Vector Field Segmentation Based on Integral Curve Attributes

Lei Zhang, Robert S. Laramee, David Thompson, Adrian Sescu, and Guoning Chen

Abstract— We propose a segmentation method for vector fields. Our segmentation is driven by integral curve attributes that are used to classify different behaviors of integral curves. In particular, we assign an attribute value to each spatio-temporal position based on the integral curve that passes through it. With this attribute information, our segmentation first performs a region classification. Then, the connected components are constructed from the derived classification to obtain an initial segmentation. After merging and filtering small segments, we extract and refine the boundaries of the segments. Because the points that are correlated by the same integral curves have the same or similar attribute values, the proposed segmentation typically results in segments that are well-aligned with the flow direction. Therefore, additional processing is not required to generate other geometric descriptors within the individual segments to illustrate the flow behaviors. We apply our method to a number of synthetic and CFD simulation data sets and compare it with existing methods to demonstrate its effectiveness.

Index Terms—Vector field data, integral curves, flow visualization, flow segmentation.

1 Introduction

Vector field analysis is a ubiquitous tool employed to study a wide range of dynamical systems involved in applications including automobile and aircraft engineering, climate study, combustion dynamics, earthquake engineering, and medicine, among others. With the continuous increase in size and complexity of the generated vector field data sets, many partition-based techniques have been developed to present an overview of the behavior of the vector fields.

Existing techniques typically partition the flow domain based on either certain local flow characteristics, such as vector magnitude and orientation, or via its topological structure [29]. Specifically, segmentation techniques that are based on local flow information usually perform vector field clustering in a hierarchical fashion, i.e., either top-down [9] or bottom-up [21, 33]. Since this clustering is based on the local flow information, the segments and their boundaries need not to be aligned with the flow direction (see Figure 9(b) for an example). Therefore, additional computation is needed in order to generate other visual primitives, such as stream-lets or glyphs [21], to convey information about the flow behavior within each segment. In addition, vector field topology [13, 23] reveals the essential flow structure that helps partition the flow domain into regions with homogeneous behaviors. However, it is yet to be applied to unsteady flow, and its visual representation in high dimensional space can be too complicated for the domain experts to comprehend.

Integral curve attributes have been recently applied to cluster [16] and select [38] integral curves to generate an overview of the vector fields. However, depending on the dissimilarity metric used for integral curve comparison, some important features may be overlooked due to the insufficient sampling of integral curves. To remedy this, very dense integral curves can be computed as the input, which will significantly increase the memory and storage consumption in computation. In addition, integral surfaces may be a better descriptor than integral curves for depicting important flow dynamics in higher-dimensional spaces, as mentioned in a recent survey [4]. Nonetheless, integral surface placement is a much harder problem than integral curve placement [4].

In this paper, we introduce a segmentation framework based on the integral curve attributes applicable to both 2D and 3D vector fields. Our method combines the advantages of the vector field clustering and integral curve attribute approaches, and attempts to achieve an intuitive and expressive segmentation of the flow domain whose resulting segments are aligned with the flow. Our framework consists of two major components. First, we derive several *attribute fields* from the integral curve attributes. The attribute field value, i.e. a scalar data, at each spatio-temporal position equals to the attribute value of the integral curve that passes through this position. The attribute value of an integral curve is computed by integrating local properties along it [22, 31]. If the integral curve is computed over a long time window, the attribute field will encode the global behaviors of the vector field [39]. With the derived attribute fields, the vector field segmentation problem is converted into a scalar field partitioning problem. In fact, a similar idea has been described in [12]. Different from their goal of generating implicit clusters induced by semantic dependencies, our method generates a segmentation based on the flow information only. With the aid of the derived attribute fields, we first classify the sampled spatial positions according to their attribute values. Then, the connected components of this classification are extracted to provide an initial segmentation. This initial segmentation may contain some smaller segments due to the numerical error in the attribute field computation. We then perform dilation operations to remove those small segments. After cleaning the initial segmentation, the boundaries of the obtained segments may still be non-smooth. We then further smooth these boundaries. Unlike the boundary refinement applied in [15] our method does not require to compute many contours and compare them to the coarse segmentation curves. The pipeline of this work is shown in Figure 1.

In summary, the contributions are as follows:

- We introduce an integral curve attribute based flow segmentation framework, which generates segments that are better aligned with the flow than those produced by existing local methods. In addition, our framework is flexible and efficient so that various attribute fields can be used to guide the segmentation and help users inspect different flow behaviors.
- We present a complete pipeline for generating the initial segmentation and filtering for both 2D and 3D vector fields. This includes a dilation operation for removing small segments and an effective boundary refinement algorithm for removing noise at segment boundaries. Our framework is simple to implement and computationally efficient. We demonstrate its effectiveness via applications to a number of synthetic and CFD simulation data. In particular, we wish to point out that our segmentation approach can aid the visualization of high dimensional vector fields without explicitly placing stream surfaces (Figure 13(b)).

The rest of the paper is structured as follows. Section 2 reviews

- *Lei Zhang and Guoning Chen are with University of Houston. Email: {lzhang38, gchen16}@uh.edu.*
- *Robert S. Laramee is with Swansea University, Email: rlaramee@gmail.com.*
- *David Thompson ad Adrian Sescu are with Mississippi State University, Email: {dst, asescu}@cavs.msstate.edu*

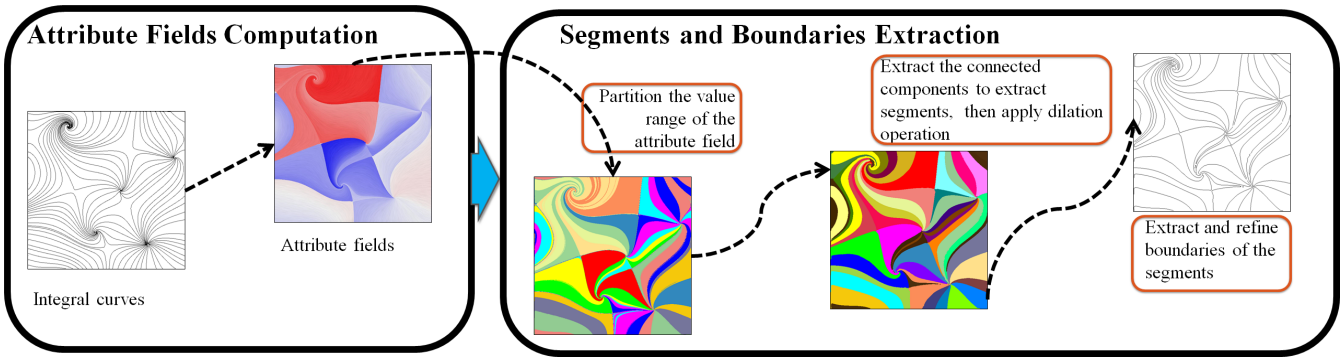

Fig. 1: The pipeline of our method.

the previous work related to the proposed method. Section 3 briefly reviews the important concepts of vector fields and the attribute fields and their computation. Section 4 describes the segmentation algorithm based on the integral curves and the boundary refinement. The applications of the segmentation algorithm to a number of steady and unsteady flows are reported in Section 5. Section 6 summarizes out work and discusses its limitations.

2 Related Work

There is a large body of literature on the analysis and visualization of flow data. Interested readers are encouraged to refer to recent surveys [4, 11, 13, 23] that provide systematic classifications of various analysis and visualization techniques. In this section, we focus on the most relevant work on integral curve attributes and partition-based flow visualization.

Vector field topological analysis Vector field topology provides a streamline classification strategy based on the origin and destination of individual streamlines. Since its introduction to the visualization community [10], vector field topology has received extensive attention. A large body of work has been introduced to identify different topological features, including fixed points [24, 35] and periodic orbits [1, 34, 37]. Recently, Chen et al.[2] studied the instability of trajectory-based vector field topology and, for the first time, proposed Morse decomposition for vector field topology computation, which leads to a more reliable interpretation of the resulting topological representation of the vector field.

The success of vector field topology for the analysis of steady vector fields has inspired efforts to extend it to the analysis of unsteady vector fields. The most successful strategies are based on the Lyapunov exponent. Specifically, *Lagrangian Coherent structures (LCS)* were introduced to identify separation structure for pathlines for time-dependent flow. The identification of LCSs was first introduced by Haller [8] by computing the *Finite Time Lyapunov Exponent (FTLE)* of the flow. Using FTLE fields computed from forward and backward time integrations of pathlines, two scalar fields are derived whose values are large in magnitude in areas of diverging and converging flow, respectively. LCSs are then defined as the ridge-lines/-surfaces of the two FTLE fields with negligible flux [14, 30]. Since its introduction, FTLE has been compared with the separatrices in the steady cases [26], and its computational efficiency has been improved substantially [5].

Streamline and pathline attributes Sadarjoen and Post introduced the winding angle concept for streamlines and utilized it to classify the streamlines within vortical regions [25]. Salzbrunn and Scheuermann introduced *streamline predicates*, which classifies streamlines by interrogating them as they pass through certain features, e.g., vortices [28]. Later, this approach was extended to the classification of pathlines [27]. At the same time, Shi et al. presented a data exploration system to study the different characteristics of pathlines based on their various attributes, including winding angles [31]. Recently, a statistics-based method was proposed to help select the proper set of pathline attributes to improve the interactive flow analysis [22]. Different from the approaches of the pathline predicate and clustering based on pathline attributes, our work is to segment the flow domain where pathlines are located. There are other techniques for classifying streamlines that are constrained to flow separation and vortex structures. The recently introduced streamline bundling technique [38] is such an example. More recently, McLoughlin et al. introduced the idea of a streamline signature based on a set of curve-based attributes including curvature and torsion [20]. This streamline signature is used to compute the similarity between streamlines and help domain experts place and filter streamlines for the creation of an informative and uncluttered visualization of 3D flow.

Flow segmentation A top-down method and a bottom-up approach were proposed by Heckel et al. [9] and Telea et al. [33], respectively. Recently, an image-space mesh-driven vector field clustering algorithm is introduced by Peng et al. [21]. They provided a bottom-up approach to generate a hierarchical cluster of vector fields defined on 2-manifolds. However, these hierarchical methods only consider local flow behavior, therefore, global flow behavior may not be revealed. McKenzie et al. [19] implemented an error-driven approach for variational clustering. Li et al. [15] proposed an approach for 2D vector field segmentation based on Hodge decomposition and the normalized cut algorithm. The Green Function Method (GFM) was used to approximate the curl-free and the divergence-free components to segment the vector field. Guan et al. [7] introduced a feature-emphasized clustering method for 2D vector fields. A 3D vector field clustering approach based on integral curvature was proposed by Kuhn et al. [12]. The authors detected regions of similar geometric properties such as integral curvature and visualizes them by means of compact cluster boundaries. More partition-based techniques for flow segmentatoin are described in the survey by Salzbrunn et al. [29].

3 Vector Field Background and Trajectory Attributes

Consider a d-manifold $\mathbb{M} \subset \mathbb{R}^d$(here d=2 or 3), a vector field can be expressed as an ordinary differential equation (ODE) $\dot{\mathbf{x}} = V(\mathbf{x},t)$ or a map $\varphi : \mathbb{R} \times \mathbb{M} \to \mathbb{R}^d$. There are a number of curve descriptors that depict different aspects of the translational property in vector fields.

- A *streamline* is a solution to the initial value problem of the ODE system confined to a given time t_0: $\mathbf{x}_{t_0}(t) = \mathbf{p}_0 + \int_{t_0}^{t} V(\mathbf{x}(\eta);t_0)d\eta$.
- *Pathlines* are the paths of the massless particles released in the flow domain at a given time t_0: $\mathbf{x}(t) = \mathbf{p}_0 + \int_{t_0}^{t} V(\mathbf{x}(\eta);t_0+\eta)d\eta$.
- A *streakline*, $\tilde{s}(t)$, is the connection of the current positions of the particles, $\mathbf{p}_{t_i}(t)$, that are released from position p_0 at consecutive times t_i.

There are a number of features in *steady flows*, $V(\mathbf{x})$, that are of interest. A point $\mathbf{x}_0$ is a *fixed point* (or *singularity*) if $V(\mathbf{x}_0) = 0$, and a trajectory is a *periodic orbit* if it is closed. Hyperbolic fixed points, periodic orbits and their connectivity define the *vector field topology* [1].

Vortices are another important flow feature that are of interest to domain experts. Unfortunately, there is no unified definition for vortices. Informally, a vortex is a region where the flow particles are rotating around a common axis (reduced to a point in 2D) [17]. In this work, we consider streamlines with larger winding angles than a user-specified threshold, e.g. 2π, are within vortices. In unsteady flows, topology is not well-defined. One typically looks for certain *coherent structures* that correspond to structures in the flow that are present for a *relatively* long time. The LCS, i.e., the ridges of the FTLE field, is one such coherent structure [8].

3.1 Attribute Fields

Various attributes can be extracted for the analysis and classification of integral curves [22]. Among these attributes, many of them can be obtained by accumulating certain local flow properties measured at the integration points, such as the arc-length and the winding angle of an integral curve. We adopt the Eulerian representation from texture-based methods and store the accumulated values at the sampled spatial positions. The value at each sample is determined by the integral curve passing through it. This Eulerian representation gives rise to a derived attribute field.

Considering an integral curve, $\mathscr{C}$, that starts from a given spatio-temporal point $(\mathbf{x}, t_0)$, the attribute field value at this point is computed as [39]:

$$\mathscr{F}(\mathbf{x}, t_0) = \mathscr{F}(\mathscr{C}(\mathbf{x})|_{t_0}^{t_0+T}) \tag{1}$$

where $\mathscr{C}(\mathbf{x})|_{t_0}^{t_0+T}$ denotes an integral curve, i.e., either a streamline or a pathline starting at time t_0 with an integral time window $[t_0, t_0+T]$. $\mathscr{F}(\cdot)$ indicates a specific attribute of interest of $\mathscr{C}$. Note that, for the rest of the paper, we consider only forward integration of the integral curves if an unsteady flow is considered; and both forward and backward integration if a steady flow is given. Assume that an integral curve $\mathscr{C}$ is represented by N integration points P_i and $(N-1)$ line segments (P_i, P_{i+1}). We then define a number of attribute fields based on Eq. (1) using the integral curves attributes discussed in [22, 31]. The attribute fields we use to segment the vector fields are as follows. The reason for selecting the following attributes is partially discussed in previous work [39].

- **Rotation Field Φ:** $\Phi_{\mathscr{C}} = \sum_{i=1}^{N-1} d\theta_i$, where $d\theta_i = (\angle(\overrightarrow{P_iP_{i+1}}, \overrightarrow{X}) - \angle(\overrightarrow{P_{i-1}P_i}, \overrightarrow{X})) \in (-\pi, \pi]$ represents the angle difference between two consecutive line segments on an integral curve. $\overrightarrow{X}$ is the X axis of the XY Cartesian space. $d\theta_i > 0$ if the vector field at P_i is rotating counter-clockwise with respect to the vector field at P_{i-1}, while $d\theta_i < 0$ if the rotation is clockwise. Φ field describes the total signed rotation along the trajectory in a global view.
- **Non Straight Velocity Field *nsV*:** $nsV_{\mathscr{C}} = \frac{(Ł - seDist)}{T}$, where T is the time window of the trajectory tracing, Ł is the length of $\mathscr{C}$, and *seDist* is the Euclidean distance between the starting and end points of $\mathscr{C}$. The *nsV* field encodes the flow rotational (or curvature) information, which can be used to identify integral curves that enter vortices.
- **Integrated Jacobian properties** The spatial gradient of $V(\mathbf{x}, t)$ is called its *Jacobian*, denoted by $\mathbf{J}_t = \nabla V(\mathbf{x}, t)$. From $\mathbf{J}_t$, a number of important physical properties can be computed, including vorticity (or curl in 2D), divergence, and its determinant that measures the amount of distortion (e.g, stretching). In this paper we derived the curl field (Figure 2(a)) and the determinant field *determinant* (Figure 2(b)) from the integration of the Jacobian property curl and determinant along pathlines.

2D and 3D attribute fields For a 2D vector field, if the attribute field is computed based on streamlines, it is a 2D field. Figure. 3(a) shows the rotation field of a synthetic steady flow based on streamlines. To visualize the attribute fields, we utilize a blue-white-red color coding scheme, unless stated otherwise, with blue corresponding to negative values and red for positive values. Pathlines-based attribute fields of a 2D unsteady flow are 3D fields. That is, given any spatio-temporal position $(\mathbf{x}, t_0)$, its attribute value is determined by the pathline starting from this position and following the forward flow direction (Eq. 1). Figure. 3(b) (top) shows a volume rendering of the pathline-based Φ field of the Double Gyre flow [30] within the time range $[0, 10]$. Figure. 3(b) (bottom) shows a 2D attribute field at a specific time step, i.e., a cross section of the 3D field shown in Figure. 3(b) (top). Figure. 2 shows a number of attribute fields of the flow behind a cylinder based on pathline tracing staring at $t = 0$ with integral time window size $T = 3$. For a 3D steady flow, the derived fields are 3D attribute fields. Figure 4 provides some examples of 3D attribute fields for two 3D steady flows. The rainbow color coding is applied for the 3D rendering of these two 3D attribute fields whose values are always positive.

In our implementation, we employ a regular sampling strategy to

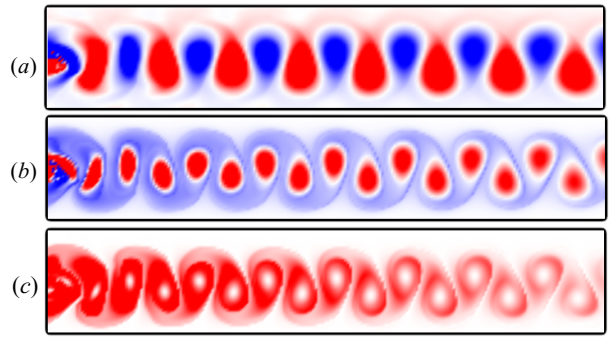

Fig. 2: A number of attribute fields of a 2D unsteady flow behind a square cylinder (a)*curl* field; (b) *determinant* field; (c) *nsV* field.

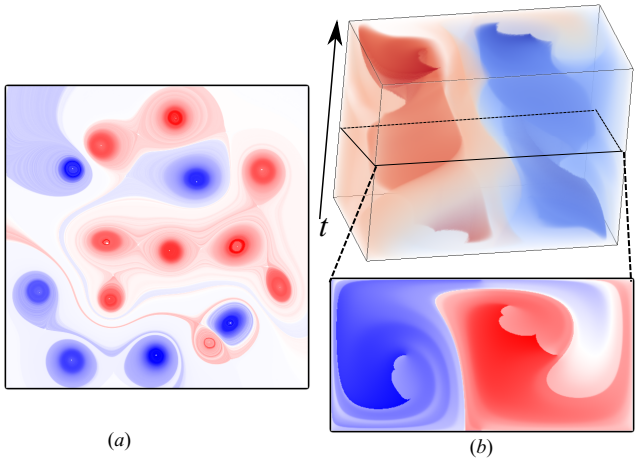

Fig. 3: The rotaion fields Φ of a steady flow and an unsteady flow. (a) the streamline-based rotation field of a synthetic steady flow; (b) The volume rendering (top) of the pathline-based Φ field of the Double Gyre flow. The bottom shows one slice at $t = 5$.

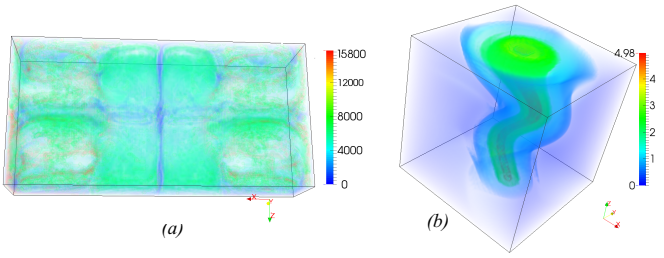

Fig. 4: The 3D attribute fields of two steady flows. (a)*curl* field of a Bernard data ; (b) arc-length field of a Tornado data. Both are visualized with the rainbow color scheme.

compute the attribute fields and the subsequent segmentation. That is, for the 2D steady flow, we partition the flow domain into $N_x \times N_y$ cells, and for 2D unsteady and 3D steady flows, we partition the domain into $N_x \times N_y \times N_z$ cubes. Here, N_x, N_y, N_z are the sampling resolutions along the X, Y, Z axes, respectively. From each seed $\mathbf{x}$ (at the center of each cell or cube), an integral curve is computed using a $4^{th}-$order Runge-Kutta integrator. A linear interpolation scheme is applied in both space and time during integration. In our experiments, N_x, N_y and N_z typically match the resolution of the data set unless stated otherwise. Using the integral curves, the attribute fields are computed and stored at each sampled spatial position. We illustrate the storage of the 2D attribute fields in Figure 5(a), where each cell, i.e. the sample position at the center of the cell, stores the derived attribute field. The cells are labeled with the classification information discussed in Section 4.1.

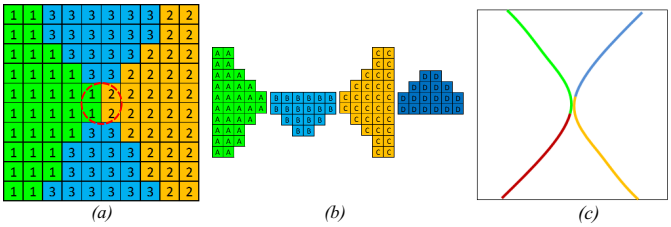

Fig. 5: The illustration of the representation of attribute values and segments. (a) An input attribute field with labels of bin IDs. (b) The segments based on the attribute field. (c) The boundaries of the segments.

4 Segmentation Algorithm

Based on the attribute fields described in Section3.1, we convert the flow segmentation into a scalar field partition problem. Figure 5 illustrates the output segments of the vector fields and their boundaries. Our algorithm can be divided into two steps. As illustrated in Figure 6, the first step is to extract the segments from the region classification (the top row of Figure 6) based on the attribute field values. The second step is to extract and clean the boundaries of the segments (the bottom row of Figure 6). The details are described in Sections 4.1 and 4.2, respectively.

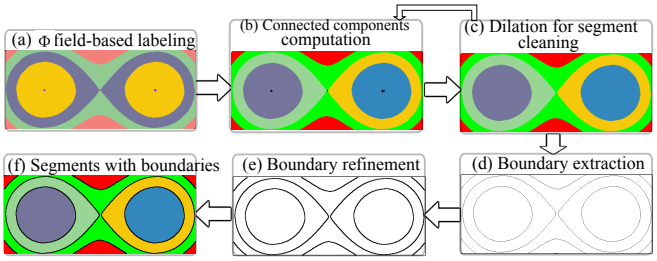

Fig. 6: The pipeline of our segmentation algorithm.

4.1 Segment Extraction and Cleaning

Given a specific attribute field, e.g. the rotation field Φ, our method consists of the following steps.

Region classification We first partition the value range of the attribute field Φ, i.e., $[\Phi_{min}, \Phi_{max}]$, into a finite number of bins $B_i, i = 1...m$. Here, m is a user-controllable parameter. The thresholds of the bins are determined by the distribution of the attribute fields. For simplicity, a uniform distribution is applied so that the bins are evenly distributed within the value range, i.e., the i^{th} bin corresponds to the range $[\Phi_{min} + (i-1)\frac{\Phi_{max}-\Phi_{min}}{m}, \Phi_{min} + i\frac{\Phi_{max}-\Phi_{min}}{m}]$. The sample points in the domain are assigned to m clusters. As shown in Figure 5(a), the sample points are classified into 3 clusters. After this step each sample point is labeled with the ID of the corresponding bin. Those points with the same label belong to the same cluster, even though they may not be physically connected. For instance, the clusters that are colored with yellow in Figure 6(a) are disconnected, even though they are classified into the same bin.

Segment extraction Next, we extract the segments from the clusters. This can be achieved by computing the connected components of those sample points based on their labeled IDs obtained in the previous step. A standard breadth first search algorithm can be used to accomplish this task. The connected components are identified using 4-connectivity in 2D and 6-connectivity in 3D. After identifying the connected components, the sampled points are re-labeled based on the index of the connected components to which they belong. This provides us the initial segmentation of the domain. Figure 6(b) illustrates the result of this step.

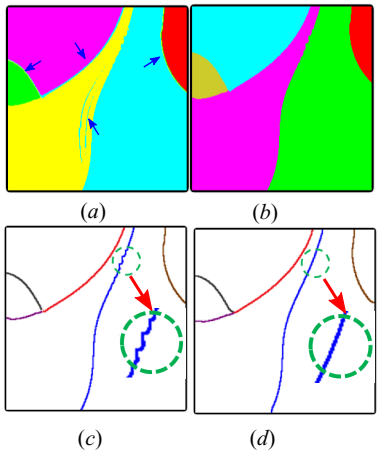

Fig. 7: The illustration of the effect of dilation operation and boundary refinement. (a) The segmentation before dilation operation. (b) The segmentation after dilation operation. (c) The extracted boundaries of (b) without refinement. (d) The extracted boundaries of (b) with refinement.

Segment cleaning via dilation Due to the numerical error in the attribute fields, the above initial segmentation may contain some small segments with little number of sampled points. These small segments will increase the complexity of the segmentation results and lead to visual distraction. Therefore, we need to filter these small or noise segments.

In order to determine whether a small segment is noise or not, we introduce a *noise segment threshold* γ, which is the percentage of the size of the bins. A *noise segment* is a segment who satisfies both of the following conditions: (1) Its size, the number of sample points in this segment, is smaller than the product of γ and the size of the bins in the initial clustering; (2) The attribute value in this segment is close to that in its neighboring segments. In implementation, this can be identified by the difference of their corresponding bin IDs. Figure 7(a) shows a number of noise segments highlighted by arrows. We apply the dilation operation to remove theses noise segments, which is one of the basic morphological operations in image processing [6]. Specifically, we first convert the flow domain into a gray scale image. The intensity of each sample point is determined by the size of the segment it belongs to. The larger the size is, the larger the intensity is. Then, the standard dilation operation is applied to this gray scale image. After dilation operation, the segment ID of each point in the noise segment is changed to the ID of one of its neighboring segments who has the largest size, i.e., with the largest number of sampled points within it. We also extend the dilation to 3D to remove the noise segments in the 3D vector field. Here, we assume a 3D uniform sampling strategy. Given a 3D noise segment, the spatio-temporal point on the boundary is relabeled with the ID of one of the six neighboring segments whose segment size is the largest. Figure 7(b) illustrates the segmentation result after applying dilation to Figure 7(a), where the noise segments

are removed.

4.2 Boundary Extraction

After filtering noise segments, the boundaries of the remaining segments may not be smooth and aligned with the flow direction. A similar issue has been reported by Li et al. [15]. They proposed a boundary refinement method via contours computed from various derived scalar fields or streamlines from the original discrete vector field. However, it requires to compute a large number of contours or streamlines and then select an optimal one with additional computation. Based on the output of our segmentation algorithm, we propose a boundary extraction algorithm which consists of the following two steps.

Extract coarse segment boundaries We describe the extraction of the boundaries for 2D and 3D segmentation separately. In 2D cases, given the unique label for each sample point based on the above segmentation, a well-known normal cut technique [3] can be applied to identify the boundaries between segments. In practice, we estimate the boundary curves between segments using the boundary points of one of the two neighboring segments. In order to smooth the boundary curve, we need to connect these boundary pixels in the correct order.

To achieve that, we first distribute these boundary points into an array. Starting from any point in the array, we trace in two directions along which the boundary is constructed. The tracing is stopped when the next point is on the boundary of the flow domain or when the point has more than one neighboring segment. If a boundary point is next to two neighboring segments, it indicates that one boundary curve of the segment is generated. The four points highlighted in the circle in Figure 5(a) are the ends of several boundary curves (Figure 5(c)). All the boundary curves of a segment have been identified when all the points in the array are traced. However, since the boundary curves of each segment are generated independently, two neighboring segments may share two boundary curves rather than one, as shown in Figure 8(a). So we also need to refine the coarse segment boundaries. For the 3D vector field, we utilize an iso-surface to estimate the boundaries of the 3D segments. Specifically, we first re-assign the ID of a selected 3D segment as 1, and the rest as 0. Therefore, an iso-surface of the value 0.99 would be a close estimation of the boundary of the segment. Some estimated boundaries of the 3D segments are shown in Figure 13(b).

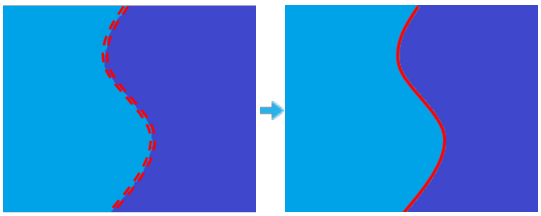

Fig. 8: Boundary refinement.

Refine boundaries A Laplacian smoothing algorithm is applied to refine the boundaries. Each point x on the two boundary curves is replaced with the average position of x and its adjacent neighbors that are also on the boundary curves. As illustrated in Figure 8(right), two shared boundary curves are merged into one with the Laplacian smoothing algorithm. Dilation only removes the noise inside a segment, while there may be noise at or near the boundary of the segment, which is caused by the initial clustering based on the simple range classification of the attribute fields. This can be adjusted according to the flow direction. Figure 7 (c) and (d) illustrates the extracted boundaries of Figure 7 (b). With the adjustment, the extracted boundaries are smoother as highlighted with the red arrows. This Laplacian smoothing can be extended to 3D to smooth the extracted iso-surfaces that correspond to the boundaries of different segments.

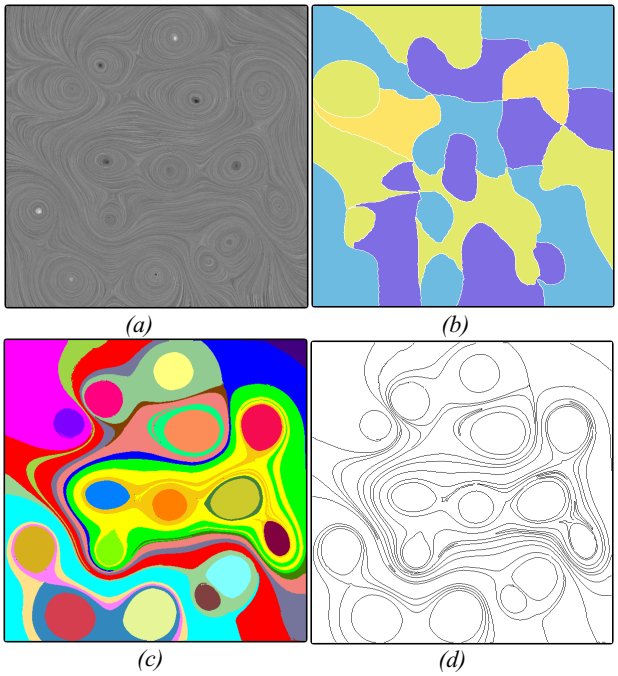

Fig. 9: Comparison between the bottom-up algorithm and our method with a synthetic flow . (a) The LIC of the flow; (b) The segmentation result of the bottom-up algorithm based on the direction of the flow. (c) The segmentation result of our algorithm based on the rotation field. (d) The boundaries of the segments in (c).

5 Results

We have applied our method to a number of synthetic and real-world vector field datasets. The first one is the 2D steady vector field taken from the top layer of a 3D simulation of global oceanic eddies for 350 days of the year 2002 [32]. Each time step corresponds to one day. The 2D unsteady vector field has a spatial resolution of 3600×2400. We extract tiles from the central Atlantic Ocean (60×60). We select the first time step of the dataset, i.e., slice #20106, to generate the 2D steady vector field. In our implementation, the sampling resolution is 600×600. The second example of real-world dataset is a simulation of a 2D unsteady flow behind a square cylinder with a Reynolds number of 160 [36]. This simulation with covers a subset of the spatio-temporal domain, i.e., $[-0.5, 7.5] \times [-0.5, 0.5] \times [15, 23]$. The resolution of the dataset is $400 \times 50 \times 1001$ (number of grid points in x,y,t-direction). We choose the first 200 time steps and use the resolution $400 \times 50 \times 200$ to compute the attribute fields based on pathlines. The time window for pathline computation is 3. Our third read-world dataset is a simulation of a 3D steady flow behind a square cylinder with a Reynolds number of 160 [36]. It covers the spatial domain $[-12, 20] \times [-4, 4] \times [0, 6]$. The spatial resolution of this dataset is $192 \times 64 \times 48$. We compute the attribute fields based on 3D streamlines. With the pre-computed attribute fields, the speed of our segmentation algorithm depends on the resolution of the spatio-temporal domain. It typically takes 2.5 seconds for a 2D steady vector field with 600×600 resolution and 4.25 seconds for a 3D steady vector field with $192 \times 64 \times 48$ resolution on a PC with an Intel Core i7-3537U CPU and 8GB RAM.

5.1 2D steady vector fields

Figure 1(right), Figure 9(c) and Figure 10(column (a)) show the segmentation results for a number of synthetic and real-world 2D steady flows, respectively.

As a comparison, we compute the segmentation using the image-space vector field clustering technique introduced by [21]. It adopts a bottom-up strategy and aggregates two most similar clusters into one each time until only one cluster is left. Figure 9(b) shows the result

of the image-space clustering of a 2D synthetic steady vector field with an error threshold of 0.081. Only showing these segments cannot provide insights into the flow behavior. In comparison, our result as shown in Figure 9(c) generates segments that are better aligned with the flow. By looking at the segments or the boundaries of the segments (Figure 9(d)), one can easily understand the flow behavior. In order to quantify how well the boundaries of the obtained segments are aligned with the input flow, one can adapt the work by [18], which is beyond the scope of this work.

Effects of the number of bins m Figure 10(a) shows the effects of using different numbers of bins for the initial region classification for the Atlantic Ocean dataset. As we can see, with a larger m, more details of the flow are revealed. For example, as highlighted in the dashed area, there are more segments when m is larger and the flow direction, including the sharp turn in the flow, is gradually revealed.

Extracted boundaries *v.s.* Seeded streamlines Column (b) of Figure 10 shows the extracted boundaries of the corresponding segmentations shown in column (a) of Figure 10. These boundaries are similar to the streamlines seeded on the boundaries, as shown in Figure 10(c). The points on a streamline have the same or similar attribute field value, while those on a boundary of one segment have the same range rather than a specific attribute field value. Therefore, the streamlines seeded on the boundaries need not exactly match the boundaries of the segments, but they are sufficiently close to each other.

5.2 2D unsteady vector fields

Figure 11 shows the segmentation results for different attribute fields for a 2D unsteady flow behind a square cylinder, i.e., rotation field, curl field, non straight velocity field *nsV* and determinant field, respectively. The initial bin number m is 5, and the noise segment threshold γ is 0.01. The segments from the rotation field (Figure 11(a)) encode the LCS of the flow, and those from the non straight velocity field *nsV* (Figure 11(c)) reveal the Von Karman vortex street.

Noise segment threshold γ analysis Figure 12 shows the segmentation of the double gyre flow based on the rotation field. When γ increases from 0.01 to 0.05, the four segments in Figure 12(a), highlighted in black, are merged with their neighboring segments after dilation (Figure 12(b)). Interestingly, these four long thin segments have rather different attribute values compared to their neighboring segments. They form two pairs and reveal certain helix and symmetric configuration of the pathlines in the Double gyre flow that are not easy to see with other methods. As shown in Figure 12(d), the pathlines seeded in those segments display similar behaviors in each component.

5.3 3D steady vector fields

Figure 13(a) shows the segmentation of a 3D steady flow behind cylinder based on the rotation field. Eight segments are generated with the initial m as 5. Figure 13(b) shows the estimated boundaries of four segments. The left two are for the segments near the flow domain boundaries and the cylinder, while the right two are for two segments inside the flow domain and far away from the cylinder. As discussed earlier, the boundaries of the segments generated by our method are closely aligned with the flow. This is also applied to 3D segmentation results. Even though a comprehensive comparison is beyond the scope of this work, we believe our 3D segmentation results can be potentially utilized to describe the complex 3D flow behavior in a similar way to stream-surfaces without explicitly placing stream surfaces. We plan to assess this in an extended work.

6 Conclusion

In this work, we propose a vector field segmentation algorithm based on derived attribute fields. A number of attribute fields are firstly computed based on the accumulation of local properties along the integral curves. We then extract the connected components based on the classification of the attribute values and apply dilation to filter the noise segments in the segmentation results. Finally we extract and smooth

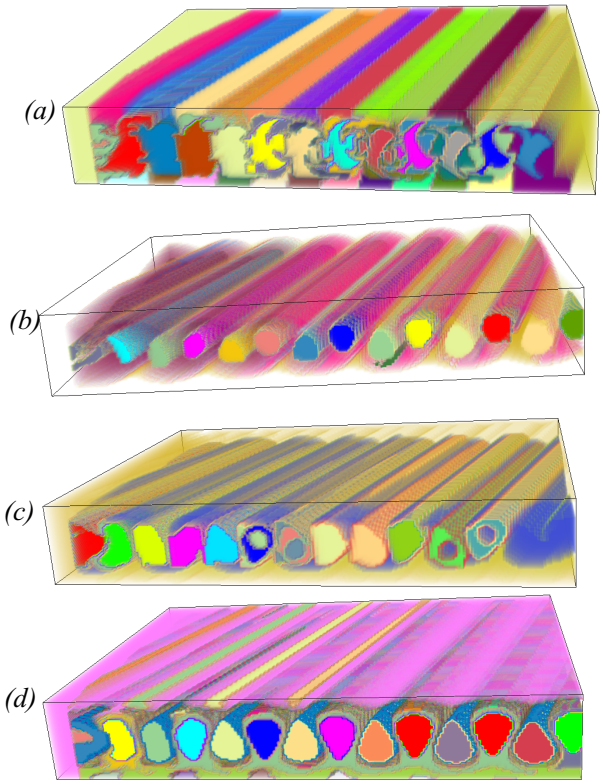

Fig. 11: Segmentation of a 2D unsteady flow behind a square cylinder based on different attribute fields: rotation field (a), *determinant* field (b), *nsV* field (c) and *curl* field (d), respectively.

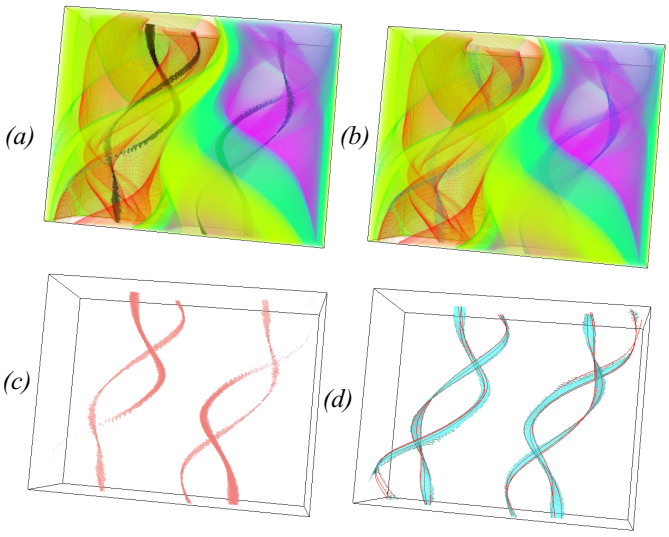

Fig. 12: The segmentation result of an unsteady Double Gyre flow with different noise segment threshold γ. (a) $\gamma = 0.01$; (b)$\gamma = 0.05$; (c) The four γ-sensitive segments; (d) The estimated boundaries of the four γ-sensitive segments with sampled pathlines (red).

the boundaries of the segments in order to obtain a cleaner segmentation for visualization. The segments generated by our algorithm are better aligned with the flow than those obtained from existing local methods. The domain experts can employ various attribute fields to explore different flow behaviors. Our segmentation can be applied to 3D steady flows, where we use iso-surfaces to estimate the boundaries of the segments. We show that these iso-surfaces could potentially be used to visualize high dimensional flows in a similar fashion to integral

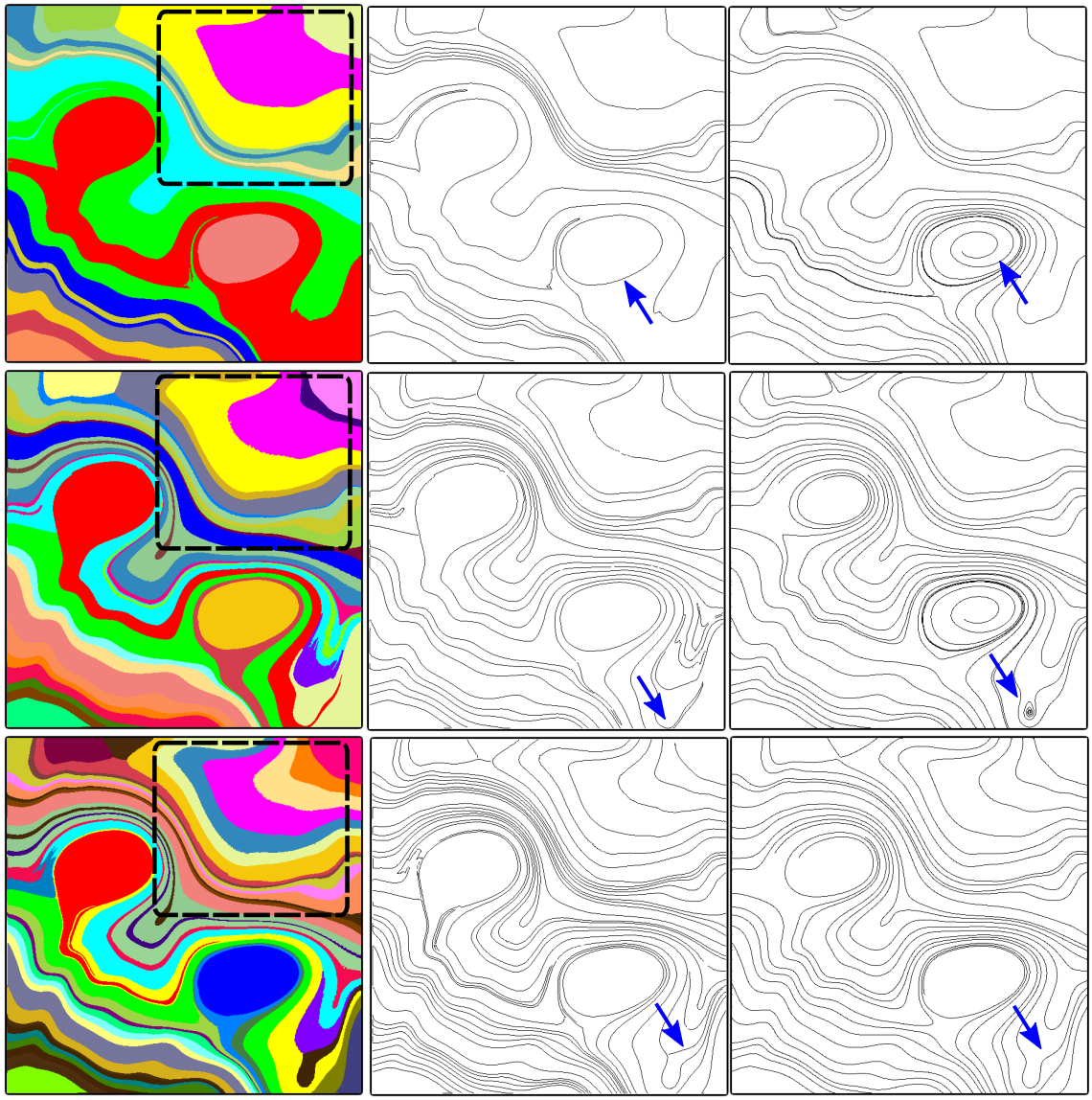

Fig. 10: The effect of the number of bins m for the initial region classification. Column (a) shows the segmentation results based on the rotation field with m as 5, 8 and 15 from top to bottom, respectively. Column (b) are the extracted boundaries of the corresponding segmentation in (a). Column (c) shows the streamlines seeded on the extracted boundaries.

surfaces.

Limitations Since we use a uniform distribution of the attribute fields to partition the range of the attribute field into a number of bins, the initial clusters are sensitive to the number of bins m, especially near the boundaries of the bins. Also, the boundaries of 3D vector field segments are estimated iso-surfaces rather than the real boundaries of the segments. In addition, the criteria to identify noise components should also consider the shape a component besides its size. Finally, the current framework does not apply to large scale 3D unsteady vector fields. We plan to address these limitations in the future.

Acknowledgements

We thank Jackie Chen, Mathew Maltude, Tino Weinkauf for the data. This research was in part supported by NSF IIS-1352722 and IIS-1065107.

References

[1] G. Chen, K. Mischaikow, R. S. Laramee, P. Pilarczyk, and E. Zhang. Vector field editing and periodic orbit extraction using Morse decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):769–785, Jul./Aug. 2007.

[2] G. Chen, K. Mischaikow, R. S. Laramee, and E. Zhang. Efficient Morse decompositions of vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):848–862, Jul./Aug. 2008.

[3] J.-L. Chen, Z. Bai, B. Hamann, and T. J. Ligocki. Normalized-cut algorithm for hierarchical vector field data segmentation. In *Electronic Imaging 2003*, pages 79–90. International Society for Optics and Photonics, 2003.

[4] M. Edmunds, R. S. Laramee, G. Chen, N. Max, E. Zhang, and C. Ware. Surface-based flow visualization. *Computers & Graphics*, 36(8):974–990, 2012.

[5] C. Garth, A. Wiebel, X. Tricoche, K. I. Joy, and G. Scheuermann. Lagrangian visualization of flow-embedded surface structures. *Computer Graphics Forum*, 27(3):1007–1014, 2008.

[6] R. C. Gonzalez. *Digital image processing*. Pearson Education India,

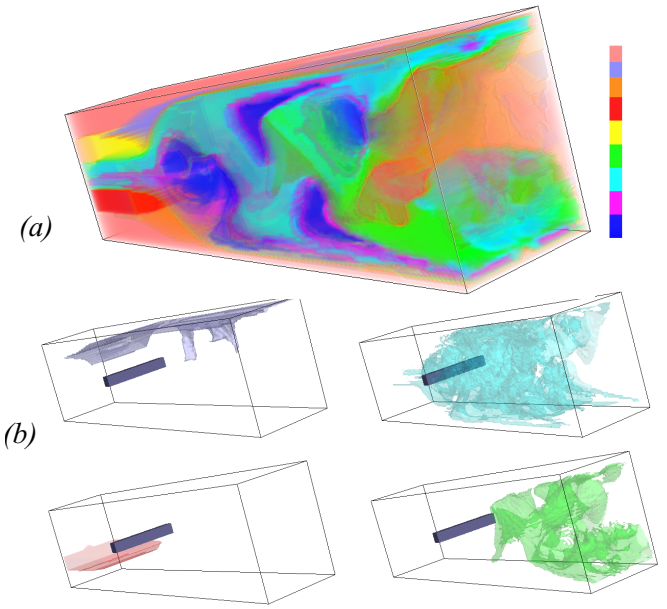

Fig. 13: Segmentation and estimated boundaries of a 3D steady flow behind cylinder. (a) The segmentation result based on the rotation field. (b) Several segmentation boundaries generated from the iso-surfaces.

2009.

[7] M. Guan, W. Zhang, N. Zheng, and Z. Liu. A feature-emphasized clustering method for 2d vector field. In *proceeding of IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 729–733. IEEE, 2014.

[8] G. Haller. Lagrangian coherent structures and the rate of strain in two-dimensional turbulence. *Phys. Fluids A*, 13:3365–3385, 2001.

[9] B. Heckel, G. Weber, B. Hamann, and K. I. Joy. Construction of vector field hierarchies. In *Proceedings of the conference on Visualization'99*, pages 19–25. IEEE Computer Society Press, 1999.

[10] J. L. Helman and L. Hesselink. Representation and display of vector field topology in fluid flow data sets. *IEEE Computer*, 22(8):27–36, August 1989.

[11] M. Jiang, R. Machiraju, and D. Thompson. Detection and visualization of vortices. In *The Visualization Handbook*, pages 295–309. Academic Press, 2005.

[12] A. Kuhn, D. J. Lehmann, R. Gasteiger, M. Neugebauer, B. Preim, and H. Theisel. A clustering-based visualization technique to emphasize meaningful regions of vector fields. In *VMV*, pages 191–198, 2011.

[13] R. Laramee, H. Hauser, L. Zhao, and F. H. Post. Topology based flow visualization: the state of the art. In *Topology-Based Methods in Visualization (Proceedings of Topo-in-Vis 2005)*, Mathematics and Visualization, pages 1–19. Springer, 2007.

[14] F. Lekien, S. Shadden, and J. Marsden. Lagrangian coherent structures in n-dimensional systems. *Journal of Mathematical Physics*, 48(6):Art. No. 065404, 2007.

[15] H. Li, W. Chen, and I.-F. Shen. Segmentation of discrete vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 12(3):289–300, 2006.

[16] K. Lu, A. Chaudhuri, T.-Y. Lee, H. W. Shen, and P. C. Wong. Exploring vector fields with distribution-based streamline analysis. In *Proceeding of IEEE Pacific Visualization Symposium*, Sydney, Australia, march 2013.

[17] H. J. Lugt. Vortex flow in nature and technology. *New York, Wiley-Interscience, 1983, 305 p. Translation.*, 1, 1983.

[18] V. Matvienko and J. Kruger. A metric for the evaluation of dense vector field visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 19(7):1122–1132, 2013.

[19] A. McKenzie, S. V. Lombeyda, and M. Desbrun. Vector field analysis and visualization through variational clustering. In *Proceedings of the Seventh Joint Eurographics / IEEE VGTC conference on Visualization*, EUROVIS'05, pages 29–35, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[20] T. McLoughlin, M. W. Jones, R. S. Laramee, R. Malki, I. Masters, and C. D. Hansen. Similarity measures for enhancing interactive streamline seeding. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1342–1353, 2013.

[21] Z. Peng, E. Grundy, R. S. Laramee, G. Chen, and N. Croft. Mesh-driven vector field clustering and visualization: An image-based approach. *IEEE Transactions on Visualization and Computer Graphics*, 18(2):283–298, Feb. 2012.

[22] A. Pobitzer, A. Lez, K. Matkovic, and H. Hauser. A statistics-based dimension reduction of the space of path line attributes for interactive visual flow analysis. In *PacificVis*, pages 113–120, 2012.

[23] A. Pobitzer, R. Peikert, R. Fuchs, B. Schindler, A. Kuhn, H. Theisel, K. Matkovic, and H. Hauser. The state of the art in topology-based visualization of unsteady flow. *Computer Graphics Forum*, 30(6):1789–1811, September 2011.

[24] K. Polthier and E. Preuß. Identifying vector fields singularities using a discrete hodge decomposition. In *Mathematical Visualization III*, pages 112–134. Ed: H.C. Hege, K. Polthier, 2003.

[25] I. Sadarjoen and F. Post. Geometric methods for vortex extraction. In *Proc. EG/IEEE Visualization Symposium*, 1999.

[26] F. Sadlo and R. Peikert. Efficient visualization of lagrangian coherent structures by filtered amr ridge extraction. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1456–1463, 2007.

[27] T. Salzbrunn, C. Garth, G. Scheuermann, and J. Meyer. Pathline predicates and unsteady flow structures. *The Visual Computer*, 24(12):1039–1051, 2008.

[28] T. Salzbrunn and G. Scheuermann. Streamline predicates. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1601–1612, 2006.

[29] T. Salzbrunn, T. Wischgoll, H. Jänicke, and G. Scheuermann. The state of the art in flow visualization: Partition-based techniques. In H. Hauser, S. Strassburger, and H. Theisel, editors, *In Simulation and Visualization Proceedings*, pages 75–92. SCS Publishing House, 2008.

[30] S. Shadden, F. Lekien, and J. Marsden. Definition and properties of lagrangian coherent structures from finite-time lyapunov exponents in two-dimensional aperiodic flows. *Physica D*, 212(3–4):271–304, 2005.

[31] K. Shi, H. Theisel, H. Hauser, T. Weinkauf, K. Matkovic, H.-C. Hege, and H.-P. Seidel. Path line attributes - an information visualization approach to analyzing the dynamic behavior of 3D time-dependent flow fields. In H.-C. Hege, K. Polthier, and G. Scheuermann, editors, *Topology-Based Methods in Visualization II*, Mathematics and Visualization, pages 75–88, Grimma, Germany, 2009. Springer.

[32] P. Skraba, B. Wang, G. Chen, and P. Rosen. 2d vector field simplification based on robustness. In *IEEE Pacific Visualization Symposium*, pages 49–56, 2014.

[33] A. Telea and J. J. van Wijk. Simplified representation of vector fields. In *Proceedings of the conference on Visualization'99*, pages 35–42. IEEE Computer Society Press, 1999.

[34] H. Theisel, T. Weinkauf, and H.-P. Seidel. Grid-independent detection of closed stream lines in 2D vector fields. In *Proceedings of the Conference on Vision, Modeling and Visualization 2004 (VMV 04)*, pages 421–428, Nov. 2004.

[35] X. Tricoche, G. Scheuermann, and H. Hagen. Continuous topology simplification of planar vector fields. In *Proceedings of IEEE Visualization 2001*, pages 159–166, 2001.

[36] T. Weinkauf and H. Theisel. Streak lines as tangent curves of a derived vector field. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1225–1234, November - December 2010.

[37] T. Wischgoll and G. Scheuermann. Detection and visualization of closed streamlines in planar fields. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):165–172, 2001.

[38] H. Yu, C. Wang, C.-K. Shene, and J. H. Chen. Hierarchical streamline bundles. *IEEE Transactions on Visualization and Computer Graphics*, 18(8):1353–1367, Aug. 2012.

[39] L. Zhang, R. S. Laramee, D. Thompson, A. Sescu, and G. Chen. Compute and Visualize Discontinuity Among Neighboring Integral Curves of 2D Vector Fields. In *Proceedings of TopoInVis*, Germany, 2015.