A REGULARIZED K-MEANS AND MULTIPHASE SCALE SEGMENTATION

SUNG HA KANG
School of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332-0160, USA

BERTA SANDBERG
Physical Optics Corporation
Torrance, CA 90501-1821, USA

ANDY M. YIP
Department of Mathematics
National University of Singapore
10, Lower Kent Ridge Road, 119076, Singapore

(Communicated by Xue-Cheng Tai)

Abstract. We propose a data clustering model reduced from variational approach. This new clustering model, a regularized k-means, is an extension from the classical k-means model. It uses the sum-of-squares error for assessing fidelity, and the number of data in each cluster is used as a regularizer. The model automatically gives a reasonable number of clusters by a choice of a parameter. We explore various properties of this classification model and present different numerical results. This model is motivated by an application to scale segmentation. A typical Mumford-Shah-based image segmentation is driven by the intensity of objects in a given image, and we consider image segmentation using additional scale information in this paper. Using the scale of objects, one can further classify objects in a given image from using only the intensity value. The scale of an object is not a local value, therefore the procedure for scale segmentation needs to be separated into two steps: multiphase segmentation and scale clustering. The first step requires a reliable multiphase segmentation where we applied unsupervised model, and apply a regularized k-means for a fast automatic data clustering for the second step. Various numerical results are presented to validate the model.

1. Introduction. Image segmentation is used to partition images and facilitates the identification of certain objects or features in the images. Image segmentation and active contour are widely studied and various extensions are proposed in different settings such as [10, 25, 33, 39]. Since the work by Mumford and Shah [25] and Chan-Vese’s successful level set implementation [3], numerous extensions have been explored and various properties have been studied in variational settings. The Chan-Vese model made it possible to identify objects without a sharp boundary, and its main driving force is the intensity difference of the objects. This model is

2000 Mathematics Subject Classification. Primary: 65D18, 94A08, 62H35.
Key words and phrases. K-means, multiphase, scale, segmentation, variational model.

The first author is supported by NSF grant DMS-0707184. The third author is supported by the Academic Research Grant R146-000-116-112 from National University of Singapore.

©2011 American Institute of Mathematical Sciences
extended to incorporate more complicated settings such as multi-channel \cite{2}, texture \cite{31}, and logic model \cite{30}.

We explore the connections between data clustering and image segmentation, focusing on automatically deciding the number of clusters. Inspired by the variational setting of image segmentation, we propose a one-dimensional data clustering model. This new regularized k-means algorithm is an extension from the classical k-means model, and explores the connections between data clustering and image segmentation from the modeling point of view. Image segmentation is closely related to data clustering. Their connections are explored by some authors, e.g. a subspace approach and a graph-cut approach for clustering is adapted for image segmentation in \cite{17, 33} respectively.

The main motivation of this work is from the unsupervised multiphase image segmentation model presented in \cite{32}. Unlike two-phase\footnote{A phase, introduced in \cite{3} consists of pixels that can be well-modeled by a common set of parameters. One phase can contain multiple isolated segments.} image segmentation, when a multiple number of phases are to be found, the results become more sensitive to the choice of regularization parameters and the initial condition (in case of iterative methods). In many applications \cite{4, 14, 19, 36, 40}, the number of phases is predetermined (or a reasonable estimate is given \textit{a priori}), which parallels the case of the classical k-means algorithm. The \textit{phase balancing model} proposed in \cite{32} addresses this issue of automatically choosing the number of phases $k$ through the minimization of energy functional:

\begin{equation}
E [\bar{K}, \{\chi_i\}, \{c_i\}|u_o] = \mu \left( \sum_{i=1}^{\bar{K}} \frac{P(\chi_i)}{|\chi_i|} \right) \sum_{i=1}^{\bar{K}} P(\chi_i) + \sum_{i=1}^{\bar{K}} \int_{\chi_i} |u_o - c_i|^2 \, dx.
\end{equation}

The notations $P(A)$ and $|A|$ represent the one-dimensional Hausdorff measure and the Lebesque measure of a phase $A$ respectively. Note that in addition to the phases $\{\chi_i\}$ and the intensity averages $\{c_i\}$, the number of phases $\bar{K}$ is also an unknown; only the observed image $u_o$ is given. By minimizing the functional, a reasonable number of phases is found as the image is segmented. As a numerical method, a fast algorithm is proposed which demonstrated stable results. The model has one parameter $\mu$ that allows different choices for the number of phases, while presetting it to $\mu=1$ (for intensity range $[0, 255]$), unsupervised segmentation is possible. Motivated by the ability to automatically find a reasonable number of phases, we propose a one-dimensional data clustering model which is reduced from the main idea of this phase balancing model (1).

We also consider multiphase scale segmentation in this paper. We classify objects by the scale in addition to the intensity. A typical Mumford-Shah-based image segmentation is driven by the intensity difference, despite of the fact that the size of the objects can also provide a meaningful classification. The scale of an object is not a local value, that scale classification needs to be proceeded by a dependable multiphase segmentation. We apply unsupervised model (1), then we further classify the objects according to the size of each connected component by the new regularized k-means model.

In this paper, we present a regularized k-means model, explore its properties, and present a numerical algorithm. In section 2, the details of the the regularized k-means model are presented. Comparisons with the classical k-means are discussed.
in subsection 2.1, and related works of the regularized k-means are listed in subsection 2.2. The proposed model shows preference for finding bigger clusters and stability against outliers. In section 3, a fast algorithm for the regularized k-means is proposed for stable realizations of the model, while its stopping criterion gives some insights on how the number of clusters is determined. In section 4, the main process of multiphase scale segmentation is presented, and various numerical results in subsection 4.1 show how using scale clustering can improve segmentation results. In section 5, numerical experiments show the performance of the regularized k-means are farther presented, followed by concluding remarks in section 6.

2. A regularized k-means model. For a given data set \(D = \{d_1, d_2, \ldots, d_n\} \subset \mathbb{R}^1\), the classical k-means problem formalized by MacQueen [22] is to find a set of \(\hat{k}\) clusters which minimizes the following energy,

\[
E[\{I_i\}|D, \hat{k}] = \sum_{i=1}^{\hat{k}} \sum_{d_j \in I_i} |d_j - c_i|^2.
\]

Here \(I_i\) represents the \(i\)-th cluster and \(c_i\) is the average over \(I_i\) for \(i = 1, \ldots, \hat{k}\). The number of clusters \(\hat{k}\) is typically given a priori or determined by some heuristic steps. When the number of clusters \(\hat{k}\) is given, there are many algorithms to compute a solution. Most classical algorithms [9, 12, 20, 22] are based on two basic steps: (1) assign a point to its nearest center; (2) update the cluster centers. These algorithms differ in how the two steps are interlaced. The Ward’s method [41] uses an agglomerative hierarchical procedure which locally optimizes the objective for each \(\hat{k}\). Some more recent algorithms include deterministic annealing [29], cutting-plane [28], and genetic algorithm [16]. The k-means problem assumes a fixed \(\hat{k}\). However, in many practical cases, \(\hat{k}\) is unknown. If \(\hat{k}\) is allowed to vary, then a trivial solution exists—simply let each datum form its own cluster. Therefore, some additional constraints or a priori knowledge about the clusters are necessary to make the problem meaningful.

We propose to add a regularization term to determine the number of clusters via minimization of the energy. We propose the following regularized k-means energy,

\[
E[\{k, \{I_i\}, \{c_i\}|D] = \lambda \left( \sum_{i=1}^{k} \frac{1}{n_i} \right) + \sum_{i=1}^{k} \sum_{d_j \in I_i} |d_j - c_i|^2,
\]

which is to be minimized for a given one-dimensional data set \(D = \{d_1, d_2, \ldots, d_n\}\). Here \(k\) is the number of clusters, \(n_i = |I_i|\) is the number of data in the cluster \(I_i\), and \(c_i\) is the average of data in \(I_i\). In this paper, we use \(f(t) = \frac{1}{t} \) for regularization (as long as \(f(t)\) is convex and satisfy \(f(s + t) < f(s) + f(t)\) for all \(s, t > 0\), different functions can be applied).

The first term regularizes the solution via the size of each cluster. The second term measures the spread (intra-cluster dissimilarity) of the clusters. In (3), if \(\lambda\) is set to zero and the number of clusters \(k\) is not given, the minimum of the second term will be achieved when each data point is its own cluster, i.e. \(k = n\). To regularize, the size of the clusters are incorporated into the first term which prefers to have all the data in one cluster. This regularization favors large clusters, and bounds the number of clusters such that \(k < n\). Together, the proposed model
(3) maintains the tightness of the clusters while avoiding over-fitting by having too small clusters.

2.1. Analysis of the regularized k-means compared to the classical k-means. In the classical k-means model (2), since a fixed number $\hat{k}$ is given, the distribution of data will strongly affect the clustering result. In particular, it prefers to have similar distribution (squared error) $|d_j - c_i|^2$ among the $\hat{k}$ clusters. To illustrate this effect, we consider the distribution $\gamma_i$ in each cluster:

$$\sum_{i=1}^{\hat{k}} \gamma_i \min_{n_i} \gamma_i \leq \sum_{i=1}^{\hat{k}} \sum_{d_j \in I_i} |d_j - c_i|^2 \leq \sum_{i=1}^{\hat{k}} \gamma_i \max_{n_i} \gamma_i.$$ 

Here $\gamma_i \min_{n_i}$ and $\gamma_i \max_{n_i}$ represent the minimum and maximum distribution within the cluster $I_i$ respectively. Let us assume there exist $\gamma_i^*$ for $i = 1, 2, \ldots, \hat{k}$ such that

$$\sum_{i=1}^{\hat{k}} \sum_{d_j \in I_i} |d_j - c_i|^2 \approx \sum_{i=1}^{\hat{k}} \gamma_i^* n_i.$$ 

This setting shows that, to minimize such an energy, the result will try to balance the value of $\gamma_i^* n_i$ among the $\hat{k}$ clusters. This is due to having a fix $\hat{k}$ and minimizing the summation term. Each cluster will prefer to have a similar value of $\gamma_i^* n_i$, i.e. a cluster $I_1$ with many data will be distributed closer to the center $c_1$, compared to another cluster $I_2$ with a smaller number of data which can be distributed farther from the center $c_2$, as long as $\gamma_1^* n_1$ and $\gamma_2^* n_2$ are similar. Therefore, the clustering result is heavily influenced by the choice of $k$ and strongly depends on how the data are distributed (via the values of $|d_j - c_i|^2$).

The proposed model is an improvement over the classical k-means. From a similar setting as above, the proposed model becomes,

$$\lambda \sum_{i=1}^{k} \frac{1}{n_i} + \sum_{i=1}^{\hat{k}} \gamma_i^* n_i.$$ 

This model has the same fitting term and it will give some balance among the distribution of the clusters. The regularization term prefers bigger clusters (a bigger number for each $n_i$) and stabilizes the process. The fact that $k$ is chosen from the minimizer of the functional makes this method flexible, and outliers are not forced into a cluster due to a set number of clusters. (Flexibility of the choice $\lambda$ is further discussed in Section 5.)

2.1.1. Stability of bigger clusters. With a regularization term such as $f(t) = \frac{1}{t}$, a clustering result favors fewer and bigger clusters. Figure 1 is an example showing the stability of the regularized k-means compared to the classical k-means. The size of the data set is $n = 10,000$, among which 70% are drawn from the normal distribution $N(0, 0.06^2)$, 15% from $N(0.8, 0.02^2)$, and the other 15% from $N(0.5, 0.02^2)$. The graphs are representing the data in histogram form. Plot (a) is the regularized k-means result using $\lambda = 0.1$ and plot (b) is the classical k-means result with $k = 3$ assigned. The result in (a) is more favorable compared to the result in (b). If the classical k-means algorithm is run multiple times, one can get a similar result to (a), yet, it only happened once in about five trials for this example. (The regularized k-means gave the same result regardless of number of trials.)
2.1.2. Stability against outliers. Since the classical k-means algorithm depends strongly on the distribution of data, it can become unstable when outliers or additional datum are introduced. With the regularized k-means this effect is reduced, thanks to flexibility of automatically adjusting the number of clusters. Additional clusters can be created to deal with the outliers, that the clustering of the remaining data is largely unaffected.

Figure 2 (the size of the data set is $n = 10,000$) (a) shows a result of using the regularized k-means with $\lambda = 0.1$, which automatically gave three clusters. This result is exactly the same as using the classical 3-means. When new additional datum are introduced to (a), using the classical 3-means gave (b) as a result. Here the intervals are all shifted: in (a) the intervals are $I_1= [0.0052, 0.2858]$, $I_2= [0.2861, 0.5156]$, and $I_3= [0.5163, 0.8490]$, while in (b), $I_1= [0.0052, 0.2888]$, $I_2= [0.2892, 0.5368]$, and $I_3= [0.5370, 1]$. With the regularized k-means (with the same $\lambda = 0.1$), the algorithm automatically gave 4 clusters as in (c). When the classical 4-means are used, the result was exactly the same as (c).

We can observe a couple of effects: (i) For a clearly separated data set, the classical k-means and the regularized k-means can give exactly the same results. (ii) When $k$ is not properly changed in the classical k-means, all of the cluster intervals can shift, showing it can easily be effected by outliers. In (a) the intervals are $I_1= [0.0052, 0.2858]$, $I_2= [0.2861, 0.5156]$, and $I_3= [0.5163, 0.8490]$, and in (c), $I_1= [0.0052, 0.2858]$, $I_2= [0.2861, 0.5156]$, $I_3= [0.5163, 0.7703]$ and $I_4= [0.7723, 1]$. The first two intervals are not effected, while only the third interval is adjusted to handle the outliers. These results are that of the regularized k-means (with the same $\lambda = 0.1$) and the classical k-means with a properly chosen $k = 3$ and $k = 4$. Notice in (b), all three intervals are shifted to include the new datum with a fixed $k = 3$ for the classical k-means.
Figure 2. [Automatic k adjustment] Given data set of \( n = 10,000 \). (a) using the regularized k-means with \( \lambda = 0.1 \) and the classical 3-means result (both are exactly the same). A classical 3-means result with new added datum in (b). All intervals are shifted compared to (a) to handle the new datum. (c) using the regularized k-means with \( \lambda = 0.1 \) - which automatically adjusted \( k \) to handle the new datum. The classical 4-means result give the same result as in (c) when \( k = 4 \) is given. This shows, the regularized k-means can handle the data change automatically with a fixed parameter \( \lambda \).

2.1.3. Effect of the regularization term. The regularization term we used has two basic properties. First, it decreases with \( k \) in the sense that if two clusters of size \( n_1 \) and \( n_2 \) respectively are merged, then the regularization term decreases, i.e. \( \frac{1}{n_1 + n_2} < \frac{1}{n_1} + \frac{1}{n_2} \). The regularization term is minimized when all data points are assigned to one cluster. Second, clusters of equal sizes are preferred, i.e. \( \frac{1}{2n} \cdot \frac{1}{2n} \leq \frac{1}{n_1} + \frac{1}{n_2} \) for any \( n_1 \) and \( n_2 \) such that \( n_1 + n_2 = n \). In fact, we have

\[
\frac{1}{n_1 + n_2} < \frac{1}{2n} \cdot \frac{1}{2n} \leq \frac{1}{n_1} + \frac{1}{n_2}.
\]

Therefore, the “small \( k \) effect” will eventually overrule the “balancing effect” when the regularization parameter is large enough. Intuitively speaking, decreasing \( k \) by one would induce a relatively large increase in the sum-of-squares error. Thus,
before $\lambda$ is large enough to compensate such an increase in the fitting error, the balancing effect takes place to fine tune the clusters (although the changes may not be very significant).

We illustrate these properties in Figure 3. In (a), a data set with two “true” clusters (red and blue) and the optimal 2-means clustering (obtained by enumerating all contiguous clusterings with $k = 2$) are shown. The true blue and red clusters have 100 and 200 points respectively. Due to the difference between the spread of the two clusters, there are 14 points that are “misclassified” by the 2-means method. In (b), the optimal clustering of the regularized k-means with respect to $\lambda = 20, 30, \ldots, 180$ are shown. We remark that when $\lambda \leq 10$, the optimal clustering has $k \geq 3$; when $\lambda \geq 180$, the optimal clustering has $k = 1$. We observe that as $\lambda$ increases from 20 to 80, the number of misclassified points decreases due to the tendency of the regularization term to balance the size of the clusters. When $80 \leq \lambda \leq 170$, where the size of the two clusters are 100 and 200 respectively, the balancing effect ceases. This is because the left-most points of the red cluster are quite far from the center of the blue cluster. Thus, further balancing would cause a relatively large increase to the sum-of-squares error. When $\lambda \geq 180$, the small $k$ effect takes over so that a single big cluster becomes the optimal choice. The graph (c) depicts the number of misclassified points for $20 \leq \lambda \leq 170$.

2.2. Related work on regularized k-means. Appending a regularization objective to the k-means objective has been studied by many authors for different purposes. In [15, 18, 24], an entropy of the cluster membership is added to achieve fuzzy clustering. The fuzziness is controlled through the regularization parameter. The algorithm in [18] is agglomerative which finds a sequence of nested clusterings corresponding to a decreasing sequence of regularization parameters. The final number of clusters is then chosen by using an external cluster validation measure. In [27], an entropy of the cluster sizes is added so that the number of clusters $k$ is part of the objective to optimize. In [38], the number of outliers is added to make the k-means problem less sensitive to outliers.

If we take a broader view that the k-means problem is a special instance of mixture modeling [6, p.526], then appending a regularization term is equivalent to introducing a prior distribution to the model parameters in the Bayesian framework, under some mild assumptions on the distributions. In particular, several prior distributions of $k$ have been considered. They include the Laplace-empirical criterion, Bayesian inference criterion, minimum description length, minimum message length, Bayesian information criterion, see [23, 8] and the references therein.

In our approach, we do not regularize $k$ directly. Instead, we regularize the size of the clusters via the term $\sum_{i=1}^{k} \frac{1}{n_i}$ which favors a smaller $k$. A similar approach is considered in [27] where an entropy of the cluster sizes is used. We consider those $f$s that stem from geometric concepts in variational image segmentation. The regularization term in [27] is

$$\sum_{i=1}^{k_{\text{max}}} f(n_i)$$

where $k_{\text{max}}$ is a predetermined upper bound of $k$ and $f(n_i) = -(n_i/n) \ln(n_i/n)$. Here, $n_i$ can take zero value to account for empty clusters. The regularization term used in our model does not allow $n_i = 0$; empty clusters are simply dropped from the expression. This implies that moving one point from a large cluster to a
new cluster would induce a relatively large increase in our regularization term (i.e. $f(n_i) \ll f(n_i - 1) + f(1)$), yet only a small increase in entropy (i.e. $f(n_i) + f(0) \approx f(n_i - 1) + f(1)$). Thus our model has a great resistance to the formation of small spurious clusters.

Modifying the standard k-means to balance the clusters has been studied in different contexts, e.g., vehicle routing [13] and scheduling in wireless sensor networks [37]. These algorithms are domain specific which take into consideration some special structure of the data. The model proposed in [42] assumes the providence of a set of class labels. The goal is to construct clusters such that each cluster contains a similar number of data from each class and that the cluster size is balanced. Each of the two objectives is achieved by adding a sum-of-squares error term.

3. A regularized k-means algorithm. In variational settings, a typical approach to find a minimizer of the functional is to consider its Euler-Lagrange form and apply a gradient descent method. For the proposed regularized k-means model, the number of clusters $k$ is also an unknown that considering a gradient direction requires tedious computation of topological derivatives. Therefore, we propose to directly consider the change in the energy for each data point to find a solution, as in [11, 32, 34]. This is also possible since we are working on a discrete setting of

Figure 3. [Effect of the regularizing term] Data contains two clusters of size 100 and 200 respectively. (a) The data set and the optimal 2-means clustering. (b) The optimal solution of the regularized k-means model for $20 \leq \lambda \leq 180$. (c) A plot of the number of misclassified points versus $\lambda$. 
A regularized k-means algorithm

- Input data set $D$ and $\lambda$. Let $k = 1$ and assume all the data are in one cluster.
- Iterate
  1. For each datum $d_j \in I_i$, compute $\Delta E_{il}$ as in (4) for all $l \neq i$, and set $\Delta E_{ii} = 0$.
  2. Update $k$ if necessary, update $n_i = |I_i|$ and $c_i$ for the clusters that undergo changes.

Table 1. A regularized k-means algorithm

$D$. The main idea of [34] is to consider the change in the energy directly, that each pixel is moved to the phase which locally minimizes the energy compared to the previous phase. Each iteration of this greedy algorithm is very fast, due to a simple format of the change in the energy functional. As in [32], we also add an option of creating a new cluster to allow changing the number of clusters $k$.

It is important to notice that this algorithm starts with a simple initial condition: all data are assumed to be in one cluster and the number of clusters is set to be $k = 1$. The model dynamically adjusts the number of clusters during the iteration of algorithm. From the proposed model (3), we consider the energy difference $\Delta E_{il}$ for each pair $(i, l)$, which represents the energy change when a datum $d_j \in D$ is moved from cluster $I_i$ to $I_l$. For each data $d_j \in D$, the energy change is computed by

$$\Delta E_{il} = E_i - E_i = \lambda \left[ \frac{1}{n_i(n_i - 1)} - \frac{1}{n_l(n_l + 1)} \right] + (d_j - c_l)^2 \frac{n_l}{n_l + 1} - (d_j - c_i)^2 \frac{n_i}{n_i - 1}.$$

If $\Delta E_{il} > 0$, then the datum $d_j$ will not be moved to $I_l$ since that will increase the energy. If $\Delta E_{il}$ is negative, then it is better to move $d_j$ to cluster $I_l$. A new cluster $I_{k+1}$ with $n_{k+1} = 1$ is created, when the following is the smallest negative value among $\Delta E_{il}$ for $l = 1, 2, \ldots, k+1$:

$$\Delta E_{i,k+1} = E_{k+1} - E_i = \lambda \left( \frac{1}{n_i(n_i - 1)} + 1 \right) - (d_j - c_l)^2 \frac{n_i}{n_i - 1}.$$

Table 1 shows the algorithm, where each datum $d_j$ is moved to the cluster that minimizes the energy.
Figure 4. Typical intensity based segmentation. (a) Original given image. (b) and (c) are showing two out of five phases using unsupervised multiphase segmentation model [32]. These are represented using characteristic functions, indicating the location of black and one of darker gray regions. The segmentation is driven by the intensity, that each phase has many different sized objects mixed together.

A necessary condition of creating a new cluster can be computed by considering the sign of $\Delta E$. The algorithm will add a new cluster as long as the following holds

$$\lambda \left(1 - \frac{1}{n_i} + \frac{1}{n_i^2}\right) < (d_j - c_i)^2.$$  

As $n_i$ increases, $|d_j - c_i|^2$ needs to be bigger to create new clusters, e.g. when one cluster has many data points, it will attract more points, unless it is far from the center $c_i$. This inequality also shows that $\lambda$ should be chosen depending on the squared-error $(d_j - c_i)^2$ and $n_i$ (or $f(n_i)$). For example,

$$\lambda \approx C(d_j - c_i)^2 n_i^2 \left(\text{or } C(d_j - c_i)^2 \frac{n_i}{f(n_i)}\right).$$

4. **Multiphase scale segmentation.** We propose image segmentation using the scale of objects. One can achieve a better classification of objects by utilizing intensity and size information together. Figure 4 shows a motivation of this approach. The original image (a) has many disks with different sizes and intensities. Using intensity based segmentation gives results such as (b) and (c). This is showing two out of five phases, indicating the location of black and one of darker gray regions respectively. The segmentation is driven by the intensity, that each phase has many
different sized objects mixed together. For some medical applications, it may be meaningful to cluster the objects according to the size of objects in addition to the intensity.

For the notion of size, we use the scale term such as

\[ S(A) := \frac{P(A)}{|A|}, \]

where \( P(A) \) denotes the perimeter of a set \( A \) and \( |A| \) denotes the 2-dimensional area of a set \( A \). This term is inversely proportional to the size of the object. The notation is used as an automatic parameter scaling in unsupervised multiphase segmentation model [32] and in [35], the inverse is used in the context of total variation (TV) denoising [35]. This term \( S \) is related to the Cheeger Set, where the objective is to find a nonempty set \( A \subset \Omega \) of finite perimeter which minimizes \( \min_{A\subset\Omega} S(A) \). This is widely studied in the calculus of variation analysis and some references include [1, 7].

We propose to use scale information for a better classification. Using scale for object classification is considered in a different context and a good literature review on recent approaches can be found in [21]. Unlike intensity based image segmentation, scale is not a local value. Each pixel is not aware of the scale of the connected component it belongs to, until each connected component is identified. Therefore, scale segmentation and intensity based segmentation can not be performed simultaneously and a stable multiphase segmentation is necessary as a pre-processing step. Any multiphase segmentation method that partitions image into separate objects can be applied. We apply the phase balancing multiphase segmentation model (1) from [32]:

\[
E[\bar{K}, \{\chi_i\}, \{c_i\}] = \mu \left( \sum_{i=1}^{K} \frac{P(\chi_i)}{|\chi_i|} \right) + \sum_{i=1}^{K} \sum_{d_j \in I_i} |u_{a_i} - c_i|^2,
\]

since it is stable and able to handle sensitivity issues of multiphase segmentation, such as initial condition dependence and the pre-assigned number of phases.

After the step of multiphase segmentation, the image domain \( \Omega \) is now partitioned by phases, \( \chi_a \), according to their intensity value, i.e. \( \Omega = \bigcup_{a=1}^{k} \chi_a \) and \( \chi_a \cap \chi_b = \emptyset \) for \( a \neq b \). Each phase \( \chi_a \) may contain many separate connected components, and we further label each connected component by \( \chi_{a,b} \) such that

\[ \chi_a = \bigcup_{b=1}^{l_a} \chi_{a,b}, \quad \chi_{a,b} \cap \chi_{a,c} = \emptyset \quad \text{for} \quad b \neq c. \]

The scale value for each connected component \( \chi_{a,b} \) is computed by

\[ S(\chi_{a,b}) = \frac{P(\chi_{a,b})}{|\chi_{a,b}|}. \]

We define an one-dimensional data set to be

\[ D = \{ S(\chi_{a,b}) | b = 1, 2, \ldots, l_a, \ a = 1, 2, \ldots, k \}. \]

This data set \( D \) contains various scale values of objects in the image. The size of \( D \), \( n \), is the number of connected components (objects) in the image. We only consider the case when \( n < \infty \), that there are finitely many objects which is the case for the discrete setting. A grouping of the objects is found by minimizing the energy (3),

\[
E[k, \{I_i\}, \{c_i\}] = \lambda \left( \sum_{i=1}^{k} \frac{1}{n_i} \right) + \sum_{i=1}^{k} \sum_{d_j \in I_i} |d_j - c_i|^2.
\]
Figure 5. (a) Original given image. (b) A result using intensity based segmentation, showing the light green regions. (c) A result using scale segmentation to identify the clover clearly. $\lambda = 0.25$ is used.

4.1. Numerical experiments for multiphase scale segmentation. For numerical experiments, after the data set $D = \{S(\chi_{a,b})|\chi_{a,b} \subset \Omega\}$ is computed, we deleted big values of scale (if exist), which represent noise (or very small objects). A typical size of the data $D$ for image scale segmentation was around 40-50 in our experiments.

Figure 6 & 7: general effects of scale segmentation. Figure 5 shows an example where using scale segmentation can further improve the segmentation result. Image (b) is using intensity segmentation, where light green regions are all identified together. By further clustering scale values with a regularized k-means algorithm, the clover is clearly identified in image (c). The large clover has a different scale value compared to that of the narrow leaves.

Figure 6 is an example using a color blind test image. Since there are some color differences in color-blind test images, using intensity based segmentation method can identify the letters in some of multiple phases. Figure 6 (b) and (c) are showing two phases of intensity based segmentation, where the letter B is clearly identified.

Using scale segmentation, objects are classified according to the size of disks in the images. Third row of Figure 6, image (d), (e) and (f), shows different phases separating objects according to their scale values. Here $\lambda = 0.02$ is used and Figure 6 are showing three out of six phases of scale segmentation. Graph (e) shows the distribution of the scale values in Figure 6 (a). Red intervals indicate clusters of
Figure 6. (a) Original given image. (b) and (c) are two phases using intensity segmentation, clearly showing the letter. (d), (e) and (f) are using scale segmentation. Considering only scale of the connected components, the image is separated according to the size of each disk. $\lambda = 0.02$ is used and only showing three phases among six phases. (g) The scale distribution of image (a). We discarded values bigger than 1.5 for more reliable clustering and to remove noise. Red intervals indicate a clustering result of the scale data. The three phases (d), (e) and (f) correspond to the first three intervals in the histogram (e).

the scale values obtained using the regularized k-means algorithm. We discarded values bigger than 1.5 to disregard noisy small objects. Three phases image (d), (e) and (f) correspond to the first three intervals in the histogram (e), and other three intervals give even smaller objects.
Figure 7. (a) Original given image. (b) and (c) are two phases using intensity segmentation. (d) and (e) are two phases using scale segmentation with $\lambda = 0.5$. Each phase is clearly separated according to the size of the object. Bigger cluster is clearly identified in (d) while smaller stars are in one phase (e).

Figure 7 shows a scale segmentation result for a star cluster image. Image (b) and image (c) are two phases (among three phases) using the intensity segmentations. Image (b) represents very bright regions and image (c) less brighter regions. Second row results, image (d) and (e), clearly distinguish different sizes by the scale clustering: bigger clusters are clearly identified in image (d) and smaller stars in (e).

Figure 8 & 9: medical application, cell images. Original cell image (a) contains many different colors and shapes of cells. Using intensity based multiphase
Figure 8. (a) Original given image. (b) and (c) are two phases using intensity segmentation (background phase not shown). Using the regularized k-means, one can further distinguish objects of different scales. (d) and (e) show two of nine phases further separated from image (b) and (c). The cells with different shape/sizes are clearly distinguished compared to only using intensity values.

Using the regularized k-means in addition, we can further distinguish objects according to their scales. Figure 8 (d) and (e) show two of nine phases separated according to the scale of objects. The model further separates each connected components in (b) and (c) according to their scale, and makes it easier to identify irregular cells or differently shaped objects. Compared to image (b), using scale segmentation clearly separates different types of cells. We used scale values below 0.4 and used $\lambda = 0.0005$. The objects with scale values above 0.4 represent small noisy regions and it is shown in Figure 9 (b). Figure 9 (a) shows a scale value distribution in histogram form, and red bars represent each cluster interval using the regularized k-means.

4.2. Remarks on using scale term. The scale value $\frac{P(A)}{|A|}$ only uses a boundary length term and an area term. Therefore, there are some limitations on what kind of objects this scale term can distinguish. This term will distinguish disks from thin
Figure 9. (a) The scale value histogram of Figure 8(a). Red intervals indicate scale clustering using $\lambda = 0.0005$. We discarded scale values bigger than 0.4 for more reliable clustering and this value is roughly chosen to remove the noise. (b) shows all the objects with scale values above 0.4.

Figure 9. (a) The scale value histogram of Figure 8(a). Red intervals indicate scale clustering using $\lambda = 0.0005$. We discarded scale values bigger than 0.4 for more reliable clustering and this value is roughly chosen to remove the noise. (b) shows all the objects with scale values above 0.4.

lines pretty clearly, but it may not be ideal for distinguishing small oscillations on the boundary. For example, a regular polygon with corners can have the same scale value as a disk without any corners. The following property shows this example when different objects can have the same scale value [32].

**Proposition 1.** Any regular (equilateral) polygon $B$ that inscribes a circle with radius $r$ has a scale $P(B) = \frac{2}{r}$. So does the limiting circle with radius $r$.

Since scale segmentation is a two step processing, a scale clustering result will be influenced by the multiphase segmentation result. If two of the same shaped objects are touching and identified as one connected component by the multiphase segmentation, then the two objects will be identified as one big object. The scale value of this big object can be different from that of each separate object.

Using scale information is different from using a shape prior for image segmentation, but it is flexible as a general classification. It may not be ideal for identifying specific shape, such as finding a particular shaped triangle. But, using scale segmentation will find similar sizes and will distinguish elongated shapes from rounded compact objects.

This approach of scale clustering can be related to histogram segmentation such as [5, 26]. A clear difference is that we are proposing a data clustering algorithm, and the histogram clustering result is not directly related to the performance in the object identification.

**Note:** we considered some variations of the model to include the scale as well as the intensity in one segmentation model, such as

$$E[k; \{\chi_i\}, \{c_i\}, \{s_i\}|u_0] = \mu \sum_{i=1}^{k} \left( \frac{P(\chi_i)}{\lambda_i} - s_i \right) \sum_{i=1}^{k} P(\chi_i) + \sum_{i=1}^{k} \int_{\chi_i} |u_0 - c_i|^2 dx,$$

where $s_i$ represents scale values of connected components. This model and variations of such model were not successful, due to a simple but an important fact that the scale value is a non-local term. For example, while one region is growing, it can get stuck at the size where many of other objects are, and over-segment one connected component. In addition, each pixel is not aware of its own location, finding the
scale value $s_i$ became very unstable. One can consider adding the scale term in the model, however, most likely it will require a two-step algorithm separating scale computation and intensity segmentation.

5. **Numerical experiments of the regularize k-means model.** In this section, we further present the effects of the regularized k-mean algorithm (3). The data sets are generated either from probability distributions or from a histogram of some images. These graphs are plotting in the form of probability distribution and different colors indicates different clustering results by the regularized k-mean algorithm (3).

[Different $\lambda$ effect] The first example shows how different choice of $\lambda$ effects the results. Figure 10 (a) shows a given data set of size $n = 20,000$. Data derived from a histogram of a general image. The parameter $\lambda$ is changed from 0.1, 0.15, to 0.3, and the algorithm gives four, three and two number of clusters respectively. The given data is reasonably separated among the region with a high peak and other regions. This example shows an effect of the parameter $\lambda$.

[$\lambda$ vs $n$] The stopping criteria (5) and the $\lambda$ term (6) show that the parameter can depend on the size of the data set. We experimented with different sizes of data set, and for a data set of size between $n \approx 100$ and $100,000$, the parameter $\lambda$ stayed the same and stable. We kept the same distribution and changed the size of the data: 25% of the data are given from the normal distribution $\mathcal{N}(0.3, 0.1^2)$, 25% from $\mathcal{N}(0.6, 0.02^2)$ and the other 50% from $\mathcal{N}(0.8, 0.08^2)$. With the same value of $\lambda = 0.1$, the regularized k-means gives three clusters consistently. Figure 11 (a) shows one such case of $n = 10,000$. When a significantly small (or large) number of data is used, $\lambda$ can be adjusted. Figure 11 (b) and (c) show examples when only ten data points are given. With $\lambda = 0.1$ the proposed method gave two clusters (different from three clusters of $n > 100$). Using a smaller value $\lambda = 0.05$, the model gave three clusters in (c). With a very small number of data set, it is not clear what is a good clustering and both (b) and (c) are reasonable results.

[$\lambda$ vs $k$] The regularized k-means algorithm does not require any prior knowledge of the number $k$, however, the results are influenced by the choice of $\lambda$. It is important to explore how sensitive this choice of $\lambda$ is compared to the clustering result. Figure 12 shows graphs of a choice of $\lambda$ versus the number of clusters $k$. Clearly the graph is a step function (since $k$ is an integer) and decreasing. Notice there are large flat intervals with the same $k$ values, which shows the stability of the choice of $\lambda$. The shape of this graph has relations to the distribution of the data.
Figure 11. $\lambda$ vs the size of datum $n$] Data is generated with the same distribution while the size of the data set changed from $n = 100,000$ to $n = 100$, and (a) is showing the case of $n = 10,000$. With $\lambda = 0.1$, all experiments with $n = 100,000$ to $n = 100$ gave three clusters similar to (a). The histogram (b) and (c) are data set of size 10. With $\lambda = 0.1$, the regularized k-means gave two clusters in (b) - which is a reasonable result with only 10 datum. Using a smaller value $\lambda = 0.05$, one can also achieve three clusters as in (c).

Figure 12. $\lambda$ vs $k$] (a) Given data. (b) A graph of $\lambda$ verses the number of clusters $k$. Clearly the graph is step function, since $k$ is an integer and decreasing. Notice there are large flat regions with the same $k$ values.

We experimented with the distribution of the data. Consider a data set generated with $\mathcal{N}(0.3, 0.05^2)$ and $\mathcal{N}(0.7, 0.05^2)$ with different proportions. We fixed the parameter to be $\lambda = 0.1$. Figure 13 is the case when only 1% of data is in the interval around 0.3, and the rest is around 0.7. The algorithm finds two intervals.
Figure 13. Distribution of datum (a) Only 1% of datum are around 0.3, while 99% of the datum are around 0.7. With $\lambda = 0.1$, the algorithm gives two clusters down to 1% around 0.3. At this distribution of 1% and 99%, some trials farther separates the big cluster and give three clusters as in image (b).

separately. Until 1%, the algorithm seems stable and consistently finds two clusters. Around 1% with $\lambda = 0.1$, sometimes it can also give three clusters further separating the big cluster into two clusters (b).

[**distribution vs $\lambda$**] We tested a data set with two clusters of uniform distribution and normal distribution around $[0.4 - \delta, 0.4 + \delta]$ and $[0.6 - \delta, 0.6 + \delta]$, with a varying $\delta$. The two intervals have the same number of data. For each different $\delta$, $\lambda$ is found to give two clusters as a result. These experiments show that the parameter $\lambda$ is dependent on the distribution of the data. As the intervals get wider and closer to each other, a bigger $\lambda$ is needed to identify the two clusters.

Figure 14 (a) shows one case of uniform distribution with $\delta = 0.09$, (b) the graph of $\delta$ (x-axis) verse $\lambda$ (y-axis) for uniform distribution. (c) shows the case of normal distribution with $\delta = 0.1$. (d) and (e) are the graph of $\delta$ (x-axis) verse $\lambda$ (y-axis) for normal distribution. The graph (d) has a jump around $\delta \approx 0.06$, since it is where two clusters start to merge and start to look like one cluster as in (c). The graph (e) is the case of two normal distributions at $[0.3 - \delta, 0.3 + \delta]$ and $[0.7 - \delta, 0.7 + \delta]$ (data not shown). This case the graph (e) is more uniform compared to (d). (Note these values of $\lambda$ are not a definite value, but one among many possible values, as illustrated in the case of $\lambda$ vs $k$ in Figure 12.) Furthermore, the range of $\lambda$ varies a little for uniformly distributed data (b), y-axis range $[0.01, 0.022]$, while normal distributed data, the value of $\lambda$ is larger. The range of the graph in (d) is $[0, 0.14]$ and (e) $[0.04, 0.24]$. This indicates that for completely separated clusters, the choice of $\lambda$ can be more sensitive, compared to the case of normal distribution case.

6. **Concluding remark.** We propose a data clustering model reduced from variational approach. The model is regularized from the classical k-means using the number of data in each cluster. We consider $\frac{1}{n}$ that stems from geometric concepts in variational image segmentation. A fast greedy algorithm was directly utilized to each data point to efficiently obtain a solution. The proposed regularized k-means algorithm is stable and gives reasonable number of clusters automatically. If the data are scaled between $[0,1]$, $\lambda = 0.1$ is a reasonable choice. The algorithm is
(a) uniform $\delta = 0.09$. (b) $\delta$ vs $\lambda$

(c) normal $\delta = 0.1$. (d) and (e) are the graph of $\delta$ (x-axis) verse $\lambda$ (y-axis) for normal distribution as in (c). The graph (d) has a jump around $\delta \approx 0.06$, since it is where two clusters start to merge and start to look like one cluster as in (c). The graph (e) is the case of two normal distributions at $[0.3 - \delta, 0.3 + \delta]$ and $[0.7 - \delta, 0.7 + \delta]$ (data not shown). This case the graph (e) is more uniform compared to (d).

Figure 14. [Different probability distribution] (a) Two clusters of uniform distribution around $[0.4 - \delta, 0.4 + \delta]$ and $[0.6 - \delta, 0.6 + \delta]$ with $\delta = 0.09$. (b) A graph of $\delta$ (x-axis) verse $\lambda$ (y-axis) for uniform distribution. The parameter $\lambda$ is chosen to give two clusters. (c) Two clusters of normal distribution of $[0.3 - \delta, 0.3 + \delta]$ and $[0.7 - \delta, 0.7 + \delta]$ with $\delta = 0.1$. (d) and (e) are the graph of $\delta$ (x-axis) verse $\lambda$ (y-axis) for normal distribution as in (c). The graph (d) has a jump around $\delta \approx 0.06$, since it is where two clusters start to merge and start to look like one cluster as in (c). The graph (e) is the case of two normal distributions at $[0.3 - \delta, 0.3 + \delta]$ and $[0.7 - \delta, 0.7 + \delta]$ (data not shown). This case the graph (e) is more uniform compared to (d).

stable with respect to the size of the data $n \approx 100$ to $n \approx 100,000$. The $\lambda$ and $k$ relation shows a stable choice of $k$ for a wide range of $\lambda$. Various experiments are performed to show the effects of the model.

For this type of NP-hard problem, we are not guaranteed convergence to the global minimum. Variations in the minimum of the algorithm are illustrated in Figure 15. This example is similar to Figure 12, except for each different $\lambda$, we randomly permuted the order of how we input the data $D$. The data set itself is the same and only the input order is permuted randomly. The oscillations around the transitions indicate that the algorithm may get stuck in the local minimum. On the other hand, even if this is the case, the result would be reasonable, since the oscillations are only around the transition between two neighboring intervals where both solutions would make sense. Also, the oscillations are only between two neighboring integers showing the stability of this algorithm.

Moreover, we explored using scale information for multiphase image segmentation. As shown in the experiments, this allows to further identify the objects more clearly. This application is different from using shape priors, and it clusters objects according to the scale. Adding the intensity and scale, we can also consider vectorial version of the clustering algorithm.
Figure 15. The graph of $\lambda$ verses the number of cluster $k$. This experiment is similar to that of Figure 12, except a random re-ordering of the input datum is used. The graph is a step function and decreasing, but shows some oscillations around the transition. This indicates a possibility of being stuck in the local minimum. However, the result will be reasonable, since the oscillations are only around the transitions and the values are only between two neighboring integers showing the stability of this model.

REFERENCES

[1] V. Caselles, A. Chambolle and M. Navaga, *Uniqueness of the Cheeger set of a convex body*, Pacific Journal of Mathematics, 232 (2007), 77–90.

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

[3] T. Chan and L. Vese, *Active contours without edges*, IEEE Trans. Image Processing, 10 (2001), 266–277.

[4] J. Chung and L. Vese, *Energy minimization based segmentation and denoising using a multilayer level set approach*, EMMCVPR, 3457 (2005), 439–455.

[5] J. Delon, A. Desolneux, J-L. Lisani and A-B. Petro, *A non parametric approach for histogram segmentation*, IEEE Transactions on Image Processing, 16 (2007), 253–261.

[6] R. O. Duda, P. E. Hart and D. G. Stork, “Pattern Classification,” Wiley-Interscience, New York, 2nd edition, 2001.

[7] A. Figalli, F. Maggi and A. Pratelli, *A note on Cheeger sets*, Proc. Amer. Math. Soc., 137 (2009), 2057–2062.

[8] M. A. T. Figueiredo and A. K. Jain, *Unsupervised learning of finite mixture models*, IEEE Trans. Pattern Anal. Machine Intell., 24 (2002), 381–396.

[9] E. W. Forgy, *Cluster analysis of multivariate data: efficiency vs interpretability of classifications*, Biometrics, 21 (1965), 768–769.

[10] S. Geman and D. Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Trans. Pattern Anal. Machine Intell., 6 (1984), 721–741.

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

[12] J. A. Hartigan and M. A. Wong, *A K-means clustering algorithm*, Applied Statistics, 28 (1979), 100–108.

[13] R. He, W. Xu, J. Sun and B. Zuo, *Balanced k-means algorithm for partitioning areas in large-scale vehicle routing problem*, In “Proc. 3rd Int. Symp. Intelligent Information Technology Application,” volume 3, pages 87–90, Nanchang, China, (2009).

[14] Y. M. Jung, S. H. Kang and J. Shen, *Multiphase image segmentation via Modica-Mortola phase transition*, SIAM Applied Mathematics, 67 (2007), 1213–1232.
Inverse Problems and Imaging

Volume 5, No. 2 (2011), 407–429

[15] N. B. Karayiannis, Meca: Maximum entropy clustering algorithm, IEEE World Congress on Computational Intelligence, Fuzzy Systems, 1 (1994), 630–635.

[16] K. Krishna and M. Narasimha Murty, Genetic k-means algorithm, IEEE Trans. Systems, Man and Cybernetics—Part B: Cybernetics, 29 (1999), 433–439.

[17] Y. N. Law, H. K. Lee and A. M. Yip, Semi-supervised subspace learning for mumford-shah model based texture segmentation, Optics Express, 18 (2010), 4434–4448.

[18] M. J. Li, M. K. Ng, Y. Cheung and J. Z. Huang, Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters, IEEE Transactions on Knowledge and Data Engineering, 20 (2008), 1519–1534.

[19] J. Lie, M. Lysaker and X.-C. Tai, A variant of the level set method and applications to image segmentation, AMS Mathematics of Computation, 75 (2006), 1155–1174.

[20] S. P. Lloyd, Least squares quantization in PCM, IEEE Transaction Information Theory, 28 (1982), 129–137. Technical Note, Bell Laboratories (1957).

[21] B. Luo, J.-F. Aujol and Y. Gousseau, Local scale measure from the topographic map and application to remote sensing images, SIAM Journal on Multiscale Modeling and Simulation, 8 (2009), 1–29.

[22] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, In “Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability, Volume I: Statistics,” pages 281–297, (1967).

[23] G. McIachlan and D. Peel, “Finite Mixture Models,” John Wiley and Sons, New York, 2000.

[24] S. Miyamoto and M. Mukaidono, Fuzzy c-means as a regularization and maximum entropy approach, In “Proc. Seventh Int’l Fuzzy Sysmtes Assoc. World Congress” (IFSA ’97), 2 (1997), 86–92.

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

[26] K. Ni, X. Bresson, T. Chan and S. Esedoglu, Local histogram based segmentation using the wasserstein distance, International Journal of Computer Vision, 84 (2009), 57–68.

[27] G. Palubinskas, X. Descombes and F. Kruggel, An unsupervised clustering method using the entropy minimization, Proceedings. Fourteenth International Conference on Pattern Recognition, 2 (1998), 1816–1818.

[28] J. Peng and Y. Xia, A cutting algorithm for the minimum sum-of-squared error clustering, In “Hillol Kargupta, Jaideep Srivastava, Chandrika Kamath and Arnold Goodman, editors, Proc. of the 5th SIAM Int. Conf. on Data Mining,” pages 150–160, Newport Beach, CA, 2005.

[29] K. Rose, E. Gurewitz and C. G. Fox, A deterministic annealing approach to clustering, Pattern Recognition Letters, 11 (1990), 589–594.

[30] B. Sandberg and T. Chan, “Logic Operators for Active Contours On Multi-channel Images,” UCLA CAM report 02-12, 2002.

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

[32] B. Sandberg, S. H. Kang and T. Chan, Unsupervised multiphase segmentation: A phase balancing model, IEEE Transaction in Image Processing, 19 (2010), 119–130.

[33] J. Shi and J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Machine Intell., 22 (2000), 888–905.

[34] B. Song and T. Chan, “A Fast Algorithm for Level Set Based Optimization,” UCLA CAM Report 02-68, 2002.

[35] D. Strong, J.-F. Aujol and T. Chan, Scale recognition, regularization parameter selection, and Meyer’s G norm in total variation regularization, SIAM Journal on Multiscale Modeling and Simulation, 5 (2006), 273–303.

[36] X.-C. Tai and T. Chan, A survey on multiple level set methods with applications for identifying piecewise constant functions, International J. Numer. Anal. Modelling, 1 (2004), 25–48.
[37] L. Tan, Y. Gong and G. Chen, *A balanced parallel clustering protocol for wireless sensor networks using k-means techniques*, In “Proc. 2nd Int. Conf. Sensor Technologies and Applications,” pages 300–305, 2008.

[38] G. C. Tseng, *Penalized and weighted k-means for clustering with scattered objects and prior information in high-throughput biological data*, Bioinformatics, 23 (2007), 2247–2255.

[39] Z. W. Tu and S. C. Zhu, *Image segmentation by Data-Driven Markov Chain Monte Carlo*, IEEE Trans. Pattern Anal. Machine Intell., 24 (2002), 657–673.

[40] L. Vese and T. Chan, *A multiphase level set framework for image segmentation using the Mumford and Shah model*, International Journal of Computer Vision, 50 (2002), 271–293.

[41] J. Ward, *Hierarchical grouping to optimize an objective function*, J. Amer. Statist. Assoc., 58 (1963), 236–244.

[42] S. Yan, H. Zhang, Y. Hu and B. Zhang, *Discriminant analysis on embedded manifold*, In “Tomáš Pajdla and Jiří Matas, editors, Proc. 8th ECCV,” pages 121–132, 2004.

Received September 2010; revised March 2011.

E-mail address: kang@math.gatech.edu
E-mail address: berta.sandberg@gmail.com
E-mail address: andyyip@nus.edu.sg