An Edge-Weighted Centroidal Voronoi Tessellation Model for Image Segmentation

Jie Wang, Lili Ju, and Xiaoqiang Wang

***Abstract*—Centroidal Voronoi tessellations (CVTs) are special Voronoi tessellations whose generators are also the centers of mass (centroids) of the Voronoi regions with respect to a given density function and CVT-based methodologies have been proven to be very useful in many diverse applications in science and engineering. In the context of image processing and its simplest form, CVT-based algorithms reduce to the well-known k-means clustering and are easy to implement. In this paper, we develop an *edge-weighted centroidal Voronoi tessellation* (EWCVT) model for image segmentation and propose some efficient algorithms for its construction. Our EWCVT model can overcome some deficiencies possessed by the basic CVT model; in particular, the new model appropriately combines the image intensity information together with the length of cluster boundaries, and can handle very sophisticated situations. We demonstrate through extensive examples the efficiency, effectiveness, robustness, and flexibility of the proposed method.**

***Index Terms*—Active contours, centroidal voronoi tessellations, clustering, edge detection, image segmentation.**

I. Introduction

PEOPLE have benefited a lot from technological advances in communications, entertainment, medicine, mapping, and manufacturing [2], [15], [20], which are often brought by the development of image processing techniques. Typical image processing includes image enhancement, restoration, compression and segmentation. These techniques are widely used in computer vision, feature detection, medical image processing, morphological image processing, remote sensing, and so on. To further develop these techniques and apply them in more sophisticated situations, we first need a better understanding of images. Clustering is one of the very powerful tools for retrieving generic structural information from a large set of data [9], [20]. Roughly speaking, clustering classifies a large data set into smaller data groups, such that the data in each cluster share some similarities [16], [21] which can be specified according to different applications. In the context of image processing, the data sets take the form of images.

Manuscript received October 29, 2008; revised March 27, 2009. First published June 23, 2009; current version published July 10, 2009. This work was supported in part by the National Science Foundation under Grant DMS-0609575. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sharathchandra Pankanti.

J. Wang and X. Wang is with the Department of Scientific Computing, Florida State University, Tallahassee, FL 32306-4120 USA (e-mail: xwang@scs.fsu.edu).

L. Ju is with the Department of Mathematics, University of South Carolina, Columbia, SC 29208 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2009.2021087

The model of centroidal Voronoi tessellations (CVTs) [10] has been introduced to numerous fields and applications such as image processing, data analysis, computational geometry, sensor network, numerical partial differential equations, cellular biology, statistics, and the territorial behavior of animals [10]–[12], [19], [22], [25]. In its simplest form, CVT-based algorithms reduce to the well-known k-means clustering technique. When applying the CVT model to image segmentation problems, the partition of a data set actually becomes an optimization process of choosing *generators* with respect to a special *energy*. Much of the effectiveness of CVT-based algorithms originates from this feature in image segmentation and other image processing applications. Moreover, CVT provides a general framework for the energy minimization process and allows convenient improvements, substantial generalizations of existing clustering strategies.

The central task of image segmentation is to partition an image into subsets so that the elements of each subset share similar attributes and properties. Once the partition is determined, we can easily identify the boundaries or edges which separate the clusters. In the past few years, there have been many methods developed for image segmentation, see [2]–[9], [14], [15], [20], [28], [31], [34], [35], [38], and references cited therein. Some of popular and successful techniques include the level-set method [4], [6], [31] which is typically a partial differential equation based variational method and spectral clustering algorithm which is an eigenvector based method [28]. More recently, graph-based algorithms have attracted a lot of attentions as a highly efficient and effective approach of partitioning the image into a small number of homogeneous regions [18], e.g., see the state-of-the-art works done in [14], [27], and [28].

In this paper, we improve the basic CVT-based clustering method proposed in [12] and develop a new *edge-weighted centroidal Voronoi tessellation* (EWCVT) model and corresponding EWCVT-based algorithms for image segmentation. Compared with some existing methods, our new method has several desirable advantages.

- The mathematical model of EWCVT is easy to understand and implement just like the basic CVT model, but its performance in segmenting images over the CVT model is much better, i.e., the segmentation results are more accurate. The criterion we adopt to produce an "optimal" segmentation is to minimize the *edge-weighted clustering energy* defined in Section III-A.
- The EWCVT model and its segmentation algorithms can be directly and easily applied to handle multiclusters situations [12], i.e., dividing an image into any specified number of clusters. It is well-known that the level set based algo-

rithms are often difficult to be generalized to deal with the cases with more than two clusters, e.g., in the active contour model [38], by adding more level set functions, the number of clusters could be preset as 2^n in which n is a positive integer but the formulations would be very complicated when $n \geq 2$.

- The new EWCVT-based algorithms are essentially clustering algorithms like the k-means method, so they are computationally much less expensive than the popular level set based algorithms, especially when the number of clusters is large. In addition, it is convenient to generalize and refine the CVT-based algorithms to meet various application requirements without substantially increasing the computational cost [12] while the segmentation criterion is generally sensitive to the graph-based segmentation algorithms (even small changes of the criterion may result in certain computational difficulty, like NP-hard [14], [23]).
- Each of the clusters produced by the EWCVT-based algorithms for an image can consist of several unconnected pieces in the physical space. This feature often enable us to segment the image using small number of clusters compared with other approaches.

In Section II, we review the basic CVT model and related algorithms for image segmentation. The new edge-weighted centroidal Voronoi tessellations model and corresponding implementation algorithms are discussed in details in Section III. Applications and numerical examples are then presented in Section IV together with some discussions to demonstrate the efficiency, effectiveness, robustness, and flexibility of the proposed method. Some concluding remarks are finally given in Section V.

II. Centroidal Voronoi Tessellation and Image Segmentation

Image segmentation is a process of subdividing an image into smaller pieces. In particular, the elements in each piece share some common features, e.g., roughly the same color or same brightness. The edges are naturally the boundaries between different segments of the image.

A digital image is often stored in the form of pixels, so an image can be regarded as a function u defined on a domain $\Omega \subseteq \mathbb{R}^N$ in the Euclidean space where the values of u represent the colors or the gray levels of the pixels. In this paper, we consider the most familiar images whose domains are 2-D rectangles, i.e., $\Omega \subset \mathbb{R}^2$. We note all of the ideas and algorithms below can be easily applied to higher dimensional and nonrectangular images. Let the values of u represent the intensities of the digital image. Since the pixels of a digital image are usually indexed by integer pairs, we can treat u as a discrete function defined over a set of points with integer coordinates, i.e., the point $(x, y) = (i, j)$, where (i, j) are integer pairs that range over the image domain. Thus, the domain of a rectangular image u is an index set $D = \{(i, j) : i = 1, \ldots, I,\ j = 1, \ldots, J\}$ for some positive integers I and J.

A. Centroidal Voronoi Tessellation and Clustering Energies

Let $\mathbb{U} = \{u(i, j)\}_{(i,j)\in D}$ denote the set of (not necessarily distinct) color values of the original image and $\mathcal{W} = \{w_l\}_{l=1}^L$ a set of typical colors (or brightness levels). We can easily compute the *Voronoi* region V_k in $\mathbb{U}$ corresponding to the color w_k, which is defined by

$$V_k = \{u(i, j) \in \mathbb{U} : |u(i, j) - w_k| \leq |u(i, j) - w_l| \quad \text{for } l =, 1, \ldots, L\} \quad k = 1, \ldots, L \tag{1}$$

where $|\cdot|$ is some predefined metric measure. Note that in (1) the differences between color values are compared instead of the physical distances between pixels. The set $\mathcal{V} = \{V_l\}_{l=1}^L$ is called a *Voronoi tessellation* or *Voronoi clustering* [30] of the set $\mathbb{U}$, color values of the original image. The set of chosen colors $\mathcal{W} = \{w_l\}_{l=1}^L$ are referred as the *Voronoi generators*. Clearly, we have $V_i \cap V_j = \emptyset$ if $i \neq j$ and $\mathbb{U} = \cup_{l=1}^L V_l$. The Voronoi tessellation $\mathcal{V}$ can be viewed as a special partition of $\mathbb{U}$.

Given a partition of $\mathbb{U}$, denoted by $\{U_l\}_{l=1}^L$, we normally define the *centroid* (*center of mass* or *cluster means*) of every cell U_l to be the color $\bar{w}_l \in U_l$ that minimize

$$\min_{w\in U_l} \sum_{u(i,j)\in U_l} |u(i, j) - w|^2. \tag{2}$$

For an arbitrary Voronoi tessellation $\left(\{w_l\}_{l=1}^L; \{V_l\}_{l=1}^L\right)$ of $\mathbb{U}$, we often have $w_l \neq \bar{w}_l$ for $l = 1, \ldots, L$, where $\{\bar{w}_l\}_{l=1}^L$ are the corresponding centroids of $\{V_l\}_{l=1}^L$. In other words, we generally can not expect the generators which generate the Voronoi tessellation to happen to be the centroids of the corresponding clusters.

Definition 1: If the generators of the Voronoi regions $\{V_l\}_{l=1}^L$ of $\mathbb{U}$ coincide with their corresponding centroids, i.e.,

$$w_l = \bar{w}_l, \text{ for } l = 1, \ldots, L$$

then we call the Voronoi tessellation $\{V_l\}_{l=1}^L$ a centroidal Voronoi tessellation (CVT) [10] of $\mathbb{U}$ and refer to $\{w_l\}_{l=1}^L$ as the corresponding CVT generators.

We note that the CVT may not be unique for a given image [10]. Therefore, determining a CVT of $\mathbb{U}$ is actually a process to find a set of generators $\{w_l\}_{l=1}^L$ such that $\{w_l\}_{l=1}^L$ are simultaneously the centroids of the associated Voronoi clusters $\{V_l\}_{l=1}^L$. The construction of CVTs often can be viewed as an "*energy*" minimization process [10]. For a given image u and a set of generators $\mathcal{W} = \{w_l\}_{l=1}^L$, let us define the VT energy of $\mathcal{W}$ as follows:

$$E_{VT}(\mathcal{W}) = \sum_{(i,j)\in D} \mathcal{E}_{VT}(i, j) \tag{3}$$

where $\mathcal{E}_{VT}(i, j)$ denotes the VT energy of the pixel (i, j) defined as

$$\mathcal{E}_{VT}(i, j) = \min_{l=1,\ldots,L} |u(i, j) - w_l|^2. \tag{4}$$

More generally, for any set of points (color values) $\mathcal{W} = \{w_l\}_{l=1}^L$ and any partition $\mathcal{U} = \{U_l\}_{l=1}^L$ of $\mathbb{U}$, the classical *clustering energy* of $(\mathcal{W}; \mathcal{U})$ can be defined as follows:

$$E(\mathcal{W}; \mathcal{U}) = \sum_{l=1}^{L} \sum_{u(i,j)\in U_l} |u(i, j) - w_l|^2. \tag{5}$$

Combining (1) and (3) together, we can actually rewrite the VT energy as

$$E_{VT}(\mathcal{W}) = \sum_{l=1}^{L} \sum_{u(i,j)\in V_l} |u(i,j) - w_l|^2 = E(\mathcal{W};\mathcal{V}) \quad (6)$$

where $\mathcal{V} = \{V_l\}_{l=1}^L$ are the corresponding Voronoi regions associated with $\{w_l\}_{l=1}^L$. The following result has been shown in [10].

Theorem 1: The classical clustering energy $E(\mathcal{W};\mathcal{U})$ is minimized only if $(\mathcal{W};\mathcal{U})$ form a CVT of $\cup$, i.e, $\mathcal{U}$ are Voronoi regions of $\cup$ associated with the generators $\mathcal{W}$ and simultaneously $\mathcal{W}$ are the corresponding centroids of the regions $\mathcal{U}$.

Suppose that we have determined the clusters $\{U_l\}_{l=1}^L$ for a given image represented in color space by $u(i,j)$ for $(i,j) \in D$. Then there is a natural segmentation of the image, which has L segments $\mathcal{D} = \{D_l\}_{l=1}^L$ in physical space defined by

$$D_l = \{(i,j) : u(i,j) \in U_l\}.$$

Consequently, we can rewrite the clustering energy (5) in this physical segmentation terminology as

$$E(\mathcal{W};\mathcal{D}) = \sum_{l=1}^{L} \sum_{(i,j)\in D_l} |u(i,j) - w_l|^2. \quad (7)$$

Let us define

$$\mathcal{E}(i,j) = |u(i,j) - w_{\pi_{\mathcal{U}}(i,j)}|^2 \quad (8)$$

where the function $\pi_{\mathcal{U}}$: $D \to \{1,\ldots,L\}$ tells which cluster $u(i,j)$ belongs to, i.e., for any pixel $(i,j) \in D$

$$\pi_{\mathcal{U}}(i,j) = l^*, \text{ if } (i,j) \in D_{l^*} \text{ (i.e., } u(i,j) \in U_{l^*}). \quad (9)$$

Then it also holds

$$E(\mathcal{W};\mathcal{D}) = \sum_{(i,j)\in D} \mathcal{E}(i,j).$$

B. Basic CVT-Based Algorithms for Image Segmentation

Image segmentation is a quite direct application of CVT clustering. The following algorithm can be used to construct CVT clustering; see, e.g., [17], [36], and [37] for details.

Algorithm CVT-A. Given a positive integer L and a digital image $\cup = \{u(i,j)\}_{(i,j)\in D}$, choose arbitrarily L color values $\{w_\ell\}_{\ell=1}^L$.

1) Determine the Voronoi clusters $\{V_\ell\}_{\ell=1}^L$ of $\cup$ associated with $\{w_\ell\}_{\ell=1}^L$.
2) For each cluster V_ℓ, $\ell = 1,\ldots,L$, determine the cluster means $\overline{w}_\ell$.
3) If $\overline{w}_\ell$ and w_ℓ are the same, return $\left(\{w_\ell\}_{\ell=1}^L; \{V_\ell\}_{\ell=1}^L\right)$ and exit; otherwise, set $w_\ell = \overline{w}_\ell$ for $\ell = 1,\ldots L$ and return to Step 1.

Algorithm CVT-A is also called the Lloyd algorithm [26]. It is easy to see that steps 1 and 2 will result in a strict decrease in the VT energy unless a local minimizer is reached. Thus, it is guaranteed that the algorithm will converge due to the compactness of the VT energy. The complexity of Algorithm CVT-A is, in general, $\mathcal{O}(k \times L \times I \times J)$ where k denotes the total number of iterations. The initial set of color values should be chosen so that none of the associated Voronoi clusters are empty. Since a digital image is a finite data set, the algorithm terminates in a finite number of steps. However, it is often the case that a very good approximation to the final CVT configuration can be obtained in substantially fewer steps. For this reason, at each iteration, one should calculate the VT energy of the current configuration and terminate when that it is within some prescribed tolerance compared with that of the previous configuration; In our segmentation experiments, we terminate the iteration when the change in the energy is less than $L\%$ of the value of the current energy.

Algorithm CVT-A does not transfer elements of $\cup$ from one cluster to another until the end of each iteration, i.e., it does not account for the change in the cluster means until all means are computed. The following algorithm is an accelerated version of Algorithm CVT-A that takes into account changes in cluster means as soon as they are determined.

Algorithm CVT-B. Given a positive integer L and a digital image $\cup = \{u(i,j)\}_{(i,j)\in D}$, choose arbitrarily L color values $\{w_\ell\}_{\ell=1}^L$ and determine the associated Voronoi clustering $\{V_\ell\}_{\ell=1}^L$.

1) For every $u(i,j) \in \cup$,
   a) evaluate the VT energy for all possible transfers of $u(i,j)$ from its current cluster V_ℓ to any of the other clusters V_k, $k = 1,\ldots,L$, $k \neq \ell$;
   b) if moving $u(i,j)$ from its current cluster V_ℓ to the cluster V_m decreases the VT energy the most, then
      i) transfer $u(i,j)$ from cluster V_ℓ to cluster V_m;
      ii) replace w_ℓ and w_m by the means of the newly modified clusters V_ℓ and V_m, respectively.
2) If no transfers occurs, return $\left(\{w_\ell\}_{\ell=1}^L; \{V_\ell\}_{\ell=1}^L\right)$ and exit; otherwise, go to Step 1.

Algorithms CVT-A and CVT-B result in a k-means clustering (in color space) of the digital image. Both are guaranteed to reduce the energy after every iteration, and finally converge to a minimizer of the VT energy. Numerical experiments indicate that Algorithm CVT-B is often more reliable than Algorithm CVT-A [12] even though each iteration of the former is more costly since one must determine the effect each potential transfer has on the energy. The gain lies in the fact that an iteration of Algorithm CVT-B leads to a larger decrease in the energy than does an iteration of Algorithm CVT-A, and thus, a much smaller number of iterations are required for Algorithm CVT-B for convergence. A hybrid approach is also quite feasible in which one starts with the Algorithm CVT-A and then switches to Algorithm CVT-B. Presumably, after several iterations of Algorithm CVT-A, only a very few of the more expensive iterations of Algorithm CVT-B are needed to obtain accurate results.

The costs of Algorithms CVT-A and CVT-B may be reduced at the price of increased storage. One recognizes that points that are close to their current cluster means will likely not be transferred to another cluster. Thus, for Algorithm CVT-A, we do not need to consider every point $u(i,j)$ when we determine a new clustering; we only need to consider those points whose

distances to their old cluster mean is larger than the average distance of all points in its cluster to the cluster mean. Similarly, for Algorithm CVT-B, we do not need to loop over every point $u(i,j) \in \cup$; we only need to loop over points whose distances to their current cluster mean are larger than the average distance of all points in its cluster to the cluster mean. This modification requires one to determine and store the average distance of each cluster. However, such a strategy will roughly halve the cost of each iteration of Algorithm CVT-B. It is obvious that a suitable data structure that takes into account such stored information with a quick searching algorithm could lead to a much more efficient implementation [25].

Another improvement to Algorithm CVT-B is not to compare reductions in the VT energy for possible transfers to far away clusters. Thus, in Step 1(a) of Algorithm CVT-B, one would only consider clusters with means having a distance to the mean of the current cluster for the point $u(i,j)$ that is less than twice of the distance from the point $u(i,j)$ itself to its own cluster mean. Implementing this strategy requires the computation and storing of the distances between cluster means, but it is very useful in the case that the number of clusters L is very large since the cost decreases by almost a factor of $O(L^{-1})$.

We also note that Algorithm CVT-A is easier to parallelize. For more discussions concerning algorithms for computing CVTs including their parallelization, we refer to [12], [13], and [22]. Once the CVT clusters are determined, the boundaries could be easily detected. Let us define the *edges* to be the boundary points (pixels) in the physical space of the images between different clusters, i.e., $(i,j) \in D_l$ is an edge point if one of its neighboring points belongs to a different segment D_k, i.e., $k \neq l$. Equivalently, (i,j) in the segment D_l is an edge point if $(i^*,j^*) \notin D_l$ for some (i^*,j^*) which is a neighbor of (i,j).

C. Deficiencies in the Basic CVT-Based Segmentation

The CVT-based segmentation has been applied successfully to images with homogeneous regions, each having a distinct centroid, i.e., a mean intensity value of the associated CVT cluster. From Section II-A, one sees that the CVT-based algorithms use global intensity information while local information is often ignored. The resulting segmentation may give us too many details and sometimes those extra details may be just noises [12]. Even worse, the objects may be too weak to be detected by the CVT-based algorithm since they might be overwhelmed by unwanted details.

In Fig. 1, the original image describes a simulated noisy minefield. After it is divided into two clusters by the CVT-based segmentation (Algorithm CVT-A or CVT-B), one still can not tell where the minefield is, i.e., the CVT-based segmentation fail to accurately identify which objects are mines and which are not. In Fig. 2, the CVT-based segmentation does a good job of identifying the lights, but one fails to recognize the boundary between the European landmass and the Atlantic ocean. These deficiencies stimulate us to look for new ideas to make CVT-based segmentation more accurate and robust. Let us carefully examine Figs. 1 and 2. Both images are divided into two segments by the CVT-based clustering in the color space, however, these two clusters are broken into pieces in the physical space of the images, i.e., the basic CVT-based segmentation gives us *too many*

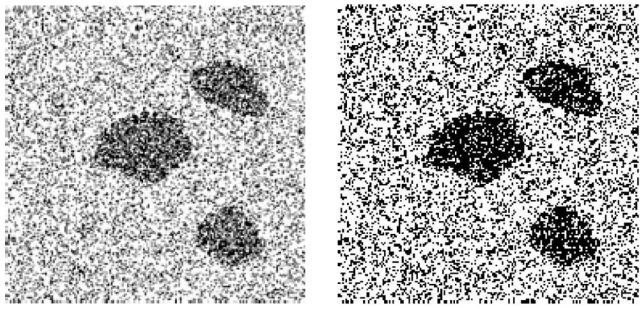

Fig. 1. Left: Original image of a simulated noisy minefield. Right: CVT-based segmentation into two clusters.

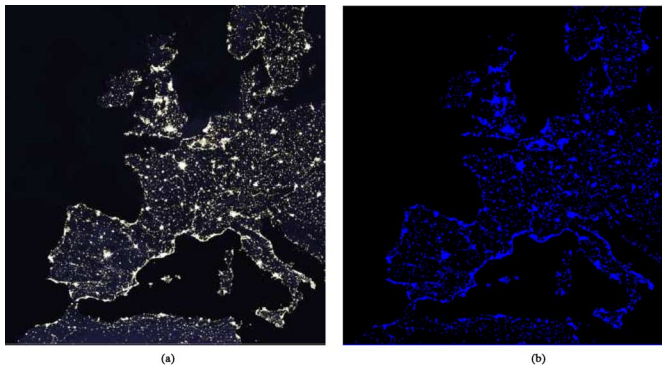

Fig. 2. Left: Original image of "Europe-at-night". Right: CVT-based segmentation into two clusters.

edges. From this observation, we may think that appropriately employing the information of the edges in the physical space into the CVT model will be useful in controlling the accuracy of segmentation. In Section III, we will introduce an *edge related energy* into the classical clustering energy to derive our new model.

III. Edge-Weighted Centroidal Voronoi Tessellation Model and Its Implementation Algorithms

A. Edge Related Energy

Let us first define a new energy relevant to the edges for a given clustering $(\mathcal{W};\mathcal{U})$. One may find several ways to define this energy. For example, in the most natural way, we may define it as the total number of the edge points, i.e.,

$$\mathcal{L} = \sum_{i=1}^{I} \sum_{j=1}^{J} \mathcal{X}(i,j). \tag{10}$$

where $\mathcal{X}$ is certain characteristic function for the edge points defined on D depending on $(\mathcal{W};\mathcal{U})$.

Here we take a more general approach and define the edge related energy as the sum of the number of edge points within a predefined neighborhood of every pixel. To be rigorous, for each pixel $(i,j) \in D$, denote by $\mathbb{N}_\omega(i,j)$ a local neighborhood for it, which can be a $\omega \times \omega$ square centered at (i,j) or a disk centered at (i,j) with radius ω. Let us call any $(i',j') \in \mathbb{N}_\omega(i,j)$ a

neighbor point (or neighbor pixel) of (i,j). For each $(i,j) \in D$, define

$$\mathcal{F}(i,j) = \sum_{(i',j') \in \mathbb{N}_\omega(i,j)} \mathcal{X}_{(i,j)}(i',j') \tag{11}$$

and then the total edge energy is given by

$$F = \sum_{(i,j) \in D} \mathcal{F}(i,j). \tag{12}$$

It is easy to see that $\mathcal{F}$ is nonnegative and so does F. In general, we can expect lager value of F means longer length of the edges (i.e., more edge points), and vice versa.

Changing in the formula of $\mathcal{F}(i,j)$ in (11) can result in different form of the edge energy. In this paper, we define $\mathcal{F}(i,j)$ to be the number of points in $\mathbb{N}_\omega(i,j)$ which does not belong to the same cluster as that of the point (i,j). In other words, for each $(i,j) \in D$, we define a local characteristic function $\mathcal{X}_{(i,j)}$: $\mathbb{N}_\omega(i,j) \to \{0,1\}$ as

$$\mathcal{X}_{(i,j)}(i',j') = \begin{cases} 1, & \text{if } \pi_{\mathcal{U}}(i',j') \neq \pi_{\mathcal{U}}(i,j) \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Note that D_l denotes the segments in the physical space of the image associated with $\mathcal{U}$. Now let us replace (11) by

$$\mathcal{E}_{\mathcal{L}}(i,j) = \lambda \sum_{(i',j') \in \mathbb{N}_\omega(i,j)} \mathcal{X}_{(i,j)}(i',j') \tag{14}$$

where λ is a positive weighting factor and then the total edge energy is given by

$$E_{\mathcal{L}}(\mathcal{D}) = \sum_{(i,j) \in D} \mathcal{E}_{\mathcal{L}}(i,j). \tag{15}$$

Note that the edge energy $E_{\mathcal{L}}$ only depends on the clusters $\{D_l\}_{l=1}^{L}$ in the physical space. If the neighbor (i',j') of (i,j) belongs to a different cluster compared with that of (i,j), we name such a neighbor pixel as an *local edge point* with regard to (i,j). Thus, $\mathcal{E}_{\mathcal{L}}(i,j)$ actually is the weighted number of local edge points with regard to the pixel (i,j). We note that a local edge point associated with the pixel (i,j) may not be a true edge point at all.

Together with the classical clustering energy (7), we define the following *edge-weighted clustering energy* as follows:

$$\begin{aligned} \widehat{E}(\mathcal{W};\mathcal{D}) &= E(\mathcal{W};\mathcal{D}) + E_{\mathcal{L}}(\mathcal{D}) \\ &= \sum_{(i,j) \in D} [\mathcal{E}(i,j) + \mathcal{E}_{\mathcal{L}}(i,j)] \\ &= \sum_{l=1}^{L} \sum_{(i,j) \in D_l} |u(i,j) - w_l|^2 \\ &\quad + \lambda \sum_{(i,j) \in D} \sum_{(i',j') \in \mathbb{N}_\omega(i,j)} \mathcal{X}_{(i,j)}(i',j'). \end{aligned} \tag{16}$$

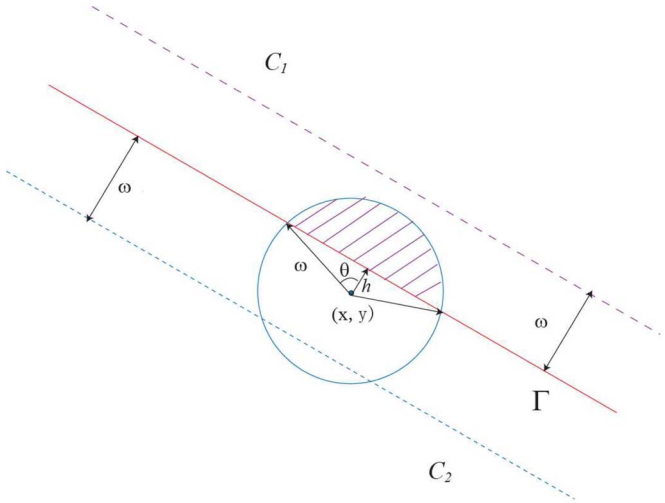

Fig. 3. Purple shadow area is the set of local edge points associated with (x,y) defined in (14). Point (x,y) is the center of the circle.

The second term, the edge energy will play a crucial role in our new model. Our goal is to find a clustering $(\mathcal{W};\mathcal{D})$ that minimizes the edge-weighted clustering energy (16). By carefully examining (14), one may notice that the variation of edge energy due to the transfer of pixel (i,j) will fall into two parts: one is from the edge energy variation of pixel (i,j) itself and the other is from the edge energy variation of pixels inside $\mathbb{N}_\omega(i,j)$.

B. Relation Between the Edge Energy and the Boundary Length

All definitions and discussions above are for the discrete case: $D = \{(i,j) : i = 1,\ldots,I,\ j = 1,\ldots,J\}$ is a discretization of the image domain. Now let us assume that the image u is a continuous function over the bounded domain $\Omega \in \mathbb{R}^2$. We will show that there exists an asymptotic relationship between the edge energy and the boundary length in the continuous case.

Assume that a smooth curve Γ divides Ω into two clusters C_1 and C_2 (in the physical space). We let $\mathbb{N}_\omega(x,y)$ be a disk centered at $(x,y) \in \Omega$ with radius ω. Under these notations, the edge energy (14) of the point (x,y) is

$$\mathcal{E}^c_{\mathcal{L}}(x,y) = \lambda \in t_{\mathbb{N}_\omega(x,y)} \mathcal{X}_{(x,y)}(x',y')dx'dy' \tag{17}$$

and the total edge energy (15) in the continuous case is given by

$$\begin{aligned} E^c_{\mathcal{L}}(\mathcal{D}) &= \in t_\Omega \mathcal{E}^c_{\mathcal{L}}(x,y)dxdy \\ &= \lambda \in t_\Omega \in t_{\mathbb{N}_\omega(x,y)} \mathcal{X}_{(x,y)}(x',y')dx'dy'. \end{aligned} \tag{18}$$

Without loss of generality, let us assume $(x,y) \in C_2$. Then (17) gives the weighted area of $\mathbb{N}_\omega(x,y) \cap C_1$. Assume that ω is small enough, then the intersection of Γ and $\mathbb{N}_\omega(x,y)$ can be viewed as a straight line segment (Fig. 3). The edge energy of the point (x,y) is equal to the weighted area of $\mathbb{N}_\omega(x,y) \cap C_1$,

i.e., the purple shadow area in Fig. 3 multiplied by λ, which can be written as

$$\begin{aligned}\mathcal{E}^c_{\mathcal{L}}(x,y) &= \lambda(\omega^2\theta - \omega^2\sin\theta\cos\theta)\\ &= \lambda(\theta - \sin\theta\cos\theta)\omega^2. \end{aligned} \tag{19}$$

Note that $h = \omega\cos\theta$ denotes the distance from (x,y) to C_1.

Now we can rewrite the edge energy as

$$\begin{aligned}E^c_{\mathcal{L}}(\mathcal{D}) &= \in t_\Omega \mathcal{E}^c_{\mathcal{L}}(x,y)dxdy\\ &= 2\lambda \in t_0^{\omega} \in t_\Gamma(\theta - \sin\theta\cos\theta)\omega^2 dldh\\ &= 2\lambda \in t_0^{\pi/2} \in t_\Gamma(\theta - \sin\theta\cos\theta)\omega^2(\omega\sin\theta)dld\theta\\ &= 2\lambda\omega^3 \in t_0^{\pi/2}(\theta\sin\theta - \sin^2\theta\cos\theta)d\theta \in t_\Gamma dl\\ &= \frac{4}{3}\lambda\omega^3 H \end{aligned} \tag{20}$$

where $H = \in t_\Gamma dl$ is the length of the boundary curve Γ. Therefore, we obtain a proportional relation between the total edge energy and the length of the boundary.

For simplicity, our proof is given only for the case of two clusters. For the cases with more than two clusters, the proof can be generalized correspondingly and the relation (20) still holds. The difference only occurs around the intersections of more that two boundary curves, but for a sufficient small ω, the effect of those intersections can be ignored compared to the total length H. We also note that when $\mathbb{N}_\omega(x,y)$ is a $\omega\times\omega$ square centered at (x,y), then we will have a similar relation such as

$$E^c_{\mathcal{L}}(\mathcal{W};\mathcal{D}) = \alpha_\Gamma\lambda\omega^3 H$$

where α_Γ is a constant depending on the position and shape of the curve Γ.

The relation (20) indicates that the relationship between the weight λ and the radius of the neighborhood ω is

$$\lambda \sim \frac{1}{\omega^3}$$

when the total edge energy is fixed.

C. Edge-Weighted Distance

Now we construct an edge-weighted distance function to measure the distance from a pixel to a generator. We want to guarantee that moving a pixel to the associated cluster of the generator to which it has the shortest edge-weighted distance decreases the total edge-weighted clustering energy the most.

Let us rewrite (16) as

$$\begin{aligned}\widehat{E}(\mathcal{W};\mathcal{D}) = &\left[\sum_{(i',j')\in D\setminus(i,j)} |u(i',j') - w_{\pi_{\mathcal{U}}(i',j')}|^2\right]\\ &+ |u(i,j) - w_{\pi_{\mathcal{U}}(i,j)}|^2\\ &+ \left[\sum_{(i',j')\in D\setminus(i,j)} \mathcal{E}_{\mathcal{L}}(i',j')\right]\\ &+ \mathcal{E}_{\mathcal{L}}(i,j). \end{aligned} \tag{21}$$

Now we need to determine variation of the total edge-weighted clustering energy when we transfer a pixel (i,j) from its current cluster D_l to another cluster D_m. Obviously, the first term in the right-hand side of (21) has no change. The change of the second term is given by

$$|u(i,j) - w_m|^2 - |u(i,j) - w_l|^2. \tag{22}$$

Now, let us denote by $n_k(i,j)$ the number of pixels within $D_k \cap \mathbb{N}_\omega(i,j)\setminus(i,j)$. Before we move the pixel (i,j), its edge energy is

$$\begin{aligned}\mathcal{E}_{\mathcal{L}}(i,j) &= \lambda \sum_{(i',j')\in\mathbb{N}_\omega(i,j)} \mathcal{X}_{(i,j)}(i',j')\\ &= \lambda \sum_{k\in\{1,\ldots,L\}\setminus\{l\}} n_k(i,j). \end{aligned} \tag{23}$$

Similarly, after we transfer (i,j) to cluster D_m, its edge energy changes to

$$\begin{aligned}\mathcal{E}_{\mathcal{L}}(i,j) &= \lambda \sum_{(i',j')\in\mathbb{N}_w(i,j)} \mathcal{X}_{(i,j)}(i',j')\\ &= \lambda \sum_{k\in\{1,\ldots,L\}\setminus\{m\}} n_k(i,j). \end{aligned} \tag{24}$$

Clearly, the change in the fourth term $\mathcal{E}_{\mathcal{L}}(i,j)$ after the transferring is

$$\begin{aligned}&\lambda \sum_{k\in\{1,\ldots,L\}\setminus\{m\}} n_k(i,j) - \lambda \sum_{k\in\{1,\ldots,L\}\setminus\{l\}} n_k(i,j)\\ &\qquad = \lambda\left[n_l(i,j) - n_m(i,j)\right]. \end{aligned} \tag{25}$$

Now let us consider the change of $\mathcal{E}_{\mathcal{L}}(i',j')$ due to the moving of the pixel (i,j) from cluster D_l to D_m. If pixel (i',j') is outside of $\mathbb{N}_\omega(i,j)$, the moving of (i,j) will not affect this term. Suppose that $(i',j') \in \mathbb{N}_w(i,j)\cap D_p$. If $p \neq l,m$, obviously there is no change in the edge energy $\mathcal{E}_{\mathcal{L}}(i',j')$. If $p = l$, moving (i,j) out of cluster D_l will increase the edge energy by λ. If $p = m$, moving (i,j) into the cluster D_m will decrease the edge energy by λ. Therefore, the change of $\sum_{(i',j')\in D\setminus(i,j)} \mathcal{E}_{\mathcal{L}}(i',j')$ is

$$\lambda n_l(i,j) - \lambda n_m(i,j) = \lambda[n_l(i,j) - n_m(i,j)]. \tag{26}$$

Summing (22), (25), and (26) together, we then obtain the variation of the total edge energy $\widehat{\mathcal{E}}(\mathcal{W})$ due to the transferring of the pixel (i,j) from D_l to D_m as follows:

$$|u(i,j) - w_m|^2 - |u(i,j) - w_l|^2 + 2\lambda(n_l(i,j) - n_m(i,j))$$

which can be rewritten in a more symmetrical way as

$$\begin{aligned}&\left(|u(i,j) - w_m|^2 - 2\lambda n_m(i,j)\right)\\ &\qquad - \left(|u(i,j) - w_l|^2 - 2\lambda n_l(i,j)\right). \end{aligned} \tag{27}$$

Now let us define the *edge-weighted distance* from a pixel (i,j) to a generator w_k as

$$\mathrm{dist}((i,j),w_k) = \sqrt{|u(i,j) - w_k|^2 + 2\lambda\tilde{n}_k(i,j)} \tag{28}$$

where $\tilde{n}_k(i,j) = |\mathbb{N}_\omega(i,j)| - n_k(i,j) - 1$, the number of pixels within $\mathbb{N}_\omega(i,j) \setminus (D_k \cup (i,j))$. Note that this distance beautifully combines the color information together with the physical information of the pixel (i,j).

In conclusion, moving a pixel to the cluster of a generator to which it has the shortest edge-weighted distance defined by (28) will decrease the edge-weighted clustering energy $\widehat{E}(\mathcal{W};\mathcal{D})$ the most.

D. Edge-Weighted Voronoi Regions

Given a set of generators $\mathcal{W} = \{w_l\}_{l=1}^L$ in the color space, we define the the edge-weighted voronoi regions $\widetilde{\mathcal{D}} = \{\widetilde{D}_l\}_{l=1}^L$ in the physical space D by

$$\begin{aligned}\widetilde{D}_l = \{&(i,j) \in D : \mathrm{dist}((i,j), w_l) \le \mathrm{dist}((i,j), w_k) \\ &\text{for } l = 1,\ldots,L\} \\ k = 1,&\ldots,L. \end{aligned} \tag{29}$$

If the generators $\mathcal{W}$ are fixed, then it is obvious that the edge-weighted Voronoi tessellation $\{\widetilde{D}_l\}_{l=1}^L$ corresponds to the minimizer of the edge-weighted energy $\widehat{E}(\mathcal{W},\mathcal{D})$, i.e.,

$$\widetilde{\mathcal{D}} = \mathrm{argmin}_{\mathcal{D}}\, \widehat{E}(\mathcal{W};\mathcal{D}).$$

We then define the EWVT energy for a given set of generators $\mathcal{W} = \{w_l\}_{l=1}^L$ by

$$\widehat{E}_{EWVT}(\mathcal{W}) = \widehat{E}(\mathcal{W};\widetilde{\mathcal{D}}). \tag{30}$$

where $\widetilde{\mathcal{D}}$ are the edge-weighted Voronoi regions associated with $\mathcal{W}$. The following algorithm can be used to efficiently construct the edge-weighted Voronoi regions.

Algorithm EWVT. Given a set of generators $\mathcal{W} = \{w_l\}_{l=1}^L$. Choose an arbitrarily partition $\{\widetilde{D}_l\}_{l=1}^L$ in the physical space of the digital image $\mathbb{U} = \{u(i,j)\}_{(i,j)\in D}$.

---

1) For every $(i,j) \in D$,
   a) calculate and compare the edge-weighted distance defined in **(28)** from the point (i,j) to all of the generators $\{w_l\}_{l=1}^L$;
   b) move the point (i,j) to the cluster whose generator has the shortest edge-weighted distance to it. (Note that this also equivalently results in an update of the partition $\{\widetilde{D}_l\}_{l=1}^L$).
2) If no point is moved, return $\{\widetilde{D}_l\}_{l=1}^L$ and exit the loop; otherwise, go to Step 1.

It is obvious that Algorithm EWVT leads to strict decrease in the EWVT energy until an edge-weighted Voronoi tessellation of D is found. The algorithm terminates in a finite number of steps because the digital image is a finite data set.

E. Edge-Weighted Centroidal Voronoi Tessellations: Model and Algorithms

Algorithm EWVT determines the edge-weighted Voronoi regions $\widetilde{\mathcal{D}} = \{\widetilde{D}_l\}_{l=1}^L$ for a given set of generators. By the spirit of CVTs, we also need a way to determine the corresponding edge-weighted centroids of $\{\widetilde{D}_l\}_{l=1}^L$, i.e., the minimizer of the edge-weighted energy $\widehat{E}(*,\widetilde{\mathcal{D}})$ ($\widetilde{\mathcal{D}}$ are now fixed). In fact, according to (16), it is easy to see that the edge-weighted centroids are exactly the standard CVT centroids of $\{\widetilde{D}_l\}_{l=1}^L$ since the edge energy $\mathcal{E}_{\mathcal{L}}(i,j)$ at each pixel (i,j) is fixed. Explicitly, the edge-weighted centroid for each $\widetilde{D}_l$ is given by

$$\tilde{w}_l = \frac{1}{|\widetilde{D}_l|} \sum_{(i,j)\in\widetilde{D}_l} u(i,j) \tag{31}$$

where $|\widetilde{D}_l|$ is the number of pixels in $\widetilde{D}_l$.

Definition 2: For a given edge-weighted Voronoi tessellation $\left(\{\tilde{w}_l\}_{l=1}^L; \{\widetilde{D}_l\}_{l=1}^L\right)$ of D, we call it an edge-weighted centroidal Voronoi tessellation (EWCVT) of D if the generators $\{w_l\}_{l=1}^L$ are also the corresponding centroids of the associated edge-weighted Voronoi regions $\{\widetilde{D}_l\}_{l=1}^L$.

Then it is easy to prove the following result about EWCVT from the above analysis.

Theorem 2: The edge-weighted energy $\widehat{E}(\mathcal{W};\widetilde{\mathcal{D}})$ is minimized only if $(\mathcal{W};\widetilde{\mathcal{D}})$ form a EWCVT of D, i.e., $\widetilde{\mathcal{D}}$ are edge-weighted Voronoi regions of D associated with the generators $\mathcal{W}$ and simultaneously $\mathcal{W}$ are the corresponding centroids of the regions $\widetilde{\mathcal{D}}$.

The following algorithm can be used to apply the EWCVT model to the application of image segmentation.

Algorithm EWCVT-A. Given an integer L and choose arbitrarily a partition $\{\widetilde{D}_l\}_{l=1}^L$ of the digital image $\mathbb{U} = \{u(i,j)\}_{(i,j)\in D}$.

---

1) For each cluster $\widetilde{D}_l$, $l = 1,\ldots,L$, determine its cluster centroid $\tilde{w}_l$ by (31).
2) Take $\{\tilde{w}\}_{l=1}^L$ as the generators, determine the corresponding edge-weighted Voronoi clustering $\widetilde{\mathcal{D}}' = \{\widetilde{D}_l'\}_{l=1}^L$ according to Algorithm EWVT.
3) If the edge-weighted Voronoi clustering $\{\widetilde{D}_l'\}_{l=1}^L$ and $\{\widetilde{D}_l\}_{l=1}^L$ are the same, return $\left(\{w_l\}_{l=1}^L; \{\widetilde{D}_l\}_{l=1}^L\right)$ and exit the loop; otherwise, set $\widetilde{D}_l = \widetilde{D}_l'$ for $l = 1,\ldots,L$ and go to Step 1.

Note that steps 1 and 2 will result in a strict decrease in the EWVT energy until a local minimizer is reached. Thus, it is guaranteed that the algorithm will converge a minimizer of the EWVT energy and terminate in a finite number of steps. The initial set of clusters should be chosen such that none of the clusters $\{\widetilde{D}_l\}_{l=1}^L$ is empty.

In step 2 of Algorithm EWCVT-A, the point transferring between clusters is made immediately but the generators will not be updated until the loop goes back to step 1. Allowing update of the generators by the centroids of the clusters after every pixel transferring again can result in the following accelerated version of Algorithm EWCVT-A, where the configuration will be changed immediately after a point is determined to transfer from its current cluster to another one. In other words, the point (pixel) will be transferred in the current iteration step and the cluster centroid will be re-calculated before dealing with the next point of the data set.

Algorithm EWCVT-B. Given an integer L and choose arbitrarily a partition $\{\widetilde{D}_l\}_{l=1}^{L}$ of digital image $\mathbb{U} = \{u(i,j)\}_{(i,j)\in D}$. Determine the centroids $\{w_l\}_{l=1}^{L}$ of $\{\widetilde{D}_l\}_{l=1}^{L}$ and take them as the generators.

1) For every $(i,j) \in \mathcal{D}$:
   a) calculate and compare the edge weighted distance defined in **(28)** from the point (i,j) to all the generators $\{w_l\}_{l=1}^{L}$;
   b) transfer the point (i,j) to the cluster whose generator has the shortest distance to it, say, from $\widetilde{D}_l$ to $\widetilde{D}_m$;
   c) replace w_l and w_m with the centroids of the newly modified clusters $\widetilde{D}_l$ and $\widetilde{D}_m$ respectively.
2) if no point is moved, return $\left(\{w_l\}_{l=1}^{L}; \{\widetilde{D}_l\}_{l=1}^{L}\right)$ and exit the loop; otherwise, go to Step 1.

Both Algorithms EWCVT-A and EWCVT-B will definitely converge to a local minimizer of the EWVT energy. It is also clear that Algorithms EWCVT-A and EWCVT-B will reduce to the CVT-based algorithms CVT-A and CVT-B respectively when the weight $\lambda = 0$ or the size of the neighborhood $\omega = 1$, i.e., no other points in a point's neighborhood except itself. The nature of Algorithm EWCVT-B, i.e., real time modification of the configuration in each iteration, leads to a larger energy decrease each step than that of Algorithm EWCVT-A as usual, and thus Algorithm EWCVT-B is often faster. The complexity of Algorithm EWCVT-B is $\mathcal{O}(k \times \omega^2 \times L \times I \times J)$ where k denotes the number of iterations, so it is more expensive than the CVT-based algorithms. Note that, in most practical cases, we again do not need to wait until the condition to exit the loops is strictly satisfied for in both algorithms. In fact, we will calculate and record the EWVT energy of the current configuration at each iteration. If the decrease of the EWVT energy is within some prescribed tolerance, the algorithm will be terminated. In our numerical experiments, we stop the iteration [12] if

$$\frac{|\widehat{E}_{i+1} - \widehat{E}_i|}{\widehat{E}_i} < L\%. \tag{32}$$

Adjusting the prescribed tolerance of the energy may result in slightly different final EWCVT configuration. The EWCVT-based algorithms provide us an effective way to control the segmentation accuracy.

There are a lot of accelerating strategies for determining CVTs [12], [25], [32]. Most of them are also applicable to our EWCVT model. We just need to replace the VT energy with the EWVT energy in any of the accelerated strategies. Presumably, the points close the segment boundary are most likely to be moved in each iteration step. Thus, we can only consider the moving of edge points at each iteration in Algorithms EWCVT-A and EWCVT-B. In practice, this scheme has been proven to be a satisfied strategy for acceleration purpose. Certainly, one also can implement a hybrid approach in which one starts with Algorithm EWCVT-A or EWCVT-B by evaluating all of the points and then switches to the scheme of evaluating the edge points only. Another possible improvement may be first carrying out the basic CVT clustering (Algorithm CVT-A or CVT-B) and then switching to the EWCVT-based algorithms with the CVT clusters as the initial guess since the CVT clustering can quickly find a very good initial configuration for the EWCVT-based algorithms (this approach is used in most of our numerical examples).

IV. Experiments and Discussions

In this section, we will discuss the selection of parameters in our EWCVT model for image segmentation and present results obtained by applying our EWCVT-based segmentation algorithms to various synthetic and real images from different modalities.

A. Principles of Determining Parameters

There are two parameters in our EWCVT model: λ as the weighting factor used to balance the classical clustering energy in the color space and the edge energy involving the physical space, and ω as the size of the neighborhood for each pixel used to control local details of the image. The final segmentation results are effected by both parameters, i.e., λ and ω determine the segmentation accuracy of the proposed method in some extend. Therefore, we need to carefully select these parameters in order to obtain satisfactory segmentation results for given images. Some principles of determining the parameters are discussed below.

The EWCVTs are constructed through evaluating the EWVT energy of each point/pixel and minimizing the total sum. Note that the EWVT energy of each point consists of two parts: one is the classical clustering energy $E(\mathcal{W};\mathcal{D})$ and the other is the edge energy $E_{\mathcal{L}}(\mathcal{D})$. In order to avoid any one of these two energies dominates during the construction process, we need to make sure that they are almost kept in the same order of magnitude, i.e., the weighting factor λ should be chosen such that

$$\frac{\max\{E(\mathcal{W};\mathcal{D}), E_{\mathcal{L}}(\mathcal{D})\}}{\min\{E(\mathcal{W};\mathcal{D}), E_{\mathcal{L}}(\mathcal{D})\}} < C \tag{33}$$

holds for some positive constant C. Without explicit declaration, λ in our experiments is always chosen to satisfy (33) with $C = 10$.

In our experiments, we take $\mathbb{N}_\omega(i,j)$ to be a $\omega \times \omega$ square centered at (i,j) (One may choose it to be a circle with radius ω too). The segmentation results are clearly dependent on the values of ω. Generally speaking, a larger ω can eliminate more noises, while a smaller ω can be more accurate in finding details of the image. If the noises take the form of a bunch of unconnected blocks in the physical domain, our experiments show that all noise blocks whose diameters are less than ω will be eliminated after the segmentation. Thus, the selection of ω often depends on the scale of the noises of the given image.

The principles presented here are guidelines for choosing parameters to obtain satisfactory segmentation results in most situations. Certainly, sometimes one may want to set the parameters to meet some other requirements for certain purposes.

B. Illustration of the Effect of Parameters and Initial Clusters

In our numerical experiments, most of the examples presented below only take a few iterations (for the EWCVT-based algorithms to stop) to obtain the final segmented images due to the use of CVT clusters as the initial configuration

for the EWCVT-based algorithms. For these examples, we will provide all of the intermediate images obtained during the segmentation process. For other examples which require much more iterations due to a simple initial configuration, only some typical intermediate images will be presented. For all example in this paper, the first image presented will be the original image and the second and last one will be the image of the initial clusters and the final segmentation of the EWCVT-based algorithms, respectively. We also note that the outcomes of Algorithms EWCVT-A and EWCVT-B are essentially the same (maybe a very slight difference) since they are just different implementation schemes for the construction of EWCVT of the same image. Thus, we only test Algorithm EWCVT-B since it it generally more efficient than Algorithm EWCVT-A as discussed in Section III-E). Information about the parameters ω and λ, image size, number of iterations, and CPU time will be listed in the caption of each example figure. The experiments are performed on a Compaq Presario PC with Core Duo 1.66-GHz processor and 1.0-GB RAM.

Fig. 4 shows the segmentation result of Algorithm EWCVT-B applied to a heavily noised synthetic image. The original image consists of three separated geometric objects in a noisy background. In order to show the difference in segmentation between the CVT-based model and the EWCVT model, in this example, we first carry out a CVT-based segmentation algorithm (Algorithm CVT-B) to the original image to get a CVT clustering with two clusters (see the first image of the second row in Fig. 4), then the CVT clusters are used as the initial configuration for Algorithm EWCVT-B to compute the EWCVT segmentation. Totally only 2 iterations are needed for Algorithm EWCVT-B to stop in this example. It is clear that the EWCVT model does generate better segmentation than the basic CVT model in this example, i.e., the boundary between segments is smoother and Almost all background noises are removed. We also plot the edge-weighted clustering energy $\widehat{E}(\mathcal{W}, \widetilde{\mathcal{D}})$ (i.e., the EWVT energy $\widehat{E}_{EWVT}(\mathcal{W})$), the classical clustering energy $E(\mathcal{W}, \widetilde{\mathcal{D}})$ and the edge energy $E_{\mathcal{L}}(\mathcal{W}, \widetilde{\mathcal{D}})$ respectively in Fig. 4 (bottom row). From the energy plot, we observe that:

1) As we expect, the edge-weighted clustering energy (16) keeps decreasing during the iterations of Algorithm EWCVT-B. This shows that our algorithm is truly a monotonic energy minimization process with respect to EWVT energy.
2) The classical clustering energy keeps increasing during the iteration. As stated above, we take the CVT clustering as the initial clusters. Note that the clustering energy $E(\mathcal{W}, \mathcal{D})$ (i.e., the VT energy $E_{VT}(\mathcal{W})$) is minimized by the CVT-based algorithms. Thus, the clustering energy of the initial guess is the minimum among all iterations. Our algorithm decreases the edge-weighted clustering energy but the classical clustering energy may increase along the iterations.
3) The edge energy keeps decreasing. Observing that in the first image of the second row, there are a lot of edge points since the CVT clusters give us a bunch of unconnected noise blocks. During the construction process of the EWCVT, most of the noise blocks are eliminated; thus, the number of edge points reduces significantly.
4) In addition, the decrease of the edge energy bypasses the increase of the clustering energy; thus, the total edge-weighted clustering energy decreases.

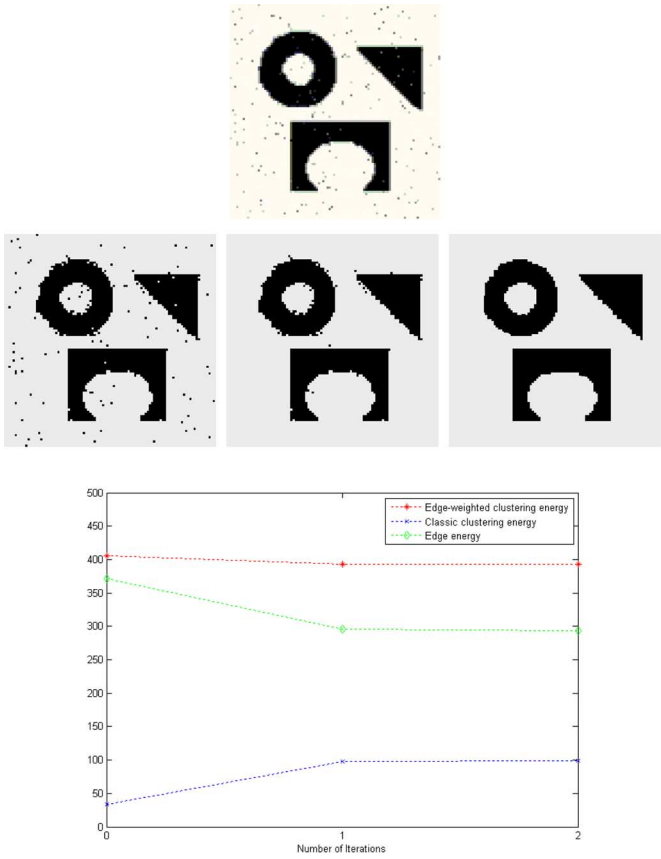

Fig. 4. Detection of three geometric objects from a noisy background using Algorithm EWCVT-B with the CVT clustering (2 clusters) as the initial configuration; image size $= 100 \times 100, \omega = 5, \lambda = 0.02$, 2 iterations, CPU time $= 0.89$ s. Top row: Original image; second row: results after 0, 1, and 2 iterations; bottom row: plot of variations of the energies.

Our EWCVT model and algorithms are also quite robust in removing the noises. The original image in Fig. 5 is a similar but much more heavily noised synthetic image compared with that of Fig. 4. Algorithm EWCVT-B still works very well for this example and the computational cost only increase by 1 iteration (CPU time increases by 0.43 s). The variations of the energies again have similar patterns as that of the previous example.

The effect of the parameter λ and the role of the edge energy in the EWCVT model are well illustrated in Figs. 6–8 where Algorithm EWCVT-B is applied to the image of three separated balls in a homogeneous background, with different values of the parameter $\lambda = 0.01, 0.05, 0.2$. If λ is smaller, then more details of boundary between segments the EWCVT model can be detected, and the segmentation result is closer to that obtained by the basic CVT model; if λ is larger, then the EWCVT model will more likely group the objects in the image together, and, thus, the boundary of final segmentation is shorter and smoother. It is easy to see verify these points from what happens in Figs. 6–8. Equation (14) implies that the edge energy is proportional to

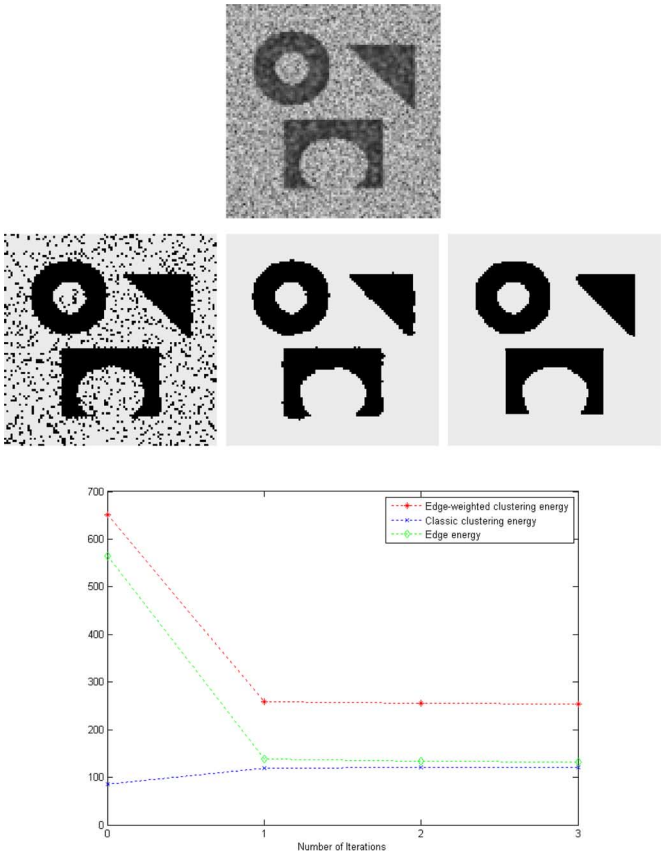

Fig. 5. Detection of three geometric objects from a noisy background using Algorithm EWCVT-B with the CVT clustering (2 clusters) as the initial configuration; Image size $= 100 \times 100, \omega = 5, \lambda = 0.01$, 3 iterations, CPU time $= 1.32$ s. Top row: original image; second row: results after 0, 1, and 3 iterations; bottom row: plot of variations of the energies.

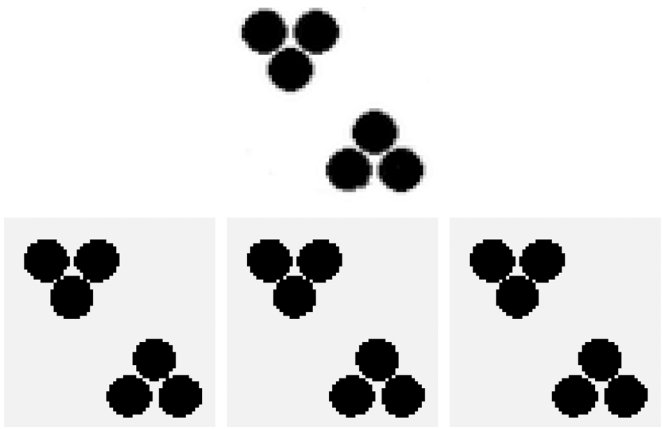

Fig. 6. Detection of three balls using Algorithm EWCVT-B with the CVT clustering (2 clusters) as the initial configuration. Image size $= 6 \times 64, \omega = 5$, $\lambda = 0.01$, 2 iterations, CPU time $= 0.43$ s.

λ. Algorithms EWCVT-A and EWCVT-B are all monotonic minimization processes for the EWVT energy, larger λ leads to a shorter edge length due to the second term in (16). It is easy to see that the grouping based on the so-called *Kanizsa's "proximity rule"* happens when $\lambda = 0.05$ (Fig. 7) and $\lambda = 0.2$ (Fig. 8). Among Figs. 6–8, λ in Fig. 8 is smallest and gives us the shortest boundary.

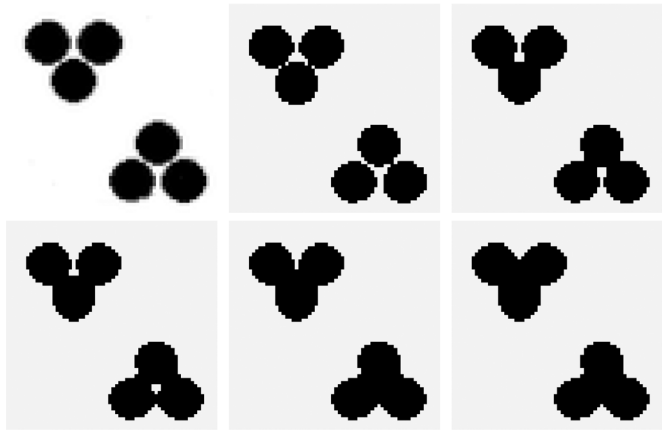

Fig. 7. Grouping of three balls based on Kanizsa's "proximity rule" when using Algorithm EWCVT-B with the CVT clustering (2 clusters) as the initial clusters. Image size $= 64 \times 64, \omega = 5, \lambda = 0.05$, 4 iterations, CPU time $= 0.57$ s.

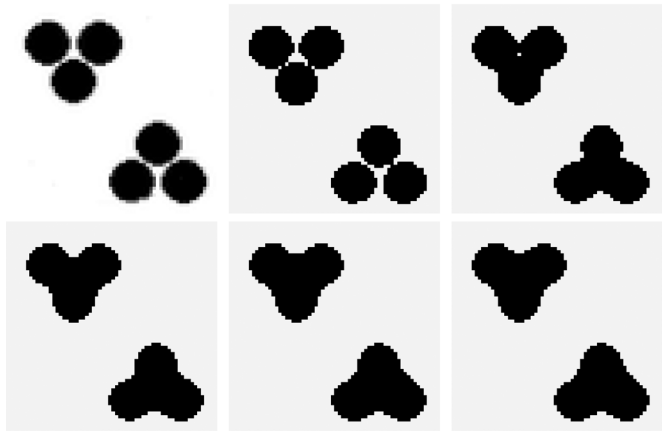

Fig. 8. Grouping of three balls based on Kanizsa's "proximity rule" when using Algorithm EWCVT-B with the CVT clustering (2 clusters) as the initial configuration. Image size $= 64 \times 64, \omega = 5, \lambda = 0.2$, 4 iterations, CPU time $= 0.57$ s.

As stated above, ω should be chosen according to the *size* of the noise blocks. In Fig. 9, Algorithm EWCVT-B with two clusters is applied to detect an airplane from a noisy background. We set ω to be 9 (that almost means that the largest diameter of the noise blocks which can removed after segmentation is 9). In this example, the initial two clusters of D are defined as follows: one is the set of all pixels inside the circle $\left\{(i,j) \in D : \sqrt{(i-50)^2 + (j-50)^2} < 40\right\}$ and the other is the set of all remaining pixels. As shown in Fig. 9, almost all noise blocks are eliminated after the segmentation. On the other hand, some parts of the airplane are also missing in the final segmented image. The reason is the size of some noise blocks is comparable with, or even bigger than those of certain parts of the airplane in the original image. In order to balance both influences, we surely need to use a more appropriate ω. Let us change ω to a smaller value, say 5 and keep all other settings the same. The segmentation results are presented in Fig. 10. Although there are some noise blocks left in the final segmented image, all of important features of the airplane are well preserved and overall we obtain a better segmentation for this image.

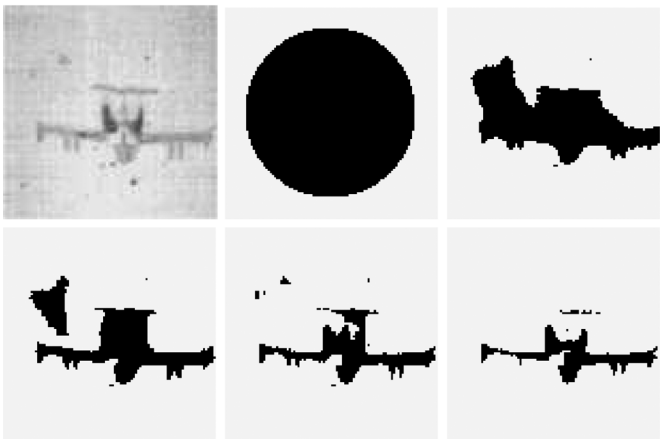

Fig. 9. Detection of an airplane from a noisy background using Algorithm EWCVT-B with 2 clusters. Image size $= 100 \times 100, \omega = 9, \lambda = 0.00015$, 37 iterations, CPU time $= 7.42$ s. The images except the original one are from iteration 0, 31, 34, 36, 37, respectively.

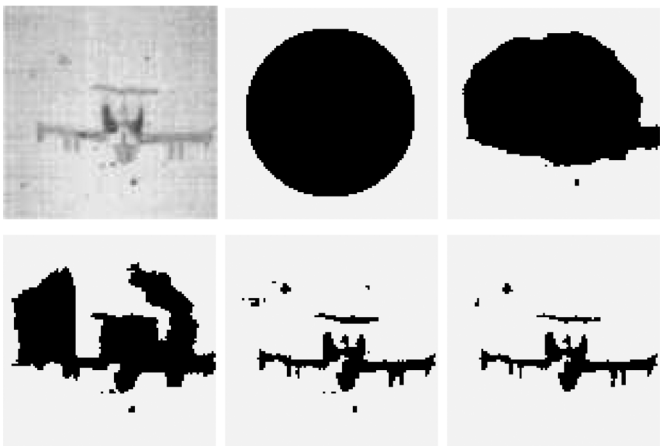

Fig. 10. Detection of an airplane from a noisy background using Algorithm EWCVT-B with 2 clusters. Image size $= 100 \times 100, w = 5, \lambda = 0.00015$, 34 iterations, CPU time $= 5.26$ s. The images except the original one are from iteration 0, 16, 26, 31, 34, respectively.

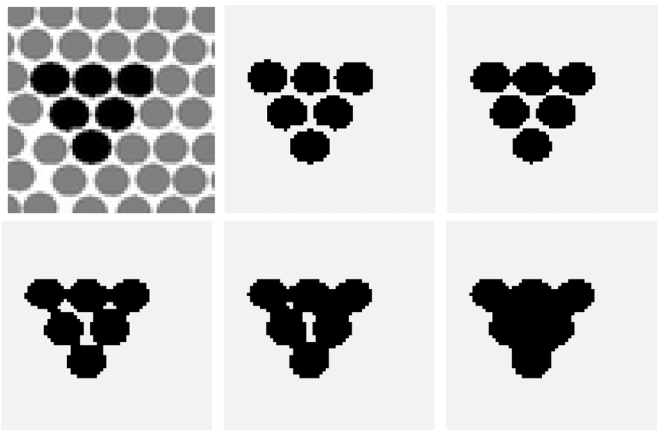

Fig. 11. Grouping based on chromatic identity when Algorithm EWCVT-B is applied with the CVT clustering (2 clusters) as the initial configuration. Image size $= 64 \times 64, \omega = 5, \lambda = 0.05$, 4 iterations, CPU time $= 0.57$ s.

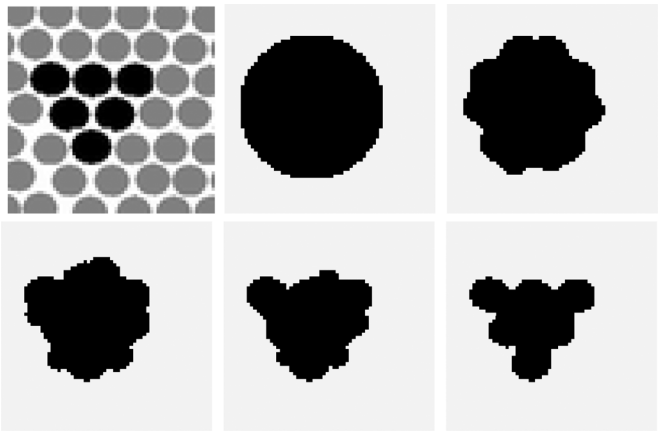

Fig. 12. Grouping based on chromatic identity when Algorithm EWCVT-B is applied with simpler initial clustering (2 clusters): one is the set of all pixels inside the circle $\left\{(i,j) \in D : \sqrt{(i-32)^2 + (j-27)^2} < 22\right\}$ and the other is the set of all remaining pixels. Image size $= 64 \times 64, \omega = 5, \lambda = 0.02, 20$ iterations, CPU time $= 1.57$ s. The images except the original image are from iteration 0, 6, 14, 16, 20, respectively.

We also would like to remark that our EWCVT-based algorithms are quite robust with respect to the selection of initial clusters. In Figs. 11 and 12, we present the segmentation results of Algorithm EWCVT-B with two clusters applied to the image of a group of well-organized balls, with different initial clusters (the second image in each of the figures). Grouping based on chromatic identity happens in both examples. The results show that the final segmented images are almost the same although the initial clusters are quite different. We also note that in Fig. 12 much more iterations are needed due to the much worse initial configuration compared with that in Fig. 11 where the initial CVT clusters already identify the six balls quite well.

C. More Two-Cluster Examples

In Fig. 1, we have shown the deficiency of the CVT-based segmentation to accurately identify the minefield when it is applied to a simulated noisy minefield image. We apply the EWCVT-based segmentation to the the same original image and the segmentation results are presented in Fig. 13. It takes 2 iterations for Algorithm EWCVT-B with the CVT clustering (two clusters) as the initial configuration and the final segmented image is really superior. A similar application of the EWCVT-based segmentation is presented in Fig. 14 for the image "Europe at night" where the basic CVT-based segmentation fails to obtain a good result. Note that in this example the initial clusters we choose are not a CVT clustering but a very simple circle-type one, see the second image of the first row in Fig. 14. It takes 73 iterations for Algorithm EWCVT-B to stop. The corresponding variations of energies during the iterations are also plotted in Fig. 14. We observe in this example that:

1) The classical clustering energy is two orders of magnitude larger than the edge energy. By carefully examining the intermediate images, we find that the boundary length increases greatly during iterations in order to approximate the coastline of European landmass. If the edge energy dominates the iterative process, the EWCVT model will result in a shorter boundary. This implies the classical clustering energy should dominate the iterative process in this specific application in order for the EWCVT-based segmentation to identify the Europe coastline accurately;

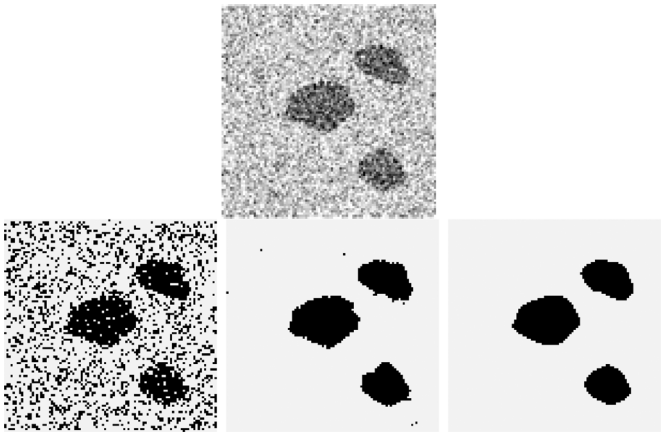

Fig. 13. Detection of a simulated minefield using Algorithm EWCVT-B with the CVT clustering (2 clusters) as the initial configuration. Image size $= 100 \times 100, \omega = 5, \lambda = 0.1$, 2 iterations, CPU time $= 1.15$ s.

thus, the value of λ should be small enough to allow the boundary to grow longer.

2) In Fig. 2, we already saw that the resulting image obtained by the basic CVT-based segmentation has too many edges. The EWCVT model gives us the ability to effectively control the number of the edge points in the final segmentation.
3) In this example, the edge energy keeps increasing while the edge-weighted clustering energy and the classical clustering energy keeps decreasing. This is partly due to the fact that the edge length (*edge points*) of the initial clusters is too small, i.e., far away from the accurate segmentation.
4) An important outcome of our EWCVT-based segmentation algorithm might be the clear identification of the English Strait.

In Fig. 15, we present the segmentation result of Algorithm EWCVT-B applied to an art picture from the Los Angeles Times by Brian Forrest. The image is segmented into two clusters and the boundary of the spiral in the resulting segmentation is again quite accurate and smooth.

Fig. 16 is used to show the ability of the EWCVT-based method to group objects together according to their orientation identity. This is done by replacing the original intensity values $\mathbb{U}$ by $orientation(\mathbb{U}) = \tan^{-1}(\mathbb{U}_y/\mathbb{U}_x)$ and then apply Algorithm EWCVT-B to the modified $\mathbb{U}$. Only 4 iterations are need and we obtain a nice grouping result of two clusters.

D. Multicluster Examples

Our model can also be easily applied to multicluster segmentation of images. The original image in Fig. 17 is almost identical to that in Figs. 4 and 5 except each of the three geometric objects has a distinct color: red, green and blue respectively. Thus, four clusters are needed in order to detect them accurately from the noisy background. Algorithm EWCVT-B starts with a CVT clustering (4 clusters) as the initial configuration and stops after 2 iterations. The CPU time is only 2.51 s. not much more than that used in Figs. 4 and 5. The segmentation result is clearly very accurate. The plot of variations of energies during the iterations are also presented in Fig. 17.

We now provide some more complex examples from real-world scenes in Figs. 18–20 where the left one is the original

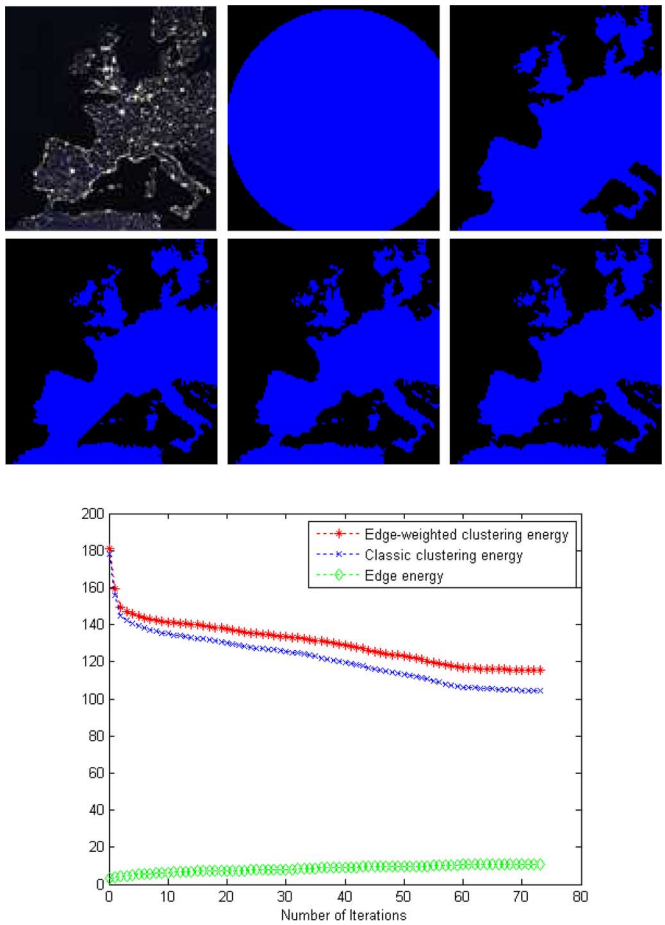

Fig. 14. Top to second row: Detection of the Europe land from the "Europe at night" image using Algorithm EWCVT-B with simple initial clustering (2 clusters): one is the set of all pixels inside the circle $\left\{(i,j) \in D : \sqrt{(i-55)^2 + (j-60)^2} < 60\right\}$ and the other is the set of all remaining pixels. Image size $= 106 \times 113, \omega = 5, \lambda = 0.0004$, 73 iterations, CPU time $= 16.85$ s. The images except the original image are from iteration 0, 26, 56, 66, 73, respectively. Bottom row: plot of variations of the energies.

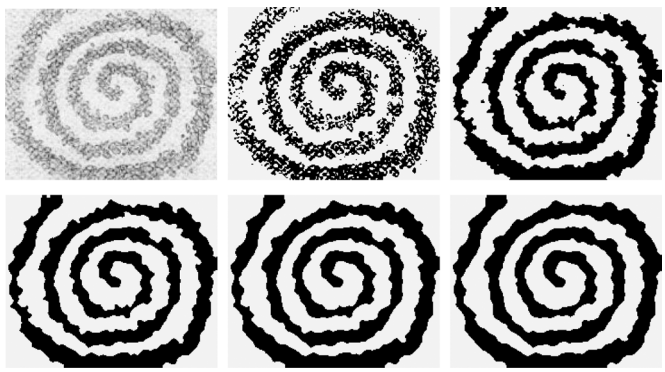

Fig. 15. Top to second row: Detection of a spiral from an art picture using Algorithm EWCVT-B with the CVT clustering (2 clusters) as the initial configuration. Image size $= 234 \times 191, \omega = 5, \lambda = 0.01$, 4 iterations, CPU time $= 6.20$ s.

image and the right one the final segmented image in all figures. The CVT clustering is used as the initial configuration for Algorithm EWCVT-B in all examples. In Fig. 18, we present the detection of skiers from a real-world background using Algorithm EWCVT-B with three clusters. The algorithm stops after

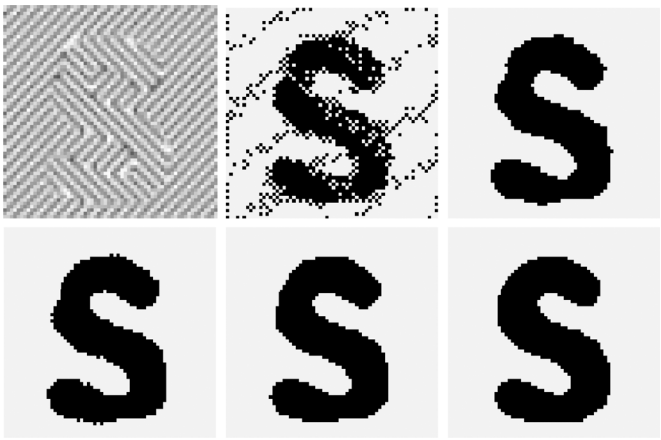

Fig. 16. Grouping based on orientation identity when using Algorithm EWCVT-B with the CVT clustering (2 clusters) as the initial configuration. Image size $= 64 \times 64, \omega = 5, \lambda = 1.0$, 4 iterations, CPU time $= 0.97$ s.

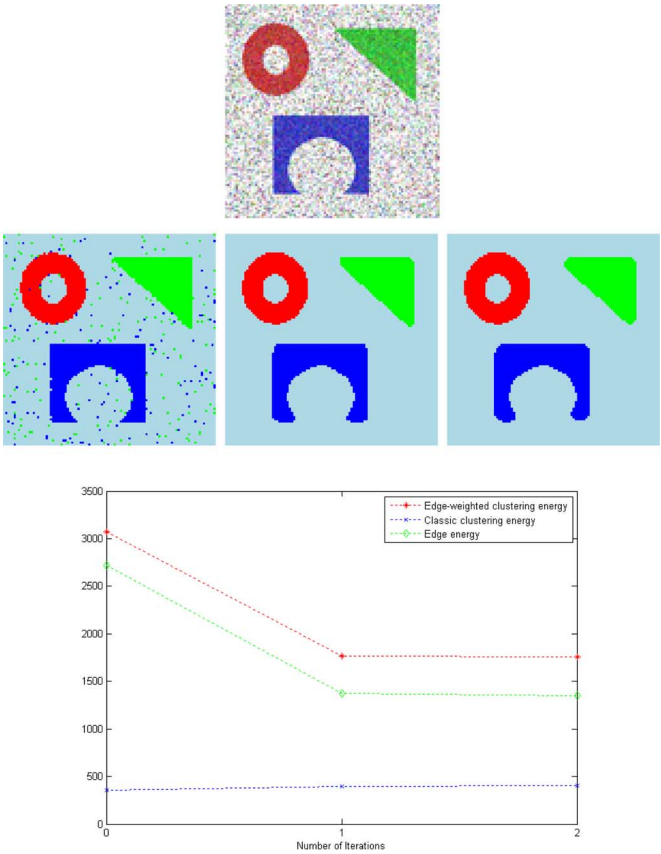

Fig. 17. Top row: Detection of three colorful geometric objects from a noisy background using Algorithm EWCVT-B with the CVT clustering (4 clusters) as the initial configuration. Image size $= 100 \times 100, \omega = 5, \lambda = 1.0$, 2 iterations, CPU time $= 2.51$ s. Bottom row: Plot of variations of the energies.

2 iterations and only takes 10.57 s. All skiers are well identified by our segmentation algorithm. Fig. 19 shows the segmentation results using Algorithm EWCVT-B with three clusters to the image of a group of elephants in a savanna scene. The algorithm also stops after two iterations and the running time is almost as same as the former example. Again, the elephants are

Fig. 18. Detection of skiers from a real-world background using Algorithm EWCVT-B with the CVT clustering (3 clusters) as the initial configuration. Left: Original image; right: final segmented image. Image size $= 320 \times 240, \omega = 5, \lambda = 0.01$, 2 iterations, CPU time $= 10.57$ s.

Fig. 19. Detection of elephants from a savanna scene using Algorithm EWCVT-B with the CVT clustering (3 clusters) as the initial configuration. Left: Original image; right: final segmented image. Image size $= 320 \times 214, \omega = 5, \lambda = 0.01$, 2 iterations, CPU time $= 10.65$ s.

Fig. 20. Detection of a starfish from a sea-bottom scene using Algorithm EWCVT-B with the CVT clustering as the initial configuration. Image size $= 320 \times 214$. Top-left: Original image; top-right: segmented image with 3 clusters; bottom-left: segmented image with 4 clusters; bottom-right: segmented image with 5 clusters. For the segmentation with 5 clusters: $\omega = 5, \lambda = 0.005$, 2 iterations, CPU time $= 15.75$ s.

clearly detected by our algorithm. Fig. 20 presents the detection of a star-fish from a real sea-bottom scene using Algorithm EWCVT-B with three, four and five clusters, respectively. Due to the strong inhomogeneous background, the segmentation result with three clusters is clear not good, but that with four and five clusters are both quite accurate. The algorithm stops after two iterations for all cases. The running time for the case of five clusters is 15.75 s, slightly longer than former two examples since more clusters are used. These examples demonstrate that our model and algorithms are quite effective and efficient for segmenting complex images of real-world scenes. The results also indicate the flexibility and robustness of our method with respect to the number of clusters.

V. Concluding Remarks

In this paper, we generalize the basic centroidal Voronoi tessellation model for image segmentation to a new edge-weighted centroidal Voronoi tessellation model and develop some efficient algorithms for its implementation in practical applications. The EWCVT-based algorithms are essentially classical clustering algorithms so that they are often computationally less expensive than the popular and powerful partial differential equation based segmentation methods. Through extensive examples presented in the preceding section, we demonstrate many advantages of our method such as the efficiency in computational cost, the ability to handle any number of clusters, the robustness with respect to noises, and the flexibility to control the segmentation accuracy. Some of our future work includes the intensity inhomogeneous image segmentation and reconstruction of multichannel images based on our EWCVT model.

Acknowledgment

The authors also would like to thank the referees for their valuable suggestions which significantly improved the paper. The original images in Figs. 1, 6, 7, 8, 13, 15, and 16 are from [6]. The original image in Fig. 18 is from dataset provided by LabelMe [33] (*http://labelme.csail.mit.edu/LabelMeToolbox/index.html*). The original image in Figs. 19 and 20 is from the Berkeley Segmentation Dataset and Benchmark [29].

References

[1] S. Byers and A. Raftery, "Nearest-neighbor cluster removal for estimating features in spatial point processes," *J. Amer. Statist. Assoc.*, vol. 93, pp. 577–584, 1998.
[2] K. Castleman, *Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1990.
[3] 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.
[4] T. Chan, J. Shen, and L. Vese, "Variational PDE models in image processing," *Notices AMS*, vol. 50, pp. 14–26, 2003.
[5] T. Chan and L. Vese, "An efficient variational multipase motion for the Mumford-Shah segmentation model," in *Proc. 34th Asilomar Conf. Signals, Syst., Comput.*, 2000, vol. 1, pp. 490–494.
[6] T. Chan and L. Vese, "Active contours withour edges," *IEEE Trans. Image Process.*, vol. 10, pp. 266–277, 2001.
[7] T. Chan and L. Vese, "Active contour and segmentation moderls using geometric PDE's for medical imaging," in *Geometric Methods in Bio-Medical Image Processing*, R. Malladi, Ed. Berlin, Germany: Springer, 2002, pp. 63–75.
[8] L. Cohen, E. Bardinet, and N. Ayache, "Surface reconstruction using active contour models," in *Proc. SPIE Conf. Geometric Methods in Computer Vision*, Bellingham, WA, 1993.
[9] E. D. Simon, *Clustering Analysis in Digital Pattern Recognition*, K. Fu, Ed. Berlin, Germany: Springer, 1980, pp. 47–94.
[10] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: Applications and algorithms," *SIAM Rev.*, vol. 41, pp. 637–676, 1999.
[11] Q. D. M. Gunzburger and L. Ju, "Constrained centroidal Voronoi tessellations on general surfaces," *SIAM J. Sci. Comput.*, vol. 24, pp. 1488–1506, 2003.
[12] Q. Du, M. Gunzburger, L. Ju, and X. Wang, "Voronoi tessellation algorithms for image compression and segmentation," *J. Math. Imag. Vis.*, vol. 24, pp. 177–194, 2006.
[13] Q. Du and X. Wang, "Tessellation and clustering by mixture models and their parallel implementations," in *Proc. 4h SIAM Data Mining Conf.*, Orlando, FL, 2004, pp. 257–268.
[14] P. Felzenszwalb and D. Huttenlocher, "Effient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, pp. 167–181, 2004.
[15] R. Gonzalez and R. Woods, *Digital Image Processing*. Upper Saddle River, NJ: Prentice-Hall, 2002.
[16] J. Hartigan, *Clustering Algorithms*. New York: Wiley, 1975.
[17] J. Hartigan and M. Wong, "Algorithm as 136: A k-means clustering algorithm," *Appl. Statist.*, vol. 28, pp. 100–108, 1979.
[18] D. Hoiem, A. Efros, and M. Hebert, "Geometric context from a single image," in *Proc. 10th IEEE Int. Conf. Computer Vision*, 2005, vol. 1, pp. 654–661.
[19] A. Hausner, "Simulating decorative mosaics," in *Proc. 28th Annu. Conf. Computer Graphics and Interactive Techniques*, 2001, pp. 573–580.
[20] A. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[21] A. Jain and R. Dubes, *Aglorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[22] L. Ju, Q. Du, and M. Gunzburger, "Probabilistic methods for centroidal voronoi tessellations and their parallel implementations," *Parallel Comput.*, vol. 28, pp. 1477–1500, 2002.
[23] R. Kannan, S. Vempala, and A. Vetta, "On clusterings-good, bad and spectral," in *Proc. 41st Annu. Symp. Foundations of Computer Science*, 2000, pp. 367–377.
[24] K. Kanizsa, *La Grammaire du Voir Essais Sur La Perception: Diderot Editeur*, Arts et Sciences, 1997.
[25] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 24, pp. 881–892, 2002.
[26] S. Lloyd, "Least squarse quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, pp. 129–137, 1982.
[27] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997, pp. 731–737.
[28] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 888–905, 2000.
[29] D. Martin, C. Fowlkes, D. Tai, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Computer Vision*, 2001, vol. 2, pp. 416–423.
[30] A. Okabe, B. Boots, K. Sugihara, and S. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Chichester, U.K.: Wiley, 2000.
[31] S. Osher, Level Set Method: Applications to Imaging Science, Univ. California, Los Angeles, 2002, CAM Rep. 02-43.
[32] S. Phillips, "Acceleration of k-means and related clustering algorithms," in *Proc. ALENEX*, 2002, pp. 166–177.
[33] B. Russell and A. Torralba, "Labelme: A database and web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, pp. 157–173, 2008.
[34] B. Sandberg, T. Chan, and L. Vese, A Level-Set and Gabor-Based Active Contour Algorithm for Segmenting Textured Images, Univ. California, Los Angeles, 2002, CAM Rep. 02-39.
[35] G. Sapiro, *Geometric Partial Differential Equations and Image Processing*. Cambridge, U.K.: Cambridge Univ., 2001.
[36] D. Sparks, "Algorithm as 58: Euclidean cluster analysis," *Appl. Statist.*, vol. 22, pp. 126–130, 1973.
[37] H. Späth, *Cluster Dissection and Analysis, Theory, FORTRAN Programs, Examples*. Upper Saddle River, NJ: Prentice-Hall, 1985.
[38] 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, pp. 271–293, 2002.

Jie Wang is a graduate student in scientific computing at the Florida State University, Tallahassee, working under the supervision of Prof. X. Wang starting in September 2007.

His research interests are in the fields of applied mathematics and scientific computation. His work involves numerical simulation and analysis, algorithms for image processing and data mining, parallel algorithms, and high-performance computing.

Lili Ju received the B.S. degree in mathematics from Wuhan University, China, in 1995, the M.S. degree in computational mathematics from the Chinese Academy of Sciences in 1998, and the Ph.D. degree in applied mathematics from Iowa State University in 2002.

He is an Associate Professor of mathematics at the University of South Carolina, Columbia. From 2002 to 2004, he was an Industrial Postdoctoral Researcher at the Institute of Mathematics and Its Applications, University of Minnesota. His research interests include numerical analysis, mesh generation, image processing, and parallel computation.

Xiaoqiang Wang received the B.S. degree in mathematics from Wuhan University, China, in 1995, the M.S. degree in mathematics from the Chinese Academy of Sciences in 1998, and the Ph.D. degree in applied mathematics from Pennsylvania State University in 2005.

He is an Assistant Professor of scientific computing at the Florida State University, Tallahassee. From 2005 to 2006, he was an Industrial Postdoctoral Researcher at the Institute of Mathematics and Its Applications, University of Minnesota. His research interests include numerical analysis, mathematical biology, image processing, scientific visualization, data mining, and high-performance scientific computing.