Segmentation of Piecewise Linear Vector Fields

Jiann-Liang Chen[1], Zhaojun Bai[2], and Terry Ligocki[3]

[1] Visualization and Graphics Research Group,
Center for Image Processing and Integrated Computing,
Department of Computer Science,
University of California, Davis
One Shield Avenue,Davis, CA 95616, USA
chenjj@cs.ucdavis.edu

[2] Department of Computer Science,
University of California, Davis
One Shield Avenue, Davis, CA 95616, USA
bai@cs.ucdavis.edu

[3] National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory
Mail Stop 50A-1148, One Cyclotron Road, Berkeley, CA 94720, USA
tjligocki@lbl.gov

Summary. In order to obtain insight into a complex vector field, it is often necessary to construct a hierarchical representation of the field. One way to construct such a hierarchy is based on grouping vectors together using certain similarity criteria. In this paper, we present a study of a 2D vector field clustering technique that is based on piecewise linear vector field approximations and an extension of a data clustering method called Normalized Cut (NC). Specifically, two steps are taken to implement the extended NC method. First, a similarity measurement for vector data is defined. Second, an eigenproblem solver is used to find the eigenvector used for partitioning. After the construction of first-level clusters, we can obtain a finer-level clustering by recursively applying the same procedure to intermediate clusters. The resulting clusters capture the features around the critical points.

Key words: Normalized Cut, Linearly Approximated Vector Field, Vector Field Segmentation, Similarity Measurement

1 Introduction

Visualizing vector field data is challenging due to the size and the complexity of the datasets involved. Our goal is to build a hierarchal representation of the vector field based on the partitions obtained using data clustering methods. On the finest level of the hierarchy, each cluster should contain exactly one feature. To obtain a coarser level of representation, clusters are then combined together to form a larger cluster. These larger clusters will imply cruder data approximation, but they will preserve global structure reasonably well.

It is often desirable to provide a capability of displaying the global structure of the underlying vectors. Knowledge about the global structure aids users in understanding the datasets and finding the features that are of interest for a particular application. Once the interesting features have been identified, the users can track the evolution of these features over time. Interesting features in 2D flow fields are, for example, critical points. For 3D datasets, features of interest include the vortex cores.

Based on the goal of extracting global structure of the vector field, a cluster is defined as the region that contains the vectors which are associated with a certain type of critical point. In the case of linear vector field, the vectors within a cluster can be expressed and approximated by the coefficient matrices computed from the cluster.

To achieve the proposed goals, we investigate the use of a graph-based clustering technique that automatically extracts vector features. One key aspect of the graph-based clustering technique is the computation of the connection weights between the vectors. The measurement itself does not determine the cluster membership directly, but rather it is used by the graph-based clustering method to partition the vector field.

Many hierarchy construction techniques have been proposed to deal with the complexity and the size of the vector field datasets. Different criteria for combining and approximating vectors have been introduced to achieve the goal of faithfully preserving the topology of the original vector field. In [9], a top-down approach that iteratively splits a vector filed into clusters was introduced. Based on the concept of Hardy's multiquadric method, a single vector is used to represent all the vectors inside each cluster. The representative vector is obtained by averaging the coordinates of the points and the associated vector values. Vectors inside a cluster are approximated by the computation of the local Hardy interpolant. With the error measurement defined as the difference between the streamlines generated from the original vectors and the ones generated from the approximated vectors, a bisecting plane that minimizes the error is used to split a cluster. This split process is repeated until the maximum cluster error is less than a predefined value or until the number of clusters has reached a certain number. Another top-down approach for vector field clustering is based on the Cahn-Hilard model for phase separation and particle coarsening [7, 8].

Based on topology analysis [10], the topology of vector fields can be simplified [2]. The implicit method is essentially an averaging technique that filters out the critical points with high spatial frequency. The explicit method removes pairs of critical points that are close spatially. A prescribed distance threshold is used to define the term of "spatially close". A more elaborate method was later proposed to ensure preservation of the flow structure, see [3]. Based on the notation of indexing [6], a collection of critical points is combined to form a new critical point with the same index. The index of a critical point is defined as the number of counter-clockwise revolutions made

by the arrows travelling on a closed curve around the critical point. There should be no other critical points inside this closed curve, and the arrow is drawn in the direction of the vector at the position on the curve.

A bottom-up approach was proposed to obtain a simplified representation of vector fields [21]. Initially, each cluster contains exactly one vector from the original dataset. Similar clusters are then merged to form a larger cluster with the new representative vector computed as the average of the original representative ones. This merging process is continued until one large cluster is formed that represents the entire dataset. The similarity measurement for the clusters is split into two parts. One part of the measurement deals with direction and magnitude comparisons, while the other part deals with position comparison. The final measuring function is defined as a linear combination of these two measurements. The coefficient used in the linear combination is then used to control clustering preference. Clusters based on either directional-and-quantitative similarity or positional similarity can be produced during the merging process.

Vector field simplification based on higher-order critical points was also proposed [23]. Critical points within a prescribed radius are combined to form a higher-order critical point. As the result of this combining process, each cluster contains exactly one such critical point. A hierarchy of the vector field is obtained by using different radii to form clusters with different sizes. The study of the higher-order critical points could be useful for our proposed method when dealing with the effect of combining first-order critical points.

The rest of this paper is organized as follows. In section 2, we introduce the concept of clustering, summarize the classification of the 2D critical points, and define the similarity measurement for vectors. In section 3, we review the normalized cut (NC) method, one of emerging graph-based clustering techniques. We introduce an extension of the NC method to vector field data partitioning and discuss related computational issues. Preliminary results of applying the proposed similarity measurement and the extension of the NC method are presented in section 4. Concluding remarks and further work are presented in section 5.

2 Similarity Measurement and Clustering for Vector Data

The premise of our proposed clustering technique is the fact that in a vector field every vector is associated with a certain type of critical point. Our idea is to group the vectors that are associated with the same critical point together, therefore forming a nature cluster. On the finest level of hierarchical representation of the vector field, a cluster is the set of all vectors that can be expressed or approximated with the same critical point. The size of the cluster, in a geometrical sense, represents the "influence region" of the underlying critical point. To obtain a coarser level of representation on

the hierarchy, adjacent clusters can be combined together, subject to certain combination criteria such as the sizes and types of critical points.

There are several advantages of grouping vectors together according to the proposed concept of cluster:

1. All the interesting features in a vector field can be identified, along with their influence regions. No averaging method is used on the vectors. The importance of the critical points is based only on their range of influence. Unlike [2, 3], the distance between them would not be used as the only criterion for combining clusters.
2. The global and the dominant structures of the vector field can be extracted by the types and the sizes of the clusters.
3. It depends on the hierarchy construction method used, the global and the dominant structure can be preserved at different level of representation.
4. The vectors within a cluster can be computed fairly accurate once the associated linear approximation matrices have been obtained. Compared to the method introduced in [9], we could actually obtain a truly representative vector for each cluster. This property of accurate approximation leads to the reduction of storage space and the ease of construction for multi-level representation.
5. The vectors within a cluster could be very different in terms of direction and magnitude. For example, vectors influenced by a focus-type critical point can point to completely opposite directions, but they would still be grouped together by our method. Unlike [21], our definition of "similar vectors" is more consistent with the underlying structure of the vector field.

The interesting features of a 2D vector field are the critical points, or features closely related to critical points. Critical points are the locations where the velocity vector is zero. They can be classified according to the behavior of nearby vectors, subject to a certain local polynomial approximation, see, for example, [10]. Several techniques have been proposed and widely used to understand the nature of the 2D critical points and to classify them. These techniques are summarized in this section.

A linear 2D vector field can be expressed as:

$$\begin{aligned}\mathbf{v}(x,y) &= \begin{bmatrix} v_1(x,y) \\ v_2(x,y) \end{bmatrix} = \begin{bmatrix} a_{1,1}x + a_{1,2}y + b_1 \\ a_{2,1}x + a_{2,2}y + b_2 \end{bmatrix} \\ &= \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \\ &= \mathbf{A}\mathbf{x} + \mathbf{b} \end{aligned} \tag{1}$$

A critical point (x_c, y_c) is defined by

$$\mathbf{v}(x,y)|_{x_c,y_c} = \begin{bmatrix} v_1(x_c,y_c) \\ v_2(x_c,y_c) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The eigenvalues of the Jacobian matrix of $\mathbf{v}$ (which is $\mathbf{A}$ in the case of linear vector fields) evaluated at the critical point (x_c, y_c) define the types of the critical points, see [4, 10, 14]. The Jacobian matrix is defined as

$$\mathbf{J}(x_c, y_c) = \frac{\partial \mathbf{v}(x,y)}{\partial x, \partial y}\bigg|_{x_c,y_c} = \begin{bmatrix} \frac{\partial v1(x,y)}{\partial x} & \frac{\partial v1(x,y)}{\partial y} \\ \frac{\partial v2(x,y)}{\partial x} & \frac{\partial v2(x,y)}{\partial y} \end{bmatrix}\Bigg|_{x_c,y_c}$$

In particular, the Jacobian matrix for linear vector fields (1) is

$$\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \tag{2}$$

If λ_1 and λ_2 denote the eigenvalues of the Jacobian matrix (2),

$$\lambda_1 = R_1 + iI_1 \quad \text{and} \quad \lambda_2 = R_2 + iI_2$$

then the signs of the real parts (R_1, R_2) and imaginary parts (I_1, I_2) of the eigenvalues λ_1 and λ_2 can be used to identify six different types of 2D critical points. These critical points are repelling focus, attracting focus, saddle point, center, repelling node (called stable node in [15]), and attracting node (called unstable node in [15]).

Another usage of the Jacobian matrix to identify critical points was proposed in [15], which is called p-q method. Let p and q be defined by

$$p = -\text{trace}(\mathbf{A}) = -(a_{1,1} + a_{2,2})$$

and

$$q = \det(\mathbf{A}) = a_{1,1} \cdot a_{2,2} + a_{1,2} \cdot a_{2,1},$$

then the sign as well as the values of them can be used to classify the critical points. In addition to the critical points identified by the signs of the eigenvalues, the p-q method identifies three more types of degenerated critical points, namely node-saddle, star node, and pure shear.

Furthermore, based on the concept of Earth Movers Distance (EMD) [16] and Clifford algebra [17], the star node can be further classified into attracting star and repelling star [12]. This approach computes the variables α and β of a so-called normalized (α, β)-space as follows:

$$\alpha = \frac{\hat{\alpha}}{\sqrt{\hat{\alpha} + \hat{\beta}}} \quad \text{and} \quad \beta = \frac{\hat{\beta}}{\sqrt{\hat{\alpha} + \hat{\beta}}},$$

where

$$\hat{\alpha} = p \quad \text{and} \quad \hat{\beta} = \text{sign}(p^2 - 4d) \cdot \sqrt{|p^2 - 4d|}.$$

The d and p components of $\hat{\alpha}$ and $\hat{\beta}$ are defined by

$$d = div(\mathbf{v}) = \frac{\partial v_1(x,y)}{\partial x} + \frac{\partial v_2(x,y)}{\partial y}$$

and

$$p = \det(\mathbf{J}).$$

The normalized (α, β)-space was further extended to the so-called (γ, λ)-space by converting α and β into an angle $\gamma \in [0, 2\pi]$ and a radius $\lambda \in [0, 1]$. As the result of this extension, even more 2D critical points are identified [22]. The additional first-order 2D critical points recognized by this method are:

1. Repelling saddle (RSa): a saddle with more outflow than inflow
2. Attracting saddle (ASa): a saddle with more inflow than outflow
3. Center, attracting focus, repelling focus, attracting star, and repelling star are each divided into subclass 1 and 2. (Critical points in the subclass 1 have non-negative curvatures, while critical points in subclass 2 have non-positive curvatures.)

As a summary, we put all discussed classification methods in Table 1.

We now turn to define similarity between vectors. The definition should consider the fact that a vector quantity conveys both direction and magnitude. Other desirable features of the similarity definition would include a smoothly varying similarity measure that ranges from 0 to 1. With these requirements in mind, for vectors $\mathbf{v}_i = \begin{bmatrix} v_{i,x} & v_{i,y} \end{bmatrix}$ located at position (x_i, y_i), and $\mathbf{v}_j = \begin{bmatrix} v_{j,x} & v_{j,y} \end{bmatrix}$ located at position (x_j, y_j), we propose the following definition of similarity measurement:

$$w(\mathbf{v}_i, \mathbf{v}_j) = \alpha \cdot e^{-\text{dist}(\mathbf{v}_i, \mathbf{v}_j)} + (1 - \alpha) \cdot e^{-\text{diff}(\mathbf{v}_i, \mathbf{v}_j)} \tag{3}$$

where

$$\text{dist}(\mathbf{v}_i, \mathbf{v}_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{4}$$

and

$$\text{diff}(\mathbf{v}_i, \mathbf{v}_j) = (v_{i,x} - v_{j,x})^2 + (v_{i,y} - v_{j,y})^2. \tag{5}$$

To ensure that the similarity measure varies between 0 and 1, the following condition is imposed on the parameter $\alpha : 0 \leq \alpha \leq 1$. The parameter α can be used to control the emphasis of the measurement. A small α will emphasize the measurement difference in direction and magnitude of the vectors; while a large α will put more weight on the distance between the vector locations. Equation (4) is simply a Euclidean distance function, and equation (5) is a measurement for both direction and magnitude. Of course, other equations could also be used as long as they have the desired characteristics.

3 Spectral Methods for Vector Data Segmentation

Spectral partition method for data clustering and segmentation is based on the utilization of the second eigenvector of the discrete Laplacian matrix of the graph representation of the underlying data. For overview of spectral partition methods, see [1, 13, 18].

Table 1. Classification of 2D critical points

Type	R_1, R_2 I_1, I_2	p, q	α, β	γ, λ
Center (C)	$R_1 = R_2 = 0 \;\; I_1 = -I_2 \neq 0$	$p = 0,$ $q > 0$	$\alpha = 0,$ $\beta < 0$	$\gamma = \frac{\pi}{2}$ $\frac{1}{2} < \lambda \leq 1$
$C1$				
$C2$				$\gamma = \frac{-\pi}{2}$ $\frac{1}{2} < \lambda \leq 1$
Attracting Focus (AF)	$R_1 = R_2 < 0 \;\; I_1 = -I_2 \neq 0$	$p > 0,$ $q > 0,$ $q > \frac{p^2}{4}$	$\alpha < 0,$ $\beta < 0$	$\frac{\pi}{2} < \gamma < \pi,$ $\frac{1}{2} < \lambda < \frac{1}{1+sin^2\gamma}$
$AF1$				
$AF2$				$\pi < \gamma < \frac{3\pi}{2},$ $\frac{1}{2} < \lambda < \frac{1}{1+sin^2\gamma}$
Repelling Focus (RF)	$R_1 = R_2 > 0 \;\; I_1 = -I_2 \neq 0$	$p < 0,$ $q > 0,$ $q > \frac{p^2}{4}$	$\alpha > 0,$ $\beta < 0$	$0 < \gamma < \frac{\pi}{2},$ $\frac{1}{2} < \lambda < \frac{1}{1+sin^2\gamma}$
$RF1$				
$RF2$				$\frac{-\pi}{2} < \gamma < 0,$ $\frac{1}{2} < \lambda < \frac{1}{1+sin^2\gamma}$
Attracting Node (AN)	$R_1 \neq R_2 < 0 \;\; I_1 = I_2 = 0$	$p > 0,$ $q < 0,$ $q < \frac{p^2}{4}$	$\lvert\alpha\rvert > \lvert\beta\rvert,$ $\alpha < 0,$ $\beta > 0$	$\frac{\pi}{2} < \gamma < \frac{3\pi}{2},$ $\frac{1}{1+sin^2\gamma} < \lambda \leq 1$
Attracting Star (AS)	$R_1 = R_2 < 0 \;\; I_1 = I_2 = 0$	$p > 0,$ $q > 0,$ $q = \frac{p^2}{4}$	$\alpha < 0,$ $\beta = 0$	$\frac{\pi}{2} < \gamma < \pi,$ $\lambda = \frac{1}{1+sin^2\gamma}$
$AS1$				
$AS2$				$\pi < \gamma < \frac{3\pi}{2},$ $\lambda = \frac{1}{1+sin^2\gamma}$
Repelling Node (RN)	$R_1 \neq R_2 > 0 \;\; I_1 = I_2 = 0$	$p < 0,$ $q > 0,$ $q < \frac{p^2}{4}$	$\lvert\alpha\rvert > \lvert\beta\rvert,$ $\alpha > 0,$ $\beta > 0$	$\frac{-\pi}{2} < \gamma < \frac{\pi}{2},$ $\frac{1}{1+sin^2\gamma} < \lambda \leq 1$
Repelling Star (RS)	$R_1 = R_2 > 0 \;\; I_1 = I_2 = 0$	$p < 0,$ $q > 0,$ $q = \frac{p^2}{4}$	$\alpha > 0,$ $\beta = 0$	$0 < \gamma < \frac{\pi}{2},$ $\lambda = \frac{1}{1+sin^2\gamma}$
$RS1$				
$RS2$				$\frac{-\pi}{2} < \gamma < 0,$ $\lambda = \frac{1}{1+sin^2\gamma}$
Saddle Point (SA)	$R_1 < 0, R_2 > 0 \;\; I_1 = I_2 = 0$	$q < 0,$	$\lvert\alpha\rvert < \lvert\beta\rvert,$ $\beta > 0$	$\gamma = \frac{\pi}{2}, \lambda < \frac{1}{2}$ or $\gamma = \frac{-\pi}{2}, \lambda = \frac{1}{2}$ or $\lambda = 0$
SA				
RSa				$\frac{-\pi}{2} < \gamma < \frac{\pi}{2},$ $0 < \lambda < \frac{1}{2}$
RSb				$\frac{\pi}{2} < \gamma < \frac{3\pi}{2},$ $0 < \lambda < \frac{1}{2}$
Node-Saddle 1	$R_1 > 0, R_2 = 0 \;\; I_1 = I_2 = 0$			
Node-Saddle 2	$R_1 < 0, R_2 = 0 \;\; I_1 = I_2 = 0$			
Pure Shear	$R_1 = R_2 = 0 \;\; I_1 = I_2 = 0$			

3.1 Normalized cut method for scalar data segmentation

The normalized cut (NC) method was developed for scalar data clustering, such as image segmentation [19]. Based on graph theory, the NC method models a given scalar dataset as a weighted undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. The nodes in $\mathbf{V}$ represent the data points, while the weighted edges in $\mathbf{E}$ denote similarities. The goal of this method is to minimize the disassociation between the two disjoint subsets A and B, where $A \cup B = \mathbf{V}$ and $A \cap B = \emptyset$. Following [19], the disassociation is defined as

$$\mathrm{Ncut}(A, B) = \frac{\mathrm{cut}(A, B)}{\mathrm{assoc}(A, V)} + \frac{\mathrm{cut}(A, B)}{\mathrm{assoc}(A, V)},$$

where $\mathrm{cut}(A, B) = \sum_{u \in A, v \in B} w(u, v)$, $\mathrm{assoc}(A, V) = \sum_{u \in A, t \in V} w(u, t)$, and $\mathrm{assoc}(B, V) = \sum_{u \in B, t \in V} w(u, t)$. The segmentation problem is then formulated as a generalized eigenvalues problem with the signs of eigenvector components used as partition indicators. The NC method, described on a high level, works as follows:

1. Define a weight function $w(v_i, v_j)$ that measures the similarity between nodes v_i and v_j. Weights should reflect the likelihood of two data points belonging to the same group. For image data, the criteria considered for the weight function definition are usually color or intensity values and locations. Place the measured results into the weight matrix $\mathbf{W}(v_i, v_j)$.
2. Calculate the weights between all nodes. Usually, this step can be simplified to only calculate the weights between a node and its neighboring nodes that are within a prescribed radius r.
3. Construct a vector $\mathbf{d}$ as follows:

$$\mathbf{d}(i) = \sum_{j=1}^{N} \mathbf{W}(v_i, v_j),$$

   where $N = |\mathbf{V}|$, the total number of nodes.
4. Construct an $N \times N$ diagonal matrix $\mathbf{D}$, having $\mathbf{d}$ as its diagonal.
5. Compute the eigenvector associated with the second-smallest eigenvalue of the matrix

$$\mathbf{D}^{-\frac{1}{2}} \cdot (\mathbf{D} - \mathbf{W}) \cdot \mathbf{D}^{-\frac{1}{2}}$$

6. The eigenvector would have N components, where each component corresponds to a data point. The signs of these components are then used as partition indicators. The data points with corresponding positive component signs are clustered into one group, while the ones with non-positive signs are placed into another group.
7. Additional portioning of the generated groups can be accomplished by either executing the described procedure again on each partition, or by using the eigenvectors associated with the third- and fourth-smallest eigenvalues as indicator vectors.

3.2 Extension of the NC method to vector field data

To extend the NC method to vector field data, under the context of the proposed cluster concept, we will not apply equations (3), (4) and (5) directly on any given pair of vectors. Instead, we will consider how well they can be approximated by the linear squares approximation and use these equations on a vector and its approximated neighboring vectors. Therefore, the similarity between vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ is measured by first computing $\hat{\mathbf{v}}_j$, the approximated $\mathbf{v}_j$, then apply the weight function (3) on the vectors $\mathbf{v}_i$ and $\hat{\mathbf{v}}_j$.

We now proceed to present a modified NC method to vector datasets. The process of measuring similarity and constructing a partition of vector field data is described by the following procedure:

1. Construct the weight (or association) matrix:
   a) For each vector $\mathbf{v}_i = \begin{bmatrix} v_{i,x} & v_{i,y} \end{bmatrix}$ at location (x_i, y_i), randomly pick m neighboring vector data within a circle of radius r centered at (x_i, y_i). The neighboring vectors can be represented as $\mathbf{v}_n = \begin{bmatrix} v_{n,x} & v_{n,y} \end{bmatrix}$, where $n = 1, 2, 3, \ldots, m$. The locations of these neighboring vectors are $\mathbf{x}_n = (x_n, y_n)$.
   b) Use the vector components and the coordinates of the vectors $\mathbf{v}_n$ and $\mathbf{v}_i$ to define the following linear least squares equation:

$$\begin{bmatrix} x_i & 0 & y_i & 0 & 1 & 0 \\ 0 & x_i & 0 & y_i & 0 & 1 \\ x_1 & 0 & y_1 & 0 & 1 & 0 \\ 0 & x_1 & 0 & y_1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m & 0 & y_m & 0 & 1 & 0 \\ 0 & x_m & 0 & y_m & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{1,1} \\ a_{2,1} \\ a_{1,2} \\ a_{2,2} \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} v_{i,x} \\ v_{i,y} \\ v_{1,x} \\ v_{1,y} \\ \vdots \\ v_{m,x} \\ v_{m,y} \end{bmatrix} \tag{6}$$

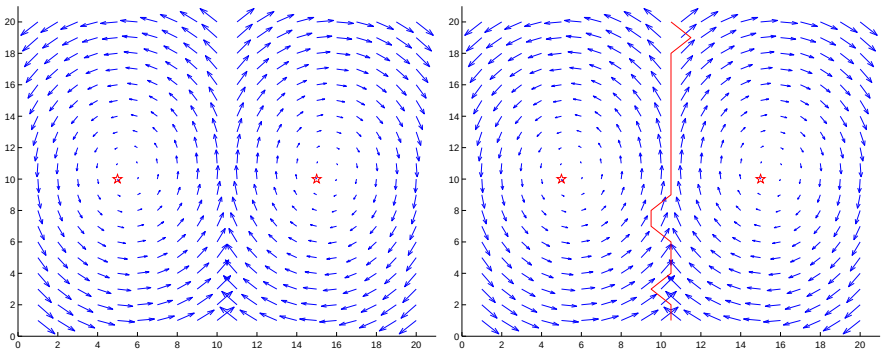
   c) Solve the linear least squares equation (6) to obtain the coefficient matrix $\mathbf{A}$ and the vector $\mathbf{b}$ of the vector field (1).
   d) Evaluate the linear least squares approximation $\hat{\mathbf{v}}_n = \mathbf{A} \cdot \mathbf{x}_n + \mathbf{b}$ at the locations of the chosen neighboring vectors. The resulting approximating vectors are
$$\hat{\mathbf{v}}_n = \begin{bmatrix} \hat{v}_{n,x} & \hat{v}_{n,y} \end{bmatrix}$$
   e) The similarity between $\mathbf{v}_i$ and $\mathbf{v}_n$ are then computed by applying equation (3) to $\mathbf{v}_i$ and $\hat{\mathbf{v}}_n$. The computed similarity is then used to build the weight matrix $\mathbf{W}$.
2. Compute the eigenvector associated with the second-smallest eigenvalue of the matrix $\mathbf{D}^{-\frac{1}{2}} \cdot (\mathbf{D} - \mathbf{W}) \cdot \mathbf{D}^{-\frac{1}{2}}$.
3. Partition the vector field:

a) Use the signs of the eigenvector components as indicators to partition the vector dataset. Vectors corresponding the same signs are placed in one group. The resulting partition is the first-level partition.
b) Finer partitioning is achieved by either executing the described procedure again on each partition, or by using the eigenvectors associated with the third- and fourth-smallest eigenvalues as indicator vectors [19].

4 Experimental Results

Several 2D synthetic vector fields are used to test the validity of the proposed method. All datasets used in our tests contain two or more critical points. Vector data are sampled on a grid. Different parameter combinations are applied to the test datasets. The parameters are the radius r, the number of neighboring vectors m, and the α used in the similarity measurement equation (3). The goal of our experiment is to test the validity of the proposed method.

The first test case is a two-center vector field with one center rotating counter-clockwise and the other one rotating clockwise. Figure 1 shows the resulting one-level partition, with parameter values $r = 5, m = 20$, and $\alpha = 0.1$. By comparing the resulting partitions with the original vector field, we observe that the NC method segments the dataset into two clusters. Each cluster contains almost all the vectors that one would associate with the corresponding center.

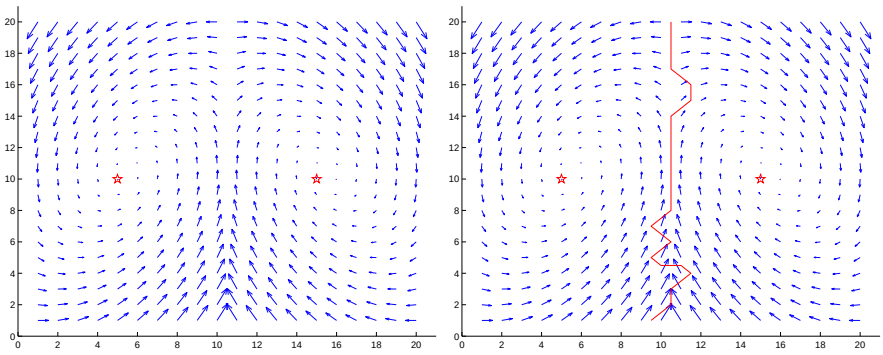


(a) (b)

Fig. 1. (a) A two-center vector field. (b) The partition with parameters $r = 5$, $m = 20$ and $\alpha = 0.1$.

The second dataset is a two-focus vector field with an attracting and a repelling focus. Figure 2 shows the resulting one-level partition, using parameter values $r = 5$, $m = 16$, and $\alpha = 0.1$. The resulting partition is almost

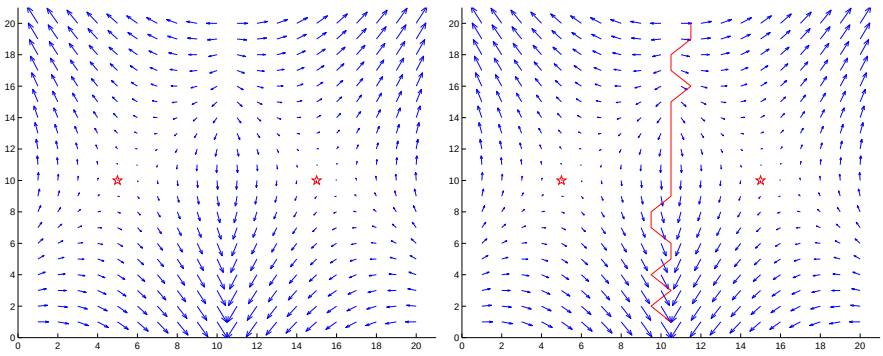
perfect with each cluster containing the majority of the vectors that should be associated with each focus.



(a) (b)

Fig. 2. (a) A two-focus vector field. (b) The partitions with parameter values $r = 5, m = 16,$ and $\alpha = 0.1$.

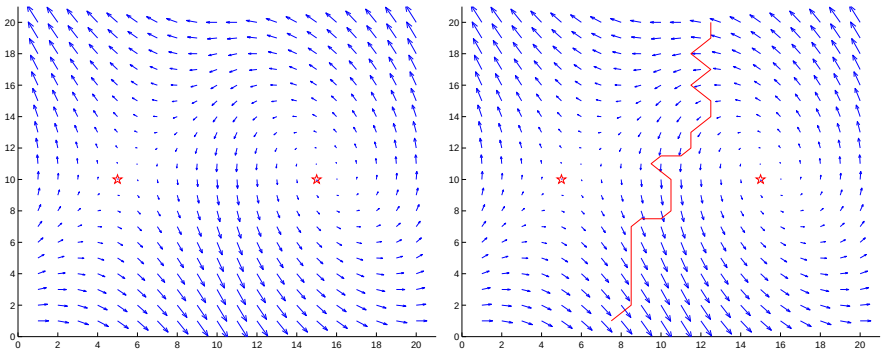
The third dataset is a two-saddle vector field. Figure 3 shows the resulting one-level partition using $r = 5$, $m = 16$ and $\alpha = 0.1$. Again, the resulting partitioning contains almost all the vectors that should be associated with the respective saddles.



(a) (b)

Fig. 3. (a) A two-saddle vector field. (b)shows the partitions with parameter values $r = 5, m = 16,$ and $\alpha = 0.1$.

The fourth dataset is a vector field with a saddle and a repelling focus. Figure 4 shows the results after a one-level partition using parameter values $r = 5, m = 14,$ and $\alpha = 0.0$. The resulting partitioning contains almost all the vectors that should be associated with the critical points. Another



(a) (b)

Fig. 4. (a) A saddle-focus vector field. (b) The partitions with parameter values $r = 5, m = 14,$ and $\alpha = 0.0$.

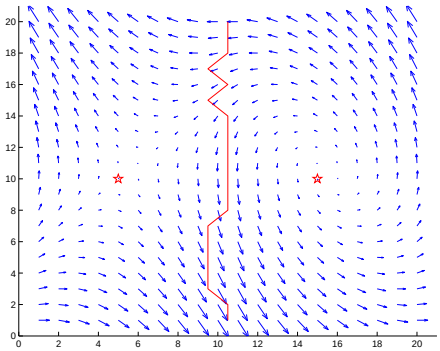
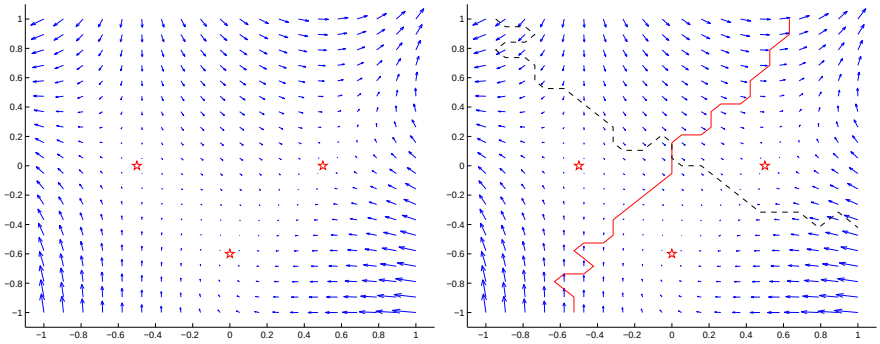


Fig. 5. A different partitioning of the saddle-focus vector field with parameter values $r = 5, m = 14,$ and $\alpha = 0.1$.

partitioning, using parameter values $r = 5, m = 14,$ and $\alpha = 0.1$, is shown in Figure 5. Although the partitions obtained by these two tests are different, both results are acceptable since the vectors in the lower middle section can be associated with either critical point.

Datasets with more than two critical points were also used to test the extended NC method. Furthermore, we consider and evaluate the results of clustering based on the eigenvector associated with the third-smallest eigenvalue. A synthetic dataset with three critical points is used to carry out this test. Figures 6 and 7 show the results of different parameter settings and scaling the length and locations of the vectors. It is interesting to observe that the first two levels of partitioning do not cut through any critical point. This is a desirable feature for vector field clustering.

The next experiment is carried out on the first two-center vector field but with two different settings:



(a) (b)

Fig. 6. (a) A two-saddle and one-focus vector field. (b) The partitions with parameter values $r = 3, m = 6,$ and $\alpha = 0.1$. Solid line indicates the 1st-level partitions while the dashed line shows the 2nd-level partitions.

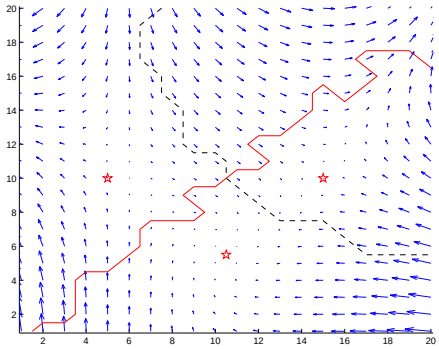


Fig. 7. Partitions of the two-saddle and one-focus vector field with different coordinates and same parameters values ($r = 3, m = 6,$ and $\alpha = 0.1$.).

1. Eight immediate neighboring vectors are always used in the computation of similarity.
2. If the similarity between any pair of vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ has been measured twice from each vector's perspective, then the similarity between them is averaged.

The resulting partition of using parameter values $r = 5, m = 20,$ and $\alpha = 0.1$ is shown in Figure 8.

5 Conclusions and Future Work

In preparation of achieving the goal of constructing a hierarchical representation for vector fields, we have presented the concept and definition of clusters

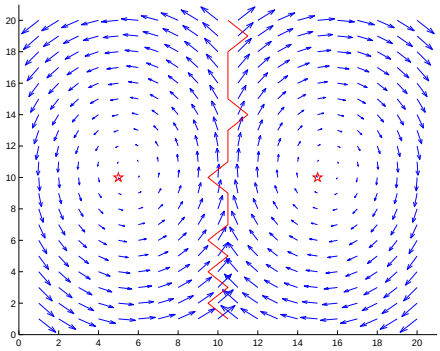


Fig. 8. Partitions of the two-center vector field with immediate neighbors, averaged similarity measurement and parameters values $r = 5, m = 20,$ and $\alpha = 0.1$.

and similarity. The utilization of the normalized cut method along with the proposed similarity measurement was shown to be plausible from the preliminary experimental results. But many issues are still need to be resolved in order to realize the goal of applying the proposed method on realistic vector datasets, and to extend the approach to 3D time-varying vector fields. Some of these issues are:

1. Cluster refinement: two issues arise as a consequence of the preliminary results produced by the proposed method. First, the extracted clusters do not contain every vector that should be included. Second, the terminating condition of the clustering process must be defined. A suitable cluster refinement scheme would be needed. One possible solution is the Linkage Refinement scheme introduced in [5]. A prescribed error margin between the original and the approximated vectors within the cluster could be used to stop the partitioning process.
2. Constructing the hierarchy: with the extended NC method, data hierarchies can be built iteratively as the vector field is being partitioned. The representative vectors could be obtained by evaluating the local linear least squares approximant at the center of each cluster. One issue is the preservation of flow topology. More investigation will be done to derive good approaches for constructing the hierarchies and to obtain the representative vectors.
3. Computational issue: since using the NC method on a $M \times N$ vector field will require solving a $MN \times MN$ eigenproblem. Efficient techniques are needed for practical purpose.
4. Handling 3D datasets: the first step is to obtain the capability to identify vortex cores [11, 20]. The next step is to incorporate the vortex core identification technique into the similarity measurement. Alternatively, one can also perform the extraction of the vortex cores prior the clustering

process, then use the extracted volumes as the building blocks in the hierarchy construction process.

Acknowledgment

Bernd Hamann has worked with us on and has contributed substantially to this research over the past two years. We would like to express our gratitude to the members of CIPIC's Visualization and Graphics Research Group at UC Davis, and to the members of the Applied Numerical Algorithms Group and the Visualization Group at Lawrence Berkeley National Laboratory for their valuable suggestions, insights, and support.

This work was supported by the Lawrence Berkeley National Laboratory; the National Science Foundation under contract ACI 9624034 (CAREER Award), through the Large Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251, and through the National Partnership for Advanced Computational Infrastructure (NPACI). We also acknowledge the support of ALSTOM Schilling Robotics and SGI.

References

1. Chung, F. R. K. (1997): Spectral Graph Theory. American Mathematical Society, Providence, R.I.
2. de Leeuw, W., Van Liere, R. (1999): Visualization of Global Flow Structures using Multiple Levels of Topology. In: Groller, E., Loffelmann, H., Ribarsky, W. (eds) Data Visualization 1999. Springer-Verlag/Wien
3. de Leeuw, W., Van Liere, R. (1999): Collapsing Flow Topology using Area Metrics. Proceedings Visualization 1999, 349–354
4. Dickinson, R.R (1991): Interactive Analysis of the Topology of 4D Vector Fields. IBM Journal of Research and Development, **35**, 59-66
5. Ding, C., He, X., Zha, H., Gu, M., Simon, H. (2001): Spetrcal Min-Max Cut for Graph Partitioning and Data Clustering. Tech. Report LBNL-47848, Natonal Engery Research, Scientific Computing Division, Lawrence Berkeley National Laboratory
6. Firby, P. A., Gardiner, C. F. (1991): Surface topology, 2nd ed., John Wiley, New York
7. Garcke, H., Preusser, T., Rumpf, M., Telea, A., Weikard, U., Van Wijk, J. J. (2000): A Continuous Clustering Method for Vector Fields. Proceedings Visualization 2000, 351–358
8. Garcke, H., Preusser, T., Rumpf, M., Telea, A.C., Weikard, U., van Wijk, J. J. (2001): A Phase Field Model for Continuous Clustering on Vector Fields. IEEE Transactions on Visualization and Computer Graphics, **7**, 230–241
9. Heckel, B., Weber, G., Hamann, B., Joy, K.I. (1999): Construction of Vector Field Hierarchies. Proceedings Visualization 1999, 19–25
10. Helman, J., Hesselink, L. (1989): Representation and Display of Vector Field Topology in Fluid Flow Data Sets. Computer, **22**, 27–36

11. Kenwright, D.N., Haimes, R. (1998) Automatic Vortex Core Detection. IEEE Computer Graphics and Applications, **18**, 70-74
12. Lavin, Y., Batra, R., Hesselink, L. (1998): Feature Comparisons of Vector Fields using Earth Mover's Distance. Proceedings Visualization 1998, 103-109
13. Mohar, B. (1991): The Laplacian Spetrum of Graphs. In: Alavi, Y.; Chartrand, G.; Oellermann, O. R.; Schwenk, A. J. (ed) Graph Theory, Combinatorics, and Applications. Wiley
14. Nielson, G.M., Jung, I.-H., Srinivasan N., Sung, J., Yoon, J.-B. (1997): Tools for Computing Tangent Curves and Topological Graphs for Visualizing Piecewise Linearly Varying vector Fields over Triangulated Domains, In: Nielson, G. M., Hagen, H., Müller, H. (ed) Scientific Visualization, Overviews, Methodologies, and Techniques, IEEE Computer Society Press
15. Perry, A. E., Chong, M. S. (2000): Introduction of Flow Visualization, In Smits, A. J., Lim, T.T. (ed) Flow Visualization, Imperial College Press
16. Rubner, Y., Tomasi, C., Guibas, L.J. (1998): A Metric for Distributions with Applications to Image Databases. Proceedings of IEEE 6th International Conference on Computer Vision, 59-66
17. Scheuermann, G., Hagen, H., Kruger, H., Menzel, M., Rockwood, A. (1997): Visualization of Higher Order Singularities in Vector Fields. Proceedings Visualization 1997, 67-74
18. Schoegel, K., Karypis, G., Kumar, V. (2002): Graph Partitioning for High Performance Scientific Simulations. In: Dongarra, J., Foster, I., Fox, G., Kennedy, K., White, A. (ed) The Sourcebook of Parallel Computing, Morgan Kaufmann
19. Shi, J., Malik, J. (2000): Normalized Cuts and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, **22**, 888–905
20. Sujudi, D., Haimes, R. (1995) Identification of Swirling Flow in 3-D Vector Fields, Technical Report, Department of Aeronautics and Astronautics, MIT
21. Telea, A., Van Wijk, J.J. (1999): Simplified Representation of Vector Fields. Proceedings Visualization 1999, 35-42
22. Theisel, H., Weinkauf, T. (2002): Vector Field Metrics based on Distance Measures of First Order Critical Points, Proceedings WSCG 2002
23. Tricoche, X., Scheuermann, G., Hagen, H. (2000): A Topology Simplification Method for 2D Vector Fields. Proceedings Visualization 2000, 359–366