Unsupervised Multiphase Segmentation: A Phase Balancing Model

Berta Sandberg, Sung Ha Kang, and Tony F. Chan

Abstract—**Variational models have been studied for image segmentation application since the Mumford–Shah functional was introduced in the late 1980s. In this paper, we focus on multiphase segmentation with a new regularization term that yields an unsupervised segmentation model. We propose a functional that automatically chooses a favorable number of phases as it segments the image. The primary driving force of the segmentation is the intensity fitting term while a phase scale measure complements the regularization term. We propose a fast, yet simple, brute-force numerical algorithm and present experimental results showing the robustness and stability of the proposed model.**

Index Terms—**Cheeger Set, image segmentation, multiphase, scale, variational model.**

I. Introduction

IMAGE segmentation separates the image into different regions to simplify the image and identify the objects easily. Image segmentation has been extensively studied by various approaches, such as mixture random-field models [10] and the Mumford–Shah variational image model [17]. New segmentation models like Monte-Carlo Markov chain model [27], the graph-cutting and spectral method [23], and the variational texture segmentation models [20], [21] incorporate more complexity. Since the publication of Mumford–Shah's image segmentation model, many extensions and properties to the variational approaches have been studied. The Chan–Vese model [6] is well known for a successful level set implementation and this work has been extended to multichannel [5], texture segmentation [20], logic model [19], and multiphase segmentation [7], [28].

In this paper, we focus on multiphase image segmentation, that is to identify more than two phases from a given image. Notable region-based multiphase segmentation models include works by Vese and Chan [28], Chung and Vese [7], Brox and Weickert [2], Tai and Chan [26], Lie *et al.* [16], Jung *et al.* [13], and Bae and Tai [1]. Vese and Chan [28] first introduced a generalization of a two-phase segmentation model [6] by using $\log_2 n$ number of level sets to identify n number of phases. A multilayer method was introduced by Chung and Vese [7] where more than one level of a level set is used to represent the discontinuity of the image and is modeled after island dynamics in epitaxial crystal growth. Minimization strategies have been proposed in [2] using the level set framework for improving region competition methods [29]. A piecewise single level set model for multiphase segmentation has also been proposed [16]. A multiphase Mumford–Shah model utilizing the graph-cut algorithm has been successfully demonstrated [1]. A relaxed model for multiphase segmentation using Γ-convergence analysis has been developed [13]. Other related studies have also been published [3], [9], [14], [15], [18].

The functional can be expressed in the framework of the Mumford–Shah model. Let Ω be a bounded Lipschitz image domain and $u_o : \Omega \longrightarrow \mathbb{R}_+ \cup \{0\}$ be a given image. Recall that the classical Mumford–Shah segmentation is to minimize

$$\mathcal{E}_{ms}[u,\Gamma|u_o] = \alpha \int_{\Omega\backslash\Gamma} |\nabla u|^2 dx + \beta\mathcal{H}^1(\Gamma) + \int_{\Omega}(u-u_o)^2 dx \quad (1)$$

where $\Gamma \in \Omega$ denotes the *edge set* of the image u, and $\mathcal{H}^1$ represents the 1-D Hausdorff measure. Multiphase segmentation identifies different phases by keying the intensity discontinuities [7], [13], [16], [26], [28]. To identify piecewise constant objects, the reduced Mumford–Shah functional can be written as minimizing

$$\mathcal{E}_{\text{cv}}[u,\Gamma|u_0] = \beta\mathcal{H}^1(\Gamma) + \sum_{i=1}^{N}\int_{\Omega_i}|u_0 - c_i|^2 \quad (2)$$

where Ω_i phases are the connected components of $\Omega\backslash\Gamma$ and c_i is the intensity average of u_0 in each Ω_i. With a successful level-set implementation, this reduced Mumford–Shah model is frequently referred to as the Chan–Vese (CV) model for two-phase segmentation [6].

One of the limitation of these multiphase segmentation methods is that a successful result is dependent on *a priori* knowledge of the number of image segments. For example, in [13] and [16], a reasonable number of phases are given *a priori* . In [28], the authors attempts to overcome this limitation by starting with many initial contours hoping irrelevant contours degenerate. In practice, this can lead to over-segmentation as illustrated in Fig. 1, that the algorithm incorrectly separates one object into several regions. In [2], the authors also noticed the difficulties of having arbitrary number of regions and proposed a region competition method in each scale for the functional in [29]. In addition, often multiphase segmentation of this type

Manuscript received January 31, 2008; revised August 11, 2009. First published September 15, 2009; current version published December 16, 2009. This work was supported in part by NSF:DMS-0908517, ONR:N00014-06-1-0345, and NSF:DMS-0610079. The associate editor coordinating the review of this manuscript and approving it for publication was Sharathchandra Pankanti.

B. Sandberg is with Adel Research, Inc., Los Angeles, CA USA, and also with the Department of Mathematics, University of California, Los Angeles 90095–1555 USA (e-mail: berta.sandberg@adelresearch.com).

S. H. Kang is with the School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: kang@math.gatech.edu).

T. F. Chan is with the Mathematics and Computer Science Departments, Hong Kong University of Science and Technology, Hong Kong, and also with the Department of Mathematics, University of California, Los Angeles 90095–1555 USA (e-mail: chan@math.ucla.edu).

Digital Object Identifier 10.1109/TIP.2009.2032310

1057-7149/$26.00 © 2009 IEEE

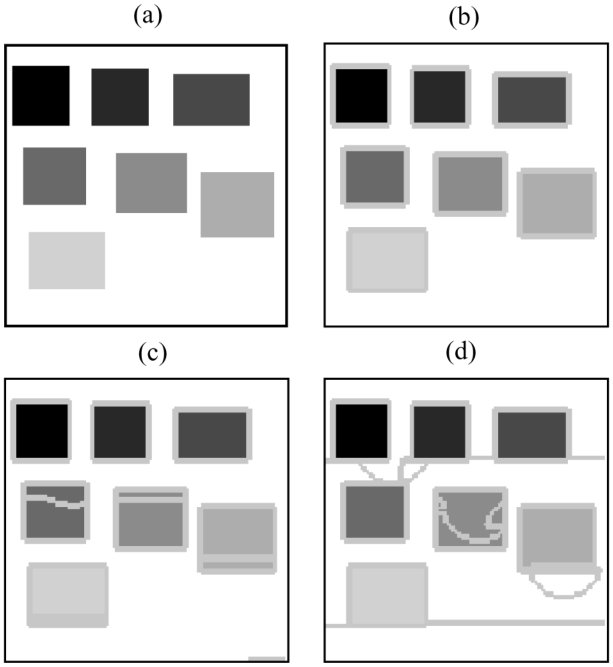

Fig. 1. (a) Original test image is a piecewise constant image with eight different constant intensities; (b), (c), and (d) are segmentation results using the Chan–Vese model [28] where (b) three level sets, (c) four level sets, and (d) five level sets are used, respectively. Boundaries of contours are superposed over the segmented results; (b) uses $2^3 = 8$ phases matching the 8 regions. When excessive contours are used; (c), (d), regions are incorrectly identified with multiple phases and in some cases spurious regions are identified by empty level sets.

(1) and (2) yield results that are highly correlated to the static initial value of the free parameters.

We propose a new functional that automatically chooses a favorable number of phases as it segments the image. In Section II, we discuss a new model for unsupervised segmentation which autonomously optimizes the phase scale without *a priori* tuning of free parameters. This model reliably converges to a favorable phase scale as it segments the image. The novel significance is that this new functional achieves unsupervised multiphase segmentation and can be implemented using a simple yet computationally efficient numerical algorithm. In Section III, we discuss the details of the numerical implementation notably disencumbered from computationally expensive Euler-Lagrange gradients. To demonstrate the robustness of the functional, the results of the numerical experiments are presented in Section IV.

II. Unsupervised Multiphase Segmentation Model

We represent the segmented image as a linear combination of distinct phases which are defined by characteristic functions χ_i. The $\sum \chi_i$ covers the entire image domain Ω and $\chi_i \cap \chi_j = \emptyset$. Each χ_i is associated with one intensity average value c_i, and it is possible that one phase may consist of many spacially separate regions. The average intensity value is computed in the canonical way: $c_i = \int_{\chi_i} u_o(x)dx / \int_{\chi_i} 1dx$. The final segmented result is denoted as

$$u = \sum_{i=1}^{K} c_i * \chi_i.$$

In the CV model (2), the two major components are composed of the intensity fitting and regularization terms. The segmentation is driven by favoring intensity similarities with minimal phase boundaries. In our functional we add two more objectives to achieve an unsupervised model that can automatically discover a favorable number of phases as it segments the image. The objectives are as follows.

1) [Phase] Favor image segments with greater area. We prefer to identify objects with significant sizes rather than regions which are extremely small.
2) [Balance] Equal importance of each phases. We assume there is no *a priori* information on which phase is more important, and we use equal weight for all phases. Outlier regions which are extremely small or extremely large compared to other regions is disfavored.

A. Phase

To identify objects with significant sizes, we consider the the scale term used by Strong *et al.* [25], in the context of total variation (TV) denoising. The authors defined the scale term to be the relation between the length and the area of objects, and denoted as $\text{scale} := (\text{area})/(\text{length})$. Here, the scale term is positively correlated to the object size, i.e., when the size of a region is large, the scale term is large.

To meet the first objective while minimizing a functional, we use an *inverse scale term*

$$\mathcal{S}_i := \frac{P(\chi_i)}{|\chi_i|} \tag{3}$$

where $P(\chi_i)$ denotes the perimeter of a phase χ_i and $|\chi_i|$ denotes the 2-D area of a phase χ_i. By minimizing this term, the segmentation favors phases with large areas. This is demonstrated in Fig. 2. In image (a), the most noticeable objects are the two big boxes on the right. Using S_i as a part of the regularization term, the segmentation would give wights to these two boxes. In addition, since the segmentation is driven by the intensity value, the similar intensity boxes will go the corresponding phase [see image (b) and image (c)]. Fig. 2(d) illustrates another example: the given image has only one definite object, the disk in the center, so this object and similar intensity areas are identified as one phase χ_1.

Remark 1: The inverse scale $\mathcal{S}_i$ is defined within each phase χ_i, and it is possible for a phase to have many disconnected regions. The total length of the edge in each phase χ_i is divided by the total area of the pixels regardless of the number of connected components. For example, several different objects with similar intensities will be contained in a common phase χ_i and contribute to $\mathcal{S}_i$ jointly.

Remark 2: This notation of $\mathcal{S}_i$ is related to the *Cheeger Set*, which is widely studied in the calculus of variation analysis. The objective is to find a nonempty set $A \subset \Omega$ of finite perimeter which minimizes the following:

$$\min_{A \subset \Omega} \frac{P(A)}{|A|}.$$

There are various studies on qualitative properties of Cheeger Sets, and [4] and [8] give good overviews and

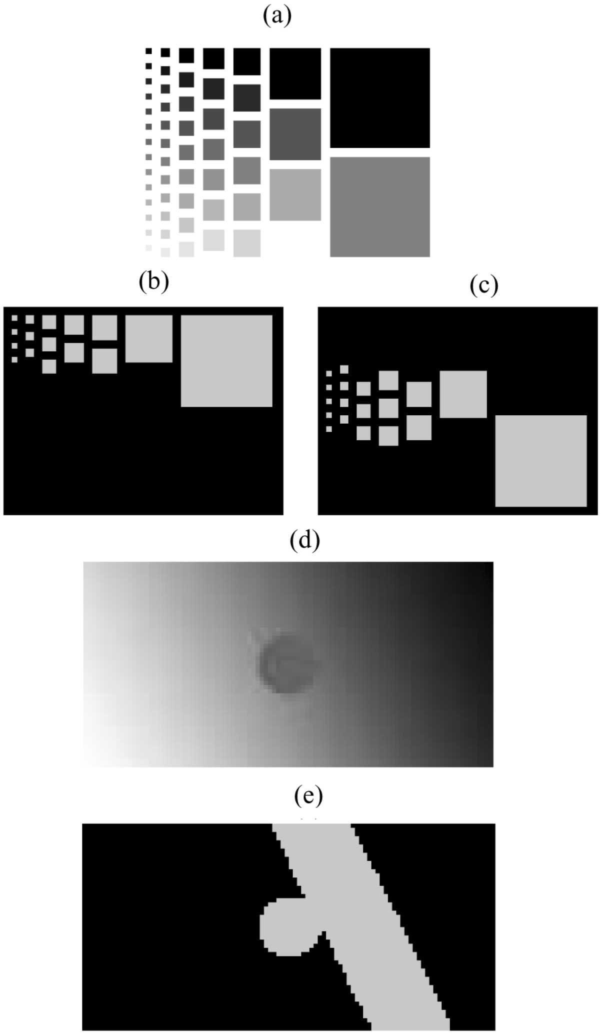

Fig. 2. Effect of the inverse scale term $\mathcal{S}_i$. (a) The original image; (b) and (c) show the phases with two biggest boxes on the right (one black and another gray). After two biggest objects are identified via $\mathcal{S}_i$, the similar intensity gray boxes are also included in the corresponding phase. (These two phases χ_1 and χ_2 are two among many other phases, full results showing all other phases are in Fig. 4). In the second row, from the original image (d), one disk in the center is identified as one phase together with all the similar intensity area. (The result showing all other phases are in Fig. 3).

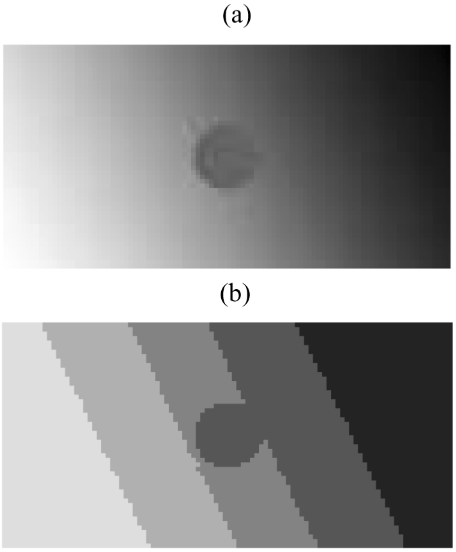

Fig. 3. Effect of the phase balancing term $\sum \mathcal{S}_i$. (a) Original image. (b) The segmented result $\sum c_i * \chi_i$ using the proposed model (5). The segmented phases are similar in scale.

references for related recent works on Cheeger sets. By defining a finite perimeter $\lambda := (P(G))/|G|$ and considering $\min_{A \subset \Omega} P(A) - \lambda|A|$, these studies of Cheeger sets are related to Total Variation minimization, $\min_{u \in BV(\Omega)} \int_\Omega |Du| + \lambda \int_\Omega |u(x) - u_o(x)|^2$, which is consistent with the scale notation studied in [25]. Note, Cheeger Set considers a single set minimization while we consider multiset problems.

Property 1: Consider a single connected region χ_i. If the perimeter is fixed, convex regions have lower $\mathcal{S}_i$ compared to concave regions. This is consistent with Cheeger Set theories, and by minimizing $\mathcal{S}_i$ the functional favors convex regions.

Property 2: Different shaped objects can have the same inverse scale value. For example, any regular (equilateral) convex polygon B which incircles a circle with radius r has $(P(B))/|B| = (2/r)$ which is identical to the inverse scale value of a circle with radius r.

B. Balance

In order to give balance among the phases, we do not assume any bias among different phases, and we consider the summation with all phases with equal weight

$$\sum_{i=1}^{K} \mathcal{S}_i = \sum_{i=1}^{K} \frac{P(\chi_i)}{|\chi_i|}.$$

For a given discrete bounded image with $|\Omega| < \infty$ and $\sum_{i=1}^{K} P(\chi_i) < \infty$, for a fixed K, if χ_i is nonempty for all $i = 1, \ldots, K$, the minimum of the summation is achieved when $\mathcal{S}_i$ are all identical for $\forall i = 1, \ldots, K$. Therefore, by minimizing this term, the objects of various sizes in the image will uniformly be distributed among all different phases and $\mathcal{S}_i$ value will be similar to each other. We refer to this term as *the phase balancing term*. Fig. 3 shows this effect of this unbiased phase selection, and uniform segmentation.

Proposition 1: Consider a piecewise constant image with m number of objects $(m < \infty)$, all of which has identical inverse scale value

$$\frac{P(B_j)}{|B_j|} := s_1, \quad \forall j = 1, \ldots, m$$

and $B_j \cap B_i = \emptyset$ for $i \neq j$ (nonoverlapping). Assume K is fixed, i.e., each χ_i are nonempty and the background is represented with χ_K. Then, any distribution of these objects B_j to different phases χ_i gives minimum of the phase balancing term

$$\sum_{i=1}^{K} \frac{P(\chi_i)}{|\chi_i|} = (K-1)s_1 + \frac{P(\chi_K)}{|\chi_K|}.$$

Proof: The inverse scale term $\mathcal{S}_i$ in (3) is define for each phase, not for each object. For example, if one phase χ_i has one circle with radius $r, \mathcal{S}_i = 2/r$, and another phase χ_j has two circles with radius $r, \mathcal{S}_j = (2\pi r + 2\pi r)/(\pi r^2 + \pi r^2) = 2/r$. Both phases have the same phase balancing value independent to the number of objects in each phases.

Therefore, if we let the length of an object to be $P(B_j) = a$, the area to be a/s_1 and m_i be the number of the objects in each phase χ_i, then, $\mathcal{S}_i = m_i a / m_i(a/s_1) = s_1$ independent to the number m_i. The phase balancing term will be $(K-1)s_1$ except for the background χ_K. ■

The above Proposition shows that if an image has only one kind of object, the number of objects in each phases can be quite different since the phase balancing value becomes independent to the number of objects in each phase. This is due to the way we compute the $\mathcal{S}_i$ (Remark 1), and can be somewhat counterintuitive to the balancing effect.

C. Proposed Model

Incorporating these additional objectives, we propose the following functional for unsupervised multiphase segmentation, *a phase balancing model*

$$E[K, \chi_i, c_i | u_o] = \hat{\mu}\left(\sum_{i=1}^{K} \mathcal{S}_i\right)\mathcal{H}^1(\Gamma) + \sum_{i=1}^{K}\int_{\chi_i} |u_o - c_i|^2 \quad (4)$$

where Γ is set of all the boundaries of χ_i for $i = 1, \ldots, K$, i.e., $\Gamma = \bigcup_{i=1}^{K}\{\partial\chi_i\}, \mathcal{H}^1$ represents the 1-D Hausdorff measure as in (1) and (2), and the average value c_is are defined as in (2), $c_i = \int_{\chi_i} u_o(x)\,dx / \int_{\chi_i} 1\,dx$. Notice that K, χ_is and c_is in $E[K, \chi_i, c_i | u_o]$ are unknown variables while only the original image u_o is given. Calculating the number of phases K while segmenting, significantly differentiates this model from the other multiphase models. Using $P(A)$ to represent the finite perimeter of the set A, the functional can be represented as

$$E[K, \chi, c_i | u_o] = \mu\left(\sum_{i=1}^{K}\frac{P(\chi_i)}{|\chi_i|}\right)\sum_{i=1}^{K}P(\chi_i) + \sum_{i=1}^{K}|u_0 - c_i|^2\chi_i. \quad (5)$$

Here, $\mu = \hat{\mu}/2$ from $\hat{\mu}$ in (4): by adding the perimeter of each phase, the total length of boundaries is doubled.

In this functional, μ is the only free parameter, and for an unsupervised segmentation we typically set $\mu = 1$. The first term of (5) is essentially unit-free, while the second term $\sum_{i=1}^{K}|u_0 - c_i|^2\chi_i$ corresponds to the area of the phase, therefore, this μ is a parameter representing the area of the segmentation. If μ is large, the segmentation favors fewer phases with larger areas, and the smaller value for μ prefers more phases with smaller area for a segmentation. The effects of the changes of μ are explored in Section IV.

To illustrate the full effect of this model, we consider the example in Fig. 2 again. After the two main features are identified, the other regions are well divided to give balance among the phases and follow the intensity differences. Fig. 4 illustrates the full result. The original image Fig. 2(a), has at least 15 different intensity values, however, the image is automatically (with $\mu = 1$) identified to 4 different phases. The segmentation is driven by the two biggest objects, then the rest are well distributed following the intensity similarities while giving the balance among the phases.

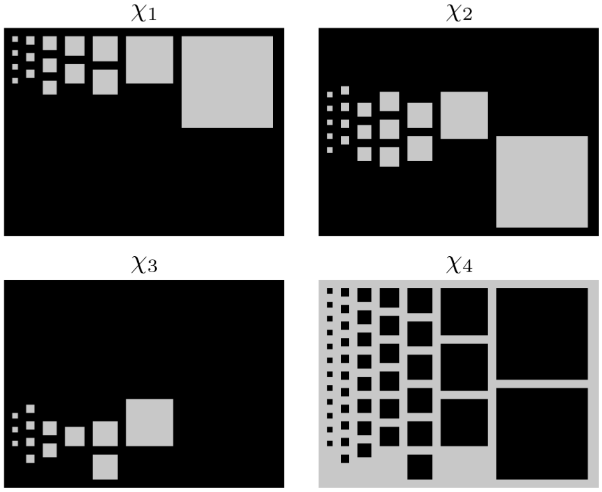

Fig. 4. Original image is Fig. 2(a). Each column of boxes are changing the intensity consistently from top to bottom, and the size of the boxes are increasing from left to right (length of the width is doubling each time). The proposed model automatically chooses four phases with $\mu = 1$, and it shows that the boxes are well distributed among the phases. Notice the strong similarity among the phases.

Remark 3: In the proposed model (5), the perimeter $P(\chi_i)$ appears twice, once in the phase balancing term and the total length term. According to the Property 2 in Section II-A, the smoothness of the boundary is independent from the minimum of $\mathcal{S}_i$, since the circle and a convex polygon can have the same inverse scale value. Therefore, the total length term $\mathcal{H}^1(\Gamma)$ is included for its smoothness property (and well-fitted boundaries).

Remark 4: As shown in Figs. 3 and 4, the segmentation is primarily driven by the intensity. The model favors larger continuous region, however, the smaller regions with similar intensity will be incorporated into the phase with the similar intensity. The small objects, such as noise or small stars will be included as a feature of the image rather than being completely denoised. Denoising occurs within a certain difference in intensity. If the intensity is close to one of the phases, it will be included in that phase (see Fig. 12).

Remark 5: There were several possible ways to incorporate the phase balancing term $\sum_{i=1}^{K}\mathcal{S}_i$ to propose a new model. Case 1, one can consider addition of the phase balancing term to CV model: the phase balancing term and the total length terms have different scales, thus creating a need for another free parameter. This inherits limitations of CV model with additional parameter, making the functional unstable. Case II, one can consider multiplying $\sum(1)/(\mathcal{S}_i)$ to the total length term, making two terms in the function both represent the area, or Case III, multiplying the phase balancing term $\sum_{i=1}^{K}\mathcal{S}_i$ to the fitting term, making both terms represents length of the regions. In these cases, Case II and Case III, in addition to being sensitive to the choice of the parameter, we observed that the number of phase K increases to minimize to energy. The total length term is never zero, while the fitting term can become zero by increasing the number of phases continuously, therefore, both cases typically result in a big number of phases. On the other hand, by multiplying the phase balancing term by the total length term, our proposed model maintain the global smoothness capability of the total length term while maximizing the area locally in the specific phase.

In the following section, we present a brute-force algorithm for a fast and easy computation of the proposed model (5).

III. Fast Algorithm for Multiphase Segmentation

We represent the segmented result as $u = \sum_{i=1}^{K} c_i * \chi_i$, where each phase is represented with a separate characteristic function χ_i, for $i = 1, \ldots, K$. A typical method for solving segmentation functionals is the level set method [7], [16], [19], [20], [26], [28], and recently one function is used to represent multiphases as in [13] and [16]. However, using characteristic function χ_i to represent each phase allows us to easily add new phases and does not introduce any bias on the transition boundary from one phase to another.

This unsupervised segmentation model (5) is nonlinear and non trivial to identify the Euler-Lagrange equation with three different types of unknowns: the number of phases K, χ_is and the average intensity c_is. Instead, we directly minimize the functional using the simple and elegant technique such as [11], [24]. We adapt the main idea of Song and Chan [24] to consider the change in the difference of the functional using a greedy algorithm to decide if one pixel belongs to the inside or the outside of the phase. These settings, [11], [24], are well equipped to handle fast computation for two phase computations. A fast algorithm for multiphase level set method for CV model was developed in [12]. The authors discuss the danger of using these algorithms for the length term by considering the topological derivative. In this paper, we chose to work with the discrete setting of the functional that the length change is explicitly computed by the four edge changes.

From the proposed model (5)

$$E[K, \chi, c_i | u_o] = \mu \left(\sum_{i=1}^{K} \frac{P(\chi_i)}{|\chi_i|} \right) \sum_{i=1}^{K} P(\chi_i) + \sum_{i=1}^{K} |u_0 - c_i|^2 \chi_i$$

we consider the difference in the functional and pick the phase according to the minimum value. For $(x, y) \in \Omega$, the change in energy when (x, y) moves from one phase l to another phase j is computed by

$$\Delta E_{lj} = \mu \Delta F + (u - c_j)^2 \frac{n_j}{n_j + 1} - (u - c_l)^2 \frac{n_l}{n_l - 1} \quad (6)$$

where $u = u(x, y)$ is the intensity value at the pixel (x, y), c_i is the average of each phase i, and n_i is the number of pixels in phase i, i.e., area $|\chi_i| = n_i$. This ΔF is the change of the first term in (5), the phase balancing term and the total length term multiplied, and other two terms are the change of the intensity fitting term which is used in [24].

This ΔF is

$$\begin{aligned} \Delta F &= \mathbb{S}_j T_j - \mathbb{S}_l T_l = \mathbb{S}_j (T_l + \Delta T) - \mathbb{S}_l T_l \\ &= (\mathbb{S}_j - \mathbb{S}_l) T_l + \mathbb{S}_j \Delta T \\ &:= T_l \Delta \mathbb{S} + \mathbb{S}_j \Delta T \end{aligned} \quad (7)$$

where $\mathbb{S}_l$ presents the phase balancing energy $(\sum \mathcal{S}_i)$ and T_l the total length energy $(\sum P(\chi_i))$ when (x, y) is in phase l, $\mathbb{S}_j$ and T_j are when (x, y) is in phase j. To compute the total length T_l, since each phase is represented by a characteristic function χ_i, we simply add all the edges in the phase to get the length

$$P(\chi_i) = \sum_{(x,y) \in \Omega} \{ |\chi_i(x+1, y) - \chi_i(x, y)| + |\chi_i(x, y+1) - \chi_i(x, y)| \}$$

then, by summing them, we have $T_l = \sum_{i=1}^{K} P(\chi_i)$ when (x, y) is in phase l. In (7), $\Delta \mathbb{S}$ is defined to be $\Delta \mathbb{S} := \mathbb{S}_j - \mathbb{S}_l$ and this needs to be computed separately using $\mathbb{S}_l = \sum (P(\chi_i)/(n_i))$ for when (x, y) is in phase l, and $\mathbb{S}_j$ for when (x, y) is in phase j.

The difference in the total length energy ΔT is the addition of the change of perimeter in phase j and the change in phase l, $\Delta T = \Delta P(\chi_j) + \Delta P(\chi_l)$. Since we are working in the discrete settings, we count the number of old edges compared to the new edges to compute the perimeter change. The edges are checked for the four neighboring points (N, S, E, W) of (x, y), and we refer to this set of four neighboring points as $\mathcal{N}$. The perimeter change in phase j, $\Delta P(\chi_j)$, can be computed by the neighboring points in $\mathcal{N}$ by the following:

$$\Delta P(\chi_j) = 4 - 2 \sum_{(a,b) \in \mathcal{N}} \chi_j(a, b).$$

This is due to the fact that there are only five choices for $\Delta P(\chi_j)$ when the value $\chi_j(x, y)$ changes from 0 to 1 (the pixel (x, y) is moving to phase j from phase l), which is completely determined by the values of the neighborhood. For example, when there were no edges $(\forall (a, b) \in \mathcal{N}, \chi_j(a, b) = 0)$, by changing this pixel from 0 to 1, it creates four new edges, i.e., $\Delta P(\chi_j) = 4$. If there is one edge, $\sum_{(a,b) \in \mathcal{N}} \chi_j(a, b) = 1$, by flipping $\chi_j(x, y) = 0 \to 1$, it makes three edges, i.e., $\Delta P(\chi_j) = 2$. If there were two edges, the change from 0 to 1 creates no additional number of edges, $\Delta P(\chi_j) = 0$, but if there were three edges, $\sum_{(a,b) \in \mathcal{N}} \chi_j(a, b) = 3$, the change will only give one edge, i.e., the change becomes $\Delta P(\chi_j) = -2$. Finally, if there were four edges, it will remove those four edges, $\Delta P(\chi_j) = -4$. This can be similarly applied to the change in the perimeter of phase l as $\Delta P(\chi_l) = -4 + 2 \sum_{(i,j) \in \mathcal{N}} \chi_l(i, j)$, where the negative sign is due to the change from 1 to 0 (the pixel (x, y) is moving from phase l to phase j). Putting this together, the difference in the total length becomes

$$\Delta T = -2 \left(\sum_{(i,j) \in \mathcal{N}} \chi_j(i, j) - \sum_{(i,j) \in \mathcal{N}} \chi_l(i, j) \right). \quad (9)$$

The difference ΔE_{lj} in (6) can be computed by gathering all these terms

$$\begin{aligned} \Delta E_{lj} &= \mu (T_l \Delta \mathbb{S} + \mathbb{S}_j \Delta T) \\ &\quad + (u - c_j)^2 \frac{n_j}{n_j + 1} - (u - c_l)^2 \frac{n_l}{n_l - 1}. \end{aligned} \quad (10)$$

This is the calculation of the explicit difference of the energy when the pixel changes from one phase l to another phase j, which is used in the algorithm, Table I (8). If $\Delta E_{lj} > 0$, the

TABLE I
PIXELWISE BRUTE-FORCE ALGORITHM

```
Algorithm
• Set an initial phase: $|\chi_1| = |\Omega|$ with $k_o = 1$, where $k_o$ is the number of phase.
• Iterate
    1) At each pixel $(x, y) \in \Omega$ which belongs to phase $l$ ($\chi_l(x, y) = 1$ and $\chi_i(x, y) = 0$ for $\forall i \neq l$), compute $value$ using $\Delta E_{lj}$ in (10),
       $value = \min_j \{\Delta E_{lj} | j \neq l, j = 1, \ldots, k+1\}$,   (8)
       and let $h = arg\min_j \{\Delta E_{lj} | j \neq l, j = 1, \ldots, k+1\}$. Here $k+1$ refers to the new empty phase.
       Then,
           if $value < 0$,   set $\chi_h(x, y) = 1$ and $\chi_l(x, y) = 0$.
           if $value > 0$,   do nothing
    2) Update $k = h$, calculate $n_i = |\chi_i|$ and $c_i$ for each phase $i = 1, \ldots, k$.
```

pixel will not change to phase j since that will increase the energy. While, if this value ΔE_{lj} is negative, it is better to move (x, y) to phase j, then leave it in phase l.

It is important to notice that the number of phases are initially set to be $K = 1$ for all the experiments. In Table I, the algorithm compares the change ΔE_{lj} among all the existing different phases $j = 1, \ldots, k$ and, if necessary, it creates a new phase to minimize the energy. This new phase is represented as phase $k+1$ and the difference in the energy (8) is calculated using $n_{k+1} = 0$.

The complexity calculation for this algorithm is straightforward. Let m be the total number of pixels in the image. At the very first pixel, there are two choices: the current phase χ_1 and a new phase χ_2. As the algorithm sweeps through the domain, at each pixel it has $k+1$ choice of phases: k is the current number of phases and one additional choice for a new phase. If the number of phases is fixed as r, the complexity is $\mathcal{O}(rm)$ and if the method allows to increase the number of phases with each iteration, the complexity becomes $O(m + 2m + \cdots + rm)$ where r is the maximum number of regions attempted. The second type of method ends up being $O(r^2 m)$. For the proposed algorithm, the number of phases is typically identified after a single sweep of the image, the complexity calculation becomes $\mathcal{O}(km) = \mathcal{O}(m)$ for upto k number of phases. This is similar to the fast algorithm method [24] which gives $O(m)$ complexity.

The proposed method allows for a real-time processing of a large image set, since there is no need to preprocess the data to identify the number of regions. The run time on a single core Intel processor computer for the 100 × 100 image is 100 frames/second using C++. Typically, the algorithm converges in less than six iterations.

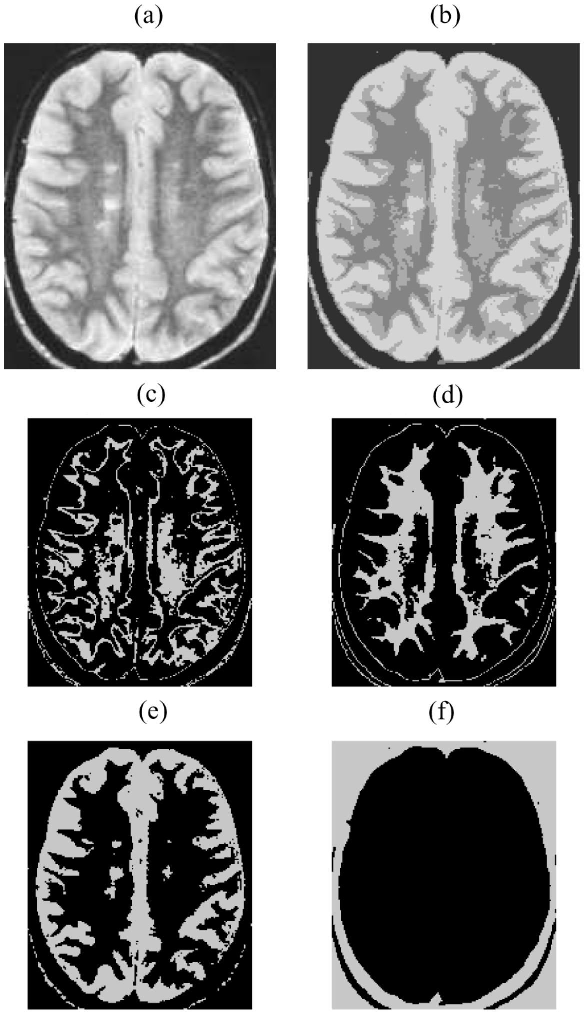

Fig. 5. MRI brain image is automatically segmented into four regions. (a) Original, (b) result, (c) phase 1, (d) phase 2, (e) phase 3, (f) phase 4. The model segments the different parts of the brain accurately.

IV. NUMERICAL EXPERIMENTS

All of the experiments are initialized with $|\chi_1| = |\Omega|$, i.e., $K = 1$, and $\mu = 1$, unless otherwise stated. The following experiments are done with left to right, and down sweeping order. These initial conditions eliminate some of the difficulties associated with multiphase segmentation.

Fig. 5 shows an experiment on a real image, a MRI brain scan. The model automatically identifies four phases, while segmenting the image. This segmentation is similar to standard segmentation algorithms which requires the number of phases *a priori*. Fig. 6 shows that the model segments the image with sharp corners and clear edges. This model automatically identifies five phases: white, gray, black (blue), green and yellow.

This method is extended to color images in Fig. 6. As in [5], we added all the channels for this experiment, using

$$\mu \left(\sum_{i=1}^{K} \frac{P(\chi_i)}{|\chi_i|} \right) \sum_{i=1}^{K} P(\chi_i) + \sum_{i=1}^{K} \int_{\chi_i} \sum_{l=1}^{3} |u_{0,l} - c_{i,l}|^2 dx \tag{11}$$

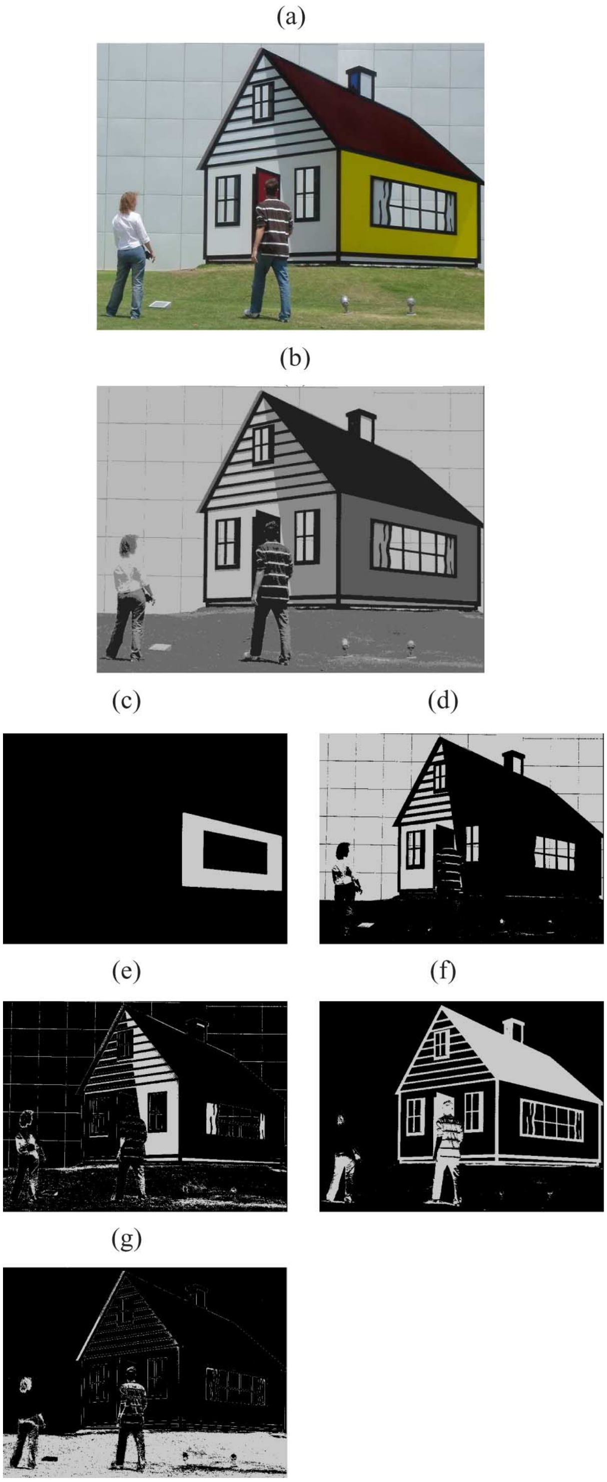

Fig. 6. Color image example. (a) Original Image, (b) segmentation, (c) phase 1, (d) phase 2, (e) phase 3, (f) phase 4, (g) phase 5. The model automatically segments the original image to five phases. Each segments refers to χ_1 (yellow), χ_2 (white), χ_3 (gray), χ_4 (black/blue), and χ_5 (green). The model is extended to a vector model using the fitting term as in [5], and for the image with sharp corners and edges, this model successfully finds shape boundary information.

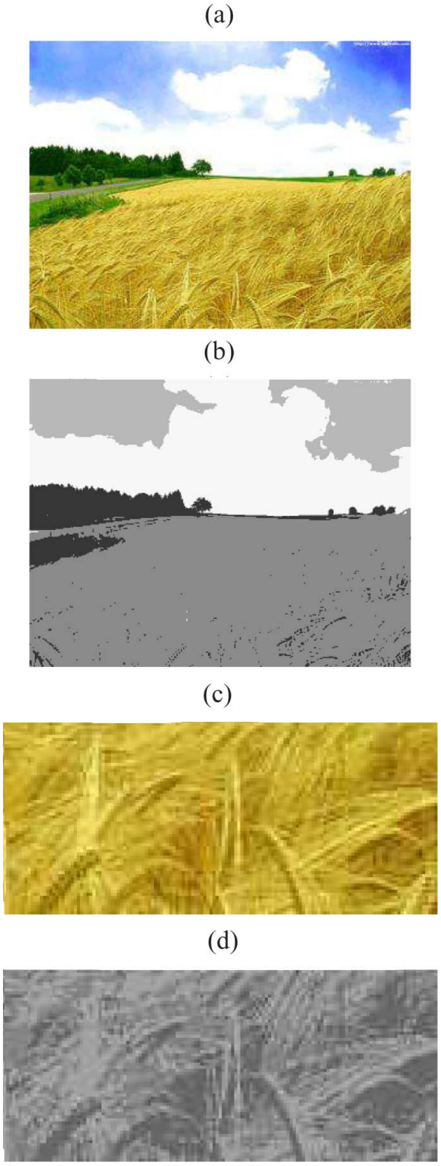

Fig. 7. Focus of the image: (a) Original Image, (b) segmentation of image (a) to four phases, (c) zoom into original image (a) field region, (d) segmentation of image (c) to three phases. The original image (a) is automatically segmented to four phases in image (b). When zoom into the one phase, the field region (yellow), image (c) is automatically further segmented to three phases in image (d). This shows that the proposed model adjusts to the scale of the given image.

here index l refers to each RGB (red, green and blue) channels. The proposed model automatically calculates the number of phases depending on the image. In Fig. 7, the original image (a) is automatically segmented to 4 phases (blue, white, yellow and green). And when zoom into the one phase the field region (yellow), image (c), is automatically further segmented to three phases. This shows that the proposed model (5) adjusts to the scale of the given image. In the original image, the scale of the wheat is insignificant and became a part of field. When zoomed into the wheat, the scale of the kernels increases relative to the image.

A. *Application to Image Quantization*

The proposed model can be applied to image quantization. Image quantization is used when high-resolution images are displayed on low-resolution or low-bit devices, such as in the calculators or in cell phones. If the low-bit device is a q-bit device, the range of the image can only take few discrete values from 0

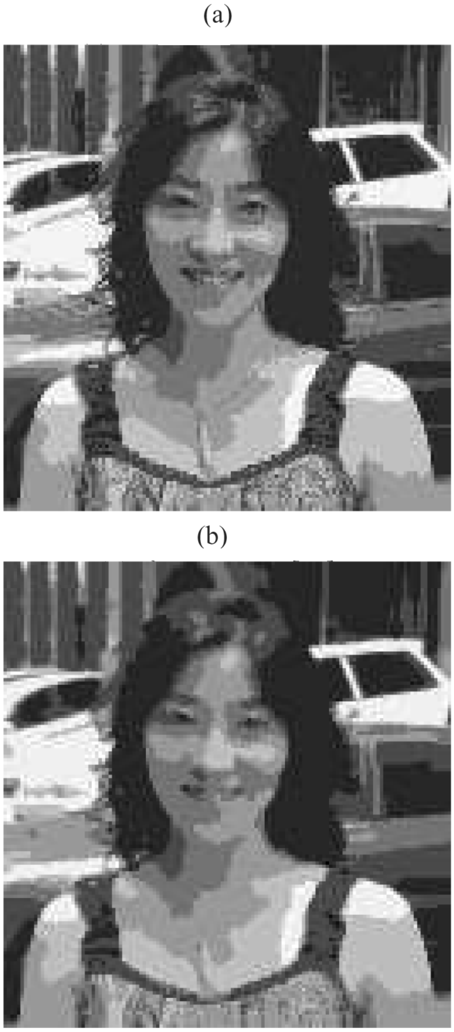

Fig. 8. Proposed model can be applied to image quantization. (a) The proposed method. (b) Quantum TV [22]. This model automatically segments the image to six phases. Compared to Quantum TV [22], this model keeps more details and features of the original image, such as the necklaces, details of the face and texture of the clothing.

to $2^q - 1$. With vast developments in cell phones and hand-held devices, good quality and fast quantization became more important. Fig. 8 shows such an application. Using this model, the image is automatically segmented to five phases. In [22], the authors computed the segments by minimizing the TV functional which is constrained on the quantum set Q. Here, the cardinality K is given

$$\min_{u \in BV^Q} E[u] = \min_{u \in BV^Q} \int_\Omega |Du| + \frac{\lambda}{2}\int_\Omega (f-u)^2 dx$$

where $BV^Q = BV(\Omega; Q) = \{u \in BV(\Omega) : u(x) \in Q,\ a.e.\ x \in \Omega\}$. With the given cardinality $K = 6$, the girl image is quantized with method [22] [Fig. 8(c)] and the optimized quanta set is found to be $Q = \{0.1314, 0.2860, 0.4514, 0.6009, 0.7484, 0.9417\}$. In our experiment, Fig. 8(a) is automatically segmented to six phases in image (b), and the c_i values are found to be $\{0.1468, 0.2915, 0.4628, 0.6121, 0.7554, 0.9508\}$ these values are within 2% difference. This example shows an application of the unsupervised segmentation model to image quantization.

B. Number of Phase K: Automatic Stopping

Notice from the algorithm, Table I, that initial condition is always set as $K = 1$ with $\chi_1 = \chi_\Omega$, and as the algorithm sweeps through the domain, the number of phases increases. It is important to understand when the algorithm stops adding new phase, to automatically give a reasonable number of phases K.

In the algorithm (8), a new phase is created only if it gives the minimum of ΔE_{lj} among all possible choices of i, j, and that minimum is a negative value. Therefore, we look into the details of when the energy becomes negative for possibly creating a new phase, i.e., $\Delta E_{lj} < 0$, for $j = k+1$ and $n_j = 0$. For this new phase, the intensity difference is zero, $(u - c_j)^2 = 0$, and the energy (10) is negative when

$$\mu(T_l \Delta\mathbb{S} + \mathbb{S}_j \Delta T)\left(1 - \frac{1}{n_l}\right) < (u - c_l)^2 \qquad (12)$$

for at least one $l \leq k$. This left-hand-side value, $\text{test} := \mu(\Delta F)(1 - (1/n_l))$ gives the lower bound on how big the intensity difference (between the current pixel $u(x,y)$ and each phase intensity average c_l) should be to create a new phase $j = k+1$. First of all, as n_l increases (as the area of existing phase gets bigger), the *test* increases, so it requires the intensity to be more different from averages c_l to create a new phase by itself. Second, the *test* value is depended on $\Delta\mathbb{S}$ multiplied by T_l, so, when the segmentation is already complicated with a large total length T_l, it becomes more difficult to add a new phase. Thirdly, ΔT is multiplied by $\mathbb{S}_j$, and since the new phase j only have one pixel, this $\mathbb{S}_j$ is quite big. These three terms make sure that a new phase is created only if there is a significantly large variance of the intensity between $u(x,y)$ and c_l. After this test (12) is satisfied, the energy ΔE_{lj} still needs to be the minimum among all other possible phases that have already been created.

To summarize, as the size of the phases increases, it becomes more difficult to add new phase. As soon as all the phases reach certain sizes, the algorithm's sensitivity to the intensity fitting term decreases, and pixels are added to already existing phases. This is independent to the location of the regions, since the area and perimeter is computed over each phase without considering the local connected components separately. The inequality (12) demonstrates the bound on adding new phases, which allows the automatic selection of K and unsupervised segmentation.

As a comparison, we considered the reduced piecewise constant Mumford–Shah, Chan–Vese [6] model as in [7]. We designed a brute-force algorithm for multiphase CV model similar to (10)

$$\Delta\mathcal{E}_{\text{cv}} = \beta\Delta T + (u - c_j)^2 \frac{n_j}{n_j + 1} - (u - c_l)^2 \frac{n_l}{n_l - 1}. \qquad (13)$$

We find that this algorithm also works very well if the original image is a piecewise constant function. However, for nonpiecewise constant images, these type of brute-force algorithms continuously add new phases, or the algorithm becomes very sensitive to the choice of parameter β. The reason for this problem

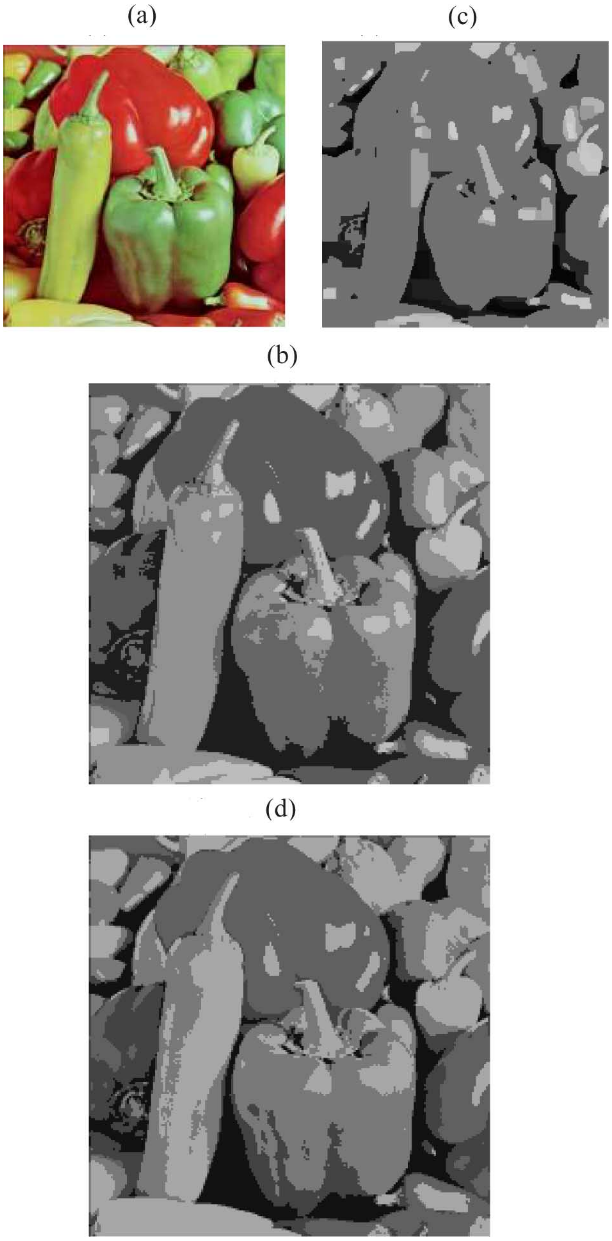

Fig. 9. (a) Original image. (b) The result of the proposed unsupervised model (6). (c) A result using CV model with a brute-force algorithm (13) with a big β. (d) A result using CV model (13) for a given fixed number of phase $K = 5$. When K is not given for CV model in image (c), the algorithm continuous to add new phases even with a big value of β. This image (c) has about 128 phases, many of which are very small regions. Comparing image (b) and image (d), the proposed model gives comparable result to CV model with an automatic selection of K.

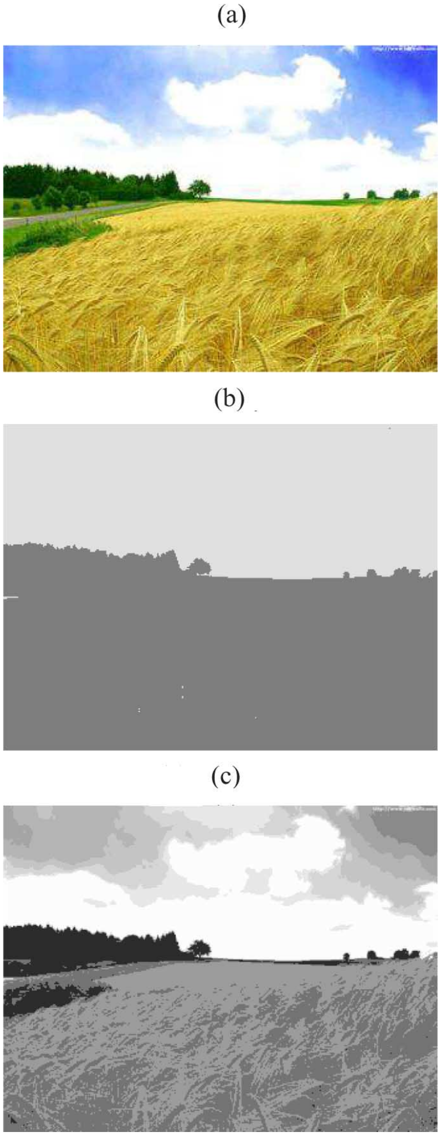

Fig. 10. (a) Original Image, (b) $\mu = 10$, (c) $\mu = 0.1$. Bigger values of μ give the bigger regions for segmentation. When $\mu = 10$, the model gives only two phases separating the sky from the earth (white and blue, and yellow and green regions are merged). While by using $\mu = 0.1$, the model enforces the intensity fitting term, and it gives ten phases with more detailed separations of the image. Fig. 7(b) have four phases with $\mu = 1$.

can be observed from a similar analysis as (12). The energy is negative, when

$$\beta \Delta T \left(1 - \frac{1}{n_l}\right) < (u - c_l)^2.$$

It is clear this algorithm will be sensitive to the choice of β. This test is only depended on the change in total length, which is typically very small, therefore, unless β is large, this algorithm will continue to add new phases. In (2), this effect corresponds to the intensity fitting term going to zero as more and more phases are added.

Fig. 9 illustrates this comparison. Given the original image (a), the proposed model segments the image with five phases automatically in (b), while as in image (c), the CV model using a brute-force algorithm (13) will continue to increase the number of phases. Even with a big value of β which gave a smooth result, this image (c) has about 128 number of phases with many small regions. When the phase number is given for CV model [image (d)], in this case $K = 5$, the brute-force algorithm (13) becomes less sensitive to the choice of β and gives a good segmentation result. Comparing image (b) and image (d), we observe that the proposed model gives comparable result to CV model with an automatic selection of K.

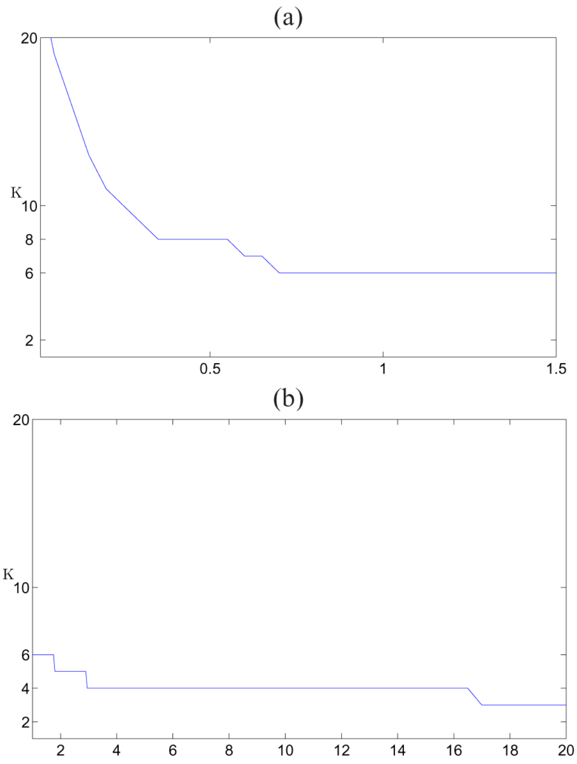

Fig. 11. Plots of K verses μ for the girl image in Fig. 8. The range of μ are (a) $0.01 < \mu < 1.5$ and (b) $1 < \mu < 20$. Notice that each piecewise intervals are quite large showing this model is not very sensitive to the choice of μ.

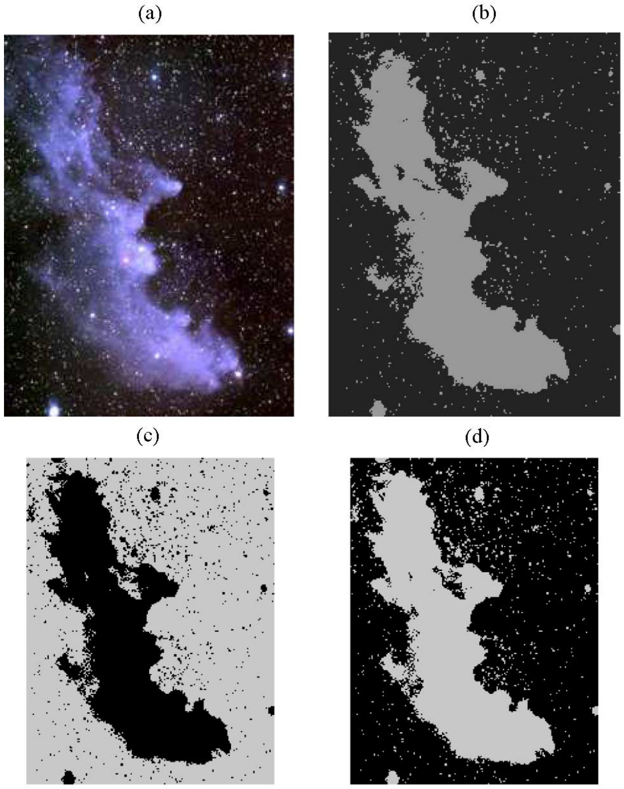

Fig. 12. Cluster segmentation: (a) Original image. (b) Segmentation result. (c) Phase 1. (d) Phase 2. A color image of a nebula is segmented into two phases using $\mu = 10$. Notice the details of the boundaries are kept while other small stars within the cluster is denoised and identified as the part of the main cluster.

C. Effects of Different μ

Up to now, all the experiments were using $\mu = 1$, which makes the proposed model (5) essentially parameter free. When $\mu \neq 1$ is used, different results can be achieved. This is illustrated in Fig. 10: compared to the Fig. 7(b) when $\mu = 1$, Fig. 10(b) with $\mu = 10$ returns only two phases separating the sky from the earth. On the other hand, with a smaller $\mu = 0.1$ in Fig. 10(c), the proposed model gives more detailed separations and identifies ten different phases.

Fig. 11 shows the changes in the phase number K verses the changes in μ. This experiment is for the image of the girl in Fig. 8. Both graphs are a plot of the number of phases K verses the μ values: (a) $0.01 < \mu < 1.5$ and (b) $1 < \mu < 20$. As predicted in Fig. 10 [also related to the inequality (12)], this is a decreasing function, and to have an integer value K, the graph becomes a step function. Notice, that each piecewise intervals are quite large showing this model is not sensitive to a careful choice of μ.

By using different values of μ, the proposed model can also handle a cluster segmentation. Fig. 12 shows an example of segmentation of a nebula: this is a color image and vector-valued model (11) is used. Using $\mu = 10$, the image is segmented into two regions, while $\mu = 1$ gives 6 different phases. Notice the region inside the nebula is smooth, while the small disconnected stars are not smoothed into the background but became a part of the nebula phase. This is due to the way the phase length and area are computed as mentioned in Section II-C, Remark 4. Noticed the bright starts within the nebula is included in the phase without being identified separately, showing this model have denoising effect while keeping details of the boundary.

D. Histogram Comparison

Without an unsupervised model, one way to give a phase number *a priori* is to consider the image's histogram. We compare the histogram of original image and the histogram of the segmented image to see the effect of the proposed model. In Fig. 13, the blue dotted line is the histogram of the original image and the red solid line is the histogram of the segmented result. Two plots are for the brain MRI, Fig. 5 and the image of the girl, Fig. 8. Looking at the histogram (b) of the girl Fig. 8, there is ambiguity in how many phases are needed for the segmentation, nevertheless, the proposed model picks out a stable number of phases, which allows for a quality segmentation of the image.

This example illustrates the difference between the proposed model and the typical k-mean or GMM methods. The image segmentation and data mining are two different applications and the minimization functional and the objective is somewhat different. Most of these methods, the number K is given *a priori* , then after iterations they correct the locations, or many preprocessing and learning process is added before these methods are used. However, in the proposed model, the number of phases K is calculated from the minimization of the proposed functional. This proposed functional has a fitting term and a regularization

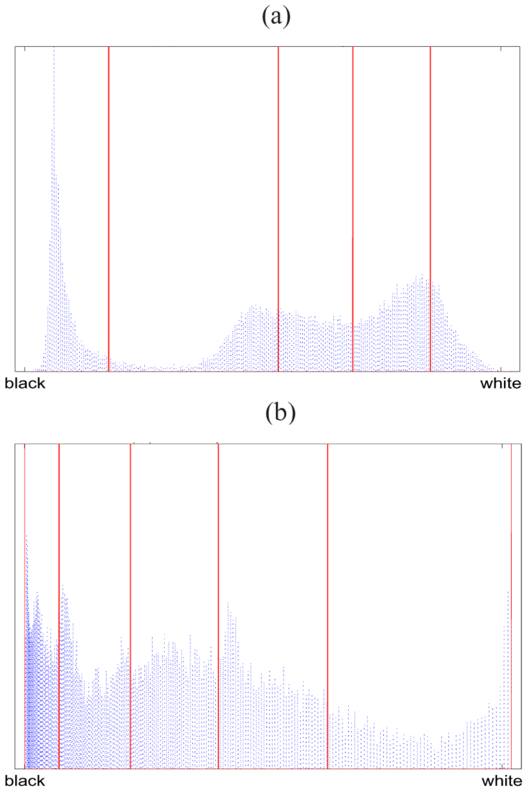

Fig. 13. Histogram comparison: (a) Brain MRI, Fig. 5, (b) the girl image, Fig. 8.The blue dotted lines are the histogram of the original image, and the red solid red lines are the histogram of the segmented result (i.e., c_i values). The range of the y-axis is artificially shortened to show the details, i.e., the red lines have a bigger value than represented in these graphs. Looking at the histogram of the original images, it is unclear how many number of phases are needed for the segmentation. Image (a) and (b) show that the proposed model automatically gives reasonable number of phases as well as the locations of c_i.

term with the newly added phase balancing term, which allows the automatic segmentation.

V. Conclusion

We propose a new unsupervised multiphase segmentation model that balances the phases and automatically gives a stable number of phases. Since the new method does not require knowledge of the number of phases before it segments, it gives a true unsupervised segmentation.

We achieved automatic segmentation which gives a reasonable number of phases K and finds each phase. This proposed method has interesting properties and many different extensions and applications are possible. To find the minimum of this nonlinear functional, we used a brute-force algorithm for a fast and accurate computation. The algorithm gives an insight on why this model has automatic stopping criteria for choosing the number of phases K. We experimented with synthetic and real images, considered applications to image quantization, extension to color images, and cluster segmentation.

This work marks a beginning of research in this area. By identifying the number of phases for a segmentation, many new and interesting applications are possible. This includes combining k-means with this clustering method for data mining applications and incorporating logic frameworks [19] in identifying key objects in the image. This furthers the pursuit of a robust and flexible computer vision system.

References

[1] E. Bae and X.-C. Tai, Graph Cuts for the Multiphase Mumford–Shah Model Using Piecewise Constant Level Set Methods, UCLA CAM Rep. 08–36, 2008.

[2] T. Brox and J. Weickert, “Level set based image segmentation with multiple regions,” in *Pattern Recognition*. Berlin/Heidelberg: Springer, 2004, vol. 3175, Lecture Notes in Computer Science, pp. 415–423.

[3] T. Brox and J. Weickert, “A TV flow based local scale measure for texture discrimination,” in *Computer Vision*. Berlin/Heidelberg: Springer, 2004, vol. 3022, Lecture Notes in Computer Science, pp. 578–590, ECCV.

[4] V. Caselles, A. Chambolle, and M. Navaga, “Uniqueness of the cheeger set of a convex body,” *Pacific J. Math.*, vol. 232, no. 1, pp. 77–90, 2007.

[5] T. Chan, B. Sandberg, and L. Vese, “Active contours without edges for vector-valued images,” *J. Vis. Commun. Image Represent.*, vol. 11, pp. 130–141, 1999.

[6] T. Chan and L. Vese, “Active contours without edges,” *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 266–277, Feb. 2000.

[7] J. Chung and L. Vese, Image Segmentatin Using a Multilayer Level-Set Approach, UCLA CAM Rep. 03–53, 2001.

[8] A. Figalli, F. Maggi, and A. Pratelli, “A note on cheeger sets,” *Proc. Amer. Math. Soc.*, vol. 137, pp. 2057–2062, 2009.

[9] S. Gao and T. D. Bui, “Image segmentation and selective smoothing by using Mumford–Shah model,” *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1537–1549, 2005.

[10] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, pp. 721–741, 1984.

[11] F. Gibou and R. Fedkiw, “Fast hybrid k-means level set algorithm for segmentation,” presented at the 4th Annual Hawaii Int. Conf. Stat. Math., 2002.

[12] L. He and S. Osher, “Solving the Chan–Vese model by a multiphase level set algorithm based on the topological derivative,” *Scale Space Var. Meth. Comput. Vis.*, vol. 4485/2008, pp. 777–788, 2007.

[13] Y. M. Jung, S. H. Kang, and J. Shen, “Multiphase image segmentation via Modica-Mortola phase transition,” *SIAM Appl. Math.*, vol. 67, pp. 1213–1232, 2007.

[14] H. Li and X.-C. Tai, “Piecewise constant level set methods for multiphase motion,” *Int. J. Numer. Anal. Model.*, vol. 4, no. 2, pp. 291–305, 2007.

[15] J. Lie, M. Lysaker, and X.-C. Tai, “A binary level set model and some applications to Mumford–Shah image segmentation,” *IEEE Trans. Image Procss.*, vol. 15, no. 5, pp. 1171–1181, May 2006.

[16] J. Lie, M. Lysaker, and X.-C. Tai, “A variant of the level set method and applications to image segmentation,” *AMS Math. Comput.*, vol. 75, pp. 1155–1174, 2006.

[17] D. Mumford and J. Shah, “Optimal approximation by piecewise-smooth functions and associated variational problems,” *Commun. Pure Appl. Math.*, vol. 42, pp. 577–685, 1989.

[18] Y. Pan, J. D. Birdwell, and S. M. Djouadi, “Bottom-up hierarchical image segmentation using region competition and the Mumford–Shah functional,” in *Proc. 18th Int. Conf. Pattern Recognition*, Washington, DC, 2006, pp. 117–121, IEEE Computer Society.

[19] B. Sandberg and T. Chan, “Logic operators for active contours on multi-channel images,” *J. Vis. Commun. Image Represent.*, vol. 15, pp. 333–358, 2005.

[20] B. Sandberg, T. Chan, and L. Vese, A Level-Set and Gabor-Based Active Contour Algorithm for Segmenting Textured Images, UCLA CAM Rep. 02-39, 2002.

[21] J. Shen, “Piecewise $H^{-1} + H^0 + H^1$ images and the Mumford–Shah-Sobolev model for segmented image decomposition,” *AMRX Appl. Math. Res. Exp.*, vol. 4, pp. 143–167, 2005.

[22] J. Shen and S. H. Kang, “Quantum TV and applications in image processing,” *Inv. Probl. Imag.*, vol. 1, no. 3, pp. 557–575, 2007.

[23] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[24] B. Song and T. Chan, A Fast Algorithm for Level Set Based Optimization, UCLA CAM Rep. 02-68, 2002.

[25] D. Strong, J.-F. Aujol, and T. Chan, "Scale recognition, regularization parameter selection, and Meyer's G norm in total variation regularization," *SIAM J. Multiscale Model. Simul.*, vol. 5, no. 1, pp. 273–303, 2006.

[26] X.-C. Tai and T. Chan, "A survey on multiple level set methods with applications for identifying piecewise constant functions," *Int. J. Numer. Anal. Model.*, vol. 1, no. 1, pp. 25–48, 2004.

[27] Z. W. Tu and S. C. Zhu, "Image segmentation by data-driven Markov chain Monte Carlo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 657–673, May 2002.

[28] L. Vese and T. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *Int. J. Comput. Vis.*, vol. 50, no. 3, pp. 271–293, 2002.

[29] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and bayes/mdl for multi-band image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, pp. 884–900, 1996.

Berta Sandberg received the B.S. degree in applied math and engineering and applied science from CalTech, Pasadena, CA, and the Ph.D. degree in mathematics from University of California, Los Angeles (UCLA), in 2002.

She has been a researcher at UCLA and is currently with Adel Research, Inc. Her research interests are in developing real-time image processing techniques for various applications in medical and defense related fields.

Sung Ha Kang received the Ph.D. degree in mathematics from the University of California, Los Angeles, in 2002.

She was with the Department of Mathematics, University of Kentucky, from 2002–2008. She has been an Assistant Professor at the School of Mathematics, Georgia Institute of Technology, Atlanta, since 2008. Her research interests are in mathematical image processing such as denoising, inpainting, color images, video and texture analysis, as well as numerical schemes and methods for scientific computing.

Tony F. Chan received the B.S. and M.S. degrees in engineering from CalTech, Pasadena, CA, and the Ph.D. degree in computer science from Stanford University, Stanford, CA.

He was with CalTech (applied math) as a Research Fellow and taught computer science at Yale University, New Haven, CT, before joining the faculty at the University of California, Los Angeles (UCLA), in 1986 as Professor of mathematics. He served as a Chair of the Department of Mathematics, UCLA, from 1997–2000, and as Director of the Institute for Pure and Applied Mathematics (IPAM) from 2000 to 2001. From July 2001 to June 2006, he served as Dean of Physical Sciences, UCLA, and became an Assistant Director, Directorate for Mathematics and Physical Sciences at National Science Foundation (2006–2009). He has been the President at the Hong Kong University of Science and Technology since October 2009, holding positions in the Mathematics and Computer Science Departments. His current research interests include mathematical image processing, computer vision, and computer graphics, computational brain mapping, VLSI physical design optimization, multiscale computational methods, multigrid and domain decomposition algorithms, iterative methods, Krylov subspace methods, parallel algorithms, and computational linear algebra.