

**FLOW VISUALIZATION AND ANALYSIS: FROM GEOMETRY
TO PHYSICS**

A Dissertation Presented to
the Faculty of the Department of Computer Science
University of Houston

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

By
Lei Zhang
August 2017

FLOW VISUALIZATION AND ANALYSIS: FROM GEOMETRY TO PHYSICS

Lei Zhang

APPROVED:

Guoning Chen, Chairman
Computer Science Department, University of Houston

David Thompson
Aerospace Engineering, Mississippi State University

Zhigang Deng
Computer Science Department, University of Houston

Christoph Eick
Computer Science Department, University of Houston

Dean, College of Natural Sciences and Mathematics

Acknowledgements

This dissertation could not have been completed without the great support that I have received from so many people over the years.

First and most important, I would like to express my special appreciation and thanks to my advisor Professor Dr. Guoning Chen, you have been a tremendous mentor for me. I would like to thank you for encouraging and supporting my research. Your advice on both research as well as on my life have been invaluable. I feel very lucky and proud to be your student. During the last five years, we have one-on-one meetings every week. Thank you so much for your time and effort on helping me growing and growing. The most correct decision I've ever made in the last five years is to choosing you as my advisor.

I would also like to thank my committee members Prof. Zhigang Deng, Prof. David Thompson and Prof. Christoph F Eick for serving as my committee members and for your time and valuable advice.

I would especially like to thank my colleagues: Xifeng Gao, Kaoji Xu, Marizeh Berenjkoub, Lieyu Shi, and Duong Nguyen for the discussion and help. The time spent with you is always a good memory for me.

Finally I thank my wife Fei Liu, my parents and my parents-in-law. I could not have finished my PhD program without your support. Thank you for your accompany and encouragement. Thanks my son Bohan Zhang for bringing so much fun and happiness in my life.

FLOW VISUALIZATION AND ANALYSIS: FROM GEOMETRY TO PHYSICS

An Abstract of a Dissertation
Presented to
the Faculty of the Department of Computer Science
University of Houston

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

By
Lei Zhang
August 2017

Abstract

As the size and complexity of flow data sets continuously increase, many vector field visualization techniques aim to generate an abstract representation of the geometric characteristics of the flow to simplify its interpretation. However, most of the geometric-based visualization techniques lack the ability to reveal the physically important features. Additional efforts are needed to interpret the physical characteristics from the geometric representation of the flow.

In this work, the Lagrangian accumulation framework is introduced first, which accumulates various local physical and geometric properties of individual particles along the associated integral curves. This accumulation process results in a number of attribute fields that encode the information of certain global behaviors of particles, which can be used to achieve an abstract representation of the flow data. This framework is utilized to aid the classification of integral curves, produce texture-based visualizations, study property transport structures, and identify discontinuous behaviors among neighboring integral curves.

Although the accumulation framework is simple and effective, the detailed flow behavior at individual integration points (and times) along the integral curves is suppressed, leading to incomplete analysis and visualization of flow data. In order to achieve a more detailed exploration, a new flow-exploration framework is investigated based on the time-series data or Time Activity Curves (TAC) of local properties. In this framework, the physical behavior of the individual particles can be described via their respective TACs. An event detector based on TACs is proposed to capture the local and global similarity of any spatial point with its neighboring points with a new dissimilarity metric. A hierarchical clustering framework is then developed based on this metric, upon which a level-of-detail

representation of the flow can be obtained. This new framework is applied to a number of 2D and 3D unsteady-flow data sets to demonstrate its effectiveness.

Contents

1	Introduction	1
2	Related Work	7
2.1	Vector Field Topological Analysis	7
2.2	Streamline and Pathline Attributes	9
2.3	Lagrangian Framework for Flow Analysis	10
2.4	Time-varying Data Visualization	11
3	Vector Field Background	13
3.1	Concepts of Vector Fields	13
3.2	Local Characteristics	14
4	A Lagrangian Accumulation Framework	16
4.1	Introduction to Lagrangian Accumulation and Attribute Fields	17
4.2	Properties of L-Accumulation and the \mathcal{A} Fields	19
4.2.1	Existence and Uniqueness of $\mathcal{A}(\mathbf{x}, t)$	19
4.2.2	Discontinuity in the \mathcal{A} Fields	19
4.2.3	Inequality Property	23
4.3	Computation and Visualization of \mathcal{A} fields	25
4.4	A Case Study: Φ Field	28

4.4.1	Computation	30
4.4.2	Visualization	31
5	Applications of Lagrangian Accumulation Framework	35
5.1	\mathcal{A} Field Based Flow Exploration	35
5.1.1	Integral Curve Filtering	36
5.1.2	Integral Curve Seeding	36
5.1.3	An \mathcal{A} Field Guided Ribbon Placement	37
5.1.4	An \mathcal{A} Field Guided Surface Seeding	39
5.1.5	Visualizing Particle-based Data Aided by \mathcal{A}	40
5.2	Discontinuity Study	42
5.2.1	Background of Discontinuity	42
5.2.2	Discontinuity Extraction	46
5.2.3	Combined Attribute Fields	47
5.2.4	Results and Applications	49
5.2.5	Conclusion	50
5.3	\mathcal{A} Field Based Flow Segmentation	52
5.3.1	Motivation	52
5.3.2	Segmentation Algorithm	55
5.3.3	Results	61
5.3.4	Conclusion	65
5.4	Discussion	74
5.4.1	An Informal Study of Relations Among Attributes	74
5.4.2	Relation of \mathcal{A} and $ \nabla\mathcal{A} $ Fields to Flow Features	80
5.4.3	Which Characteristics to Accumulate?	81
5.4.4	Accumulation Window Size	82
5.4.5	Average of the Accumulated Value	84

5.4.6	Comparison with the Eulerian Accumulation	85
5.4.7	Extension to Non-integral Curves – Streak Lines	86
6	A TAC-based Framework	89
6.1	Time Activity Curve (TAC) Generation	91
6.1.1	Definition of TAC	91
6.1.2	TAC Analysis	94
6.1.3	TAC-based Event Definition	97
6.2	Methodology	98
6.2.1	TAC-based Dissimilarity Metrics	99
6.2.2	TAC-based Clustering Algorithm	104
6.2.3	A Hierarchical Clustering Algorithm in Time Dimension	113
6.3	Results	124
6.3.1	Hierarchical Clustering of Eulerian TACs	124
6.3.2	Hierarchical Clustering of Lagrangian TACs	127
6.3.3	Temporal Hierarchical Clustering	128
7	Conclusion	132
8	Appendix	135
	Bibliography	140

List of Figures

1.1	The illustration of the gap between geometrics representation and physical interpretation. (a) A number of pathlines with homogeneous behaviors. (b) The time activity curves of the physical attribute Q . (c) Accumulated Q field values of each corresponding pathlines in (a).	3
4.1	A number of \mathcal{A} fields computed based on the accumulation of λ_2 for a 2D synthetic steady flow (a), arc-length of the integral curves for an unsteady double gyre flow (b), and the absolute winding angle of the integral curves for a 3D steady flow behind cylinder (c), respectively. A blue-white-red color coding is used for (a) and (c), while a rainbow color coding is used for (b).	18
4.2	Discontinuity of the \mathcal{A} field at a separatrix connecting a saddle (blue dot) and a source (green dot). The \mathcal{A} field is sampled along the line segment traversing through the separatrix. \mathbf{c}_i indicate the samples along this segment. Case (a) shows a scenario of the discontinuity that the discrete sampling may miss (illustrated by the orange curve), while the discontinuity in case (b) could be captured with sufficient samples.	21
4.3	(a) highlights the places (in red) of an \mathcal{A} field where Eq.(4.3) does not hold. (b) highlights the places (in red) where Eq.(4.4) does not hold. As can be seen, all places satisfy this new inequality condition. (c) shows the $ \nabla\mathcal{A} $ using a rainbow color coding. As can be seen, most places have very small gradient values (blue), indicating slow change of \mathcal{A} field there; while places near the separatrices that correspond to certain discontinuity in the \mathcal{A} have large gradient (red). The horizontal direction is X and the vertical direction is Y.	24
4.4	\mathcal{A} fields of a synthetic surface flow (a), a cooling jacket simulation (b) and a gas engine simulation (c), respectively.	27

4.5	\mathcal{A} (top) and $ \nabla\mathcal{A} $ (bottom) fields of a tile of the ocean simulation with different window sizes for accumulation. (a)10% of the size of the bounding box of the data domain; (b) 50%; (c) 2,000%. (d) shows the \mathcal{A} field by accumulating the divergence along streamlines.	28
4.6	An example of Φ field (b) and its gradient magnitude field $ \nabla\Phi $ (c). (a) shows the LIC of the flow.	29
4.7	The result of the double gyre flow with $T = 10$. (a-b) show the forward Φ field, the comparison between LCS and $ \nabla\Phi $ ridges at time steps 0 and 20, respectively. For FTLE, a rainbow color coding is used where red indicates larger FTLE values and blue smaller. For the comparison of LCS and $ \nabla\Phi $ ridges (b), the extracted FTLE ridge points are highlighted in red on top of the $ \nabla\Phi $ visualization that utilizes a gray scale color map. In addition to the LCS structure, the Φ and $ \nabla\Phi $ fields also capture the cusp-like seeding curves indicated by the yellow arrows. (c) shows the Φ and $ \nabla\Phi $ field based on streaklines at $t = 0$, which captures the singularity paths of the two vortices over time as highlighted by the black arrows. (d) shows the pathlines seeded at the cusp-like ridges of the $ \nabla\Phi $ field. Note that the intersections of these pathlines with the (horizontal) time planes form some coherent structures rotating over time. (e) shows a set of selected pathlines by filtering the Φ field.	32
4.8	The visualization of the Φ (a) and $ \nabla\Phi $ (b) fields of the cylinder flow data [65]. The time direction increases away from the view. The 2D slices correspond to time step $t_0 = 200$ are shown. (c) and (d) show the comparison of LCS and $ \nabla\Phi $ field computed based on forward pathlines (c) and backward pathlines (d) at $t_0 = 0$, respectively. The LCS is highlighted in red on top of the $ \nabla\Phi $ visualization. For each 2D slice horizontal direction is X and the vertical direction is Y.	34
5.1	The volume rendering of the Φ field (a) and $ \nabla\Phi $ field (b) of the Bernard flow. (c) shows several filtered streamlines based on the Φ field. (d) shows the local attribute and Q	37
5.2	Ribbon placement results for the Bernard data (a) and the tornado data (b), respectively. The left image of each group shows the ribbon placement guided by local helicity information, while the right image shows the placement guided by the derived \mathcal{A} field based on helicity.	38

5.3	Comparison of two strategies of seeding curve generation. The red surface is constructed from a seeding curve derived using the small variation strategy, while the yellow is from a seeding curve derived using the large variation strategy. The seed of seeding curve for the blue surface is located inside the bundle, where its $ \nabla \mathcal{A} $ value is small, i.e. the \mathcal{A} values along this seeding curve are almost constant. In contrast, the seed position of the seeding curve for the red surface is located near the boundary of the domain, where the $ \nabla \mathcal{A} $ value is large, and the variation of the \mathcal{A} values on this seeding curve is also large.	40
5.4	Visualization of an \mathcal{A} field derived from a dam-breaking particle based simulation. Blue means the change of particle moving direction is small, while red mean large. It shows that the particles that hit the boundary have larger change of moving direction, as highlighted by the arrows.	41
5.5	The illustration of the relation between the attribute field and a number of well-known flow features, including the flow separation (a) and vortices (b). The left column shows the vector fields illustrated by streamlines, the middle column shows the rotation field.	43
5.6	The plots of the rotation field of the streamlines intersecting with given seeding line segments (shown in Figure 5.5). Note that the discontinuities (sharp gradients) in the rotation field indicate the flow features. The values shown in the plot have no units as they do not correspond to any physical quantities.	44
5.7	Discontinuity detection on a Φ field derived from a synthetic steady flow using the Canny edge detector with different combinations of parameters. (a) The differential topology with LIC as the background; (b) Φ field; (c-e) Detected edges with different parameters of the Canny edge detector: (c) - $\sigma = 3.0 \alpha = 0.3 \beta = 0.8$, (d) - $\sigma = 3.0 \alpha = 0.6 \beta = 0.8$, (e) - $\sigma = 3.0 \alpha = 0.3 \beta = 0.86$; (f) The gradient of Φ field. The horizontal direction is the X direction and the vertical direction is the Y direction.	46
5.8	Illustration of a number of attribute fields derived from the Double Gyre flow and their detected edges. (a)–(d) show the attribute fields Φ , L , $avgDir$ and $acceleration$, $avgDir$ computed from pathlines, respectively. (e) is the rotation field Φ from streaklines. (f) is the rotation field Φ from pathlines using backward integration. The parameters of Canny edge detector are $\sigma = 2.0$, $\alpha = 0.3$, $\beta = 0.8$. The horizontal direction is the X direction and the vertical direction is the Y direction.	47

5.9	(a) Overlap of edges in Φ (yellow) and \mathbb{L} (purple) fields; (b) Simple combination of Φ and \mathbb{L} fields. The horizontal direction is the X direction and the vertical direction is the Y direction.	48
5.10	Illustration of weighted combination of Φ and <i>avgDir</i> field. (a) Combined attribute field; (b) Edges detected from the super field. The dark gray curves on top are stable edges that do not change with weights. The weights of Φ and <i>avgDir</i> from left to right are $\alpha_1 = 0.1, \alpha_2 = 0.9; \alpha_1 = 0.5, \alpha_2 = 0.5; \alpha_1 = 0.9, \alpha_2 = 0.1$, respectively. The parameters of Canny edge detector are $\sigma = 1.0, \alpha = 0.3, \beta = 0.8$. The horizontal direction is the X direction and the vertical direction is the Y direction.	49
5.11	Results of the forced-damped Duffing system. (a)–(b) Φ and <i>avgDir</i> fields and their detected edges. The parameters of Canny edge detector are $\sigma = 1.8, \alpha = 0.3, \beta = 0.9$. (c) A super field using the equally weighted combination of $\Phi, avgDir, L, nsV$ and <i>seDist</i> . The horizontal direction is the X direction and the vertical direction is the Y direction.	49
5.12	Attribute fields of the flow behind cylinder and detected edges. (a) <i>avgV</i> field; (b) <i>nsV</i> field; (c) \mathbb{L} field; (d) <i>FTLE</i> and <i>LCS</i> . The parameters of Canny edge detector are $\sigma = 1.0, \alpha = 0.3, \beta = 0.8$. The horizontal direction is the X direction and the vertical direction is the Y direction.	51
5.13	The pipeline of the proposed method. Note that it only illustrate the procedural of the algorithm and does not provide the results, therefore, there is no need to show axes or units.	53
5.14	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.	55
5.15	The pipeline of the segmentation algorithm.	55
5.16	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 $m = 5$	56
5.17	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.	59

5.18	Comparison between the bottom-up algorithm and the proposed 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 the proposed algorithm based on the rotation field. (d) The boundaries of the segments in (c).	67
5.19	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.	68
5.20	Segmentation of a 2D unsteady flow behind a square cylinder based on different attribute fields: <i>determinant</i> field (a), <i>nsV</i> field (b) and <i>curl</i> field (c), respectively. The bin number $m = 6$	69
5.21	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 $m = 6$	70
5.22	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.	71
5.23	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, where the X axis shows the range of the attribute field in each bin, and the Y axis illustrates the number of samples in each bin.	72
5.24	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 $m = 2$. (d) The legend shows the partitioning strategy for initial region classification, where the X axis shows the range of the attribute field in each bin, and the Y axis illustrates the number of samples in each bin.	73

5.25	The scatter plot matrix of different \mathcal{A} fields of the Double Gyre flow. Note that the scatter plots associated with FTLE shows the correlation among the magnitude of the gradient of the individual attribute fields with the FTLE field. Note that tis figure is a matrix not a real plot, therefore there are no axes or unites. Due to the limitation of page size, this figure is split into four parts and put in the Appendix (Figure 8.1, Figure 8.2, Figure 8.3, and Figure 8.4).	75
5.26	Comparison of the \mathcal{A} fields computed by accumulating the curl (bottom) and the change of the flow direction (top), i.e. winding angle, respectively. (a) shows the \mathcal{A} fields of a synthetic 2D steady flow. Their corresponding edges are in (b). (c) shows the \mathcal{A} fields of a 2D force duffing system. . .	78
5.27	Comparison of the FTLE fields (top) and a derived fields (bottom) from the \mathcal{A} fields–vector fields defined by V_{SE} for the double gyre flow (a) and the force duffing system (b).	79
5.28	The influence of the accumulation window size. (a) shows the \mathcal{A} field computed with the integration length equal 10% of the size of the bounding box of the flow domain, while (b) shows the \mathcal{A} field with the length equal twice of the size of the bounding box. (c) shows the plots of the \mathcal{A} values sampled along two seeding line segments. As can be seen, even they have the same length, the two segments encode different amount of information quantified by the range of the \mathcal{A} values along the segments. This also demonstrates how the varying density of the integral curve may influence the encoding of certain flow features in the \mathcal{A} field.	83
5.29	The \mathcal{A} fields computed without (left) and with average (right).	84
5.30	Comparison of Eulerian (a) and Lagrangian (b) accumulations using various attributes of the flow behind cylinder data. Note that the Eulerian accumulation highlights the places where the vortices sweep through, while the Lagrangian accumulation emphasizes the oscillating behaviors of the individual vortices.	85
5.31	The $ \nabla\Phi $ fields based on streaklines for a number of synthetic unsteady flows.	87
6.1	(a) An example TAC. (b) Event of the TAC in (a). (c) An example of time interval segments of TACs.	93

6.2	(a) Eulerian TAC samples of attribute <i>curl</i> . (b) Lagrangian TAC samples of attribute Q . X axis is the time and Y axis stands for the attribute values. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them.	96
6.3	Different types of primitive trends.	97
6.4	Two examples of TACs that cannot be accurately measured by Euclidean distance, Pearson correlation or DTW. (a) $D_e(\Gamma_{base}, \Gamma_1) = D_e(\Gamma_{base}, \Gamma_2)$, $DTW(\Gamma_{base}, \Gamma_1) = DTW(\Gamma_{base}, \Gamma_2)$, Γ_2 has different trend from Γ_{base} . (b) $D_p(\Gamma_{base}, \Gamma_1) = D_p(\Gamma_{base}, \Gamma_2)$, $DTW(\Gamma_{base}, \Gamma_1) = DTW(\Gamma_{base}, \Gamma_2)$, Γ_2 is farther away from Γ_{base} w.r.t temporal difference.	102
6.5	Illustration of clustering result generation. (a) distribution of distance values at each level of the tree. (b)-(d) clustering results with $K = 6, K = 5, K = 4$, respectively. The corresponding distance thresholds are $\varepsilon = 26\%$, $\varepsilon = 30\%$ and $\varepsilon = 43\%$, respectively. X axis is the time and Y axis stands for the attribute value. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them. . .	108
6.6	An example of TAC-based clustering. (a) direct visualization of the two clusters. (b) edge-bundling visualization of the two clusters. X axis is the time and Y axis stands for the attribute value. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them.	110
6.7	Illustration of edge-bundling method for TACs cluster visualization. TACs are divided into three parts: head, body, and tail to construct the edge-bundling. X axis is the time and Y axis stands for the attribute value. . . .	113
6.8	Motivation of temporal clustering. (a) two clusters (in red and blue, respectively) of TACs show different global behaviors in the entire time period. (b) Similar behaviors exhibit for the two groups of TACs in a local time interval (highlighted in cyan). X axis is the time and Y axis stands for the attribute value.	114
6.9	An example of temporal clustering. (a) initial hierarchical clustering result of $M = 4$ leaf nodes. (b) clustering result after merging the two most similar clusters in (a). X axis is the time and Y axis stands for the attribute value. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them.	116

- 6.10 Visualization of changes through time intervals. (a) result before cluster ID adjustment. (b) result after cluster ID adjustment. (c) edge-bundling visualization of the result. (d) modified edge-bundling visualization. T_1 and T_2 represent the two time intervals. X axis is the time and Y axis stands for the attribute value. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them. 123
- 6.11 Different views in the system. (a) shows the visualization of clustering result. (b) shows the representative TAC in each cluster in (a) and a hierarchical tree of the clustering result. (c) shows the visualization of temporal hierarchical clustering result. (d) shows the clustering on flow domain based on the result in (a). X axis is the time and Y axis stands for the attribute value of curl. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them. For the texture-based visualization of the flow, the horizontal direction is X and the vertical direction is Y. 125
- 6.12 Clustering result of the Double Gyre flow based on Eulerian TACs of attribute λ_2 . (a) Clustering result based on Euclidean distance metric. (b) and (c) Clustering results based on *EDTW* distance metric with the cluster number $K = 2$ and $K = 3$, respectively. The columns from left to right show the visualization of TAC clusters, representative TAC in each cluster and clustering of flow domain, respectively. X axis is the time and Y axis stands for the attribute value of λ_2 . Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them. For the texture-based visualization of the flow, the horizontal direction is X and the vertical direction is Y. 126
- 6.13 Clustering result of Cylinder flow based on Eulerian TACs of attribute *curl*. (a), (b) and (c) Clustering results based on *EDTW* distance metric with the cluster number $K = 3$, $K = 4$, and $K = 5$, respectively. The columns from left to right show the visualization of TAC clusters, representative TAC in each cluster and clustering of flow domain, respectively. X axis is the time and Y axis stands for the attribute value of *curl*. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them. For the texture-based visualization of the flow, the horizontal direction is X and the vertical direction is Y. 127

6.14	Clustering result of the Double Gyre flow based on Lagrangian TACs of attribute <i>curl</i> with different cluster number K . (a) TAC clusters and its hierarchical tree. (b) representative TACs in each cluster. (c) Clustering of flow domain. X axis is the time and Y axis stands for the attribute value of <i>curl</i> . Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them. For the texture-based visualization of the flow, the horizontal direction is X and the vertical direction is Y.	129
6.15	Clustering result of Cylinder flow based on Lagrangian TACs of attribute <i>curl</i> with different cluster number K . (a) TAC clusters and its hierarchical tree. (b) Clustering of flow domain. X axis is the time and Y axis stands for the attribute value of <i>curl</i> . Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them. For the texture-based visualization of the flow, the horizontal direction is X and the vertical direction is Y.	130
6.16	Clustering result of Double Gyre flow based on Lagrangian TACs of attribute <i>momentum</i> with different number of time intervals M . (a) $M = 2$. (b) $M = 3$. (c) $M = 4$. X axis is the time and Y axis stands for the attribute value. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them.	131
8.1	The scatter plot matrix of different \mathcal{A} fields of the Double Gyre flow. Part 1 of Figure 5.25	136
8.2	The scatter plot matrix of different \mathcal{A} fields of the Double Gyre flow. Part 2 of Figure 5.25	137
8.3	The scatter plot matrix of different \mathcal{A} fields of the Double Gyre flow. Part 3 of Figure 5.25	138
8.4	The scatter plot matrix of different \mathcal{A} fields of the Double Gyre flow. Part 4 of Figure 5.25	139

Chapter 1

Introduction

Vector field visualization is a ubiquitous technique that is employed to study a wide range of dynamical systems involved in applications, such as automobile and aircraft engineering, climate study, combustion dynamics, earthquake engineering, and medicine, among others. Many effective approaches have been developed to visualize these complex data [10, 28, 43, 51]. With the continuous increase in size and complexity of the generated flow data sets, there is a strong need to develop an effective abstract (or reduced) representation that addresses the complexity of data interpretation and user interaction.

There is a large body of work on generating a reduced representation of the flow based on its geometric behaviors. One classic representation is vector field topology, which aims to extract special streamlines related to fixed points and periodic orbits. By extracting these special streamlines and their connectivity, the flow fields can be segmented into different regions with homogeneous behaviors. Such segmentation reduces drastically the information to be displayed, conveying a holistic understanding of the flow in a more semantic

level. Recently, streamline and pathline attribute-based flow exploration [16, 42] was introduced. Other work generates a reduced representation of the flow by classifying integral curves into different clusters based on their individual attributes. The representative curves for each cluster are computed to provide an abstract visualization of the flow [70].

Most existing vector field visualization techniques focus on the representation of the geometric characteristics of the vector fields, which is understandable as geometric features (e.g., some flow patterns) are intuitive to understand. However, there are two limitations of these methods. (1) Important physical relevant features could be missed. For example, vector field topology only encodes hyperbolic features in the flow, other physically relevant information, such as vortices, and flow separation, is not included. On the other hand, integral curve clustering methods place seeds based on spatial importance (i.e. to ensure sufficient spatial coverage and to reduce cluttering) rather than physical importance. Therefore, they cannot guarantee that physically important features will be revealed. (2) Geometric representation may not intuitively reveal the physical behaviors of the flow. Domain experts may still need additional tools or mental translation to interpret the physical characteristics from the geometric representation of the flow. Figure 1.1 provides an example where the physically interesting behaviors may not be revealed in the traditional geometric representation of the flow. The pathlines shown in Figure 1.1(a) have geometrically similar behaviors, while their time activity curves of a local physical property Q -criterion [22], shown in Figure 1.1(b), show otherwise. In particular, the black pathline (in the middle of the pathline bundle shown in (a)) has rather different behaviors in the attribute space from the others (i.e. lower valley values).

This dissertation aims to fill the gap between the geometric representation of the flow

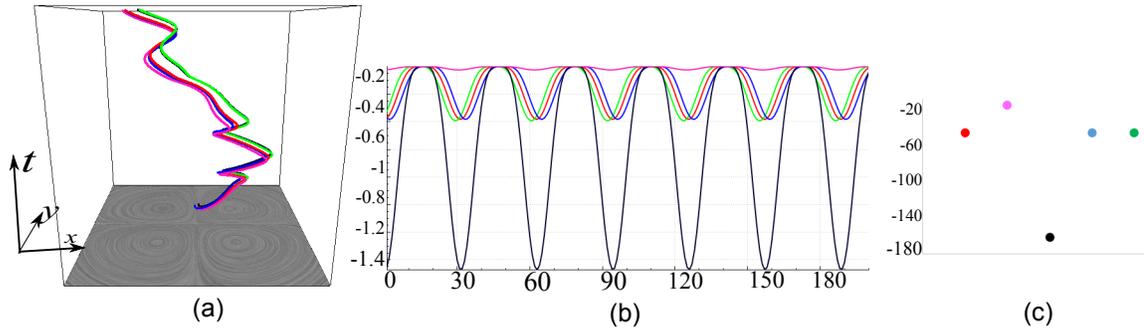


Figure 1.1: The illustration of the gap between geometrics representation and physical interpretation. (a) A number of pathlines with homogeneous behaviors. (b) The time activity curves of the physical attribute Q . (c) Accumulated Q field values of each corresponding pathlines in (a).

and the physical interpretation of the flow behaviors that are of interest to the experts. However, the local physical characteristics of the flow are typically not intuitive to understand and describe only the local and short-term behaviors of the flow. To address that, flow data visualization and exploration need to be performed, which are guided by the physical information associated with either the individual particles or spatial samples over a given time window. To address that, the following two new frameworks are introduced in this work.

The first framework is called the *Lagrangian accumulation* framework, in which the local physical or geometric attributes of the individual particles are accumulated along the integral curves associated with the particles. The accumulated values are assigned to the starting positions of those particles, which are typically evenly distributed in the beginning. This results in a number of attribute fields, each of which corresponds to a specific attribute. These fields are referred to as the *Lagrangian accumulation fields*, or the \mathcal{A} fields. In this way, the overall, long-term behaviors of the particles are encoded

in the spatio-temporal locations that they pass through in an Eulerian fashion. With this representation, the spatial points that are located on integral curves that have similar behaviors will have similar attribute fields values. Note that each integral curve indicates the trajectory of a flow particle seeded at a sampled spatial position. In that sense, the computation of the derived attribute fields summarizes the behaviors of the particles along their individual trajectories, which provides us a global, abstract representation of their individual trajectories. Figure 1.1(c) shows the values of the Q field, an example of \mathcal{A} fields, of the pathlines shown in Figure 1.1(a). Though the geometric behaviors of the pathlines are similar, their overall Q -attribute behaviors are quite different as shown in Figure 1.1(c). The Lagrangian accumulation framework has been applied to assist a number of flow visualization tasks, including flow segmentation, integral curve seeding and discontinuity discovery. The result included here demonstrates that these tasks can be conducted more efficiently with the aid of the obtained \mathcal{A} fields.

However, \mathcal{A} fields have some significant limitations. (1) They cannot provide the details of flow behaviors, especially the occurring time and period of an interesting feature. (2) They cannot study the contributions of the individual local events to the global behavior of a particle, i.e, the reason that causes the specific behavior.

Therefore, the second framework, i.e. a new flow exploration framework based on the Time Activity Curves (TAC) of the local attributes, is introduced to achieve a more detailed exploration. Specifically, a number of time-series data of the local physical properties are derived along each integral curve, which provides us a series of TACs of different local attributes, based on which an event detector of interesting flow behaviors are introduced. With the concepts of TAC and TAC-based event, a new dissimilarity metric is defined to

characterize the similar attribute behaviors of different particles. In order to provide different levels of details of the flow behaviors, a TAC-based hierarchical clustering algorithm is presented. A comprehensive distance metric $EDTW$ is derived to describe the dissimilarity of TACs based on the detected events, which incorporates the global correlation of a pair-wise TACs, DTW distance and temporal difference of events. This TAC-based hierarchical clustering algorithm adopts a modified BIRCH-based clustering to resolve the scalability issue and accelerate the computation. A similar hierarchical clustering is then applied to the time dimension to reveal the changes of flow behaviors through different time intervals. A statistic-based segmentation is proposed to generate a number of initial time intervals, from which a hierarchical tree is built. This hierarchical tree and the corresponding clusters can be utilized to aid the user exploration of the flow behaviors.

In summary, this dissertation has made the following contributions.

- The Lagrangian accumulation along integral curves is revisited and extended to the study of flow behaviors using any well-defined local flow characteristics. In addition, a systematic study and discussion on the properties of the resulting \mathcal{A} fields is provided. This work demonstrates how to use the properties of the \mathcal{A} fields to achieve a number of interesting visualizations for 2D and 3D flows. It also shows that the \mathcal{A} field and the information encoded in it can be utilized to enhance a number of flow visualization tasks, including seeding curve generation for integral curve and surface placement and flow domain segmentation.
- The Lagrangian accumulation frame work is extended based on TACs to obtain a more detailed exploration. A new hierarchical clustering algorithm is proposed and a

new and comprehensive TAC-based framework is introduced to assist flow exploration. In addition, the framework is flexible and efficient in that various attributes can be used to help users investigate different flow behaviors.

Without confusion, for all the 2D visualization of the data, the horizontal direction is the X direction and the vertical direction is the Y direction. For 3D visualization, we provide the reference frames to indicate the orientation of the viewpoint. Illustration and pipeline figures do not have axes and units, because they describe the process of the proposed methods rather than showing the actual results.

Chapter 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 for the dense and texture-based visualization techniques [29], geometric-based methods [37], illustrative visualization [1], topology-based methods [28, 43], and partition-based techniques [51] for various flow data. This section focuses on the most relevant work.

2.1 Vector Field Topological Analysis

Vector field topology provides a streamline classification strategy based on the origin and destination of the individual streamlines. Since its introduction to the visualization community [21], vector field topology has received extensive attention. A large body of work has been introduced to identify different topological features, including fixed

points [44, 59] and periodic orbits [5, 58, 68]. Recently, Chen et al. (2008) [6] 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 vector fields. Szymczak et al. (2012) [57] introduced a new approach to converting the input vector field to a piecewise constant (PC) vector field and computing the Morse decomposition on triangulated manifold surface.

For the topological analysis of unsteady flows, *Lagrangian Coherent structures (LCS)*, i.e. curves (2D) or surfaces (3D) in the domain across which the flux is negligible, were introduced to identify separation structures in unsteady flows. The computation of LCSs was first introduced by Haller (2001) [19] by computing the *Finite Time Lyapunov Exponent (FTLE)* of the flow, whose ridges indicate the LCS. Since its introduction, FTLE has been compared with the separatrices in the steady cases [47], and its computational performance has been improved substantially [14]. Recently, Fuchs et al.(2010) [13] presented an extended critical point concept applying vector field topology in unsteady flow. Sadlo and Weiskopf (2010) introduced a streakline-based topology based on the concept of generalized streaklines [48]. It successfully characterizes the saddle type of hyperbolic features. This was extended to study the 3D unsteady flow topology [60].

2.2 Streamline and Pathline Attributes

Sadarjoen and Post (1999) introduced the winding angle concept for streamlines and utilized it to classify the streamlines within vortical regions [46]. Salzbrunn and Scheuermann (2006) introduced *streamline predicates*, which classify streamlines by interrogating them as they pass through certain user-specified features, e.g., vortices [50]. Later, this approach was extended to the classification of pathlines [49]. At the same time, Shi et al. (2007) [54] presented a data exploration system to study the characteristics of pathlines based on 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 [42]. This work differs from pathline predicate and pathline attribute approaches in that it utilizes the attributes of the individual pathlines to construct a smooth scalar field to classify the spatial locations where the pathlines are seeded. This enables us to study the flow structure as well as to classify the integral curves via the behavior of the derived scalar field. More recently, McLoughlin et al.(2013) [36] introduced the idea of a streamline signature based on a set of curve-based attributes including curvature and torsion. This streamline signature is used as a measure of the similarity between streamlines, and helps domain experts place and filter streamlines for the creation of an informative and uncluttered visualization of 3D flow.

2.3 Lagrangian Framework for Flow Analysis

In fluid dynamics, there are two different views for the study of flow behaviors, i.e. observing the flow at fixed location—Eulerian point of view, or observing it on a moving particle—Lagrangian point of view. This work specifically focuses on the Lagrangian framework, which studies the behavior of particles along their individual paths, i.e. integral curves computed from the initial positions of the particles. Based on this characteristic, the Finite-Time Lyapunov Exponent (FTLE) [19], the streamline [50] and pathline [49] predicates, the pathline attribute approaches [18, 42, 54], and the streamline and pathline dissimilarity for streamline clustering [70], selection [36], and ensemble analysis [17] are all examples of Lagrangian approaches. Among them, the FTLE approach aims to measure the rate of separation at individual spatial sampling points. Its flow map computation is essentially a special case of Lagrangian accumulation that sums up all the vector values scaled by the integration step size along the path of the particle, which leads to the end position of a particle given its starting position. This accumulation neglects all intermediate position as well as other information of the particle that is not relevant to the particle separation. The computed rate of separation at each point is encoded as a scalar field, which facilitates the identification of its ridges, known as the Lagrangian Coherent Structure (LCS). This Eulerian representation of the FTLE fields is similar to the derived \mathcal{A} fields. Nonetheless, the Lagrangian accumulation and the resulting \mathcal{A} fields are more general than the FTLE approach, and can be used to encode attributes of the particles along their paths rather than just at their starting and ending positions.

The idea of accumulating local characteristics along the paths of particles and assigning the accumulated values to the corresponding integral curves has been employed in the pathline attribute approaches. Specifically, Shi et al. (2007) [54] presented a data exploration system to study the different characteristics of pathlines based on their various attributes. Pobitzer et al. (2012) [42] applied a statistics-based method to select a proper subset of pathline attributes to improve the interactive flow analysis. While not directly accumulating the local attributes, Gao et al. (2014) [18] proposed to accumulate the square difference between the local attributes along pairs of integral curves to define the distance between them. Differentiating from the pathline attribute approach, \mathcal{A} fields adapt the Eulerian representation of the texture-based visualization and the FTLE approach, and store the accumulated attribute values to the seeding positions of the individual integral curves, whose properties offer a number of opportunities to support various flow data exploration tasks.

More recently, Lagrangian representation has been employed to address the scalability of the visualization of large scale unsteady flows [18, 4].

2.4 Time-varying Data Visualization

In scientific simulations, features of interest are commonly represented as Time Activity Curves (TAC). TAC has been studied in scientific visualization in recent years [62, 69]. Lee et al. (2009) proposed a visualization framework to analyze time-varying data sets with a TAC-based distance field [30]. This field provides a visualization to highlight the position the features, while it still cannot provide the details of an individual TAC, especially the

time of occurrence and period of an interesting feature. Wei et al. (2011) introduce a dual-space method to analyze such data, starting by clustering the time series curves in the phase space of the data, and then visualizing the corresponding trajectories of each cluster in the physical space [64]. Ferstl et. al. (2017) proposed a time-hierarchical clustering approach for analyzing the temporal growth of the uncertainty in ensembles of weather forecasts [12]. For a thorough overview of approaches for the time-varying data, please refer to the surveys [32, 11].

Chapter 3

Vector Field Background

In this chapter, some of the most important concepts in vector fields are reviewed, which will be used for the later discussion of the proposed frameworks.

3.1 Concepts of Vector Fields

Consider a spatio-temporal domain $\mathbb{D} = \mathbb{M} \times \mathbb{T}$ where $\mathbb{M} \subset \mathbb{R}^d$ is a d -manifold ($d = 2, 3$ in this work) and $\mathbb{T} \subset \mathbb{R}$, a general vector field can be expressed as an ordinary differential equation (ODE) $\dot{\mathbf{x}} = V(\mathbf{x}, t)$.

In the *steady* (i.e. *time-independent*) case, $\dot{\mathbf{x}} = V(\mathbf{x})$, whose solution gives rise to a *flow* which is a continuous function (or map) $\varphi : \mathbb{R} \times \mathbb{M} \rightarrow \mathbb{M}$ satisfying $\varphi(0, \mathbf{x}) = \mathbf{x}$, for all $\mathbf{x} \in \mathbb{M}$, and $\varphi(t, \varphi(s, \mathbf{x})) = \varphi(t + s, \mathbf{x})$ for all $\mathbf{x} \in \mathbb{M}$ and $t, s \in \mathbb{R}$. The *trajectory* of $\mathbf{x}_0 \in \mathbb{M}$ is a *streamline*, $\mathbf{x}_{\mathbf{x}_0}(t) = \mathbf{x}_0 + \int_0^t V(\mathbf{x}_{\mathbf{x}_0}(\tau))d\tau$. A point $\mathbf{x}_0 \in \mathbb{M}$ is a *fixed point* if

$\varphi(t, \mathbf{x}_0) = \mathbf{x}_0$ for all $t \in \mathbb{R}$, that is, $V(\mathbf{x}_0) = 0$. A trajectory is called a *periodic orbit* (or closed), if given any point, \mathbf{x} , on it, it satisfies $\varphi(T, \mathbf{x}) = \mathbf{x}$, where $T \in \mathbb{R}$ and $T \neq 0$. All the hyperbolic fixed points and periodic orbits and their connectivity defines the *topology* of the steady vector field [5].

For an *unsteady* (or *time-dependent*) vector field, $V(\mathbf{x}, t)$, the trajectory of a particle starting at \mathbf{x}_0 and at time t_0 is called a *pathline*, denoted by $\mathbf{x}_{\mathbf{x}_0, t_0}(t) = \mathbf{x}_0 + \int_0^t V(\mathbf{x}_{\mathbf{x}_0, t_0}(\tau), t_0 + \tau) d\tau$. The collection of the particles that are released from the same spatial position but at consecutive times forms a *streakline*. In contrast, the collection of particles that are released from different spatial locations but at the same time forms a *timeline*. In this work, we focus only on the integral curves, i.e. streamlines and pathlines, derived from a given vector field.

3.2 Local Characteristics

There are a number of local characteristics that are of interest to the domain experts. Specifically, the *acceleration* of \mathbf{v} is defined as $\mathbf{a}(\mathbf{x}, t) = \frac{D\mathbf{v}}{Dt} = \frac{\partial \mathbf{v}(\mathbf{x}, t)}{\partial t} + (\mathbf{v}(\mathbf{x}, t) \cdot \nabla) \mathbf{v}(\mathbf{x}, t)$, where $(\mathbf{v}(\mathbf{x}, t) \cdot \nabla) = \nabla_{\mathbf{x}} \mathbf{v}$ represents the spatial gradient of \mathbf{v} , i.e. *Jacobian*, denoted by \mathbf{J} for simplicity. Important deformation modes of the flow can be obtained through the decomposition of \mathbf{J} . Specifically, the Jacobian matrix can be decomposed as $\mathbf{J} = \mathbf{S} + \mathbf{R}$, where $\mathbf{S} = \frac{1}{2}[\mathbf{J} + (\mathbf{J})^\top]$ and $\mathbf{R} = \frac{1}{2}[\mathbf{J} - (\mathbf{J})^\top]$ are the symmetric and antisymmetric components of \mathbf{J} , respectively. From this decomposition, the local shear rate is measured as the Frobenius norm of \mathbf{S} [9], and the Q and λ_2 values at each point can be computed as $Q = \frac{1}{2}(\|\mathbf{R}\|^2 - \|\mathbf{S}\|^2)$ [22], while λ_2 is the second largest eigenvalue for the tensor

$\mathbf{S}^2 + \mathbf{R}^2$ [24]. They are typically used to characterize the vortical or rotational behaviors of the flow.

In this work, the following local attributes, a_l , are utilized for various experiments.

- a_1 : vorticity, $\|\nabla \times \mathbf{v}\|$.
- a_2 : divergence, $tr(\mathbf{J})$, i.e. trace of \mathbf{J} .
- a_3 : helicity, $\nabla \times \mathbf{v} \cdot \mathbf{v}$.
- a_4 : λ_2 , the second largest eigenvalue of the tensor $\mathbf{S}^2 + \mathbf{R}^2$ [24].
- a_5 : $Q = \frac{1}{2}(\|\mathbf{R}\|^2 - \|\mathbf{S}\|^2)$ [22].
- a_6 : local shear rate, the Frobenius norm of \mathbf{S} .
- a_7 : determinant of \mathbf{J} .
- a_8 : change of flow direction (also known as winding angle), $\angle(\mathbf{v}(\mathbf{p}_i), \mathbf{v}(\mathbf{p}_{i+1}))$ where \mathbf{p}_i denotes a point on an integral curve. This geometric attribute essentially measures the curvature of the integral curve at \mathbf{p}_i .
- a_9 : velocity vector \mathbf{v} .
- a_{10} , acceleration, $\mathbf{a}(\mathbf{x}, t) = \frac{D\mathbf{v}}{Dt} = \frac{\partial \mathbf{v}(\mathbf{x}, t)}{\partial t} + (\mathbf{v}(\mathbf{x}, t) \cdot \nabla)\mathbf{v}(\mathbf{x}, t)$.

Chapter 4

A Lagrangian Accumulation Framework

In this chapter, the first framework proposed by this dissertation, a Lagrangian accumulation framework, is discussed. The attribute fields derived from Lagrangian accumulation are first introduced, followed by the discussion of their properties and a case study of the Lagrangian accumulation framework.

4.1 Introduction to Lagrangian Accumulation and Attribute Fields

Consider an integral curve, \mathcal{C} , starting from a given point (\mathbf{x}, t_0) , the Lagrangian accumulation can be formulated as the following convolution process:

$$A_g((\mathbf{x}, t_0), t) = \int_{t_0}^t k(\tau) a_l(\mathcal{C}(\tau), t_0 + \tau) d\tau \quad (4.1)$$

where $k(\tau)$ is a filter kernel following the integral curves [2, 61]. For simplicity, this work assumes a simple *box filter* [53], for all examples. $a_l(\mathcal{C}(\tau), t_0 + \tau)$ is the value of the selected local flow property, a_l , measured at location $\mathcal{C}(\tau)$ and at time $t_0 + \tau$, which can be either scalar, vector, or tensor values. For the later discussion, we mainly consider scalar properties. In most cases, a_l is continuous in \mathbb{D} except at some special locations, such as fixed points in the steady cases. $A_g((\mathbf{x}, t_0), t)$ represents the accumulated value. $t \in \mathbb{R}$ is the integration window size. Note that t can be negative to account for backward integration. In addition, both forward and backward integration starting at (\mathbf{x}, t_0) are also possible. Nonetheless, this work concentrates on the forward integration at this moment.

The above formulation is appropriate for accumulation in time-dependent settings. In the steady cases, the local attribute values are not dependent on the current integration time but only the location, i.e. denoted by $a_l(\mathcal{C}(\tau))$. More often, in steady cases, the accumulation is performed with a specified length s along the streamlines as in

$$A_g(\mathbf{x}, s) = \int_0^s k(\eta) a_l(\mathcal{C}(\eta)) d\eta. \quad (4.2)$$

Again, this accumulation along a streamline can also be performed in both forward and backward directions. To simplify the subsequent discussion, the Lagrangian accumulation

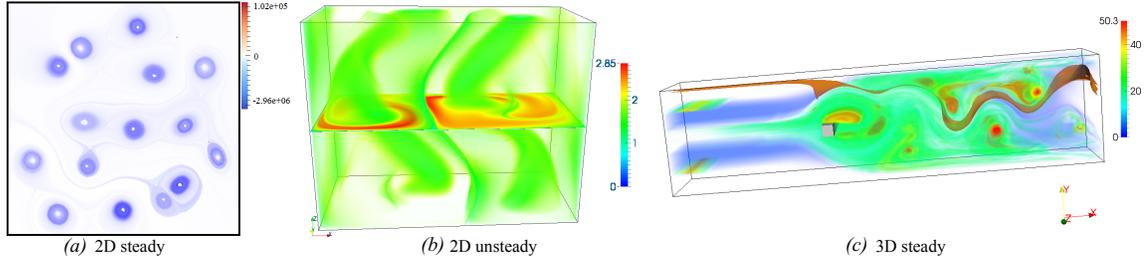


Figure 4.1: A number of \mathcal{A} fields computed based on the accumulation of λ_2 for a 2D synthetic steady flow (a), arc-length of the integral curves for an unsteady double gyre flow (b), and the absolute winding angle of the integral curves for a 3D steady flow behind cylinder (c), respectively. A blue-white-red color coding is used for (a) and (c), while a rainbow color coding is used for (b).

is referred to as the *L-accumulation* for the rest of the dissertation.

Given a spatio-temporal domain $\mathbb{D} = \mathbb{M} \times \mathbb{T}$, a derived scalar field can be obtained (assuming a_l is scalar) from the above convolution, where the value at each sample position is determined by Eq.(4.1) or (4.2). This field is referred to as a Lagrangian Accumulation field or an \mathcal{A} field. The scalar fields discussed in [55] are essentially examples of \mathcal{A} fields. Given different local characteristics of interest to accumulate, one can obtain various \mathcal{A} fields. A discussion on the relationships between several \mathcal{A} fields is provided in the later section. Figure 4.1 provides examples of the \mathcal{A} fields computed based on the collection of λ_2 , arc-length, and change of flow direction along the integral curves, respectively. Given an \mathcal{A} field, its gradient, $\nabla \mathcal{A}$, and the gradient magnitude can be computed, which are used to identify places where the \mathcal{A} field has large changes.

4.2 Properties of L-Accumulation and the \mathcal{A} Fields

This section proposes some detailed discussion on the L-accumulation framework and its Eulerian representation, as well as some of its important properties, in addition to those inherited from the convolution process [38], that may be utilized to understand what flow information can be encoded in the resulting \mathcal{A} field and how to use it to generate some interesting visualizations.

4.2.1 Existence and Uniqueness of $\mathcal{A}(\mathbf{x}, t)$

One important property of a \mathcal{A} field is that given any point $(\mathbf{x}, t) \in \mathbb{D}$ (except at fixed points in steady flows), there is exactly one \mathcal{A} value returned by Eq.(4.1) or (4.2) given the specified parameters. This is due to the uniqueness of integral curves, i.e. in theory there exists exactly one integral curve passing through any give point except at fixed points. This property may seem trivial but it indicates that \mathcal{A} field achieves complete spatial coverage, which enables us to generate a dense visualization of the flow.

4.2.2 Discontinuity in the \mathcal{A} Fields

In mathematics, a function $f(\mathbf{x})$ defined in \mathbb{M} is said to be continuous at \mathbf{c} if for every $\varepsilon > 0$, there exists a $\delta > 0$ such that for all $\mathbf{x} \in \mathbb{M}$ $|\mathbf{x} - \mathbf{c}| < \delta \Rightarrow |f(\mathbf{x}) - f(\mathbf{c})| < \varepsilon$. This condition need not be satisfied everywhere in \mathbb{D} for a \mathcal{A} field. Specifically, for a steady vector field that consists of fixed points, the integral curves (or streamlines) passing through them reduce to points. Therefore, the obtained \mathcal{A} field is not continuous at a fixed point. Also,

the \mathcal{A} field is not continuous across the integral curves that end or start from saddles, i.e. *separatrices*—a special type of streamlines, if the accumulation is performed in infinite time window. This is because an arbitrarily small perturbation in the direction other than the flow direction will lead to another integral curve with length much different from the separatrix, making the \mathcal{A} field accumulated using Eq.(4.2) discontinuous at separatrices. In fact, there are two distinct configurations for this discontinuity, as demonstrated in Figure 4.2. Case(a) of Figure 4.2 shows a scenario that is sensitive to the discrete sampling. The sampling (black dots) shown in this case will miss this discontinuity, since the two opposite sides of the separatrix have similar \mathcal{A} values. In the meantime, the discontinuity illustrated in case (b) is not sampling sensitive compared to (a). Given a sufficient sampling rate, this discontinuity can be captured with certain discrete operators, such as an edge detector [3] or a *discrete* gradient computation, $\nabla\mathcal{A}$. This observation can be used to guide the selection of characteristics for accumulation if the goal is to encode the information of flow separation. That is, the selected characteristic should lead to an \mathcal{A} field that possesses the behavior shown in case(b) of Figure 4.2 at different sides of the separation structure. More details about the discontinuity property and its applications will be discussed in Section 5.2.

In mathematics, a function $f(\mathbf{x})$ defined in \mathbb{M} is said continuous at \mathbf{c} if for every $\varepsilon > 0$, there exists a $\delta > 0$ such that for all $\mathbf{x} \in \mathbb{M}$ $|\mathbf{x} - \mathbf{c}| < \delta \Rightarrow |f(\mathbf{x}) - f(\mathbf{c})| < \varepsilon$. However, this condition may not be satisfied everywhere in \mathbb{D} by a \mathcal{A} field. Specifically, for a steady vector field that consists of fixed points, the integral curves (or streamlines) passing through them reduce to points. Therefore, the obtained \mathcal{A} field is not well-defined there.

The second place where \mathcal{A} may exhibit discontinuous behavior is at the separation

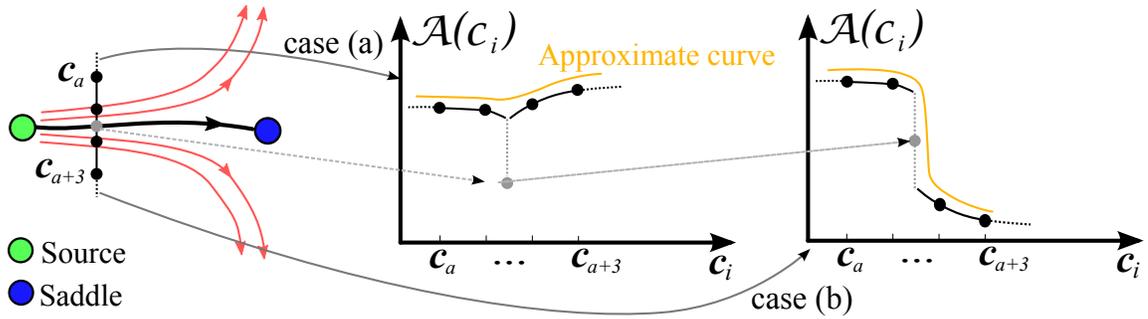
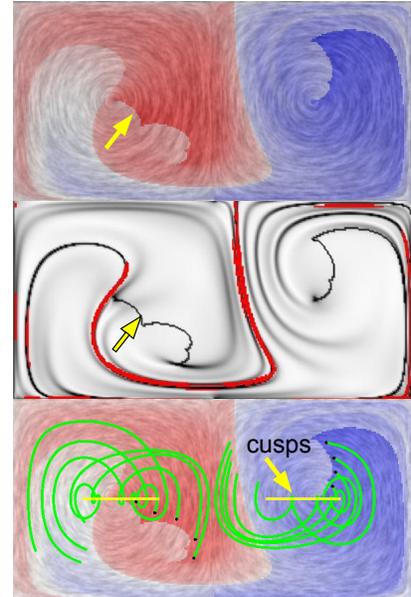


Figure 4.2: Discontinuity of the \mathcal{A} field at a separatrix connecting a saddle (blue dot) and a source (green dot). The \mathcal{A} field is sampled along the line segment traversing through the separatrix. c_i indicate the samples along this segment. Case (a) shows a scenario of the discontinuity that the discrete sampling may miss (illustrated by the orange curve), while the discontinuity in case (b) could be captured with sufficient samples.

structures of the flow. Consider a smooth vector field, the transition of the (geometric) behaviors of neighboring integral curves is smooth. However, this smooth transition is violated at places where the integral curves have structural changes (e.g., end at different fixed points or two far away locations). Those locations correspond to the separation structures in the flow. In many cases, especially in unsteady vector fields, these separation structures are not unique and are sensitive to the selection of the integration time. In contrast, vector field topology is a rigorous notion of the separation structures of steady vector fields, which is defined in infinite long time. In either case, this geometric discontinuous behavior of integral curves may or may not be reflected by the \mathcal{A} fields that accumulate the local characteristics along integral curves. Figure 4.2 shows two possible cases where the \mathcal{A} field (a) misses or (b) captures the topological discontinuity across a separatrix. In case (a), the accumulation values on both sides of the separatrix are similar despite different geometric behaviors of their associated streamlines. Depending on the seeding location and possibly the numerical error, this discontinuity may be missed. In case (b), the

accumulation values on both sides are sufficiently different, capturing the discontinuous geometric behavior across the separatrix.

Does this mean that the discontinuity exhibiting in \mathcal{A} is always a sub-set of the separation structures of the vector fields? To answer this question, let us consider another example shown in the inset to the right. This example shows an \mathcal{A} computed by accumulating the change of the flow direction along the densely placed pathlines for the Double Gyre flow. Beside the well-known separation structure defined as the ridges of the FTLE field, there exists additional discontinuity in the obtained \mathcal{A} as highlighted by the arrows. By close inspection, this cusp-like discontinuity is caused by the abrupt direc-



tional change in the integration of the involved pathlines due to the two oscillating centers. This behavior has already been reported in a previous work [66]. This example indicates that the discontinuity in \mathcal{A} may correspond to discontinuous behavior in the accumulated quantity on neighboring pathlines other than geometric characteristics of the integral curve.

Based on the above discussion and analysis, it can be concluded that under the numerical error free assumption the discontinuity exhibited in \mathcal{A} indeed corresponds to the discontinuous geometric and/or physical behaviors of neighboring integral curves. However, in practice, not all this discontinuity can be captured by \mathcal{A} due to the selection of integration times and seeding strategy. With this observation, it can be argued that the

accumulation framework and the resulting \mathcal{A} fields are simple and effective means to produce an approximate overview on the potential discontinuity in integral curve behaviors, which is known to be relevant to a number of important flow features.

Remark: The highlighted discontinuity in \mathcal{A} may not provide the precise locations and times where and when it happens. Recall the example shown in the above inset. Although the sharp direction change occurs in a later time in the flow, the discontinuity occurs in the first time step where those pathlines are seeded. Although this looks like a disadvantage of the accumulation framework and \mathcal{A} fields, it indeed provides a robust way for seeding and selecting of integral curves that may possess interesting behaviors (i.e. the abrupt change of direction) without extracting those features precisely. Nonetheless, there are still cases that knowing the exact local spatio-temporal regions where those features/events occur is necessary. In that case, additional information needs to be utilized in addition to the accumulated value. One possible solution is to study the variation and distribution of the local attributes along integral curves to provide more detailed information about integral curve behaviors, which should be a valuable direction for future research.

4.2.3 Inequality Property

If k is a box filtering function and the time window is equal to \mathbb{T} , i.e. the whole time range of an unsteady flow or ∞ for a steady flow, then all the points correlated via the same integral curve will get the *same* \mathcal{A} value, while neighboring points that are *not* correlated by the same integral curve may have different values. In this case, the following

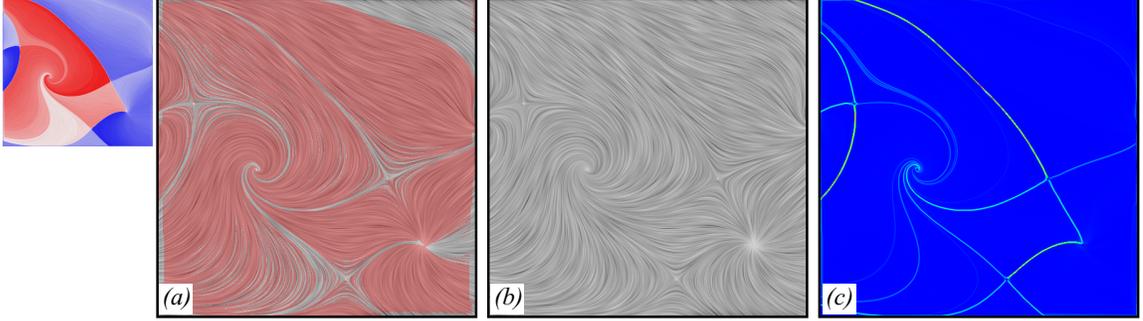


Figure 4.3: (a) highlights the places (in red) of an \mathcal{A} field where Eq.(4.3) does not hold. (b) highlights the places (in red) where Eq.(4.4) does not hold. As can be seen, all places satisfy this new inequality condition. (c) shows the $|\nabla\mathcal{A}|$ using a rainbow color coding. As can be seen, most places have very small gradient values (blue), indicating slow change of \mathcal{A} field there; while places near the separatrices that correspond to certain discontinuity in the \mathcal{A} have large gradient (red). The horizontal direction is X and the vertical direction is Y.

inequality [35] is satisfied.

$$|\langle \nabla\mathcal{A}, V^\perp \rangle| > |\langle \nabla\mathcal{A}, V \rangle| \quad (4.3)$$

where $\nabla\mathcal{A}$ represents the gradient of \mathcal{A} . This inequality depicts that the change of \mathcal{A} along the flow direction is smaller than along a direction perpendicular to the flow. This inequality is employed to evaluate the quality of texture-based flow visualization techniques. However, this may not be always satisfied due to the discrete sampling and the smoothing and blurring effect that occurs during the accumulation in practice as discussed above. Figure 4.3(a) provides an example plot that highlights the places (in red) in an \mathcal{A} field, where the above inequality does not hold. This \mathcal{A} field is computed based on the winding angles of the integral curves. As can be seen in this example, in the majority of the domain the gradient direction of the \mathcal{A} field are closer to the flow direction than its perpendicular direction, except near the discontinuity. Based on [35], visualizing this \mathcal{A}

field will not make a good contrast. In the meantime, this does not mean that the gradient of \mathcal{A} will be parallel to the flow direction. Quite the opposite, in most places they do not match, as shown in Figure 4.3(b). In addition, in most places where Eq.(4.3) does not hold, the gradient value there is rather small (Figure 4.3(c)). Therefore, the above inequality is weakened to introduce the following inequality for the study.

$$|\nabla \mathcal{A}| > \left| \langle \nabla \mathcal{A}, \frac{\mathbf{V}}{\|\mathbf{V}\|} \rangle \right| \quad (4.4)$$

This is true if all points along the same integral curve have the same values using an infinite window size. In this case, the patterns in $|\nabla \mathcal{A}|$, which now are parallel to the patterns of \mathcal{A} , are also aligned with the flow direction. In practice, given different choices of the kernel and window size, the above inequality may not be satisfied everywhere. Nonetheless, due to the blurring effect of the L-accumulation operator, the difference along the flow direction should still be very small if not the least. This property enables us to study the flow patterns based on the *difference* between the accumulated (or average) behaviors of neighboring integral curves, e.g., the extraction of the boundaries of different flow regions.

4.3 Computation and Visualization of \mathcal{A} fields

In the implementation of the Lagrangian framework, a uniform dense sampling strategy is taken to avoid any bias under the assumption that no priori knowledge of the data is known. Given any sample point, an integral curve (i.e. a streamline for a steady vector field or a pathline for an unsteady vector field) is computed using the standard Runge-Kutta fourth-order integrator (RK4) with a fixed step size. The local attribute values are interpolated

at the integration points based on the pre-computed values at the uniform dense samples. It is worth noting that due to the uniform sampling strategy and an axis dependent order, the computed \mathcal{A} may possess certain artifacts or numerical errors. To address this, two additional processes to the original accumulation framework are introduced. First, a dual grid is constructed with the uniform samples as the centers of the grid cells. For each grid cell, a list of the computed integral curves passing through it is recorded. As long as a cell is traversed by an integral curve, this cell is marked visited, and its \mathcal{A} value is computed as the weighted sum of the \mathcal{A} values of the integral curves passing it. The weights are selected based on their distance to the center of the cell. Second, after the initial \mathcal{A} field is obtained, it is further smoothed along the flow direction in a fashion similar to the enhanced-LIC approach [39]. That is, another low-pass filtering process is performed along the short integral curves seeded at the sampling points with the \mathcal{A} field as the input. This additional smoothing can be very useful in cases in which the samples are irregular (i.e. the vertices of a triangle mesh), which is typical for surface flows. Figure 4.4 provide a few examples of the \mathcal{A} fields computed on triangle meshes. In these examples, the streamlines are seeded at the vertices of its respective triangle mesh and integrated sufficiently long (e.g., twice the size of the bounding box of the geometry). Figure 4.4(a, left) shows the initial accumulated \mathcal{A} , which is not smooth. After performing the aforementioned smoothing, the \mathcal{A} field is better aligned with the flow (Figure 4.4(a, right)). The computation times for \mathcal{A} fields depend on the size of the data, the resolution of the samples and the integration time, which can range from a few seconds (e.g., the 2D steady flow) to two hours (e.g., the surface flows) on a PC with an Intel Xeron 1.6 GHz CPU and 8 GB RAM without parallelization.

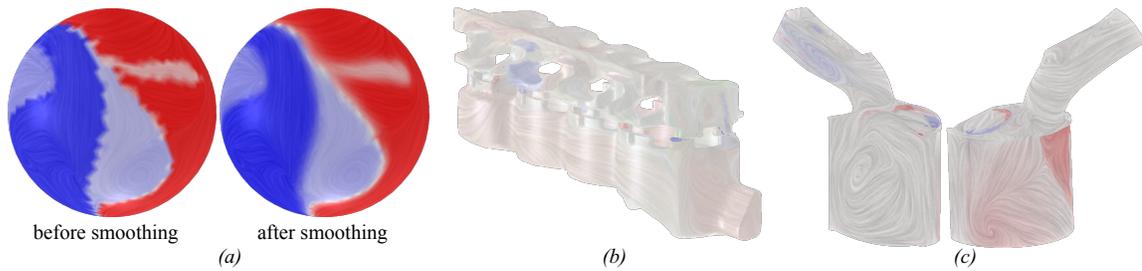
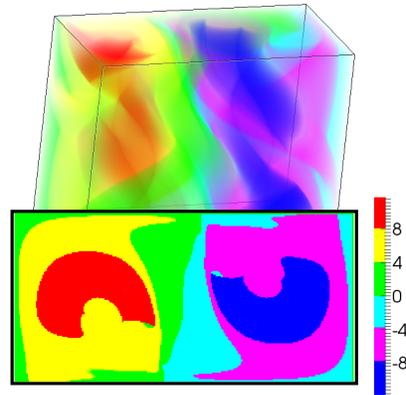


Figure 4.4: \mathcal{A} fields of a synthetic surface flow (a), a cooling jacket simulation (b) and a gas engine simulation (c), respectively.

Pseudo segmentation via discrete color coding With the spatial coverage property and the inequality property that makes the patterns in the \mathcal{A} field aligned with the flow direction, a visualization can be created using discrete color coding to achieve an effect similar to a flow domain segmentation. The inset provides an example of discrete color visualization. Note that no actual segmentation is performed in this visualization. However, a true segmentation may be obtained with this discrete color assignment as the input [73].



Remarks: Because the patterns of the \mathcal{A} fields are aligned with the flow except at fixed points, the direct visualization of \mathcal{A} and $\nabla\mathcal{A}$ fields often provides us an overview of the flow behavior. However, one should also realize that the sensitivity of the \mathcal{A} fields w.r.t the integration times, which may reveal local or global behaviors of the flow in different scales.

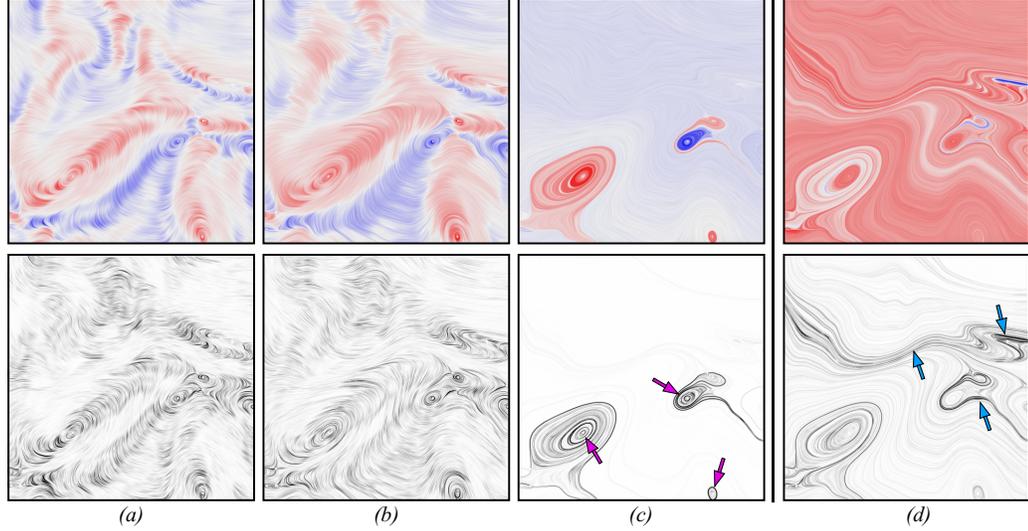


Figure 4.5: \mathcal{A} (top) and $|\nabla\mathcal{A}|$ (bottom) fields of a tile of the ocean simulation with different window sizes for accumulation. (a) 10% of the size of the bounding box of the data domain; (b) 50%; (c) 2,000%. (d) shows the \mathcal{A} field by accumulating the divergence along streamlines.

4.4 A Case Study: Φ Field

In this section, Φ Field, a derived attribute field based on the characteristic of flow rotation, is taken as an example to show the computation and visualization of \mathcal{A} fields.

Φ Field: The Φ field is determined from:

$$\Phi_{\mathcal{C}} = \sum_{i=1}^{N-1} d\theta_i \quad (4.5)$$

where for 2D vector fields $d\theta_i = (\angle(\overrightarrow{P_i P_{i+1}}, \overrightarrow{X}) - \angle(\overrightarrow{P_{i-1} P_i}, \overrightarrow{X})) \in (-\pi, \pi]$ represents the angle difference between two consecutive line segments on an integral curve. \overrightarrow{X} is the direction of 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

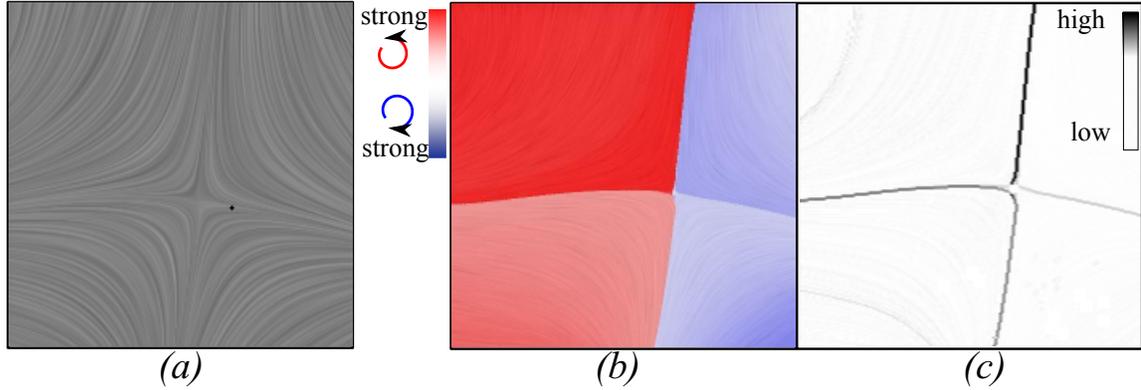


Figure 4.6: An example of Φ field (b) and its gradient magnitude field $|\nabla\Phi|$ (c). (a) shows the LIC of the flow.

rotation is clockwise. For 3D vector fields, $d\theta_i = \arccos\left(\frac{\vec{P}_i\vec{P}_{i+1} \bullet \vec{P}_{i-1}\vec{P}_i}{|\vec{P}_i\vec{P}_{i+1}| |\vec{P}_{i-1}\vec{P}_i|}\right)$ is used, i.e. the absolute difference of the direction of two consecutive line segments on an integral curve. From a global perspective, the Φ field describes the total signed rotation for 2D flows and total absolute rotation for 3D flows along the trajectory. Figure 4.6(b) shows an example of the Φ field visualized using a blue-white-red color coding with blue color corresponding to the negative Φ values, red for positive, and white for zero values.

$|\nabla\Phi|$ Field: The gradient of the Φ -field is defined as $\nabla\Phi = \left(\frac{\partial\Phi}{\partial x}, \frac{\partial\Phi}{\partial y}, \frac{\partial\Phi}{\partial z}\right)$. It is well-known that $\nabla\Phi$ points in the direction where Φ increases the fastest, and its magnitude $|\nabla\Phi|$ indicates the rate of variation of Φ in this direction. Figure 4.6(c) shows a visualization of the gradient of the Φ -field shown in Figure 4.6(a). A gray scale color coding is used with black corresponding to larger $|\nabla\Phi|$ values.

4.4.1 Computation

The computation of the Φ field starts with a flow map estimation. In order to capture the detailed flow information, a *regular sampling* strategy is employed. That is, the domain is partitioned into $N_X \times N_Y \times N_Z$ grid cells. In the experiments, $N_X \times N_Y \times N_Z$ matches the resolution of the image plane in 2D or the original resolution of the data in 3D unless stated otherwise. From each seed \mathbf{x} (at the center of each cell), an integral curve is computed using a 4th-order Runge-Kutta integrator. A linear interpolation scheme is applied in both space and time during integration.

The termination conditions for streamline and pathline computations based on the definition of the Φ fields for steady and unsteady flows is considered separately. For steady flows, given a starting point (x_0, y_0) , a streamline is computed in both forward and backward flow directions until it 1) reaches a boundary of the data domain; 2) reaches a singularity; 3) forms a closed loop; or 4) reaches the maximum number of integration time (or steps) T . Note that criterion 3) is needed because the unpredictably large Φ value near a center-type singularity may make the visualization of the Φ -field challenging. For unsteady flows, given a starting point $(x_0, y_0; t_i)$, a pathline is computed in both positive and negative time until it reaches a boundary of the data domain \mathbb{D} , e.g., the boundary of the physical domain or the boundary of the time range. In other words, a complete pathline for each sample position is computed.

After computing the integral curves starting from the sampled positions with the given integration time (or integration steps) T , the flow rotation is accumulated along the integral curve based on Equation 4.5. Since a regular sampling strategy is employed, $\nabla\Phi$ can be

estimated by evaluating the central difference along each axis direction.

4.4.2 Visualization

The visualization of the Φ field is straight forward. For 2D flows, the Φ field can be shown using color plots or volume rendering with a blue-white-red color coding, where blue represents negative rotation and red represents positive rotation (Figure 4.8(a), Figure 4.7(a)). For 3D flows, since the Φ field is always positive, it can be visualized using volume rendering with a standard rainbow color coding where blue maps small attribute values and red indicates large attribute values. The $|\nabla\Phi|$ field is always visualized with a gray scale color coding (Figure 4.8(b), Figure 4.7(b)).

Figure 4.7 shows the results for the Double Gyre flow. In particular, Figure 4.7(a) shows the volume rendering of the Φ field, from which one can easily observe the strong rotational behavior of the pathlines that is induced by the two vortices can be observed. This can be better revealed by a set of pathlines selected via filtering the Φ field (Figure 4.7(e)). Figure 4.7 (b) shows the $|\nabla\Phi|$ field in the spatio-temporal domain. Two cross sections of this gradient field at $t = 0$ and 20 are shown in the bottom two images of (b), respectively, which are overlapped with the LCS ridges (i.e. the red pixels). From this visual comparison, it can be figured out that the $|\nabla\Phi|$ field captures behavior very similar to the FTLE field. It is worth noting that the proposed accumulation process may indicate the presence of a dynamic event (e.g., strong rotation) that occurs at a different location in the field and at a later time through the value assigned to the seeding location as long as they are correlated by the same pathline. In contrast, the FTLE field does not employ

an accumulated value but instead represents the rate of divergence (convergence) of particles released from a region. The LCS, i.e. the ridges of the FTLE field, does accurately reflect the spatial location of the divergence. Nonetheless, this method does provide a classification for pathlines based on the similarity of their rotational behavior.

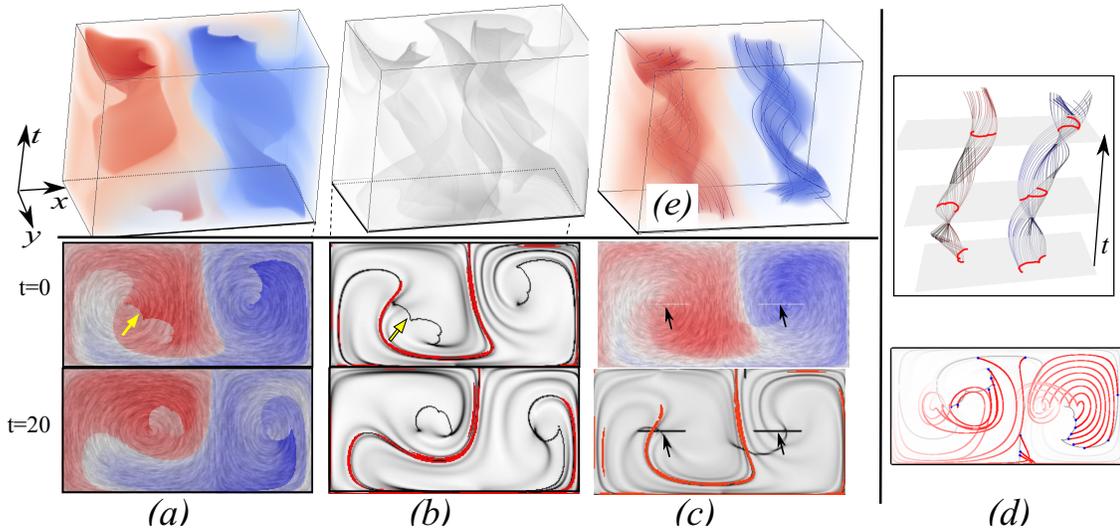


Figure 4.7: The result of the double gyre flow with $T = 10$. (a-b) show the forward Φ field, the comparison between LCS and $|\nabla\Phi|$ ridges at time steps 0 and 20, respectively. For FTLE, a rainbow color coding is used where red indicates larger FTLE values and blue smaller. For the comparison of LCS and $|\nabla\Phi|$ ridges (b), the extracted FTLE ridge points are highlighted in red on top of the $|\nabla\Phi|$ visualization that utilizes a gray scale color map. In addition to the LCS structure, the Φ and $|\nabla\Phi|$ fields also capture the cusp-like seeding curves indicated by the yellow arrows. (c) shows the Φ and $|\nabla\Phi|$ field based on streaklines at $t = 0$, which captures the singularity paths of the two vortices over time as highlighted by the black arrows. (d) shows the pathlines seeded at the cusp-like ridges of the $|\nabla\Phi|$ field. Note that the intersections of these pathlines with the (horizontal) time planes form some coherent structures rotating over time. (e) shows a set of selected pathlines by filtering the Φ field.

Figure 4.8 shows the volume rendering (top) and a number of 2D slices (bottom) of the Φ and $|\nabla\Phi|$ fields for the flow past a cylinder [65] using ParaView. This data set consists of 1001 time steps. To compute the Φ field, we uniformly sample 400×50 particles at

each time step, and integrate the pathlines in both forward and backward time directions until they reach the domain boundary (i.e. $T = 8$). For each pathline, the rotation integration starts from the position where the particle first enters the domain and along the positive time direction until it exits the domain. The measurement of each local rotation is performed by projecting the pathline onto the 2D plane upon which the vector field is defined. The computation of this Φ field took about 3 hours, and the output file storing the pathlines is of size 184 MB. From the result, it can be seen that the Φ field (Figure 4.8(a)) reveals the regions with strong rotational behavior. The blue-white-red color coding effectively conveys the orientation of the rotation. The $|\nabla\Phi|$ field (Figure 4.8(b)) highlights the structure of the discontinuity in the Φ field, which assembles the combined forward and backward FTLE ridges, as shown in (c) and (d). The red pixels represent the LCS whose FTLE values are larger than $0.65f_{max}$ (f_{max} is the maximum FTLE value).

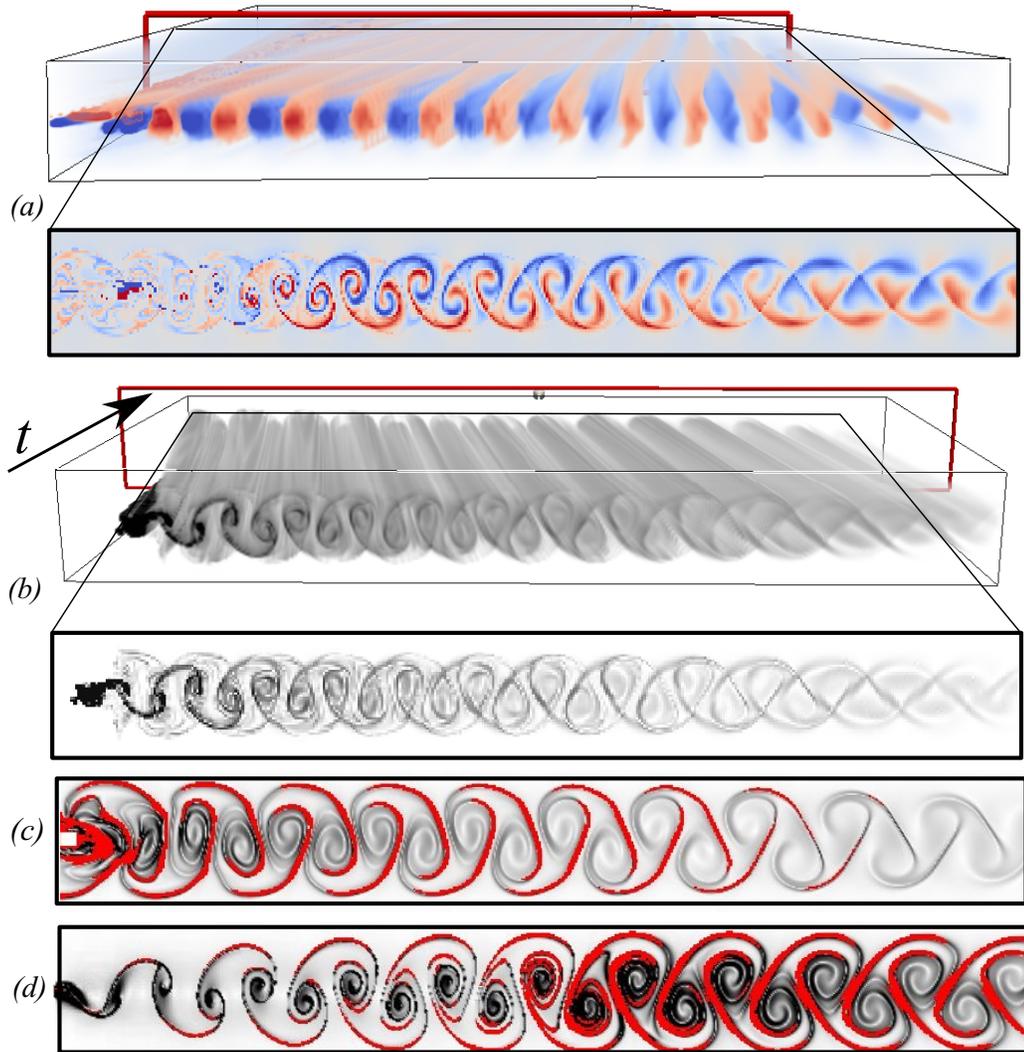


Figure 4.8: The visualization of the Φ (a) and $|\nabla\Phi|$ (b) fields of the cylinder flow data [65]. The time direction increases away from the view. The 2D slices correspond to time step $t_0 = 200$ are shown. (c) and (d) show the comparison of LCS and $|\nabla\Phi|$ field computed based on forward pathlines (c) and backward pathlines (d) at $t_0 = 0$, respectively. The LCS is highlighted in red on top of the $|\nabla\Phi|$ visualization. For each 2D slice horizontal direction is X and the vertical direction is Y.

Chapter 5

Applications of Lagrangian Accumulation Framework

In this chapter, a number of flow exploration tasks are conducted under the the Lagrangian accumulation framework, including integral curve filtering and seeding, ribbon placement and surface seeding. Then the discontinuity study and \mathcal{A} field based flow segmentation are discussed in details.

5.1 \mathcal{A} Field Based Flow Exploration

From the introduction in Section 4.4, certain interesting features of Φ and $|\nabla\Phi|$ fields shown in the visualization are observed. The derived Φ and $|\nabla\Phi|$ fields encode useful information that can be applied to a number of flow exploration tasks.

5.1.1 Integral Curve Filtering

An interface is provided for users to highlight regions with a specified rotation behavior. Specifically, the user can adjust two thresholds, α and β , for the Φ -field visualization, so that regions with $\alpha < \Phi < \beta$ will be colored white. Figure 4.7 (e) provides an example of this type of visualization. Because all points at a given time t_0 on the same integral curve have the same or similar Φ values, performing the above filtering is equivalent to eliminating integral curves whose Φ values are less than the threshold. However, this may still generate a set of integral curves that are too dense. To address this, the data ranges $[\Phi_{min}, \alpha]$ and $[\beta, \Phi_{max}]$ are evenly subdivided into M intervals where M is specified by the user. For each interval, an integral curve whose Φ value is closest to the median of the data range will be shown. With this interface, the user can easily highlight regions with strong positive and negative rotations that may indicate the vortices in the flow. Figure 5.1 (a) shows the vortex regions highlighted by the Φ field.

5.1.2 Integral Curve Seeding

The inequality property and the discontinuity in the Φ fields can be utilized to guide the generation of seeds for integral curves. For a 2D flow, seeds are placed at the points where the $|\nabla\Phi|$ value is larger than a user-specified threshold, e.g., $0.8|\nabla\Phi|_{max}$, where $|\nabla\Phi|_{max}$ is the maximal gradient magnitude in the domain. Integral curves are then computed from these seeds. Randomly started from one seed, those seeds that are too close to the remaining seeds are filtered out. This process is repeated until no more seeds are available. Figure 4.7(d) shows an example of the seeded pathlines in an unsteady flow. The seeds of

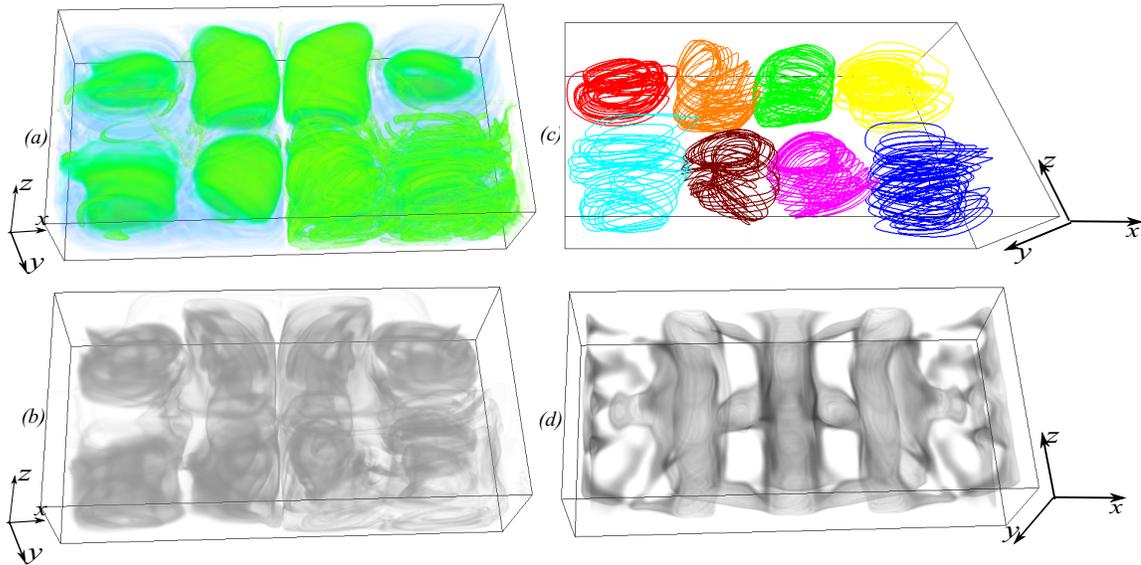


Figure 5.1: The volume rendering of the Φ field (a) and $|\nabla\Phi|$ field (b) of the Bernard flow. (c) shows several filtered streamlines based on the Φ field. (d) shows the local attribute and Q .

the pathlines are placed on the cusp-like seeding curves indicated by the yellow arrow in Figure 4.7 (b).

5.1.3 An \mathcal{A} Field Guided Ribbon Placement

3D ribbons are known to represent effectively flow characteristics that neither integral curves nor integral surfaces can effectively convey. An example of such flow characteristic is the helicity of the flow that quantifies the rotational behavior around an integral curve. To utilize this information to guide the seeding and placement of ribbons, in addition to aggregating the helicity along the individual streamlines to obtain an \mathcal{A} , the standard deviation of the helicity values along each streamlines is derived, denoted by σ . For

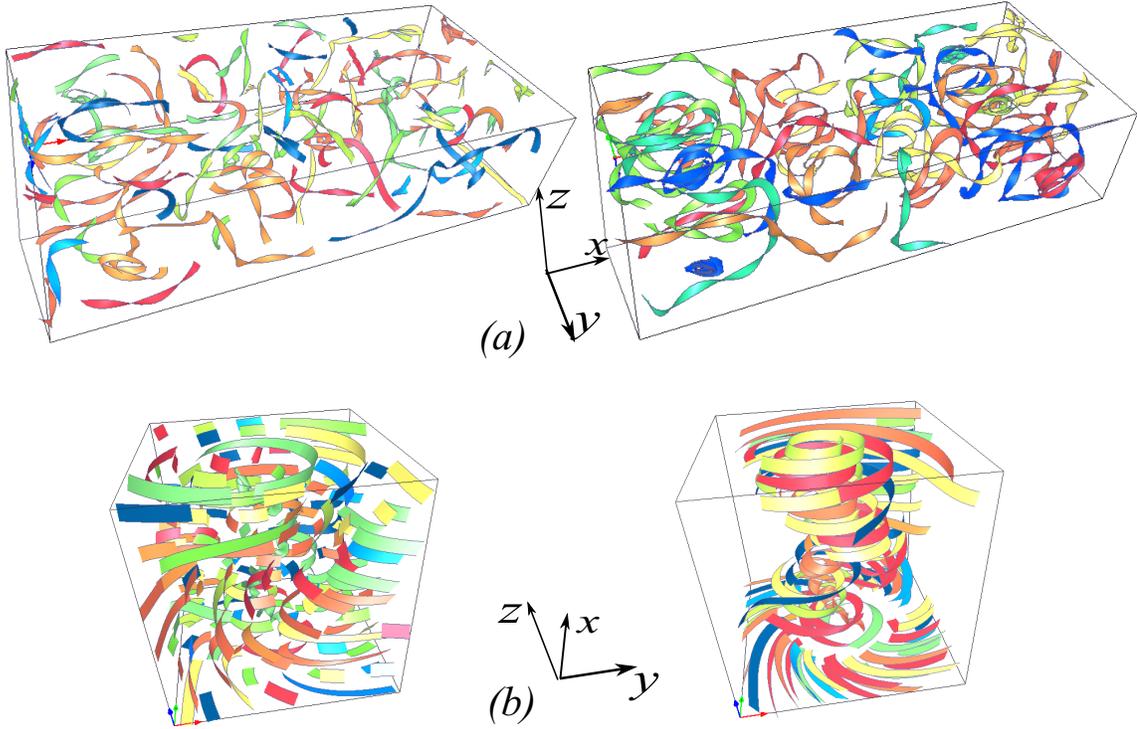


Figure 5.2: Ribbon placement results for the Bernard data (a) and the tornado data (b), respectively. The left image of each group shows the ribbon placement guided by local helicity information, while the right image shows the placement guided by the derived \mathcal{A} field based on helicity.

each candidate seed \mathbf{p} , a value of $\mathcal{A}(\mathbf{p}) + \sigma(\mathbf{p})$ is assigned. Based on this value, all candidate seeds that are uniformly distributed in \mathbb{D} are ranked. From the top-ranked seeds, a series of ribbons as the initial set of ribbons can be constructed. Then, new ribbons are iteratively inserted to fill the blank region of \mathbb{D} while keeping a minimum user-specified distance away from other existing ribbons. The similarity metric introduced by Chen et al. (2007) [8] is used to further remove redundant ribbons that are too similar to the existing ones. Figure 5.2 shows the ribbon placement results using the proposed \mathcal{A} field guided framework. Compared to the ones that are produced using only the local attributes (i.e.

the initial ribbons are placed at locations with maximum local attribute values), our results tend to generate ribbons with longer length that can provide more coherent information about the flow behaviors (i.e. the tornado and the four vortices of the Bernard data are easily identifiable), which is expected.

5.1.4 An \mathcal{A} Field Guided Surface Seeding

An integral surface is the integration of a 1D curve (i.e. seeding curve) through 3D flows. Compared with the individual integral curves, integral surfaces can more effectively convey 3D flow information with the additional visual cues (e.g., lighting, transparency and textures). However, not all integral surfaces are intrinsic. They highly depends on the selection of the seeding position and the shape and orientation of the seeding curve. Generating good seeding curves that can lead to expressive surface representations of the flow is still a challenging task. With the computed \mathcal{A} and its gradient information, a simple yet effective seeding curve generation strategy is proposed. In particular, the candidate seed \mathbf{p}_c that has the smallest $|\nabla\mathcal{A}|$ value is selected. Let us denote the \mathcal{A} value at \mathbf{p}_c by g . Next, a seeding curve starting from \mathbf{p}_c is generated and guided by the curvature field [45], whose points have \mathcal{A} values falling in the range $[g - \delta, g + \delta]$. The obtained seeding curve encodes streamlines, the variation of whose \mathcal{A} values is not larger than δ . Thus, the computed stream surface from this seeding curve is expected to have small variation. In the meantime, a candidate seed \mathbf{p}'_c that has the largest $|\nabla\mathcal{A}|$ value can be selected, from which a seeding curve guided by the $\nabla\mathcal{A}$ field is generated. The computed stream surface from this seeding curve is expected to have large variation according to the meaning of the $\nabla\mathcal{A}$ field (i.e. it highlights the places where \mathcal{A} has large changes). Figure 5.3 shows two

surfaces computed from the two seeding curves constructed using the above two strategies for the flow behind the cylinder data, respectively. The blue surface was generated from a seeding curve with small variation of \mathcal{A} values along it, which highlights the boundary of a small vortex bundle next to the cylindrical object. In contrast, the red surface was generated from a seeding curve with large variation of \mathcal{A} values. This surface exhibits rich and varying flow behaviors around the boundaries of various vortices.

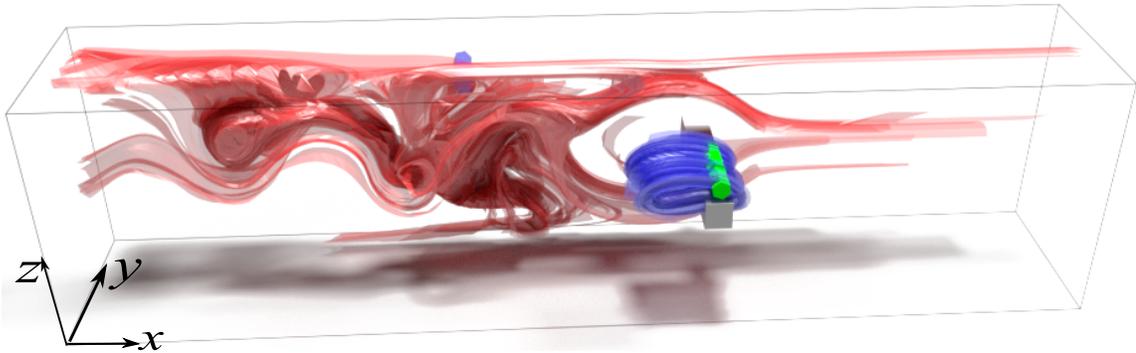


Figure 5.3: Comparison of two strategies of seeding curve generation. The red surface is constructed from a seeding curve derived using the small variation strategy, while the yellow is from a seeding curve derived using the large variation strategy. The seed of seeding curve for the blue surface is located inside the bundle, where its $|\nabla\mathcal{A}|$ value is small, i.e. the \mathcal{A} values along this seeding curve are almost constant. In contrast, the seed position of the seeding curve for the red surface is located near the boundary of the domain, where the $|\nabla\mathcal{A}|$ value is large, and the variation of the \mathcal{A} values on this seeding curve is also large.

5.1.5 Visualizing Particle-based Data Aided by \mathcal{A}

In addition to applying the accumulation framework to mesh-based vector field data, it is also utilized to aid the visual exploration of the particle-based flow data. Different from the previous examples where the integral curves are computed to depict the trajectories of

mass-less particles, the particles in the particle-based data have mass and their trajectories need not be the integral curves of the corresponding velocity field. Nonetheless, the accumulation framework still applies. In this case, the accumulated value of a particle indeed describes the overall attribute behavior of the particle. Figure 5.4 shows an \mathcal{A} field computed based on the change of the moving direction (i.e. a_8) of the particles produced by a dam-breaking simulations computed using the position-based fluid method [34]. From the result it can be seen that particles that hit the boundary have larger change of moving direction, as highlighted by the arrows

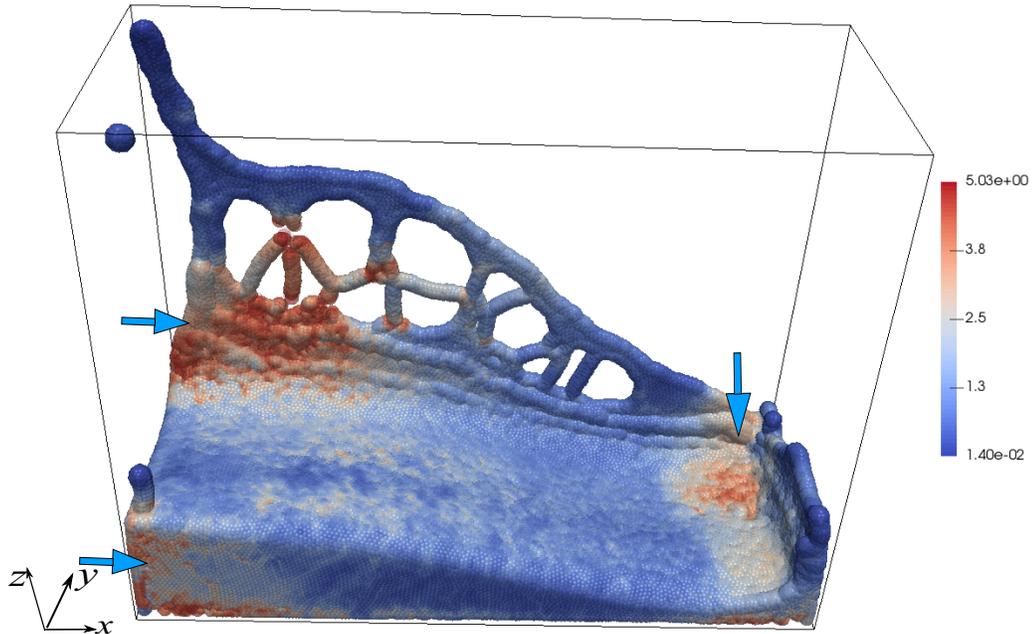


Figure 5.4: Visualization of an \mathcal{A} field derived from a dam-breaking particle based simulation. Blue means the change of particle moving direction is small, while red mean large. It shows that the particles that hit the boundary have larger change of moving direction, as highlighted by the arrows.

5.2 Discontinuity Study

The discontinuity property of \mathcal{A} fields is introduced in Section 4.2.2. In this section, more details of this property are discussed and an algorithm to extract the discontinuity among neighboring integral curves is proposed.

5.2.1 Background of Discontinuity

From the discussion of discontinuity, it has already been known that the spatio-temporal positions correlated by the same integral curves will have similar attribute values, while those neighboring points traversed by integral curves that possess different behavior will have different attribute values. The 1D plots in Figure 5.6 show the attribute values along the seeding line segments (i.e. the red segments in Figure 5.5 left) . They exhibit cliff-like, sharp changes (highlighted by the blue arrows), which correspond to certain *discontinuities* in the corresponding attribute field. This discontinuity may be closely related to certain flow features, such as flow separation, as shown in Figure 5.5.

There are a number of potential reasons leading to this discontinuity. In general, without providing a rigorous proof that the obtained \mathcal{A} field from a steady flow exhibits discontinuity at the locations where the integral curves have abrupt changes in their geometric property, e.g., being truncated. Some examples of these integral curves include those that connect with boundary switch points, where the flow transits from inflow to outflow or vice versa, and those that form closed loops where the integration is terminated earlier. Again, if the goal is to encode this discontinuity in the \mathcal{A} fields, one should select the appropriate characteristic that leads to an \mathcal{A} field that possesses distinct behaviors at different sides of

these special integral curves to avoid the influence of discrete sampling, although this may not be trivial in practice.

For an unsteady flow, the above discontinuity may not exist due to its finite time range. However, sharp changes in the \mathcal{A} field may still be observed at the ridges of transportation, i.e. the Lagrangian coherent structure (LCS) due to a similar reason. To simplify the subsequent discussion, this sharp change is still referred to as some discontinuity.

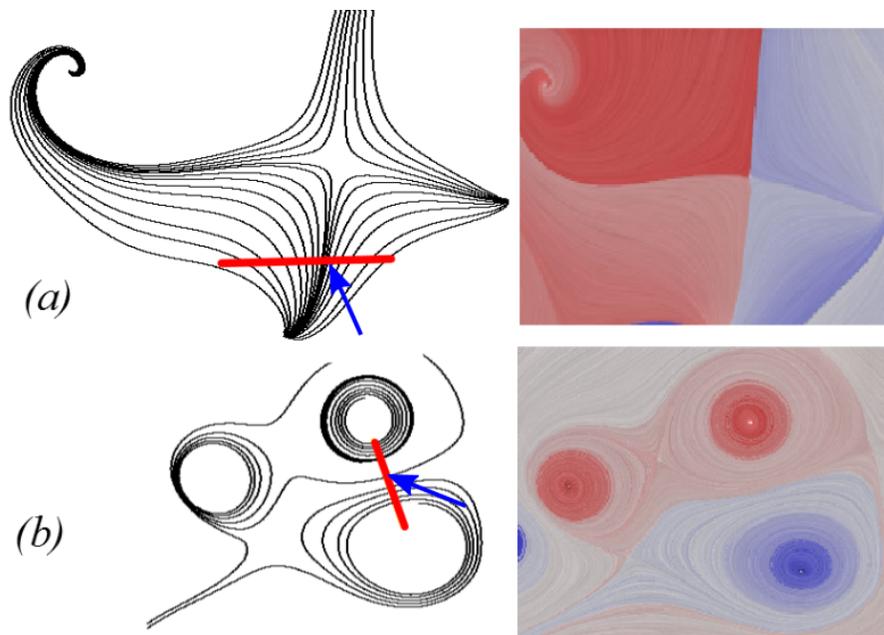


Figure 5.5: The illustration of the relation between the attribute field and a number of well-known flow features, including the flow separation (a) and vortices (b). The left column shows the vector fields illustrated by streamlines, the middle column shows the rotation field.

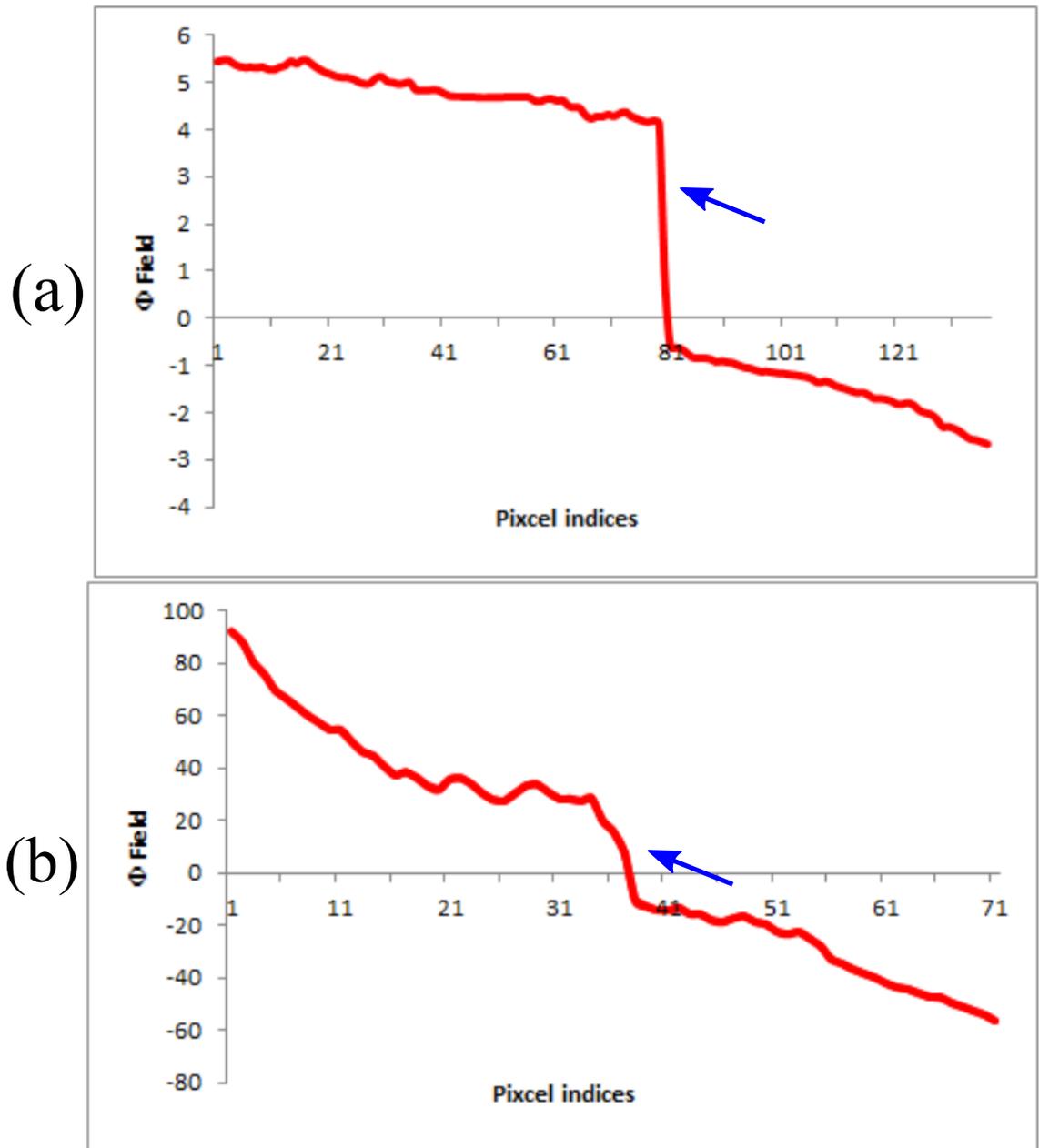


Figure 5.6: The plots of the rotation field of the streamlines intersecting with given seeding line segments (shown in Figure 5.5). Note that the discontinuities (sharp gradients) in the rotation field indicate the flow features. The values shown in the plot have no units as they do not correspond to any physical quantities.

It is also worth mentioning that other abrupt changes of the local characteristics along the integral curves could also lead to certain discontinuity in the resulting \mathcal{A} field. One example that has been recently reported is the cusp-like behavior in pathlines [66], which is caused by the abrupt change in the pathline direction, i.e. almost a π angle difference between the previous and current directions. This abrupt change is caused by the intersection of a pathline with the paths of singularity. The \mathcal{A} field that is computed by accumulating the change of flow direction along the pathlines can effectively capture this discontinuity (Figure 4.7). Based on the same reason, the abrupt change in the local physical properties of the flow along the integral curves may also lead to similar discontinuity in the resulting \mathcal{A} field.

In practice, due to numerical errors introduced by the integrators and the varying window sizes used to compute the accumulation (discussed next), the discontinuity may occur in places that need not correspond to well-known flow features as discussed above. However, if the accumulation is performed along the entire integral curves in the domain \mathbb{D} so that all points along the same integral curves obtain the same \mathcal{A} value, due to the inequality property (Section 4.2.3), the direction of the discontinuity in the \mathcal{A} field is aligned or can be made to align along the flow direction, which guarantees that no misleading information will be provided in the subsequent visualization. In the rest of this work, the discontinuity in a \mathcal{A} field is referred as to the *salient flow patterns* under the given accumulation window.

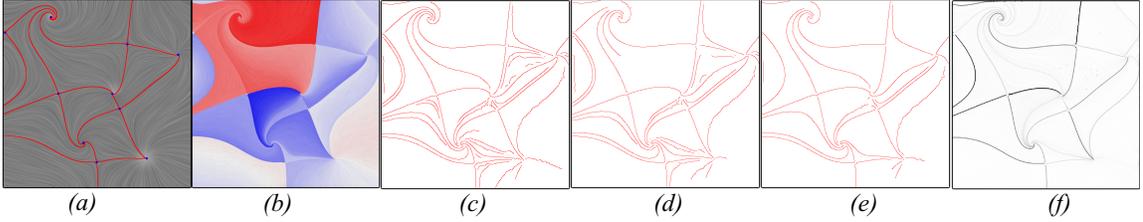


Figure 5.7: Discontinuity detection on a Φ field derived from a synthetic steady flow using the Canny edge detector with different combinations of parameters. (a) The differential topology with LIC as the background; (b) Φ field; (c-e) Detected edges with different parameters of the Canny edge detector: (c) - $\sigma = 3.0$ $\alpha = 0.3$ $\beta = 0.8$, (d) - $\sigma = 3.0$ $\alpha = 0.6$ $\beta = 0.8$, (e) - $\sigma = 3.0$ $\alpha = 0.3$ $\beta = 0.86$; (f) The gradient of Φ field. The horizontal direction is the X direction and the vertical direction is the Y direction.

5.2.2 Discontinuity Extraction

As shown in Figure 5.5, the attribute fields may contain discontinuities that correspond to the sharp gradients in the integral curve behavior. These discontinuities in the attribute fields are similar to the edges in a digital image. The gradient of the attribute field may be able to locate these discontinuities (Figure 5.7(f)), but may require non-intuitive thresholds to reveal the salient ridges. Therefore, the more robust Canny edge detector [3] is selected to locate this discontinuity in the attribute fields, which can be converted into 2D images. The Canny edge detector has three input parameters: σ - the standard deviation of the Gaussian smoothing filter, α - the low threshold and β - the high threshold. The lower row of images in Fig 5.8 shows the detected edges from the corresponding attribute fields of the Double Gyre flow. Note that for the average direction field *avgDir*, the field values of the two neighboring pathlines be equal to or close to 0 and 2π , respectively, as highlighted with the arrows in Figure 5.8(b). But it does not indicate the discontinuity because they are in the same (or be close to) direction. Therefore, this case is filtered using the Canny edge detector.

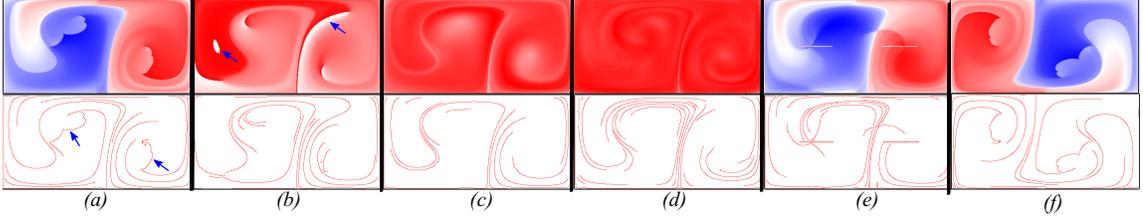


Figure 5.8: Illustration of a number of attribute fields derived from the Double Gyre flow and their detected edges. (a)–(d) show the attribute fields Φ , L , $avgDir$ and $acceration$, $avgDir$ computed from pathlines, respectively. (e) is the rotation field Φ from streaklines. (f) is the rotation field Φ from pathlines using backward integration. The parameters of Canny edge detector are $\sigma = 2.0$, $\alpha = 0.3$, $\beta = 0.8$. The horizontal direction is the X direction and the vertical direction is the Y direction.

5.2.3 Combined Attribute Fields

The study of the combined attribute fields is proposed to understand the behavior of the discontinuities in different attribute fields. To achieve this, one can simply overlap the detected edges from different attribute fields, as shown in Figure 5.9(a). However, the detected edges from the individual fields are independent of each other. With this simple overlapping, it is difficult to know whether their corresponding attribute fields have similar behavior or not (i.e. both are descending, or one is descending while the other is ascending) at the locations that exhibit sharp change. This information may be revealed in the combined attribute field.

Assume $\mathcal{A}_i, i = 1, 2, \dots, n$ represent the attribute fields introduced in Chapter 4. Three combination strategies are proposed to compute a super attribute field \mathcal{A}_{com} .

Linear combination is defined as $\mathcal{F}_{com} = \mathcal{F}_i + \mathcal{F}_j$, where \mathcal{F}_i and \mathcal{F}_j are selected attribute fields from the attribute fields pool. However, if one of the selected attribute fields

has a much larger value range, the super field will be dominated by this attribute field. Figure 5.9(b) is the result of combined super field from the rotation field Φ ($[-11.73, 11.73]$) and the length field \mathcal{L} ($[0., 2.8]$), which shows mostly the features of the rotation field.

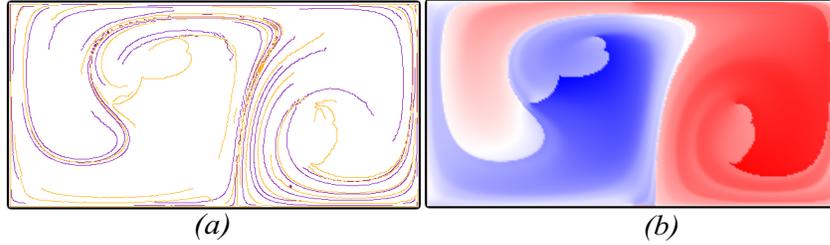


Figure 5.9: (a) Overlap of edges in Φ (yellow) and \mathcal{L} (purple) fields; (b) Simple combination of Φ and \mathcal{L} fields. The horizontal direction is the X direction and the vertical direction is the Y direction.

Weighted combination is employed to address the issue of the simple combination. Here, $\mathcal{F}_{com} = \alpha \widehat{\mathcal{F}}_i + \beta \widehat{\mathcal{F}}_j$, where $\alpha + \beta = 1$ and satisfies $0 \leq \alpha \leq 1, 0 \leq \beta \leq 1$. $\widehat{\mathcal{F}}_i$ and $\widehat{\mathcal{F}}_j$ are the normalized values of the attribute field \mathcal{F}_i and \mathcal{F}_j , respectively. Figure 5.10(a) shows the super fields computed using the weighted combination of the Φ and $avgDir$ fields of the Double Gyre flow, with the weight for the Φ field being 0.1, 0.5, and 0.9, respectively. With this weighted combination, the discontinuity structure in the super field that is non-sensitive to the choices of weights can be further identified. That is, no matter what weight combination is selected, the derived super field always contains this discontinuity, which is composed of *stable edges*. Figure 5.10(b) shows these stable edges as gray curves super-imposed onto the edges extracted from the corresponding super field. In this example, nine super fields, in which the weight of the Φ field is $\alpha_1 = 0.1, 0.2, \dots, 0.9$, respectively, were generated to identify the stable edge.

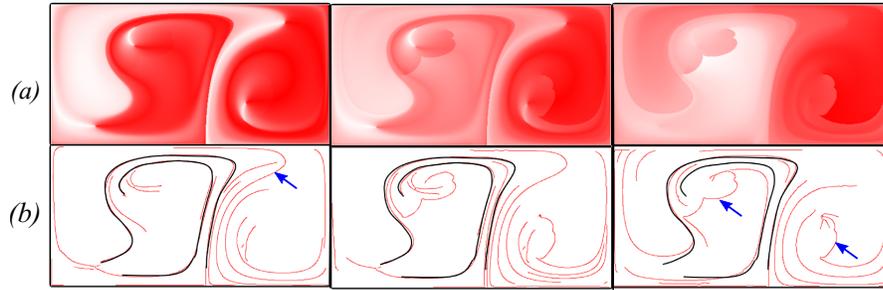


Figure 5.10: Illustration of weighted combination of Φ and $avgDir$ field. (a) Combined attribute field; (b) Edges detected from the super field. The dark gray curves on top are stable edges that do not change with weights. The weights of Φ and $avgDir$ from left to right are $\alpha_1 = 0.1, \alpha_2 = 0.9$; $\alpha_1 = 0.5, \alpha_2 = 0.5$; $\alpha_1 = 0.9, \alpha_2 = 0.1$, respectively. The parameters of Canny edge detector are $\sigma = 1.0, \alpha = 0.3, \beta = 0.8$. The horizontal direction is the X direction and the vertical direction is the Y direction.

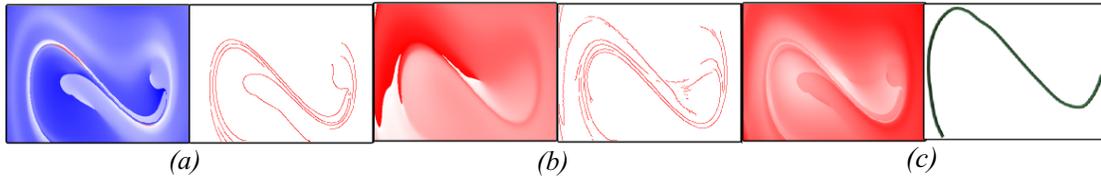


Figure 5.11: Results of the forced-damped Duffing system. (a)–(b) Φ and $avgDir$ fields and their detected edges. The parameters of Canny edge detector are $\sigma = 1.8, \alpha = 0.3, \beta = 0.9$. (c) A super field using the equally weighted combination of $\Phi, avgDir, L, nsV$ and $seDist$. The horizontal direction is the X direction and the vertical direction is the Y direction.

5.2.4 Results and Applications

The attribute field has been applied based analysis and exploration framework to a number of synthetic and real-world 2D vector fields. The cost of pre-computation of attribute fields depends on the resolution of the spatio-temporal domain and the time window of trajectory integration. Pathline-based attribute field computation requires 10 to 32 seconds for the data sets considered in this work, while streakline-based attribute field computation requires about 4 to 15 minutes. All processing times are measured on a PC with an Intel

Core i7-3537U CPU and 8 GB RAM.

The first example is the Double Gyre flow with a spatial resolution of 256×128 , which has been shown earlier. For the second example, a dynamical system defined by the forced-damped Duffing oscillator [20] is considered

$$\begin{aligned}u(x, y, t) &= y \\v(x, y, t) &= x - x^3 - 0.25y + 0.4 \cos(t)\end{aligned}\tag{5.1}$$

with the constant spatial divergence operator -0.25 , which is a non-area-preserving. A spatial resolution of 800×600 and a time window $T = 5$ is chosen. The attribute fields of the system and the corresponding detected edges are shown in Figure 5.11(a). The detected edges from each attribute field encode the LCS information. Figure 5.11(c) upper shows a super field generated from an equally weighted combination of all six attribute fields. Figure 5.11(c) bottom illustrates the detected common edges.

Another example is a simulation of a 2D unsteady flow behind a square cylinder with a Reynolds number of 160 [65]. A spatial resolution of 400×50 is used to compute the attribute fields. The time window for this data set is 3. Figure 5.12 shows the attribute fields and the corresponding detected edges. While the edges detected in all of the attribute fields encode at least part of the LCS of the flow, the non straight velocity field nsV (Figure 5.12(b)) also reveals the swirling behavior of the flow clearly.

5.2.5 Conclusion

In this chapter, an algorithm to extract the discontinuity of the behavior between neighboring integral curves is proposed. It shows that this discontinuity may be closely related to a

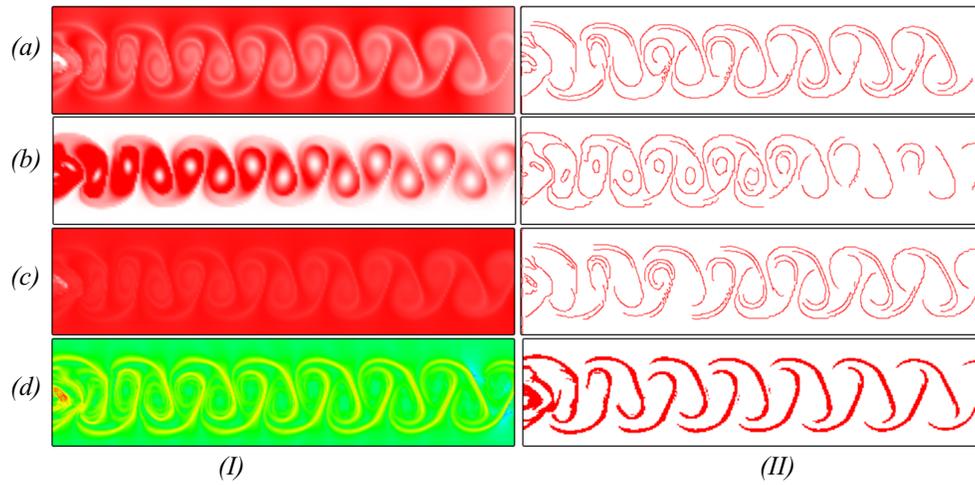


Figure 5.12: Attribute fields of the flow behind cylinder and detected edges. (a) $avgV$ field; (b) nsV field; (c) L field; (d) $FTLE$ and LCS . The parameters of Canny edge detector are $\sigma = 1.0$, $\alpha = 0.3$, $\beta = 0.8$. The horizontal direction is the X direction and the vertical direction is the Y direction.

number of flow features. Different strategies to combine individual attribute fields to form a super attribute field to study the spatial correlation of the attribute fields are also studied.

5.3 *A* Field Based Flow Segmentation

5.3.1 Motivation

Existing flow segmentation techniques typically partition the flow domain based on either certain local flow characteristics, such as vector magnitude and orientation, or its topological structure [51]. Specifically, segmentation techniques that are based on local flow information usually perform vector field clustering in a hierarchical fashion, i.e. either top-down or bottom-up [41]. 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 5.18(b) for an example). Therefore, additional computation is needed in order to generate other visual primitives, such as stream-lets or glyphs [41], to convey information about the flow behavior within each segment. On the other hand, vector field topology [28, 43] 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 [33] and select [70] 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

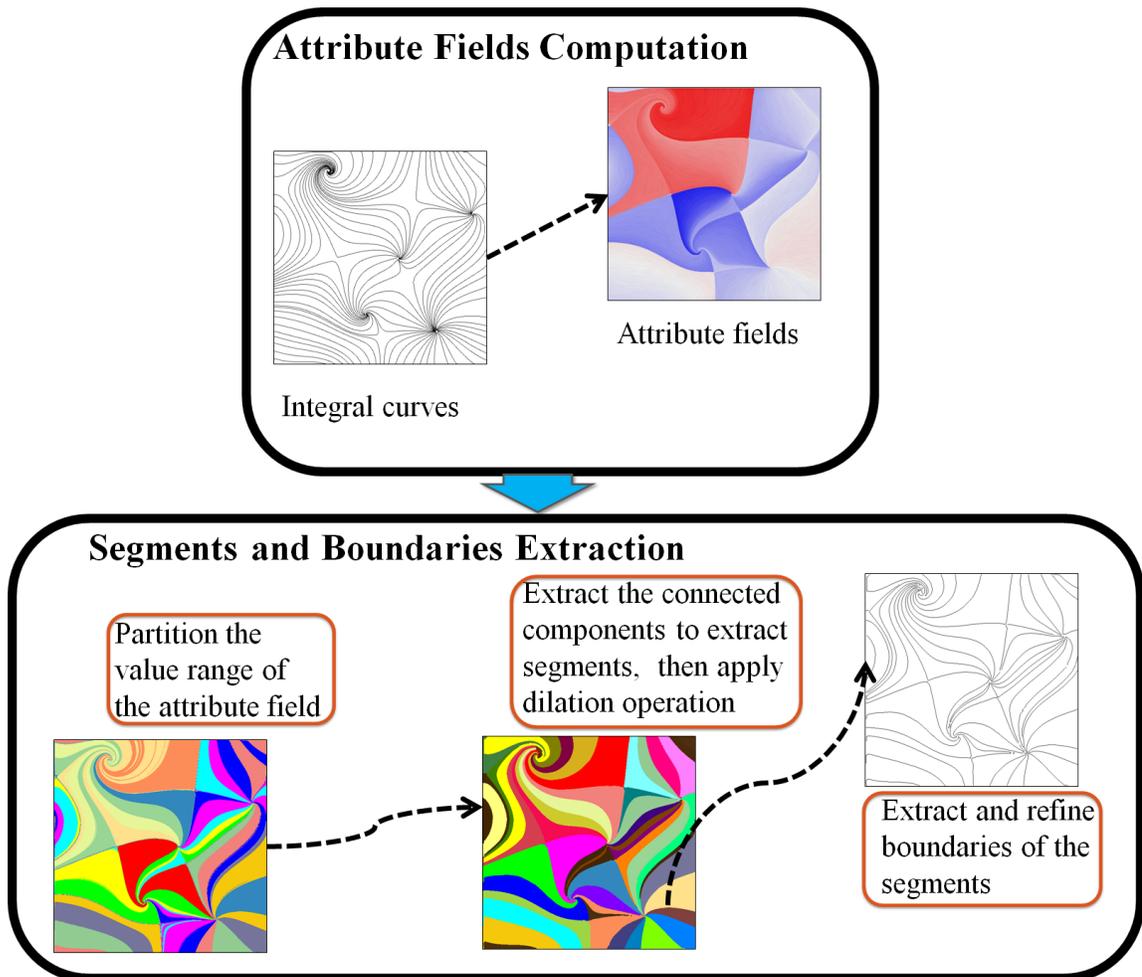


Figure 5.13: The pipeline of the proposed method. Note that it only illustrate the procedural of the algorithm and does not provide the results, therefore, there is no need to show axes or units.

higher-dimensional spaces, although integral surface placement is a much harder problem than integral curve placement [10].

In this section, a segmentation framework is introduced, which is based on integral curve attributes and is applicable to both 2D and 3D vector fields. The proposed 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. This framework consists of two major components (Figure 5.13). First, several *attribute fields* are derived 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 [42, 54]. If the integral curve is computed over a long time window, the attribute field will encode the global behaviors of the vector field [72].

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 [26]. Compared to [26], where the goal was to generate implicit clusters produced by semantic dependencies, this method generates a segmentation based on flow information only. With the aid of the derived attribute fields, the sampled spatial positions are first classified 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. Then dilation operations are performed 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, these boundaries are further smoothed. Unlike the boundary refinement strategy applied in [31] This method does not require computing many contours and comparing them to the coarse segmentation curves.

5.3.2 Segmentation Algorithm

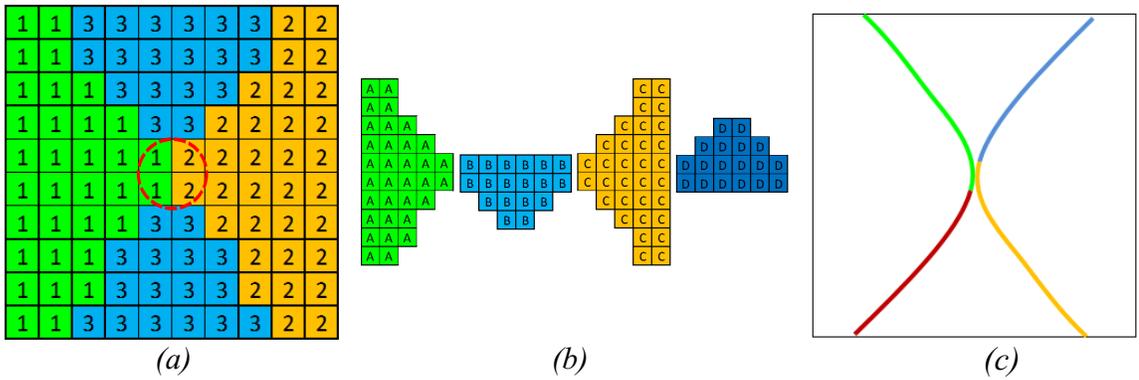


Figure 5.14: The illustration of the representation of attribute values and segments. (a) An input attribute field with labels of bin IDs. (b) The segments based on the attribute field. (c) The boundaries of the segments.

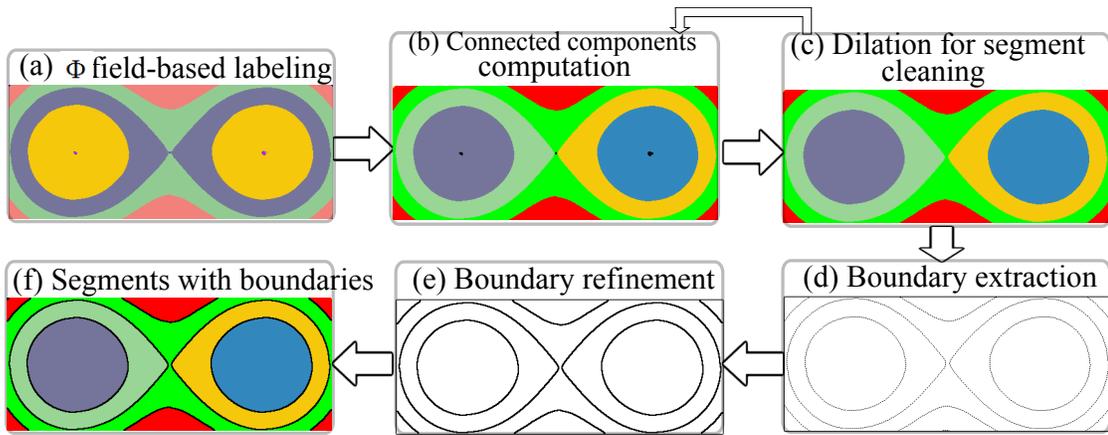


Figure 5.15: The pipeline of the segmentation algorithm.

Based on the derived attribute fields, the flow segmentation problem is converted into a scalar field partitioning problem. Figure 5.14 illustrates the output segmented regions of the vector fields and their boundaries. This algorithm can be divided into two steps. As

illustrated in Figure 5.15, the first step is to extract the segments from the region classification (top row of Figure 5.15) based on the attribute field values. The second step is to extract and clean the boundaries of the segments (bottom row of Figure 5.15). The details will be described later.

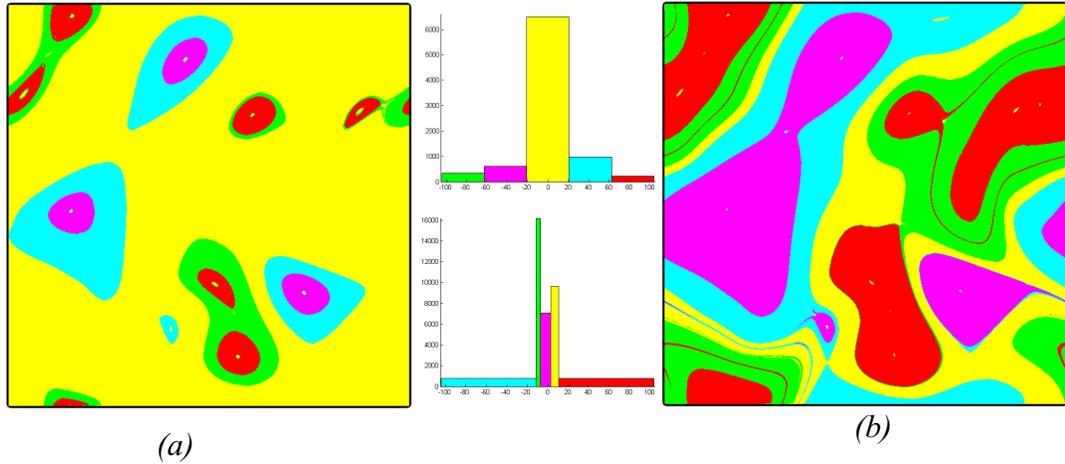


Figure 5.16: Region classification based on different methods. (a) uniform distribute on data range. (b) equal-sizes bins. The histograms show the data distribute under each method. The bin number $m = 5$.

5.3.2.1 Segment Extraction and Cleaning

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

Region classification Different flow regions are firstly classified 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 5.14(a)). However, this simple strategy may result in initial region classification with largely different sizes, as shown in Figure 5.16(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 5.16(b)). Without further specification, most 2D segmentation results shown in this section adopts the second initial partitioning strategy. For the 3D flow data sets, the region classification is performed manually based on the data distribution histogram to avoid generating too large bins (i.e. clusters with too many points). However, it does not enforce equal size requirement in 3D. The actual partitioning strategy is provided 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, the segments are extracted 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 5.15(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, it needs to remove these small or noise segments.

In order to determine whether a small segment is noise or not, a *noise segment threshold* γ is introduced, 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 between the corresponding bin IDs. Figure 5.17(a) shows a number of noise segments highlighted by arrows. The dilation operation is applied, which is one of the basic morphological operations in image processing [15], to remove these segments. Specifically, the flow domain is converted 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. The dilation to 3D is extended to remove the noise segments in 3D vector fields. 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 5.17(b) illustrates the segmentation result after applying dilation to Figure 5.17(a), where the noise segments are removed.

Note that, when equal-size bin partitioning strategy is applied to generate the region

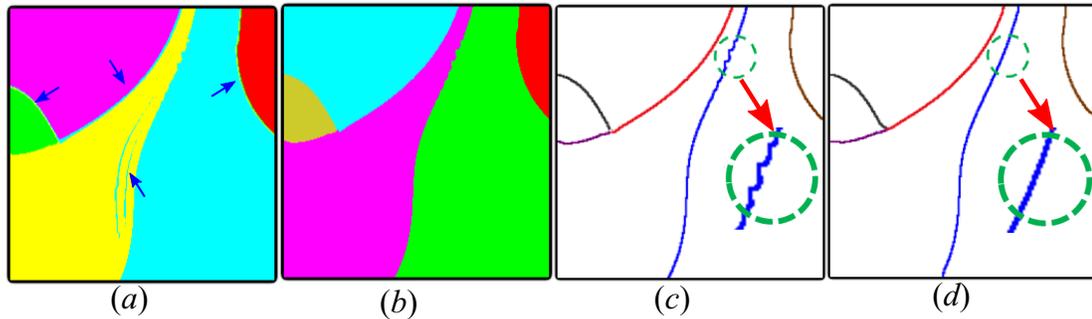


Figure 5.17: The illustration of the effect of dilation operation and boundary refinement. (a) The segmentation before dilation operation. (b) The segmentation after dilation operation. (c) The extracted boundaries of (b) without refinement. (d) The extracted boundaries of (b) with refinement.

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, the local strategy described above is taken, 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).

5.3.2.2 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. (2006) [31]. 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 this segmentation algorithm, a boundary extraction algorithm is proposed, which consists of two steps. In the following, the method to handle 2D segmentation is described first, 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 [7] can be applied to identify the boundaries between segments. In practice, the boundary curves between segments can be estimated using the boundary points of one of the two neighboring segments. In order to smooth the boundary curve, it needs to connect these boundary pixels in the correct order.

To extract boundary curves, the boundary points can be saved in an array. Starting from any point in the array, two directions are traced 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 5.14(a) are the ends of several boundary curves (Figure 5.14(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 it needs 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 5.17 (c) and (d) illustrate the extracted boundaries of Figure 5.17 (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, an iso-surface is utilized to estimate the boundaries of the 3D segments. Specifically, the ID of a selected 3D segment is re-assigned 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 5.22 and 5.23. 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.

5.3.3 Results

The method has been applied to a number of synthetic and real-world vector field datasets, including steady and unsteady flows for both 2D and 3D.

5.3.3.1 2D Steady Vector Fields

Figure 5.13(right), Figure 5.18(c) and Figure 5.19(column (a)) show the proposed segmentation results for a number of synthetic and real-world 2D steady flows, respectively.

As a comparison, the segmentation is computed using the image-space vector field clustering technique introduced by Peng et al. (2012) [41], which adopts a bottom-up strategy and aggregates the two most similar clusters each time until only one cluster is left. Figure 5.18(b) shows the result of the image-space clustering for 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, result generated by the proposed method shown in Figure 5.18(c) generates segments whose boundaries are better aligned with the flow. By looking at the segments or the boundaries of the segments (Figure 5.18(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 (2013) [35], which is beyond the scope of this work.

Effects of the number of bins m Figure 5.19(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 [56]. Each time step corresponds to one day. The first time step of the dataset, i.e. slice #20106, is selected to generate the 2D steady vector field. As what can be seen, 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 5.19 (b) shows the extracted boundaries of the corresponding segmentations shown in Figure 5.19 (a). These boundaries are

similar to the streamlines seeded on the boundaries, as shown in Figure 5.19 (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 5.19 (b) and (c), but they are sufficiently close to each other.

5.3.3.2 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 [65]. 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). The first 200 time steps are chosen and a resolution of $400 \times 50 \times 200$ is used to compute the attribute fields based on pathlines. The time window for the pathline computation is 3. Figure 5.20 shows the segmentation results of this dataset based on different attribute fields, i.e. determinant field, non-straight velocity 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 5.20(c)) encode the LCS information of the flow and those from the non straight velocity field nsV (Figure 5.20(b)) reveal the Von Karman vortex street.

Noise segment threshold γ Figure 5.21 shows the segmentation of the Double Gyre flow [52] based on the rotation field. When γ increases from 0.01 to 0.05, the four segments, highlighted in black in Figure 5.21(a), merge with their neighboring segments after

dilation (Figure 5.21(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 5.21(d), the pathlines seeded in those regions display similar behaviors in each region.

5.3.3.3 3D Steady Vector Fields

To test this method in a steady 3D flow framework, an instant is taken from a 3D time-dependent flow behind a square cylinder with a Reynolds number of 160 [65]. It covers the spatial domain $[-12, 20] \times [-4, 4] \times [0, 6]$. The spatial resolution of this dataset is $192 \times 64 \times 48$. The attribute fields based on 3D streamlines are computed first. Figure 5.22(a) shows the segmentation of this dataset based on the curl field. Eight segments are generated with the bin number $m = 3$. Figure 5.22(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.

Figure 5.23 shows the segmentation and estimated boundaries of the Bernard flow [67], whose domain is $[-16, 16] \times [-4, 4] \times [-8, 8]$. From the eighteen segments generated with bin number $m = 5$, eight are selected, whose average rotation field value is above 525.6. The estimated boundaries of the eight segments shown in Figure 5.23 (a) highlight the eight vortex regions of the Bernard flow. There are also some segmented regions covering the whole flow domain rather than a specific vortex region. Figure 5.23 (b) shows such a segment.

As discussed earlier, the boundaries of the segments generated by the proposed 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, the 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.

5.3.3.4 3D Unsteady Vector Fields

This framework is also applied to a 3D unsteady flow, i.e. the unsteady ABC vector field [27]. The parameters are $A = \sqrt{3} + 0.5t\sin(\pi t)$, $B = \sqrt{2}$ and $C = 1$. The spatial domain considered here 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 5.24(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 5.24(b)). Three of them highlight the vortex regions of the flow, whose boundaries are estimated with the iso-surfaces shown in Figure 5.24(c).

5.3.4 Conclusion

In this chapter, a vector field segmentation algorithm based on derived attribute fields is proposed. A number of attribute fields are computed based on the accumulation of local properties along the integral curves. Then the connected components are extracted based on the classification of the attribute values and apply dilation to filter the noise segments in the segmentation results. Finally the boundaries of the segments are extracted

and smoothed in order to obtain a cleaner segmentation for visualization. The segments generated by the proposed 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. The proposed segmentation can be applied to 3D steady flows, where iso-surfaces is computed to estimate the boundaries of the segments. It shows that these iso-surfaces could potentially be used to visualize high dimensional flows in a similar fashion to integral surfaces.

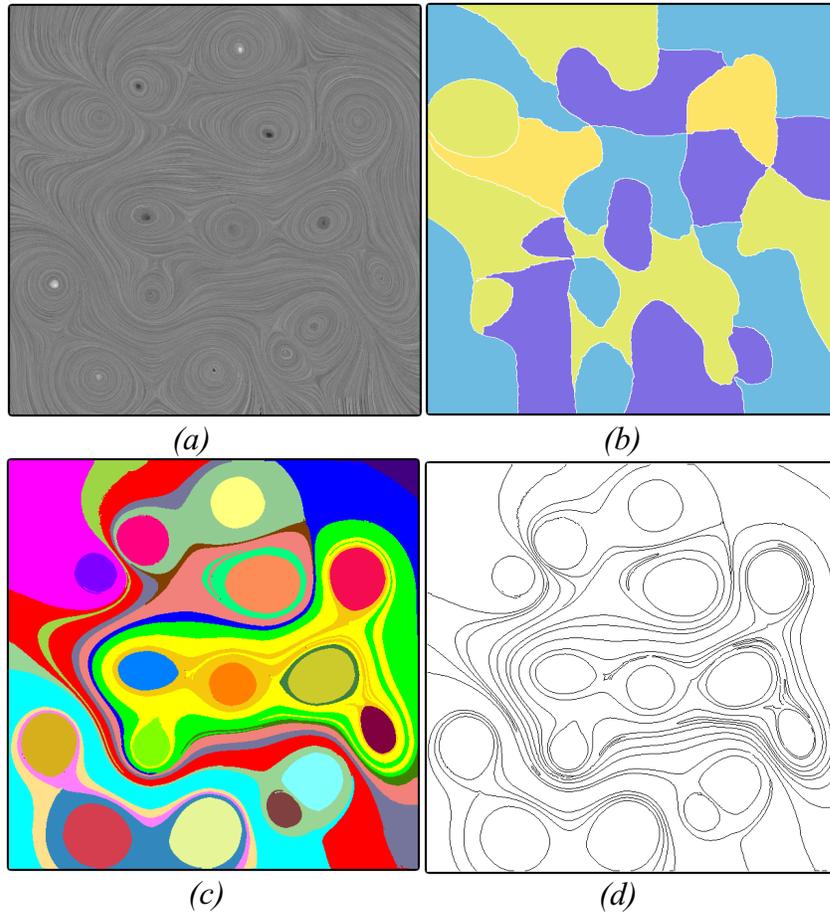


Figure 5.18: Comparison between the bottom-up algorithm and the proposed method with a synthetic flow . (a) The LIC of the flow; (b) The segmentation result for the bottom-up algorithm based on the direction of the flow. (c) The segmentation result for the proposed algorithm based on the rotation field. (d) The boundaries of the segments in (c).

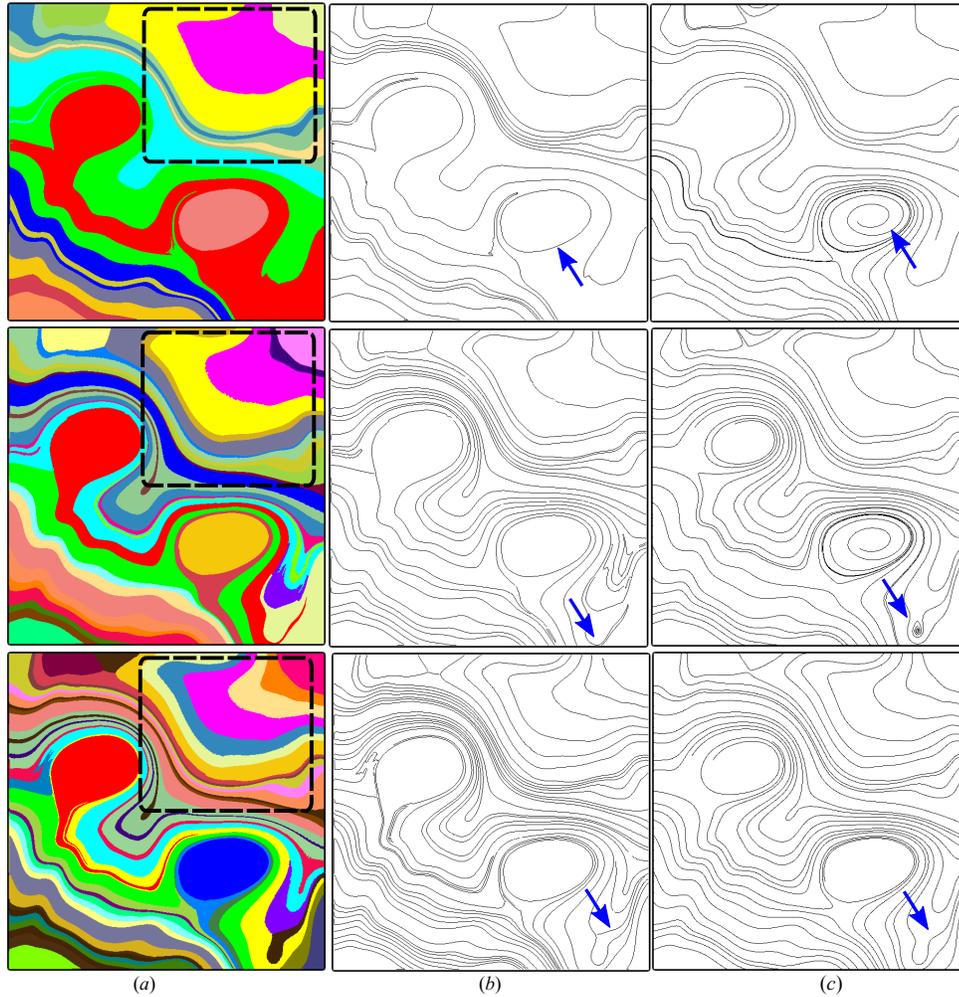


Figure 5.19: The effect of the number of bins m for the initial region classification. Column (a) shows the segmentation results based on the rotation field with m as 5, 8 and 15 from top to bottom, respectively. Column (b) are the extracted boundaries of the corresponding segmentation in (a). Column (c) shows the streamlines seeded on the extracted boundaries.

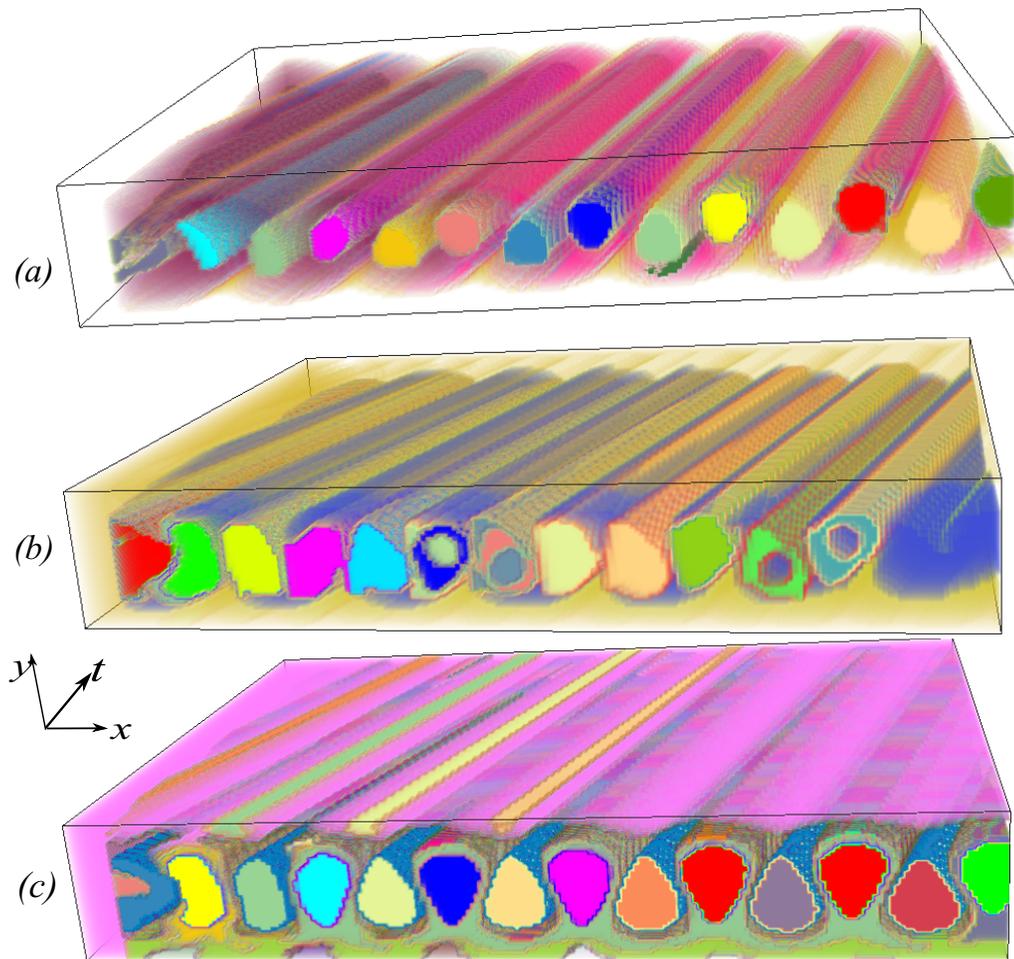


Figure 5.20: Segmentation of a 2D unsteady flow behind a square cylinder based on different attribute fields: *determinant* field (a), *nsV* field (b) and *curl* field (c), respectively. The bin number $m = 6$.

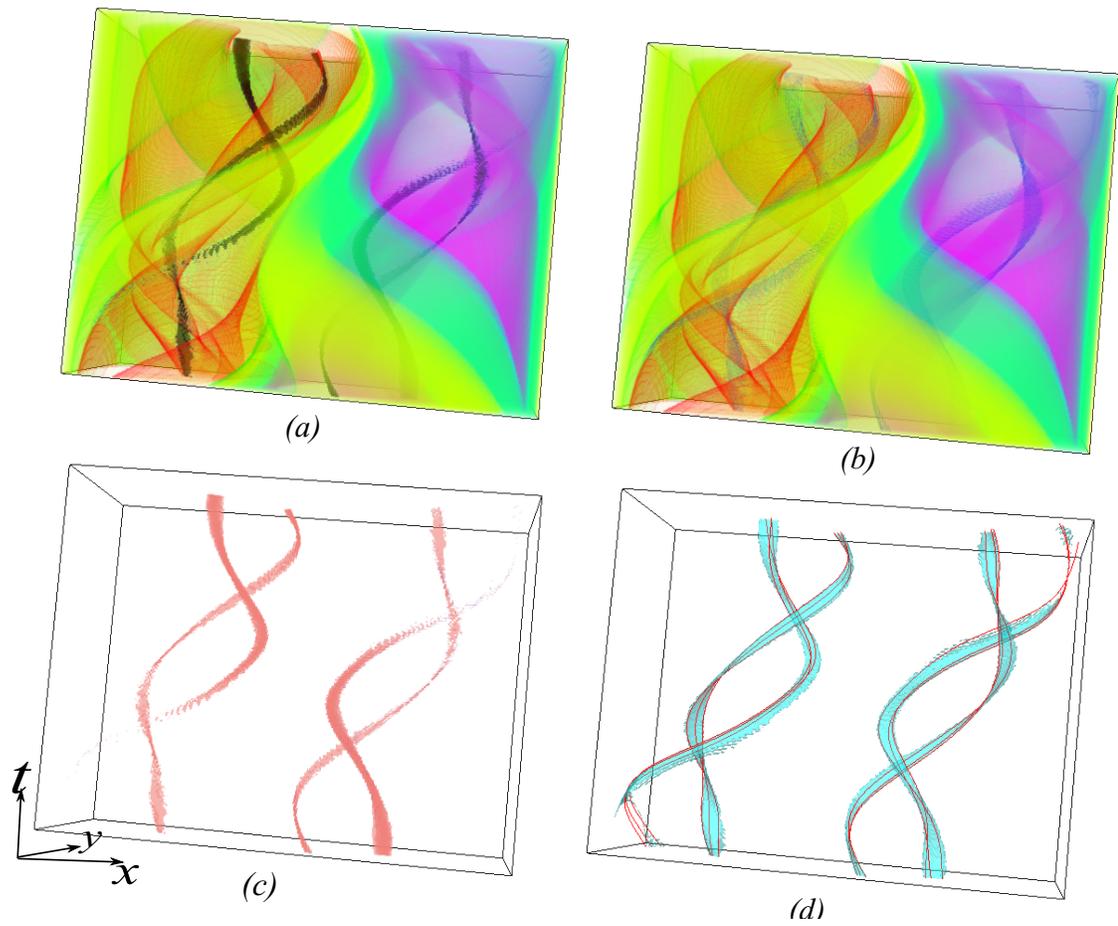


Figure 5.21: The segmentation result of an unsteady Double Gyre flow with different noise segment threshold γ . (a) $\gamma = 0.01$; (b) $\gamma = 0.05$; (c) The four γ -sensitive segments; (d) The estimated boundaries of the four γ -sensitive segments with sampled pathlines (red). The bin number $m = 6$.

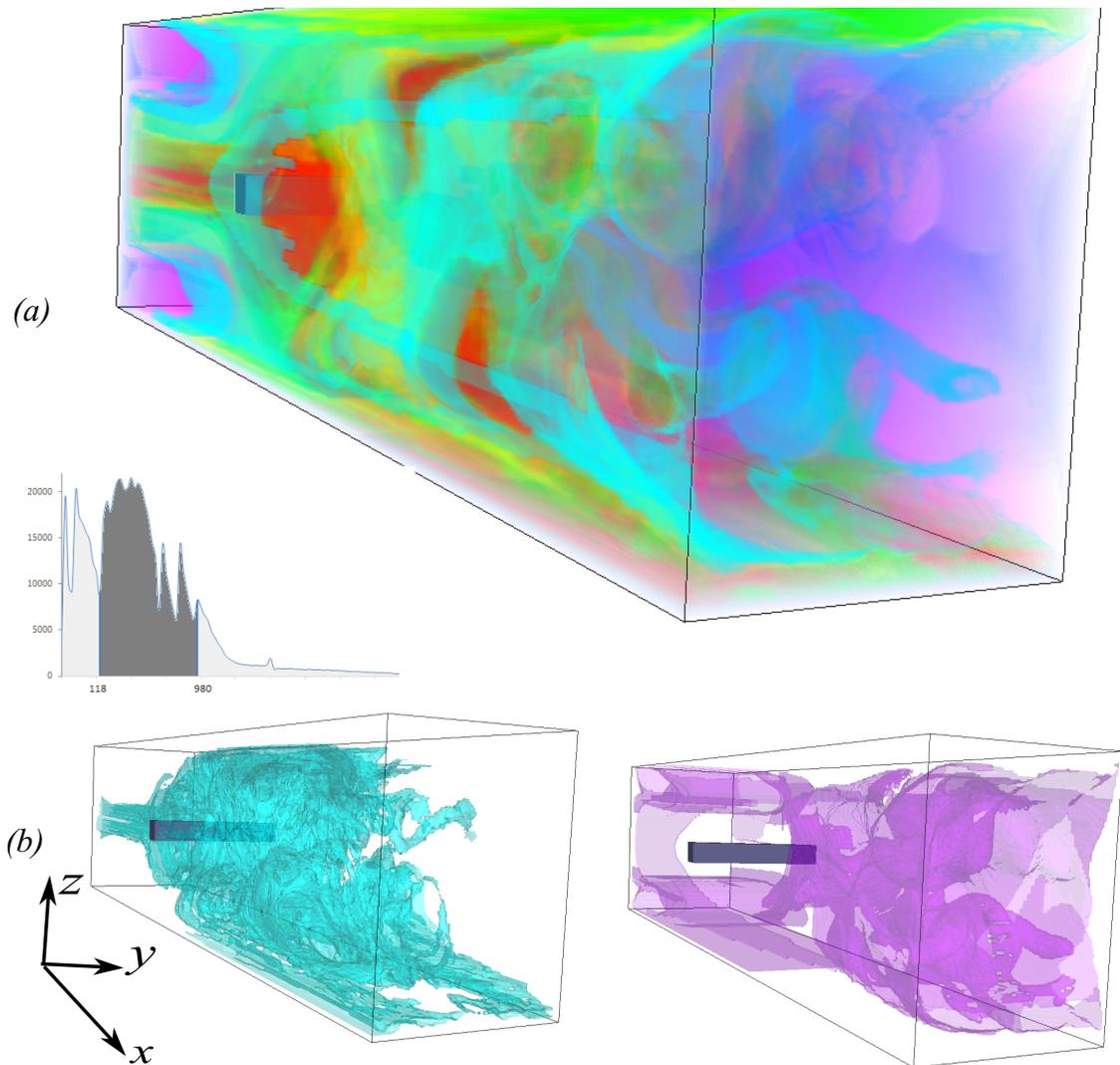


Figure 5.22: Segmentation and estimated boundaries of a 3D steady flow behind cylinder. (a) The segmentation result based on the curl field. (b) Several segmentation boundaries generated using the iso-surfaces. The bin number $m = 3$. The legend aside shows the partitioning strategy for initial region classification.

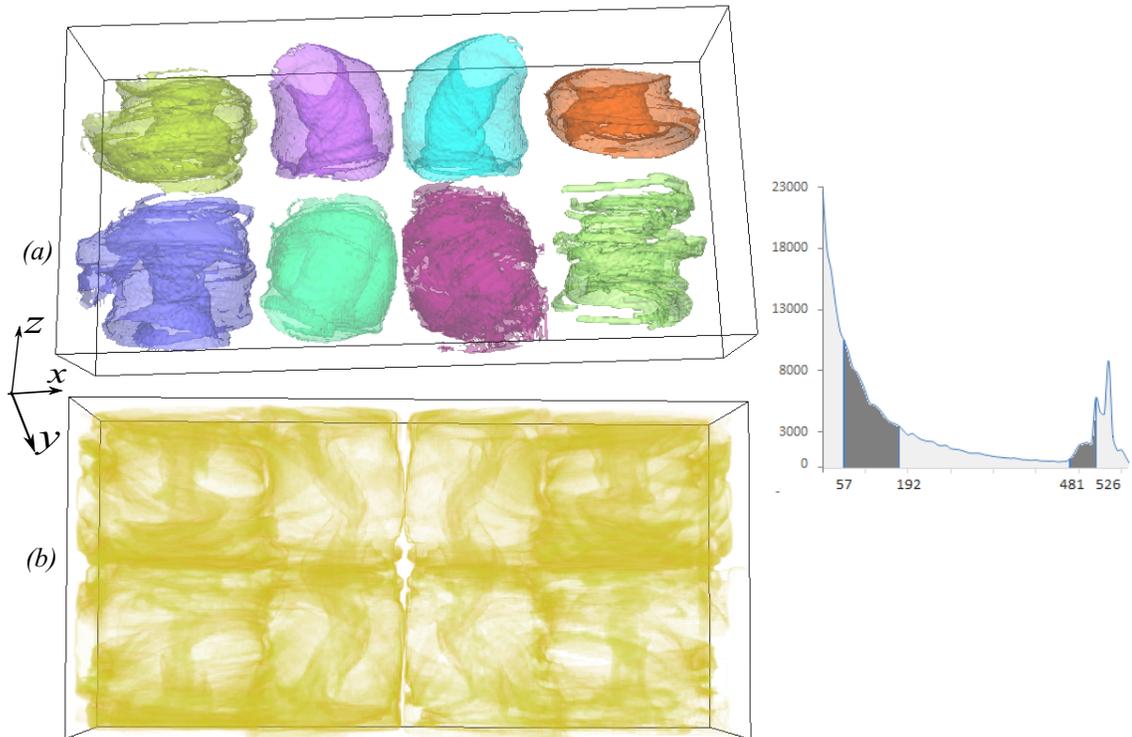


Figure 5.23: Segmentation and estimated boundaries of the Bernard flow. (a) Eight segmentation boundaries generated from the iso-surfaces. (b) one of the generated segments. The bin number $m = 5$. The legend aside shows the partitioning strategy for initial region classification, where the X axis shows the range of the attribute field in each bin, and the Y axis illustrates the number of samples in each bin.

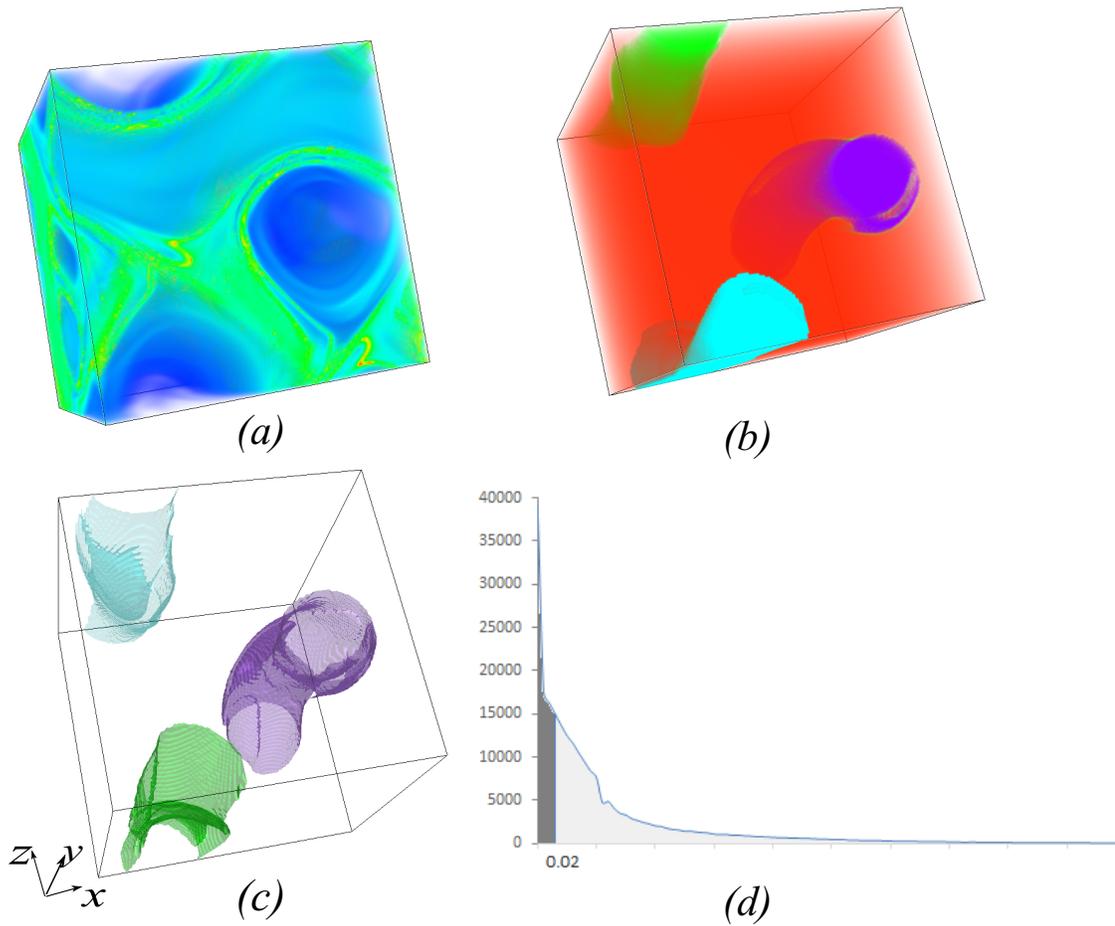


Figure 5.24: Results of the unsteady ABC flow projected into $t = 0$. (a) Visualization of the rotation field. (b) Four segments of the rotation field with bin number $m = 2$. (c) Three segmentation boundaries generated from the iso-surfaces. The bin number $m = 2$. (d) The legend shows the partitioning strategy for initial region classification, where the X axis shows the range of the attribute field in each bin, and the Y axis illustrates the number of samples in each bin.

5.4 Discussion

In this section, An informal study of relations among different attribute is first conducted. Then the attribute selection and the window size parameter is discussed. At the end, Lagrangian accumulation is compared with the Eulerian accumulation, and it shows the ability of the Lagrangian accumulation frame work extended to non-integral curves.

5.4.1 An Informal Study of Relations Among Attributes

Considering the large number of \mathcal{A} fields that can be used to describe various flow behaviors, it would be interesting to see how their corresponding attribute fields are correlated. In this section, An informal study is conducted, which discusses the relation among a number of selected geometric characteristics of the integral curves and their corresponding flow properties. It starts with the pairwise scatter plots among different \mathcal{A} fields.

To understand how well different \mathcal{A} fields correlate with one another, a scatter plot matrix is constructed based on the Double Gyre flow, as shown in Figure5.25. Each of the entries of this matrix shows a scatter plot with two attributes as its X and Y axes, respectively. Based on this matrix, the following useful relations can be found.

Length Field L vs. Average Particle Velocity Field $avgV$ These two attributes show a strong linear relationship (entry highlighted by the purple box in the matrix). This is because the arc-length of each integral curve is equal to the sum of the velocity magnitude, multiplied by the integration step-size, measured along this curve.

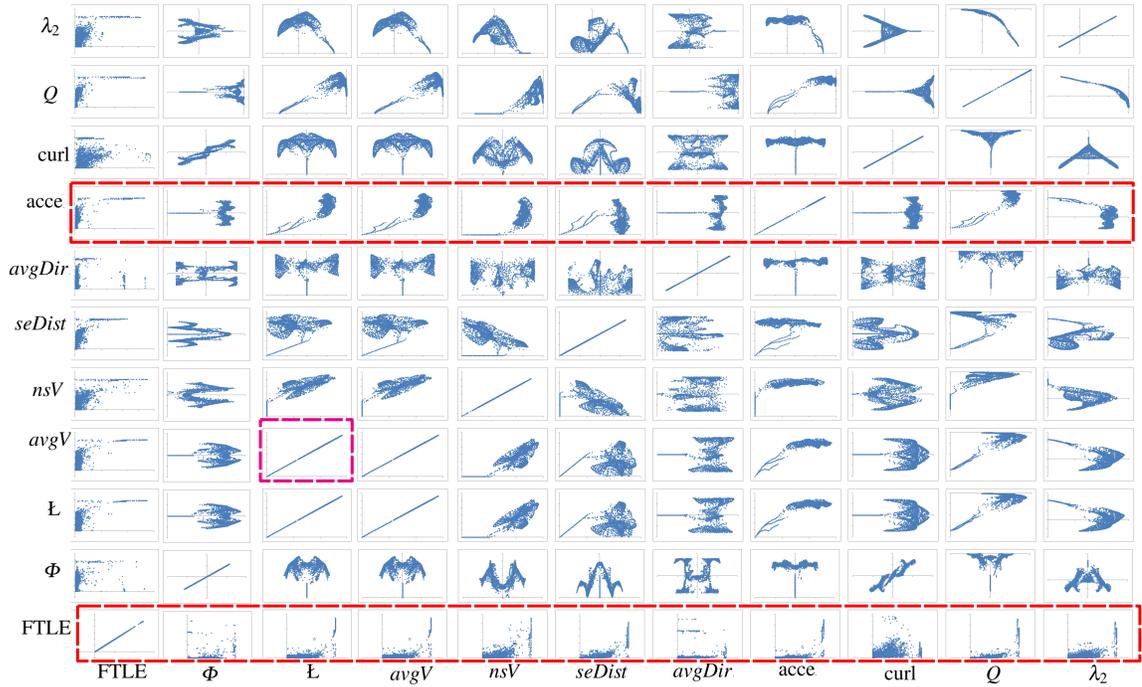


Figure 5.25: The scatter plot matrix of different \mathcal{A} fields of the Double Gyre flow. Note that the scatter plots associated with FTLE shows the correlation among the magnitude of the gradient of the individual attribute fields with the FTLE field. Note that this figure is a matrix not a real plot, therefore there are no axes or units. Due to the limitation of page size, this figure is split into four parts and put in the Appendix (Figure 8.1, Figure 8.2, Figure 8.3, and Figure 8.4).

5.4.1.1 Arc-length vs. Velocity Magnitude

It is not surprising that these two properties are directly related, as the arc-length of each segment of an integral curve is determined by the length of the vector value at the starting point of this segment scaled by the integration step size, i.e. scaled velocity magnitude. As shown in Figure 5.25 (entry highlighted by the purple box in the matrix), these two attributes show a strong linear relationship.

5.4.1.2 Acceleration Field vs. other Attribute Fields

The scatter plots of the acceleration field, which is computed by integrating the acceleration magnitude along a set of pathlines, and the remaining attribute fields (raw *acce*) generally display clear patterns. In particular, when the value of the *acce* field is small, the other attributes tend to be small. When the value of the *acce* field is increasing and becomes sufficiently large, the other attribute values tend to be large as well. This is consistent with the knowledge that the acceleration, a result of the net applied external force based on Newton Second Law, is the source of many different flow behaviors, such as flow separation and rotation. However, this relation is not true between *acce* and λ_2 or Q . That is, the smaller the *acce* value, the larger the the absolute values of λ_2 and Q . This in fact matches the result of the work [25] that utilizes the local minima of the acceleration field to detect vortex cores.

5.4.1.3 Winding Angle vs. Curl

Figure 5.26(a-c) shows a comparison of two \mathcal{A} fields computed by accumulating the change of the flow direction, i.e. winding angle (top) and curl (bottom) for some 2D flows, respectively. As can be seen, they exhibit almost identical patterns in the steady case (a-b). This is because curl quantifies the amount of rotation of the flow, i.e. twice the angular velocity in 2D, at a point in the flow domain, while the angle difference of the two vectors at two consecutive points along integral curve measures the amount of turning of this curve. If these two points are infinitely close, this angle change will tend to be the curl with the difference of a scale factor. Nonetheless, in general the curl-based \mathcal{A} fields

tend to be smoother than the winding angle based \mathcal{A} fields. This is because the curl at any given integration point is obtained via interpolation during the accumulation, while the angle difference between flow vectors is estimated via the angle change of the orientation of the two consecutive line segments of the integral curve, which is subject to numerical error. However, curl-based \mathcal{A} fields may not be able to capture some discontinuity of the geometric behaviors of the integral curves. As shown in Figure 5.26(c), the cusp-like behavior of pathlines (highlighted by the arrows) is not captured by the curl-based \mathcal{A} field. This is because this cusp-like behavior corresponds to a sharp angle (i.e. π) change which makes the flow directions before and after the cusp pointing to almost opposite directions, i.e. they are almost co-linear. Thus, the discrete curl computation that is performed while cross product computation will return zero or a very small value. Nonetheless, the relation between curl and the change of flow direction, as well as relation among other vortex identification criteria, such as λ_2 and Q , should be systematically studied as a potential path forward to address the problem of the current lack of a unified definition of vortices.

5.4.1.4 FTLE Approach vs. Accumulating Flow Vectors along Pathlines

In addition to accumulating the scalar quantities along the integral curves, vector-valued properties can also be accumulated. The resulting \mathcal{A} field is then a vector field. This vector-valued accumulation is used to study the relation of the FTLE computation and a derived scalar field computed from an \mathcal{A} field by accumulating the flow vectors scaled by the integration step size along integral curves. Assume a forward accumulation is considered, i.e. $t > 0$ in Eq.(4.1), the resulted vector is an orientation vector that points from the starting point to the end point of the integral curve [54] based on vector calculus, denoted

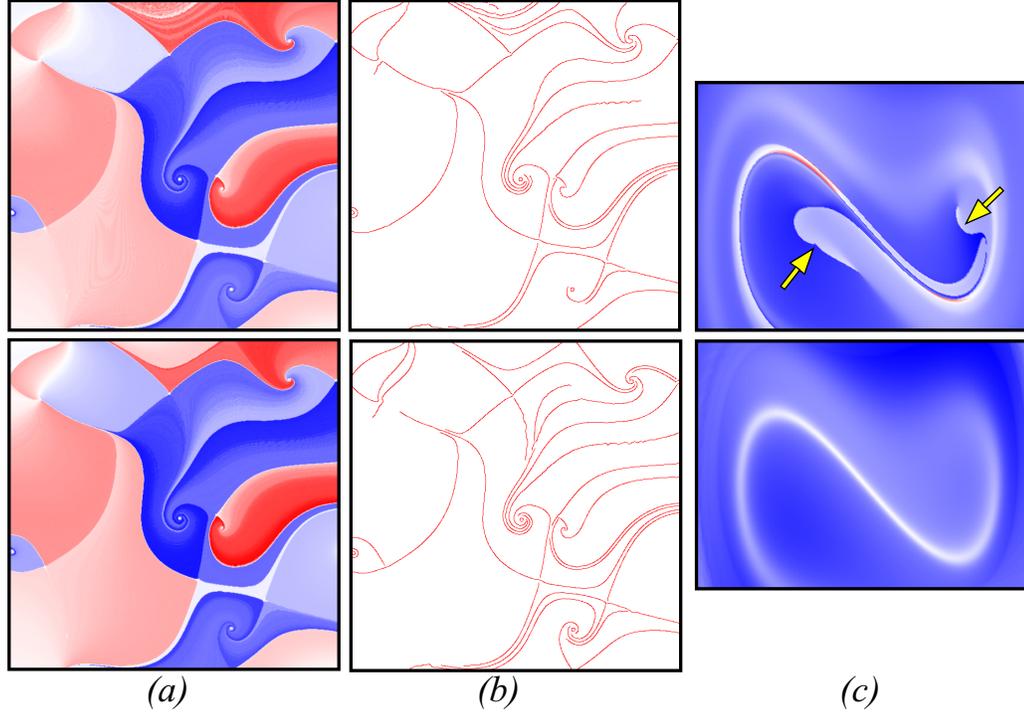


Figure 5.26: Comparison of the \mathcal{A} fields computed by accumulating the curl (bottom) and the change of the flow direction (top), i.e. winding angle, respectively. (a) shows the \mathcal{A} fields of a synthetic 2D steady flow. Their corresponding edges are in (b). (c) shows the \mathcal{A} fields of a 2D force duffing system.

by $V_{SE}(\mathbf{x}) = \varphi_{t_0}^{t_0+t}(\mathbf{x}) - \varphi_{t_0}^{t_0}(\mathbf{x})$ based on the notion of flow map [19]. This accumulated vector is saved to the corresponding seeding point of the integral curve, resulting in a vector-valued version of the \mathcal{A} field. It is not difficult to verify that

$$F = \frac{dV_{SE}(\mathbf{x})}{d\mathbf{x}} = \frac{d\varphi_{t_0}^{t_0+t}(\mathbf{x})}{d\mathbf{x}} - I_2 \quad (5.2)$$

where $\frac{dV_{SE}(\mathbf{x})}{d\mathbf{x}}$ denotes the gradient of the vector-valued \mathcal{A} field, $\frac{d\varphi_{t_0}^{t_0+t}(\mathbf{x})}{d\mathbf{x}}$ denotes the flow map deformation, and I_2 is an 2×2 identity matrix. Then $s_{t_0}^t(\mathbf{x}) = \frac{1}{t} \ln \sqrt{\lambda_{\max}(G)}$ is computed, where $G = F^T F$ —a Cauchy tensor and λ_{\max} is the maximum eigen-value of G . This gives rise to a scalar field that seems to have similar patterns to the corresponding FTLE field computing using the same time window according to Eq.(5.2). Figure 5.27 provides

the comparison of the original FTLE fields (top) and the derived scalar fields (bottom) from V_{SE} for a number of 2D unsteady flows. This indicates that the attribute that quantifies the difference from the starting point to the end point of an integral curve encodes the information of flow separation. Nonetheless, the accumulation of vectors using direct vector summation may lead to degeneracy. For instance, accumulating tangent vectors along a closed integral curve results in a zero vector. Therefore, a more appropriate accumulation may be to separate the accumulation of the direction and magnitude components, which requires further investigation.

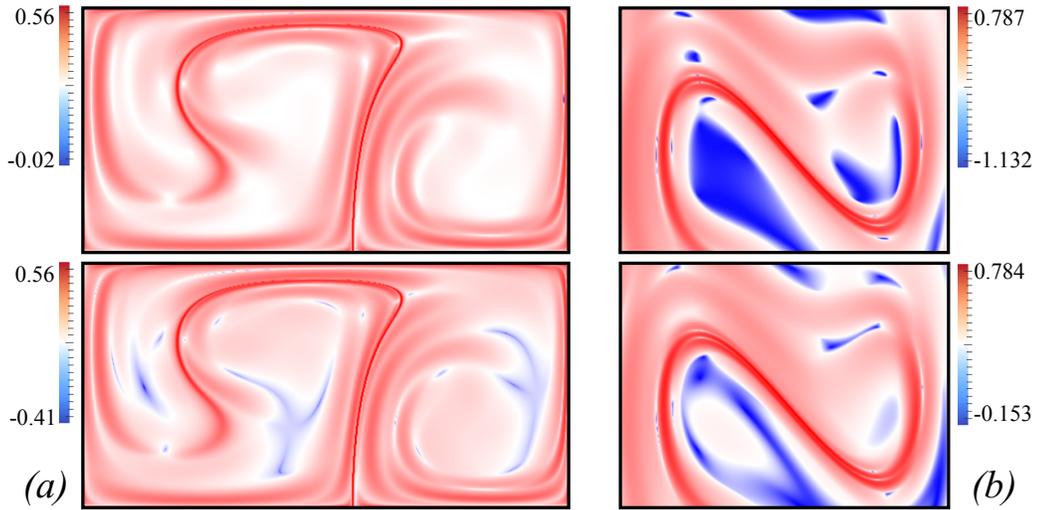


Figure 5.27: Comparison of the FTLE fields (top) and a derived fields (bottom) from the \mathcal{A} fields—vector fields defined by V_{SE} for the double gyre flow (a) and the force duffing system (b).

Similarly, one can use this above accumulation to verify the relation among other vector quantities, such as the difference vector between two consecutive flow vectors along integral curves and the acceleration of the flow. In addition, the Jacobian of the vector field—an asymmetric tensor [71], may be accumulated along the integral curves, which could provide additional insights into the general deformation of the flow particles along

their paths.

5.4.2 Relation of \mathcal{A} and $|\nabla\mathcal{A}|$ Fields to Flow Features

Steady flow features Many discontinuities (i.e. edges identified by the edge detector) of these attribute fields share characteristics with certain well-known flow features. For example, Figure 5.7 compares the discontinuity structure of the rotation field Φ of a synthetic steady flow to its topology (Figure 5.7(a)), which is illustrated via a set of integral curves that end or start from saddles, i.e. *separatrices*—a special type of streamline. The Φ field is not continuous across the separatrices if the accumulation is performed using an infinite time window. This is because an arbitrarily small perturbation in the direction other than the flow direction will result in another integral curve with a length much different from the separatrix, making the Φ field accumulated using Eq.(4.1) discontinuous at separatrices. With different parameters, different levels of detail of the discontinuity in the Φ field can be revealed (Figures 5.7(c)-(e)). Figure 5.7(f) shows the gradient of the Φ field, which does not provide a clean discontinuity structure.

LCS Lagrangian Coherent Structures (LCS) are defined as the ridges of the corresponding FTLE field. It indicates the regions of the domain with relatively large separation. Compared to LCS, it appears that the edges detected from all the attribute fields of the Double Gyre flow encode at least part of this information. This is also true for the other data sets that have been investigated. The discontinuity may be observed at the ridges of transport, i.e. LCS due to a similar reason to the separatrices in steady flow. A pathline seeded on the ridges may have behaviors different from its neighboring pathlines caused

by the separation, leading to the discontinuity in the attribute fields.

Cusp seeding curves The cusp seeding curve has been discussed in [66] to reduce self-intersecting pathlines in the pathline placement. These cusp seeding curves of the Double Gyre flow can be identified from the discontinuities in the rotation field Φ as shown in Figure 4.7(a). This cusp-like behavior in pathlines is caused by the abrupt change in the pathline direction, i.e. almost angle of π difference between the previous and current directions, which is in turn caused by the intersection of the pathlines with the paths of singularities.

Singularity path Singularity paths reveal the trajectories of fixed points in an unsteady flow. Among all the attribute fields studied, only the Φ field computed based on streaklines encodes such information. See Figure 4.7(c) for an example showing where the paths of the two vortices of the Double Gyre flow are revealed by the edges detected from the streakline-based Φ field. This is because singularity paths induce the cusp-like behavior in pathlines, also discussed in [66]. This cusp-like behavior corresponds to a large local angle change, which in turn leads to a large change, i.e. a discontinuity, in the Φ field. In addition, the temporal behavior illustrated by the translation of the singularities can only be captured by measuring the attributes of particles released at the same position at consecutive times, i.e. streaklines.

5.4.3 Which Characteristics to Accumulate?

Based on the existing results in the literature, it can be observed that if the goal is to study the transport behavior of the flow or the variation of the state of the particles along

their paths, then physical properties are typically selected [55]. On the other hand, for the integral curve dissimilarity computation, geometric characteristics are usually considered over physical properties [36]. However, this should not be treated as a general rule, as demonstrated by a recent work [18] in which the physical properties can also be used to define the distance between integral curves.

In addition, different local characteristics may be related by physical principles [42]. Nonetheless, given certain flow behaviors of interest, there could have more than one characteristic to measure it, and the \mathcal{A} fields that are computed from different characteristics may encode overlapping flow information. For the specific applications, selection of the appropriate characteristics deserves a detailed and comprehensive discussion as provided in [42], which is beyond the scope of this work.

5.4.4 Accumulation Window Size

The definition of \mathcal{A} is unfortunately sensitive to the specified accumulation window size, i.e. integration time/length. That means different \mathcal{A} s computed with different integration times/lengths may exhibit different patterns (i.e. different discontinuity structures). Figure 5.28 provides an example showing the \mathcal{A} fields based on the accumulation of the change of flow direction (aka. signed curvature) of a simple separation flow with different integration times/length. From the results, it can be figured out that with a smaller integration length (Figure 5.28 a), the \mathcal{A} field tends to capture the local and short-term flow behaviors. Interestingly, it captures places with large flow curvature. In contrast, a larger integration length may reveal the global and long-term flow behaviors (Figure 5.28

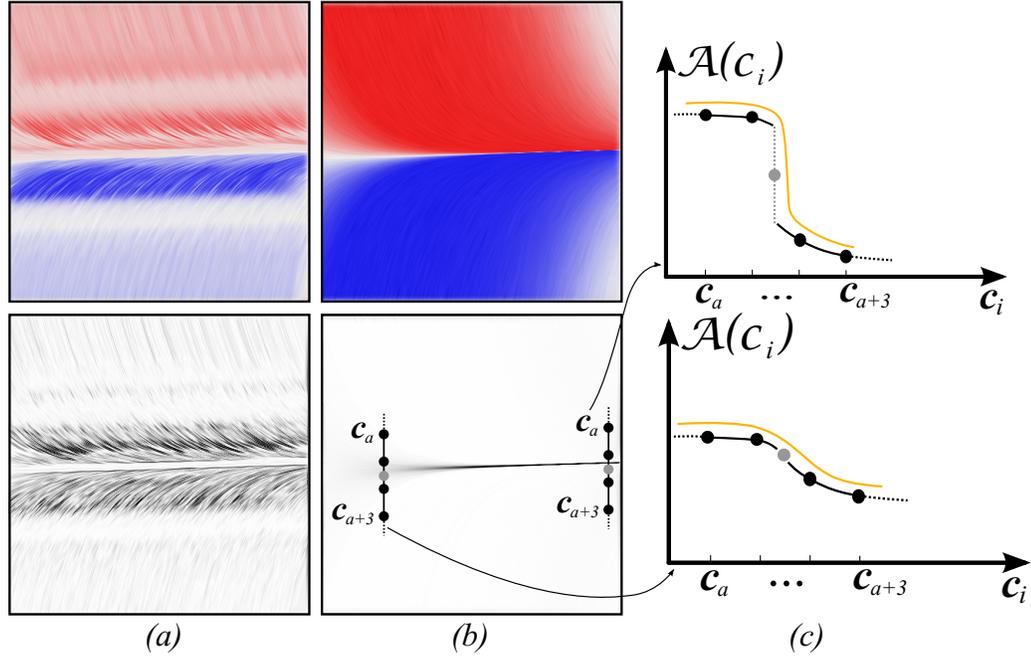


Figure 5.28: The influence of the accumulation window size. (a) shows the \mathcal{A} field computed with the integration length equal 10% of the size of the bounding box of the flow domain, while (b) shows the \mathcal{A} field with the length equal twice of the size of the bounding box. (c) shows the plots of the \mathcal{A} values sampled along two seeding line segments. As can be seen, even they have the same length, the two segments encode different amount of information quantified by the range of the \mathcal{A} values along the segments. This also demonstrates how the varying density of the integral curve may influence the encoding of certain flow features in the \mathcal{A} field.

b), and produce smoother \mathcal{A} fields at the same time. This effect is similar to the observation for the convolution process used by the texture-based techniques [35]. Figure 5.28 (c) shows the plots of the \mathcal{A} values along two line segments (shown in Figure 5.28 (b)). As can be seen, the ranges of the \mathcal{A} values on these two sampled segments are not identical. This again can be attributed to the sensitivity of the sampling location on the separation structure and the smeared effect of long integration. In practice, the selection of the integration times/lengths depends on the needs of the applications. If the local characteristic

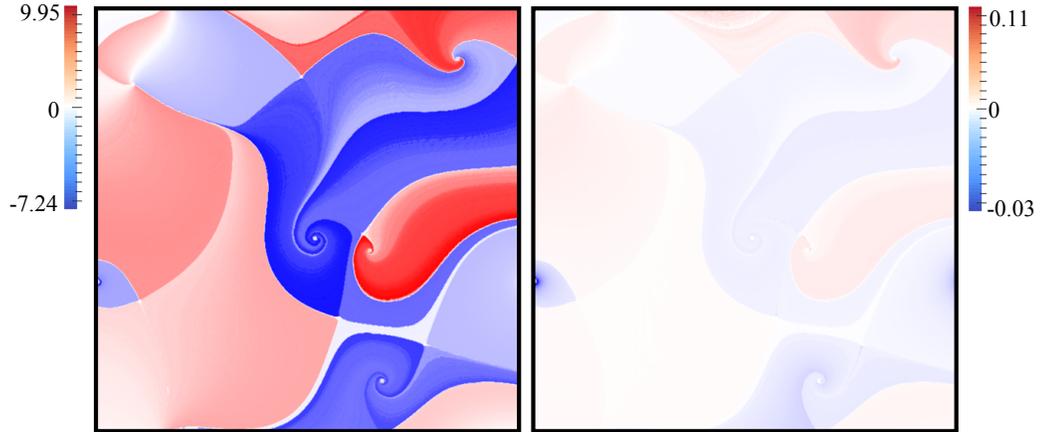


Figure 5.29: The \mathcal{A} fields computed without (left) and with average (right).

of the flow is of interest, a small integration time can be selected, while if the global and structure information of the flow is the focus, a long integration may be used. A similar consideration on the selection of integration time can be seen in the FTLE computation.

5.4.5 Average of the Accumulated Value

To avoid the possible artifacts introduced by the number of integration steps, especially when the integral curves are getting closer to fixed points, $A'_g(\mathbf{x}, t) = \frac{1}{t}A_g((\mathbf{x}, t_0), t)$ is also computed for unsteady flow and $A'_g(\mathbf{x}, s) = \frac{1}{s}A_g(\mathbf{x}, s)$ for steady flow, which essentially describes the average behavior of the particle along its path. The resulting \mathcal{A} fields with and without this average computation are compared. They in general have similar behaviors with the difference of some scalar factor (Figure 5.29). The benefit of using the average value is to enable us to inspect the overall attribute behavior along the integral curves. This can be useful when studying the behaviors of particles in unsteady flows. However, the differences between the \mathcal{A} values near the discontinuity tend to become smaller (Figure 5.29(right)), which may make the identification of these occurrences challenging.

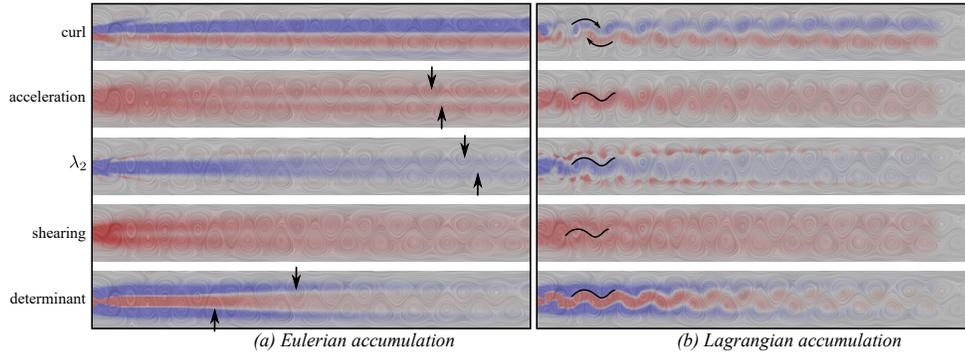


Figure 5.30: Comparison of Eulerian (a) and Lagrangian (b) accumulations using various attributes of the flow behind cylinder data. Note that the Eulerian accumulation highlights the places where the vortices sweep through, while the Lagrangian accumulation emphasizes the oscillating behaviors of the individual vortices.

Therefore, in most of the experiments, we use the non-average version of the \mathcal{A} fields.

5.4.6 Comparison with the Eulerian Accumulation

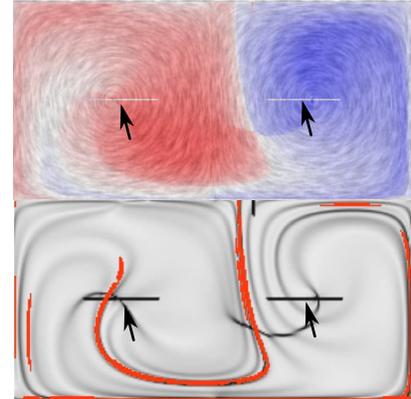
To some extent, the above Lagrangian accumulation framework allows us to inspect the aggregated (or overall) behaviors of particles during their advection (especially in the unsteady setting). In the meantime, the attribute values measured at the fixed locations but over time can be accumulated (or aggregated) to obtain the overall information of the flow at those locations. This scenario shares some similarity with the way of how different weather measurements are collected at those fixed stations. This accumulation is referred to as the *Eulerian accumulation*.

Figure 5.30 (a) shows the Eulerian accumulation results of a number of attributes for the 2D flow behind cylinder data. Most of these attributes are relevant to the vortical behaviors of the flow. As the vortex street pattern behind the cylinder in this flow is well known (which is also depicted by the texture image of the original flow minus the

ambient component), we can clearly observe that the obtain \mathcal{A} fields all highlight the regions where the vortices sweep through. In particular, the regions highlighted by the accumulation of acceleration magnitude, λ_2 and the determinant of the Jacobian clearly highlight the places that the vortex centers pass, which induce two tails in the later part of the domain (highlighted by the arrows). In contrast, the Lagrangian accumulation of the same attributes (Figure 5.30 (b)) does not provide this overall aggregated information of vortex regions but rather highlights the oscillatory behaviors of the individual vortices.

5.4.7 Extension to Non-integral Curves – Streak Lines

The accumulation framework for integral curves can be extended to other geometric curves derived from the vector fields, such as streak lines. A *streak line*, $\bar{s}(t)$, is the connection of the current positions of the particles, $\mathbf{p}_{t_i}(t)$, that are released from position p_0 at consecutive time t_i . Since the meaning of accumulating physical attributes along a streak line is yet to be clarified, this work concentrates on the local geometric characteristics, such as



the curvature or the change of the streak line direction. To reduce the memory overload, this work limited the number of particles released for each streak line to 200. This may affect the smoothness of streak lines depending on the time window used for the computation. To handle boundaries, the computation of a streak line is simply terminated once any of its particles hit a boundary. The inset shows the result for the Double Gyre flow. From

this result, it shows two edge segments in both the \mathcal{A} field (top) and the $|\nabla\mathcal{A}|$ field (bottom) (highlighted by the arrows). With a closer look, it can be figured out that these two edge segments correspond to the paths of the two oscillating centers. To further verify the conjecture, accumulation is performed along streak lines derived from a number of synthetic unsteady vector fields that possess various moving singularities. Figure 5.31 shows the results. Not surprisingly, the highlighted ridges in the $|\nabla\mathcal{A}|$ fields of these examples indeed correspond to the paths of the singularities.

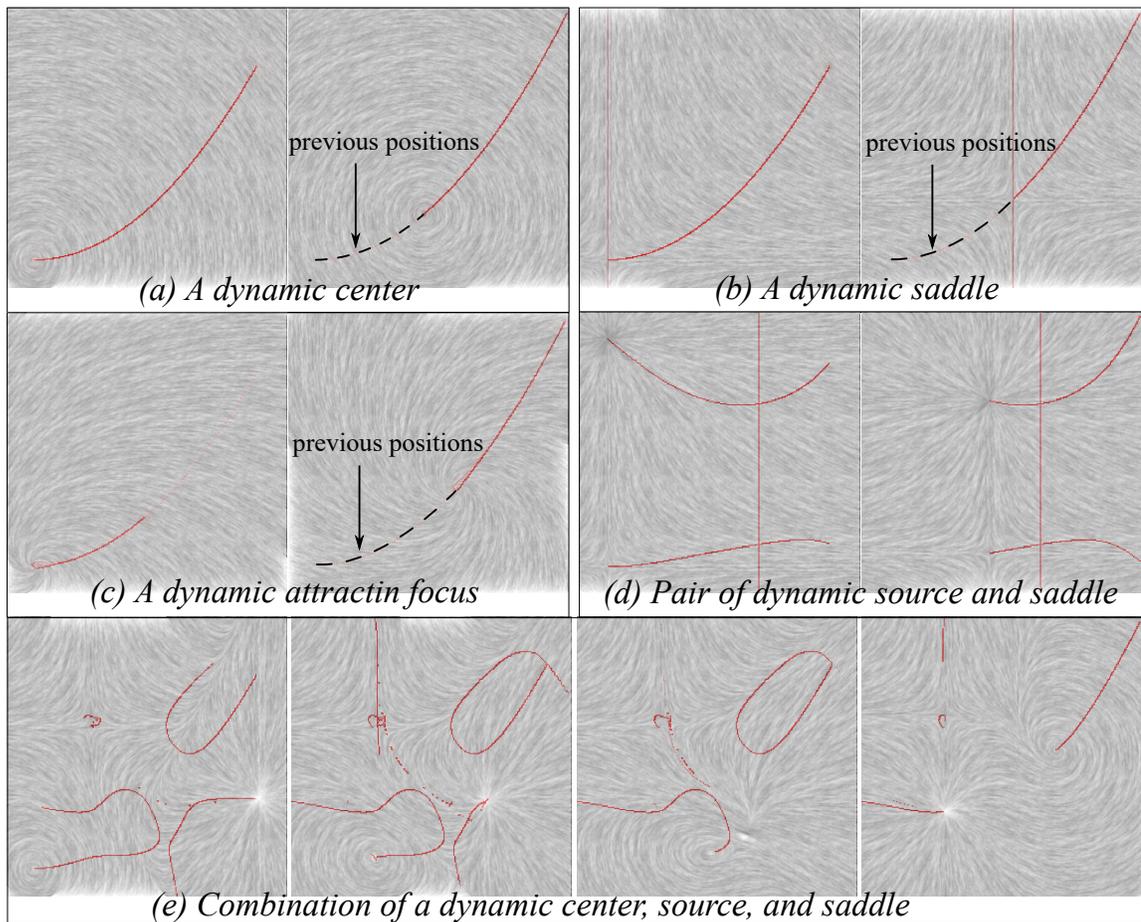


Figure 5.31: The $|\nabla\Phi|$ fields based on streaklines for a number of synthetic unsteady flows.

Why the \mathcal{A} field computed based on streak links reveal the singularity paths, while the one based on the pathlines cannot? To explain this, let us consider a pathline starting at position \mathbf{x}_0 at time t_i , which defines a flow map $\phi_i^t(\mathbf{x}_0)$. Once it moves away from \mathbf{x}_0 , information about what happens at \mathbf{x}_0 after t_i is not encoded in that pathline. In contrast, a streak line starting from \mathbf{x}_0 and perceived at time $t_j (> t_i)$ is a collection of particles that are released at \mathbf{x}_0 from t_i to t_j . Therefore, it naturally encodes the temporal variation of flow maps passing \mathbf{x}_0 after t_i . As showed before, the translation of the singularities will cause the sharp change in the direction of integral curves. This abrupt geometry change is captured by the accumulation along streak lines. Nonetheless, additional effort should be made to provide a more rigorous interpretation of the patterns revealed in the streak line based \mathcal{A} fields.

Chapter 6

A TAC-based Framework

The previous chapters have shown that to provide an overview of the general or average behavior of a particle in the flow, certain (physical or geometric) attributes of it are typically accumulated, which derives *fields*. However, this accumulated attribute information suppresses the detailed information of the flow that is of paramount importance in identifying local features in space and time, leading to incomplete representation of flow behaviors. To address this issue, a new framework for the visualization and exploration of unsteady flows is presented based on the time-dependent attributes measured at any given position and time. These time-dependent attributes can be represented as series of time activity curves (TAC) that are associated with either the individual spatial locations (Eulerian TAC) or the individual particles (Lagrangian TAC). In this chapter, the relations of the characteristics of these TACs with some well-known flow features is studied. Based on their characteristics, a hierarchical clustering framework for TACs is developed. In addition, a new distance metric is proposed to measure the similarity of segments of TACs

for the clustering. Our hierarchical clustering supports a level-of-detail representation of flow behavior in both space and time.

Lee et al. (2009) proposed a visualization framework to analyze time-varying data sets with a TAC-based distance field [30]. This field provides a visualization to highlight the position the features. However, there are still some limitations of the global attribute fields. (1) They cannot provide the details of an individual TAC, especially the occurring time and period of an interesting feature. In other words, the identification of an interesting behavior, i.e. the spacial location, the time information and duration, cannot be revealed. (2) They cannot study the contributions of the individual local events to the global behavior of a particle, i.e, the reason that causes the specific behavior.

To solve these problems, a novel framework is proposed to visualize and analyze time-dependent flows based on local physical properties. Specifically, the Time Activity Curves (TAC) of those local attribute is utilized to describe time-varying flow features of interests.

Our contributions are summarized as follows.

- An event detector for interesting flow behaviors is introduced based on local attribute TACs. The events are expressed as a combined information including the details, such as time and location of a given behavior.
- In order to provide different levels of details of the attributes, a TAC-based hierarchical clustering algorithm is presented. A comprehensive distance metric $EDTW$ is derived to describe the dissimilarity of TACs based on the detected events, which incorporates the global correlation of a pair-wise TACs, DTW distance of the clustering and temporal difference of events. A modified BIRCH-based clustering is

applied to resolve the scalability and accelerate the computation.

- The hierarchical clustering algorithm is applied to the time dimension to reveal the changes of flow behaviors through different time intervals in the whole time period. A statistic-based segmentation is proposed to generate a number of initial time intervals, and a hierarchical tree of the time intervals is built.
- A modified edge-bundling visualization of TAC clusters is proposed, which is able to represent the general behaviors of TACs in a cluster. In addition, the connection of clusters among different time intervals is well illustrated.
- The above algorithms have been integrated into an exploration system and which is applied to a number of 2D and 3D unsteady flows to reveal their respective behaviors of interest.

6.1 Time Activity Curve (TAC) Generation

This section describes how TACs are derived based on a given local attribute and provides a discussion on the potential benefits of TACs in revealing flow behaviors. An event detection algorithm applied to TACs is proposed, which is the basis of the distance metric in Section 6.2.1. In the following, I start with the introduction of TACs.

6.1.1 Definition of TAC

Given a specific local attribute \mathbf{A} , its value at a spatial position $\mathbf{x} \in \mathbb{M}$ at time $t \in \mathbb{T}$ can be denoted as $\mathbf{A}(\mathbf{x}, t)$. Computing $\mathbf{A}(\mathbf{x}, t)$ at a location \mathbf{x} over times gives us a time series of

the attribute values, referred to as a Time Activity Curve (TAC), which can be expressed as follows.

$$\Gamma_{\mathbf{A},\mathbf{x}}[i] = \mathbf{A}(\mathbf{x}, t_i) | i = 1, 2 \dots n \quad (6.1)$$

where $t_1, t_2 \dots t_n$ are time steps in the time window $T \subset \mathbb{T}$.

Since \mathbf{x} in Eq.6.1 is a fixed spatial position in the flow domain, the time series of \mathbf{A} is generated in an Eulerian way, which is called an Eulerian TAC. Similarly, the local attribute \mathbf{A} can also be computed along the pathline, \mathcal{C} , seeded at the particle released at \mathbf{x} at time t , which gives rise to a time series in the Lagrangian view, which can be expressed as follows.

$$\Gamma_{\mathbf{A},\mathcal{C}}[i] = \mathbf{A}(\mathcal{C}(\mathbf{x}, t_i), t_i) | i = 1, 2 \dots n \quad (6.2)$$

where $\mathcal{C}(\mathbf{x}, t_i)$ is the location of \mathbf{x} on the pathline, \mathcal{C} , at time t_i , and $t_i \in T \subset \mathbb{T}$.

In general, a Time Activity Curve is defined as a time serial data $\Gamma = \{\Gamma[i] | i = 1, 2 \dots n\}$ where $\Gamma[i]$ is the local attribute value at time t_i . Figure 6.1(a) is an example TAC where x axis indicates time series and y axis shows the local attribute values at the corresponding times. The length of a TAC is the number of elements in the TAC, indicating the lifespan of the time series data.

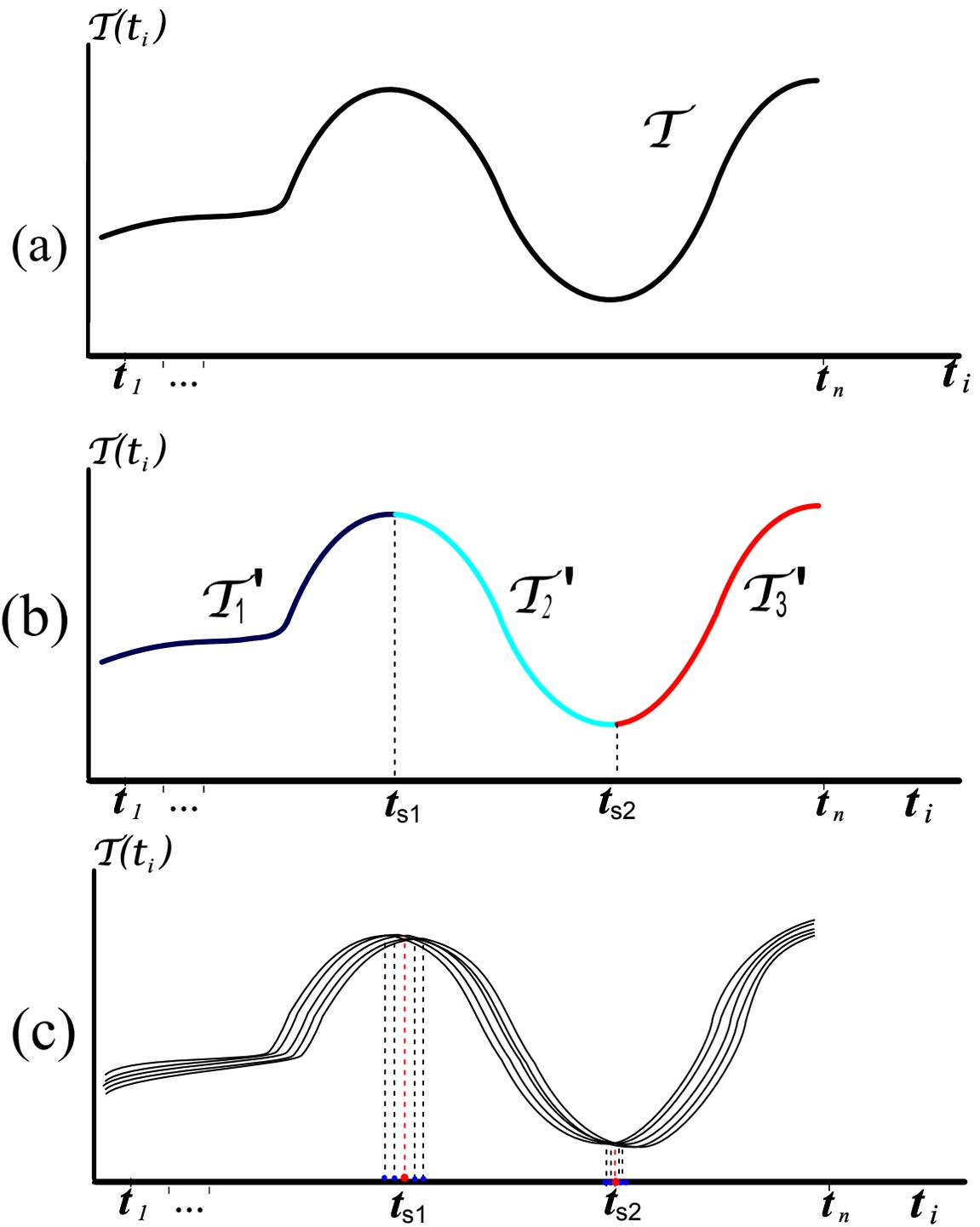


Figure 6.1: (a) An example TAC. (b) Event of the TAC in (a). (c) An example of time interval segments of TACs.

6.1.2 TAC Analysis

One benefit of TACs is that their geometric characteristics show the trend of the corresponding local attribute. Figure 6.2(a) shows several Eulerian TACs sampled in a 2D unsteady flow behind a square cylinder. The attribute used to construct these TACs is *curl*. The seeding positions are highlighted with the colored dots (Figure 6.2(a), left). Their corresponding TAC curves are shown in the TAC plot (Figure 6.2(a), right) with the corresponding colors. While the periodicity of the TACs in Figure 6.2(a) indicates the left-to-right transportation behavior of the flow, the peak-to-peak amplitude of a TAC reflects the distance between the seeding position of the TAC and the vortex center. That is, the larger the peak magnitude, the closer its seeding position to the vortex center (see the red dot and its corresponding TAC). This observation is consistent with the fact that the changes of the attribute *curl* are smaller at the locations that far away from the vortex center.

In contrast to Eulerian TACs that show the trends of the attribute at a given seeding position in a given time range, Lagrangian TACs are capable of collecting the attribute information along pathlines, indicating the transport of the local attributes carried by particles. Figure 6.2(b) are two Lagrangian TACs of the attribute Q seeded at the vortex center and on the vortex boundary, respectively. The TAC associated with the particle seeded at (or near) the vortex center (the red dot) is mostly flat (see the TAC plot on the right), indicating that the vortex center has a rather stable behavior.

The above examples show that compared to integral curves that only convey the geometric characteristics of the flow, Eulerian and Lagrangian TACs can reveal not only the

geometric but also the physical characteristics of the flow with the appropriate attributes. This indicates that analyzing TACs may allow us to develop an effective exploration framework to reveal more fruitful flow behaviors than only inspecting the integral curves.

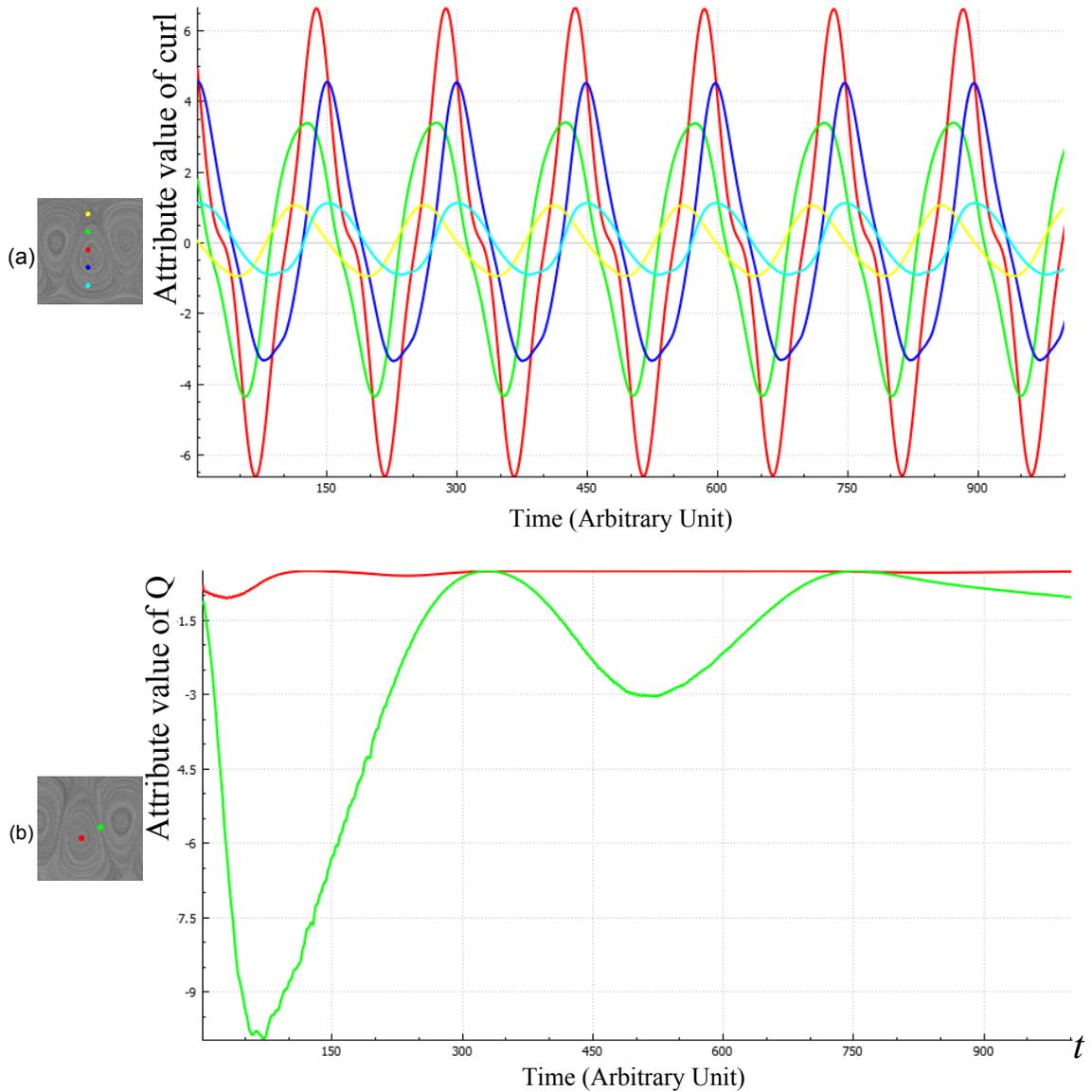


Figure 6.2: (a) Eularian TAC samples of attribute $curl$. (b) Lagrangian TAC samples of attribute Q . X axis is the time and Y axis stands for the attribute values. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them.

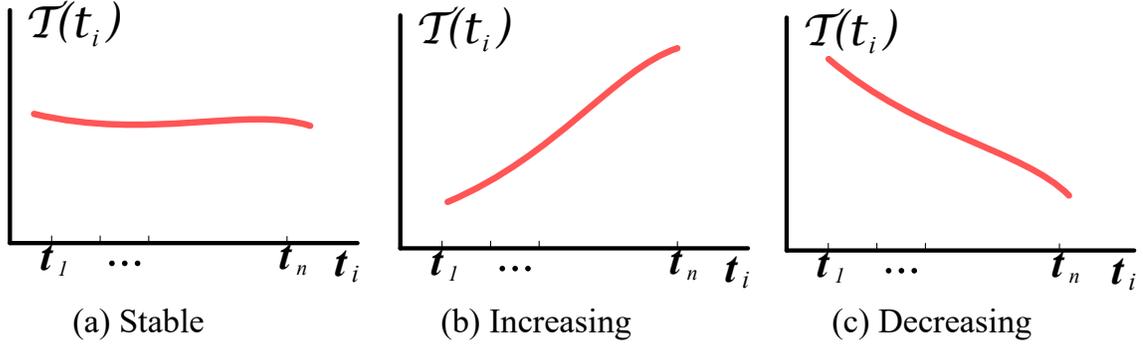


Figure 6.3: Different types of primitive trends.

6.1.3 TAC-based Event Definition

In order to describe the behavior of a TAC, it needs to identify one or multiple interesting temporal trends of the local attribute either at a fixed location (Eularian TAC) or along an integral curve (Lagrangian TAC). The detected temporal trends are used to determine the similarity (or dissimilarity) between TACs. Given a TAC Γ , its *event* is defined as the segmentation of the TAC that contains a sequence of sub-TACs, $\Gamma = \langle \Gamma'_1, \Gamma'_2, \dots, \Gamma'_z \rangle$ where Γ'_i is a sub-TAC, i.e. a continuous subset of Γ , so that:

- (a) The union of the sub-TACs $\Gamma'_i \in \Gamma, i = 1, 2, \dots, z$ forms the initial TAC Γ , i.e. $\bigcup_{i=1}^z \Gamma'_i = \Gamma$.
- (b) $F_i \cap F_j = \emptyset (i \neq j)$
- (c) Each sub-TAC consists of only one primitive trend, which can be either stable, increasing or decreasing, as illustrated by Figure 6.3. In other words, no $\Gamma'_i \in \Gamma$ contains combined trends.
- (d) Each pair of two neighboring sub-TACs have different primitive trends.

In our implementation, given an individual TAC $\Gamma = \Gamma[1 \dots n]$, the 1D Morse decomposition [5] is applied to generate a number of temporal sequences of TAC segments, i.e. sub-TACs. Assume there are z primitive trends in Γ , corresponding to $z - 1$ splitting points $s_1, s_2 \dots s_{z-1}$, $s_i \in [1, n]$. The event of Γ , i.e. the sequence sub-TACs can be extracted as follows.

$$\Gamma'_i = \begin{cases} \Gamma[1 \dots s_i - 1] & i = 1 \\ \Gamma[s_{i-1} \dots s_i - 1] & 1 < i < z \\ \Gamma[s_{i-1} \dots n] & i = z \end{cases} \quad (6.3)$$

Figure 6.1(b) shows an example of event defined on the TAC in Figure 6.1(a). There are three primitive trends (in the order of increasing, decreasing and increasing) in Γ , so the event consists of three sub-TACs, i.e. $\Gamma = \langle \Gamma'_1, \Gamma'_2, \Gamma'_3 \rangle$ corresponding to splitting positions s_1 and s_2 . Note that the correspondent splitting times are t_{s_1} and t_{s_2} , respectively.

6.2 Methodology

Given the above concepts of TACs, a new framework is proposed to use TACs to perform clustering to aid the subsequent TAC-based flow exploration. In particular, given sequences of TACs that are computed based on the densely sampled seeds in the flow domain, a hierarchical clustering based on the characteristics of the entire TACs is performed to characterize their similarity in both the attribute and physical spaces. Second, based on the clustering result obtained in the first step, a hierarchical temporal clustering of TACs

is performed to provide a level-of-detail characterization of their temporal behaviors (i.e. split or merge over time). To assist these two-stage clustering of TACs, a new dissimilarity metrics for TACs is first discussed.

6.2.1 TAC-based Dissimilarity Metrics

Given two TACs with equal length, $\Gamma_1 = \Gamma_1[1 \dots n]$ and $\Gamma_2 = \Gamma_2[1 \dots n]$, their distance can be measured using either

Euclidean distance:

$$D_e(\Gamma_1, \Gamma_2) = \sqrt{\sum_{i=1}^n (\Gamma_1[i] - \Gamma_2[i])^2} \quad (6.4)$$

or Pearson correlation:

$$D_p(\Gamma_1, \Gamma_2) = \frac{cov(\Gamma_1, \Gamma_2)}{\sigma_{\Gamma_1} \sigma_{\Gamma_2}} \quad (6.5)$$

where cov is the covariance and σ_{Γ} is the standard deviation of Γ .

Although both Euclidean distance and Pearson correlation measure are easy to compute, they have their own drawbacks as TAC-based dissimilarity metrics. Euclidean distance is not able to reveal the transformation of TACs. As shown by an example in Figure 6.4(a), Γ_2 has a flipped trend of Γ_{base} and Γ_1 has the same trend as Γ_{base} , but their Euclidean distance to Γ_{base} is the same. Pearson correlation is a measure of the linear correlation between two TACs, which has a value between -1 and 1. It can reveal the trend difference between two TACs. $D_p(\Gamma_1, \Gamma_2) > 0$ indicates Γ_1 has a similar trend to Γ_2 ($D_p(\Gamma_1, \Gamma_2) = 1$ means they have exactly the same trend). $D_p(\Gamma_1, \Gamma_2) < 0$ indicates Γ_1

has a different trend than Γ_2 . ($D_P(\Gamma_1, \Gamma_2) = -1$ means they have flipped trends). Pearson correlation cannot measure the dissimilarity of two TACs when they have the same trend (i.e. shifting in TACs). Figure 6.4(b) is such an example. Γ_1 and Γ_2 have the same trend as Γ_{base} , but the event in Γ_2 happens latter than that in Γ_1 . In addition, both Euclidean distance and Pearson correlation measure require the TACs to have the same length. This may not be satisfied when the events of TACs have different durations. For example, a pathline stopping at the boundary of flow domain will lead to the early termination of the corresponding Lagrangian TAC.

To address the above issues of the Euclidean distance and Pearson correlation measure, another distance metric, Dynamic Time Warping (DTW), has been employed to consider both the shift and deformation of the time series [30]. DTW is a dynamic programming algorithm that aligns two time series with the smallest distortion. With DTW, TACs of similar shapes with different temporal shifts and time spans can be classified into the same group. In addition, DTW applies to TACs with different length because a time warping is considered. DTW is defined and computed as $DTW(\Gamma_1, \Gamma_2) = M[n, n]$, where M can be derived from a recursive equation:

$$M[i, j] = |\Gamma_1[i] - \Gamma_2[j]| + \begin{cases} 0 & i = 1 \& j = 1 \\ M[i-1, j] & i > 1 \& j = 1 \\ M[i, j-1] & i = 1 \& j > 1 \\ N[i, j] & i > 1 \& j > 1 \end{cases} \quad (6.6)$$

where $N[i, j] = \min(M[i-1, j], M[i, j-1], M[i-1, j-1])$, $i = 1 \dots n$, $j = 1 \dots n$.

DTW metric focuses on the shape of the TACs, considering the shift and deformation of the time series data. However, only the sequence of the sampled positions is taken into account, while the difference in temporal dimension is ignored. For example, in $DTW(\Gamma_1, \Gamma_2)$ computation, the first elements in Γ_1 and Γ_2 are aligned, no matter Γ_1 and Γ_2 start at the same time or not. Obviously, leaving the absolute temporal difference out of the definition of the distance functions omits important information regarding the temporal distribution of relationships between TACs. In Figure 6.4(a), Γ_1 and Γ_2 have similar DTW distance to Γ_{base} since Γ_1 can be deformed to Γ_2 by a flipping operation. Obviously, Γ_1 should be grouped to Γ_{base} rather than Γ_2 considering that they have similar trends. In Figure 6.4(b), according to the definition of DTW, Γ_1 and Γ_2 have the same DTW distance to the TAC Γ_{base} because the different temporal shifts do not influence the DTW computation after the dynamic warping. In fact, compared to Γ_1 , Γ_2 is more different from Γ_{base} because the main feature in Γ_2 happens much later than that in Γ_1 . In addition, since DTW considers the deformation of TACs, the trend information is not reflected by DTW distance metric.

Since all the previous methods fail to measure the similarity of two TACs properly and more importantly, to capture the similarity of events decomposed from the two TACs, a new distance metric, called Event-based Dynamic Time Warping (EDTW) to calculate the dissimilarity of TACs based on the events of TACs, is introduced. As discussed in Section 6.1.3, the events of a given TAC are a sequence of sub-TACs that reveal trend information in different time intervals.

The distance between Γ_1 and Γ_2 is defined as the dissimilarity of the corresponding events detected from Γ_1 and Γ_2 , $\Gamma_1 = \langle \Gamma_{1'_1}, \Gamma_{1'_2}, \dots, \Gamma_{1'_k} \rangle$ and $\Gamma_2 = \langle \Gamma_{2'_1}, \Gamma_{2'_2}, \dots, \Gamma_{2'_k} \rangle$. The

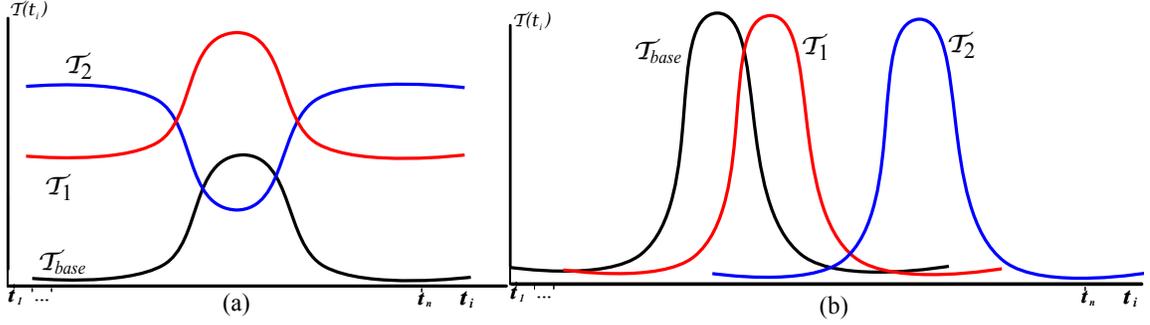


Figure 6.4: Two examples of TACs that cannot be accurately measured by Euclidean distance, Pearson correlation or DTW. (a) $D_e(\Gamma_{base}, \Gamma_1) = D_e(\Gamma_{base}, \Gamma_2)$, $DTW(\Gamma_{base}, \Gamma_1) = DTW(\Gamma_{base}, \Gamma_2)$, Γ_2 has different trend from Γ_{base} . (b) $D_p(\Gamma_{base}, \Gamma_1) = D_p(\Gamma_{base}, \Gamma_2)$, $DTW(\Gamma_{base}, \Gamma_1) = DTW(\Gamma_{base}, \Gamma_2)$, Γ_2 is farther away from Γ_{base} w.r.t temporal difference.

Event-based Dynamic Time Warping (EDTW) distance is defined as follows:

$$EDTW(\Gamma_1, \Gamma_2) = (1 + P_c \times Dist_{corr}(\Gamma_1, \Gamma_2)) \times \sum_{i=1}^k (DTW(\Gamma'_{1i}, \Gamma'_{2i}) \times (1 + P_t \times Dist_{time}(\Gamma'_{1i}, \Gamma'_{2i}))) \quad (6.7)$$

$$Dist_{time}(\Gamma'_{1i}, \Gamma'_{2i}) = 1 - 2 \frac{\|\Gamma'_{1i} \cap \Gamma'_{2i}\|}{\|\Gamma'_{1i}\| + \|\Gamma'_{2i}\|} \quad (6.8)$$

$$Dist_{corr}(\Gamma_1, \Gamma_2) = 0.5 - \frac{cov(\Gamma_1, \Gamma_2)}{2\sigma_{\Gamma_1} \sigma_{\Gamma_2}} \quad (6.9)$$

In the above definition, $DTW(\Gamma'_{1i}, \Gamma'_{2i})$ measures the DTW distance of sub-TACs Γ'_{1i} and Γ'_{2i} . $Dist_{time}(\Gamma'_{1i}, \Gamma'_{2i})$ quantifies the difference of sub-TACs Γ'_{1i} and Γ'_{2i} in temporal

dimension. If Γ_{1i}' and Γ_{2i}' have the same duration of the temporal period, $Dist_{time}(\Gamma_{1i}', \Gamma_{2i}')$ is 0. While if they have completely different lifespan, i.e. the time ranges have no intersection, $Dist_{time}(\Gamma_{1i}', \Gamma_{2i}')$ is 1. $Dist_{time}(\Gamma_{1i}', \Gamma_{2i}')$ is introduced to resolve the issue illustrated in Figure 6.4(a). $Dist_{corr}(\Gamma_1, \Gamma_2)$ measures the global correlation between TACs Γ_1 and Γ_2 . $Dist_{corr}(\Gamma_1, \Gamma_2) = 1$ means Γ_1 and Γ_2 have opposite trends while $Dist_{corr}(\Gamma_1, \Gamma_2) = 0$ means they have the same trend. $Dist_{corr}(\Gamma_1, \Gamma_2)$ is incorporated into *EDTW* distance measure to resolve the issue illustrated in Figure 6.4(b). $Dist_{time}(\Gamma_{1i}', \Gamma_{2i}')$ and $Dist_{corr}(\Gamma_1, \Gamma_2)$ participate to *EDTW* distance measure by two penalty factors P_t and P_c which represent user-specified importance to temporal difference and global correlation, respectively.

There are two potential issues in the above *EDTW* computation. One is it requires that the events of Γ_1 and Γ_2 must consist of the same number of sub-TACs. To address the case where Γ_1 and Γ_2 have different numbers of events, the following adjustment is made. Assume $\Gamma_1 = \langle \Gamma_{11}', \Gamma_{12}', \dots, \Gamma_{1z}' \rangle$ and $\Gamma_2 = \langle \Gamma_{21}', \Gamma_{22}', \dots, \Gamma_{2k}' \rangle$, $z > k$. The $z - k$ iterations are applied to Γ_1' . In each iteration the smallest sub-TAC Γ_{1i}' is merged into its neighboring sub-TAC Γ_{1i}' or Γ_{1i+1}' depending on which one is smaller. With this strategy the larger sub-TACs that well represent the event are kept as original while the smaller ones are merged together until the two TACs have the same number of sub-TACs. The other potential issue is that Γ_1 and Γ_2 may have different length, which contradicts the requirement of the computation of $cov(\Gamma_1, \Gamma_2)$ in $Dist_{corr}(\Gamma_1, \Gamma_2)$. In this case, interpolation is applied to the smaller TACs so that they have equal length. The interpolation operation only affects the sample rate on a TAC, so the trend of the TAC will not be modified, i.e. making the computation of the correlation between two TACs with different length possible.

6.2.2 TAC-based Clustering Algorithm

In order to provide different levels of details of flow behaviors w.r.t. the local attributes, based on the above new distance metric, we perform the clustering of TACs using their entire temporal samples. Specifically, the agglomerative hierarchical clustering (AHC) is employed, which is a popular clustering algorithm that sequentially combines smaller clusters into larger ones until one cluster that represents all the data is left. To resolve the scalability and complexity of AHC, a modified BIRCH-based strategy in the context of TACs is adopted.

6.2.2.1 Agglomerative Hierarchical Clustering (AHC)

AHC is one of the most commonly used clustering methods due to its applicability to many dissimilarity metrics types such as Euclidean distance, Pearson correlation or DTW. Given n TACs $\Gamma_1, \Gamma_2, \dots, \Gamma_n$, AHC builds a hierarchy of clusters, i.e. an unbalanced binary tree, from which different numbers of clusters can be derived. First, each TAC is treated as a separate initial cluster (leaf node), and the hierarchy is then built by repeatedly merging pairs of similar clusters until all TACs are contained in a single cluster (root node). For the given n TACs, $n - 1$ merge iterations are carried out. In each iteration, it needs to decide which two clusters are merged together. Therefore, a distance metric and a so-called linkage type needs to be specified.

The distance metric determines the similarity between individual TACs and can be generally specified through a distance matrix $D \in \mathbb{R}^{(n \times n)}$, where each entry D_{ij} stores the dissimilarity, i.e. the distance, between the pair of Γ_i and Γ_j . Throughout this work, the

Event-based Dynamic Time Warping (EDTW) distance between TACs discussed in Section 6.2.1 is used as the distance metric. The entry D_{ij} is computed as $D_{ij} = EDTW(\Gamma_i, \Gamma_j)$.

The linkage type, on the other hand, specifies how distances between two clusters, each of which consists of a number of TACs, are determined from D_{ij} . According to the specified linkage type, in every merge iteration the pair of clusters with the smallest distance can be merged. In this work, given two clusters C_p and C_q , a complete linkage is used, which is defined as follows:

$$Dist(C_p, C_q) = \max(D_{ij} | \forall \Gamma_i \in C_p, \forall \Gamma_j \in C_q) \quad (6.10)$$

Note that several other linkage types, such as average linkage, single linkage, average linkage and Ward's linkage [63], can also be applied here. The discussion of linkage selection is beyond the scope of this work, interested readers can refer to the overview article by Jain (2010) [23] for details.

The agglomerative hierarchical clustering algorithm is implemented as follows:

- Step 1: Compute TACs at the samples in the flow domain, which are the initial clusters.
- Step 2: Compute the distance matrix $D_{(n \times n)}$ of the initial clusters.
- Step 3: Find the two clusters with smallest cluster distance according to the linkage type, merge them into a new cluster in a higher level and update the distance matrix.
- Step 4: Repeat step 3 until one cluster is left.

6.2.2.2 A BIRCH-based Clustering

The classic AHC algorithm does not scale well in terms of memory overhead and computation time with the increase of the TACs (and their lifespans). According to AHC algorithm, all the TACs are equally important for the clustering purpose, and in each clustering decision (i.e. selecting the closest pair of clusters), all the TACs and currently existing clusters are inspected equally. However, not all the TACs are equally important. In other words, those TACs that are close or similar enough can be considered collectively instead of individually.

To address that, the classic AHC algorithm is incorporated to the BIRCH clustering [74]. BIRCH is designed for clustering a large amount of numerical data by the integration of hierarchical clustering at the initial stage, which is based on two main structures: clustering feature (CF) and clustering feature tree (CF-tree). A clustering feature is a triplet summarizing information that represents a cluster, including the number of data points in the cluster, the linear sum and the square sum of those data points. A CF-tree is a height-balanced tree that stores the clustering features for a hierarchical clustering. CF-tree is a very compact representation of the dataset because each entry in a leaf node is not a single data point but a sub-cluster, which absorbs as many data points as the specific threshold value allows [74].

However, BIRCH method cannot be applied to TAC-based clustering directly because the above clustering feature cannot well represent a cluster of TACs. A clustering feature is essentially a summary of the statistics information for the given cluster using linear sum and square sum of data points. The TAC generated by applying linear sum or square sum

to a group of TACs cannot represent this group because different trends of TACs may not exist after the sum operation. For example, the linear sum of an increasing TAC and a decreasing TAC may give rise to a flat TAC. Therefore, instead of building up a CF-tree based on the above clustering features, BIRCH strategy is adapted to the TAC-based AHC. Specifically, in the first step of AHC discussed in Section 6.2.2.1, instead of treating each TAC as an initial cluster, i.e. a leaf node, the leaf node to store a sub-cluster of TACs enlarged. Given a new TAC, if its distance to any leaf node, i.e. the *EDTW* distance between the new TAC and the TAC that creates the leaf node, is less than a user-specified threshold, the new TAC will be absorbed into that leaf node. Otherwise a new leaf node will be created by the new TAC. The rest steps are the same as AHC algorithm. With the aid of BIRCH-based clustering, the TAC-based clustering algorithm can handle large numbers of input TACs with long lifespans.

6.2.2.3 Clustering Result Generation

Once the hierarchical tree is created, given a cluster number K , the top K clusters on the tree is output. According to the AHC, the higher level in the tree, the larger distance is among the TACs in the corresponding cluster on that level. Therefore, another way to select a number of clusters is to input a distance threshold ε . Given any ε , a binary search on the distance values on each level of the tree is conducted, then the tree at the target level is cut to generate the output clusters. In practice, users may not have the prior knowledge about the distance range, so a normalized distance range is provided.

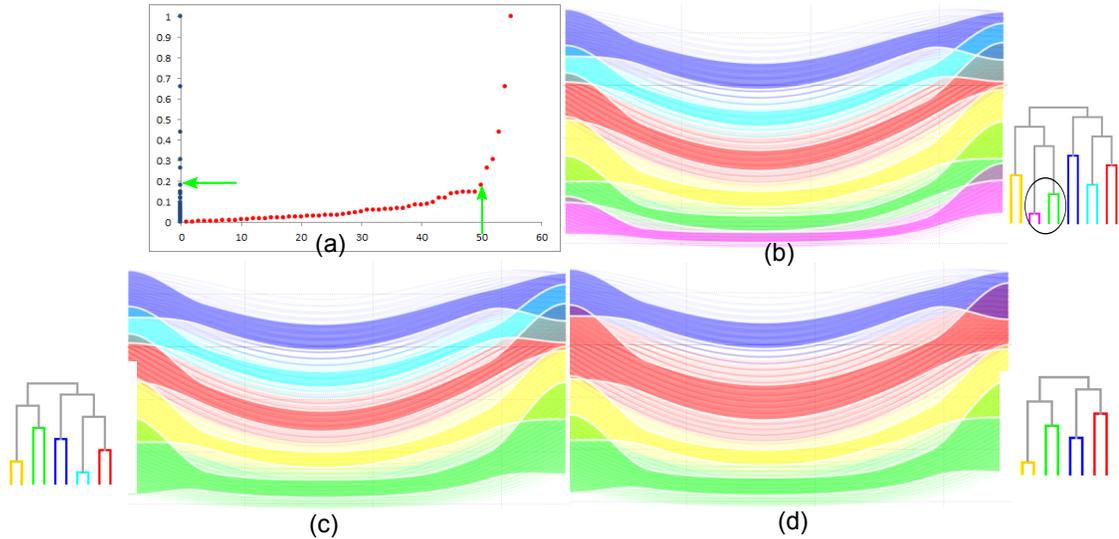


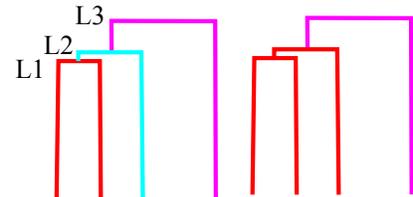
Figure 6.5: Illustration of clustering result generation. (a) distribution of distance values at each level of the tree. (b)-(d) clustering results with $K = 6, K = 5, K = 4$, respectively. The corresponding distance thresholds are $\varepsilon = 26\%, \varepsilon = 30\%$ and $\varepsilon = 43\%$, respectively. X axis is the time and Y axis stands for the attribute value. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them.

However, choosing an appropriate K or distance threshold ε to generate a desired level of TAC clusters may require the user to carry out a few trial-and-errors.

To accelerate this process, a candidate cut is provided,

which corresponds to certain K and its distance threshold ε without user specification. Recall that during the AHC clustering process, in each merging iteration, two clusters with smallest cluster distance are selected and merged into a new cluster in a higher level. In other words, there is a distance value corresponding to each level of the tree. As shown in the inset, if ε is set as the distance at level $L1$, three clusters are generated (left figure).

The ε at level $L2$ gives rise to two clusters (right figure). In the implementation, the height



of each level in the tree is mapped to a distance value. The heights of level $L1$ and level $L2$ are close to each other, which indicates their distance values are similar. The distribution of the distance values on each level is then analyzed to select a cut in the tree at the level where the distance value has a large increase (see Figure 6.5a for an example with more details provided next). This idea is based on the observation that changing the cut level where the distance values are densely distributed (close to each other) does not change the clustering result much. As shown in the inset, cutting the tree at level $L2$ (right figure) is preferred as the candidate cut compared with the one on the left based on the height information.

An example of automatic selection of clusters for visualization is shown in Figure 6.5. In Figure 6.5(a), the red dots show the changes of distance values (y axis) of each merging operation (x axis) and the blue dots show the distribution. The level highlighted by the green arrow is selected as the candidate cut, because before that the distance values are densely distributed, and starting from this level, the distance value shows a large increase. In this case, this candidate cut gives rise to $K = 6$ and $\varepsilon = 26\%$. Those six clusters are shown in Figure 6.5(b). The visualization of the clustering results is discussed in the following Section 6.2.2.4.

6.2.2.4 Visualization of TAC-based Clusters

Traditionally, visualizing clusters is achieved by assigning each cluster a specific color. In the proposed TAC-based clustering, the members in a cluster are individual TACs. Showing all TACs with colors assigned based on their cluster IDs will easily result in clutter, making it difficult to recognize the behaviors of the TACs. Also, TACs are often plotted

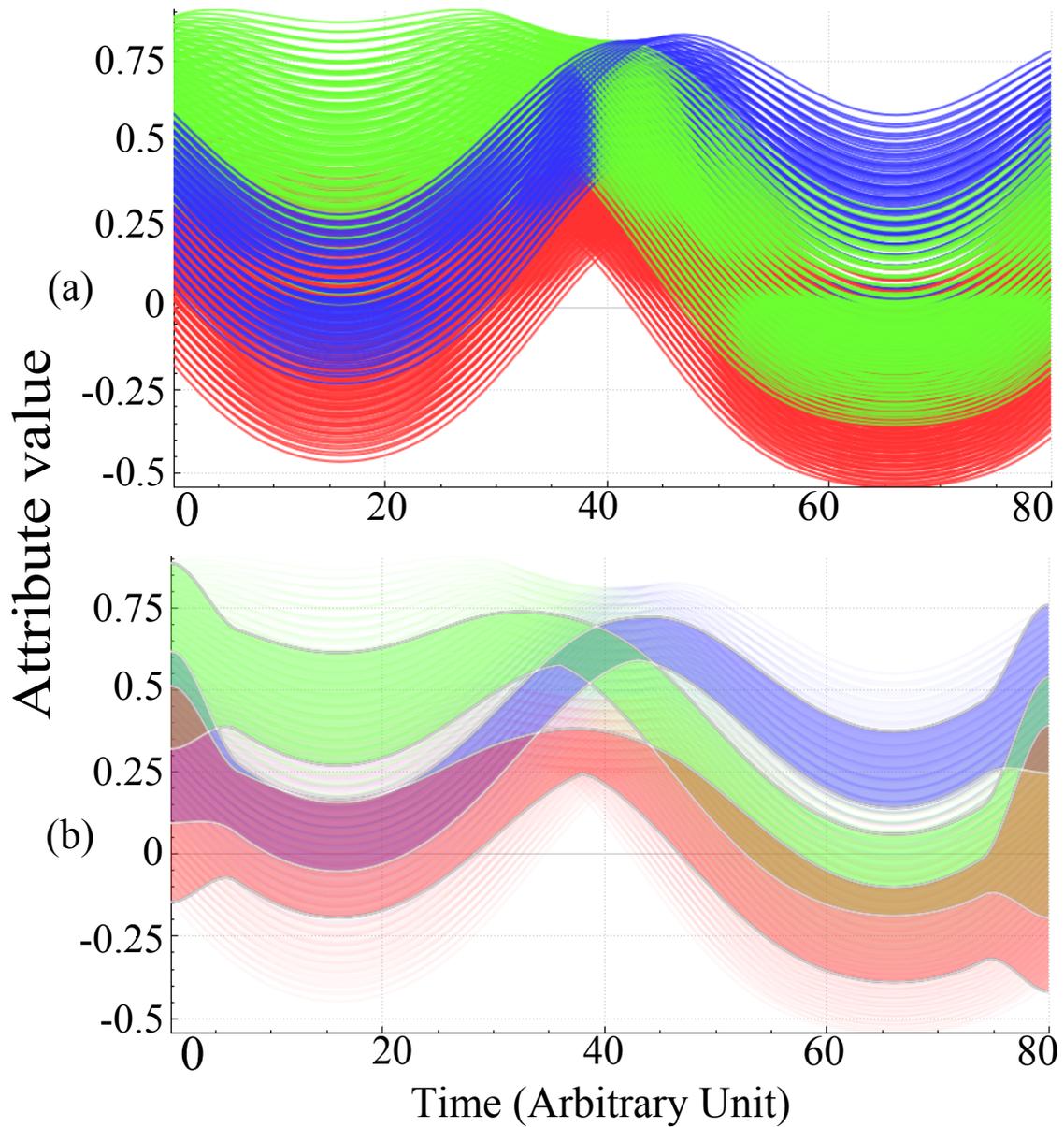


Figure 6.6: An example of TAC-based clustering. (a) direct visualization of the two clusters. (b) edge-bundling visualization of the two clusters. X axis is the time and Y axis stands for the attribute value. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them.

on top of each other, which may hide the information of the TACs that are covered. As shown in Figure 6.6(a), the three clusters are overlapping at certain time range, resulting in the visual clutter issue. This cluttering issue is worsened with the increase of the number of TACs.

If t_s and t_e , i.e. the starting and end times of the entire period of TACs, are considered as two data axis, the TACs plot is similar to the classic parallel coordinates plot (PCP). The difference is that PCP is applied to data where the axes do not correspond to points in time, and a point in PCP only has values at axes. While in a TAC plot, a TAC also has its values between the two axes (i.e. t_s and t_e), and different TACs may have different behaviors between t_s and t_e . Therefore, PCP technique cannot be directly applied to the TAC plots. Inspired by the work [40] that uses an edge-bundling method to generate an abstract version of the classic PCP to reduce visual clutter in PCP, an edge-bundling like layout of the clusters of TACs is created, aiming to produce an informative visualization of the TACs clusters with reduced clutter.

Different from the method in the work [40], data clustering cannot be applied on the values at the axes in TAC setting because this will change the coverage of TACs clusters at the beginning and the end. Another difference is that a cluster of TACs has its own behaviors, which cannot be represented by a general Bezier strip. To address this, a new visualization for the TAC clusters is presented. Specifically, given a cluster of TACs, the centroid and boundaries of the cluster are first derived (Figure 6.5(b)). Then the two boundaries are offset (or shrunk) towards to the centroid. The offset distance is set as 80% of the distance from the boundary to the centroid in this work (Figure 6.5(b)). The offset operation does not change the overall behaviors of the TACs in the cluster, while

the range of the cluster, i.e. the coverage of the attribute values at t_s and t_e is changed. To preserve the coverage information of the cluster, a head and a tail segments are created for the edge-bundling (Figure 6.5(c)), whose lengths are 10% of the TAC length, respectively. In the head and tail segments, the maximum and minimum of the attribute values at t_s and t_e are kept. In order to keep the smooth transition between the head (tail) segment and the body segment, a cubic Bezier spline is constructed (Figure 6.5(d)). Furthermore, halos around the edge-bundling of each cluster are rendered to reduce ambiguity at intersections. After applying the edge-bundling method, the clusters in Figure 6.6(a) are shown in Figure 6.6(b), which greatly reduces the clutters.

Since the clustering result is obtained from the hierarchical tree based on the AHC algorithm, it helps to better understand the relation between clusters, i.e. the merge or split changes with different K or ϵ , if the hierarchical structure of the clusters is provided. Therefore, a hierarchical tree of clusters is generated for the clustering result. In this hierarchical tree, each node represents a cluster (with the same color coding as the edge-bundling visualization) at different level of the tree. The hierarchy information can be easily extracted from the entire hierarchical tree of the AHC algorithm. Note that each node is not necessary an individual TAC, but rather a cluster of TACs during clustering. From this hierarchical tree, it can be easily figured out the merge/split relations among the clusters. As shown by the hierarchical tree in Figure 6.5(b), the purple cluster and the green cluster will be merged together when K changes from 6 to 5 (Figure 6.5(b)-(c)). In addition, the height of the levels is adjusted in the hierarchical tree according to the distance values on each level. In this way, even after the value of K is changed, the relative height of clusters is reserved.

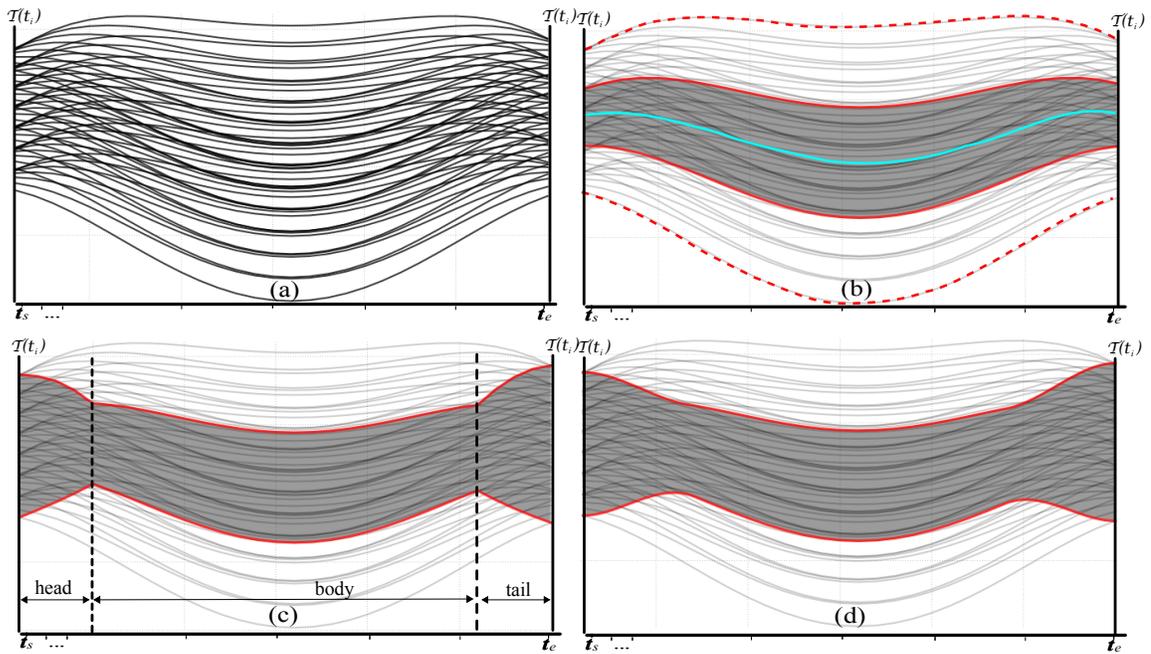


Figure 6.7: Illustration of edge-bundling method for TACs cluster visualization. TACs are divided into three parts: head, body, and tail to construct the edge-bundling. X axis is the time and Y axis stands for the attribute value.

6.2.3 A Hierarchical Clustering Algorithm in Time Dimension

Since TACs similarities could vary greatly in different time intervals, simply clustering TACs on the entire time period could miss interesting local temporal behaviors of TACs. Also, two TACs that belong to two clusters globally may possess segments having similar behaviors (see Figure 6.8 for an example), which cannot be captured in the above global clustering along the entire time period. To address this, a hierarchical clustering algorithm is proposed in time dimension, i.e. a temporal hierarchical clustering.

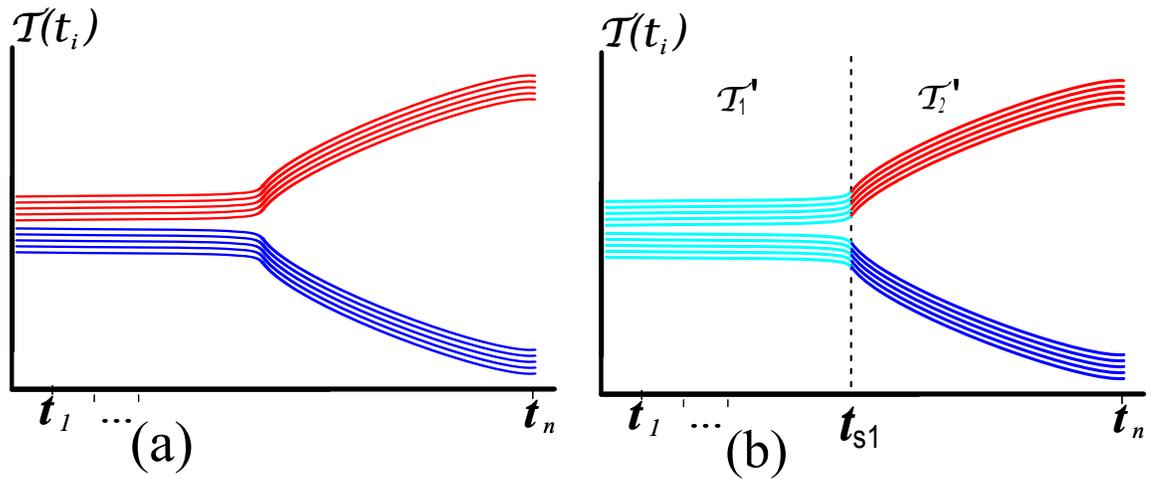


Figure 6.8: Motivation of temporal clustering. (a) two clusters (in red and blue, respectively) of TACs show different global behaviors in the entire time period. (b) Similar behaviors exhibit for the two groups of TACs in a local time interval (highlighted in cyan). X axis is the time and Y axis stands for the attribute value.

Three tasks need to be accomplished in the temporal hierarchical clustering: (1) achieving the appropriate temporal partitioning; (2) performing the AHC within each time interval obtained from the temporal partitioning in (1); (3) properly handling the transition of AHC results between consecutive time intervals. Achieving these goals is not trivial. First, all TACs have different temporal behaviors. Even the TACs that belong to the same clusters may still have slightly different behaviors (Figure 6.1(c)), which makes the selection of cutting points (or cuts) for temporal partitioning difficult. Second, the AHC performed in the individual time intervals and along the time axis should be consistent (e.g., in terms of the error threshold and similarity characterization of the clusters that they belong to). Third, the AHC results obtained in consecutive time intervals may not be identical. It is important to keep track of their transition relation (i.e. split or merge) across the cuts.

To address the above challenges and achieve the goals, the temporal AHC first performs a time interval segmentation (Section 6.2.3.1), which segments the entire time period T into a number of intervals with varying lengths. For each time interval, the TAC-based hierarchical clustering discussed in Section 6.2.2 is applied to show the similarities of sub-TACs in the specific time range (Section 6.2.3.2). Then a bottom-up hierarchy of the time intervals is constructed (Section 6.2.3.3). In order to accurately capture and represent the transition of AHC clusters across neighboring time intervals, a novel data structure and visualization (Section 6.2.3.4) is introduced. Figure 6.9 shows an example of hierarchical clustering on time dimension.

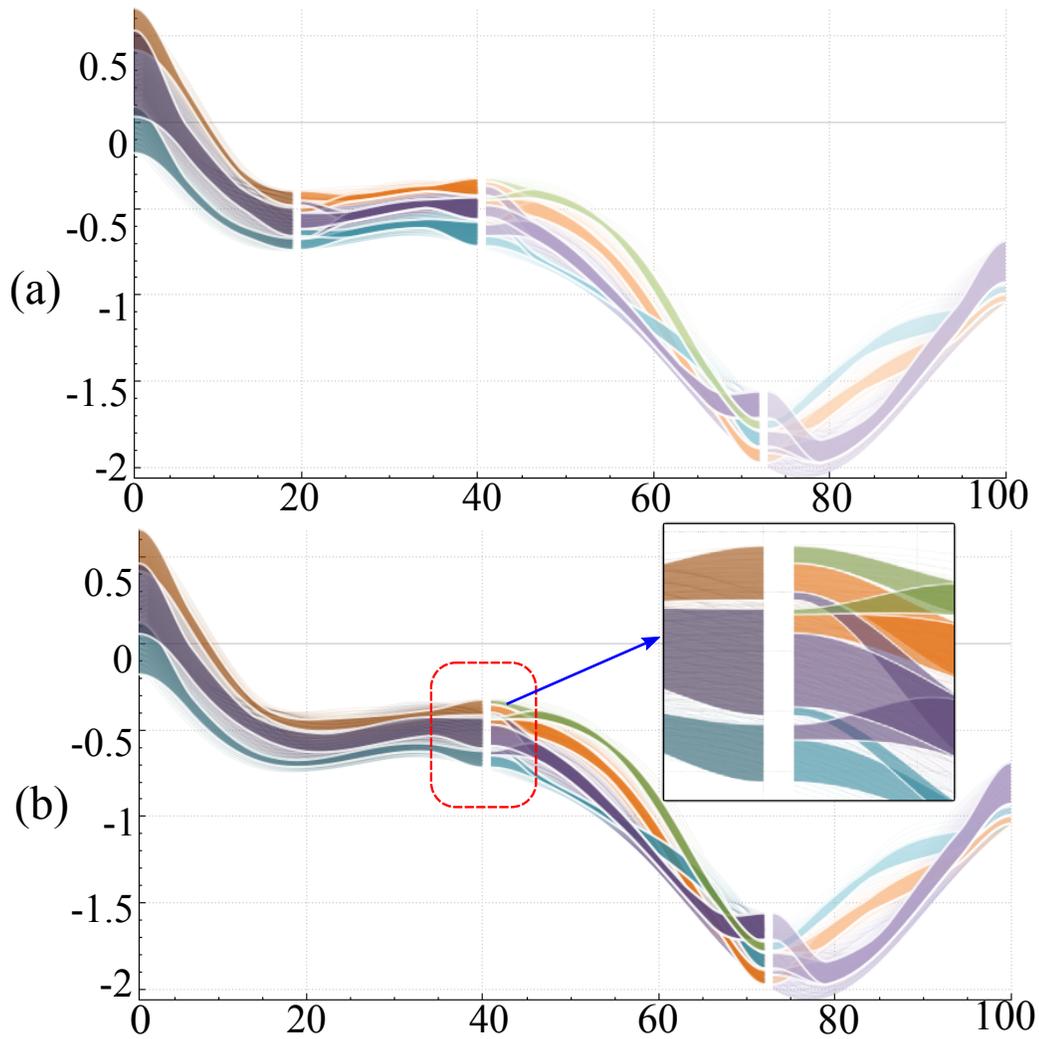


Figure 6.9: An example of temporal clustering. (a) initial hierarchical clustering result of $M = 4$ leaf nodes. (b) clustering result after merging the two most similar clusters in (a). X axis is the time and Y axis stands for the attribute value. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them.

6.2.3.1 Time Interval Segmentation of TAC

In order to study TACs in a level-of-detail fashion, time interval segmentation is applied to a group of TACs. The time intervals that segment TACs must preserve TAC characteristics. In other words, one primitive trend of a TAC is not expected to be segmented into two, which causes fragmentariness. For an individual TAC, the 1D Morse decomposition can be applied to generate the temporal sequences of TAC segments, as shown in Figure 6.1(b). However, for a group of TACs, it is not guaranteed that the segment splitting points are identical for all the TACs. To address this, the distribution of the inflexion points of all the TACs along the time axis are analyzed. Specifically, the statistics of the inflexion points in several small time ranges is obtained first, from which the top $M - 1$ ranges with a larger number of inflexion points are selected. Then in each range, a cut can be obtained by the average position of the inflexion points in the range. These $M - 1$ splitting positions segment the entire time period T into a number of intervals with varying length, referred to as $T = \langle T_1, T_2, \dots, T_M \rangle$. In this way, all TACs are segmented by these splitting positions which could preserve the most common characteristics of all the TACs. An example of time interval segmentation is illustrated in Figure 6.1(c).

6.2.3.2 Sub-TACs AHC within Local Time Intervals

After performing the temporal partitioning and obtained the local time intervals, the AHC algorithm can be performed within each interval. Different from the computation of *EDTW* distance of two TACs in the entire time period that may have different numbers of events and sizes, in a given time interval, all the sub-TACs have the same length. In

addition, all the TACs in a time interval have the same lifespan, making $Dist_{time}(\cdot, \cdot) = 0$. Therefore, to reduce the computation complexity, for a given pair Γ_1 and Γ_2 in a time interval $T_i = (t_s, t_e)$ where t_s and t_e are the starting and end times of T_i , their distance can be computed as follows:

$$Dist(\Gamma_1, \Gamma_2) = (1 + P_c \times Dist_{corr}(\Gamma_1, \Gamma_2)) \sqrt{\sum_{i=t_s}^{t_e} (\Gamma_1[i] - \Gamma_2[i])^2} \quad (6.11)$$

where $Dist_{corr}(\Gamma_1, \Gamma_2)$ is the correlation between Γ_1 and Γ_2 defined as Equation 6.9, and P_c is the penalty factor.

Assigning the cluster number for each time interval is difficult because on one hand the number of time intervals in temporal hierarchical clustering is varying, on the other hand the dissimilarities in different time intervals may be different. To show the consistent changes cross time intervals, the same standard needs to be applied to all the time intervals. Therefore, the distance threshold ε for the global clustering is used to guide the clustering among time intervals. Specifically, the distance threshold ε_i for time interval T_i is decided by the time range of an interval, i.e. $\varepsilon_i = \frac{|T_i|}{|T|} \varepsilon$. In this way, it is foreseeable that there are more clusters generated in the time intervals where the TACs behave more diversely, i.e. TACs have larger dissimilarities.

6.2.3.3 Temporal Hierarchical Clustering

The goal of temporal clustering is to build up a hierarchical tree of the input M time intervals obtained in the previous temporal partitioning, i.e. M leaf nodes of the tree, so that the level-of-details of TAC behaviors can be revealed in time dimension. Different

from the spatial hierarchical clustering, in which any two clusters can be selected for a merging operation, in temporal hierarchical clustering, considering the continuity of time, only two clusters that are neighboring in time can be merged together, which makes the merging operation easier. For example, for the initial M leaf nodes, there are only $M - 1$ candidate pairs to choose. In the implementation, starting from the initial M leaf nodes, a distance array $D \in \mathbb{R}^{(M-1)}$ is created. Each entry indicates the dissimilarity changes after a pair of consecutive time intervals is merged into one. $D[i]$ can be computed as follows.

$$D[i] = \eta(T_i) + \eta(T_{i+1}) - \eta(T_i \cup T_{i+1}) \quad (6.12)$$

where $\eta(T_k)$ is the average pairwise dissimilarity within a time interval T_k , $T_i \cup T_{i+1}$ is the new time interval by merging T_i and T_{i+1} .

$$\eta(T_k) = \sqrt{\frac{\sum_{i=1}^{|T_k|} \sum_{j=i}^{|T_k|} (\text{Dist}(\Gamma_i, \Gamma_j))^2}{|T_k|(|T_k| - 1)}} \quad (6.13)$$

$\eta(T_k)$ reflects the compactness of the TACs in the time interval T_k . The larger $\eta(T_k)$ is, the further the TACs in T_k are located away from the centroid.

In the spatial hierarchical clustering, two clusters with the smallest distance are selected for merging. Similarly, in temporal hierarchical clustering, two time intervals with the smallest dissimilarity changes are merged together first. In other words, time interval T_k and its neighboring time interval T_{k+1} that satisfies $D[k] \leq D[i], \forall 1 \leq i \leq M$, are first merged together to generate a new time interval $T_k + T_{k+1}$ and then removed from the node list. Consequently, a new $M - 2$ dimension distance array $D_{(M-2)}$ is generated with the

remaining $M - 1$ nodes. The above merge process is iteratively applied until only one time interval, i.e. the entire time period, is left, which is the root of the temporal hierarchical tree. The height of the temporal hierarchical tree built on the M time intervals is $M - 1$. On the i^{th} level of the tree, i.e. height is i , there are $M - i$ time intervals.

6.2.3.4 Visualization of Changes of Local AHC Results Across Cuts

In the visualization of TAC clusters, a specific color is assigned for each cluster according to its unique ID, which is clear when the entire TACs are considered for clustering (i.e. no temporal clustering). In contrast, in temporal clustering among a number of time intervals, each time interval is clustered independently, therefore, the colors (i.e. cluster IDs) through time intervals may not be consistent. An example is shown in Figure 6.10(a). The colors of the three clusters in time interval T_2 do not match with those in time interval T_1 .

To resolve the above issue, an adjustment of cluster IDs is required. The cluster IDs in time interval T_k is adjusted based on its previous time interval T_{k-1} . Specifically, a cluster C_p in T_k should be adjusted according to its *main source cluster* in T_{k-1} , i.e. the cluster from which most TACs in C_p originate. For example, in Figure 6.10(b), most of the TACs in cluster C_1 in T_2 originate from C_1 in T_1 , so its main source cluster is C_1 in T_1 . In the following, $C_{p,k}$ is used to denote the p^{th} cluster in the time interval T_k . To achieve this adjustment, a connection matrix $D_{m \times n}$ is created where m and n are the numbers of clusters in T_{k-1} and T_k , respectively. The value of entry D_{ij} stores the number of TACs from the cluster $C_{i,k-1}$ to the cluster $C_{j,k}$. The adjustment is completed by re-identifying the clusters in T_k one by one. For each column j in the connection matrix $D_{m \times n}$, it finds the largest value, for example, $D(p, j)$. If the cluster $C_{p,k-1}$ is not assigned as the main

source cluster to other clusters in T_k , the cluster ID of $C_{j,k}$ is re-assigned as p . Otherwise it keeps searching for the next largest value. If all the clusters in T_{k-1} are already assigned, which is possible when $n > m$, a new cluster ID, i.e. a new color, is assigned to C_j . In Figure 6.10(b), the clusters $C_{1,2}$, $C_{2,2}$ and $C_{3,2}$ are adjusted with Cluster IDs as 2, 1 and 3, respectively. After the adjustment, the colors of the clusters from T_1 to T_2 become consistent.

With the color consistence, it is easy to identify the changes through time intervals, especially for the sources of the clusters. However, the visual clutter and overplotting issue still exists Figure 6.10(b). One solution is to apply an edge bundling visualization as introduced in the work [40] to aggregate the sub-TACs in each time interval, as shown in Figure 6.10(c), from which the main source of each cluster in T_2 is easily identified by the consistent colors and with the reduced clutter. However, the transition of the clustering of two neighboring intervals across a cut is not fully revealed. Take the matrix in Figure 6.10(a) for an example, among 17 TACs in $C_{2,2}$, 10 are from $C_{1,1}$ and 7 are from $C_{2,1}$. The main source cluster $C_{1,1}$ is indicated by the consistent color from $C_{1,1}$ in T_1 to $C_{2,2}$. However, the information of the *minor source cluster* $C_{2,1}$ is ignored.

To fully reveal the connections among time intervals, given two neighboring time intervals T_{k-1} and T_k , the edge-bundling visualization on both the tail of T_{k-1} and the head of T_k can be improved. The tail of T_{k-1} is first adjusted non-overlapping. As discussed in Section 6.2.2.4, the tail of a cluster indicates the range of TACs at the end of the time interval. TACs belonging to two clusters may overlap at the tail. One simple way is to equally split the total range at the end of the tail into some non-overlapping sub-ranges and assign

each cluster with one sub-range. But the sub-ranges cannot reveal the real ranges of clusters, and the relative range is also hidden by the equal sub-ranges. In the proposed method, an offset is applied to both clusters whose tails are overlapping. The offset is proportional to the size of the clusters. As illustrated by the red arrow in Figure 6.10(d), the minimum value of C_2 at the tail end is increased and the maximum value of C_3 at the tail end is decreased, eliminating the overlapping between C_2 and C_3 while keeping the relative range size simultaneously.

Removing overlapping at the tail of T_{k-1} makes the boundaries of source clusters clear. In order to reveal all the source of clusters in T_k , including both main source and minor sources, the head of the clusters in T_k is re-constructed. Specifically, if a cluster C_i in T_{k-1} is split into a number of clusters in T_k , i.e. multiple destinations, a head from the tail of C_i is plot no matter C_i is the main source or not. For example, as illustrated by the green arrow in Figure 6.10(d), C_2 in T_1 is split into three clusters in T_2 , accordingly, three heads from the tail of C_2 are connected to the three clusters in T_2 . Note that the width of the heads is proportional to the number of TACs from C_2 , which enables the differentiation of the main sources and minor sources. From Figure 6.10(d), it is easy to identify the changes of the clusters from T_{k-1} to T_k .

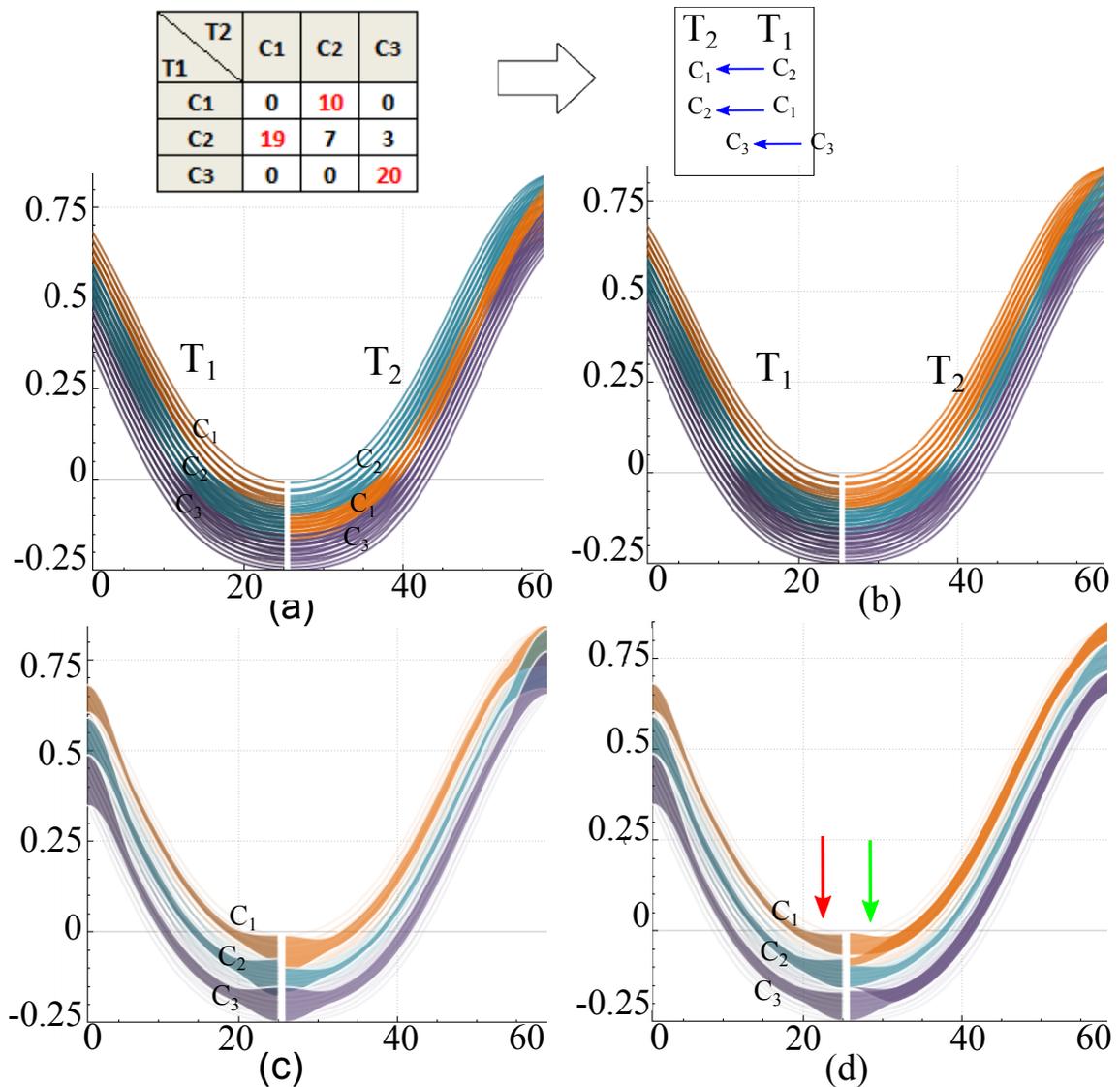


Figure 6.10: Visualization of changes through time intervals. (a) result before cluster ID adjustment. (b) result after cluster ID adjustment. (c) edge-bundling visualization of the result. (d) modified edge-bundling visualization. T_1 and T_2 represent the two time intervals. X axis is the time and Y axis stands for the attribute value. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them.

6.3 Results

The proposed TAC-based clustering has been applied to a number of flow data sets to demonstrate its effectiveness. In this section, the results of TAC clustering based on different attributes are discussed. After selecting/setting the threshold for the hierarchical clustering tree, a number of clusters of the TACs will be generated. An example of the TAC-based clustering result is shown in Figure 6.11(a). Due to the large number of input TACs, overlap of the TACs in different clusters cannot be resolved by a naive transparency control. Therefore, for each cluster, a representative TAC is generated, as shown in Figure 6.11(b), which provides an overview of the attribute space. Figure 6.11(c) provides the visualization of the flow domain, which is colored based on the corresponding clusters.

6.3.1 Hierarchical Clustering of Eulerian TACs

Figure 6.12 shows the TAC-based clustering result of Double Gyre flow based on Eulerian TACs of attribute λ_2 . λ_2 is one of the local criteria for the identification of a vortex [24]. The region within the vortex core has negative λ_2 values. Figure 6.12(a) is the clustering result based on Euclidean distance metric with the cluster number $K = 2$, from which two clusters are generated. One includes TACs with local negative λ_2 , i.e. the green cluster that indicates the two vortices. The other cluster contains TACs with local positive λ_2 , i.e. the red cluster that indicates non-vortex regions. As discussed in Section 6.2.1, the transformation information of TACs cannot be revealed by the Euclidean distance. Compared to Figure 6.12(a), the clustering result based on the *EDTW* distance metric in Figure 6.12(b) does not indicate vortex regions directly, but rather show the oscillation

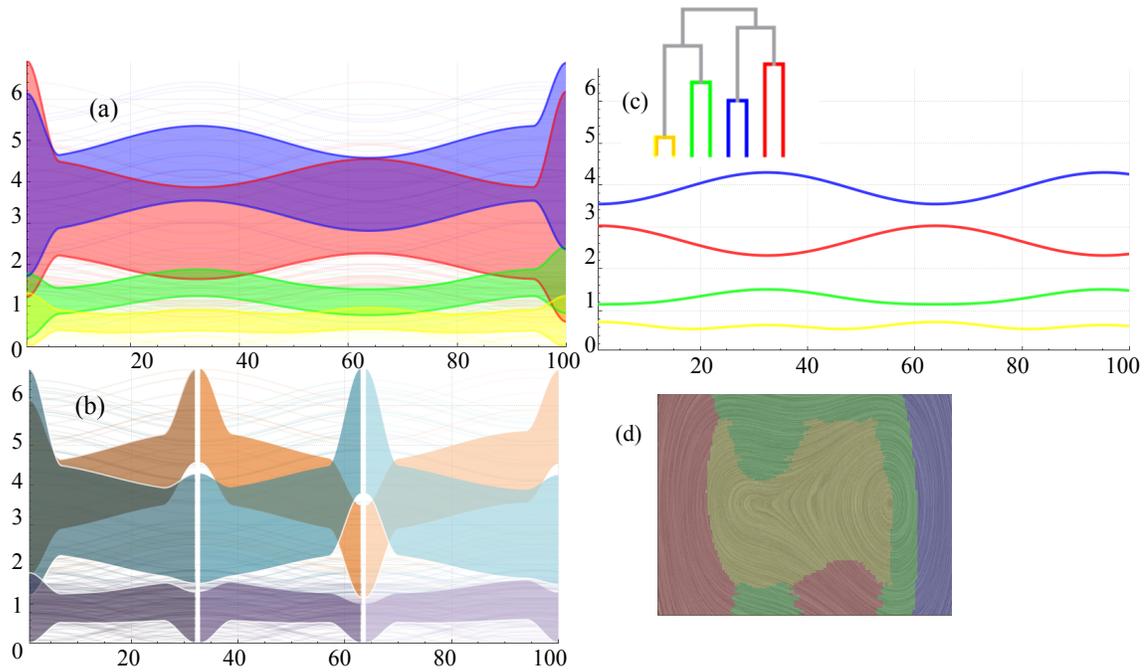


Figure 6.11: Different views in the system. (a) shows the visualization of clustering result. (b) shows the representative TAC in each cluster in (a) and a hierarchical tree of the clustering result. (c) shows the visualization of temporal hierarchical clustering result. (d) shows the clustering on flow domain based on the result in (a). X axis is the time and Y axis stands for the attribute value of curl. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them. For the texture-based visualization of the flow, the horizontal direction is X and the vertical direction is Y.

behaviors of the two vortex systems. The TACs in the green cluster change from positive λ_2 to negative λ_2 , which indicates that the corresponding locations are not in a vortex (or near the center of the vortex) at the beginning, but the vortex enters this region in a later time. Such a temporal behavior may be better depicted via the temporal clustering. Figure 6.12(c) shows the clustering result with $K = 3$ based on *EDTW* distance metric.

Figure 6.13 shows the hierarchical clustering result of the Cylinder Flow [65] based on Eulerian TACs of attribute curl. When K is 3, the blue and red clusters show the regions

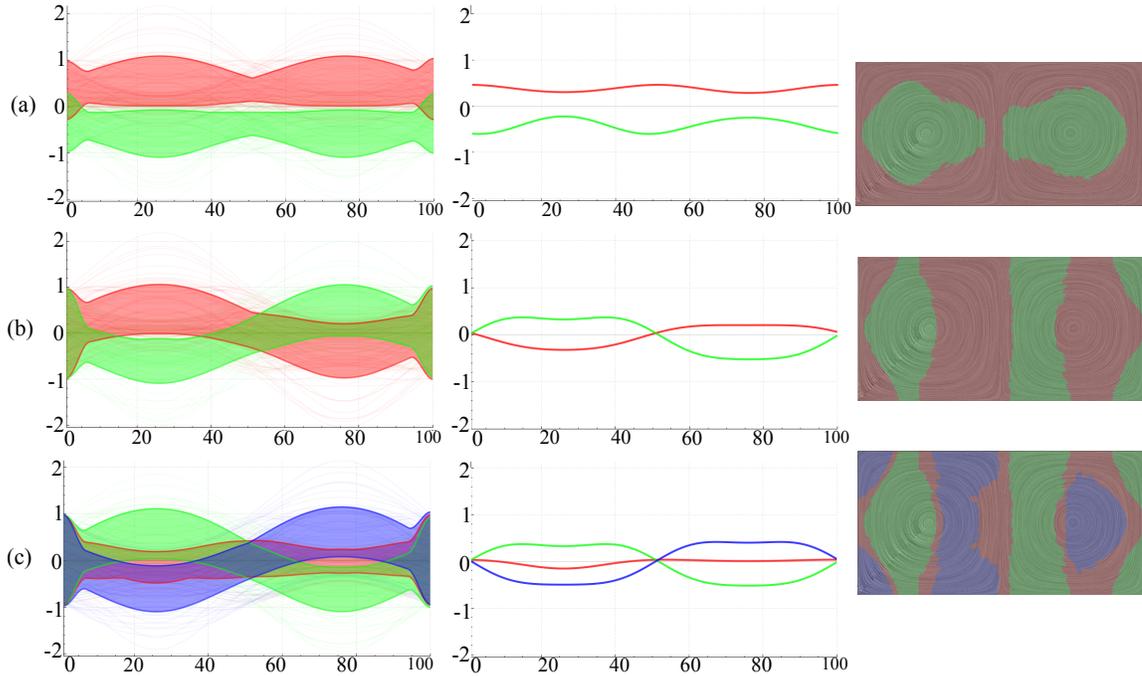


Figure 6.12: Clustering result of the Double Gyre flow based on Eulerian TACs of attribute λ_2 . (a) Clustering result based on Euclidean distance metric. (b) and (c) Clustering results based on *EDTW* distance metric with the cluster number $K = 2$ and $K = 3$, respectively. The columns from left to right show the visualization of TAC clusters, representative TAC in each cluster and clustering of flow domain, respectively. X axis is the time and Y axis stands for the attribute value of λ_2 . Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them. For the texture-based visualization of the flow, the horizontal direction is X and the vertical direction is Y.

near the vortices. They are not exactly vortex regions because of the translational behavior of the vortices in the flow. When K increases to 4, a yellow cluster is separated from the green cluster, which exhibits similar behavior to the red cluster but with certain shifting. In fact, the regions highlighted by the yellow cluster correspond to the boundaries between two neighboring vortices with different rotational orientations. When we increase K to 5, a cyan cluster is separated from the green cluster again, revealing similar behavior to the yellow cluster. By changing the number of generated clusters, more and more details can

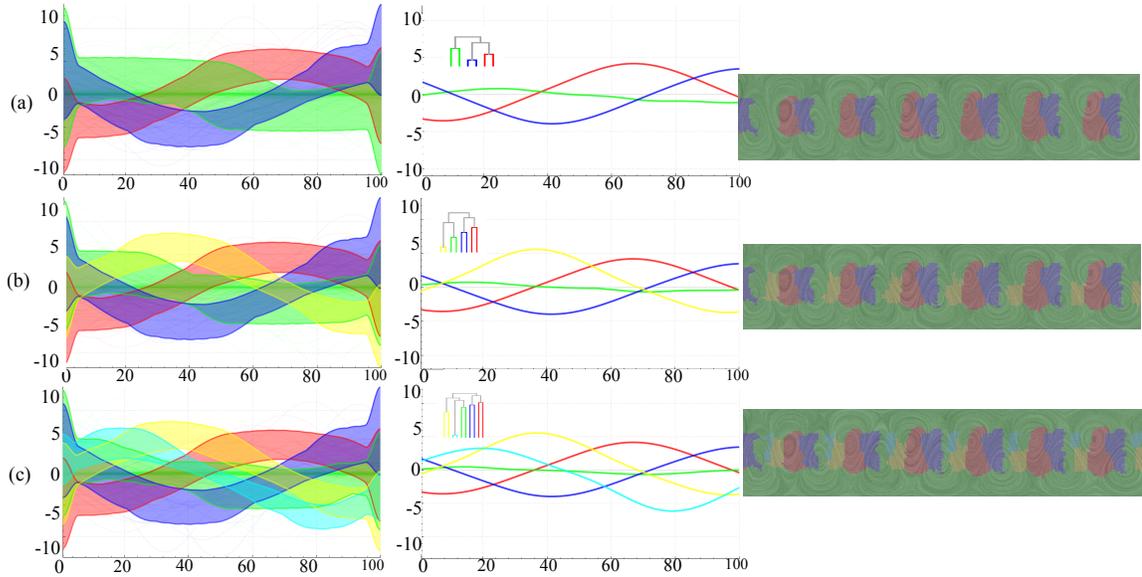


Figure 6.13: Clustering result of Cylinder flow based on Eulerian TACs of attribute *curl*. (a), (b) and (c) Clustering results based on *EDTW* distance metric with the cluster number $K = 3$, $K = 4$, and $K = 5$, respectively. The columns from left to right show the visualization of TAC clusters, representative TAC in each cluster and clustering of flow domain, respectively. X axis is the time and Y axis stands for the attribute value of *curl*. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them. For the texture-based visualization of the flow, the horizontal direction is X and the vertical direction is Y.

be revealed.

6.3.2 Hierarchical Clustering of Lagrangian TACs

Figure 6.14 shows the hierarchical clustering result of Double Gyre flow based on Lagrangian TACs of attribute *curl* with different cluster number K . With the increasing number of clusters, more details are revealed. The clustering of flow domain indicates the LCS structure, which is expected from the Lagrangian view. One noteworthy observation

is that the vortex structure is stable when the cluster number is changed.

Figure 6.15 shows the hierarchical clustering result of the Cylinder flow based on Lagrangian TACs of attribute *curl* with different cluster number K . With the increasing number of clusters, more details in the vortices are revealed. The clustering on the flow domain from the curl field reveals the Von Karman vortex street of the Cylinder flow, compared to the results obtained with the Eulerian TACs (Figure 6.13).

6.3.3 Temporal Hierarchical Clustering

Figure 6.16 shows the temporal hierarchical clustering result of the Double Gyre flow based on Lagrangian TACs of attribute *momentum* with different number of time intervals M . With the increasing number of time intervals, the TACs in each time interval show the merging and splitting behaviors, which indicate the temporal behaviors of the movement of the two vortices. the temporal behavior of λ_2 . This temporal movement of the two vortices cannot directly be revealed by the clustering based on Eulerian TACs shown in Figure 6.12 (b) - (c).

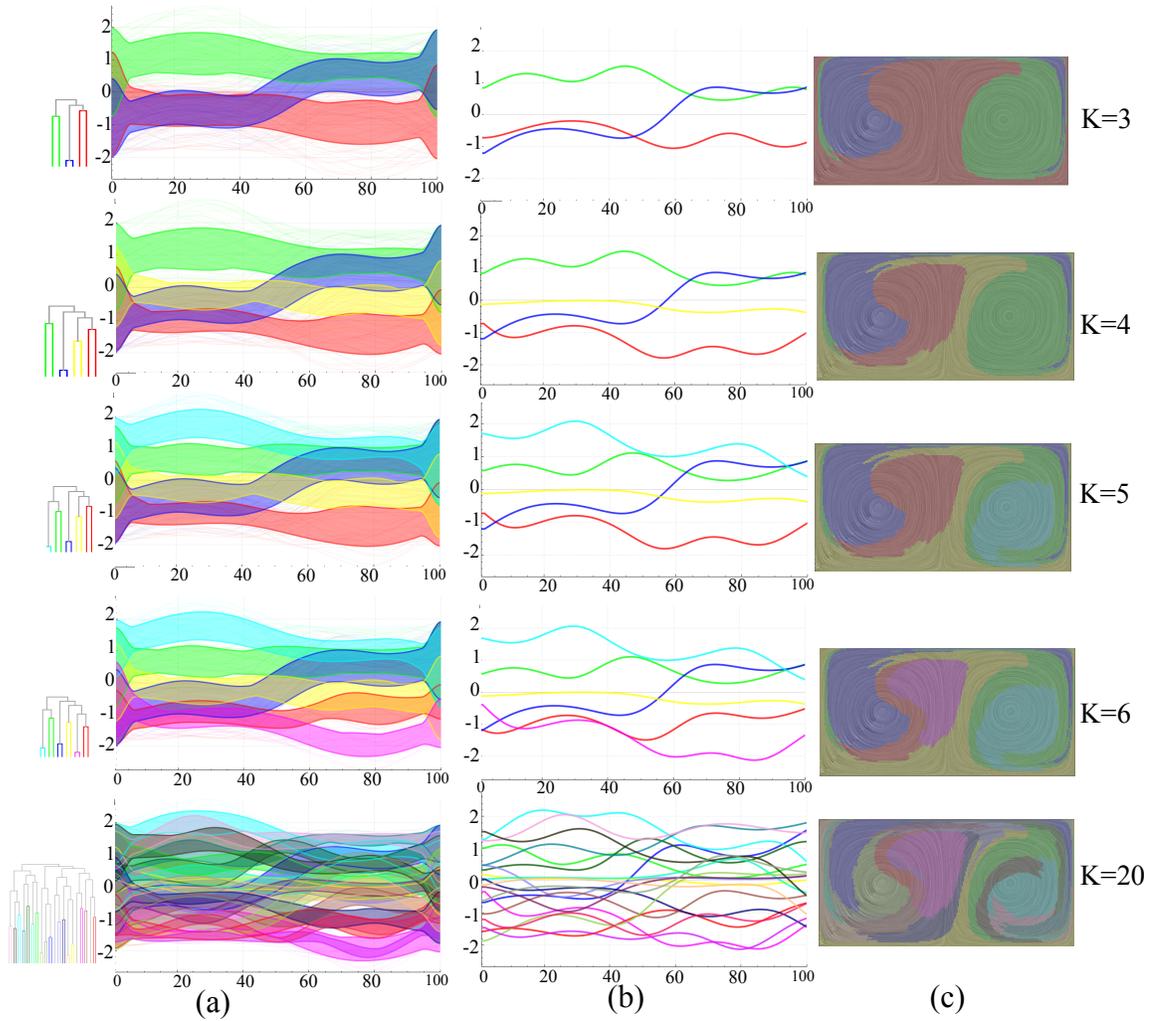


Figure 6.14: Clustering result of the Double Gyre flow based on Lagrangian TACs of attribute *curl* with different cluster number K . (a) TAC clusters and its hierarchical tree. (b) representative TACs in each cluster. (c) Clustering of flow domain. X axis is the time and Y axis stands for the attribute value of *curl*. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them. For the texture-based visualization of the flow, the horizontal direction is X and the vertical direction is Y.

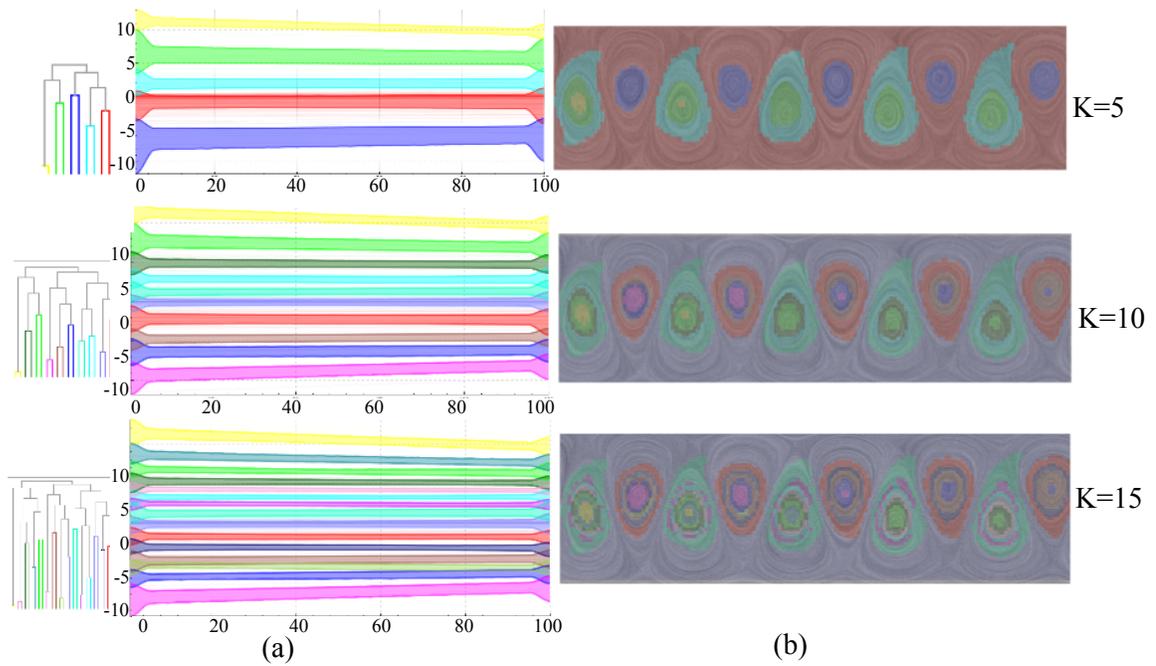


Figure 6.15: Clustering result of Cylinder flow based on Lagrangian TACs of attribute *curl* with different cluster number K . (a) TAC clusters and its hierarchical tree. (b) Clustering of flow domain. X axis is the time and Y axis stands for the attribute value of *curl*. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them. For the texture-based visualization of the flow, the horizontal direction is X and the vertical direction is Y.

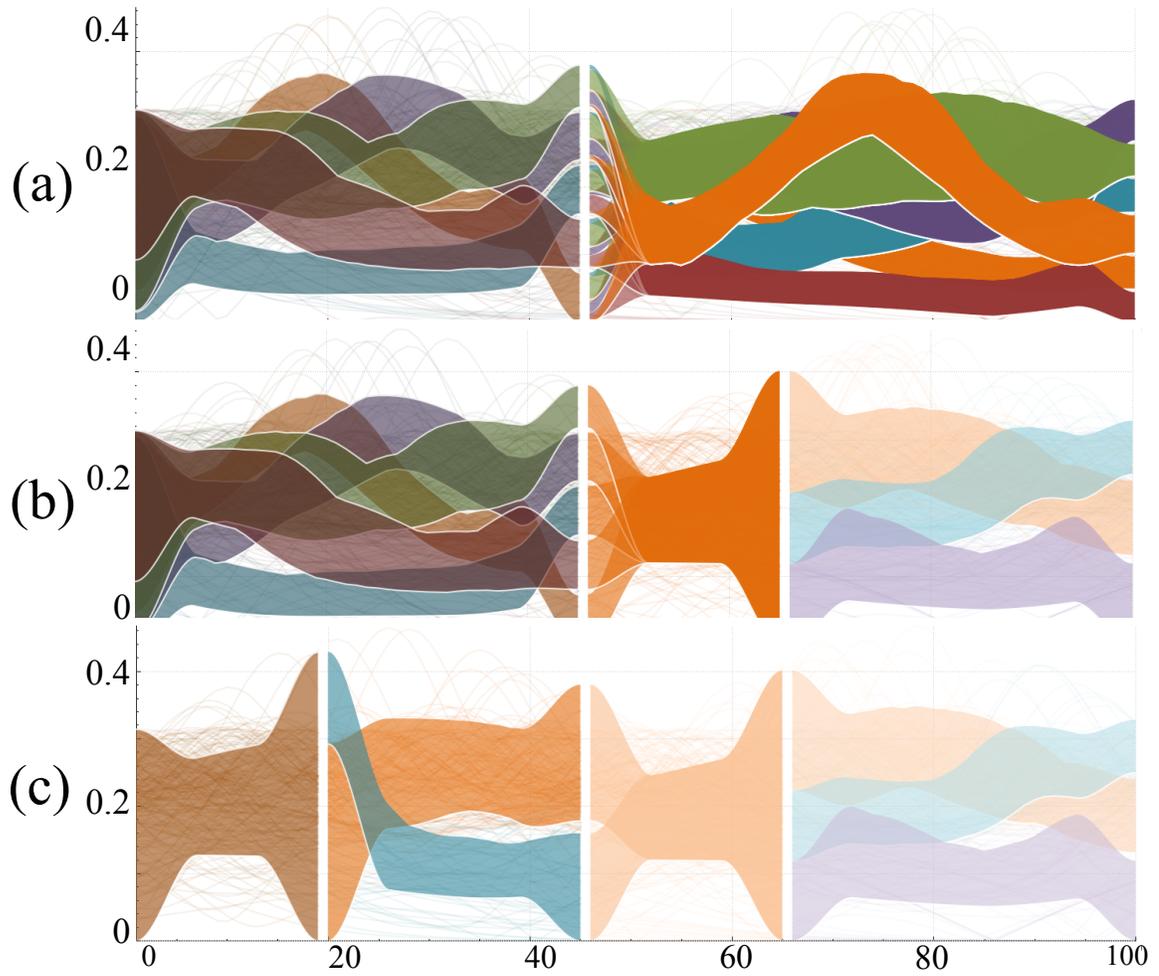


Figure 6.16: Clustering result of Double Gyre flow based on Lagrangian TACs of attribute *momentum* with different number of time intervals M . (a) $M = 2$. (b) $M = 3$. (c) $M = 4$. X axis is the time and Y axis stands for the attribute value. Those values are derived from the velocity and are not real physical quantities, therefore, there is no unit for them.

Chapter 7

Conclusion

In this work, a Lagrangian accumulation framework is first proposed. By accumulating various local physical and geometric properties along the associated integral curves, a number of \mathcal{A} fields are derived. The properties of \mathcal{A} fields are discussed systematically, which are utilized to achieve interesting visualizations for both 2D and 3D vector fields. Some well-known flow features are compared with the results of \mathcal{A} fields and its gradient fields. The applications of the Lagrangian accumulation framework on a number of flow visualization tasks are then discussed, including seeding curve generation for integral curves and surface placement and flow domain segmentation. In addition, an algorithm to extract the discontinuity of the behavior between neighboring integral curves based on \mathcal{A} fields is introduced, which shows that this discontinuity may be closely related to a number of flow features. Different strategies to combine individual attribute fields are studied to form a super attribute field to study the spatial correlation of the attribute fields. In order to address the limitation of the Lagrangian accumulation framework on revealing

the detailed behaviors in flow exploration, a new TAC-based flow exploration framework is introduced. In addition, a new distance metric is proposed to measure the similarity of segments of TACs for the clustering. The hierarchical clustering supports a level-of-detail representation of flow behavior in both space and time.

Limitations and future work There are a number of limitations that the user should be aware of. For the Lagrangian accumulation framework, first, even though this work has shown that choosing different window sizes for the accumulation may be employed to generate various visualizations, the selection of an appropriate window is highly application-dependent, which may influence both the computational cost and the revealed patterns. Similarly, the sampling strategy could affect the information that can be captured by the \mathcal{A} fields. Second, during the accumulation, the characteristic values may cancel each other. For instance, if one accumulates the change of the flow direction along a symmetric integral curve that has the behavior similar to a sine function, the resulted value can be zero. Third, the discussed accumulation is also a dimensionality reduction process (i.e. reducing the 1D information into a single value), which will surely result in information loss. However, this information loss and a solution to reducing it have not been carefully discussed, which will be investigated in the future. For the TAC-based framework, the relationship between TACs of neighboring particles is lack of discuss. The study of this relationship can start with the concept of stable sets. A stable set can be defined as a group of geometrically neighboring particles whose TACs have high similarity to a given TAC of interest. The particles in a stable sets show coherent behaviors not only in geometrics but also in attribute space. Exploring stable sets in the flow domain will enable domain experts to better understand the relation between geometrics attributes and physical attributes. The

current TAC-based framework utilize individual TACs derived from a specific attribute. If multiple attributes are considered, multi-TACs can be generated. Therefore, a potential extension of the current work is the multi-TAC based flow exploration, which is a way to study the relationship between different attributes. At last, the application of TAC-based framework to 3D unsteady flow should also be studied in the future.

Chapter 8

Appendix

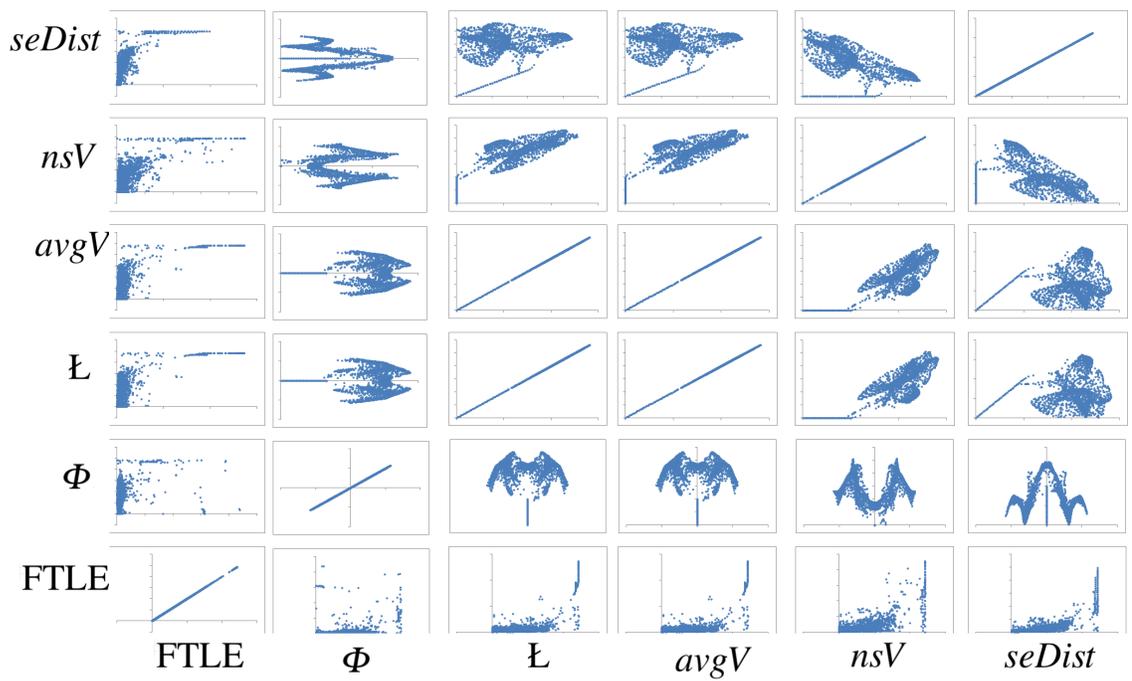


Figure 8.1: The scatter plot matrix of different \mathcal{A} fields of the Double Gyre flow. Part 1 of Figure 5.25

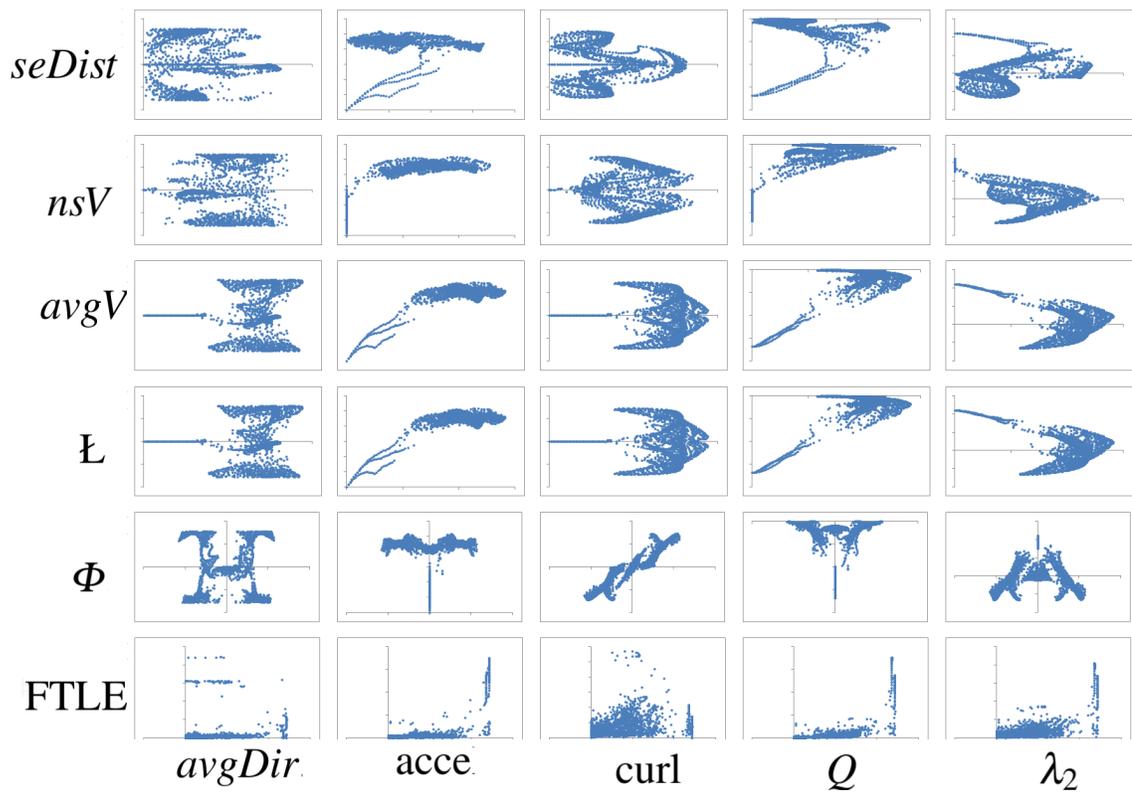


Figure 8.2: The scatter plot matrix of different \mathcal{A} fields of the Double Gyre flow. Part 2 of Figure 5.25

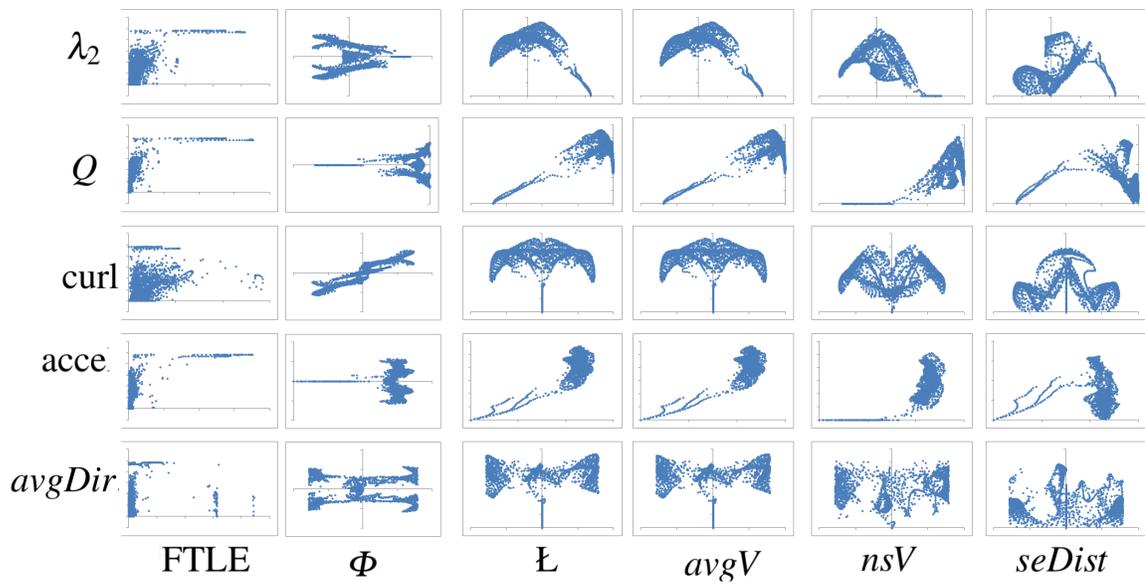


Figure 8.3: The scatter plot matrix of different \mathcal{A} fields of the Double Gyre flow. Part 3 of Figure 5.25

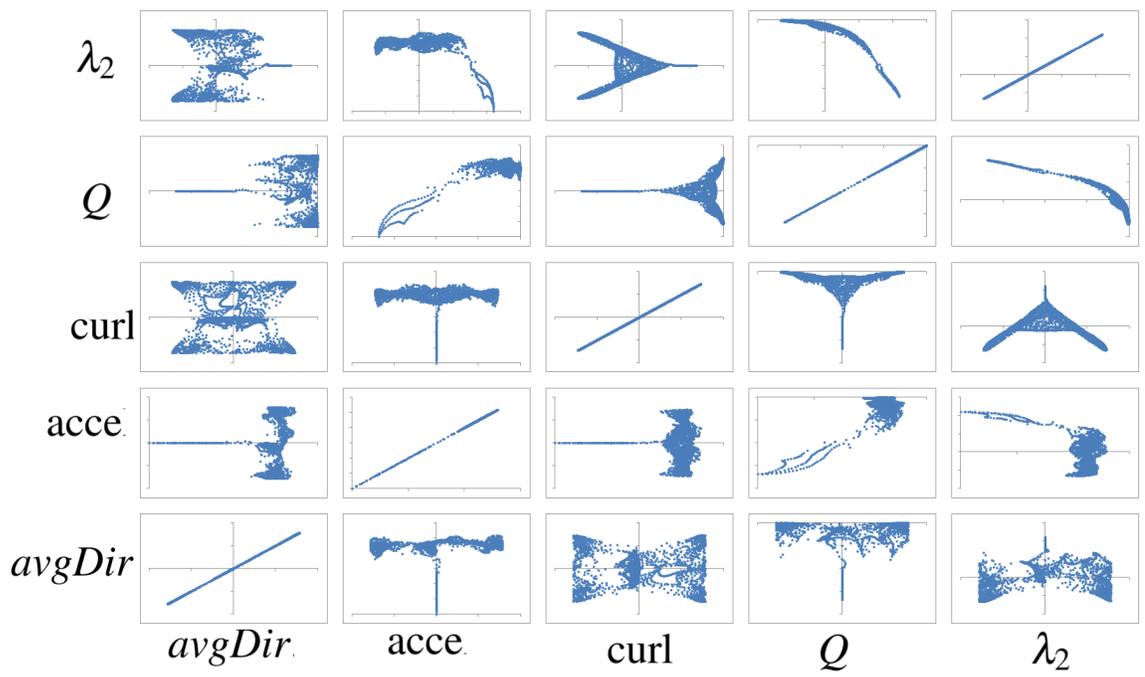


Figure 8.4: The scatter plot matrix of different \mathcal{A} fields of the Double Gyre flow. Part 4 of Figure 5.25

Bibliography

- [1] A. Brambilla, R. Carnecky, R. Peikert, I. Viola, and H. Hauser. Illustrative Flow Visualization: State of the Art, Trends and Challenges. In M.-P. Cani and F. Ganovelli, editors, *Eurographics 2012 - State of the Art Reports*, pages 75–94. The Eurographics Association, 2012.
- [2] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 263–270. ACM, 1993.
- [3] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.
- [4] J. Chandler, H. Obermaier, and K. Joy. Interpolation-based pathline tracing in particle-based flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 21(1):68–80, 2015.
- [5] G. Chen, K. Mischaikow, R. S. Laramée, 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, 2007.
- [6] G. Chen, K. Mischaikow, R. S. Laramée, and E. Zhang. Efficient Morse decompositions of vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):848–862, 2008.
- [7] J.-L. Chen, Z. Bai, B. Hamann, and T. J. Ligocki. Normalized-cut algorithm for hierarchical vector field data segmentation. In *Electronic Imaging*, pages 79–90, 2003.
- [8] Y. Chen, J. D. Cohen, and J. Krolik. Similarity-guided streamline placement with error evaluation. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1448–1455, 2007.

- [9] R. Cucitore, M. Quadrio, and A. Baron. On the effectiveness and limitations of local criteria for the identification of a vortex. *European Journal of Mechanics-B/Fluids*, 18(2):261–282, 1999.
- [10] M. Edmunds, R. S. Laramée, G. Chen, N. Max, E. Zhang, and C. Ware. Surface-based flow visualization. *Computers & Graphics*, 36(8):974–990, 2012.
- [11] P. Esling and C. Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12, 2012.
- [12] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann. Time-hierarchical clustering and visualization of weather forecast ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):831–840, 2017.
- [13] R. Fuchs, J. Kemmler, B. Schindler, J. Waser, F. Sadlo, H. Hauser, and R. Peikert. Toward a lagrangian vector field topology. In *Computer Graphics Forum*, volume 29, pages 1163–1172. Wiley Online Library, 2010.
- [14] 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.
- [15] R. C. Gonzalez. *Digital image processing*. Pearson Education India, 2009.
- [16] H. Guo, F. Hong, Q. Shu, J. Zhang, J. Huang, and X. Yuan. Scalable lagrangian-based attribute space projection for multivariate unsteady flow data. In *Pacific Visualization Symposium (PacificVis), 2014 IEEE*, pages 33–40. IEEE, 2014.
- [17] H. Guo, X. Yuan, J. Huang, and X. Zhu. Coupled ensemble flow line advection and analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2733–2742, 2013.
- [18] H. Guo, J. Zhang, R. Liu, L. Liu, X. Yuan, J. Huang, X. Meng, and J. Pan. Advection-based sparse data management for visualizing unsteady flow. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2555–2564, 2014.
- [19] G. Haller. Lagrangian structures and the rate of strain in a partition of two-dimensional turbulence. *Physics of Fluids*, 13(11):3365–3385, 2001.
- [20] G. Haller and T. Sapsis. Lagrangian coherent structures and the smallest finite-time lyapunov exponent. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 21(2):023115, 2011.

- [21] J. L. Helman and L. Hesselink. Representation and display of vector field topology in fluid flow data sets. *IEEE Computer*, 22(8):27–36, 1989.
- [22] J. C. Hunt, A. Wray, and P. Moin. Eddies, streams, and convergence zones in turbulent flows. In *Studying Turbulence Using Numerical Simulation Databases*, 2, volume 1, pages 193–208, 1988.
- [23] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.
- [24] J. Jeong and F. Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–94, 1995.
- [25] J. Kasten, J. Reininghaus, I. Hotz, and H.-C. Hege. Two-dimensional time-dependent vortex regions based on the acceleration magnitude. *Transactions on Visualization and Computer Graphics (Vis’11)*, 17(12):2080–2087, 2011.
- [26] 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 *Vision Modeling and Visualization*, pages 191–198, 2011.
- [27] 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.
- [28] 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)*, pages 1–19. Springer, 2007.
- [29] R. S. Laramee, H. Hauser, H. Doleisch, F. H. Post, B. Vrolijk, and D. Weiskopf. The state of the art in flow visualization: dense and texture-based techniques. *Computer Graphics Forum*, 23(2):203–221, June 2004.
- [30] T.-Y. Lee and H.-W. Shen. Visualizing time-varying features with tac-based distance fields. In *Visualization Symposium, 2009. PacificVis’ 09. IEEE Pacific*, pages 1–8. IEEE, 2009.
- [31] H. Li, W. Chen, and I.-F. Shen. Segmentation of discrete vector fields. *Visualization and Computer Graphics, IEEE Transactions on*, 12(3):289–300, 2006.
- [32] T. W. Liao. Clustering of time series dataa survey. *Pattern Recognition*, 38(11):1857–1874, 2005.

- [33] 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 PacificVis '13: IEEE Pacific Visualization Symposium*, Sydney, Australia, march 2013.
- [34] M. Macklin and M. Müller. Position based fluids. *ACM Transactions on Graphics (TOG)*, 32(4):104, 2013.
- [35] 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.
- [36] T. McLoughlin, M. W. Jones, R. S. Laramée, 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.
- [37] T. McLoughlin, R. S. Laramée, R. Peikert, F. H. Post, and M. Chen. Over two decades of integration-based, geometric flow visualization. In *Computer Graphics Forum*, volume 29, pages 1807–1829. Wiley Online Library, 2010.
- [38] S. K. Mitra and Y. Kuo. *Digital signal processing: a computer-based approach*, volume 2. McGraw-Hill New York, 2006.
- [39] A. Okada and D. L. Kao. Enhanced line integral convolution with flow feature detection. In *Electronic Imaging'97*, pages 206–217. International Society for Optics and Photonics, 1997.
- [40] G. Palmas, M. Bachynskyi, A. Oulasvirta, H. P. Seidel, and T. Weinkauff. An edge-bundling layout for interactive parallel coordinates. In *Visualization Symposium (PacificVis), 2014 IEEE Pacific*, pages 57–64. IEEE, 2014.
- [41] Z. Peng, E. Grundy, R. S. Laramée, 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.
- [42] 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.
- [43] 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, 2011.

- [44] 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.
- [45] M. Roth. *Automatic extraction of vortex core lines and other line type features for scientific visualization*. Hartung-Gorre, 2000.
- [46] I. Sadarjoen and F. Post. Geometric methods for vortex extraction. In *Proc. EG/IEEE Visualization Symposium*, 1999.
- [47] 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.
- [48] F. Sadlo and D. Weiskopf. Time-dependent 2-d vector field topology: An approach inspired by lagrangian coherent structures. 29(1):88–100, 2010.
- [49] T. Salzbrunn, C. Garth, G. Scheuermann, and J. Meyer. Pathline predicates and unsteady flow structures. *The Visual Computer*, 24(12):1039–1051, 2008.
- [50] T. Salzbrunn and G. Scheuermann. Streamline predicates. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1601–1612, 2006.
- [51] 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 2008 Proceedings*, pages 75–92. SCS Publishing House, 2008.
- [52] 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.
- [53] H.-W. Shen, C. R. Johnson, and K.-L. Ma. Visualizing vector fields using line integral convolution and dye advection. In *Volume Visualization, 1996. Proceedings., 1996 Symposium on*, pages 63–70. IEEE, 1996.
- [54] K. Shi, H. Theisel, H. Hauser, T. Weinkauff, 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 *Topology-Based Methods in Visualization II*, pages 75–88. Springer, 2009.
- [55] K. Shi, H. Theisel, T. Weinkauff, H.-C. Hege, and H.-P. Seidel. Visualizing transport structures of time-dependent flow fields. *IEEE Computer Graphics and Applications*, 28(5):24–36, 2008.

- [56] P. Skraba, B. Wang, G. Chen, and P. Rosen. 2d vector field simplification based on robustness. In *Pacific Visualization Symposium (PacificVis), 2014 IEEE*, pages 49–56. IEEE, 2014.
- [57] A. Szymczak and E. Zhang. Robust morse decompositions of piecewise constant vector fields. *Visualization and Computer Graphics, IEEE Transactions on*, 18(6):938–951, 2012.
- [58] H. Theisel, T. Weinkauff, 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*, pages 421–428, 2004.
- [59] X. Tricoche, G. Scheuermann, and H. Hagen. Continuous topology simplification of planar vector fields. In *Proceedings of IEEE Visualization 2001*, pages 159–166, 2001.
- [60] M. Üffinger, F. Sadlo, and T. Ertl. A time-dependent vector field topology based on streak surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):379–392, 2013.
- [61] J. J. van Wijk. Image based flow visualization. In *ACM Transactions on Graphics (SIGGRAPH 02)*, volume 21, pages 745–754, Jul 2002.
- [62] C. Wang, H. Yu, and K.-L. Ma. Importance-driven time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1547–1554, 2008.
- [63] J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.
- [64] J. Wei, H. Yu, R. W. Grout, J. H. Chen, and K.-L. Ma. Dual space analysis of turbulent combustion particle data. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 91–98. IEEE, 2011.
- [65] T. Weinkauff and H. Theisel. Streak lines as tangent curves of a derived vector field. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1225–1234, 2010.
- [66] T. Weinkauff, H. Theisel, and O. Sorkine. Cusps of characteristic curves and intersection-aware visualization of path and streak lines. In *Topological Methods in Data Analysis and Visualization II*, pages 161–176. Springer, 2012.

- [67] 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.
- [68] 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.
- [69] J. Woodring and H.-W. Shen. Multiscale time activity data exploration via temporal clustering visualization spreadsheet. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):123–137, 2009.
- [70] 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, 2012.
- [71] E. Zhang, H. Yeh, Z. Lin, and R. S. Laramée. Asymmetric tensor analysis for flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):106–122, 2009.
- [72] L. Zhang, R. S. Laramée, 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.
- [73] L. Zhang, R. S. Laramée, D. Thompson, A. Sescu, and G. Chen. An integral curve attribute based flow segmentation. *Journal of Visualization*, 19(3):423–436, 2016.
- [74] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1(2):141–182, 1997.