Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2010.32s.7

Vol. **3**, No. 2, pp. 223-244
May 2010

Edge-Weighted Centroidal Voronoi Tessellations

Jie Wang and Xiaoqiang Wang*

Department of Scientific Computing, Florida State University, Tallahassee, FL 532306-4120, USA.

Received 25 September 2009; Accepted (in revised version) 6 January 2010

Available online 8 March 2010

Abstract. Most existing applications of centroidal Voronoi tessellations (CVTs) lack consideration of the length of the cluster boundaries. In this paper we propose a new model and algorithms to produce segmentations which would minimize the total energy — a sum of the classic CVT energy and the weighted length of cluster boundaries. To distinguish it with the classic CVTs, we call it an Edge-Weighted CVT (EWCVT). The concept of EWCVT is expected to build a mathematical base for all CVT related data classifications with requirement of smoothness of the cluster boundaries. The EWCVT method is easy in implementation, fast in computation, and natural for any number of clusters.

AMS subject classifications: 52B55

Key words: Centroidal Voronoi tessellations, cluster boundary, edge detection, clustering, image processing.

1. Introduction

Centroidal Voronoi tessellations (CVTs) are special Voronoi tessellations whose *generators* are also the *centroids* of the associated Voronoi regions, with respect to a given density function. In the past few years, CVT-based methodologies have been applied successfully to diverse disciplines, including but not limited to, image processing and analysis [8, 14, 32], vector quantization and data analysis [17, 18], model reduction [12], high-quality point sampling [23], meshless computing [13], mesh generation and optimization [1, 16, 20], numerical partial differential equations [11, 21], and computer graphics and vision [18, 27]. The application list is still growing.

Lots of applications of clustering require the cluster boundaries to be smooth, while keeping the total length of the boundaries to be as small as possible. For example, we often obtain zigzag cluster boundaries when the basic CVT technique is applied to classify data sets. Especially in image segmentation, those zigzag boundaries are mainly due to noises or natural properties of the images, smoothing the boundaries can help us reduce or even eliminate the noises or unnecessary details [32].

*Corresponding author. *Email addresses:* jw05j@fsu.edu (J. Wang), wwang3@fsu.edu (X. Wang)

The popular level set method provides us a classic method to partially solve this problem [2–7, 9, 25, 28, 31]. However, it becomes very complicated when dealing with more than two clusters since multi-phase level sets have to be considered [31] in this case. In contrast, the CVT based techniques do not cause significant increase of computational cost due to increment of the number of clusters. In fact, we successfully developed an improved CVTs model for image segmentation recently and obtained very satisfactory results [32]. Our new model produces smooth boundaries in a controllable manner. More important, our model can be easily generalized to handle the cases of more than two clusters without introducing much difficulty in both theoretical and computational considerations. The key idea of our new model is to introduce a new energy term related to the *cluster boundaries*. And thus we call our model an Edge Weighted Centroidal Voronoi Tessellations (EWCVT) model. In this paper, we apply the similar idea to general data spaces, not restricted to image and even not restricted to 2D data set. The main contribution of our work here is to build a mathematical base for all CVT related data classification/clustering with requirement of smoothness of the cluster boundaries.

Here we would like to emphasize the differences between the method proposed in this paper and the one discussed in our another paper [32]. These two methods have the same name — EWCVT, because they both from the same idea: adding the weighted edge length to the classic CVT energy. In [32] the edge length considered is from physical space while the CVT energy is specifically from the color space (the image intensity), however, this paper handles the edge length and the CVT energy in the same physical space. Another major difference is that the method in [32] is designed only for 2D images, while the method in this paper can handle any dimensional data clustering problems. One may think about these two papers in this way: suppose we have a box filled by several soap bubbles, paper [32] takes a photo and tries to divide them by the color difference, this paper tries to simulate them by calculating their occupancy and surface tension.

We organize our paper as follows. First, we give a brief review of the classic CVT models and related algorithms in Section 2. In Section 3, the new EWCVT model and corresponding implementation algorithms are carefully developed. Together with some analysis and discussions, extensive numerical examples are presented to demonstrate special features of the EWCVT model in Section 4. Concluding remarks are finally given in Section 5.

2. Review of Classic Centroidal Voronoi Tessellations

2.1. Basic definition

Generally speaking, the computational domain is an open subset of $\mathbb{R}^n$, say Ω. A tessellation of Ω is in fact a non-overlapping covering $\mathscr{V} = \{V_l\}_{l=1}^L$ of Ω. Rigorously, we require $V_i \cap V_j = \emptyset$ if $i \neq j$ and $\overline{\Omega} = \cup_{l=1}^L \overline{V}_l$. The *Voronoi* region V_k of Ω can be easily computed once we are given a set of points $\mathscr{Z} = \{z_l\}_{l=1}^L$ according to

$$V_k = \Big\{ x \in \Omega : |x - z_k| \le |x - z_l|, \text{ for } l = 1, \cdots, L \Big\}, \qquad k = 1, \cdots, L. \tag{2.1}$$

where $|\cdot|$ is some predefined metric measure, e.g., the Euclidean distance on $\mathbb{R}^n$. The set of computed regions $\mathscr{V}=\{V_l\}_{l=1}^L$ is called a *Voronoi tessellation* or *Voronoi clustering* [24] of Ω. The set of chosen points $\mathscr{Z}=\{z_l\}_{l=1}^L$ is referred as the *Voronoi generators*. The Voronoi tessellation $\mathscr{V}$ can be viewed as a special partition of Ω.

On the other hand, if we are given an arbitrary partition $\{U_l\}_{l=1}^L$ of $\Omega\in\mathbb{R}^n$, and a density function $\rho(x)$ defined for each $x\in\Omega$, the so-called *centroid* (*center of mass* or *cluster means*) of every cell U_l is the point $\bar{z}_l\in U_l$ defined by

$$\min_{z\in U_l}\int_{U_l}\rho(x)|x-z|^2dx. \tag{2.2}$$

By simple calculation, the explicit form of $\bar{z}_l$ is given by:

$$\bar{z}_l=\frac{\displaystyle\int_{U_l}x\rho(x)dx}{\displaystyle\int_{U_l}\rho(x)dx}. \tag{2.3}$$

For an arbitrary Voronoi tessellation $\left(\{z_l\}_{l=1}^L;\{V_l\}_{l=1}^L\right)$ of Ω, normally we have $z_l\neq\bar{z}_l$ for $l=1,\cdots,L$. In other words, we generally can not expect the generators which generate the Voronoi tessellation to be happen to be the centroids of the corresponding clusters.

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

$$z_l=\bar{z}_l,\ \text{for } l=1,\cdots,L,$$

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

We note that for a given domain Ω, the CVT may not be unique [10]. Therefore, determining a CVT of Ω is actually a process to find a set of generators $\{z_l\}_{l=1}^L$ such that $\{z_l\}_{l=1}^L$ are simultaneously the centroids of the associated Voronoi regions $\{V_l\}_{l=1}^L$.

2.2. Clustering energies and Lloyd's algorithm

The construction of CVTs can be viewed as an *energy* minimization process [10]. For a given set Ω and a set of generators $\mathscr{Z}=\{z_l\}_{l=1}^L$, let us define the VT energy of $\mathscr{Z}$ as follows:

$$E_{VT}^c(\mathscr{Z})=\int_{x\in\Omega}\mathscr{E}_{VT}(x)dx, \tag{2.4}$$

where

$$\mathscr{E}_{VT}(x)=\min_{l=1,\cdots,L}\rho(x)|x-z_l|^2. \tag{2.5}$$

The superscript “c” denotes continuous case in order to distinguish with the discrete case. More generally, for any set of points $\mathscr{Z} = \{z_l\}_{l=1}^L$ and any partition $\mathscr{U} = \{U_l\}_{l=1}^L$ of Ω, the classical *clustering energy* of $(\mathscr{Z};\mathscr{U})$ can be defined as:

$$E^c(\mathscr{Z};\mathscr{U}) = \sum_{l=1}^{L} \int_{x\in U_l} \rho(x)|x - z_l|^2 dx. \tag{2.6}$$

Note that $\mathscr{U} = \{U_l\}_{l=1}^L$ are not necessarily the Voronoi regions corresponding to the set of generators $\mathscr{Z} = \{z_l\}_{l=1}^L$ in the definition.

Combining Eqs. (2.1) and (2.4) together, we can rewrite the VT energy as

$$E_{VT}^c(\mathscr{Z}) = \sum_{l=1}^{L} \int_{x\in V_l} \rho(x)|x - z_l|^2 dx = E^c(\mathscr{Z};\mathscr{V}), \tag{2.7}$$

where $\mathscr{V} = \{V_l\}_{l=1}^L$ are the corresponding Voronoi regions generated by $\{z_l\}_{l=1}^L$. According to Eq. (2.5), it is clear that for a fixed set of generators $\{z_l\}_{l=1}^L$, the VT energy

$$E_{VT}^c(\mathscr{Z}) = \min_{\mathscr{U}=\{U_l\}_{l=1}^L} E^c(\mathscr{Z};\mathscr{U}).$$

In fact we have a stronger conclusion stated below which is proved in [10]:

Theorem 2.1. *For a given domain Ω, the classical clustering energy $E(\mathscr{Z};\mathscr{U})$ reaches its minimum only if $(\mathscr{Z};\mathscr{U})$ form a CVT of Ω, i.e, $\mathscr{U}$ are Voronoi regions of Ω generated by the generators $\mathscr{Z}$ and simultaneously, each z_l is the centroid of U_l.*

Let us define a projection function $\pi_{\mathscr{U}}\colon \Omega \to \{1,\cdots,L\}$ which maps each point x to its cluster index, i.e., for any point $x \in \Omega$,

$$\pi_{\mathscr{U}}(x) = l^*, \ \text{ if } \ x \in U_{l^*} \ \text{ (i.e., } \ x \in U_{l^*}). \tag{2.8}$$

Then the classic clustering energy (2.6) can be rewritten in a much general manner

$$E^c(\mathscr{Z};\mathscr{U}) = \int_{x\in\Omega} \mathscr{E}(x)dx,$$

where

$$\mathscr{E}(x) = \rho(x)|x - w_{\pi_{\mathscr{U}}(x)}|^2. \tag{2.9}$$

Once the clusters $\mathscr{U} = \{U_l\}_{l=1}^L$ are determined, the boundaries are defined by

$$\partial\mathscr{U} = \cup_{l=1}^L \partial U_l.$$

A well known algorithm to construct CVTs is the so called Lloyd’s algorithm, see, e.g., [19,29,30] for details.

Algorithm CVT-A (Lloyd's). Assume we are given a domain Ω, a positive integer L and a density function $\rho(x)$ defined on Ω. Let us choose arbitrarily L points $\{z_\ell\}_{\ell=1}^L \in \Omega$.

1. Compute the Voronoi clusters $\{V_\ell\}_{\ell=1}^L$ of Ω with respect to $\{z_\ell\}_{\ell=1}^L$.
2. For each cluster V_ℓ, $\ell = 1, \cdots, L$, compute the cluster mean $\overline{z}_\ell$.
3. Once $\overline{z}_\ell$ and z_ℓ are the same, terminate the loop and return the current configuration $\left(\{z_\ell\}_{\ell=1}^L;\ \{V_\ell\}_{\ell=1}^L\right)$; otherwise, set $z_\ell = \overline{z}_\ell$ for $\ell = 1, \cdots L$ and go to Step 1.

According to the definition of Voronoi regions and centroids, the VT energy will be guaranteed to be decreased along the iterations unless a local minimizer is reached.

2.3. Discrete cases

We can easily derive the discrete version of CVT by simply replacing the integration by summation. Let us say the discretized domain of Ω is $\mathscr{D}$. Given the density function $\rho(\cdot)$ defined on $\mathscr{D}$, the centroid $\bar{z}$ of its subset U can be written as:

$$\bar{z}^d = \frac{\sum_U x\rho(x)}{\sum_U \rho(x)}. \tag{2.10}$$

If we choose randomly a set of points $\mathscr{Z} = \{z_l\}_{l=1}^L$ as generators and an arbitrary partition $\mathscr{U} = \{U_l\}_{l=1}^L$ of $\mathscr{D}$ is given, the classic clustering energy then is:

$$E^d(\mathscr{Z};\mathscr{U}) = \sum_{x\in\mathscr{D}} \mathscr{E}(x) = \sum_{x\in\mathscr{D}} \rho(x)|x - z_{\pi_{\mathscr{U}}(x)}|^2.$$

Moreover, if the partition $\mathscr{U} = \{U_l\}_{l=1}^L$ of $\mathscr{D}$ is the Voronoi tessellation $\mathscr{V} = \{V_l\}_{l=1}^L$ associated with $\mathscr{Z} = \{z_l\}_{l=1}^L$, we can derive the VT energy of $\mathscr{Z}$ for the discrete case similarly:

$$E_{VT}^d(\mathscr{Z}) = \sum_{l=1}^{L} \sum_{x\in V_l} \rho(x)|x - z_l|^2. \tag{2.11}$$

Algorithm CVT-A is also applicable to the discrete cases without significant modification. The computational complexity of Algorithm CVT-A is in general $\mathscr{O}(k \times L \times N)$ where k denotes the total number of iterations and N is the number of points which belong to $\mathscr{D}$. Points transfers will not happen until the end of each iteration in Algorithm A, i.e., it does not account for the change of the cluster means until all centroids are computed. In order to take into account changes of cluster means as soon as they are determined, the following accelerated version was developed.

```
Algorithm CVT-B. Assume the discrete domain $\mathscr{D}$ is given. We have a positive integer
$L$ and choose arbitrarily $L$ points $\{z_\ell\}_{\ell=1}^L \in \mathscr{D}$. Then we compute the Voronoi clusters
$\{V_\ell\}_{\ell=1}^L$ associated with $\{z_\ell\}_{\ell=1}^L \in \mathscr{D}$.
  1. For each $x \in \mathscr{D}$,
      (a) evaluate the classic clustering energy for all possible transfers of $x$ from its
          current cluster $V_\ell$ to any of the other clusters $V_k$, $k = 1, \ldots, L$, $k \neq \ell$;
      (b) if the test transfer of $x$ from $V_\ell$ to $V_m$ decreases the classic clustering energy
          the most, then
            i. set $x \in V_m$;
           ii. recompute the centroids $\bar{z}_\ell$ and $\bar{z}_m$ of the modified clusters $V_\ell$ and $V_m$,
               then set $z_\ell = \bar{z}_\ell$ and $z_m = \bar{z}_m$, respectively.
  2. If no transfer occurs, terminate the loop and return the current configuration
     $\left(\{z_\ell\}_{\ell=1}^L; \{V_\ell\}_{\ell=1}^L\right)$; otherwise, go to Step 1.
```

Both Algorithms CVT-A and CVT-B result in a k-means clustering. The classic clustering energy is guaranteed to be decreased after each iteration in both Algorithms CVT-A and CVT-B, and finally converges to its minimum. The efficiency of Algorithm CVT-B lies on the fact that an iteration of Algorithm CVT-B leads to a much larger decrease in the energy than does an iteration of Algorithm CVT-A. Thus a much smaller number of iterations is required for Algorithm CVT-B to converge. Practically, a hybrid approach is feasible in which one starts with the Algorithm CVT-A and then switches to Algorithm CVT-B. For further discussions on the comparison of Algorithms CVT-A and CVT-B and some possible improvements, see [14, 17, 32] for details.

3. An edge-weighted Centroidal Voronoi Tessellation model and its implementation algorithms

In this section, first we introduce an edge energy to represent the length of cluster boundaries. And the further introduced the so called edge-weighted clustering energy is the key improvement we have made to the classic CVTs. Second, due to the new introduced edge energy, we derive a concise formula of the new distance function incorporated with the edge-weighted CVT clustering energy. Accordingly, the new *Voronoi regions* can be easily computed based on the new distance function. Finally the detailed implementations of our algorithms are provided with some discussions.

3.1. Edge energy and the length of boundaries

To be rigorous, let us make some conventions first. Recall that, we are handling an open set $\Omega \in \mathbb{R}^n$. In addition we are given a partition $\mathscr{U} = \{U_l\}_{l=1}^L$ of Ω and a set of points $\mathscr{Z} = \{z_l\}_{l=1}^L$ in Ω. Note that, $\mathscr{U} = \{U_l\}_{l=1}^L$ is an arbitrary partition, and as a result, $\mathscr{Z} = \{z_l\}_{l=1}^L$ is not necessarily the generators of $\mathscr{U} = \{U_l\}_{l=1}^L$.

For each point $x \in \Omega$, let $\mathbb{N}_r(x)$ denote a local neighborhood of x, which can be a cube or a ball centered at x (if $\mathbb{N}_r(x)$ represents a cube, its side length is r; or r is the radius if $\mathbb{N}_r(x)$ denotes a ball). Every $y \in \mathbb{N}_r(x)$ is called a neighbor point of x. Furthermore, we define a local characteristic function for each $x \in \Omega$, i.e., $\mathscr{X}_{(x;\mathscr{U})}$: $\mathbb{N}_r(x) \to \{0,1\}$ as

$$\mathscr{X}_{(x;\mathscr{U})}(y) = \begin{cases} 1, & \text{if } \pi_{\mathscr{U}}(y) \neq \pi_{\mathscr{U}}(x); \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

With the help of these notations, the edge energy of a point x can be written as

$$\mathscr{E}^c_{\mathscr{L}}(x;\mathscr{U}) = \int_{\mathbb{N}_r(x)} \mathscr{X}_{(x;\mathscr{U})}(y)dy. \tag{3.2}$$

Therefore the total edge energy is simply the integration of $\mathscr{E}^c_{\mathscr{L}}(x;\mathscr{U})$, i.e., edge energy of each single point x, over the whole domain:

$$E^c_{\mathscr{L}}(\mathscr{U}) = \int_{\Omega} \mathscr{E}^c_{\mathscr{L}}(x;\mathscr{U})dx = \int_{\Omega}\int_{\mathbb{N}_r(x)} \mathscr{X}_{(x;\mathscr{U})}(y)dydx. \tag{3.3}$$

In fact, the edge energy $E^c_{\mathscr{L}}(\mathscr{U})$ is proportional to the “*real length*" of the cluster boundaries if r is small enough. The analytical results are given in [32] which we will restate in Theorem 3.1.

Theorem 3.1. *Define the total edge energy as* (3.3)*, we have*

$$\lim_{r\to 0} E^c_{\mathscr{L}}(\mathscr{U}) \to \alpha r^3 L,$$

where L is the length (*surface area for $n \geq 3$*) *of the clusters boundaries, and α is a constant only depending on the shape of the neighborhood $\mathbb{N}_r(x)$, for the special case of the disc shaped neighborhood of 2D data set, $\alpha = 4/3$.*

Note that for 2D data set, L is the length of the boundaries; for $n \geq 3$, the boundaries are $(n-1)$ dimensional manifolds and L is actually the surface area or volume of those manifolds. Theorem 3.1 is proved only for 2D data set in [32], the proof in higher dimensional space is essentially the same.

Combining the classic clustering energy (2.6) and the edge energy (3.3), we define the *edge-weighted clustering energy* as follows:

$$\begin{aligned}\widehat{E}^c(\mathscr{W};\mathscr{U}) &= E^c(\mathscr{W};\mathscr{U}) + \lambda E^c_{\mathscr{L}}(\mathscr{U}) \\ &= \int_{x\in\Omega} \left[\mathscr{E}(x) + \lambda \mathscr{E}^c_{\mathscr{L}}(x;\mathscr{U})\right] dx \\ &= \sum_{l=1}^{L} \int_{x\in U_l} \rho(x)|x-w_l|^2 dx + \lambda \int_{x\in\Omega}\int_{y\in\mathbb{N}_r(x)} \mathscr{X}_{(x,\mathscr{U})}(y)dydx,\end{aligned} \tag{3.4}$$

where $\lambda > 0$ is a positive weight parameter. The energy consists of two parts:

- First, classic clustering energy which comes from the classic CVT model.
- Second, edge energy which is related to the *"length of boundaries*".

Formulas (3.2)-(3.4) for the discrete case can be easily derived as follows. For a single point $x \in \mathscr{D}$, the edge energy of x is given by:

$$\mathscr{E}^d_{\mathscr{L}}(x;\mathscr{U}) = \sum_{y\in\mathbb{N}_r(x)} \mathscr{X}_{(x,\mathscr{U})}(y), \tag{3.5}$$

and then the total edge energy is the summation of the edge energy of each points instead of integration in the continuous case:

$$E^d_{\mathscr{L}}(\mathscr{U}) = \sum_{x\in\mathscr{D}} \mathscr{E}^d_{\mathscr{L}}(x;\mathscr{U}). \tag{3.6}$$

Similarly, we can write the *edge-weighted clustering energy* for the discrete case as:

$$\begin{aligned} \widehat{E}^d(\mathscr{W};\mathscr{U}) &= E^d(\mathscr{W};\mathscr{U}) + \lambda E^d_{\mathscr{L}}(\mathscr{U}) \\ &= \sum_{x\in\mathscr{D}} \left[\mathscr{E}(x) + \lambda \mathscr{E}^d_{\mathscr{L}}(x;\mathscr{U})\right] \\ &= \sum_{l=1}^{L}\sum_{x\in U_l} \rho(x)|x-w_l|^2 + \lambda \sum_{x\in\mathscr{D}}\sum_{y\in\mathbb{N}_r(x)} \mathscr{X}_{(x;\mathscr{U})}(y). \end{aligned} \tag{3.7}$$

Clearly, the second term is the weighted edge energy of the whole data set. The new energy incorporates information of the boundaries length and will play a crucial role in our new method. Due to the weight parameter λ, smoothness of the resulting boundaries can be controlled effectively. Moreover, if we carefully examine equation (3.7), one may find that once we transfer a point x from its current cluster to another cluster, not only the edge energy of x, but the edge energy of neighbor points y of x changes. In contrast, the transfer of x will not affect the classic clustering energy of any other points.

3.2. Edge-weighted distance

Now we need to construct an *edge-weighted distance* function to determine the distance from a point to a generator. For simplicity, let us consider the discrete case.

The equation (3.7) can be rewritten as

$$\begin{aligned} \widehat{E}^d(\mathscr{W};\mathscr{U}) = & \left[\sum_{y\in\mathscr{D}\backslash x} \rho(y)|y-w_{\pi_{\mathscr{U}}(y)}|^2\right] + \rho(x)|x-w_{\pi_{\mathscr{U}}(x)}|^2 \\ & + \lambda\left[\sum_{y\in\mathscr{D}\backslash x} \mathscr{E}^d_{\mathscr{L}}(y;\mathscr{U})\right] + \lambda\mathscr{E}^d_{\mathscr{L}}(x;\mathscr{U}). \end{aligned} \tag{3.8}$$

If we transfer a point x from its current cluster U_l to another cluster U_k, the total energy variation will come from the variation of the four terms on the right hand side of Eq. (3.8). Obviously, the transfer will not affect the first term on the right-hand side of (3.8). The change of the second term can be easily derived as

$$\rho(x)\left[|x-w_k|^2-|x-w_l|^2\right]. \tag{3.9}$$

Let n_s denote the number of points within $U_s \cap \mathbb{N}_r(x) \setminus x$. Under the current configuration, the weighted edge energy of point x is

$$\lambda \mathcal{E}^d_{\mathcal{L}}(x;\mathcal{U}) = \lambda \sum_{y\in\mathbb{N}_r(x)} \mathcal{X}_{(x;\mathcal{U})}(y) = \lambda \sum_{i\in\{1,\cdots,L\}\setminus\{l\}} n_i. \tag{3.10}$$

Similarly, after we move x to cluster U_k, its weighted edge energy changes to

$$\lambda \mathcal{E}^d_{\mathcal{L}}(x;\mathcal{U}) = \lambda \sum_{y\in\mathbb{N}_w(x)} \mathcal{X}_{(x;\mathcal{U})}(y) = \lambda \sum_{i\in\{1,\cdots,L\}\setminus\{k\}} n_i. \tag{3.11}$$

Therefore, transfer point x from its current cluster U_l to another cluster U_k will result in a change in the fourth term $\lambda\mathcal{E}^d_{\mathcal{L}}(x)$, which can be derived as

$$\lambda \sum_{i\in\{1,\cdots,L\}\setminus\{k\}} n_i - \lambda \sum_{i\in\{1,\cdots,L\}\setminus\{l\}} n_i = \lambda\left(n_l - n_k\right). \tag{3.12}$$

Note that transferring point x from U_l to U_k not only affects the edge energy of x itself but also all of the neighbor points y of x. Let us turn our attention to the term $\mathcal{E}^d_{\mathcal{L}}(y)$. If point y is outside of $\mathbb{N}_r(x)$, its edge energy will not be affected by the transfer of x. However, if $y \in \mathbb{N}_r(x) \cap U_p$, i.e., y is a neighbor point of x, we need to consider three different cases.

- First, if $p \neq l,k$, the edge energy $\mathcal{E}^d_{\mathcal{L}}(y)$ will be kept the same.
- Second, if $p = l$, moving x out of cluster U_l means $y \in U_l$ has one more neighbor point x which does not belong to the same cluster as y. Thus, the edge energy of y will be increased by λ.
- Third, if $p = k$, moving x into the cluster U_k implies $y \in U_k$ has one more neighbor point which belongs to the same cluster as y. Therefore, the edge energy of y will be decreased by λ.

Combining all of the discussions above and transferring x from U_l to U_k will result in a change in $\lambda\sum_{y\in\mathcal{D}\setminus x}\mathcal{E}^d_{\mathcal{L}}(y)$, which can be expressed as:

$$\lambda n_l - \lambda n_k = \lambda(n_l - n_k). \tag{3.13}$$

Based on all of the above analysis, we finally arrive at the variation of the total edge energy $\widehat{E}^d$ due to the moving of x from U_l to U_k by simply summing (3.9), (3.12) and

(3.13) together $\rho(x)|x-w_k|^2-\rho(x)|x-w_l|^2+2\lambda(n_l-n_k)$ which can be rewritten in a more symmetrical manner as

$$(\rho(x)|x-w_k|^2-2\lambda n_k)-(\rho(x)|x-w_l|^2-2\lambda n_l). \tag{3.14}$$

Formula (3.14) indicates that the *edge-weighted distance* from a point x to a generator w_k can be defined as

$$\mathrm{dist}(x,w_k)=\sqrt{\rho(x)|x-w_k|^2+2\lambda\tilde{n}_k}, \tag{3.15}$$

where $\tilde{n}_k=|\mathbb{N}_r(x)|-n_k-1$, which is the number of points within $\mathbb{N}_r(x)\setminus(U_k\cup x)$.

In conclusion, in order to decrease the total edge-weighted clustering energy $\widehat{E}(\mathscr{W};\mathscr{U})$ the most, we need to transfer a point x to the cluster whose generator is the *closest* one to it, in the sense that the edge-weighted distance defined by (3.15) is the shortest.

Note that for the continuous case, the above derivation is also applicable: the edge-weighted distance function has the same form except that $\tilde{n}_k=|\mathbb{N}_r(x)\setminus U_k|$ represents the area of $\mathbb{N}_r(x)\setminus U_k$ instead of the number of points inside $\mathbb{N}_r(x)\setminus U_k$. Moreover we should be aware that the edge-weighted distance is not a rigorous norm since it does not satisfy the triangular inequality. But it is good enough to determine the Voronoi regions to the corresponding generators.

3.3. Edge-weighted Voronoi regions

Although concepts developed above are applicable to both the continuous and discrete cases, we will consider the discrete cases only in the following for simplicity.

Assume we have a set of generators $\mathscr{Z}=\{z_l\}_{l=1}^L\in\mathscr{D}$, the edge-weighted Voronoi regions $\widetilde{\mathscr{Q}}=\{\widetilde{Q}_l\}_{l=1}^L$ are defined as

$$\begin{aligned}\widetilde{Q}_k=&\Big\{x\in\mathscr{D}:\mathrm{dist}(x,w_k)\le\mathrm{dist}(x,w_l),\\ &\quad\text{for } l=1,\cdots,L\Big\},\qquad k=1,\cdots,L.\end{aligned} \tag{3.16}$$

The distance function is defined by (3.15).

If we fix the generators $\mathscr{Z}$, Eq. (3.16) indicates that the edge-weighted Voronoi tessellation $\{\widetilde{Q}_l\}_{l=1}^L$ is the minimizer of the edge-weighted energy $\widehat{E}^d(\mathscr{Z},\mathscr{U})$, i.e.,

$$\widetilde{\mathscr{Q}}=\mathrm{argmin}_{\mathscr{U}}\,\widehat{E}^d(\mathscr{Z};\mathscr{U}).$$

The EWVT energy for a fixed set of generators $\mathscr{Z}=\{z_l\}_{l=1}^L$ can be written as

$$\widehat{E}^d_{EWVT}(\mathscr{Z})=\widehat{E}^d(\mathscr{Z};\widetilde{\mathscr{Q}}). \tag{3.17}$$

Thus, if we are given a set of generators $\mathscr{Z}=\{z_l\}_{l=1}^L$, we can adopt the following algorithm to construct the edge-weighted Voronoi regions associated with $\mathscr{Z}=\{z_l\}_{l=1}^L$.

Algorithm EWVT. Assume we have a set of generators $\mathscr{Z} = \{z_l\}_{l=1}^{L}$ and an arbitrary segmentation $\{\widetilde{U}_l\}_{l=1}^{L}$ of the discrete set $\mathscr{D}$. (Again, $\{\widetilde{U}_l\}_{l=1}^{L}$ are not necessarily the edge-weighted Voronoi regions generated by $\mathscr{Z} = \{z_l\}_{l=1}^{L}$.)

1. For every $x \in \mathscr{D}$,
   (a) evaluate the distance defined by Eq. (3.15) from the point x to all of the generators $\{z_l\}_{l=1}^{L}$;
   (b) transfer the point x to the cluster whose generator is the closest one to it. (Note that this is equivalent to an update of the partition $\{\widetilde{U}_l\}_{l=1}^{L}$).
2. If no point is moved, terminate the loop and $\{\widetilde{U}_l\}_{l=1}^{L}$ is the edge-weighted Voronoi region $\{\widetilde{Q}_l\}_{l=1}^{L}$ associated with $\mathscr{Z} = \{z_l\}_{l=1}^{L}$; otherwise, go to Step 1.

Clearly algorithm EWVT results in strict decrease in the energy until we arrive at an edge-weighted Voronoi tessellation of $\mathscr{D}$. Thus, the compactness of $\mathscr{D}$ guarantees the convergence of the algorithm.

3.4. Edge-weighted centroidal Voronoi tessellations: definition and algorithms

Algorithm EWVT provides an effective and efficient approach to determine the edge-weighted Voronoi regions $\widetilde{\mathscr{Q}} = \{\widetilde{Q}_l\}_{l=1}^{L}$ for a given set of generators. However on the other hand, if we have computed the edge-weighted Voronoi regions $\widetilde{\mathscr{Q}} = \{\widetilde{Q}_l\}_{l=1}^{L}$, a method to determine the corresponding edge-weighted centroids of $\{\widetilde{Q}_l\}_{l=1}^{L}$, i.e., the minimizer of the edge-weighted energy $\widehat{E}(*, \widetilde{\mathscr{Q}})$ ($\widetilde{\mathscr{Q}}$ are fixed), is in demand. In fact, because the edge energy $\mathscr{E}^{d}_{\mathscr{L}}(x)$ depends only on the current partition instead of the generators, Eq. (3.4) implies that the edge-weighted centroids are exactly the classic centroids of $\{\widetilde{D}_l\}_{l=1}^{L}$ given by Eq. (2.10).

Definition 2. *Assume we have an edge-weighted Voronoi tessellation* $(\{\tilde{z}_l\}_{l=1}^{L}; \{\widetilde{Q}_l\}_{l=1}^{L})$ *of* $\mathscr{D}$*, if the generators* $\{z_l\}_{l=1}^{L}$ *are simultaneously the centroids of the associate edge-weighted Voronoi regions* $\{\widetilde{Q}_l\}_{l=1}^{L}$*, we call it an* edge-weighted centroidal Voronoi tessellation (*EWCVT*) *of* $\mathscr{D}$.

From the above analysis, the following theorem is pretty straightforward:

Theorem 3.2. *Only if* $(\mathscr{Z}; \widetilde{\mathscr{Q}})$ *form an EWCVT of* $\mathscr{D}$*, i.e.,* $\widetilde{\mathscr{Q}}$ *are edge-weighted Voronoi regions of* $\mathscr{D}$ *associated with the generators* $\mathscr{Z}$ *and simultaneously* $\mathscr{Z}$ *are the corresponding centroids of the regions* $\widetilde{\mathscr{Q}}$*, the minimum of edge-weighted energy* $\widehat{E}(\mathscr{Z}; \widetilde{\mathscr{Q}})$ *is achieved.*

Based on the principle of energy reduction, we construct the following algorithm to calculate the EWCVTs.

Algorithm EWCVT-A. Suppose we are given an integer L and an arbitrary partition $\{\widetilde{U}_l\}_{l=1}^L$ of the discrete set $\mathscr{D}$.

1. Compute centroid $\bar{z}_l$ by (2.10) for each cluster $\widetilde{U}_l,\ l=1,\cdots,L,$.
2. Set $\{\bar{z}\}_{l=1}^L$ as generators, adopt Algorithm EWVT to generate the associate edge-weighted Voronoi clusters $\widetilde{\mathscr{Q}}' = \{\widetilde{Q}_l'\}_{l=1}^L$.
3. If there is no difference between the edge-weighted Voronoi clusters $\{\widetilde{Q}_l'\}_{l=1}^L$ and $\{\widetilde{U}_l\}_{l=1}^L$, terminate the loop and return $(\{z_l\}_{l=1}^L;\{\widetilde{Q}_l'\}_{l=1}^L)$; otherwise, set $\widetilde{U}_l = \widetilde{Q}_l'$ for $l=1,\cdots,L$ and go to Step 1.

In Step 2 of Algorithm EWCVT-A, the generators will not be updated until one iteration completes even the point transfer occurs during the iteration. Therefore an accelerated version of Algorithm EWCVT-A can be naturally developed by updating the generators of the clusters immediately as long as point transfer occurs.

Algorithm EWCVT-B. Assume we are given an integer L and an arbitrary partition $\{U_l\}_{l=1}^L$ of discrete set $\mathscr{D}$. Let us compute the centroids $\{z_l\}_{l=1}^L$ of $\{U_l\}_{l=1}^L$ and take them as the generators.

1. For every $x \in \mathscr{D}$,
    - (a) evaluate the distance defined by Eq. (3.15) from the point x to all of the generators $\{z_l\}_{l=1}^L$;
    - (b) transfer the point x to the cluster whose generator is closest to it in the sense of the distance, say, from U_l to U_k;
    - (c) recompute the centroids $\bar{z}_\ell$ and $\bar{z}_m$ of the modified clusters U_ℓ and U_k, then set $z_\ell = \bar{z}_\ell$ and $z_k = \bar{z}_k$, respectively.
2. if no point is moved, terminate the loop and return $(\{z_l\}_{l=1}^L;\{U_l\}_{l=1}^L)$ (i.e., $\{\widetilde{Q}_l\}_{l=1}^L) = \{U_l\}_{l=1}^L)$); otherwise, go to Step 1.

Due to the compactness of $\mathscr{D}$, both EWCVT-A and EWCVT-B will be guaranteed to converge to a local minimizer of the EWVT energy. Because of real time modification of the configuration in each iteration, Algorithm CVT-B normally leads to a larger energy decrease in each step than that of Algorithm EWCVT-A. Therefore algorithm EWCVT-B is more efficient than EWCVT-A. The complexity of algorithm EWCVT-B can be represented by $\mathscr{O}(k \times r^2 \times L \times N)$ where k denotes the number of iterations, so it is more expensive than the CVT-based algorithms by r^2 times. But if r is small, the extra computational cost can be ignored. Moreover, because the edge-weighted energy variation will be quite small compared with that of the starting steps, we do not need to wait until the condition to exit the loops is strictly satisfied for both algorithms. In most practical cases, the EWVT energy of each iteration will be recorded. The algorithm will be terminated once the decrease of the EWVT energy is within some prescribed tolerance. In most of our numerical experiments, we exit the loop [14] if

$$\frac{|E_{i+1} - E_i|}{E_i} < L\%. \tag{3.18}$$

In fact a lot of accelerating strategies for determining CVTs [14, 22, 26] are available. Fortunately most of them are applicable to our EWCVT model without significant modification. The VT energy only needs to be replace with the EWVT energy in any of the accelerated strategies. As a matter of fact, the most likely point transfer will occur for the points which are closer to the segment boundary. Thus in each iteration in EWCVT-A and EWCVT-B, we only need to consider the test transfer of edge points . Extensive numerical experiments have shown this scheme is a satisfied strategy for acceleration purpose. Moreover, it is also a good option to implement a hybrid approach in which one starts with EWCVT-A or EWCVT-B by evaluating all of the points transfers and then switches to the scheme to evaluate the edge points only.

4. Experiments and discussions

Due to Theorem 2, if the neighborhood size r is small enough, our algorithms EWCVT-A and EWCVT-B are actually minimizing the weighted energy

$$\mathscr{E}_{VT} + \alpha\lambda r^3 L \tag{4.1}$$

a balance between VT energy $\mathscr{E}_{VT}$ and the edge length L with weight $\alpha\lambda r^3$, where constant α depends on the shape of neighborhood $\mathbb{N}_r(x)$. If $\lambda r^3 \to 0$ our model degenerates to the classic CVT method. Moreover increasing λ or r results in increment of the edge length weight. In the following subsections, we will illustrate the effect of these parameters through the numerical examples and the corresponding discussions. Note, without explicit declarations, we assume density function $\rho(\cdot)$ is uniform.

4.1. Effect of weighting factor λ

To compare with our EWCVT model, we first give the segmentation result produced by classic CVTs in Fig. 1. We will first show the effect of the weight parameter λ in Figs. 2, 3,

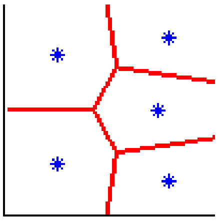

Figure 1: Segment a square into 5 clusters. Image size = 100 × 100. The segmentation is obtained by using classic CVT model, i.e., Algorithm CVT-B.

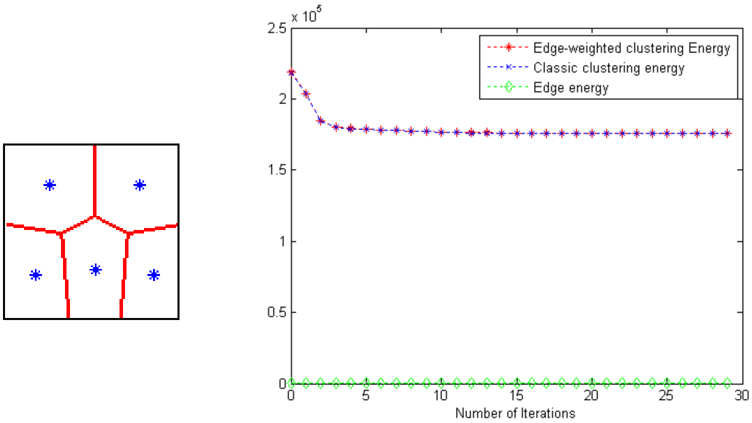

Figure 2: Segment a square into 5 clusters. Image size = 100×100. The segmentation is obtained by EWCVT-B. $r = 7$, $\lambda = 0.01$. Right image is the corresponding energy variation plot.

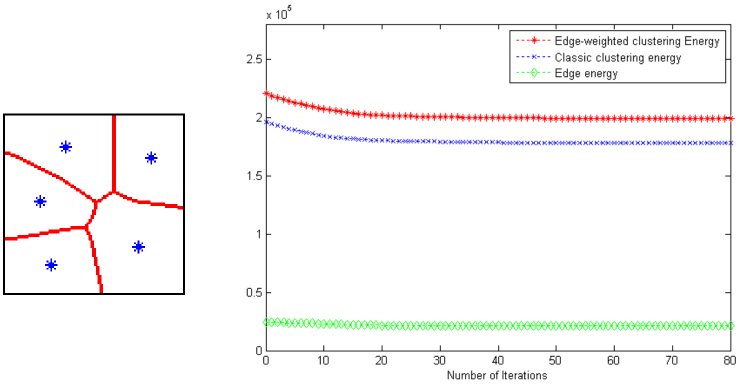

Figure 3: Segment a square into 5 clusters. Image size = 100×100. The segmentation is obtained by EWCVT-B. $r = 7$, $\lambda = 1$. The image in the right is the corresponding energy variation plot.

4 and 5. In Fig. 2, the value of λ is set to be 0.01. Note that, if $\lambda = 0$, our EWCVT model reduces to the classic CVT model since the edge energy is always equal to 0. Thus, if λ is quite small, we can expect the segmentation produced by EWCVT model to be pretty similar with that of classic CVT model since the classic VT clustering energy will dominate the iteration process. Besides, the energy variation plot of Fig. 2 further supports our assertions. Note, the edge energy is almost zero through the iteration process and the *edge-weighted clustering energy* coincides with the classic clustering energy very well which implies the dominate energy term is the classic clustering energy. Therefore, the final segmentation in Fig. 2 is almost exactly the same as that of classic CVTs if we simply rotate the segmented image in Fig. 2 counter-clock wise by 90°.

The segmentation in Fig. 3 is obtained by increasing λ to 1. Clearly, the resulting boundaries are not straight line segments anymore. Moreover, from the energy variation plot, the classic clustering energy is of the same magnitude with the edge energy. Thus, the iteration process is guided by the balance of the classic clustering energy and the edge

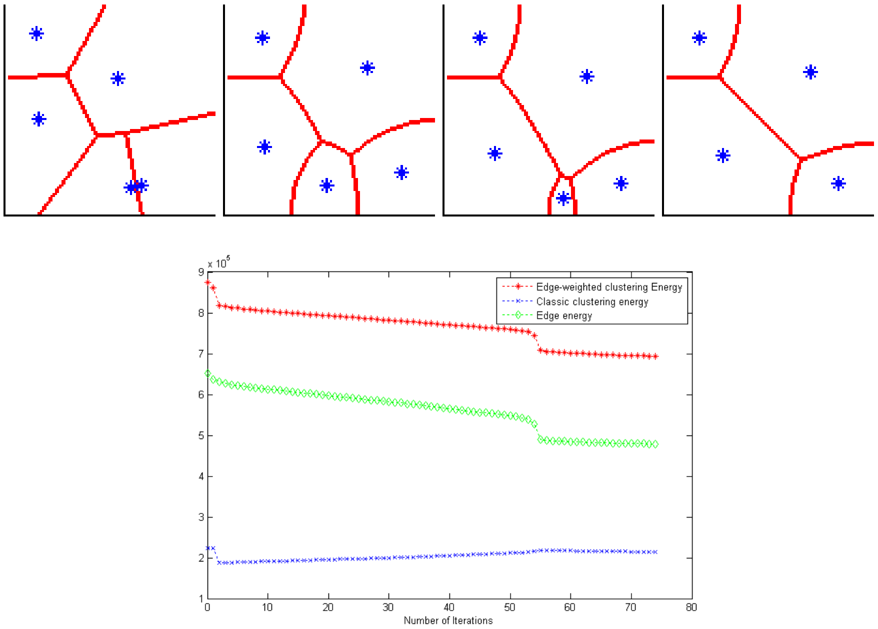

Figure 4: Segment a square into 5 clusters. Image size $= 100 \times 100$. The segmentation is obtained by EWCVT-B. For the first row: the first image is the Voronoi region corresponding to a set of randomly chosen generators. The other images are from iteration 17, 52, 74, respectively. $r = 7$, $\lambda = 5$. For the second row, the image is the corresponding energy variation plot.

energy. The segmentation does not look like the segmentation result obtained by classic CVTs.

If we keep increasing λ, something interesting happens. Let us look at Fig. 4. One of the clusters disappears! This unusual event well reveals the nature of our EWCVT model. Since the value of λ is so large, the edge energy will dominate the iteration process other than the classic clustering energy which is clearly illustrated by the energy variation plot. Thus, the *"length of boundaries"* will be dramatically decreased, with the cost that some clusters disappear. It is easy to see that the *"length of boundaries"* of the last image in Fig. 4 is much shorter than that of the initial image.

Furthermore, we give an "*extreme*" example in Fig. 5 in which the value of λ is set to be 10, 1000 times larger than that of Fig. 2. Finally, we have only one cluster winning out. No boundaries at all! The edge energy decreases to 0 at the end. In addition, the plot of *edge weighted clustering energy* is pretty similar as that of the edge energy which implies the edge energy dominates the iteration process.

Practically, examples shown in Figs. 4 and 5 may be viewed as *bad* results since one may not want any cluster disappears during the iteration. But due to the existence of the edge energy, some clusters with defects will naturally disappear. Those clusters introduce too much edge energy to the total energy and then will be eliminated during the process. This property is completely different with the classic CVT where the number of clusters is fixed in the whole process. Further discussion will be provided later.

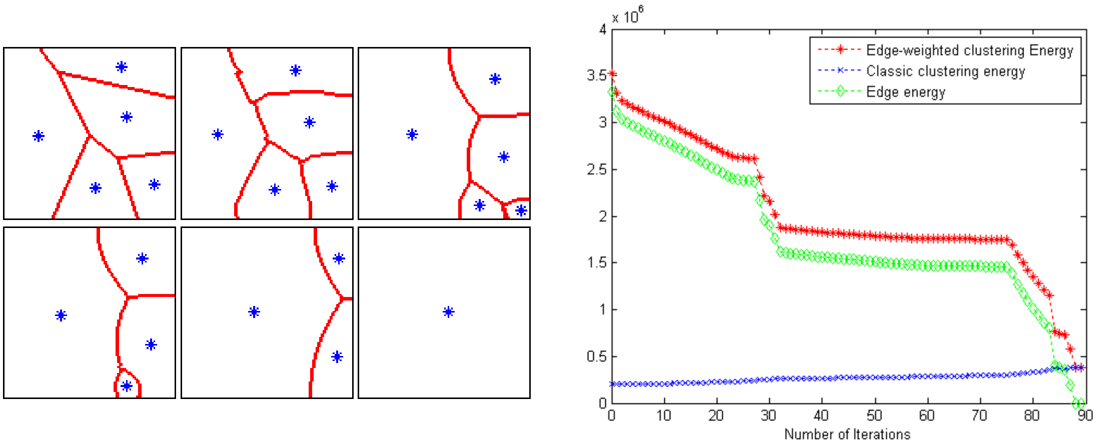

Figure 5: Segment a square into 5 clusters. Image size = 100×100. The segmentation is obtained by EWCVT-B. The first image is the Voronoi region corresponding to a set of randomly chosen generators. The other images are from iteration 1, 27, 30, 75 and 89 respectively. $r = 7$, $\lambda = 10$. The image in the last row is the corresponding energy variation plot.

4.2. Effect of the size of $\mathbb{N}_r$

The effects of r roughly fall into two parts: the accuracy of (4.1), and the weight of edge length. Theoretically speaking, a smaller value of r implies the edge energy approximates the edge length more accurately as stated in Theorem 2. Thus the effect of r is more complex than λ. In this subsection, we will show the effect of r through examples in Figs. 6, 7, 8 and 9. In all of these examples, we adopt the same initial clusters, i.e., one disk whose radius is 15, one little square whose side length is 10, and the left part of the domain.

In Fig. 6, the size of the neighborhood $\mathbb{N}_r$ is 3 which is smaller than both of the disk and the little square. The length of boundaries dramatically increases during the first several iteration steps. And the final result looks like the result produced by the classic CVT model. In fact, smaller $\mathbb{N}_r$ means smaller edge energy. The classic clustering energy is the main factor resulting in the final segmentation in Fig. 6.

If we set r equal to 5, $\mathbb{N}_r$ becomes comparable to both of the disk and the little square, but no larger than them. The edge energy and the classic clustering energy will have a balanced effect during the iteration. Therefore, the resulting segmentation in Fig. 7 is a balanced result of the edge energy and the classic clustering energy. The boundary length of the disk and the little square increases a little bit and the iteration stops very soon.

Then a question will naturally rise – What will happen when $\mathbb{N}_r$ is bigger than one of the clusters? Fig. 8 answers the question. In this example, we set r to be 7 such that $\mathbb{N}_r$ is bigger than the little square but smaller than the disk. The little square disappears! This is a reasonable consequence since larger $\mathbb{N}_r$ implies the edge energy will take more effect during the iteration. Once again, “larger" edge energy will result in shorter boundary.

Fig. 9 shows another “extreme" case when we set $\mathbb{N}_r$ much larger than both of the disk and little square. Again, we have no boundary left which is an expectable result.

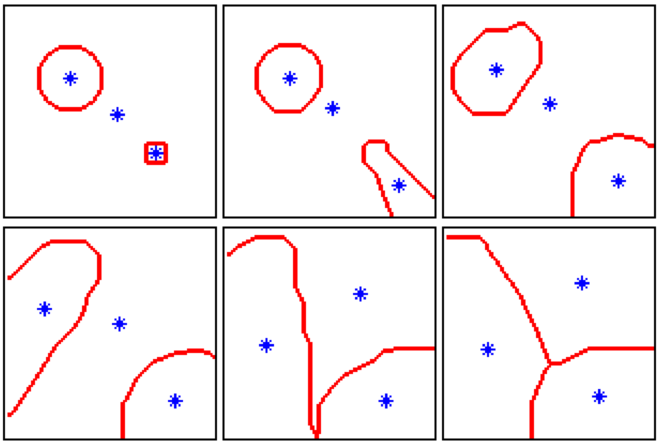

Figure 6: Segment a square into 3 clusters. Image size = 100×100. The segmentation is obtained by EWCVT-B. The first image is the initial clusters. The other images are from iteration 1, 12, 17, 28 and 41 respectively. $r=3$, $\lambda=3$.

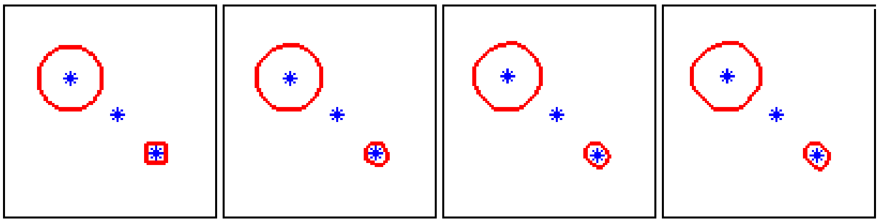

Figure 7: Segment a square into 3 clusters. Image size = 100×100. The segmentation is obtained by EWCVT-B. The first image is the initial clusters. The other images are from iteration 1, 2 and 3 respectively. $r=5$, $\lambda=3$.

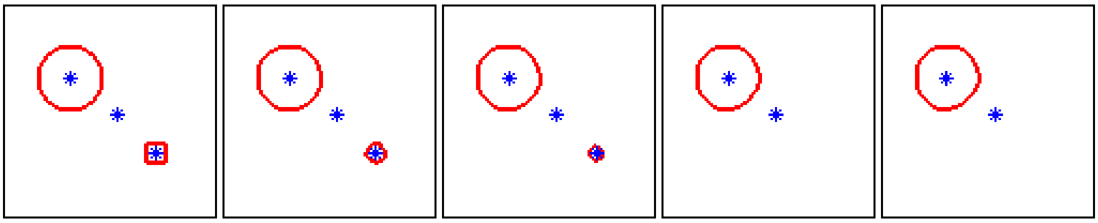

Figure 8: Segment a square into 3 clusters. Image size = 100×100. The segmentation is obtained by EWCVT-B. The first image is the initial clusters. The other images are from iteration 1, 2, 3 and 4 respectively. $r=7$, $\lambda=3$.

From (4.1), if we change λ and r simultaneously while keeping λr^3 constant we can expect same clustering results. Let us look at Figs. 3 and 10 in which the value of λ and r are carefully chosen such that λr^3 is roughly a constant. The edge energy plots in both of Figs. 3 and 10 are almost the same which demonstrate the edge energy is well kept as a constant. In addition, the final segmentation in Figs. 3 and 10 are almost the same as each other, i.e., they are just mirror symmetric to each other.

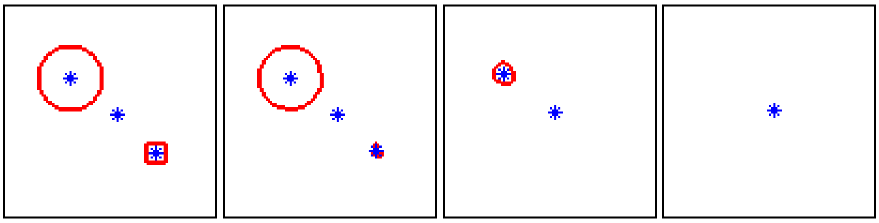

Figure 9: Segment a square into 3 clusters. Image size $= 100 \times 100$. The segmentation is obtained by EWCVT-B. The first image is the initial clusters. The other images are from iteration 1, 18 and 21 respectively. $r = 13$. $\lambda = 3$

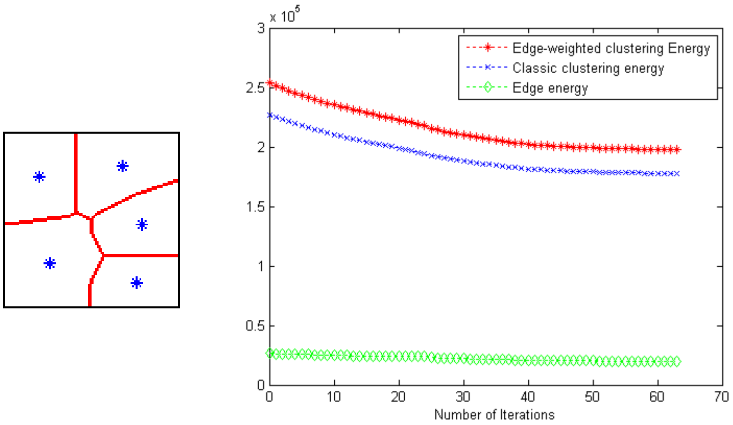

Figure 10: Segment a square into 5 clusters. Image size $= 100 \times 100$. The segmentation is obtained by EWCVT-B. $r = 5$, $\lambda = 2.7$. The image in the right is the energy variation plot.

4.3. Large number of clusters

The number of clusters in previous experiments is less than 10. For large number of clusters, the experiments show that the difference between EWCVT and the classic CVT becomes insignificant. Besides, it suggests that the classic CVT has already done a good job to make short boundaries in the case of large number of clusters.

A much more sophisticated example is given in Fig. 12. We divide the domain into 64 clusters with a nonuniform density function defined by

$$\rho(x,y) = A\exp\left(-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2}\right) + 1, \tag{4.2}$$

where $A = 100$, $(x_0, y_0) = (30, 30)$ and $\sigma_x = \sigma_y = 20$. Clearly, in the region of higher value of the density function, the clusters are much denser and smaller. In the rest part of the domain, the density function roughly equals to 1 which makes it roughly uniform. Therefore, the clusters in such region are much uniformly distributed and their size are roughly the same. One can compare it with the right image in Fig. 11, which is obtained by the classic CVT method. One can find that there is no much difference between the results of the EWCVT and the classic CVT.

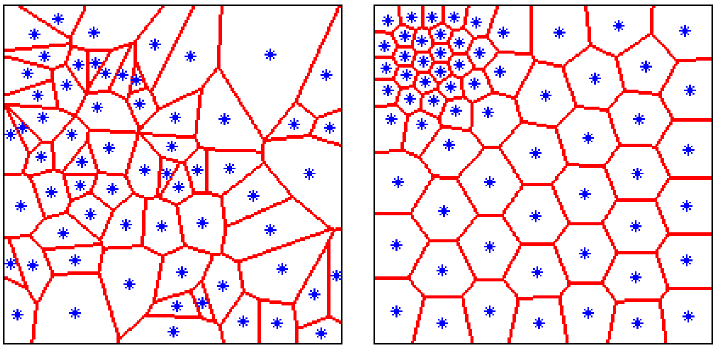

Figure 11: Segment a square into 64 clusters. The density function is not uniform. Image size = 200 × 200. The segmentation is obtained by CVT-B. The first image is the Voronoi region corresponding to a set of randomly chosen generators. The second image is the final segmentation. Totally, 195 iterations involved.

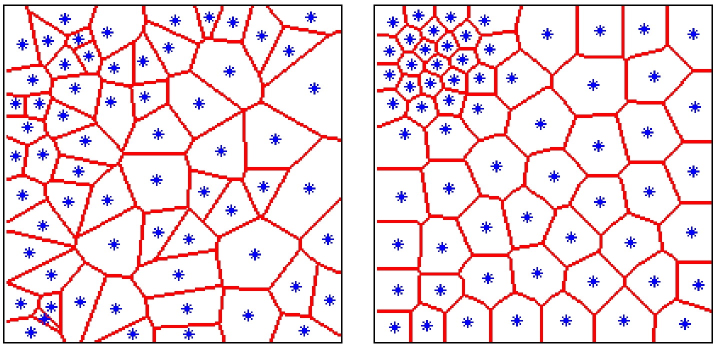

Figure 12: Segment a square into 64 clusters. The density function is not uniform. Image size = 200 × 200. The first image is the Voronoi region corresponding to a set of randomly chosen generators. The second image is the final segmentation obtained by EWCVT-B. Totally 263 iterations involved.

In Fig. 13, we adopt 500 clusters as the initial clusters. In the final segmentation, only 344 clusters left. Some clusters disappear due to the reason discussed in previous sections. It is pretty close to the classic CVT configuration with 344 clusters.

5. Concluding remarks

In this paper, we propose an Edge-Weighted Centroidal Voronoi Tessellation model for general domain clustering. Extensive examples are provided to illustrate the unique nature of our model compared with the classic CVT model. It preserves the beauty of the classic CVT as well as obtaining a shorter boundary. The EWCVT is especially useful for the case of small number of clusters. Our model can be used as a general data clustering technique. Although only 2D data sets are used in the experiments of this paper, the proposed methods are applicable to 3D or higher dimensional data sets. We expect more sophisticated applications of our methods in the disciplines such as data mining, signal processing, computer graphics and vision, etc., will be developed in the near future.

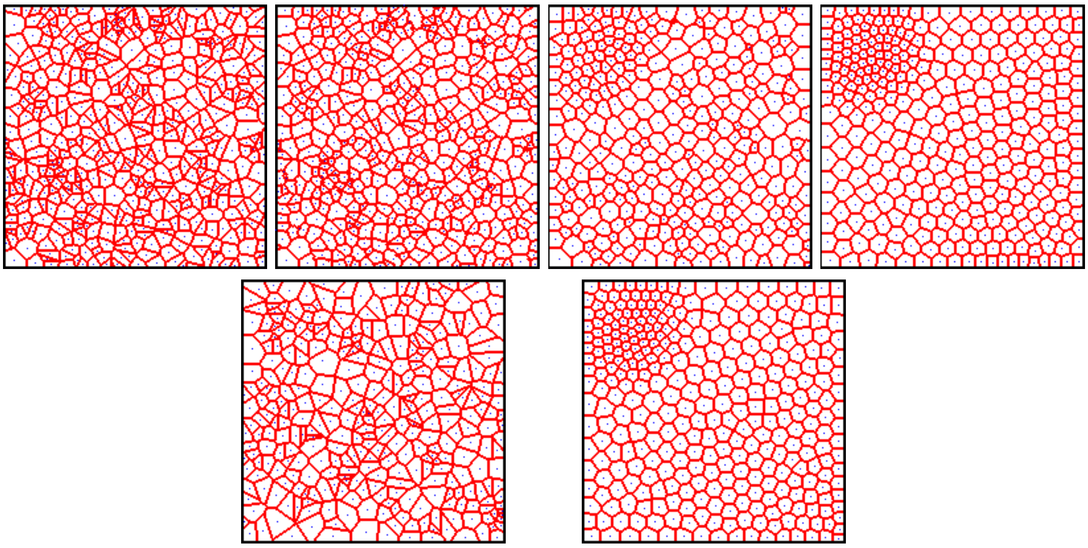

Figure 13: Segment a square into 500 clusters. The density function is not uniform. Image size $= 200 \times 200$. The segmentation is obtained by EWCVT-B. The first image is the Voronoi region corresponding to a set of randomly chosen generators. The other images of the first row are from iteration 1, 20, and 96 respectively. The bottom row are the initial and final clustering of the classic CVT respectively. $r = 7$, $\lambda = 10$.

Acknowledgments This research is supported in part by the U.S. National Science Foundation under grant number DMS-0913491.

References

[1] L. Antani, C. Delage, and P. Alliez, Mesh sizing with additively weighted Voronoi diagrams, *Meshing Roundtable Conference Proceedings*, 2007, pp. 335-346.

[2] K. Castleman, *Digital Image Processing*, Prentice Hall: Englewood Cliffs,1990.

[3] T. Chan, B. Sandberg, and L. Vese, Active contours without edges for vector-valued images, *J. Visual Comm. Image Rep.*, **11**, pp. 130-141, 1999.

[4] T. Chan, J. Shen, and L. Vese, Variational PDE models in image processing, *Notices of AMS*, **50**, pp. 14-26, 2003.

[5] T. Chan and L. Vese, An efficient variational multipase motion for the Mumford-Shah segmentation model, *Proceedings of 34th Asilomar Conference on Signals, Systems, and Computers*, **1**, pp. 490-494, 2000.

[6] T. Chan and L. Vese, Active contours withour edges, *IEEE Trans. Image Process*, **10**, pp. 266-277, 2001.

[7] T. Chan and L. Vese, Active contour and segmentation moderls using geometric PDE's for medical imaging, *Geometric Methods in Bio-Medical Image Processing*, R. Malladi (Ed.), Springer: Berlin, 2002, pp. 63-75.

[8] M. Cappellari and Y. Copin, Adaptive spatial binning of integral-field spectroscopic data using Voronoi tessellations, *Monthly Notices of the Royal Astronomical Society*, **342**, 2003, pp. 345-354.

[9] L. Cohen, E. Bardinet, and N. Ayache, Surface reconstruction using active contour models, *Proceedings of SPIE Conference on Geometric Methods in Computer Vision*, SPIE, Bellingham, 1993.

[10] Q. Du, V. Faber, and M. Gunzburger, Centroidal Voronoi tessellations: Applications and algorithms, *SIAM Review*, **41**, 1999, pp. 637-676.

[11] Q. Du and M. Gunzburger, Grid generation and optimization based on centroidal Voronoi tessellations, *Appl. Math. Comptut.*, **133**, 2002, pp. 591-607.

[12] Q. Du and M. Gunzburger, Model reduction by proper orthogonal decomposition coupled with centroidal Voronoi tessellation, *Proceedings of Fluids Engineering Division Summer Meeting*, FEDSM 2002-31051, ASME, 2002.

[13] Q. Du, M. Gunzburger, and L. Ju, Meshfree, probabilistic determination of points, sets and support regions for meshless computing, *Comput. Methods Appl. Mech. Engrg.*, **191**, 2002, pp. 1349-1366.

[14] Q. Du, M. Gunzburger, L. Ju, and X.-Q. Wang, Centroidal Voronoi tessellation algorithms for image compression, segmentation and multichannel restoration, *J. Math. Imaging Vision*, **24**, 2006, pp. 177-194.

[15] Q. Du, L. Ju, and L. Tian, Analysis of a mixed finite-volume discretization for fourth-order equations on general surfaces, *IMA J. Numer. Anal.*, **29**, pp. 376-403, 2009.

[16] Q. Du and D. Wang, Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations, *Inter. J. Numer. Methods Engrg.*, **56**, 2003, pp. 1355-1373.

[17] Q. Du and X. Wang, Tessellation and clustering by mixture models and their parallel implementations, *Proceedings of 4th SIAM Inter. Conf. Data Mining*, Orlando, FL, 2004, SIAM, 2004, pp. 257-268.

[18] Q. Du and X. Wang, Centroidal Voronoi tessellation based algorithms for vector fields visualization and segmentation, *IEEE Proceedings of Visualization'04*, Austin, Texas, 2004, pp. 43-50.

[19] J. Hartigan and M. Wong, Algorithm AS 136: A k-means clustering algorithm, *Appl. Stat.*, **28**, pp. 100-108, 1979.

[20] L. Ju, Conforming centroidal Voronoi Delaunay triangulation for quality mesh generation, *Inter. J. Numer. Anal. Model.*, **4**, pp. 531-547, 2007.

[21] L. Ju, M. Gunzburger, and W.-D. Zhao, Adaptive finite element methods for elliptic PDEs based on conforming centroidal Voronoi Delaunay triangulations, *SIAM J. Sci. Comput.*, **28**, pp. 2023-2053, 2006.

[22] 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.*, **24**, pp. 881-892, 2002.

[23] V. Ostromoukhov, C. Donohue, and P. Jodoin, Fast hierarchical importance sampling with blue noise properties, *ACM Trans. Graphics* **23**, pp. 488-495, 2004.

[24] A. Okabe, B. Boots, K. Sugihara, and S. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, Wiley, Chichester, 2000.

[25] S. Osher, Level set method: Applications to imaging science, CAM Report 02-43, UCLA, 2002.

[26] S. Phillips, Acceleration of k-means and related clustering algorithms, *ALENEX*, pp. 166-177, 2002.

[27] M. Samozino, M. Alexa, P. Alliez, and M. Yvinec, Reconstruction with Voronoi centered radial basis functions, *Proceedings of Sympos. Geom. Processing*, pp. 51-60, 2006.

[28] B. Sandberg, T. Chan, and L. Vese, A level-set and Gabor-based active contour algorithm for segmenting textured images, CAM Report 02-39, UCLA, 2002.

[29] D. SPARKS, Algorithm AS 58: Euclidean cluster analysis, *Appl. Stat.*, **22**, pp. 126-130, 1973.

[30] H. SPÄTH, *Cluster Dissection and Analysis, Theory, FORTRAN Programs, Examples*, Prentice Hall, Upper Saddle River, 1985.

[31] L. VESE AND T. CHAN, A multiphase level set framework for image segmentation using the Mumford and Shah model, Inter. J. Comput. Vis., **50**, pp. 271-293, 2002.

[32] J. WANG, L. JU, AND X. WANG, An Edge-Weighted Centridal Voronoi Tessellation Model For Image Segmentation, *IEEE Trans. Image Process.*, **18**, pp. 1844-1858, 2009.