w_{pq}$ are different because a different neighborhood system is used to describe the discrete representation of the surface area. For this purpose, a neighborhood system that consists of 26 neighbors is used. Each edge angular orientation is determined by the spherical angles $\Theta_{pq}=\{\psi_{pq},\theta_{pq}\}$. Fig. 3 shows the system used to approximate the surface area.

**Table 2**
Comparison of the speed between our discrete implementation and the classical implementation introduced in [11].

| | Graph cuts | | Gradient descent | |
|---|---|---|---|---|
| Image | Iterations | Total CPU time | Iterations | Total CPU time |
| Topology | 9 | 2203 µs | 1845 | $50.850\times10^6$ µs |
| Spiral | 9 | 2944 µs | 16421 | $653.24\times10^6$ µs |
| Grouping | 4 | 1121 µs | 354 | $95.200\times10^6$ µs |
| Lung | 4 | 1041 µs | 6910 | $334.08\times10^6$ µs |
| Mammogram 1 | 5 | 1041 µs | 2312 | $73.735\times10^6$ µs |
| Mammogram 2 | 13 | 2724 µs | 14793 | $541.072\times10^6$ µs |

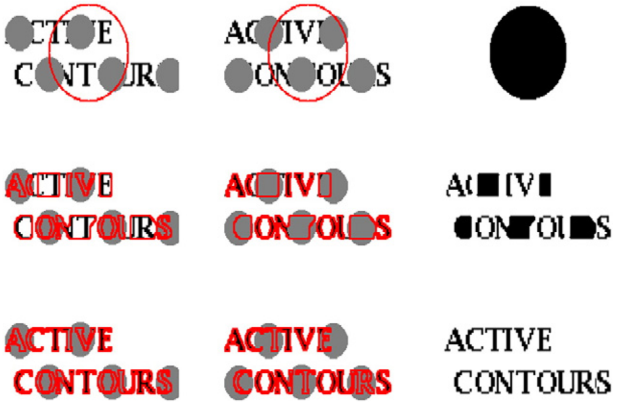

**Fig. 15.** Active contours, image size = 188 × 98 — Segmenting the word ACTIVE CONTOURS from two channels each channel has some occlusion. Columns 1, 2 and 3 represent channel 1, channel 2, and the image model, respectively. The first row is the initialization $C:\sqrt{(x-94)^2+(y-49)^2}=40$ for channels 1 and 2 and the initialization model. The second row is an intermediate step of the evolution and the third row is the image model. Final segmentation result is obtained after 4 iterations, cpu = 1532 µs with $\lambda_{i1}=1$, $\lambda_{i2}=0.7$ and $\mu=0.001\times255^2$.

The weights $w_{pq}$ are calculated as follows:

$$w_{pq} = \frac{\delta^3\Delta\Theta_{pq}}{\pi|e_{pq}|} \tag{30}$$

where $\Delta\Theta_{pq}=\Delta\psi_{pq}\Delta\theta_{pq}$ corresponds to the partitioning of the unit sphere among the angular orientation of the edges of the neighborhood system shown in Fig. 3.

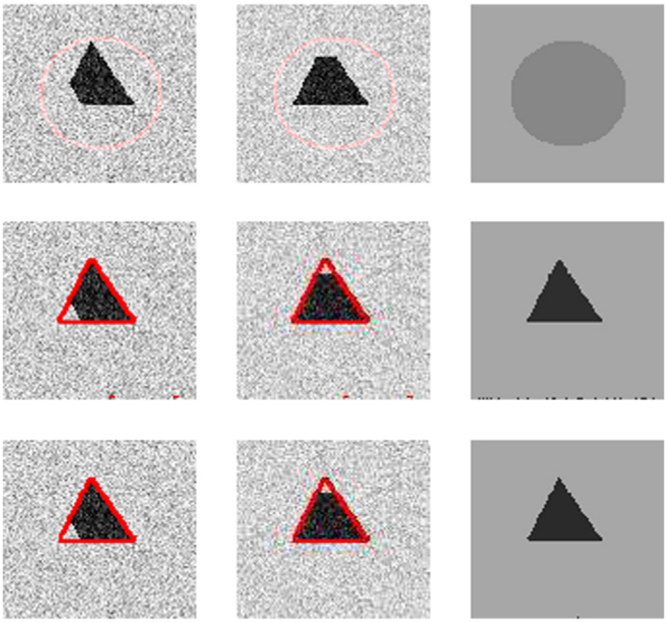

**Fig. 16.** Triangles, image size = 100 × 100 — segmenting the triangle from two channels each channel has a missing corner. Columns 1, 2 and 3 represent channel 1, channel 2, and the image model, respectively. The first row is the initialization $C:\sqrt{(x-50)^2+(y-50)^2}=40$ for channels 1 and 2 and the initialization model. The second row is an intermediate step of the evolution and the third row is the image model. Final segmentation result is obtained after 5 iterations, cpu = 1496 µs with $\lambda_{i1}=0.7$, $\lambda_{i2}=1$ and $\mu=0.05\times255^2$.

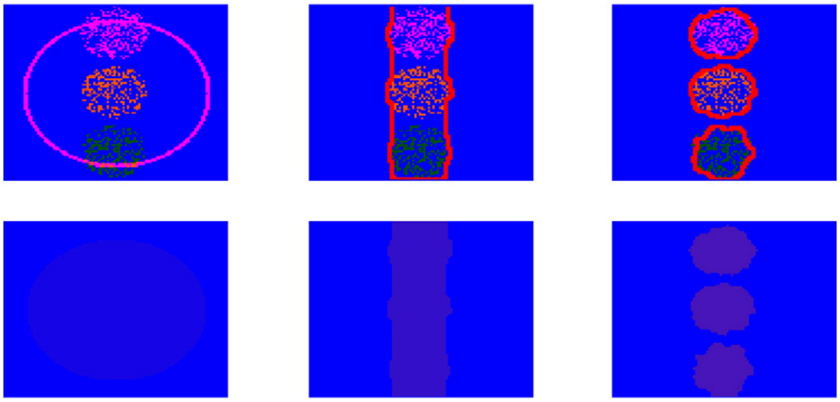

**Fig. 17.** Lights, image size = 100 × 100 — segmenting a color image. Rows 1 and 2 represent the evolution steps and the corresponding approximation model, respectively. The first column is the initialization $C:\sqrt{(x-50)^2+(y-50)^2}=40$ superimposed on the image and the initialization model. The second column shows an intermediate step of the evolution and its model and the third column shows the final segmentation result and the image model. Final segmentation result is obtained after 4 iterations, cpu = 2384 μs with $\lambda_{i1}=1$, $\lambda_{i2}=1$ and $\mu=0.06\times255^2$.

## 6. Experimental results

This section introduces experimental results for several images. Preliminary results have been reported in [21]. Some of these images are synthetic images used to test the robustness of the algorithm to noise and topology changes. The others are real medical images used to show that the algorithm is generic and can be adopted and applied to several applications. We have applied the algorithm on some of the images published in [11]. We have also implemented the original level set formulation introduced in [11] and applied it on the chosen images on the same machine (2 GHz Intel Core Duo, 2GB RAM) to compare the speed of our algorithm and investigate whether graph cut optimization reduces the processing time or not. Most of the images were resized to 256 × 256, and we used $\lambda_1=\lambda_2=1$ and $\nu=0$ for numerical implementation. The value of $\mu$ changes according to the type of image and the object of interest. However, $\mu=0.1\times255^2$ was suitable for most of the applications.

In each of our results, the initialization, the final evolution results, as well as some of the intermediate steps of the evolution will be included. The piecewise constant approximation model associated with each image will also be shown. Fig. 4 shows the robustness of the algorithm to noise. Moreover, the initialized circle split to detect different shapes inside the image which illustrates the robustness of the algorithm to topology changes. Fig. 5 shows that the algorithm can work properly when the edges are blurred. The results in these two figures emphasize that the algorithm still preserves all the advantages of the original level set implementation presented in [11].

Fig. 6 shows the effectiveness of the algorithm in detecting the objects defined by grouping and shows the importance of the regularization. The transitions from (b) to (c) and from (c) to (d) show how the ripples of the contours of the black circles have been dramatically suppressed due to the minimization of the contour length. Fig. 7 illustrates the segmentation of the lungs from an MRI of the chest.

### 6.1. Insensitivity to initialization

Figs. 8–10 illustrate the stability of the graph cut optimization technique and the insensitivity to initialization when extracting the boundary of a breast mass from a cropped mammogram. Although the three initializations used are completely different, the algorithm returns the exact same minimum and the exact same boundary of the mass. On the contrary, when we used the same initializations and applied the classical level set implementation, the algorithm result in three different solutions. Fig. 11 shows two of them. When the initialization was very far from the object of interest, the classical level set implementation completely failed to detect the boundaries of the object. Fig. 12 shows the energy minimization over 20 iterations for the three different initializations shown in Figs. 8–10. It emphasizes

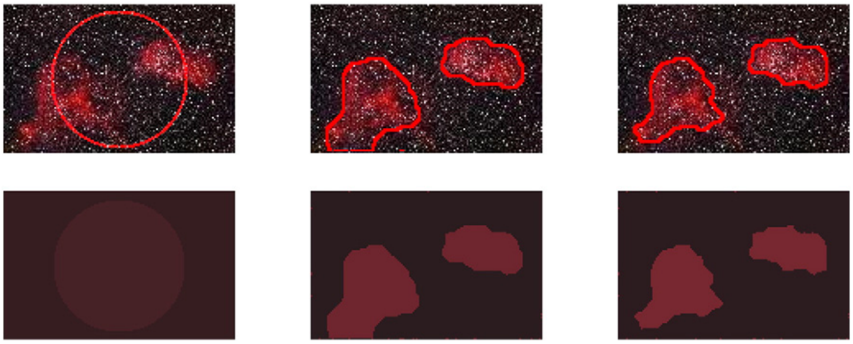

**Fig. 18.** Real color image, image size = 141 × 90 — segmenting a real color image with blurred boundaries and noise. Rows 1 and 2 introduce the evolutionsteps and the corresponding approximation model, respectively. The first column shows the initialization $C:\sqrt{(x-50)^2+(y-50)^2}=40$ superimposed on the image and the initialization model. The second column shows an intermediate step of the evolution and the corresponding model and the third row shows the final segmentation results and the image model. Final segmentation result is obtained after 6iterations, cpu = 2384 μs with $\lambda_{i1}=1$, $\lambda_{i2}=1$ and $\mu=0.03\times255^2$.

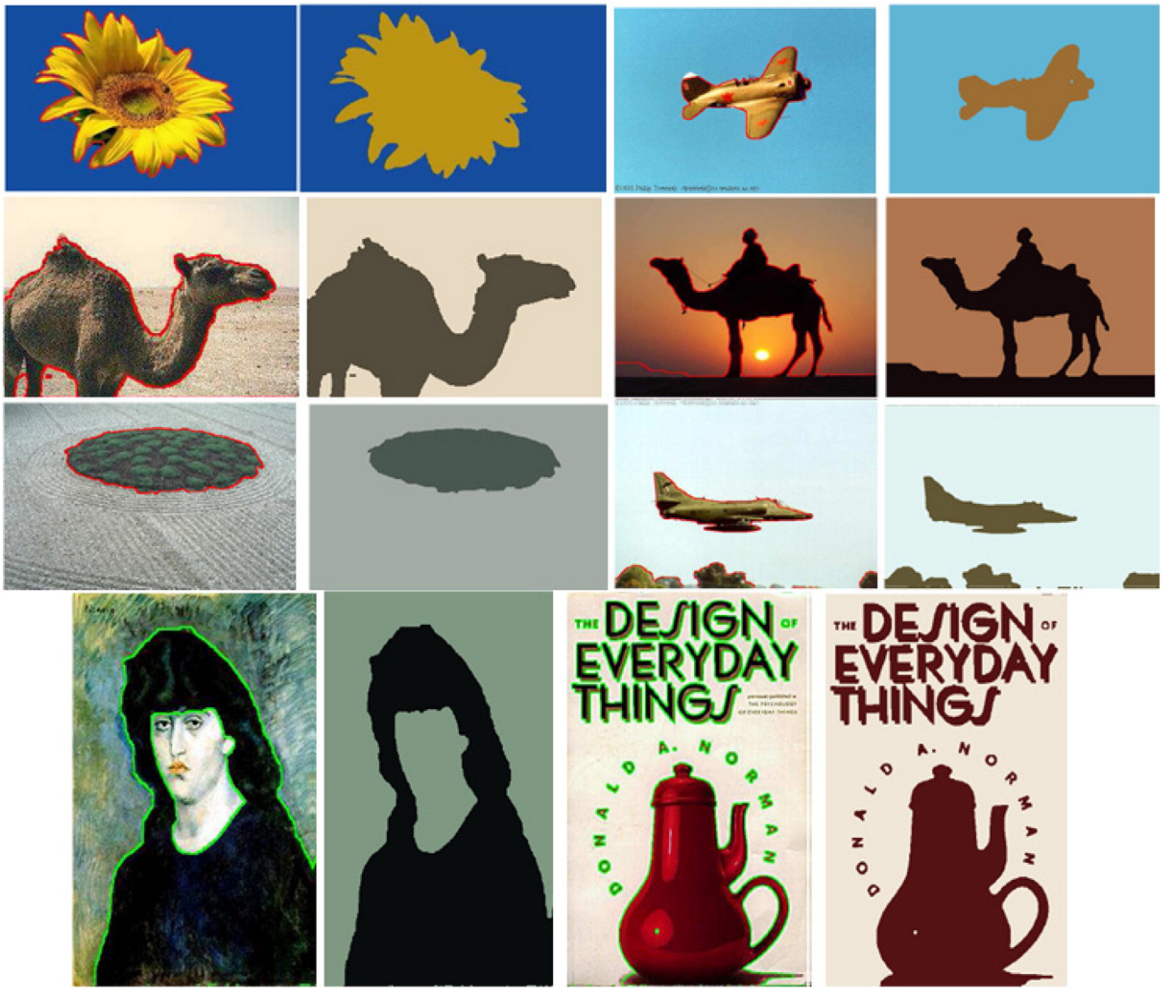

**Fig. 19.** Segmentation of a collection of RGB images from the Berkley segmentation data set and Caltech objects data set.

the insensitivity to initialization and the global optimization of the energy function regardless of the initial values of the binary variables.

In order to quantitatively assess the robustness of the algorithm to the initialization, we have conducted the mammogram segmentation experiment 50 times with 50 different initializations for both the level sets framework and our combinatorial optimization framework. In our experiments with level sets, we endeavored the parameters that provide the best possible segmentation.

Fig. 14 depicts a sample of our experiment. It shows an example of the overlap between two segmentation results obtained by the level set approach and the overlap between the corresponding segmentation results using the combinatorial optimization framework when the same initializations were used. It is obvious that even if the level set approach succeeds to provide a segmentation close to the boundary of the mass, it does not perform consistently when different initializations are used.

For quantitative assessment, we have consulted an expert for the correct segmentation and for each result of the 50, we have calculated the Dice Similarity Index (SI) between the segmentation result and the correct segmentation.

For the combinatorial optimization framework, we get the exact same result each time we run the algorithm with 100% similarity.

For the level set framework, in 10 out of the 50 initializations (i.e. 20% of the cases), the algorithm totally failed in capturing the mass. The initial contour shrinks until it disappears. We have observed that this usually happens if the initial contour is far from the boundary of the mass such as the initialization depicted in Fig. 10. The contour may also disappear if it is totally contained by the mass but still far away from the correct mass boundary. For our calculations, we excluded the cases where the contour shrinks until it disappears which favors the level set framework. Figs. 13 shows 8 segmentation results of the Chan–Vese model when 8 different initialization were used. The bottom row exhibit examples where the segmentation results were very far from the appropriate segmentation. These cases that exhibit less than 90% Dice coefficient (8 out of the 50) were excluded from our quantitative assessment were also excluded from the calculations, once more, favoring the Chan–Vese model in the evaluation. For the rest of the cases, the average dice coefficient was 98.22% and the standard deviation was 1.47%. It is worth mentioning that these results were obtained by a careful choice of parameters for the Chan–Vese model and that some of them took more than 5 min to terminate. While the similarity measure for all the 50 cases in GC was 100% and all of our results were obtained in a fraction of the second.

Table 1 is a concise summary for this experiment.

### 6.2. Improvement in the processing time

To compare the efficiency of our algorithm (speed-wise) relative to active contour without edges [11], the model parameters were fixed in both implementations. Then both algorithms were applied to the same images at the same machine. Table 2 provides the number of iterations and the total cpu time for each of the images shown in

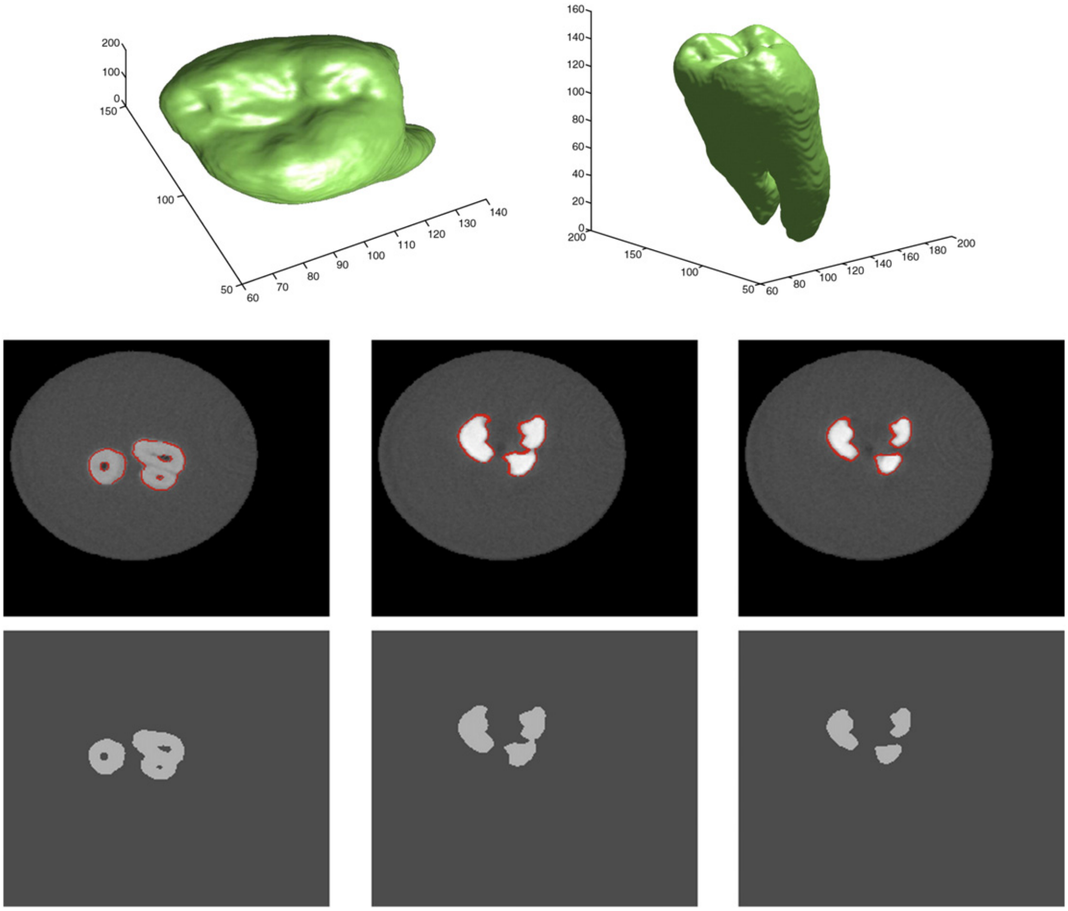

**Fig. 20.** 3D segmentation for a tooth. The first row shows the 3D results for our segmentation in two different views. The second row of the first, second and third columns shows the intersection of the final surface with the planes z = 135, z = 149 and z = 151, respectively, superimposed on the input data slices. The second row of the first, second and third rows shows the corresponding projection of the piecewise constant approximation at the same z levels. The final segmentation results were obtained after 3 iterations with $\lambda_1 = \lambda_2 = 1$ and $\mu = 0.001 \times 255^2$.

Figs. 4–10. The comparison shows that the numerical implementation using our approach is much faster than the level set framework of the active contour without edges model introduced in [11].

### 6.3. Vector valued image segmentation

The vector valued segmentation algorithm described in Section 4 has been applied to the same images used by Chan et al. in [28]. The results of applying our algorithm to these images illustrate two situations. First, segmentation of the object of interest from different channels where each channel has some missing information and the object can only be recovered by integrating the information from the different channels. Second, segmentation of R-G-B images where each color component represents a channel. In Fig. 15, the object of interest is the words ACTIVE CONTOURS, in each channel there is some occlusion. These words can be segmented properly by combining the information from the two channels.

Fig. 16 shows the segmentation of a triangle from two different channels, each channel has one missing corner, the two channels are superimposed with noise.

Fig. 17 shows the segmentation of a simulated color image. The image has three spots on a blue background. Each color component of the image contains only two spots and so as the corresponding intensity image obtained by the transformation 0.342 R + 0.5 G + 0.158 B (see [28] for more details and illustrations about the image construction) so it is impossible to segment the image using only one color component or the corresponding intensity image. Hence, we used the vector valued segmentation algorithm and the results are shown in Fig. 17. Fig. 18 shows the segmentation of a real color image. Finally, we have tested our algorithm on images from the Berkley Segmentation data set and the Caltech data sets. Fig. 19 shows a sample segmentation result of these images along with the piecewise constant approximation.

### 6.4. Volumetric segmentation

The volumetric image segmentation approach described in Section 5 has been applied to synthetic and real images. This section will present a sample of our results. We will show the 3D segmentation results for each data set and for illustrations we will also provide 2D projections of the segmentation superimposed on the 2D slices.

Fig. 20 shows the 3D segmentation of a tooth. The figure illustrates how the model is topology preserving, this is clear from the way it splits to capture the details of the root part of the tooth. Moreover, the first figure in the second row shows that the importance of topology preservation property extends to capturing the internal details of the tooth.[2]

Fig. 21 shows the 3D segmentation results of a daisy pollen grain, as well as, projections of the 3D segmentation at three different z

[2] If the internal details are not of a relevant interest in certain applications, $\mu$ can be adjusted to group the small details just as we have handled it in the 2D case illustrated in Fig. 6.

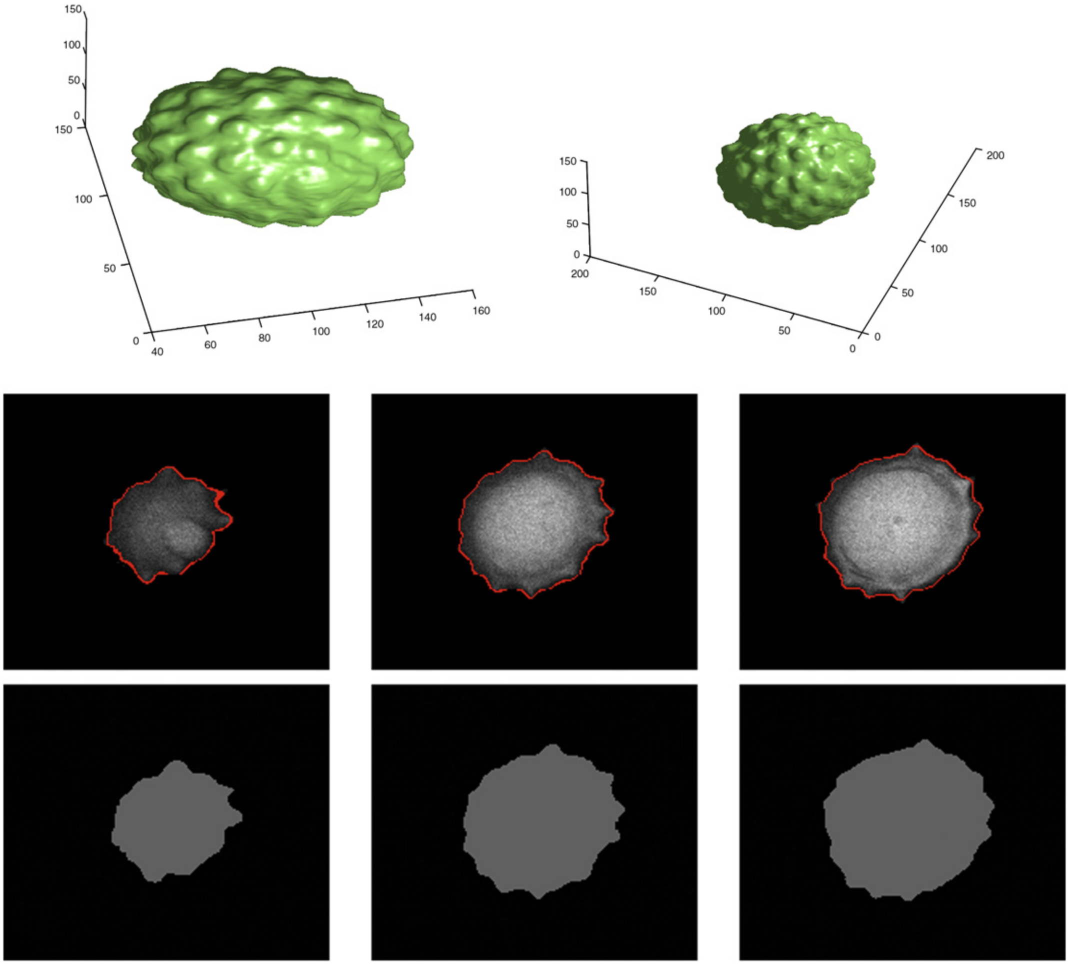

**Fig. 21.** 3D segmentation for a daisy pollen grain. Volume size is 190 × 190 × 165. The first row shows the 3D results for our segmentation in two different views. The second row shows the intersection of the final surface with the planes z = 43, z = 68 and z = 84, respectively, superimposed on the input data slices. The third row shows the corresponding projection of the piecewise constant approximation at the same z levels. $\mu = 0.1 \times 255^2$, the results were reached after 4 iterations.

levels superimposed on the input volume slices. Fig. 22 shows the segmentation of a spherical fullerene molecule (buckyball). The figure illustrates the robustness of the segmentation approach. The size of the input volume was 32 × 32 × 32 with very fine details and sudden transitions between the successive slices in the different directions, and yet the segmentation approach could capture these fine details as illustrated in the 3D model and the piecewise approximation shown in rows 2–5.

## 7. Discussion and conclusion

We have presented a novel numerical technique for front propagation. A discrete formulation of the Chan–Vese model has been presented. The submodularity of discrete energy function has been justified and hence the energy function was represented using a graph. Min-cut/max-flow has been used to optimize the energy function. The presented mathematical formulation handled the front propagation problem more efficiently than level sets algorithm in two main aspects. First the numerical implementation of the min-cut/max-flow algorithms has a polynomial time complexity which makes our model much faster than the level set algorithms. Second, a global optimum is always obtained regardless of the location of the initialization contour because of using graph cuts instead of gradient descent in the optimization stage. The results of the algorithm and the processing time recorded for both scalar and vector valued images are very promising. The algorithm can be adopted to improve the computational complexity of many existing segmentation algorithms in the literature.

### 7.1. Relationship to graph cuts segmentation methods

Graph cuts have been extensively used in the literature of image segmentation [18,27,29,30]. Here, we would like to highlight the difference between our approach and the currently existing graph based segmentation techniques. Generally, graph cut segmentation algorithms work by constructing a graph with submodular weights that reflect the location of the object boundary. Seed initializations (hard constraints in the graph construction) are used to initialize the foreground and max-flow/min-cut algorithm is used to find the cut that separates the seeds, often by cutting the edges that represent the highest gradient.

Our model shares some of the previous characteristics such as the construction of a graph with submodular weights and performing the optimization via finding the min cut of the graph. However, It differs from the graph cuts segmentation in different aspects:

1. We do not use seeds to initialize the optimization process and our model does not require user interaction. The initialization we used is arbitrary and can be placed anywhere in the image and does not highlight foreground and background as discussed in the results section.
2. our formulation is independent of the image gradient which makes it more robust to noisy, blurred and ill-defined edges.

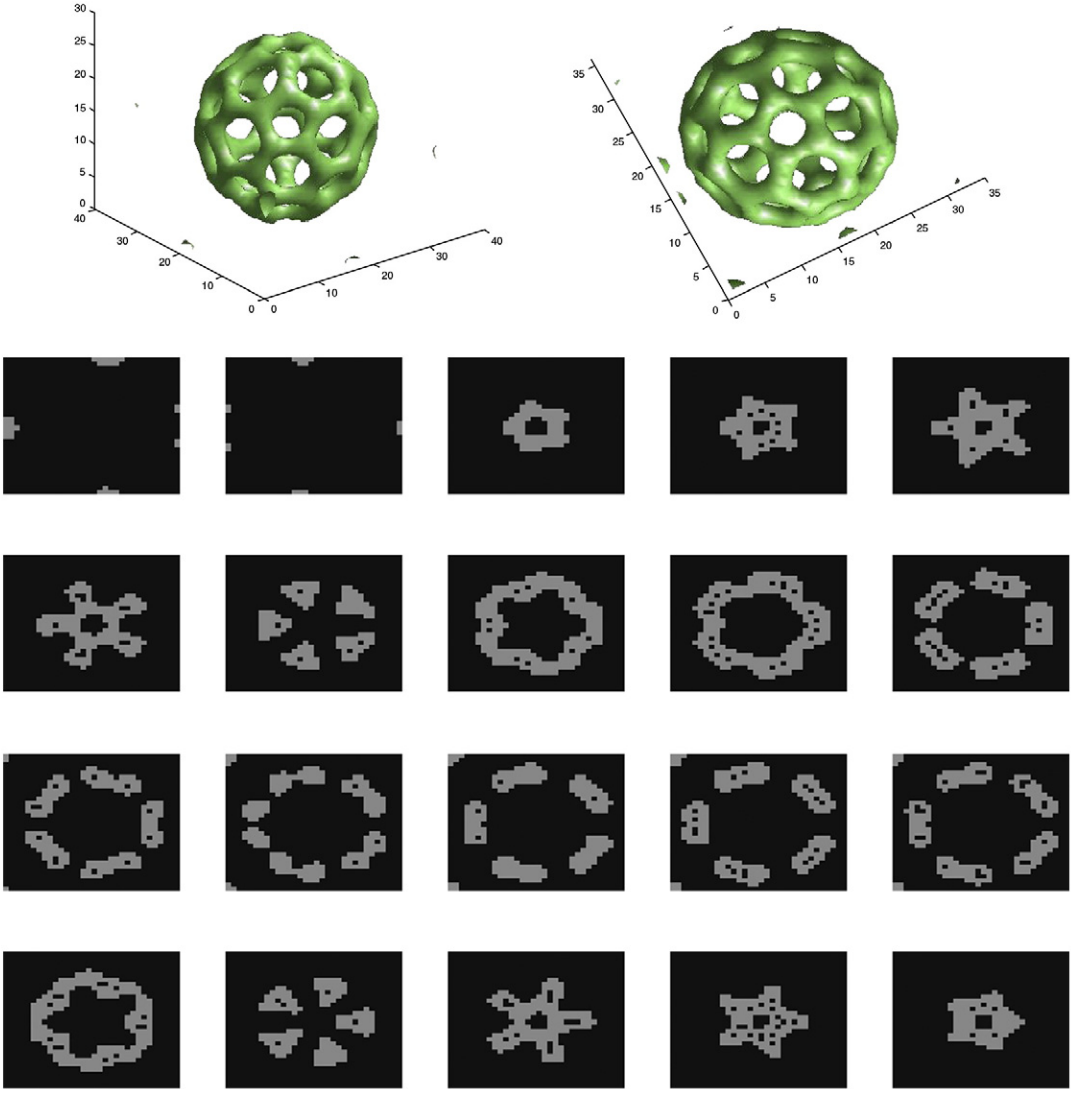

**Fig. 22.** Buckyball segmentation. Volume size = 32 × 32 × 32. The first row shows two different views for the 3D surface. Rows 2–4 show the peicewise constant approximation for the buckyball volume at 20 different z levels out of the given 32. $mu = 0.0001$, final results were reached after 3 iterations with CPU time = 592 ms.

3. In our algorithm, graph cuts is not the core of the segmentation. But, rather, it is one part of a larger scheme that achieves the segmentation task using the Mumford–Shah functional. It serves as an optimization tool that solves the Mumford–Shah model at every step of the evolution.

Although some studies such as [30] also use the graph optimization as a step in an iterative scheme, they use seed initialization in their algorithm that requires user interaction. Moreover, the edge weights they used are dependent on the image gradient.

### 7.2. Relationship to recent development in level sets segmentation models

In the past few years, several research studies such as [31–33] targeted the development of more efficient level set segmentation schemes based on the Mumford–Shah functional. The efforts of such papers are classifies in two categories: First, some studies target the improvement in the formulation itself. For example, Chan–Esedogulo–Nikolova [31] aim at presenting an equivalent convex representation for the Chan–Vese model to eliminate the local minima associated with the non convex nature of the Mumford–Shah. Second, other studies aim at providing fast implementation to optimize such energy functions. For example, the recent contribution of Goldstein, Bresson and Osher [33] applied the Split Bregman method to segmentation and reconstruction problems to provide much faster implementation than most (if not all) the currently existing level set methods. Here, we would like to highlight the differences between our model and the aforementioned contributions. The importance of solving the Mumford–Shah model on a discrete lattice is due to the following reasons:

1. Generally, level set segmentation uses small steps for contour evolution yielding very slow algorithms. Graph cut optimization techniques have polynomial time complexity when optimizing submodular functions (experimentally almost linear). our proposed segmentation is much faster than aforementioned level sets segmentation with a probable exception of the noticeable very recent contribution of Goldstein–Bresson–Osher [33] that achieved time close to ours for some of the images.
2. Most level set methods require several implementation choices (e.g. time step, discretization ...etc ) and the final results may be affected by the implementation parameters.
3. Last but not least, formulating the problem on a graph empowers the model by extending its applicability to several problems such as data clustering for example. The reason is that our model does

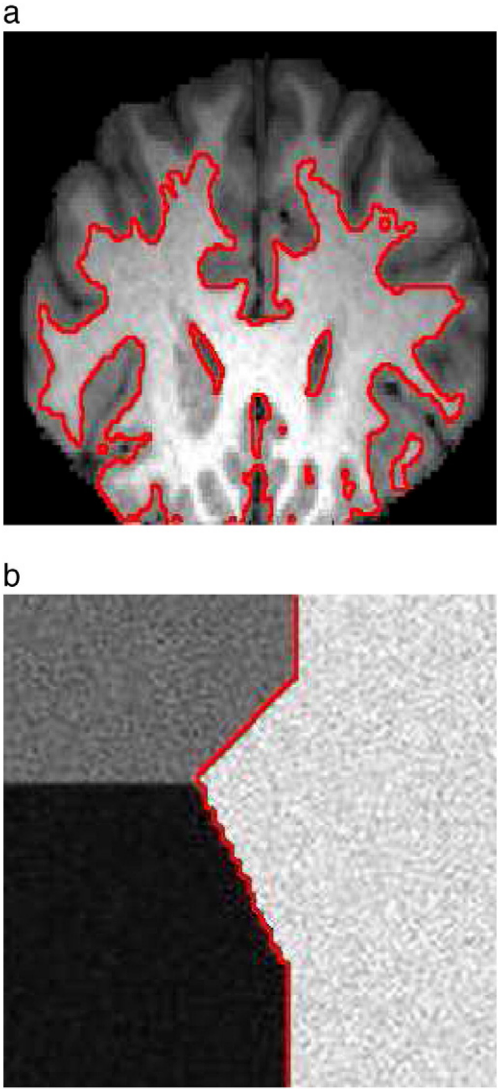

**Fig. 23.** Limitation of the model a segmentation of a brain MRI slice without bias field correction. (b) Segmentation of an image with 3 classes.

not restrict the topology of the graph to a cartesian grid and can be simply applied to arbitrary graphs that represent any kind of data.

### *7.3. Limitation of the model and future work*

Despite the huge impact of the piecewise constant Mumford–Shah functional. The assumptions of the model mainly, bimodal segmentation of piecewise constant intensity distribution, limit the application of our model to problems that satisfy these constraints. If these constraints are violated, the model has to be altered to accommodate the new constraints. To exemplify, Fig. 23 shows two examples where our model would fail to provide an appropriate segmentation. For example, in (a), we show a brain MRI slice without bias field correction. Since the image exhibit a very high level of inhomogeneity that extensively violate the piecewise constant intensity constraint, our approach does not perform well. For future, we plan to relax the homogeneity constraint of the model to be locally piecewise constant rather than globally piecewise constant. Another example is shown in (b) where the image contains more than two classes, ours is designed to provide a binary labeling that differentiate between two classes only. For future, we will investigate the extension of our algorithm to multi-phase image segmentation by introducing more than two labels to the model.

Last but not least, we are planning in incorporating prior information to the model which will have enabled us to separate objects that share the same intensity profile. This can be done if we include statistical prior information from atlas or training data.

## References

[1] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, International Journal of Computer Vision, IJCV 1 (4) (1987) 321–331.

[2] L.D. Cohen, On active contour models and balloons, Computer Vision, Graphics, and Image Processing 53 (2) (1991) 211–218.

[3] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, A. Yezzi, Conformal curvature flows: from phase transitions to active vision, Archive for Rational Mechanics and Analysis 134 (3) (1996) 275–301.

[4] R. Malladi, J.A. Sethian, B.C. Vemuri, Shape modeling with front propagation: a level set approach, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (2) (1995) 158–175.

[5] A. Amini, T. Weymouth, R. Jain, Using dynamic programming for solving variational problems in vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (9) (1990) 855–867, ISSN 0162-8828.

[6] A.J. Yezzi, S. Kichenassamy, A. Kumar, P.J. Olver, A. Tannenbaum, A geometric snake model for segmentation of medical imagery, IEEE Transaction on Medical Imaging 16 (2) (1997) 199–209.

[7] N. Paragios, R. Deriche, Geodesic active contours for supervised texture segmentation, IEEE Computer Vision and Pattern Recognition, CVPR, vol.2, 1999, pp. 422–427.

[8] R. Ronfard, Region based strategies for active contour models, International Journal of Computer Vision 13 (2) (1994) 229–251.

[9] D. Mumford, J. Shah, Optimal approximation by peicewise smooth functions and associated variational problems, Communications on Pure and Applied Mathematics 42 (4) (1989) 577–685.

[10] L.A. Vese, T.F. Chan, A multiphase level set framework for image segmentation using the Mumford and Shah model, International Journal of Computer Vision 50 (3) (2002) 271–293, ISSN 0920-5691.

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

[12] X. Bresson, P. Vandergheynst, J. Thiran, A variational model for object segmentation using boundary information and shape prior driven by the Mumford–Shah functional, International Journal of Computer Vision 28 (2) (2006) 145–162.

[13] T. Chan, W. Zhu, Level set based shape prior segmentation, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, vol.2, 2005, pp. 1164–1170.

[14] D. Cremers, N. Sochen, C. Schnšrr, Towards recognition-based variational segmentation using shape priors and dynamic labeling, Scale Space Methods in Computer Vision (2003) 388–400.

[15] K. Fundana, N. Overgaard, A. Heyden, Deformable Shape Priors in Chan–Vese Segmentation of Image Sequences, IEEE International Conference on Image Processing vol. 1, ISSN 1522-4880, 285–288, 2007.

[16] F. Huang, J. Su, Moment-based shape priors for geometric active contours, Proceedings of the 18th International Conference on Pattern Recognition (ICPR06), ISBN: 0-7695-2521-0, 2006, pp. 56–59.

[17] S. Shah, A. Abaza, A. Ross, H. Ammar, Automatic tooth segmentation using active contour without edges, Proc. of the Biometric Consortium Conference (BCC), 2006, pp. 1–6.

[18] N. Xu, R. Bansal, N. Ahuja, Object segmentation using graph cuts based active contours, IEEE conference on Computer Vision and Pattern Recognition, vol.2, 2003, pp. 46–53,, ISSN 1063-6919.

[19] J. Darbon, A note on the discrete binary mumford–shah model, MIRAGE, 2007, pp. 283–294.

[20] L. Grady, C. Alvino, The piecewise smooth Mumford–Shah functional on an arbitrary graph, IEEE Transaction on Image Processing 18 (11) (2009) 2547–2561.

[21] N. El-Zehiry, P. Sahoo, S. Xu, A. Almaghraby, Graph cut optimization for the Mumford Shah model, Proccedings of the Seventh IASTED International Conference on Visualization, Imaging, and Image Processing, VIIP, 2007, pp. 182–185.

[22] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, vol. 153, Springer-Verlag, 2003.

[23] C. Samson, L. Blanc-Fraud, G. Aubert, J. Zerubia, Multiphase evolution and variational image classification, Tech. Rep. 3662, INRIA. 1999.

[24] Y. Boykov, V. Kolmogorov, Computing geodesics and minimal surfaces via graph cuts, International Conference for Computer Vision, ICCV, vol.1, 2003, pp. 26–33.

[25] V. Kolmogorov, Y. Boykov, What metrics can be approximated by geo-cuts, or global optimization of length/area and flux, Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05), vol.1, 2005, pp. 564–571.

[26] V. Kolmogorov, R. Zabih, What energy functions can be minimizedvia graph Cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2) (2004) 147–159, ISSN 0162-8828.

[27] Y. Boykov, 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) (2004) 1124–1137.

[28] T.F. Chan, B. Yezrielev, L.A. Vese, Active contours without edges for vector-valued images, Journal of Visual Communication and Image Representation 11 (2000) 130–141.

[29] Y. Boykov, M.-P. Jolly, Interactive graph cuts for optimal boundary and region segmentation of objects in N–D images, ICCV, 2001, pp. 105–112.

[30] C. Rother, V. Kolmogorov, A. Blake, GrabCut: an interactive foreground extraction using iterated graph cuts, SIGGRAPH, 2004, pp. 309–314.

[31] T. Chan, S. Esedoglu, M. Nikolova, Algorithms for finding global minimizers of image segmentation and denoising models, SIAM Journal on Applied Mathematics 66 (5) (2006) 1632–1648.

[32] X. Bresson, S. Esedoglu, P. Vendergheynst, J.-P. Thiran, S. Osher, Fast global minimization of the active contour/snake model, Journal of Mathematical Imaging and Vision 28 (2) (2007) 151–167.

[33] T. Goldstein, X. Bresson, S. Osher, Geometric Applications of the Split Bregman Method: segmentation and surface reconstruction, To appear in Journal of Scientific Computing- Available at ftp://ftp.math.ucla.edu/pub/camreport/cam09-06.pdf. 2009.