

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 16-08-2023	2. REPORT TYPE Thesis or Dissertation	3. DATES COVERED (From - To) -
---	--	-----------------------------------

4. TITLE AND SUBTITLE Tracking and evolution of compressible turbulent flow structures	5a. CONTRACT NUMBER W911NF-20-1-0096
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 611102

6. AUTHORS Jonas Buchmeier	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES University of Southern California University of Southern California 3720 South Flower Street LOS ANGELES, CA 90089 -0001	8. PERFORMING ORGANIZATION REPORT NUMBER
---	--

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSOR/MONITOR'S ACRONYM(S) ARO
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) 75112-WS.3

12. DISTRIBUTION AVAILABILITY STATEMENT

13. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:	17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Ivan Bermejo-Moreno
a. REPORT	b. ABSTRACT	c. THIS PAGE	19b. TELEPHONE NUMBER 213-740-2881

REPORT DOCUMENTATION PAGE (SF298)
(Continuation Sheet)

Continuation for Block 13

Proposal/Report Number: 75112.3-WS

Report Title: Tracking and evolution of compressible turbulent flow structures

Report Type: Ph.D. Dissertation

Publication Type: Thesis or Dissertation

Institution: University of Southern California

Date Received: 16-Aug-2023

Completion Date: 1/23/22 6:05AM

Title: Tracking and evolution of compressible turbulent flow structures

Authors: Jonas Buchmeier

Acknowledged Federal Support: Y

Tracking and evolution of compressible turbulent flow structures

by

Jonas Buchmeier

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(Aerospace Engineering)

January 2022

Acknowledgements

First of all, I would like to thank my advisor, Prof. Iván Bermejo-Moreno, for all his guidance and patience throughout my PhD journey. I felt in good hands while working with Iván. My PhD work builds up on his own PhD on the geometry of turbulence and would not have been possible without it. Also I want to thank him for taking me to the Summer Research Program of the Center for Turbulence Research at Stanford University in 2018, which was quite a special experience and honor.

I would like to thank Profs. Julian Domaradzki, Carlos Pantano-Rubino, Aiichiro Nakano, Paul Newton and Mihailo Jovanovic for agreeing to be in my qualifying exam and defense committee and showing interest in this research. I also want to express my thanks to Prof. Stefan Hickel for his advice during my Master thesis at the Technical University of Munich and Dr. Bernhard Eisfeld of the German Aerospace Center for awakening my passion for fluid dynamics during my Bachelor thesis project.

My deepest gratefulness belongs to my parents, Harald and Ulla, for their endless support and encouragement throughout this journey.

I would like to thank my lab mates Xiangyu, Jon and Naili and also my office mates Arturo, Raye, Ruiyang and Keyvan for all the fun we had in and outside the office. Special thanks goes to Alex for the great collaboration during his stay at USC and his friendship.

This work would not have been possible without financial support received from the Army Research Office (Grant W911NF2010096) and computational resources provided by USC's Center for Advanced Research Computing and an award of computer time provided by the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

Table of Contents

Acknowledgements	ii
List of Tables	v
List of Figures	vi
Abstract	ix
Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Survey of flow feature tracking methods	3
1.2.1 Implicit feature extraction and tracking	4
1.2.2 Explicit feature extraction and correspondence based tracking	5
1.3 Outline of the present work	7
Chapter 2: Flow feature tracking methodology	9
2.1 Introduction	9
2.2 Event definitions in correspondence search	11
2.3 Structure attributes for event detection and rejection	12
2.4 Detection of correspondences	15
2.4.1 Projection of spatial attributes	16
2.4.2 Attribute scaling	17
2.4.3 Nearest-neighbor search in the attribute space	17
2.4.4 Radius search in the spatial subspace	19
2.4.5 Detection of unmatched structures	22
2.4.6 Collecting correspondences form the multi-stage search	22
2.5 Constraining of correspondences	23
2.5.1 Bounding volume constraints	23
2.5.2 Surface proximity constraints	27
2.6 Detection of events	28
2.7 Constraining and optimization of events	28
2.7.1 Event-driven correspondence constraining	29
2.7.2 Combinatorial extraction and conservation-based constraining of subevents	30
2.7.3 Sequence of event realizability constraints	31
2.8 Treatment of periodic boundaries	32
2.9 Graph representation of temporal evolution of flow features	33
2.10 Graph mining for pattern recognition in flow feature evolution	39
2.10.1 Criteria for pattern evaluation	44
2.10.2 Application to a synthetic tracking graph	45

Chapter 3: Passive scalar structures in isotropic turbulence and shock-turbulence interaction	51
3.1 Introduction	51
3.2 Numerical setup and initialization of passive scalar structures	52
3.3 Downstream distribution of events	60
3.4 Structure volumes of event types	64
3.5 Split and merge events in geometrical feature space	66
3.6 Downstream evolution surface-averaged physical quantities	69
3.7 Correlations between local surface geometry and physical quantities on the surface	73
3.8 Trajectories of primary structures in the feature space	76
3.8.1 Geometrical trajectories and surface area evolution	76
3.8.2 Trajectories with scalar dissipation and vorticity	83
Chapter 4: Vortical structures in temporally developing compressible mixing layers	88
4.1 Introduction	88
4.2 Problem formulation and numerical setup	89
4.2.1 Setup of the flow configuration	89
4.2.2 Setup of the tracking algorithm	92
4.3 Validation of self-similarity and growth rate of the mixing layer	93
4.4 Identification, extraction and general statistics of vortical structures	98
4.5 Creations neighboring existing vortices	101
4.6 Detection of hairpin-like vortical structures	104
4.7 Hairpins in the geometrical feature space	110
4.8 Temporal distribution and formation of hairpin-like vortices	112
4.9 Cross-stream and spanwise growth of hairpin vortices	118
Chapter 5: Conclusion	123
Bibliography	126

List of Tables

3.1	Initial configuration of passive scalar structures used in the DHIT and STI configuration	54
3.2	Settings for NN and radius search and correspondence constraining in passive scalar application	58
3.3	Settings of event constraining in passive scalar application	59
3.4	Settings for graph building in passive scalar application	59
3.5	Mean disappearance location, start location and spatial length of split cascade of primary structures .	81
4.1	Grid properties for mixing layer configurations.	91
4.2	Settings for the correspondence search and constraining.	92
4.3	Settings for event constraining and event building in the graph.	92
4.4	Begin, end and growth rate of the self-similar regimes.	93
4.5	Input parameter of the hairpin filter algorithm	106

List of Figures

1.1	Classification of flow feature tracking methods	3
2.1	Overview of the tracking algorithm	10
2.2	Generic shape enclosed by AABB and OBB and illustration of radius search	21
2.3	Illustration of AABB volume and distance constraint	24
2.4	Illustration of separating axis theorem and local surface proximity	26
2.5	Duplication of boundary intersecting or crossing structures	32
2.6	Event reconnection under periodic boundary conditions	34
2.7	Graph representation of structure evolution	35
2.8	Vertex clustering strategies for graph building	37
2.9	Classification of branches in graph	39
2.10	Synthetic graph for graph mining demonstration	41
2.11	Labeled synthetic input graph	42
2.12	Illustration of graph compression and graph isomorphism	43
2.13	Illustration of graph compression and generation of parallel edges	44
2.14	Compressed synthetic graph after the first Subdue iteration	47
2.15	Compressed input graph after the 8-th iteration	48
2.16	Best pattern extracted from a synthetic input graph	49
2.17	Embedding of pattern instances	49
2.18	Embedding and decompression of pattern instances	50
2.19	Embedding and decompression of pattern instances of varying compressed nodes	50
3.1	Initial radial scalar profile and regions of scalar discontinuities	54
3.2	Renderings of passive scalar structure evolution in DHIT	55
3.3	Plane cuts of transitional regions of scalar gradient	57
3.4	Problem setup of passive scalar mixing in STI and blended DHIT precursor simulations	58
3.5	Number of extracted iso-surfaces depending on iso-value	59
3.6	Hierarchical layout of the graph of the $\phi_{4\lambda}$ scalar in DHIT	60
3.7	Hierarchical layout of the graph of the $\phi_{2\lambda}$ scalar in DHIT	61
3.8	Downstream distribution of different event types	62
3.9	Representative compound events of $\phi_{2\lambda}$ structures in the post-shock region	63

3.10	Volume histogram of disappearing ϕ_λ (a) and $\phi_{4\lambda}$ (b) structures in DHIT and STI $M = 1.5$	64
3.11	Joint histogram of volumes V_s of source and V_t targets of continuation events	65
3.12	Joint histogram of volumes V_s of source and V_t of summed targets of split events	66
3.13	Joint histogram of volumes of summed source and summed targets of compound events	67
3.14	Joint histograms of geometrical attributes of source and target structures of split events	67
3.15	Absolute shape index mapped on $\phi_{2\lambda}$ structures in DHIT during split events	68
3.16	Joint histograms of geometrical attributes of source and target structures of merge events	69
3.17	Ensemble mean of the downstream evolution of the scalar gradient magnitude, vorticity magnitude and the alignment between the scalar gradient and vorticity averaged on isosurfaces.	71
3.18	Ensemble mean of the downstream evolution of the scalar gradient alignment with strain-rate eigenvectors averaged on scalar iso-surfaces	72
3.19	Dimensionless curvedness, absolute shape index and the alignment between the eigenvector γ of the strain-rate tensor and the scalar gradient $\nabla\phi_{4\lambda}$ mapped on an isosurface of the $\phi_{4\lambda}$	74
3.20	Alignment analysis conditioned on local surface geometry	76
3.21	Trajectories in the geometric feature space of individually tracked primary $\phi_{4\lambda}$ structures in DHIT	78
3.22	Early and later higher-order split of $\phi_{4\lambda}$ structures in DHIT	78
3.23	Ensemble mean trajectories of primary structures in geometric space for DHIT and STI	79
3.24	Temporal evolution of the normalized surface area of primary structures in DHIT.	80
3.25	Streamwise evolution of the surface area of primary ϕ_λ and $\phi_{2\lambda}$ structures in DHIT and STI normalized by its initial DHIT value	81
3.26	Ensemble mean trajectories of primary structures in geometric space with scalar gradient and vorticity magnitude for DHIT	83
3.27	Ensemble mean trajectories of primary structures in geometrical feature space with scalar gradient and vorticity magnitude (STI $M = 1.5$).	84
3.28	Ensemble mean trajectories of primary structures in geometrical feature space with scalar gradient and vorticity magnitude (STI $M = 3.0$).	85
3.29	Renderings of primary $\phi_{4\lambda}$ and ϕ_λ structures in DHIT	86
3.30	Renderings of $\phi_{4\lambda}$ and ϕ_λ in STI $M = 3.0$	87
4.1	Illustration of the initial condition of the temporally developing mixing layer	90
4.2	Grid specification of the mixing layer configurations	91
4.3	OBB tree presentation of a hairpin vortex	92
4.4	Density xy -planes of the $M_c = 0.7$ mixing layer at different times	94
4.5	Momentum thickness evolution and growth rate	95
4.6	Self-similar streamwise velocity profiles	96
4.7	Individual streamwise turbulence intensity profiles in the cross-stream direction	97
4.8	Ensemble mean profile of streamwise, cross-stream and spanwise turbulence intensities and Reynolds shear stress	97
4.9	Iso-value evolution and number of extracted structures from the mixing layer	99
4.10	Detected event types of Q structures in the mixing layer over time	99

4.11	Temporal evolution of the geometrical signature in the early stage of the $M_c = 0.3$ mixing layer	100
4.12	Selection of Λ -vortices	101
4.13	Alignment between Q-surface normal and vorticity. Cross-stream locations of creation events	102
4.14	Geometrical variety of the extracted Q-structures in the self-similar regimes	102
4.15	Newly created structures and their first event in geometric feature space	103
4.16	Creation-merge pattern in the self-similar regime of the $M_c = 0.3$ mixing layer	103
4.17	Creation and subsequent merge bending a tube-like structures	104
4.18	Creation and subsequent merge forming legs on tube-like structures	104
4.19	Velocity field surrounding an individual hairpin vortex	105
4.20	Large connected hairpins and shape index on individual hairpin	106
4.21	Elements of the hairpin filter algorithm	108
4.22	Structures geometrically similar, but physically dissimilar to a hairpin vortex.	108
4.23	Leg vorticity component - cross coordinate clustering rejections during hairpin filtering.	109
4.24	Examples of detected hooks and arches.	110
4.25	Dimensionless curvedness mapped on hairpins	110
4.26	Hairpin vortices in the geometrical feature space	111
4.27	Geometrical signatures of hairpin-like vortices grouped by mixing layer stage	112
4.28	Temporal distribution of hairpin detections	113
4.29	Connected structures breaking into individual hairpin-like structures	115
4.30	Hooks developing hairpin shape	115
4.31	Short-lived hairpin in the $M_c = 0.3$ mixing layer	116
4.32	Long-lived hairpin in the $M_c = 0.3$ mixing layer	117
4.33	Hairpin in geometrical feature space enhanced by surface-averaged streamwise vorticity component	118
4.34	Lifetime of a structure developing temporarily hairpin-like shape ($\tau = 280 - 376$)	119
4.35	Lifetime of a structure developing temporarily hairpin-like shape ($\tau = 96 - 172$)	120
4.36	Mean cross stream locations of hairpins	121
4.37	Cross-stream extent of center plane crossing hairpins	121
4.38	Cross-stream and spanwise growth of hairpin vortices	122

Abstract

An algorithm is introduced to track the evolution of individual structures in turbulent flows. Flow structures are defined as closed surfaces of any kind in numerical or experimental data sets. To track evolving structures over time and to capture their interactions with each other, correspondences between structures in consecutive time steps are found in a higher-dimensional feature space, which is constructed from differential geometric and spatial structure attributes. After detected correspondences are tested for physical realizability, structure interactions are mapped into a graph data structure which represents the evolution and interactions of all tracked structures over time. The hybrid attribute- and region-based algorithm allows large sampling intervals in the database by capturing complex compound interactions resulting from fast feature dynamics. The tracking graph is queried to retrieve individual and statistical information on the evolution of the structures. Graph data mining techniques are adapted to recognize common patterns in the flow feature evolution.

In a first application, the tracking algorithm is utilized to conduct a structure-based numerical analysis of passive scalar mixing in decaying homogeneous isotropic turbulence (DHIT) and shock-turbulence interaction (STI) canonical configurations. Structures of a scalar field are defined as disconnected isosurfaces of the particular underlying scalar field in shock-capturing direct numerical simulations (DNS) of DHIT, with Taylor microscale Reynolds number $Re_\lambda = 40$ and turbulence Mach number $M_t = 0.2$, and STI cases in which the turbulence is processed by a shock wave at Mach numbers $M = 1.5$ and 3.0 . The analysis focuses on the temporal evolution of ensembles of passive scalar structures, initialized as collections of uniformly spaced spheres of varying scales, commensurate with the initial Taylor microscale, λ , of the underlying turbulence. Changes in the feature surface geometry are related to the underlying physical processes relevant to the mixing (turbulent transport, scalar dissipation, and shock compression). Temporal surface convolution increases for initially larger structures, resulting in a higher probability of locally hyperbolic geometries where break-up into smaller structures occurs. Shock-induced deformation of the structures amplifies breakup processes, enhancing mixing, particularly for larger structures. Mixing enhancement by the shock is manifested as an amplification of the surface-averaged scalar gradient, which increases for initially larger structures. The alignment between the scalar gradient and the most extensional strain-rate eigendirection on the scalar isosurfaces also increases across the shock. Larger magnitudes of the scalar gradient and its alignment with the most compressive strain-rate eigendirection correlate with flatter surface regions. Shock-induced structure compression increases the area coverage of flat regions, where the amplification of scalar gradient is localized.

In a second application of the tracking algorithm, the inhibiting effect of compressibility on the turbulence intensities in temporarily developing mixing layers is analyzed from a structural viewpoint. In the configurations of initial

momentum thickness Reynolds number $Re_{\theta,0} = 160$ and varying convective Mach numbers $M_c = \{0.3, 0.7, 1.1\}$ the dynamics and geometrical properties of vortical structures defined by iso-surfacing the Q-criterion field are investigated. Using a time-dependent iso-value based on statistics of the Q-field, less vortices are extracted with increasing convective Mach number, expressing the inhibiting effect of compressibility on the structural level. Higher compressibility lowers the curvedness of the extracted Q-structures. The creation of vortices in the neighborhood of existing vortices and the subsequent merging of both vortices is recognized as a prominent pattern of the vortex dynamics. Other patterns of statistical significance are split-dominated or represent the short-lived evolution of isolated structures which disappear shortly after their creation without interacting with others. To evaluate the occurrence, persistence and significance of hairpin-like vortices in the mixing layers, a detection method for such special vortices is introduced. For all convective Mach numbers, the number of hairpin detections peaks at $Re_{\theta} \approx 400$ in the pre-self-similar stages of the mixing layers and decays afterwards, suggesting that hairpins are not a dominant flow feature once the turbulent background is developed. The time span in which vortices preserve their hairpin-like shape varies, but most hairpins are rather short-lived as they cross the center planes of the mixing layers and are partially pulled into the opposite free streams. The majority of individual hairpins is formed by the structural breakup process of largely connected hairpins which originate from Λ vortices which grow in the early stage of the mixing layer through merging of tube-like vortices. A number of hairpins have a tilted orientation with respect to the center plane, especially in the stages after the pre-self-similar regime. For hairpins that have quasi-streamwise orientation cross-stream and spanwise growth of equal rates occurs simultaneously.

Chapter 1

Introduction

1.1 Motivation

Despite its chaotic, multi-scale character, organized motions and turbulence structures have been observed in turbulent fluid flow for a long time. Hussain (1986) links the motivation to study coherent structures to a desire to find order in apparent disorder. Although ambiguous definitions of what a coherent structure is can be found in the literature, there is an intuitive understanding that coherent structures are associated with motions that have some sort of spatial organization and temporal coherence leading to a certain lifetime of the coherent structure in which its appearance actively or passively evolves. Complementing statistical approaches to study the nature of turbulent flow, structural analyses have provided fundamental insight into the mechanisms of momentum and energy transfer in turbulence. For example, in the buffer layer of near-wall turbulence streaks of streamwise velocity are found to carry most of the turbulent kinetic energy (TKE) (Kline *et al.*, 1967; Smith & Metzler, 1983) and quasi-streamwise vortices organize dissipation and Reynolds stresses (Jiménez, 2013), whereas in the logarithmic layer, most of the momentum transfer originates from wall-attached eddies (Townsend, 1961) and vortices organize into self-similar clusters (Del Alamo *et al.*, 2006). Structural analyses of turbulent flows have led to the development of structure-based models of turbulence fine scales (Townsend, 1951; Tennekes, 1968; Lundgren, 1982; Pullin & Saffman, 1993) and subgrid-scale models for large eddy simulations (Misra & Pullin, 1997).

To gain further insight into the structural mechanisms of turbulent flow, different description techniques for the geometrical properties of turbulence structures have been suggested in the literature (Bermejo-Moreno & Pullin, 2008; Leung *et al.*, 2012). By utilizing these techniques the geometry of an individual turbulence structures can be related to the underlying flow physics. The structural approach can be expanded from turbulence structures to other flow features such as droplets in multiphase flows, transition zones or shock waves. Silver (1995) and Post *et al.* (2002) summarize different techniques to define, extract and visualize features from a flow field. Since flow features are an abstraction of the original data and typically occupy a relatively small region of the computational domain, their extraction in general leads to a reduction of data, as emphasized, for example, by Van Walsum *et al.* (1996). Extracted features can be isolated from the overall data set, which simplifies their individual exploration compared to visualizing the entire data set at once.

Time-resolving computational and experimental techniques to study fluid motion, such as direct numerical simulation (DNS) or time-resolved three-dimensional particle image velocimetry (3D-PIV), have granted access to instantaneous, three-dimensional flow fields and hence to the dynamics of turbulence structures. Whereas the aforementioned geometrical techniques describe turbulence structures for instantaneous snapshots of a given data set, the present work focuses on the evolution of those geometrical properties over time and the relation to the temporally or spatially evolving physics of the flow field. Therefore, a novel method to track individual flow structures throughout the flow field is introduced in §2 adding temporal coherence to the geometrical analysis. The method enables simultaneous tracking of intertwined structures with significant concave regions over large sampling intervals, by combining region- and attribute-based algorithms. The tracking attributes incorporate geometric properties of the structures obtained from differential geometry analyses.

In our present tracking methodology, flow features are defined as closed surfaces of any kind embedded in a three-dimensional space, but could be generalized to other dimensions (e.g., flow features in two-dimensional turbulence). In principle, the surfaces could be derived from a numerical simulation, as well as originate from experimental data. In this work, the surfaces are extracted from volumetric scalar fields by isosurfacing. However, we emphasize that this flow feature extraction is extrinsic to the tracking itself, as the tracking is explicit, see §1.2.2. The algorithm tracks individual, disconnected isosurface components of the chosen field. The polygonal mesh discretization of the isosurface is obtained by applying the marching cube algorithm (Lorenson & Cline, 1987) followed by a disconnection step to isolate individual isosurfaces. Features, structures and isosurfaces are used synonymously in this work. The shape of the structures, characterized from their differential geometry properties, plays a key role in the presented tracking methodology. In addition to its contribution to the tracking effort via geometrical attributes used in the correspondence search, we aim to relate the temporal development of the geometry of the structures to the underlying flow physics. The method is designed to track all structures extracted from the flow simultaneously over large sampling intervals, which pose challenges to existing tracking methods due to the increased inter-frame evolution. To study the evolution of individual structures and their mutual interactions, the tracking results are encoded in a graph data structure, which is a representation of the evolution of all tracked features.

In a first application presented in this work the proposed tracking method is used to investigate passive scalar mixing in compressible turbulence from a structural point of view. Adding to findings based on statistical analyses, the tracking of individual finite-sized scalar structures enables to investigate how the scalar mixing is represented at the structural level in the form of the structural breakup process and the structure lifetime until complete diffusion. Physical quantities relevant to the mixing, such as the magnitude of the scalar gradient and its alignment with eigendirections of the strain-rate tensor, can be mapped on the tracked surfaces and hence are also examined from a structural viewpoint. The tracking application and geometrical analysis allows to study how the shock-induced structure deformation amplifies the structural mixing process. In particular, the temporal geometrical analysis of the scalar structures permits to identify the predominant shapes of increased scalar dissipation as the structures pass through the shock wave.

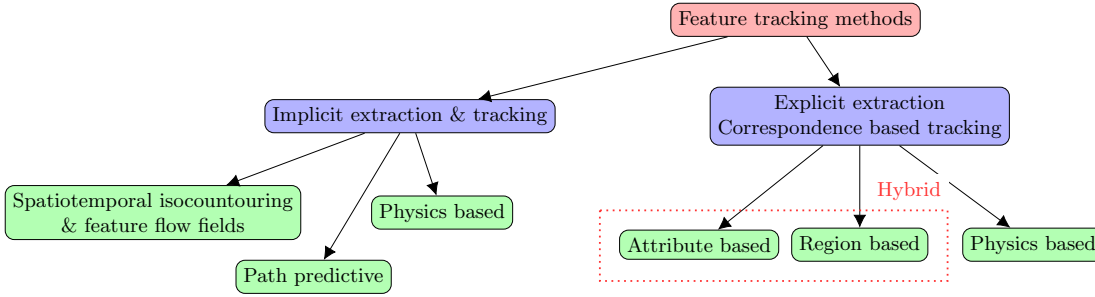


Figure 1.1: Classification of flow feature tracking methods and proposed hybrid approach of attribute and region based tracking.

Another flow configuration addressed in this work is the mixing layer formed between two parallel streams. Statistical analyses show an inhibiting effect of compressibility on the turbulence intensities in the mixing layer (Papamoschou & Lele, 1993). By utilizing the proposed tracking algorithm the effect of compressibility investigated in regard to the dynamics and geometrical properties of individual vortical structures in the mixing layer. Despite of their proposal in the 1950s, the occurrence, persistence and relevance of hairpin vortices in shear flow configurations (such as the mixing layer) is still subject to ongoing debate. Therefore, a filter is proposed to detect hairpin-like structures during the development of the mixing layer. After identifying structures which are temporarily of hairpin-like shape, their individual evolution and geometry is examined. The tracking allows to study how hairpin vortices are formed and how they persist in the developing turbulent background of the mixing layer.

1.2 Survey of flow feature tracking methods

The features of a time-depending flow field may have evolving characteristics such as their spatial location and shape. Analyzing the dynamics of flow features in physical space is subject to feature tracking. Different feature tracking techniques have been introduced in the literature that differ on how the features are defined and extracted from the flow field. A classification of feature tracking algorithms is presented in figure 1.1. In *explicit* feature tracking the extraction of features is conducted as a pre-step to the tracking itself. Because the extraction is performed independently for each time instant (frame) and the temporal coherency of each feature is not utilized in the extraction process, the *correspondence problem* between features in successive frames arises. Solving the correspondence problem means finding the relation between feature in successive frames. For example, if two features extracted from successive frames are matched in a one-to-one correspondence, the two features represent the same feature at the two different time instants. Without this correspondence information the individual feature evolution cannot be analyzed. In general, each feature can have multiple forward and backward correspondences which adds significant complexity to the correspondence problem. Avoiding the correspondence analysis between frames is one motivation for *implicit* feature tracking methods, which either apply a spatio-temporal feature extraction rather than a pure spatial extraction (as in explicit tracking) or use a prediction in the extraction process based on features extracted in previous frames, and hence also include temporal coherence in the feature extraction process.

1.2.1 Implicit feature extraction and tracking

Weigle & Banks (1998) and Ji *et al.* (2003) present implicit algorithms to track time-varying isosurfaces and interval volumes by isocontouring in higher dimensions across the entire spatio-temporal domain. Assuming that corresponding \mathbb{R}^3 isosurfaces from two consecutive time steps overlap with each other, isocontouring in \mathbb{R}^4 is used to find correspondences implicitly. Given a \mathbb{R}^3 isosurface at time t_n the overlapping isosurface at time t_{n+1} is found by computing the connected isosurface in \mathbb{R}^4 that intersects with the given \mathbb{R}^3 isosurface and then slicing it along the time dimension to get the overlapping \mathbb{R}^3 isosurface at t_{n+1} . Hence the correspondence problem for isosurfaces in \mathbb{R}^3 is implicitly solved. To reduce the computational cost of isocontouring in \mathbb{R}^3 and \mathbb{R}^4 during the tracking, in a follow-up work, Ji & Shen (2004) used a precomputed correspondence table between isosurface components for any possible isovalue. Then, in the actual tracking, the correspondence search reduces to look-up operations from this pre-computed table. The construction of the look-up table is based on the observation that the correspondence relationship between \mathbb{R}^3 isosurfaces in consecutive time steps can only change at critical isovalues. At critical isovalues the number of isosurface components in \mathbb{R}^4 changes, since the temporal correspondences change.

The tracking method proposed by Bajaj *et al.* (2002) is based on generating the isosurfaces in consecutive time steps progressively from the previous time step by performing a temporal contour propagation of existing isosurfaces and seed generation of new components, instead of extracting isosurfaces independently from different time steps. Sohn & Bajaj (2005) construct the topology of time-varying isosurfaces resulting in a time-dependent contour tree. Correspondences are computed between two instants of the contour tree. Topological events, such as the split of one isosurface into multiple ones or the merge of multiple isosurfaces into one, are represented in a topology-change graph.

All of the aforementioned implicit spatio-temporal tracking methods are based on a feature-overlapping criterion, equivalently found in a certain class of explicit methods, as described later. Theisel & Seidel (2003) introduced an alternative approach to represent the time-varying behavior of features not as a higher-dimensional isosurface but as stream lines or surfaces of higher-dimensional vector fields. This approach of “feature flow fields” is applicable to local flow feature and was tested, in particular, to track critical points of 2D and 3D vector fields. Even though in feature flow fields no explicit correspondence matching is performed, features may not be connected over time by the streamlines of the feature flow field, if the spatial displacement between the features in consecutive time steps is too large and, therefore, an overlap criterion is somewhat implied. Theisel *et al.* (2005) apply the concept of ‘feature flow fields’ to extract and track vortex core lines in a formulation based on ‘parallel vectors’ as introduced by Peikert & Roth (1999). The concept of parallel vectors derives two vector fields from the original vector field, such as itself and its curl, and a vortex core is located where these two vector fields are parallel. Streamline/surface integration of the feature flow field is used to extract/track the parallel vector lines. Another implicit tracking method for line-type features was proposed by Bauer & Peikert (2002) to track vortex cores through different spatial and temporal scales.

Rather than utilizing a spatio-temporal extraction process, Banks & Singer (1995) apply a vorticity-predictor pressure-corrector scheme to reconstruct and implicitly track elongated vortex tubes in turbulent flow. Muelder & Ma (2009) also propose an implicit tracking method in a prediction-correction manner. Assuming the features have

fairly predictable paths, prediction methods of different order are used to estimate the feature region in a frame based on the previous frames. In a correction step the estimated feature region is either shrunk or grown to extract the actual feature in the current frame. Hence, the method assumes an overlap between the estimated and actual feature region.

In general, low temporal resolution in the flow field constitutes a challenge to tracking its features. To track volumetric flow features, especially over large sampling intervals, Sauer *et al.* (2014) proposed a method using trajectories of corresponding Lagrangian particle data to predict the location of the feature in the next available time instant.

Another particle-based, path-predictive tracking method was applied by Schafhitzel *et al.* (2008) on segmented vortex structures to study perturbation events on arbitrary temporarily tracked vortex structures. The vortex core line is tracked in a predictor-corrector manner by placing particles on the core line. The particles are integrated along a given unsteady vector field and tested against overlapping with core lines in the consecutive time step.

1.2.2 Explicit feature extraction and correspondence based tracking

In contrast to the described implicit tracking methods, the feature extraction step is a pre-step to the actual tracking effort in explicit tracking. As mentioned earlier, in explicit tracking the correspondence problem arises due to the exclusion of temporal coherency in the feature extraction. Explicit tracking is therefore also called correspondence-based tracking and related methods can be divided in three categories based on how the search for correspondences is conducted: physics-, region- and attribute-based correspondence search. Another way of classification is distinguishing between local and global tracking techniques as described later on.

Since the features are extracted from the flow field, their dynamics are in principle governed by the same equations as the fluid motion. Clyne *et al.* (2012) proposed a feature path prediction method which is based on the physics of the underlying solutions to the equation of fluid motion. To start the path prediction, for each feature, the point of minimum dissipation is found and advected by the velocity field to the consecutive frame. A streamline passing through this advected point is integrated in forward and backward directions. Possible candidates for correspondence with the original feature are features in the consecutive frame which intersect the streamline. The best match is found by minimizing the relative volume difference between candidate and original features.

Early work on region-based correspondence search for explicit tracking was conducted by Silver & Wang (1996, 1997, 1998). The main motivation of the region-based search is based on the observation that, if the sampling frequency is high enough, corresponding features in consecutive time steps usually have a spatial overlap. To track one feature, first, all overlapping features in the consecutive frame are found. To select the best match from those candidates a degree of overlap is computed in terms of a normalized volume-difference test. This approach has been extended by Chen *et al.* (2003) to be applied to feature tracking in data sets obtained by application of adaptive mesh refinement (AMR) techniques, whereby each feature can span multiple refinement levels. A “feature tree” is obtained representing the feature in multiple levels of refinement.

A tracking method based on region-based correspondence was applied to track vortical structures in forced and decaying isotropic turbulence by Kareem *et al.* (2007) and Abdel-Kareem (2011). Lozano-Durán & Jiménez (2014) use region-based correspondence search to study the evolution of coherent structures in turbulent channels, organizing the found correspondences over time into a graph that represented temporal structure interactions. Another region-based tracking method was proposed by Saikia & Weinkauff (2017), where feature regions are defined by merge trees that, in general, segment a scalar field into different regions. The similarity between two subtree regions in consecutive frames is measured by considering the volume overlap and the ξ^2 distance between voxel intensity histograms of two regions. A graph data structure is built over all time steps, where the nodes are the subtree regions in each time step and the edges represent all possible connections between regions. The edge weight is defined by the combined distance from the volume overlap and the histogram distance. Since the graph incorporates connection information for all regions over all time steps, it can be used to make tracking decisions globally instead of considering two consecutive time steps. By applying Dijkstra’s shortest path algorithm, the tracking path for each feature is determined. A similar approach, using a graph to obtain tracking paths, was used by Widanagamaachchi *et al.* (2012), which also utilizes merge trees for the definition of regions. In this method the features contained in a track (branch) of the graph have similar intensity levels, whereas in Saikia & Weinkauff (2017) the intensity level of a region of the merge tree can change over time. The same method was later adapted by Widanagamaachchi *et al.* (2015) to track features in embedded surfaces with application to turbulent combustion.

As mentioned earlier, region-based methods rely on the spatial overlap of feature in consecutive frames. Hence, they require a fairly high sampling frequency, especially when small, fast moving features are present in the data. In contrast to the region-based feature tracking methods, attribute-based methods try to solve the correspondence problem by considering only a small set of abstract attributes describing the feature. Therefore attribute-based matching costs significantly less memory and a correspondence search in multiple stages becomes affordable. The feature attributes are utilized to uniquely represent each feature in the search for correspondences. Therefore, the choice of attributes used in this representation is fundamental and may depend on the application. To solve the correspondence problem based on an attribute representation of the features, matching criteria need to be established for the attributes of source and target feature.

Samtaney *et al.* (1994) introduced a list of evolutionary events in the lifetime of a feature: continuation, creation, disappearance, split and merge, see §2.2. To detect those events an attribute-based method which uses the feature centroid as the primary attribute in the search is proposed. In a first step, continuing features are determined by finding the closest feature in the consecutive frame in terms of the centroids. The matched candidates are tested with respect to additional attributes, such as second order moments of inertia, volume and mass. In a second step remaining unmatched features are tested for split and merge events. Feature evolution over multiple frames and interactions between features in the form of the detected events are represented in a directed acyclic graph. A similar tracking method based on centroid location and volume as key attributes was adapted by Chan *et al.* (2018) to track bubbles in breaking waves.

Reinders *et al.* (1999) and Reinders *et al.* (2001) propose an attribute-based method in a prediction-verification scheme. Their attribute set contains centroid position, volume, mass and the best fitting ellipsoid (in terms of position, scaling, and orientation) of each feature. The prediction utilizes a linear extrapolation using the previous two frames and is applied in forward and backward directions. Hence, multiple passes are executed for each frame.

The previously described tracking methods make the common assumption that the feature motion in between consecutive frames is limited and fall into the category of local tracking techniques. This assumption is fairly obvious in terms of the region-based methods, since they rely on the spatial overlap. The attribute-based method of Samtaney *et al.* (1994) also tests the closest centroids in consecutive frames for continuation events. The prediction-verification scheme in Reinders *et al.* (2001) makes the assumption that features evolve predictably. Hence, it also implies certain restriction on the feature evolution between frames. Contrary, global tracking techniques do not impose any restrictions on feature motion between frames. In other words, features may move anywhere in the domain in between frames. Hence, these methods do not perform the tracking of each feature independently with local search approaches. In a global tracking algorithm all the features from two consecutive frames are considered together and a globally best match is pursued. The technique introduced by Ji & Shen (2006) seeks the globally best match by using the Earth Mover’s Distance (EMD), which is also known as Wasserstein metric, as a matching cost measure and propose a “branch-and-bound” method to find the global minimal cost efficiently. EMD is a statistical tool used to measure the distance between two probability distributions over a region. However, although in principle no restrictions are applied, “the cost algorithm is weighted heavily in favor of features that are in close proximity to the tracked feature” as pointed out by Clyne *et al.* (2012). The Wasserstein metric was also used by Soler *et al.* (2018) to track topological features in time-varying scalar data.

Caban *et al.* (2007) introduce a non-local tracking method which is attribute-based with no bias toward nearby features. It is based on the observation that volume segments in time-varying data have textural features that are characteristic to that particular subvolume. Here a feature is associated with a subvolume of the domain. The texture of a subvolume is defined by 26 textural attributes. Correspondences are found in terms of the best texture match in subsequent time steps. First, a tracking window is estimated based on the dynamics between frames in the data set. For data sets where little is known about those dynamics, the window can be large and, in principle, span the entire time domain. Then, similarity measurements in terms of texture are conducted for candidate subvolumes. Since correspondences are found based on textural properties, the best match is not necessarily found in the features spatial neighborhood.

1.3 Outline of the present work

The flow feature tracking algorithm proposed as a local, *hybrid* approach combining attribute- and region-based principles is described in §2. After outlining the algorithm in §2.1, the different types of flow structure interactions which need to be captured by the algorithm are defined in §2.2. §2.3 presents the structure attributes used to abstract flow features in the methodology. The multi-stage search for structure correspondence candidates is described in §2.4.

The constraining process of obtained correspondence candidates is addressed in §2.5. §2.6 describes how candidates for complete events are formed from the constrained structure-to-structure correspondences, while the constraining and optimization procedure for event candidates is detailed in §2.7. A strategy to treat periodic domain boundaries in the context of flow feature tracking is proposed in §2.8. The design of the graph data structure storing the accepted flow feature interactions over time is explained in §2.9. §2.10 introduces a graph mining approach to extract common patterns of flow feature interactions from the tracking graph.

The introduced algorithm is applied to track flow features extracted from scalar (i.e., single-component) fields governed by different physical mechanisms. §3 investigates the temporal evolution of passive scalar structures with well-defined initial geometry in canonical homogeneous isotropic turbulence and shock-turbulence interaction flow configurations. The motivation and numerical setup of this flow configurations is given in §3.1 and §3.2. The temporal (DHIT) and downstream (STI) distribution of detected events is presented in §3.3. Structure volumes associated with different event types are shown in §3.4. Split and merge events are investigated in geometrical feature space in §3.5. §3.6 presents the downstream evolution of surface-averaged physical quantities relevant for the mixing process. Correlations between the local surface geometry and those physical quantities are highlighted in §3.7. The evolution of primary structures is presented in §3.8.1 focusing on geometrical feature space trajectories, surface area evolution and relation to scalar dissipation and vorticity on the surface.

In a second application, motivated in §4.1, the flow feature tracking methodology the evolution of vortical structures is studied in temporarily evolving compressible mixing layers, for which the problem formulation and numerical setup is described in §4.2. The mixing layer simulations are validated in §4.3. §4.4 addresses how vortices are extracted from the mixing layer and presents their general statistics. By utilizing the graph mining technique introduced in §2.10, §4.5 analyses merge events of structures newly created in close proximity of existing vortices. A detection method for individual hairpin-like vortices is presented in §4.6. The geometrical signature of detected hairpin-like structures is investigated in §4.7. The temporal distribution, and hence the relation to the Reynolds number, of hairpin-like structure detections is discussed in §4.8, which also investigates the formation of individual hairpins. The analysis of the dynamics of vortical structures in mixing layer concludes with a discussion of the cross-stream and spanwise growth of hairpin-like structures in the pre-self-similar stage of the mixing layer. A conclusion reflects the thesis in §5.

Chapter 2

Flow feature tracking methodology

2.1 Introduction

We propose a tracking algorithm that follows the idea of explicit, correspondence-based tracking. Hence, the extraction of structures from each given time instant of the flow field and the generation of attributes describing those features are performed as a pre-step to the actual tracking. This extraction and characterization step is conducted for each time frame independently and described in §2.3 in detail. The exclusion of temporal coherence in the feature extraction leads to the correspondence problem, which is the starting point of the tracking effort. The proposed tracking algorithm can be organized in three main blocks: 1) correspondence search & constraining; 2) event formation & constraining; and 3) dynamic graph building. While we seek to solve the correspondence problem in first block of the algorithm, the event formation and, especially, graph building go beyond the solution of that correspondence problem. The three blocks are carried out in sequence for two consecutive frames and repeated in a time loop for which the sampling interval length does not have to be constant, as illustrated in figure 2.1(a).

After the structures have been extracted and their attributes are generated, a frame-to-frame correspondence search is conducted in multiple stages utilizing different subsets of the structure attribute set. The search process involves a higher dimensional neighborhood in the attribute space and, optionally, a radius search in the spatial attribute subspace. Each search is conducted in the forward direction from source to target frame and also backwards from target to source as described in detail in §2.4. The outcome of the search is a list of potential structure-to-structure correspondences. The proposed search method has its key counterpart in a constraining process by which the found correspondences are tested for physical realizability on a structure-to-structure correspondence level (see §2.5). The algorithm allows for flexibility in terms of the types and order of the correspondence constraints. In general, the correspondence constraining can be based on structure attributes, regions, and/or physical laws. However, all correspondence constraints here discussed are region-based, adding the hybrid character to our algorithm. Contrary to existing tracking approaches, the region-based constraining presently introduced does not rely on spatial overlap between source and target structures. The structure region information is used to constrain what has already been found and is not meant to be utilized in the search.

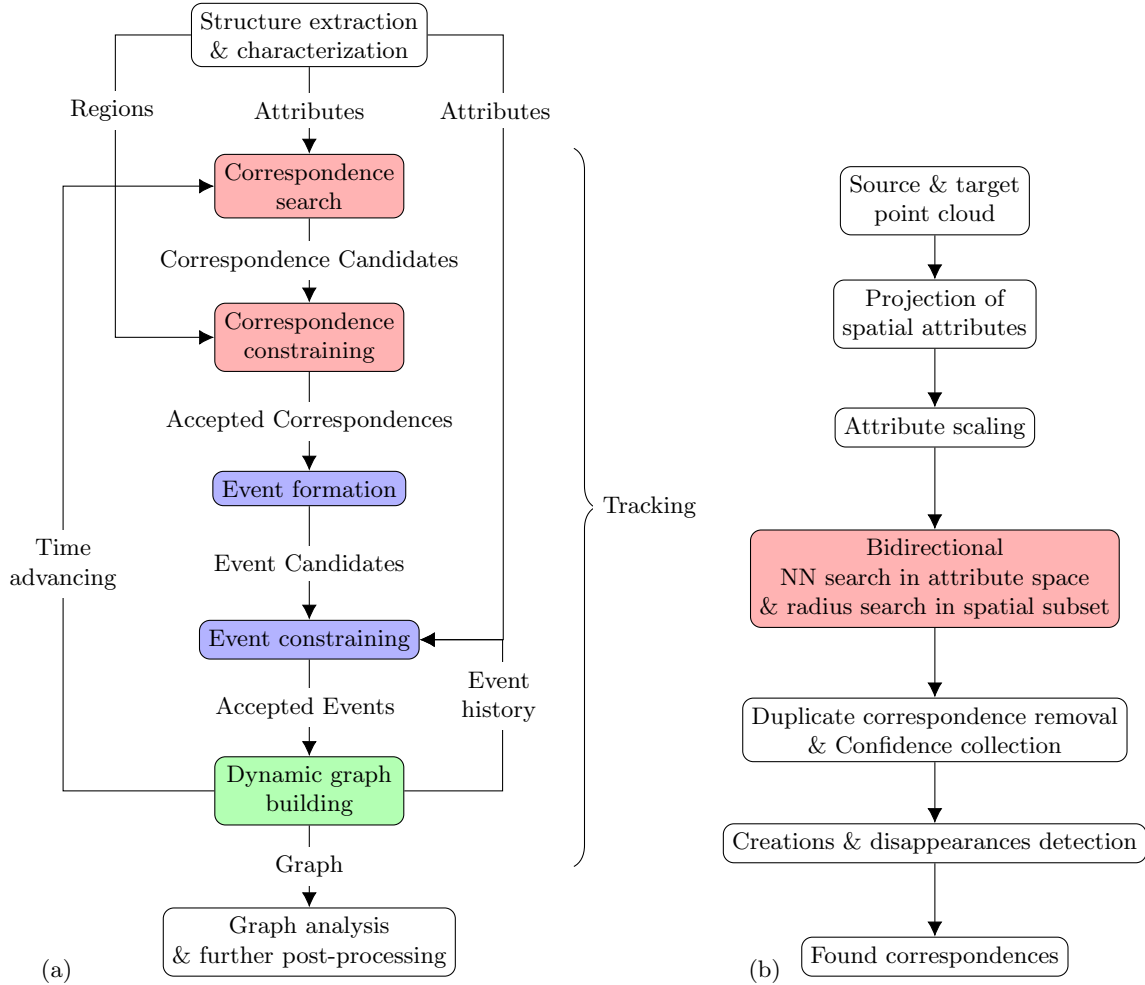


Figure 2.1: (a) Overview of the tracking workflow identifying three main blocks: correspondence search & constraining, event formation & constraining, and dynamic graph building. (b) Overview of the correspondence search process with the bidirectional nearest-neighbor search in the full attribute space and the radius search in the spatial subspace as its central element.

As described in 2.2, only continuation events consist of a single structure-to-structure correspondence. Multi-correspondence events, such as split and merge, are formed based on the structure-to-structure correspondences that pass the constraining process, 2.6. The formation of potential events is adding information that can be used to further test for physical realizability. Again, in principle the event constraining is open to be performed on a attribute, region or physics basis. In contrast to the implemented correspondences constraints, all implemented event constraints are attribute-based, §2.7. Similar to the correspondence constraints, the algorithm allows flexibility regarding the type and order of the event constraints. Depending on the application, in a sequence of event constraints the same event constraint type may be applied to different event types with different tolerances. The event constraining process often leads to significant event simplifications, emphasizing again that the proposed search and constraining process needs to be seen as a unity.

The accepted events are used to dynamically extend a graph data structure at the end of each tracking step (§2.9). The graph is directed and acyclic indicating the progressing tracking time, its vertices represent structures at

a certain time instant and its edges display the found structure to structure correspondences between successive time steps. The building of multi-correspondence events requires a decision making process in terms of the continuation, creation and termination of branches which is based on the overall confidence that was assigned to correspondence in the search and structure properties, such as lifetime and volume. The entire graph represents the lifetime and interactions of all structures present in the flow field over time. Each branch of the graph represents the evolution of one individual structure. Hence, the graph can be queried to further post process the individually tracked structures. Even though the main purpose of the graph is to store the found structure evolution and interactions over time, the event history information contained in the graph can also be utilized in the event constraining of subsequent tracking steps.

Periodic boundaries are commonly found in numerical datasets from which structures are extracted (e.g., instantaneous volumetric snapshots / frames obtained from simulations of turbulent flows). An adequate handling of domain periodicity poses additional challenges to the development of the tracking algorithm, as it affects not only the structure extraction pre-step, but also both the correspondence and event components. To provide an overview of the tracking process, the treatment of periodic boundaries in the tracking is not shown in figure 2.1(a). This aspect of the algorithm is discussed in section 2.8. The modular character of the algorithm makes certain steps optional and depending on the application and purpose of the tracking. In addition, new types of realizability constraints which may be needed for new application can be added easily and without modification of the architecture.

The presented algorithm requires user input in form of search parameter and constraint settings. To tune these settings for a specific application, the algorithm offers several forms of visualization which provide feedback about the accuracy of the detected feature interactions to the user. Visualizing the feature interaction can be seen as a first verification step that delivers useful insight into how the search and constrain parameter need to be adjusted to obtain reasonable tracking results for the specific application.

2.2 Event definitions in correspondence search

To track an evolving structure over time the structure needs to be identified in a series of consecutive time instants (frames). The sampling interval (or tracking time step) between consecutive frames may not be constant, but is assumed to be sufficiently small to match structures in consecutive frames and capture interactions between them. In the matching process, structures in a frame at time t_n are referred to as ‘source’ structures, while structures in the consecutive frame t_{n+1} are called ‘target’ structures. Samtaney *et al.* (1994) introduced a classification of events that describe interactions of structures within a sampling interval as follows:

1. Continuation: A structure continues from one frame to the next without interacting with another structure. The source structure has only one forward correspondence.
2. Creation: A new structure appears in the target frame. The target structure has no backward correspondence.

3. Split: A source structure fractures into two or more target structures. The source structure has multiple forward correspondences.
4. Merge: Two or more source structures merge into one target structure. The target structure has multiple backward correspondences.
5. Disappearance: A source structure disappears. The source structure has no forward correspondence.

While a continuation is a one-to-one correspondence, split and merge events may be seen as a group of one-to-one correspondences. A generalized event can be defined as a collection of one-to-one correspondences. The order of a split or merge is defined as the difference between the number of target and source structures. The complexity of events is related to the sampling interval. For example, as the sample interval length is increased, a cascade of two first-order splits would eventually be detected as a single second-order split. A reduced sampling frequency decreases the amount of data which needs to be stored for the tracking application. Since our tracking algorithm aims to be applicable to datasets of large sampling intervals, we add a generalized event type, that combines splits and merges of multiple order, to the list of events:

6. Compound event: Multiple source structures merge and split into multiple target structures.

While compound events have been neglected by most tracking algorithms found in the literature, Guo *et al.* (2015) reported the observation of compound events during the tracking of magnetic flux vortices. A higher temporal resolution is needed to decompose compound events into regular events.

The goal of the tracking is to generate a list of those events for each individual structure when the matching process is repeated over a sequence of frames. This way the temporal evolution of each structure and its interactions with others are recorded. In addition to these events from one frame to the next, events occurring over several frames as combinations of the previously defined frame-to-frame events may be defined. For example, a merge of a structure with another structure from which it was previously split may be seen as a ‘reconnection event’ and a sequence of split events can be seen as a ‘cascade split event.’ However, the occurrence of those multi-frame events highly depends on the sampling frequency. For example, a reconnection event (split and merge) may be seen as a continuation if the sampling frequency is lower. In contrast to the frame-to-frame events that need to be detected by the tracking algorithm, multi-frame events do not have to be taken into account in solving the correspondence problem and are subject to the graph analysis following the tracking.

2.3 Structure attributes for event detection and rejection

In attribute-based correspondence search, correspondences between structures in successive time steps are based on the comparison of attributes of the structures. An attribute of a structure is a quantitative property, such as its size or orientation. A group of attributes is a fingerprint used to identify a structure, in other words, to distinguish a structure from other structures in the same time instant. In general, the fingerprint of a structure may change from

one time instant to the next. Different event types affect the attributes of a structure from one time instant to the next in different ways. For example, in most cases, a split event may affect the size of the structure more significantly than a continuation event. Since the unique description of a structure at each time instant is an essential part of the tracking, the choice of attributes to form the fingerprint of the structure is a key element.

Since a good fingerprint separates structures, including the spatial location of a structure in the group of attributes is a natural choice. The location of a structure can be represented by a number of attributes. The full information about the spatial region occupied by a structure is contained in the spatial coordinates of all points forming the isosurface. In the present algorithm, this spatial information is reduced to three attributes that are reserved to represent the location of the structure. Depending on the application different choices for those three attributes may be made. Candidates to represent the spatial information of a structure in three attributes can be based on solely the geometry of the structure, such as the three coordinates of the center of the bounding box of the structure, or including other properties such as its centroid. In all applications of tracking algorithm presented in this work, we use the center of either the axis-aligned or oriented bounding box of a structure to define its location. Note that, although attributes like the center of the bounding box are meant to describe the position of the structure, the location defined by those three attributes is not always within the actual boundaries of the structure.

In addition to the spatial location, the shape of a structure can be used to distinguish the structure from other structures in the time instant. Similar to the three attributes representing the location of a structure, we are looking for a reduced set of geometrical attributes that classifies the geometry of isosurfaces. Different geometry description techniques have been suggested in the literature. Geometrical descriptors based on Minkowski functional have been used by Leung *et al.* (2012). In this study, we apply the geometrical characterization method introduced by Bermejo-Moreno & Pullin (2008). From the three main steps of this methodology, extraction, characterization and classification, we do not make use of the latter one. Starting from a three dimensional scalar field at a time instant the extraction of structures begins with a multi-scale decomposition of the given fields into component fields representing different scales of the original field. This first step is not applied in our study and we reduce the extraction method to an isocontouring of the original field. Disconnected regions of the isosurfaces extracted from a volumetric scalar field define the individual structures to be tracked over time. Thus, the structures subject to the tracking are disconnected closed isosurfaces of the original field. Since the geometrical characterization and hence the tracking is limited to closed surfaces, it is essential to reconnect and close isosurfaces that intersect periodic domain boundaries, if those are present in the simulation. Structures intersecting non-periodic domain boundaries are not closed, however, may still be included in the geometrical analysis and tracking attempt by closing them either with simply using the boundary that they intersect or by closing them using their mirrored extension, Demanet & Ying (2007).

The geometrical characterization of each individual isosurface starts with the computation of two differential-geometry properties at each point of the surface. Following Koenderink & Van Doorn (1992) the absolute value

of the shape index S and the dimensionless curvedness C are locally determined based on the principal curvatures $\{\kappa_1, \kappa_2\}$ as

$$S = \left| -\frac{2}{\pi} \arctan \frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2} \right| \quad C = 3 \frac{V}{A} \sqrt{\frac{\kappa_1^2 + \kappa_2^2}{2}} \quad (2.1)$$

, where the structure volume V and area A are used for a non-dimensionalization of the curvedness. Higher values of the curvedness correspond to lower values of the radius of curvature and hence a stronger locally curved surface. The absolute shape index classifies the local shape of the surface as hyperbolic $S < 0.5$, parabolic $S = 0.5$ or (concave or convex) elliptical $0.5 < S < 1$. Extreme values of S are associated with a symmetrical saddle point $S = 0$ and an umbilical point $S = 1$, while the shape index is undefined for a planar surface. Since the attribute representation of each structure is based on attribute of different types, the value range of each attribute needs to be considered and a scaling operation into a common range becomes necessary, see section 2.4.2. While the absolute shape index is bounded as $0 \leq S \leq 1$, the dimensionless curvedness only has the planar surface as a lower bound, thus $C \in \mathbb{R}^+$. However typical values of C range in $[0, 3]$. The area-based joint probability density function $P(S, C)$ of the pointwise quantities S and C is obtained provide a non-local description of the geometry of the surface. Based on the first and second moments of the joint probability density function $P(S, C)$ a so-called feature center \hat{S}, \hat{C} is defined which lies closer to region of high density than the mean center (asymmetry of the joint pdf). In addition upper and lower distances $d_u^S, d_l^S, d_u^C, d_l^C$ of the joint pdf in each of its variable are obtained. The set of parameter which span the so called feature space of parameters is completed by the global compactness parameter, which is defined as

$$\hat{\lambda} = \sqrt[3]{36\pi} \frac{V^{2/3}}{A} \quad (2.2)$$

and has a range $\hat{\lambda} \in [0, 1]$. The compactness parameter is also known as the sphericity since it measures how close the shape of an object is to a perfect sphere ($\hat{\lambda} = 1$) (Wadell, 1935). As a result of the geometrical characterization, each isosurface is represented as a point in the 7-dimensional feature space of parameters $\{\hat{S}, \hat{C}, \hat{\lambda}, d_u^S, d_l^S, d_u^C, d_l^C\}$. The feature space of parameters is designed to geometrically classify surfaces of different shape, e.g. identify clusters in the high-dimensional space. However, combing the feature space of parameters with the three attributes describing the location of each surface would construct a 10-dimensional space in which the search for correspondences needs to be conducted. Searches in high dimensional space are subject to the ‘‘curse of dimensionality’’ problem as observed by Bellman (2015). Furthermore with k being the number of dimensions, in kd-tree data structures which are suitable for searches in space of low to medium dimensionality the number of data points N should be $N \gg 2^k$ in order to make the search significantly more efficient than an brute-force search. Thus, the number of dimensions of the space in which the correspondence search in conducted should be large enough. to distinguish structures but not bigger than that. Consequently we combine the location information of the structure not with the full feature space of geometrical parameters, but with the subset $\{\hat{S}, \hat{C}, \hat{\lambda}\}$. Typical regions in this reduced parameter space are near the $(\{\hat{S}, \hat{C}, \hat{\lambda}\}) = (1, 1, 1)$ point for blob-like structures and $(1/2, 1, \hat{\lambda})$ for tube-like structures, which become more stretched with a lower value of $\hat{\lambda}$. Sheet-like structures are characterized by low values of \hat{C} and $\hat{\lambda}$.

The proposed tracking method is not limited to a particular choice of feature attributes. Depending on the application, different attribute sets may be chosen. However, in the present work, the set of attributes describing each structure in any given time frame is formed by the three spatial attributes $\{x_c, y_c, z_c\}$ and the three geometrical attributes $\{\hat{S}, \hat{C}, \hat{\lambda}\}$. Hence each structure is represented as a point in a six dimensional attribute space and all structures in a frame form a cloud of points in this attribute space. As described in section §2.4.2, a scaling and weighting operation is applied to the group of attributes. Different events affect the structure attributes in different ways. While the geometrical shape of the structure is marginally evolving in a continuation, splits and merges can significantly affect the geometrical signature. Thus, to detect all types of events different subsets of the group of attributes are used in the search of correspondences (see §2.4).

In addition to the attributes that are used for the correspondence search, another set of attributes can be used to test and reject found correspondences and events based on physical realizability. To test for realizability, the algorithm follows region-based approaches to constrain correspondences and attribute-based approaches to constrain events. In flow configurations with a dominant mean flow direction, the direction of the integrated velocity of the structures may be used as an attribute to test directionality of the found event. Depending on the application a structure attribute may be identified which can be assumed to be conserved within the event within a user-defined tolerance. Examples for such attributes could be the overall volume or mass which may be assumed to conserved from all source to all target structures of an event (see §2.7.2).

2.4 Detection of correspondences

After all structures from the source and target frame have been mapped into the attribute space, a multi-stage search for correspondences between these two clouds of points is conducted utilizing first the full set of attributes and then its spatial subset only. Matching source points with target points by comparing the structure attributes translates into minimizing the distance between matched points in a space which is spanned by those attributes or a subset of them. The search for correspondences is conducted in multiple stages in which both the source and the target clouds participate as query and searched clouds. An overview is presented in figure 2.1(b). In general, the difficulty of frame-to-frame correspondence detection and the complexity of the resulting events highly depend on the sampling frequency of the dataset. The proposed combined method of searching and constraining aims to be applicable to datasets with low sampling frequency where corresponding structures do not necessarily spatially overlap in consecutive frames. However, a sufficiently high sampling rate is assumed. The limiting value for the sampling frequency is application dependent. In a dataset with relatively slow feature dynamics, in terms of the attributes spanning the attribute space, a lower sampling frequency may be sufficient than in applications characterized by quick feature dynamics. The minimal required sampling frequency also depends on the scale of the structures subject to the tracking in relation to the structure velocity. For most tracking methods (especially region-based) small, fast moving features result in the most restrictive sampling frequency. However, in the attribute space, structures are represented in terms of spatial location and geometric shape. In other words, structures that move fast relative to

their size but are characterized by minor changes in shape from frame to frame may result in a comparable minimum sampling frequency than structures that move more slowly relative to their size but deform faster. Thus, measures to evaluate the sampling frequency based on location and shape are needed.

2.4.1 Projection of spatial attributes

As a pre-step to the correspondence search, the velocity of the structures can be utilized to compute an estimated position used to improve the spatial subset $\{x_c, y_c, z_c\}$ of the structure attributes. The goal of the projection is to further minimize the distance of matching points in the attribute space before the actual search is performed. Either the source structures can be projected forward or the target structures backward. The direction in which the spatial projection is conducted prior to each stage of the correspondence search is chosen based on the event type whose detection is the main objective of that particular stage, as detailed in §2.4.3-2.4.4.

In contrast to Muelder & Ma (2009), who predict the feature region in the target frame as an extrapolation using the feature’s positions in the previous frames, we compute the estimate $\mathbf{x}' = (x'_c, y'_c, z'_c)$ for the spatial subset of attributes in the previous or successive frame from the current subset $\mathbf{x} = (x_c, y_c, z_c)$ as a linear projection $\mathbf{x}' = \mathbf{x} \pm \Delta t \mathbf{v}_{\text{avg}}$ (plus / minus sign for forward / backward projection) based on the average structure velocity defined as

$$\mathbf{v}_{\text{avg}} = \frac{1}{A} \int_A \mathbf{v} dA \quad (2.3)$$

In general, the quality of the projection depends on the length of the sampling interval. A higher-order projection could be obtained by utilizing the velocity information, not only from the current source or target frame but also from previous or subsequent frames. However, when evaluating the relevance of the spatial projection in general, and especially of including higher order contributions, it needs to be considered that topology-changing events, such as split and merge, can change the structure’s center location from source to target frame more significantly than predicted by the velocity-based displacement, where no change in topology is assumed. In addition, application-specific phenomena that accelerate or decelerate structures (such as shock waves in a compressible flow) further diminish the benefits of the projection. To account for these effects the spatial projection may be zonally adapted, for example turned off across a shock wave.

The idea of a spatial projection will be further utilized in the constraining process of found correspondences (discussed in §2.5), where instead of the structure’s spatial attributes, the structure’s bounding volumes or the structure itself will be projected. In contrast to the projection of the spatial attributes $\{x_c, y_c, z_c\}$, this spatial projection of the structure or its bounding volume may benefit from including a rotational contribution, in addition to the translational component.

2.4.2 Attribute scaling

Since the spatial attributes describe the location of a structure in physical space, the domain size of the application limits their range. In calculating the Euclidean distance with coordinates of different scales the coordinate with larger range would dominate the distance. Hence, a rescaling (standardization) of the attributes is necessary. We rescale an attribute α as $\tilde{\alpha} = (\alpha - \alpha_0)\omega$, where α_0 is the attribute origin and ω is the weight of the attribute. Different weights can be assigned to the geometric attributes, depending what attributes are to be emphasized. For the tracking application presented in this work, a weight defined as the inverse domain length is used for the spatial attributes to rescale them into a range $[0, 1]$ and balance the contribution from spatial and geometrical attributes to the distance calculation.

2.4.3 Nearest-neighbor search in the attribute space

Regarding the inter-frame evolution of a structure that continues from a source to a target frame and has no interactions with other structures neither in the source nor the target frame, the following observations can be made: 1. The spatial displacement of the structure is limited. Thus, the corresponding target structure may not be the spatially closest to the source structure, but can be found somewhere in its spatial neighborhood. 2. A continuation event can be characterized by small topology changes compared to split and merge events. The geometrical deformation of the structure is limited. Therefore, for a continuation event, a corresponding target structure can be expected in the close neighborhood of the source structure in the attribute space. Hence, a nearest-neighbor (NN) search in the full six-dimensional attribute space is conducted to detect potential continuation events. The outcome of a NN search depends on the search direction, e.g. if the nearest neighbor in the target cloud is found for each source point or vice versa. Hence, we conduct the NN search in both directions, starting with source-to-target as the first step of the first stage of the correspondence search. Different data structures and algorithms have been proposed in the literature to conduct a NN search in higher dimensional space, as for example summarized by Yianilos (1993). We conduct the NN search in the attribute space based on the construction of a kd-tree data structure as introduced by Bentley (1975); Friedman *et al.* (1977). In general, a kd-tree is a space-partitioning data structure that hierarchically organizes the data points based on their position in the k-dimensional space aiming to allow efficient access to the organized points by positionally querying. To organize the data, the space is recursively split into half-spaces by axis-aligned hyperplanes. Since half-spaces are constructed in the k-dimensional space, the kd-tree is a binary tree. Each node of the tree is not only associated with a half-space but also with one of the k dimensions that is used as the splitting direction to generate the node’s children. Although each nodes defines the direction of the splitting hyperplane, the construction of the kd-tree is not unique, because the plane can in principle be placed anywhere in the half-space. A standard rule is to split before the median point along the longest dimension of the half-space. Hence, assuming the overall space has comparable depth in each dimension, one cycles through the dimensions when moving down the tree.

Once the kd-tree is constructed, several ways exist to find the nearest neighbor to a query point. We note that always choosing the “good side” (i.e. in direction of the query point) of the binary tree when traversing down the tree can have a high probability of failure, especially when many data points are located close to half-space boundaries. Detailed strategies can be found in Liu *et al.* (2005), for example.

We refer to the cloud which is formed by all query points in the NN search as the *query cloud* and the searched cloud which is partitioned in form of a kd-tree as the *match cloud*. In the search for the nearest target neighbor to each source point in our attribute space, the source points serve as the query cloud and the target points form the match cloud. Opposite roles are taken when the NN search is conducted to find the nearest source neighbor for each target point. At each stage of the correspondence search, we define a confidence value for each found correspondence in the particular stage of the search, allowing the assignment of different weights on specific stages of the search. At the end of the search a total confidence is computed (see §2.4.6). The confidence evaluation is continued in the constraining process. For the bidirectional NN search, we determine the confidence value depending on the user preference either as a general value for the NN search $c_i = c = \omega_{NN}c_{NN}$ or as a correspondence-specific value

$$c_i = \omega_{NN} \left[c_{NN} + (1 - c_{NN}) \left(1 - \frac{d_i}{\bar{d} + 3\sigma} \right) \right] \quad (2.4)$$

where ω_{NN} and $c_{NN} < 1$ are the user-provided weight and minimum value for the NN confidence. d_i is the Euclidean distance between query and match points in the attribute space for the i -th found correspondence, and \bar{d} and σ are the mean and standard deviation of the distances of all found correspondences in this stage. Note that, before weighting, the confidence value is bounded between c_{NN} and unity, where a unitary confidence is associated with a vanishing distance in the attribute space and hence maximal confidence in the correspondence. Depending on the relative location of point in the source and target clouds in the attribute space, a correspondence may be found in both search directions and a confidence value will be assigned to the same correspondence in both directions. If the mean and the standard deviation, \bar{d} and σ , are included in the confidence determination (equation 2.4), then the confidence value will not be the same in both search directions, in general.

In contrast to a range search, the NN is not restricted to a maximal distance. Events other than continuation lead to a more pronounced separation of source and target points in the attribute space because of the associated topology changes. However, since the NN search is not rejecting correspondences by comparing to a certain maximal allowed distance, correspondences are still going to be found for the source and target points involved in these non-continuation events. The validity of these correspondences depends significantly on the number of source and target structures and their location in the attribute space. The NN search for each source point has a tendency to detect either potential continuation or merge events, while the NN search for each target point detects predominantly continuation and split candidates. To detect merging structures, a forward projection of the source spatial attributes reflects the converging character of the merge event, while for a split event, a backward target projection captures the diverging targets. Hence, prior to the NN search from source to target, a forward projection of the source spatial attributes is conducted. Likewise, prior to the NN search from target to source, a backward projection of the target

spatial attributes is performed, as described in §2.4.1. However, the validity of split and merge events found in the NN search stages of the search is uncertain due to the unpredictable changes of the geometrical attributes $\{\hat{S}, \hat{C}, \hat{\lambda}\}$ that occur in a merge or split event, however certain tendencies exist, see §3.5. This problematic can be further illustrated by considering the behavior of a NN search in terms of disappearance and creation events. Consider a (synthetic) frame pair where a structure continues from the source to target frame and another small source structure disappears, e.g. has no true forward correspondence. However, visible to the NN search are two structures in the source frame and one in the target frame. Hence, the unrestricted NN search for both source structures will result in matching with the same target structure, while the NN search for the target structure will match the target structure with only one of the two source structures. Combining all found correspondences of the two NN searches a false merge event is detected. Analogously, a false split event is detected when in reality only a single continuation and creation takes place. These simple thought experiments emphasize a need to test the validity all correspondences found in the NN search, especially the non-continuation events, and a need to have an alternative search method to find candidates for split and merge events. However, applying the NN searches without restriction still has a justification and certain advantages. The hypothetical example with two source structures and one target structure could also be used to illustrate an event where the source which was disappearing in the previous example is instead a truly merging / incoming structure which is moving fast and is small (hence is causing only minor topology changes in the merge). The dominant correspondence of this merge is the correspondence found by the NN search in both directions. However, a false disappearance and continuation instead of a merge may have been detected if the NN search would have been conducted with a maximum allowed distance. The proposed bidirectional NN search needs to be seen as one part of a multi-stage search and in combination with the constraining process. The purpose of the search is to find all possible correspondence candidates, which will be tested for realizability in the constraining stage.

2.4.4 Radius search in the spatial subspace

The geometry of a continuing structure evolves smoothly between the source and target frame. The structure is locally deformed, but the target structure is characterized by a global geometric signature $\{\hat{S}, \hat{C}, \hat{\lambda}\}$ which is comparable to its counterpart in the source frame. Contrary to the continuation, events involving interactions with other structures, such as split and merge, cause significant changes in the shape of the structure. Consequently, corresponding structures are separated in the full attribute space. The change in shape not only modifies the geometrical attributes of the structure, but also relocates its center, leading to a larger spatial displacement, in terms of structure centers, compared to continuation events. However, topology-changing events affect the geometrical attribute subspace much more abruptly than the spatial subspace: Despite the spatial displacement, corresponding structures are still expected to be found in their spatial neighborhood. Therefore, a correspondence search in the spatial subspace is conducted to exclude the geometrical attributes from the search. Although excluding the geometry from the search is motivated by jumps in the geometric signature caused by split and merge events, the search in

the spatial subspace is still going to detect potential continuation candidates, which are not only neighboring each other in the full attribute space, but also in the spatial subspace.

In contrast to the correspondence search employing the full spatial and geometrical attribute set, the search in the spatial subset is conducted in form of a radius search rather than a nearest-neighbor search as illustrated in fig. 2.2(b). A k-nearest-neighbor search in the spatial subset would determine the k-nearest neighbors of the query point without restriction to their distance. In contrast, a radius search around a query point finds all match points within a fixed distance from the query point. Conducting a k-nearest-neighbor search is not feasible, because the number k of nearest neighbors to be found cannot be specified, since no assumption of the order of the split or merge event can be made at this stage. Instead, searching for all candidates involved in the event comes with the requirement of having to define a search radius.

For the NN search of both stages an attribute scaling and weighting as described in section 2.4.2 is necessary because of the different scales of spatial and geometrical attributes. To conduct the radius search in the spatial subset a scaling is not applied, since all spatial attributes are used with equal weights.

An individual search radius is assigned to each query point based on the size of the structure. Different options to measure the structure size can be considered. The spatial extent of objects is often measured in form of their minimum bounding box. A minimum bounding box of three dimensional object is defined as the rectangular cuboid with minimal volume which encloses all points of the object. Choosing a rectangular box as a bounding rather than any other shape such as a bounding sphere is computationally convenient when a domain is described using a Cartesian coordinate system. Depending on the constraints that need to be satisfied by the box different types of bounding boxes exists. Figure 2.2(a) shows an isosurface enclosed by its axis-aligned bounding box (AABB) and oriented bounding box (OBB).

While the AABB is defined by the minimum and maximum position of the structure in each coordinate direction, the OBB can be characterized by a center point indicating the location, an orthonormal set of three basis vectors indicating the orientation and three scalars indicating the symmetric extent of the box in the direction of its principal axes. The box's three principal axes are referred to as major, medium and minor axis and the corresponding scalars describing the extent are referenced as half-width, half-height and half-depth. In case of the oriented bounding box no restriction apply to the orientation of the box and the orientation is chosen to minimize the box's volume. The axis-aligned bounding box also minimizes its volume, but under the constraint that its principal axes have to be aligned with the axes of the coordinate system. Hence, the axis-aligned bounding box is simply defined by the structures minimum and maximum x, y, z coordinate.

Once the axis-align or oriented bounding box has been found for each query structure in the radius search, it is used to define the the structures search radius. Possible choices for the search radius are the half-diagonal of the box, its longest half-dimension or its rms-value of all three dimensions. The definition of the structure's search radius is completed by specifying a tolerance factor (> 1). Tracking application presented in this work apply a search radius based on the OBB half-diagonal with a tolerance factor of 1.3. An alternative to forming the search radius based on

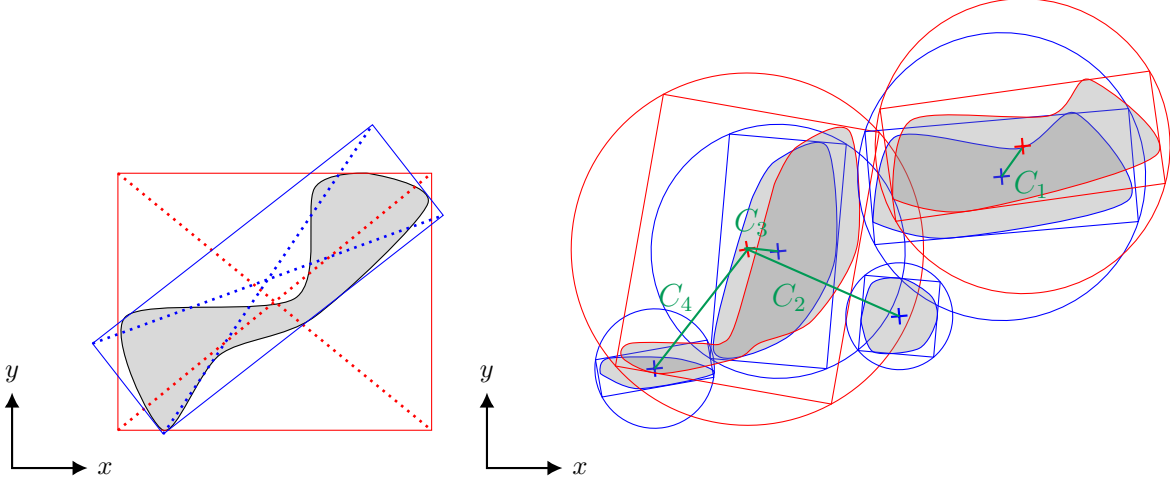


Figure 2.2: (a) Generic two dimensional shape enclosed by its axis-aligned bounding box (red) and oriented bounding box (blue). (b) Illustration of a continuation, first order merge and disappearance event of generic shapes. OBB of source structures in blue, OBB of target structures in red. Search radius based on OBB half-diagonal without tolerance. Events detected by the radius search are a true continuation and a false second order merge, which again emphasizes the need for a constraining process based physical realizability.

the structure’s bounding box is to use the structure’s μ parameter, $\mu = 3V/A$, which has been previously used to non-dimensionalize the curvature of the structure, see section 2.3.

Similar to the nearest-neighbor search in the full attribute set, the radius search is conducted in both directions. In sequence of the multi-stage correspondence search, we apply the radius search where the source points in the spatial attribute subspace serve as query points and the target cloud is the searched match cloud after the NN search is conducted with the source points as query points. To organize the multi-stage search, we define a first stage where the source points are the query points and the target points are the match points. The first step of this first stage is the NN search in the full attribute space from source to target cloud, followed by the radius search in the spatial subspace. The second stage of the search consists of the same two steps, but conducting the NN search in opposite direction and using the target points as query points and the source points as match points for the radius search. To perform the radius searches efficiently, we organize the respective three dimensional match clouds in a kd-tree data structure. While the query clouds always contain all points of the source or target clouds, the user can optionally filter out points in the match cloud that have been already matched in previous search steps aiming to minimize the time spent on the kd-tree construction and search.

Since the search radius is associated with the query structure size, split events (or split parts of a compound event) are more likely detected when a source structure is used as a query, while merge events (or merge parts of a compound event) have a higher likelihood to be found when a target structure is used as query. Hence, prior to the radius search using the source points as query, the target backward projection of spatial attributes is conducted. Likewise, the source cloud forward projection is carried out prior to the radius search using target points as query.

In spite of this spatial projection, the detection of split and merge events still benefits from the non-directionality (isotropy) of the radius search. For example, a highly convoluted, elongated source structure could split in a way where the line connecting the center of the splitting source structure and the center of the splintered target structure is largely deviating from the direction of the projection.

Similar to the NN search, a confidence value is calculated for each found correspondence in the radius search. The confidence in each correspondence is defined as

$$c_i = \omega_R \left(1 - \frac{d_i}{r_i} \right) \quad (2.5)$$

where ω_R is a specified weight, d_i is the center distance between the matched structures and r_i is the search radius of the query. Note that, before weighting, the confidence value is bounded in $[0, 1]$, where 1 indicates maximal possible confidence in the found correspondence. The distance d_i of the correspondence is the distance between the projected, rather than the original, source (target) point and the target (source) point.

2.4.5 Detection of unmatched structures

Based on the assumption that the source and target frames contain structures, carrying out the bidirectional and unrestricted NN search guarantees that all structures have at least one corresponding structure. Consequently, no creation and disappearance events can be found when the NN searches are performed. However, structures which are newly created in the target frame or have their final occurrence in the source frame should not be part of any correspondence. The falsely detected correspondences involving these structures do not pass the constraining process. If the bidirectional NN search is conducted, the constraining process is the only mechanism to find creation and disappearance events. In contrast, if only the spatial radius search is conducted, structures may be present in the source or target frame that have not been matched with a corresponding structure. Thus, a search is performed for unmatched source and target structures. The found structures participate in a creation or disappearance event. However, the pool of creation and disappearance events is enhanced during the constraining process, see §2.5 and §2.7. For example, a detected continuation that does not satisfy one of the applied constraints generates a disappearance and a creation event.

2.4.6 Collecting correspondences form the multi-stage search

At the end of the multi-stage correspondence search a list of detected correspondences is obtained. A correspondence identifies the involved source and target structure as well as the confidence assigned to the correspondence. In the multi-stage search, the same pair of source and target structures can be matched with each other multiple times. For example, a correspondence may have been established in both direction of the NN search or the radius search generates a correspondence that has previously been found by the NN search. Those correspondences share the information of the involved structures. However, they are not complete duplicates, because the confidence values differ based on the distances in the different spaces and the specified weight for each search step. The group of

correspondences with common source and target structures can be merged into a single correspondence, storing the duplicated source and target information and summarizing the search record of this group in the form of a cumulative confidence.

2.5 Constraining of correspondences

In contrast to event constraints (§2.7) that test an event in its entirety for physical realizability, correspondence constraints evaluate the plausibility of single correspondence candidates found in the search as described in S2.4. Correspondences deemed not physically realizable on a structure to structure basis are rejected as a pre-step to the correspondence-based formation of events, to avoid the formation of complex, non-physical events in the first place. Although attributes of the involved features can be applied to constrain correspondences, all constraints proposed in this section are region-based, indicating the hybrid nature of the tracking algorithm. The presented region-based correspondence constraints can be classified in two groups working either with the feature’s bounding volume, such as axis-aligned or oriented bounding box, or directly with the polygonal mesh of the structure. Hence, the different types of correspondence constraints are characterized by different degrees of accuracy and computational cost. In particular, the axis-aligned bounding box representation of a structure, and thus all constraints based on it, are associated with the lowest accuracy, while the constraints invoking the oriented bounding box and polygonal mesh of the structure result in intermediate and high accuracy, respectively.

To minimize the overall cost of the correspondence constraining, a sequence of constraints ordered by ascending accuracy and cost is applied to each correspondence. Assuming the tolerance setting to be consistent throughout the constraints, it can be concluded that a correspondence that is rejected by a constraint of low accuracy would also be rejected by all subsequent constraints of higher accuracy. Hence, the correspondence does not need to be tested in the subsequent constraints. Especially, the more computationally expensive surface proximity constraint (§2.5.2), that works with the polygonal meshes, is only applied to a reduced number of correspondences.

To continue the confidence evaluation for each correspondence found in the correspondence search, a confidence value is computed in each constraint. A found correspondence is rejected, if the calculated confidence falls below the specified threshold for any correspondence constraint in the sequence. However, a source will be marked as disappearing only if all its correspondences are rejected. Analogously, for the involved target structure, a creation event will be generated only if there is no other correspondence which passes the constraining and has the same target.

2.5.1 Bounding volume constraints

A computationally inexpensive, region-based evaluation of the physical realizability of a structure to structure correspondence can be conducted by utilizing a bounding volume representation of the correspondence’s source and target structure. The correspondence constraints proposed in this section use either the AABB or OBB of a structure as its bounding volume, and examine the spatial relation between source and target bounding volumes.

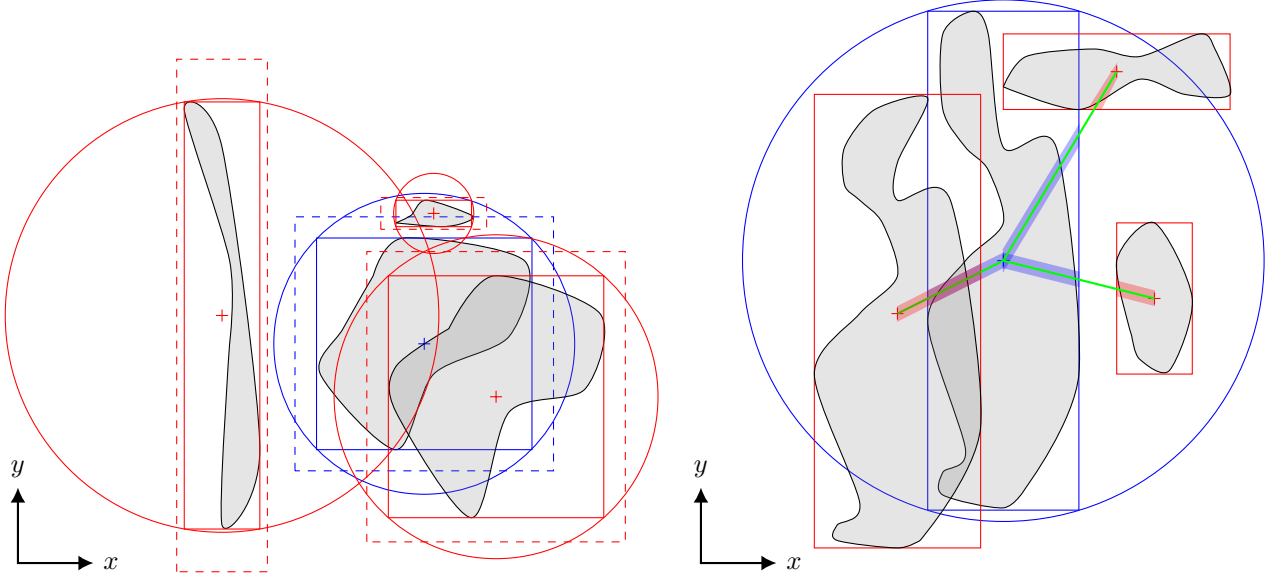


Figure 2.3: (a) Two dimensional illustration of the axis-aligned bounding box volume constraint. While the solid boxes are the AABBs, the dashed boxes indicate the resized AABB of each structure. The constraint evaluates correspondence as previously found in a radius search (even without tolerance) based on the overlap region between a resized source AABB (blue) and three resized target AABB (red). Here, the correspondence between the source and the tube-like target structure would be rejected by the constraint. (b) Two dimensional illustration of the axis-aligned bounding box distance constraint. Source AABB in blue, while target AABB drawn in red. Displacement between box centers shown in green. Box radius in direction of the displacement highlighted.

Similar to the projection of spatial attributes in the correspondence search, all bounding volume constraints optionally utilize the surface-averaged structure velocity to either project the bounding volume of the source forward or the bounding volume of the target backward, depending on which projection minimizes the distance between the bounding volumes (defined as the distance between the bounding volume centers).

The *axis-aligned bounding box volume constraint* tests the physical realizability of a correspondence by inspecting the spatial (volumetric) overlap between the resized AABB of the source and target structure involved in this correspondence. If the resized AABBs do not overlap, the correspondence is deemed physically unrealizable. The AABB of the source and target structures are resized by a common specified tolerance factor preserving the AABBs aspect ratios, as illustrated in figure 2.3(a).

The volume V_o of the overlapping region between the two AABBs can be computed as

$$V_o = \prod_{i=1}^3 \max \left[\min (S_i^{\max}, T_i^{\max}) - \max (S_i^{\min}, T_i^{\min}), 0 \right] \quad (2.6)$$

where $S_i^{\max}, S_i^{\min}, T_i^{\max}$ and T_i^{\min} are the maximal and minimal coordinates of the source and target AABB in i -th dimension. If the resized source and target AABB do not overlap ($V_o = 0$), the correspondence is rejected by the

constraint. Overlapping resized AABBs allows the correspondence to pass the constraint and a confidence value is assigned as

$$c = \omega \frac{V_o}{\min(V_S, V_T)} \quad (2.7)$$

where ω is a specified weight and V_S and V_T are the volumes of the resized source and target AABB. The computed confidence is added to the value generated in the search for each correspondence.

Alternatively to the volumetric overlap test of the resized source and target AABBs, a correspondence can be evaluated by examining the spatial extent or radius of both boxes in the direction given by the displacement between the centers of the AABBs. A two dimensional illustration of the *axis-aligned bounding box distance constraint* is presented in figure 2.3(b).

The radii $r_{s \rightarrow t}$ and $r_{t \rightarrow s}$ of the source AABB in direction of the target AABB and vice versa can be computed from the axis-aligned extents of both boxes. If the Euclidean norm of the center displacement is denoted as $\|d\|$, the correspondence passes the constraint if $\|d\|/(r_{s \rightarrow t} + r_{t \rightarrow s}) < \varepsilon$, where ε is a specified threshold. The assigned confidence value of a passing correspondence is determined relative to the threshold ε and defined as

$$c = \omega \left(1 - \frac{\|d\|}{\varepsilon (r_{s \rightarrow t} + r_{t \rightarrow s})} \right) \quad (2.8)$$

which is, for a passing correspondence, bounded between $[0, 1]$ before weighting with the specified factor ω .

Depending on the orientation of the structure, the OBB can enclose significantly less volume compared to that of the structure's AABB. Analogously to the volumetric overlap test of source and target AABB, the *oriented bounding box volume constraint* conducts a volumetric overlap test in terms of the resized OBBs of the structures involved in the correspondence. A brute-force algorithm conducts $2 \times 12 \times 6 = 144$ edge-face checks to determine if two OBBs overlap or not. In contrast, a fast overlap test for two OBBs can be conducted by applying Minkowski's separating axis theorem (SAT) as suggested in Gottschalk *et al.* (1996). In general, the SAT can be used to test if two convex sets are disjoint sets in n-d Euclidean space. The SAT examines the disjointedness of two convex objects by conducting an axial projection of both objects onto an axis on which two intervals are formed by the projections. If those intervals do not overlap, then the axis is a *separating axis* and a *separating plane* orthogonal to the separating axis exists that separates the two objects, which are consequently proven to be disjoint. However, if there is an overlap between the intervals, the objects may still be disjoint. Further testing using another axial projection onto a different axis is needed.

For any two points of a (object oriented bounding) box, the box contains the line segment which joins the two points. Hence an OBB is a convex set and the SAT can be used to test the overlap status of two OBBs. In addition, an OBB is also a polytope, which simplifies the selection of axes for the axial projection of the SAT. If two convex polytopes are disjoint and there exists a separating plane between them, then there also exists a separating plane which is either parallel to a face of either polytope or parallel to an edge from each polytope. Since the separating axis is orthogonal to the separating plane, the axis can be found either orthogonal to a face or edge of either polytope. Since each OBB has three face orientations and there are nine pairwise combinations of edges, there are 15 cases

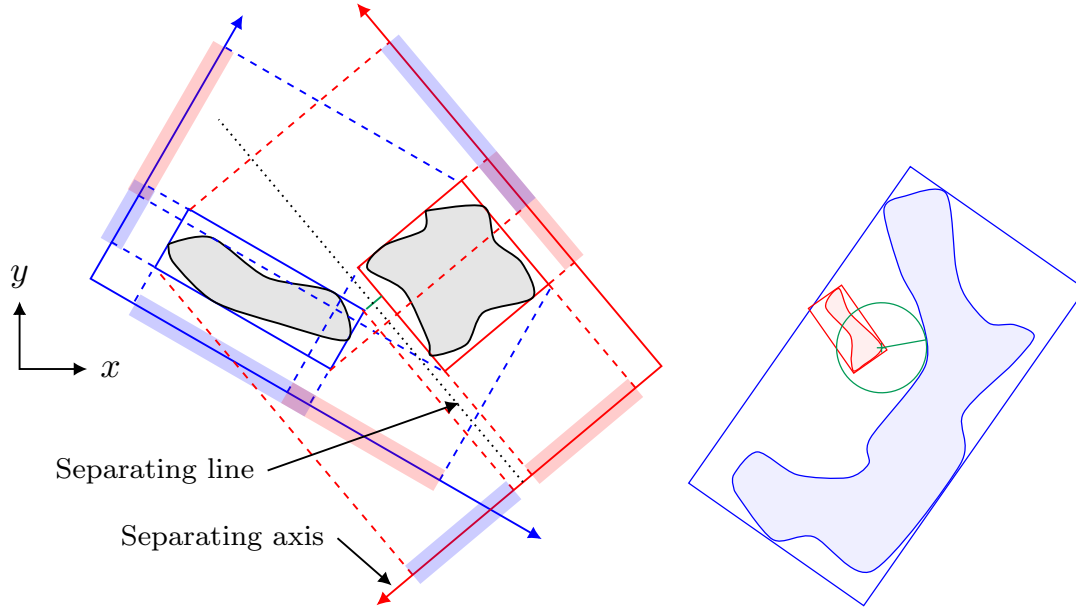


Figure 2.4: (a) Illustration of the separating axis theorem in two dimensions applied to rectangular shapes. (b) Illustration of the minimal proximity between two generic shapes with overlapping oriented bounding boxes. The target (red) structure is enclosed in the concave region of the source (blue) structure. In spite of overlapping OBBs, the structures themselves do not intersect each other, but show a significant distance (green) between each other compared to their size.

of potential separating axes. A two dimensional illustration of the SAT applied to two disjoint boxes is presented in figure 2.4(a). The 15 test cases defined by the SAT are sufficient to determine if two OBBs overlap or not. The testing process may be terminated early, as soon as one of the tests reveals that a separating axis exists.

Following a similar approach to allow for some tolerance in the constraining as in the *axis-aligned bounding box volume constraint*, the SAT can be applied to resized OBBs of the source and target structure. Alternatively to testing the resized OBBs for spatial overlap, a certain distance between the original OBBs can be allowed. The Gilbert-Johnson-Keerthi (GJK) algorithm, Gilbert *et al.* (1988), iteratively computes the distance between two convex shapes and is therefore applicable to determine the distance between the source and target OBBs in the *OBB distance constraint*. To solve the distance query between two convex shapes A and B, the GJK elegantly recasts the problem by considering their Minkowski difference

$$A \ominus B = \{\mathbf{a} - \mathbf{b} \mid \mathbf{a} \in A, \mathbf{b} \in B\} \quad (2.9)$$

The motivation to consider $A \ominus B$ rather than the physical space becomes clear, when rephrasing the question about the spatial overlap of A and B to a question about common points of A and B. If A and B intersect, the two objects have at least one common point. Consequently, their Minkowski difference $A \ominus B$ contains the origin. In general, the distance between two convex shapes is the minimum distance between the origin and the Minkowski difference of the objects. Hence, the GJK iteratively searches for the point on the Minkowski difference closest to the origin, which is the point of minimum norm of the Minkowski difference. Instead of computing $A \ominus B$ explicitly,

its point of minimum norm is approximated by iteratively building a series of simplices, which move closer to the origin in each iteration. A simplex is a convex hull of a set of affinely independent points in \mathbb{R}^n . As the simplex moves closer to the origin, its point of minimum norm becomes a better approximation of the point of minimum norm of the Minkowski difference. To update the simplex in each iteration, points of the simplex which do not contribute to the closest point to the origin computation, are removed from the simplex, while a vertex from the Minkowski difference which is closer to the origin is added to the simplex. This vertex is selected by evaluating a support function on the Minkowski difference in the direction of the current minimal distance vector approximation. Hence, the vertex of $A \ominus B$ is selected which is farthest in the direction of the current distance vector. Further implementation details of the GJK can be found in Bergen (1999) and Montanari *et al.* (2017). In collision detection application the GJK is typically followed by an application of the Expanding Polytope Algorithm, Van Den Bergen (2001), to obtain the penetration depth of intersecting convex shapes. However, in the presented correspondence constraining, a correspondence passes the *OBB distance constraint* if the source and target OBBs intersect, and the penetration depth is not further considered. If the OBBs are not intersecting, their distance needs to be evaluated to decide whether to accept or reject this correspondence. The computed OBB distance is compared to a constraint measure characterizing the size of the source and target structure. Depending on the application different choices of constraint measure r can be made, such as the maximum of the half diagonal of source and target OBBs or the maximum of $\mu = 3V/A$ of the source and target structures, which is the constraint measure used in the tracking application presented in this work. A confidence value is assigned to passing correspondences as

$$c = \omega \left(1 - \frac{\|d_{\min}\|}{\varepsilon \max(r_s, r_t)} \right) \quad (2.10)$$

where d_{\min} is the minimum distance between the OOBs and ω and ε are specified confidence weight and tolerance values, respectively. The constraint measure of the source and target structure is indicated as r_s and r_t .

2.5.2 Surface proximity constraints

Although their flexible orientation makes OBBs more accurate representations of the flow feature’s spatial extent than AABBs, both bounding volume types, being convex objects, are inaccurate representations if the structure is highly convoluted and has significant concave regions. Figure 2.4 (b) illustrates a configuration where the OBBs of source and target structure overlap, while the actual structures enclose, rather than intersect, each other. Therefore, the narrow phase of the correspondence constraining solves distance queries between the flow features’ actual isosurfaces rather than their bounding volumes.

Distance queries for concave objects typically require increased computational effort compared to convex shapes. As an optional pre-step to the distance computation the isosurfaces of a correspondence are spatially projected according to the projection in the bounding volume constraining of the same correspondence.

The distance between the surfaces is determined by means of a point-to-point distance calculation where the surfaces are interpreted as point clouds. Therefore, the distance query translates into a three dimensional nearest-neighbor (NN) search that determines the nearest neighbor in the points of the target structure for each point in the source structure. The minimum distance of the surfaces is obtained as the minimum of these point-to-point distances. Similar to the nearest-neighbor search in the correspondence search, a kd-tree data structure is used to conduct the nearest-neighbor searches. Here, the NN search is conducted in lower dimensionality compared to the NN search used to find correspondences, and a kd-tree specialized for three dimension can be applied (Blanco & Rai, 2014). Furthermore, the surface distance computation can be terminated early, if the current point-to-point distance is smaller than a threshold, based on the grid size, below which the structure are considered to be intersecting. Correspondences of intersecting source and target surfaces are accepted and pass the constraining process based on local surface proximity. Analogously to the *OBB distance constraint* the distance between non-intersecting structures is compared to a constraint measure, taken as the maximum of source and target structure, to decide if the correspondence should be accepted or rejected. A confidence value is assigned to a passing correspondence as in equation 2.10.

2.6 Detection of events

At the end of the correspondence constraining process described in §2.5, a unique list of accepted one-to-one correspondences between source and target structures is obtained. By inspection of this list of corresponding structures a list of events as defined in §2.2 can be formed. An *event* groups of correspondences that have common source or target structures, as a structure in the source or target frame can correspond to zero, one or multiple structures in its complementary frame. Disappearing and newly created structures are detected among all source and target structures as those which do not have any forward or backward correspondence, respectively. A continuation event is identified as a correspondence in which both the query and the match indices are unique and neither the query nor the match index is part of any other correspondence. A group of correspondences with a shared source structure but distinct targets forms a pure split event, in which the duplicated source is the splitting structure. Analogously, a pure merge event of multiple order is build from multiple correspondences, which share the same match index. In the most general case, compound event is generated form a group of correspondences that not only has duplicates in the involved query indices, but also has common match indices. This event formation process generates a list of events describing the multi-structure interactions between source and target frames. Each structure from the source and target frames participates in one single event and, hence, appears only once in the obtained list of events.

2.7 Constraining and optimization of events

In contrast to the constraining of isolated structure-to-structure correspondences presented in §2.5, the event constraining process evaluates the physical realizability of single correspondences in the context of the entire event to

which they belong and tests the whole event for plausibility. Additionally, the event constraining evaluates if an event can be simplified into subevents, which consist of subsets of the correspondences of the whole event, under satisfaction of the constraint, i.e. each subevent extracted from the original event satisfies the event constraint individually.

2.7.1 Event-driven correspondence constraining

The correspondence constraining described in §2.5 assesses the realizability of structure-to-structure correspondences based on information contained only in this particular correspondence itself. For example, at the stage of correspondence constraining the type of event an investigated correspondence potentially is involved in is undetermined. Contrary, the event-driven correspondence constraining also tests single structure-to-structure correspondences, but utilizes information from the entire event that the correspondence is part of. In the tracking framework a number of event-driven correspondence constraints are implemented, while their applicability depends on the flow configuration and also on the type of feature of the flow to be tracked.

For example, in the feature tracking of passive scalar structures in compressible turbulence presented in §3, small structures discretized as triangulations with a number of points may disappear below specified minimum due to diffusion, which should be accounted as a disappearance event. If a small structure disappears in close proximity to another continuing structure, the presented correspondence search could detect a false merge event rather than a continuation and a disappearance. To correct those false merge events or merge-parts of compound events a simple constraint based on the number of points of a structure is introduced. If the number of points of discretized source structures involved in a merge event (or the merge part of a compound event) is below the specified minimum number of points for a structure to be considered in the tracking increased by a user-provided offset, the constraint removes this source structure from the merge event, while the remaining part of the event is kept. A disappearance event is generated for the removed source. If the original event was a pure merge event, removing the source either reduces the order of the merge event or transforms the merge into a continuation event. Analogously, removing a source from a compound event reduces the complexity of the compound or transforms it into a pure split event. Since the *number-of-points constraint* is only applied to merging sources, the event type needs to be known prior to its application and the constraint cannot be used at the correspondence constraining stage.

For flows characterized by a dominant mean flow direction, a constraint evaluating the directionality of an event can be useful to detect falsely formed events. For events other than creation and disappearance, measuring the plausibility of the directionality can be achieved by computing the angle between the surface-averaged velocity of the source structure and the distance vector between the structure centers (spatial attributes) of source and target for each correspondence in the event. The surface-averaged velocity indicates the estimated direction of motion of a source structure. The deviation from this estimation of the actual direction of the distance vector of the detected correspondence should be bounded by specified limits, if there is a dominant direction in the flow. Since topology changing events are expected to lead to larger deviations from the estimated direction of motion of a structure, different bounds are suitable for different event types.

2.7.2 Combinatorial extraction and conservation-based constraining of subevents

In addition to evaluating single correspondences in the context of an event, a constraining process is applied to test the physical realizability of all correspondences of an event at once and examines the plausibility of simplified subevents extracted from the original event. This subevent constraining can be applied based on one or multiple attributes, which are assumed to be conserved in total during the (sub)event within a specified tolerance and have not been utilized in the correspondence search. For example, the structure volume, mass or another physical quantity can be chosen. Since the attribute is assumed to be conserved, the realizability of the (sub)event can be tested by measuring the similarity between the source and target sum of the attribute(s). Subevents are extracted from an event by erasing one or multiple correspondences from the original event. Based on the similarity between the source and target sum of the conserved attribute a score can be assigned to each subevent, indicating the realizability of the subevent. The score $s_{i,c}$ of the i -th subevent with n sources and m targets based on a conserved attribute α is defined as

$$s_{i,\text{conv}} = \frac{\min(\alpha_{i,s}, \alpha_{i,t})}{\max(\alpha_{i,s}, \alpha_{i,t})} \quad \alpha_{i,s} = \sum_{j=0}^n \alpha_{i,j}^s \quad \alpha_{i,t} = \sum_{j=0}^m \alpha_{i,j}^t \quad (2.11)$$

where $\alpha_{i,j}^s$ and $\alpha_{i,j}^t$ are the values of the conserved attribute of the event's sources and targets, respectively. A (sub)event is considered to be realizable if its score is bigger than a specified threshold.

Depending of the event type under consideration different types of subevents can be extracted from the original event. The only subevents that can be extracted from a continuation event are one disappearance and one creation. Hence, if a continuation event does not satisfy the conservative event constraint, there is no alternative other than decoupling the continuation into a disappearance and a creation.

The extraction of subevents aims to avoid creation and disappearances as subevents, if possible. Original split and merge events that satisfy the constraint are therefore accepted without considering their potential subevents. A split event can fail the conservation constraint because of two reasons: (1) not all targets of the source are found in the search or meaningful correspondences have been rejected by previously applied constraints. Since at this point in the algorithm it is impossible to add correspondences to an event, there is no mechanism to recover from this rejection by the constraint. The split may be either kept or is completely decoupled into one disappearance and a creation event for each target of the original split. (2) the split is rejected because too many targets have been assigned to the source and passed the previous constraining process. Analogously, a merge can be rejected either because not enough sources are present in the event to fulfill the constraint or too many source are connected to the merging target. A mechanism to recover from a split (or merge) for which too many targets (or sources) are present in the original event is to extract subevents by removing one or multiple targets (or sources) from the event and testing the realizability of the resulting subevents. For a split (merge) event with n targets (sources), the number of subevent to be tested is given by $c = \sum_{r=1}^n n!/(n-r)!r!$.

The subevent with the maximum score may be a split (merge) of reduced order or a continuation event. If the optimal score satisfies the constraint, this subevent is accepted and a creation (disappearance) event is generated for each of the remaining targets (sources). If the maximum score which can be achieved with the given pool of targets

(sources) still does not satisfy a specified tolerance of the constraint, the optimized event may still be kept and either a creation (disappearance) is only generated for the remaining targets (sources) or the original event is completely decoupled.

Under certain circumstances, the proposed correspondence search can generate compound events in which a pair of sources exist that has a pair of common targets, called *double connections* in the following. The correspondences between these sources and targets typically differ significantly in their confidence values. To simplify the event and especially to enable the extraction of subevents, the correspondence between these subgroup of source and targets within the compound event with the lowest confidence may be removed if its confidence is lower than a specified threshold.

A compound event which has no double connections or from which double connections have been successfully removed previously can be decomposed into simpler subevents by classifying its targets based on their backward connections. In a compound without double connections three types of targets can be identified: *regular targets* have a single backward correspondence; *merge targets* have multiple backward correspondences and are the only target to one of their sources; *split targets* have multiple backward correspondences and all of their sources have multiple targets. Due to its single backward correspondence, a compound event cannot be decomposed into subevents at a regular target. In contrast, both merge and split targets allow the decomposition of the compound event. However, to avoid disappearances, the way subevents are extracted differs between decompositions based on a split or merge target. Because their sources have multiple forward correspondences, split targets are assigned to all candidate subevents that are generated by decomposing the compound event at the split target. After the score of all candidate subevents is determined the split target is ultimately assigned to the subevent with the maximum score, while the other reduced subevents still need to satisfy the constraint after the split target has been removed. In contrast, since a merge target has one source which has only the merge target as a target, the merge target can only be assigned to the subevent in which this source participates, to avoid that this source becomes a disappearing source.

Since the subevents extracted from compound events can themselves be compound events of reduced complexity, the compound decomposition needs to be applied in a recursive manner. Furthermore, the compound decomposition can be applied to the lowest possible level, i.e. aiming for subevents which are as simple as possible and still satisfy the constraint, or stopped at an intermediate level. In the latter case, the decomposition is stopped where further decomposing the subevents would lead to simpler events, which, on the other hand, have lower scores.

2.7.3 Sequence of event realizability constraints

If an event is constrained by one of the described constraints, the resulting sub-events are generated immediately. Therefore the order matters in a series of these constraints, as the events a constraint is acting on depends on the position of the constraint in the series. A general guideline to order event constraints is to go from complexity to simplicity, i.e. constraints acting on complex compound events should be applied first as they may generate simplified regular sub-events which need to be tested later on in the series in constraints acting on these simpler event types.

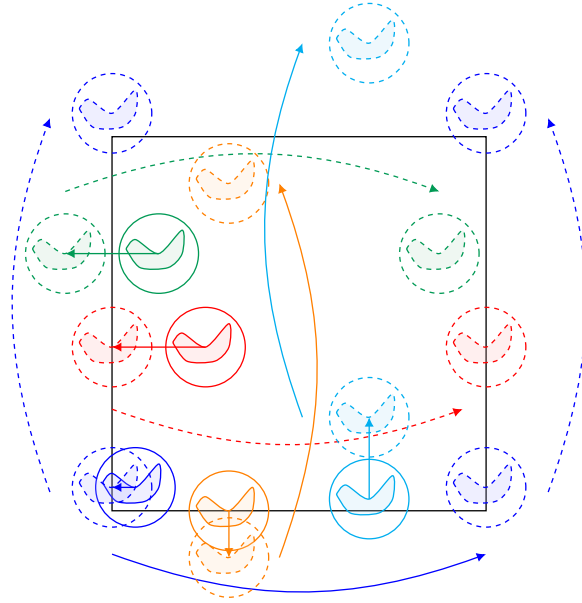


Figure 2.5: Duplication of boundary intersecting or crossing structures under periodic boundary conditions for one frame. Straight arrows indicate the estimated location of the structure in the successive or previous frame due to forward source or backward target projection. Bent arrows show the estimated location of the duplicate structure at the opposite side of the domain. Bent solid arrows illustrate duplication due to periodic structure reconnection, while bent dashed arrows show duplication due to boundary crossing structures.

2.8 Treatment of periodic boundaries

The applied geometrical characterization and classification of disconnected isosurfaces of the original field is limited to closed surfaces. As the tracking depends on the characterization as a pre-step from which the feature attributes are obtained, the tracking itself is also restricted to closed surfaces. However, this limitation could be overcome if a different set of attributes would be used. Because the features are required to be closed surfaces, any interaction of a structure with a domain boundary that leads to open surfaces requires a strategy to close the structures and, consequently, enable their tracking. Structures intersecting with non-periodic domain boundaries may be included in the geometrical analysis and tracking attempt by closing them either by simply using the boundary that they intersect or by closing them using their mirrored extension. Contrary, structures intersecting periodic boundaries are continued at the opposite side of the domain and, hence, these structures can be closed by reconnecting their pieces across the periodic boundary. To reconnect a structure across a periodic domain boundary Bermejo-Moreno & Pullin (2008) apply a matching algorithm for the intersection curves between the boundary and the structure on both sides of the domain. After the structure pieces are reconnected in the periodic direction, the structure is present as one closed surface at one arbitrarily chosen side of the domain used in the reconnection process. Note that if all three dimensions are periodic, the periodic reconnection needs to be applied in all three dimensions simultaneously to close a structure that intersects all three boundaries.

Periodic reconnection enables the geometrical characterization of previously open surfaces intersecting periodic domain boundaries. However, additional steps are required to apply the tracking under periodic boundary conditions.

In order to use the spatial attributes of a feature in the search for correspondences under periodic conditions, boundary crossing structures need to be detected and duplicated according to the periodicity of the domain. The spatial attributes of the duplicated structure are obtained from those of the original structure by applying a translation as specified by the domain size in the periodic direction, while the geometrical attributes are unaffected by the duplication. Furthermore, to solve the correspondence problem between source and target frames under periodic conditions, boundary crossing structures must be detected in both frames prior to the correspondence search. To detect potentially boundary-crossing structures, their location in the subsequent (source) or previous (target) frame is estimated by a linear projection, as described in §2.4.1. Structures that have not been reconnected prior to their geometrical characterization are duplicated if the radius search region or the bounding volume (whichever is bigger) of the projected structure intersects or crosses the boundary. Structures that have been reconnected across periodic boundaries are duplicated according to the original location of the structure fragments. However, the potential combination of boundary crossing and intersecting structure also needs to be taken into account (see blue structure in figure 2.5 for an illustrative example). The need to duplicate both sources and targets occurs because a source may correspond to a target which does not meet the criteria to be duplicated (source duplication needed) or vice versa.

The search for correspondences does not distinguish between duplicated and original structures. Correspondences are individually established for each duplicate (figure 2.6a). However, each original structure and its duplicates are fundamentally the same structure, therefore, all correspondences found need to be collected and assigned to one structure after the correspondence search (figure 2.6b). For example, if both the source and the target of a correspondence are duplicated, the correspondence will also be duplicated, i.e. two correspondences are established at different sides of a periodic domain, with common source and target and are duplicated correspondence between the same structures. Hence, during the collection of correspondence, duplicated correspondences are removed.

2.9 Graph representation of temporal evolution of flow features

Graph data structures represent pairwise relationships between objects and are therefore ideal to organize the temporal evolution of flow features by mapping the accepted events solving the correspondence problem between pairs of frames onto a graph at the end of each tracking step. Objects are abstracted to form the set of vertices of the graph and each related pair of vertices is an edge of the graph. In the present tracking methodology of turbulent flow structures each vertex of the graph represents a flow structure at a certain instant in time, while each edge indicates the correspondences between source and target structures. Each vertex and edge carries properties associated to the represented structure or correspondence, respectively. In particular, the structure identification and the time instant in which the structure exists are essential vertex properties to interrogate the graph for post-processing purposes. An edge can, for example, store the confidence of the correspondence associated with it. Here, edges connect from source to target vertex, i.e. have a direction associated with them and the resulting graph is directed. The source vertex of an edge always represents a structure at one time instant earlier than the structure represented by the target vertex

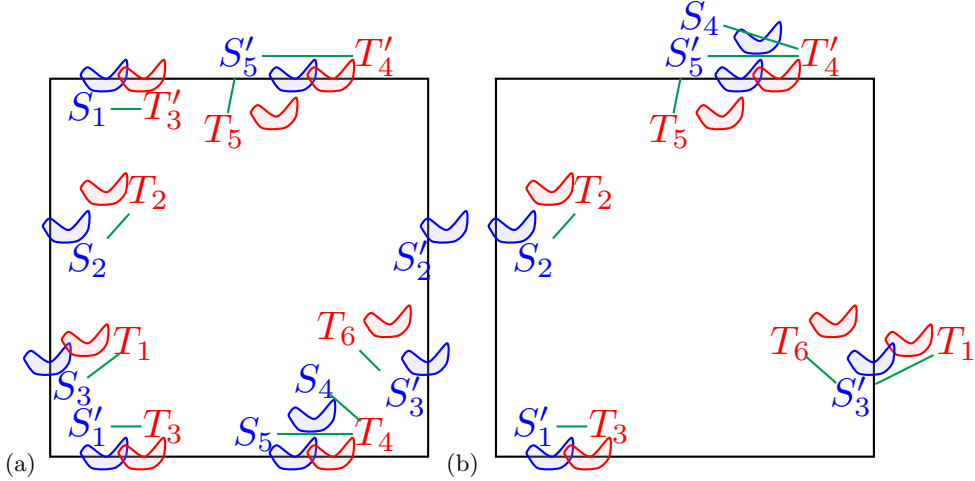


Figure 2.6: Correspondence search and event reconnection under periodic boundary conditions. (a) individually found correspondences (green) between duplicated and original source (blue) and target (red) structures. (b) resulting reconnected events. Primes indicate duplicated structures. Each structure is present in only one of the reconnected events in form of the original structure or one of its duplicates.

of the edge. At the end of each tracking step, the accepted events are mapped onto the graph and extend it by one frame, resulting in an ordering of vertices into a temporal sequence. Therefore the graph is characterized by a topological ordering, which forbids directed cycles. The graph is a directed acyclic graph.

After the list of accepted events is obtained for a tracking step, the graph needs to be updated according to this list of events. How different event types are incorporated in the graph building process is explained in the following. Intrinsic to the graph building process is an implicit vertex clustering which results into two-layer hierarchy of subgraphs, defined as *tree* and *branch*, of the overall graph data structure as illustrated in figure 2.7. By clustering the vertices and introducing the subgraph hierarchy, the graph reflects the life time and relationships of the tracked features. Each tree graph is a subgraph of the overall graph data structure, while each branch is a subgraph of a tree. Each branch is a representation of the lifetime of one individual structure, i.e. the first vertex of a branch is associated with the frame in which this particular structure first appears, either resulting from a creation or split event. On the other end, the last vertex of a branch represents the final presence of the structure either due to a following disappearance or a merge event. A tree is a composition of branches. All branches of a tree constitute the lifetime of structures that originated from the same parent structure, which is the root vertex of the tree. From this unique root vertex, all other vertices of the tree are reachable, which requires a minimum of connectivity between the branches of a tree. If structures originating from different root structures interact with each other, then their branches within the different trees will be connected. The only events that can connect trees are a merge or a compound. Due to the orientation of the edges, for a root vertex of one tree, there are unreachable vertices in the other tree and the trees are weakly connected. On the other hand, there may exist trees in the graph which are disjoint from all other trees. The structures of such an isolated tree only interact with structures that have the same

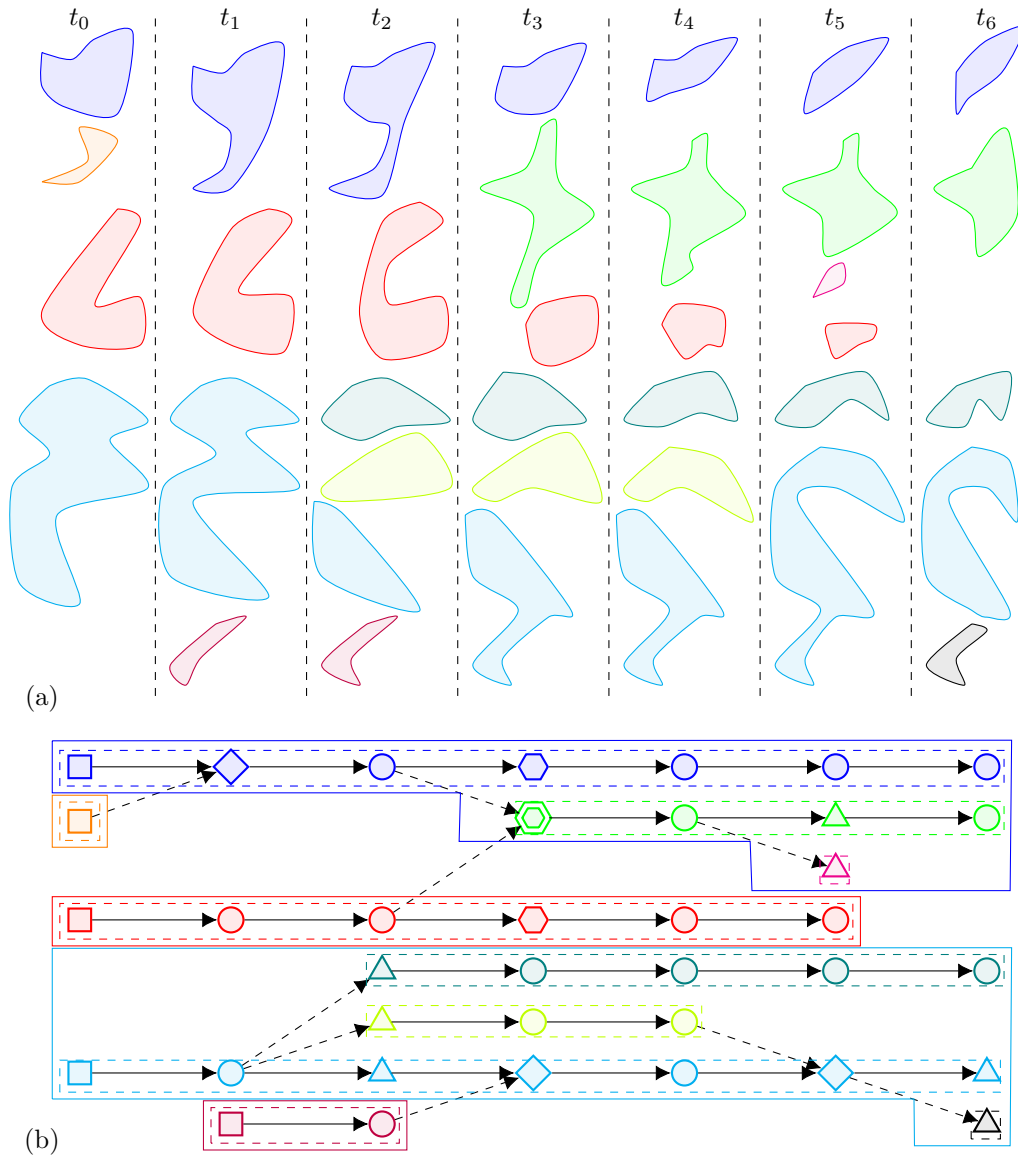


Figure 2.7: (a) Illustrative representation of the temporal evolution (from left to right) and interaction of structures of arbitrary shapes, where each color indicates an individual structure (undergoing merges, splits, creation and disappearances). (b) Representation of the structure evolution and interactions in (a) as a directed acyclic graph, with the same color scheme as the individual structures. Dominant edges in events indicated as solid arrows \longrightarrow , while non-dominant edges are shown as dashed arrow $-\!-\!-\!>$. Node and structure colors identify the same structure in multiple frames. Node shapes indicate the event type from which the structure results: \square creation, \circ continuation, \triangle split, \diamond merge, and \hexagon compound events. Trees and branches indicated as solid and dashed wrapping. Branches classified as primary ($-,-,-,-$), incoming ($-,-,-,-$), outgoing ($-,-,-,-$) and reconnecting ($-,-$). Note that a branch can be outgoing from or incoming to multiple branches through compound events. Not shown are branches of connecting type which are outgoing from a branch and incoming to another branch. Branches that have connecting and re-connecting character are considered to be of mixed type, see also fig. 2.9.

root structure. Finally, the union of all weakly connected or disjoint trees describes the evolution of all structures present over all frames of the dataset.

Each structure in the first source frame of the tracking application is represented as a root vertex, which seeds a directed tree with currently one branch. After a root vertex has been generated for each structure in the first source frame, the initial graph data structure consists of a number of trees equal to the number of structures in the frame. Each of these trees has one branch containing one vertex, which is the root vertex of the tree. Each root vertex is uniquely associated with a source of an event of the first tracking step. As the tracking progresses over time, the source structures of all events can always be mapped to the set of added vertices in the previous tracking step. On the other hand, to extend the graph according to newly found events, the target structures of the events are added as new vertices to existing trees, but not necessarily to already existing branches. The only exceptions are target structures that are involved in creation events. By definition, a newly created target does not have an antecessor and is therefore represented as a new root vertex in the graph, similar to the structures in the first source frame. A creation event always seeds a new tree with one active branch. Contrary, source structures that disappear do not have a successor and, consequently, a disappearance event terminates the branch to which the vertex representing the source structure has been added in the previous tracking step. Depending on how many active branches the tree has, the termination of the branch may also terminate the tree. The degree of a vertex can be used to distinguish different types of vertices. In a directed graph, the number of incoming edges of a vertex is its indegree, while the number of outgoing edges is the outdegree. Consequently, a vertex associated with a target of a creation event has an indegree of zero, while a disappearing source is a vertex of outdegree zero. A continuation event extends an existing branch of an existing tree. Since a continuation event is a single correspondence, the source vertex of the continuation has an outdegree of one, while its target vertex has an indegree of one.

Because the vertices involved in creation, disappearance or continuation events have degrees of either zero or one, the vertex clustering into branches to build these events in the graph is unique. In contrast, the source vertex of split and the target vertex of a merge have degrees higher than one. Multiple vertices of degree higher than one exist in a compound event.

As illustrated in figure 2.8, different strategies for the vertex clustering can be applied to build these event types in the graph. For example, the targets of these events could always start a new branch. This strategy applied to a split event does not need further decision making, as the newly created branches are part of the tree of the splitting vertex. However, the same approach used to build a merge event between trees already requires a process that decides to which of the merging trees the new branch should belong, as a branch cannot exist outside of a tree. To avoid this decision making process an alternative would be to not only branch out the target, but seed a new tree all together.

An alternative vertex clustering approach aims to continue branches in the building of these event types. In case of a split event, the target vertex that continues the branch of the splitting source vertex would still be interpreted as the same structure after the split rather than a new structure which starts its lifetime after the split. Depending on the split order, multiple target candidates exist to continue the branch of the splitting vertex. Therefore, a

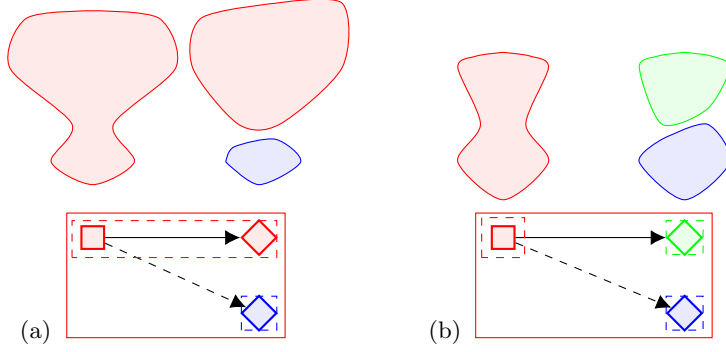


Figure 2.8: Different vertex clustering strategies to build a first order split in the graph. In (a) the branch of the splitting vertex is continued by one of its target vertices (continuation of the parent branch). Contrary, each of the targets in (b) starts a new branch, while the branch of the splitting vertex is terminated (out branching of the child branch). In both building variants all involved branches are part of the same existing tree. Solid arrows indicate a dominant edge, while dashed arrows represent a non-dominant edge.

strategy is needed to select the dominant correspondence (edge) of the split whose target is continuing the parent branch. The decision can be based on multiple criteria, however the result should consider that this selected target will be interpreted as the source structure after the split. Analogously, for a merge event, a dominant source needs to be selected, so that the merged target continues the branch of the dominant source. For a compound event, the decision-making process is applied to the sub-split and sub-merge events to select dominant sources and targets.

To allow a decision based on multiple criteria, a total score is calculated for each of the candidate correspondences as a weighted sum of scores of this correspondence for each criterion. Hence, the total score of the i -th candidate correspondence based on n criteria is defined as

$$s_{\text{total},i} = \sum_{j=0}^n \omega_j s_{ij}, \quad \sum_{j=0}^n \omega_j = 1, \quad s_{ij} \in [0, 1] \quad (2.12)$$

where ω_j is the specified weight for the j -th criterion and s_{ij} is the score of the i -th candidate in the j -th criterion, which is bounded by $[0, 1]$. Since the dominant target of the (sub)split event will be seen as the successor of the splitting source, and the dominant source of a (sub)merge event will be interpreted as the predecessor of the merged target, a criterion to decide on the dominant edge(s) of an event can be based on the comparison between the volume of candidate and the reference volume. For a (sub)split, the reference is the splitting source, whereas for a (sub)merge, the reference is the merged target. The score of the i -th candidate in this volume-based criterion is determined as

$$s_{i,\text{volume}} = \frac{\min(V_i, V_{\text{ref}})}{\max(V_i, V_{\text{ref}})} \quad (2.13)$$

Other criteria based on any physical quantity of the structures could be introduced in the same way as the volume-based criterion. In addition to the volume-based criterion, the confidence value of each correspondence in the event can be used as an score for the candidate as well. This way, a dominant edge is linked to a correspondence of high confidence. The confidence includes a contribution from the geometrical signature due to the NN search. However a criteria explicitly based on shape could be added.

In contrast to dominant target candidates for (sub)splits, the history of source candidates for (sub)merge events is available and can be utilized to form another criterion based on the lifetime of the source structure before the (sub)merge event. To assign a lifetime-based score bounded in $[0, 1]$, the score of the i -th source candidate is defined as

$$s_{i,\text{lifetime}} = 1 - e^{-\frac{1}{2}(t_i - t_{i,0})} \quad (2.14)$$

, where $t_{i,0}$ is the time of the frame in which the i -th source candidate is first present as a individual structure. If two source candidates differ by Δt in their lifetime, their difference in the lifetime score depends on the lifetime of the structures due to the exponential slope of the score. Two long-living source structures differing in lifetime by a small fraction of both structures lifetime Δt obtain a similar lifetime score and the decision will be dominated by another criterion.

After the scores of each criterion have been determined for each candidate, the candidate that maximizes the total score $s_{i,\text{total}}$ is selected as the dominant source or target of the (sub)merge or (sub)split event. While the dominant target of a (sub)split continues the parent branch, each non-dominant target generates a new branch within the same tree. The outdegree of a splitting source vertex equals the number of targets. Contrary to a split, a merge can connect branches of the same or different trees. For a merge between trees selecting the dominant source, not only specifies the continued branch but also answers the question to which tree the merged target is assigned since the continued branch is uniquely associated with a tree. The branches of the non-dominant sources of a merge event are terminated. The indegree of the merged target equals the number of sources of the merge.

Finally, a hybrid vertex clustering approach can be used to outbranch targets of split or merge events if their total score or specified subscore is below a threshold. The hybrid approach especially useful for higher order events, where the targets may be interpreted as individual structures due to the multiple interactions of the event. For example, if a source generates multiple approximately equally sized targets in a higher order split, the targets may be outbranched.

While post-processing the graph data structure resulting from the application of the tracking algorithm, the branches can be classified based on the connectivity of their start and end vertices as suggested by Lozano-Durán & Jiménez (2014). The start vertex of a primary branch results from a creation event, i.e. it has no antecessor, whereas its end vertex is a disappearing structure, i.e. it has no successor. Branches that do not satisfy both criteria are considered secondary branches. More specifically, incoming branches have start vertices without an antecessor, but their end vertex has a successor in another branch. Their counterpart, outgoing branches, are characterized by having a start vertex with antecessor, but having an end vertex without successor. In contrast, connecting and re-connecting branches are described by start and end vertices that are connected to other branches, as illustrated in figure 2.9. An end vertex of a re-connecting branch is connected to the same branch as its start vertex, whereas the end vertex of a connecting branch is connected to a different branch than its start vertex. Note that the connections between branches need not be regular split or merge events, but can also be more complex compound events. For example, an incoming branch could also end in a compound event rather than a pure merge. Therefore, the branches

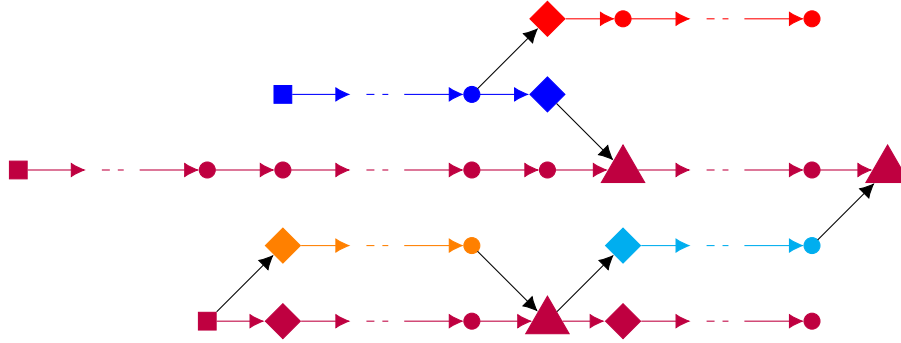


Figure 2.9: Classification of branches based on how they emerge and disappear. Start and end vertices of primary branches (—) do not have an antecessor or successor, respectively. The start vertex of an incoming branch (—) has no antecessor, but its end vertex has at least one successor, while the opposite scenario defines an outgoing branch (—). The start vertices of connecting (—) and re-connecting (---) branches have at least one antecessor and their end vertices have at least one successor. The end vertex of a reconnecting branch is connected to a vertex of a branch from which the reconnecting branch originated, while the end vertex of a connecting branch is connected to a separate branch. —> indicates branch connecting edges. Figure adapted from Lozano-Durán & Jiménez (2014).

are classified by the indegree of their start vertex and the outdegree of their end vertex, rather than the event types these vertices participate in.

2.10 Graph mining for pattern recognition in flow feature evolution

The application of the flow feature tracking algorithm described in the previous sections results in a graph data structure representing the temporal evolution and interactions of all individually tracked structures. To extract common *patterns* in the structure evolution, data mining techniques can be applied to its graph representation. Mined patterns are subgraphs of the tracking graph which occur frequently and hence are of significance for the flow feature evolution. While the input graph obtained from the tracking may have disjoint components, patterns are required to be connected subgraphs. In contrast to the frame-to-frame events detected by the correspondence search, patterns span multiple frames and therefore have a certain temporal extent. The actual occurrences of a pattern in the input graph are denoted as the *instances* of the pattern, while the pattern itself refers to the general definition of the subgraph.

One of the established graph mining approaches is the Subdue graph-based relational learning system (Cook & Holder, 1993; Holder *et al.*, 1994; Ketkar *et al.*, 2005). The following describes how the Subdue algorithm is adapted in the present work to recognize common patterns in the temporal interactions of flow features. One of the common challenges of graph mining approaches is the immense number of potentially interesting subgraphs extracted from large graphs. In contrast to mining techniques focusing on frequent subgraph discovery, such as the gSpan algorithm (Yan & Han, 2002), Subdue, in its unsupervised form, searches for patterns which best compress the graph dataset based on the minimum description length (MDL) principle (Rissanen, 1998) and typically reports fewer, but ideally more relevant patterns (Cook & Holder, 2006). To further reduce the number of extracted patterns

by allowing slight variations between instances of the same pattern, Subdue applies an inexact graph matching between a pattern and its instances (approximate graph isomorphism).

During the flow feature evolution, continuation events typically occur much more frequently than all other event types. Consequently, most subgraphs extracted from a graph representing the feature evolution are also dominated by continuation events. These subgraphs have a high value in terms of the overall graph compression. Therefore the majority of extracted patterns by Subdue in its unsupervised form are continuation-dominated and less interesting in terms of interactions between structures. Hence, a number of measures are introduced to guide the Subdue pattern search towards patterns which allow a certain amount of continuations, but focus on structure interactions and events other than continuations.

As a pre-step to the Subdue pattern search, the tracking graph is processed by *labeling functions* which assign additional attributes to the nodes and edges of the graph. In a *labeled graph* the pattern mining not only takes the connectivity (structure) of the extracted subgraphs into account, but also the attributes of the involved nodes and edges. To illustrate the graph labeling process, a synthetic tracking graph is considered in fig. 2.10.

The first two node attributes to be used in the Subdue pattern mining are the in- and outdegree of the node in the tracking graph. The node degrees significantly constrain the instance matching. For example, without these attributes, a first-order split could be matched with a sub-split of a second-order split. The unequal outdegree of the splitting node prevents matching those two subgraphs to the same pattern. A third node attribute evaluates the forward and backward distance of the node to another node involved in an event other than continuation. The attribute, denoted as *continuation attribute* in the following, has a value of 0 if the node is closer than a user-specified limit to a source or target node of a non-continuation event, otherwise the value of this node attribute is 1. The purpose of this continuation node attribute is to guide the pattern search towards patterns including events other than continuations as explained below in more detail.

In addition to node labels, attributes are also assigned to the edges of the tracking graph to be accounted for during the pattern mining. For each edge, the outdegree of the source node of the edge and the indegree of the target node of the edge are used as two edge attributes. An additional attribute stores the information whether or not the edge is a dominant edge in the tracking graph. This edge attribute preserves the tree-branch subgraph hierarchy of the tracking graph in the pattern mining process. Assigning the described attributes to the nodes and edges of the synthetic tracking graph of fig. 2.10 results in the labeled graph of fig. 2.11. Contrary to the previously described attributes, which are based on the properties of the tracking graph itself, attributes used in the pattern mining process can also be based on information extrinsic to the graph. For instance, a node attribute can describe the shape of the structure that the node represents. Introducing such an attribute places a geometric constraint on the pattern mining process. Rather than using the geometric signature of a structure directly as node attribute, the attribute can be obtained from clustering all tracked structures in the geometric feature space. The ID of the cluster to which the structure belongs to is then assigned to the geometric node attribute. Attributes extrinsic to the graph are not utilized in the demonstration of the graph mining using the synthetic graph, but will be employed in mining the graph representing vortical structures in mixing layers in chapter §4.

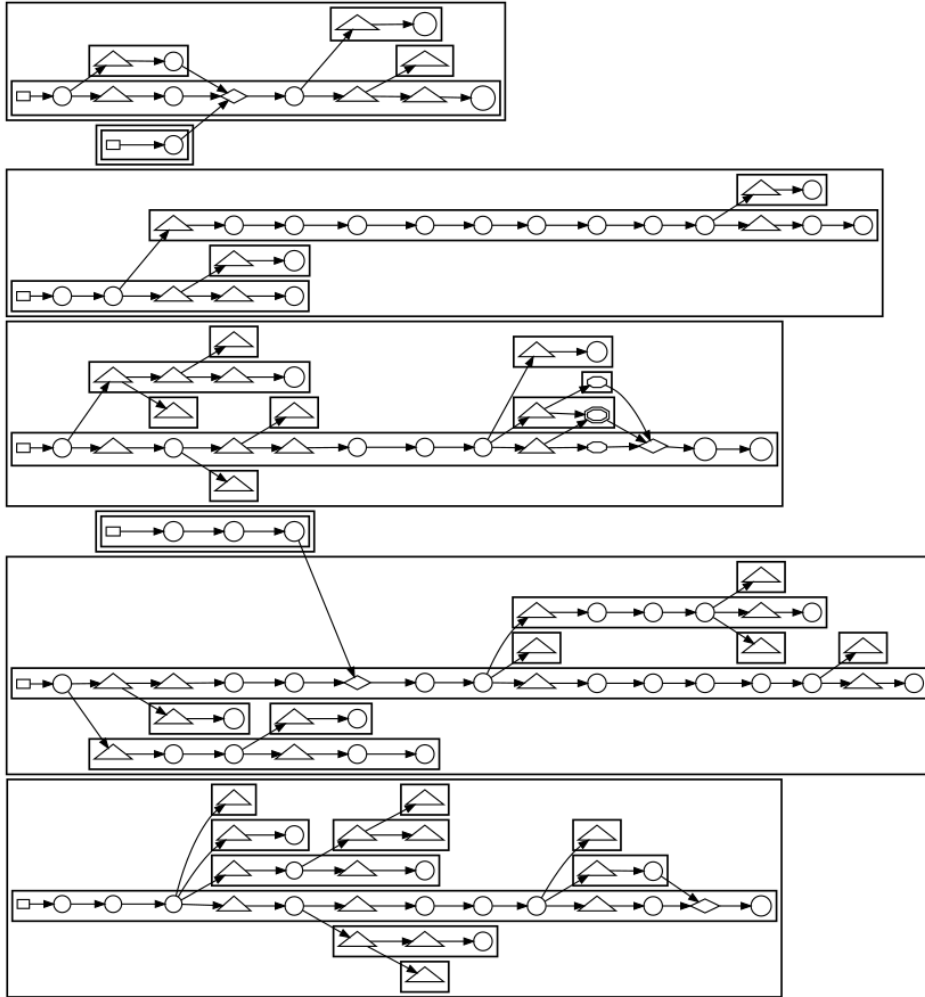


Figure 2.10: Synthetic graph used to demonstrate the graph labeling process and serving as input for the Subdue graph miner. Boxes indicate the tree-branch subgraph hierarchy of the graph. Node shapes symbolize event types as detailed in fig. 2.7.

Node and edge attributes other than those introduced by the labeling functions, such as the structure ID or time attribute of a node, are ignored when the tracking graph is fed into the Subdue algorithm. The initial state of the pattern search is the set of all uniquely labeled nodes of the tracking graph. Subdue applies a variant of beam search to expand those pattern candidates: pattern candidates are extended in all possible ways by a single edge and a vertex. The obtained pattern candidates are evaluated and ordered in terms of graph compression (or domain-specific rules as described below). Only the top beam patterns are expanded in the next iteration of the pattern candidate extension, where the beam width is a user-specified input value. This pattern growth process terminates when the search space is exhausted and no further growth can be realized. To focus the pattern candidates on some sort of event activity other than solely continuations, a barrier is introduced into Subdue's pattern growth approach by utilizing the previously introduced continuation node attribute: Instead of extending the pattern candidate in all possible ways, only those extensions are allowed which add nodes to the pattern which have the same continuation attribute value as the nodes already in the pattern.

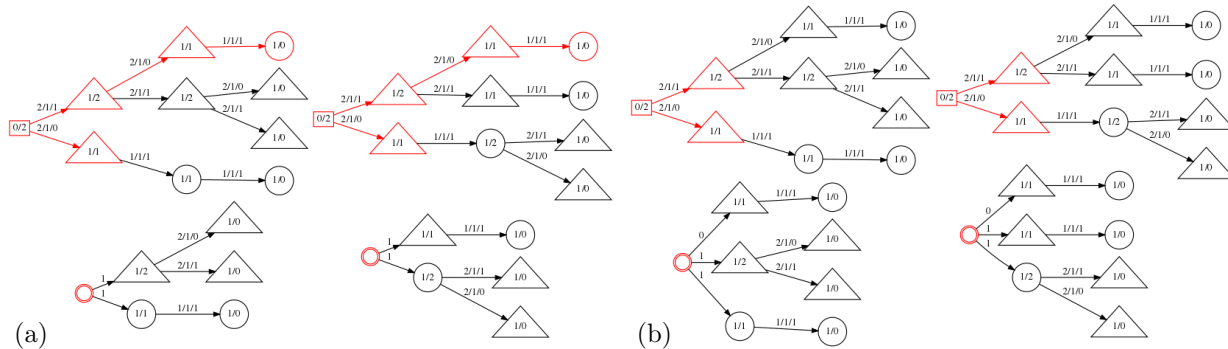


Figure 2.12: Illustrative split-dominated graphs labeled with in- and outdegree node attributes and in-, outdegree and dominant edge attributes. The two input graphs in the top row of (a) are similar but not isomorphic. Compression of the subgraphs highlighted in red leads to two isomorphic graphs in the bottom row. Compression of a different subgraph selection of the same two input graphs also generates two isomorphic graphs in (b). Figure continued in 2.13 (a)

Once the pattern search terminates, a graph compression procedure is invoked which substitutes all instances of the *best pattern* (based on the evaluation method, see below) in the tracking graph by single nodes, illustrated in fig. 2.12 and 2.13. Those compressed nodes represent the pattern in the compressed graph. During the compression procedure, edges, incoming or outgoing from best pattern instances, are redirected to be incoming to or outgoing from the compressed node. The graph compression operation also affects the node and edge attributes of the graph. In contrast to regular nodes, the compressed nodes do not carry the two node degree attributes, but contain an attribute indicative of the the pattern this compressed node is an instance of. The continuation node attribute value is inherited from the nodes the instance compressed. As a consequence, compressed nodes can only be matched with other compressed instances of the same pattern. The source outdegree and target indegree attribute are removed from the edges incoming to or outgoing from compressed nodes, respectively, but the dominant edge attribute is kept to maintain the tree-branch hierarchy of the graph. After compressing the graph, the Subdue pattern search is carried out in another iteration on the compressed graph. This process is repeated, while a list of patterns ordered by value is reported after each iteration and always the best pattern of the current iteration is used for compression. (The Subdue algorithm is applied iteratively on the successively compressed graphs, using the best pattern for compression in each iteration.) Due to the compression the graph size is iteratively reduced and Subdue has a higher chance to find larger pattern as pattern can involve compressed nodes. The pattern search of each iteration can be constrained by a minimum and maximum number of edges a pattern needs to consist of. A higher minimum number of edges can ensure a certain amount of complexity in the pattern, especially if only a few iterations of the Subdue algorithm are carried out. However, smaller patterns found in early iterations can serve as building blocks of complex pattern of later iterations. It's worth noting, that graph compression can disrupt the temporal ordering of the tracking graph and also generate parallel edges between nodes, which makes the graph a directed multi-graph as in the example shown in fig. 2.13b. In contrast to the original Subdue version, the creation of self-loops (edges where the source and target is the same node) is prevented during the graph compression to avoid paths of infinite length.

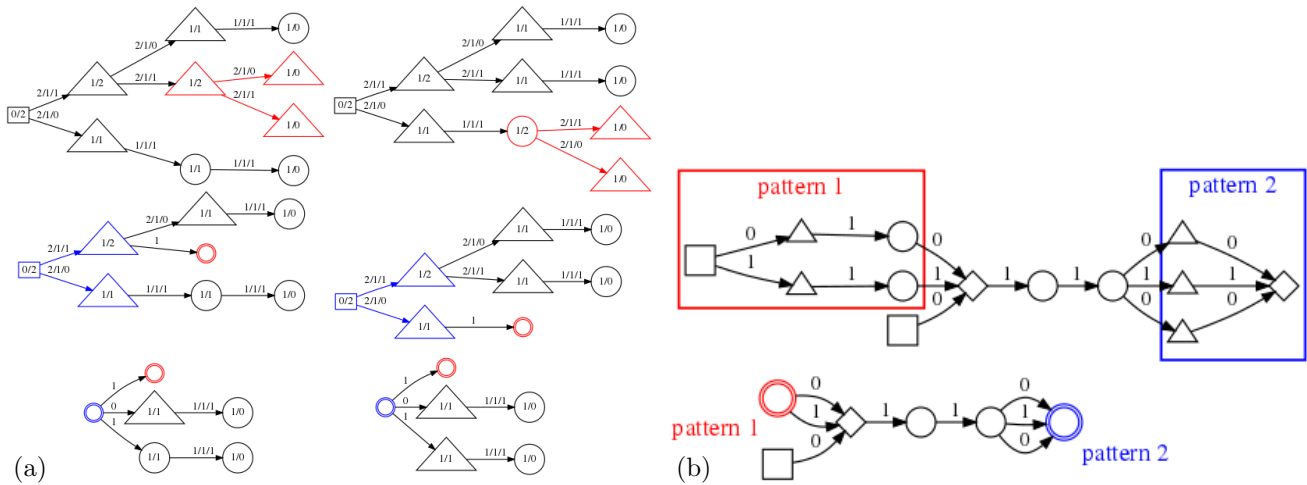


Figure 2.13: (a) Iterative graph compression leading to isomorphism.(b) Generation of parallel edges through partial compression of reconnecting branches. The example illustrates the preservation of the dominant edge attribute during compression.

2.10.1 Criteria for pattern evaluation

The pattern evaluation based on the graph compression using the MDL principle is replaced by a set of factors measuring the value of a pattern in terms of the application of the mining to flow feature evolution. First, a value of 0 is assigned to patterns consisting solely of nodes having a continuation attribute value of 1. Therefore those pattern candidates are never considered for expansion during the beam search. The *coverage* and *compactness* of a pattern are two measures suggested as domain-independent (Holder *et al.*, 1994). The coverage is defined as the ratio of the number of edges in the pattern to the total number of edges in the graph (at the current state of compression) multiplied by the number of instances of the pattern candidate (minus 1). Despite of differing from the MDL principle, the coverage value of a pattern still values the graph compression potential of the pattern. The compactness value of a pattern is defined as the ratio between edges and vertices of the pattern.

To favor patterns with structure interactions (rather than just continuation events), the product of all in- and outdegrees of the pattern's nodes is considered as another factor of the pattern evaluation. Here, the degrees of the nodes in the subgraph extracted from the graph at the current state of compression are used. The subgraph may contain compressed nodes which are not present in the original graph. For nodes having a degree of zero in the subgraph it is inspected if this degree value coincides with the degree value of the node in the original graph, if the node exists in the original graph. If this is the case, those nodes contribute with a factor of 2 to the product of node degrees to value creation and disappearance events, otherwise the node does not contribute to the node degree product. Note that a node may contribute with its outdegree but not with its indegree and vice versa. The final degree product of the subgraph is decreased by one before multiplication with the other factors of the pattern evaluation.

To favor pattern candidate of high temporal length, the length of the longest path in the subgraph is considered. A path in a directed graph is a sequence of connected, directed edges. For the longest path computation edges

are weighted in terms of event activity they are part of: For edges which are not incoming to and outgoing from compressed nodes the sum of the indegree and outdegree edge attribute decreased by one is used as an edge weight. This way, edges of continuations have a weight of 1 and edges of other event types get higher weights. For edges incoming to or outgoing from compressed nodes, the number of edges the node compressed is used as a weight to add attraction to those nodes. For nested compressed nodes the cumulative number of compressed edges is used as a weight. If the edge connects compressed nodes, the sum of the edges compressed by both nodes is used as a weight. The overall value of the pattern candidate is calculated as the product of its value in coverage and compression and its node degree-based and longest path-based values.

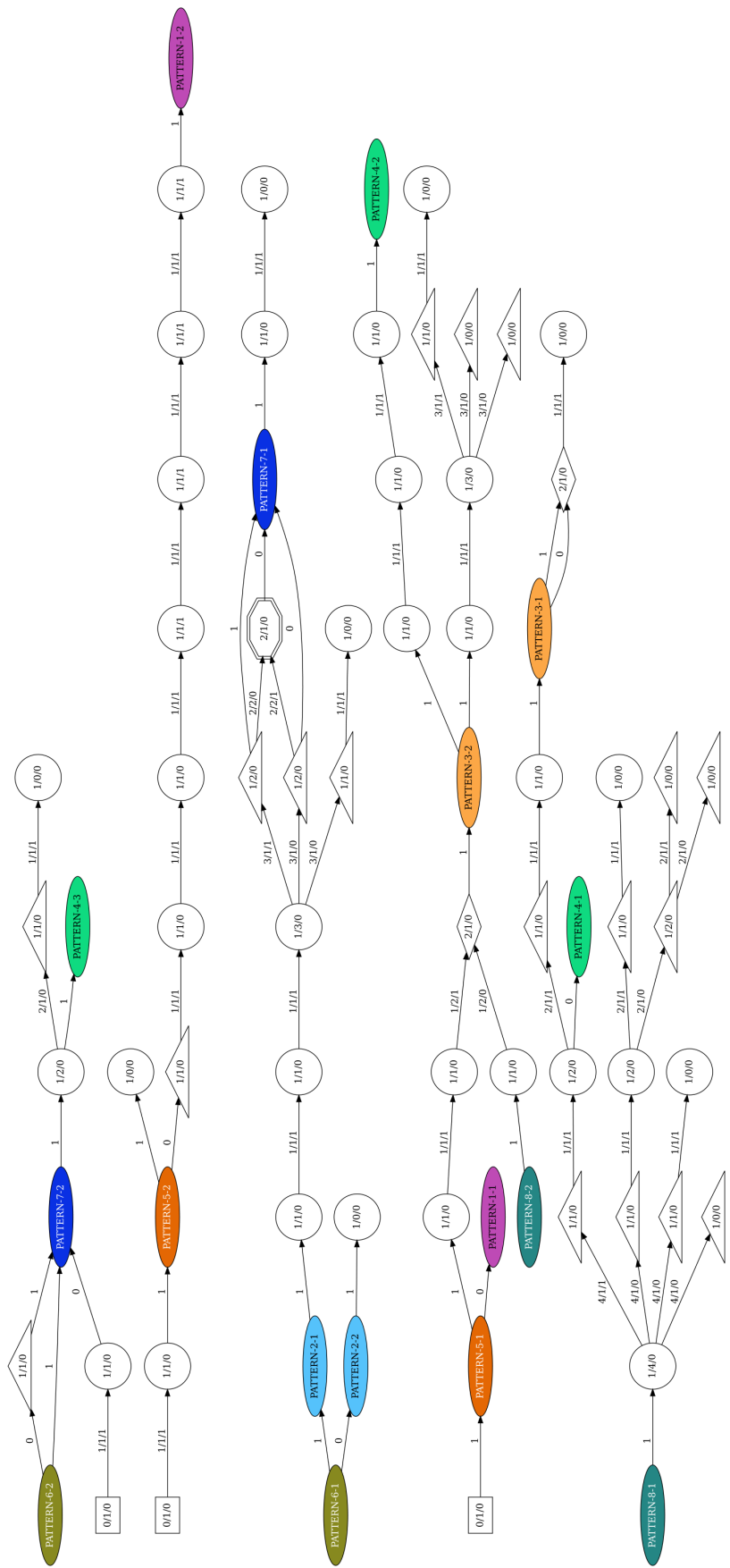
2.10.2 Application to a synthetic tracking graph

The Subdue algorithm with the described modifications and pattern evaluation rules is applied to mine patterns from the labeled synthetic tracking graph of fig. 2.11. The minimum number of edges constraint for a pattern is set to 2 and the node distance limited to determine the value of the node continuation attribute is also set to 2 in this application. The synthetic graph compressed by the two instances of the best pattern of the first iteration is drawn in fig. 2.14. With described settings, the best patterns of the Subdue iterations after the 8-th iteration have only one instance in the graph and the mining is terminated at this stage. The compressed tracking graph after the 8-th iteration is drawn in fig. 2.15, where the colored compressed nodes correspond to the best pattern of the particular iteration plotted in fig.2.16. The node and edge attributes of the best patterns show their values based on the original graph(fig. 2.11) rather than based on the connectivity of the pattern itself. When inspecting the best pattern of iterations 6 and 7, it is evident that the subgraphs only partially express the connectivity of the graph region from which they are extracted. For example, the merge node of the best pattern of iteration 7 has a indegree of 3 in the original graph, but in the extracted pattern the node has only 2 incoming edges. It is therefore useful to embed the instances of a pattern into their surroundings in the original graph to gain context information. Naturally, the embedding differs between the instances of the same pattern.

The two instances of the best pattern of iteration 7 are embedded in fig. 2.10.2. Nodes and edges drawn in solid indicate a pattern instance, while dashed nodes and edges show the connectivity of the instance to the surrounding input graph. One layer of predecessors and successors of the instance nodes is considered to represent the embedding. From the context information of the embedding it is clear that one of the source nodes of the second-order merge differs in its attributes from the other node and the extracted pattern only partially covers the merge.

If a pattern instance contains compressed nodes, those are recursively decompressed before embedding the instance into the surroundings of the the tracking graph. Two scenarios can lead to pattern instances involving compressed nodes: First, a compressed node can be added in the pattern growth during the pattern search. In this case, all instances of the pattern will have a compressed node representing an instance of the same best pattern of a previous iteration. The second way a compressed node can be integrated into a pattern instance is in cases where a pattern instance overlaps with an instances of a best pattern. This best pattern can be the best pattern of a previous, the same as the current or even a subsequent iteration. Here, the compressed node is only added to the instance which

overlaps with the other best pattern instance. Consequently, the variation between the pattern instances is increased. The second scenario is shown in fig. 2.18, where only the first instance of the pattern carries a compressed node. Depending on the overlap, it is also possible that different instances of the same pattern include compressed nodes representing different best patterns as illustrated in fig. 2.19.



Compressed graph after iteration 8

Figure 2.15: Compressed input graph after the 8-th Subdue iteration. The graph has been compressed by the best pattern of each iteration. Each best pattern is highlighted with a color corresponding to the pattern definitions in fig. 2.16

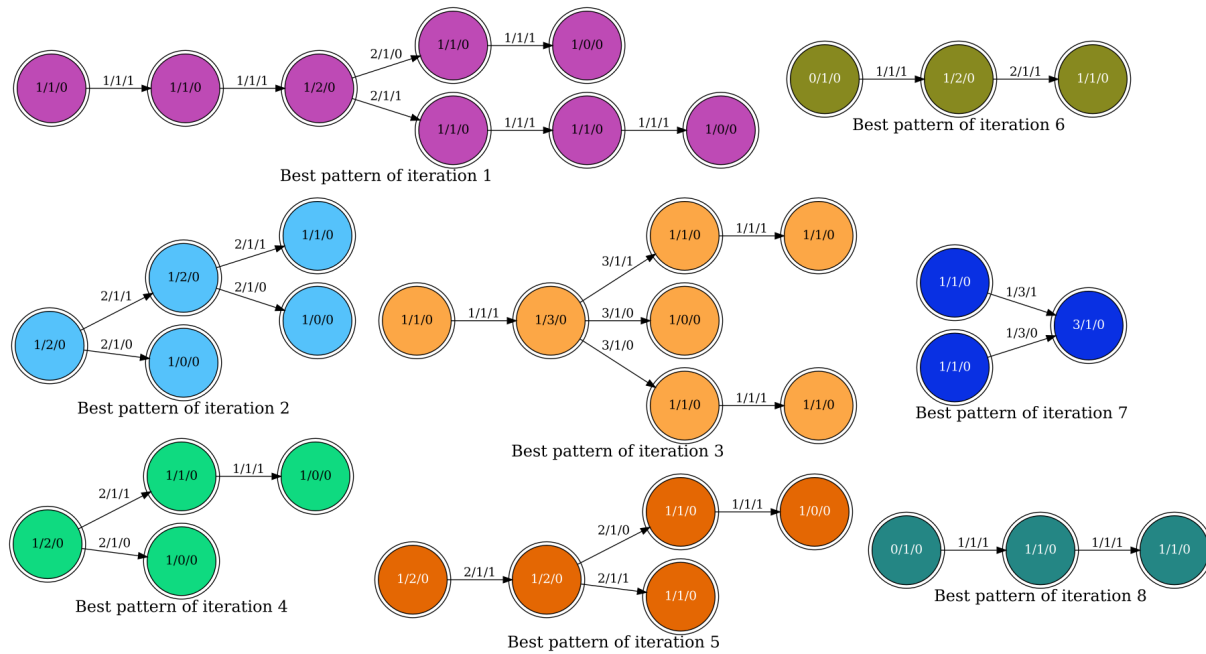


Figure 2.16: Best pattern of each iteration of the Subdue algorithm applied to the synthetic input graph. Colors corresponding to compressed nodes in fig. 2.15. Since node shapes can vary among the instances of a pattern, the double circle shape is used as a generic shape. Node and edge attributes show their values based on the original graph (fig. 2.11).

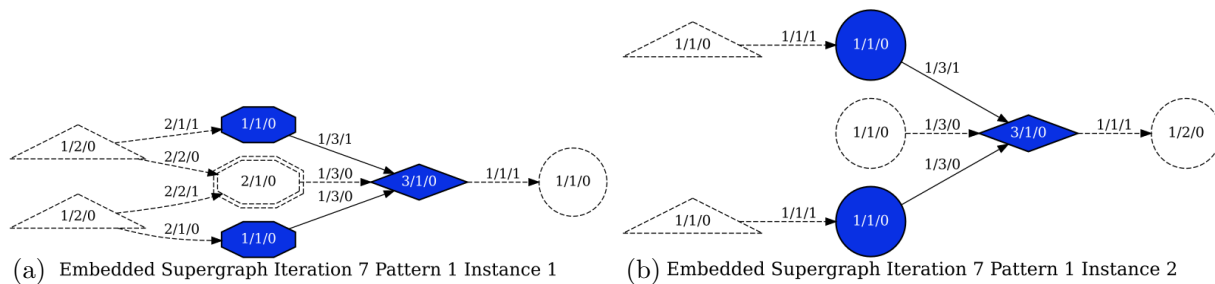
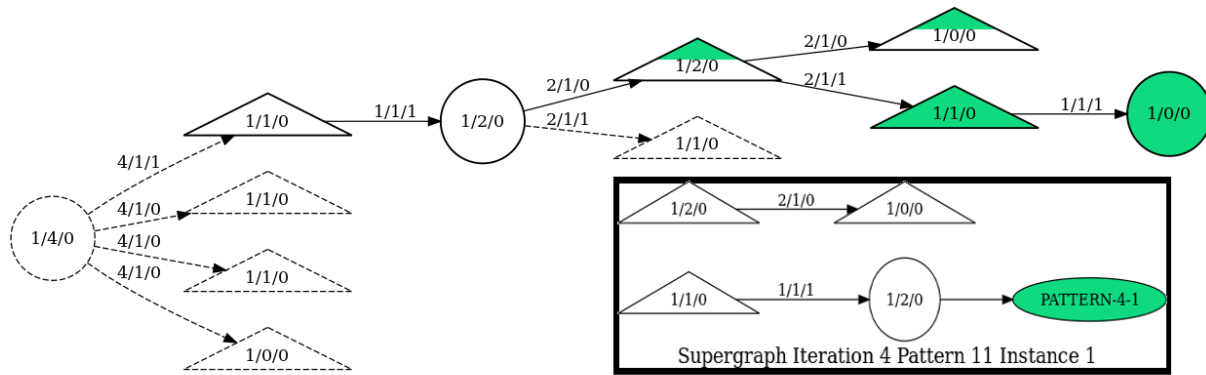
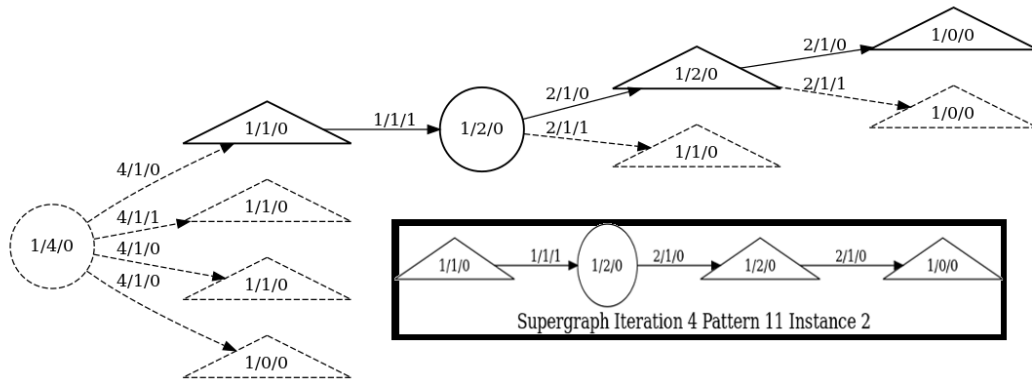


Figure 2.17: Embedding of the two instances of the best pattern of iteration 7. The nodes and edges drawn in solid indicate the actual pattern instances, while the dashed nodes and edges are the embedding of the instances in the original graph.

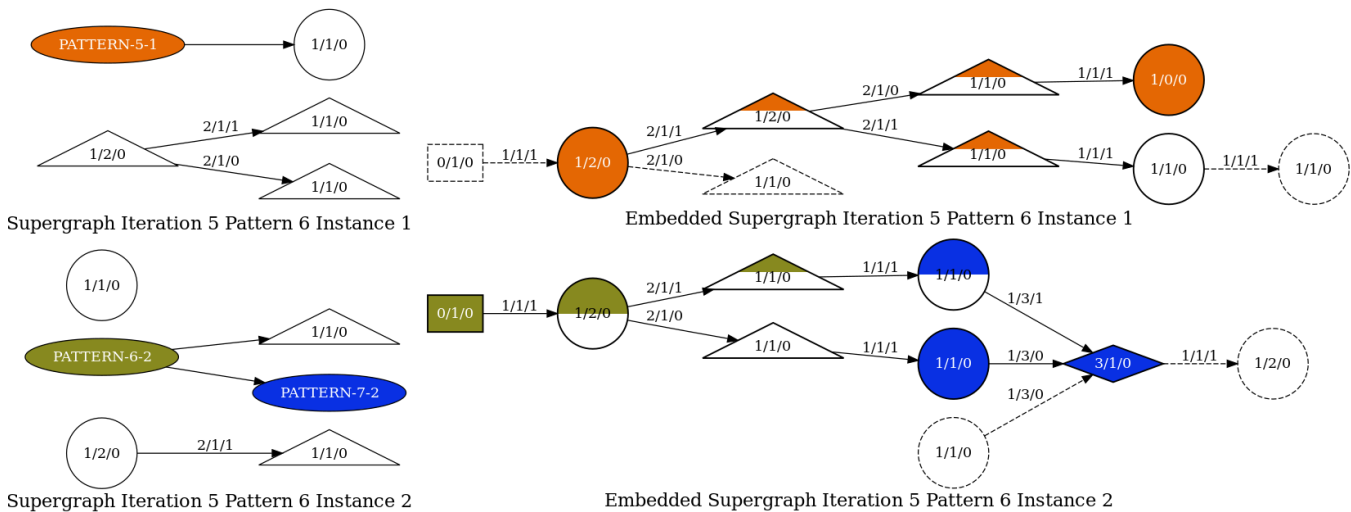


Embedded Supergraph Iteration 4 Pattern 11 Instance 1



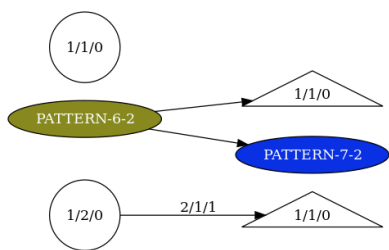
Embedded Supergraph Iteration 4 Pattern 11 Instance 2

Figure 2.18: Embedding and decompression of the two instances of pattern 11 of iteration 4. The first instances overlaps with an instance of the best pattern of iteration 4 in two nodes, highlighted in half-white half-green. Therefore the overlapping instance of the best pattern of iteration 4 is added to the first instance of pattern 11 of iteration 4. The definition of the best pattern of iteration 4 is given in 2.16. Inserting overlapping instances of best patterns into instances of another pattern increases the variation among these instances.

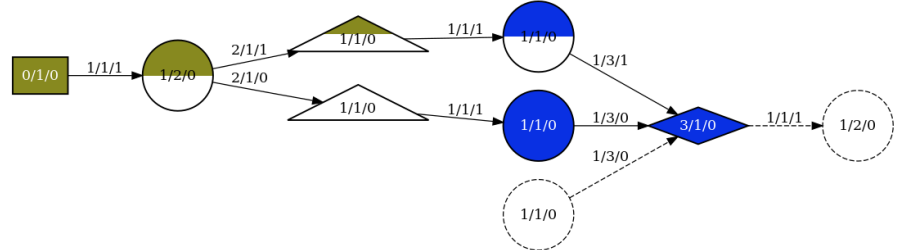


Supergraph Iteration 5 Pattern 6 Instance 1

Embedded Supergraph Iteration 5 Pattern 6 Instance 1



Supergraph Iteration 5 Pattern 6 Instance 2



Embedded Supergraph Iteration 5 Pattern 6 Instance 2

Figure 2.19: Embedding and decompression of the two instances of pattern 6 of iteration 5. The definitions of the involved best patterns are given in 2.16. Both instances contain compressed nodes of different pattern types, leading to quite a variation between the instances, when they are decompressed and embedded.

Chapter 3

Passive scalar structures in isotropic turbulence and shock-turbulence interaction

3.1 Introduction

A comprehensive understanding of compressibility effects on turbulent mixing processes is of great relevance to advance air-breathing super- and hypersonic propulsion systems (Andreopoulos *et al.*, 2000; Urzay, 2018). Shock wave-induced mixing enhancement has been proposed to accommodate the very short time scale requirements for nonpremixed combustion in these applications (Budzinski *et al.*, 1992). Furthermore, compressibility effects can originate from heat release in non-premixed and, especially, in premixed combustion (Bilger, 2004). Despite the identification of the scalar dissipation rate as a key parameter in non-premixed combustion several decades ago (Bilger, 1976; Pitsch & Steiner, 2000), the study of compressibility effects on scalar mixing has been the focus only in more recent direct numerical simulations (DNS) of homogeneous isotropic turbulence (HIT) (Pan & Scannapieco, 2010, 2011; Ni, 2015, 2016; Danish *et al.*, 2016; Panickacheril John *et al.*, 2020) and also shock-turbulence interaction (STI) (Tian *et al.*, 2017; Boukharfane *et al.*, 2018; Gao *et al.*, 2020). These numerical studies confirm that the amplification of turbulence kinetic energy (TKE) and transverse vorticity variance in conjunction with decreased turbulence length scales (Lee *et al.*, 1993, 1994; Ryu & Livescu, 2014) result in an increase of the scalar dissipation rate across a shock wave, as suggested by earlier experiments (Hesselink & Sturtevant, 1988; Agui *et al.*, 2005).

While the TKE amplification saturates (Lee *et al.*, 1997) for Mach number, M , larger than 3, the amplification of the scalar dissipation rate does not saturate at least up to $M = 5$, but is lowered for larger Taylor-Reynolds number, Re_λ , and turbulence Mach number, M_t (Gao *et al.*, 2020). The shock-induced enhancement of scalar mixing is expressed by a reduced scalar Taylor-microscale and an intensified decay of scalar variance in the post-shock region (Boukharfane *et al.*, 2018). The interaction of spatial velocity and scalar gradients contributes to the scalar transport (Danish *et al.*, 2016). The increase of scalar dissipation across the shock is associated with the evolution of the alignment between the scalar gradient, $\nabla\phi$, the vorticity, ω , and the strain-rate tensor eigenvectors, denoted as α , β and γ (Kerr, 1985; Ashurst *et al.*, 1987; Parashar *et al.*, 2017), and ordered by their respective eigenvalues from most extensional to most compressive, $\alpha \geq \beta \geq \gamma$. Recent statistical volumetric analyses of HIT (Ashurst *et al.*, 1987;

Danish *et al.*, 2016) and STI (Gao *et al.*, 2020; Boukharfane *et al.*, 2018) have shown that the scalar gradient is most aligned with the most compressive eigendirection γ of the strain-rate tensor, followed by α and β (Batchelor, 1959; Vedula *et al.*, 2001; Donzis *et al.*, 2010), consistent with Batchelor’s turbulence theory. The scalar gradient alignment with the eigenvectors of the solenoidal component of the strain-rate tensor shows very similar behavior to that of incompressible turbulence (John *et al.*, 2019). Scalar gradient alignment with the most extensional eigenvector, α , increases across the shock. For reactive scalars in premixed combustion, the α alignment is also known to increase due to dilatational motion induced by heat release in the reaction zone (Swaminathan & Grout, 2006; Kim & Pitsch, 2007). In contrast, the alignment with the intermediate and most compressive eigenvectors decreases due to enhanced scalar dissipation immediately after the shock. A weakening of alignment between the scalar gradient and the most compressive strain-rate eigenvector γ in flow regions of high dilatation was recently reported (Panickacheril John *et al.*, 2020). A strong anti-correlation found between the scalar gradient and the vorticity (Kerr, 1985) has been interpreted structurally as the passive scalar being wound around vortex tubes. The decrease of scalar gradient alignment with the intermediate eigenvector β and the vorticity ω is thus expected since the alignment between β and ω is strengthened by the shock-interaction (Livescu & Ryu, 2016; Boukharfane *et al.*, 2018). The alignment analysis has been utilized in the development of SGS models for the flux of a passive scalar (Pullin, 2000).

To complement statistical studies of the scalar mixing, the research objective this chapter is to investigate the dynamics of this process in decaying HIT and its enhancement by shock-interaction in STI from a structural point of view.

3.2 Numerical setup and initialization of passive scalar structures

The presented tracking algorithm is applied to study compressible turbulent mixing of passive scalars from a structural rather than volumetric view. To elucidate the effect of shock waves on the geometry of the scalar structures and mixing enhancement, we compare temporally decaying homogeneous isotropic turbulence (DHIT), and the canonical statistically stationary interaction of a nominally planar shock wave with spatially decaying isotropic turbulence (STI) at two different Mach numbers. Shock-capturing direct numerical simulations of these two flow configuration are performed based on a numerical method described in detail in Larsson & Lele (2009) and Larsson *et al.* (2013). To include passive scalars in these flow simulations the compressible Navier-Stokes equations for a calorically perfect gas are complemented with additional advection-diffusion equations

$$\partial_t(\rho\phi_m) + \partial_j [\rho(\phi_m u_j - D_m \partial_j \phi_m)] = 0 \quad (3.1)$$

for the passive scalar transport as described in detail in Gao *et al.* (2020). Here, ρ is the density, ϕ_m are the scalars, u_j the j -th velocity component ($j = 1, 2, 3$), and D_m the scalar diffusivities. The most relevant dimensionless parameters for the DHIT are the Taylor microscale Reynolds number, $Re_\lambda = \bar{\rho} u'_{\text{rms}} / \bar{\mu}$, where μ is the dynamic viscosity, $\lambda = [\overline{u'_2 u'_2} / (\partial_2 u_2)^2]^{(1/2)}$ is the Taylor microscale and $u'_{\text{rms}} = (\overline{u'_i u'_i} / 3)^{1/2}$ is the r.m.s value of the velocity fluctuation, and the turbulent Mach number, $M_t = \sqrt{3} u'_{\text{rms}} / \tilde{c}$ and additionally the mean flow Mach number, $M = \tilde{u}_{1,u} / \tilde{c}$, for

the STI simulation. In these definitions, Reynolds averages of a flow quantity, f , are indicated as \bar{f} , while \tilde{f} denotes a Favre averaged flow quantity. The corresponding fluctuations are $f' = f - \bar{f}$ and $f'' = f - \tilde{f}$. In the STI flow configuration, these dimensionless parameters are chosen as $Re_\lambda = 40$, $M_t = 0.2$ and $M = \{1.5, 3.0\}$, where Re_λ and M_t are taken just upstream of the shock location. The initial values of Re_λ and M_t in the corresponding DHIT configuration are chosen as $(Re_\lambda, M_t) = (46, 0.23)$ for the $M = 1.5$ STI and $(Re_\lambda, M_t) = (42, 0.21)$ for the $M = 3.0$ STI to account for the decay from the inlet to the shock location in the STI. The Schmidt number, defined as the ratio of the kinematic viscosity ν of the fluid and the diffusivity D_m of the m -th passive scalar, is chosen to be $Sc = 1$ in all presented DHIT and STI simulations, such as typically for supersonic mixing of gases. For the STI flow configuration, a resolution of $1280 \times 384 \times 384$ was found to be grid converged, see Gao *et al.* (2020). The grid resolution in each coordinate direction of the DHIT configuration matches the transverse resolution the STI simulation.

In a volumetric study of passive scalar mixing in (compressible) decaying isotropic turbulence the passive scalars are typically initialized to have defined (initial) spectra, such as $E(k) \propto k^4 e^{-2k^2/k_0^2}$ with $k_0 = 4$. In contrast, to study the mixing from a structural point of view, we initialize the passive scalar fields with a well-defined initial structure such that their isosurfaces present a specific initial shape. Precursor DHIT simulations are performed and the initial passive scalar fields are added to a time instant in which the turbulence is fully developed (velocity derivative skewness approximately -0.5, temporally increasing dissipation-based length scale $L_\varepsilon = K^{3/2}/\varepsilon$, $K = \widetilde{u_i'' u_i''}/2$, $\varepsilon = \overline{\tau_{ij} S_{ij}}/\bar{\rho}$) and the Taylor microscale Reynolds number and the turbulent Mach number match their target values of $Re_\lambda = 46$ (42) and $M_t = 0.23$ (0.21), respectively. This time instant of fully developed turbulence complemented with initialized passive scalar structures serves as the initial field of the DHIT simulations and also as the inflow database for the STI configuration, as described at the end of this section. To study geometric changes of canonical shapes in a turbulent background, for each simulation the passive scalar fields, denoted as ϕ_λ , $\phi_{2\lambda}$, and $\phi_{4\lambda}$, are initialized as collections of uniformly spaced spheres of uniform radius equal to one, two and four times, respectively, the initial Taylor microscale, λ , of the underlying decaying turbulence.

The scalar fields are initialized using a smooth radial profile that transitions between zero outside and unity inside the targeted spherical shapes (see figure 3.1). The smooth transition aims to avoid discontinuities in the scalar fields that can result in spurious numerical instabilities. The transition profile describes the scalar value $\phi(r)$ as a function of the distance r from the center of the ellipsoid as

$$\phi(r) = \frac{\phi_{\max} + \phi_{\min}}{2} + \frac{\phi_{\max} - \phi_{\min}}{2} \operatorname{erf} \left(2 - \frac{4(r - r_{\text{in}})}{r_{\text{out}} - r_{\text{in}}} \right) \quad (3.2)$$

, where r_{in} and r_{out} are the inner and outer radius, i.e. $\phi(r = r_{\text{in}}) = \phi_{\max}$ and $\phi(r = r_{\text{out}}) = \phi_{\min}$.

The three initial configurations of passive scalar structures whose temporal evolutions are studied in this work are summarized in table 3.1. The configurations of each structure type in the initial DHIT frame and in the corresponding STI inflow database are identical. The same sampling interval length of $\Delta t = \tau_0/12.7$ is used for the tracking for all structures in the STI $M = 1.5$ and corresponding DHIT flow configurations, where $\tau_0 = \lambda/u_{\text{rms}}$ is the initial large-eddy turnover time of the DHIT configuration. To account for the higher upstream velocity, the sampling

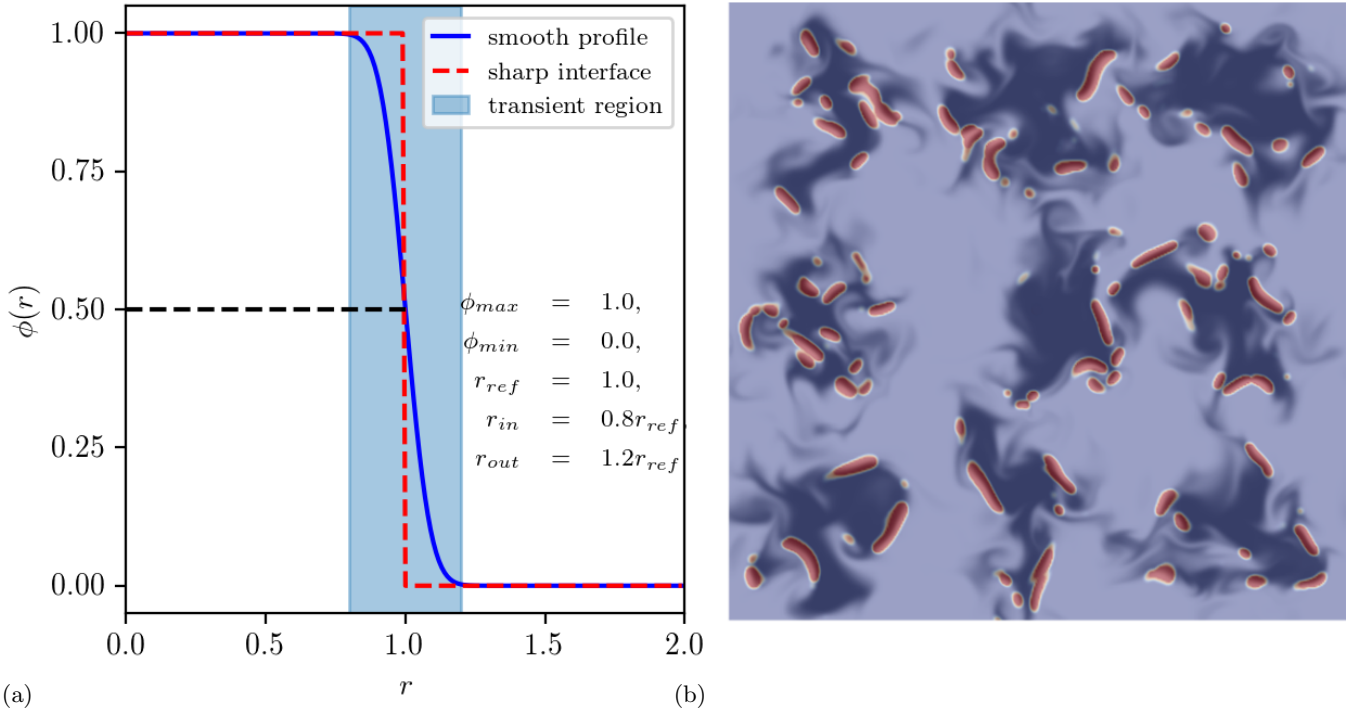


Figure 3.1: (a) Radial scalar profile at initialization. (b) Detected regions of scalar discontinuities (red) during the development from the initial condition in DHIT in a slice of the $\phi_{4\lambda}$ field.

Symbol	Radius	Transition	Spacing	Number
ϕ_λ	λ	5η	3λ	12^3
$\phi_{2\lambda}$	2λ	10η	6λ	6^3
$\phi_{4\lambda}$	4λ	20η	12λ	3^3

Table 3.1: Initial configuration of passive scalar structures used in the DHIT and STI configuration. Initialized number of structures per DHIT domain block. Radial transition length from ϕ_{\min} to ϕ_{\max} .

frequency is doubled in the $M = 3.0$ STI and in the corresponding DHIT. Figure 3.2 shows the initialized $\phi_{4\lambda}$ structures and their temporal evolution in the DHIT configuration at three different time instants. The application of the reconnection of surfaces intersecting periodic domain boundaries is demonstrated in 3.2 (c).

Despite of initializing the passive scalar structures using the smooth radial profile 3.2, large gradients are detected in the passive scalar fields by applying a jump-based scalar discontinuity sensor, figure 3.1. In domain regions marked by the scalar discontinuity sensor, a fifth-order weighted essentially non-oscillatory (WENO) scheme with HLL flux splitting, Ducros *et al.* (2000), is applied, while a sixth-order central difference scheme is applied elsewhere. Discontinuities in the hydrodynamic flow field near the shock (or shocklets, for sufficiently large M_t) as detected by a dilatation-vorticity based sensor ($-\theta > 1.2\omega_k\omega_k$) are analogously treated numerically.

Since the radial length $r_{\text{out}} - r_{\text{in}}$ in which the scalar initially transitions from ϕ_{\min} to ϕ_{\max} is a fix fraction of the nominal initial radius of the passive scalar structures, the region of non-zero initial scalar gradient thickens with increasing nominal radius (figure 3.3). As the structures deform after the initialization, stretching of the transitional region induces local increases of the scalar gradient magnitude, which are more pronounced the longer the initial

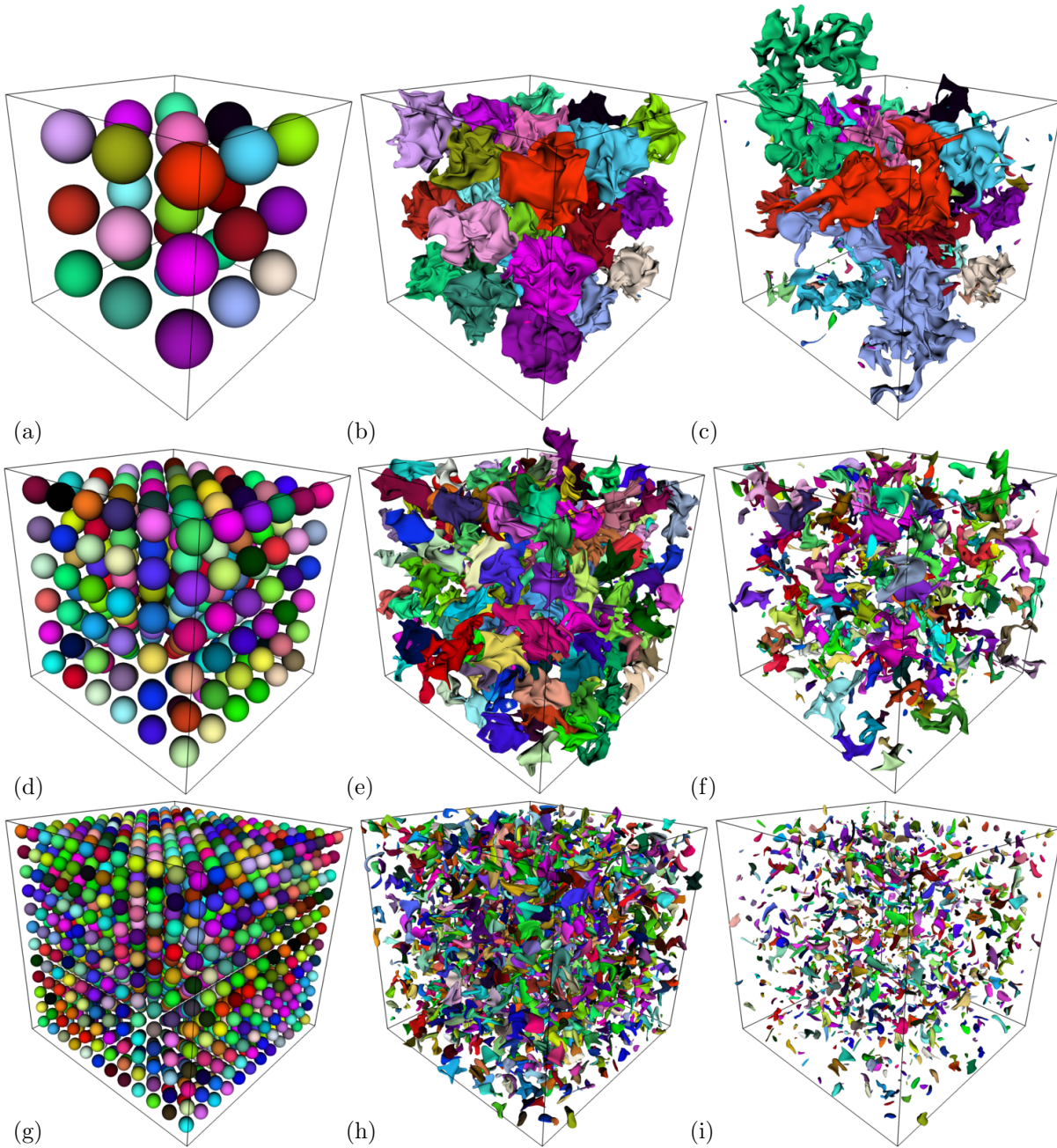


Figure 3.2: Evolution of structures educed from $\phi_{4\lambda}$ (a-c), $\phi_{2\lambda}$ (d-f) and ϕ_λ (g-i) scalars in DHIT captured in different frames from left to right: $t/\tau_0 = 0.0$ (a,d,g), $t/\tau_0 \approx 1.65$ (b,e,h), $t/\tau_0 = 4.5$ (c), $t/\tau_0 = 3.5$ (f), $t/\tau_0 = 2.4$ (i). Coloring indicates the identification of the individually tracked structures. Thus, color changes are associated with split and merge events. Periodic reconnection has been applied to structures crossing boundaries. For instance, in (c) the top part of the green structure is located outside the computational domain as it has been periodically reconnected to its lower part continuation inside the domain, leaving the void region on the opposite side (see also fig. 2.5 for an illustration of the reconnecting algorithm). The rendering shows the periodically reconnected structure at one of its possible locations, all of which are considered in the correspondence search.

transitional length is. The increased stretching of larger transitional regions enhances the mixing of the passive scalar field (Villermaux, 2019).

To generate the inflow database used for the STI simulation, three individual DHIT precursor simulations without passive scalars are performed and from each the time instant in which the Taylor-Reynolds number and turbulent Mach number reach their target values $Re_\lambda = 46(42)$ and $M_t = 0.23(0.21)$ to start the DHIT simulation including the passive scalar fields. From each of those three time instants two additional field are obtained by applying rotations with respect to the y and z coordinate axes. The nine obtained boxes are blended together following a technique introduced in Larsson (2009). After the blending operation the passive scalar structures are initialized on the obtained cuboid grid in the same way as previously described for the DHIT flow configuration, figure 3.4. The same initial geometry is used for the DHIT and STI study to make comparison between the passive scalar mixing in both flow configurations, table 3.1. The nominally planar shock wave is located stationarily a distance ≈ 2 downstream of the inlet, which will be taken as the origin of the x axis ($k_0x \approx 0$). To avoid acoustic reflections from the outflow boundary into the subsonic region of the domain, a sponge layer of length π is introduced before the outlet boundary (Larsson & Lele, 2009). Since damping terms are added to the Navier-Stokes equation in the sponge layer region, the flow field in the sponge layer is considered to be not physically meaningful. Therefore the effective downstream length of the STI configuration relevant to the tracking is reduced to the region starting at the position downstream of the inlet where the structures first become closed surfaces, which depends on the initial size of the structures, to the the starting position of the sponge layer.

To obtain individual passive scalar structures from the underlying field by iso-surfacing in principle any value $\phi_{\text{iso}} \in [\phi_{\text{min}} = 0.0, \phi_{\text{max}} = 1.0]$ can be used as an iso-value. Not only the shape of the passive scalar isosurfaces extracted at each frame of the simulation depend on the chosen contour value, but also the total number of extracted structures in each step. For the STI application, figure 3.5a reports the total number of extracted structures in a certain frame in the statistically stationary state for contour values ranging from ϕ_{min} to ϕ_{max} for the considered passive scalar fields $\phi_\lambda, \phi_{2\lambda}$ and $\phi_{4\lambda}$. The number of extracted structures peaks around $\phi_{\text{iso}} = 0.3$ and decreases for larger iso-values. The downstream length of the region in which structures are present is significantly affected by the choice of the iso-value. In general, a lower contour value leads to an increased lifetime of the scalar structures and hence an increased number of extracted structures. For $\phi_{\text{iso}} < 0.3$ very large connected regions are obtained and the number of extracted structures is reduced. Due to the structure deformation induced by the shock, the post-shock region is prone to the formation of widely connected isosurfaces as visualized in figure 3.5b. Large isosurfaces in the post-shock region are unfavorable in terms of tracking individual structures. Additionally, surfaces which span across the domain and connect to themselves via periodic boundaries have infinite extent, remaining open and therefore lead to a failure of their geometrical characterization. The size of the isosurface of largest extent is found to be quite sensitive to the chosen iso-value ϕ_{iso} and also depends on the choice of the initial spacing of the passive scalar structures. For the given spacing, an isovalue of $\phi_{\text{iso}} = 0.6$ is chosen to define the passive scalar structure in all presented configurations, which decreases the initial size of the structures insignificantly compared to the nominal radius but prevents the formation of large isosurfaces in the post-shock region. From all isosurfaces extracted for

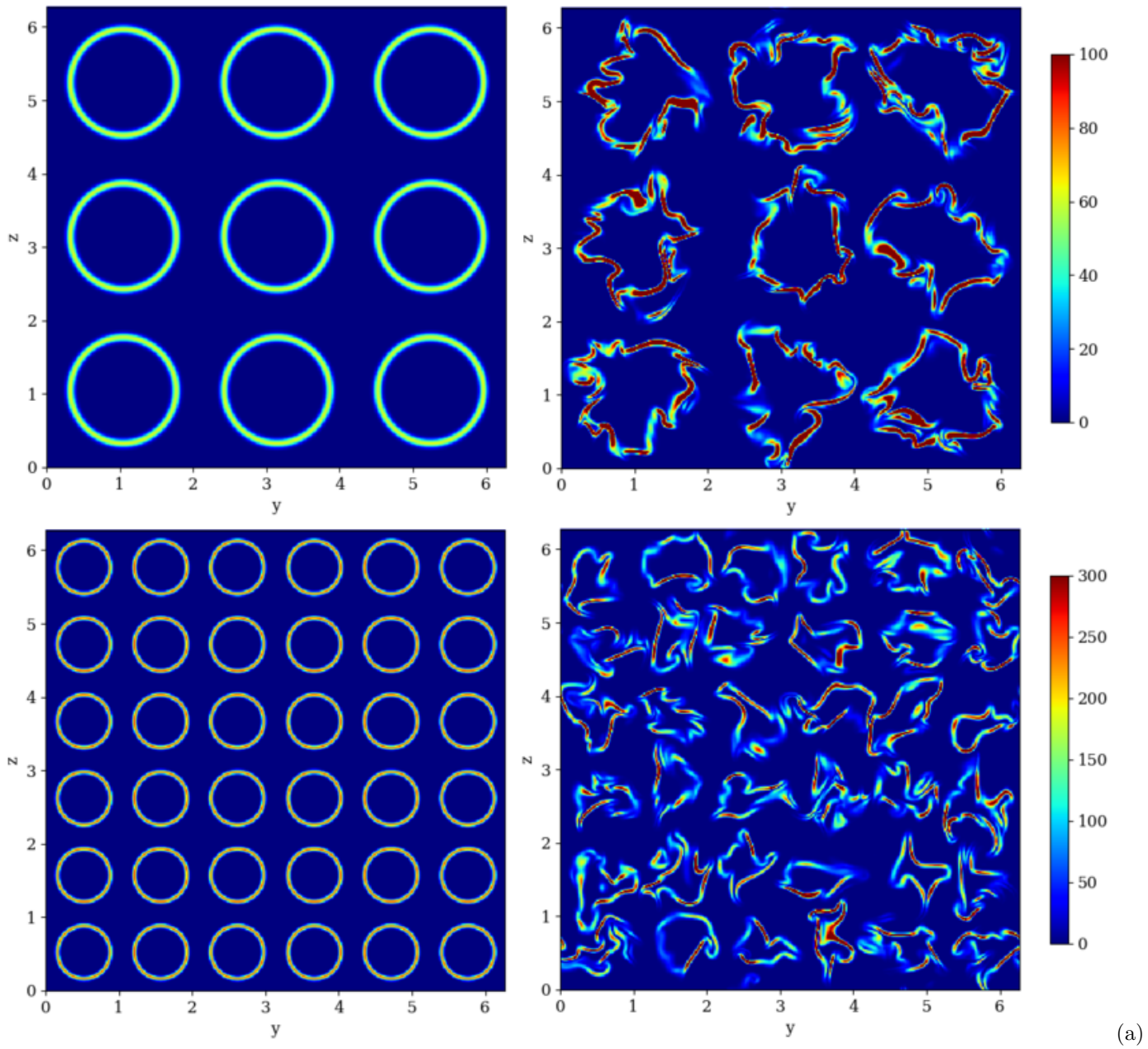


Figure 3.3: Scalar gradient of $\phi_{4\lambda}$ (top row) and $\phi_{2\lambda}$ (bottom row) visualized in plane cuts of the DHIT $M = 1.5$ domain: Initial transitional region $\phi_{4\lambda}$ (top,left), Transitional region $\phi_{4\lambda}$ at $t/\tau_0 = 0.98$ (top,right), Initial transitional region $\phi_{2\lambda}$ (bottom,left), Transitional region $\phi_{2\lambda}$ at $t/\tau_0 = 0.98$ (bottom,right).

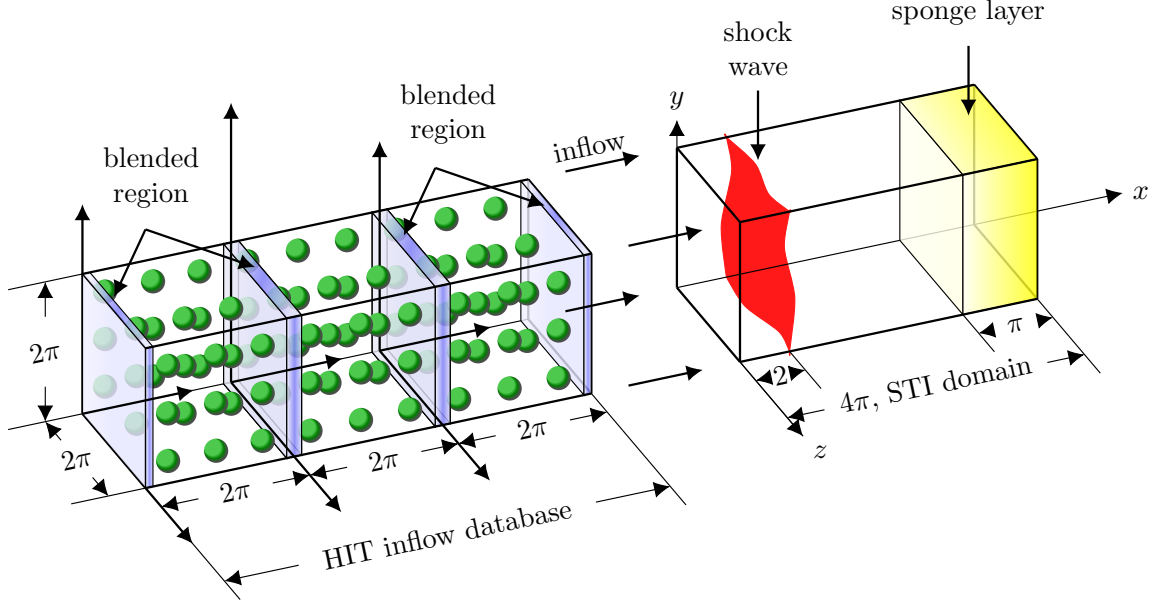


Figure 3.4: Problem setup of passive scalar mixing in STI and blended DHIT precursor simulations. Length of inflow database reduced in figure. Adapted from Gao *et al.* (2020).

		Nearest-neighbor search							Radius search		Correspondence constraints			
f_x	f_y	f_z	$f_{\hat{S}}$	$f_{\hat{C}}$	$f_{\hat{\lambda}}$	c_{NN}	ω_{NN}	ε_R	ω_R	l_{ref}	ε	ε_{rel}	ω	
$1/2\pi$	$1/2\pi$	$1/2\pi$	1	1	1	0.5	0.5	1.5	0.1	μ	0.2	0.4	0.1	

Table 3.2: Scaling factors, confidence value, and weight for the NN search. Tolerance and weight utilized in the radius search. Structure reference length, tolerance, relaxed tolerance and weight used for both OBB distance and surface proximity correspondence constraints, as defined in §2.4.2-2.5.

$\phi_{\text{iso}} = 0.6$, only those having a polygonal mesh consisting of at least 50 points will be considered in the geometrical characterization (and, hence, in the tracking), to ensure a reliable calculation of the differential-geometric properties.

The tracking methodology is applied to the analysis of structures of passive scalar mixing with the settings presented in tables 3.2-3.4. In the NN search, the spatial attributes are scaled using by the inverse of the transverse domain length, such that they are balanced with the geometrical attributes (table 3.2). This scaling gives more emphasis to the mean shock-normal dominant flow direction in the STI configuration. In the radius search, a large tolerance ε_R is applied to the OBB half-diagonal to form a search radius that captures all possible correspondence candidates. The same settings are used for both applied correspondence constraints. The obtained distances are compared to a fraction ε of the maximum of the μ values of source and target structures of the tested correspondence. For small structures, defined by having a number of surface triangulation points smaller than 1000, a relaxed fraction $\varepsilon_{\text{rel}} = 2\varepsilon$ is used to provide a higher tolerance. Table 3.3 lists the parameter settings for the event constraints ordered according to their sequence of application. First, a conservation-based constraint is applied to compound events where each extracted sub-event needs to satisfy a constraint threshold ε in terms of the estimated mass. A relaxed threshold ε_{rel} is used if the extracted sub-event is a continuation. Correspondences of the compound event with less confidence than c_{min} may be removed to resolve *double connections* (see §2.7). After a surface point constraint is deployed

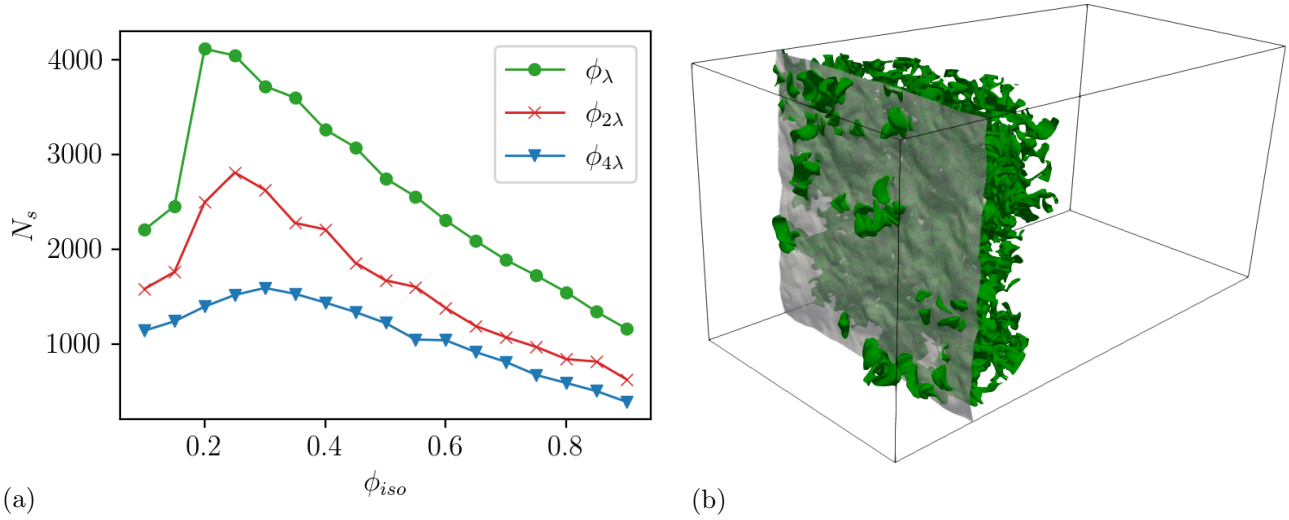


Figure 3.5: (a) Number isosurfaces N_s of each scalar field $\phi_\lambda, \phi_{2\lambda}, \phi_{4\lambda}$ for different isovalues ϕ_{iso} extracted from a frame in the statistically stationary state of the STI. (b) Formation of a ϕ_λ isosurface of largest extent (green) for $\phi_{iso} = 0.3$ in the post-shock region. Shock visualized as a dilatation isosurface (gray).

Conservation			Surface points	Conservation					
Compound			Merge, Sub-merge	Compound			Merge	Split	Continuation
ε	ε_{rel}	c_{min}	N_{pts}	ε	ε_{rel}	c_{min}	ε	ε	ε
0.5	0.25	0.75	250	0.4	0.15	0.8	0.6	0.3	0.2

Table 3.3: Settings of the event constraining. Type of the constraint and event type the constraint acts on. Event constraints are applied in sequence from left to right.

to test merge events and potentially simplify remaining compound events, another conservation-based constraint is applied to compound events with a less restrictive setting. Subsequently, merge, split and continuation events are tested by a conservation-based constraint that skips splitting and continuing source structures with less than 1000 surface triangulation points. The weights used for the dominant target/source selection for the construction of split, merge or compound events in the graph data structure are listed in table 3.4. Here, the conservation-based criteria use again the estimated mass of the structures involved in the event to build.

(Sub-) Split			(Sub-) Merge	
Conservation	Confidence	Lifetime	Conservation	Confidence
0.8	0.2	0.5	0.3	0.2

Table 3.4: Weights used for the dominant target/source selection to build (sub-)split/merge events in the graph data structure.

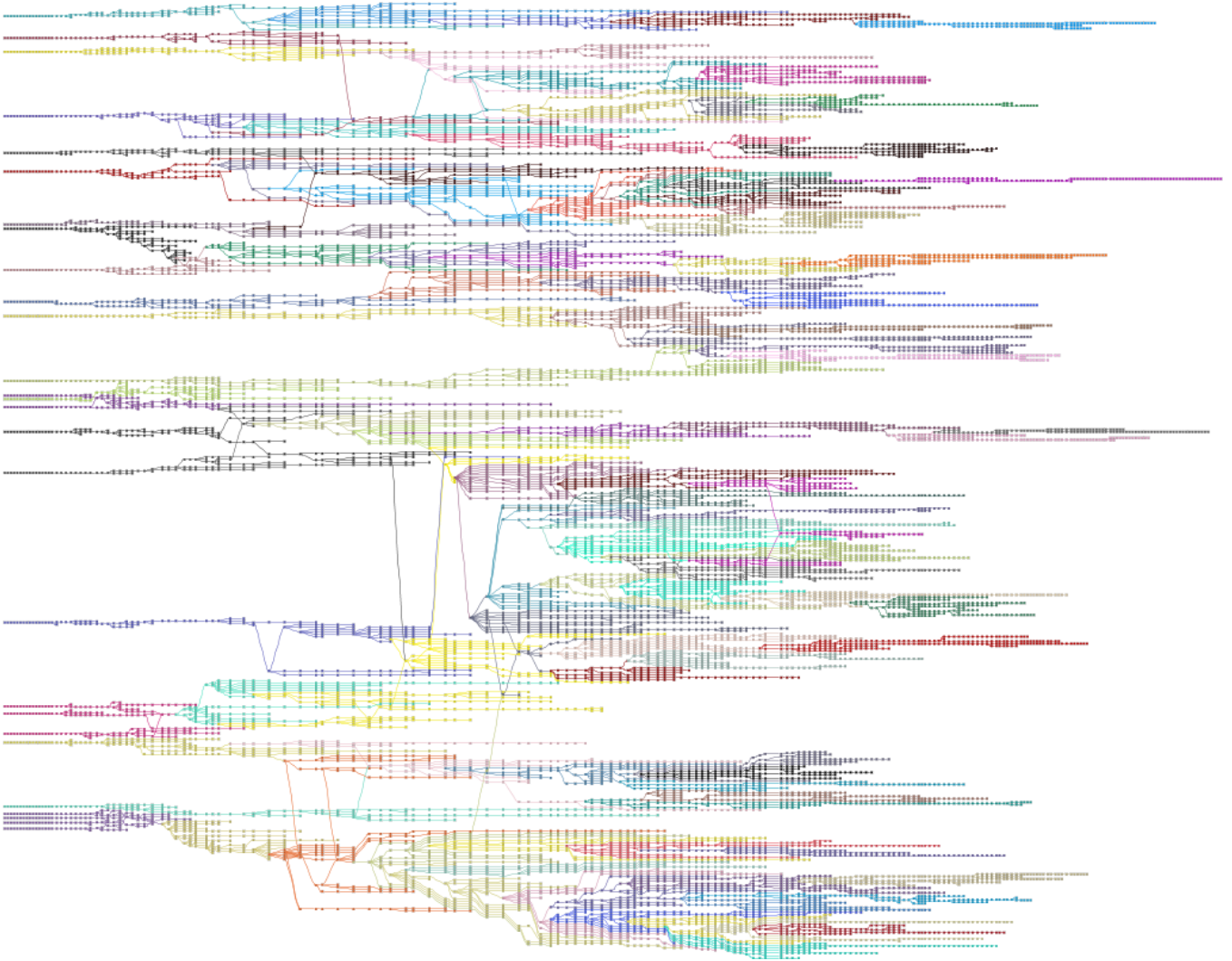


Figure 3.6: Hierarchical layout of the graph of the $\phi_{4\lambda}$ scalar in DHIT (evolving from left to right). Colors highlight communities as detected by edge betweenness clustering, Girvan & Newman (2002).

3.3 Downstream distribution of events

The graph representation of the temporal evolution of all tracked $\phi_{4\lambda}$ structures in the DHIT $M = 1.5$, figure 3.6, indicates that the majority of detected events are continuations of structures which evolve from source to target frames without interacting with others. Second to continuing structures, the graph is dominated by split and, consequently, disappearance events. The reduced lifetime of the $\phi_{2\lambda}$ structures compared to the $\phi_{4\lambda}$ structures of larger initial size is apparent in the reduced length of the branches of the graph representing the $\phi_{2\lambda}$ evolution in figure 3.7, as the sampling frequency is the same in both applications of the tracking. The number of outgoing branches relative to the number of primary branches is reduced in the graph representation of the $\phi_{2\lambda}$ structure evolution.

These findings motivated by the appearance of the graphs are quantified in the following by investigating the temporal or equivalently spatial (downstream) distribution of different event types. Since the STI configuration is statistically stationary, the temporal evolution of individual passive scalar structures is investigated by means of their

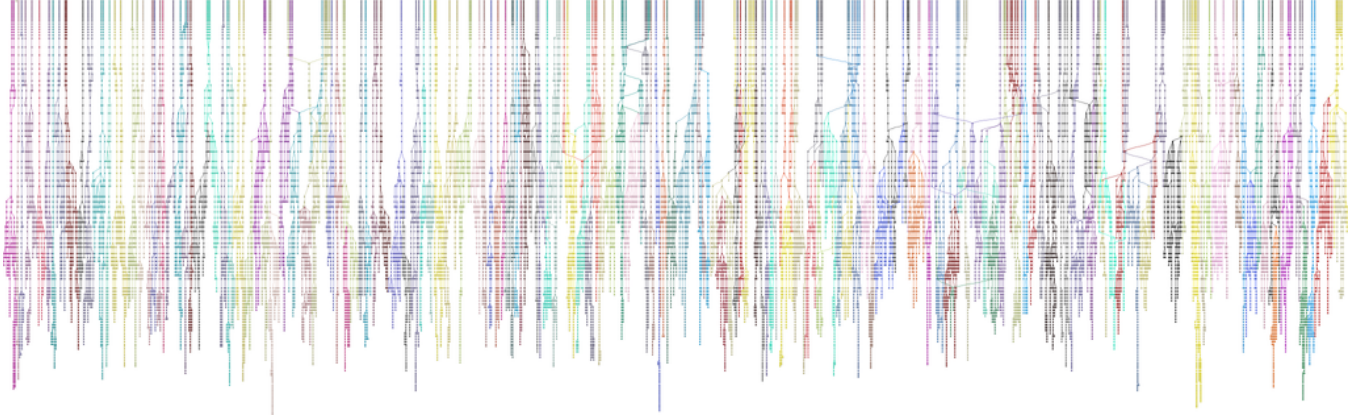


Figure 3.7: Hierarchical layout of the graph of the $\phi_{2\lambda}$ scalar in DHIT (evolving from top to bottom). Colors highlight communities as detected by edge betweenness clustering, Girvan & Newman (2002).

downstream evolution. Since the structure velocity in x is always positive the mapping from time to downstream location is monotonic, $t^n < t^{n+1} \Rightarrow x(t^n) < x(t^{n+1})$. To compare the structure evolutions in the temporally decaying DHIT configuration to those of the STI, the time coordinate of the DHIT configuration is translated into a equivalent downstream coordinate as

$$x_{\text{hit}} = \begin{cases} u_u t & \text{for } t < x_s/u_u \\ x_s + u_d(t - x_s/u_u) & \text{else} \end{cases} \quad (3.3)$$

, where x_s is the shock location and u_u and u_d are the mean downstream flow velocities upstream and downstream of the shock. The downstream distribution of different event types during structure evolution in the STI and DHIT configuration is presented in figure 3.8, where the downstream coordinate is normalized by the k_0 wavenumber and the initial Taylor micro-scale, λ , of the DHIT configuration. Here, the histogram count of number of events per downstream interval is normalized by the interval length and the number of incoming (STI) or initial (DHIT) structures.

Due to scalar dissipation, no creation of new structures are expected to occur from the underlying passive scalar field, as the scalar value stays below the threshold used for iso-surfacing. The only expected source of creation events are incoming structures of the inflow database of the STI configurations. Figure 3.8 shows the expected sharp peak of creation events close to the inlet boundary. The peak location is shifted downstream as the structures increase in initial size from ϕ_λ to $\phi_{4\lambda}$ and become closed surfaces at a later downstream location. A small number of creation events is detected in the post-shock region, especially for the $\phi_{4\lambda}$ structures. Those structures falsely detected as being newly created are small in volume and are small continuing structures or targets of split events in most cases.

The distributions of the other event types are also shifted downstream with increasing initial size leading to practically no disappearance, split or merge events for $\phi_{2\lambda}$ and $\phi_{4\lambda}$ structures in the pre-shock region. The number and distribution of disappearance events are linked to the number and distribution of split events as the majority of (early) splits have one dominant structure similar in size of the parent structure, see §3.8.1, and a number of smaller targets which disappear in close proximity to the split event. The peak of disappearance events is closer to the shock

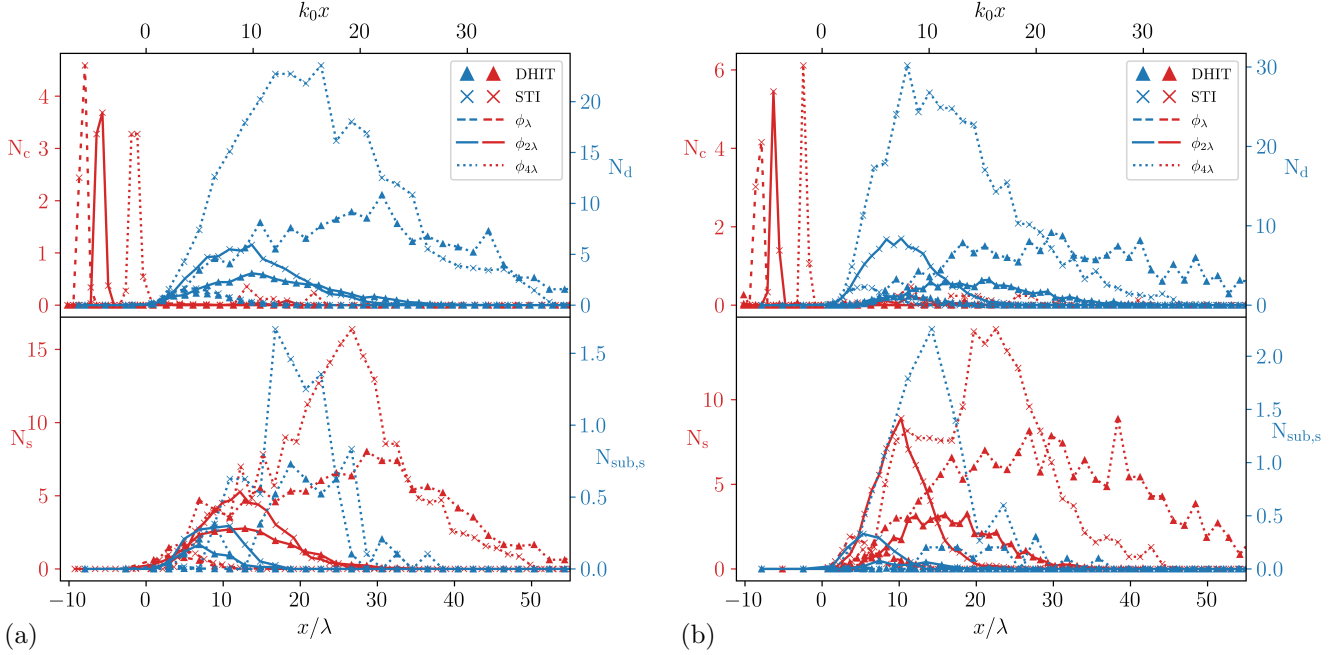


Figure 3.8: Downstream distribution of different event types for the $\phi_\lambda, \phi_{2\lambda}$ and $\phi_{4\lambda}$ passive scalar structures in DHIT and STI configurations for $M = 1.5$ (a) and $M = 3.0$ (b). Relative (normalized by bin size) histogram counts are normalized by its bin size and the number of initial (DHIT) or incoming (STI) structures: N_c : number of creations, N_d : number of disappearances, N_s : number of splits, $N_{sub,s}$: number of sub-splits of compound events. Shock location at $x/\lambda = 0$.

location in STI than in the corresponding DHIT, which is a consequence of the shock-induced mixing enhancement. This difference between STI and DHIT becomes larger for the higher Mach number $M = 3.0$. Note that figure 3.8 is not distinguishing between split or merge events of different order. While disappearance and split events start to occur approximately at the same downstream location, the downstream range of disappearance events is larger than for split events as structures (which have been continuing for a while) disappear without prior splitting.

For all structure types ($\phi_\lambda, \phi_{2\lambda}, \phi_{4\lambda}$), the number of split and, consequently, disappearance events per initial (DHIT) or incoming (STI) structure is larger in the STI than in the DHIT configuration due to the added structure deformation induced by the shock. This difference between STI and DHIT further increases with the shock Mach number and becomes larger with increasing initial size of the structures. While the peak location is approximately at the same downstream location for ϕ_λ and $\phi_{2\lambda}$, the disappearances and splits peak further upstream (closer to the shock) in the STI than in the DHIT for the $\phi_{4\lambda}$ structures, which indicates again the dependence of the structure deformation on the (initial) size of the structures. For both, DHIT and STI, the number of disappearance and splits per initial and incoming structure, respectively, increased with initial size, which can also be seen in the graphs, figure 3.6 and 3.7. The higher shock Mach number $M = 3.0$ induces more splits in closer proximity to the shock.

The number of compound events is larger in the STI than in the DHIT, increasing with higher Mach number and peaks closely after the shock location. The sub-split events hidden in the compound events explain the reduced number of pure split events in the STI and, also, contribute to the number of disappearing structures closely behind

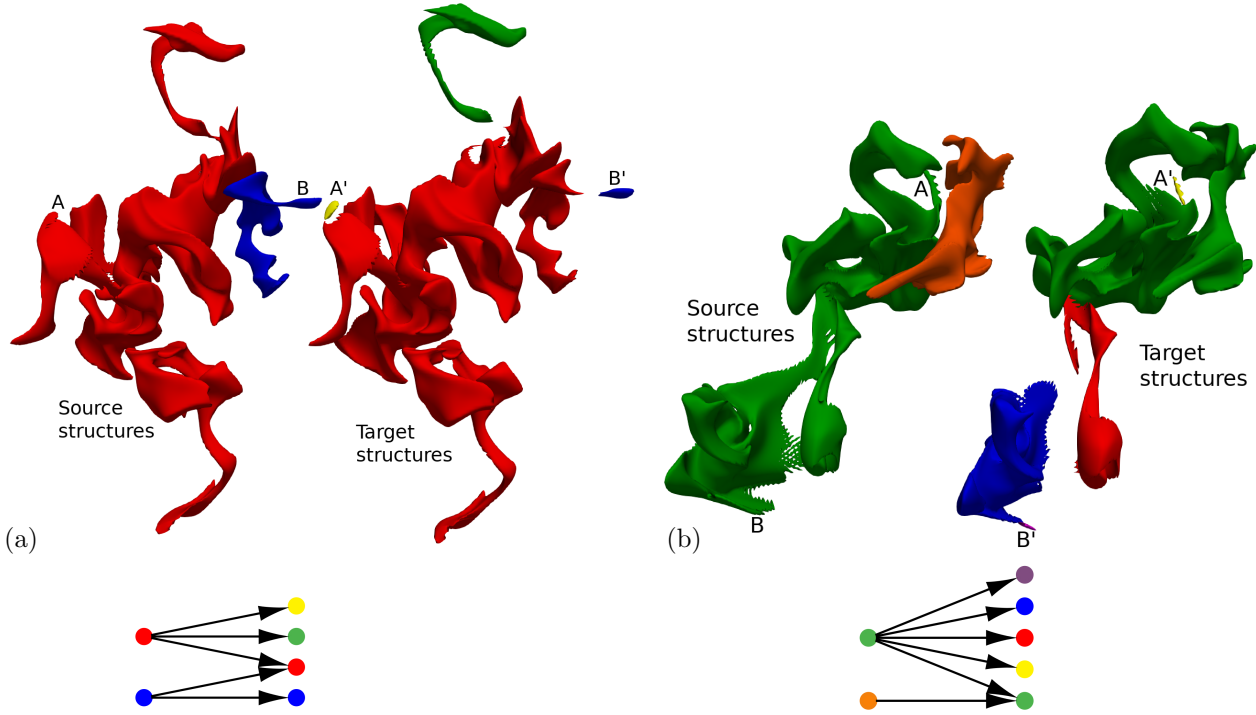


Figure 3.9: Representative compound events of $\phi_{2\lambda}$ structures in the post-shock region of the STI configuration. For each event (a)-(b) target structures are artificially displaced for clarity. The diagram below each rendering represents the structures as nodes (with colors matching those of the associated rendered structures) and the correspondences as arrows, directed from source to target structures. Structure colors indicate branches as illustrated in fig. 2.7.

the shock, especially for the $\phi_{4\lambda}$ structures, as the number of compound events per incoming structure is larger for increasing initial size of the structure, figure 3.8. Note that, despite the higher sampling frequency, which reduces the number of sub-splits in the DHIT used for the STI with $M = 3.0$ compared to the DHIT used for the STI with $M = 1.5$, the number of sub-splits increases for the higher Mach number in STI. In both DHIT and STI flow configurations and for all structure types the number of merge events per initial/incoming structure is small compared to the number of splits. Similar to sub-split events, there is a considerable number of sub-merge events hidden in compound events. While the number of pure split events is clearly larger than the number of compound events, the number of compound events is comparable or even larger than the number of pure merge events, especially in the STI configuration where more compound events, and hence more sub-merge events, occur. While there is typically one sub-merge per compound event, multiple sub-split events of potentially higher order occur per compound event, see figure 3.9 for some representative examples of compound events. Compound events are of importance to capture all cases of structure interactions in the scalar mixing process. Therefore it is essential for the tracking algorithm to account for compound events rather than neglecting them as in previous feature tracking methodologies.

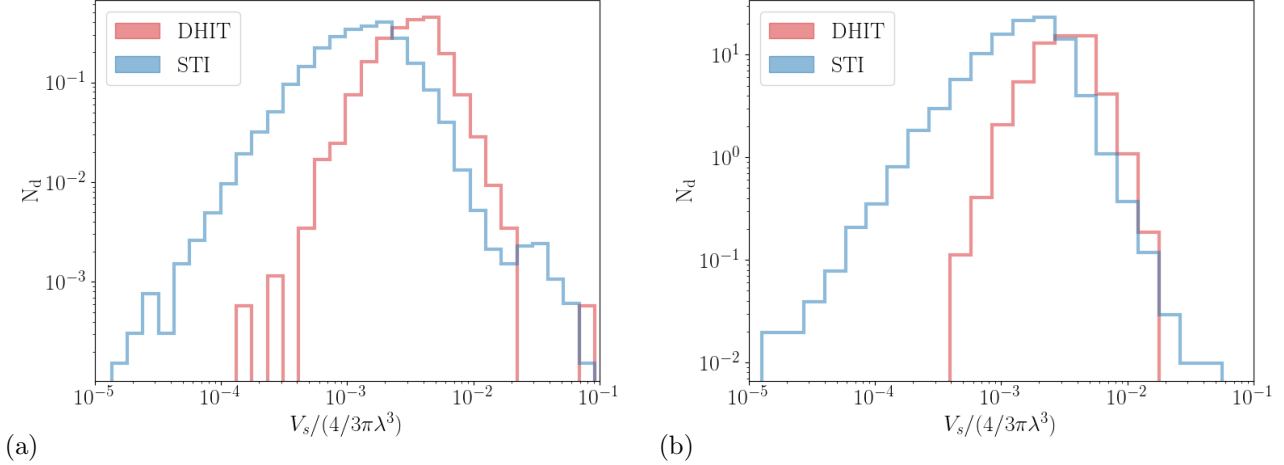


Figure 3.10: Volume histogram of disappearing ϕ_λ (a) and $\phi_{4\lambda}$ (b) structures in DHIT and STI $M = 1.5$ normalized by the number of initial and incoming structures. Volume V_s of source structures relative to a sphere of radius of the initial Taylor-microscale λ .

3.4 Structure volumes of event types

In this section we investigate the volumetric size of source and target structures of different event types. In addition to providing physical insight, this section also serves as a verification of the detected events. The volume histograms of disappearing ϕ_λ and $\phi_{4\lambda}$ structures in STI $M = 1.5$ and corresponding DHIT are shown in 3.10. As already seen in the downstream distribution of disappearance events presented in the previous section, the normalized number of disappearing structures per initial/incoming structures is larger for the $\phi_{4\lambda}$ structures of larger initial size than the ϕ_λ structures and for the same initial size more structures per incoming / initial structure disappear in the STI than in the DHIT, due to the increased number of splits per structure. As expected, the volume of disappearing structures of both structure types and in both flow configuration is small compared to the volume of a sphere of radius equal to the initial Taylor micro-scale in the DHIT simulation. The volume range of disappearing structures is approximately the same for both structure types. While the largest disappearing structure in the DHIT configuration is similar in volume to the largest disappearing structure in STI, the smallest disappearing structure in the DHIT is significantly larger than the smallest disappearing structure in the STI configuration, because of the increased grid resolution of the STI in downstream direction. As both configurations apply the same minimum number of surface points criterion for a structure to be considered in the tracking, a smaller volume can be realized with this minimum number of points in the STI configuration due to the higher resolution.

A similar grid resolution effect can be observed in the joint source and target volume histograms of continuing ϕ_λ and $\phi_{4\lambda}$ structures in STI $M = 3.0$ and DHIT, shown in figure 3.11. A deviation from volume-preserving continuation events is observed on small scales due to scalar dissipation and for larger shock-crossing structures due to the compression of the structure and the scalar dissipation amplification by the shock.

The joint histograms of splitting ϕ_λ and $\phi_{4\lambda}$ structures in STI and DHIT for both considered Mach numbers are formed by the volume of the source structure and the sum of the volume of the target structures of the split events,

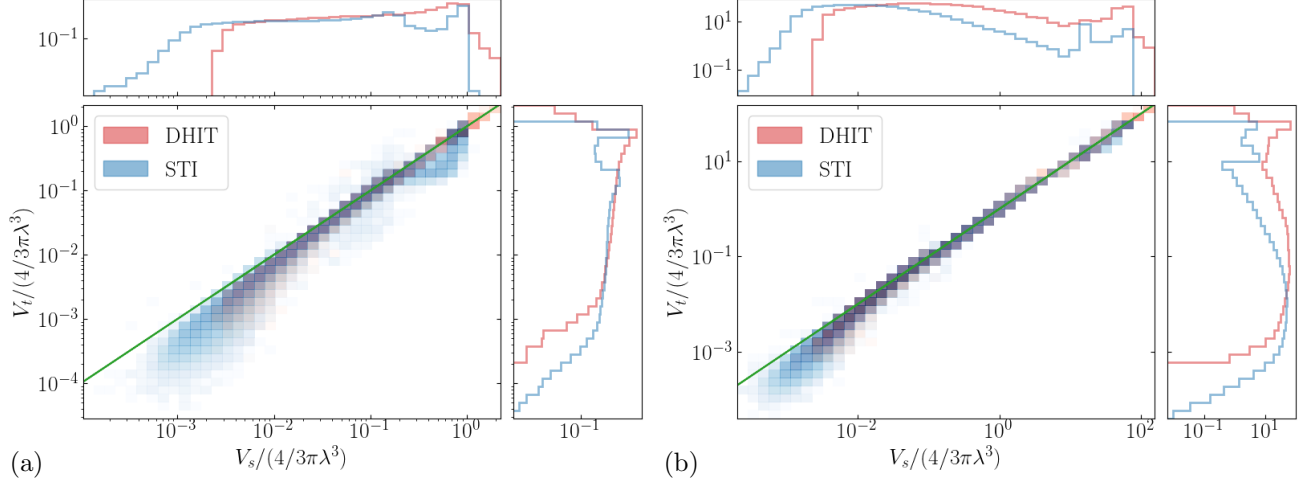


Figure 3.11: Joint histogram of volumes V_s of source and V_t targets of continuation events in the evolution of ϕ_λ (left) and $\phi_{4\lambda}$ (right) structures in DHIT and STI for $M = 3.0$ normalized by the number of initial and incoming structures. Volumes of the source and target structures are normalized by the volume of sphere of radius of the initial Taylor-microscale. Green reference line indicates volume preserving splits. Marginal histograms included as projections.

figure 3.12, where the grid resolution effect can be observed again. In accordance with the downstream distribution of split events, figure 3.8, the number of splitting structures per initial/incoming structure is larger for the initially bigger sphere $\phi_{4\lambda}$. Note that the splitting of a structure can occur after the structure participated in a merge event, which allows the ratio of the splitting volume to the reference volume, $V_s/(4/3)\pi\lambda^3$, to be larger than the ratio of the volume of the initialized structures to this reference volume. While the largest splitting volumes are naturally found for the $\phi_{4\lambda}$ structures, the smallest splitting structures have similar volumes for both scalar fields as the dissipation affects both fields on the same scales.

Similar to the disappearing volumes, the largest splitting volume in the DHIT and STI are comparable, while the smallest splitting structure have less volume in the STI than in the DHIT due to the increased grid resolution. For both, DHIT and STI, the smallest splitting volume are significantly larger than the smallest disappearing volumes. Hence, structures that have shrunk to a certain volume due to scalar dissipation or previous splits, do not split further and simply continue until they reach the end of their lifetime and disappear by dissipation. In agreement, structures of small volumes, near the end of their lifetime, are characterized by a relatively high compactness, $\hat{\lambda}$, which decreases the probability of a split (see §3.8.1). Figure 3.12 also indicates that the sum of the volume of the target structures of a split approximately conserves the volume of large source structures, while the sum of the target volumes is reduced compared to the source volume for small source volumes due to increased scalar dissipation on smaller scales.

Figure 3.13 expresses an increased number of compound events in the STI compared to the DHIT for both structure types, ϕ_λ and $\phi_{4\lambda}$. The number of occurring compound events per initial/incoming structure increases from ϕ_λ to $\phi_{4\lambda}$ structures for both DHIT and STI. Compared to pure split events, the smallest structures participating in compound events are larger as, by definition, a compound event has at least one sub-merge event, and merge events

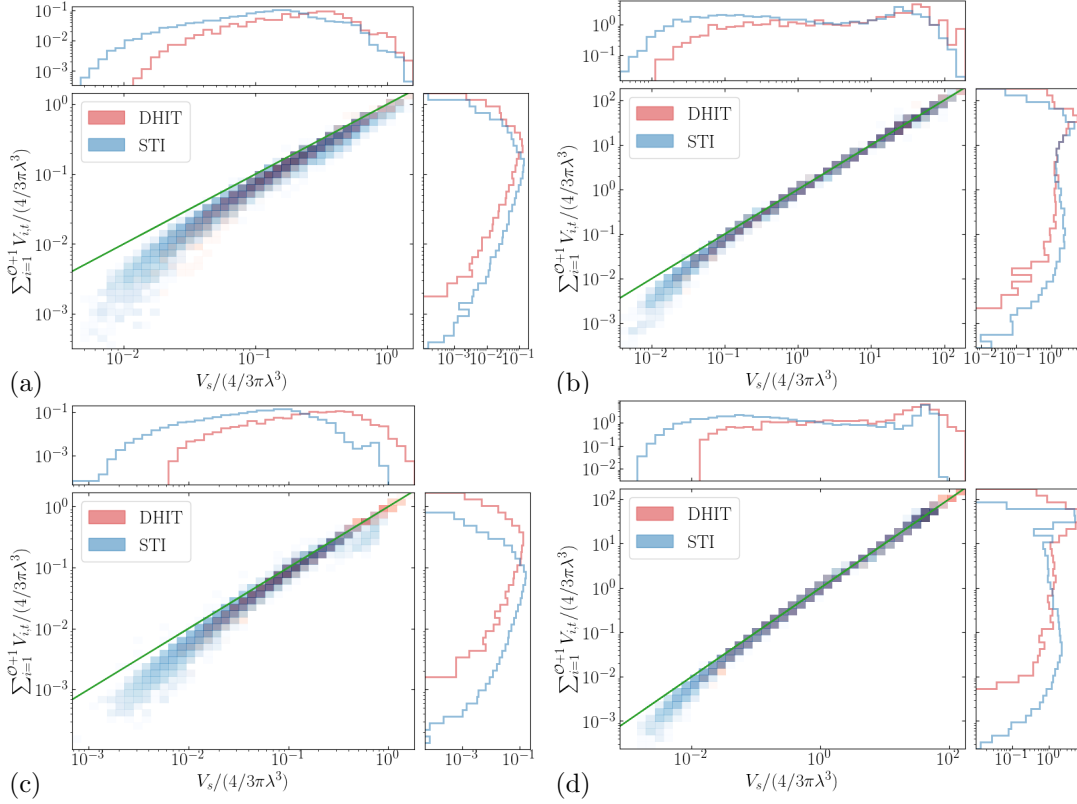


Figure 3.12: Joint histogram of volumes V_s of source and V_t of summed targets of split events in the evolution of ϕ_λ (left) and $\phi_{4\lambda}$ (right) structures in DHIT and STI for $M = 1.5$ (top) and $M = 3.0$ (bottom) normalized by the number of initial and incoming structures. Volumes of the source and target structures are normalized by the volume of sphere of radius of the initial Taylor-microscale. Green reference line indicates volume preserving splits. Marginal histograms included as projections.

typically occur between structures of relatively large volume (as can be inferred from the volume histogram for merge events, not shown). Similar to split events, for $\phi_{4\lambda}$ structures the portion of compound events for which the volume is approximately conserved is higher than for the smaller ϕ_λ structures as the deviation occurs for smaller volumes.

3.5 Split and merge events in geometrical feature space

The geometrical signature $\{\hat{S}, \hat{C}, \hat{\lambda}\}$ of source and target structures of split and merge events are investigated in this section, focusing on the effects of the initial structure size and the shock. The joint histograms $(\hat{\lambda}, \hat{C})$ and (\hat{S}, \hat{C}) of the geometrical attributes of source and target structures of split events in DHIT (figure 3.14 (a)) and STI (figure 3.14 (b)), indicate that splitting source structures are characterized by lower values of the compactness, $\hat{\lambda}$, and the absolute shape index than their targets, while the dimensionless curvedness \hat{C} is comparable between sources and targets. Structures characterized by low compactness are more stretched and likely to contain a ligament portion, which increases the probability of these structures becoming the source of a split.

Figure 3.15 suggests that once the highly stretched ligament section of a source breaks, the produced target structures exhibit a dome-like region in the region previously connected to the former ligament section, thus increasing

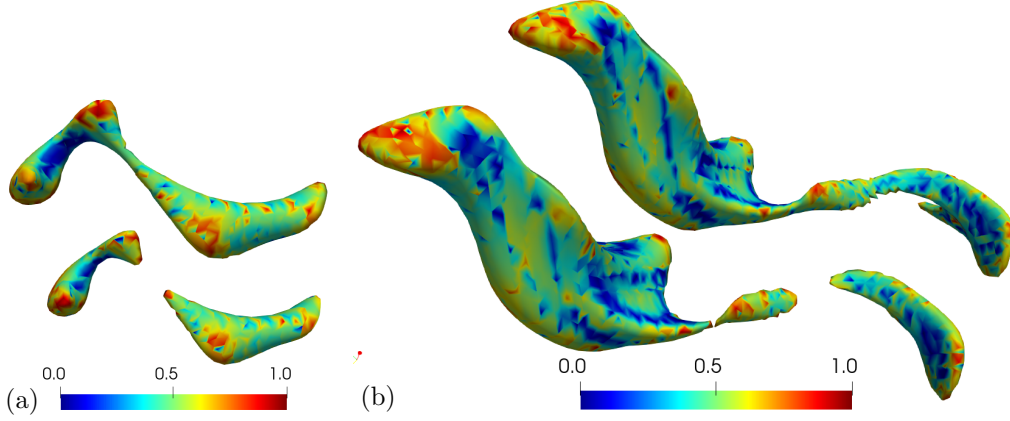


Figure 3.15: Absolute shape index mapped on $\phi_{2\lambda}$ structures in DHIT during split events of order one (a) and order two (b). Source structures artificially displaced with source on the top and target on the bottom.

their absolute shape index. The compactness $\hat{\lambda}$ of splitting sources in DHIT (figure 3.14 (a)) decreases with larger initial size of the structures from ϕ_λ to $\phi_{4\lambda}$, as the structures become more convoluted and stretched before they split. The higher convolution of the source structures for bigger initial size also shows in a higher dimensionless curvedness \hat{C} and an increased amount of hyperbolic surface points, which lowers the absolute shape index ($\hat{S} < 0.5$). In contrast, splitting ϕ_λ structures tend to be dominated by elliptical points ($\hat{S} > 0.5$). This dependence of the geometric signature of splitting sources on the initial size is linked to geometrical trajectories of the structures, which have a common starting point in the geometrical feature space as all of them are spheres but develop very differently depending on the initial structure size (see §3.8.1). Furthermore, for $\phi_{2\lambda}$ and $\phi_{4\lambda}$ structures, the \hat{S} - \hat{C} joint PDF of source structures is essentially enclosed by that of the target structures. As the geometrical trajectories indicate, split events early in the structure lifetime typically contain a dominant target structure which is similar in shape to the splitting source structure. Hence, this target is located in close proximity to the splitting source in the geometrical feature space. The compression of the passive scalar structures by the shock wave leads to lower values of compactness, $\hat{\lambda}$, and absolute shape index, \hat{S} , of splitting source structure in the STI, figure 3.14 (b), compared to the DHIT. This effect is most pronounced for $\phi_{4\lambda}$ structures, which causes the increase of in the number of split events from the DHIT to the STI to be the highest for the $\phi_{4\lambda}$ structures, as already observed in §3.3.

The joint histograms $(\hat{\lambda}, \hat{C})$ and (\hat{S}, \hat{C}) of the geometrical attributes of source and target structures of merge events in DHIT and STI presented in figures 3.16(a) and (b), respectively, show opposite relations between source and target geometrical signature than previously observed for split events. Here, the merged target structure is characterized by a lower compactness. The difference between merging sources and merged targets in terms of the absolute shape index is less pronounced than previously observed for split events. The dependencies of the geometric signature on the initial structure size previously observed for splitting source structures apply to merged target structures.

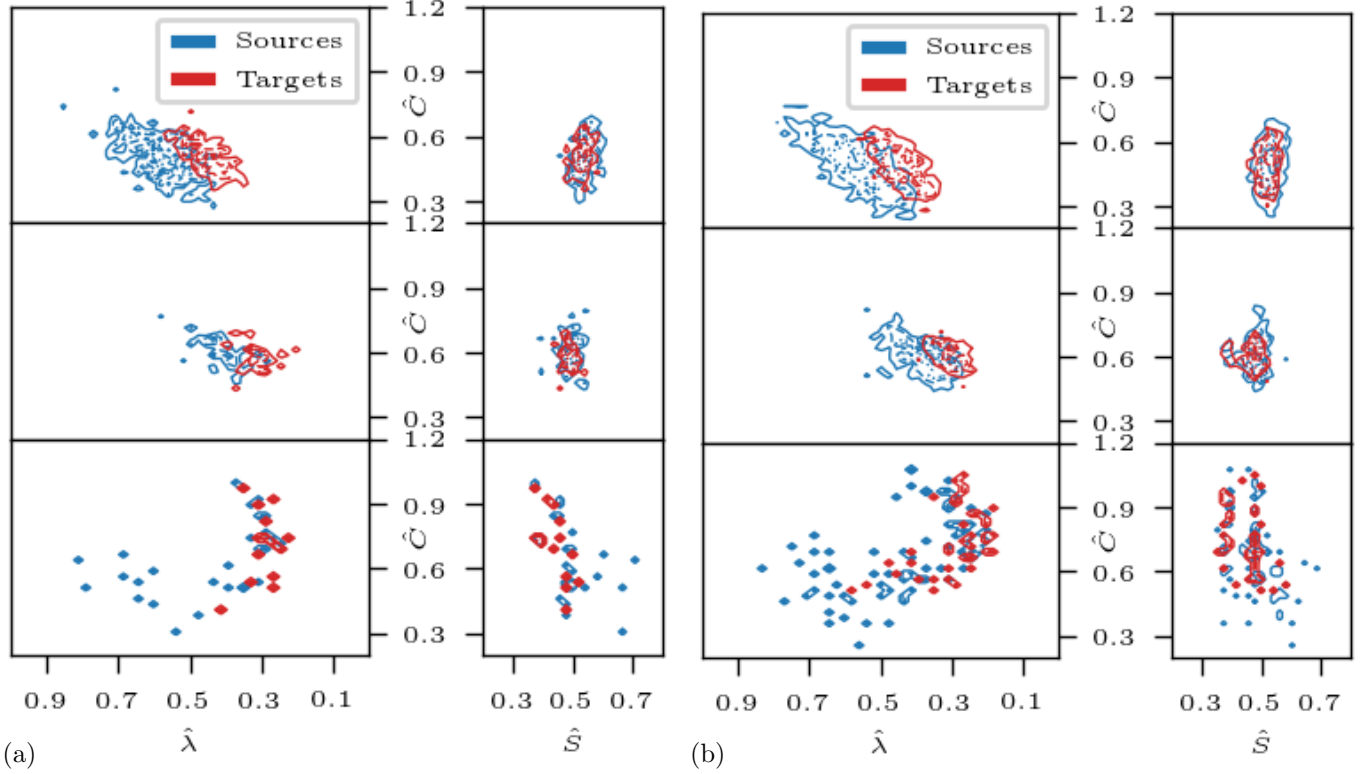


Figure 3.16: Joint histograms of geometrical attributes of source and target structures of merge events in DHIT (a) $M = 1.5$ STI (b). Top: ϕ_λ , Center: $\phi_{2\lambda}$, Bottom: $\phi_{4\lambda}$. Contour line: —, —, ···, ···, ···, ···, ···, ···: 0.2, —, —, ···, ···, ···, ···, ···: 0.4, ···, ···, ···, ···, ···, ···, ···: 0.6, ···, ···, ···, ···, ···, ···, ···: 0.8.

3.6 Downstream evolution surface-averaged physical quantities

Ensemble statistics of the downstream evolution of surface-averaged physical quantities of the flow field computed as

$$q_{\text{avg}} = \frac{1}{A} \int_A q dA \quad (3.4)$$

on individually tracked isosurfaces of ϕ_λ and $\phi_{4\lambda}$ passive scalar in STI and corresponding DHIT configurations are investigated in this section. The effect of the shock on these surface-averaged quantities is more pronounced for a higher Mach number $M = 3.0$ in the STI configuration. The downstream evolution of the ensemble mean and standard deviation of the magnitude of the scalar gradient $|\nabla\phi|$ normalized by its initial mean value in the DHIT is presented in figure 3.17. In both flow configurations, the influence of the size of the transitional length of the initialized structures (see figure 3.3) on the early development of the scalar gradient magnitude is apparent. While the scalar gradient magnitude of ϕ_λ structures immediately decays after the initialization, stretching of the non-zero scalar gradient region of the $\phi_{4\lambda}$ structures cause an initial increase of the scalar gradient magnitude. After the initial increase the magnitude of scalar gradient monotonically decays in the DHIT configuration as the scalar mixing progresses. In contrast, the scalar gradient magnitude is amplified while the structures pass through the shock wave in the STI configurations. This amplification is larger for a larger initial size of the structures which indicates the scalar dissipation enhancement is larger for those structures. After the scalar gradient magnitude has reached its

maximum, its decay is more pronounced than in the corresponding DHIT as the scalar dissipation rate is amplified by the shock (Gao *et al.*, 2020; Boukharfane *et al.*, 2018).

Contrary to the downstream evolution of the scalar gradient magnitude, the vorticity magnitude does not exhibit a transient development as the turbulence is already fully developed at the initialization of the structures. The downstream evolution of the ensemble statistics of the surface-averaged vorticity magnitude shown in figure 3.17 is normalized by the initial mean value in the DHIT. As the turbulence decays in the DHIT, the surface-averaged vorticity magnitude monotonically decreases over time. In contrast, the vorticity magnitude is amplified as the structures pass through the shock wave. Unlike the scalar gradient magnitude, the increase of the vorticity magnitude is similar for all initial structure sizes. Similar to the evolution of the scalar gradient magnitude, the vorticity magnitude decays more rapidly in the post-shock region than in the DHIT.

The scalar gradient and the vorticity show no preferential alignment at initialization since the turbulence is homogeneous and isotropic. After initialization a transient develops in which the alignment decreases, figure 3.17.A strong anti-correlation between the scalar gradient and the vorticity was found by Kerr (1985), interpreted as the scalar being wound around vortex tubes. In the present STI configuration, the alignment is further decreased when the structures pass through the shock wave. The minimal alignment during transient becomes smaller with increasing initial structure size. Again, the downstream evolution in the STI is shifted against the DHIT evolution, especially in the initial rapid decay of the alignment. The alignment is increasing for all initial structure size in the post-shock region. While the alignment in the STI has a lower minimum than in the DHIT and stays below the alignment in the DHIT for ϕ_λ structures, the minimum is bigger in the STI than in the DHIT for $\phi_{4\lambda}$ and the alignment reaches similar values in both configurations further downstream in the post-shock region.

The amplification of the passive scalar dissipation rate across the shock is associated with evolution of the alignment between the eigenvectors of the strain-rate tensor and the scalar gradient (Kerr, 1985; Ashurst *et al.*, 1987; Danish *et al.*, 2016; Parashar *et al.*, 2017). Alignment analysis of these vectors has been utilized in the development of SGS models for the flux of a passive scalar (Pullin, 2000). Figure 3.18 show the downstream evolution of the ensemble mean and standard deviation of the surface-averaged alignments between $\nabla\phi$ and the strain-rate eigenvectors, denoted as α, β and γ with their respective eigenvalues ordered from most extensional to most compressive, $\alpha \geq \beta \geq \gamma$.

Due to the ad-hoc initialization of the passive scalar structures and the isotropic, homogeneous background turbulence, the upstream starting point in the DHIT corresponds to no preferential alignment (all cosines of those angles equal approximately 0.5). After the initialization, a transient development is observed for all alignments on all considered isosurfaces, which lasts longer for larger initial size of the structures. According to Batchelor’s theory, the scalar gradient is most aligned with the most compressive eigendirection of the strain-rate tensor Batchelor (1959); Vedula *et al.* (2001); Donzis *et al.* (2010). In agreement, the surface-averaged mean alignment with γ becomes the largest, followed by α and β . These results, obtained for ensembles of initially spherical structures from surface-averaged data, are consistent with previous volumetric analyses of HIT (Ashurst *et al.*, 1987; Danish *et al.*, 2016) and STI (Boukharfane *et al.*, 2018; Gao *et al.*, 2020). In addition, the scalar gradient alignment with the eigenvectors of

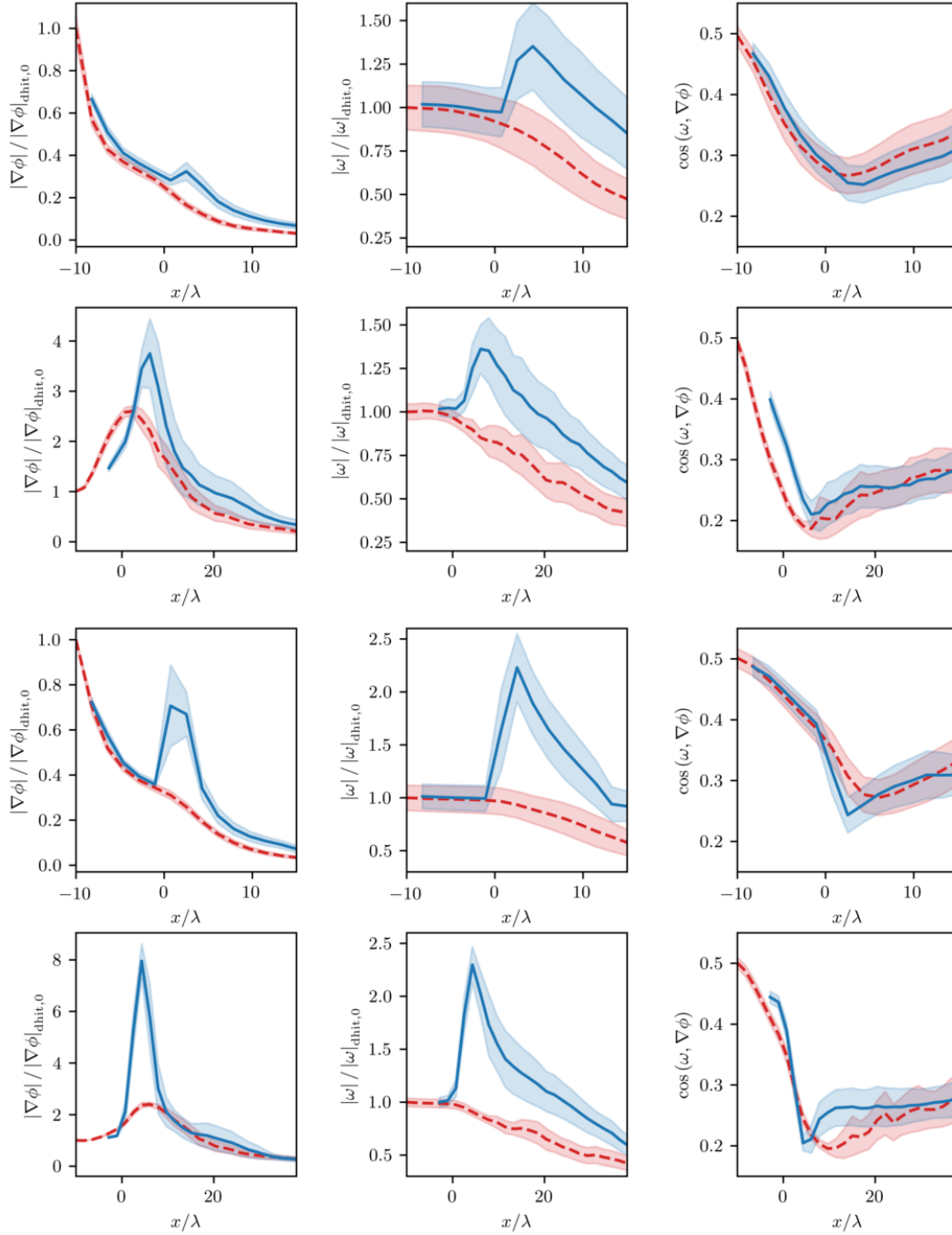


Figure 3.17: Ensemble mean of the downstream evolution of the scalar gradient magnitude (left), vorticity magnitude (center) and the alignment between the scalar gradient and vorticity averaged on ϕ_λ (row 1 for $M = 1.5$ and 3 $M = 3.0$) and $\phi_{4\lambda}$ (row 2 $M = 1.5$ and 4 $M = 3.0$) isosurfaces. Color filled region indicate one standard deviation. Structure location defined by its most downstream surface point.

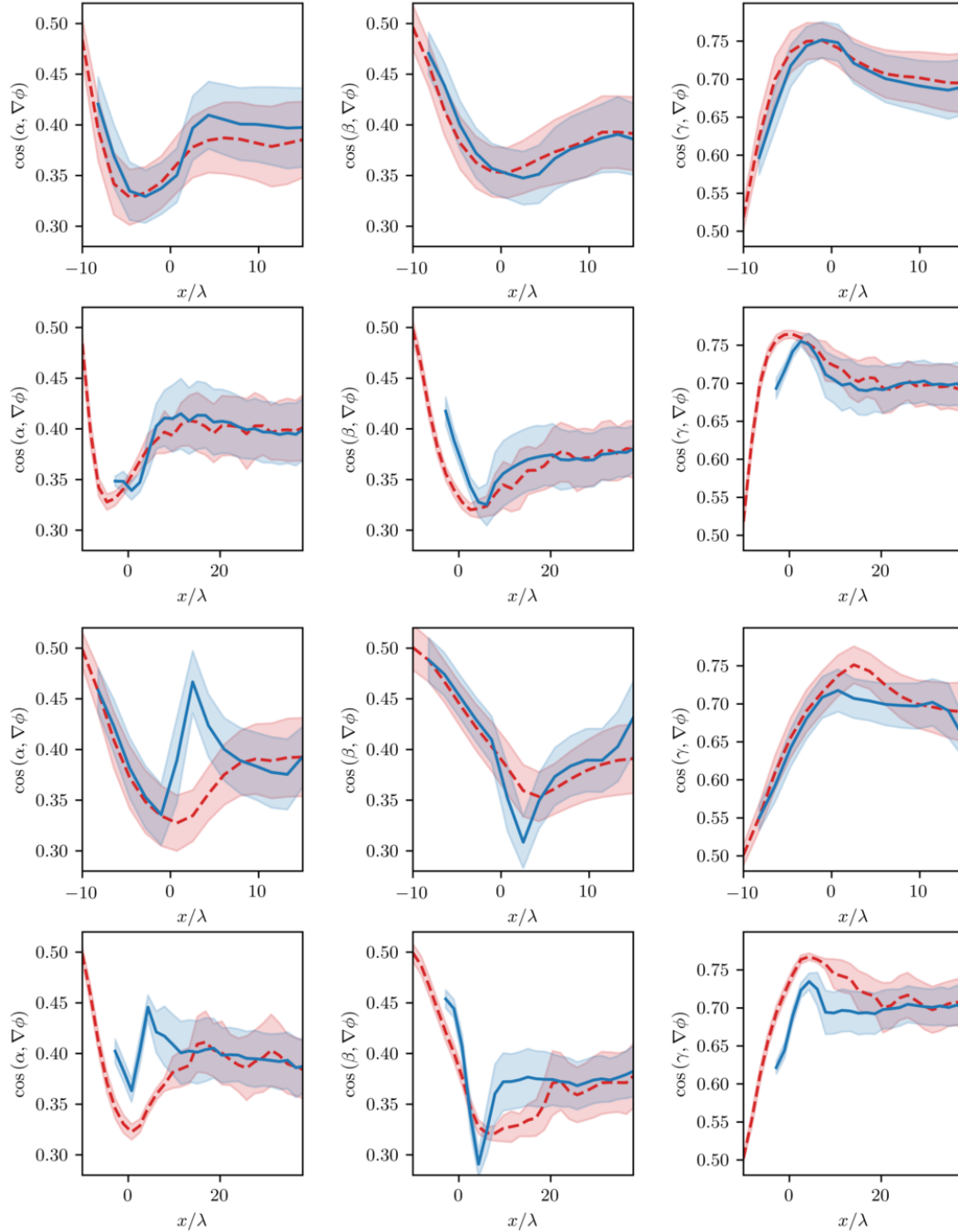


Figure 3.18: Ensemble mean of the downstream evolution of the scalar gradient alignment with the α (left), β (center) and γ (right) eigenvector of the strain-rate tensor averaged on ϕ_λ (row 1 for $M = 1.5$ and $3 M = 3.0$) and $\phi_{4\lambda}$ (row 2 $M = 1.5$ and $4 M = 3.0$) isosurfaces. Color filled region indicate one standard deviation. Structure location defined by its most downstream surface point.

the solenoidal component of the strain-rate tensor shows very similar behavior to that of incompressible turbulence (John *et al.*, 2019).

After the transient, the alignment with α becomes nearly constant, whereas alignment with γ (β) slightly decreases (increases) in a similar way for all considered initial structure sizes. The alignments in the STI exhibit a similar transient development than in the DHIT with a temporal (downstream) shift which becomes more significant with increasing initial structure size. While the transient development affects the entire structures immediately after their initialization in the DHIT, only the part of an incoming structure in the STI configuration which is already inside of the domain experiences the transient development, while the part of the structure outside of the domain remains in the initial conditions of the inflow database. Once the incoming structures become closed surfaces, the transient development is more advanced on the further downstream region of the surface. Hence, the surface-averaged transient development is shifted compared to the DHIT structures and the shift is more significant for larger incoming structures, as they become closed surfaces at a more downstream location. A change of slopes is observed as the structures traverse the shock, which is more pronounced for the small ϕ_λ structures due to the shorter streamwise distance of interaction and for the higher Mach number $M = 3.0$. In particular, the scalar gradient alignment with the most extensional eigenvector α increases across the shock. For reactive scalars in premixed combustion, the α alignment is also known to increase due to dilatational motion induced by heat release in the reaction zone Swaminathan & Grout (2006); Kim & Pitsch (2007). The alignment with the intermediate and most compressive eigenvector decreases due to enhanced scalar dissipation immediately after the shock. The decrease in alignment with the intermediate eigenvector β and the vorticity ω is thus expected since the alignment between β and ω is strengthened by the shock-interaction (Livescu & Ryu, 2016; Boukharfane *et al.*, 2018). A weakening of alignment between the scalar gradient and the most compressive strain-rate eigenvector γ in flow regions of high dilatation was recently reported by (Panickacheril John *et al.*, 2020). In the post-shock region, the alignments in the STI reach similar values than in the DHIT. Especially for the $\phi_{2\lambda}$ and $\phi_{4\lambda}$ structures the variance of the ensemble statistics increases over time in the DHIT and STI configuration.

3.7 Correlations between local surface geometry and physical quantities on the surface

The local geometry of a surface can be characterized by the pointwise values of dimensionless curvedness C and the absolute shape index S on the surface (see figure 3.19a,b). Additionally, pointwise physical quantities can be mapped on the surface, such as the alignment between the scalar gradient and the most compressive eigenvector γ of the strain-rate tensor (figure 3.19c). To elucidate correlations of local physical quantities on a passive scalar isosurface with its local geometry, the average alignment of scalar gradient and γ conditioned on S and C accounting for the area-based joint PDF is obtained (figure 3.19e). For the sample structure shown in figure 3.19, a clear correlation between flat regions (low C) and high alignment of $\nabla\phi$ and γ is observed on the surface renderings (a,c) and the conditioned joint PDF (e).

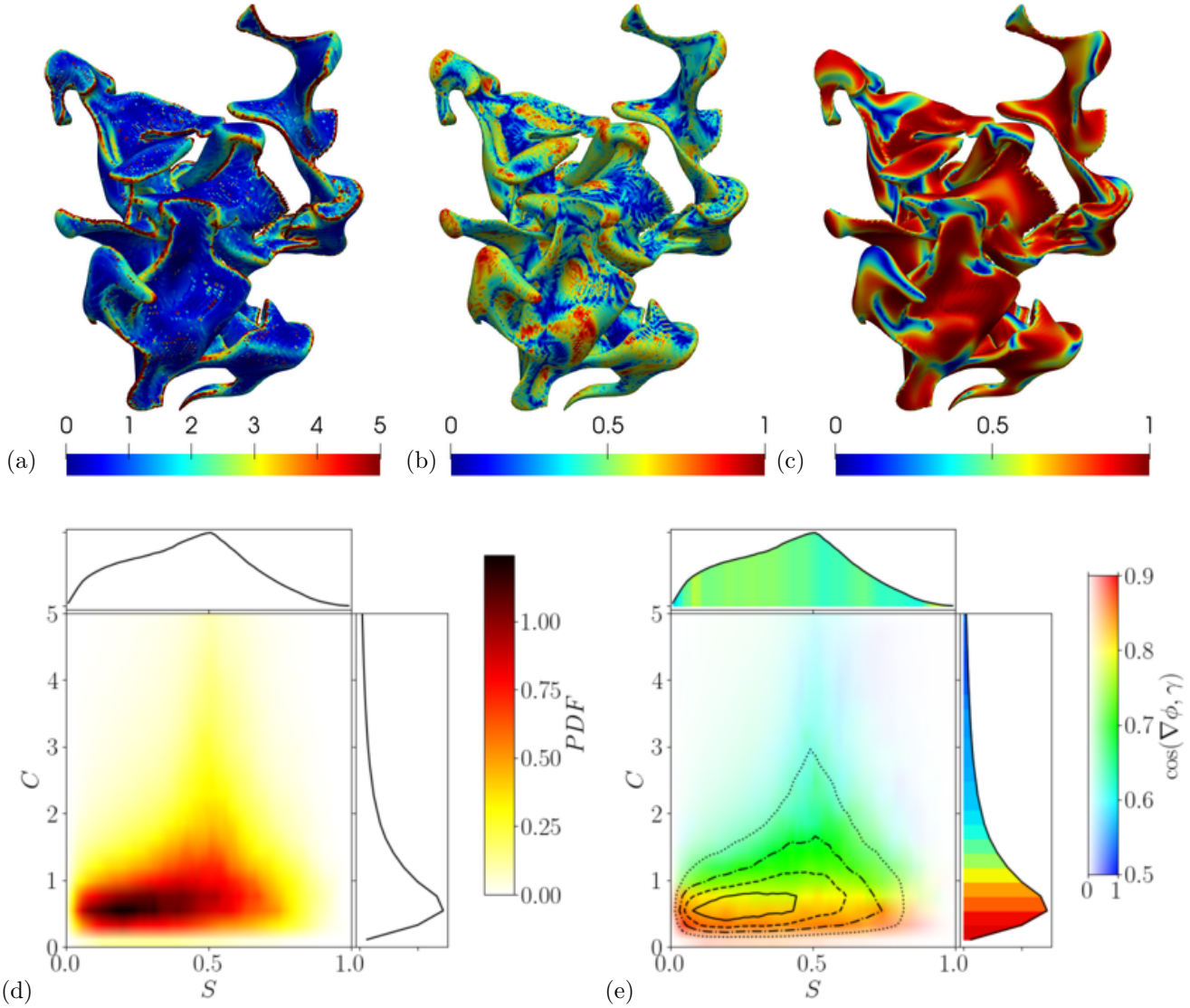


Figure 3.19: Local values of the dimensionless curvedness (a), absolute shape index (b) and the alignment between the most compressive eigenvector γ of the strain-rate tensor and the scalar gradient $\nabla\phi_{4\lambda}$ (c) mapped on a isosurface of the $\phi_{4\lambda}$ at $t/\tau_0 = 4.3$. (d) Area-based joint and marginal probability density functions of S and C . (e) Average alignment of scalar gradient and γ conditioned on S and C (hue) accounting for the area-based joint PDF (saturation, from white for zero area coverage to the full color for the maximum area coverage), with area-based marginal PDFs of S and C colored by the average alignment conditioned on each variable. Equispaced contour lines for 0.2 (dotted), 0.4 (dash-dotted), 0.6 (dashed), and 0.8 (solid) of the peak joint PDF value.

Figure 3.20 shows ensemble mean and standard deviation of the surface distributions, conditioned on S and C , of alignments between $\nabla\phi$ and strain-rate eigenvectors and the vorticity, and the scalar gradient magnitude obtained on $\phi_{4\lambda}$ isosurfaces, for collections of isosurfaces centered around four different streamwise intervals at any given time of the simulation. Positive correlations with $C < 5.0$ are observed for the alignment of $\nabla\phi$ (which is parallel to the normal to the isosurface at every point, \mathbf{n}) with $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, indicating that regions of higher curvature present better alignment between the scalar gradient and the most extensional and intermediate strain-rate eigenvectors. The opposite trend is observed for the alignment with the most compressive strain-rate eigenvector, $\boldsymbol{\gamma}$, indicating that flatter regions of the isosurfaces present larger alignment between $\nabla\phi$ (and thus \mathbf{n}) and $\boldsymbol{\gamma}$. For $C < 5.0$ these correlations are strengthened as the isosurfaces are transported downstream. For regions of very large curvedness ($C > 5.0$) the expected alignment with $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ slightly decreases, whereas the opposite trend is observed for the alignment with $\boldsymbol{\gamma}$. Across the shock, the scalar gradient alignment with $\boldsymbol{\alpha}$ ($\boldsymbol{\gamma}$) increases (decreases) for the entire range of the absolute shape index, S . A decreased alignment with $\boldsymbol{\beta}$ downstream of the shock is more pronounced at parabolic points ($S = 0.5$) of the isosurfaces.

For the scalar gradient alignment with the most extensive strain-rate eigendirection $\boldsymbol{\alpha}$, the correlation with S are less pronounced than with C : at the further downstream locations there is a slightly positive correlation with the shape index in its hyperbolic range ($S < 0.5$). The same correlation is found for the alignment with the intermediate eigendirection $\boldsymbol{\beta}$ farther downstream. The ordering of the ensemble means by the three considered locations is according to the downstream evolution of their surface-averaged values, §3.6. Contrary to the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ alignment with $\nabla\phi$, the alignment with the most compressive eigendirection $\boldsymbol{\gamma}$ shows a negative correlation with the absolute shape index for hyperbolic points. In the elliptical range of S , however, the correlation is positive, but becomes less pronounced further downstream.

The alignment between $\nabla\phi$ and ω shows a positive correlation when conditioned on C and with S in its hyperbolic range. Hence, the anti-correlation between $\nabla\phi$ and ω is more pronounced on flatter surface regions. The correlation between the magnitude of $\nabla\phi$ and the local geometry exhibits similar trend to found for the alignment between $\nabla\phi$ and $\boldsymbol{\gamma}$. This implies that most of the dissipation after the transient and its enhancement while crossing the shock occurs on flatter regions (lower \hat{C}). Again, the ordering of the ensemble means by location is in agreement with the downstream evolution of the surface-averaged scalar gradient magnitude described in 3.6. Similar correlations are found in the DHIT and STI $M = 3.0$ configurations and for all initial structure sizes (not shown).

The correlation between the Q -criterion on the passive scalar isosurface and the surface geometry shows similarities to the findings for the vorticity magnitude (not shown). At the most upstream location Q is positively correlated to C over its entire range. At locations further downstream this correlation becomes more pronounced for $C < 2.5$. A slightly negative correlation is established for highly curved regions, but the expected value of the Q -criterion is still significantly higher than for flat regions. When conditioned on S Q shows a positive correlation in the hyperbolic range of S and for the more downstream locations peaks at the parabolic value of $S = 0.5$, which is also observed for the vorticity magnitude. Note that the ordering of the ensemble means conditioned on S by downstream location is opposite for the Q -criterion than for the vorticity magnitude.

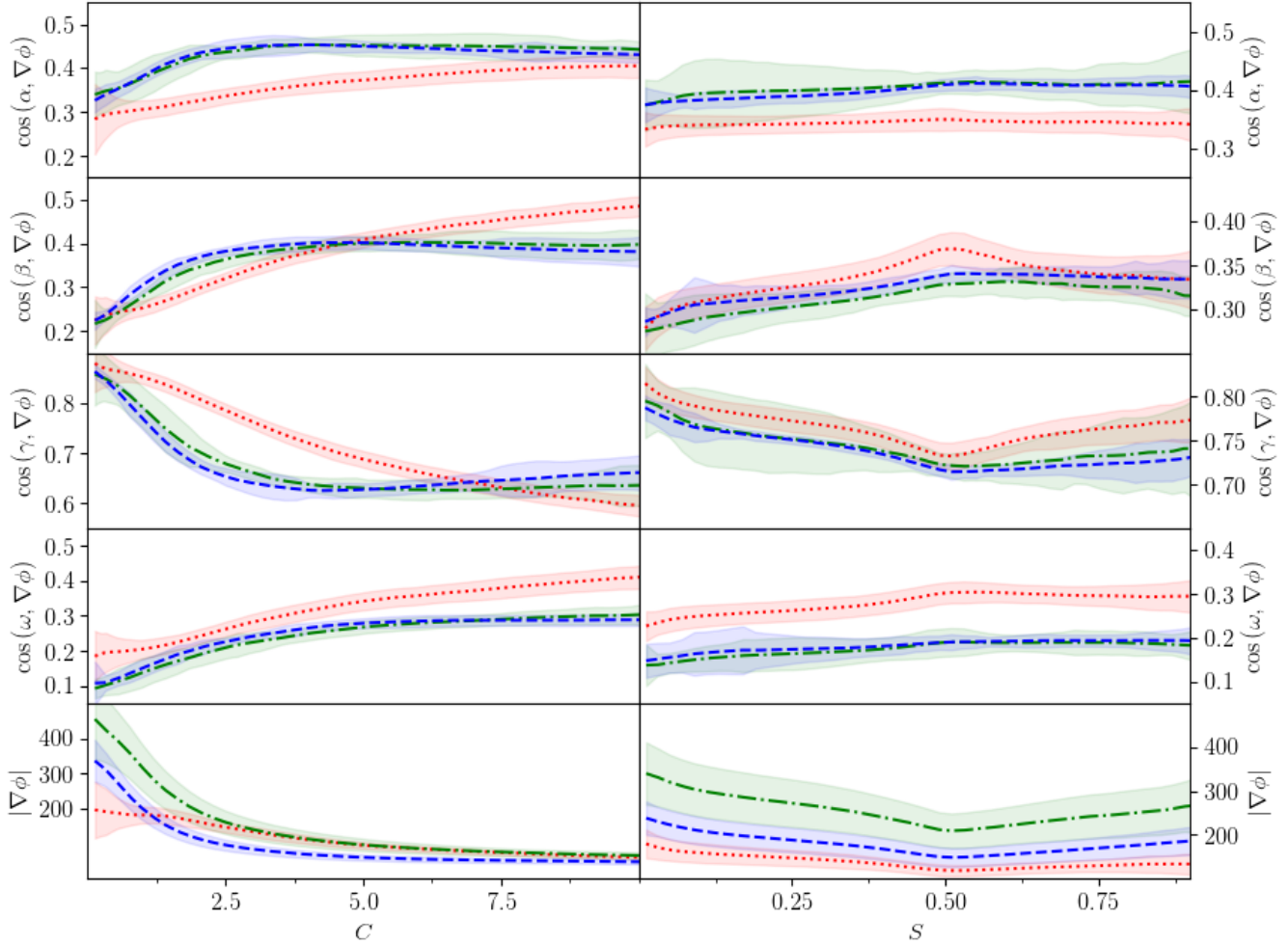


Figure 3.20: Surface distributions of the alignment between the scalar gradient and the eigenvectors α, β, γ of the strain-rate tensor and the vorticity ω and the magnitude of the scalar gradient conditioned on local dimensionless curvedness (left) and absolute shape index (right) of $\phi_{4\lambda}$ in STI $M = 1.5$ at different downstream locations: $k_0x \in [-2.2, -1.8]$ (\cdots), $k_0x \in [4.0, 4.4]$ ($- \cdot -$), $k_0x \in [8.0, 8.4]$ ($- -$). Lines represent ensemble mean of the conditional pdf, while colored region enclose two standard deviations.

3.8 Trajectories of primary structures in the feature space

3.8.1 Geometrical trajectories and surface area evolution

The deformation of passive scalar structures during their temporal evolution can be quantified by investigating the structure's trajectory (or path) in the geometrical feature space $\{\hat{S}, \hat{C}, \hat{\lambda}\}$. Here, we focus on the temporal evolution of primary structures. Hence, from the graph representing the evolution of all tracked structures only branches whose start vertex results from a creation event and end vertex is a disappearing structure are considered. In the DHIT configuration only the initialized structures and in the STI only the incoming structures are candidates for primary structures, but depending on the structure interaction only a subset of them are truly primary structures. At initialization, all $\phi_{4\lambda}$ passive scalar structures in the DHIT are spheres and thus their trajectories have a common

starting point in the feature space at $\{\hat{S}, \hat{C}, \hat{\lambda}\} = (1, 1, 1)$ (see figure 3.21). As visualized in figure 3.2 (a)-(c), the isosurfaces become highly convoluted shortly after the initialization, causing a rapid increase in the dimensionless curvedness \hat{C} and a decrease in the absolute shape index \hat{S} , due to the occurrence of hyperbolic surface points where $S < 0.5$. The deviation from the spherical shape also lowers the compactness, $\hat{\lambda}$, of the structures, as they are stretched. After reaching the maximum dimensionless curvedness, all structures follow a nearly linear trajectory in the \hat{C} - $\hat{\lambda}$ plane towards the (0.7, 0.25) location, thus lowering both their dimensionless curvedness and compactness. The absolute shape index, \hat{S} , follows a rapid initial transition from 1.0 (spherical shape) to 0.5, reached as the maximum \hat{C} is attained. The subsequent decrease of \hat{C} and $\hat{\lambda}$ has a lesser effect on \hat{S} , indicating a balance between the area coverage of hyperbolic and elliptic geometries on the convoluted surfaces.

From the start of the trajectories to the occurrence of the first non-continuation event the change in the geometric signature from one frame to the next (given by the distance between consecutive trajectory points corresponding to tracking steps) monotonically decreases due to the viscous decay of the underlying turbulent background field. As the stretching increases ($\hat{\lambda} < 0.5$) first split events occur in the structure evolution. The highly stretched structures tend to split over a sequence of frames. However, despite this cascade of split events, the change in geometrical signature of the structure is comparable to that of a continuation event, implying that these early splits do not alter the geometry significantly. As visualized in figure 3.22(a), the split events that occur in this early stage of the structure evolution tend to be dominated by one target structure, similar in shape to the splitting source structure, and other smaller targets. As the dominant target continues the branch of the primary structure, its geometrical signature is moderately changed by the split event. This geometrical similarity between the source and the dominant target structure is also observed in the joint histograms of geometrical attributes (see figure 3.14 (a)). After these cascades of split events, the structures have reached their minimum compactness (maximum stretching), and the jump in length between consecutive tracking steps in the trajectory in the geometric feature space is increased as subsequent split events tend to affect the geometrical signature more significantly (figure 3.22 (b)). The later generations of splits do not form a cascade, even though the branch section between splits remain short in some cases. The structure evolution in between splits is again characterized by a decreasing dimensionless curvedness and compactness. The later generations of splits affect the geometric signature of the primary structure in accordance with the findings in the joint histograms of split events in the feature space (see §3.5): while the dimensionless curvedness remains relatively constant, the compactness and by trend the absolute shape index increase during the split event. The final stage of the lifetime of primary structures is characterized by continuation events with increasing area coverage of elliptical surface points and increasing compactness, which makes a further splitting of the structure less likely. This trend is in accordance with the downstream distribution of split compared to disappearance events, discussed in §3.3.

From the collection of individual geometrical trajectories, we calculate ensemble statistics resulting in the mean trajectory and its standard deviation (at each tracking step) for structures of each scalar field ($\phi_\lambda, \phi_{2\lambda}, \phi_{4\lambda}$) in DHIT and STI ($M = 1.5, 3.0$) configurations. These ensemble trajectories, shown in figure 3.23, enable a characterization of the effects of the initial size and the shock compression on the geometric evolution of the structures. Whereas all structures start from a spherical shape ($\hat{\lambda}, \hat{S}, \hat{C}) = (1, 1, 1)$ in DHIT, the initial increase in curvedness is significantly

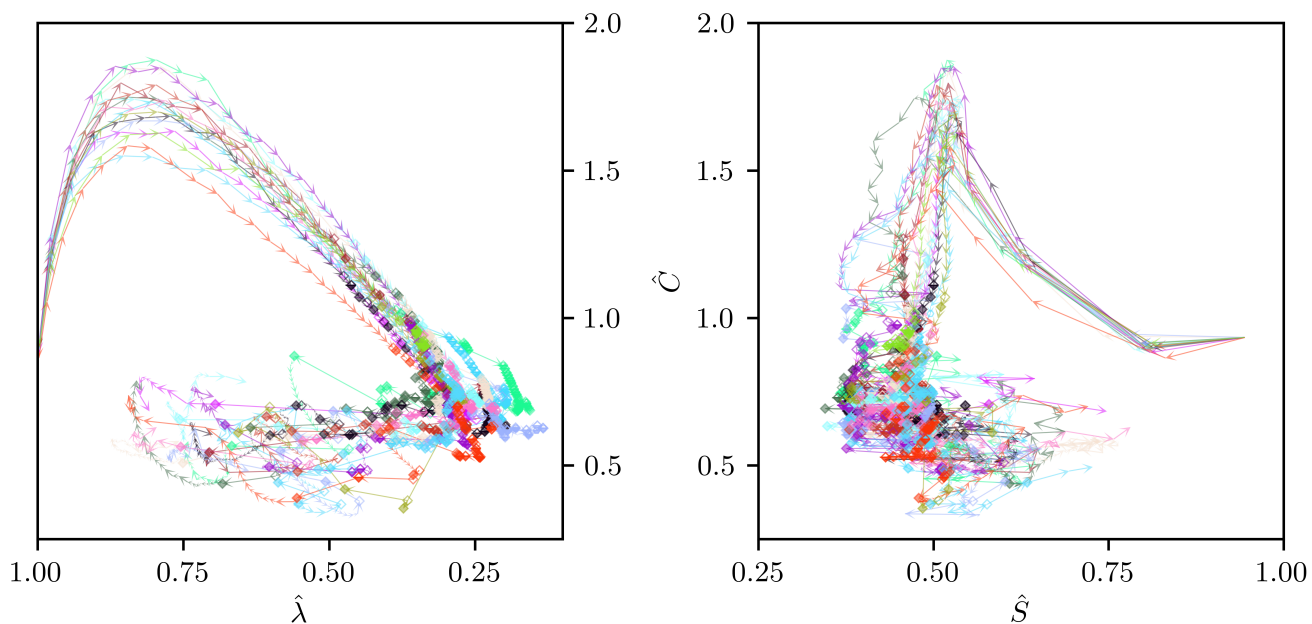


Figure 3.21: Trajectories in the geometric feature space of individually tracked primary $\phi_{4\lambda}$ structures in DHIT. Markers indicate start (filled) and end (empty) points of sections of the branch where only continuation events occur: \blacklozenge denotes a section start resulting from a split event, \blacklozenge indicates a splitting section end. Half-filled markers \blacklozenge are used if the section consists only of one frame. Analogously, \blacklozenge and \blacklozenge are used for merge and compound events, respectively.

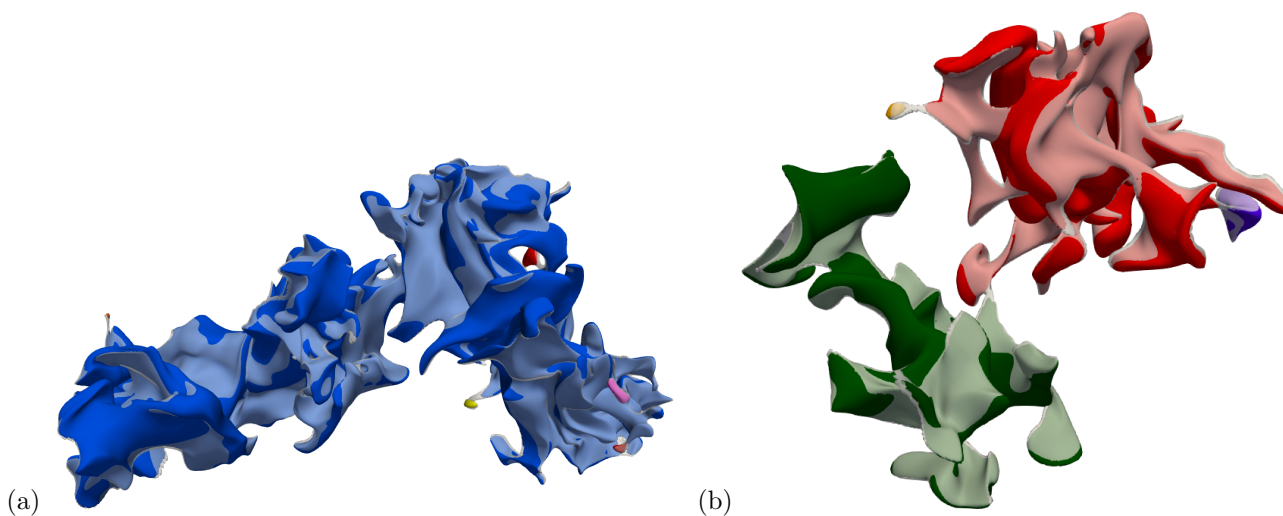


Figure 3.22: Higher-order split of $\phi_{4\lambda}$ structures (translucent) in DHIT with one dominant target structure (blue) occurring at an earlier stage of a structure's evolution (a) and a higher-order split with two dominant target structures (in red and green) occurring at a later stage of a structure's evolution (b).

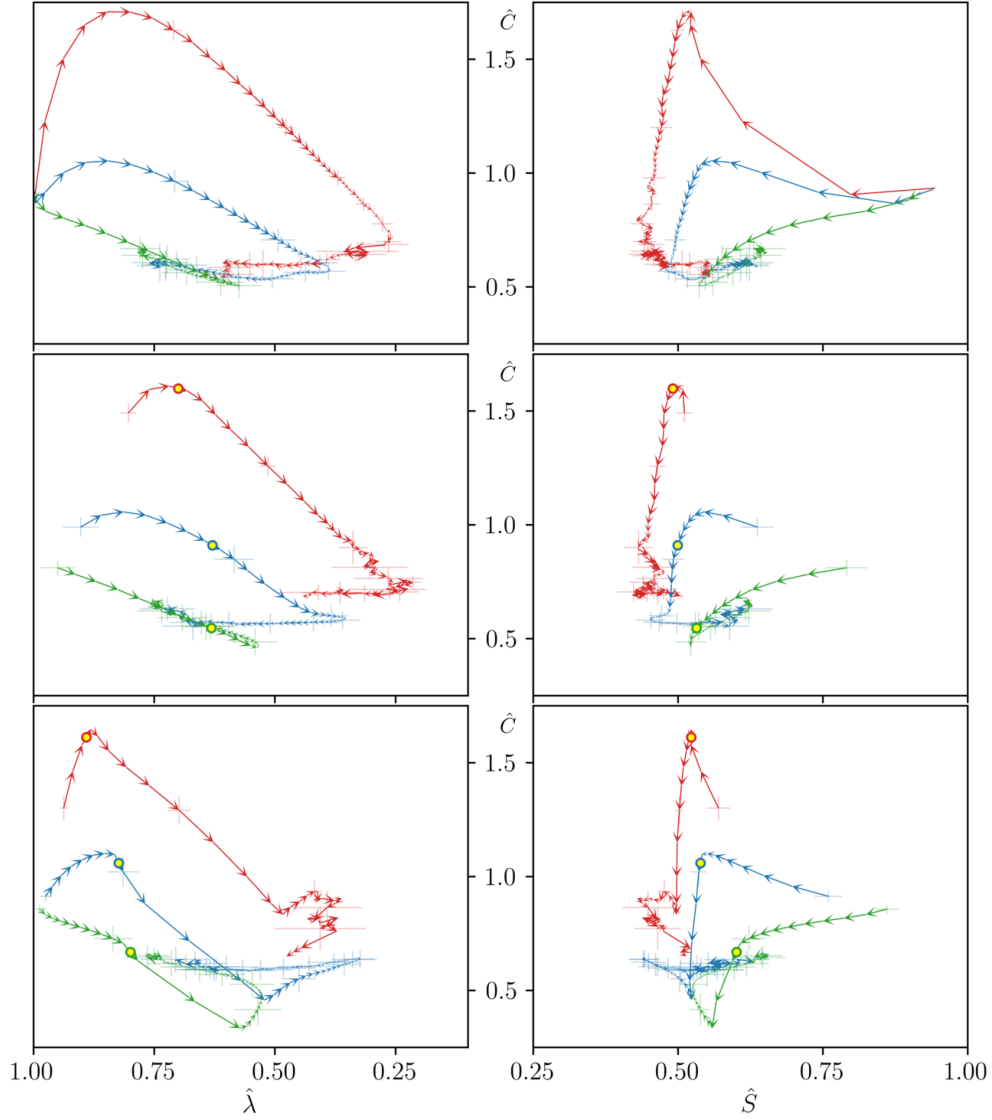


Figure 3.23: Ensemble mean trajectories of primary structures of ϕ_λ (green), $\phi_{2\lambda}$ (blue), and $\phi_{4\lambda}$ (red), in the $(\hat{S}, \hat{C}, \hat{\lambda})$ geometric space, for DHIT (top), STI with $M = 1.5$ (center), and STI with $M = 3.0$ (bottom) cases. Arrows at each tracking time step indicate the direction of time. Error bars indicate one standard deviation of ensemble trajectories in each geometric parameter, plotted every eighth tracking step for clarity. The yellow circle marks the mean shock-crossing tracking step for the STI cases. Each trajectory considers tracking steps relative to its own creation event.

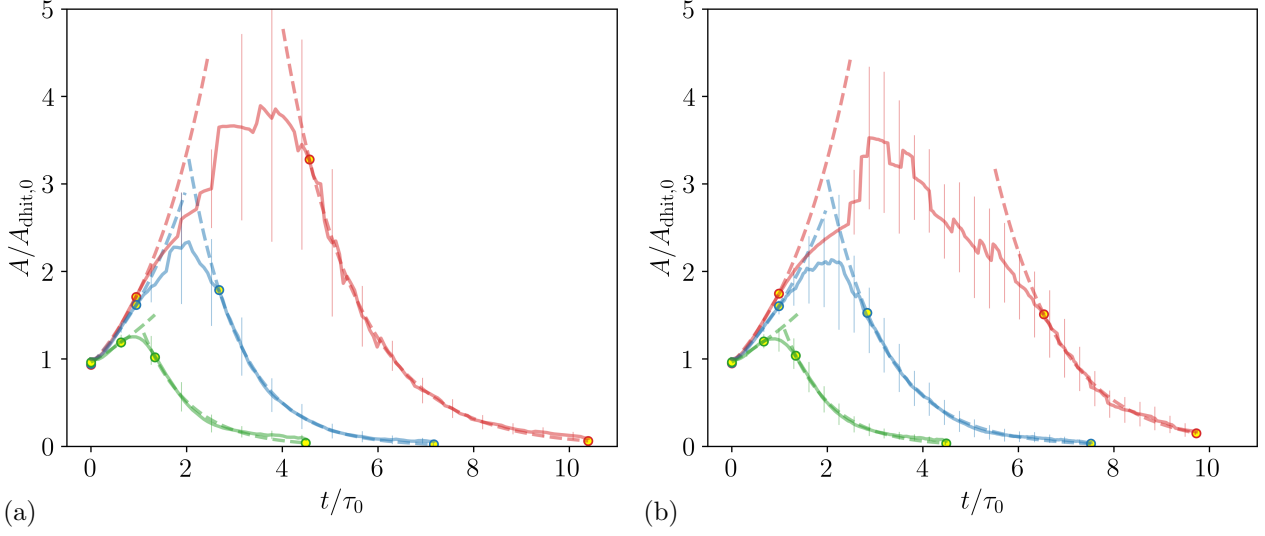


Figure 3.24: Temporal evolution of the normalized surface area (solid) of primary ϕ_λ (green), $\phi_{2\lambda}$ (blue) and $\phi_{4\lambda}$ (red) structures in DHIT corresponding to STI $M = 1.5$ (a) and STI $M = 3.0$ (b). Errorbars include one standard deviation, plotted every 8-th data point. Dashed line represent fitted curves of exponential growth and decay. Yellow marker indicate the range used for the curve fitting.

less pronounced for the $\phi_{2\lambda}$ than for the $\phi_{4\lambda}$ structures. Furthermore, the mean ensemble trajectory of ϕ_λ structures does not exhibit an initial increase in \hat{C} at all, since the initially smaller ϕ_λ structures are less convoluted by the background turbulence eddies.

Due to the differences in surface convolution and stretching, the mean area evolution of primary structures in DHIT, shown in figure 3.24, also depends on their initial size, but suggests a certain degree of self-similarity. Exponential area growth has been proposed in the literature for the evolution of Lagrangian structures in statistically stationary turbulence (Etemadi, 1990; Girimaji & Pope, 1990, 1992; Goto & Kida, 2007; Yang *et al.*, 2010). In the present study, despite the temporally decaying turbulence and diffusion acting on the (non-Lagrangian) passive scalar structures, an exponential time dependence of the form $A(t/\tau_0)/A_0 = a \exp(bt/\tau_0)$ can be fitted to the early, stretching-dominated surface area evolution, with a and b constant. A larger initial area growth is observed for initially larger structures. The area growth rate is larger for the DHIT with a slightly higher initial Taylor-Reynolds number ($Re_\lambda = 46$ compared with 42, corresponding to inflow turbulence used for STI cases with $M = 1.5$ and 3.0, respectively). After the initially exponential growth, the area growth rate decreases as the dissipation takes over the surface stretching and the surface area reaches its maximum. In the following diffusion-dominated regime, an exponential decay can be fitted to the area decrease. The area decay rate is faster for more stretched surfaces with a larger surface area (i.e., $\phi_{4\lambda}$ structures). Cascade splits, which potentially shrink the surface area significantly, occur in this period of exponential surface area decay.

As the initial structure size increases (from ϕ_λ to $\phi_{4\lambda}$), the ensemble mean trajectory reaches lower values of the shape index, \hat{S} , favoring hyperbolic points, and compactness, $\hat{\lambda}$, indicative of an increased convolution. With increasing initial size, also the mean structure lifetime increases, as it takes longer for the structure to be diffused below the threshold used for iso-surfacing. Increased lifetime translates into a mean disappearance location which is

	STI with $M = 1.5$					STI with $M = 3.0$				
	DHIT			STI		DHIT			STI	
	ϕ_λ	$\phi_{2\lambda}$	$\phi_{4\lambda}$	ϕ_λ	$\phi_{2\lambda}$	ϕ_λ	$\phi_{2\lambda}$	$\phi_{4\lambda}$	ϕ_λ	$\phi_{2\lambda}$
$(\bar{x}_d - x_s)/\lambda$	10.17	28.99	54.2	7.78	15.33	12.9	31.5	64.6	5.9	11.94
$(\bar{x}_{c,s} - x_s)/\lambda$	3.61	9.93	15.71	3.64	7.34	6.5	12.9	23.9	3.11	5.5
$(\bar{x}_{c,e} - \bar{x}_{c,s})/\lambda$	1.09	2.63	7.40	0.81	3.07	0.51	0.94	2.1	0.39	1.6

Table 3.5: Normalized mean disappearance location (top row), mean start location of split cascade (center row) and mean spatial length of split cascade (bottom row) for primary structures of type $\phi_\lambda, \phi_{2\lambda}$ and $\phi_{4\lambda}$ in STI for $M = 1.5$ and $M = 3.0$ and corresponding DHIT flow configurations.

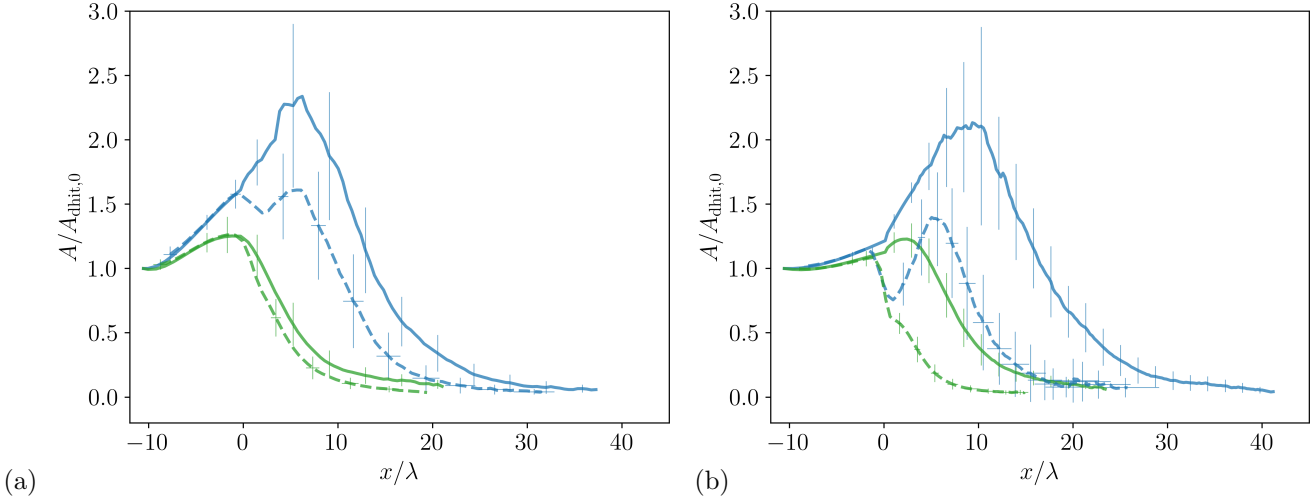


Figure 3.25: Streamwise evolution of the surface area of primary ϕ_λ (green) and $\phi_{2\lambda}$ (blue) structures in DHIT (solid) and STI (dashed) normalized by its initial DHIT value. STI $M = 1.5$ and corresponding DHIT shown in (a), while STI $M = 3.0$ and corresponding DHIT in (b). Errorbars include one standard deviation, plotted every 8-th data point. Time coordinate in DHIT is transformed into spatial coordinate using equation 3.3.

farther downstream after translating time to streamwise coordinate (equation 3.3). The lifetime reduction indicates that the mixing of larger structures benefits more from the shock-interaction: for the shock interaction with $M = 1.5$, the mean disappearance location of ϕ_λ structures is reduced by 24%, while for $\phi_{2\lambda}$ structures the reduction is 47%. For the higher Mach number case, the reductions are 54% and 62%, respectively. The dependence of the lifetime reduction on the structure size correlates with that of the surface-averaged scalar gradient magnitude on the structure size. Furthermore, the cascade splits start at a later streamwise location and are spread over a larger streamwise length (see table 3.5). As the pre- and post-shock mean streamwise velocity differs between the two considered STIs, the streamwise locations of the corresponding DHITs differ in table 3.5. At the end of their lifetime, structures of all scalar fields considered tend towards the same geometrical attributes: moderate compactness and dimensionless curvedness with predominantly elliptical points.

In the STI flow configurations, the structures have already been deformed individually at the time when they become closed surface. Therefore, their trajectories in the feature space do not have a common starting point and differ based on initial structure size (see center and bottom rows of figure 3.23). The more upstream half of the

structure is less convoluted than the more downstream half when the structures become closed as the downstream half already spent more time within the domain. This discrepancy between front and rear half is more pronounced for bigger structures (as it takes longer for them to be closed) and for the lower Mach number case $M = 1.5$ due to the lower pre-shock velocity. As a consequence, especially for the $\phi_{4\lambda}$ structures the maximum curvedness is less than in the DHIT as the more upstream part of the structure is less convoluted than the more downstream part and thus overall the maximum curvedness is less than in the DHIT. The shock interaction lowers the lifetime of the primary structures in comparison to the corresponding DHIT as the mixing is enhanced (table 3.5). Additionally, the shock induces an earlier starting point of the cascade split events. Compression of the structures by the shock wave lowers the dimensionless curvedness \hat{C} (flattening the surface normal to the compression) and the compactness $\hat{\lambda}$ (higher stretching, thus favouring split events) compared to the DHIT. The deformation induced across the shock, particularly for $M = 3$ is more pronounced for the smaller ϕ_λ structures, as a larger portion of the structure is compressed by the shock at once. For the STI $M = 3$ case, the curvedness of the ϕ_λ and $\phi_{2\lambda}$ structures reaches a minimum after the structures are completely processed by the shock, and increases afterwards due to shock-induced turbulence amplification.

The amplification of turbulence across the shock also leads to a surface area re-growth of $\phi_{2\lambda}$ structures in the post-shock region that is stronger for the higher Mach number (figure 3.25). This period of area re-growth correlates with the decreasing behavior of the compactness parameter, $\hat{\lambda}$, in the feature space trajectories (figure 3.23) after the structures have been processed by the shock. While the amplification of the surface-averaged vorticity magnitude is similar for all initial structure sizes (figure 3.17), the area re-growth is larger for $\phi_{2\lambda}$ than for ϕ_λ structures. At this post-shock location the size of the ϕ_λ structures is smaller than the Taylor microscale and, therefore, those structures are mainly advected by the (amplified) eddies. In contrast, $\phi_{2\lambda}$ structures are still large enough to be also stretched by the (smaller) turbulence eddies. Prior to the surface area re-growth in the post-shock region, the higher Mach number compression decreases the surface area more significantly. Structures of $\phi_{4\lambda}$ exhibit a similar behavior up to a certain post-shock location where consecutive structures in the streamwise direction, decelerated by the shock, merge with each other and pile up, generating structures of large streamwise extent. A bigger initial spacing of the structures is needed to avoid this pile up process. The cascades of split events start at slightly higher values of dimensionless curvedness and end at slightly lower values of the compactness. Even though the cascade (sequence of consecutive splits) ends there, the structures are still highly stretched, thus further splits are likely to occur at later times of their evolution. In contrast to the DHIT, a number of splits (or compound events) occur for the STI cases, which affect the geometric signature of the primary structures early in their evolution. The number of primary $\phi_{4\lambda}$ structures in the STI configurations with very long lifetime is too small to be statistically significant and therefore the final steps of their lifetime are excluded from the mean ensemble trajectory. Therefore those mean trajectories do not reach similar values of the geometrical properties as the other two mean trajectories at the end of the trajectory.

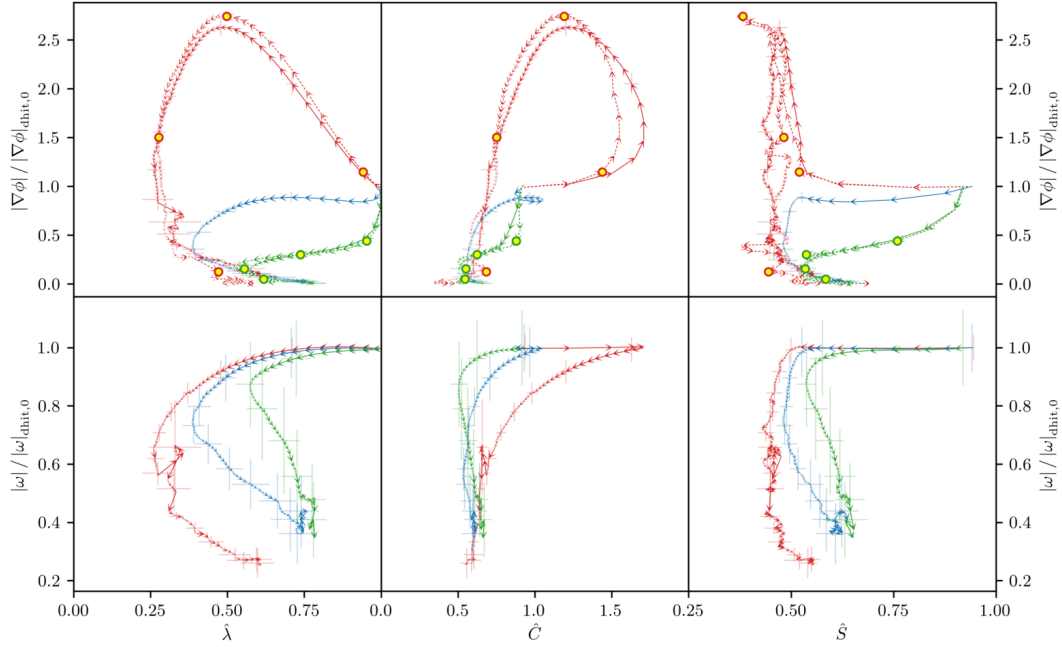


Figure 3.26: Ensemble mean trajectories (solid) of primary structures of ϕ_λ (green), $\phi_{2\lambda}$ (blue), and $\phi_{4\lambda}$ (red) for DHIT in the $(\hat{S}, \hat{C}, \hat{\lambda})$ geometric space with scalar gradient (top) and vorticity magnitude (bottom). Physical quantities normalized by their initial value in DHIT. Error bars include one standard deviation of ensemble trajectories in each geometric and physical parameter, plotted every eighth tracking step for clarity. For ϕ_λ and $\phi_{4\lambda}$ structures the individual closest-to-mean trajectory (dotted) is plotted with highlighted steps (yellow markers) corresponding to the structures rendered in figure 3.29.

3.8.2 Trajectories with scalar dissipation and vorticity

The study of ensemble trajectory statistics is extended to include the evolution of two physical quantities averaged on the surfaces: the magnitude of the scalar gradient (indicative of the scalar dissipation on the surface) and the vorticity magnitude, shown in figure 3.26 for the DHIT case. For $\phi_{4\lambda}$ structures, the scalar gradient magnitude exhibits a rapid initial increase, due to stretching, of the initial scalar transitional region, as previously observed in the streamwise evolution (figure 3.17). The peak in curvedness is reached significantly earlier than the peak in the scalar gradient magnitude: while the curvedness already decreases, the scalar gradient magnitude still increases. The geometry of $\phi_{4\lambda}$ structures, especially the curvedness and the absolute shape index, develops much faster than both the scalar gradient and the vorticity magnitude at early stages of the trajectory. In contrast, for the smaller ϕ_λ structures, the decrease in scalar gradient magnitude is initially more pronounced than the geometrical changes. During the cascade split events, not only is the geometrical signature of the primary structure marginally affected, as previously explained, but also both surface-averaged physical quantities. Since the turbulence is already fully developed at structure initialization, the surface-averaged vorticity magnitude exhibits a monotonic decay in DHIT. Due to their longer lifetime, the vorticity magnitude reaches lower values on the larger $\phi_{4\lambda}$ structures.

The individual trajectory closest to the ensemble mean in terms of geometry and scalar gradient magnitude (and thus, scalar dissipation) is also shown in figure 3.26 for $\phi_{4\lambda}$ and ϕ_λ structures. Figure 3.29 shows the structures

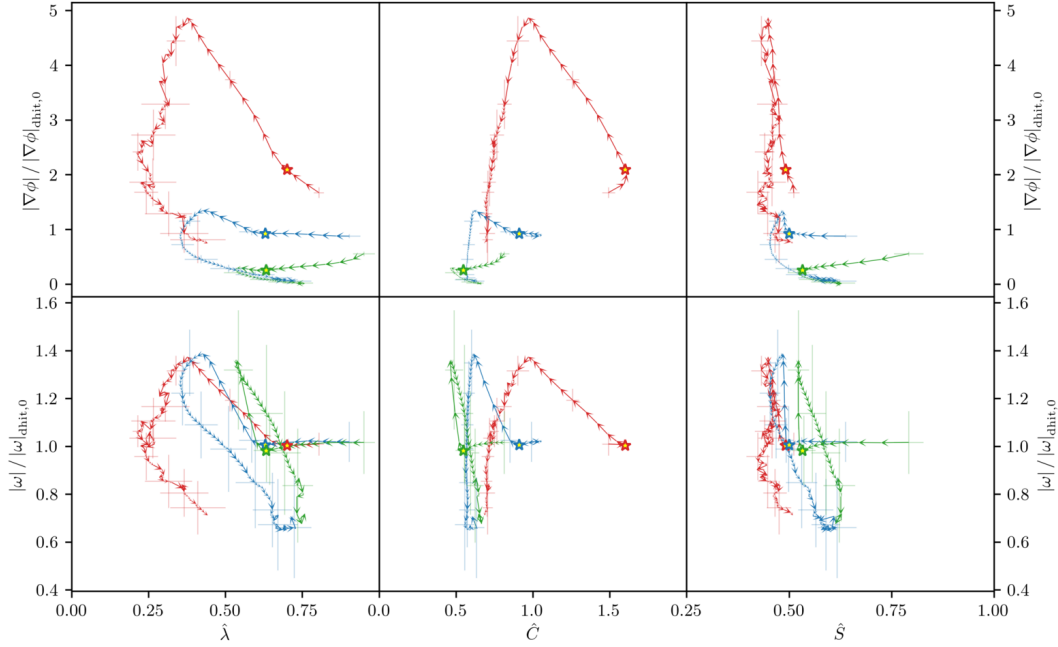


Figure 3.27: Ensemble mean trajectories of primary structures of ϕ_λ (green), $\phi_{2\lambda}$ (blue), and $\phi_{4\lambda}$ (red) for STI with $M = 1.5$ in the $(\hat{S}, \hat{C}, \hat{\lambda})$ geometric space with normalized scalar gradient (top) and vorticity magnitude (bottom). The yellow star marks the mean shock-crossing tracking step. Other plot elements are the same as in figure 3.26.

corresponding to four highlighted steps of the closest-to-mean trajectory rendered with the alignment between the scalar gradient and the γ eigenvector of the strain-rate mapped on the surface. The area-based joint and marginal probability density functions of S and C for the structures at each highlighted step are also shown, indicating that for both structure sizes (and for all four highlighted steps) the largest curvedness on the surfaces is found at parabolic points ($S = 0.5$) that occur on the ridges of the convoluted surfaces. This is in agreement with Girimaji & Pope (1992) who reported that highly curved surface elements of propagating surfaces in isotropic turbulence are of cylindrical shape. The correlation, found in §3.7, between flatter surface regions and close alignment between the scalar gradient and the most compressive strain-rate eigenvector, γ , emerges shortly after the ad-hoc spherical initialization, weakening later in the structure lifetime.

The ensemble mean trajectories of primary structures in STI with $M = 1.5$ differ from those in DHIT due to the open structure evolution close to the inlet and the impact of the shock on the geometrical and physical surface properties (see figure 3.27). As previously noted, the number of primary $\phi_{4\lambda}$ structures with very long lifetime is too small to be statistically significant and therefore the final steps of their lifetime are excluded from the mean ensemble trajectory. Hence, the mean $\phi_{4\lambda}$ trajectory does not reach scalar gradient and vorticity magnitudes as low as in the mean trajectories of ϕ_λ and $\phi_{2\lambda}$ structures. While the increase in scalar gradient magnitude depends on the size of the primary structures, the increase of the vorticity magnitude is the same for all structure sizes. These findings are consistent with figure 3.17, where all structure types, not only primary ones, have been investigated.

The stronger compression and quicker closure of incoming structures in the STI $M = 3.0$ configuration change the ensemble mean trajectories in terms of geometrical and physical structure properties compared to the lower

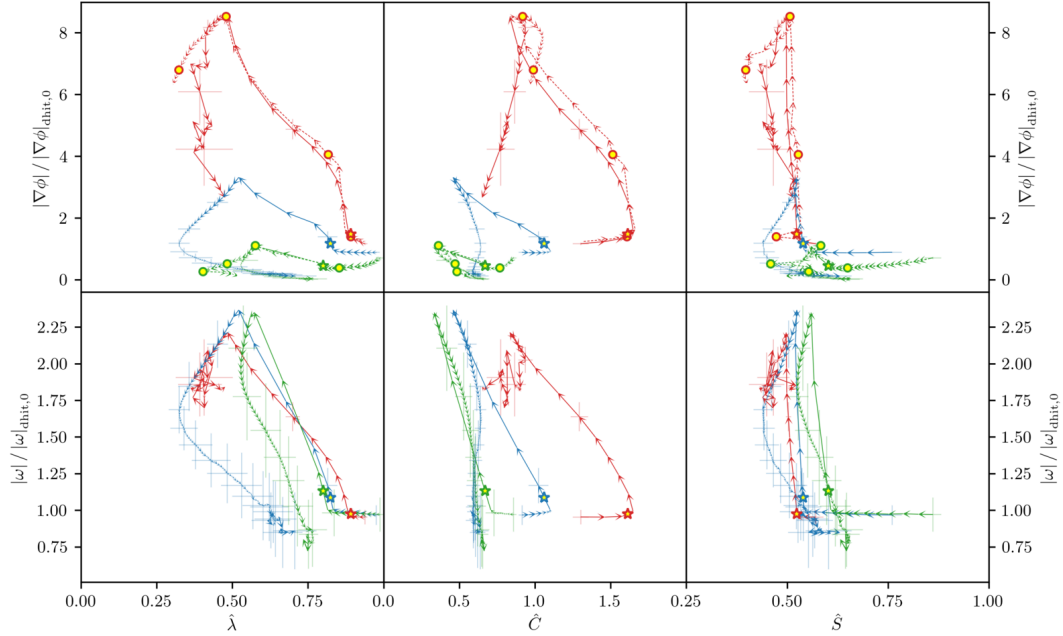


Figure 3.28: Ensemble mean trajectories (solid) of primary structures of ϕ_λ (green), $\phi_{2\lambda}$ (blue), and $\phi_{4\lambda}$ (red) for STI with $M = 3.0$ in the $(\hat{S}, \hat{C}, \hat{\lambda})$ geometric space with normalized scalar gradient (top) and vorticity magnitude (bottom). For the ϕ_λ and $\phi_{4\lambda}$ structures the individual closest-to-mean trajectory (dotted) is plotted with highlighted steps (yellow markers) corresponding to the structures rendered in figure 3.30. Other plot elements are the same as in figure 3.27.

Mach number case (figure 3.28). The smaller the initial size of the structures, the lower their curvedness is at the point of highest scalar dissipation. Compared to the DHIT (figure 3.26), $\phi_{4\lambda}$ structures reach significantly higher average scalar dissipation, and at lower curvedness resulting from the flattening normal to the shock. Considering the evolution after the peak in scalar gradient magnitude, the curvedness of the ϕ_λ and $\phi_{2\lambda}$ structures decays in DHIT, is nearly constant for STI with $M = 1.5$, and has a period of increase for the STI with $M = 3.0$. Since the pile-up process of $\phi_{4\lambda}$ structures farther downstream of the shock generates structures of large streamwise extent, the scalar gradient and vorticity magnitude do not evolve toward the low values observed for smaller (ϕ_λ , $\phi_{2\lambda}$) structures in the final stage of their ensemble mean trajectories.

The average alignment between scalar gradient and the most compressive strain-rate eigenvector conditioned on S and C at highlighted steps of the closest-to-mean trajectory (figure 3.30) confirms again the correlation between flatter regions and high alignment for both initial structure sizes.

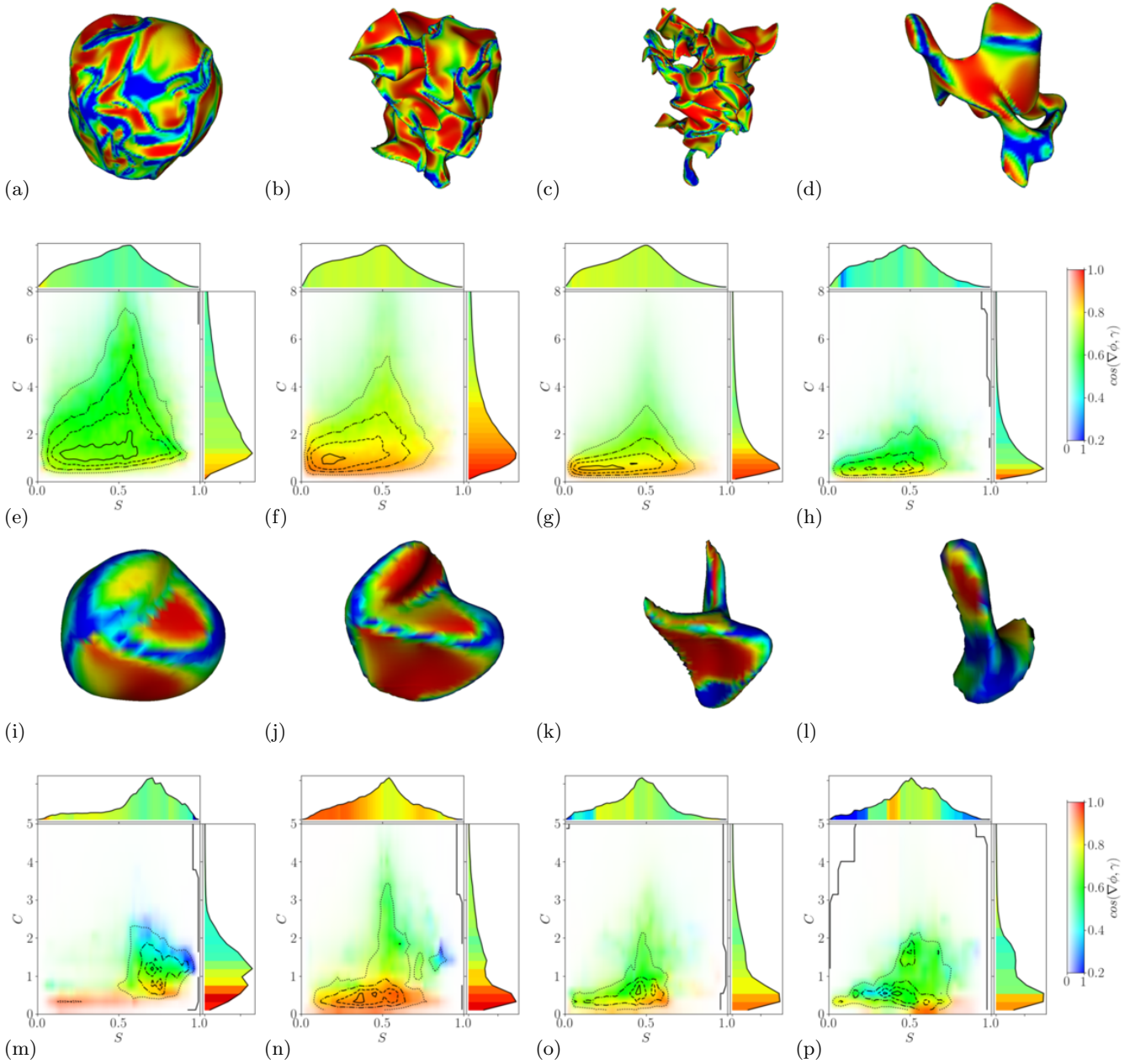


Figure 3.29: Renderings of primary $\phi_{4\lambda}$ (a-d) and ϕ_λ (i-l) structures in DHIT with the alignment between scalar gradient and the γ eigenvector of the strain-rate mapped on the surface, corresponding to the highlighted steps of the closest-to-mean trajectory in figure 3.26 (left to right, not to scale). The corresponding distribution of average alignment of scalar gradient and γ conditioned on S and C (hue), accounting for the area-based joint PDF is shown underneath each rendered structure (e-h,m-p). The horizontal direction in the renderings corresponds to the average shock-normal direction.

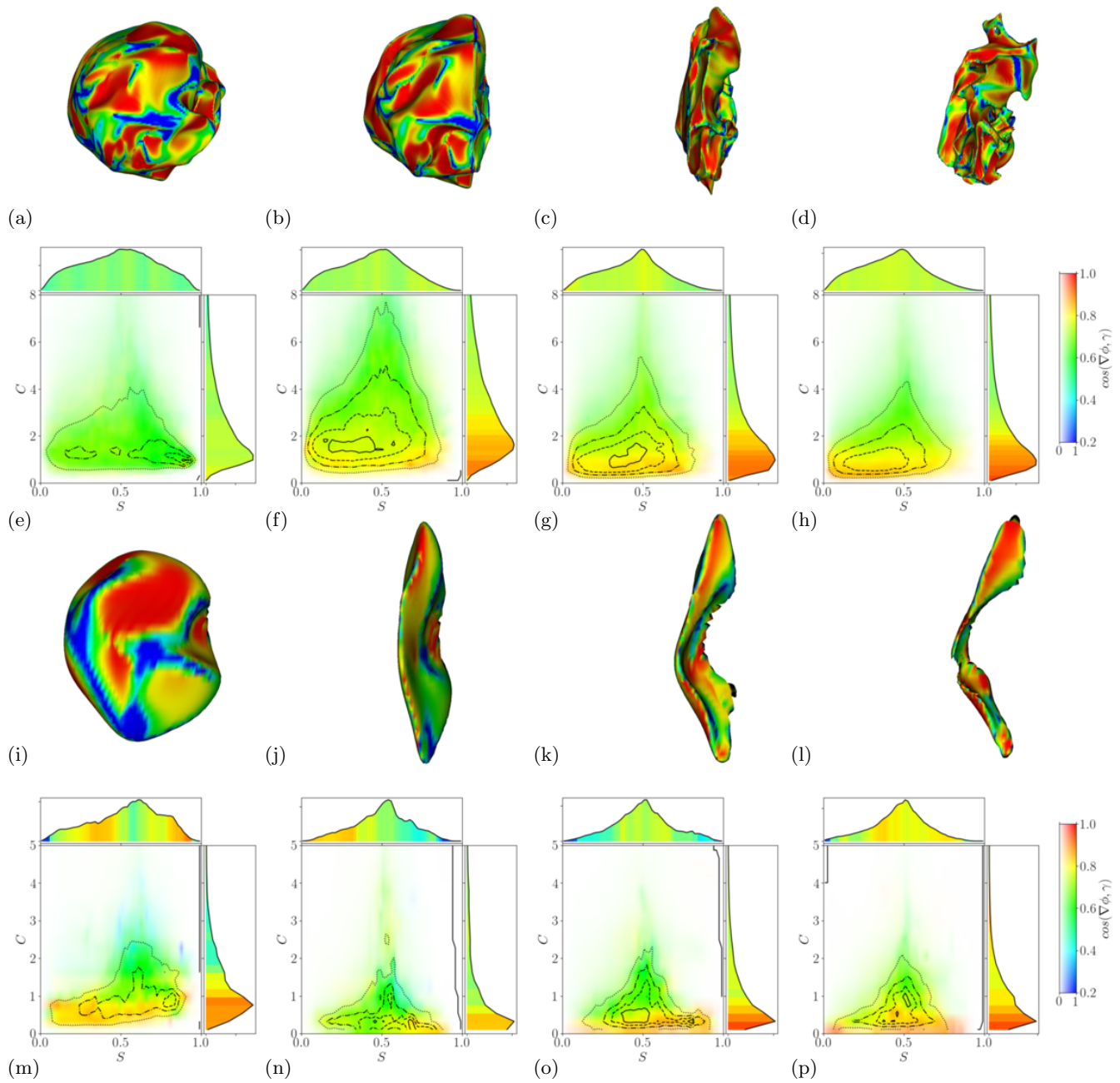


Figure 3.30: Renderings of $\phi_{4\lambda}$ (a-d) and ϕ_λ (i-l) in STI $M = 3.0$ and corresponding distribution of average alignment of scalar gradient and γ conditioned on S and C (hue) accounting for the area-based joint PDF (e-h,m-p) corresponding to the highlighted steps of the closest-to-mean trajectories in figure 3.28. Other plot elements are the same as in figure 3.29.

Chapter 4

Vortical structures in temporally developing compressible mixing layers

4.1 Introduction

The mixing of two parallel streams and, hence, the growth rate of the mixing / shear layer forming between them is substantially inhibited at high Mach number compared to its incompressible counterpart. Statistical analyses have shown that compressibility has a stabilizing effect which reduces the turbulence levels and growth rate of the mixing layer (Papamoschou & Lele, 1993; Cambon *et al.*, 1992; Friedrich & Bertolotti, 1996). To measure compressibility a set of dimensionless parameters is used. Denoting the free stream velocity, density and speed of sound in the two free streams as U_i, ρ_i, c_i , for $i = 1, 2$, the convective Mach number, $M_c = (U_2 - U_1)/(c_2 + c_1)$, is a measure of the compressibility of the mixing layer, where equal specific heat capacities are assumed (Bogdanoff, 1983). Other compressibility measures are the turbulent Mach number, $M_t = u_{\text{rms}}/c$, where u_{rms} is the root mean square velocity, and the gradient Mach number, $M_g = Sl/c$, where S denotes the mean velocity gradient and l is a characteristic length scale in the direction of the gradient. The gradient Mach number was found especially useful to distinguish the stronger compressibility effects in shear layers from those in boundary layers (Sarkar, 1995).

In the literature, the canonical flow configuration of a mixing layer has been studied in terms of a temporally or spatially developing mixing layer, focusing on the dynamics of large, organized structures (Brown & Roshko, 1974), process of transition to turbulence and the self-similar state of the shear layer including passive and reactive scalar transport (Clemens & Mungal, 1995; Karasso & Mungal, 1996). From perturbed laminar initial (or inlet) conditions the flow develops to a turbulent, self-similar state via Kelvin-Helmholtz instability (Moser & Rogers, 1993; Rogers & Moser, 1994). During the transition to turbulence the occurrence of different structures, such as rollers and rib vortices (Moser & Rogers, 1991; Rogers & Moser, 1992) and Λ -vortices, harpin-vortices and “flower” structures (Zhou *et al.*, 2012) has been observed. The dynamics of these structures have been found to be sensitive to initial (inlet) conditions. Statistical features of the turbulence in its self-similar state, such as the longitudinal velocity structure functions and scaling exponents, follow the observations for homogeneous isotropic turbulence for small scales, but differ from those for larger scales (Attili & Bisetti, 2012). Linear stability analysis revealed that the mixing layer becomes truly three dimensional above a convective Mach number of $M_c = 0.6$ (Sandham & Reynolds, 1990). The formation of shocklets in three-dimensional compressible mixing layer was reported for a convective Mach number of

$M_c = 1.2$ (Vreman *et al.*, 1995; Fu & Li, 2006). However, Zhou *et al.* (2012) observed shocklets in a spatial developing mixing layer at a lower convective Mach number of $M_c = 0.7$.

It is of interest how the phenomenology of a decreased growth rate of mixing layers with increased compressibility can be explained from a standpoint of turbulence structures. To gain insight into these structural mechanisms, the geometrical analysis and the flow feature tracking, §2, are applied to vortical structures extracted as isosurfaces of the Q-criterion from the mixing layer. Despite of the introduction of the hairpin vortex concept by Theodorsen in the 1950s, the occurrence, persistence and significance of such vortices in shear flow configurations is still subject of debate. Therefore, special interest exists to study the evolution of hairpin-like vortices in the mixing layer.

4.2 Problem formulation and numerical setup

4.2.1 Setup of the flow configuration

Two measures of the thickness of the mixing layer are commonly used in the literature: the vorticity thickness

$$\delta_\omega(t) = \frac{\Delta U}{|\partial \tilde{u}/\partial y|_{\max}} \quad (4.1)$$

and the momentum thickness

$$\delta_\theta(t) = \frac{1}{\rho_0 \Delta U^2} \int_{-\infty}^{\infty} \bar{\rho} \left(\frac{\Delta U}{2} - \tilde{u}_1 \right) \left(\frac{\Delta U}{2} + \tilde{u}_1 \right) dy \quad (4.2)$$

where the Favre-averaged streamwise velocity component is $\tilde{u}_1 = \rho \bar{u}_1 / \bar{\rho}$ and the free stream velocity difference is denoted as $\Delta U = U_2 - U_1$. Those two thickness measures can be used as characteristic length scales to form the vorticity thickness Reynolds number $Re_\omega = \rho \Delta U \delta_\omega / \mu$ and the momentum thickness Reynolds number $Re_\theta = \rho \Delta U \delta_\theta / \mu$. Besides the convective Mach number, M_c , the density ratio between the free streams ρ_2 / ρ_1 was also found to affect the growth rate of the mixing layer. This density ratio may be affected by the gas composition and heat release, such as in high-speed propulsion applications. In high-speed flow, a reduced (momentum thickness) growth rate for unequal free stream densities has been linked to a shift of the dividing streamline towards the low-density side (Pantano & Sarkar, 2002). In the present application the equal-density case $\rho_2 / \rho_1 = 1$ is considered.

Following the configuration setup of a temporally developing mixing layer by Pantano & Sarkar (2002) and Foyi & Sarkar (2010), the mixing of two equally but opposite free streams, $-U_1 = U_2 = \Delta U / 2$, is presently studied by conducting direct numerical simulations at higher spatial resolution than in previous studies, followed by a geometric characterization and temporal tracking of the resulting turbulent structures. Figure 4.1(a) presents the simulation domain and setup. The flow is initialized such that the mean streamwise velocity component \bar{u}_1 is prescribed by a hyperbolic tangent profile, while the cross-stream and spanwise mean velocity components are set to zero:

$$\bar{u}_1 = \frac{\Delta U}{2} \tanh \left(-\frac{y}{2\delta_{\theta,0}} \right), \quad \bar{u}_2 = 0, \quad \bar{u}_3 = 0 \quad (4.3)$$

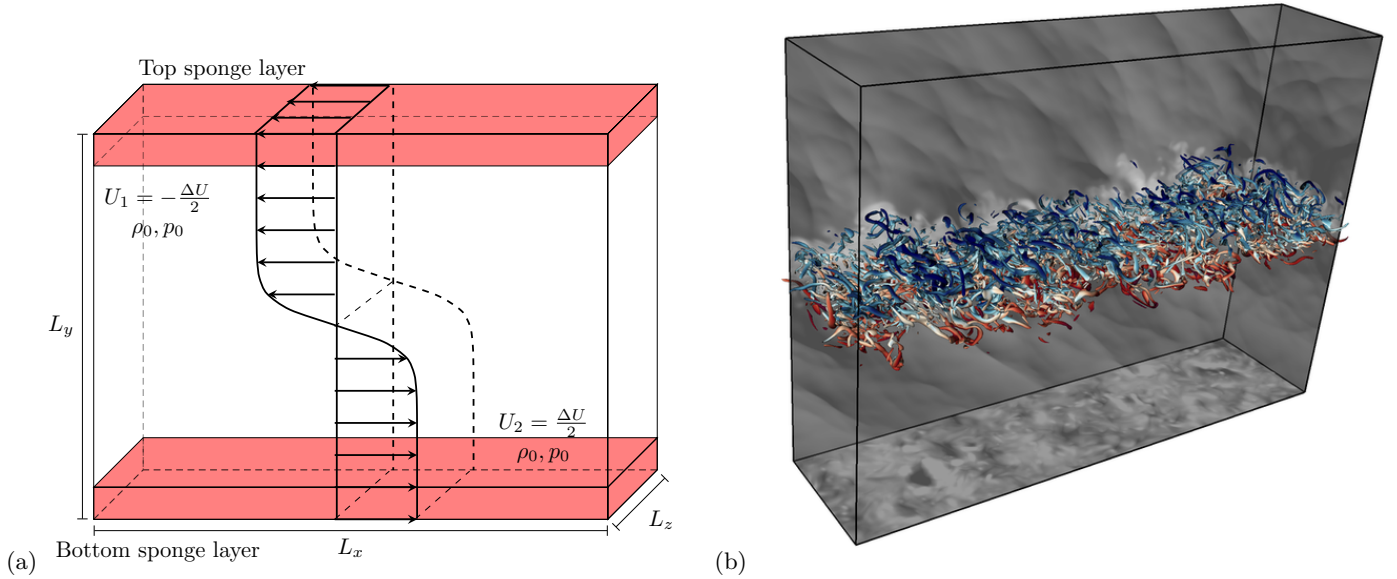


Figure 4.1: (a) Illustration of the initial condition of the temporally developing mixing layer. Mean streamwise velocity profile and sponge boundary damping indicated in red. (b) The $M_c = 1.1$ mixing layer at the beginning of its self-similar regime ($\tau = 650$). Planes with reduced opacity show the density field (white: $\rho = 0.7$, black: $\rho = 1.1$) where the center plane is artificially displaced to the bottom for visibility. Vortical structures are identified as iso-surfaces of the Q-criterion and colored by the streamwise velocity component (blue: $u_1 = -1.0$, red: $u_1 = 1.0$).

Pressure and density are initially uniform across the domain. The vorticity and momentum thickness Reynolds number have initial values of $Re_{\omega,0} = 640$ and $Re_{\theta,0} = 160$. The initial vorticity thickness is specified to be $\delta_{\omega,0} = 1$, while the momentum thickness is $\delta_{\theta,0} = 0.25$ initially. To trigger the transition to turbulence broadband fluctuations, u'_i , are added to the mean velocity components. Those fluctuations are obtained from a random field on which a isotropic turbulence spectrum (Ristorcelli & Blaisdell, 1997) is imposed as

$$E(k) \propto k^4 e^{-2k^2/k_0^2} \quad (4.4)$$

with $k_0 = 48$. Applying this initialization technique leads to a nonzero compressible velocity component derived based on a small M_t expansion of the Navier–Stokes equations. Additionally, (incompressible) fluctuations of the pressure field and linearly related density and temperature fluctuations are also initially present. To limit the velocity fluctuation to the center region of the domain a damping cross-stream function, $f(y)$, is applied in the form of a hyperbolic secant shape following Li & Fu (2003):

$$f(y) = C \operatorname{sech}^2 \left(\frac{y}{2\delta_{\theta,0}} \right) \quad (4.5)$$

with $C = 0.25$. The velocity components are initialized as $u_i = \bar{u}_i + f(y)u'_i$.

Since the size of turbulent structures in mixing layers decreases with increasing convective Mach number (Lesieur *et al.*, 2005), three convective Mach numbers of $M_c = 0.3, 0.7$ and 1.1 are considered. Whereas periodic boundary

M_c	$\tilde{L}_x \times \tilde{L}_y \times \tilde{L}_z$	$N_x \times N_y \times N_z$	$\tilde{L}_{y,\text{core}}$	$N_{y,\text{core}}$
0.3	$345 \times 560 \times 86$	$1024 \times 1127 \times 256$	299.7	908
0.7	$345 \times 480 \times 86$	$1024 \times 973 \times 256$	253.5	768
1.1	$345 \times 400 \times 86$	$1024 \times 931 \times 256$	253.5	768

Table 4.1: Grid properties of the mixing layer configuration for different convective Mach number M_c simulation cases. The domain extent is normalized by the initial momentum layer thickness.

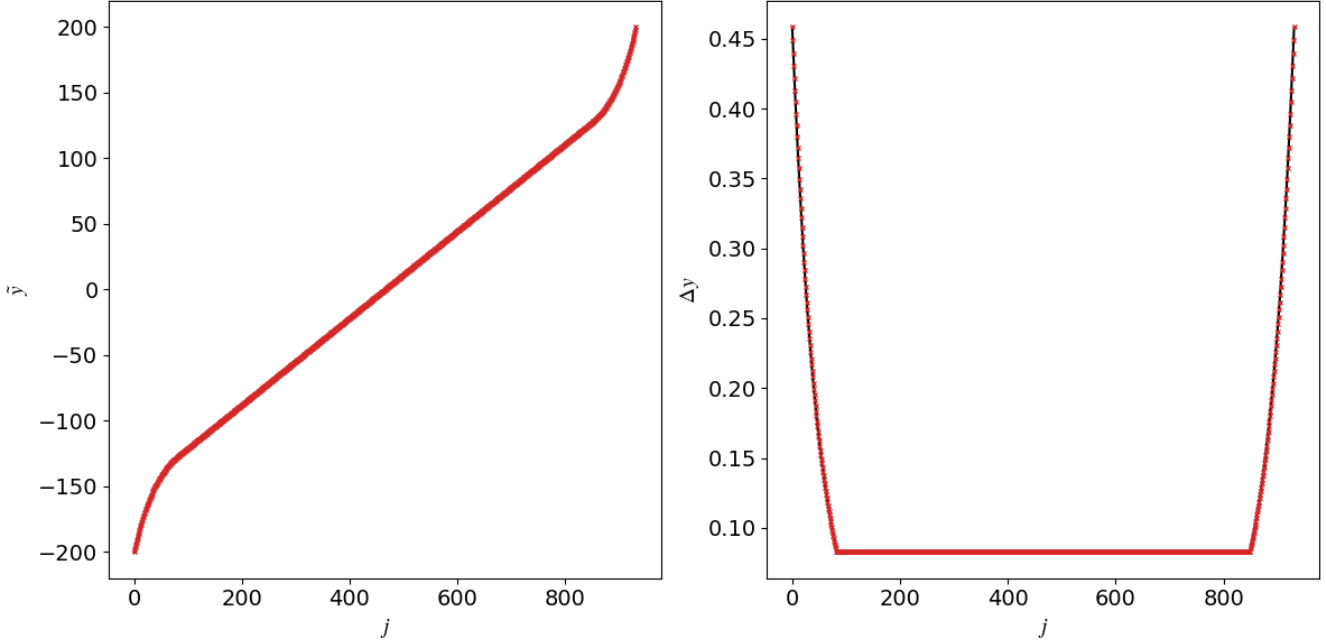


Figure 4.2: Cross-stream grid coordinates (a) and spacing (b) of the mixing layer domain for $M_c = 1.1$ as a function of the grid point index, j in that cross-stream direction.

conditions are utilized in streamwise and spanwise directions, non-reflective sponge layers are added to the bottom and top boundaries of the domain to avoid spurious acoustic reflections (Mani, 2012). Different sponge lengths of normalized cross-stream extent of $\tilde{L}_s = 153.25, 113.25$ and 73.25 are chosen depending on the convective Mach number of the mixing layer $M_c = 0.3, 0.7$ and 1.1 , respectively. All grids share the same spatial resolution in the uniformly-spaced core region of the domain (see table 4.1). The grids have a uniform spacing in stream- and spanwise direction. For the cross-stream core region of the domain, $|\tilde{y}| < 150$ ($M_c = 0.3$) or $|\tilde{y}| < 126$ ($M_c = 0.7, 1.1$), the grid spacing in the y direction is uniform, whereas a grid stretching based on the geometric series is applied towards the bottom and top boundary (figure 4.2). For the $M_c = 0.7$ and $M_c = 1.1$ configurations the beginning of the sponge layers coincides with the beginning of the grid stretching, whereas for the $M_c = 0.3$ case the sponges start in the outer region of the core of the grid.

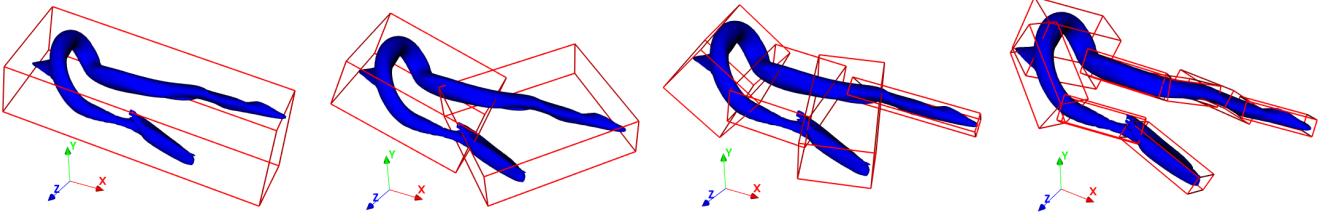


Figure 4.3: A hairpin vortex (blue) extracted from the $M_c = 1.1$ mixing layer at $\tau = 196$ and its OBB tree representation (red) of increasing depth from left to right. To capture the bent shape of the structure, the level of the tree needs to be sufficiently high.

Nearest-neighbor search								Radius search				Correspondence constraints				
f_x	f_y	f_z	$f_{\hat{S}}$	$f_{\hat{C}}$	$f_{\hat{\lambda}}$	c_{NN}	ω_{NN}	l_{ref}	ε_R	$\varepsilon_{R,rel}$	ω_R	l_{ref}	ε	ε_{rel}	ω_{OBB}	ω_{local}
$1/L_x$	$1/L_x$	$1/L_x$	1	1	1	0.5	0.5	d_{OBB}	1.2	1.4	0.1	d_{OBB}	0.075	0.15	0.1	0.3

Table 4.2: Scaling factors, confidence value, and weight for the NN search. Tolerance and weight utilized in the radius search. Structure reference length, tolerance, relaxed tolerance and weight used for both OBB distance and surface proximity correspondence constraints, as defined in §2.4.2-2.5.

4.2.2 Setup of the tracking algorithm

Vortical structures in the mixing layer, identified and extracted as detailed in §4.4, are individually tracked by applying the flow feature tracking methodology as described in §2. To increase the performance of the tracking algorithm, in the *OBB distance constraint*, see §2.5, a OBB tree representation of varying depth, fig. 4.3, is used rather than always using the root node of the tree to represent the structures in the GJK-distance calculation. Benefiting from an improved surface representation, the distances between the leaf nodes of a source-target correspondence pair are calculated and more false correspondences are rejected in the *OBB distance constraint*, which otherwise would be rejected by the computationally more expansive *surface proximity constraint*. If a distance between leaf nodes satisfies the constraint, computation of all other (not yet determined) distances is skipped. The level of the OBB tree used to represent a structure is based on the compactness parameter $\hat{\lambda}$ and the absolute shape index \hat{S} : Larger stretching and an increased number of hyperbolic surface points are used as indicators that a higher level of the tree is needed to capture the bent surface shape correctly.

The settings of the tracking algorithm are summarized in tables 4.2 and 4.3. The main difference in the setting compared to the tracking of the passive scalar structures, see §3, are: (i) Since the Q-structures are not convoluted

Conservation			Surface points	Conservation							(Sub-)Split/Merge		
Compound			Merge, Sub-merge	Compound			Merge		Split	Continuation		Conservation	Confidence
ε	ε_{rel}	c_{min}	N_{pts}	ε	ε_{rel}	c_{min}	ε	ε	ε				
0.8	0.7	0.5	350	0.6	0.6	0.75	0.5	0.5	0.25		0.8	0.2	

Table 4.3: Settings of the event constraining and event building in the graph. Type of the constraint and event type the constraint acts on. Event constraints are applied in sequence from left to right. Weights used for the dominant target/source selection to build (sub-)split/merge events in the graph data structure.

M_c	$\tau_{ss,s}$	$\tau_{ss,e}$	$\dot{\delta}_{\theta,ss}$
0.3	350	700	0.01308
0.7	500	950	0.009572
1.1	650	1050	0.008739

Table 4.4: Start $\tau_{ss,s}$ and end $\tau_{ss,e}$ time of the self-similar regime for different convective Mach number M_c . $\dot{\delta}_{\theta,ss}$ denotes the slope of the linear regression conducted in the self-similar regime.

compared to the passive scalar structures, the diagonal of the structure OBB is used as a reference length for the radius correspondence search and also for the spatial constraints rather than the μ parameter. (ii) The conservation-based events constraints are based on the volume rather than on the estimated mass of the structure. (iii) The weights used for the dominant target/source selection for the construction of split, merge or compound events in the graph data structure are based on conservation (volume) and confidence, i.e. the lifetime weight is not used for (sub-)merge events.

4.3 Validation of self-similarity and growth rate of the mixing layer

The temporal development of the mixing layer configuration of $M_c = 0.7$ is visualized in fig. 4.4 by means of the density distribution on the $z = 0$ plane spanning in streamwise (x) and cross-stream (y) direction. Due to friction between the two streams the temperature rises in the mixing layer, which results in a decrease of density as the overall pressure is nearly constant. Local pressure minima associated with vortices and acoustic waves emanating from the mixing layer are hinted in the density visualization.

Self-similar solutions can be derived for the equations and boundary conditions governing a turbulent mixing layer of sufficiently high Reynolds number (Townsend, 1980). The time necessary to reach self-similar state is influenced by the choice of method and its perturbation strength used to generate initial random fluctuation in the flow field. The requirements for the mixing layer to be in the self-similar state are linear thickness growth and the independence of the shapes of the mean velocity and Reynolds stress profiles of time (or downstream distance in spatial developing layers) when scaled by the velocity difference and local mixing layer thickness (Mehta, 1991). These three indications of a self-similar period in the mixing layer evolution are validated in the following for the three considered simulation cases.

The growth of the mixing layer, expressed in terms of momentum thickness, is shown in Fig. 4.5a. A period of linear growth can be found in all three momentum thickness evolutions as indicated by the linear regression of high correlation coefficient in this period. With increasing convective Mach number a longer time of transition is observed until the mixing layer reaches its period of linear growth. Again indicative of turbulence inhibitions by higher M_c , the momentum thickness growth rate

$$\dot{\delta}_{\theta} = \frac{1}{\Delta U} \frac{d\delta_{\theta}}{dt} = \frac{1}{\delta_{\theta,0}} \frac{d\delta_{\theta}}{d\tau} \quad (4.6)$$

in the self-similar period, $\dot{\delta}_{\theta,ss}$, is lower for higher M_c , table 4.4.

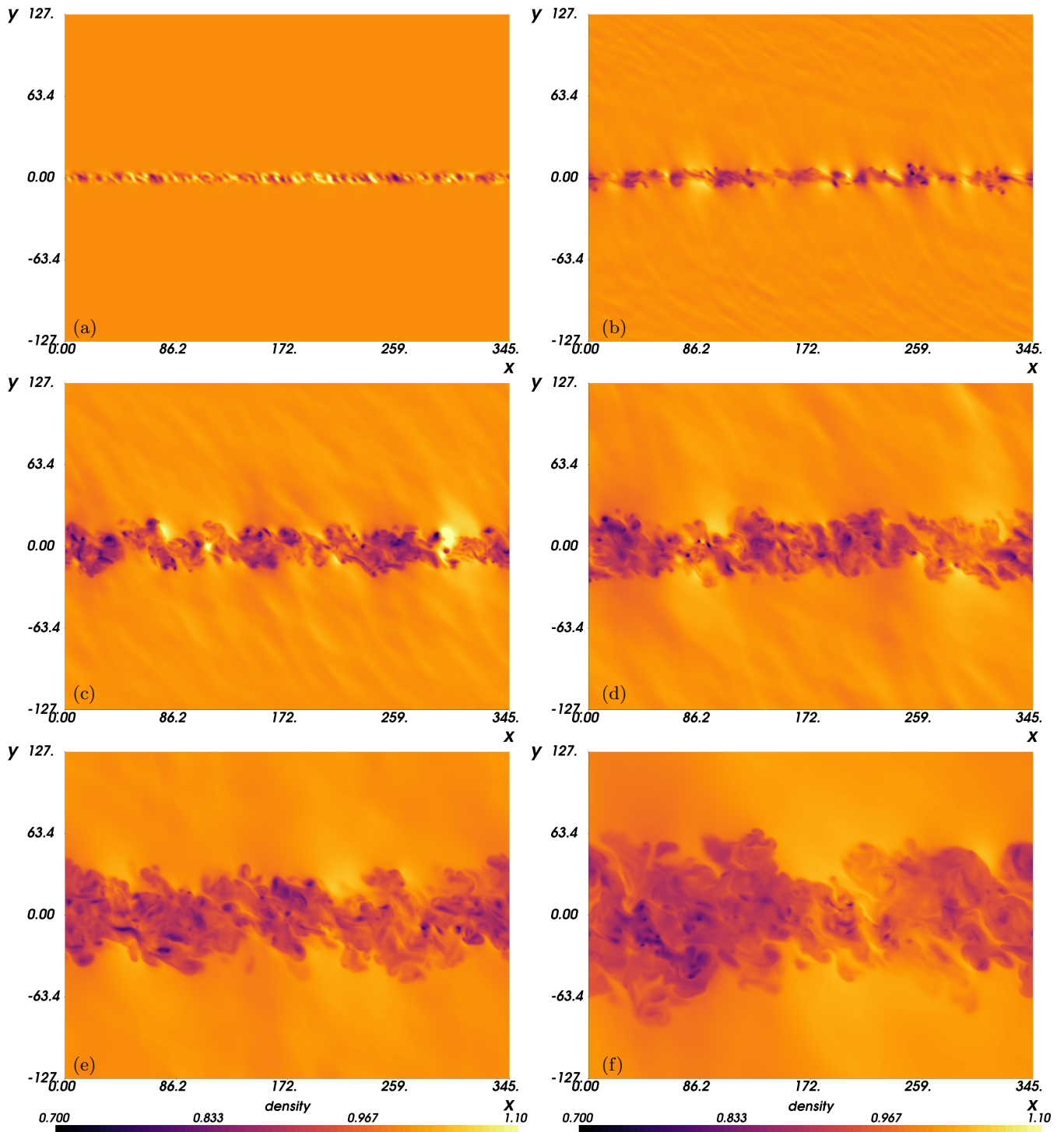


Figure 4.4: Density xy -planes of the $M_c = 0.7$ mixing layer configuration at different times: (a) $\tau = 4$, (b) $\tau = 200$, (c) $\tau = 400$, (d) $\tau = 600$, (e) $\tau = 1000$, (f) $\tau = 1800$. Frames (c)-(e) roughly fall within the self-similar regime of the mixing layer $\tau = 500 - 960$. The figure is limited to the transversal core region of the domain, see table 4.1.

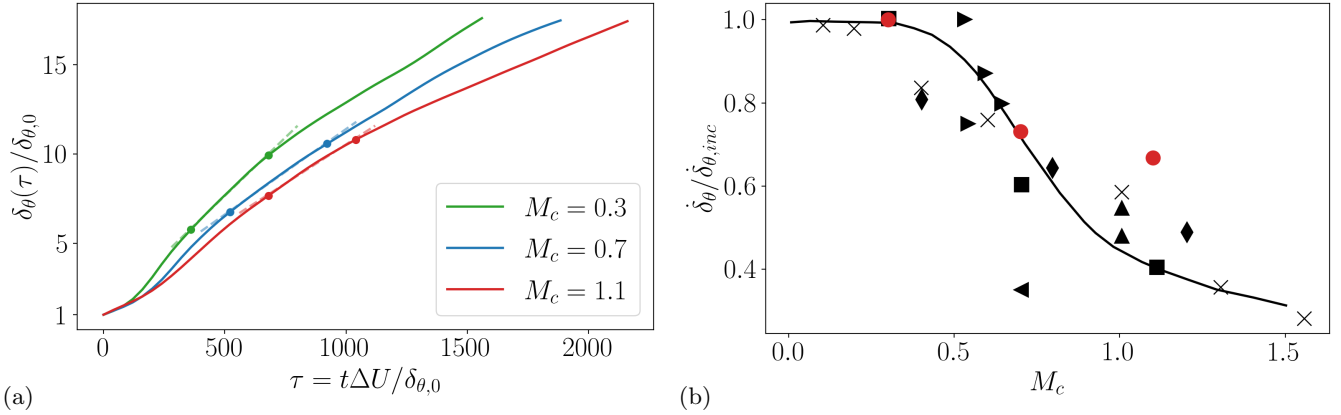


Figure 4.5: (a) Temporal evolution of the momentum thickness for different convective Mach numbers M_c . Markers indicate evolution interval used for the linear regression in the self-similar period. (b) Mixing layer growth rate in self-similar period for present simulations (●) and references: ‘Langley curve’ by Birch & Eggers (1972) (—), Barre *et al.* (1996) (▲), Freund *et al.* (2000) (×), Bonnet *et al.* (1993) (▶) Fu & Li (2006) (◆), Pantano & Sarkar (2002) (■), Zhou *et al.* (2012)(◄).

Following Pantano & Sarkar (2002), the lower convective Mach number case, $M_c = 0.3$, is taken as quasi-incompressible in fig. 4.5b to determine the influence of the Mach number on the mixing layer growth rate in the self-similar regime. For the $M_c = 0.3$ case, the present DNS has a growth rate of $\dot{\delta}_{\theta,ss} = 0.01308$, while Pantano & Sarkar (2002) report a slightly higher value of 0.016. While a good agreement with the reference data is found for $M_c = 0.7$, the decrease in $\dot{\delta}_{\theta,ss}$ relative to the quasi-incompressible growth rate is less pronounced in the present DNS compared to other simulations. Better agreement could be reached by slightly modifying the fitting-ranges of the $M_c = 0.3$ and $M_c = 1.1$ cases. However, the time periods leading to the highest amount of self-similarity in the velocity and Reynolds stress profiles, discussed in the following, were chosen as the periods of self-similarity.

The streamwise velocity profiles averaged over homogeneous flow directions are considered in fig. 4.6. The velocity is normalized by the free-stream velocity difference and cross-stream coordinate is scaled by the momentum or vorticity thickness of the time the profile is extracted from. The individual profiles taken from a collection of time instants throughout the self-similar regime show indeed a high degree of collapse.

A collapse of Reynolds stress profiles $R_{ij} = \widetilde{u_i''u_j''} = \overline{u_i u_j} - \tilde{u}_i \tilde{u}_j$ is considered to be a more sensitive evidence for self-similarity than a collapse of mean velocity profiles (Rogers & Moser, 1994). Therefore instantaneous cross-stream profiles of the streamwise turbulence intensity are examined for self-similarity in fig.4.7. Instead of individual, ensemble mean profiles of the streamwise, cross-stream and spanwise turbulence intensities and Reynolds shear stress profile are examined in fig. 4.8. The present DNS shows good agreement with previous simulations and experiments. Peak turbulence intensities are found in decreasing order along the streamwise, spanwise, and cross-stream directions, respectively, in agreement with Pantano & Sarkar (2002).

4.4 Identification, extraction and general statistics of vortical structures

To investigate the temporal mixing layer configuration, described in §4.2.1, from a structural viewpoint, the present analysis focuses on the inhibiting effect of compressibility, expressed as a set of convective Mach numbers $M_c = \{0.3, 0.7, 1.1\}$, on the evolution and geometrical properties of vortical structures in the mixing layer. As the vortex identification method, the compressible version of the Q-criterion originally introduced by Hunt *et al.* (1988) is utilized. Like the Δ -criterion by Dallmann (1983); Vollmers *et al.* (1983); Chong *et al.* (1990), λ_2 -criterion by Jeong & Hussain (1995) or the λ_{ci} -swirl-strength criterion by Berdahl & Thompson (1993); Zhou *et al.* (1999), the Q-criterion identifies vortices based on the velocity gradient tensor $\mathbf{A}_{ij} = \partial V_i / \partial x_j$. Comprehensive reviews of vortex identification methods can be found in Epps (2017) and Zhang *et al.* (2018). To summarize the motivation of the Q-criterion, the characteristic polynomial of the velocity gradient tensor $\det(\mathbf{A}_{ij} - \lambda \mathbf{I}) = 0$, which can be formulated as

$$\lambda^3 + \mathbf{P}\lambda^2 + \mathbf{Q}\lambda + \mathbf{R} = 0 \quad (4.7)$$

is considered. Here \mathbf{I} is the identity matrix and λ are the eigenvalues of the velocity gradient tensor. The tensor's invariants P, Q and R can be expressed as (Suman & Girimaji, 2010; Epps, 2017)

$$\mathbf{P} = -tr[\mathbf{A}] = -\mathbf{S}_{ii} \quad (4.8)$$

$$\mathbf{Q} = \frac{1}{2} \left(\mathbf{P}^2 - tr[\mathbf{A}^2] \right) = \frac{1}{2} \left(\mathbf{P}^2 - \mathbf{S}_{ij}\mathbf{S}_{ji} - \mathbf{\Omega}_{ij}\mathbf{\Omega}_{ji} \right) \quad (4.9)$$

$$\mathbf{R} = \frac{1}{3} \left(-\mathbf{P}^3 + 3\mathbf{P}\mathbf{Q} - tr[\mathbf{A}^3] \right) = \frac{1}{3} \left(-\mathbf{P}^3 + 3\mathbf{P}\mathbf{Q} - \mathbf{S}_{ij}\mathbf{S}_{jk}\mathbf{S}_{ki} - 3\mathbf{\Omega}_{ij}\mathbf{\Omega}_{jk}\mathbf{S}_{ki} \right) \quad (4.10)$$

where $\mathbf{\Omega}_{ij} = 1/2(\mathbf{A}_{ij} - \mathbf{A}_{ji})$ is the rotation-rate tensor and $\mathbf{S}_{ij} = 1/2(\mathbf{A}_{ij} + \mathbf{A}_{ji})$ is the strain-rate tensor. These two tensorial quantities are defined as the antisymmetric and symmetric part of the velocity gradient tensor \mathbf{A}_{ij} . The Q-criterion defines a vortex as a region where $Q > Q_{\text{iso}}$, where nominally $Q_{\text{iso}} > 0$. In incompressible flow, this condition is satisfied where the vorticity magnitude is larger than magnitude of rate of strain.

The Q-structures of interest are obtained by iso-contouring the Q scalar field. Due to the threshold-based character of the Q-criterion, the choice of the iso-contour value Q_{iso} is crucial for the shape, spatial extent and number of Q-structures extracted from the flow field at an instant. A low iso-value generates large, widely ramified structures not suitable for the tracking of individual structures, while a high iso-value shrinks the extracted Q-structures reducing their complexity and lifetime. To account for the temporally evolving character of the flow field, a time-dependent iso-value is chosen: From the mean, $\langle Q \rangle$, and standard deviation, $\sqrt{\langle Q^2 \rangle - \langle Q \rangle^2}$, of the scalar Q field of each frame the iso-value used for extraction is formed as $Q_{\text{iso}} = \langle Q \rangle + n\sqrt{\langle Q^2 \rangle - \langle Q \rangle^2}$. Here, the standard deviation is found to be the dominant contribution to Q_{iso} . For the lower convective Mach number case ($M_c = 0.3$), a choice of $n = 6.0$ was found to extract a satisfactory number of individual Q structures throughout the simulation and mainly structures of tube-like shape in the self-similar regime and later stages of the simulation. Due to the inhibiting effect of compressibility on the turbulence intensities in the mixing layer, the lower Mach number case is

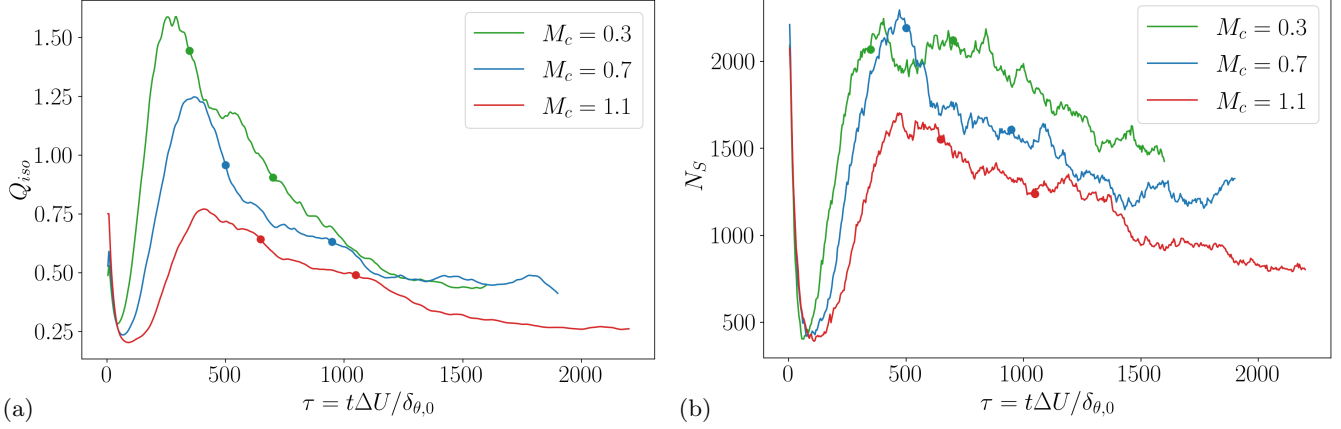


Figure 4.9: (a) Evolution of the iso-contour value Q_{iso} used to extract individual Q structures from the mixing layers of varying convective Mach number M_c . (b) Number of extracted structures based on the iso-value evolution. The yellow marker indicate the begin and end of the self-similar regimes of the mixing layers.

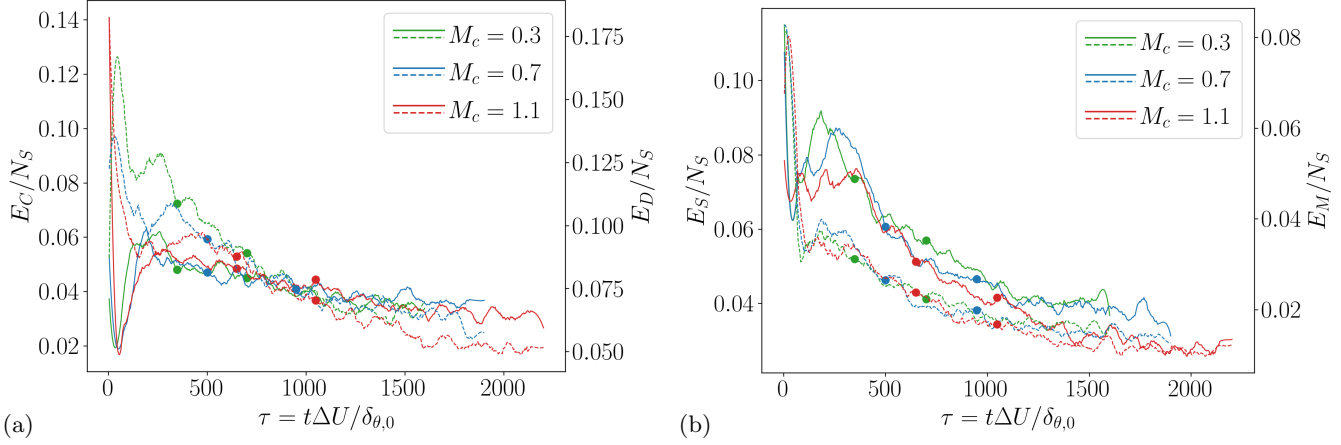


Figure 4.10: Smoothed number of detected events of different types over time: (a) Creations (solid), disappearances (dashed), (b) splits (solid) and merger (dashed) of any order.

the most prone to formation of large, ramified Q structures. Therefore, a lower choice for n would still avoid largely connected structures for the higher Mach number cases. However, for the sake of consistency and to limit the number of parameter influencing the study, the same choice of $n = 6.0$ is made for $M_c = 0.7, 1.1$. Note that, despite the same choice of n , at a given time instant the iso-value Q_{iso} tends to be lower for the higher Mach number, as the standard deviation of the Q field is lower, see fig. 4.9a. After an initial transient, which is longer for larger M_c , the iso-value reaches its maximum before the beginning of the self-similar regime for all considered Mach number M_c and decays afterwards. Indicative of the expression of turbulence inhibition on the structural level, a larger number of structures is extracted throughout the simulation for lower Mach number, fig. 4.9b. Similar to the iso-value evolution, a transient is present in the number of extracted structures and the temporal distribution of detected event types, fig. 4.10. Based on the applied iso-contouring and extraction method, the number of structures decays in the self-similar regime for $M_c = \{0.7, 1.1\}$ and only in the $M_c = 0.3$ case the number of extracted structures

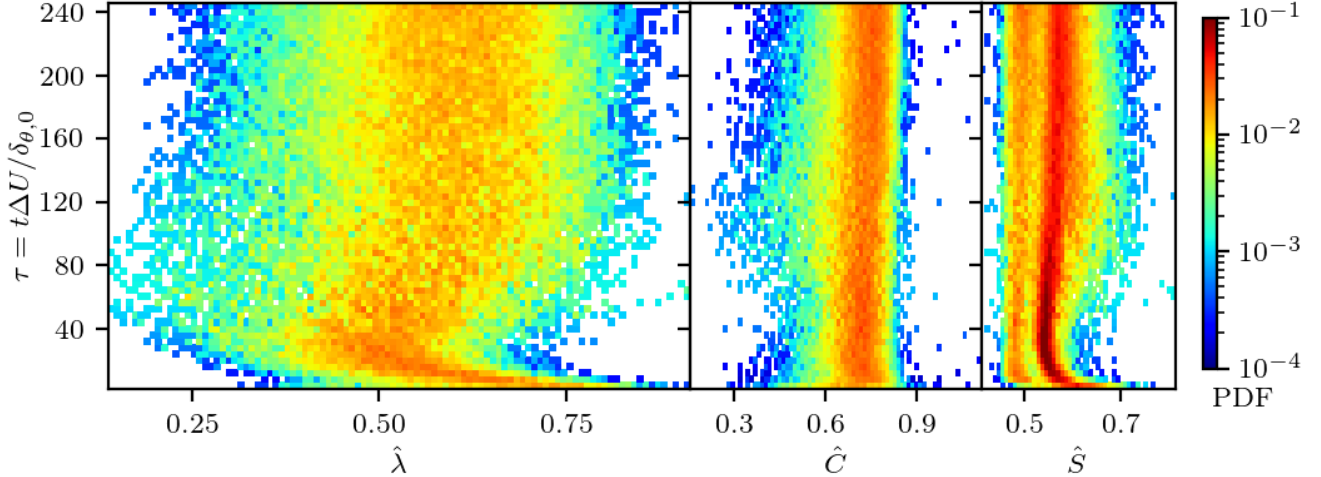


Figure 4.11: Temporal evolution of the geometrical signature of all structures extracted during the early, pre-self-similar stage ($\tau = 0 - 240$) of the $M_c = 0.3$ mixing layer configuration. Each sample is weighted by the inverse of the total number of structures of the frame from which the sample is extracted.

remains constant throughout the self-similar regime of the mixing layer. The peak event activity coincides with the peak in extracted structures (and iso-value) and is ordered by Mach number. The number of merger exceeds the number of split events only in the very early stage of the simulation.

The initialized velocity fluctuations, see §4.2.1, generate small, rather compact ($\hat{\lambda} > 0.6$) Q-structures in the very first frames of the simulations. While these structures grow and merge with others, they become more stretched shortly after initialization ($\tau < 40$ for $M_c = 0.3$), see fig. 4.11. The continued merging process ($\tau = 40 - 120$ for $M_c = 0.3$) leads to the formation of large structures characterized by a low value of compactness and more elliptical than hyperbolic surface points ($\hat{S} > 0.5$). These largely connected (ramified) structures, visualized in fig. 4.12, are reminiscent of Λ -vortices observed near the inlet of spatially developing mixing layers (Zhou *et al.*, 2012). As the structures span across the center plane of the mixing layer, they split into smaller, less ramified structures shortly after, when parts of the structures are pulled into the oppositely directed free streams.

Another indicator of the inhibiting effect of compressibility on the turbulent dynamics of the mixing layer can be seen in the statistics of the (anti-)alignment between the normal vector of the Q-isosurfaces and the vorticity vector, fig. 4.13(a), as it takes longer for the vorticity to become approximately tangential to the surfaces after initialization.

Consistent with the Reynolds stress profiles, see figure 4.7 and 4.8, the majority of new vortices is created near the center plane of the mixing layer, fig. 4.13(b), where the cross-stream location of a structure is defined as the mean of the maximal and minimal cross-stream coordinate of the structure's surface points. Less new vortices are created per frame of the self-similar regime with increasing convective Mach number M_c .

The geometrical variety of the Q-structures extracted during the self-similar regimes of the mixing layers is shown in fig. 4.14. The majority of the structures is characterized by a tube-like shape of decreasing curvedness with increasing convective Mach number M_c . The tube-like structures show a large variation in stretching, which decreases with decreasing curvedness.

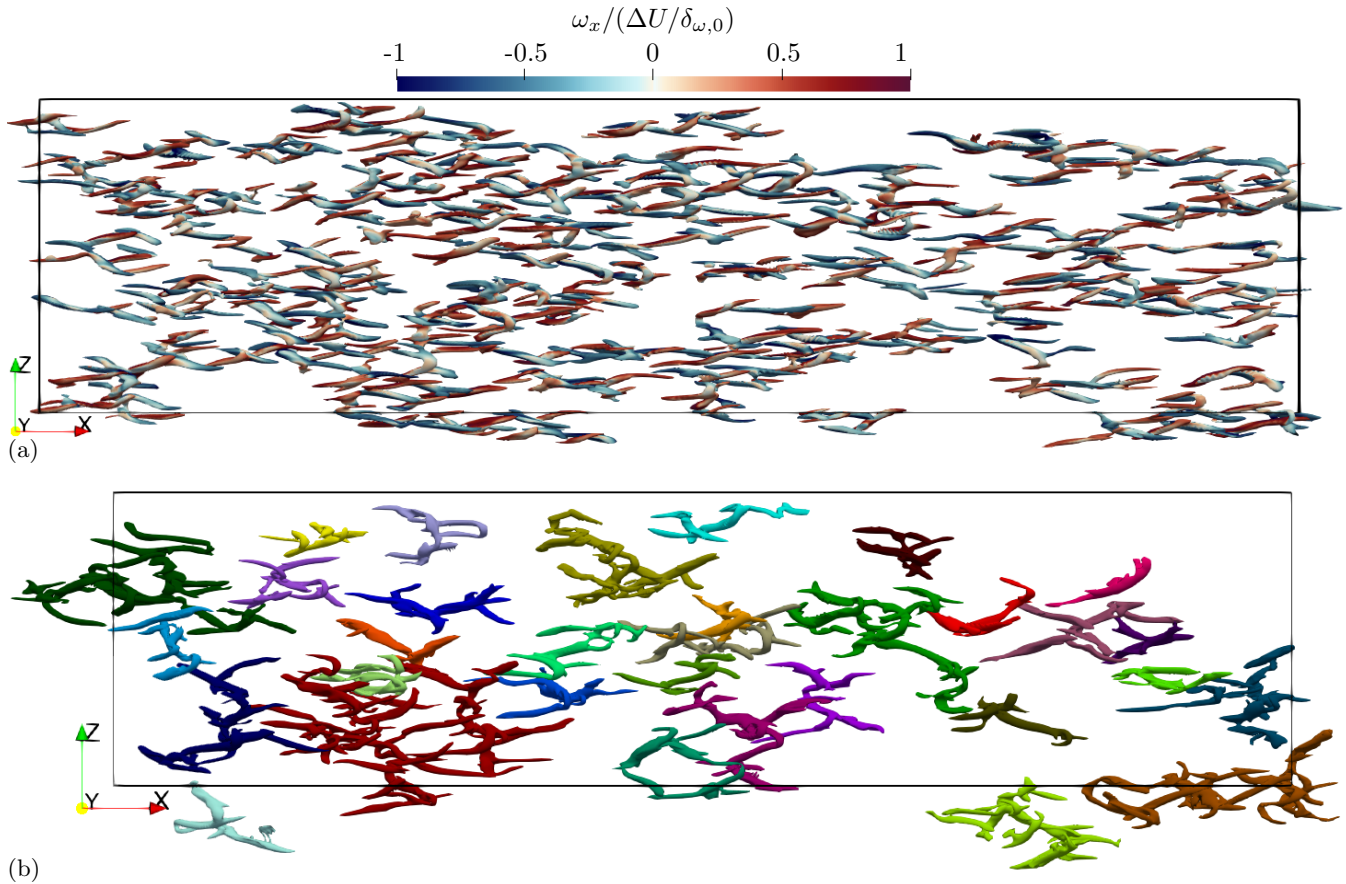


Figure 4.12: Selection of largely connected vortical structures extracted at $\tau = 40$ (a) and $\tau = 80$ (b) from the $M_c = 0.3$ mixing layer configuration colored by streamwise vorticity and individual structure color, respectively. Periodic reconnection of domain boundary intersecting structures is applied in streamwise and spanwise direction. The highly stretched structures span across the center plane of the mixing layer. The shape of the vortices is reminiscent of Λ -vortices observed near the inlet of spatially developing mixing layers (Zhou *et al.*, 2012).

4.5 Creations neighboring existing vortices

Newly created Q-structures in the self-similar regime of the $M_c = 0.3$ mixing layer are geometrically characterized as generally of tube-like shape with a relatively higher compactness $\hat{\lambda}$, fig. 4.15. A large portion of the newly created structures disappears below the iso-value Q_{iso} used for surface extraction in the particular frame within a few tracking time steps and without any interaction with other structures. Contrary to these short-lived, isolated structures, other structures participate in structure interactions shortly after their creations. Therefore, this section focuses on the first event (other than continuation and disappearance) in the lifetime of newly created structures. Compound events rarely occur as such (at least in the self-similar regime) and created structures mainly have merge or split events as their first structure interaction. Presented in fig. 4.15, created structures that are the source of a split in their first event typically live longer than merging newly created structures and become more stretched tube-like before the split occurs.

If the first event in the lifetime of a newly created structure is a merge, the structure either merges into another structure (i.e., the newly created structure is a non-dominant source of the merge) or the new structure continues

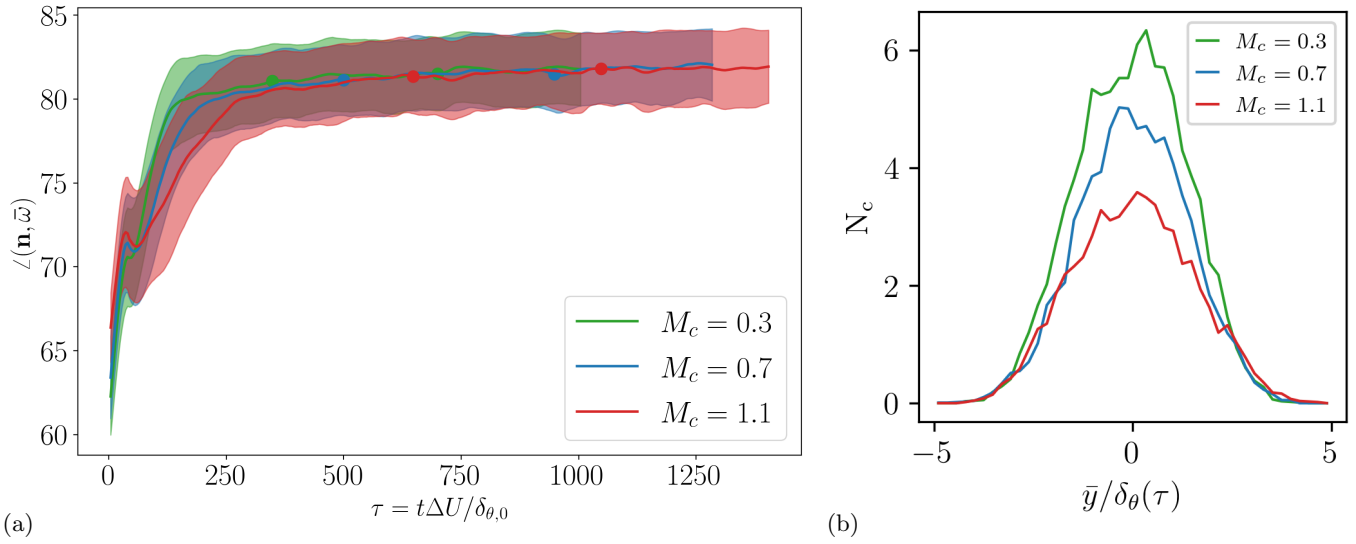


Figure 4.13: (a) Temporal evolution of the ensemble mean of of the surface-averaged angle between the surface normal and the vorticity on the surface. Filled area encloses two standard deviations. After an initial transient, which takes longer for increasing M_c , the vorticity becomes approximately tangential to the Q-isosurfaces. (b) Distribution of cross-stream locations of creation events during the self-similar periods of the mixing layers. Counts normalized by number of frames in the self-similar regime.

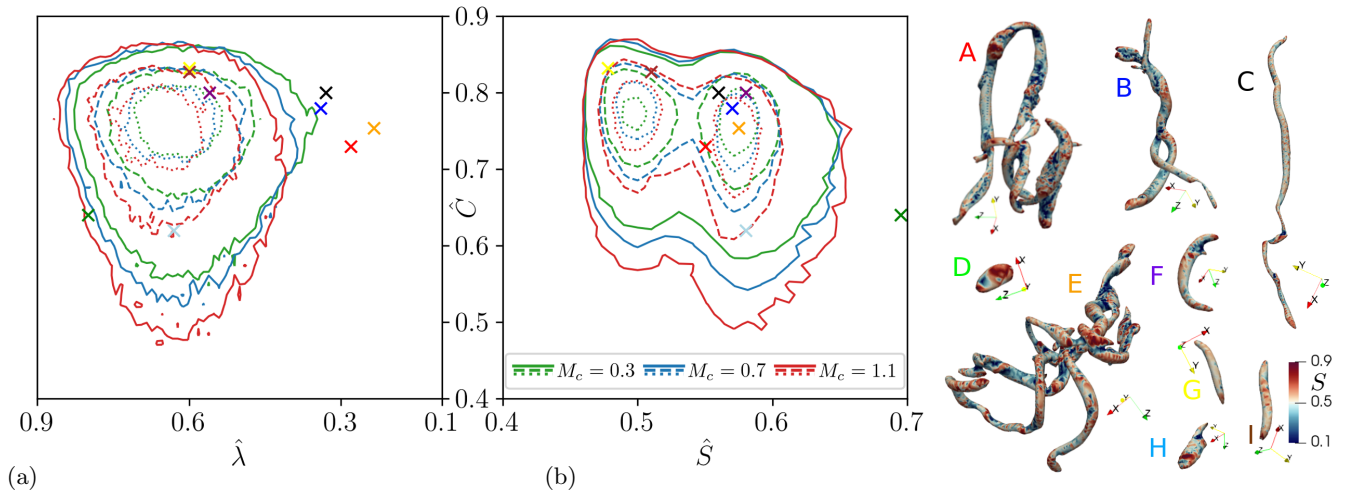


Figure 4.14: (a) Geometrical variety of all extracted Q-structures with more than 100 surface points in the self-similar regimes of the mixing layers for different convective Mach numbers. Contour lines of the PDF indicate 0.1 (solid), 0.4 (dashed) and 0.7 (dotted) of the maximum count. (b) Exemplary structures extracted from the $M_c = 0.3$ at $\tau = 344$ case corresponding to the letters in the feature space plot. Structures colored by absolute shape index S . Structure renderings are not to scale.

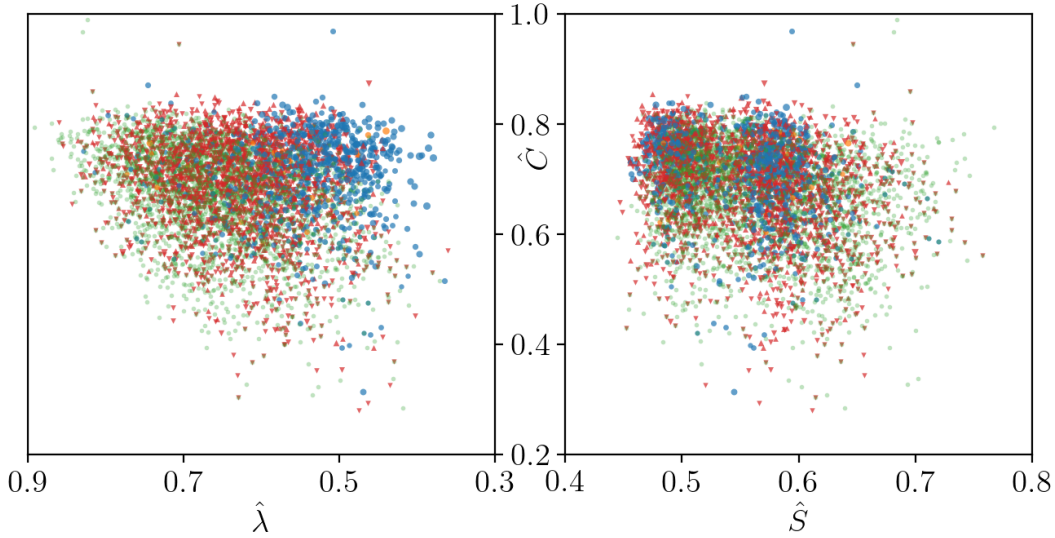


Figure 4.15: Newly created structures during the self-similar regime of the $M_c = 0.3$ mixing layer and their geometric signature while participating as a source in the first merge (red), split (blue) or compound (orange) event after their creation. Newly created structures which do not have any structure interaction during their (short-lived) existence are excluded from the figure. The size of the event source marker is indicative of the lifespan from the creation to the first event. For merge events, the \blacktriangledown shows a merge into another existing structure, while a \blacktriangle marker is used if another structure merges into the newly created structure.

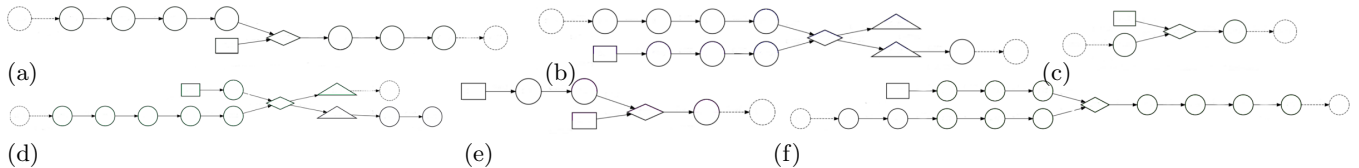


Figure 4.16: Examples of patterns of a newly created structure merging with an existing structure extracted from the self-similar regime of the $M_c = 0.3$ mixing layer. Number of instances of each pattern: a) 50, b) 4, c) 122, d) 3, e) 12, f) 11. Dashed nodes do not belong to the pattern instance itself but to its embedding of the entire tracking graph. Node shapes signal event types following fig. 2.7

its life after another structure merged into it. While both scenarios occur, the first one is found to be (much) more frequent. In this case, the structure is created in close proximity to an existing structure and merges into such shortly after its creation.

To further support these findings based on fig. 4.15, the graph mining method, presented in §2.10 is applied. Due to the size of the tracking graph, this graph mining effort is conducted only on the graph representing all structures evolutions in the self-similar regime of the $M_c = 0.3$ mixing layer rather than searching the graph of the entire data set for frequent subgraphs. While the graph miner detects many split-dominated patterns and also the previously mentioned short-lived structure evolution only consisting of a creation, a sequence of continuation varying in length and a disappearance, this section focuses on found patterns involving creation and merge events. Examples of detected creation-merge patterns and their statistical significance are shown in fig. 4.16.

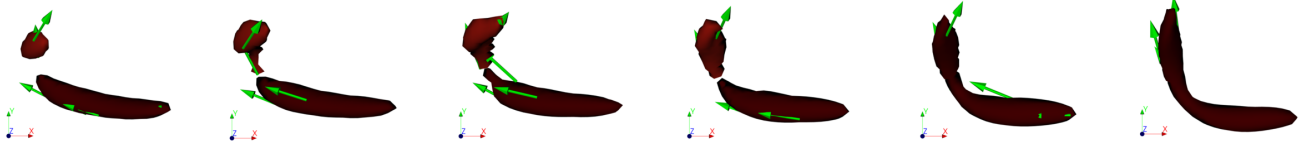


Figure 4.17: Creation of a new vortex of rather compact shape in the neighborhood of an existing straight, tube-like vortical structure. The merging of both structures yields a tube-like structure of bent shape. Temporal order from left to right. Green arrows indicate vorticity on the structures colored by tracking ID.

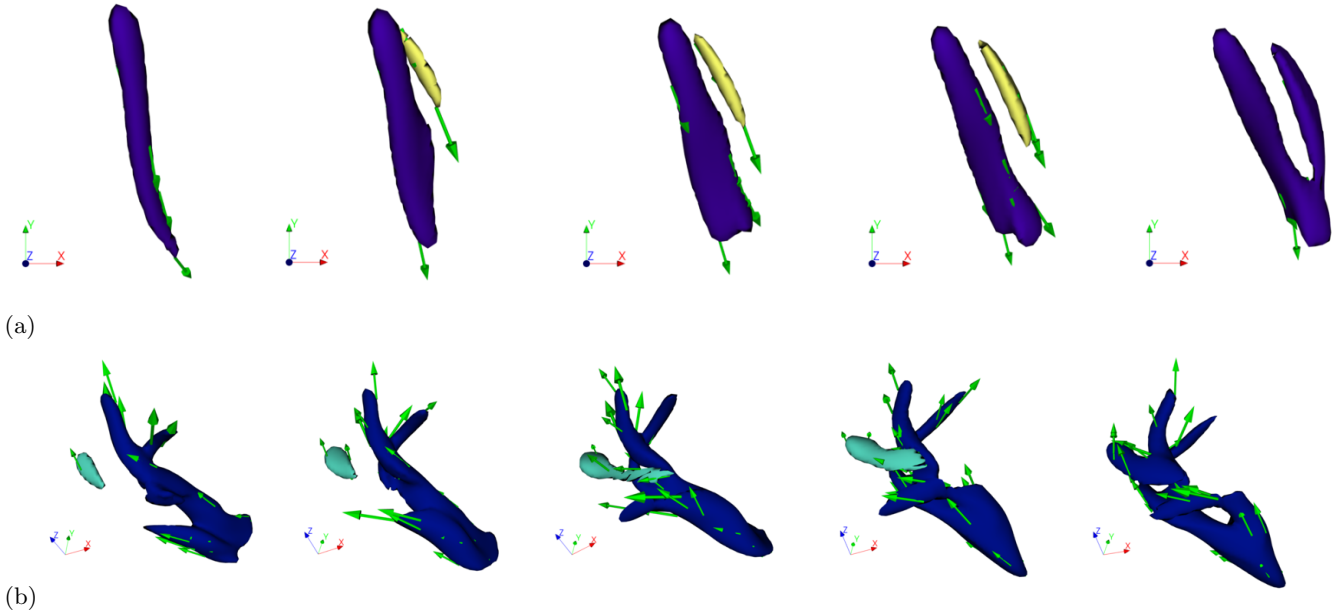


Figure 4.18: The incoming (newly created) structure often develops an additional “finger” on the existing structures. These addition are often found to be twisted as in (b)

Consistent with fig. 4.15, the patterns indeed exhibit merging structures shortly after their creation. Depending on the relative position and orientation of the merging structures, the interaction of an existing straight tube-like structure with a newly created structure can yield a bent shape of the resulting structure, fig. 4.17.

The existing, dominant structure often develops a additional branch (“finger”) in the course of merging with the incoming (newly created) structure. During the merge process the incoming, merging structure undergoes a stretching, as seen in fig. 4.18(b). The new branch of the merge target structure can be of straight or rather twisted shape depending on the stretching during the merging process.

4.6 Detection of hairpin-like vortical structures

Considering the geometrial variety of the extracted vortical structures, see fig. 4.14, the conceptual model of a hairpin vortex (Theodorsen, 1952, 1955; Robinson, 1991) is proposing a vortical structure of distinctive shape. While observed in other shear flow configurations, such as homogeneous shear turbulence (Rogers & Moin, 1987; Vanderwel & Tavoularis, 2011; Dong *et al.*, 2017) and mixing layers (Brown & Roshko, 1974; Rogers & Moser, 1992; Moser

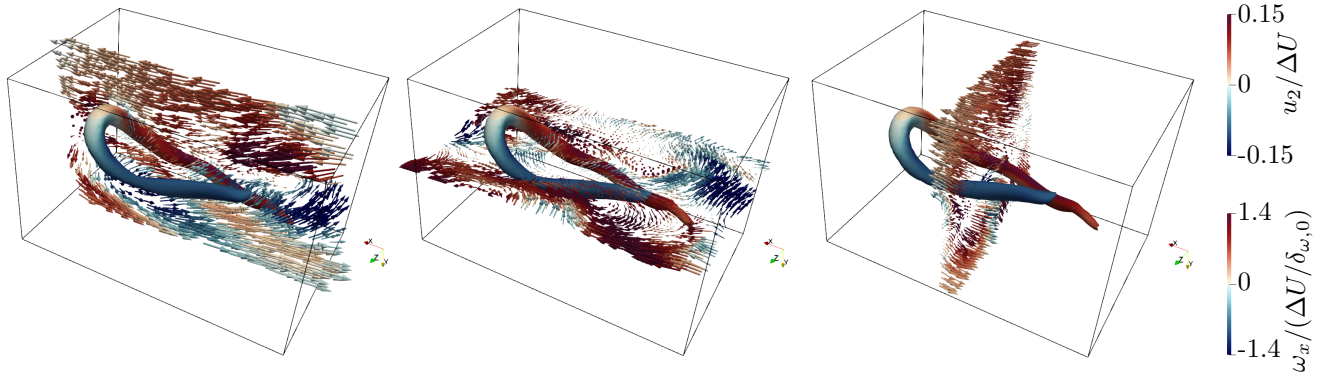


Figure 4.19: Plane cuts of the velocity field surrounding an individual hairpin extracted at $\tau = 144$ from the $M_c = 0.3$ mixing layer crossing its center plane. The velocity vector is colored by the cross-stream velocity component, while the streamwise vorticity component is mapped on the surface.

& Rogers, 1993; Zhang *et al.*, 2019; Zhou *et al.*, 2012), hairpin vortices are frequently reported in wall-bounded turbulence (Acarlar & Smith, 1987; Chacín *et al.*, 1996; Zhou *et al.*, 1999; Tomkins & Adrian, 2003; Adrian, 2007; Wu & Moin, 2009; Schlatter *et al.*, 2014; Eitel-Amor *et al.*, 2015). The characteristic elements of a hairpin vortex are the spanwise oriented head connecting a pair of quasi-streamwise, counterrotating vortices referred to as the legs of the hairpin (Adrian *et al.*, 2000). The growth of hairpin vortices is related to Townsend’s attached eddy hypothesis (Townsend, 1980), which states that the size of eddies are proportional to their distance from the wall. In wall turbulence the hairpin concept offers a mechanism to generate Reynolds shear stress, low-speed streaks and to transport vorticity of the mean shear at the wall outward (Adrian, 2007). Conceptually, the hairpin head undergoes a (small) upward (wall-normal) motion and lies further away from the wall as the leg pair. In agreement with observations of hairpins in boundary layers, the velocity field (perturbed by smaller-scale turbulence) surrounding an individual hairpin extracted from the $M_c = 0.3$ mixing layer displays an ejection in between the legs of the hairpin, see fig. 4.19. The inner part of the head of such individual hairpin is dominated by hyperbolic surface points, while the outer head part and the tips of the legs are elliptical in shape, fig. 4.20(a).

To investigate the temporal evolution of hairpin-like vortical structures in mixing layers, an algorithm to filter hairpin-like structures from all vortical structures extracted from a turbulent flow field is proposed in the following. The goal of the algorithm is to filter individual hairpin-like structures, whereas (depending on the iso-value used for iso-surfacing) largely connected hairpins, fig. 4.20(b), are extracted from the flow field but rejected by the filter. While there is no exact mathematical definition of a hairpin vortex in the literature, hairpin-like structures are filtered by imposing a sequence of geometrical and physical constraints on each extracted vortical structure. A set of input parameters required by the algorithm is listed in table 4.5 with parameter values used for all applications in this work. Processing one structure at a time, the algorithm starts with an application of spectral clustering to the surface points of the structure. To identify candidates for the head and leg pair of the potentially hairpin-like structure, this spatial clustering is conducted with a prescribed number of three clusters. No scaling operation is performed prior to the spectral clustering. Structures having a spatial clustering where any of the three clusters has less than a user-defined minimum size s_{\min} are rejected.

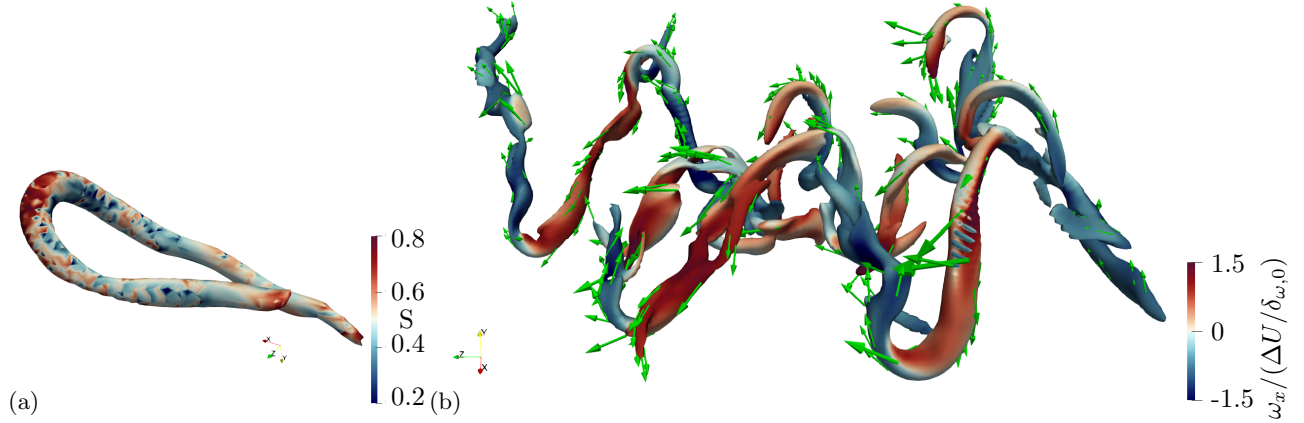


Figure 4.20: (a) Absolute shape index mapped on the surface of an individual hairpin vortex extracted at $\tau = 144$ from the $M_c = 0.3$ case. (b) Large, connected hairpin vortices extracted at $\tau = 128$ from the $M_c = 0.3$ case. Streamwise vorticity component mapped on surface, while green arrows represent the vorticity at sampled surface points. Renderings not to scale.

Required		Optional			
s_{\min}	α_{leg}	ε	σ	α_{dip}	d_{dip}
250	$\pi/4$	0.5	150	0.05	10

Table 4.5: Input parameter of the hairpin filter algorithm: s_{\min} minimum size of spatial clusters, α_{leg} maximum angle between the leg pair, ε maximum distance between two samples to be considered in the neighborhood of each other in the leg vorticity - cross coordinate clustering, σ minimum cluster size in the leg vorticity - cross coordinate clustering. Parameter used for the optional bimodality test of the leg cross coordinate distribution: α_{dip} significance threshold and d_{dip} interval merge distance of the UniDip algorithm.

The three spatial clusters are assigned as a head candidate and two leg candidates based on the number of openings each surface cluster has: The head candidate has two openings, whereas each leg has one opening. If the clustering leads to any other combination of openings, the structure is rejected. Next, the OBB tree of each surface cluster is obtained, where the tree depth is one for a leg and two for the head to better capture its bent shape, see fig. 4.21. The two OBB leaf nodes of each leg are used to define the direction of the leg as the line connecting the centers of the leaf nodes and pointing away from the root node of the head candidate. The angle between the leg directions is computed and compared against a user-provided maximum angle α_{leg} . It is observed that a local coordinate system $\{\hat{x}_1^{\text{OBB}}, \hat{x}_2^{\text{OBB}}, \hat{x}_3^{\text{OBB}}\}$ based on the major, medium and minor direction of the OBB of the entire surface may lead to a poor capturing of the direction of the legs if the structure is rather asymmetric due to unevenly long legs for example, see fig. 4.21 (a). To correct this tilt a local coordinate system is constructed as follows: From the two leg directions a mean leg direction \vec{l} is obtained and projected onto the major-medium plane of the structure OBB as $\vec{l}_{\text{proj}} = \vec{l} - (\vec{l} \cdot \hat{x}_3^{\text{OBB}})\hat{x}_3^{\text{OBB}}$. A orthonormal basis $\{\tilde{x}_1, \tilde{x}_2, \tilde{x}_3\}$ is formed using the unit projected mean leg direction $\tilde{x}_1 = \hat{l}_{\text{proj}}$, the minor direction \hat{x}_3^{OBB} of the structure OBB and their cross product \tilde{x}_3 . Using this local coordinate system it is required for the head candidate to have at least one leaf node in between the legs in terms of the \tilde{x}_3 coordinate.

A number of extracted structures approximate a hairpin-like shape, but their leg pairs do not exhibit counter-rotating character which is essential for hairpin vortices, fig. 4.22. Therefore, simple statistical constraints are imposed on the distribution of the \tilde{x}_1 component of the vorticity on the candidate surface: (i) The mean of the distribution on the surface cluster is required to have opposite signs for the two legs due to the counter-rotation. (ii) The absolute value the mean is expected larger than the standard deviation. To include tolerance, the coefficient of variation is required to be smaller than two. (iii) Since the vorticity changes orientation along the head, the absolute value of the mean on the head cluster is expected smaller than on both legs.

A hairpin with straight, approximately parallel legs exhibits a bimodal distribution of \tilde{x}_3 surface point coordinates of the leg pair, i.e., the distribution excludes the head cluster, fig. 4.21 (a). To detect such bimodality the probability density function is obtained from the \tilde{x}_3 coordinates of the legs using a kernel density estimation (KDE) where the bandwidth is optimized by utilizing a cross-validation approach. After adding noise to this estimated PDF, random samples are obtained from the distribution and fed into the UniDip algorithm (Maurus & Plant, 2016) to determine the modality of the distribution.

However, hairpins are often distorted by surrounding eddies. For example, if a leg is bent inwards, the distribution of the \tilde{x}_3 coordinates of the legs may not be bimodal. To detect those more asymmetric hairpins the bimodality is neither a necessary nor a sufficient criterion. It is kept in the algorithm to optionally filter the subset of hairpins that satisfy this criterion, but not applied to obtain the results in the present work. Complementing the \tilde{x}_3 coordinates, the \tilde{x}_1 component of vorticity, $\tilde{\omega}_1$, is considered as another feature of the leg pair. The z-scores of those two features of the leg pair are obtained independently and the standardized features are clustered by applying the DBSCAN algorithm (Ester *et al.*, 1996; Schubert *et al.*, 2017). In contrast to the earlier used spectral clustering, the number of clusters is not prescribed but estimated by the DBSCAN algorithm. Requiring a certain amount of compactness in \tilde{x}_3 and $\tilde{\omega}_1$, it is expected that each leg contributes one cluster to the data, i.e., the DBSCAN algorithm estimates two clusters in total. This approach is robust against smaller eddies which merged into the hairpin impairing not only its shape (\tilde{x}_3) but also its vorticity distribution due to the noise detection of the DBSCAN algorithm. Fig. 4.23 illustrates different types of rejections of vortical structures which passed all other prior constraints.

As mentioned earlier, hairpins are often lopsided and the (quasi-streamwise) legs are only rarely counter-rotating pairs of equal strength leading to asymmetry (Robinson, 1991). The proposed hairpin filter detects those imperfect candidates, referred to as hooks and arches in fig. 4.24, as hairpin-like. The number of detections of these imperfect hairpins can be greatly reduced by requiring the candidate head surface cluster to have an OBB root node center which has the utmost (smallest or largest) major coordinate \hat{x}_1 of the entire surface OBB compared to the OBB root node centers of the other two surface clusters.

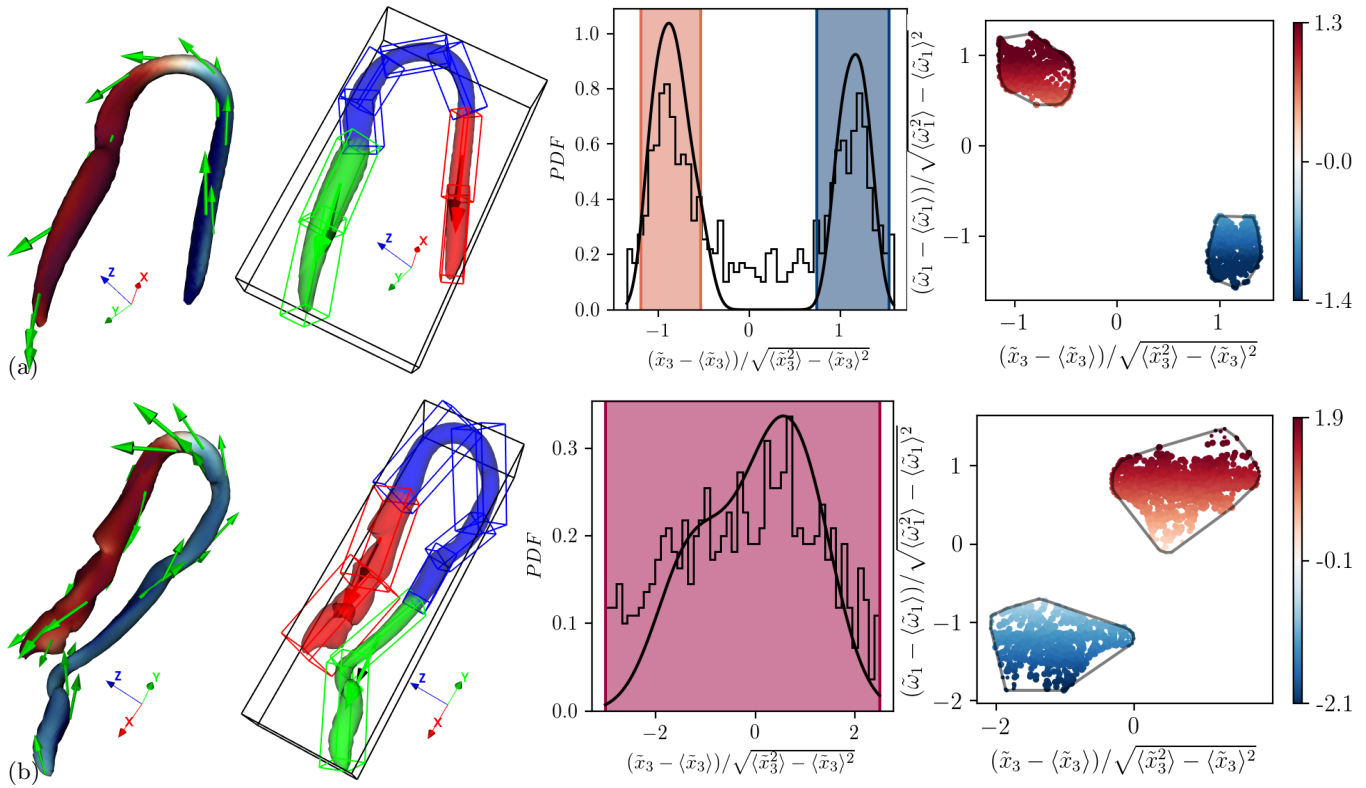


Figure 4.21: Elements of the hairpin detection method from left to right: Leg component of the vorticity mapped on the extracted surface. Green arrows represent the vorticity at sampled surface points. The three clusters resulting from the spatial clustering of the surface are indicated in red, green and blue. The OBB trees of each surface cluster are colored according to the cluster they enclose. A tree depth of one is used for the legs, while the depth is two for the head cluster. Red and green arrows show the direction of each leg, while black arrows are the mean leg direction projected onto the major-medium plane of the global OBB of the surface (black cuboid). Modes in the distribution of the cross coordinates of the legs are highlighted as density intervals. Probability density function obtained from the KDE of the leg cross coordinates shown in black. Histogram obtained from the samples fed into the UniDip algorithm with added noise. Convex hulls enclose the clusters obtained from the clustering of the leg component vorticity - cross coordinate clustering of the legs. Cluster core points plotted with increased point size. Scatter points colored by the leg component of the vorticity. Range of color bar based on leg vorticity component of entire surface to also apply to the surface mapping (left). (a) showing a bimodal distribution of cross coordinates of the legs, while a unimodal distribution was detected for (b). Structures extracted from the $M_c = 0.3$ case at $\tau = 148$ and $\tau = 232$, respectively.

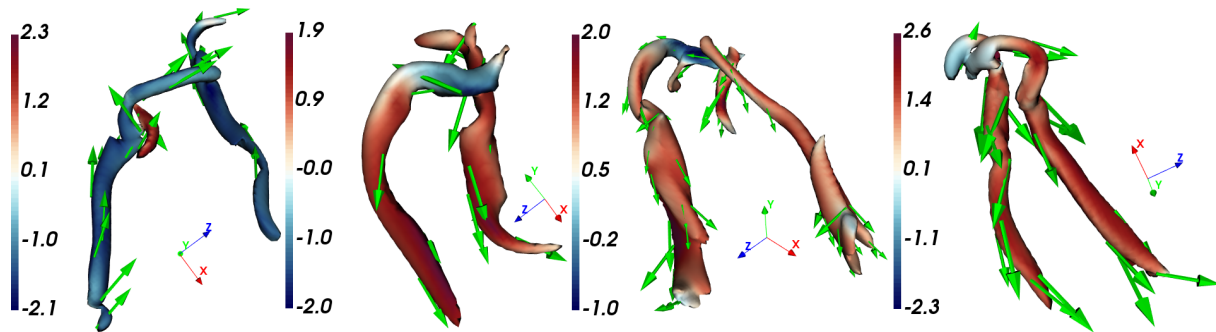


Figure 4.22: Representative structures approximating a hairpin-like structure in shape, while differing in vorticity distribution on their surface from a hairpin vortex. Leg component of the vorticity mapped on the surfaces, while green arrows indicate vorticity. The structures are detected as non-hairpin-like based on their physical dissimilarity to a hairpin vortex.

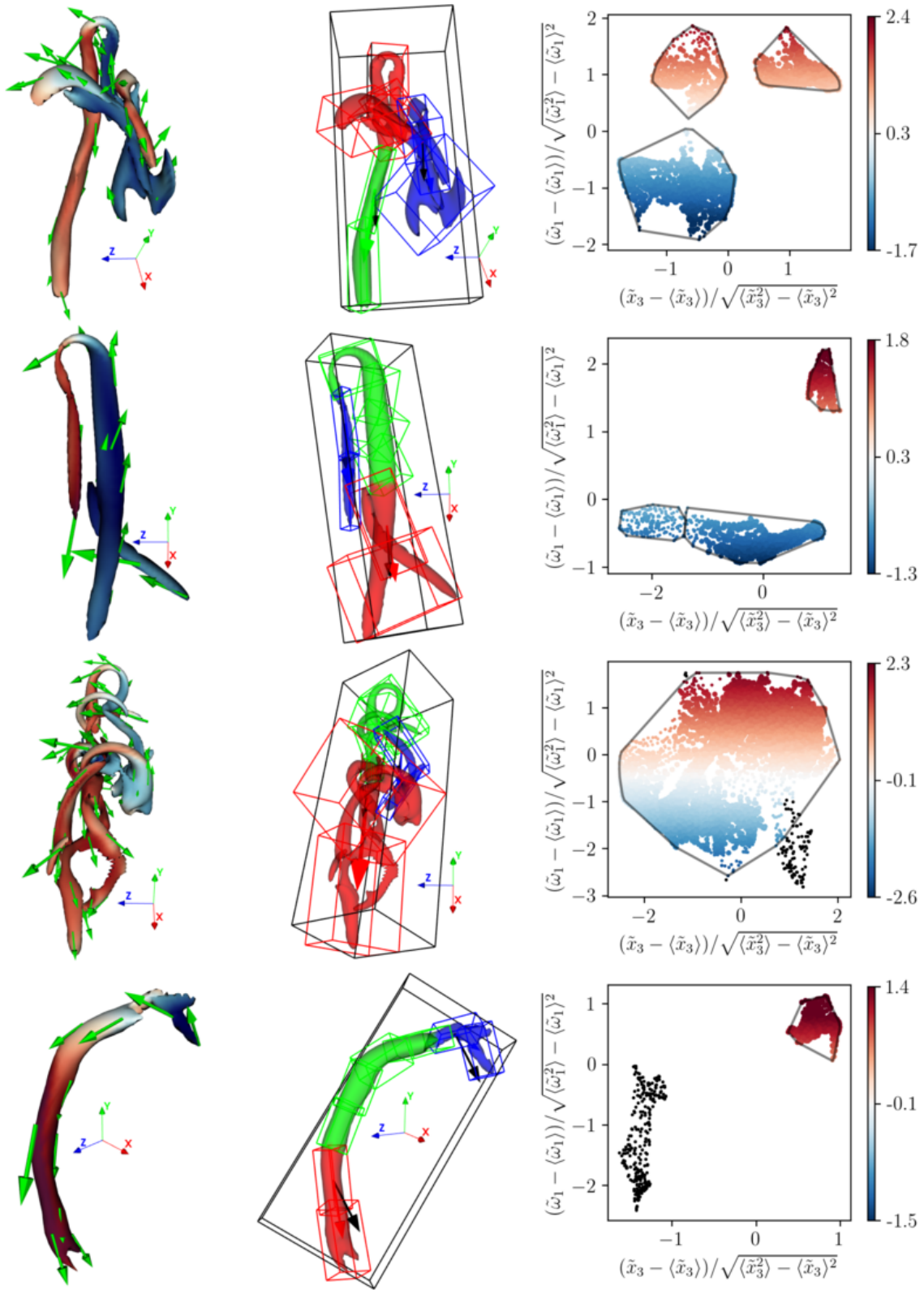


Figure 4.23: Representation of different types of rejections in the leg vorticity component - cross coordinate clustering step of the hairpin detection method. Convex hulls enclose detected clusters. Core and non-core cluster members are colored by the leg vorticity component, where non-core members are reduced in size. Noise points indicated in black.

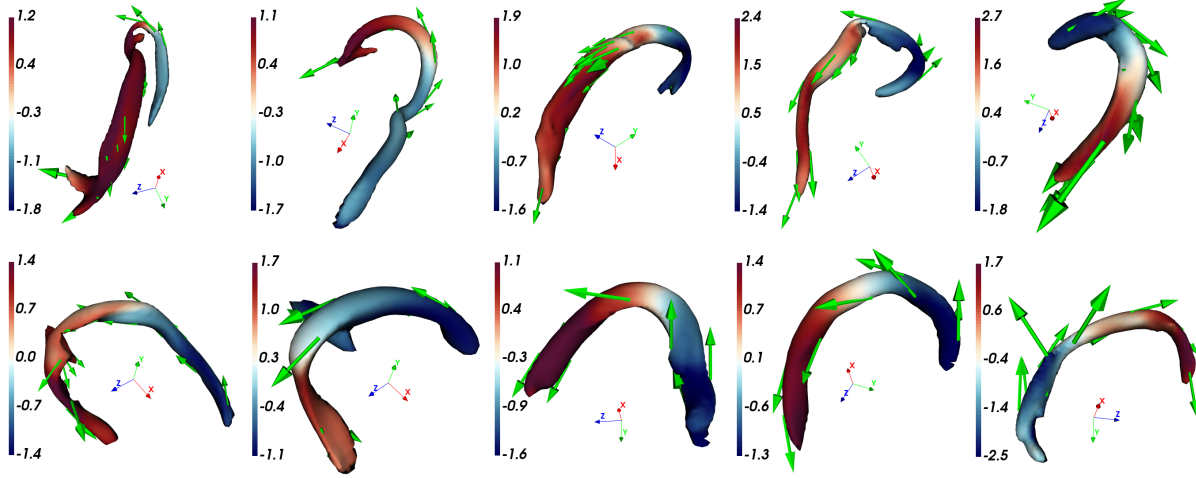


Figure 4.24: Examples of hooks (top row) and arches (bottom row) detected as hairpin-like in the pre-self-similar stage of the $M_c = 0.3$ mixing layer case. Other plot elements as in fig. 4.22.

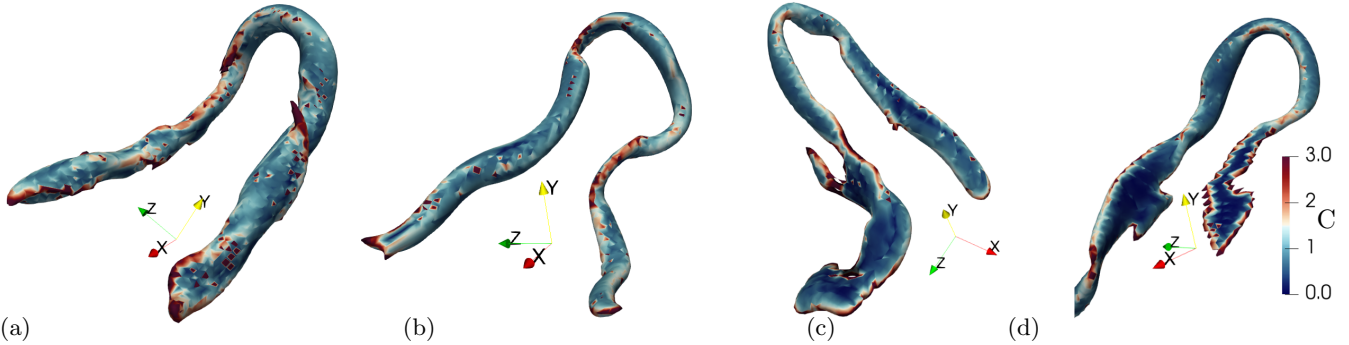


Figure 4.25: Dimensionless curvedness mapped on hairpins extracted from the $M_c = 0.3$ (a, $\tau = 232, \hat{C} = 0.78$), (b, $\tau = 360, \hat{C} = 0.82$) and $M_c = 1.1$ (c, $\tau = 464, \hat{C} = 0.63$), (d, $\tau = 424, \hat{C} = 0.58$) mixing layer. For the higher convective Mach number flatter regions are found on the legs of the hairpin.

4.7 Hairpins in the geometrical feature space

The proposed detection method, §4.6, is applied to filter hairpin-like structures from all extracted structures of the compressible mixing layers considering all frames from initialization until the end of the simulations in the post-self-similar regime. The filter settings are listed in table 4.5. The detected hairpin-like structures are mapped into the geometrical feature space in fig. 4.26. The hairpin vortices are characterized as stretched tube-like structures. While there is a number of detections showing a balance of hyperbolic and elliptic surface points ($\hat{S} \approx 0.5$), elliptic surface points slightly dominate for most cases due to the shape of the outer head region and the tips of the legs, fig. 4.20 (a). For the lower convective Mach number ($M_c = 0.3$) the range in geometric signature is in agreement with hairpins recently reported in stably stratified atmospheric boundary layers (Harikrishnan *et al.*, 2021). Similar to the observations regarding structures of all shapes in the self-similar regimes, see fig. 4.14, the curvedness of the detected hairpins decreases with increasing convective Mach number. The decrease in curvedness is mainly localized on the hairpin legs rather than on the head of the structure. Highly stretched hairpin detections are typically

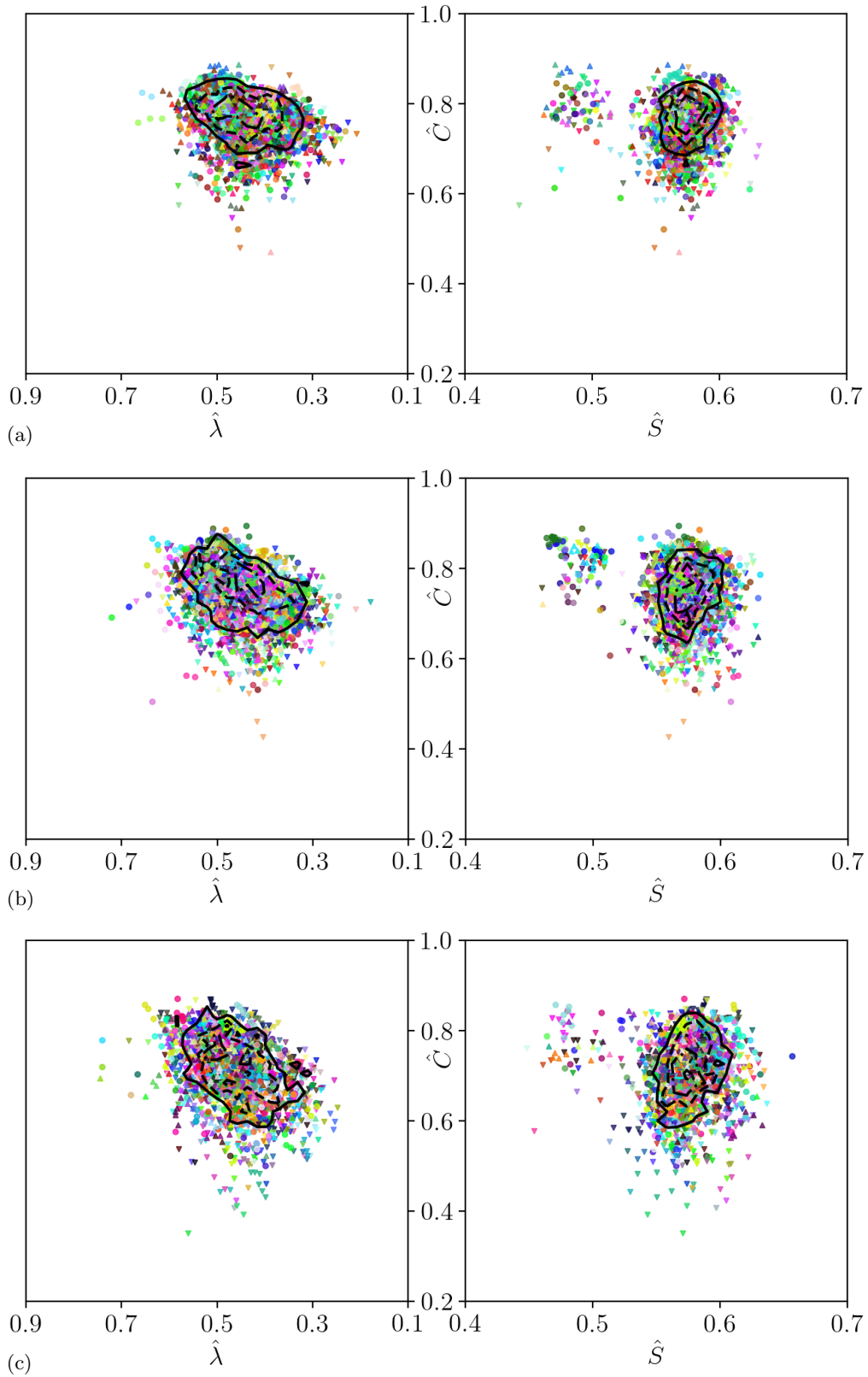


Figure 4.26: Hairpin vortices in the geometrical feature space. Hairpins detected in the $M_c = 0.3$ (a), $M_c = 0.7$ (b) and $M_c = 1.1$ (c) mixing layer. Markers of the same color indicate the same structure detected as hairpin-like in different (not necessarily successive) frames. Marker types are indicative of the time of extraction: Pre-self-similar \blacktriangledown , self-similar \bullet , post-self-similar \blacktriangle . The lines —, --, -.- correspond to 0.25, 0.5 and 0.75 contour level respectively.

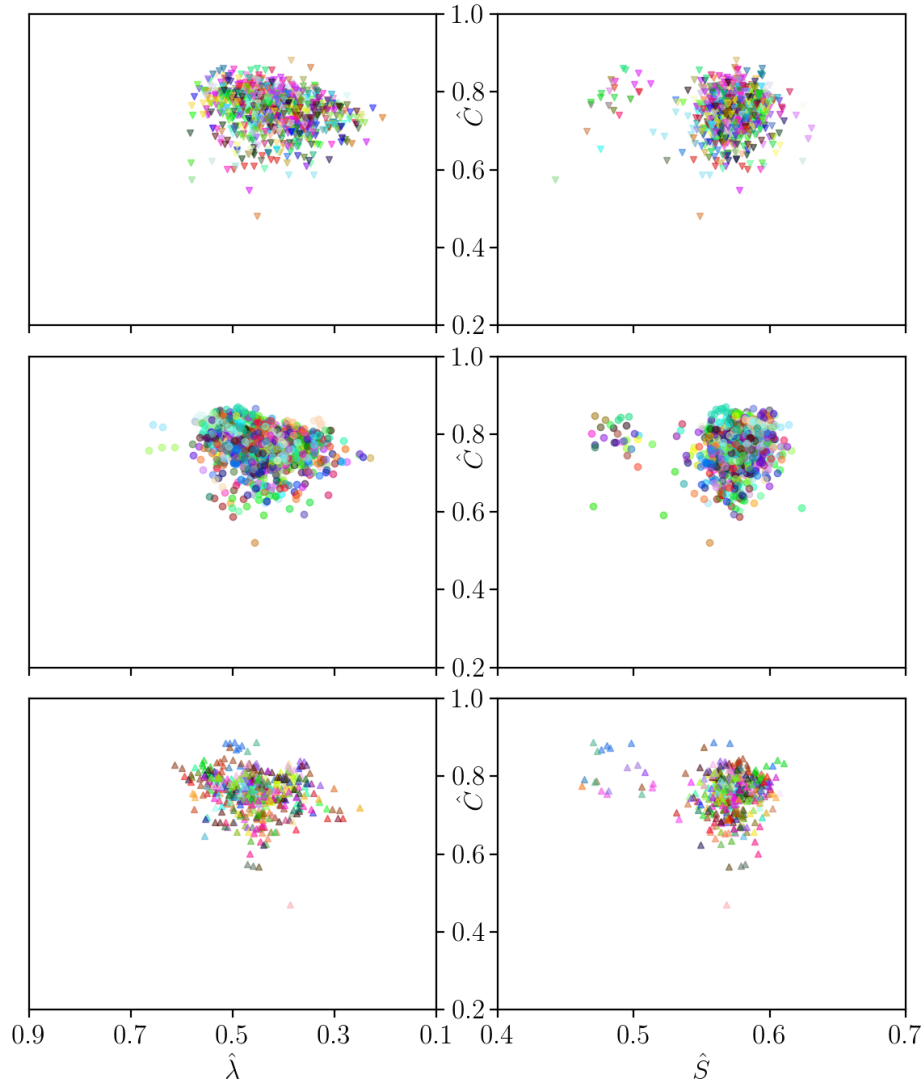


Figure 4.27: Geometrical signatures of hairpin-like vortices detected in the pre-self-similar (top), self-similar (center) and post-self-similar stage of the $M_c = 0.3$ mixing layer. Marker color indicative of tracking ID.

characterized by a medium curvedness compared to other detections of the same convective Mach number. Similar geometrical signatures are obtained for hairpins detected at different stages of the mixing layer evolution (pre-/post-/self-similar), see fig. 4.27.

4.8 Temporal distribution and formation of hairpin-like vortices

The temporal distribution of hairpin-like structure detections during the mixing layer evolution is presented in fig. 4.28. If an individual structure is detected to be of hairpin-like shape in multiple frames within its lifetime, each detection contributes to the distribution. Throughout their lifetime Q-structures can develop and lose hairpin-like shape multiple times with or without interacting with other structures. Therefore the lifetime of a structure

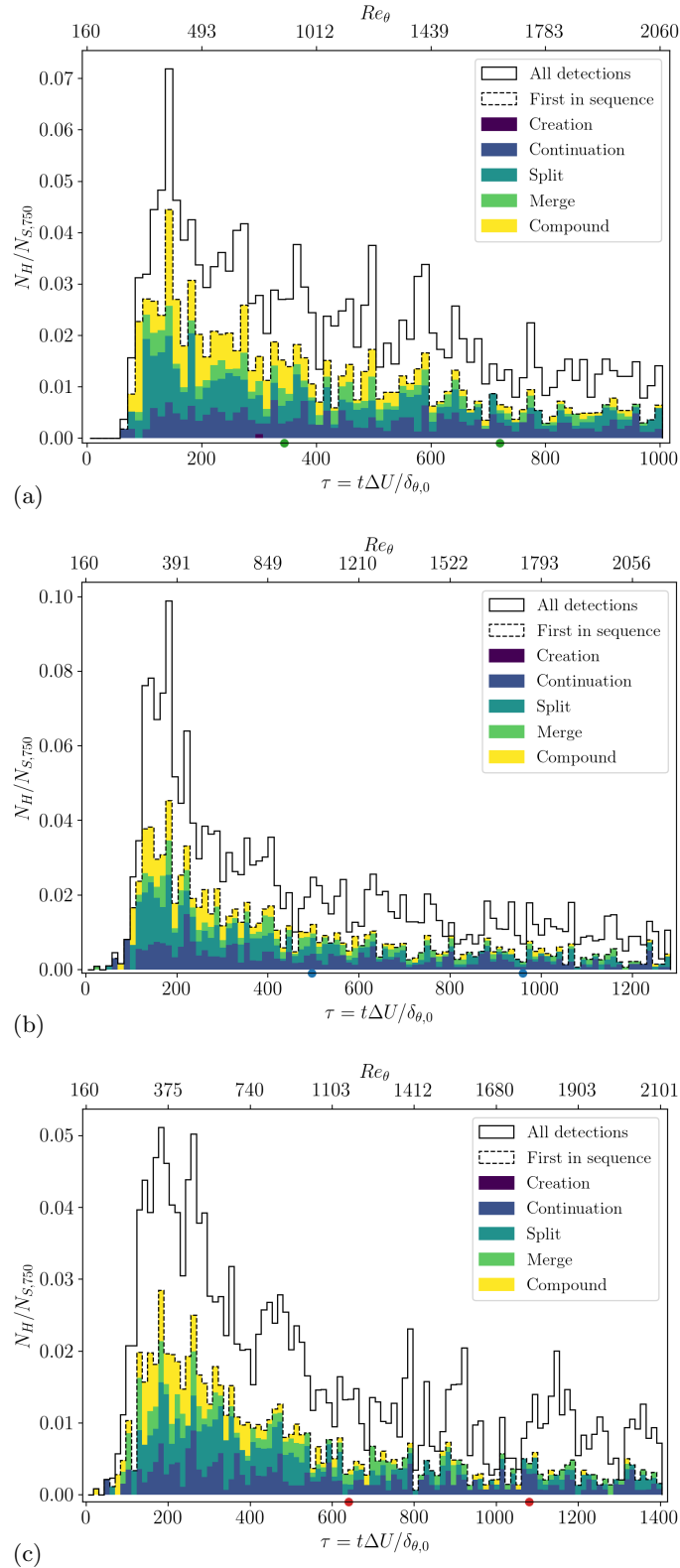


Figure 4.28: Temporal distribution of hairpin detections in mixing layers of convective Mach number $M_c = 0.3$ (top), $M_c = 0.7$ (center) and $M_c = 1.1$ (bottom). Number of detections normalized by the total number of extracted structures with at least 750 surface points of the time step. While the solid distributions include all detections, the dashed distributions are based on the first in sequence detections. Markers on the temporal axis show self-similar periods. The event types forming the hairpin-like structures are included as stacked contributions to the first in sequence distribution.

temporarily in hairpin-like shape can be split into sequences of detections in consecutive frames of the simulation. The sequences of consecutive detections vary in temporal length, however, the majority is found to be rather short-lived ($\Delta\tau < 20$, or even shorter).

For all considered convective Mach numbers, the number of hairpin detections peaks in the pre-self-similar stage and decays afterwards. This concentrated occurrence is related to the breakup process of the large, connected structures, see fig. 4.12 and fig. 4.20(b), which were formed through (sub-)merge events in the very early stage of the mixing layer development. Consequently, split and compound event, i.e., sub-split events, contribute the most to the peak occurrence of individual, disconnected hairpins. Examples of this structural breakup process are rendered in fig. 4.29 for the $M_c = 0.7$ mixing layer. The breakup process is observed for connected structures within which the hairpin heads are located in the same free stream side as well as in opposite sides. The breakup of large, connected hairpins is influenced by the choice of the iso-value used for extracting the Q-structures, i.e., a structure which is observed to breakup for the used iso-value of the frame, may stay connected for a lower iso-value.

As discussed in §4.4, the $M_c = 0.3$ case exhibits the most restrictive requirement for the choice of n in $Q_{\text{iso}} = \langle Q \rangle + n\sqrt{\langle Q^2 \rangle - \langle Q \rangle^2}$ to prevent the formation of very large connected structures in the early mixing layer development. While a choice of $n = 6$ indeed prevents the formation of domain-spanning structures, a significant portion of hairpin-like structure may still be connected for this choice of n and is therefore not detected as individual hairpin-like structures, which decreases the count of detections in fig. 4.29(a). The number of connected hairpins (not accounted for in this analysis) need to be kept in mind when evaluating the peak count of detections for the different convective Mach numbers. For the given choice of $n = 6$ the normalized peak count decreases with increasing convective Mach number from $M_c = 0.7$ to $M_c = 1.1$. With increasing convective Mach number the occurrence of the peak in the temporal distribution of the hairpin-like detections is delayed in time, but occurs approximately at the same momentum thickness Reynolds number of $Re_\theta \approx 400$. In the self-similar regimes of the mixing layers, which are less prone to the formation of largely connected structures compared to the early stage of the mixing layer development, the normalized number of detections may slightly decrease with increasing convective Mach number. The reduced detection of hairpin-like structures for increasing momentum thickness Reynolds number Re_θ supports observations of hairpins in turbulent boundary layers for which the hairpin regeneration process was reported to be not sustained once the turbulent background is developed (Eitel-Amor *et al.*, 2015). The same authors find no evidence of hairpin vortices reaching into the wall region for $Re_\theta > 400$, which matches with the momentum thickness Reynolds number of the peaks in the temporal distribution of the hairpin-like detections in fig. 4.28. In the subsequent stages of the mixing layer evolution the detected, individual hairpin-like structures contribute less than 5% to the total number of structures with more than 750 surface points of the frame and are therefore not considered as a dominant flow feature. However, a number of hairpins connected to other (not necessarily hairpin-like) structures may still exist, but is not accounted for.

Fig. 4.30 demonstrates how hook-like structures with quite unevenly pronounced legs can develop arch- or hairpin-like shape by either splitting the dominated leg or merging with another tube-like structure which serves as the second

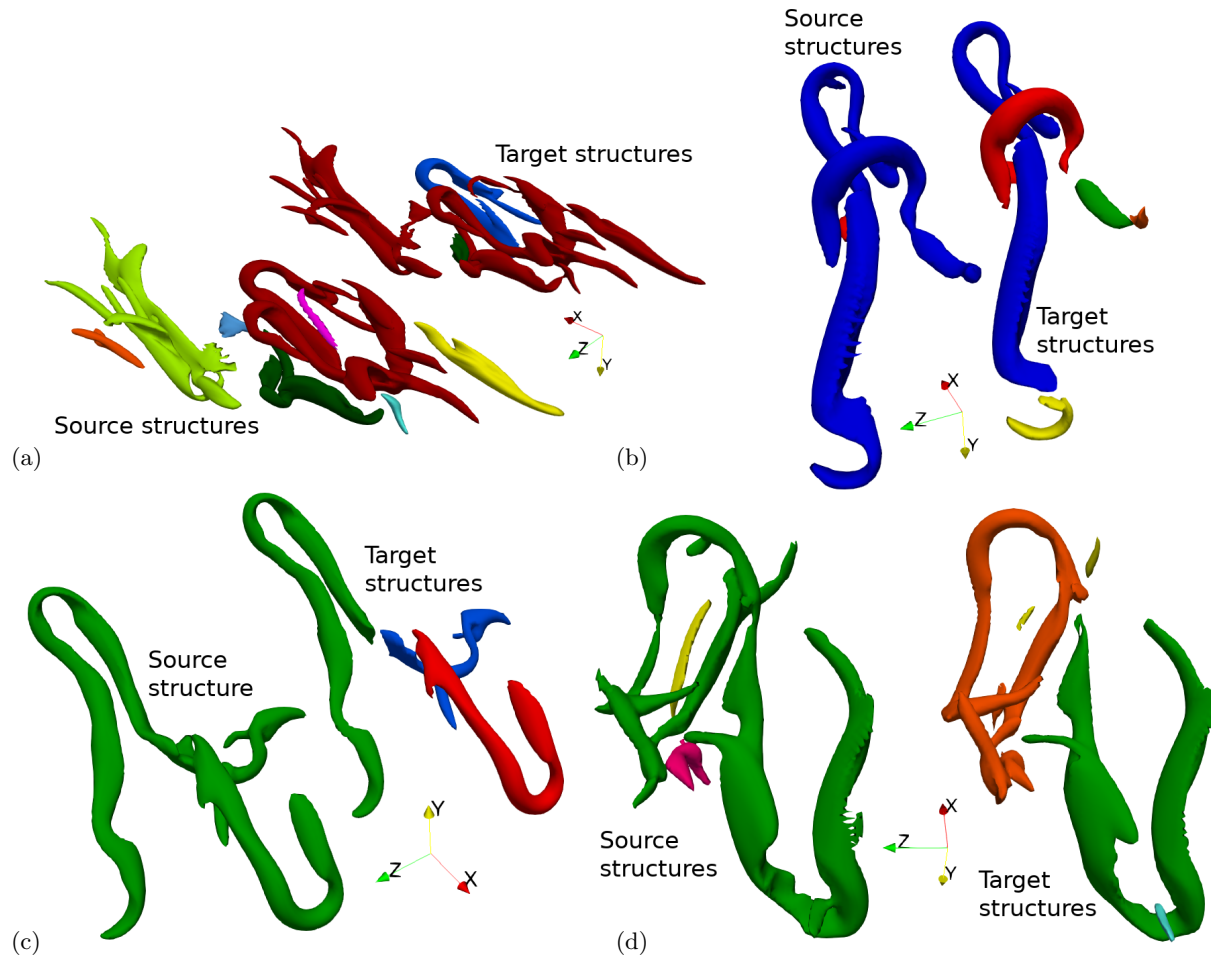


Figure 4.29: Connected, partially hairpin-like structures, see also fig. 4.12(b) and fig. 4.20(b), breaking into individual hairpins during the pre-self-similar stage of the $M_c = 0.7$ mixing layer. The breakup process is observed for connected structures within which the hairpin heads are located in the same free stream side (b, $\tau = 164$), opposite sides (c, $\tau = 172$) and (d, $\tau = 192$) as well as often in mixed cases (a, $\tau = 104$). Target structures artificially displaced for visibility, renderings are not to scale.

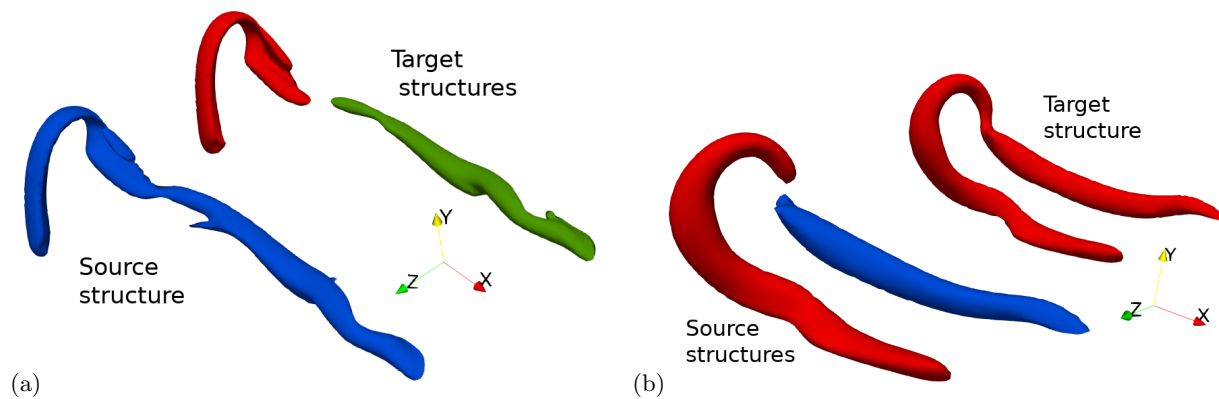


Figure 4.30: Hook-like structures developing in arch-like and hairpin-like shape by splitting the dominant leg (a) ($\tau = 156$) and merging with a tube-like vortex (b) ($\tau = 252$) in the $M_c = 0.7$ mixing layer, not to scale.

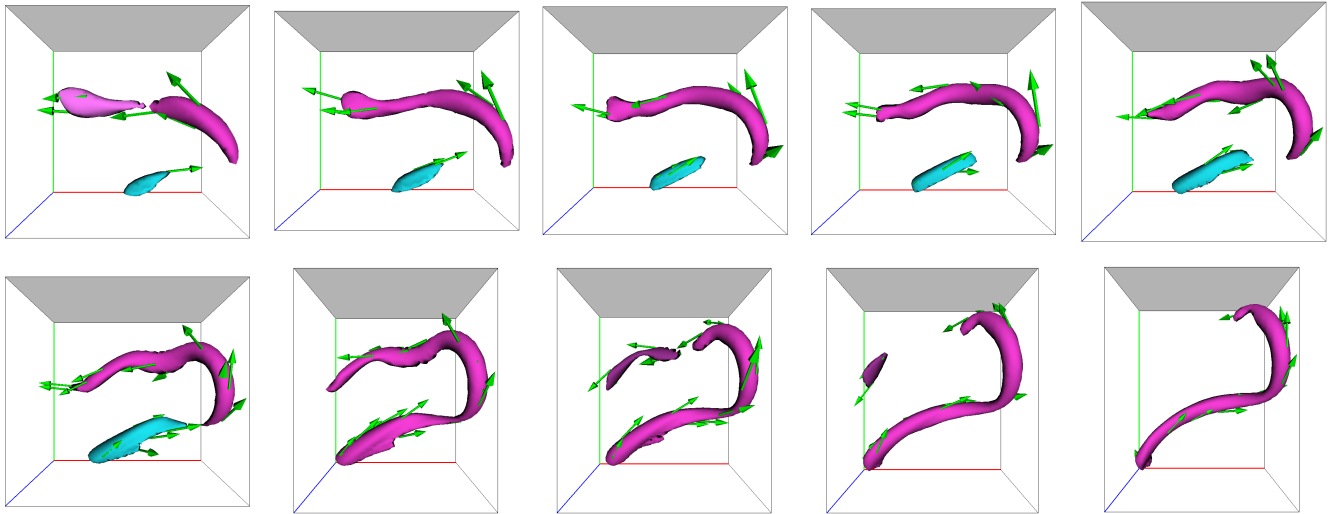


Figure 4.31: Side view (xy) of a series of merge events generating a short-lived hairpin vortex in the $M_c = 0.3$ mixing layer and subsequent breakup from $\tau = 304$ (top, left) to $\tau = 340$ (bottom, right). Structure color represents structure ID, while green arrows are the down sampled vorticity on the structures. Note the tilted orientation of the hairpin with respect to the center plane of the layer indicated in gray.

leg of the hairpin. Note, however, that hook-like structures can also develop more hairpin-like shape in a sequence of continuation events without structure interactions by gradually growing or losing a leg.

The formation of a hairpin vortex in a series of merge events is shown in fig. 4.31. First, the merging of two tube-like structures generates a hook which develops a bent shape and eventually merges with another structure to form a hairpin. Note, that both, the hook and the more tube-like structure, grow towards each other in this merging process. Interestingly, the hairpin has a tilted orientation with respect to the center plane of the mixing layer. The resulting hairpin is short-lived as it loses its first leg only one frame after its formation.

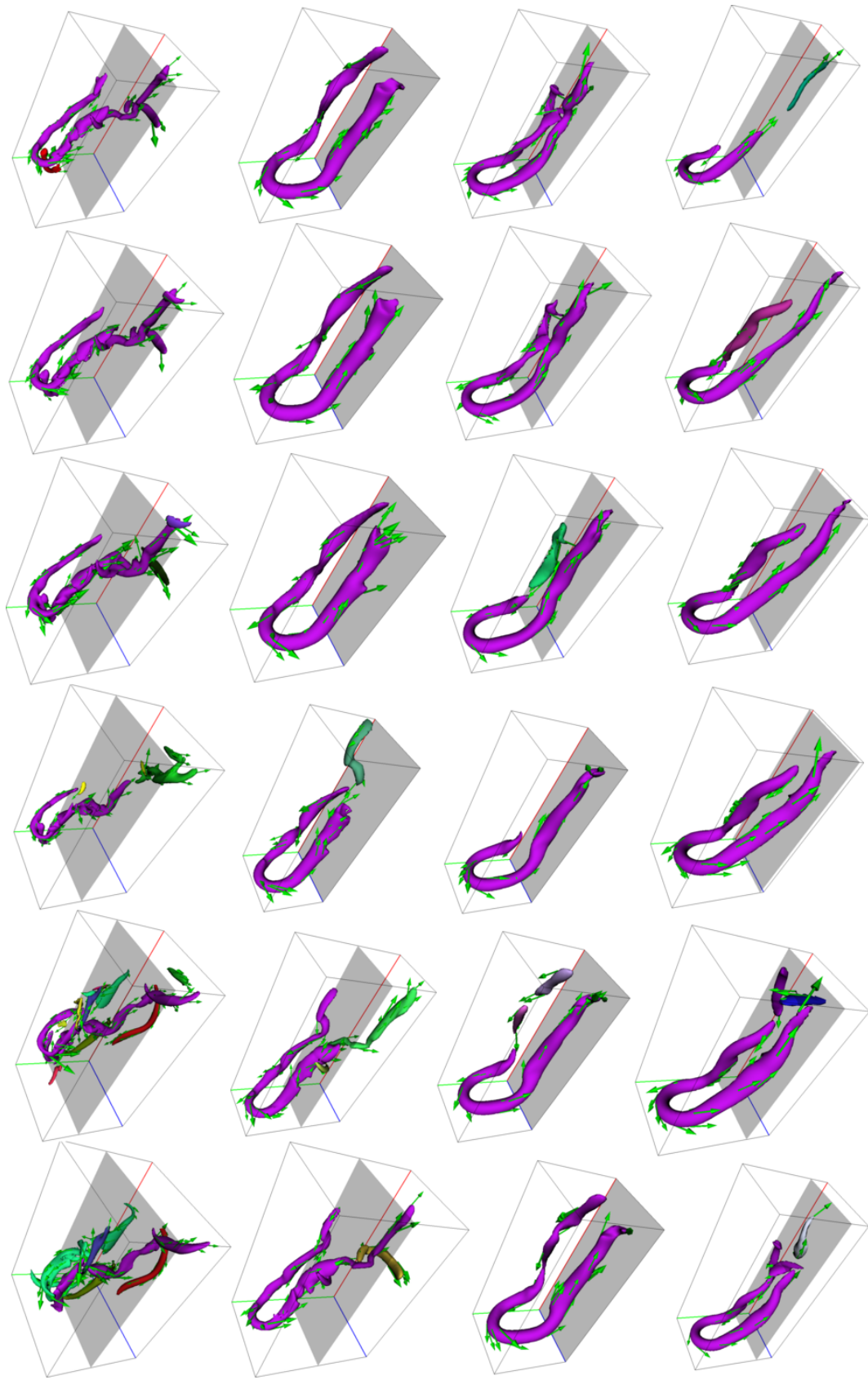


Figure 4.32: Section ($\tau = 204 - 296$) of a lifespan of a vortical structure developing hairpin-like shape in the $M_c = 0.3$ mixing layer, from left to right and top to bottom. Interacting structures included as source or target structures of the respective event. The structure is mostly located above the center plane of the mixing layer indicated in translucent black. Several structure interactions contribute to the persistence of the hairpin in the rendered section and subsequent frames (not shown).

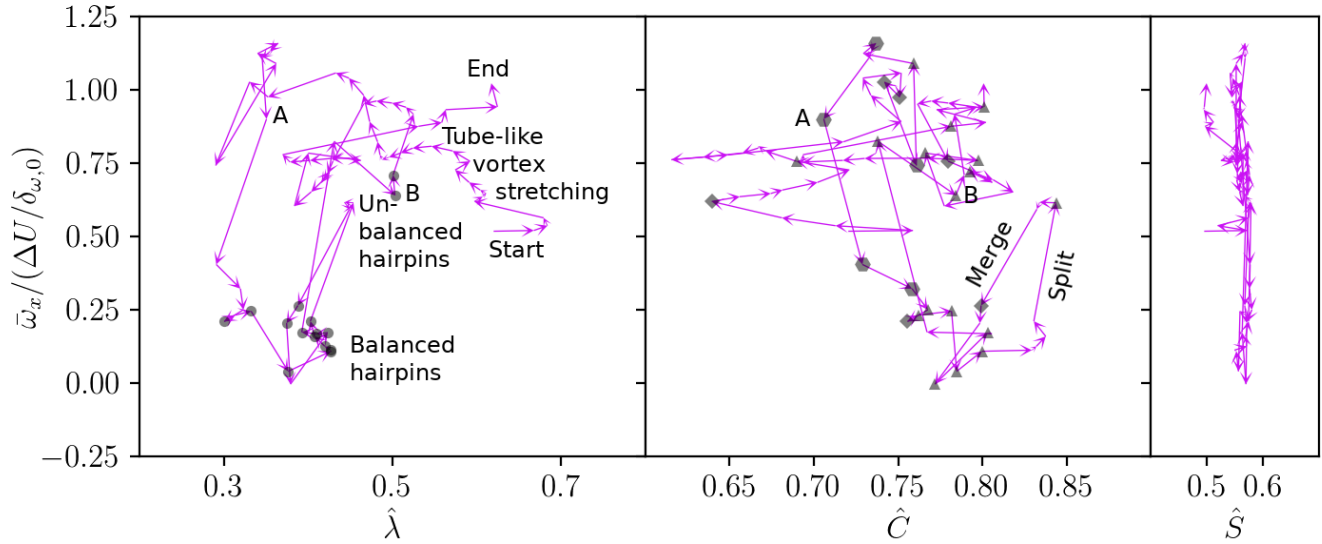


Figure 4.33: Trajectory of a vortical structure temporarily developing hairpin-like shape in geometrical feature space enhanced with surface-averaged streamwise vorticity component. The trajectory section between points A and B corresponds to the renderings in fig. 4.32. Detected hairpin-like structures are highlighted as \bullet in the left subplot. In the center subplot markers \blacktriangle , \blacklozenge and \bullet indicate if a step of the trajectory is the target of a split, merge or compound event respectively.

Fig. 4.33 presents the full trajectory of a structure which preserves its hairpin shape for a longer sequence of frames in the geometrical feature space enhanced with the surface-averaged streamwise vorticity component. The temporal evolution of the structure is rendered for a subsection of its trajectory in the corresponding fig. 4.32. The structure originates from a creation event at $\tau = 80$ and develops tube-like shape, while being orientated mainly in streamwise direction and intersecting with the center plane of the mixing layer. As the tube-like structure becomes more stretched, its streamwise vorticity component intensifies. The structure develops hook-like shape as a result of a complex compound event and eventually becomes hairpin-like over a series of frames within which one leg intensifies while the other leg is shortened in splits. At the time the structure is first detected as a hairpin the streamwise vorticity component is almost balanced on the structure. The geometric signature of the hairpin detections is in agreement with those presented in fig. 4.26. The sequence of consecutive hairpin detection ends in a split of a leg, which leads to an imbalance in the surface-averaged streamwise vorticity. However, the structure retrieves its hairpin shape in a merge with another structure serving as the new leg.

4.9 Cross-stream and spanwise growth of hairpin vortices

The entire lifespan of two structures which temporarily develop hairpin-like shape in the $M_c = 0.3$ mixing layer are shown in fig. 4.34 and fig. 4.35. Both structures remain close to the center plane of the mixing layer throughout their lifetime, often intersecting with it. Therefore, the streamwise component of the surface-averaged velocity changes its sign multiple times during the structure evolution. Resulting from a split, the structure shown in fig. 4.35 has

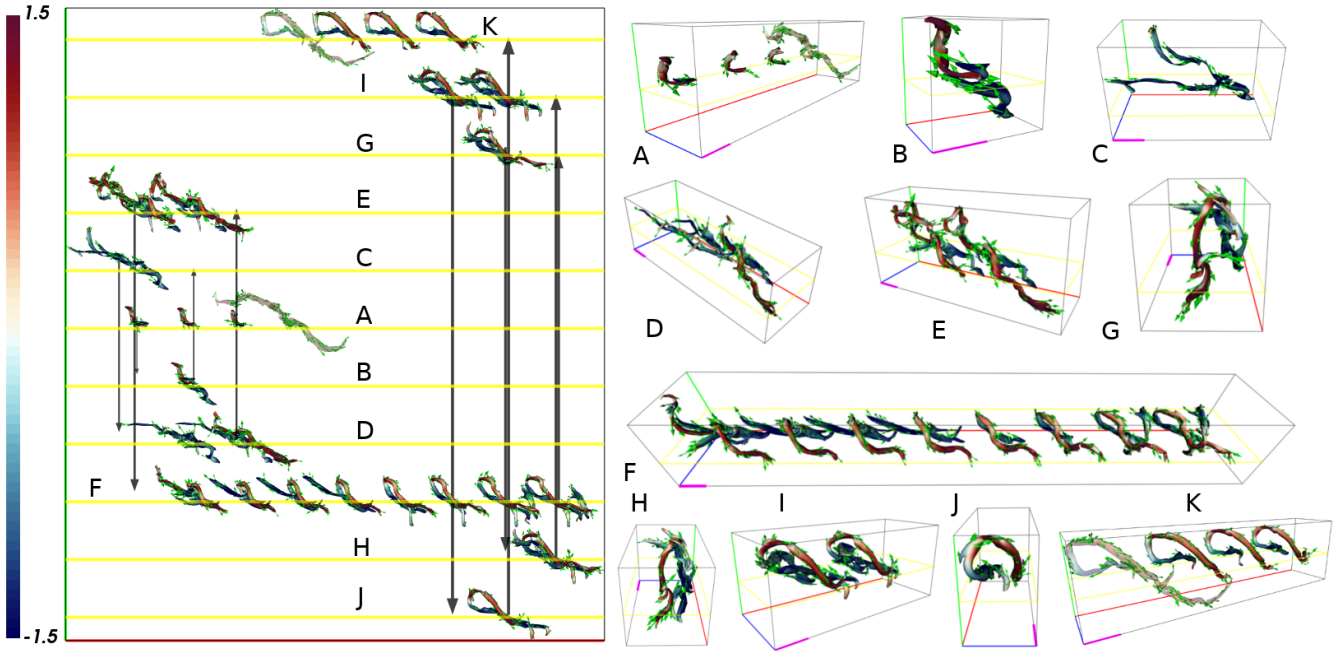


Figure 4.34: Lifetime ($\tau = 280 - 376$) of a structure temporarily developing hairpin-like shape in the $M_c = 0.3$ mixing layer. Streamwise vorticity mapped on the structure and vorticity sampled as green arrows. Due to the proximity to the center plane of the mixing layer, the surface-averaged streamwise velocity of the surface often changes its sign. Artificial structure displacement added for visibility based on the sign of this surface-averaged velocity: Starting from level A a transition upwards indicates a change towards an (on average) left moving structure, while a downwards transition indicates a shift toward a right moving structure. Size of the black arrow indicates the temporal order of these transitions. The yellow plane indicates the center plane of the mixing layer at each level of the figure. The 3D renderings on the right show the structures of each level of the left side of the figure (parallel projection). While the rendering are not to scale, the pink line serves as a reference scale (10 times the initial momentum layer thickness). The structures from which the hairpin-like structure originally split (source of (sub-)split) and eventually merges (target of (sub-)merge) are added with reduced opacity. Note that the head of the hairpin becomes quite tilted at the end of the lifetime.

a hairpin-like shape with the head located in the upper free stream. However, multiple structure interactions later, the structure has developed into a larger hairpin with the head located in the lower free stream.

The distribution of the cross-stream location of all detected hairpin-like structures and the first-in-sequence subset is presented in fig. 4.36 for the three considered convective Mach numbers. Here, the cross-stream location of a structure is defined as the mean of its minimal and maximal cross-stream position. Following the Reynolds stress profiles, fig. 4.7 and fig. 4.8, the distributions peak at the center plane and are about symmetrical. The distribution including only the first-in-sequence detections shows slightly less variance and more detections occur closer to the center plane.

Fig. 4.37 investigates the subset of the center plane crossing detected structures and their extent on each free stream side of the mixing layer relative to the momentum layer thickness at the time. The majority of the structures shows an uneven top and bottom cross stream extent, which is also supported by the previous renderings of detected structures. For hairpins whose legs are oriented in the streamwise direction and the head in spanwise direction, the

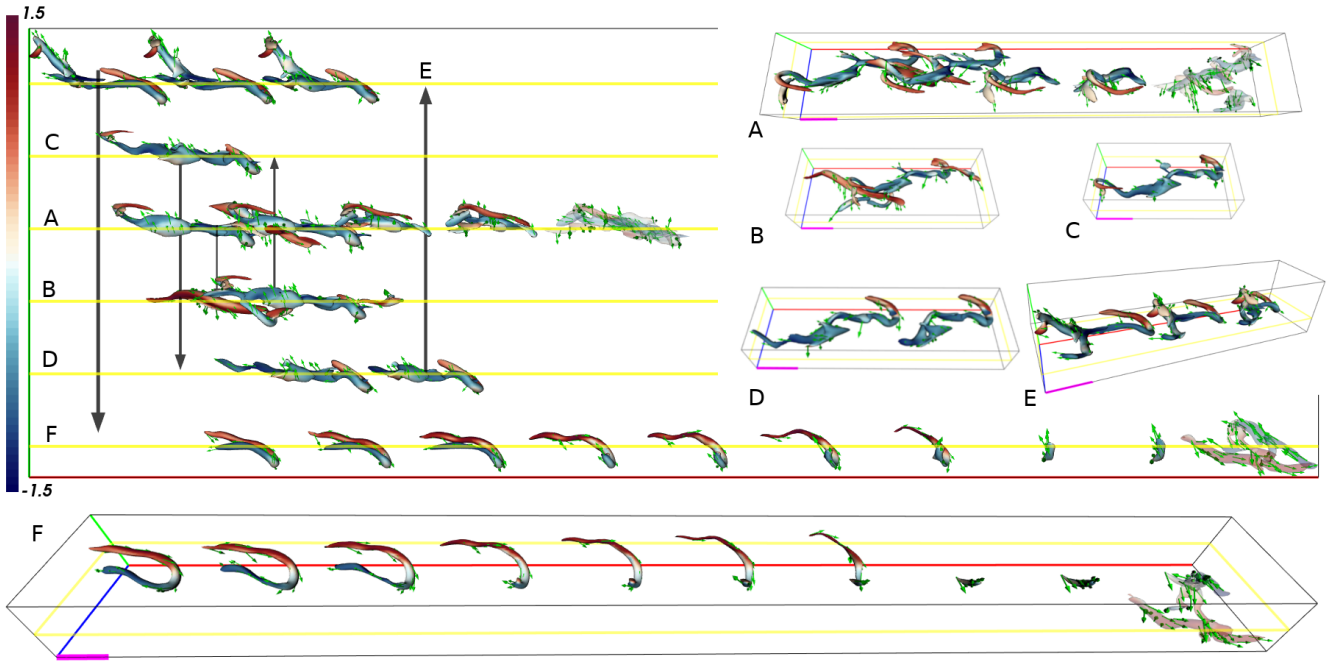


Figure 4.35: Lifetime ($\tau = 96 - 172$) of a structure developing temporarily hairpin-like shape while crossing the center plane of the mixing layer. As frequently observed, the hairpin loses one leg over time and becomes a hook-like structure. Other plot elements as in fig. 4.34.

dominant cross-stream side is the side on which the head is located. Fig. 4.37 includes structures which have a tilted orientation with respect to the center plane of the mixing layer.

In turbulent boundary layers the cross-stream and spanwise growth rate are reported to be similar to each other and is associated with Townsend's attached eddy hypothesis (Adrian, 2007). Therefore, the hairpin growth behavior is investigated in the context of the mixing layer configuration, in fig. 4.38. The minimal and maximal surface point coordinate in the cross-stream and spanwise direction is used to define the extent of a hairpin structure in the particular direction, Δy and Δz . The evolution of each structure temporarily in hairpin shape is inspected visually to exclude structures that show a significant tilted orientation and hairpins that are distorted by connecting to other structures at any point in their evolution. While no exact measure is used to exclude those structures, it is ensured that the spanwise extent is representative for distance of the outsides of the legs and the cross-stream extent is an indication of the distance between the head and the legs. Due to these requirements mainly hairpins occurring in the pre-self-similar regime are included in the analysis. While the growth behavior differs between individual hairpins, the overall trend for the $M_c = 0.3$ case is in rough agreement with the findings reported for hairpins in wall-bounded turbulence. The cross-stream and spanwise structure extents are similar to each other. The results are less clear for the $M_c = 1.1$ mixing layer as a number of hairpins is found to grow in cross-stream direction while approximately preserving their spanwise extent. However, the overall ranges of cross-stream and spanwise extents are similar for both convective Mach numbers.

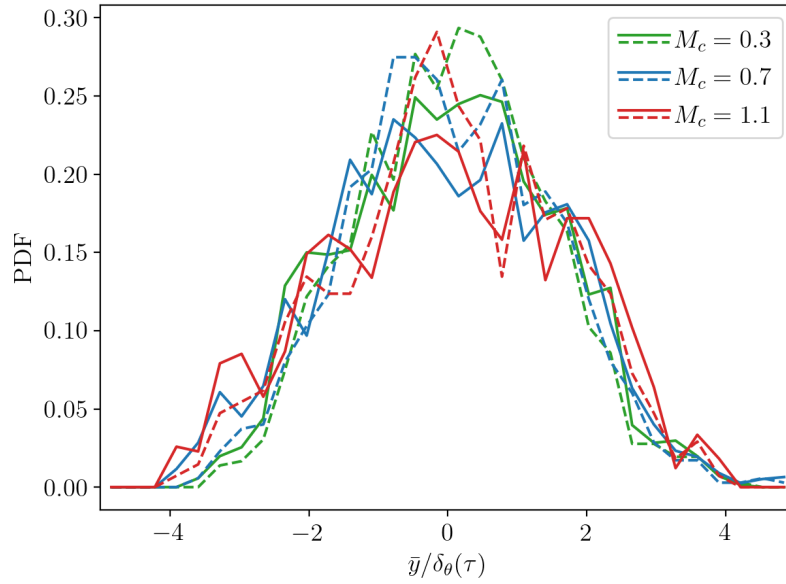


Figure 4.36: Distribution of the cross-stream location of all detected hairpins (solid) and the first-in-sequence subset of hairpins (dashed) normalized by the momentum thickness. Compare to the Reynolds stress profiles in fig. 4.7 and fig. 4.8 as well as the cross-stream distributions of newly created structures of all shapes in the self-similar regimes in fig. 4.13 (b).

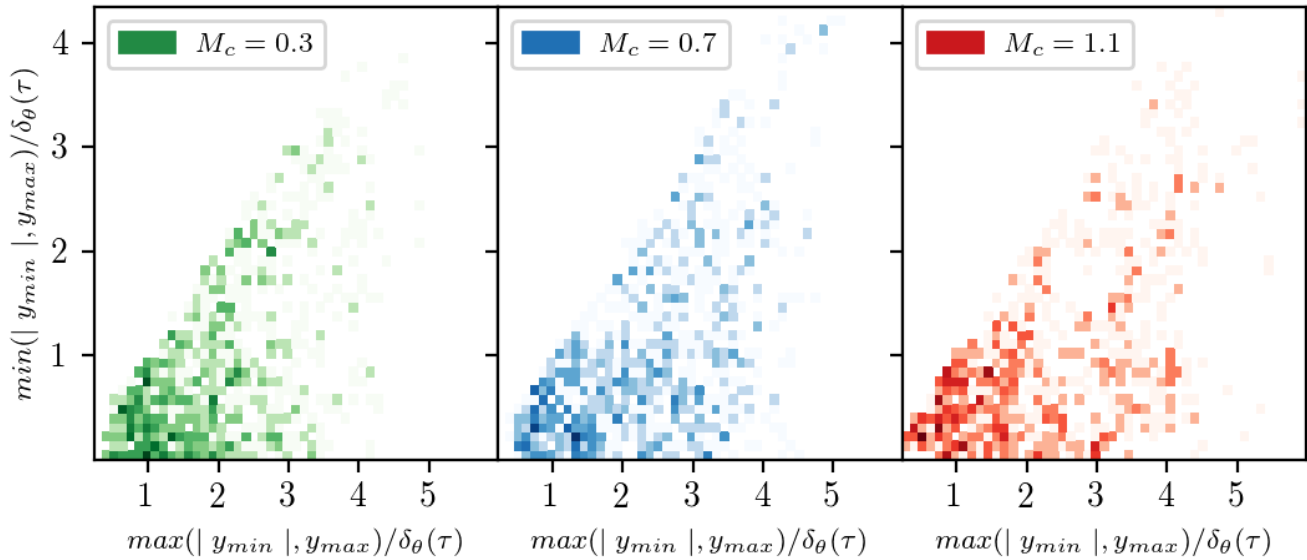


Figure 4.37: The cross-stream extent of center plane crossing hairpins. Most hairpin-like structures emerge unevenly into the top and bottom free stream. On the basis of prior renderings, the dominant cross-stream side is the side where the head is located.

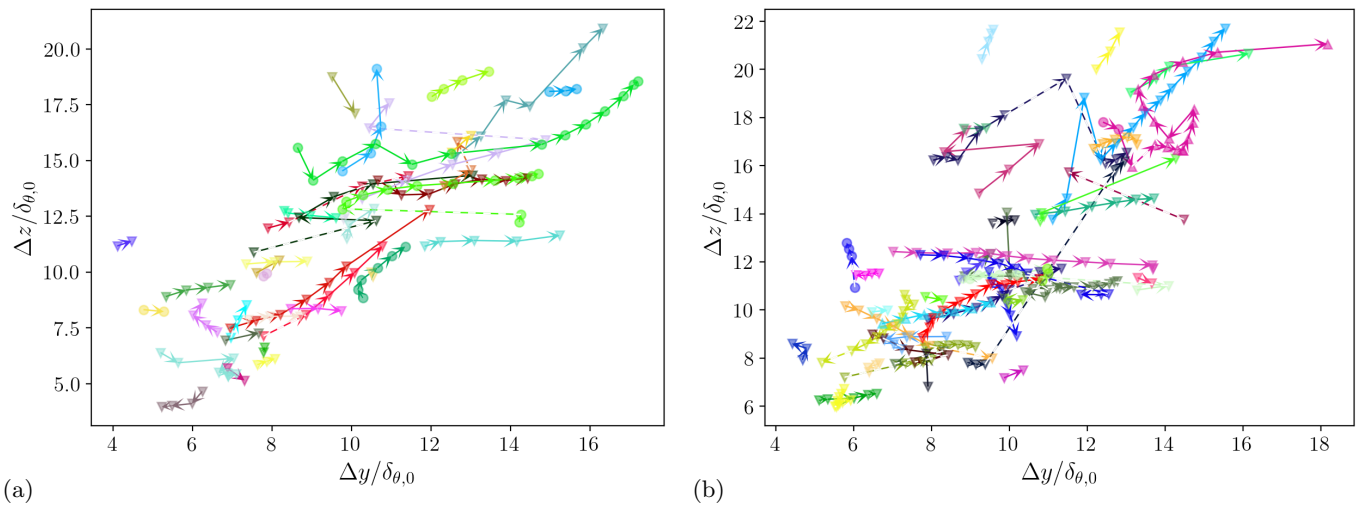


Figure 4.38: Growth of the cross-stream Δy and spanwise Δz extent of selected hairpin-like vortices in the $M_c = 0.3$ (a) and $M_c = 1.1$ (b) mixing layers. Colors indicate individual tracking structure ID, while temporal order is shown with arrows. For hairpin detections in non-consecutive frames dashed arrows are used. Note that the hairpins may be affected by structure interactions at any time, which influence the extent of the structure. Hairpin detection in the pre-self-similar self-similar and post-self-similar stage plotted as \blacktriangledown , \bullet and \blacktriangle , respectively.

Chapter 5

Conclusion

A hybrid approach of attribute- and region-based local tracking has been introduced to study the temporal evolution of flow features, defined as closed surfaces of any kind in a turbulent flow field. As a pre-step of the explicit tracking, flow features are extracted and characterized for each frame of the given data set independently. The resulting spatial and geometrical feature attributes are utilized in the search for correspondences between features in consecutive frames aiming to generate candidates of temporal feature relation. A sequence of region-based correspondence and attribute-based event realizability constraints has been proposed to test the physical plausibility of the detected correspondence candidates. A graph data structure is built to organize the evolution and record the interactions of individually tracked flow features over time. The graph data structure is analyzed and queried to gain insight of the flow feature dynamics by means of individual structure evolution as well as ensemble statistical evolution. Graph data mining techniques have been adapted to search for common patterns in the interactions between flow features.

The tracking algorithm was applied to investigate the evolution of passive scalar structures in temporally decaying homogeneous isotropic turbulence and statistically stationary shock-turbulence interaction at Mach numbers 1.5 and 3.0. Here, the flow features were defined as iso-surfaces of the underlying passive scalar field. The tracking application and the comparison of both flow configurations allowed to elucidate the effect of shock waves on the scalar mixing on finite-sized structure evolution and surface-mapped physical quantities relevant to mixing.

Shock-induced deformation increases the occurrence of split and subsequent disappearance events, shifting those events closer to the shock for increasing Mach number. The structural breakup process facilitates the scalar mixing. As expected, initially larger structures take longer to diffuse, but their mixing benefits more from the shock-interaction as the number of splits and disappearances per incoming structure increases more (compared to the DHIT) and the upstream shift of event activity is more significant than for smaller structures.

Small-structure interactions were found to deviate from volume-preservation as a consequence of increased scalar dissipation at fine scales, whereas events involving larger structures show a high degree of volume preservation (apart from larger shock-crossing structures which are compressed and affected by the shock-amplified scalar dissipation). Structures that have shrunk to a certain volume due to scalar dissipation or previous splits, tend not to split further, and simply continue until they reach the end of their lifetime and disappear by diffusion.

The geometric signature of structures involved in split and merge events show opposite trends: splitting (merging) structures are more stretched (compact) and show a larger (smaller) area of hyperbolic points than the resulting split (merged) structures. Compared to DHIT, shock-induced compression of the passive scalar structures increases their stretching and, in turn, the probability of the structure to split in the STI configuration.

The enhancement of scalar dissipation by the shock is expressed at the structural level as an amplification of the surface-averaged scalar gradient. This amplification increases with initial structure size, indicating a better mixing enhancement by the shock-interaction for larger structures. Surface-averaged alignments of the scalar gradient with the most extensional strain-rate eigendirection increases across the shock, while the alignment with the intermediate and most compressive eigenvector decreases. This result, previously reported in volumetric statistical analysis of fully-developed scalar mixing, is here confirmed also at the structure level from scalar fields initialized with uniform patterns of well-defined shapes.

Larger scalar gradient (hence, dissipation), its shock-induced amplification, and alignment with the most compressive strain-rate eigendirection were found to correlate with flatter surface regions. These correlations emerged shortly after the initial convolution of the initially spherical structures by the background turbulence. The geometrical trajectories of the passive scalar structures depend on their initial sizes (relative to the eddy sizes of the background turbulent flow). The amount of surface convolution early in the structure's lifetime increases with initial size and leads to larger curvedness and area-coverage of hyperbolic points, resulting in an increased number of splits per structure. Shock compression alters the geometrical trajectories in the STI configurations, lowers the compactness and curvedness of the structures. Less compact (i.e. more stretched) structures are more likely to split, while a decreased curvedness shows increased area-coverage of flat surface regions where scalar dissipation dominates. The sudden shock interaction flattens the structures and enhances mixing more for larger structures. The post-shock amplification of turbulence increases the curvedness again, inducing a period of re-growth of the surface area and stretching of the structures, which increases the probability of splits and enhances scalar mixing.

In a second application, the flow feature tracking was applied to study the dynamics and geometrical properties of vortical structures in temporarily developing mixing layers of varying convective Mach numbers $M_c = \{0.3, 0.7, 1.1\}$ to complement existing statistical analyses on the inhibiting effect of compressibility on the turbulence intensities in such flow configuration. The compressible Q-criterion was used to identify vortices in the flow field. The tracking algorithm was successfully applied with the same types of realizabilty constraints as in the passive scalar mixing. In contrast to the passive scalar application, a time-dependent iso-value was used for the structure extraction to account for the temporarily developing character of the flow. With increasing convective Mach number less vortices were extracted which pronounces the inhibiting effect of compressibility on the structural level. Higher compressibility was observed to lower the curvedness of the extracted Q-structures.

The application of the graph data mining approach focused on the creation of vortices in the neighborhood of existing vortices. The subsequent merging of both vortices is recognized as a prominent pattern of the vortex dynamics. Other patterns of statistical significance are split-dominated or highlight the short-lived evolution of isolated structures without structure interactions in their lifetime.

A detection method for hairpin-like vortices was introduced based on geometrical and physical constraints. In combination with the tracking, the hairpin filter was used to evaluate the occurrence, persistence and significance of hairpin-like vortices in the mixing layers. For all convective Mach numbers, the number of hairpin detections peaks at $Re_\theta \approx 400$ in the pre-self-similar stages of the mixing layers and decays afterwards, suggesting that hairpins are not a dominant flow feature once the turbulent background is developed. These findings support previous reports of the hairpin regeneration process, which was found not to be sustained at higher Reynolds number in wall-bounded turbulence. It would be of great interest to evaluate what influence different choices for the iso-value have on the number of detected hairpins, especially for the lower Mach number case.

The time span in which vortices preserve their hairpin-like shape varies, but most hairpins are rather short-lived as they cross the center planes of the mixing layers and are partially pulled into the opposite free streams. This mechanism could explain the reduced lifetime of hairpins in the mixing of two opposite free streams compared to those found in boundary layer flow. The majority of individual hairpins is formed by the structural breakup process of largely connected hairpins which originate from Λ vortices that grow in the early stage of the mixing layer through merging of tube-like vortices. A number of hairpins have a tilted orientation with respect to the center plane, especially in the later stages of the mixing layer. Hairpins with quasi-streamwise orientation exhibit equal rates of simultaneous growth in cross-stream and spanwise direction.

Future extensions of the presented flow feature tracking methodology could target to track features not of only one type, but of multiple types simultaneously. This would allow to study relations between the dynamics of flow feature of different types. The graph mining technique would greatly benefit from a pattern classification method to summarize the pattern extracted from large graphs better. To increase the performance of the algorithm, parallelism could be added to the implementation, especially to the local surface distance computation that is currently the computational bottleneck of the method. Another mode can be introduced to the algorithm in which it does not operate as a pure post-processing tool of a given data set, but in which the tracking is coupled with the solver generating the data set. Thereby an adaptive sampling frequency can be determined on the fly by evaluating the complexity of the detected events.

Bibliography

- ABDEL-KAREEM, W. 2011 Tracking of vortical structures in three-dimensional decaying homogeneous isotropic turbulence. *Int J Mod Phys C* **22** (12), 1373–1391.
- ACARLAR, M. S. & SMITH, C. R. 1987 A study of hairpin vortices in a laminar boundary layer. Part 1. hairpin vortices generated by a hemisphere protuberance. *J. Fluid Mech.* **175**, 1–41.
- ADRIAN, R. J. 2007 Hairpin vortex organization in wall turbulence. *Phys. Fluids* **19** (4), 041301.
- ADRIAN, R. J., MEINHART, C. D. & TOMKINS, C. D. 2000 Vortex organization in the outer region of the turbulent boundary layer. *J. Fluid Mech.* **422**, 1–54.
- AGUI, J. H., BRIASSULIS, G. & ANDREOPOULOS, Y. 2005 Studies of interactions of a propagating shock wave with decaying grid turbulence: velocity and vorticity fields. *J. Fluid Mech.* **524**, 143.
- ANDREOPOULOS, Y., AGUI, J. H. & BRIASSULIS, G. 2000 Shock wave—turbulence interactions. *Ann. Rev. Fluid Mech.* **32** (1), 309–345.
- ASHURST, W. T., KERSTEIN, A., KERR, R. & GIBSON, C. 1987 Alignment of vorticity and scalar gradient with strain rate in simulated navier–stokes turbulence. *Phys. Fluids* **30** (8), 2343–2353.
- ATTILI, A. & BISETTI, F. 2012 Statistics and scaling of turbulence in a spatially developing mixing layer at $re_\lambda = 250$. *Phys. Fluids* **24** (3), 035109.
- BAJAJ, C., SHAMIR, A. & SOHN, B.-S. 2002 Progressive tracking of isosurfaces in time-varying scalar fields .
- BANKS, D. C. & SINGER, B. A. 1995 A predictor-corrector technique for visualizing unsteady flow. *IEEE Trans Vis Comput Graph* **1** (2), 151–163.
- BARRE, S., ALEM, D. & BONNET, J. 1996 Experimental study of a normal shock/homogeneous turbulence interaction. *AIAA J.* **34** (5), 968–974.
- BATCHELOR, G. K. 1959 Small-scale variation of convected quantities like temperature in turbulent fluid part 1. general discussion and the case of small conductivity. *J. Fluid Mech.* **5** (1), 113–133.
- BAUER, D. & PEIKERT, R. 2002 Vortex tracking in scale-space. In *Proceedings of the symposium on Data Visualization 2002*, pp. 233–ff. Eurographics Association.

- BELL, J. H. & MEHTA, R. D. 1990 Development of a two-stream mixing layer from tripped and untripped boundary layers. *AIAA J.* **28** (12), 2034–2042.
- BELLMAN, R. E. 2015 *Adaptive control processes: a guided tour*. Princeton university press.
- BENTLEY, J. L. 1975 Multidimensional binary search trees used for associative searching. *Commun. ACM* **18** (9), 509–517.
- BERDAHL, C. & THOMPSON, D. 1993 Eduction of swirling structure using the velocity gradient tensor. *AIAA J.* **31** (1), 97–103.
- BERGEN, G. V. D. 1999 A fast and robust gjk implementation for collision detection of convex objects. *J. graph. tools* **4** (2), 7–25.
- BERMEJO-MORENO, I. & PULLIN, D. 2008 On the non-local geometry of turbulence. *J. Fluid Mech.* **603**, 101–135.
- BILGER, R. W. 1976 The structure of diffusion flames. *Combust. Sci. Technol.* **13** (1-6), 155–170.
- BILGER, R. W. 2004 Some aspects of scalar dissipation. *Flow, Turb. and Combust.* **72** (2), 93–114.
- BIRCH, S. F. & EGGERS, J. 1972 A critical review of the experimental data for developed free turbulent shear layers. *Free Turbulent Shear Flows, Volume I-Conference Proceedings, NASA SP-321* pp. 11–40.
- BLANCO, J. L. & RAI, P. K. 2014 nanoflann: a C++ header-only fork of FLANN, a library for nearest neighbor (NN) with kd-trees. <https://github.com/jlblancoc/nanoflann>.
- BOGDANOFF, D. 1983 Compressibility effects in turbulent shear layers. *AIAA J.* **21** (6), 926–927.
- BONNET, J., DEBISSCHOP, J. & CHAMBRES, O. 1993 Experimental studies of the turbulent structure of supersonic mixing layers. In *31st Aerospace Sciences Meeting*, p. 217.
- BOUKHARFANE, R., BOUALI, Z. & MURA, A. 2018 Evolution of scalar and velocity dynamics in planar shock-turbulence interaction. *Shock Waves* **28** (6), 1117–1141.
- BROWN, G. L. & ROSHKO, A. 1974 On density effects and large structure in turbulent mixing layers. *J. Fluid Mech.* **64** (4), 775–816.
- BUDZINSKI, J., ZUKOSKI, E. & MARBLE, F. 1992 Rayleigh scattering measurements of shock enhanced mixing. In *28th Joint Propulsion Conference and Exhibit*, p. 3546.
- CABAN, J., JOSHI, A. & RHEINGANS, P. 2007 Texture-based feature tracking for effective time-varying data visualization. *IEEE Trans Vis Comput Graph* **13** (6), 1472–1479.
- CAMBON, C., COLEMAN, G. & MANSOUR, N. 1992 Rapid distortion analysis and direct simulation of compressible homogeneous turbulence at finite mach number .

- CHACÍN, J. M., CANTWELL, B. J. & KLINE, S. J. 1996 Study of turbulent boundary layer structure using the invariants of the velocity gradient tensor. *Exp. Therm. Fluid Sci.* **13** (4), 308–317.
- CHAN, W., DODD, M., JOHNSON, P., URZAY, J. & MOIN, P. 2018 Formation and dynamics of bubbles in breaking waves: Part i. algorithms for the identification of bubbles and breakup/coalescence events .
- CHEN, J., SILVER, D. & JIANG, L. 2003 The feature tree: Visualizing feature tracking in distributed amr datasets. In *IEEE Symposium on Parallel and Large-Data Visualization and Graphics, 2003. PVG 2003.*, pp. 103–110. IEEE.
- CHONG, M. S., PERRY, A. E. & CANTWELL, B. J. 1990 A general classification of three-dimensional flow fields. *Phys Fluid Fluid Dynam* **2** (5), 765–777.
- CLEMENS, N. & MUNGAL, M. 1995 Large-scale structure and entrainment in the supersonic mixing layer. *J. Fluid Mech.* **284**, 171–216.
- CLYNE, J., MININNI, P. & NORTON, A. 2012 Physically-based feature tracking for cfd data. *IEEE Trans Vis Comput Graph* **19** (6), 1020–1033.
- COOK, D. J. & HOLDER, L. B. 1993 Substructure discovery using minimum description length and background knowledge. *J Artif Intell Res* **1**, 231–255.
- COOK, D. J. & HOLDER, L. B. 2006 *Mining graph data*. John Wiley & Sons.
- DALLMANN, U. 1983 Topological structures of three-dimensional vortex flow separation. In *16th Fluid and Plasma-dynamics Conference*, p. 1735.
- DANISH, M., SUMAN, S. & GIRIMAJI, S. S. 2016 Influence of flow topology and dilatation on scalar mixing in compressible turbulence. *J. Fluid Mech.* **793**, 633–655.
- DEL ALAMO, J. C., JIMENEZ, J., ZANDONADE, P. & MOSER, R. D. 2006 Self-similar vortex clusters in the turbulent logarithmic region. *J. Fluid Mech.* **561**, 329–358.
- DEMANET, L. & YING, L. 2007 Curvelets and wave atoms for mirror-extended images. In *Wavelets XII*, , vol. 6701, p. 67010J. International Society for Optics and Photonics.
- DONG, S., LOZANO-DURÁN, A., SEKIMOTO, A. & JIMÉNEZ, J. 2017 Coherent structures in statistically stationary homogeneous shear turbulence. *J. Fluid Mech.* **816**, 167–208.
- DONZIS, D. A., SREENIVASAN, K. & YEUNG, P. 2010 The batchelor spectrum for mixing of passive scalars in isotropic turbulence. *Flow Turb. Combust.* **85** (3), 549–566.
- DUCROS, F., LAPORTE, F., SOULÈRES, T., GUINOT, V., MOINAT, P. & CARUELLE, B. 2000 High-order fluxes for conservative skew-symmetric-like schemes in structured meshes: application to compressible flows. *J. Comp. Phys.* **161** (1), 114–139.

- EITEL-AMOR, G., ÖRLÜ, R., SCHLATTER, P. & FLORES, O. 2015 Hairpin vortices in turbulent boundary layers. *Phys. Fluids* **27** (2), 025108.
- EPPS, B. 2017 Review of vortex identification methods. In *55th AIAA aerospace sciences meeting*, p. 0989.
- ESTER, M., KRIEGEL, H.-P., SANDER, J., XU, X. & OTHERS 1996 A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, , vol. 96, pp. 226–231.
- ETEMADI, N. 1990 On curve and surface stretching in isotropic turbulent flow. *J. Fluid Mech.* **221**, 685–692.
- FOYSI, H. & SARKAR, S. 2010 The compressible mixing layer: an les study. *Theor Comput Fluid Dyn* **24** (6), 565–588.
- FREUND, J. B., LELE, S. K. & MOIN, P. 2000 Compressibility effects in a turbulent annular mixing layer. part 1. turbulence and growth rate. *J. Fluid Mech.* **421**, 229–267.
- FRIEDMAN, J. H., BENTLEY, J. L. & FINKEL, R. A. 1977 An algorithm for finding best matches in logarithmic expected time. *ACM Trans Math Softw (TOMS)* **3** (3), 209–226.
- FRIEDRICH, R. & BERTOLOTTI, F. 1996 Compressibility effects due to turbulent fluctuations. *Applied scientific research* **57** (3-4), 165–194.
- FU, S. & LI, Q. 2006 Numerical simulation of compressible mixing layers. *Int. J. Heat Fluid Flow* **27** (5), 895–901.
- GAO, X., BERMEJO-MORENO, I. & LARSSON, J. 2020 Parametric numerical study of passive scalar mixing in shock turbulence interaction. *J. Fluid Mech.* **895**.
- GILBERT, E. G., JOHNSON, D. W. & KEERTHI, S. S. 1988 A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Robot Autom Mag* **4** (2), 193–203.
- GIRIMAJI, S. & POPE, S. 1990 Material-element deformation in isotropic turbulence. *J. Fluid Mech.* **220**, 427–458.
- GIRIMAJI, S. & POPE, S. 1992 Propagating surfaces in isotropic turbulence. *J. Fluid Mech.* **234**, 247–277.
- GIRVAN, M. & NEWMAN, M. E. 2002 Community structure in social and biological networks. *Proc. Natl. Acad. Sci. U.S.A* **99** (12), 7821–7826.
- GOTO, S. & KIDA, S. 2007 Reynolds-number dependence of line and surface stretching in turbulence: folding effects. *J. Fluid Mech.* **586**, 59.
- GOTTSCHALK, S., LIN, M. C. & MANOCHA, D. 1996 Obbtree: A hierarchical structure for rapid interference detection. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 171–180.

- GUO, H., PHILLIPS, C. L., PETERKA, T., KARPEYEV, D. & GLATZ, A. 2015 Extracting, tracking, and visualizing magnetic flux vortices in 3d complex-valued superconductor simulation data. *IEEE Trans Vis Comput Graph* **22** (1), 827–836.
- HARIKRISHNAN, A., ANSORGE, C., KLEIN, R. & VERCAUTEREN, N. 2021 Geometry and organization of coherent structures in stably stratified atmospheric boundary layers. *arXiv preprint arXiv:2110.02253* .
- HESSELINK, L. & STURTEVANT, B. 1988 Propagation of weak shocks through a random medium. *J. Fluid Mech.* **196**, 513–553.
- HOLDER, L. B., COOK, D. J., DJOKO, S. & OTHERS 1994 Substructure discovery in the subdue system. In *KDD workshop*, pp. 169–180. Washington, DC, USA.
- HUNT, J., WRAY, A. & MOIN, P. 1988 Eddies, stream, and convergence zones in turbulent flows. *Proceedings of the Summer Program (Center for Turbulence Research)* pp. 193–208.
- HUSSAIN, A. F. 1986 Coherent structures and turbulence. *J. Fluid Mech.* **173**, 303–356.
- JEONG, J. & HUSSAIN, F. 1995 On the identification of a vortex. *J. Fluid Mech.* **285** (-1), 69.
- JI, G. & SHEN, H.-W. 2004 Efficient isosurface tracking using precomputed correspondence table. In *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization*, pp. 283–292. Eurographics Association.
- JI, G. & SHEN, H.-W. 2006 Feature tracking using earth mover’s distance and global optimization. In *Pacific graphics*, , vol. 2.
- JI, G., SHEN, H.-W. & WENGER, R. 2003 Volume tracking using higher dimensional isosurfacing. In *IEEE Visualization, 2003. VIS 2003.*, pp. 209–216. IEEE.
- JIMÉNEZ, J. 2013 Near-wall turbulence. *Phys. Fluids* **25** (10), 101302.
- JOHN, J. P., DONZIS, D. A. & SREENIVASAN, K. R. 2019 Solenoidal scaling laws for compressible mixing. *Phys. Rev. Lett.* **123** (22), 224501.
- KARASSO, P. & MUNGAL, M. 1996 Scalar mixing and reaction in plane liquid shear layers. *J. Fluid Mech.* **323**, 23–63.
- KAREEM, W. A., IZAWA, S., XIONG, A. & FUKUNISHI, Y. 2007 Extraction and tracking of multi-scaled vortices from a homogeneous isotropic turbulent field. *J. Turb.* (8), N12.
- KERR, R. M. 1985 Higher-order derivative correlations and the alignment of small-scale structures in isotropic numerical turbulence. *J. Fluid Mech.* **153**, 31–58.
- KETKAR, N. S., HOLDER, L. B. & COOK, D. J. 2005 Subdue: Compression-based frequent pattern discovery in graph data. In *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, pp. 71–76.

- KIM, S. H. & PITTSCH, H. 2007 Scalar gradient and small-scale structure in turbulent premixed combustion. *Phys. Fluids* **19** (11), 115104.
- KLINE, S. J., REYNOLDS, W. C., SCHRAUB, F. A. & RUNSTADLER, P. W. 1967 The structure of turbulent boundary layers. *J. Fluid Mech.* **30** (4), 741–773.
- KOENDERINK, J. J. & VAN DOORN, A. J. 1992 Surface shape and curvature scales. *Image Vis Comput* **10** (8), 557–564.
- LARSSON, J. 2009 Blending technique for compressible inflow turbulence: algorithm localization and accuracy assessment. *J. Comput. Phys.* **228** (4), 933–937.
- LARSSON, J., BERMEJO-MORENO, I. & LELE, S. K. 2013 Reynolds-and mach-number effects in canonical shock-turbulence interaction. *J. Fluid Mech.* **717**, 293.
- LARSSON, J. & LELE, S. K. 2009 Direct numerical simulation of canonical shock/turbulence interaction. *Phys. Fluids* **21** (12), 126101.
- LEE, S., LELE, S. K. & MOIN, P. 1993 Direct numerical simulation of isotropic turbulence interacting with a weak shock wave. *J. Fluid Mech.* **251**, 533–562.
- LEE, S., LELE, S. K. & MOIN, P. 1994 Corrigendum: direct numerical simulation of isotropic turbulence interacting with a weak shock wave. *J. Fluid Mech.* **264**, 373–374.
- LEE, S., LELE, S. K. & MOIN, P. 1997 Interaction of isotropic turbulence with shock waves: effect of shock strength. *J. Fluid Mech.* **340**, 225–247.
- LESIEUR, M., MÉTAIS, O., COMTE, P. & OTHERS 2005 *Large-eddy simulations of turbulence*. Cambridge university press.
- LEUNG, T., SWAMINATHAN, N. & DAVIDSON, P. 2012 Geometry and interaction of structures in homogeneous isotropic turbulence. *J. Fluid Mech.* **710**, 453–481.
- LI, Q. & FU, S. 2003 Numerical simulation of high-speed planar mixing layer. *Comput. Fluids* **32** (10), 1357–1377.
- LIU, T., MOORE, A. W., YANG, K. & GRAY, A. G. 2005 An investigation of practical approximate nearest neighbor algorithms. In *Advances in neural information processing systems*, pp. 825–832.
- LIVESCU, D. & RYU, J. 2016 Vorticity dynamics after the shock–turbulence interaction. *Shock Waves* **26** (3), 241–251.
- LORENSEN, W. E. & CLINE, H. E. 1987 Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph comput. graph.* **21** (4), 163–169.

- LOZANO-DURÁN, A. & JIMÉNEZ, J. 2014 Time-resolved evolution of coherent structures in turbulent channels: characterization of eddies and cascades. *J. Fluid Mech.* **759**, 432–471.
- LUNDGREN, T. S. 1982 Strained spiral vortex model for turbulent fine structure. *Phys. Fluids* **25** (12), 2193–2203.
- MANI, A. 2012 Analysis and optimization of numerical sponge layers as a nonreflective boundary treatment. *J. Comput. Phys.* **231** (2), 704–716.
- MAURUS, S. & PLANT, C. 2016 Skinny-dip: clustering in a sea of noise. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1055–1064.
- MEHTA, R. 1991 Effect of velocity ratio on plane mixing layer development: Influence of the splitter plate wake. *Exp. Fluids*. **10** (4), 194–204.
- MISRA, A. & PULLIN, D. I. 1997 A vortex-based subgrid stress model for large-eddy simulation. *Phys. Fluids* **9** (8), 2443–2454.
- MONTANARI, M., PETRINIC, N. & BARBIERI, E. 2017 Improving the gjk algorithm for faster and more reliable distance queries between convex objects. *ACM Trans. Graph. (TOG)* **36** (3), 1–17.
- MOSER, R. & ROGERS, M. 1991 Mixing transition and the cascade to small scales in a plane mixing layer. *Phys Fluid Fluid Dynam* **3** (5), 1128–1134.
- MOSER, R. D. & ROGERS, M. M. 1993 The three-dimensional evolution of a plane mixing layer: pairing and transition to turbulence. *J. Fluid Mech.* **247**, 275–320.
- MUELDER, C. & MA, K.-L. 2009 Interactive feature extraction and tracking by utilizing region coherency. In *2009 IEEE Pacific Visualization Symposium*, pp. 17–24. IEEE.
- NI, Q. 2015 Compressible turbulent mixing: Effects of schmidt number. *Phys. Rev. E* **91** (5), 053020.
- NI, Q. 2016 Compressible turbulent mixing: Effects of compressibility. *Phys. Rev. E* **93** (4), 043116.
- PAN, L. & SCANNAPIECO, E. 2010 Mixing in supersonic turbulence. *Astrophys. J.* **721** (2), 1765.
- PAN, L. & SCANNAPIECO, E. 2011 Passive scalar structures in supersonic turbulence. *Phys. Rev. E* **83** (4), 045302.
- PANICKACHERIL JOHN, J., DONZIS, D. A. & SREENIVASAN, K. R. 2020 Compressibility effects on the scalar dissipation rate. *Combust. Sci. Technol.* **192** (7), 1320–1333.
- PANTANO, C. & SARKAR, S. 2002 A study of compressibility effects in the high-speed turbulent shear layer using direct simulation. *J. Fluid Mech.* **451**, 329.
- PAPAMOSCHOU, D. & LELE, S. K. 1993 Vortex-induced disturbance field in a compressible shear layer. *Phys Fluid Fluid Dynam* **5** (6), 1412–1419.

- PARASHAR, N., SINHA, S. S., DANISH, M. & SRINIVASAN, B. 2017 Lagrangian investigations of vorticity dynamics in compressible turbulence. *Phys. Fluids* **29** (10), 105110.
- PEIKERT, R. & ROTH, M. 1999 The” parallel vectors” operator-a vector field visualization primitive. In *Proceedings Visualization’99 (Cat. No. 99CB37067)*, pp. 263–532. Ieee.
- PITSCH, H. & STEINER, H. 2000 Scalar mixing and dissipation rate in large-eddy simulations of non-premixed turbulent combustion. *Proc. Combust. Inst.* **28** (1), 41–49.
- POST, F. H., VROLIJK, B., HAUSER, H., LARAMEE, R. S. & DOLEISCH, H. 2002 Feature extraction and visualisation of flow fields. In *Eurographics (STARs)*.
- PULLIN, D. I. 2000 A vortex-based model for the subgrid flux of a passive scalar. *Phys. Fluids* **12** (9), 2311–2319.
- PULLIN, D. I. & SAFFMAN, P. G. 1993 On the Lundgren–Townsend model of turbulent fine scales. *Phys. Fluids A: Fluid Dyn.* **5** (1), 126–145.
- REINDERS, F., POST, F. H. & SPOELDER, H. J. 1999 Attribute-based feature tracking. In *Data visualization’99*, pp. 63–72. Springer.
- REINDERS, K. F., FREDERIK, K. & REINDERS, J. 2001 Feature-based visualization of time-dependent data .
- RISSANEN, J. 1998 *Stochastic complexity in statistical inquiry*, , vol. 15. World scientific.
- RISTORCELLI, J. R. & BLAISDELL, G. A. 1997 Consistent initial conditions for the dns of compressible turbulence. *Phys. Fluids* **9** (1), 4–6.
- ROBINSON, S. K. 1991 Coherent motions in the turbulent boundary layer. *Annu. Rev. Fluid Mech.* **23** (1), 601–639.
- ROGERS, M. M. & MOIN, P. 1987 The structure of the vorticity field in homogeneous turbulent flows. *J. Fluid Mech* **176**, 33–66.
- ROGERS, M. M. & MOSER, R. D. 1992 The three-dimensional evolution of a plane mixing layer: the kelvin–helmholtz rollup. *J. Fluid Mech.* **243**, 183–226.
- ROGERS, M. M. & MOSER, R. D. 1994 Direct simulation of a self-similar turbulent mixing layer. *Phys. Fluids* **6** (2), 903–923.
- RYU, J. & LIVESCU, D. 2014 Turbulence structure behind the shock in canonical shock–vortical turbulence interaction. *J. Fluid Mech.* **756**.
- SAIKIA, H. & WEINKAUF, T. 2017 Global feature tracking and similarity estimation in time-dependent scalar fields. In *Computer Graphics Forum*, , vol. 36, pp. 1–11. Wiley Online Library.
- SAMTANEY, R., SILVER, D., ZABUSKY, N. & CAO, J. 1994 Visualizing features and tracking their evolution. *Computer* **27** (7), 20–27.

- SANDHAM, N. & REYNOLDS, W. 1990 Compressible mixing layer-linear theory and direct simulation. *AIAA J.* **28** (4), 618–624.
- SARKAR, S. 1995 The stabilizing effect of compressibility in turbulent shear flow. *J. Fluid Mech.* **282**, 163–186.
- SAUER, F., YU, H. & MA, K.-L. 2014 Trajectory-based flow feature tracking in joint particle/volume datasets. *IEEE Trans Vis Comput Graph* **20** (12), 2565–2574.
- SCHAFHITZEL, T., BAYSAL, K., RIST, U., WEISKOPF, D. & ERTL, T. 2008 Particle-based vortex core line tracking taking into account vortex dynamics. In *Proceedings International Symposium on Flow Visualization*, , vol. 8.
- SCHLATTER, P., LI, Q., ÖRLÜ, R., HUSSAIN, F. & HENNINGSON, D. S. 2014 On the near-wall vortical structures at moderate Reynolds numbers. *Eur. J. Mech.-B/Fluids* **48**, 75–93.
- SCHUBERT, E., SANDER, J., ESTER, M., KRIEGEL, H. P. & XU, X. 2017 DbSCAN revisited, revisited: why and how you should (still) use dbSCAN. *ACM Trans. Database Syst. (TODS)* **42** (3), 1–21.
- SILVER, D. 1995 Object-oriented visualization. *IEEE Comput Graph Appl* **15** (3), 54–62.
- SILVER, D. & WANG, X. 1996 Volume tracking. In *Proceedings of Seventh Annual IEEE Visualization'96*, pp. 157–164. IEEE.
- SILVER, D. & WANG, X. 1997 Tracking and visualizing turbulent 3d features. *IEEE Trans Vis Comput Graph* **3** (2), 129–141.
- SILVER, D. & WANG, X. 1998 Tracking scalar features in unstructured data sets. In *Proceedings Visualization'98 (Cat. No. 98CB36276)*, pp. 79–86. IEEE.
- SMITH, C. R. & METZLER, S. P. 1983 The characteristics of low-speed streaks in the near-wall region of a turbulent boundary layer. *J. Fluid Mech.* **129**, 27–54.
- SOHN, B.-S. & BAJAJ, C. 2005 Time-varying contour topology. *IEEE Trans Vis Comput Graph* **12** (1), 14–25.
- SOLER, M., PLAINCHAULT, M., CONCHE, B. & TIERNY, J. 2018 Lifted wasserstein matcher for fast and robust topology tracking. *arXiv preprint arXiv:1808.05870* .
- SUMAN, S. & GIRIMAJI, S. S. 2010 Velocity gradient invariants and local flow-field topology in compressible turbulence. *J. Turbul.* (11), N2.
- SWAMINATHAN, N. & GROUT, R. 2006 Interaction of turbulence and scalar fields in premixed flames. *Phys. Fluids* **18** (4), 045102.
- TENNEKES, H. 1968 Simple model for the small-scale structure of turbulence. *Phys. Fluids* **11** (3), 669–671.

- THEISEL, H., SAHNER, J., WEINKAUF, T., HEGE, H.-C. & SEIDEL, H.-P. 2005 Extraction of parallel vector surfaces in 3d time-dependent fields and application to vortex core line tracking. In *VIS 05. IEEE Visualization, 2005.*, pp. 631–638. IEEE.
- THEISEL, H. & SEIDEL, H.-P. 2003 Feature flow fields. In *Proceedings of the symposium on Data visualisation 2003*, pp. 141–148. Eurographics Association.
- THEODORSEN, T. 1952 Mechanisms of turbulence. In *Proceedings of the 2nd Midwest Conference on Fluid Mechanics, 1952.*
- THEODORSEN, T. 1955 The structure of turbulence. In *50 Jahre Grenzschichtforschung*, pp. 55–62. Springer.
- TIAN, Y., JABERI, F. A., LI, Z. & LIVESCU, D. 2017 Numerical study of variable density turbulence interaction with a normal shock wave. *J. Fluid Mech.* **829**, 551–588.
- TOMKINS, C. D. & ADRIAN, R. J. 2003 Spanwise structure and scale growth in turbulent boundary layers. *J. Fluid Mech.* **490**, 37–74.
- TOWNSEND, A. 1980 *The structure of turbulent shear flow*. Cambridge university press.
- TOWNSEND, A. A. 1951 On the fine-scale structure of turbulence. *Proc. Math. Phys. Sci.* **208** (1095), 534–542.
- TOWNSEND, A. A. 1961 Equilibrium layers and wall turbulence. *J. Fluid Mech.* **11** (1), 97–120.
- URZAY, J. 2018 Supersonic combustion in air-breathing propulsion systems for hypersonic flight. *Annu. Rev. Fluid Mech.* **50**, 593–627.
- VAN DEN BERGEN, G. 2001 Proximity queries and penetration depth computation on 3d game objects. In *Game developers conference*, , vol. 170.
- VAN WALSUM, T., POST, F. H., SILVER, D. & POST, F. J. 1996 Feature extraction and iconic visualization. *IEEE Trans Vis Comput Graph* **2** (2), 111–119.
- VANDERWEL, C. & TAVOULARIS, S. 2011 Coherent structures in uniformly sheared turbulent flow. *J. Fluid Mech.* **689**, 434–464.
- VEDULA, P., YEUNG, P. & FOX, R. O. 2001 Dynamics of scalar dissipation in isotropic turbulence: a numerical and modelling study. *J. Fluid Mech.* **433**, 29.
- VILLERMAUX, E. 2019 Mixing versus stirring. *Ann. Rev. Fluid Mech.* **51**, 245–273.
- VOLLMERS, H., KREPLIN, H. & MEIER, H. 1983 Separation and vortical-type flow around a prolate spheroid-evaluation of relevant parameters. *Tech. Rep.*. DEUTSCHE FORSCHUNGS-UND VERSUCHSANSTALT FUER LUFT-UND RAUMFAHRT EV

- VREMAN, B., KUERTEN, H. & GEURTS, B. 1995 Shocks in direct numerical simulation of the confined three-dimensional mixing layer. *Phys. Fluids* **7** (9), 2105–2107.
- WADELL, H. 1935 Volume, shape, and roundness of quartz particles. *J. Geol.* **43** (3), 250–280.
- WEIGLE, C. & BANKS, D. C. 1998 Extracting iso-valued features in 4-dimensional scalar fields. In *IEEE Symposium on Volume Visualization (Cat. No. 989EX300)*, pp. 103–110. IEEE.
- WIDANAGAMAACHCHI, W., CHEN, J., KLACANSKY, P., PASCUCCI, V., KOLLA, H., BHAGATWALA, A. & BREMER, P.-T. 2015 Tracking features in embedded surfaces: Understanding extinction in turbulent combustion. In *2015 IEEE 5th Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 9–16. IEEE.
- WIDANAGAMAACHCHI, W., CHRISTENSEN, C., PASCUCCI, V. & BREMER, P.-T. 2012 Interactive exploration of large-scale time-varying data using dynamic tracking graphs. In *IEEE symposium on large data analysis and visualization (LDAV)*, pp. 9–17. IEEE.
- WU, X. & MOIN, P. 2009 Direct numerical simulation of turbulence in a nominally zero-pressure-gradient flat-plate boundary layer. *J. Fluid Mech.* **630**, 5–41.
- YAN, X. & HAN, J. 2002 gspan: Graph-based substructure pattern mining. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pp. 721–724. IEEE.
- YANG, Y., PULLIN, D. & BERMEJO-MORENO, I. 2010 Multi-scale geometric analysis of lagrangian structures in isotropic turbulence. *J. Fluid Mech.* **654**, 233–270.
- YIANILOS, P. N. 1993 Data structures and algorithms for nearest neighbor search in general metric spaces.
- ZHANG, D., TAN, J. & YAO, X. 2019 Direct numerical simulation of spatially developing highly compressible mixing layer: Structural evolution and turbulent statistics. *Phys. Fluids* **31** (3), 036102.
- ZHANG, Y., LIU, K., XIAN, H. & DU, X. 2018 A review of methods for vortex identification in hydroturbines. *Renew. Sust. Energ. Rev.* **81**, 1269–1285.
- ZHOU, J., ADRIAN, R. J., BALACHANDAR, S. & KENDALL, T. 1999 Mechanisms for generating coherent packets of hairpin vortices in channel flow. *J. Fluid Mech.* **387**, 353–396.
- ZHOU, Q., HE, F. & SHEN, M. 2012 Direct numerical simulation of a spatially developing compressible plane mixing layer: flow structures and mean flow properties. *J. Fluid Mech.* **711**, 437.