Centroidal Voronoi tessellation based algorithms for vector fields visualization and segmentation

Qiang Du[*] Xiaoqiang Wang[†]

Abstract

A new method for the simplification and the visualization of vector fields is presented based on the notion of Centroidal Voronoi tessellations (CVT's). A CVT is a special Voronoi tessellation for which the generators of the Voronoi regions in the tessellation are also the centers of mass (or means) with respect to a prescribed density. A distance function in both the spatial and vector spaces is introduced to measure the similarity of the spatially distributed vector fields. Based on such a distance, vector fields are naturally clustered and their simplified representations are obtained. Our method combines simple geometric intuitions with the rigorously established optimality properties of the CVTs. It is simple to describe, easy to understand and implement. Numerical examples are also provided to illustrate the effectiveness and competitiveness of the CVT-based vector simplification and visualization methodology.

CR Categories: I.4.6 [Computing Methodologies]: Image Processing and Computer Vision—Segmentation; I.3.3 [Computing Methodologies]: Computer Graphics—Picture/Image Generation

Keywords: Flow Visualization, Vector Field, Simplification, Segmentation, Clustering, Centroidal Voronoi tessellation

1 Introduction

Large and complex data sets are being generated at an enormously fast speed with the advent of modern computing technology. Effective strategies for data mining that include the representation, simplification, characterization and manipulation of data become increasingly important.

The clustering and segmentation of spatially distributed data are important tools for data mining and information retrieval. The format of the spatially distributed data may vary, ranging from color intensity for images to various statistics for geographical regions. In abstract terms, the spatially distributed data set may be viewed as vector fields defined in a spatial domain. It has always been a computational challenge to visualize large sets of vector fields including those collected from various scientific and engineering disciplines. The vector fields we have in mind here include not only those typified by the velocities of wind on surfaces or flow currents in the ocean, but also other distributed statistics representing a much broader class of information.

Many vector data visualization methods have been developed in the past. They may be roughly divided into three kinds. The first kind of method uses arrows to help visualize the vector field, among which the most ubiquitous method for flow visualization uses hedgehogs. Although sub-sampling can be used to reduce the arrow count, it may not provide the best arrow distribution of the dataset. A recently proposed approach [Telea and vanWijk 1999] is to cluster the dataset by a hierarchical clustering tree using a bottom-up approach. In the beginning, every single point forms a cluster, then cluster-merging takes place according to a measure of the difference of their positions and orientations. However, this kind of method requires the input of many parameters so that the results may be very sensitive to the different choices.

Another kind of method is to display vector fields by texture synthesis. Line integral convolution [Cabral and Leedom 1993; Shen et al. 1996] and spot noise [deLeeuw and vanWijk 1995] are two well designed approaches in illustrating the direction of vector fields. Line integral convolution stretches a given image along the paths directed by a given vector fields to generate textures. Spot noise creates noise like texture by distributing many replicas of a shape. Texture based algorithms are very effective ways to display vector fields. But it can not display the directions of vector fields and it is very difficult to compress the vector field in an efficient way.

The third kind of method gaining popularity in recent years is the PDE based methods. It is natural to think that streamlines, streamtubes or flow ribbons can be used to express the flow once they can be conveniently calculated. In a recent work [Turk and Banks 1996], energy minimization is used to distribute the streamlines. Another example is the work in [Garcke et al. 2001] where PDE based phase field model is used to generate the continuous clustering of vector fields. The PDE based method has also been studied by many authors in the context of image segmentation and image inpainting. In general, the PDE based method has many great advantages such as the simple description of geometric quantities and the easy handling of topological events, both are important issues in the clustering of spatial statistics. Nevertheless, the good performance of the PDE based method often comes with much more time consuming computation and the results are often affected by the different scaling parameters used in the models.

Here, we propose a clustering/segmentation method for the vector fields based on the notion of Centroidal Voronoi tessellations (CVT's) [Du et al. 1999]. CVTs are optimal tessellations of a given domain and they also give rise to a global approach to cluster a domain into Voronoi regions.

Roughly speaking, for the spatially distributed vector fields of interests to us here, they can be thought as some vector bundles (or fibers) defined in a spatial domain. However, it is more natural and more convenient to treat such vector bundles and the spatial domain together as elements of a higher dimensional manifold equipped with a suitably defined distance (metric). Then, one may obtain, from the higher dimensional distance, a centroidal Voronoi tessellation that defines the clusters of the spatial domain. Then a lifting operation can be applied to obtain the vector representations of the

[*] Department of Mathematics, Pennsylvania State University, University Park, PA 16802 (qdu@math.psu.edu).

[†] Department of Mathematics, Pennsylvania State University, University Park, PA 16802 (wang@math.psu.edu).

IEEE Visualization 2004
October 10-15, Austin, Texas, USA
0-7803-8788-0/04/$20.00 ©2004 IEEE

vector fields distributed in each spatial clusters.

Our approach belongs to the class of methods to visualize vector field by arrows. For a given number of arrows, this method gives an ideal distribution of the arrows. The optimization properties of CVT's ensures that the results of our method are superior from a global perspective. This method can be easily generalized to some sophisticate algorithms. Meanwhile, the method is very fast, and easy to implement.

In section 2, the basic concept of CVTs is described. New vector field clustering algorithms are presented in section 3. Applications and numerical examples are given in section 4, together with some discussion on the performance of the algorithms. Some conclusions are made in section 5. Some technical details concerning the algorithms for CVTs and the mathematical background are given in the appendix (section 6).

2 Centroidal Voronoi Tessellations

Given an open set $\Omega \subseteq \mathbf{R}^N$, the set $\{V_i\}_{i=1}^k$ is called a tessellation of Ω if $V_i \cap V_j = \emptyset$ for $i \neq j$ and $\bigcup_{i=1}^k \overline{V}_i = \overline{\Omega}$ where $\overline{V}_i$ and $\overline{\Omega}$ denote the closures of V_i and Ω. Let d denote a distance defined on $\mathbf{R}^N$. Given points $\{z_i\}_{i=1}^k$ belonging to $\overline{\Omega}$, the Voronoi region (or cluster) $\widehat{V}_i$ corresponding to the point z_i is defined by

$$\widehat{V}_i = \{x \in \Omega | d(x,z_i) < d(x,z_j) \text{ for } j = 1,...,k, j \neq i\} \ .$$

The points $\{z_i\}_{i=1}^k$ are called generators. The set $\{\widehat{V}_i\}_{i=1}^k$ is a Voronoi tessellation or Voronoi diagram of Ω, and each $\widehat{V}_i$ is referred to as the Voronoi region corresponding to z_i. In [Du and Wang 2004a], the above definitions have been generalized to allow the use of a one-sided distance function, that is, the Voronoi region $\widehat{V}_i$ is defined by

$$\widehat{V}_i = \{x \in \Omega | d_x(x,z_i) < d_x(x,z_j) \text{ for } j = 1,...,k, j \neq i\}$$

where $d_x(x,y)$ is a distance function defined according to some local Riemannian metric at the point x.

Given a region $V \subseteq \mathbf{R}^N$, a one-sided distance function $d_x(x,\cdot)$, the mass centroid z^* of V is a unique point in $\mathbf{R}^N$ to minimize the energy defined by the summation of distance square:

$$E(z,V) = \int_V d_x^2(x,z)\, dx \ . \tag{1}$$

The conventional mass center can be defined as the minimizer of $E(\cdot,V)$ with $d_x^2(x,y) = d^2(x,y)\rho(x)$ where ρ is the density function and d is the standard Euclidean distance. In case $\rho = 1$ is a constant density, then z^* is just the mean of each cluster, i.e.

$$z^* = \frac{1}{|V|}\int_V x\, dx \ . \tag{2}$$

Thus given k points $\{z_i\}_{i=1}^k$, we have the Voronoi tessellation formed by the Voronoi regions $\{\widehat{V}_i\}_{i=1}^k$, and given k regions $\{\widehat{V}_i\}_{i=1}^k$, we have their mass centroids $\{z_i^*\}_{i=1}^k$. A Voronoi tessellation is a Centroidal Voronoi tessellation if the generators are themselves the mass centroids of the respective Voronoi regions.

For any set of k points $\{z_i\}_{i=1}^k$ and a tessellation made of k regions $\{V_i\}_{i=1}^k$, we define the total energy by

$$E(\{z_i\}_{i=1}^k, \{V_i\}_{i=1}^k) = \sum_{i=1}^k E(z_i,V_i) = \sum_{i=1}^k \int_{V_i} d_x^2(x,z_i)\, dx \ . \tag{3}$$

The CVTs enjoy an optimality property that can be rigorous proved (see [Du and Wang 2004a]):

Theorem. The minimizer of the total energy (3) leads to a CVT.

For more detailed discussions, we refer to [Du et al. 1999; Du and Gunzburger 2002; Du et al. 2003; Du and Wang 2002]. If the distance is truly a one-sided distance function, then the corresponding CVT should in principle be called an anisotropic CVT as defined in [Du and Wang 2004a].

CVTs are very special and elegant tessellations. In figure 1, a 2D example using the standard Euclidean distance and the constant density is shown.

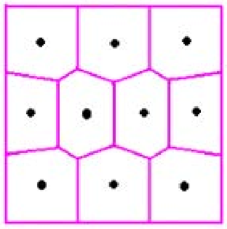

Figure 1: Illustration of a 2d CVT with 10 clusters.

Two algorithms (Algorithms 1 and 2) for generating CVTs are discussed in the appendix along with suggestions for their improvement. Based these algorithms, new vector field clustering algorithms can be developed using a proper definition of the distance between vectors in a flow field.

3 Vector Fields Visualization

3.1 Vector fields clustering

First some notations are given here. Let a vector field V be defined on a domain $\Omega \subseteq \mathbf{R}^N$, such that for every point $x \in \Omega$, $V(x) \in \mathbf{R}^M$. We use $|\cdot|$ to denote the standard vector norm. Viewing $(x, V(x))$ as a point in a higher dimensional space $\mathbf{R}^{N+M}$, we can just think that the region of each cluster, a subset of Ω, is the projection of a subset defined in $\mathbf{R}^{N+M}$ back to $\mathbf{R}^N$. Let a point p, denoted as (x_p, y_p) where $y_p = V(x_p)$, be called degenerate if $y_p = 0$. As nearby degenerate points can be grouped into the same cluster as their closest non-degenerate points, all points may be viewed as non-degenerate without loss of generality.

Given a positive scaling constant w, define the (one-sided) distance between $p = (x_p, y_p)$ and $m = (x_m, y_m)$ as

$$d_p(p,m) = \sqrt{|y_p|^2 - |y_p| y_p \cdot y_m + w|y_p|^2|x_p - x_m|^2} \ . \tag{4}$$

(For more detailed discussions of d_p, see the appendix).

Then, given a set of k generators $\{m_i\}_{i=1}^k$ under the constraint $|y_{m_i}| = 1$, the Voronoi regions $\{\widehat{C}_i\}$ corresponding to the point $\{m_i\}$ are defined by

$$\widehat{C}_i = \{x_p \in \Omega | d_p(p,m_i) < d_p(p,m_j) \text{ for } j = 1,...,k, j \neq i\} \ . \tag{5}$$

It is obvious that $\widehat{C}_i \cap \widehat{C}_j = \emptyset$ if $i \neq j$. For some p that satisfying $d_p(p,m_i) = d_p(p,m_j)$ for two distinct generators $m_i \neq m_j$, we then assign p to the Voronoi region $\widehat{C}_i$ if $|x_p - x_{m_i}| < |x_p - x_{m_j}|$.

Now, some discussions on the cluster centers are in order. Given a cluster C, the centroid m^* is obtained as the minimizer of the energy defined in (1). Using the definition of d_p, we have

$$E(m,C) = \int_C |y_p|^2 - |y_p|y_p \cdot y_m + w|y_p|^2|x_p - x_m|^2 \, dx_p \,.$$

By minimizing this energy, the following two algorithms come from the Algorithm 1 and Algorithm 2.

Algorithm 3: Vector field CVT clustering. Given a positive integer k, a weight w and a domain Ω, choose any k points $\{m_i = (x_i, y_i)\}_{i=1}^k$ and determine the associated Voronoi clustering $\{C_i\}_{i=1}^k$.

1. For each cluster C_i, $i = 1, \dots, k$, determine the centroids by

$$\bar{x}_i = \frac{\int_{C_i} |V(x)|^2 x \, dx}{\int_{C_i} |V(x)|^2 \, dx} \,, \quad \bar{y}_i = \frac{\int_{C_i} |V(x)|V(x) \, dx_p}{|\int_{C_i} |V(x)|V(x) \, dx|} \,. \qquad (6)$$

2. Determine the Voronoi clusters associated with $\{(\bar{x}_i, \bar{y}_i)\}_{i=1}^k$.

3. If the Voronoi clusters corresponding to $\{(\bar{x}_i, \bar{y}_i)\}_{i=1}^k$ and $\{(x_i, y_i)\}_{i=1}^k$ are the same, or some tolerance condition is met, exit the loop; otherwise, set $(x_i, y_i) = (\bar{x}_i, \bar{y}_i)$ for $i = 1, \dots k$, determine the new Voronoi clustering and return to Step 1.

Algorithm 4: Vector field CVT clustering. Given an integer k, a weight w and a domain Ω, choose any k points $\{m_i = (x_i, y_i)\}_{i=1}^k$ and determine the associated Voronoi clustering $\{C_i\}_{i=1}^k$.

1. For every point $p = (x, y)$,
   (a) evaluate all the distances $d_p(p, m_i)$ for $i = 1, \dots, k$;
   (b) For the shortest distance $d_p(p, m_t)$,
      i. move the point x from old group s into group t;
      ii. replace the centroid m_s and m_t by the means of the newly modified clusters V_s and V_t respectively via the formula (6).
2. Exit when some tolerance is met; otherwise, go to Step 1.

The tolerance choices are similar to that in Algorithms 1 and 2 given in the appendix. A derivation of (6) is also given there.

To end this section, we note that the above clustering relies on the input of the number of clusters k. The practical choice of k will be discussed later.

3.2 Non-uniformly distributed fields clustering

Non-uniformly distributed vector field is a vector field whose vector density is non-uniformly distributed. In the real world, we can always met this kind of vector fields. For example, the flow in the atmosphere is non-uniformly distributed because the densities of the air are different at different heights. Another example is the crowd out from a cinema where every people is associated with a vector and the density of this vector field is very high at the door or inside of the cinema, and it comes to be thinner and thinner at the place further and further from the door.

Algorithms 3 and 4 can be generalized to the non-uniformly distributed vector fields clustering with a density distribution $\rho(x_p)$. The energy for a cluster C with a centroid m is given by

$$\begin{aligned} E(m,C) &= \int_C \rho(x_p) d_{x_p}^2(x_p, m) \, dx_p \\ &= \int_C \rho(x_p)(|y_p|^2 - |y_p|y_p \cdot y_m + w|y_p|^2|x_p - x_m|^2) \, dx_p \,. \end{aligned}$$

Algorithm 5: Non-uniformly distributed vector field CVT clustering. Given a density distribution ρ, a positive integer k, a weight w and a domain Ω, choose any k points $\{m_i = (x_i, y_i)\}_{i=1}^k$ and determine the associated Voronoi clustering $\{C_i\}_{i=1}^k$.

1. For each cluster C_i, $i = 1, \dots, k$, determine the centroids by

$$\bar{x}_i = \frac{\int_{C_i} \rho(x)|V(x)|^2 x \, dx}{\int_{C_i} \rho(x)|V(x)|^2 \, dx} \,, \; \bar{y}_i = \frac{\int_{C_i} \rho(x)|V(x)|V(x) \, dx_p}{|\int_{C_i} \rho(x)|V(x)|V(x) \, dx|} \,. \quad (7)$$

2. Determine the Voronoi clusters associated with $\{(\bar{x}_i, \bar{y}_i)\}_{i=1}^k$.

3. If the Voronoi clusterings corresponding to $\{(\bar{x}_i, \bar{y}_i)\}_{i=1}^k$ and $\{(x_i, y_i)\}_{i=1}^k$ are the same, or any tolerance condition met, exit the loop; otherwise, set $(x_i, y_i) = (\bar{x}_i, \bar{y}_i)$ for $i = 1, \dots k$, find the new Voronoi clustering and return to Step 1.

Algorithm 4 can also be easily generalized to non-uniformly distributed vector fields. We omit the details.

The examples of non-uniformly distributed vector fields clustering is given in section 4.

3.3 Visualization strategies

The algorithms for the vector field clustering are given in section 3.1 and section 3.2, with each cluster represented by a unit vector y_m at point x_m. Some visualization strategies can be taken into the visualization process.

First, we can set the length of the representation vector as the average of the lengths of all the vector in the same Voronoi region C_i. For instance, we can take

$$L_{y_i} = \frac{1}{|C_i|} \int_{C_i} |V(x)| \, dx \,, \qquad (8)$$

or

$$L_{y_i} = \frac{1}{|C_i|} (\int_{C_i} |V(x)|^2 \, dx)^{1/2}. \qquad (9)$$

And define the new representation vector $z_i = L_{y_i} y_i$.

For a non-uniformly distributed vector field, after the clustering, we can also set the length of the representation vector as

$$L_{y_i} = \frac{(\int_{C_i} \rho(x)|V(x)|^2 \, dx)^{1/2}}{\int_{C_i} \rho(x) \, dx} \,, \qquad (10)$$

which is a generalized form of (9).

Second, the color of the representation vector can be used to represent the vector variance or the energy of each cluster. The vector variance is defined as

$$Var(C_i) = \frac{\int_{C_i} |y_p|^2 - |y_p|y_p \cdot y_{m_i} \, dx_p}{|C_i|} \,, \qquad (11)$$

or, for non-uniformly distributed vector field,

$$Var(C_i) = \frac{\int_{C_i} (|y_p|^2 - |y_p|y_p \cdot y_{m_i})\rho(x) \, dx_p}{\int_{C_i} \rho(x) \, dx} \,. \qquad (12)$$

Alternatively, to display the vector field, instead of plain arrows, curved arrows may also be used which are computed along streamlines from every cluster's center x_i, with their length and the color determined in the same way as for the plain arrows.

Finally, all the above strategies may be combined for different problems. Some illustrations will be given in the next section.

4 Applications and Numerical Examples

This section gives some 2D and 3D vector field visualization examples. We use some 2D examples to illustrate the parameters selection, such as the cluster number k and weight w, and the performance of our algorithms. A 3D example is given in the end.

Though our theory is applicable to much more general settings, as an illustration, the two dimensional vector fields to which we apply the CVT based vector fields clustering algorithms mostly distributed in a two dimensional square $\Omega = [-1,1]^2 \subseteq \mathbf{R}^2$, and the three dimensional vector fields are distributed in a three dimensional square $\Omega = [-1,1]^3 \subseteq \mathbf{R}^3$.

Figure 3 shows the vector CVT for the vector field with a center vortex depicted in the Figure 2 on a 300×300 Cartesian mesh. 6 clusters are used. The red color indicates higher variance while the blue color indicates less variance. Another example is given in figure 4 where a vector field with non-uniform lengths is clustered into 15 clusters.

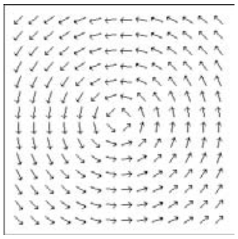

Figure 2: A vector field for a degree one vortex.

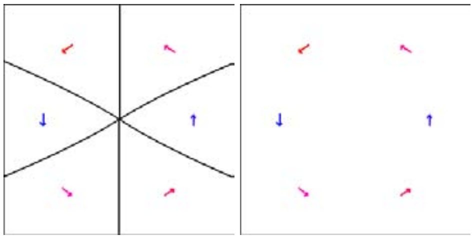

Figure 3: A clustering in 6 clusters (cluster boundaries are shown on the left).

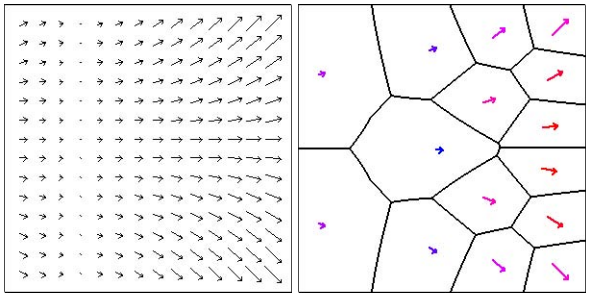

Figure 4: A vector field and the clustering with 15 clusters.

The performance of the algorithm in general depends on the choice of the parameters used in our algorithms, including the number of clusters k and the weight parameter w, and the initial distribution of the generators. For our numerical examples, we have found that the uniformly sampled and randomly distributed initial generators both lead to satisfactory convergent results.

As for the parameter tuning, it is obvious that the number of clusters k is very important in this process. If we further take 30 clusters for the vector field in figure 4, the result is showed in figure 5.

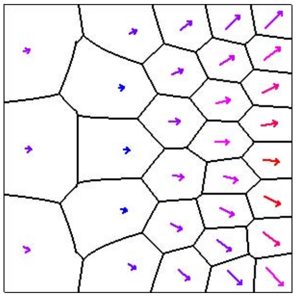

Figure 5: Clustering of the figure in Fig.4 into 30 clusters.

In general, the larger the k is, the more details the simplified graph shows, accompanied by more computations involved and smaller compression ratios. On the other hand, if the k is too small, important details of the original field may be lost. Thus, automatically choosing a good k is very important. Obviously, for a good k, the vector at the centroid should well represent the flow directions in each cluster, that is, the angle θ between each vector $V(x)$ and its centroid vector y_m should be small. We thus choose the following quantity to measure the goodness of k.

$$G(k) = \frac{1}{|\Omega|} \int_{\Omega} \frac{V(x) \cdot y_m(x)}{|V(x)|} \, dx \qquad (13)$$

where $y_m(x)$ is the vector centroid of the cluster x belongs. It is obvious that $|G(k)| \leq 1$, and the closer $G(k)$ is to to 1, the more details the simplified graph gives.

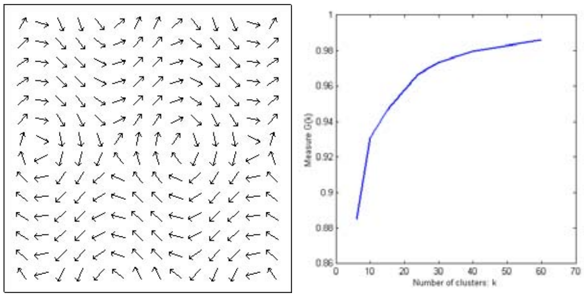

Figure 6: The original field (left) and the measure curve in k (right).

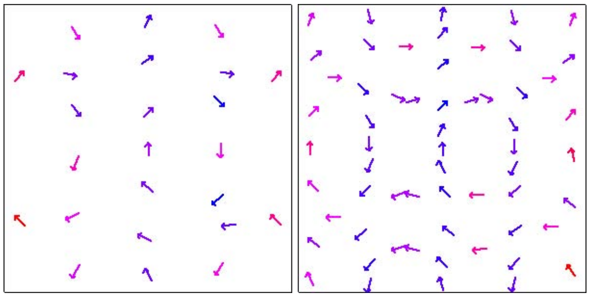

Figure 7: Visualized with 24 (left) and 60 (right) clusters.

Based on our experience, it is better to choose k such that $G(k)$ is above 0.98 and the slope of $G(k)$ is getting small. The right picture of figure 6 shows the graph of $G(k)$ for a vector field (left picture) having vortices. Figure 7 shows the visualization of this vector field with 24 clusters and 60 clusters with $G(24) = 0.9667$ and $G(60) = 0.9861$ respectively with the 60 clusters giving a clear presentation of the vortices. In practice, an adaptive estimations of k may also be developed. Relevant discussions in the context of CVT clusterings have been studied in [Du and Wang 2004b].

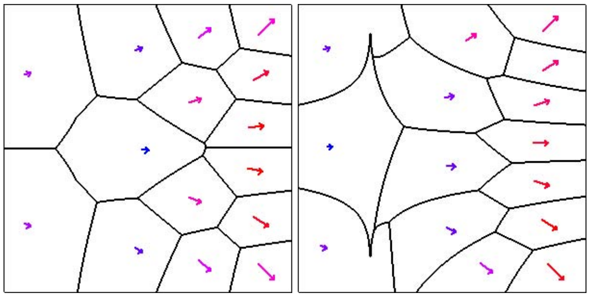

Figure 8: Visualized with weights $w = 0.5$ (left) and $w = 0.1$ (right).

We now discuss the choice of the weight w in the distance formula (note that $w = 1/L^2$ is sufficient for most applications, where L is the size/diameter of the domain). A smaller w decreases the weight in space, making the simplified graph much more likely to distribute the clusters along the flow directions represented by the vector field. However, for some fields, a smaller w may result in very irregular or even disconnected clusters, thus, it would then be more appropriate to choose a larger w to regularize the clusters. Figure 8 shows two examples of 15 clusters with different values of w for the field in figure 4.

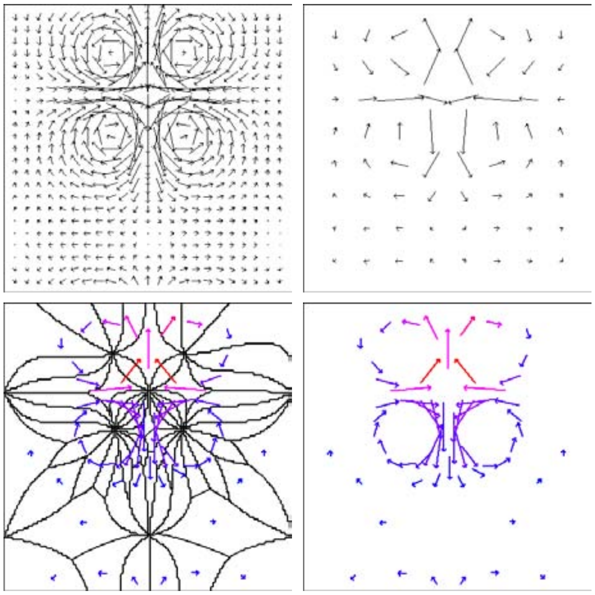

Figure 9: Top: original fluid fields (left) and 8×8 uniformly sampled arrows (right); bottom: clustering in 60 clusters (left) and the 60-arrows visualization (right) based on our method.

For a more practical example, the result of our method applied to a fluid vector field is given in Figure 9. The field is obtained from the simulation of the deformation of two bubbles in a Newtonian fluid [Du et al. 2005]. Clearly, an efficient visualization of the fluid field is obtained by our method which reveals the main characteristics of the flow field.

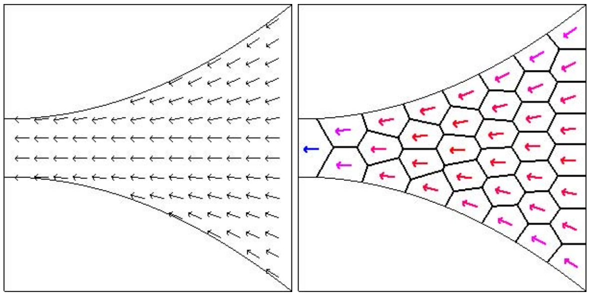

Figure 10: Vector field of ants going through a gate; the right graph is the simplified graph by 36 arrows .

We now turn to some non-uniformly distributed vector fields. Figure 10 shows a vector fields of many ants moving through a channel of different widths with the same speed. The right graph is the simplified presentation with 36 vectors. Figure 10 reveals only the vector fields, but not the distribution of the ants. As the ants move into the narrower region, they get more crowded. Assuming that they move with a constant speed, the density ρ is then inversely proportional to the width of the channel. Thus, the distribution of arrows like that in figure 10 does not provide a realistic view of the ants distribution. Figure 11 gives the graph by algorithm 5 using the density ρ with the clustering showing on the right, illustrating a good balance of the vector simplification between the flow directions and the underline density distribution.

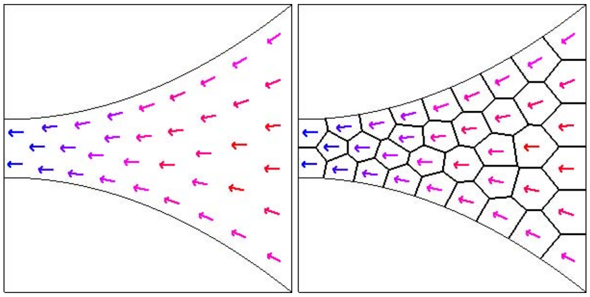

Figure 11: Visualization of the non-uniformly distributed vector fields; the right one is the clustering.

We can also use 3D curved vectors for 2D and 3D visualization. Figure 12 shows the result for the vector field in Figure 9. Clearly, the curved vectors reveal more details of the vector field, and they are more efficient for the vector visualization.

Finally, Figure 13 shows the visualization of a 3D vector field formed by two vortices pointing to different directions.

In all of the 2D experiments, a 300×300 grid is used except figure 9 which uses a 384×384 grid. The 3D experiment uses a $60 \times 60 \times 60$ grid. For the non-optimized algorithm 3, most of these examples can be done in less than a minute on a Pentium M 1.3GHz laptop except ones with more than 60 clusters in figures 7, 9 and 13 which take no more than 5 minutes. The most time consuming step is the step 2 in algorithm 3 and the step 1.(b) for algorithm 4. We note that the total operations needed is $O(kn)$ for a fixed number of iterations, where k is the number of clusters, n is the number of total grid points.

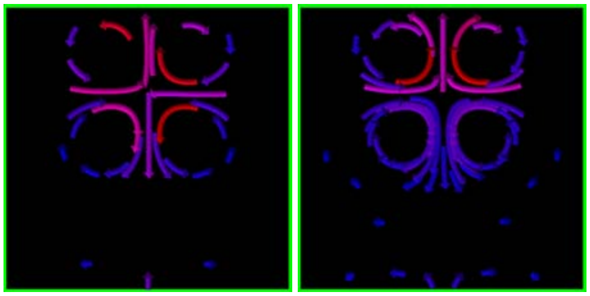

Figure 12: Visualization of the field in Figure 9: 30 (left) and 60 (right) curved arrows.

5 Conclusion

We have presented a new vector field clustering approach which is based on the technique of Centroidal Voronoi tessellations. After assigning a well defined distance, the vector field has a very natural Centroidal Voronoi tessellations. Based on the Centroidal Voronoi tessellation of the vector field, we can give simplified and elegant visualization of the vector field.

Comparing to other visualization methods, the merits of the new method are its global view, easy understanding and its efficient implementation and realization. It can be used for both uniformly distributed vector fields (say at lattice points) and non-uniformly scattered vector fields.

The set up of the vector field clustering considered is limited to distributed vector fields defined in a subset of the Euclidean space, however, it is obvious that our approach has a lot of potential to be generalized. One of such generalizations can be used to cluster and segment vector fields defined on complex surfaces and manifolds. In some of our earlier works [Du et al. 2003], we have considered the constrained Centroidal Voronoi tessellations which are the generalizations of Centroidal Voronoi tessellations in the Euclidean space to surfaces and manifolds. In the same spirit, we can extend our CVT based clustering/segmentation to vector fields defined on a compact and continuous surface/manifold $\mathbf{S} \subset \mathbf{R}^N$ given by $\mathbf{S} = \{x \in \mathbf{R}^N \ : \ g_0(x) = 0 \text{ and } g_j(x) \leq 0 \text{ for } j = 1, \ldots, m\}$ for some continuous functions g_0 and $\{g_j\}_{j=1}^m$, and given a set of vector fields $\{\vec{v}_i\}_{i=1}^k$, defined on a point set $\{z_i\}_{i=1}^k \in \mathbf{S}$, one may define their corresponding Voronoi regions on $\mathbf{S}$ by

$$V_i = \{x \in \mathbf{S} \ : \ d_x(x, z_i) < d_x(x, z_j) \quad \text{for } j = 1, \ldots, k, \ j \neq i\}$$

for $i = 1, \ldots, k$. Notice that the one-sided distance is independent of the surface $\mathbf{S}$, thus, V_i's are simply the restrictions of the Voronoi regions (5) defined in a subset of $\mathbf{R}^N$ onto $\mathbf{S}$.

Since the mass centroids $\{z_i^*\}_{i=1}^k$ of $\{V_i\}_{i=1}^k$ as defined by (2) do not in general belong to $\mathbf{S}$, then, a constrained mass centroid z_i^c on the surface is defined as a solution of the following problem:

$$\min_{z \in \mathbf{S}} E_i(z), \qquad \text{where} \qquad E_i(z) = \int_{V_i} d_x(x, z)^2 \, dx.$$

The integral over $\{V_i\}$ is understood as standard surface integration on $\mathbf{S}$. Then, we can get the *constrained centroidal Voronoi tessellation* (CCVT) for the vector fields defined on the surface $\mathbf{S}$ if and only if the points $\{z_i\}_{i=1}^k$ which serve as the generators associated with the Voronoi regions $\{V_i\}_{i=1}^k$ are the constrained mass centroids of those regions. Thus, we expect that the notion of constrained CVT can also lead us to new clustering and segmentation methods for vector fields defined on manifolds and surfaces.

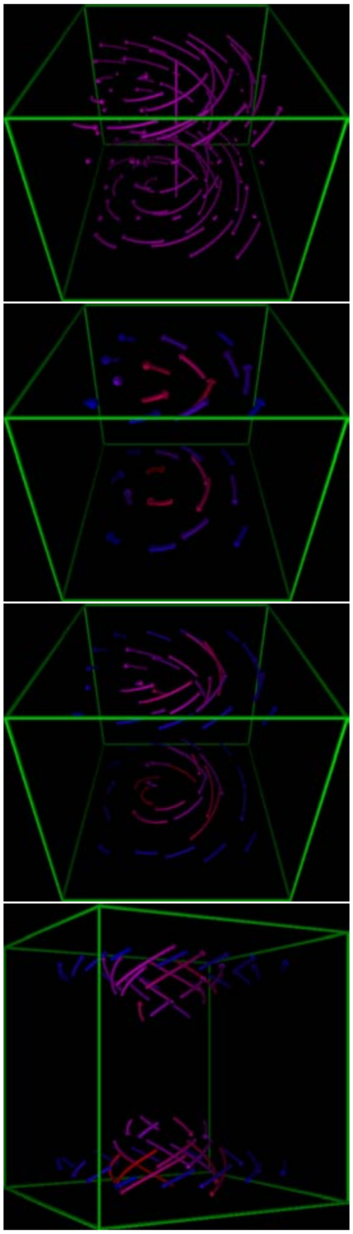

Figure 13: The 5 × 5 × 5 uniformly sampled curved arrows (top) of the original field, visualizations with 30 (second) and 60 curved arrows (the last two, with different viewing angles).

It should be noted that in [Du et al. 2005], we have also developed a theory for retrieving some useful topological information of the deformable interface based on the phase field description. As for most of applications, getting correct statistics is very important, we anticipate that such topological information retrieval tools may also be useful to vector field simplification and visualization.

6 Appendix

Algorithms for CVTs. To construct the CVT's with a given positive integer k and a domain Ω, a set of points $\{z_i\}_{i=1}^k$ are to be determined that are at the same time the generators of a Voronoi clustering of the regions and the mass centroids of the associated clusters. The following algorithm can be used to construct CVT's; see, e.g., [Hartigan and Wong 1979; Späth 1985; Sparks 1973] for details.

Algorithm 1. Given a positive integer k and a domain Ω, choose an initial distribution of k distinct points $\{z_i\}_{i=1}^k$ in Ω and determine the associated Voronoi clustering $\{V_i\}_{i=1}^k$.

```
1. For each cluster $V_i$, $i = 1, \dots, k$, determine the centroids, or the cluster means $\{\bar{z}_i\}_{i=1}^k$, in the Euclidean distance case (or for more general distances).
2. Determine the Voronoi clustering (or anisotropic Voronoi clustering [Du and Wang 2004a]) associated with $\{\bar{z}_i\}_{i=1}^k$.
3. If the (possibly anisotropic) Voronoi clusterings corresponding to $\{\bar{z}_i\}_{i=1}^k$ and $\{z_i\}_{i=1}^k$ are the same, or some tolerance condition is met, exit the loop; otherwise, set $z_i = \bar{z}_i$ for $i = 1, \dots k$, determine the new Voronoi clustering and return to Step 1.
```

It is easy to see that steps 1 and 2 result in a decrease in the energy defined in (1), which guarantees the convergence to a local minimizer of the energy. For a discrete 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 may calculate the energy of the current configuration and terminate the construction steps when the energy is within some prescribed tolerance of the energy of the previous configuration. Since the Voronoi clusterings and their centroids are uniquely determined by each other, another tolerance in the last step is to calculate the distance between $\{\bar{z}_i\}_{i=1}^k$ and $\{z_i\}_{i=1}^k$ instead of comparing their Voronoi regions.

Algorithm 1 does not transfer the elements 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 1 that takes into account the changes in cluster means as soon as they are determined.

Algorithm 2. Given a positive integer k and a domain Ω, choose an initial distribution of k points $\{z_i\}_{i=1}^k$ in Ω and determine the associated Voronoi clustering $\{V_i\}_{i=1}^k$.

```
1. For every point $x$,
   (a) evaluate all the distances $d_x(x, z_i)$ for $i = 1, \dots, k$;
   (b) For the shortest distance $d_x(x, z_t)$,
       i. move the point $x$ from old group $s$ into group $t$;
       ii. replace the centroid $z_s$ and $z_t$ by the means of the newly modified clusters $V_s$ and $V_t$, respectively.
2. Exit when some tolerance is met; otherwise, go to Step 1.
```

Algorithms 1 and 2 both result in a k-means clustering or a CVT tessellation corresponding to the CVT-energy (1). Numerical experiments indicate that Algorithm 2 is often more reliable than Algorithm 1 even though the former is more costly per iteration since one must examine the effect of each potential transfer on the energy. The gain lies in the fact that an iteration of Algorithm 2 leads to a larger decrease in the energy than that of Algorithm 1, and thus requires a much smaller number of iterations. A hybrid approach is also possible in which one starts with the Algorithm 1 and then switches to Algorithm 2. Presumably, after several iterations of Algorithm 1, only a very few of the more expensive iterations of Algorithm 2 are needed to obtain accurate results.

The costs of both Algorithms 1 and 2 may be reduced at the price of increased storage [Kanungo et al. 2002]. Another improvement to Algorithm 2 is possible by avoiding the comparison of reductions in the CVT-energy for possible transfers to far away clusters. We note that Algorithm 1 is easier to parallelize while Algorithm 2 is easier to be generalized to more general CVT's. There are many other algorithms for the computation of CVTs, including more recent works on the fully parallelizable probabilistic methods [Ju et al. 2002]. In [Du and Wang 2004b], generalizations of such probabilistic approaches are made for general mixture model based clusterings. Though the algorithmic details are more involved, the near perfect speed up does give the new algorithms significant advantage in clustering large data sets. We refer to [Ju et al. 2002] and [Du and Wang 2004b] for further discussions.

Mathematical Discussions. We here present some mathematical background for the vector fields clustering algorithms discussed in the paper.

Given a positive scaling constant w, a distance between two non-degenerate points $p = (x_p, y_p)$ and $q = (x_q, y_q)$ can be defined as

$$d(p,q) = \sqrt{1 - cos(\theta) + w|x_p - x_q|^2} \qquad (14)$$

where θ is the angle between the vector y_p and y_q, that is, $\cos(\theta)|y_p||y_q| = y_p^T y_q$. The constant w may be chosen to be dependent on L, the size of the spatial domain, so that it can be used to provide a scaling effect of different spatial domain sizes. For example, $w = 1/L^2$.

Obviously the above distance satisfies the following properties:

1. $d(p,q) = d(q,p)$;
2. $d(p,q) = 0 \Leftrightarrow p = q$;
3. $d(p,q) + d(q,r) \geq d(p,r)$.

The last inequality follows from $2(1 - cos(\theta)) = |y_p/|y_p| - y_q/|y_q||$. We now give a remark here for the distance formula (14). First of all, in some sense, if we measure the closeness of two vectors in the vector field by the difference of their directions only, then $d(\cdot,\cdot)$ is the most natural distance in the space $R^N \times S^{M-1}$ where S^{M-1} means the unit sphere in R^M. Moreover, it is also easy to see that the distance increases with a larger angle θ and a larger Euclidean distance between x_1 and x_2.

For a cluster in Ω, motivated by the geometric intuition that most of the vectors distributed in the cluster are desired to align in the direction and that the corresponding vector representation (simplification, or generator) should first be consistent with such an orientation, we first assign a constraint $|y_m| = 1$ to the generator $m = (x_m, y_m)$. Then by incorporating the idea that vectors in the cluster with larger magnitudes tend to affect the flow orientation more, into the consideration, we take the magnitude y_p as a weighting factor, and define the (one-sided) distance between p and m as

$$d_p(p,m) = |y_p|\sqrt{1 - cos(\theta) + w|x_p - x_m|^2}$$

or simply equation (4). Of course, since the constraint $|y_m| = 1$ is enforced, effectively we have $d_p(p,m) = |y_p||y_m|d(p,m)$.

Then, given a set of k generators $\{m_i\}_{i=1}^k$, the non-overlapping Voronoi regions $\{\widehat{C}_i\}$ corresponding to the points $\{m_i\}$ are defined by the equation (5). For some p that satisfying $d_p(p,m_i) = d_p(p,m_j)$ for two distinct generators $m_i \neq m_j$, we then assign p to the Voronoi region $\widehat{C}_i$ if $|x_p - x_{m_i}| < |x_p - x_{m_j}|$. Since the set of points with both $d_p(p,m_i) = d_p(p,m_j)$ and $|x_p - x_{m_i}| = |x_p - x_{m_j}|$ has zero measure (a set in a lower dimensional space), this tie-breaking rule guarantees that the Voronoi regions form a valid tessellation of the spatial domain Ω.

We note that the Voronoi tessellations defined above belong to the general class of *anisotropic Voronoi tessellations* studied in [Du and Wang 2004a]. The particular form of the distance definition is, however, unique as it pertains to our specific application. The tessellations (clusterings) of Ω are determined through generators and distances which live more naturally in a higher dimensional space $R^N \times S^{M-1}$ associated with both the spatial domain and the vector fields defined on Ω. Once the generators (and cluster centroids) in $R^N \times S^{M-1}$ are specified, some lifting operations are to be conducted to find suitable representations of the clusters and the vector fields back in the space R^{N+M}.

Now, some discussions on the cluster centers are in order. Given a cluster C, the centroid m^* is obtained as the minimizer of the energy defined in (1). Using the definition of d_p, we have

$$E(m) = \int_C |y_p|^2 - |y_p| y_p \cdot y_m + w|y_p|^2 |x_p - x_m|^2 \, dx_p \, .$$

To find such a minimizer m^* of $E(m)$ under the constraint $|y_m| = 1$, we need

$$\frac{\partial E}{\partial x_m}|_{m^*} = 2w \int_C |y_p|^2 (x_p - x_{m^*}) \, dx_p = 0 \, , \qquad (15)$$

$$\frac{\partial E}{\partial y_m}|_{m^*} = \int_C -|y_p| y_p \, dx_p = \lambda y_{m^*} \, , \qquad (16)$$

where λ is the Lagrange multiplier and $|y_{m^*}| = 1$.

From (15), (16) and $|y_{m^*}| = 1$. we get

$$x_{m^*} = \frac{\int_C |y_p|^2 x_p \, dx_p}{\int_C |y_p|^2 \, dx_p} \, , \quad y_{m^*} = \frac{\int_C |y_p| y_p \, dx_p}{|\int_C |y_p| y_p \, dx_p|} \, . \qquad (17)$$

In the context of spatially distributed vector fields, we have $y_p = V(x_p)$. Thus, based on the formula (17), together with (5), we have the algorithm 3 from the algorithm 1, and the algorithm 4 from the algorithm 2.

References

CABRAL, B., AND LEEDOM, L. 1993. Imaging vector fields using line integral convolution. *Computer Graphics (Proc. SIGGRAPH 93)*, 263–279.

DELEEUW, W., AND VANWIJK, J. 1995. Enhanced spot noise for vector field visualization. In *Proc. Visualization 95, IEEE Computer Society Press*, 233–239.

DU, Q., AND GUNZBURGER, M. 2002. Grid generation and optimization based on centroidal voronoi tessellations. *Appl. Math. Comp. 133*, 591–607.

DU, Q., AND WANG, D. 2002. Tetrahedral mesh generation and optimization based on centroidal voronoi tessellations. *Int. J. Numer. Meth. Eng. 56*, 1355–1373.

DU, Q., AND WANG, D. 2004. Anisotropic centroidal voronoi tessellations and their applications. *SIAM J. Sci. Comp. to appear*.

DU, Q., AND WANG, X. 2004. Tessellation and clustering by mixture models and their parallel implementations. In *Proceedings of the 2004 SIAM data mining conference, Orlando, FL*, SIAM.

DU, Q., FABER, V., AND GUNZBURGER, M. 1999. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review 41*, 637–676.

DU, Q., GUNZBURGER, M., AND JU, L. 2003. Constrained centroidal voronoi tessellations for surfaces. *SIAM Journal on Scientific Computing 24*, 1488–1506.

DU, Q., LIU, C., AND WANG, X., 2005. Retrieving topological information for phase field models. preprint.

GARCKE, H., PREUSSER, T., RUMPF, M., TELEA, A., WEIKARD, U., AND VANWIJK, J. 2001. A phase field model for continuous clustering on vector fields. *IEEE Transactions on Visualization and Computer Graphics 7*, 230–241.

HARTIGAN, J., AND WONG, M. 1979. Algorithm as 136: A k-means clustering algorithm. *Appl. Stat. 28*, 100–108.

JU, L., DU, Q., AND GUNZBURGER, M. 2002. Probabilistic methods for centroidal voronoi tessellations and their parallel implementations. *Journal of Parallel Computing 28*, 1477–1500.

KANUNGO, T., MOUNT, D., NETANYAHU, N., PIATKO, C., SILVERMAN, R., AND WU, A. 2002. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Machl Intel. 24*, 881–892.

SHEN, H., JOHNSON, C., AND MA, K. 1996. Visualizing vector fields using line integral convolution and dye advection. In *Proceedings of the 1996 symposium on Volume visualization, San Francisco, CA*, IEEE Press, 63–72.

SPARKS, D. 1973. Algorithm as 58: Euclidean cluster analysis. *Appl. Stat. 22*, 126–130.

SPÄTH, H. 1985. *Cluster Dissection and Analysis, Theory, FORTRAN Programs, Examples.* Ellis Horwood.

TELEA, A., AND VANWIJK, J. 1999. Simplified representation of vector fields. In *Proc. IEEE Visualization 99*, IEEE Computer Society Press, 35–42.

TURK, G., AND BANKS, D. 1996. Image-guided streamline placement. In *Computer Graphics (Proc. SIGGRAPH 96)*, ACM Press.