Noname manuscript No.
(will be inserted by the editor)

An Integral Curve Attribute Based Flow Segmentation

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

Received: date / Accepted: date

Abstract We propose a segmentation method for vector fields that employs geometrical curve attributes to classify the behavior of the integral curves. In particular, we assign to a given spatio-temporal position the attribute value associated with the integral curve initiated at that point. With this attribute information, our segmentation strategy first performs a region classification. Then, 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 points that are correlated by the same integral curve have the same or similar attribute values, the proposed segmentation method naturally generates segments whose boundaries are better aligned with the flow direction. Therefore, additional processing is not required to generate other geometric descriptors within the segmented regions to illustrate the flow behaviors. We apply our method to a number of synthetic and CFD simulation data sets and compare their results with existing methods to demonstrate its effectiveness.

Keywords Vector field data · integral curves · flow visualization · segmentation.

1 Introduction

Vector field analysis is a ubiquitous tool employed to study a wide range of dynamical systems involved in numerous applications, such as automobile and aircraft engineering, climate study, combustion dynamics, earthquake engineering, and medicine, among others. As the size and complexity of generated vector filed data sets increase continuously, many partition-based techniques have been developed to generate a thorough description of the behavior of vector fields.

Existing techniques typically partition the flow domain based on either certain local flow characteristics, such as vector magnitude and orientation, or 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 be aligned with the flow direction (see Figure 7(b) for an example). Therefore, additional computation is needed in order to generate other visual primitives, such as streamlets or glyphs [21], to convey information about the flow behavior within each segment. On the other hand, vector field topology [14,23] reveals the essential flow structure and partitions the flow domain into regions with homogeneous behaviors. However, it has yet to be applied to unsteady flow and its visual representation in high dimensional space can be too complex to be useful to domain experts.

Integral curve attributes have been recently applied to cluster [17] and select [39] 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 an inadequate spatial sampling by the integral curves. To remedy this, very dense integral curves can be computed as input, which will significantly increase the memory and storage requirements during computation. In the meantime, integral surfaces may be better descriptors than integral curves for depicting important flow dynamics in higher-dimensional spaces, although

Lei Zhang, Guoning Chen (✉)
University of Houston, Houston, TX, 77081
E-mail: {gchen16}@uh.edu

Robert S. Laramee
Swansea University, Wales, UK

David Thompson, Adrian Sescu
Mississippi State University, Mississippi State, MS, 39762

integral surface placement is a much harder problem than integral curve placement [4].

In this paper, we introduce a segmentation framework based on integral curve attributes applicable to both 2D and 3D vector fields. Our method combines the advantages of vector field clustering and integral curve attribute approaches and attempts to achieve an intuitive and expressive segmentation of the flow domain whose resulting segment boundaries are aligned with the flow. Our framework consists of two major components (Figure 1). 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 is set equal to the attribute value of the integral curve that is seeded at this position. The attribute value of an integral curve is computed by integrating local properties along the curve [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 [40]. Using 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 reference [12]. Compared to [12], where the goal was to generate implicit clusters produced by semantic dependencies, our method generates a segmentation based on 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 present in the attribute field computation. We then perform dilation operations to remove those small segments. After filtering the initial segmentation, the boundaries of the obtained segmented regions may still be non-smooth. To remedy this, we further smooth these boundaries. Unlike the boundary refinement strategy applied in [16] our method does not require computing many contours and comparing them to the coarse segmentation curves.

In summary, our contributions are as follows:

- We introduce an integral curve attribute based flow segmentation framework, which generates segments that are better aligned with the flow direction than those produced by existing local methods. In addition, our framework is flexible and efficient in that various attribute fields can be used to guide the segmentation and help users investigate 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 the effectiveness of the method by applying it 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 (see Figure 11 and Figure 12).

The rest of the paper is structured as follows. Section 2 reviews the previous work related to the proposed method. Section 3 briefly reviews the important concepts of vector fields and the attribute fields as well as their computation. Section 4 describes the segmentation algorithm based on the integral curve attributes and the refinement of the segment boundaries. The applications of the segmentation algorithm to a number of steady and unsteady flows are reported in Section 5, followed by a discussion of the limitations of our work in Section 6.

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 [4,11,14,23] that provide systematic classifications of various analysis and visualization techniques for vector field data. 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, 34,38]. 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 in time-dependent flows, by computing the *Finite Time Lyapunov Exponent (FTLE)* of the flows [8,15,30]. Since its introduction, FTLE has been compared with the separatrices for 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 streamlines within vortical regions [25]. Salzbrunn and Scheuermann [27] introduced *streamline predicates*, which classifies streamlines by interrogating them as

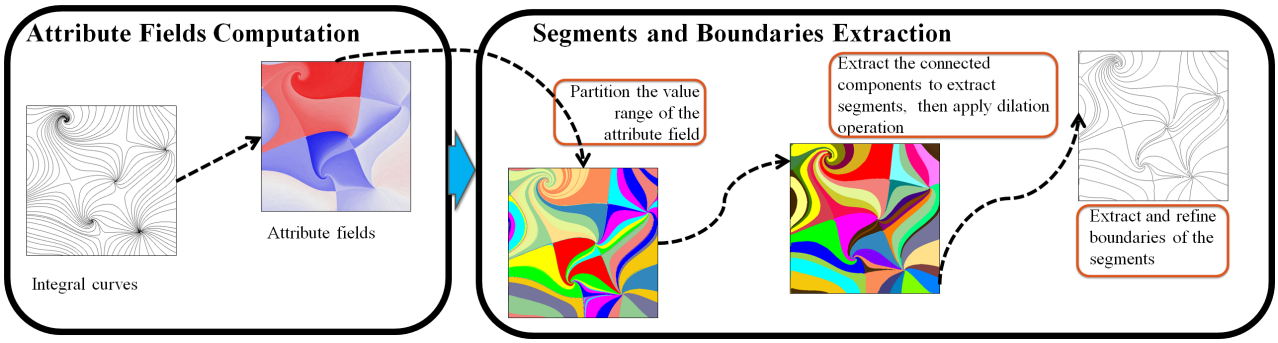

Fig. 1 The pipeline of our method.

they pass through certain features, e.g., vortices [28]. Later, this approach was extended to the classification of pathlines. At the same time, Shi et al. [31] presented a data exploration system to study the different characteristics of pathlines based on their various attributes, including winding angle. 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 approach segments the flow domain in regions where pathlines are located. There are other techniques for classifying streamlines that are constrained to flow separation and vortex structures, such as the streamline bundling technique [39]. More recently, McLoughlin et al. [20] introduced the idea of a streamline signature based on a set of curve-based attributes including curvature and torsion. This streamline signature is used to compute the similarity between streamlines and help domain experts place and filter streamlines to create an informative and uncluttered visualizations 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. An image-space, mesh-driven vector field clustering algorithm was introduced by Peng et al. [21]. They provided a bottom-up approach to generate a hierarchical clustering 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. [16] proposed a 2D vector field segmentation based on the 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 visualized them by means of compact cluster boundaries. More partition-based techniques for flow segmentation are described in the survey by Salzbrunn et al. [29].

3 Vector Field Background and Trajectory Attributes

Consider a d-manifold $\mathbb{M} \subset \mathbb{R}^d (d = 2,3)$, 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 reveal different aspects of the translational property of vector fields. In this work, we focus on only streamlines and pathlines derived from 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 trajectories of massless particles released in the flow domain at a given time t_0: $\mathbf{x}(t) = \mathbf{p}_0 + \int_{t_0}^{t} V(\mathbf{x}(\eta);t_0+\eta)d\eta$.

3.1 Attribute Fields

Various attributes can be extracted for the analysis and classification of integral curves [22,31]. Among these attributes, many can be obtained by accumulating 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 position is determined by the integral curve initiated at 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 [40]:

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

where $\mathscr{C}(\mathbf{x})|_{t_0}^{t_0+T}$ denotes an integral curve, i.e., either a streamline or a pathline initiated at $\mathbf{x}$ and computed within an integral time window $[t_0, t_0+T]$. $\mathscr{F}(\cdot)$ represents an attribute of interest of $\mathscr{C}$.

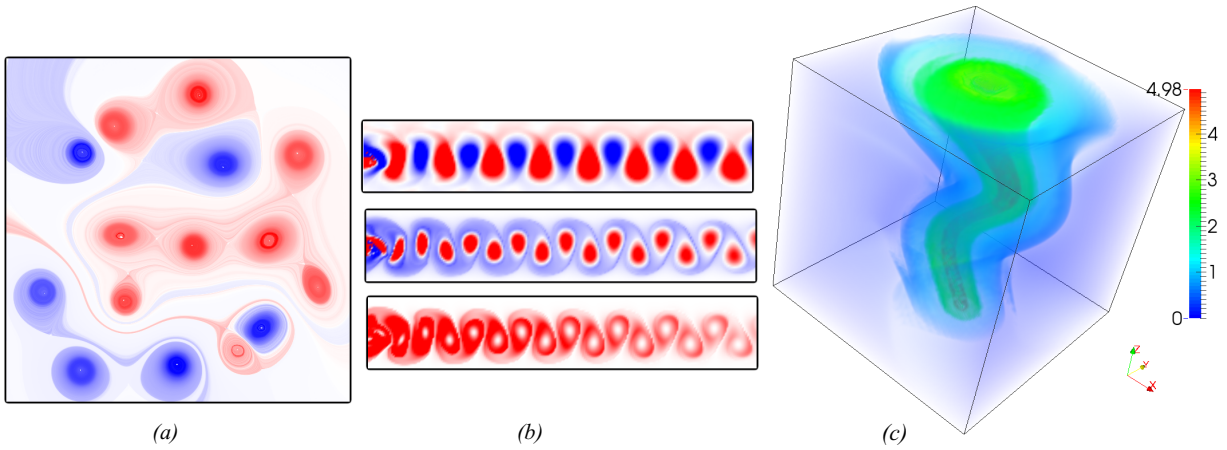

Fig. 2 The visualization of some attribute fields. (a) The rotation field Φ of a steady flow. (b) *curl*, *determinant* and *nsv* fields (from top to bottom) of a 2D unsteady flow behind a square cylinder. (c) The arc-length field of a Tornado data. A blue-white-red color coding is used for 2D attribute fields with blue for negative values and red for positive ones (a, b). Rainbow color coding is used to render a 3D volumetric attribute fields with non-negative values (c).

The attribute fields that we use to segment the vector fields include: (1) rotation field Φ – the winding angles of $\mathscr{C}$; (2) non-straight velocity field *nsV* – an attribute field encoding the characteristics of straightness of $\mathscr{C}$; (3) arc-length field *arc* – the length of $\mathscr{C}$; (4) *curl* and *determinant* field – the integration of the curl magnitude and the determinant of Jacobian along $\mathscr{C}$. Figure 2 shows a number of attribute fields in 2D and 3D. We refer the interested readers to [40] for more details about attribute fields and their visualization.

In our implementation, we employ a regular sampling strategy to 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 (or **t** for the unsteady flows) axes, respectively. From each seed **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 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 3(a), where each cell, whose center is a sample position, stores the derived attribute field. The cells are labeled with the classification information discussed in Section 4.1.

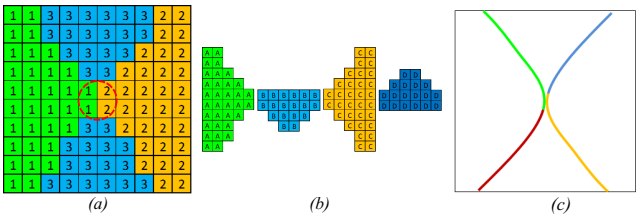

Fig. 3 The illustration of the representation of attribute values and segments. (a) An input attribute field with labels of bin IDs. (b) The segments based on the attribute field. (c) The boundaries of the segments.

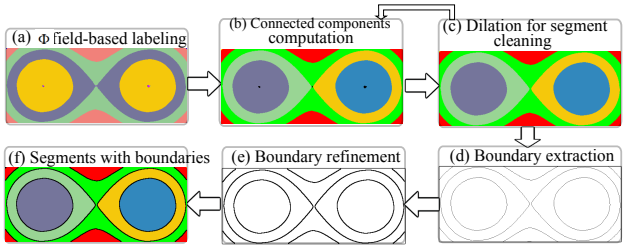

Fig. 4 The pipeline of our segmentation algorithm.

4 Segmentation Algorithm

Based on the attribute fields described in Section 3.1, we convert the flow segmentation problem into a scalar field partitioning problem. Figure 3 illustrates the output segmented regions of the vector fields and their boundaries. Our algorithm can be divided into two steps. As illustrated in Figure 4, the first step is to extract the segments from the region classification (top row of Figure 4) based on the attribute field values. The second step is to extract and clean the boundaries of the segments (bottom row of Figure 4). The details are described in Sections 4.1 and 4.2, respectively.

4.1 Segment Extraction and Cleaning

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

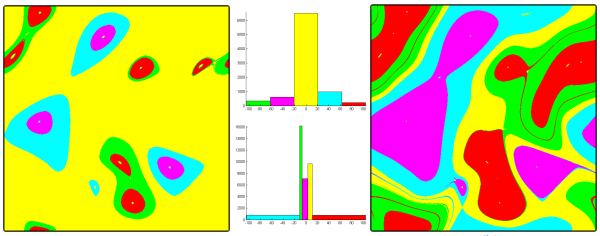

Fig. 5 Region classification based on different methods. (a) uniform distribute on data range. (b) equal-sizes bins. The histograms show the data distribute under each method. The bin number $m = 5$.

Region classification *We first classify different flow regions based on their attribute field values. A simple strategy is to evenly partition the data range of the attribute field, i.e., the i^{th} bin ($i \in [0, m-1]$) corresponds to the range $[\Phi_{min} + (i-1)\frac{\Phi_{max}-\Phi_{min}}{m}, \Phi_{min} + i\frac{\Phi_{max}-\Phi_{min}}{m}]$. The data points are labeled as in the i^{th} cluster based on the attribute field value defined on it (e.g., Figure 3(a)). However, this simple strategy may result in initial region classification with largely different sizes, as shown in Figure 5(a). An alternative way to generate m clusters is to equalize the size of bins. That is, the number of data points falling into each bin is the same. This generally yields better initial partitioning (Figure 5(b)). Without further specification, most 2D segmentation results shown in this paper adopts the second initial partitioning strategy. For the 3D flow data sets, we perform the region classification manually based on the data distribution histogram to avoid generating too large bins (i.e., clusters with too many points). However, we do not enforce equal size requirement in 3D. We provide the actual partitioning strategy via the legends aside the results of 3D data sets.*

Segment extraction After the preceding step each sample point is labeled with the ID of the corresponding bin (i.e., cluster ID). Next, we extract the segments from these 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, respectively. 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 4(b) illustrates the result of this step.

Segment cleaning via dilation Due to the numerical error in the attribute fields, the above initial segmentation may contain small segments with only a small number of sampled points. These small segments will increase the complexity of the segmentation and lead to visual distraction. Therefore, we need to remove 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 a percentage of the size of the bins. A *noise segment* is a segment that satisfies both of the following conditions: (1) Its size, i.e., 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 differences in the corresponding bin IDs. Figure 6(a) shows a number of noise segments highlighted by arrows. We apply the dilation operation, which is one of the basic morphological operations in image processing [6], to remove these segments. 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 to which it belongs. That is, the larger the size of the region, the larger the intensity of the sample. Then, the standard dilation operation is applied to this gray scale image. After the dilation operation, the segment ID of each point in the noise segment is changed to the ID of its neighboring segments with the largest size, i.e., with the largest number of sampled points. We also extend the dilation to 3D to remove the noise segments in 3D vector fields. 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 its six neighboring segments whose segment size is the largest. Figure 6(b) illustrates the segmentation result after applying dilation to Figure 6(a), where the noise segments are removed.

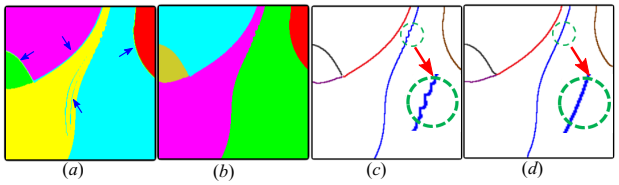

Fig. 6 The illustration of the effect of dilation operation and boundary refinement. (a) The segmentation before dilation operation. (b) The segmentation after dilation operation. (c) The extracted boundaries of (b) without refinement. (d) The extracted boundaries of (b) with refinement.

Note that, when equal-size bin partitioning strategy is applied to generate the region classifications, the above noise segmentation threshold is equivalent to using the data size (e.g., the diameter of the bounding box of the data domain) multiplied by γ. Nonetheless, we opt for the local strategy described above, as the initial partitioning need not be equal size (e.g., in 3D cases). Using a global threshold to remove noise segments may lose detailed information in those smaller segments (i.e., initial clusters that have less sample points).

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. [16]. They proposed a boundary refinement method via contours computed from various derived scalar fields or streamlines from the original discrete vector field. However, their method requires to compute a large number of contours or streamlines and select an optimal one with additional computation. Based on the output of our segmentation algorithm, we propose a boundary extraction algorithm that consists of two steps. In the following we first describe our method to handle 2D segmentation, followed by a discussion of the processing of 3D cases.

Extracting initial segment boundaries In 2D cases, given the unique label for each sample point based on the above segmentation, the 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 extract boundary curves, we first store the 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 two or more neighboring segments. If the boundary point that already belongs to a segment is also next to other two segments, it indicates that one boundary curve of the segment is generated. The four points highlighted in the circle in Figure 3(a) are the ends of several boundary curves (Figure 3(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. So we need to refine the initial segment boundaries.

Boundary refinement A Laplacian smoothing algorithm is applied to refine the boundaries. Each point $\mathbf{x}$ on the two boundary curves is replaced with the average position of $\mathbf{x}$ and its adjacent boundary neighbors. Thus two neighboring boundary curves can be merged into one. In addition, 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 6 (c) and (d) illustrate the extracted boundaries of Figure 6 (b) before (c) and after (b) refinement, respectively. With this adjustment, the extracted boundaries are smoother as highlighted with the red arrows.

For a 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 Figures 11 and 12. The Laplacian smoothing strategy for the refinement of boundaries can be extended to 3D to smooth the extracted iso-surfaces that correspond to the boundaries of different segments.

5 Results

We have applied our method to a number of synthetic and real-world vector field datasets, including steady and unsteady flows for both 2D and 3D. 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. For the 3D steady data sets considered, 4.25 seconds are required to compute a segmentation. For the 3D unsteady ABC flow, it takes 14 minutes. All numerical experiments are carried out on a PC with an Intel Core i7-3537U CPU and 8GB RAM.

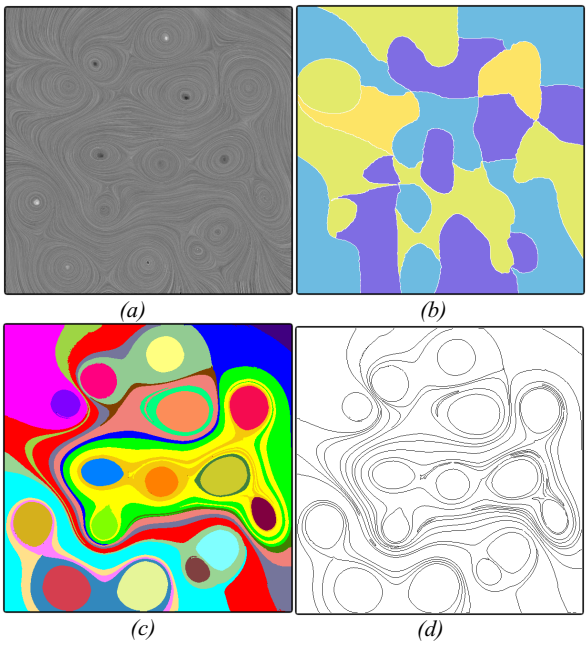

Fig. 7 Comparison between the bottom-up algorithm and our method with a synthetic flow . (a) The LIC of the flow; (b) The segmentation result for the bottom-up algorithm based on the direction of the flow. (c) The segmentation result for our algorithm based on the rotation field. (d) The boundaries of the segments in (c).

5.1 2D steady vector fields

Figure 1(right), Figure 7(c) and Figure 8(column (a)) show our 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 Peng et al. [21]. It adopts a bottom-up strategy and aggregates the two most similar clusters each time until only one cluster is left. Figure 7(b) shows the result of the image-space clustering of a 2D synthetic steady vector field with an error threshold of 0.081. However, only showing these segments cannot provide insights into the flow behavior. In comparison, our result as shown in Figure 7(c) generates segments whose boundaries are better aligned with the flow. By looking at the segments or the boundaries of the segments (Figure 7(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 Matvienko and Kruger [18], which is beyond the scope of this work. Additional comparisons can be found in the supplemental document.

Effects of the number of bins m Figure 8(a) shows the effects of using different numbers of bins for the initial region classification for the Atlantic Ocean dataset, which is taken from the top layer of a 350-day 3D simulation of global oceanic eddies for the year 2002 [32]. Each time step corresponds to one day. We select the first time step of the dataset, i.e., slice #20106, to generate the 2D steady vector field. 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 Figure 8 (b) shows the extracted boundaries of the corresponding segmentations shown in Figure 8 (a). These boundaries are similar to the streamlines seeded on the boundaries, as shown in Figure 8 (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, as indicated by the arrows in Figure 8 (b) and (c), but they are sufficiently close to each other.

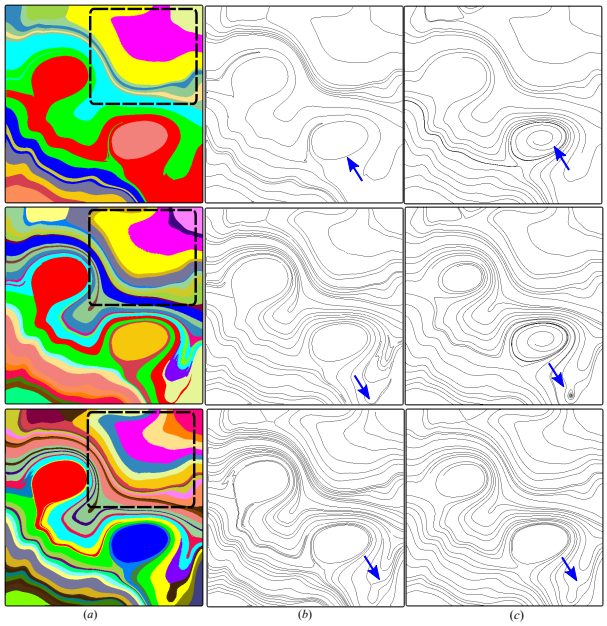

Fig. 8 The effect of the number of bins m for the initial region classification. Column (a) shows the segmentation results based on the rotation field with m as 5, 8 and 15 from top to bottom, respectively. Column (b) are the extracted boundaries of the corresponding segmentation in (a). Column (c) shows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

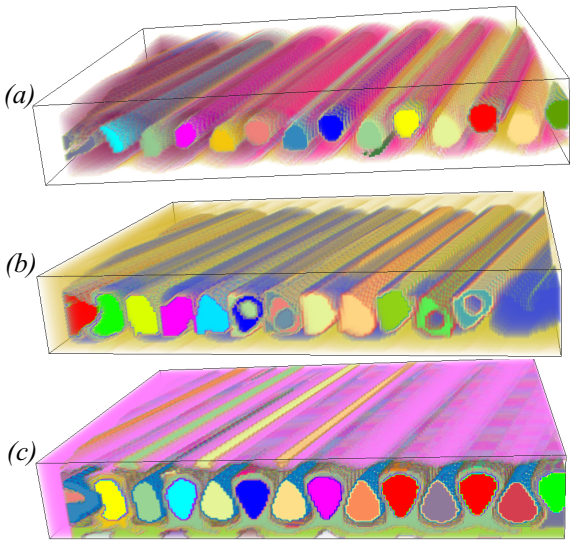

Fig. 9 Segmentation of a 2D unsteady flow behind a square cylinder based on different attribute fields: *determinant* field (a), *nsV* field (b) and *curl* field (c), respectively. The bin number $m = 6$.

5.2 2D unsteady vector fields

The second example of a real-world dataset is a simulation of a 2D unsteady flow behind a square cylinder with a Reynolds number of 160 [36]. This simulation covers a subset of the spatio-temporal domain, $[-0.5, 7.5] \times [-0.5, 0.5] \times [15, 23]$. 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 a resolution of $400 \times 50 \times 200$ to compute the attribute fields based on pathlines. The time window for the pathline computation is 3. Figure 9 shows the segmentation results of this dataset based on different attribute fields, i.e., determinant field, non-straight velocity

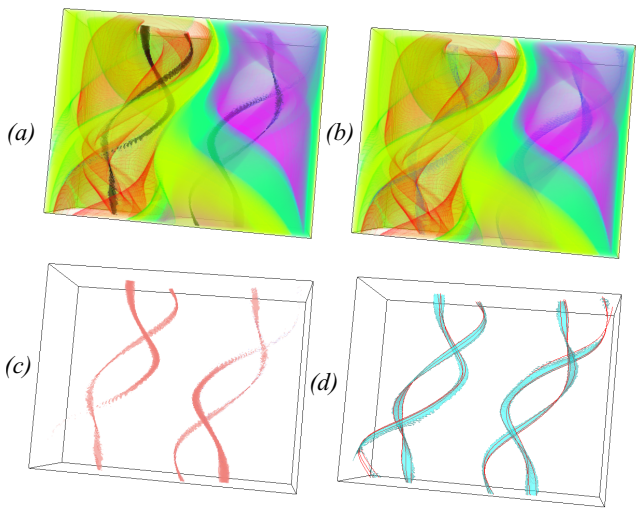

Fig. 10 The segmentation result of an unsteady Double Gyre flow with different noise segment threshold γ. (a) $\gamma = 0.01$; (b)$\gamma = 0.05$; (c) The four γ-sensitive segments; (d) The estimated boundaries of the four γ-sensitive segments with sampled pathlines (red). The bin number $m = 6$.

field *nsV*, and curl field, respectively. The bin number m is 6, and the noise segment threshold γ is 0.01. The segments from the curl field (Figure 9(c)) encode the LCS information of the flow and those from the non straight velocity field *nsV* (Figure 9(b)) reveal the Von Karman vortex street.

Noise segment threshold γ Figure 10 shows the segmentation of the Double Gyre flow [30] based on the rotation field. When γ increases from 0.01 to 0.05, the four segments, highlighted in black in Figure 10(a), merge with their neighboring segments after dilation (Figure 10(b)). Interestingly, these four long thin segments have rather different attribute values compared to their neighboring segments. They reveal a symmetric, helical configuration of the pathlines in the Double Gyre flow that are not easy to see with other methods. As shown in Figure 10(d), the pathlines seeded in those regions display similar behaviors in each region.

5.3 3D steady vector fields

To test out method in a steady 3D flow framework we take an instant from a 3D time-dependent 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. Figure 11(a) shows the segmentation of this dataset based on the curl field. Eight segments are generated with the bin number $m = 3$. Figure 11(b) shows the estimated boundaries of two segments by iso-surfaces. The left image shows a segment inside the flow domain, where the curl field has large values. The right image shows a segment near the flow domain boundaries, where the curl field has small values.

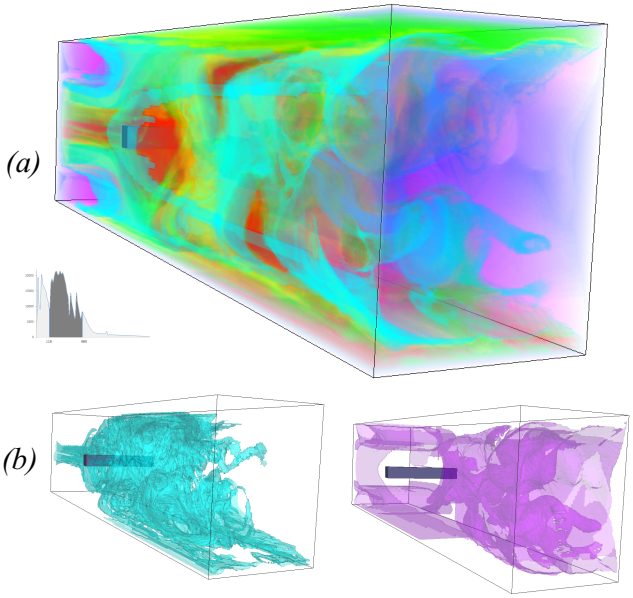

Fig. 11 Segmentation and estimated boundaries of a 3D steady flow behind cylinder. (a) The segmentation result based on the curl field. (b) Several segmentation boundaries generated using the iso-surfaces. The bin number $m = 3$. The legend aside shows the partitioning strategy for initial region classification.

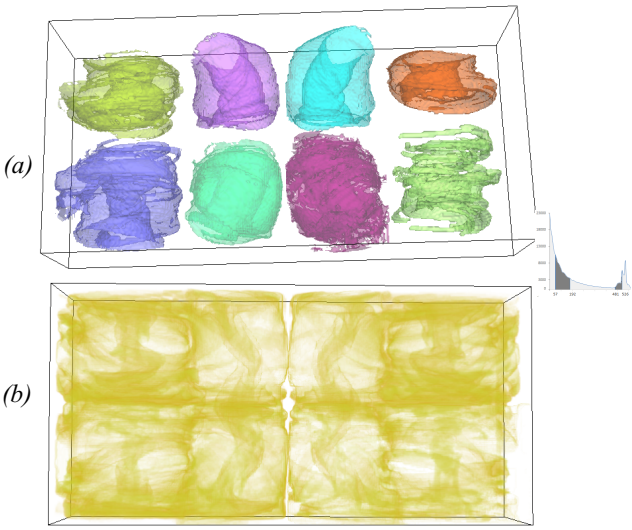

Fig. 12 Segmentation and estimated boundaries of the Bernard flow. (a) Eight segmentation boundaries generated from the iso-surfaces. (b) one of the generated segments. The bin number $m = 5$. The legend aside shows the partitioning strategy for initial region classification.

Figure 12 shows the segmentation and estimated boundaries of the Bernard flow [37], whose domain is $[-16,16] \times [-4,4] \times [-8,8]$. From the eighteen segments generated with bin number $m = 5$, we select eight whose average rotation field value is above 525.6. The estimated boundaries of the eight segments shown in Figure 12 (a) highlight the eight

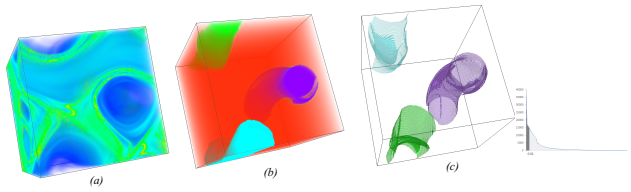

Fig. 13 Results of the unsteady ABC flow projected into $t = 0$. (a) Visualization of the rotation field. (b) Four segments of the rotation field with bin number $m = 2$. (c) Three segmentation boundaries generated from the iso-surfaces. The bin number $m = 2$. The legend aside shows the partitioning strategy for initial region classification.

vortex regions of the Bernard flow. There are also some segmented regions covering the whole flow domain rather than a specific vortex region. Figure 12 (b) shows such a segment.

As discussed earlier, the boundaries of the segments generated by our method are closely aligned with the flow. This is also true for the 3D segmentation results. Even though a comprehensive comparison is beyond the scope of this work, we believe that our 3D segmentation results can potentially be 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.

5.4 3D unsteady vector fields

We also applied our framework to a 3D unsteady flow, i.e., the unsteady ABC vector field [13]. We defined the parameters to be $A = \sqrt{3} + 0.5tsin(\pi t)$, $B = \sqrt{2}$ and $C = 1$. The spatial domain we consider is $[0, 2\pi]^3$. The attribute fields of the 3D unsteady flow are defined in a 4D space and are based on pathlines. The time window for pathlines computation is 10. Figure 13(a) is the visualization of the rotation field projected into the time slice $t = 0$. With the bin number $m = 2$, four segments are generated (Figure 13(b)). Three of them highlight the vortex regions of the flow, whose boundaries are estimated with the iso-surfaces shown in Figure 13(c).

6 Conclusion

In this work, we propose a vector field segmentation algorithm based on derived attribute fields. A number of attribute fields are 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 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. 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 surfaces.

Limitations The initial region classification used in our pipeline is rather simple and may require user adjustment. A more heuristic strategy needs to be developed. Also, the boundaries of 3D vector field segments are currently some estimated iso-surfaces rather than the real boundaries of the segments. In addition, the criteria employed to identify noise components should also consider the shape of a component in addition to its size. Take a small segment around the center point of a vortex as an example, if we only consider the size of this segment and ignore its shape, it may be classified as a noise segment and be filtered out, which is not desired. *In fact, the shape information of the segments can be described by their 1D skeletons. Furthermore, the current segmentation framework does not consider the well-known flow features, such as topology. However, we believe that the gradient of the attribute fields, which has been shown relevant to a number of flow features [40], can be employed to guide the segmentation process.* Finally, the current framework does not apply to large scale 3D unsteady vector fields because of memory limitations (due to the attribute field computation). We plan to address these limitations in the future.

Acknowledgments 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, 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. P. Kundu and I. Cohen. Fluid mechanics. 2004. *Elsevier Academic Press, San Diego). Two-and three-dimensional self-sustained flow oscillations*, 307:471–476, 2008.
14. 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.
15. F. Lekien, S. Shadden, and J. Marsden. Lagrangian coherent structures in n-dimensional systems. *Journal of Mathematical Physics*, 48(6):Art. No. 065404, 2007.
16. 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.
17. 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.
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. D. Weiskopf, T. Schafhitzel, and T. Ertl. Texture-based visualization of unsteady 3d flow by real-time advection and volumetric illumination. *Visualization and Computer Graphics, IEEE Transactions on*, 13(3):569–582, 2007.
38. 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.
39. 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.
40. 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.