DISSERTATION

The Visible Vortex –

Interactive Analysis and Extraction of Vortices in Large Time-dependent Flow Data Sets

---

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften

unter der Leitung von

Priv.-Doz. Dipl.-Ing. Dr.techn. Helwig Hauser,
Institut E186 für Computergraphik und Algorithmen,

eingereicht an der Technischen Universität Wien,
Fakultät für Informatik,

von

Dipl.-Inf. Raphael Fuchs,
Matrikelnummer 0727622,
Harkortstr. 7/3,
A-1020 Wien

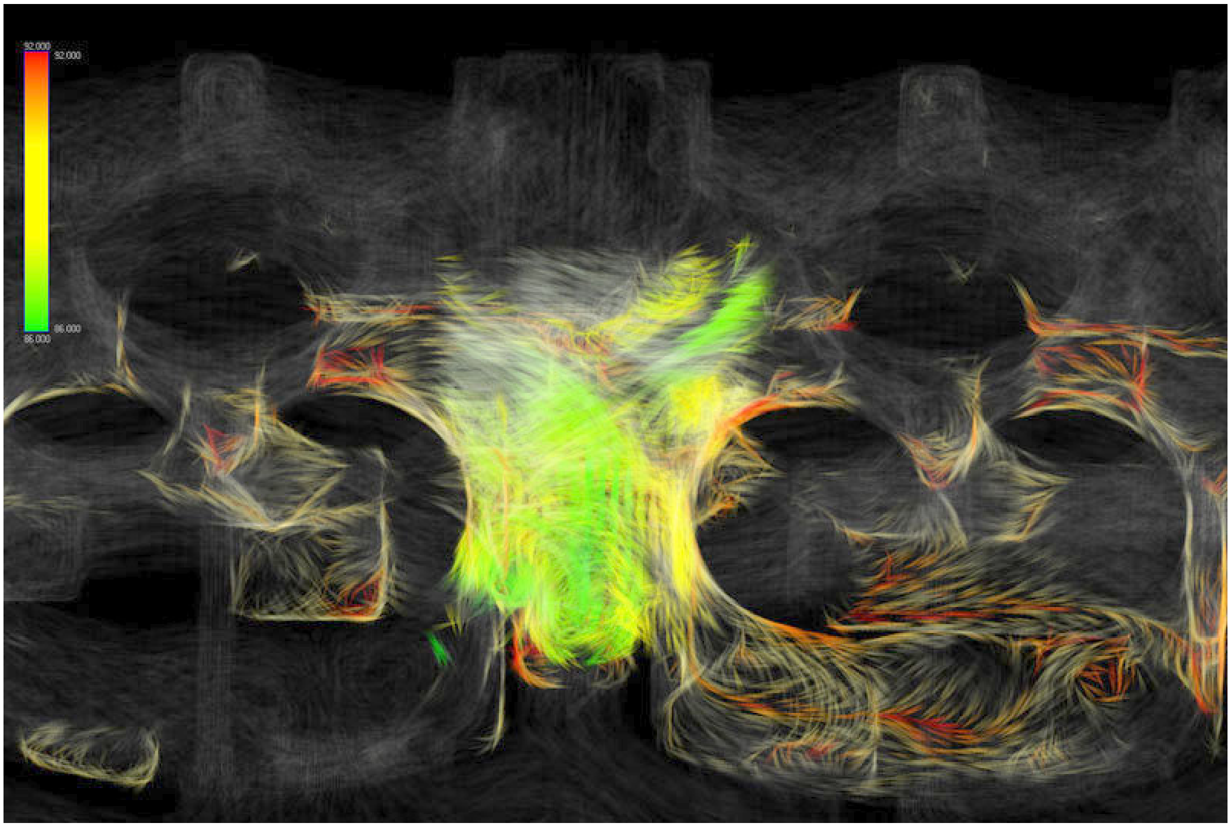
Wien, im September 2008

delivered by
EUROGRAPHICS
DIGITAL LIBRARY
www.eg.org
diglib.eg.org

The Visible Vortex –

Interactive Analysis and Extraction of Vortices in Large Time-dependent Flow Data Sets

Raphael Fuchs, PhD thesis



mailto:raphael.fuchs@VRVis.at
http://www.cg.tuwien.ac.at/~raphael/

Abstract

Computational simulation of physical and chemical processes has become an essential tool to tackle questions from the field of fluid dynamics. Using current simulation packages it is possible to compute unsteady flow simulations for realistic scenarios. The resulting solutions are stored in large to very large grids in 2D or 3D, frequently time-dependent, with multi-variate results from the numeric simulation. With increasing complexity of simulation results, powerful analysis and visualization tools are needed to make sense of the computed information and answer the question at hand. To do this we need new approaches and algorithms to locate regions of interest, find important structures in the flow and analyze the behavior of the flow interactively.

The main motives of this thesis are the extension of vortex detection criteria to unsteady flow and the combination of vortex detectors with interactive visual analysis. To develop an understanding for the simulation results it is necessary to compare attributes of the simulation to each other and to be able to relate them to larger structures such as vortices. It is shown how automatic feature detection algorithms can be combined with interactive analysis techniques such that both detection and analysis benefit.

By extending and integrating vortex detectors into the process of visual analysis, it becomes possible to understand the impact of vortex structures on the development of the flow. Using real-world examples from the field of engine design we discuss how vortex structures can have critical impact on the performance of a prototype. We illustrate how interactive visual analysis can support prototype design and evaluation. Furthermore, we show that taking the unsteady nature of the flow into account improves the quality of the extracted structures.

Kurzfassung

Computersimulation physikalischer und chemischer Prozesse sind ein essentielles Hilfsmittel zum Verständnis von Problemen aus dem Bereich der Strömungslehre geworden. Mit aktuellen Simulationspaketen ist es möglich für realistische Szenarien zeitabhängige Lösungen zu berechnen. Die berechneten Lösungen werden in großen Gitternetzen gespeichert, sind häufig zeitabhängig und enthalten die multivariaten Ergebnisse der numerischen Simulation. Mit zunehmender Komplexität der Simulationsergebnisse entsteht die Notwendigkeit geeignete Analyse- und Darstellungswerkzeuge zu verwenden, um aus den erzeugten Daten

Erkenntnisse zu gewinnen und die gegebene Fragestellung lösen zu können. Dafür sind neue Methoden und Algorithmen notwendig um wichtige Teile der Daten zu extrahieren, wichtige Strukturen in der Strömung zu erkennen und das Strömungsverhalten interaktiv analysieren zu können.

Das Hauptmotiv dieser Arbeit ist die Erweiterung von Wirbelkriterien auf zeitabhängige Lösungen und die Kombination dieser Wirbelkriterien mit den Methoden der interaktiven visuellen Analyse. Um ein Verständnis für die Simulationsergebnisse zu entwickeln ist es notwendig Attribute der Simulation miteinander vergleichen und mit größeren Strukturen wie Strömungswirbeln in Verbindung bringen zu können. Es wird gezeigt, wie automatische Wirbeldetektoren und interaktive Analyse kombiniert werden um sowohl Erkennung als auch Analyse von Wirbelstrukturen zu verbessern.

Indem klassische Wirbeldetektoren erweitert und in die visuelle Analyse integriert werden, ist es möglich die Auswirkung von Wirbeln auf wichtige Attribute der Flüssigkeit und die Entwicklung der Strömung zu verstehen. Wir zeigen anhand von praktischen Beispielen aus dem Bereich der Motorentwicklung, dass Wirbel einen entscheidenden Einfluss auf anwendungskritische Variablen der Strömung haben können und demonstrieren wie interaktive visuelle Analyse helfen kann diesen Einfluss einzuschätzen. Weiterhin besprechen wir wie zeitabhängige Größen in die Wirbelbestimmung einfließen können um die Qualität der gefundenen Merkmale zu verbessern.

Contents

Abstract, Kurzfassung **iii**

Related Publications **ix**

1 Introduction and Overview **1**
1.1 Vortex Flow 1
1.2 Interactive Visual Analysis 2
1.3 Contribution of this Work 5
1.4 Organization of this Thesis 5

2 State of the Art in Visualization of Scientific Data **7**
2.1 Motivation 7
2.2 Dealing with scalar attributes 12
  2.2.1 Techniques in the processing, filtering and visualization mapping stage 12
  2.2.2 Rendering stage techniques 15
  2.2.3 Image stage techniques 15
2.3 Vector Field and Flow Visualization 16
  2.3.1 Techniques in the processing, filtering and visualization mapping stage 17
  2.3.2 Rendering stage techniques 18
  2.3.3 Image stage techniques 21
2.4 Tensor field visualization 22
  2.4.1 Techniques in the processing, filtering and visualization mapping stage 23
  2.4.2 Rendering stage techniques 25
  2.4.3 Image stage techniques 26
2.5 General approaches to multi-dimensional visualization 27
2.6 Illustrative rendering 31
2.7 Chapter Conclusions 33

3 Binary and Non-Binary Vortex Detectors **37**
3.1 Motivation 37
3.2 Background on Vortex Detection 38
3.3 Non-binary vortex detectors 43
  3.3.1 Derivation of the Continuous Detectors 44

3.3.2 Integration in the interactive framework 47
3.4 Application Evaluation – a Cooling Jacket 50
3.4.1 Reduced heat transport due to turbulent motion 50
3.4.2 A situation of improved heat transport due to turbulent motion 52
3.5 Chapter Conclusions 52

4 Cross-Detector Analysis 55
4.1 Motivation 55
4.2 On Visual Interactive Analysis 56
4.3 PQR Plots 58
4.4 Correlation Brushing 60
4.5 Multiple Detector Views 62
4.6 Evaluation of the Multi-view approach 64
4.6.1 Two-Stroke Engine 65
4.6.2 Cooling Jacket 66
4.7 Chapter Conclusions 68

5 Delocalized Detectors 69
5.1 Motivation 69
5.2 Delocalized Vortex Detectors 71
5.2.1 New Criterion 71
5.2.2 Line View 73
5.3 Evaluation 75
5.3.1 Value distribution in real-world data sets 75
5.3.2 Cooling Jacket 76
5.3.3 Two-stroke Engine 78
5.3.4 T-Junction 78
5.3.5 Filter Properties 82
5.4 Implementation Details 82

6 Unsteady Parallel Vectors 85
6.1 Motivation 85
6.2 Analytic Considerations 87
6.2.1 A Tilting Vortex 87
6.2.2 A Rotating Vortex Rope 89
6.3 Pathline Based Feature Detectors 92
6.3.1 Sujudi-Haimes 92
6.3.2 Higher Order Vortex Core Lines 94
6.3.3 Interactive Vortex Core Line Extraction and Filtering 95
6.3.4 Pseudocode 98
6.4 Application Study – Engine Data Sets 99
6.4.1 Impact of Time-derivatives 102
6.4.2 Equivalence Ratio 103

6.5 Assessment of Numerical Behavior 104
6.6 Chapter Conclusions 106

7 Summary 107
7.1 Unsteady Criteria for Vortex Core Lines 107
7.2 Interactive Vortex Region Analysis 109
    7.2.1 Step 1: Non-binary Criteria 109
    7.2.2 Step 2: Multiple combined Criteria 111
    7.2.3 Step 3: Delocalized Criteria 112
7.3 Chapter Conclusions 113

8 Thesis Conclusions 115

A Gradients 117
A.1 Velocity Estimation 118
A.2 Gradient Reconstruction 119
    A.2.1 Green-Gauss Linear Reconstruction 119
    A.2.2 Least-Squares Linear Reconstruction 120

Acknowledgments 121

Curriculum Vitae 122

Bibliography 125

Related Publications

This thesis is based on the following publications:

Raphael Bürger[1] and Helwig Hauser,
STAR: Visualization of Multi-variate Scientific Data,
In EuroGraphics 2007 State of the Art Reports (STARs), pages 117-134, 2007.

Raphael Bürger[1], Philipp Muigg, Martin Ilčík, Helmut Doleisch and Helwig Hauser,
Integrating Local Feature Detectors in the Interactive Visual Analysis of Flow Simulation Data,
In Proceedings of Eurographics/ IEEE-VGTC Symposium on Visualization 2007 (EuroVis), pages 171-178, 2007.

Raphael Bürger[1], Philipp Muigg, Helmut Doleisch, and Helwig Hauser,
Interactive cross-detector analysis of vortical flow data,
In Proceedings of Coordinated and Multiple Views in Exploratory Visualization 2007 (CMV), pages 98-110, 2007.

Raphael Fuchs, Ronald Peikert, Helwig Hauser, Filip Sadlo, and Philipp Muigg
Parallel Vectors Criteria for Unsteady Flow Vortices,
IEEE Transactions on Visualization and Computer Graphics (TVCG), 14(3), pages 615-626, 2008.

Raphael Fuchs, Ronald Peikert, Filip Sadlo, Bilal Alsallakh, and Meister Eduard Gröller
Delocalized Unsteady Vortex Region Detectors,
In Proceedings of Vision, Modelling, and Visualization Workshop 2008 (VMV), 2008.

---

[1] now Raphael Fuchs

Chapter 1

Introduction and Overview

> "A beginning is the time for taking the most delicate care that the balances are correct." (Frank Herbert 1920 – 1986)

This thesis deals with interactive visual analysis of vortices in computational fluid dynamics simulations. We present novel methods to extract, display and analyze vortical fluid structures to help engineers and researches deal with the growing complexity of modern CFD simulations. The introduction is split into three parts: Section 1.1 discusses vortex features in fluid simulation data sets. Section 1.2 discusses the workflow and basic concepts of interactive visual analysis. Section 1.3 details contributions of the presented work.

1.1 Vortex Flow

Fluids can show complex three dimensional movements where individual particle paths are entangled to form intricate spatial structures. Such a volume of complex fluid movements is difficult to display in a single image or video without loss of much of the information. Still, as long as the fluid is not chaotic and only statistical properties of the fluid are reliable, visualizations can give the engineer or researcher valuable insight into simulation results. Thus, visualizations are a tool to help engineers and scientists understand the behavior of a fluid. Engineered objects such as turbines or engines are designed to a purpose and within operational parameters the overall movements of the fluid follow a course intended by the designer. But even inside operational parameters, the behavior of the fluid is complex and to improve their designs engineers and researchers want to refine their understanding of a given problem at hand. The key to optimization of industrial designs is understanding the behavior of the flow. During the work of this thesis and in cooperation with industrial partners it has become clear that understanding what a simulation result means can be achieved best in a process where the engineer can interactively analyze the data and produce new views on the data as his or her understanding develops.

Computer graphics can allow insight into large data sets but it is also widely accepted that due to the rapid increase in the size and complexity of the data sets produced with current simulations or acquisition devices we would like to present simplified information that helps to understand general properties of the simulation results quickly. From fluid dynamics theory and experiments we know [21] that fluids often contain regions of coherent motion, called coherent structures. One of the most important group of coherent structures are vortices. In this thesis we will discuss how vortex criteria can be included into a visualization framework and simplify understanding the simulation results for the engineer.

Vortices influence fluid behavior on all scales and the phenomenon of vortical motion in fluids is well researched. See Figure 1.1 for a some examples. Vortices can be related to many application critical properties of the flow [32, 29, 86]:

- Vortices can improve mixing of fluids, e.g., the mixing of fuel droplets and air in the combustion chamber of an combustion engine is driven by a large tumble vortex .
- Vortices can create lift and considered critical in insect flight [].
- Vortices affect the efficiency of fluid transport, e.g., in water pumps or pipe lines.
- Vortices are a major source for material wear and critical pressure fluctuations inside fluid machinery.
- Vortices can cause high levels of noise, e.g. in wind mills, which often serves as an argument against widespread use of wind power plants in populated areas.
- Vortices can reduce the effectiveness of turbines and generators.
- Vortices can cause vibrations of engine parts reducing the lifetime and reducing its profitability.
- Vortices are linked to dangerous difficulties in air travel, including stall, wake turbulence at the wing tips and trailing vortices endangering following aircraft.

Even though many vortical phenomena are well researched, a final understanding of vortical fluid motion or turbulence in general is not within reach [21]. This is one of the key facts influencing the work of this thesis: there is no final agreement to what a vortex is and how it should be defined. Therefore we need a visualization system that is flexible, allows interpretation of fuzzy information and gives the user the possibility to include his or her knowledge into the visual analysis process.

1.2 Interactive Visual Analysis

Interactive extraction of information has been a hot research topic in recent years, focusing on interactive information drill-down [134], visual data mining [167, 66] and visual analytics [146]. Important issues are advanced interaction concepts as well as procedures and algorithms to gain access to features and information in the data. In this thesis we will see that a combination of feature extraction methods with modern interactive visual analysis approaches is a promising route to gain understanding of simulation results.

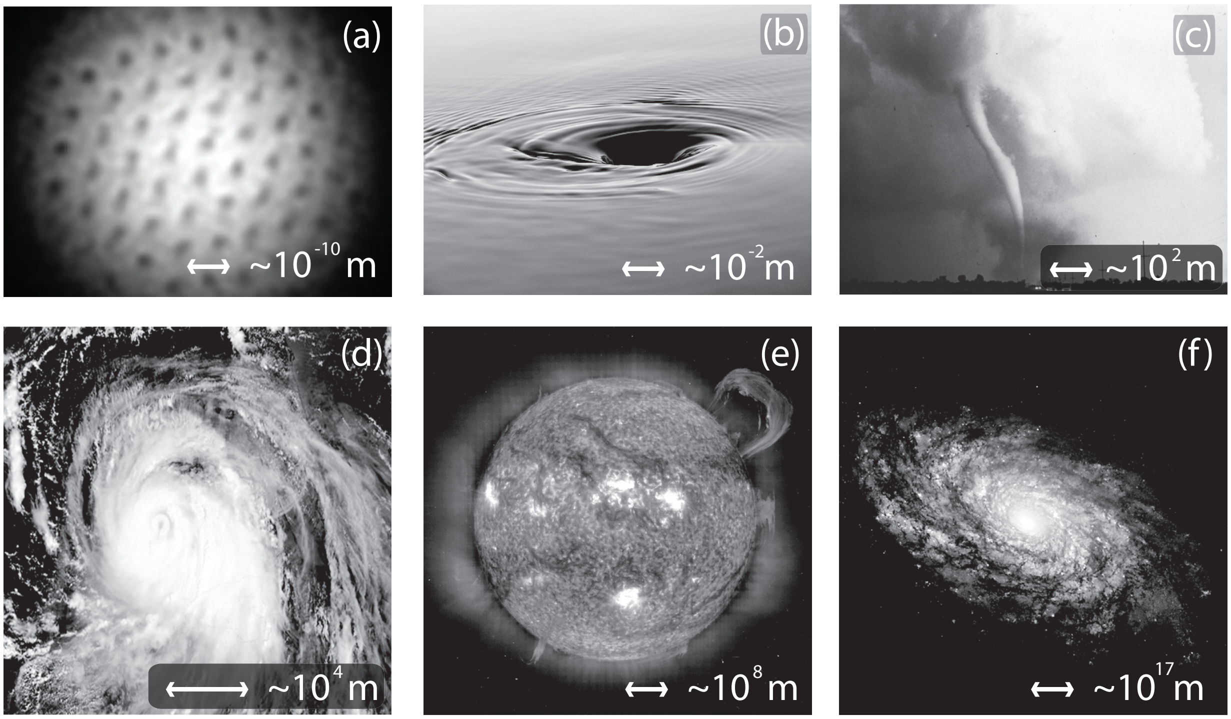


Figure 1.1: Vortices affect fluid behavior on all scales. (a) quantum vortices in a superfluid [130] (b) bathtub vortex [152] (c) tornado [109] (d) hurricane [106] (e) sun spot vortices [110] (f) spiral galaxy [105] (numbers approximate)

Interactive visual analysis provides techniques which give the user control over the visualization such that he or she can display the subset of information most relevant at the current moment. Analysis of the data is based on interaction metaphors that allow to filter and select data elements of interest interactively. The most important goal of interactive visual analysis is not in rendering individual images but enabling the user to gain insight from a given data set. The work of this thesis is based on the interactive visual analysis system SimVis [179]. There are three concepts at the heart of the SimVis [25] visualization system:

1. **Linking and brushing** allows to select subsets of interest in the data. Visual representations of the data (e.g., scatterplots, histograms, table views, etc.) provide individual metaphors to highlight data items of interest. In Figure 1.2 (a) we can see a brush selecting a subset of elements in a scatterplot. Multiple views can be linked together to form Boolean combinations of the individual selections of each view.

2. **Focus+context** visualization shows the relevant selected subset of the data in full detail while only providing a reduced depiction of the other data items to provide context to the viewer. In Figure 1.2 (b) we can see the selected attributes with pressure mapped to color.

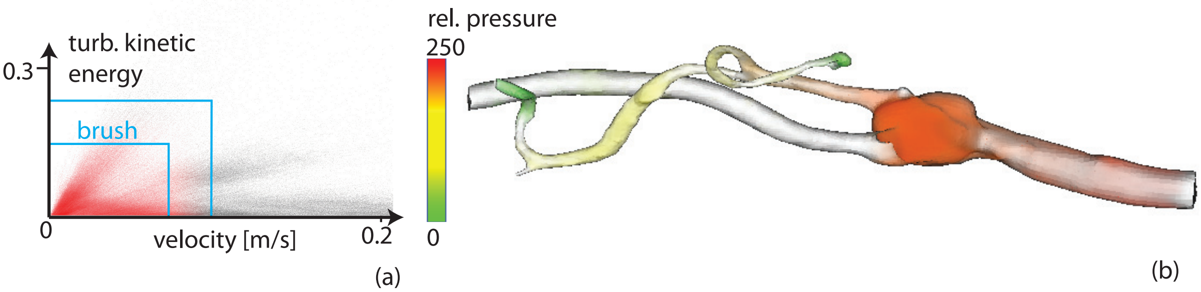


Figure 1.2: Linking and brushing combined with focus+context visualization. The data set contains a simulation of blood flow through an aneurysm. (a) Data elements are displayed inside a scatterplot, where the user has selected (brushed) data items of low velocity and low to medium turbulent kinetic energy. (b) The selected portions of the volume are rendered with color, while the rest of the data set is opaque and only the boundaries of the geometry are rendered in grey as context. Here we can see that the aneurysm contains a large quantity of low velocity fluid cells.

3. **Complex feature selection and smooth brushing** allows to combine multiple selections using fuzzy logic combinations of multiple selections. The strength of this approach lies in the possibility to express partial degrees of interest combined with a straightforward way to combine multiple partial selections using fuzzy norms. This approach is extended to vortex analysis in Chapter 3.

Each visualization technique is based on a transformation from raw data values to adequate representations which allow the user to get a better understanding. In the field of feature based visualization this approach can be considered to take three steps:

1. Domain specific features are extracted from the data (i.e., mapping raw data to feature information).
2. Features are mapped to visual representations, such as lines, glyphs or volumes.
3. A final result is rendered by combining all representations into a final image.

In this thesis we present automated feature extraction in combination with interactive feature analysis as means to understanding complex flow fields that are otherwise difficult to comprehend. We focus on extracting structures related to vortices in flow data. The extracted information can improve the analysis process in two ways: first, by including them into the visualization and second, by including them into the interactive analysis process. The application studies presented throughout this thesis originate from computational fluid dynamics (CFD) simulations, but the presented results apply to measured or synthetic data as well. Based on the realization that understanding is a process requiring different levels of detail, overview and filtered information we employ interactive visual analysis.

1.3 Contribution of this Work

The main contributions of this work are as follows:

- The contributions of this thesis include two mathematical vortex examples that model real world problems. Based on the results of these examples we derive modifications of existing vortex core line detection algorithms to extend them to the unsteady flow domain. Real world applications where the original approaches fail are presented and it is shown that the results improve using the modified approach. Finally a numerical study evaluates the impact of time-derivative estimation on the feature extraction process. In the appendix we give details on implementation details for unstructured grid data.

- Standard vortex detection criteria are adapted to allow fuzzy logic combinations of multiple detectors. While most detectors are prone to finding false positives [45], their numerical and mathematical properties differ. In Chapter 3 we discuss how the different vortex extraction schemes can be mapped into a common framework so that the user can analyze how they interact and complement each other for a given problem. This requires to extend the binary classifiers to generate non-binary response values. To convey the uncertainty that results from vortex feature derivation we use transparency coding and direct volume rendering of the selected regions. We show how derived features integrate into the process of interactive visual analysis.

- The non-binary feature detectors are integrated into a general multi-view where they can be combined for interactive analysis. This includes the adaption of the PQR plots to an analytic view based on scatterplots, an extension of classical rectangular brushing to integrate visual analysis and automated feature detection based on multiple detectors.

- We suggest an extension (delocalization) of the Eulerian vortex criteria to the Lagrangian point of view. This includes method to interactively control the crucial integration length parameter interactively in a data-driven way. The process of delocalization includes upstream information to deal with the problem of short particle trajectories. For delocalization a fuzzy accumulation and weighting technique is discussed, that deals with local numerical instabilities.

1.4 Organization of this Thesis

The second chapter gives an overview of state-of-the-art scientific visualization techniques. Chapter 3 introduces the notion of non-binary vortex detectors and gives an overview of previous vortex detection criteria. The non-binary vortex criteria are the basis for the following chapters: in Chapter 4 we discuss how multiple views can be used to interactively combine the presented detectors. Chapter 5 presents a filter technique to extend the local vortex detectors to include spatial and temporal characteristics of the flow into the extraction process. In Chapter 6 we focus on line type features and discuss how vortex core line detection can be improved in the unsteady flow setting.

Chapter 2

State of the Art in Visualization of Scientific Data

> "A process cannot be understood by stopping it. Understanding must move with the flow of the process, must join it and flow with it." (Frank Herbert 1920 – 1986)

In this chapter we discuss relevant research works related to the visualization of complex, multi-variate data. We focus on "non-classical" approaches, i.e. approaches which haven't been discussed in previous related reports, and we highlight techniques which potentially lead towards new directions in visualization research. We discuss how different techniques take effect at specific stages of the visualization pipeline and how they apply to multi-variate data sets being composed of scalars, vectors, and tensors. We also provide a categorization of these techniques in the aim for a better overview of related approaches. In the second part of this chapter we take a look at recent techniques that are useful for the visualization of complex data sets either because they are general purpose or because they can be adapted to specific problems.

2.1 Motivation

In the last decade, there has been an enormous progress in scientific visualization [96], still the visualization of multi-variate continuous 3D data, especially if the data set is time-dependent, remains a great visualization challenge. Recent publications have stated a shift in visualization research from a classical approach dealing with visualization of small, isolated problems to a new kind of challenge: visualization of massive scale, dynamic data comprised of elements of varying levels of certainty and abstraction [146]. It is a well known fact that we are experiencing an increase in the amount and complexity of data generated that exceeds our ability to easily understand and make sense of it. To address this development, Lee et al. [89] have indicated the work with multi-variate data sets as one important task for future visualization research that will require significant advances in visualization algorithms. Munzner et al. [102] addressed top scientific research problems and identified multi-field visualization as one of the central questions for future research. Therefore, in this article, we give an overview of existing work in scientific visualization that points in this direction. Intentionally, we leave out a vast amount of work focusing on solving specific visualization tasks which repeatedly have been discussed in other related reports.

Even though there is a wealth of algorithms for automated analysis to process data, these are only applicable when we know what we are looking for in the first place. Automated processing cannot generate new features and understanding beyond what is already known and might even remove important features of the data. Ward's Mantra [155] "I'll Know it When I See it", stresses the fact that we often rely on our visual pattern recognition system to help us in gaining knowledge of the data. For this reason we need to discover new ways to visualize complex scientific data or adapt the existing ones to deal with the new situation of multi-field data sets.

Different tasks need specific qualities in a visualization method. For example when trying to gain overview of the data a continuous and smooth style might be most appropriate, whereas for information drill-down a researcher might want a specific feature to be shown in a way as true to the data as possible. The application of combined hybrid visualization techniques will increase the value of any visualization application, therefore most of the techniques discussed in this chapter will be useful for other types of data as well.

By targeting the use of complex visualization techniques we do not speak in favor of a "more is better" approach, we would rather like to stress the importance of feature- and knowledge-driven visualization. The aim of this report is to give an overview of current techniques from various disciplines dealing with complex multi-dimensional scalar, vector and tensor data sets with the goal in mind to apply them to situations where these types of data are present at the same time in three dimensions. Kirby [71] defines the development of a visualization method as breaking the data into components, exploring the relationships among them, and visually expressing both the components and their relationships. To visualize the complexity of multiple data components and the relationships between them researchers have sought to find ways to combine the advantages of different types of visualization techniques. Hesselink et al. [53] give a short overview of research issues in visualization of vector and tensor fields that is still valid today. They declare four goals for future visualization research: feature-based representation of the data, comprising reduced visual complexity, increased information content and a visualization that matches the concepts of the application area. According to this, we see the three main advantages of multi-method visualization: first, improved effectiveness of visualization because each part of the data can be visualized by the most appropriate technique. Second, the ability to visualize multi-variate data sets while minimizing visual clutter at the same time. And third a separation between two questions that are too often intermingled: how to visualize vs. what to visualize. This intermingling is a drawback in the design of visualization systems featuring a single rendering technique. They very often induce a strong determination on the aspects of the data the user will be able to see. This does not mean that it is not possible to generate parameters that will show all the features but that a certain visualization tool might set the user on a track that hinders a successful investigation.

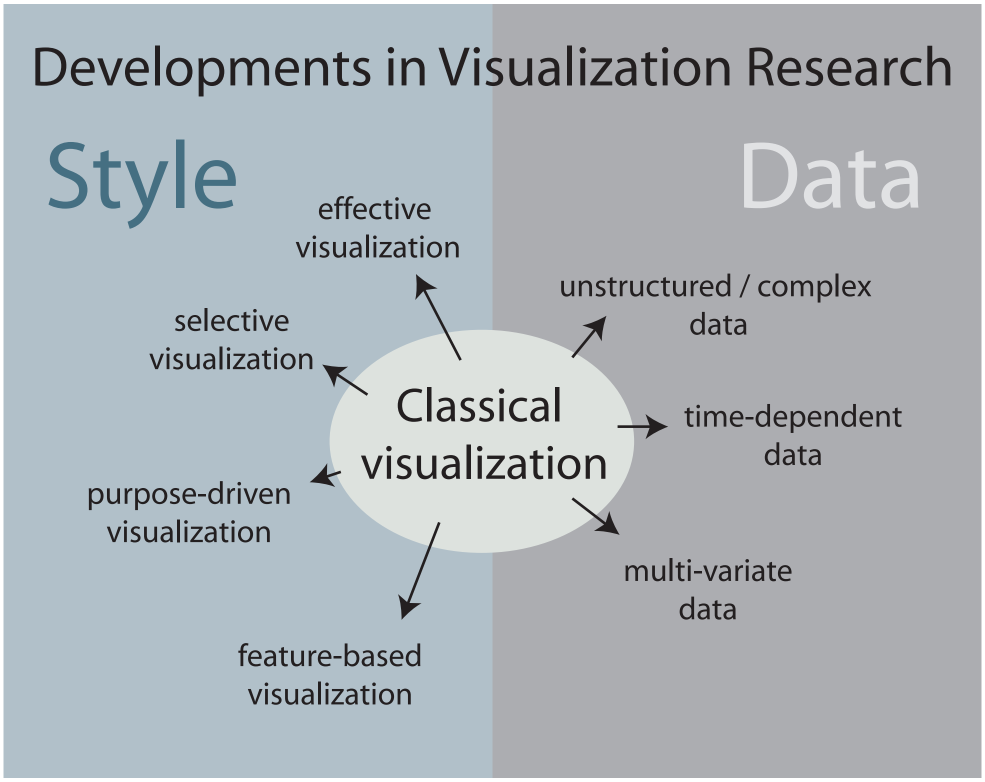

Developments in Visualization Research
Style
Data
effective visualization
selective visualization
purpose-driven visualization
feature-based visualization
Classical visualization
unstructured / complex data
time-dependent data
multi-variate data

Figure 2.1: Visualization research is developing from classical 3D structured scalar visualization in different directions. We can see two important developments in visualization research: on deals with the question how different types of data can be visualized well (data), while a second trend is to search for ways to extend visualization methods to adapt them to the requirements of the user (style). The use of techniques that allow the visualization of time-dependent, complex and multi-variate data in a useful way will be a key to successful visualization applications in the future.

An overview of current work on hybrid- and combined visualization algorithms can be a starting point for future visualization research in this direction of flexible and user-task driven visualization.

In the first part of this chapter (sections 1 to 4) we will discuss existing techniques resulting from applications in various scientific fields such as meteorology, CFD simulation, medical imaging and geology. Some types of data are of distinguished importance and appear in so many applications that visualization of these has become a research area in its own respect. Among these are flow, tensor and (time-dependent) scalar data. We will take a closer look at techniques that have been developed in these fields to deal with multi-variate data. In this context we will structure the following discussion relating to the type of data a publication mainly deals with and focus on scalar data in the first chapter, vectorial data in the second and tensorial in the third. At the beginning of each chapter we will give references to survey articles that focus on the classical approaches concentrating

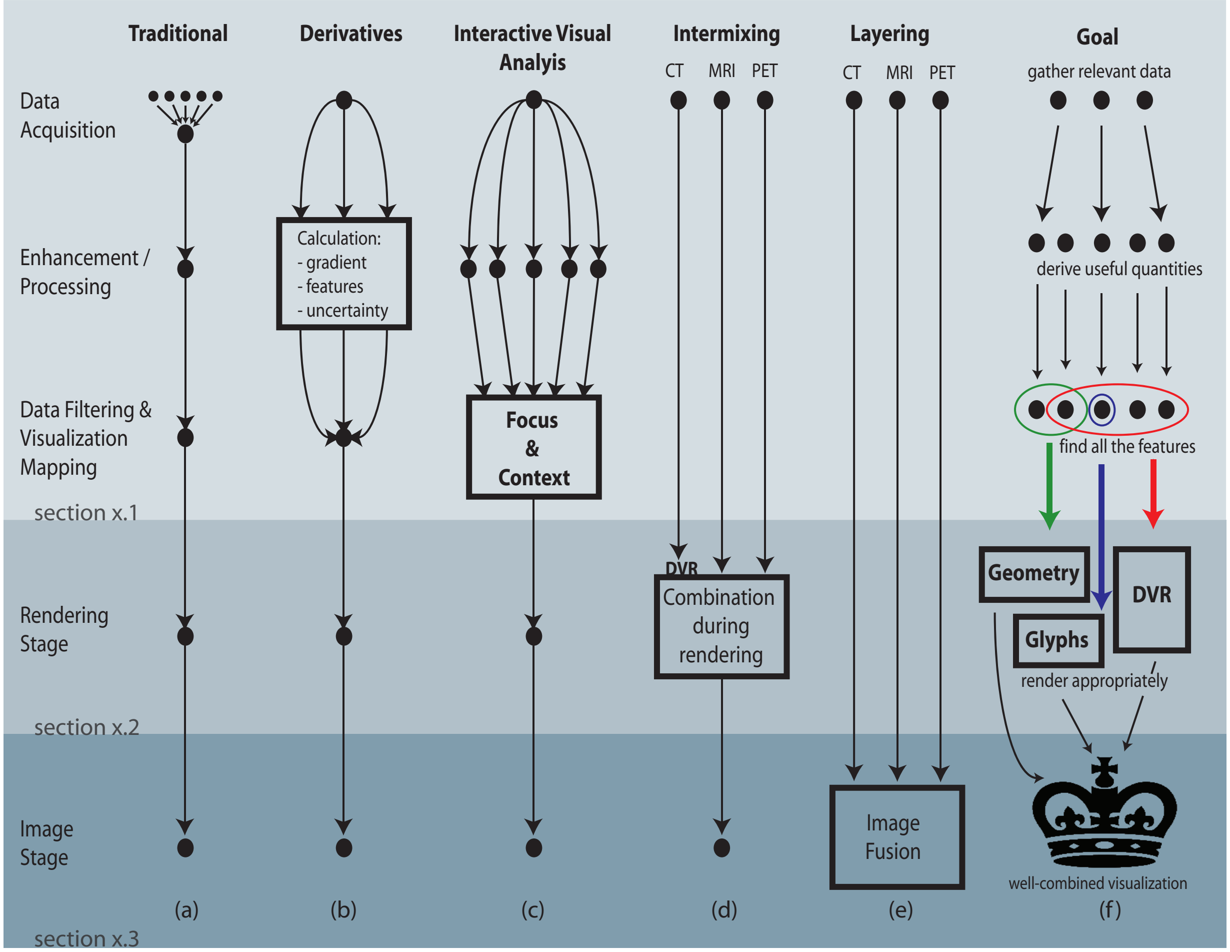


Figure 2.2: Six examples of how to deal with multi-variate data-sets: (a) traditional (b) deriving additional quantities (c) linking, brushing and other SimVis related techniques (d) data intermixing (e) layering rendered images (f) a well-combined visualization technique using all relevant data and derivations. Within the following sections on scalar, vectorial and tensorial data we will discuss techniques following the different stages of the visualization pipeline. In each section x we will begin with techniques related to data acquisition, processing and visualization mapping (subsection x.1), discuss techniques based on the rendering stage in the second subsection (subsection x.2) and techniques working on the image stage in the third (subsection x.3).

on one or few data items of the respective data type at once. In the second part of this chapter (sections 5 and 6) we will give a short overview of existing techniques that may not have been developed in the context of multi-variate visualization but that we consider as highly applicable for this purpose.

In each chapter we classify techniques dealing with multi-variate data sets according to the point of the visualization pipeline where the multi-variate nature of the data is being tackled (see Figure 2.2). The visualization pipeline is a model that describes how visual representations of data can be obtained following a procedure of well defined steps. (a) In traditional scientific visualization most of the acquired information is resampled onto a structured grid during an preprocessing and data analysis step. During data filtering the user selects the proportions of the data that will be visualized, e.g. by selecting values ranges of data attributes to be shown in focus. In the visualization mapping stage the data elements are mapped to visual representations. In the rendering stage the geometric information, data values and additional parameters such as viewing parameters and transfer functions contribute to the resulting image. The resulting image can undergo final modifications such as color enhancements or overdrawing (e.g. labels) to generate the final output of the visualization process. (b) Multi-variate visualization using derived quantities uses additionally computed values to improve the visualization, e.g. by using these values in the transfer function design [75] or for color assignment [40]. Other important types of derived features are segmentation data, classification data and cluster information. These are very often generated in a (semi-)automated fashion that also outputs uncertainty information for the generated information. These additional fields can be used to improve the visualization of the data [44]. (c) The linking and brushing concept in SimVis [25] uses different views and combines the selected features into a visualization mapping of the multi-variate data set. (d) The fourth type of multi-variate visualization combines the data coming from different modalities during the rendering step. The opacity is computed by using a combination operator to integrate the material properties into a single value [13]. (e) Layering renders different data items separately and combines them in an image fusion step [168]. (f) A goal could be a visualization technique that would collect all relevant data, derive the inherent information, (interactively) detect all features of interest, match them to an appropriate rendering algorithm and combine the results cleverly to get the optimal visualization.

When talking of complex volumetric data we distinguish between different types of data. We distinguish between multi-dimensional, multi-variate and multi-modal data. We will refer to multi-dimensional data in the scope of this article as data containing multiple independent physical dimensions (e.g., three spatial and the time dimension). Multi-variate data contains multiple variables or attributes, that do not have to be independent of each other. Multi-channel or multi-modal data is data that was acquired using multiple ways of measurement or operations. In many respects multi-modal and multi-variate data is considered equivalent in the context of visualization research. As an example we can think of a scan of the human chest using combined CT (computed tomography), MRI (magnetic resonance imaging) and ultrasound. The CT will capture the bones (e.g. ribs, spine) best, resulting in a single scalar field. The MRI scan is more accurate in measuring soft tissue, resulting in a second scalar field and with the use of a contrast agent it is possible to obtain vector information about the blood flow. Using multiple scanning modalities, we have obtained a multi-variate data set where the data elements are indexed by three spatial dimensions. At each grid point we have two scalars which describe the same physical situation and one vector with three components describing the x,y and z direction of the flow. We see that scientific data has more structure than general multi-variate data sets (e.g. census data or questionnaire results).

2.2 Dealing with scalar attributes

Scalar volume rendering is a central issue in medical data visualization. Engel et al. [30] give an extensive introduction. The acquisition devices used for medical imaging can be used for other purposes as well (e.g. industrial CT), but medical applications can be considered a large driving force of this field. Since medical data sets are often images obtained from different sources, the visualization of multi-valued data sets and their registration is a tightly coupled research area in medical image processing. Other types of scalar data result from sources such as numerical simulations, marine sonar, meteorology radar/satellite scans and photographic volumes. The most common sources for multi-valued scalar data are scanning devices as used in medial imaging and computational simulations. Also, scientific data sets are very often segmented or post-processed to extract regions containing different features which are of varying importance to the user – the location of a tumor might be of more interest than any other feature in the data set resulting in additional dimensions. An additional dimension for this kind of high dimensional data sets results from the uncertainty that comes with automated registration, segmentation and feature detection algorithms.

2.2.1 Techniques in the processing, filtering and visualization mapping stage

In this section we discuss visualization techniques that reduce the number of variables before rendering. Two important approaches are feature extraction methods and region of interest based methods. Feature extraction methods classify high dimensional data into features like iso surfaces, topological structures or other domain related features (such as vortices in flow data). They assign to each point in space a degree of membership to a feature of interest (e.g. a tumor) that can then be visualized using scalar rendering and color coding.

The techniques used in SimVis include linking and brushing for interactive feature selection [26] (see Fig 2.3). The user can specify a degree of interest in a subset of data items based on their attributes. The degree of interest functions from several linked views (scatterplots, histograms, etc.) are then accumulated using fuzzy-logic operators [25]. In the visualization, the flow features are visually discriminated from the rest of the data in a focus+context visualization style which is consistent in all views. SimVis supports smooth brushing to enable fractional degree of interest values as well as the logical combination of brushes for the specification of complex features. Brushing means to select intervals of the data values. The data elements that have attribute values inside these intervals, belong to the focus and are highlighted consistently in all views. According to this degree of interest function data samples are classified as focus or context and colored accordingly. This is an example of feature and knowledge driven visualization.

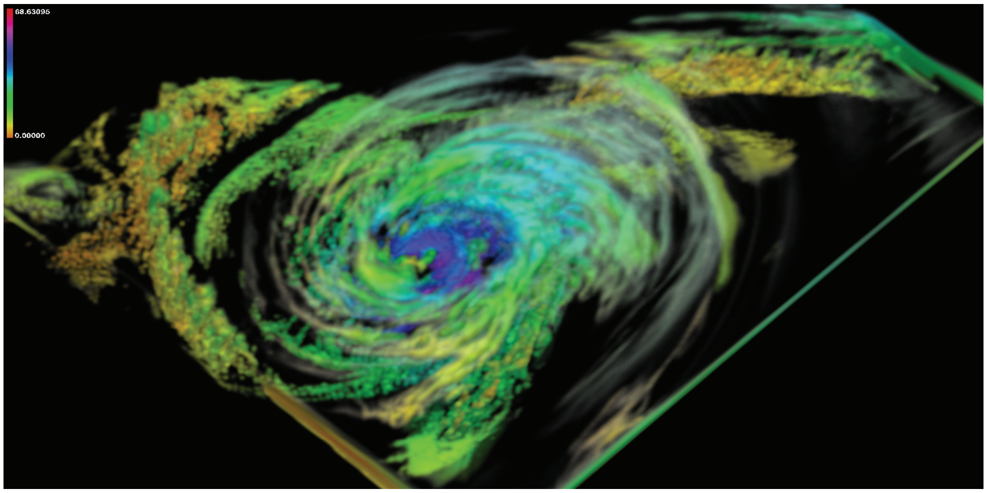


Figure 2.3: A DVR visualization of a hurricane data set using interactive feature specification and focus+context visualization [26]. (Image courtesy of H. Doleisch)

Tzeng et al. [151] suggest an intelligent systems approach to brushing. The user brushes portions of the volume that are of interest. A machine learning classifier (a neural network or support vector machine) is built from this training set. Based on the classifier the system then determines for each sample whether it belongs to the region of interest or not.

In the field of medical imaging the integration of different volumes into a single visual representation is called data intermixing (this compares to data fusion and volume fusion in other fields). The different modalities (e.g. Computed Tomography (CT), Magnetic Resonance Imaging (MRI) or Positron Emission Tomography (PET)) can show different, complementary and partially overlapping aspects of the situation. Therefore most algorithms are designed to allow flexible viewing of the original channels alone and more or less sophisticated combinations. A standard approach is to combine data sets based on segmentation information (e.g., the brain is visualized using MRI data, while the skull is shown based on data from the CT channel) combined with color coding (see Figure 2.4).

Illumination stage intermixing takes place in the visualization mapping stage: to combine the different attributes in the multi-valued volume voxel V, a combination function takes the attribute values $a_1, ..., a_n$ directly as input:

$$opacity(V) := opacity(combine(a_1, ..., a_n))$$

This way only a single transfer function is necessary, but we have to define a combination function that deals with the different value ranges of the attributes (e.g., using a multi-dimensional transfer function). To handle the complexity of higher-dimensional transfer functions, Kniss suggests the use of (local support) Gaussian transfer functions [76]. Kniss and Hansen [75] developed a technique for visualization of multi-variate data by applying multi-dimensional transfer functions and derived quantities. In a case study [74] they apply this approach to meteorological simulation data using three-dimensional transfer functions (for instance two axes map data values and the third the gradient magnitude). A drawback of this method is that multi-dimensional transfer function design is a complicated task and the results are hard to predict.

Kreeger and Kaufmann [77] present a hybrid rendering algorithm that combines volume rendering and translucent polygons embedded inside the volume. They apply their technique to combine an MRI-volume of a human head with an angiogram that visualizes blood vessels. Here the 'how' approach of the visualization (surfaces and volume) is matched to the 'what' context of the data (blood vessels and tissue).

Woodring and Shen [169] present a technique to visually compare different time steps of time-varying data sets using Boolean and other operations. The operators *over*, *in*, *out*, *atop* and *xor* compare two timesteps A and B at each voxel to derive a new field.

Another (rare) source of multi-modal data are photographic volumes. The visible human male data set contains vectorial (RGB) color information at each voxel taken by photographing each slice. Volume rendering is difficult in this context, because a high-dimensional transfer function from 3D to opacity is necessary. Ebert et al. [28] show how to use a perceptually appropriate color space for transfer function design. Ghosh et al. [39] render multi-channel color volumes consisting of CT, MRI and color information on the hardware. Muraki et al. [103] have presented a method to assign color values to voxels from multi-modal data sets using a neuronal net trained on a photographic volume.

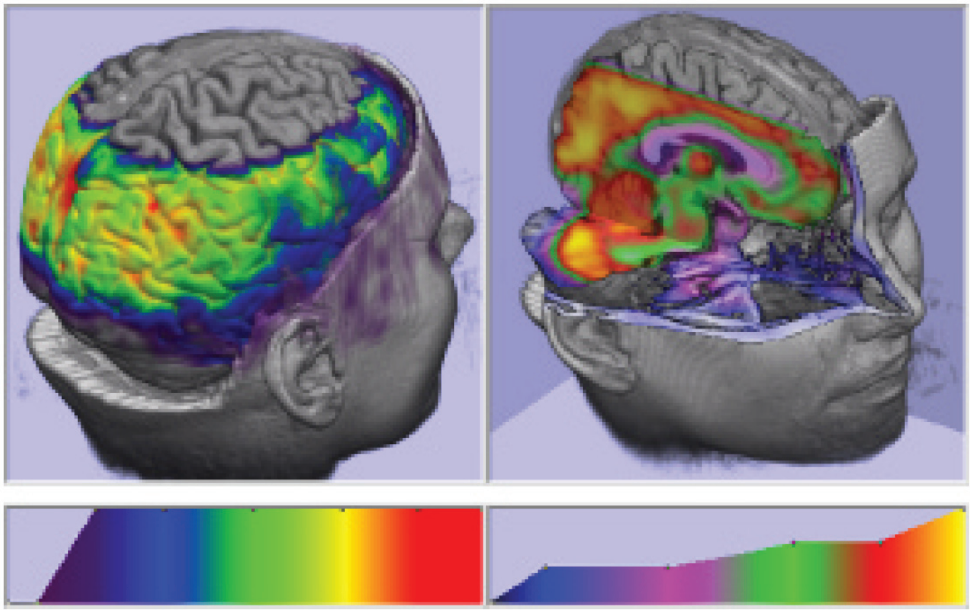


Figure 2.4: Using multiple transfer functions, region selection and color coding to combine information from multiple channels [93]. (Image courtesy of I. H. Manssour)

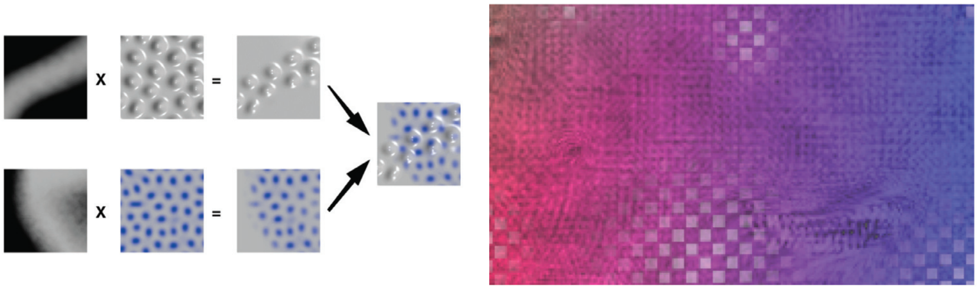


Figure 2.5: A combination of different bump-mapping styles to visualize multiple fields on the same layer [143]. (Image courtesy of R. M. Taylor)

2.2.2 Rendering stage techniques

Cai and Sakas [13] present a ray casting technique that integrates the information of multiple volumes during rendering. Data intermixing is done in the rendering pipeline during accumulation. On the accumulation stage the different modalities are already mapped to opacity and intensity values by their own transfer functions. This means they have the same intensity and opacity range ($[0, 1]$). Intermixing on the accumulation stage can then be done by defining additional opacity and intensity evaluation function taking as input the opacities of the different attributes $a_1, ..., a_n$:

$$opacity(V) := combine(opacity(a_1), ..., opacity(a_n))$$

The authors suggest to use linear or boolean operators for combination. Ferre et al. [35] discuss combination functions that take into account additional values, such as the gradient. Rössler et al. [121] present a GPU-based implementation of the DVR intermixing technique working with 3D textures and shader programs. Each data volume is rendered separately using an individual shader program allowing for different render modes for the modalities. Then intermixing is done when volume slices are combined in back-to-front order.

The Spectral Volume Rendering Technique [108] displays a multimodal volume using a physics based light interaction model: each material interacts with the light in its specific way. For different modalities, the interaction with several materials at one point in space is simulated. Spectral Volume Rendering is probably the physically most realistic technique to do illumination stage intermixing in terms of light propagation.

Grimm et al. [43] developed methods that allow efficient visualization of multiple intersecting volumetric objects that is applicable in the situation of multimodal volumes. They introduce the concept of V-Objects, which represent abstract properties like illumination, transfer functions, region of interest and transformations of an object connected to a volumetric data source.

2.2.3 Image stage techniques

Among the visual attributes that can represent data values are color, transparency, contour lines, surface albedo, texture and surface height. Textures are a versatile medium, that

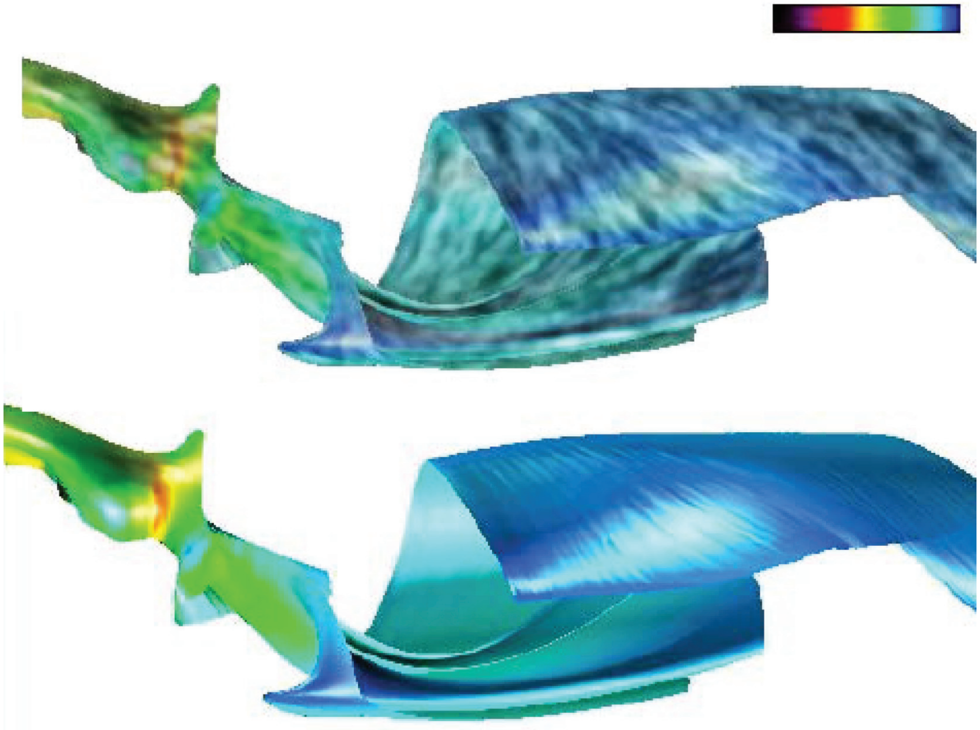


Figure 2.6: Combining texture advection and surface based flow visualization. Both the location of the iso-surface and its texture convey information about the flow [84].

can be computed with approaches such as spot noise, texture advection, bump-mapping and reaction-diffusion techniques. Shenas and Interrante [133] discuss methods to combine color and texture to represent multiple values at a single location. Taylor [143] describes a layering system for visualization of multiple fields on the same surface using data driven spots. He also discusses problems that arise due to the layering process. Their finding is that visualizing multiple data sets using a different technique for each layer is limited to four different fields, since the layers on top either mask or scramble the information below. Special care must be taken to keep the different layers distinguishable, for example by keeping the frequencies of the surface characteristics separated enough. In Figure 2.5 we see an example how bump-mapping and a reaction-diffusion texture are combined (left). On the right we see a resulting image using data driven spot textures. House et al. [54] discuss optimal textures for information visualization, including a large user study design, that investigates layered textures for visualizing information located on a surface.

2.3 Vector Field and Flow Visualization

The velocity of a flow is represented by a vector field and a vector field can define a flow, therefore in most applications their visualization can be considered equivalent [96]. Nevertheless, flow is more than just a vector field and recent visualization techniques

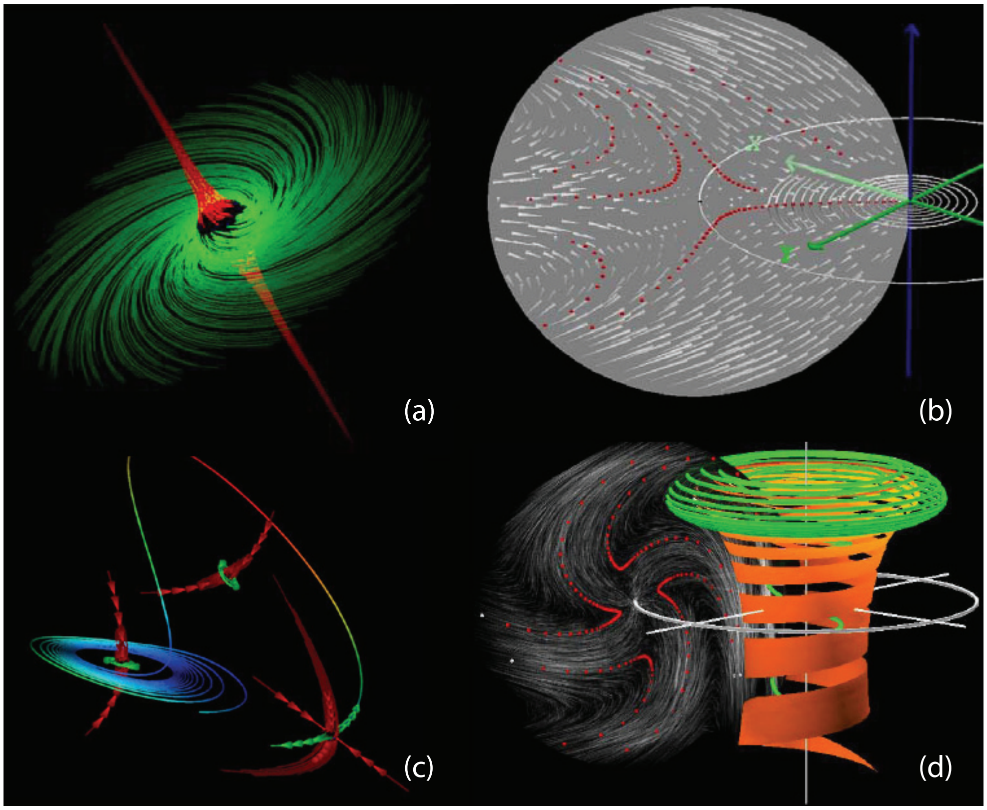


Figure 2.7: Enhanced topology visualization combining (a) streamline-based glyphs, (b) direct flow visualization, (c) solution trajectories and (d) streambands [49].

stressing the importance of the Lagrangian viewpoint address this issue.

Recent surveys and overview articles include: a classification of different flow visualization algorithms and a discussion on derived, second-order data by Hauser [48] and the state of the art report on flow visualization focusing on dense and texture-based techniques by Laramee et al. [88]. Post et al. [115] give an overview of feature extraction methods for flow fields.

2.3.1 Techniques in the processing, filtering and visualization mapping stage

A basic technique in flow visualization is to match the attributes of a data set to physically appropriate representations ('how' matched to 'what'). For example shock waves are mapped to surfaces, dispersed particles are mapped to particle traces or points. Therefore we will not repeat every application that uses combinations of standard flow visualization techniques such as lines [177, 73], surfaces [163], sub volumes [132] or dense techniques [60].

Laramee et al. [84] discuss the application of texture advection on surfaces for visualization of vector fields defined at the stream surface. In this application tumbling motion of the flow in the combustion chamber of a diesel engine is visualized by seeding a surface that depicts the swirling motion of the flow. This is based on work by van Wijk and Laramee on image space advection [164, 85]. In their approach parametrization of the surface is not necessary and advection is not computed for pixels occluded by other parts of the surface. The main steps are:

1. compute flow vectors at vertices of the surface mesh
2. project the vector field onto the image plane
3. advect texture properties according to the projected vector field
4. add shading to the image to convey shape information

This approach allows interactive frame rates for animated flow textures. Both the shape of the surface and the texture can transport meaning to the user (see Figure 2.6).

Since topology based visualization techniques feature sparse and economic screen usage, there is ample space left for additional information. Hauser and Gröller suggest a two step approach [49]. In the first step topology information is computed. Examples are fixed points and their Jacobians and higher order attractors. This is the classical step in topology visualization and in most cases the second step is not very intricate: different types of topological elements are visualized by different glyphs representing attracting, repelling and saddle points and separation lines [52]. This second step is now augmented by showing a visualization of the flow structure in a neighborhood of the critical point or visualizing the Poincaré map (see Fig. 2.7).

There is a lot of work in how the components of multi-variate data can be visualized. Sauber et al.[129] present multifield-graphs that deal with the question how the correlations between the components in the data can be shown. They introduce derived correlation fields that describe the strength of correlation between two variables at each point in space. The user can visualize correlation between scalar fields and vector fields. This also shows that the inherent information in multi-variate field that groups several variables to vectors and tensors can be useful when deriving additional information.

2.3.2 Rendering stage techniques

There is a number of flow visualization methods that render multi-valued flow data. Splatting is a very versatile technique that allows to integrate vector fields into scalar rendering by adding tiny vector particles into the splat texture [18, 19]. The examples by Max and Crawfis [97] combine surface geometries representing cloudiness with colored glyphs representing wind velocity. This is an example, where a single rendering technique shows different types of data and still uses appropriate visualizations for the components. In a data type oriented manner the ground is rendered as a surface, while the clouds have a volumetric look giving a good feeling of orientation in space. Directions and altitude are

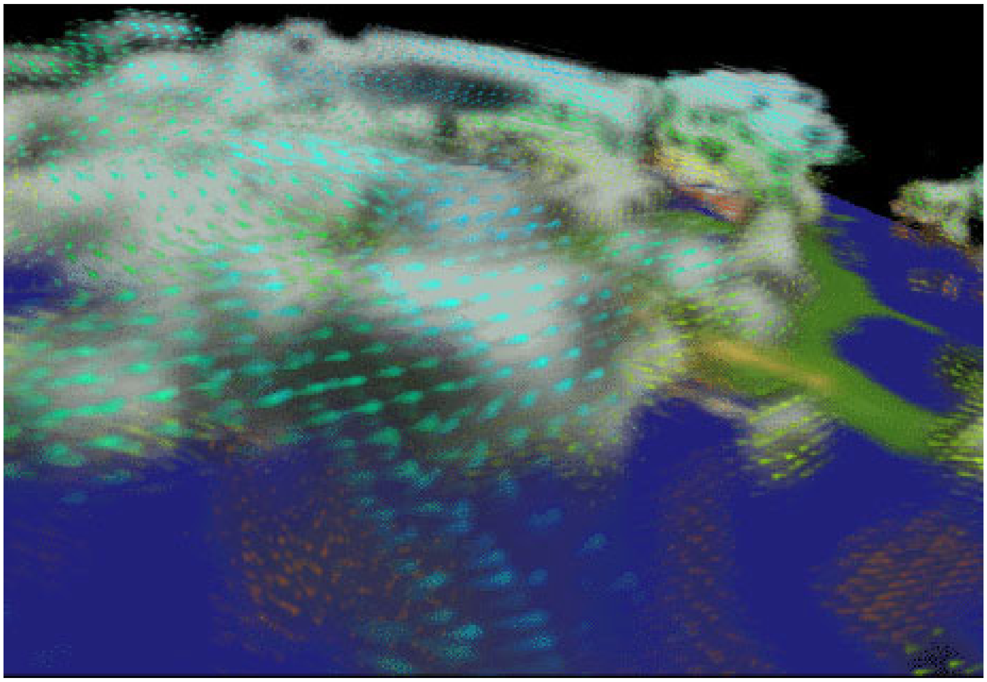


Figure 2.8: Direct volume rendering combining both realistic cloud rendering and splatted directional glyphs and color coding [18]. (Image courtesy of R. A. Crawfis)

visualized as colored glyphs, showing that they do not represent physical objects in space (see Figure 2.8).

Treinish investigated how specialized visualizations can be used to effectively visualize weather data using views of varying complexity [148] and presented a multi-resolution technique for complex weather data [147] (see Figure 2.9).

Since many of the existing flow algorithms are derived from physical models based on particles, the combination of particle and texture based flow visualization is a natural approach. Erlebacher et al. [31] developed a spatiotemporal framework that encompasses many aspects of time-dependent flow visualization. Weiskopf et al. [159] apply the spatio-temporal framework to unsteady flow visualization. In the context of dense flow visualization they identify two important types of coherence within the animation: spatial coherence, which conveys the structure of a vector field within a single picture and frame-to-frame coherence which conveys the development of these structures over time. They employ two steps: the first step is basically a propagation of particles forward in time to construct a space-time volume of trajectories. The second step applies convolution along paths through the spacetime volume that is done independently for each time step and texel. This hybrid particle and texture based approach combines advantages of particle-based representations with texture-based visualization. Particle systems are computational and memory efficient and allow accurate Lagrangian integration. Texture-based systems on the other hand have hardware acceleration for texture lookups and manipulations supported on modern graphic cards.

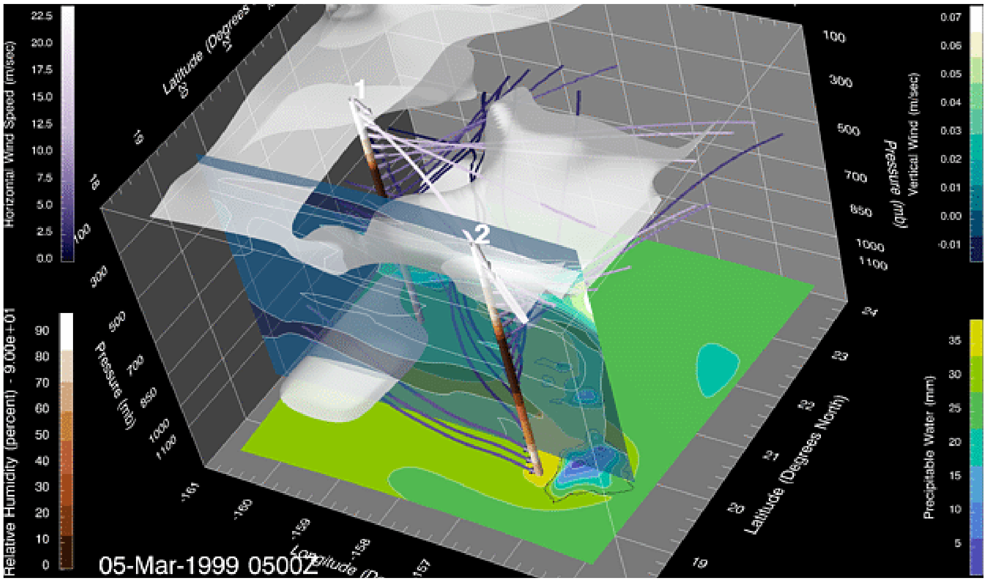


Figure 2.9: A weather visualization combining streamribbons, arrows, slices, contour bands and isosurfaces [148]. (Image courtesy of L. Treinish)

An approach that achieved not too much attention in the literature is to use more than one rendering system at the same time. Yagel et al. [172] suggested the use of four different renderers on a single CFD data set. Each is specialized to a specific task. Interactions can be visualized using a fast hardware-accelerated algorithm, high magnification images employ a specialized anti-aliasing technique. They use a ray casting algorithm specialized for the design of transfer functions while high-resolution and high-quality images are produced using a volumetric technique. Since today's computing machinery makes interactive manipulation of transfer functions, lighting parameters and other rendering attributes possible, the advantages of multiple combined renderers may be less obvious. Nevertheless an automatic selection of different rendering approaches and smooth transition between these would improve the visual experience for the user. This is an open research problem. Also, the integration of multiple renderers (e.g. illustrative and volumetric) into a single image at the same time is not investigated in much detail today. New ways to integrate different rendering algorithms is a promising route for future research. Mueller et al. [138] present an application that uses several renderers for prototyping, comparison and educational applications.

Stompel et al. [139] explore the use of illustrative techniques to visualize multi-dimensional, multi-variate data sets on structured grids. They use silhouette enhanced shading and depth-based color variation to enhance the 3D results and reduce cluttering related problems. Magnitude of temporal change is integrated using color modulation and flow properties such as direction, vorticity and velocity can be visualized using brush strokes.

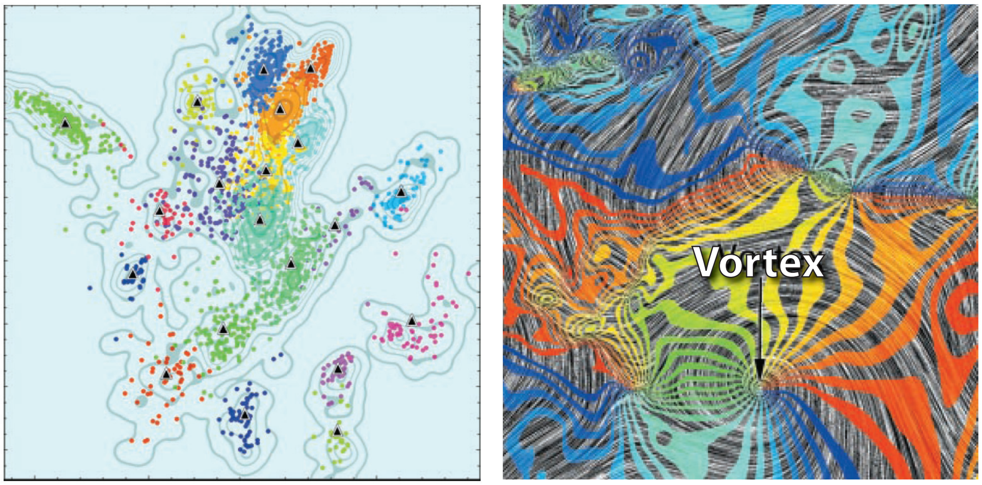


Figure 2.10: A layered combination of glyphs, color coding and isolines (left) and a filigreed layered visualization of flow data combining texture advection and color coding [168]. (Image courtesy of P. C. Wong)

2.3.3 Image stage techniques

Crawfis and Allison [17] very early recognized the power of compositing several images of rendered objects together to do scientific visualization. Their graphic synthesizer could combine several images to generate multi-variate representations of two dimensional data sets. Wong et al. [168] apply image compositing for visualization of multi-variate climate data. They present three image fusion techniques: opacity adjustments for see-through, filigreed graphics where portions of each layer are removed and elevation mapping where one scalar is mapped to the z-axis. In Figure 2.10 we see an example of layered glyph rendering (left) and a filigreed layering of color coded rendering and advection layered flow visualization (right).

Kirby [70] gives an introduction to art-based layered visualization in the context of 2D flow visualization (see Fig. 2.11). A promising approach to visualize multiple aspects of high dimensional data sets is the combination of illustrative, art- and glyph-based rendering.

They introduce the concept of layering in a similar way as done in oil-paintings: underpainting contains a low-frequency and low color-range coloring of a one-dimensional scalar vorticity value. Then two data layers follow: ellipses and arrows, depicting the most important aspects of the data. A final mask layer gives black context information (see Figure 2.11). By carefully selecting the order of layers it is possible to weight different aspects of the data differently and can suggest a viewing order for different parts of an image. Sobel [136] presents a descriptive language for modeling layered visualizations, that allows to design and share visualization parameters for layering algorithms.

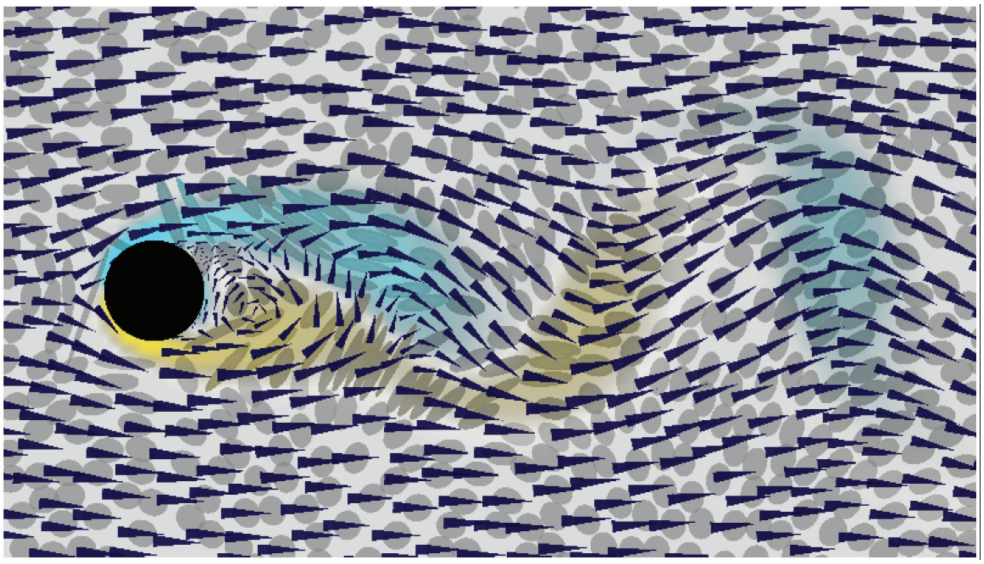


Figure 2.11: An visualization using multiple layers to visualize scalar, vectorial and tensorial information [71]. (Image courtesy of M. Kirby)

2.4 Tensor field visualization

Visualization of multi-variate data containing only tensor information is a difficult problem already. The interpretation of tensor information suffers if it is reduced to scalar information or if parts are visualized separately (e.g. in different images). Tensorial information has to be visualized fully or meaning and comprehensibility can be lost. When speaking of tensor field visualization we typically refer to second order tensors (three by three matrices). Depending on the application these tensors can be symmetric or non-symmetric. A symmetric tensor allows to calculate three orthonormal eigenvectors and corresponding eigenvalues. Non-symmetric tensor fields can be decomposed to a symmetric tensor and a vector field. Because of these properties most visualization applications focus on the visualization of symmetric tensor data – this already involves six variables at each point simultaneously. Because tensorial information is difficult to comprehend and structure, multi-style visualization techniques are common in this field. An example would be a layered visualization combining diffusion tensor glyphs and a CT reference image slice to show the organ geometry. It is also common to show basic geometry cues (e.g. the shape of the brain or the kidney) as context information in the form of a wire frame or silhouette rendering.

Important sources for tensor data are

- medical applications working with measured MRI diffusion tensors. Their visualization is the field of Diffusion Tensor Imaging (DTI) and deals with symmetric tensors with positive eigenvalues.
- materials science and geomechanics working with stress and strain tensor fields. Re-

lated tensors are symmetric with signed eigenvectors.

- fluid dynamics where several properties are tensor valued. Examples are the rate-of-strain tensor and the fluid momentum gradient tensor.
- general relativity theory simulations, where gravity is expressed as a rank two tensor and the electro-magnetic field tensor in special relativity.

Zhang et al. [153] give an extensive introduction and a state of the art overview of diffusion tensor visualization. Wünsche [171] gives a basic introduction into stress and strain tensor fields suitable for the computer scientist.

2.4.1 Techniques in the processing, filtering and visualization mapping stage

For tensor fields glyph-based visualization is the most common technique. Glyphs of stresses and strains is surveyed by Hashash et al. [47]. One basic question that many publications state is "visualize all the information of one tensor in some places or only some part of it everywhere?". The first would lead to some kind of glyph-based visualization where the information is visualized using a glyph that can represent all the degrees of freedom. Glyph-based visualization of tensor fields mainly uses the three eigenvectors (major, medium and minor) to generate a shape showing the direction of the eigenvectors. The most common is the ellipsoid, since it is possible to include all eigenvectors in a straightforward manner. Other glyphs are the Haber Glyph and the Reynolds Glyph [47].

A classification of tensor shapes was given by Westin [162]. A diffusion tensor is isotropic when the eigenvalues are about equal ($\lambda_1 \approx \lambda_2 \approx \lambda_3$), planar anisotropic where two eigenvalues are about the same and larger than the third ($\lambda_1 \approx \lambda_2 \geq \lambda_3$) or linear anisotropic where one eigenvalue is larger than the others ($\lambda_1 \geq \lambda_2 \approx \lambda_3$). The corresponding ellipsoids are spherical, disk- or needle-shaped respectively. Westin introduced the shape factors to measure which of these cases is dominant:

$$c_{linear} = \frac{\lambda_1 - \lambda_2}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}} \quad c_{planar} = \frac{2(\lambda_2 - \lambda_3)}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}$$

$$c_{spherical} = \frac{3\lambda_3}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}$$

The three shape factors sum to one and define barycentric coordinates, that can be used for glyph geometry assignment [162], opacity mapping [68], color coding, or glyph culling [174].

One way to get from glyphs to hyperstreamlines [22] is to place ellipsoids close to another along the direction of the major eigenvector. From any seed point, three hyperstreamlines can be generated using one of the three eigenvector fields for the streamlines and the other two for the cross section. This leads to a connected line along the major direction that encodes the other two eigenvalues in the cross section of the streamline. A hyperstreamline

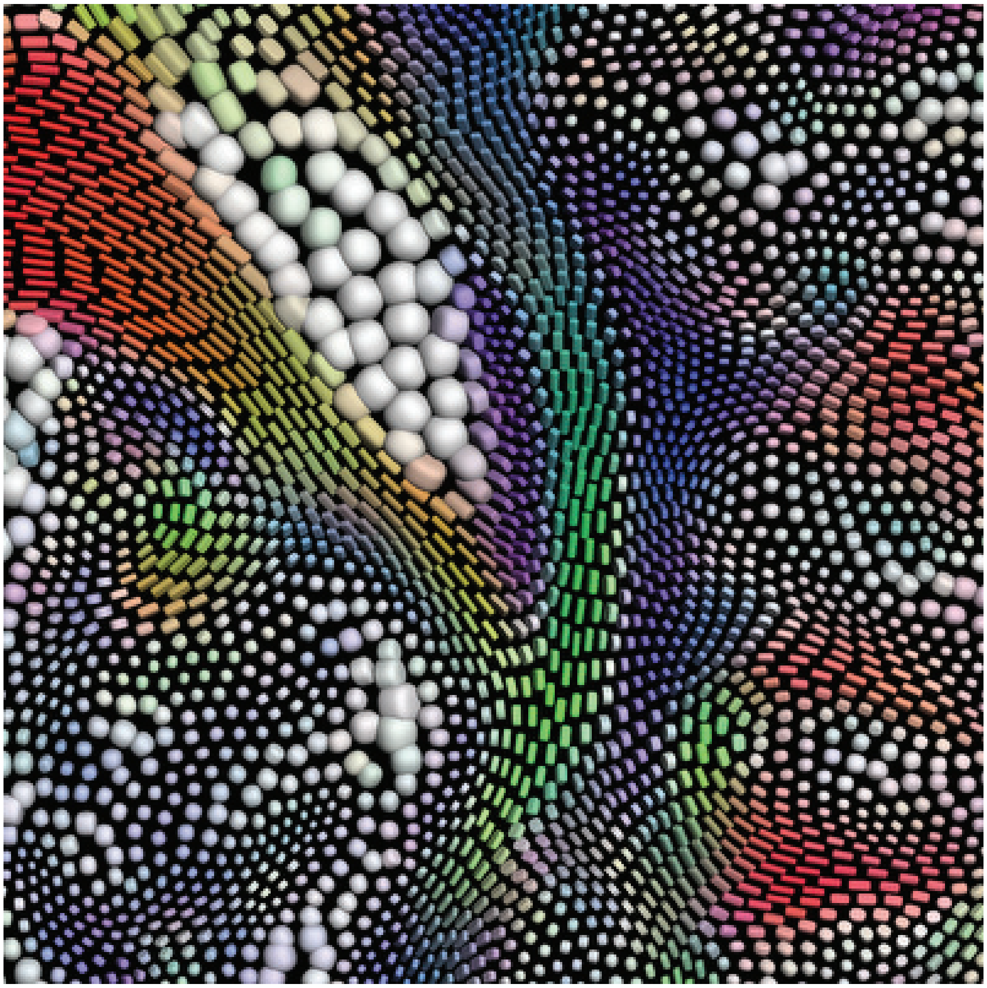


Figure 2.12: Sensible layout of glyphs allows to convey the location of fiber structures in the data. This shows how much automated parameter setting approaches (layouting) can improve the visualization [69]. (Image courtesy of G. Kindlmann)

visualizing a tensor field can be enhanced to show other properties by coloring. For non-symmetric tensors, the rotational components can be encoded as 'wings' along the main hyperstreamlines [175].

Zhang et al. [174] use stream tubes and stream surfaces for the tensor aspect of the data and contours and volumes as anatomical landmarks. The authors do not use the different visualization techniques to visualize different components of the data but show data ranges differently. Since in the brain, regions of high linear anisotropy very often correlate with regions containing densely fiber tracks, tensors having high linear anisotropy are adequately visualized using steam tubes, while tensors of high planar anisotropy are visualized using stream surfaces. This way both techniques can be used for the type of data they work best for.

In an adaption of image-based flow visualization Zhang et al. [173] visualize topological properties of tensor fields on curved surfaces. They discuss properties of critical points and an approach to extract flow directions to apply advection vectors. Furthermore they show applications to painterly rendering of 2D images.

Merhof et al. [98] present a hybrid rendering technique combining point sprites and triangle strips to display fiber tracts in the brain. They show that combining two rendering techniques can improve the comprehensibility of the visualization. This is an example how thinking about the ’what’ part of the visualization (fiber tracts) can give clues to improving the ’how’ approach.

2.4.2 Rendering stage techniques

Visualizing parts of the tensor information in a continuous fashion is done in volume rendering of tensor fields. Sigfridsson et al. [135] present a dense technique that filters noise along the eigenvector directions to produce a continuous representation of tensor fields that produce results similar to LIC. The basic tasks in volume rendering tensor fields – determining opacity, calculating shading and assigning material color – can be done by specific mappings of tensor properties based on the shape factors. Opacity is determined by using a barycentric opacity map (e.g. high opacity for linear anisotropy). Lighting is determined by using a heuristic that refers to the shape an ellipsoid glyph would have in the same position: in case of planar anisotropy the lighting model is the same as with traditional surface modeling, in the linear case the lighting model is similar to lighting of illuminated streamlines. Cases in between are interpolated. In the simplest setting color coding is done by using a colored ball and choosing color depending on the direction of the major eigenvector. This basic setting allows improvement using additional visualization techniques. Kindlmann et al. [68] present a reaction-diffusion texture that can visualize a tensor field alone but also integrate it with the volume-rendered tensor-field visualization (see Figure 2.13). The idea of a reaction-diffusion texture is to simulate a system of two differential equations. One describes the diffusion governed by Fick’s second law of two morphogens where the resulting concentration of these morphogens determines the color at each position. The other differential equation measures how much the two substances react and neutralize each other. The initial condition is that both have the same concentration everywhere. Applying diffusion relative to the given tensor field at each position generates a texture that can show information about the tensor field in its own right. The authors suggest color modulation or bump mapping to combine volume rendering and the volumetric texture. The result is similar to a surface rendering of geometry combined with diffusion glyphs, but has several advantages. The most important is that the resulting ellipsoids are distributed more naturally and are packed in a way that represents features of the data. Also the empty space between tensor ellipsoids is reduced. Furthermore it avoids the common problem of gridded ellipsoid layouting to give the false impression of structure in the data.

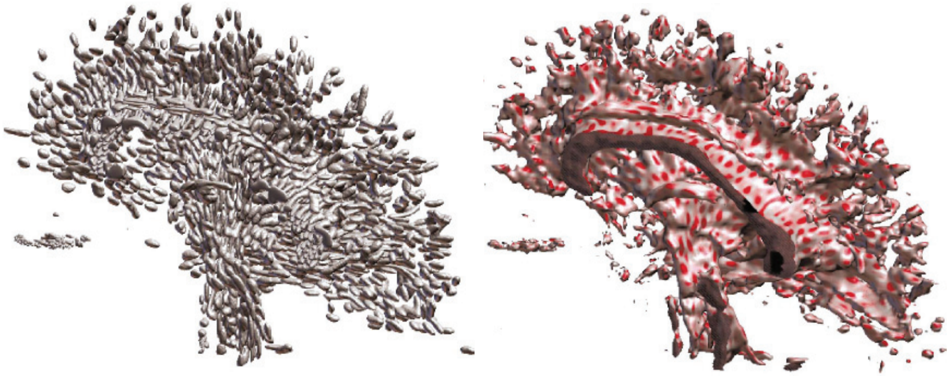


Figure 2.13: (a) The reaction diffusion texture allows natural glyph placement and geometry. (b) Alternative combined use of the texture for coloring the volume [68]. (Image courtesy of G. Kindlmann)

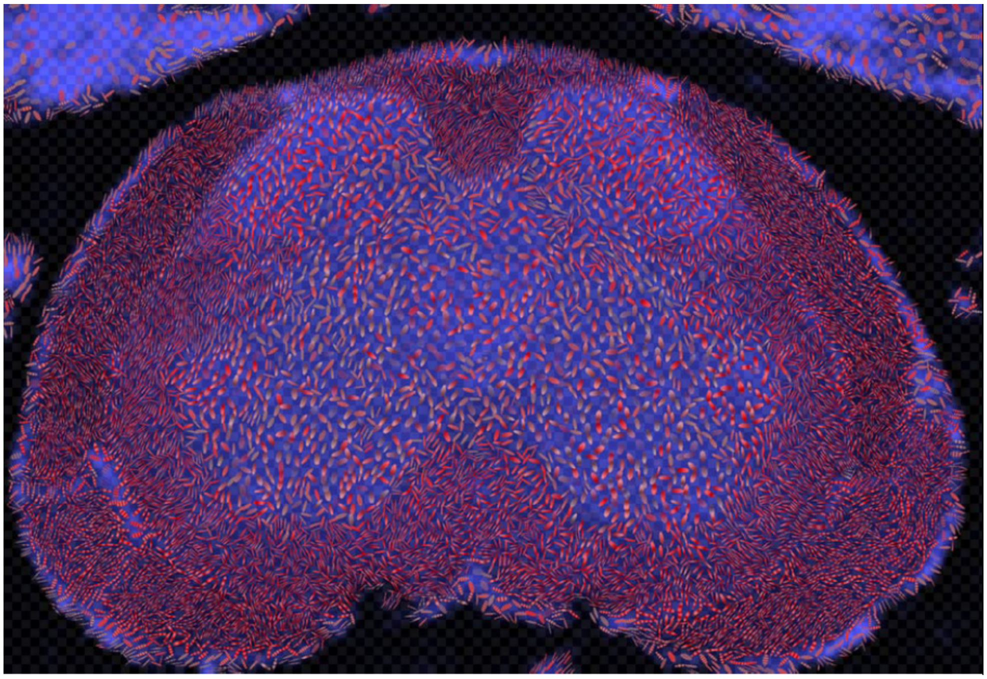


Figure 2.14: A visualization of the mouse spinal cord based on artistic techniques using multiple layers and glyphs [80]. (Image courtesy of D. H. Laidlaw)

2.4.3 Image stage techniques

For tensor visualization an image stage technique has been published by Laidlaw et al. [80]. They show that a lot of information can be conveyed in a single image using brush based glyphs and layering. Combining varying brush strokes and layering allows to display many components of the data locally, while the underpainting can show form. Contrast is used to create depth. Stroke sizes, texture and contrast help to define a focus within each image (see Figure 2.14). In a recent publication Wenger et al. [161] combine volume rendering techniques and layering into a multilayer volume rendering approach. Their method is related to two level volume rendering [51] which will be discussed in the second part of this chapter. They combine densely packed threads of colored and haloed streamlines with direct volume rendered context information. To deal with occlusion problems they designed interactive controls to change visualization parameters like thread length or opacity. Also, they heavily use transfer functions. This interesting publication is a good example of how to layer volumetric rendering successfully to visualize different portions of information.

2.5 General approaches to multi-dimensional visualization

In this section we will give an overview of the techniques we have identified to deal with complex data sets. It has become a widely accepted perspective to view visualization as a path from data to understanding [23]. We have identified a wide and diverse range of general approaches to multi-variate or complex data visualization. The following subsection cannot give a comprehensive enumeration of all the related work, but is thought to be an introductory overview. We also do not distinguish between 'how' and 'what' approaches, since several of the techniques can be used both ways.

Derivations or derived quantities are used because visualizing the measured data directly might not be useful for understanding it. Kirby et al. [71] show that in flow visualization for example showing additional components, that do not give information that was not already available from the velocity field, helps understanding the situation. In flow visualization useful derived quantities are for example vorticity, the rate-of-strain tensor, the rate-of-rotation tensor, turbulent charge and turbulent current. Smoothing the data to remove noise or calculating gradients to improve lighting will very often result in more pleasing visualizations that are easier to work with. Hauser [48] discusses the use of differential information to improve scientific visualization.

Glyphs (also referred to as icons) are a powerful communication item. A large number of data dimensions can be incorporated into the attributes of a single shape or symbol (see Figure 2.15). The particular mappings may also be customized to reflect semantics relevant to specific domains to facilitate interpretation. Since glyphs are generally not placed in dense packings, the free space between them allows the visualization of additional information. They interact therefore nicely with other visualization algorithms and are frequently added to visualization applications. Wittenbrink at al. [166] suggest glyphs for uncertainty in vector fields. Kindlmann and Westin [69] have presented a technique for packing glyphs in a way that their alignment conveys additional information. Hashash gives an overview of stress and strain tensor glyphs [47]. (See also [67, 155, 78, 165].)

Hybrid rendering is the application of several visualization techniques for the same image. This is useful especially for segmented data sets where background information is applicable to choose the appropriate rendering technique for different subregions of the

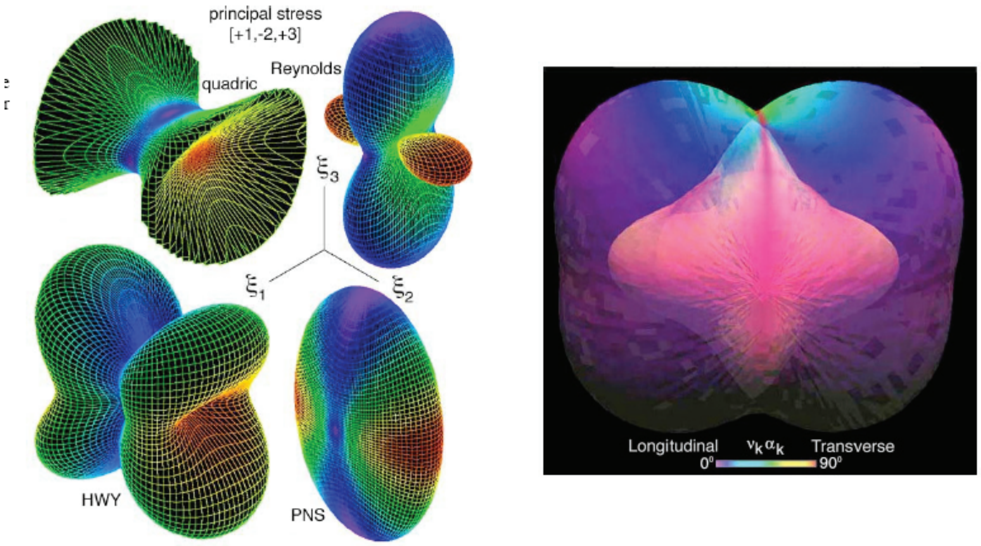
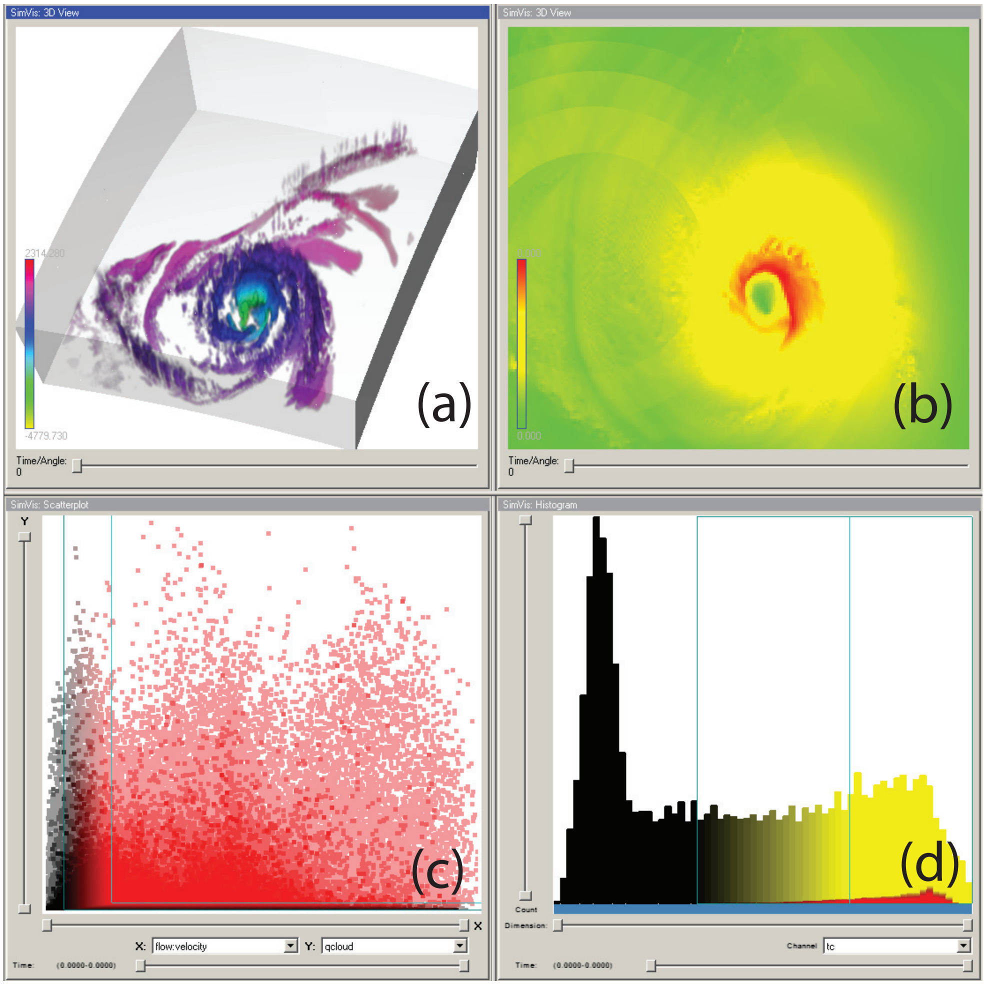


Figure 2.15: Advanced glyphs for stress tensor visualization using color and geometry for information coding (left) and transparency (right) [78]. (Image courtesy of R.D. Kriz)

data [51]. There are many examples for this approach: Jesse and Isenberg [63] describe a hybrid rendering scheme that combines photorealistic and illustrative rendering to highlight parts of a volume for presentation. Kreeger and Kaufmann [77] describe a fast method to combine volume rendering and translucent polygons to render mixed scenes. Laramee et al. [83, 85] and van Wijk [164] present rendering algorithms to visualize flow on surfaces. Wegenkittl et al. [156] combine surfaces, tubes and particles to visualize the behavior of a dynamical system.

Interaction is probably the most important tool for understanding complex data. Possible interactions are changing viewing parameters, transfer function manipulation, seeding point selection, streamline culling, streamline queries, graphical model exploration, region of interest selection and many others. An emerging trend is to use concepts from interactive visual analysis for data exploration. In Figure 2.16 we see an example of multiple linked views that work together to help understanding the data. In the attribute view (c) and (d) linking is helps to understand how different attributes are related: the data elements selected by the brush are shown red, while the elements selected in the other view are colored yellow.

Layering and Fusion has been used extensively in scientific visualization to show multiple items. Fusion-based methods combine different rendering styles in image space [168]. Layering is a generalization of this approach where multiple layers of information are visualized on top of each other. This is most applicable for two dimensional visualization but there is work where transparent stroked textures show surfaces without completely obscuring what is behind them [59, 58]. Several other layering techniques have been discussed in the first section of this chapter (see [17, 80, 168, 70]).



X: flow:velocity
Y: qcloud
Channel tc

Figure 2.16: An example of combined attribute and volumetric views. The 3D view (a) shows the location of data points in space with pressure mapped to color. A 2D slice (b) shows the velocity close to the eye of the storm. Two attribute views (scatterplot of velocity vs. cloud density (c) and a histogram of temperature (d)) are used to select which cells are shown.

Two-level volume rendering Hauser et al. [51] and Hadwiger et al. [44] present a two-level approach that combines different rendering methods for volume rendering of segmented data sets (see Figure 2.17). Each of the segmented regions can be rendered using a specific rendering method like NPR, DVR or MIP during ray accumulation. Since most users perceive three dimensional scientific data sets as built up from individual objects, the authors use the segmentation information to generate images that take this into account (one per object intersection). To compute the representative values for the objects different rendering techniques can be used, The authors also use the technique to visualize dynamical systems. This gives a hint at a more general applicability of their approach. Since the decision what rendering method to choose is given to the user, it becomes possible to use

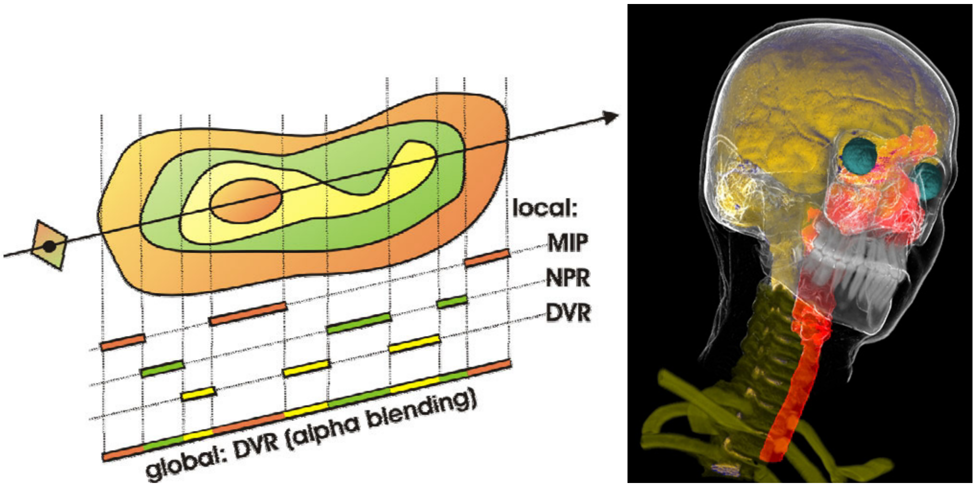


Figure 2.17: Two-level volume rendering allows to combine multiple rendering techniques using different compositing methods locally (left). A multi-level volume rendering of a human head using tone shading (brain), contour enhancement (skin), shaded DVR (eyes and spine), unshaded DVR (skull, teeth, and vertebrae) and MIP (trachea) [44]. (Image courtesy of M. Hadwiger)

the most adequate in the given moment. This approach is well suited to visualize multi-dimensional data sets by combining different rendering methods that are most appropriate for different features inside the data. (See also [161].)

Multiple Views present the information in several different views that encourage comparison, give contrast and help to generate a correct understanding. Roberts [119] describes the generation and presentation of multi-form visualizations in an abstract way and gives an introduction to multi-view visualization. Yagel et al. [172] discuss to group volume renderers that have different quality and rendering speed tradeoffs. Van Wijk and van Liere's hyperslicing approach uses multiple views to display a large set of possible projections of the data [165].

n-D viewing is based on defining hyperplanes on the high-dimensional volume or direct projection. This is done very often for time-varying data sets, where time-coherency can be exploited for compression and acceleration. The major issue for projections from n-D is to determine occlusion because a front-to-back ordering is not clearly defined after projection. Feiner and Beshers [34] suggest the World within World approach to drill down on the data by iteratively slicing away dimensions (see also [170, 107, 3, 165] and references therein). Blaas et al. [8] have developed a framework that uses interactive projection parameter specification for mapping multi-variate data values to scatterplots.

Probing is a general visualization approach for multi-variate data visualization. The user can state interest in a specific location or data range. Then a reduced amount of data is shown everywhere and for subsets of the data a local and more complex visualization conveys details. This avoids clutter and occlusion, is computationally efficient and helps the user to focus on specific aspects of the data. Examples for local detail global overview

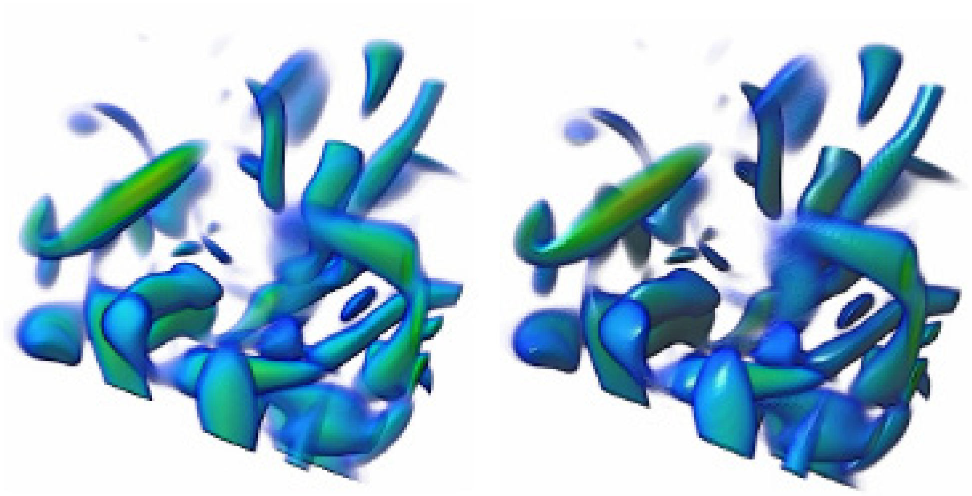


Figure 2.18: Visualizing flow data can benefit from using illustrative techniques. The image shows a closeup of turbulent vortex flow using silhouette and shading (left) and additionally gradient and depth enhancement (right) [139]. (Image courtesy of Q.-L. Ma)

techniques are: focus and context visualization [25], magic lenses [65], level-of-detail [16], clipping-probes [158] or zooming.

Reduction Dimension reduction and de-noising can remove unwanted details in the data and remove obscuring structures that hinder the process of understanding. Also, presenting views that contain a reduced amount of information and clipping are examples for data reduction tools. The importance of data reduction is very well expressed in the saying that in the future the main question will not be what to show, but what not to show. There is a trend to include attribute views (such as scatterplots, parallel sets, etc.) for interactive visual analysis of the attributes of the data-set. These views can benefit strongly from having automated clustering and reduction algorithms available.

2.6 Illustrative rendering

The visualization of multiple features and the accentuation of important structures and information has gained special attention in scalar volume rendering, especially in illustrative rendering. Illustrative rendering employs abstraction techniques to convey relevant information. In the context of scientific visualization non-photorealistic rendering refers to the adaption of techniques that have been developed by traditional artists for the generation of synthetic imagery. In the context of scientific visualization Bruckner et al. [11] have developed the VolumeShop framework for direct volume illustration.

Depth color Cues Svakhine and Ebert [141] describe depth based color variation. It gives intuitively understandable cues of the relative positions of different features in the data set. Distance color blending dims sample colors as they recede from the viewer. At the front of the volume, the voxel color remains unchanged. As the screen-depth value increases the color is gradually blended with the background color:

$$Color = (1 - depth) \cdot Color_{original} + depth \cdot Color_{background}$$

Silhouette enhancement and boundary enhancement Silhouette lines are particularly important in the perception of surface shape and the perception of volumetric features. In order to strengthen the cues provided by silhouettes, one increases the opacity of volume samples where the gradient is close to perpendicular to the view direction. Using a dark silhouette color can be effective for outlining of features. Levoy [90] proposed to scale opacity using the magnitude of the local gradient. Many applications also use the local gradient as a transfer function parameter [75]. Ebert and Rheingans [27] suggest to add scaling parameters to boundary enhancement such that the gradient-based opacity of the volume sample becomes:

$$o_g = o_v(k_{gc} + k_{gs}(||\nabla_f||)^k_{ge})$$

depending on the data o_v (original opacity), ∇_f (the value gradient of the volume at the sample) and on user specified parameters k_{gc} (scales influence of original opacity), k_{gs} (scales influence of gradient enhancement) and the exponent parameter k_{ge} that allows the user to adjust the slope of the opacity curve.

Enhanced transparent surfaces and stippling Since many features can be visualized using surfaces, transparent surface rendering offers a good possibility to show the spatial relationship between two superimposed features. To improve shape recognition, Interrante [58] uses principal direction-driven LIC [60]. The surface shape and curvature are enhanced using a stroke texture. This enables the visualization of additional variables using colored strokes and varying stroke width and density. The surface information is stored in a texture and needs recomputation if stroke parameters are changed. Nagy et al. [104] have developed a technique to do stippling of surfaces in real time using preprocessed principal directions. This allows to change stroke positions and appearance interactively but it needs an additional pass to render the strokes.

Feature halos In line drawings it is common to depict the depth ordering of two crossing lines by inserting a short gap into the line behind the other such that the lines do not touch. Therefore halos are an important technique for visualization of streamlines. For scalar volumes Ebert and Rheingans [27] propose a method for creating halo effects for features during the illumination process. Halos are created primarily in planes orthogonal to the view vector by making regions just outside features darker and more opaque. They calculate the halo-intensity by weighting the neighbors gradient magnitude and halo-direction.

Shadows and indirect illumination Even though a feature in the volume (e.g. a vortex) may not cast a shadow in reality, a shadow can very well show spatial relationships between features in the data. Especially for glyphs and disconnected features lighting is a powerful way to provide better perceptual cues [42]. Also, shadowgraphs and schlieren [142] can depict important structures inside a volume. Conceptually, due to inhomogeneities

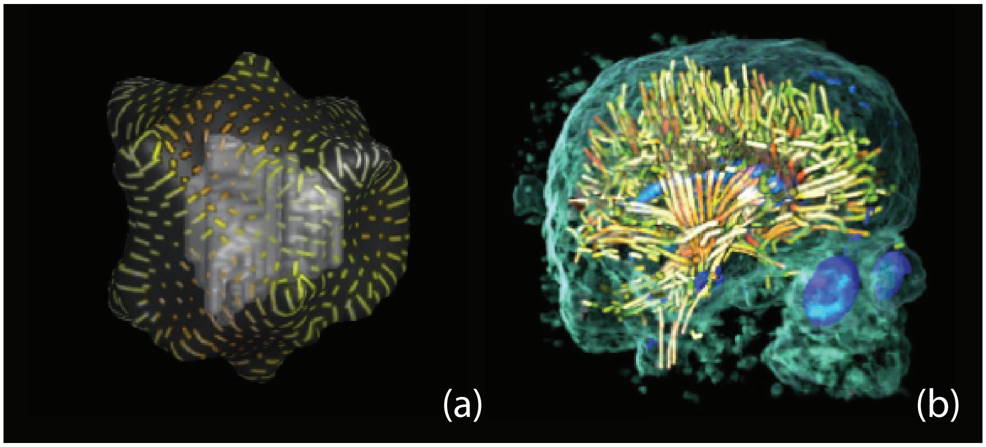


Figure 2.19: (a) Transparent surfaces can allow layered visualization for three dimensional images [60]. (Image courtesy of V. Interrante) (b) Non-photorealistic rendering of tensor information using line-glyphs and DVR of context information [161]. (Image courtesy of A. Wenger)

inside a translucent volume, refraction within the affected region displaces light rays before they reach the screen. Therefore this is similar to boundary enhancement. The technique stems from flow photography but is applicable to visualize structures in other volumetric data sets as well.

Cutaways and Ghosting Feiner and Seligmann [33] introduced cut-away views and ghosting to 3D visualization. Cut-away views are a common tool of illustrators to show important structures in a volume that would be occluded otherwise. Viola et. al [154] introduced importance-driven volume rendering to determine which parts of the volume are to be cut away (this notion of importance is also referred to as degree-of-interest (DOI) in other publications). Ghosting [43, 10] is a technique very often combined with cutaway views. Instead of removing the obscuring region completely they keep the most important parts (e.g.edges) of the occluding structures to conserve context information.

Motion blur is a widely used technique to show information from time-dependent data sets. For example Neophytou and Mueller [107] use a motion blurring effect to give hints on how the data will evolve.

2.7 Chapter Conclusions

We have discussed new developments in visualization research and have outlined recent trends in the literature. Based on these trends we have presented an abstracted goal for multifield visualization algorithms that highlights open questions on different stages of the visualization pipeline:

- How can we capture all the relevant information?

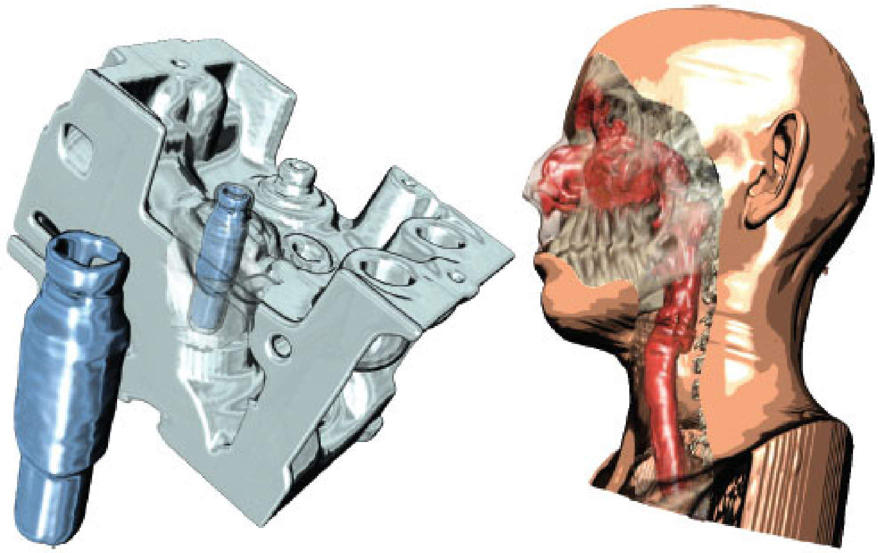


Figure 2.20: Techniques from traditional illustrative imaging can increase the information content in volume rendered images. Ghosting can be used to show important features in detail while cutaway views keep structural information independent of the viewing direction [10]. (Image courtesy of S. Bruckner)

- How can we combine relevant quantities of information belonging to the same location in space?
- What quantities can be derived from the data, that will help the user to understand the data?
- What are the features in the data and how can we combine automated and interactive techniques for finding them?
- Which visualization technique is most appropriate to the given data?
- How can we combine different visualization techniques for efficient and effective visualization?

In the first part of this chapter, we have discussed applications and algorithms employing one or several useful techniques to visualize complex scientific data sets in two and three dimensions. In the second part we have given an overview of modern approaches to for multi-variate data visualization and analysis.

Conclusively, in the literature we found researchers taking three general approaches to deal with complex, high-dimensional data sets. These approaches are highly promising and research in how these can be combined and improved will increase the value of visualization systems:

- *Interaction:* The connection between elements of the data can be explored interactively. This approach works on the assumption that the user will understand the configuration of a data set using different views and different levels of detail interacting with visualization parameters. Exploration of correspondence between different views on the data by visual comparison using the powers of the visual system to see what the data means. This is probably the most important key to understand complex multi-dimensional data and is used extensively in related fields like information visualization as well.

- *Complexity-adjusted rendering:* Due to the growth in computation power a valid approach to deal with the rising complexity of the data is to increase the computational efforts put into visualizing this data. Rendering complex visual clues (e.g. lighting models, shadows and distance clues) can make spatial interpretation easier for the user. We have seen that visualization research has produced tools to decrease clutter and to visualize complex and intertwined three-dimensional data features.

- *Multi-method visualization:* Is the approach to use a visualization technique only for the type of range of data it works best for. In the scope of this article we have discussed several publications following this approach and a tight integration of multiple techniques in a visualization gives a key advantage towards understanding the data.

- *Automated techniques:* provide powerful tools when working with complex multivariate data sets. Two important types of automated tools are feature classification algorithms and automated optimizations of visualization parameters. Classification, feature extraction and segmentation information are useful to render data in a more appropriate way and to ease the interaction with the data. Also, they may help to extract information from the data. Automated parameter optimizations (e.g. for seed point placement, color selection or viewpoint parameters) can reduce clutter, speed up rendering and help to extract meaning from the data.

This chapter has given an introduction into the field of scientific visualization in general. In the following chapters we will focus on the field of flow visualization and analysis, with the goal in mind to visually analyze vortex structures in flow simulation data. We will see that the four approaches *interaction*, *complexity-adjusted rendering*, *multi-method visualization* and *automated techniques* are relevant for understanding vortex features in flow simulation data. In the next chapter we will focus on interactive techniques combined with automatic computation of vortex detectors. In Chapter 4 we extend this approach but focus on attribute views which reduce the complexity of the information in comparison to the three dimensional renderings of the first chapter. Chapter 5 deals with improving the control of automated vortex detectors. Chapter 6 deals with vortex extraction in time-dependent flows improves a widely used vortex detection technique and also combines the visualization techniques used in the previous chapters with line-type visualizations combining volume rendering and glyph based visualization.

Chapter 3

Binary and Non-Binary Vortex Detectors

”Truth suffers from too much analysis.” (Frank Herbert 1920 – 1986)

The previous chapter has discussed scientific visualization from a more general perspective, with an emphasis on flow visualization. We have seen examples where feature detection can be an important tool for understanding scientific data sets. A CT scan of the head can include information on regions of the brain, while a part of an engine can be displayed in focus, when it has previously been identified as an important piece or feature of the engine. The process of feature extraction becomes relevant for flow visualization when vortices have an influence on the application. In this chapter we present background information and related work on vortex detection.

The first sections present motivation and an overview of binary vortex detectors. In Section 3.3 we discuss continuous formulations of common vortex detectors that allow a seamless integration into the concept of interactive visual analysis of flow simulation data. We express the originally binary feature detector results as fuzzy-sets that can be combined using the linking and brushing concepts of interactive visual analysis. Both interaction and visualization gain from having multiple detectors concurrently available and from the ability to combine them. An application study on automotive data reveals how these vortex detectors combine and perform in practise.

3.1 Motivation

There is still no ultimate agreement on how to generally define and detect vortices, even though the concept of a vortex is common in fluid dynamics and has proven useful to describe and model the behavior of fluids. Vortices belong to the most important coherent features in flow fields. They influence the behavior of the flow on all scales and are responsible for phenomena like hurricanes and tornadoes, mixing of fluid materials, have influence on the effectiveness of engines and machinery, and the drag on moving objects.

Reportedly local vortex extraction methods fail to find all vortices in real-world data [14]. For example, if there are two axes of swirl, many local detectors will indicate a direction that is a combination of the two [122]. The relations between the different criteria have been investigated on a formal basis [20, 15, 114], still the reasons for their different performances are not fully understood. At the moment, there is no answer to the question which detector will perform best in a given situation in general. Especially as long as there is no final answer to the question 'what exactly is a vortex?', we suggest to use a hybrid approach in visual analysis that combines the strengths of more than one criterion.

While most detectors are prone to finding false positives [45], they do not share exactly the same numerical and theoretical problems. In this chapter the vortex detection criteria are combined into a common framework. This is done by extending the binary classifiers to generate fuzzy logic response values. Transparency and DVR [24] are used to convey the uncertainty that results from vortex feature derivation.

3.2 Background on Vortex Detection

In this subsection we give an overview of the most common vortex detectors that serves as a basis for their continuous counterparts, described right afterwards. Feature-based flow visualization has been an active field of research for many years and it is beyond the scope of this chapter to provide a comprehensive discussion of all of this work – we refer to Post et al. [115], who published an extended overview recently. In this section we focus on selected pieces of previous work, which are tightly related to this thesis.

From experiments and from literature inspection we have found a series of flow attributes and notations to be useful for understanding the properties of vortices:

- a point $\mathbf{x} \in \mathbb{R}^3$ and an attribute value $a(\mathbf{x})$
- the data set D and data elements in a r-neighborhood of a point $\mathrm{P}_r(\mathbf{x}) = \{\mathbf{x} \in \mathrm{D}, |\mathbf{x} - \mathbf{y}|^2 < r^2\}$
- linear scaling of attribute values $a(\mathbf{x})$ to the interval $[0, 1]$ of a subset of data elements $\mathbf{x} \in \mathrm{S} \subseteq \mathrm{D}$ with minimum $\min_S = \min\{a(\mathbf{x})|\mathbf{x} \in \mathrm{S}\}$ and maximum $\max_S = \max\{a(\mathbf{x})|\mathbf{x} \in \mathrm{S}\}$ as $scale_\mathrm{S}(a(\mathbf{x})) = \frac{a(\mathbf{x}) - \min_S}{\max_S - \min_S}$
- the velocity field $\mathbf{u}$
- curl of $\mathbf{u}$ (or vorticity) $\quad \omega = \nabla \times \mathbf{u}$
- the helicity can be defined as $\omega \cdot \mathbf{u}$, normalized helicity is then $\omega \cdot \mathbf{u}/|\omega||\mathbf{u}|$. Normalized helicity physically represents the angle between velocity vector and vorticity vector.
- the velocity gradient tensor $\mathbf{J} = \nabla \mathbf{u}$ is the Jacobian of $\mathbf{u}$
- the rate-of-strain tensor $S = \frac{1}{2}(\mathbf{J} + \mathbf{J}^T)$

- the rate-of-rotation tensor $\Omega = \frac{1}{2}(\mathbf{J} - \mathbf{J}^T)$
- there are several criteria working on the parameters of a local curvilinear coordinate system such that $\mathbf{J}$ becomes

$$\nabla \mathbf{u} = [\mathbf{v_r v_{cr} v_{ci}}] \begin{pmatrix} \lambda_r & & \\ & \lambda_{cr} & \lambda_{ci} \\ & -\lambda_{ci} & \lambda_{cr} \end{pmatrix} [\mathbf{v_r v_{cr} v_{ci}}]^{-1}$$

  With $\mathbf{v_r}$, $\mathbf{v_{cr}}$ and $\mathbf{v_{ci}}$ being eigenvectors of $\mathbf{J}$, λ_r the real eigenvalue and $\lambda_{cr} \pm i\lambda_{ci}$ the complex conjugate eigenvalue pair. This differs from the eigenvalues of a symmetric matrix that has three real eigenvalues $\lambda_1 \leq \lambda_2 \leq \lambda_3$.

We list the vortex features considered in the scope of this thesis, a more in-detail discussion follows in Section 3.3.1.

Kinematic Vorticity Number N_k is a local measure that gives the quality of rotation independent of vorticity magnitude. It was introduced by Truesdell [150] as $N_k = ||\Omega||/||S||$. A value of $N_k = \infty$ corresponds to solid body rotation and $N_k = 0$ to irrotational motion.

Vorticity magnitude The straightforward way to look for vortices is to search for regions of high vorticity magnitude. Vorticity magnitude thresholding is often used as a detection method to mark cells as potential vortex regions and apply local verification using computationally more demanding methods.

Helicity and Curl Levy et al. propose the use of normalized helicity and curl and search for regions where $\mathbf{u}||\omega$ [91]. Even though this may not always correspond to the actual vortex core line, the authors used this feature with corresponding colors successfully on meteorological data. Inside streamwise vortices the velocity and the vorticity vector tend to align themselves parallel to each other. As a result, a normalized helicity value close to 1.0 or -1.0 can be a good detector for streamwise vortices.

Hunt's Q criterion compares the symmetric strain component (S) and the rotational component (Ω) of $\mathbf{J}$ [56] (see definitions above). Hunt combines this with the additional requirement of a local pressure minimum. In our framework we do not need to do this, since this additional requirement can be added interactively by brushing a derived quantity that finds local pressure minima.

'Lambda 2' introduced by Jeong and Hussain [62], the λ_2 criterion is one of the most popular vortex region detectors and has been studied extensively over the years. The criterion involves computing the symmetric matrix $S^2 + \Omega^2$ and its eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$. A vortex is detected in the connected region where λ_2 is negative.

Chong's criterion is based on critical point theory and the eigenvalues and eigenvectors of the Jacobian. A material particle is considered to show spiraling motion if $\nabla \mathbf{u}$ has two complex eigenvectors [15]. Perry and Chong give a description of the relation between the Jacobian and its invariants P, Q and R [114].

The swirling strength parameter of Berdahl [7] and related methods [176, 14] derive values measuring swirling from the imaginary and real parts of the complex eigenvalue pair. In interactive analysis these methods are useful since they give information on the

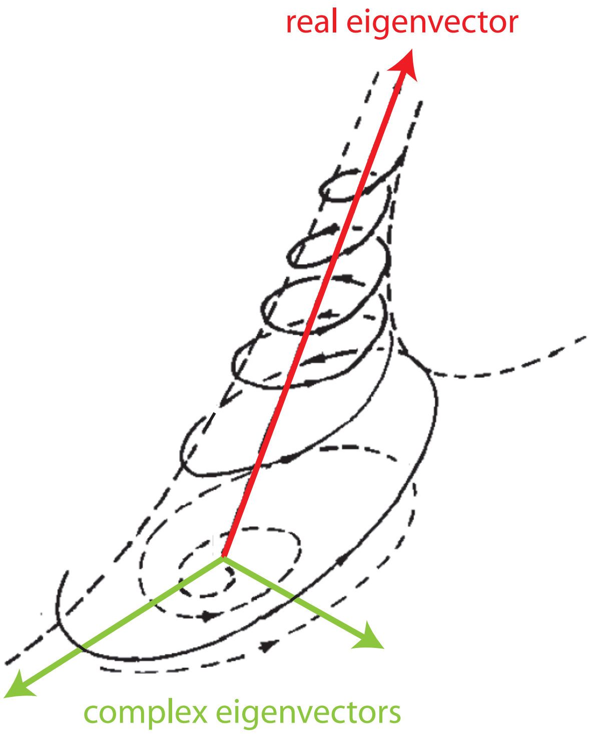


Figure 3.1: In the locally linearized field, two complex and one real eigenvalue lead to a spiralling streamline pattern. The complex Eigenvectors span the plane on which the flow is swirling. The magnitude of the complex Eigenvalue determines the speed of rotation. (Image edited based on Chakraborty et al. [14])

local strength of swirling motion. Zhou et al. reason that, when **J** has a real eigenvalue, then in the locally curvilinear coordinate system (y_1, y_2, y_3) spanned by the vectors $\mathbf{v_r}$, $\mathbf{v_{cr}}$ and $\mathbf{v_{ci}}$ and locally translating with the fluid particle, the instantaneous streamlines are given by

$$y_1(s) = y_1(0)e^{\lambda_r s},$$
$$y_2(s) = e^{\lambda_{cr} s}[y_2(0)cos(\lambda_{ci} s) + y_3(0)sin(\lambda_{ci} s),$$
$$y_3(s) = e^{\lambda_{cr} s}[y_3(0)cos(\lambda_{ci} s) - y_2(0)sin(\lambda_{ci} s).$$

It can be seen that in the time-frozen field the flow is swirling in the plane spanned by (v_{cr}, v_{ci}). The swirling strength is then the magnitude of λ_{ci} and we can use λ_r as a measure for stretching or compression along v_r. Figure 3.1 gives an illustration.

Some of these feature detectors use the same building bricks as others but calculate the final result in a different way. Especially the parallel vectors operator [122] and the critical point criterion of Chong et al. [15] are common ”building parts” for feature detection algorithms. For example by combining the two we get the detector introduced by Sujudi and Haimes [140]. This shows how multiple views and brushing can widen the possibilities of interactive analysis: by implementing the building bricks the user can combine these in standard ways using default values or combine them in new ways when the standard methods do not give satisfying results.

The classical approach to vortex feature detection are the Eulerian quantities stated above. For a three dimensional velocity field $\mathbf{u}(x,t)$, available vortex criteria use the velocity gradient decomposition

$$\mathbf{J} = S + \Omega,$$

where $S = \frac{1}{2}(\mathbf{J} + \mathbf{J}^T)$ is the rate-of-strain tensor and $\Omega = \frac{1}{2}(\mathbf{J} - \mathbf{J}^T)$ is the rate-of-rotation or vorticity tensor. Hunt et al. [56] define the Q-criterion: an element of the fluid is part of a vortex if $Q = \frac{1}{2}(||\Omega||^2 - ||S||^2)$ is larger than zero. That is, the magnitude of rotation is larger than the magnitude of strain. This criterion is refined by the λ_2 criterion of Jeong and Hussain [62]. It requires the eigenvalues of the symmetric tensor $S^2 + \Omega^2$ ordered as $\lambda_1 \geq \lambda_2 \geq \lambda_3$ to fulfill the condition $\lambda_2 < 0$. Chong et al. [15] introduced the δ-criterion, which requires the complex component λ_{ci} of one eigenvalue of $\mathbf{J}$ to be larger than zero. We can see that the rate-of-rotation tensor plays an important role for these Eulerian quantities. Haller [46] describes vortices through the stability of manifold structures which are related to fluid trajectories. The M_z criterion [46] can be considered as an accumulation of a local measure along a trajectory. Haller suggests to analyze the strain acceleration tensor

$$M := \dot{S} + S \cdot \mathbf{J} + \mathbf{J}^T \cdot S$$

where $\dot{S} := \partial_t S + (\nabla S)v$ is the material derivative of strain. In simple terms, Haller proves that a position, where M_z is positive definite, indicates this position to belong to structures which enhance finite-time turbulent mixing. Haller also introduces the notion of objectivity from continuum mechanics to vortex feature extraction. It has commonly been stressed that a vortex detector has to be Galilean invariant, i.e., it has to be invariant to translations of the reference frame (which leads to the rejection of closed trajectories as a vortex criterion). Haller argues that for special cases, such as strongly interacting vortices, and special experimental setups, such as strongly rotating fluids in a fast enough rotating tank, the vorticity tensor Ω will pronounce the whole fluid as a single vortex. Therefore he requests a vortex detection criterion to be objective, i.e., to be invariant under changes of frame-of-reference of the form

$$\tilde{\mathbf{x}} = Q(t)\mathbf{x} + b(t).$$

$Q(t)$ is a time-dependent proper orthogonal tensor, and $b(t)$ is a time-dependent translation vector. Sahner et al.[127] discuss the implementation of the M_z criterion and mention a second problem: since the material derivative of strain is involved we need to estimate second order derivatives. This introduces a serious amount of noise and degrades the accuracy of the detector for non-analytic data. Therefore they suggest to switch to an accumulation step, where the local information is combined along the trajectories by counting the number of steps where the detector response is positive. Meyers et al. [99] discuss how objective rates can be constructed for symmetric tensors with focus on Eulerian strain rates. The fact that Ω is not objective motivates focusing on the properties of strain in the fluid.

When we extract structures in flow fields from a Lagrangian perspective we can find different approaches: the first is based solely on the analysis of particle trajectories and their relations. The finite time Lyapunov exponent (FTLE) is an example from this class. In the field of chaotic systems, Lapeyre [82] studies the Lyapunov vectors and exponents to understand tracer gradient dynamics. Sadlo and Peikert [126] extract ridges from 3D FTLE for the extraction of Lagrangian coherent structures (LCS). Garth et al. [37] present a method for the direct visualization of 2D FTLE information which results in expressive images of time-dependent flow. In recent work they show how the extraction process for height ridges of the FTLE field [125] and for direct visualization [37] can be done efficiently. Cucitore et al. [20] suggest a non-local measure of swirl to extract vortices. Jiang et al. [64] search for trajectories rotating about a common axis to verify the existence of a vortex while Sadarjoen and Post [124] compute curvature centers of trajectories. In earlier work Lugt [92] requires a vortex to be a portion of the fluid moving around a common axis. As an indicator for such a structure he proposes closed or spiralling pathlines.

Reinders et al. [117] use a graph view to show the development of flow features over time and to indicate events such as birth, death, and annihilation of features. Bauer et al. [6] discuss the tracking of vortices in scale space, which improves the consideration of important features. Garth et al. [38] show the movement of singularities relative to an axis, which is of special importance compared to the others. Theisel and Seidel [144] introduce the concept of the feature flow field and use it to improve feature tracking: the paths of the critical points are tracked as the streamlines of a new vector field, i.e., the feature flow field constructed from the original vector field.

The idea of considering pathlines when analyzing time-dependent flow data is not new as such. Theisel et al. [145] present a pathline-oriented approach to extracting the topology of 2D time-dependent vector fields – similar to a streamline-based approach, they distinguish features according to attracting, repelling, or saddle-like behavior.

In general, we observe a new motivation in the field to approach even very complex cases in 3D time-dependent flow visualization. Peikert and Sadlo [112] discuss feature-based visualization for the investigation of vortex rings and vortex breakdown bubbles in recirculating flow, and Tricoche et al. [149] describe a slice-based visualization for understanding intricate flow structures where the slices are placed orthogonal to trajectories of the flow.

Another interesting class of approaches are physical criteria (instead of geometric ones) for feature extraction. Banks and Singer [4] propose a method to find vortex core lines based on a predictor/corrector method that steps through the field in the direction of the vorticity vector. At each step the normal plane is constructed and the point is reset to the nearest local pressure minimum. Jankun-Kelly et al. [61] present an improvement of this approach using a function fitting procedure to locate the extreme values, stepping along the real eigenvector of the velocity gradient. Stegmaier et al. [137] present an algorithm that combines the λ_2 method of Jeong and Hussain [62] with the predictor/corrector method of Banks and Singer. For growing the skeleton they step in the direction of the vorticity vector. In this context of physical approaches, several more methods have been presented, e.g., the Q-criterion of Hunt et al. [56], also known as the elliptic version of the Okubo–Weiss

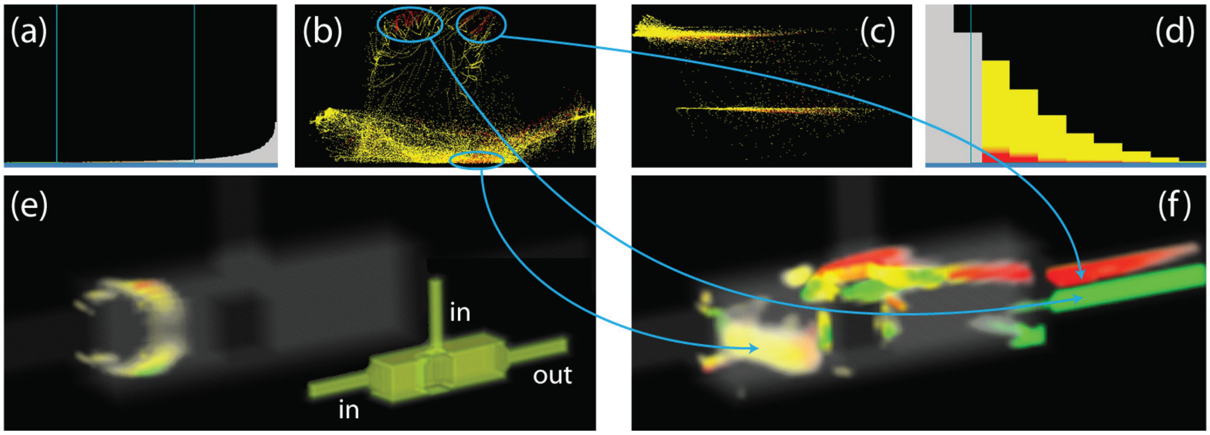


Figure 3.2: The combination of non-binary feature detectors and interactive visual analysis: (a) The histogram shows that only few data items trigger strong response of the λ_2 feature detector, (b) a scatterplot of temperature against velocity shows differences between the detected vortices, (c) a scatterplot of turbulent energy against relative pressure shows differences between the vortices near the outlet and the inlet (d) a derived distance to surface measure removes occluding elements located at the boundary. (e) This weak vortex in an early time step of the simulation would have broken into several parts using iso-value based visualization; (f) in the 3D overview of the situation we can see how the vortices differ in rotation speed and direction.

criterion by Okubo [111] and Weiss [160], or the extension of considering acceleration terms by Hua et al. [55], which includes temporal derivatives and expresses the feature extraction process from the Lagrangian perspective.

3.3 Non-binary vortex detectors

The basic insight that led to the development of the new interactive feature extraction framework is that both automated and interactive approaches have their limits that we can deal with by combining both approaches. On the one hand, interactive extraction and data analysis is limited in terms of feature complexity. It is simply not possible for the user to find features that have too many dependencies or involve elaborate computations. On the other hand, human users are very good in dealing with incoherent information, uncertainty, and fuzzy concepts. In fact a user will very often not know what he or she was looking for before he has found it [155]. Automated feature extraction algorithms (in particular vortex detectors) have received intense attention and it is reported that they are all able to detect vortices under the right circumstances in a fast and robust manner.

The detectors currently available may fail to detect non-standard features or features that do not share the same frame of reference as the detector. Furthermore, many feature detection algorithms still need a considerable amount of parameter tuning (e.g. iso-surface values) to get good results. Chakraborty et al. [14], for example, stress the importance of using appropriate thresholds when trying to find vortices with specific properties like compactness along the axis or vortex strength. When multiple features are present there may be no single threshold parameter to detect all flow features at once. Therefore our main motivation to combine visual analysis and algorithmically derived features is to add the monitoring and reasoning capabilities of the user to the exactness and computation power of the computer. See Table 3.1 for an overview.

3.3.1 Derivation of the Continuous Detectors

To investigate flow data using focus+context visualization we need to adapt the criteria in a way that they express a notion of vorticity as fuzzy-logic attributes. There are some requirements for properties of our fuzzy sets: they should **extend** the binary classifiers (i.e. they are 0 where the binary classifier outputs no vortex and 1 where a vortex would be detected with full certainty). We do not force a measure to be extensive when it is in general also used for other purposes and the user expects different behavior. Otherwise we can often guarantee extensiveness by scaling the output to the $[0,1]$ interval accordingly. We do not expect sharp feature boundaries at the scale of typical flow simulations. It is sensible to expect the classifiers to be **continuous**. To balance these two criteria we model the range of values where the detector only partially detects a vortex. Furthermore, the classifier should be **monotonic**, since this allows for good intuitive combination behavior using the classical fuzzy norms to model the Boolean 'and' and 'or' operations using the 'min' and 'max' functions [72].

- We define a fuzzy local extremum around a point $\mathbf{x}$ with numerical attribute $a(\mathbf{x})$ with minimum $\min = \min\{a(\mathbf{x}) | \mathbf{x} \in \mathrm{P}_r\}$ and maximum $\max\{a(\mathbf{x}) | \mathbf{x} \in \mathrm{P}_r\}$ in a neighborhood as

  $$extremum_{Fuzzy}(a(\mathbf{x})) = \begin{cases} 0.5 & : \quad max = min \\ scale_{\mathrm{P}_r}(a(\mathbf{x})) & : \quad otherwise \end{cases}$$

  This is, the relative position of the attribute value inside the interval of the minimal and maximal attribute values in the neighborhood of $\mathbf{x}$. Of course a local attribute extremum defined like this is dependent on the extent of the neighborhood that defines localness. This attribute was added because in many typical flow situations it is possible to find the central regions of a vortex by restricting the detected region further using the additional condition of locally minimal pressure.

- High vorticity magnitude is the good starting point for analysis of vortex structures. The actual classifier is 'large vorticity' and has reportedly been fairly successful in free shear flows [62], but there is no predefined value from which on vorticity is considered large, therefore the SimVis approach allows the user to select which values he or she considers as 'large'.

  $$\omega_{Fuzzy} = scale_{\mathrm{D}}(\omega)$$

Method	Properties	Benefits in combined application
Vorticity magnitude	fast computation	can be used for preselection of relevant cells in large data sets using a relaxed threshold, responds to sheer
N_k	independent of vorticity magnitude	can be used to cross-check regions selected using vorticity magnitude
Normalized helicity	signed, gives direction of rotation	combination with other detectors helps to distinguish between connected regions of counter-rotating vortices, ignores transversal vortices
Hunt's Q	no computation of eigenvalues necessary, in many cases equivalent to λ_2	numerically more stable for noisy data, comparison with λ_2 for confidence
λ_2	based on eigenvalues of a symmetric matrix, does not distinguish between connected vortices	very good performance, reliability confirmed in many publications
Eigenvalue related methods	detailed insight into vortex properties, need eigenvalues and eigenvectors of the rotation matrix, may introduce numerical issues, more costly	can restrict detected vortex regions to portions of fast/slow spiraling motion, give information on axial stretching and orbital compactness

Table 3.1: Comparison of detector properties.

- Levy et al. [91] introduced normalized helicity H_n. In the limiting case where $\mathbf{u}$ is parallel to ω we have $H_n = \pm 1$ and we do not scale this measure since this is the expected behavior

$$H_n(\mathbf{x}) = \frac{\mathbf{u}(\mathbf{x}) \cdot \omega(\mathbf{x})}{|\mathbf{u}(\mathbf{x})||\omega(\mathbf{x})|}$$

- The characteristic equation for $\mathbf{J}$ is given by

$$\lambda^3 + P\lambda^2 + Q\lambda + R = 0$$

  where P, Q and R are the three invariants of $\mathbf{J}$, defined as $P = -tr(\mathbf{J})$, $Q = -\frac{1}{2}(P^2 - tr(\mathbf{JJ}))$ and $R = -det(\mathbf{J})$. The invariants map both to topological critical point features and tell about physical properties of the flow (e.g. $P = 0$ holds for incompressible flows) and can be a useful addition to the other views.

- Hunt et al. suggest regions of positive Q as vortical regions [56] where the magnitude of the rate-of-rotation tensor Ω exceeds the magnitude of the rate-of-strain tensor S. The larger the difference between Ω and S the higher the certainty that we have found a vortex:

$$Q_{Fuzzy}(\mathbf{x}) = \begin{cases} 0 & : \ ||\Omega||^2 - ||S||^2 \leq 0 \\ scale_{\mathbf{D}}(||\Omega||^2 - ||S||^2) & : \ otherwise \end{cases}$$

- The parallel vectors operator was intended to extract vortex core lines and ease the notation of vortex extraction algorithms [113]. A vortex core has the necessary condition of two characteristic vectors of the flow to be parallel. Therefore the parallel vectors operator, similar to local pressure minima, can help to restrict the regions of vorticity to be closer to the core. The parallel vectors operator uses the flow vector $\mathbf{u}$ and additionally derived quantities, we found $a = \nabla\mathbf{u} \cdot \mathbf{u}$ and $b = \mathbf{JJu} + T\mathbf{uu}$ to be the most useful (T is the tensor of second derivatives). The vector a gives a formulation of the vortex core line criterion by Sujudi and Haimes [140] using vector parallelism. The vector b can be considered as a higher-order extension of the Sujudi and Haimes criterion. For more details and a derivation of the vectors a and b we refer to Chapter 6. Using a dot product on the normalized vectors we get a measure of parallelism:

$$PV(x) = \frac{\mathbf{u}(\mathbf{x})}{|\mathbf{u}(\mathbf{x})|} \cdot \frac{a(\mathbf{x})}{|a(\mathbf{x})|}$$

- Complex Eigenvalues: Critical point theory [15] tells us that a particle will show rotating motion if $\mathbf{J}$ has two complex eigenvalues. In the related regions vorticity is sufficiently strong to cause the rate-of-strain tensor to be dominated by the rate-of-rotation tensor. This can be tested by checking the characteristic polynomial of $\mathbf{J}$ for a positive discriminant as we know from Cardan's solution for cubic polynomials. Berdahl and Thompson [7] used the fact that in a locally curvilinear coordinate system spanned by the eigenvectors of $\mathbf{J}$ the eigenvalues give insight into the behavior of a fluid particle: If the eigenvalues are complex, then one plane will contain a focus and solution trajectories will wrap around the one real eigenvector. In terms of the eigenvalues the criterion of Chong reads as $\lambda_{ci} > 0$. This criterion was reportedly successfully combined with others [15, 176, 122].

$$Complex1_{Fuzzy}(\mathbf{x}) = scale_{\mathrm{D}}(\lambda_{ci}(\mathbf{x}))$$

A nice property of this criterion is also that λ_{ci} directly measures the strength of radial motion of a fluid element. Chakraborty et al. [14] suggest a combination of $\lambda_{ci} \geq \epsilon$ and $\kappa \geq \lambda_{cr}/\lambda_{ci} \geq \delta$ where ϵ, δ and κ are positive thresholds, to include a notion of orbital compactness of the vortex. From this results an extension to the complex eigenvalue criterion:

$$Complex2_{Fuzzy}(\mathbf{x}) = \begin{cases} 0 & : \ \lambda_{ci}(\mathbf{x}) < \epsilon \\ scale_{\mathrm{D}}(\frac{\lambda_{cr}(\mathbf{x})}{\lambda_{ci}(\mathbf{x})}) & : \ otherwise \end{cases}$$

We suggest to use the two criteria in combination.

- Jeong and Hussain [62] proposed the second eigenvector λ_2 of $S^2 + \Omega^2$ as a criterion for finding vortex regions. A vortex is found in regions where λ_2 is smaller than zero. We know that λ_2 requires $\mathbf{x} \cdot \Omega^2\mathbf{x}$ to be greater than $\mathbf{x} \cdot S^2 x$ in one eigenplane of $S^2 + \Omega^2$. This is only critical if $\lambda_1 > 0$ since we know that in case $\lambda_1 < 0$ these two are balanced in all directions anyhow. When $\lambda_1 > 0$ the modulus of λ_2 gives indication of the balance of $\mathbf{x} \cdot \Omega^2\mathbf{x}$ and $\mathbf{x} \cdot S^2 x$ in one eigenplane of $S^2 + \Omega^2$.

$$\lambda_{2Fuzzy}(\mathbf{x}) = \begin{cases} 0 & : \lambda_2(\mathbf{x}) \geq 0 \\ 1 & : \lambda_1(\mathbf{x}) \leq 0 \\ scale_{\mathrm{D}}(-\lambda_2(\mathbf{x})) & : otherwise \end{cases}$$

3.3.2 Integration in the interactive framework

The traditional way of integrating feature detectors in flow visualization is to use isosurfaces to represent the extracted structures. This is not always appropriate in the case of a vortex. The notion of a continuous degree of interest function tries to capture two important properties of features: the first is that flow features are not sharply defined and the second is the uncertainty that is inherent in feature extraction. Very often we cannot be absolutely confident that each data element we selected really is part of the feature we are looking for. This partial inclusion is represented by rendering data elements with opacity values according to the degree of interest they obtained after all brushes and attributes are summed up. The features are visually represented with their inherent fuzziness and the user is not tempted to assume a sharp distinction between laminar and turbulent flow.

SimVis suggests a layered work flow. The information drill-down conceptually starts on the direct data access level. The user can get an overview of the distribution of attribute values like temperatures, pressures or flow velocities in the simulation data. This allows us to gain an intuition on the situation in a straight-forward manner.

The second level is analyzing relations between different attributes and different subvolumes of the simulation. This involves using linked scatterplots of different attributes, interactive brushing and linked views. Feature complexity on this level is still limited to choosing intervals of attribute values to be part of the feature. The selection of specific value ranges involves specifying a degree of interest in parts of the data that exhibit the characteristics selected. In fluid dynamics applications it is very often the case that combinations of different attributes are of interest. For example, we will see in the next section that in the design of a cooling jacket very slow or fast portions of the flow are of critical importance when they have extremely high temperature. The linked aspect of the scatterplots allows us to get an intuition for the relations between multiple attributes.

The third level involves the computation of derived features from one or multiple attributes. On this level general properties of the data like correlation between attributes, time-derivatives of attributes, or smoothing operations result in additional synthetic attributes. After derivation, these synthetic attributes behave like the other attributes and can be brushed, mapped to color, and serve as input information for further derivations. In an iterative manner the user can now use the operations of the higher levels of inspection to gain an understanding of these attributes.

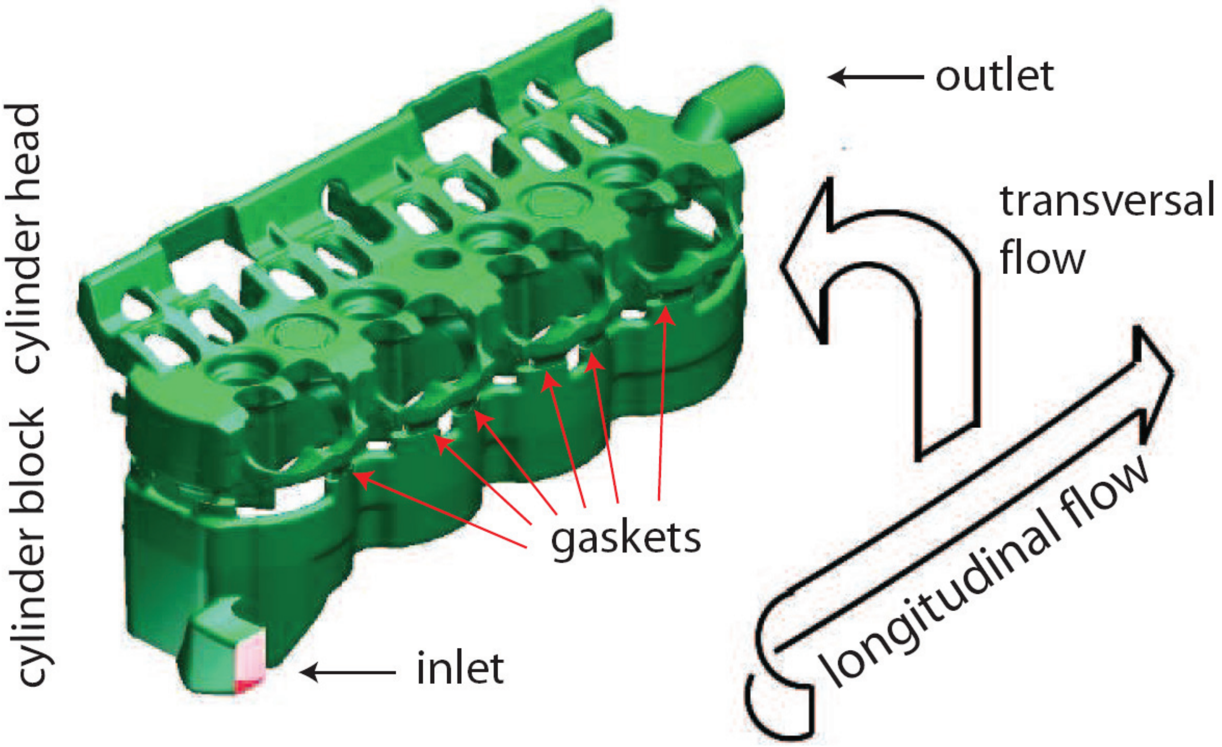


Figure 3.3: The major components of the flow through the cooling jacket include a longitudinal component from inlet to outlet and a transversal component in the upward-and-over direction through the gaskets [50].

The fourth level tackles specialized feature extraction. After the user has gained insight into the features of the flow it is now possible to choose appropriate feature detectors to extract where and when important events in the flow occur. At this point the interactive aspect of feature detection comes into play: since the interaction with the derived fuzzy set is possible in real-time, one can configure the sensitivity and related thresholds of the detectors interactively. After visual inspection and parameter tuning these features are ready for access in the higher levels of the work flow. They are ready to be inspected in detail to understand their properties using the upper levels of the SimVis work flow. The other way round the extracted features may be useful for exclusion. An engineer looking for properties of the laminar portions of the flow can extract a measure of 'vortexness' and brush only those parts that possess low membership values for this fuzzy set.

The possibility to combine different features has the benefit of being able to express complicated vortex criteria using simple combinations. The vortex core extraction operator of Sujudi and Haimes [140], for example, can be expressed as a combination of the parallel vectors operator [113] and the discriminant criterion of Chong [15]. The vortex core extraction algorithm of Miura and Kida [100] is a combination of local pressure minima and parallel vectors. Furthermore combining multiple detectors can help to compensate

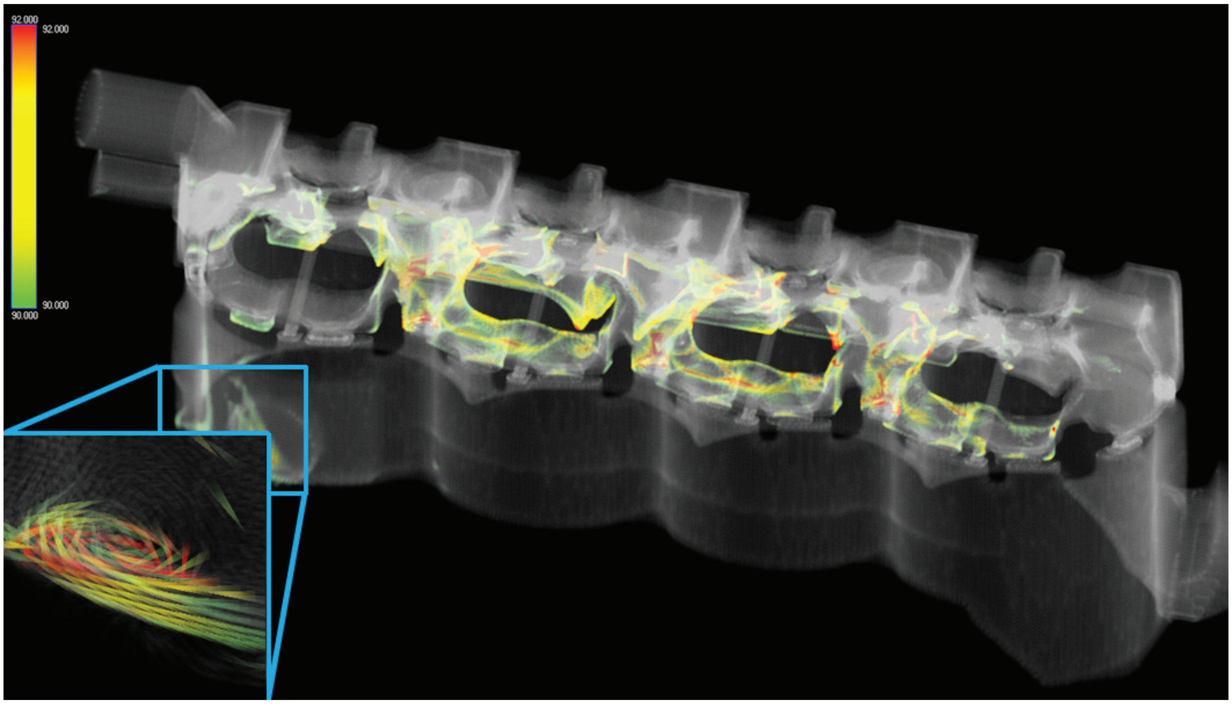


Figure 3.4: A feature-based, focus+context visualization showing regions of near stagnant, hot flow with medium to high levels of the λ_2 vortex detector. On the left we see the regions in focus. The zoomed view shows details, especially the extent of critical (red) volume. The magnification reveals a vortex structure at this point.

unwanted properties of one detector. The λ_2 method for example does not always distinguish nearby vortices. This can often be compensated for by deriving a helicity attribute to differentiate nearby vortices by means of rotational direction.

In Figure 3.2 we see an illustration of the discussed qualities: in (a) a histogram of the λ_2 values (high λ_2 to the left and low λ_2 to the right) shows that only a small proportion of cells in the data set exhibit high feature values but a substantial fraction of the data is mapped to non zero vortex membership values. (b) In the scatterplot – mapping temperature against velocity – we can see that the cells belonging to vortices (points highlighted in red) cluster for each vortex. (c) A scatterplot of turbulence against relative pressure shows that there are two main pressure levels and we can interactively check that pressure is lower near the outlet.

The derived feature attributes also add value through the linking functionality – selecting high feature values highlights these data elements in the other views. In part (e) of the image we can see how the mapping to fuzzy values of featureness changes aspects of the visualization: the horseshoe vortex is detected with high fuzzy values in the front part where it is strongest. Near the two ends of the horseshoe it is not very typical and the cells are assigned values close to zero by the detector. In a visualization using iso-value surfaces this vortex would break into several parts due to this effect, but it is conceivable as a whole as a fuzzy-region. This histogram of distance to surface measure (d) was used to remove obscuring boundary cells from focus. In part (f) we see the features that were found in a 3D view color mapped with helicity to display information on the direction of rotation and its strength.

3.4 Application Evaluation – a Cooling Jacket

In the following we briefly sketch two examples of interactive visual analysis and exploration of fluid flow through a cooling jacket. Computational fluid dynamics software is used to inspect and improve the design process and we know that engineers invest large amounts of time to optimize the geometry of cooling jackets. In this application evaluation we continue work done by Hauser et al. [50] and Laramee et al. [87] where regions of turbulence were not considered. For supplementary documentation and results to this case study, in particular video, high-resolution images and other material, please refer to the web page for this thesis [1].

The cooling jacket in focus (see Figure 3.3) is used with a four cylinder engine. The complex shape of the cooling jacket is influenced by multiple factors including the shape of the engine block and to optimal temperature for the particular engine. The cooling jacket geometry consists mainly of three components: the cylinder head on top, the cylinder block on the bottom and a thin component connecting the cylinder head and block, called the gasket. The cylinder head is responsible for transferring heat away from the intake and exhaust ports at the top of the engine block. The cylinder block is responsible for heat transfer from the engine cylinders and for even distribution of flow to the head. Between the cylinder head and block lies the cooling jacket gasket. It consists of a series of small holes that act as conduits between the block and head. These ducts can be quite small relative to the overall geometry but nonetheless are very important because they are used to govern the motion of fluid flow through the cooling jacket. There are two main components to the flow through a cooling jacket: a longitudinal motion lengthwise along the geometry and a transversal motion from cylinder block to head and form the intake to the exhaust side. Important design goals for the mechanical engineers are to obtain an even distribution of flow to each engine cylinder and to avoid regions of stagnant flow to ensure good overall heat transport.

3.4.1 Reduced heat transport due to turbulent motion

In order to find regions of the geometry that might need refinement we search for regions where slow flow motion and high temperatures come together. The resulting image is still cluttered from very small regions (that pose no problem to engine operability) and therefore difficult to understand. A similar situation appears in the earlier investigation of this property in a recent publication by Laramee [87]. From background knowledge we know that vortices may diminish heat transport. Figure 3.4 left illustrates regions in focus after brushing temperatures above the range of optimal conditions around 363° K in

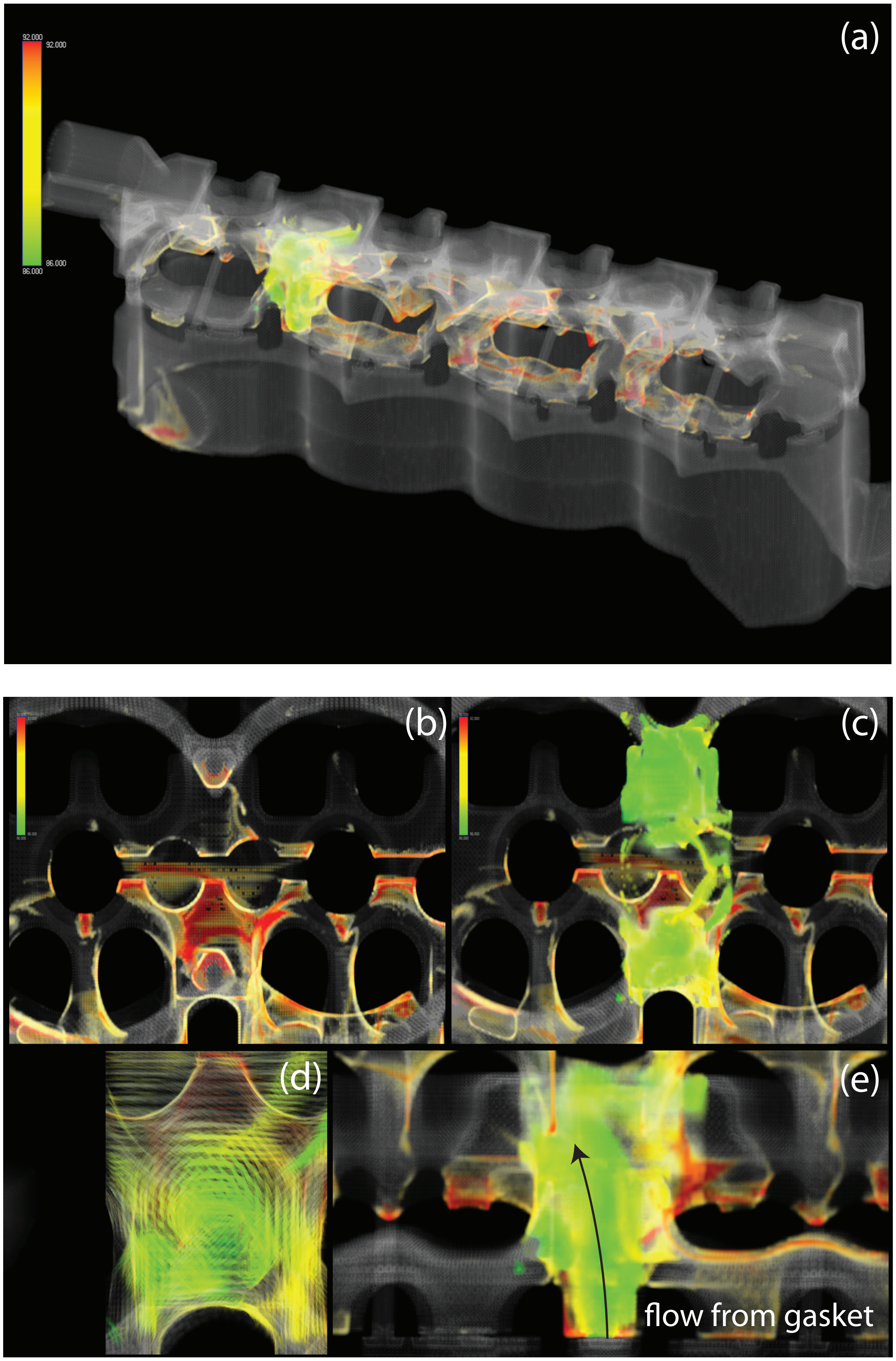


Figure 3.5: (a) We have highlighted the turbulent region of interest. (b) From the top we can see that a large patch of the surface contains high temperature values. (c) We additionally include the two regions of turbulence below the surface. The lower one has smaller extent and not enough cooling fluid is transported to this point. (d) A magnified view of the lower turbulent structure. (e) The turbulent structure caused by a gasket, viewed from the side.

combination with a derived vortex measure. The result is a less cluttered image, showing larger undesirable regions, where the cooling fluid is less effective in transporting heat away. There is a large connected region of low heat transport visible in the lower right part of the overview picture. After zooming in and visual inspection of flow behavior we can conclude that the unwanted formation of a vortex is probably the cause for this situation. Fortunately we know from engineers that the region is small enough and operation of the engine in this state remains safe.

3.4.2 A situation of improved heat transport due to turbulent motion

Even though the vortex in the preceding paragraph caused problems, there are cases when engineers intentionally induce swirling motion when designing geometry. In Figure 3.5 we see a region of turbulent motion that was provoked deliberately by choosing an appropriate gasket geometry. Why is this so? The parts of the geometry close to the inner boundary where the engine cylinders are located are critical parts of the volume. Turbulent motion mixes the fluid and transports the hot portions away from the boundary replacing them with cooler elements of the fluid. The large overview part of the image (a) tries to give a feeling for the three dimensional geometry of such a region of high turbulence behind a gasket. The next part of the image (b) shows a zoomed view from above. The two ring shaped parts of the geometry hold a cylinder, and we can see that one side has mapped high temperature values (red) while the other is not in the critical temperature interval. Now we include the regions of turbulence caused by two of the opposing gaskets into the view (c). It becomes clearly visible that the side with good temperature values is well covered by the turbulent motion. On the other side the situation is different: the hot parts are not covered by the turbulent motion and receive not enough cooling fluid. In (d) we see an enlarged view of the vortical motion. Due to fuzzy-attribute mapping one can get a good impression of the relative strength of turbulence. The last part of the image (e) shows another view on the region of turbulence from the side.

3.5 Chapter Conclusions

In this chapter we have shown that feature detection algorithms benefit from continuous representations in the context of interactive visual analysis – both with regard to effectiveness and efficiency. We have discussed how the criteria are fitted into the SimVis framework and contribute to the overall usefulness of the system. Additionally, we have applied this new approach successfully to data from the engineering domain. The results of this chapter lead to additional questions which we will focus on in the next chapter. We will go into further detail on how the different detectors and measures can be combined interactively and contribute to each other. We investigate the interrelationships of the criteria from the viewpoint of visualization. We will see how the interactive combination of several criteria can improve the understanding of complicated vortices and other features of turbulent flow.

Chapter 4

Cross-Detector Analysis

> ”Logic’s useless unless it’s armed with essential data.” (Frank Herbert 1920 – 1986)

In this chapter we discuss the application of multiple linked views for advanced vortex investigation in flow data. Local feature detectors and additional measures integrate into an interactive flow feature detection system based on multiple linked views. We discuss how linking and brushing of derived feature information can lead to increased performance in interactive feature analysis. In a case study we demonstrate how the possibility to integrate the information of multiple detectors leads to sound understanding of vortex type features. Enhanced credibility and combined advantages of several detectors can be achieved by uniting the results of multiple feature detectors in multiple coordinated views.

4.1 Motivation

Flow is everywhere. Gaseous or liquid flows occur on all scales from capillary flows transporting oxygen to a muscle to the motion of plasma inside the sun. One fundamental property of flow is the ubiquitous nature of turbulence. A century ago many people believed there might be some kind of ’universal theory of turbulence’ that was supposed to be valid under a wide range of circumstances. It is now generally agreed that such a theory does not exist [21]. Since there is no general theory that describes turbulent fluids many engineers do not fully trust automated feature extraction methods. Therefore, in the context of flow visualization, interactive approaches are an important tool to understand the complex movements inside the flow and the changes of related attributes such as temperature and pressure.

Computational fluid dynamics (CFD) methods output simulation data with ever increasing complexity. The engineer wants to answer questions like: What is the flow like now? How does the flow change over time? Where does the flow lead to? As long as the turbulence is not developed into chaos, there still exist coherent structures that allow predictions and can give insight in the behavior of the flow [57]. Vortices are an important structure and by now we know a lot on their behavior. It is widely agreed that vortices belong to the most important coherent features in flow fields. Therefore tools that can help to find and analyze the parts of the fluid that exhibit swirling motion can greatly help the engineer or physicist to gain insight into the properties of the fluid motion.

There exists a wide range of local and global feature detectors that can find vortices in the flow under the right conditions [115]. They have individual advantages and can outperform others in certain situations. Some of the intrinsic properties of a vortex detector include:

- Representation of the vortex: vortex core line methods extract line type features giving precise insight into the location of the vortex core. Vortex region extraction algorithms extract an approximate volume that describes the extent of the connected region of swirling motion.
- Galilean invariance: a vortex detection algorithm that is Galilean invariant is valid in all inertial frames of reference [62]. A detector that has this property is able to find vortices also in cases where a steady current in one direction is added to the flow field and masks the vortical motion of the fluid.
- Robustness: many detection schemes are considered to be prone to errors in special configurations of the flow. Robustness can vary with respect to noise in the data and the size of the grid in unique ways for each detector [118].

In the remainder of this chapter we will see how multiple linked views can give access to many of the useful properties of the different detectors at once, but in the scope of this chapter we will limit the analysis to vortex region extraction based approaches. These are of interest, since vortices are especially difficult to detect and even approved extraction algorithms can differ in their outputs significantly.

This is also an example on how visual analysis can benefit from multiple views that allow the user to build a complex mental model of the properties of the flow. Until recently the rising computing power has led mainly to rising complexities in the data generated. This is of course a highly favorable development, but with rising complexity of the data we also wish for more powerful analysis tools to benefit from the information in the data. In this chapter we suggest interactive visual analysis as a way to to meet the challenge of rising complexity and suggest to include automated feature detectors into this process.

4.2 On Visual Interactive Analysis

Multiple views and smooth brushing improve the flow feature analysis. In Figure 4.1 we see an example for this approach. The two 3D views show (a) regions of high local vortex-measure and (b) low velocity and high temperature in focus and serve as reference. (c) The scatterplot of a vortex detector (λ_2) vs. its local extrema can be used to get an overall impression on the distribution of detector response values and brushing can be used to select vortical regions. In all attribute views the brushed elements appear in red. Elements that are colored black are not selected in any of the views. Dots are colored

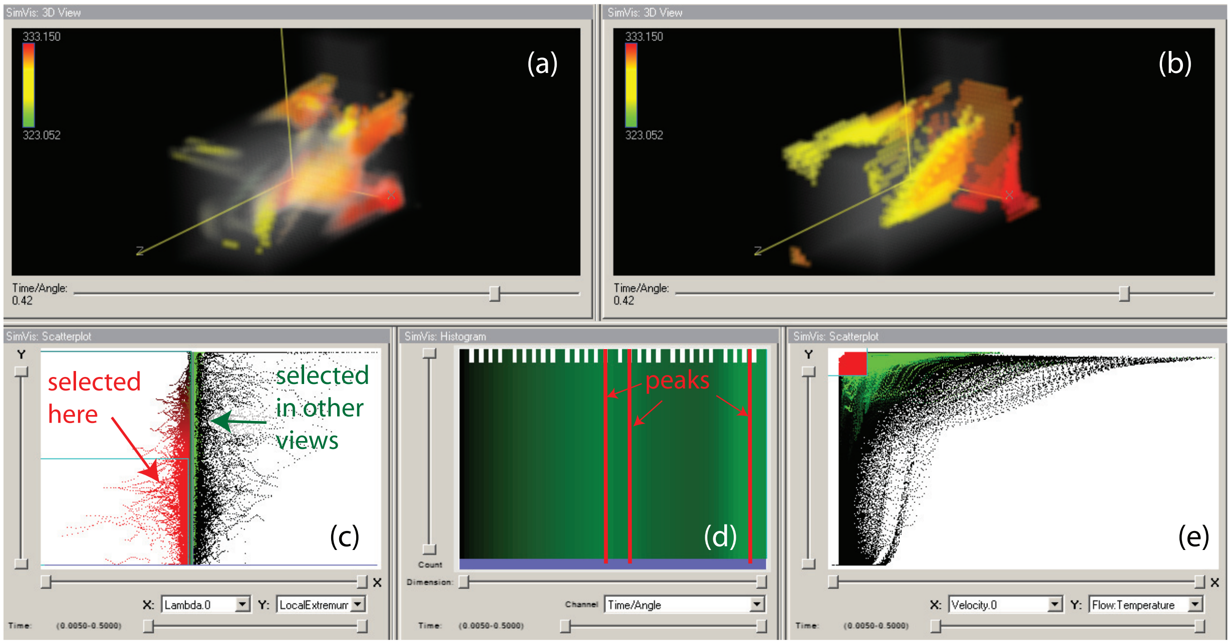
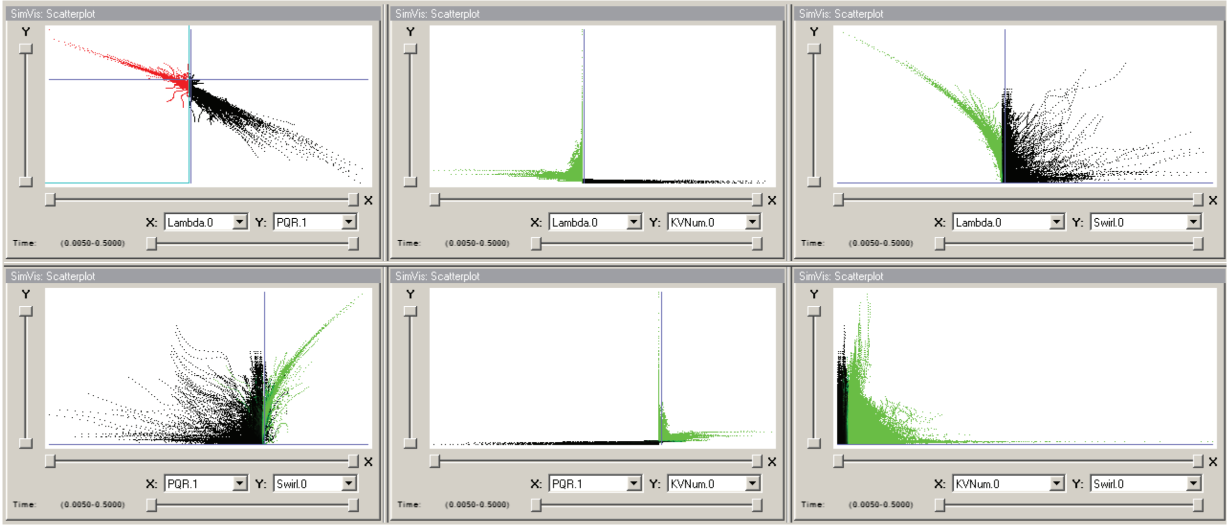


Figure 4.1: Multiple views and smooth brushing for vortex analysis. (a) A 3D view showing regions of high local vortex-measures. (b) A 3D-view with low velocity and high temperature in focus serves as reference. (c) A scatterplot of the λ_2 vortex detector output vs. local extrema of λ_2. (d) The histogram shows the development of the features selected in the scatterplots and reveals three peaks of turbulence over time. (e) A scatterplot of velocity vs. temperature allows to compare attribute values of the selection (red) with those of the vortex regions (green).

green to show data items that are not selected in the current view but in other linked vies, i.e. they show the difference between the current selection from selections in other views. (d) The histogram shows the development of the features selected in the scatterplots (using color linking between views) and reveals two peaks of turbulence over time. This could be time steps that qualify for further investigation. (e) In this scatterplot of velocity vs. temperature we compare attribute values of the selection (red) with those of the vortex regions (green).

Details on the vortex detectors and references can be found in previous Section 3.2. The combination of multiple classifiers has received much attention in the machine learning and and statistic communities. Lam [81] discusses methodologies for the combination of multiple classifiers to obtain improved recognition results. Roli and Kittler [120] present an overview on fusion methodologies.

In their standard definition the common feature detectors give a binary classification for each volume element to the classes 'vortex' and 'no vortex'. In the previous chapter we have discussed that it can be more useful to consider the whole spectrum of local feature detector response instead of using a binary threshold. Especially in combination with direct volume rendering more information is communicated by also showing the gradient



X: Lambda.0 Y: PQR.1
Time: (0.0050-0.5000)

X: Lambda.0 Y: KVNum.0
Time: (0.0050-0.5000)

X: Lambda.0 Y: Swirl.0
Time: (0.0050-0.5000)

X: PQR.1 Y: Swirl.0
Time: (0.0050-0.5000)

X: PQR.1 Y: KVNum.0
Time: (0.0050-0.5000)

X: KVNum.0 Y: Swirl.0
Time: (0.0050-0.5000)

Figure 4.2: Combining multiple detectors using linked views: (a) λ_2 vs. Q shows that the two are closely related in this data set. (b) The relation between λ_2 and kinematic vorticity number is smaller but the sharp 'edge' in the distribution helps to specify a good threshold for λ_2. (c) The comparison with the swirl criterion verifies that we have selected a good threshold. (d-e) the pairwise relations of Q, the swirl criterion and the kinematic vorticity number can be used as a sanity check for the detected region.

information in the visualization. Nevertheless, the interaction framework still allows us to use thresholding and binary brushes.

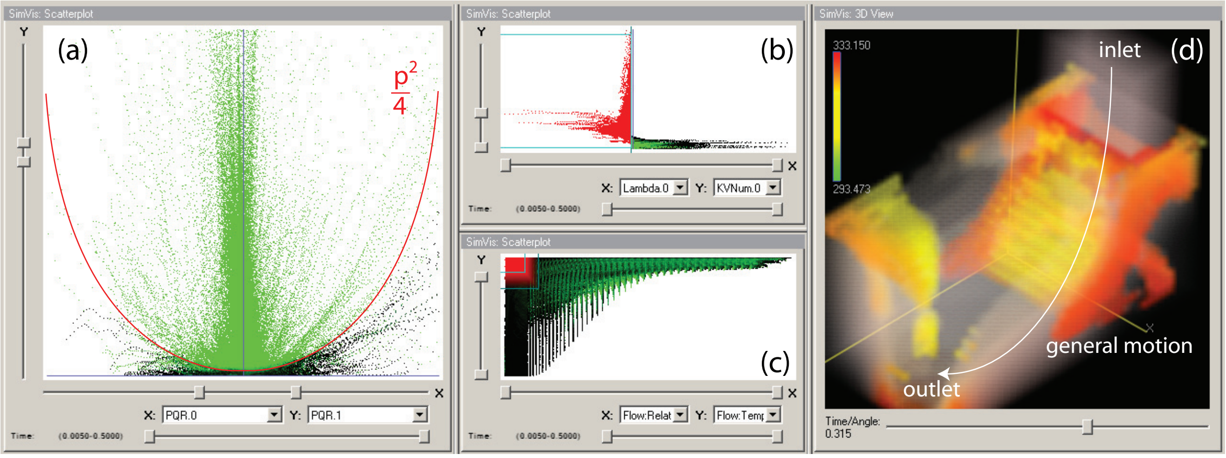
In Figure 4.2 we see that the different detectors have unique response characteristics and inspection of their interaction can help visual analysis. Since the detectors are all based on values computed from the Jacobian there is very often some correspondence.

4.3 *PQR* Plots

The analysis of critical points of the fluid field has proven to be very useful for understanding the behavior of flow fields. Perry and Chong discuss how critical point concepts can help understand eddying and swirling flow patterns [114, 15]. They use three invariants of the Jacobian, P, Q and R:

$$P = -trace(\mathbf{J}), \quad Q = \frac{1}{2}(P^2 - trace(\mathbf{J}^2))$$
$$R = -det(\mathbf{J})$$

The invariants are related to the topology of flow patterns. Therefore they give a useful complement to the other views. Since plots of P, Q and R are traditionally used for presentation and discussion in the literature and not as vortex extraction criteria we do not scale or transform them in any way but map them directly to scatterplots. Plotting



X: PQR.0 Y: PQR.1

X: Lambda.0 Y: KVNum.0

X: Flow:Relat Y: Flow:Temp

Time/Angle: 0.315

Figure 4.3: A *PQ*-plot: (a) most of the cells selected by the the OR-combination of the λ_2 method and the low-pressure-high-temperature feature lie above the parabola $q = \frac{p^2}{4}$ in the pq-plot (drawn red over the scatterplot for illustration). (b) The first feature (red) selects λ_2 values suggesting a vortex region. We can see where this differs from the second feature (green). (c) The second feature (red) selects low pressure and high temperature. We see that most of the cells belonging to a vortex unexpectedly exhibit high pressure values (green). (d) In the 3D-view we can see the region in space where we have both features present.

different planes of these invariants is a common tool when discussing flow topology features. Typically one invariant is held constant (e.g., P vs. Q with $R = 0$) such that the result can be visualized using a 2D plot, but to our knowledge this has not been done interactively using multiple linked views, where it is possible to compare the distribution of points interactively for different settings. The invariants are related to the eigenvalues of the characteristic equation by

$$det(A - \lambda I) = 0 \leftrightarrow \lambda^3 + P\lambda^2 + Q\lambda + R = 0$$

plots of P, Q and R give insight into the complex and real eigenvalues of the characteristic equation and therefore into parts of the flow which show eddying motion.

In Figure 4.3 the *PQ*-plot shows the spread of cells in the *PQ*-plane with R selected to be zero and compares this with the results of the λ_2 method. In (a) we see that most of the cells selected by the λ_2 method and the low-pressure-high-temperature feature lie inside above the parabola $q = \frac{p^2}{4}$ in the *PQ*-plot. This is the region of stable and unstable focus flow, that is a pattern of spiralling motion. In (b) the selection (red) is centered at high λ_2 values. Through linking we see where this differs from the second feature (green). (c) The selected feature (red) of low pressure and high temperature can be compared with the feature from the other views (i.e., b). We see that most of the cells belonging to a vortex exhibit high pressure values (green). This is an atypical finding, since due to the centripetal forces necessary for spiraling motion we commonly find lower pressure values

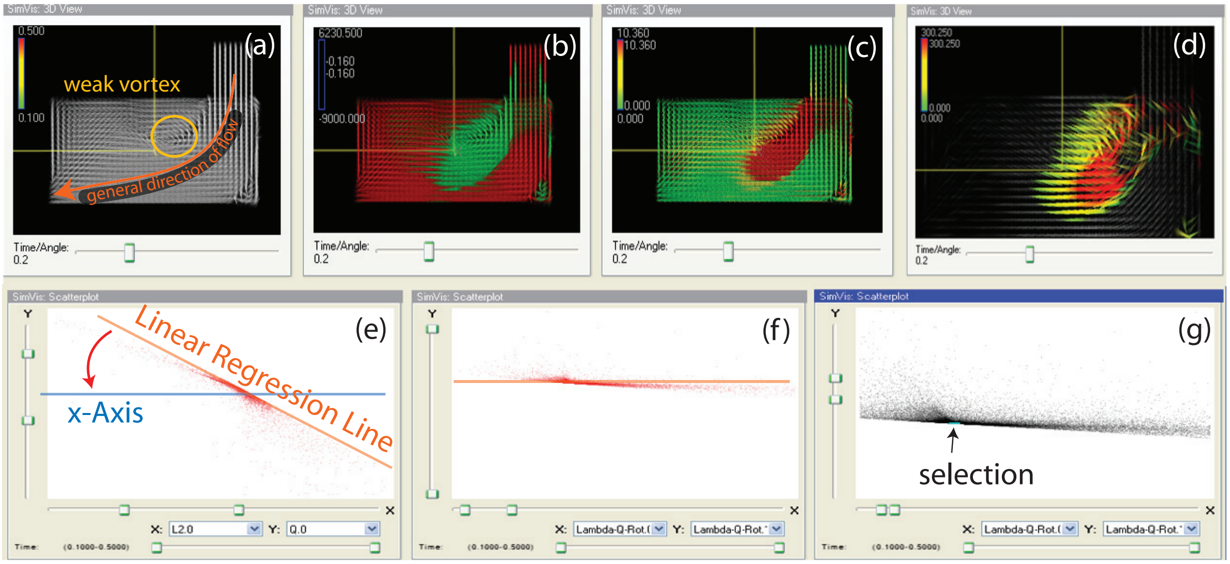


Figure 4.4: Correlating several detectors: (a) The data set contains a weak and small vortex in a slow portion of the fluid, that no single method can find satisfyingly. (b) The λ_2 method has to be set to very high sensitivity to include the vortex region. (c) The swirling strength method detects a similar region. (d) The region selected by using a combined detector that correlates λ_2 and swirling strength. (e) The rotation of the points (see text) in the scatterplot is dependent on the slope of the regression line. (f) After rotation the regression line matches with the x-Axis. (g) A rectangular selection now contains an inherent measure of correlation of the detectors.

connected with vortices. The 3D view (d) complements the analysis by showing the regions in space where we have at least one of the features present. The regions match with what would have been expected, that is to say behind the inlet, at the region on top opposite of the outlet and around the outlet.

4.4 Correlation Brushing

When trying to understand how several detectors combine we would like to understand how the response values of two detectors correlate. For this purpose we use a rotation of the scatterplot to be able to use rectangular brushing to find correlating parts of a feature. We calculate the offset and slope of the regression line and rotate the scatterplot to match the regression line with the x-Axis. See Figure 4.4 (e+f) for an example. Let X and Y be the attributes under inspection in the scatterplot. Then we can compute the mean values $\bar{x}$, $\bar{y}$ to translate the points in the plot such that $(\bar{x}, \bar{y})$ coincides with the origin of the scatterplot. The slope β of the regression line $y = \alpha + \beta x$, ($\alpha = 0$ after translation) is then

$$\beta = \frac{\sum_i ((x_i - \bar{x})(y_i - \bar{y}))}{\sum_i (x_i - \bar{x})^2}$$

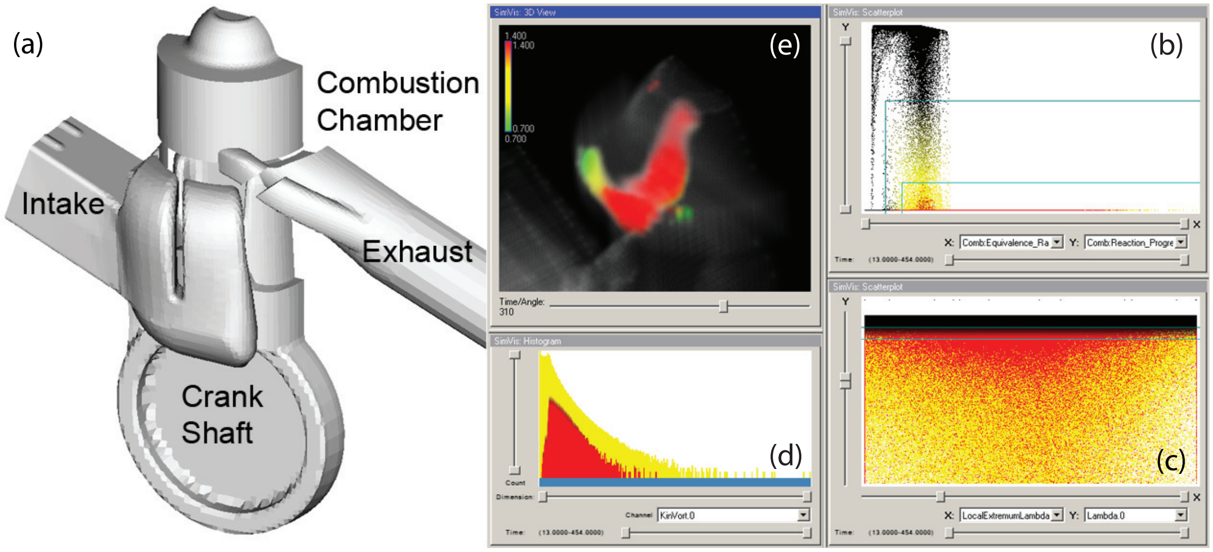


Figure 4.5: Feature localization analysis in the application case of a 2-stroke engine simulation: (a) an overview showing the engine geometry. (b) In the first step we select the parts of the data with the desired oxygen/fuel ratio. (c) To find out which portions of the vortex result in optimal mixing we plot λ_2 against local extremes of λ_2, the optimal mixing seems to result from medium vortex strength. (d) We check this result by comparing with a histogram of the kinetic vorticity number and see that the cells under strong rotation do not show optimal mixing ratios. (e) The resulting mixing vortex region colored by ER attribute (”ER” measures combustibility).

This gives the angle between the x-Axis and the regression line as arctan(β). After rotating the points in the scatterplot we can use rectangular brushing to select correlating parts of the attributes X and Y. Since linear regression using a regression line based on least-squares-fitting is very sensitive to outliers in the data, we include the option to exclude a certain percentage of the data from the calculation of the covariance (e.g. the 100th percentile) in the user interface.

In Figure 4.4 we see an example of this approach. The 3D images (a-c) are different images from the same viewpoint, while the fourth 3D view (d) has larger zoom for closeup visualization. The scatterplot (g) and the 3D view (d) are linked such that the selected cells get full opacity. We study a weak vortex in a region behind the inlet where the flow is moving slowly close to a stronger main current from inlet to outlet. For this reason the general direction of the flow from inlet to outlet is difficult to distinguish from the location of the vortex by local feature detectors. In (b) and (c) we can see that both detectors tend to include some portion of the curved flow into the detected vortex region. Still, the strengths of response differ in a systematic way from each other: in the faster portion of the flow we get detector responses that correlate differently than the responses for fluid cells inside the weak vortex. Therefore we can select a good approximation to the vortex region with a single brush. Of course we could have improved the selected region by using

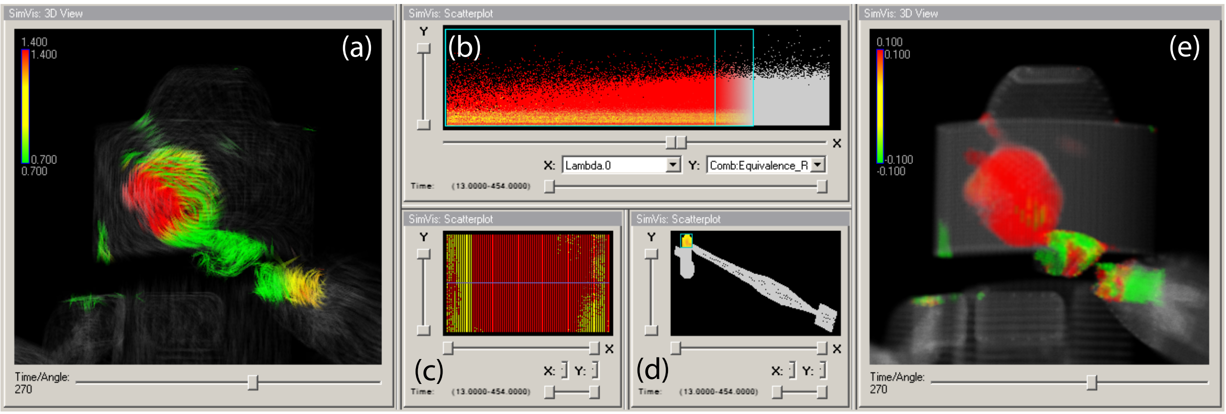


Figure 4.6: Analysis of the tumbling motion inside the combustion chamber: (a) a 3D view of the tumbling motion that causes mixing of fuel and oxygen. (b) The selection of high turbulence measure. (c) A scatterplot showing the distribution of time vs. helicity shows that the current timestep is of importance for the overall process. (d) A 2D projection of the engine were the combustion chamber is selected for spatial analysis. (e) A second 3D view with color mapped to helicity shows that the selected vortex consists of at least two counterrotating parts.

additional brushes and including other features, but in the image we can clearly see that in the rotated view a single selection already performs very well.

4.5 Multiple Detector Views

Numerical data visualization aims at supporting the tasks of exploration and analysis to find out about certain characteristics of the data set. Important tasks are: verification of simulation results by the experts for sanity, improvement and optimization of shape and geometry, definition of settings and parameters of controlling hardware (e.g., the chip controlling injection speed for different engine loads) and prediction of fluid-material interaction (e.g., transport of lubrication fluid). All these properties of the data are connected to a certain extent to the control of vortices and their development over time.

We have identified four important interaction schemes that are often used in interactive analysis – they all involve the use of multiple linked views:

- **vortex and feature localization** answers questions on the location of vortices and related features inside the data. The engineer uses interactive brushing in multiple views to select ranges of different features that are of interest. Different 3D views will show where the concerned regions are located and the user can relate regions and their properties with each other.

  For example in a simulation of two mixing fluids the user can search for regions of turbulent flow by brushing high vorticity values and compare them with regions of high concentrations of one fluid to find stable vortices that are interfering with the mixing process. (See Figure 4.5 for an example and its discussion later in the text.)

- **spatial analysis** answers questions concerning the properties of a certain subvolume of the data. The engineer selects the relevant region using projections and 3D views and uses multiple scatterplots to visualize the situation and compare it to the overall behavior of the fluid in other views. This kind of analysis is necessary to understand the situation at important parts of the geometry to find out if the flow behaves as expected. See Figure 4.6 for an example.

- **vortex-feature analysis** answers questions on the relationships between vortices and other features. The engineer uses multiple connected views to gain understanding on how a selected feature affects other attributes of the data. For example the user can use several feature detectors to detect all vortices and to exclude false positives. Afterwards the engineer can analyze how the temperature in these regions compares to the overall situation. See Figure 4.7 for an example.

- **time-dependent vortex analysis** answers questions on the progression of properties of the data in time. The engineer uses multiple views that show the situation at different time steps or the development of a feature over time. For example the user can select a vortex and analyze its progression in the course of time.

The key to these analysis guides is the linking between the different views. When we need to interpret data – that is to go from numbers to meaning – we need the built-in connection between the different views: linked viewing parameters in the 3D views during feature localization and linked feature brushing for inter-feature analysis and spatial analysis. This procedural approach to interactive drill-down analysis is based on a combination of computational, automatic feature detectors that are computed in a pre-processing step with interactive brushing and attribute selection.

We see that using multiple linked views and complementary feature detectors we can gain deepened insight of the vortex-type features inside the flow. The analysis involving several feature detectors gives two opportunities for further inspection. The first way is to interrelate different detectors to

- increase confidence in the detected region or core [146].

- find false positives and false negatives by comparing where the detectors agree and disagree.

- limit the response to a certain type of vortex feature. For example this can be beneficial to be able to discern vortices by the direction or strength of rotation.

These properties are important when dealing with a large number of automatically detected vortices, where it is necessary to analyze only the largest ones or otherwise classified as 'critical'. The combination of detectors and attributes of the data is beneficial in several ways:

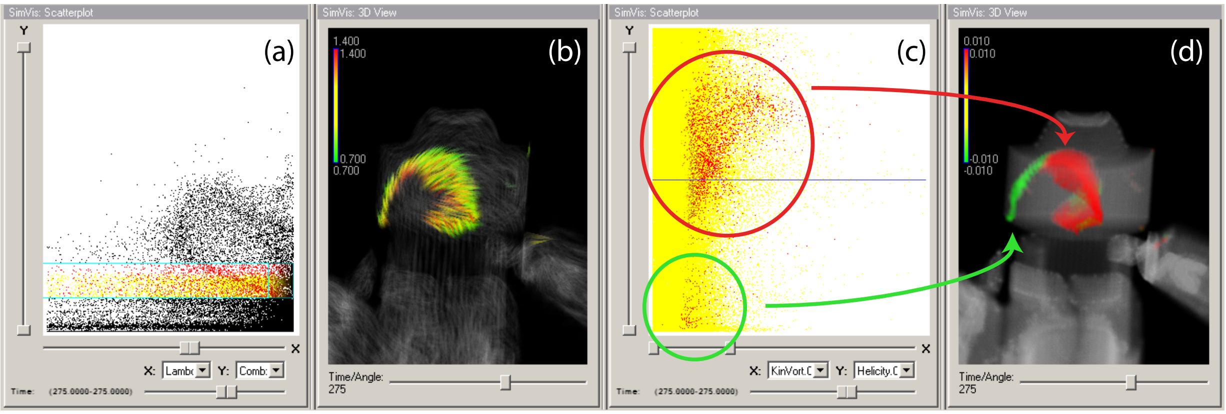


Figure 4.7: Further investigation of optimal mixing. (a) The portion of the vortex from Figure 4.6 where the mixing attribute lies inside the optimal range. (b) In the 3D view we can see that this region is located at the boundary of the rotating motion. (c) Plotting kinematic vorticity vs. helicity reveals two clusters with a tendency to slowly rotating motion. (d) In 3D we can see that the region of optimal mixing differs from the main parts of the tumbling motion (compare Fig. 4.6 (e)).

- We can find vortices that are critical in the context. For example vortices that hinder mixing, that stall the fluid motion, take away kinetic energy or cause material wearout.

- We can analyze vortices that are caused by purpose to check their effectiveness and general shape (e.g., to cause mixing).

- We can understand how the existence of vortices is related to relevant properties inside the fluid.

All these tasks inevitably involve the use of multiple views, both 2D and 3D, and require to relate features and detectors to each other.

4.6 Evaluation of the Multi-view approach

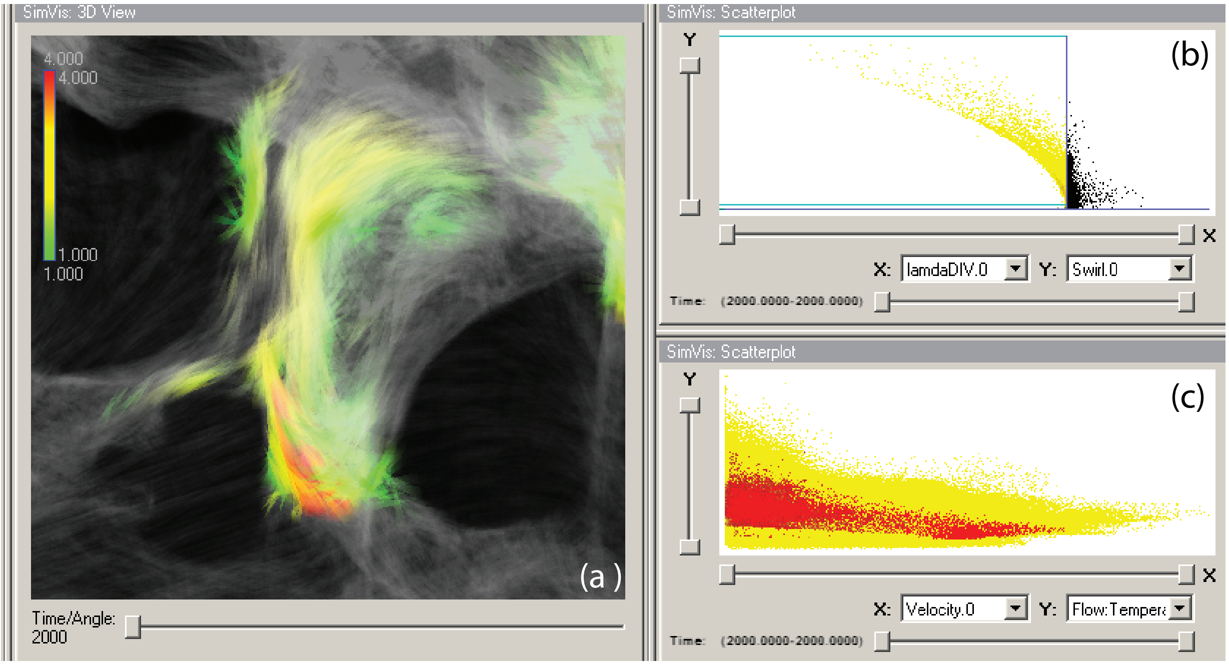
Data sets from computational fluid dynamics (CFD) simulation share specific characteristics: they cover multiple aspects of the simulation related to physical properties of the material (e.g., temperature, density or strain), the fluid component (e.g., direction, kinetic energy or pressure) and additional quantities related to the application domain (e.g., rate of combustion or soot density). In the following we will discuss two examples of interactive visual analysis of simulation data from the field of automotive research.

4.6.1 Two-Stroke Engine

The 2-stroke engine data set contains a complete simulation of the injection and combustion of fuel during one crank revolution. The model is based on a moving volume mesh such that at every time step a new unstructured grid is introduced. In a previous work, Schmidt et al. have analyzed this data set with the SimVis approach [131] and compared the results with a VR/AR method. With the use of CFD simulation initial experiments can be made without costly production of prototype hardware. In this step, unexpected processes can be investigated and the course of physical processes can be shaped according to the engine specifications. One key attribute that is related both to emission and engine performance is equivalence ratio (ER), which is the relation between fuel and air within a volume cell. It is crucial that ER lies in the optimal interval between 0.7 and 1.4 for most fluid cells at the moment of ignition. In Figure 4.5 we inspect the mixing process inside the combustion chamber – using the approach following a spatial feature analysis process: (b) in the first step we see the region of optimal mixing in a timestep at the beginning of the compression process when the reaction has not started yet. (c) This region was selected using a scatterplot showing ER and a reaction progression attribute. (d) A scatterplot that relates vortex detector output against its local extrema shows that optimal mixing is related to regions that are not at the peaks of vortex detector response. This could be related to the ambivalent nature of vortices in relation to mixing: a strong vortex can isolate portions of the flow inside and hinder mixing while chaotic motion and turbulence tend to cause full mixing of different fluid materials. (e) The histogram of kinematic vorticity shows clearly that the parts of the fluid that have reached optimal mixing (colored red in the histogram) are rotating slower than the rest.

In order to further analyze the connection between turbulent motion and mixing inside the combustion chamber we continue with a spatial analysis. In Figure 4.6 a measure of high turbulence is selected and related to the region that exhibits optimal mixing. In (a) we see a visualization of the rotating motion that is part of the tumbling motion which is the main force of the mixing process of fluid and oxygen. This first orientation step of the spatial analysis shows the vortex in nearly full development. This vortex constitutes an important part of turbulent motion inside the combustion chamber and is essential in the mixing process during compression resulting from the upward motion of the piston. Even though we have mapped color to ER, spatial relations do not become immediately clear. (b) In the scatterplot we can see that ER in the combustion chamber is contained in a very narrow band (the yellow portion) probably due to the effect of the injection setup. We have selected high vortex-measure (λ_2) values. (c) A scatterplot of time vs. helicity shows a tendency to clockwise rotation in later timesteps. (d) We have used a 2D projection of the geometry to select the combustion chamber for local analysis. (e) Mapping helicity to color of the volume cells we can distinguish the main vortex (red) from a smaller turbulent region (green).

To understand the properties of this vortex we continue with an inter-feature analysis that relates features of the fluid with ER. This is shown in Figure 4.7. (a) As a first step we select optimal ER in a scatterplot showing the situation in the current timestep. (b)



X: lamdaDIV.0 Y: Swirl.0
Time: (2000.0000-2000.0000)

X: Velocity.0 Y: Flow:Tempera
Time: (2000.0000-2000.0000)

Figure 4.8: One key part of the flow inside the cooling jacket are the vortices caused by the gaskets between the cylinder block and cylinder head parts. (a) A closeup of one of the vortices behind a gasket (see also Fig. 4.9 (a) no. 1). (b) The central parts of the vortices can be selected by combining λ_2 and high swirl. (c) When plotting velocity against temperature the gasket vortices lie inside a single cluster.

The 3D view shows that the portion of the vortex where the mixing attribute lies inside the optimal range is located at the boundary of the vortex. (c) Plotting kinematic vorticity vs. helicity reveals two clusters, both with a tendency to slower rotating motion. (d) In a 3D view we can see that the region of optimal mixing differs from the main parts of the tumbling motion (compare Fig. 4.6 (e)). We can assume that the second cluster (green) is a weaker child vortex caused by the main vortex and that the mixing process is related to the appearance of smaller regions of turbulence caused by the main tumbling motion.

4.6.2 Cooling Jacket

In the second application example we inspect fluid flow through the cooling jacket from the previous Chapter (see 3.4). In this application evaluation we continue the analysis from the previous chapter and analyze the regions of turbulence behind the gaskets further using multiple detectors.

The coolant flow inside the jacket is governed by two major directions – a longitudinal motion along the cylinders of the engine and a transversal motion from cylinder block to head. To keep the engine operative it is necessary to obtain an even distribution of flow to each engine cylinder and to avoid regions of stagnant flow to ensure good overall heat

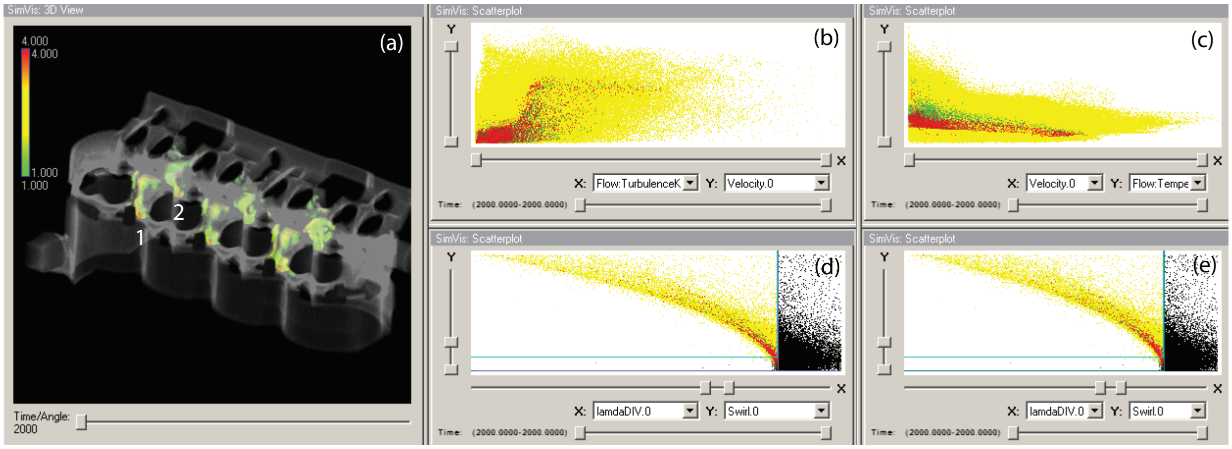


Figure 4.9: We analyze the difference between two of the vortices caused by the gasket geometry: (a) an overview of the gasket vortices in the cooling jacket geometry. (b) This scatterplot shows that second vortex (red) tends to lower turbulence and velocity but clear distinction is not possible. (c) We found the differing attribute: plotting temperature separates the two regions. (d+e) Both vortices are very similar in the outputs of the vortex detectors.

transport.

Around the combustion chambers heat is transported away from the surface by a mixing process that replaces the heated fluid. To this end the shape of the gaskets causes turbulent regions behind the gaskets. The turbulent regions differ in size and heat transport efficiency. In Figure 4.8 (b) we see a closeup of a single vortex with color mapped to velocity. (c) We use a combination of λ_2 and the swirling strength parameter to select the full region of turbulence restricted to the parts rotating stronger. In (d) we see a scatterplot involving all vortices and so far we cannot distinguish between the two vortices in terms of their attributes.

In Figure 4.9 (a) we see six of these turbulent regions in focus with the rest of the geometry in shades of gray for context visualization. In scatterplots visualizing detector outputs ((d) and (e)) we cannot see obvious differences between the turbulent regions 1 and 2. Therefore we switch to analyzing their attribute values using additional linked views. In a scatterplot (b) of turbulent kinetic energy vs. velocity we see that the second vortex (red) tends toward slower movements and less kinetic energy. A second scatterplot (c) now reveals that the two turbulent regions differ in their temperature attributes. Further analysis shows that they are similar in other aspects. We can conclude that the difference between the two regions of turbulence stems from the larger temperature of the fluid before entering gasket 2. This is due to the fact that the fluid arriving at the second gasket has covered a larger distance from the inlet than the fluid that goes through gasket 1. The result is therefore that the slightly lower cooling effect of the second gasket is not due to the gasket geometry but an acceptable effect of the overall situation.

4.7 Chapter Conclusions

We have presented a case study of vortex feature analysis using combined detectors and multiple views in the context of CFD data for automotive industries research. Using multiple views it is possible to compare and weight the response values of several detectors intuitively by means of interactive visual analysis. Furthermore we have discussed an extension to brushing features in scatterplots that allows to analyze the correlation of features using rectangular brushing. It was demonstrated that by correlating two detectors one can find features that could not be studied before. The case study reveals that the combination of multiple 2D and 3D views can help understanding complex 3D flow features when linking and brushing are combined appropriately. We have identified systematic approaches to study vortex phenomena in multidimensional data sets, based on multiple views, linking and combined feature detectors.

In the context of real world data the benefits of multiple linked views become clear: taking advantage of the combined potential of infovis views, 3D graphics visualization and automated feature detectors allows to understand the situation inside the data. Reading between the lines and gaining deepened understanding is possible by the streamlined and integrated approach of coordinated views and multiple detectors.

Chapter 5

Delocalized Detectors

> "Most deadly errors arise from obsolete assumptions." (Frank Herbert 1920 – 1986)

In this chapter we discuss generalizations of instantaneous, local vortex criteria. We incorporate information on spatial context and temporal developments into the detection process. The presented method is generic in so far that it can extend any given Eulerian criterion to Lagrangian unsteady vortex detection. Furthermore, we present a visual aid to steer and understand the feature extraction process. We show that the delocalized detectors are able to distinguish between connected vortices and help understanding regions of multiple interacting structures. The filtering properties of the delocalized detectors extract smoother structures and reduce noise in the vortex detection result.

5.1 Motivation

Recent research in the field of Lagrangian coherent structures [41, 46, 126, 127] suggests that we need to refine our approach to understanding fluid behavior. Even though the local information has shown to be highly valuable when trying to understand the nature of turbulent fluid movements, we need to look further and find ways to include information on temporal development and particle movement into the analysis. In this chapter we show that vortex feature extraction can retain the knowledge that we have on local properties of the flow and still include the Lagrangian perspective into the analysis.

The Lagrangian approach is based on taking the trajectories of particles into account for analysis. We can think of the detectors presented in this chapter as criteria where local detector responses are accumulated along trajectories to achieve both spatial and temporal coherency. The Lagrangian approach introduces new questions into the analysis. Since the result of the Lagrangian vortex feature detector is dependent on the length of the particle trajectories[1] analyzed, we get an additional parameter with significant impact

---

[1] since trajectories are streamlines for steady data and pathlines for unsteady data we will speak of trajectories when the difference is not relevant

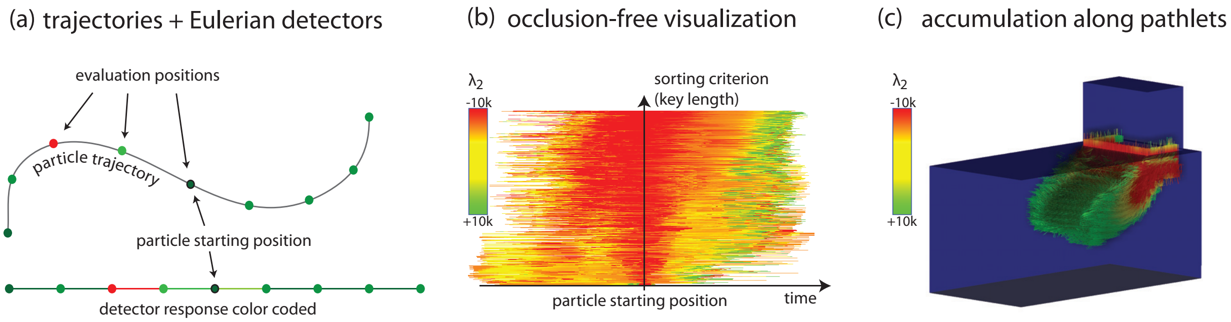

(a) trajectories + Eulerian detectors
evaluation positions
particle trajectory
particle starting position
detector response color coded
(b) occlusion-free visualization
λ_2
-10k
+10k
sorting criterion (key length)
particle starting position
time
(c) accumulation along pathlets
λ_2
-10k
+10k

Figure 5.1: We illustrate the suggested approach. (a) By analyzing Eulerian quantities along trajectories we can improve the extraction of vortices which we consider as coherent structures and limit the effect of numerical issues. (b) Using a simple two-dimensional representation of the detector responses along trajectories we can understand and steer the locality of the accumulation. (c) The resulting method promotes coherency in space and time by accumulating information along a trajectory of a fluid particle for each point.

on the results of the analysis. We need a way to control the length of the trajectory that contributes to the vortex detector response. Recent publications have suggested this as an important open research question [127, 41, 37]. We present an approach which allows to control this parameter non-uniformly using an interactive analysis view.

A problem mentioned by several publications dealing with Lagrangian coherent structures and particle trajectories in general [36, 125, 128] is the fact that particle trajectories can quickly leave the simulation domain (e.g., through an outlet). In this case we do not have enough information available to give a good accumulated detector response. The approach of delocalized Eulerian detectors gives three answers to the problem of short trajectories: firstly, local criteria have been demonstrated to give reliable results on their own, thus we are less dependent on having long trajectories available to generate good results. Secondly, we allow to include the upstream information by using backwards integration into the analysis to compensate for short particle trajectories in forward time. And thirdly, by taking the proportion of the unknown region into account, the lack of information due to extremely short trajectories is included into the detection result.
The contributions of this chapter are as follows:

- We extend Eulerian vortex criteria to extract coherent structures which improve on the features detected using λ_2, Q or the swirl criterion.
- A method to interactively control integration of the detectors along particle paths.
- An extension of trajectories to include upstream information to deal with the problem of short particle trajectories.
- Comparison and evaluation of the results according to numerical issues, smoothness and separation of vortices.

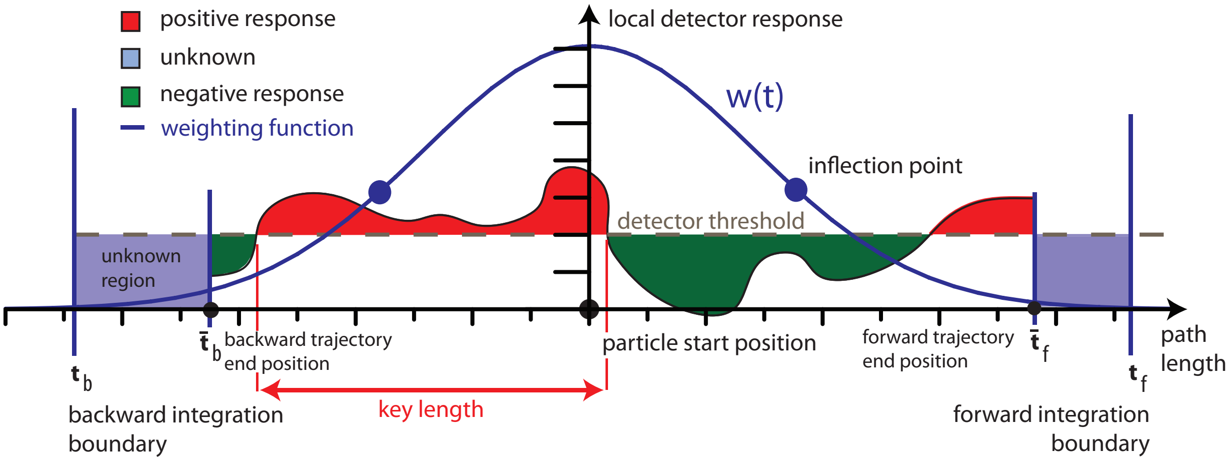


Figure 5.2: Accumulation of local detector values along a particle trajectory. The local values are weighted according to their distance from the seeding point of the trajectory.

In the next section we discuss related work. The third section presents the basic concepts of delocalized feature detectors and a 2D view of particle trajectories to analyze the local detector responses. In the fourth section we present evaluation results.

5.2 Delocalized Vortex Detectors

In the following subsection we discuss the non-local extensions of the Eulerian detectors. In the second subsection we present the line view which helps to steer the extraction. See Figure 5.1 for an illustration of the delocalized vortex region detectors approach.

5.2.1 New Criterion

For accumulation along a trajectory we need an Eulerian vortex detector $E(x, \mathbf{J}(x,t), \mathbf{u}(x,t)) \rightarrow [0,1]$ where $x \in D \subset \mathbb{R}^3$ is a point inside the simulation domain D, $\mathbf{u}(x,t)$ is the velocity at point x at time t and $\mathbf{J}(x,t)$ is the Jacobian. See Figure 5.1 (a) for an illustration. The criteria we have found to benefit most from delocalization are λ_2, Q, the swirling strength criterion and vorticity magnitude.
A pathline can be expressed as

$$p(t+\Delta t) = p(t) + \int_t^{t+\Delta t} \mathbf{u}(p(s),s)ds$$

where $p(t)$ is the position of the particle at time t, $p(t+\Delta t)$ is the new position after time Δt and $\mathbf{u}(p(t),t)$ is the velocity of the particle at position $p(t)$ at time t. The Runge-Kutta method (RK4) can be used for numeric integration of these pathlines [116]. Given a vector field $\mathbf{u}$ we call p_{t_0,x_0} the pathline starting at point x_0 and time t_0.

Now we can define the Eulerian detector response for a pathline p_{t_0,x_0} as

$$E(p_{t_0,x_0}, t) = E(p_{t_0,x_0}(t), \mathbf{J}(p_{t_0,x_0}(t), t), \mathbf{u}(p_{t_0,x_0}(t), t))$$

if $p_{t_0,x_0}(t) \in D$.

For a pathline we define the two maximal integration length parameters $\bar{t}_f$ and $\bar{t}_b$ as the maximal time this pathline remains inside the simulation domain D during forward (resp. backward) integration.
In contrast to unsteady flow LIC, where a color pixel value is advected through the flow field, in the the context of computing a vortex detector value it makes sense to include the backward direction: the vorticity of a position is not only dependent on its future contribution to a vortex, but also on its past developments (we can think of a particle inside the border of a strong vortex region to be justly assigned a high vorticity value).

The delocalized version of the Eulerian detector at position x_0 and time t_0 finally is

$$\widetilde{E}(p_{t_0,x_0}, t_b, t_f) = \frac{\int_{\max(\bar{t}_b,t_b)}^{\min(t_f,\bar{t}_f)} w(s) \cdot E(p_{t_0,x_0}, s)ds}{\int_{t_b}^{t_f} w(s)ds} \tag{5.1}$$

with $t_b < t_0 < t_f$ and $w(x)$ a weighting function. Good parameters for forward and backward integration time, t_f and t_b allow the delocalized detector improve on the local information. In case the trajectory leaves the domain before the selected integration times are met ($t_f > \bar{t}_f$ or $\bar{t}_b > t_b$) we can accumulate the requested information only partially. Weighting the result with the integral of $w(x)$ over the complete selection $[t_b, t_f]$ decreases the delocalized detector result and *incorporates the uncertainty resulting from short trajectories*. The formalism does not change for steady and unsteady flows, since for steady flows the definition of a pathline coincides with the definition for a streamline.

The weighting function should give sufficient control over the accumulation and produce predictable results for the user. The first option is linear weighting where the weight for a position on the trajectory is given by the difference in physical time from particle release time t_0. That is $w(t) := 1 - (t - t_0)/(t_f - t_0)$ for $t \geq t_0$ and $w(t) := (t_0 - t)/(t_0 - t_b)$ for $t < t_0$ with ($t_b < t_0 < t_f$). The second option is a an accumulation using a Gaussian filter $w(x) = 1/(\sigma\sqrt{2\pi}) \cdot e^{-0.5(t-t_0)^2/\sigma}$. See Figure 5.2 for an illustration. The line view presented in the next section allows the user to determine and specify the relevant parameters, i.e. σ, t_f and t_b. Using Gaussian weighting, the influence of a sampling point quickly becomes very small after the inflection point is reached, thus σ can be used to control the locality of the criterion. For brevity we will write $\tilde{E}_t^{\sigma}$ for a delocalized detector using Gaussian accumulation when $t_f = t_b = t$. Units are seconds for t, and σ has unit of 'average cell size times meters' to simplify the notation for the different data sets.

Filtering a Galilean invariant feature detector along a path line results in a Galilean invariant detector. This is because the operation of pathline integration commutes with Galilean transformations. We can understand this intuitively by considering what happens inside the fluid if the observers speed relative to the fluid changes: the position of a particle relative to a vortex remains the same as long as the detector itself is Galilean invariant. In

	Cells	Ts	Type	Grid	ROI	Lines	Acc.	Int.
T-Junc.	30 K	100	incomp.	struc.	all	137 MB	0.1 sec	1 min
Cool. J.	1538 K	1 (steady)	incomp.	unstruc.	95 K	650 MB	2 sec	5 min
2-Stroke	1156 K	91	comp.	unstruc.	81 K	570 MB	2 sec	4 min
Rankine	262 K	1	synth.	struc.	all	1.6 GB	3 sec	3 min

Table 5.1: Comparison of the data sets evaluated in the application study. We have evaluated a simulation of a pulsating T-Junction, a Cooling Jacket, a 2-stroke engine and the synthetic rankine vortex model. The region of interest (ROI) showing complex vortical behavior was always much smaller than the whole data set. (Abbreviations: Ts – time steps, ROI – cells in region of interest, Acc. – accumulation of delocalized detector, Int. – integration of trajectories)

case the detector is not galilean invariant the information accumulated along the particle path may not be correct, but since the delocalization operation commutes, we can argue that integration does not aggravate this problem.

So far we have not discussed how the integration parameters t_b and t_f can be chosen appropriately. This will be the topic of the following subsection.

5.2.2 Line View

The purpose of this view is to visualize the computed trajectories in a 2D view as straight lines. This gives more space to convey visual information and enables easier selection and brushing operations. The view is related to work of Matkovic et al.[94] that presents colored lines as an approach to visualize functions as straight lines using color coding.

The delocalized criteria are robust and only in complex flow regions a single threshold does not perform well. In this case it is necessary to use multiple thresholds, which are difficult to define using the occluded 3D trajectory rendering. By evaluating the distribution of local detector values in combination with selective 3D visualization of the relevant streamlines it becomes possible to select a few suitable integration length parameters to separate interacting vortices.

The engineers are interested in the relationships between the fluid cells of the mesh, therefore we seed one trajectory per grid cell. Each trajectory is visualized simply by placing its segments successively on a straight horizontal line. The resulting horizontal lines are spaced vertically so that they fill the available viewing space. Our main interest for the line view is to observe how the vortex classifier response is distributed along the trajectories. The view works in coordination with the other views in the visualization framework to allow filtering relevant trajectories. See Figure 5.3 for an illustration. While selecting integration length parameters in the line view, the currently relevant trajectories are rendered as lines in the 3D view, conveying the spatial information for these trajectories. Additionally, a degree of interest (DOI) can be specified by brushing in other views, thus assigning a DOI value to the sample points in the data set. Lines with zero DOI at their

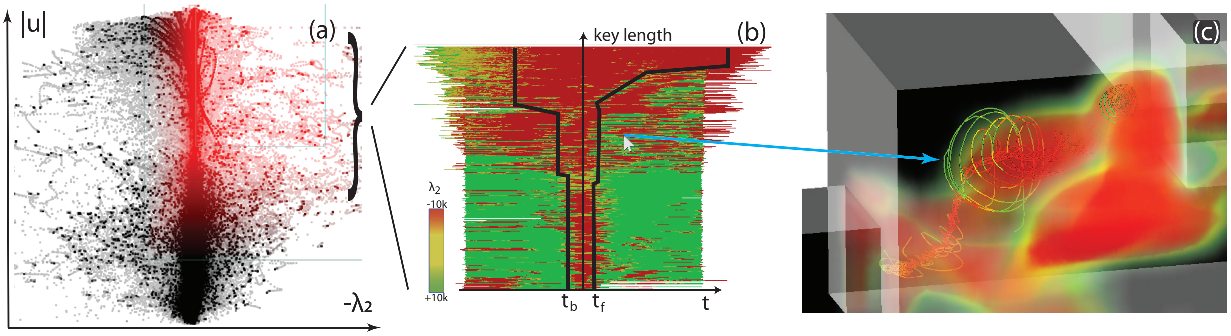


Figure 5.3: Linking the Line View. (a) Attribute selection on the multivariate simulation data set allows to filter the data points of interest. Only the lines seeded at the selected points are rendered in the line view. (b) The remaining lines are displayed in 2D and the user can specify integration parameters by drawing line segments onto the view. (c) The view is linked to the 3D rendering. While selecting the forward and backwards integration length parameters in the 2D view the trajectories below the tip of the cursor are rendered in the 3D view.

starting positions are filtered out.

Good sorting of the lines is crucial so that trajectories belonging to the same structure are ordered closely together. For that purpose the view offers a range of sorting and filtering criteria:

- Key length: is the maximal time interval including t_0 inside which the particle remains inside a region of positive local detector response without interruption (see Figure 5.2).
- Line length: lines are sorted or filtered according to their length
- Delocalized response: after selection the lines can be reordered according to their delocalized detector response

Line fusion: line fusion is necessary when more lines are currently in focus than there are pixels available on the screen. We employ post-classification, by first combining the detector responses for line segments, and then assigning color to the resulting line by means of a transfer function. To combine a group of lines into one line, we keep advancing a vertical scan-line, until all segments are drawn onto a storage texture.

Integration length specification: the selection ranges for the lines can be defined interactively by drawing two curves on the view. This way, for each group of lines (after fusion) the user can specify the parameters t_f and t_b. When Gaussian weighting is selected, the interaction allows to select the location of the inflection point. It is sufficient to select the ranges for very large groups of lines and only when this approach fails it is necessary to zoom in and perform a more elaborate selection. With good sorting parameters the delocalization is robust, and all figures in this chapter were made without zooming.

Using linking and brushing the line view allows to select appropriate integration length parameters for different regimes of the flow, which is necessary to separate interacting structures. Furthermore, the selective rendering of 3D trajectories using linking between the line view and the 3D view can serve as a useful analysis tool by itself.

5.3 Evaluation

In this section we discuss vortex feature detection results. Table 5.1 gives an overview of the evaluated cases. To be able to evaluate at which point higher thresholds start to remove important parts of the feature we include vortex core lines computed with the approach of Sujudi and Haimes [140] for the steady cooling jacket data set and its extension for time dependent flows discussed in the following chapter. These vortex core line detectors can produce spurious or shifted solutions as well, but for the strongest and largest features they are a good measure for comparison with the extracted regions.

5.3.1 Value distribution in real-world data sets

When working with CFD data sets we can roughly classifiy them into two categories: the first are direct numerical simulations (DNS) where a flow problem is solved on a super computer using the Navier-Stokes equations alone. Using super computers and sufficiently simple problems it is possible to solve the given problem exactly with very high grid resolutions. These data sets are commonly well behaved and contain few, well distinguishable structures [101].

CFD data sets from engineering on the other hand usually deal with problems that cannot be solved using DNS and need turbulence models to get results in reasonable time. These data sets tend to contain [2]:

- Large differences in cell sizes (generally three to four orders of magnitude).
- Non-planar faces on the boundary where the shape of the object is modeled.
- Large differences in face sizes within a single cell.

This can lead to extremely large gradients close to the boundary due to the extremely small cell sizes. Solution errors and non-planar faces additionally cause errors in gradient estimation. Therefore the cells close to the boundary and especially those near edges of the boundary can have strong impact on the distribution of vortex detector values. In Figure 5.4 we can see that the non-uniform distribution of detector values has to be considered during interactive analysis.

The non-uniform distribution of detector values also has to be taken into account when selecting levels for iso-surface extraction from vortex detector values. A common choice in DNS data is to use iso-surfaces at a level of 20% of maximal detector response for vortex visualization (see e.g., [62, 176]). Due to the generally higher and more variable detector values in engineering problems, appropriate levels for iso-surface extraction have to be

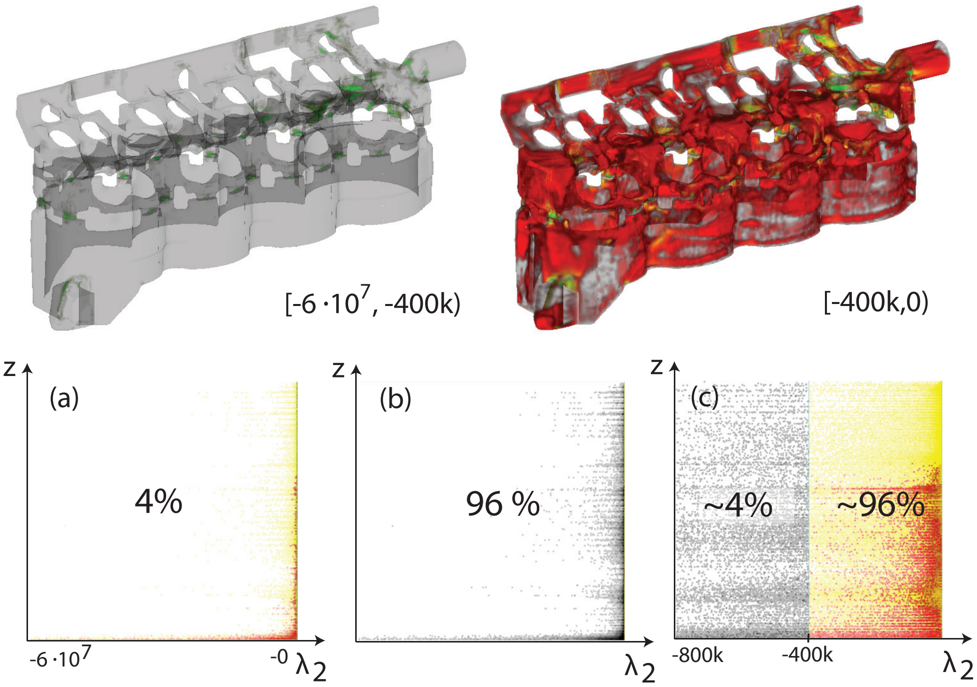


Figure 5.4: Extremely large gradients can lead to non-intuitive value distributions of the vortex detection criteria. (a) We can see that a selection of λ_2 values in the range $[-6 \cdot 10^7, -400k)$ in a scatterplot. This selects only 4% of the negative λ_2 values in the volume. (b) The remaining 96% of λ_2 values are invisible. (c) In a zoomed view of the scatterplot we can see the distribution in the selected range $[-400k, 0)$. The linked 3D views give an impression on the distribution of λ_2 values inside the data set. (all numbers rounded to next integer).

selected for each data set individually. For the cooling jacket more than 90% of positive Q-detector values lie inside the range $(0, 2 \cdot 10^6]$ and more than 90% of the λ_2-detector values lie inside the $[-400k, 0)$ range. We have found iso-values of 1000 for Q and 5000 for λ_2 appropriate to display the geometry of the extracted vortex structures.

5.3.2 Cooling Jacket

The first data set contains a steady simulation grid of fluid moving through a cooling jacket (see Figures 5.5 and 5.6). In this section we extend the analysis of turbulence inside the cooling jacket performed in Chapter 3. In Chapter 4.6.2 we have discussed the geometry of the cooling jacket in more detail. The important part for the following discussion is the cooling jacket gasket situated between the cylinder head and the cylinder block (see Figure 3.3). The gaskets are small holes that serve as conduits between the block and head. We focus on a region of interest (see Figure 5.5 (top)) behind one of these conduits used to control the motion of fluid flow through the cooling jacket. Close to the inlet the flow is

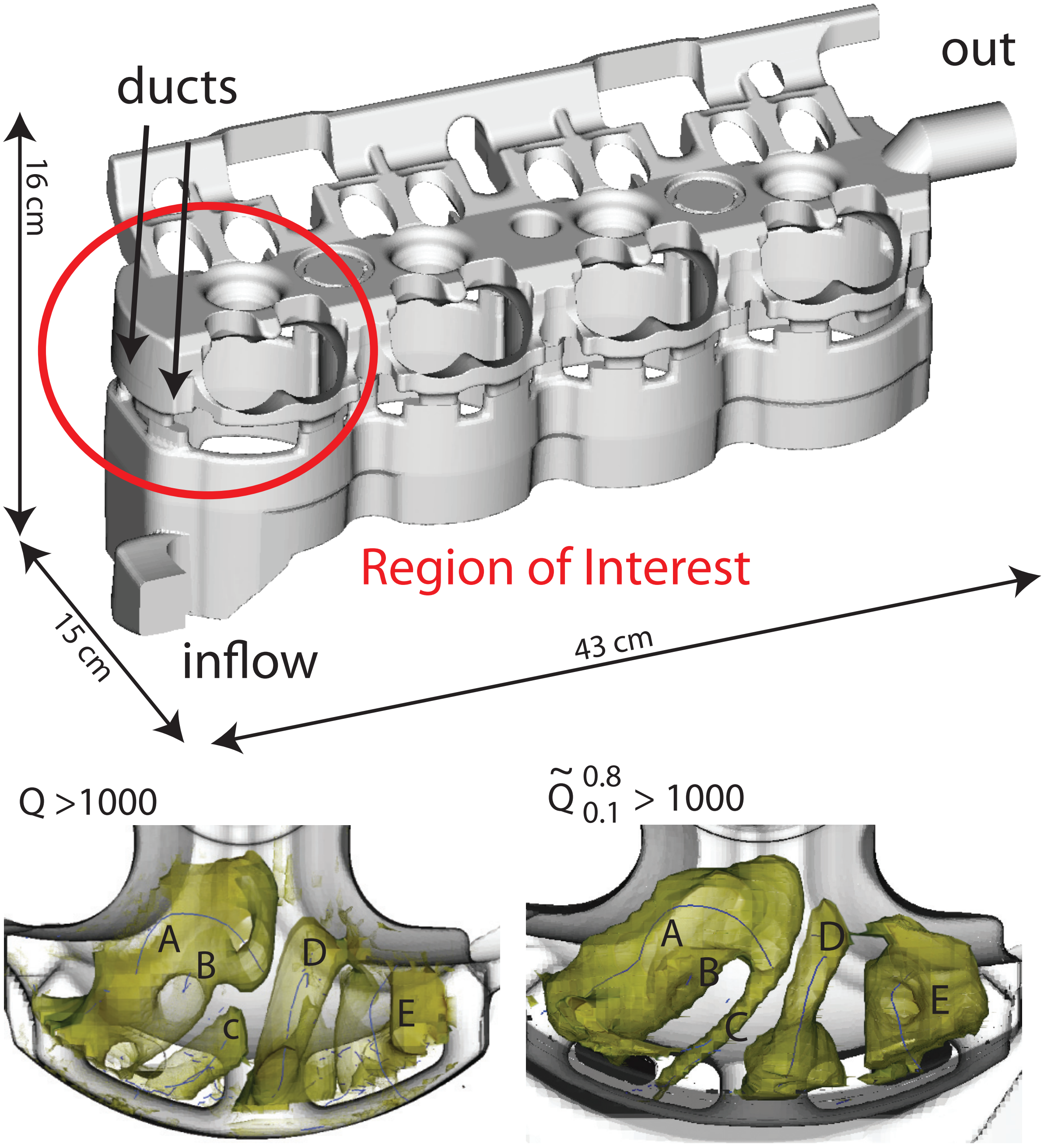


Figure 5.5: The cooling jacket data set. (top) Overview of the geometry and region of interest. (bottom) We compare the detector results for Q and delocalized Q using $\sigma = 0.8$ and 0.1 sec. integration time: large parts of the feature A are removed by the instantaneous Q criterion, while feature C becomes more localized and connected to the larger region, which is better in the light of the results obtained using λ_2.

fast and the gasket causes strong turbulence.

In Figure 5.5 (bottom) we compare the Q criterion and the delocalized Q criterion. We set the threshold to 1000 in order to remove the large amount of tiny structures, such that three larger structures become distinguishable. We can see that inside the largest structure (A) half of the length of the core line is removed and large holes appear. While the two other structures (C+D) contain the same core line features, the delocalized regions appear much smoother and are disconnected.

In Figure 5.6 we compare the λ_2 vortex detector at thresholds 1000 and 5000 with the delocalized version of λ_2. The top row shows a side view of the situation, where we can see turbulent regions appear behind the gaskets. At a threshold level of 1000 all the relevant vortex core lines are present, but the resulting iso-surface is difficult to understand and we have a single connected region. At this level the feature A is still present, but if we want to separate the different structures, we have to set a higher threshold where the core line is no longer fully inside the selected domain. If we look at the corresponding delocalized λ_2 regions, we can see that the feature A remains intact. We can also see that the core line at position C hints towards the assumption that this is a rather important feature which is correctly connected to the large region (A). Also the regions (B) and (C) are disconnected. By comparing the results of the delocalized versions of the Q and the λ_2 criterion we can make another finding: the delocalized results are extremely similar for both criteria, even though the iso-surfaces of the instantaneous versions bear little semblance.

5.3.3 Two-stroke Engine

The two-stroke engine data set contains the simulation results for injection and combustion of fuel in the combustion chamber during one crank revolution.

In Figure 5.7 we can see the results of the vortex detection in the Eulerian and the Lagrangian frame. Here we have only one extremely strong main feature and therefore a single core line is detected. For the main feature (A) the results for the Eulerian and the delocalized detector are similar. But for a smaller structure (B) we have a similar result as for the cooling jacket: to keep B intact we have to select a low threshold ($swirl > 0$) and the feature (B) is difficult to recognize. (b) At a threshold where the features become distinguishable, the feature (B) is split in two components.

5.3.4 T-Junction

The t-junction data set is a small unsteady simulation of pulsating flow through a t-junction. In the center of the domain, behind an obstacle, a small vortex appears. Du to the good temporal resolution and relative simplicity of the situation we can discuss how the delocalization process allows to concentrate on different types of features. We can observe the development of four features in the data set. These are (A) two longitudinal vortices behind the inlet, (B) a transversal vortex created by the pulsating inflow boundary condition, (C) a vortex appearing behind the obstacle and (D) a region of turbulence at the outlet. In (a) we can see that it is possible to select a threshold to separate the structures

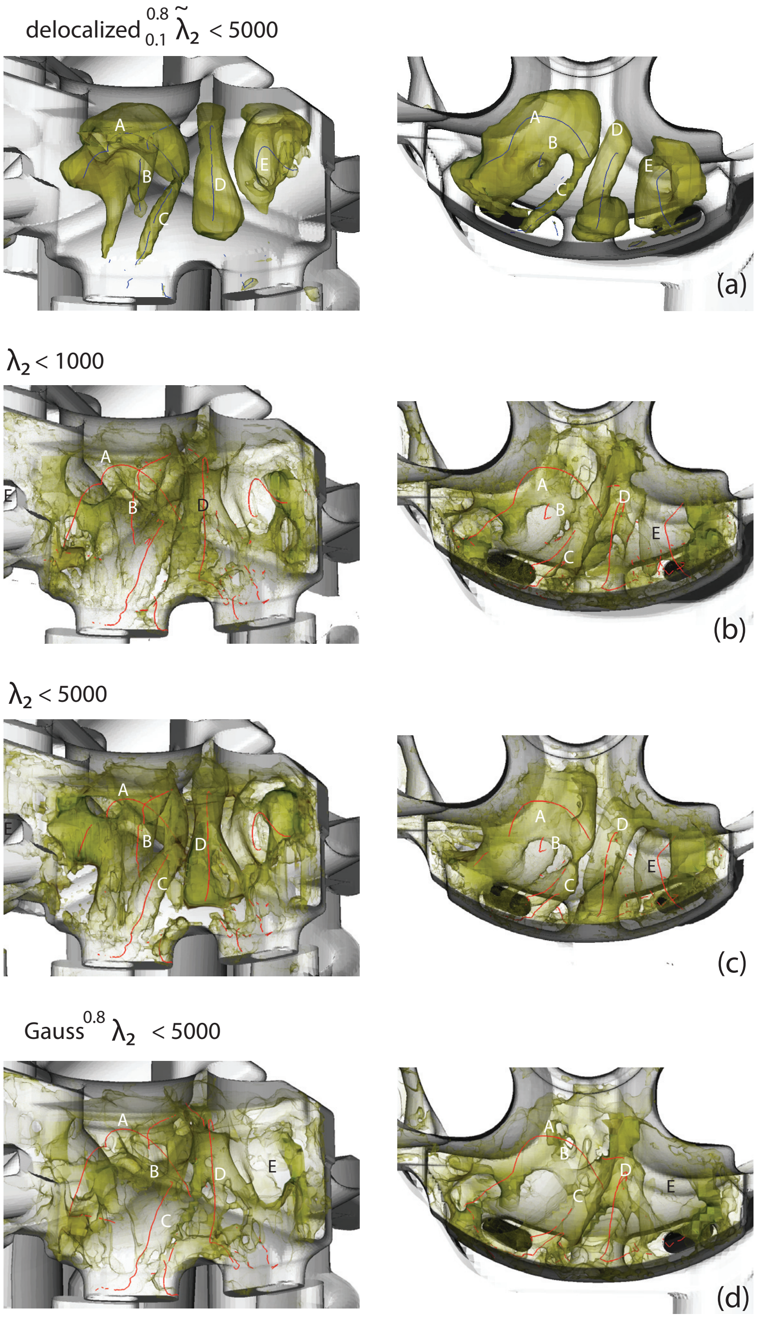

delocalized $^{0.8}_{0.1}\tilde{\lambda}_2 < 5000$
A
B
C
D
E
(a)
$\lambda_2 < 1000$
(b)
$\lambda_2 < 5000$
(c)
Gauss$^{0.8}$ $\lambda_2 < 5000$
(d)

Figure 5.6: The delocalized λ_2 outperforms λ_2 regarding feature separation and noise suppression. (a) The delocalized version of λ_2 extracts three non-connected regions which contain the strongest and longest vortex core lines. Small features and noise are removed. (b) At a threshold of $\lambda_2 < 1000$ we get a single connected region and all the vortex core line features remain intact. We also get a lot of small or weak features which we are not interested in. (c) At a level of $\lambda_2 < 5000$ different (still connected) features appear, still a lot of noise is extracted and vortex A breaks in two parts. (d) Smoothing λ_2 using a Gaussian kernel removes noise, but also feature E.

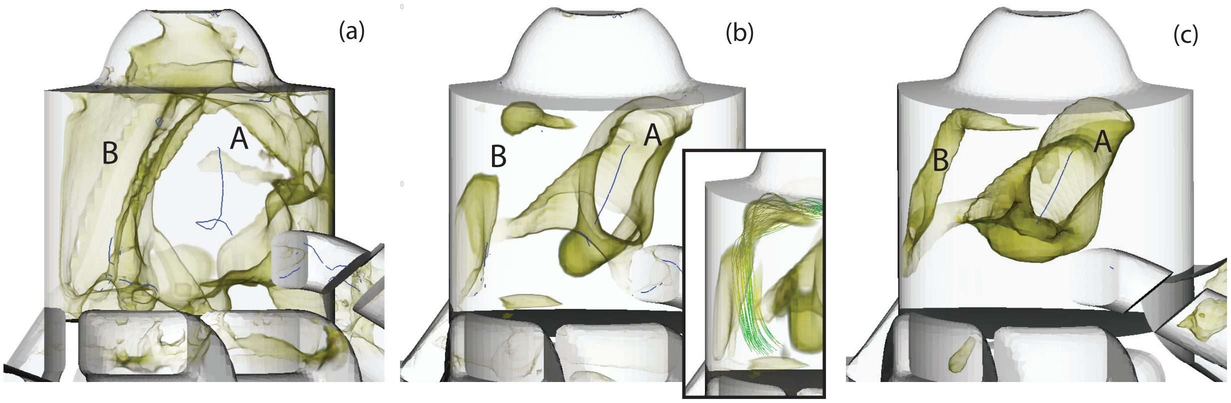


Figure 5.7: Two-stroke engine data set. We compare the result of the swirling strength criterion and the delocalized swirling strength criterion. (a) At a low threshold ($swirl > 0$) the two features are not distinguishable. (b) Searching for a better threshold we find that in order to get a good separation between the two features we have to select a value at which feature B breaks in two components ($swirl > 10000$). A visualization including streamlines seeded at the gap shows that the two components belong together. (c) The delocalized δ detector with $\sigma = 0.8$ and $t_f = t_b = 0.01$ allows to visualize both features intact.

noise	1%	5%	10%
λ_2	0.007 %	5.734 %	12.988 %
$MF(\lambda_2)$	0.007 %	0.019 %	7.813 %
$Gauss(\lambda_2)$	0.872 %	1.1 %	4.185 %
$\widetilde{\lambda_2}^{0.8}$	0 %	0 %	3.002 %

Table 5.2: Numerical evaluation of noise in the Rankine vortex model. The table shows the error rates of the classification results of the local detector (λ_2), after application of a median filter ($MF(\lambda_2)$), after application of a Gaussian filter ($\sigma = 0.8$), and the results for the delocalized λ_2 detector ($\sigma = 0.8$).

(A) and (B). In (b) we use streamlines to depict the shape of the feature (C) a the current moment. We can see that the feature has the same height as the obstacle. From this we can conclude that in (a) the threshold necessary to separate the features (A) and (B), removes too much of feature (C). In (c) we can see that the delocalized version of λ_2 allows to separate the vortex behind the obstacle and still visualize the full transversal vortex (B). By selecting specific integration length parameters for each of the now disconnected regions we have also deselected the turbulent region at the outlet.

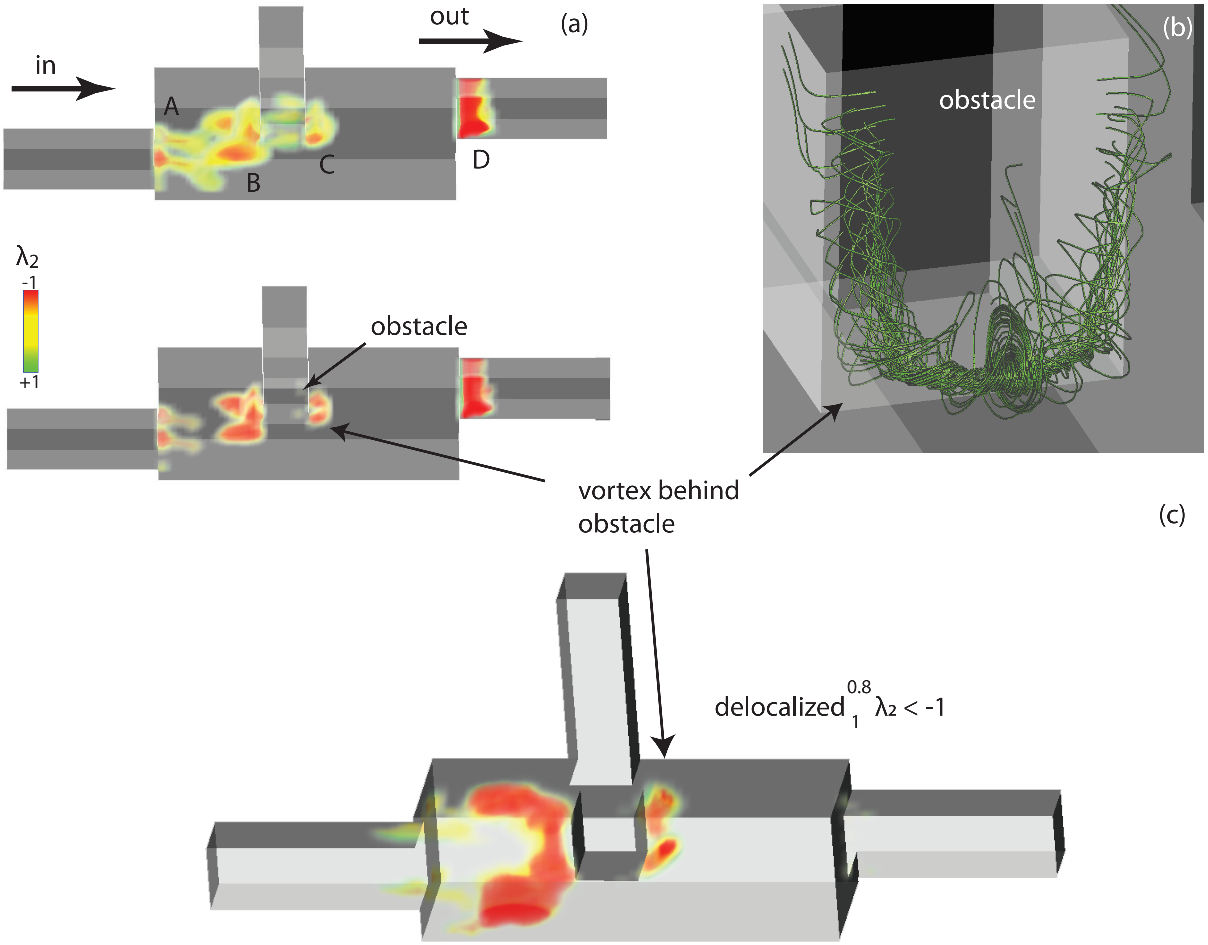


Figure 5.8: Comparison of λ_2 and the delocalized version of λ_2. (a) Behind the inlet two small longitudinal vortices appear (A). When setting a threshold to separate the transversal vortex (B) from the longitudinal vortices, the third vortex (C) behind the obstacle almost vanishes. (b) Streamlines show that the vortex C extends to the full height of the obstacle. (c) Using the delocalized λ_2 detector we can focus on the transversal vortex and still select the full vortex C while deselecting A and D.

5.3.5 Filter Properties

To test numerical stability we use a simple synthetic solution so that we can know where the vortex has to be detected. A simple model for a vortex is given by the combination of a rigid-body rotation within a core, and a decay of angular velocity outside [5]. The Rankine vortex model can be described by

$$u_\theta = \begin{cases} \omega \cdot \frac{r}{R} & , r \leq R \\ \omega \cdot \frac{R}{r} & , r > R \end{cases} \qquad u_r = 0 \quad u_z = u,$$

where R is the radius of the vortex, u controls axial velocity and ω controls the maximal tangential velocity. The model has a long history in meteorological studies of tropical cyclones and can be considered a good approximation of measured data [9]. This is also an example much in favor of the local detectors since they all have 100% correct classification in the absence of noise. Nevertheless the delocalized vortex detectors outperform their Eulerian counterparts consistently. From an image processing viewpoint one can consider the presented detector as a special case of an isotropic filter. To show that the reasoning behind the convolution actually improves the detection results, we compare our results to the error rates after the application of a median and a Gaussian filtering kernel.

In Table 5.2 we can see the results of the numerical study. Noise was added to each cell using a linear combination of random noise vectors for each cell $n_{i,j,k}$ at sample position (i, j, k) in the regular grid and the original flow value $v_{i,j,k}$ such that a noise level of $p\%$ is computed as $v_{i,j,k} + (p/100) \cdot n_{i,j,k}$. In the following evaluation we compare the results of the original Rankine vortex data set and the version with noise added – the two cases are unrelated and the features are extracted each time only using the respective data set.

We use $\sigma = 0.8$ for accumulation of the delocalized detector values. Changing the velocity vector locally will affect the estimated gradient of all the surrounding cells. Isotropic filtering therefore cannot deal with noise as well as the delocalized vortex detectors. The error of the delocalized vortex detectors at a noise level of 10% of the original signal stems from the fact that we have a very sharp vortex boundary in the model such that a small deviation from the correct trajectory can already degrade the performance. A second reason is that the trajectories at the corners of the rectangular domain have very short integration times and quickly leave the simulation domain. It is quite unnatural to have such sharp vortex boundaries and also the large percentage of short streamlines is to the disadvantage of the delocalized detector. Even though the delocalized detector outperforms the other methods. The high error of the Gaussian filter stems from the fact that for low error rates it blurs the errors and can actually increase error.

5.4 Implementation Details

To achieve quick response times appropriate for interactive analysis we need a tradeoff between accuracy, flexibility and computational cost. In the context of our application we know that the engineers are interested in understanding the relationships between the

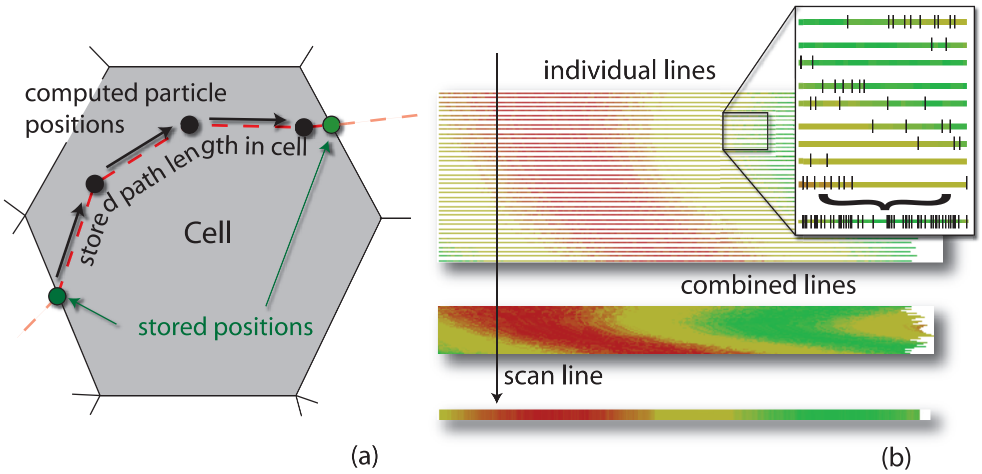


Figure 5.9: (a) A tradeoff between accuracy and interaction time: for interactive analysis we use particle paths computed offline for each volume cell. To reach interactive response rates, we restrict the system to one line segment per cell and particle and store the correct path length in an additional attribute. (b) When the number of pixels available is lower than the number of trajectories we combine groups of lines using a scan line algorithm. This can result in a large number of segments for the combined line.

computational cells of the simulation grid. Therefore we seed one path- or streamline per cell to be able to compute a delocalized detector response for each cell. During interaction the main computational cost lies in the line fusion approach and our prototype can compute the final texture for small regions of interest with up to 100K cells at interactive rates (se Figure 5.9). A more efficient implementation could be several magnitudes faster. Trajectories are computed off line and stored in an additional data set, which takes several hours for the full two-stroke data set, but for a specific region of interest at a selected time step only small subset of these lines has to be computed.

Chapter Conclusions

The main drawback of the presented method is that the detected results are no longer objective in the sense that each engineer will come to exactly the same vortex detection results. The exact location of the vortex boundary is dependent on the specification of integration length parameters. These differences are typically small and as long as we do not have a general definition for what a vortex is, this fuzziness can be considered appropriate. The second disadvantage of the presented approach is the time consumed by integrating trajectories through the fluid. We used a rather inefficient implementation where the timings cannot be considered representative, and GPU-based implementations are reported to compute trajectories at nearly interactive rates [79]. Another drawback is that interaction is often a necessity. Using bad integration length parameters, the results are more blurred and worse than λ_2 regions, even though the delocalized detectors have shown to be very robust in our experience. The approach to use a single trajectory length parameter as seen in the results presented in Figures 5.5, 5.6, 5.7 and 5.8 only works with carefully selected regions of interest and even then the line view is necessary to find good parameters for σ and $t = t_f = t_b$.

An obvious idea for estimating good integration length parameters automatically is to search for minima of detector response along the trajectory. This way we could hope to find the boundary of the vortex region without interaction. This has produced mixed results for the evaluated data sets. A criterion for good integration length parameters based on physical principles independent of user interaction would further improve the delocalized detectors.

In this chapter we have proposed delocalized vortex region detectors. With little interaction to determine reasonable parameters, the delocalized vortex detectors improve the feature extraction process. We have also discussed how the ability to control the range of integration improves the expressiveness of the detectors over their local counterparts. The delocalized detectors are a combination of the Eulerian and the Lagrangian approach to vortex region extraction. The basic message here is that the Eulerian and the Lagrangian are not different alternatives to vortex extraction, opposite to each other, but that they can be combined to one technique sharing the benefits of both. The good local vortex detection performance of the Eulerian criteria and the global information of the Lagrangian view point combine to generate well separated and smooth detection results.

Chapter 6

Parallel Vectors Criteria for Unsteady Flow Vortices

> "Often there's no need to tear off an arm to remove a splinter. "
> (Frank Herbert 1920 – 1986)

Feature-based flow visualization is naturally dependent on feature extraction. To extract flow features, often higher-order properties of the flow data are used such as the Jacobian or curvature properties, implicitly describing the flow features in terms of their inherent flow characteristics (e.g., collinear flow and vorticity vectors). In this chapter we present recent research which leads to the (not really surprising) conclusion that feature extraction algorithms need to be extended to a time-dependent analysis framework (in terms of time derivatives) when dealing with unsteady flow data. Accordingly, we present two extensions of the parallel vectors based vortex extraction criteria to the time-dependent domain and show the improvements of feature-based flow visualization in comparison to the steady versions of this extraction algorithm both in the context of a high-resolution data set, i.e., a simulation specifically designed to evaluate our new approach, as well as for a real-world data set from a concrete application.

6.1 Motivation

We present a solution to the challenge of *feature extraction* when dealing with *time-dependent* simulation data from computational fluid dynamics. We aim at feature-based flow visualization with focus on vortices and their central locations. In an extension of the state of the art we present two new methods for the extraction of *vortex core lines* (aka. *vortex axes*[1]) in *unsteady flow* which are truthful to the time-dependent nature of the extracted features.

---

[1] In other fields, e.g., in fluid mechanics, *vortex cores* are considered to be of regional type (and not of line type). We use the term *vortex core line* for line-type curve features which represent central locations in vortices.

A lot of work has been done in the field of feature extraction from *steady/time-independent* flow data, especially with focus on vortices. In the context of time-dependent flow previous work focussed on extracting features from individual time steps by interpreting the flow data as a “stack” of steady flow fields (one per time step) and by applying extraction methods for steady flow data accordingly. The time-dependent nature of these features was taken into account by connecting them afterwards over time, e.g., by tracking.

It is favorable to inherently consider time already during feature extraction and not separately in a second step. Doing so, we find ourselves aligned with others (such as Hussain already in 1983 [57]), who demand the joint consideration of space and time when investigating features in time-dependent flow data. Accordingly, we propose to formulate the extraction criterion in a way that temporal derivatives are used for the local characterization of vortices and not only the Jacobian of the flow. This is synonymous to considering pathlines for feature extraction from unsteady flow instead of streamlines. Even though we experienced in exchange with colleagues, reviewers, and others that this extension is easily and quickly considered to be logical and straight forward, the results improve more than expected.

Very often, flow phenomena such as gas flow during combustion or air flow around a vehicle are time-dependent in their nature and steady representations are just an approximation. Data sets with time-independent flow are useful for domain experts as they provide information about general or large-scale characteristics of the flow, at a relatively low cost in terms of data set size, simulation time, as well as analysis time. However, we still observe a clear trend towards more unsteady flow data in scientific as well as in commercial applications mostly because of better results, especially when doing a more careful or detailed flow analysis, and also because of the availability of increased computing and storage resources.

Accordingly, we consider it important to explicitly demonstrate that feature extraction based on time, is not only logical to do, but indeed yields better results. In certain cases, we can even observe that the traditional, streamline-oriented approaches lead to displaced “features”. Furthermore, we can find an improved agreement of the new approach with physical extraction schemes such as the low-pressure assumption in the midst of vortices (no need for a correction step). In Sections 6.2 and 6.4 we exemplify our point by means of selected cases both in analytic and computed form. The need for a new approach is demonstrated as well as the gain through improved results.

The algorithms, which we take as a basis for developing our new approach, are the proven method for extracting vortex core lines from steady flow data by Sujudi and Haimes [140] as well as the related, higher-order method by Roth and Peikert [122]. Both approaches were successfully applied in many cases, especially when dealing with time-independent data. As such, we consider them as a strong starting point for approaching the case of unsteady flow data. To do so, we adopt the principle of the parallel vectors operator [113] for extracting the vortex core lines in conjunction with modified extraction criteria that are based on temporal derivatives.

Weinkauf et al. [157] approach the question of vortex core line extraction in a similar fashion. For finding ”swirling particle cores” they analyze the real eigenvector of the velocity gradient and the acceleration vector. Even though they arrive at a similar extraction method, they reason differently (and use other, related vectors for their approach). Our solution, as presented in this chapter, is based on physical principles, resulting in a corresponding modification of existing algorithms. The swirling particle cores method is based on the space-time framework and builds primarily on a geometric approach. In future work we plan to evaluate and compare the two approaches thoroughly. We are also able to demonstrate our work in the context of an application, compare it to other, simulated quantities related to vortices, and show its good numerical behavior regarding time step size in the data set.

6.2 Analytic Considerations

In the following, we discuss two analytic examples which can be considered as models for related phenomena in actual flow data. This way we can concentrate on the demonstration of the need for a new approach. Looking at analytic cases we can avoid issues related to sampling and reconstruction. This approach is analogous to the work of others who use analytic examples for motivation and for demonstration [140, 122, 46].

6.2.1 A Tilting Vortex

To construct our first synthetic vortex example, we aimed at an as simple as possible flow model that still can demonstrate the difference between a streamline- and a pathline-based approach. To avoid a simultaneous discussion of whether our approach is Galilean invariant we decided to go for one simple vortex which tilts over time.

Accordingly, we specify our flow model as

$$\mathbf{u}\,(x, y, z,\, t) = \begin{pmatrix} -y + tz \\ x - tz \\ z \end{pmatrix}.$$

The vortex in $\mathbf{u}$ is linearly strained in the z-direction and contains a tilt which increases over time. Considering $\mathbf{u}$ in just one time step $t = t_a$ and analyzing its – in all locations equal – Jacobian

$$\mathbf{J}|_{t=t_a} = \begin{pmatrix} 0 & -1 & t_a \\ 1 & 0 & -t_a \\ 0 & 0 & 1 \end{pmatrix},$$

by considering the single real eigenvector $(t_a, 0, 1)^{\mathsf{T}}$ of this matrix we observe a virtual[2] rotation of the instantaneous flow field around an axis which is aligned with this vector and which tilts into the positive x-direction. In the top row of Fig. 6.1 this situation is illustrated for two time steps $t_a = 0$ (left) and $t_a = 0.3$ (right).

---

[2] We consider this rotation as “virtual” as it only exists for an infinitesimal short moment of time – the vortex axis which is detected locally in time does not yield any tightly related finite-time rotation of particles around this axis.

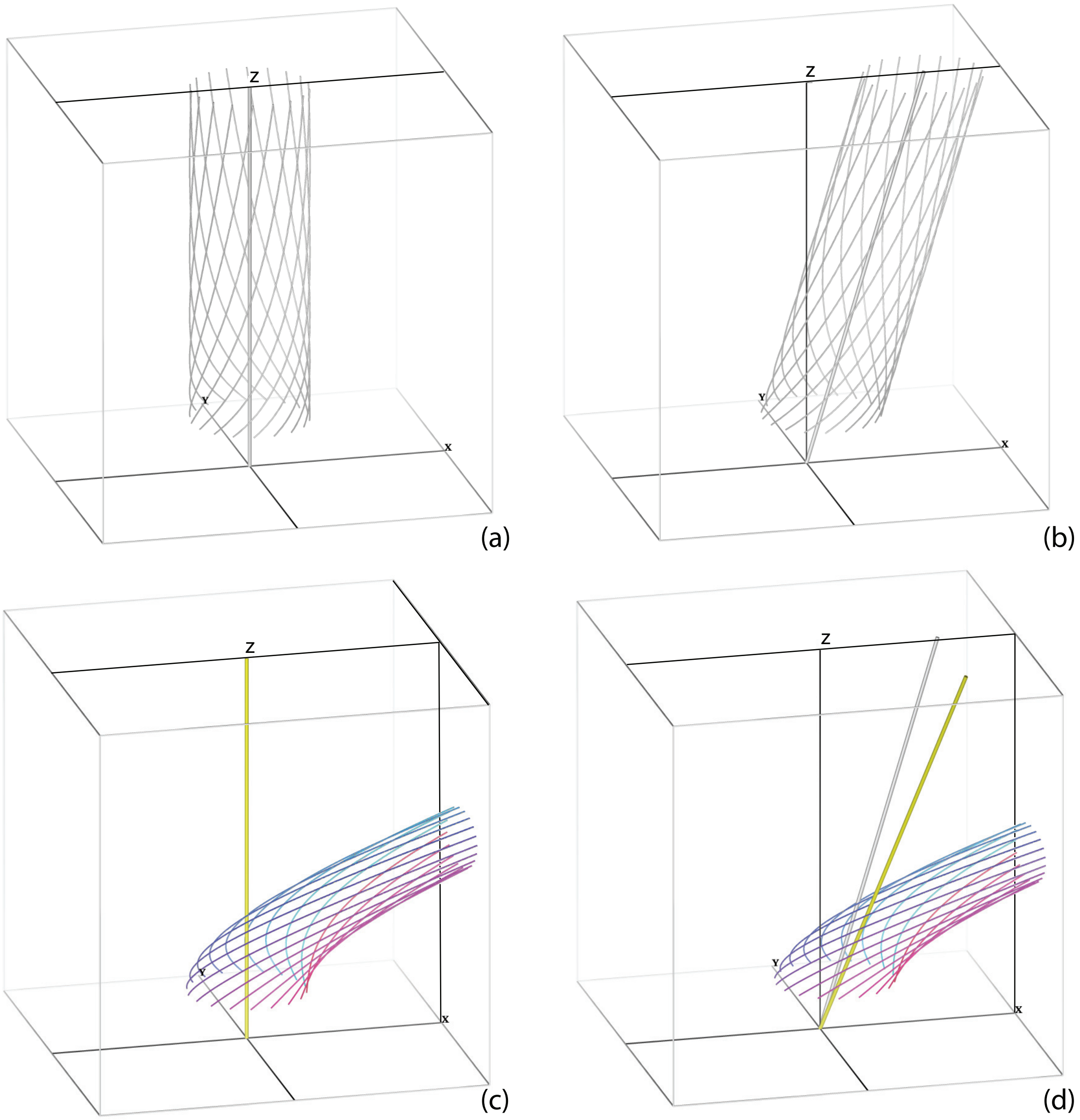


Figure 6.1: A synthetic example of a tilting vortex is shown before the tilt (at the left) and a bit later (on the right). The top row shows the vortex core line (grey tube) according to Sujudi and Haimes [140] and several streamlines – the tilt into the x-direction is obvious. The bottom row shows pathlines (in color) which exhibit an additional tilt towards the viewer (yellow vortex core line).

We abandon the restriction to only consider the flow in just one time step and see a different picture (bottom row of Fig. 6.1). In addition to the above mentioned x-tilt, there is another tilt towards the viewer. The corresponding vortex core line illustrated in yellow in Fig. 6.1 (d) reflects this additional y-tilt.

The design of this flow model allows to analytically find explicit solutions for stream- and pathlines. If we first consider just one time step $t = t_a$, we derive the streamline for seed location $(x_0, y_0, z_0)^\mathsf{T}$ in parameterized form as

$$\begin{aligned} x(\tau) &= (x_0 - t_a z_0)\cos(\tau) - y_0 \sin(\tau) + t_a z_0 e^\tau, \\ y(\tau) &= (x_0 - t_a z_0)\sin(\tau) + y_0 \cos(\tau), \\ z(\tau) &= z_0 e^\tau. \end{aligned}$$

The $t_a z_0 e^\tau$ term in the x-component of streamlines reflects the above discussed x-tilt. In the y-component of streamlines we do not see any corresponding tilt term.

Considering pathlines next, we derive the following solution (now parameterized with time t):

$$\begin{aligned} x(t) &= (x_0 + \tfrac{1}{2}z_0)\cos(t) - (y_0 + \tfrac{1}{2}z_0)\sin(t) + (t - \tfrac{1}{2})z_0 e^t, \\ y(t) &= (x_0 + \tfrac{1}{2}z_0)\sin(t) + (y_0 + \tfrac{1}{2}z_0)\cos(t) - \tfrac{1}{2}z_0 e^t, \\ z(t) &= z_0 e^t. \end{aligned}$$

Now we see corresponding tilt terms in both the x- and the y-components of the pathlines and the vortex axis is found to be along the vector $(t, -t, 1)$.

6.2.2 A Rotating Vortex Rope

As a second example, we construct a simple synthetic model of a rotating vortex rope that has characteristics which are related to an important flow phenomenon in the draft tube of large water turbines. To start, we consider the flow field

$$\mathbf{u} = \begin{pmatrix} -(y - y_1) \cdot s \\ (x - x_1) \cdot s \\ 1 \end{pmatrix}.$$

For the degenerated case of $x_1 = y_1 = 0$, this simply is a rigid rotation about the z-axis. Assuming that the points (x_1, y_1, z) lie on a helix with radius R and pitch $\frac{2\pi}{k}$, which rotates around the z-axis with angular frequency ω and phase 0, i.e., with

$$\begin{aligned} x_1 &= R \cdot \cos(kz + \omega t) \text{ and} \\ y_1 &= R \cdot \sin(kz + \omega t), \end{aligned}$$

we get a rotating vortex, i.e., a time-dependent flow field as desired – see Fig. 6.2 for selected stream- and pathlines. Note, that we assume $|k + \omega| < s$ to ensure that the structure of the helix dominates the rotation about it.

Based on this model, we can analytically derive several variants of vortex core lines (according to different extraction schemes). In all cases we obtain a helix with the same

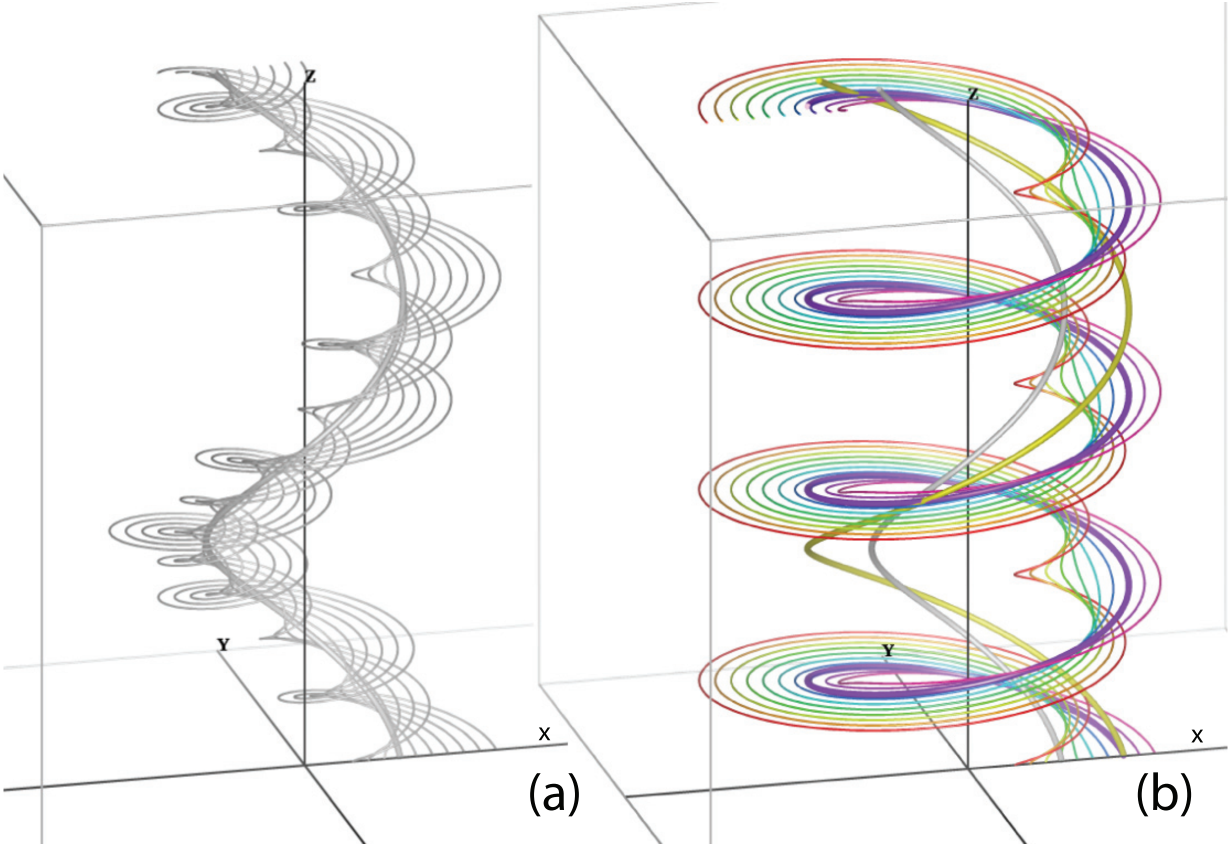


Figure 6.2: Streamlines and pathlines in a model of a rotating vortex rope. (a) The vortex core line based on streamlines (according to Sujudi and Haimes [140]) is shown as a thick grey tube (it is the only grey line which also is a helix). (b) The vortex core line based on pathlines (shown in yellow on the right) has the same pitch but a larger radius (it is the only helical pathline, shown in thick magenta).

pitch, frequency, and phase, but with different radii. See table 6.1 for an overview of the results.

The employed methods are three state of the art approaches for steady flow data: the method proposed by Levy et al. (curl parallel to velocity [91]), the one by Sujudi and Haimes (parallel first and second derivatives of streamlines [140]), and the higher-order method by Roth and Peikert (parallel first and third derivatives of streamline [122]). We apply them to the flow data of individual time steps as discussed above.

We contrast these results with those of our new approach, i.e., the unsteady extension of Sujudi and Haimes's (as described in Section 6.3.1) and the unsteady version of the higher-order approach (as described in Section 6.3.2). We see that the traditional approaches miss the rotation of the vortex rope (missing '$+\omega$' terms in all cases), since it obviously cannot be detected from considering an individual time step only.

We also compute a correct vortex core line for this unsteady flow by using a sym-

	streamline-based	pathline-based (new)
Levy et al.	$(1+\frac{k}{2\cdot s})R$	
Sujudi & Haimes	$(1+\frac{k}{s})R$	$(1+\frac{k+\omega}{s})R$
higher-order	$(1+\frac{k}{s}+(\frac{k}{s})^2)R$	$(1+\frac{k+\omega}{s}+(\frac{k+\omega}{s})^2)R$
correct		$(1-\frac{k+\omega}{s})^{-1}R$

Table 6.1: Different extraction schemes all result in helical vortex core lines, but with different radii. We compare the results for the algorithms of Levy et al. [91], Sujudi and Haimes [140], the higher-order method by Roth and Peikert [122], and an analytically determined correct variant.

metry argument. On each slice orthogonal to the z-axis ($z = z_{\text{const}}$), there is just one point $(x, y, z_{\text{const}})^{\mathsf{T}}$, with

$$x = \frac{R \cdot \cos(kz)}{1-(k+\omega)/s} \quad \text{and } y = \frac{R \cdot \sin(kz)}{1-(k+\omega)/s},$$

such that a particle which is released from this point at time 0 moves along a pathline of exact helical shape. Particles that are released from any other location yield pathlines of more complicated geometry (Fig. 6.2). In this case, we see that the material line (time line), which consists of all of these special particles, coincides with the correct vortex core line. This curve has the same radius as the helical pathlines, but exhibits a different pitch of $\frac{2\pi}{k}$ vs. $\frac{2\pi}{k+\omega}$. We note, however, that the fact that the vortex core line also is a material line is specific to this example and does not generally hold for arbitrary cases.

By comparing the different radii from table 6.1 with the correct solution and by considering the geometric series $(1-p)^{-1} = (1+p+p^2+\ldots)$, here with $p = (k+\omega)/s$, we can see a nice alignment of our new approach with the correct solution. The modified variant of the approach by Sujudi and Haimes is the first-order approximation of the correct solution and the modified variant of the higher-order approach is its second-order approximation.

The deviation of the Sujudi-Haimes lines from the correct vortex core lines is the phenomenon first observed in the "bent helix" example [123], and it is due to the combination of a weakly rotating vortex and a strongly curved vortex core line. The error becomes negligible if $|k+\omega| \ll |s|$, i.e. if the sum of the spatial and the temporal frequency is much smaller than the parameter s controlling the swirl around the vortex core line. The higher-order method yields an additional term of the Taylor series in this example.

We have seen that the extension to unsteady flow for both methods results in improved results in comparison with the time frozen analysis of vortex flow features. To understand what is happening with unsteady vortices it is necessary to extend the steady versions of the vortex extraction criteria.

6.3 Pathline Based Feature Detectors

We can generalize existing feature extraction algorithms to unsteady flow data by replacing streamlines with pathlines in the underlying model. This way they remain unchanged for steady flows.

6.3.1 Sujudi-Haimes

In this section we modify the approach by Sujudi and Haimes [140] to include time derivatives.

6.3.1.1 Original Definition

In the original definition the first step is to compute the eigenvalues of $\nabla \mathbf{u}$ per tetrahedral cell. Only cells where a pair of complex eigenvalues exists are further processed. The existence of two complex eigenvalues is determined by the discriminant of the characteristic polynomial [15].

The next step is to compute the single real eigenvector $\epsilon_{\mathbf{r}}$ for the candidate cells to extract the local direction of the vortex core line. In the final step the algorithm searches for locations where $\epsilon_{\mathbf{r}}$ is parallel to $\mathbf{u}$. Linear interpolation is used between the nodes of a grid cell when searching for parallel locations. A modification in order to get connected lines instead of disjoint straight line segments is to estimate velocity gradients per node and compute parallel positions on cell faces.

6.3.1.2 Equivalent Definition

The eigenvector computation required by the original method is quite expensive. A more efficient method [113] is to compute the matrix-vector product $\mathbf{a_s} = (\nabla \mathbf{u})\, \mathbf{u}$ instead. Given that $\epsilon_{\mathbf{r}}$ is the only real eigenvector of $\nabla \mathbf{u}$, it is parallel to $\mathbf{u}$ exactly if $\mathbf{a_s}$ is. Hence, Sujudi-Haimes vortex core lines can be equivalently defined as the locus of points where $\mathbf{u}$ and $\mathbf{a_s}$ are parallel, restricted to points where the velocity gradient has a pair of complex eigenvalues. In this context, two vectors are said to be parallel also if one or both of them are zero.

6.3.1.3 Modification for Unsteady Flow

The original formulation of the Sujudi-Haimes criterion is expressed in terms of the velocity field and its gradient tensor field. Using this formulation we cannot include the time derivative information since these quantities are the same for steady and unsteady flow. In

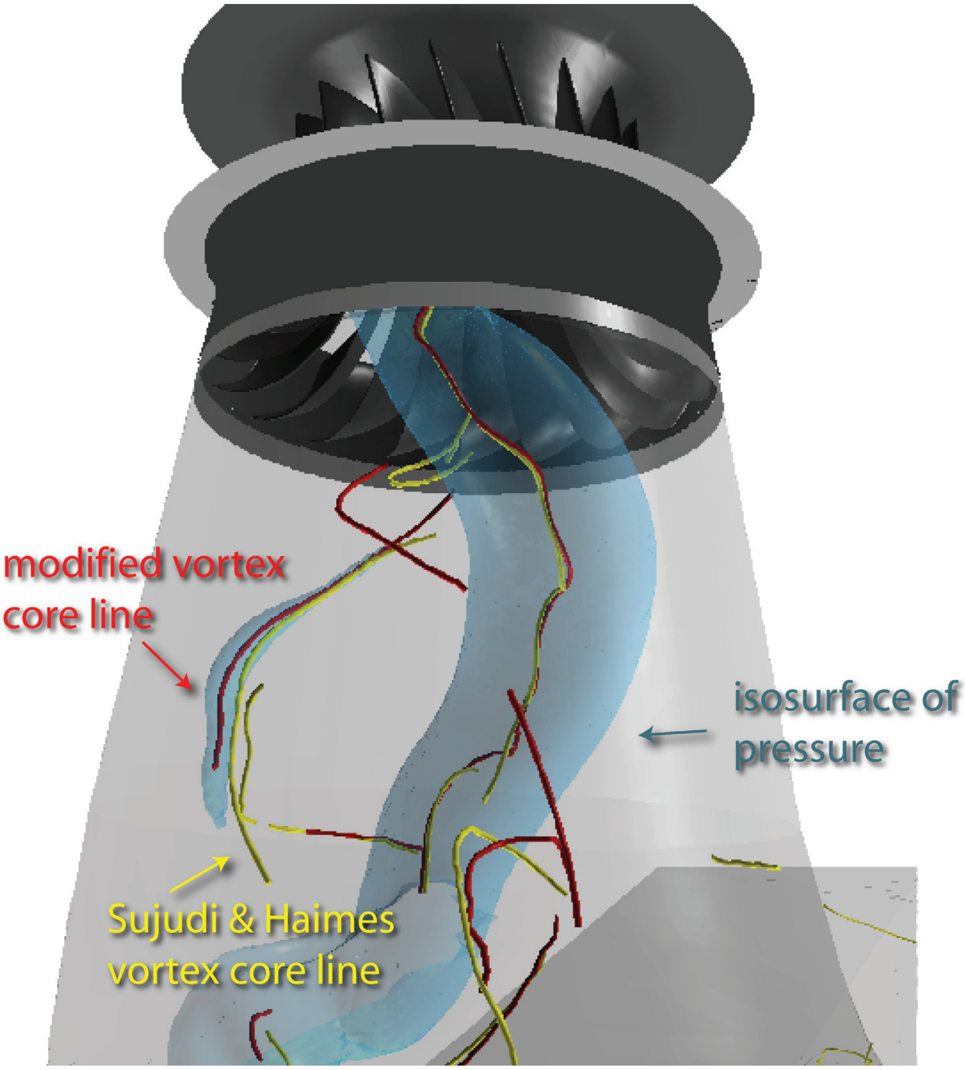


Figure 6.3: For the vortex rope in the depicted data set, iso-values of pressure give good insight on where the vortex core line is located. We can clearly see how the yellow core line (extracted using the classical approach of Sujudi and Haimes [140]) deviates from the center of pressure isosurface. The modification to time derivative aware extraction of the vortex core line improves the results visibly.

contrast, the parallel vectors formulation allows for a different extension to unsteady flow. The vector $\mathbf{a_s} = (\nabla\mathbf{u})\,\mathbf{u}$ can be viewed as the steady case of the acceleration vector

$$\mathbf{a_t} = D\mathbf{u}/Dt = (\nabla\mathbf{u})\,\mathbf{u} + \partial\mathbf{u}/\partial t$$

of a particle. An obvious modification is now to use the true acceleration vector instead of the vector $\mathbf{a_s}$, i.e. to look for points where $\mathbf{a_t}$ and $\mathbf{u}$ are parallel. Besides the justification as being the natural extension to unsteady flow, this modification is also backed up by the following observation.

Sujudi-Haimes vortex core lines can be defined in a third equivalent way, namely as the locus of zero streamline curvature, again constrained to points where the velocity gradient has a pair of complex eigenvalues. The equivalence is shown as follows. The curvature of a curve with (time) parameter is $\kappa = \|\dot{\mathbf{x}} \times \ddot{\mathbf{x}}\| / \|\dot{\mathbf{x}}\|^3$ where the dots denote temporal derivatives. For a streamline, $\dot{\mathbf{x}} = \mathbf{u}$ and $\ddot{\mathbf{x}} = \mathbf{a_s}$, so the streamline curvature is zero exactly where the Sujudi-Haimes criterion is met. For a pathline, $\ddot{\mathbf{x}}$ is $D\mathbf{u}/Dt$ so the pathline curvature is zero exactly where the modified Sujudi-Haimes criterion is met.

In principle, the zero curvature points of streamlines or pathlines could be computed to yield vortex core lines according to the original or modified Sujudi-Haimes criterion. However, numerical integration and curvature computation are too expensive operations to make this a practical alternative to the parallel vectors method.

It was a long standing open question from our application partners why the vortex core lines resulting from the original algorithm of Sujudi and Haimes very often exhibit a small phase-shift in relation to regions of low pressure. Therefore it is a common approach to do a correction step towards pressure minima when extracting vortex core lines [4, 61]. In Figure 6.3 we can see that the yellow vortex core lines extracted using the eigenvector method are shifted away from the center of the pressure isosurface. Using the pathline based extraction approach we arrive at a solution located at the pressure minima without a correction step. Therefore we can assume that the deviation in the unmodified approaches results from not taking the temporal derivative into account.

6.3.2 Higher Order Vortex Core Lines

In this section we modify the higher order approach to work on pathlines.

6.3.2.1 Original Definition

Roth and Peikert [122] present an extension of the vortex extraction approach by Sujudi and Haimes to bent vortices. The eigenvector is based on a straight line model for the vortex core line. In real world data sets we can find many types of bent vortices though. Common types are hairpin, horseshoe, and ring shaped vortices. Roth and Peikert showed [123] that the eigenvector method introduces an error as soon as the vortex is bent.

To overcome these drawbacks we can weaken the conditions on a vortex core line such that we can detect bent vortices as well, but the amount of false positives will increase significantly. It is not possible to model a curved vortex based on linear fields, therefore one has to take into account higher-order derivatives when searching for vortex core lines. The second derivative following a particle in a steady velocity field is $\mathbf{b_s} = (\nabla\mathbf{a})\mathbf{u}$.

Based on the torsion of a parametric curve in $\mathbb{R}^3$ we can relax the condition on vortex core lines such that torsion is zero and that zero torsion is preserved as well as possible when following the streamline. The extraction algorithm is based on the fact that for the bent vortex model the vector $\mathbf{b_s}$ at the vortex core line is not only restricted to the $< \mathbf{u}, \mathbf{a_s} >$ plane but that the best choice is to require that $\mathbf{b_s}$ is parallel to $\mathbf{u}$. Thus, we can state the following definition for a vortex core line: the vortex core line is the location of all points where $\mathbf{b_s}$ is parallel to $\mathbf{u}$.

6.3.2.2 Modification for Unsteady Flow

The problems observed for curved vortices in steady flow data [122] obviously extend to curved vortices on unsteady flow data. In Section 6.2.2 we have seen that the modified version of the higher order model will reproduce the correct vortex core line of the bent time dependent model if we ignore the terms of higher order in the Taylor expansion. Therefore, we can use the parallel vectors operator to apply the higher-order approach to unsteady flows.

A criterion based on zero curvature in principle searches for straight vortex core lines. The line that is classified as the vortex core line by the parallel vectors approach of the previous section can deviate to some extend from this restriction. But for strongly bent vortices the result will show the same inconsistencies as observed for streamline based geometries. For the higher order vortex core line detection algorithm the required modification is therefore to replace the vector $\mathbf{b_s}$ by the actual jerk vector (rate of change of acceleration) $\mathbf{b_t} = \mathbf{D^2u/Dt^2}$. See Figure 6.7 for an example.

6.3.3 Interactive Vortex Core Line Extraction and Filtering

Both the eigenvector method and the higher order method produce many line segments that cannot be considered as vortex core lines. For this reason we use the interactive visual analysis features of the SimVis framework to extract the meaningful vortex core lines. This way we get confidence in the extracted vortex core lines and can improve their quality. Here we rely on non-binary expressions of vortex detectors to select the vortex core lines of interest [12]. The other way round we use the extracted core lines to derive other attributes in the data. Figure 6.4 illustrates this approach.

To our knowledge there is no fully satisfying approach to extract only the relevant vortex core lines automatically from the data. The interactive multi-field approach of SimVis handles this problem using visual analysis. To be able to do this we modify the parallel vectors algorithm slightly:

1. Generate additional field $\mathbf{a_t}$ or $\mathbf{b_t}$ (see Section 6.3.1.3, Section 6.3.2.2 and Appendix A).
2. Compute closed parallel vectors lines without additional criteria (see Section 6.3.4).

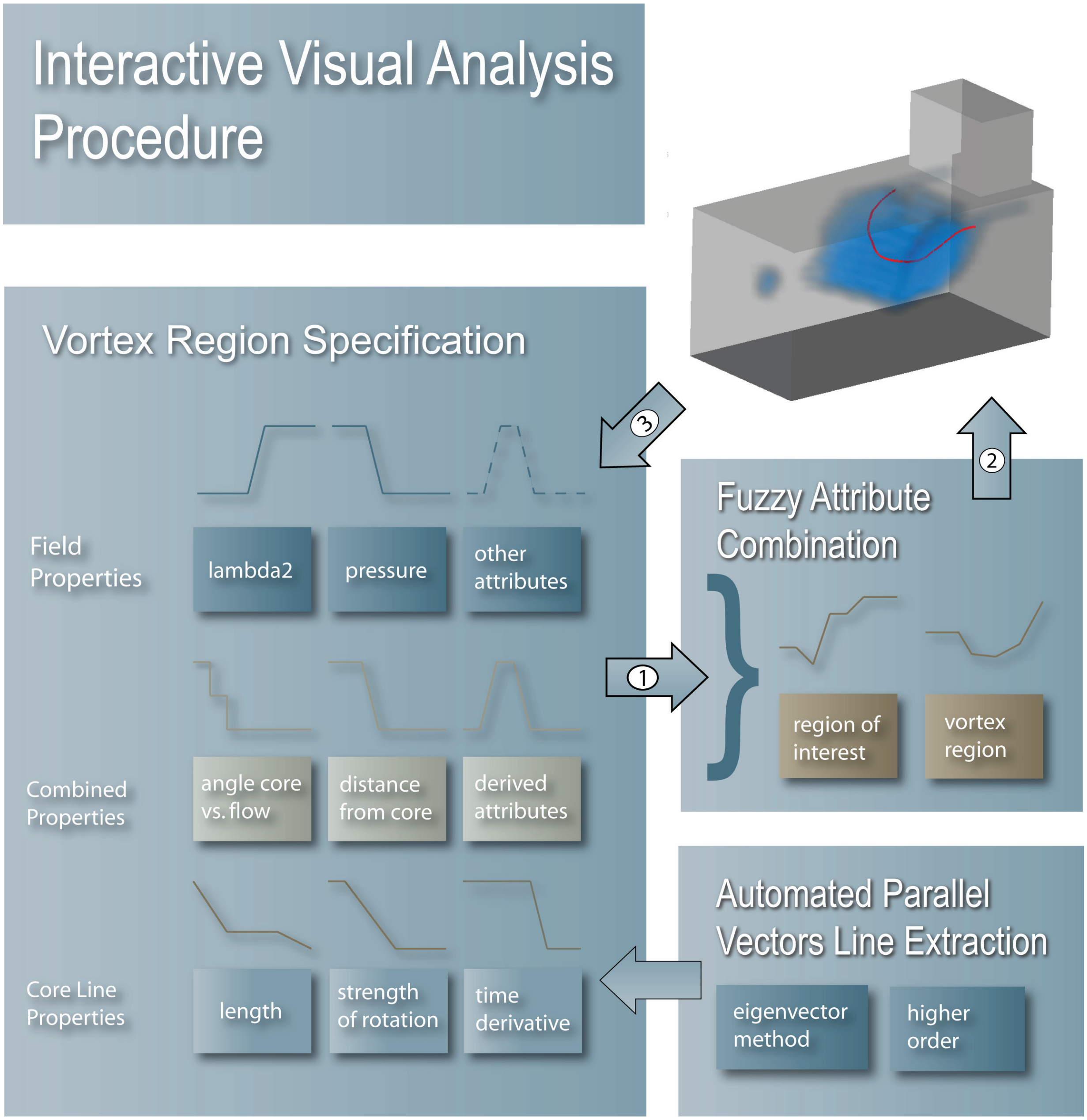


Figure 6.4: Interactive visual analysis with vortex core line extraction. After the vortex core lines are computed we use interaction to remove false positives. (1) The user can interactively specify the volume of interest in attribute views to select attribute ranges of interest and another set of attribute selections that control the vortex region. (2) The selected region of interest is visualized by volume rendering (in this example the volume selection is defined by $\lambda_2 < -100$) and the vortex region controls which line segments are visible (we have selected regions that have both complex velocity gradient eigenvalues and negative λ_2 values). (3) From the vortex core lines we can derive additional attributes such as an attribute measuring the distance from the core line for further analysis.

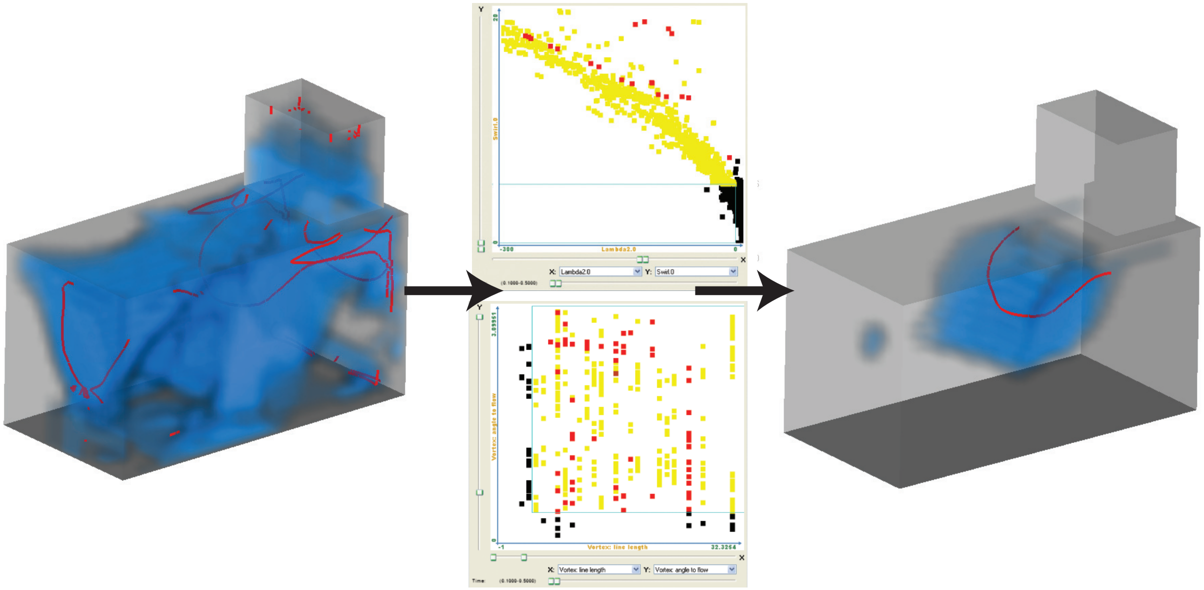


Figure 6.5: Multiple views and brushing allow the user to apply vortex core line filter rules interactively. Starting with a large number of spurious solutions we can select the main part of the largest vortex core line by applying two brushes in two scatterplots of additional flow attributes.

3. Use interactive region of interest specification to extract correct subsections of the lines (see Figure 6.4).

The δ-discriminant used as an additional criterion both by the method of Sujudi and Haimes and the higher order method was introduced by Chong et al. [15]. This physics-based criterion does not take into account the time-dependent components of the flow. Nevertheless physics-based criteria such as δ, Q, and λ_2 are often directly applicable to unsteady flow, when it is possible to derive them from instantaneous properties of the flow. The δ-criterion is prone to finding false positives in large regions of the flow (e.g. in the turbine data set it is true almost everywhere). In our experience it has shown to reduce the number of spurious solutions to use additional vortex core region detectors in combination with the δ-criterion. Another type of additional criteria includes information derived from the vortex core line [113]. Examples are the angle between flow and vortex core line, number of core line segments or vortex strength. These are difficult to tune optimally. By combining multiple vortex region criteria as suggested in [12] we can avoid criteria involving the extracted vortex core line.

Building on the information we get from the extracted vortex core lines, we get access to a whole new type of information that we can use in further analysis steps. To include information on the vortex core line we derive for each cell an attribute that measures the distance from the final vortex core line in a simple breadth-first traversal starting with cells that contain a vortex core line segment.

6.3.4 Pseudocode

The following is a general algorithm outline. The method 'getTriFaces()' returns a vector of triangulated faces for a cell. The method 'checkFaces()' applies the parallel vectors operator to each face in a list and returns a list of points where v and w are parallel. The method 'getTets()' returns a tetrahedralization of a cell using face centers and the cell center as additional points. The analytic solution of the parallel vectors problem can result in three result points per face and we can have up to 12 result points per tetrahedron. The method 'connect()' has to apply a heuristic to connect the result points. Possible heuristics include minimization of core line curvature, minimal core to flow angles or similar vortex strengths λ.

```
INPUT: unstructured grid
OUTPUT: array of line segments

lines *core = new lines();
//pre-processing
cells *c = grid->getVertexCells();

//check each cell for vortex-core
foreach cell ∈ c {
    //quick: check for two face intersections
    vector<tri> *tris = cell.getTriFaces();
    result = checkFaces(tris);

    if(result.size() == 2)
        core->add(result);

    //fallback: check all tets
    else {
        //get tetrahedralization
        tets *t = cell.getTets();
        foreach tet ∈ t
            core->add(
              connect(
                checkFaces(tet) ) );
    }
}
return core;
```

The following is pseudo code of the parallel vectors operator on a triangle. If the determinant of incrV is zero, also incrW has to be checked. The method 'inverse()' returns the inverse matrix. The method 'realEigenV()' returns a list of eigenvectors of a matrix having real eigenvalues. It is based on the nice trick that an analytic solution can be computed using matrix inversion and eigenvector computation, published by Peikert and Roth [113]. Assuming linear interpolation on a triangular face of a cell, the velocity can be written in local coordinates: $v = incrV \cdot (s,t,1)^T$, where $incrV$ is the Matrix $(v2 - v1, v3 - v1, v1)$. The velocity at each point on the triangle is then simply $v = v1 + s \cdot (v2 - v1) + t \cdot (v3 - v1)$. Two fields are parallel when $V \cdot (s,t,1)^T = \lambda W \cdot (s,t,1)$. If W (or V) is invertible we can multiply by W^{-1} (or V^{-1}) and get an eigenvector problem $W^{-1}V = \lambda x$.

```
INPUT: triangle (t1,t2,t3),
       flow (v1,v2,v3),
       accel (w1,w2,w3)
OUTPUT: parallel positions

//compute increments
mat incrV = (v2-v1, v3-v1, v1);
mat incrW = (w2-w1, w3-w1, w1);

//find parallel positions
if($det(incrV)! = 0$) {
    mat inv = incrV.inverse();
    mat sol = inv * incrW;
    vector *eig = sol.realEigenV();
    vector *pos;

    foreach e $\in$ eig {
        float s = e->x/e->z;
        float t = e->y/e->z;
        if(s>=0 && t>=0 && s+t<=1)
            pos.add(t1+s*t2+t*t3);
    }
    return pos;
}
```

6.4 Application Study – Engine Data Sets

We have implemented the presented vortex core line detection algorithms in the SimVis framework [179] and applied it to two engine data sets to verify the approach on real world data. For these data sets we have found that using the Green-Gauss approach for computing gradients gives better results than a least-squares approach (see Appendix A).

The first data set results from a simulation of the compression and combustion phase in the combustion chamber of a standard engine model. In Figure 6.6 we can see the vortex core lines based on the original and the modified versions of the parallel vectors criteria. Obviously the results differ significantly and one of the vortex core lines is not extracted

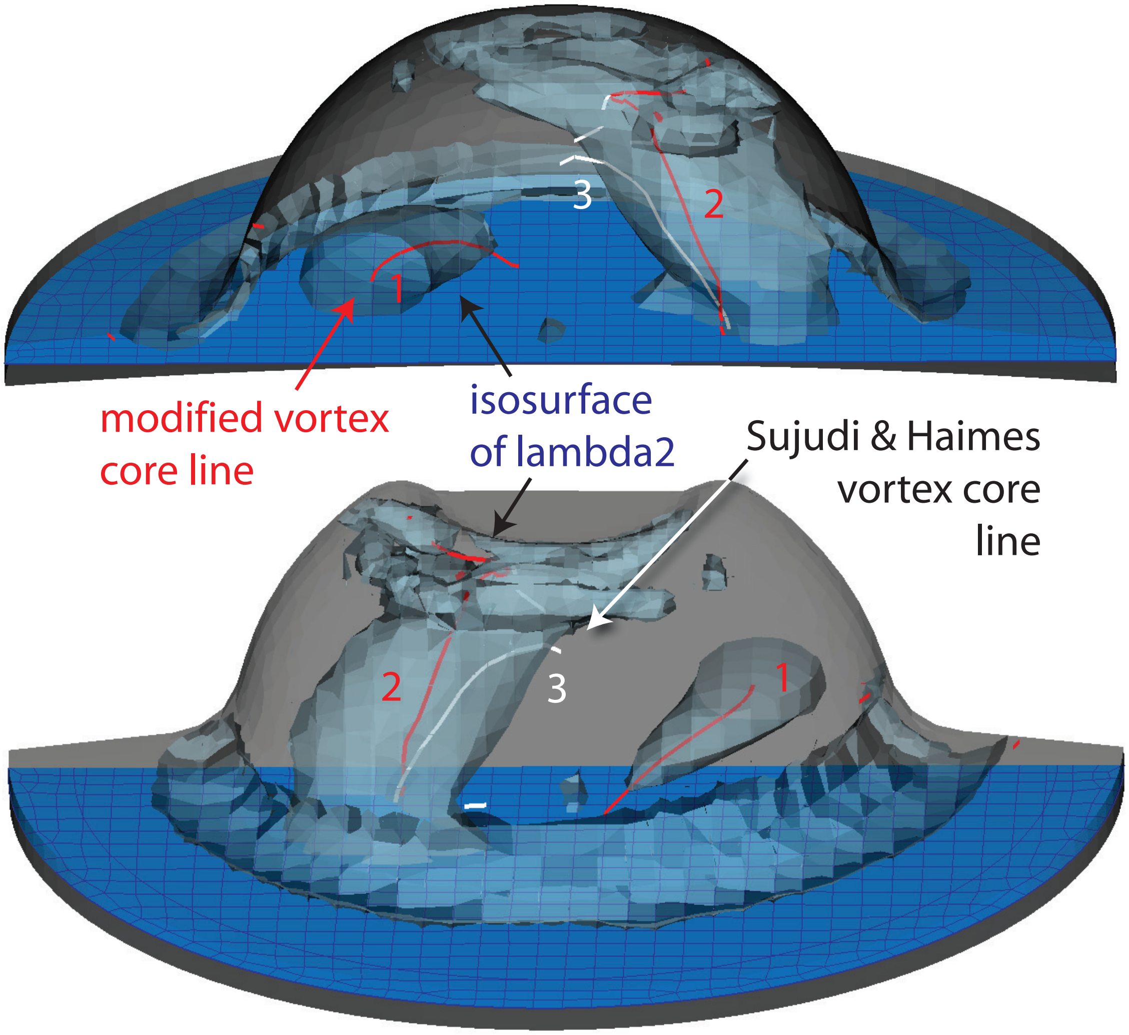


Figure 6.6: We compare the vortex core lines found by the original method of Sujudi and Haimes [140] and the modified version. Two views of the same timestep show the benefits of the modification. Both views show the same vortex core line and isosurface. (1) In one case the original method does not detect one vortex core at all. (2) The time-aware modification traverses the full length of the vortex core and continues into the region of strong turbulence at the top of the cylinder. (3) The original vortex core line leaves the core region of the vortex and vanishes in a substantial portion of the vortex region.

at all using the original algorithm.

The second data set is a high-performance two-stroke engine data set, which contains the complete simulation results from the injection and the combustion of fuel during one crank revolution. The engine geometry is shown in Figure 6.8. Table 5.1 shows a compar-

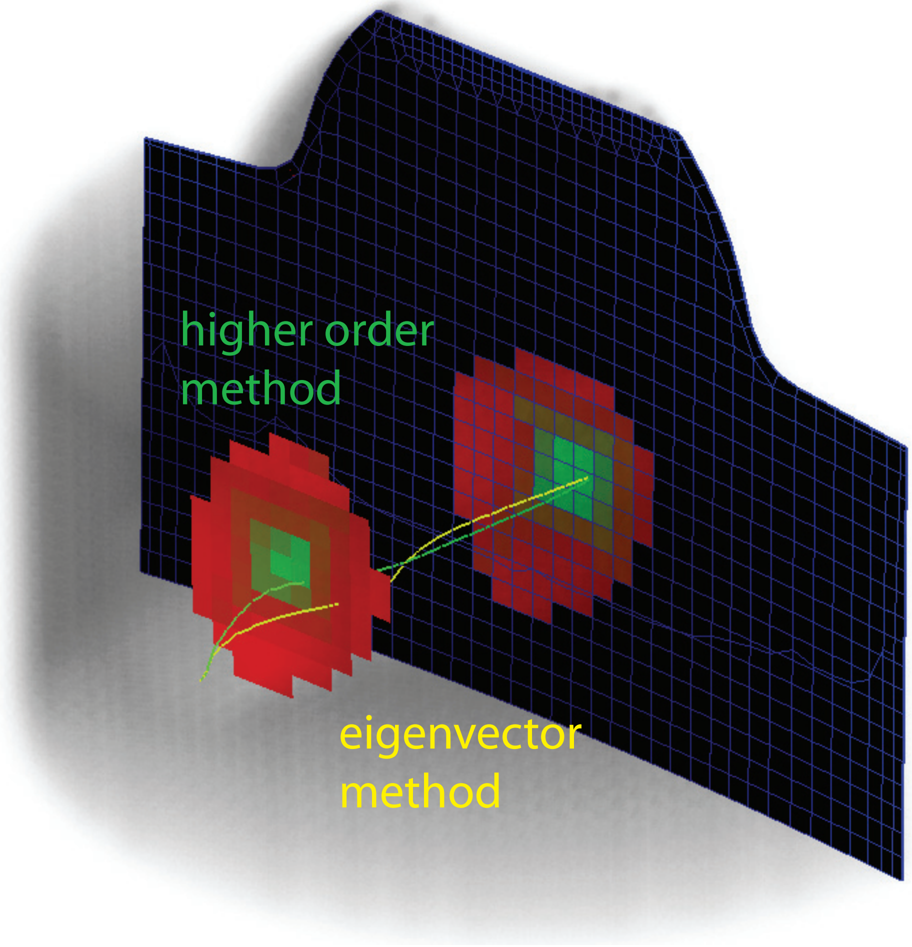


Figure 6.7: In this early timestep of the combustion chamber data set we can see that the extracted vortex core lines for the modified version of the eigenvector method and the modified higher order method differ at the weakest part of the vortex. The cutting plane with color mapped to pressure shows that the modified method of Sujudi and Haimes fails to detect the exact core line of the vortex in this case.

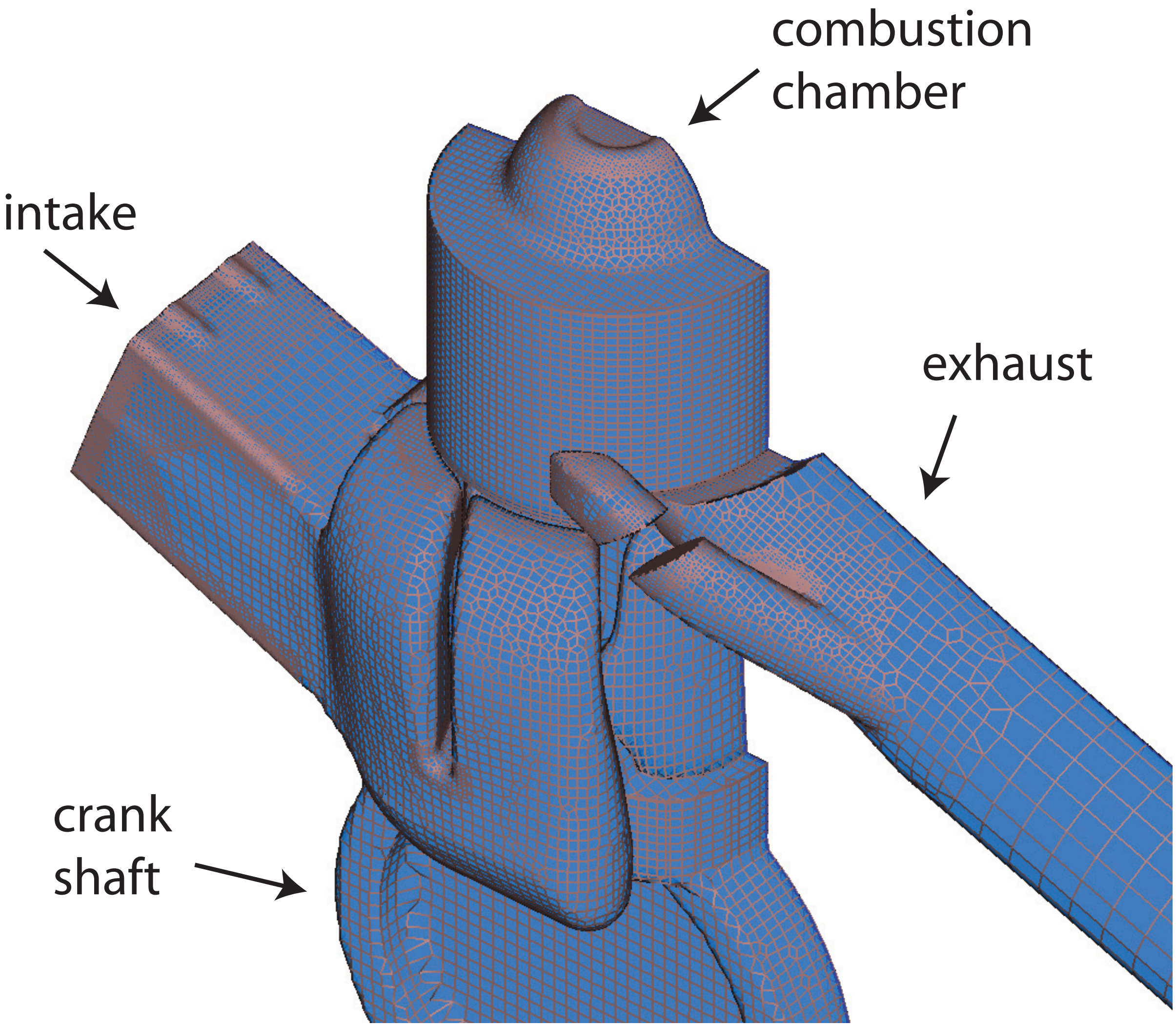


Figure 6.8: Overview of the geometry of the two-stroke engine data set.

ison of the data sets discussed in the following sections.

6.4.1 Impact of Time-derivatives

The question remains whether and where the time derivative information has significant impact on the vortex core line extraction results. In the engine data sets we have found the vortex core lines extracted by the modified and the unmodified methods to be similar but shifted for most timesteps. But in Figure 6.6 we can observe that in a timestep shortly after ignition the vortex core line based on a_s and the vortex core line based on a_t can differ significantly. This is due to the strong impact of the time derivative in these time steps. To illustrate the close correlation between these two vectors in early timesteps and the large impact of the time derivative after ignition we show the magnitudes of the vectors

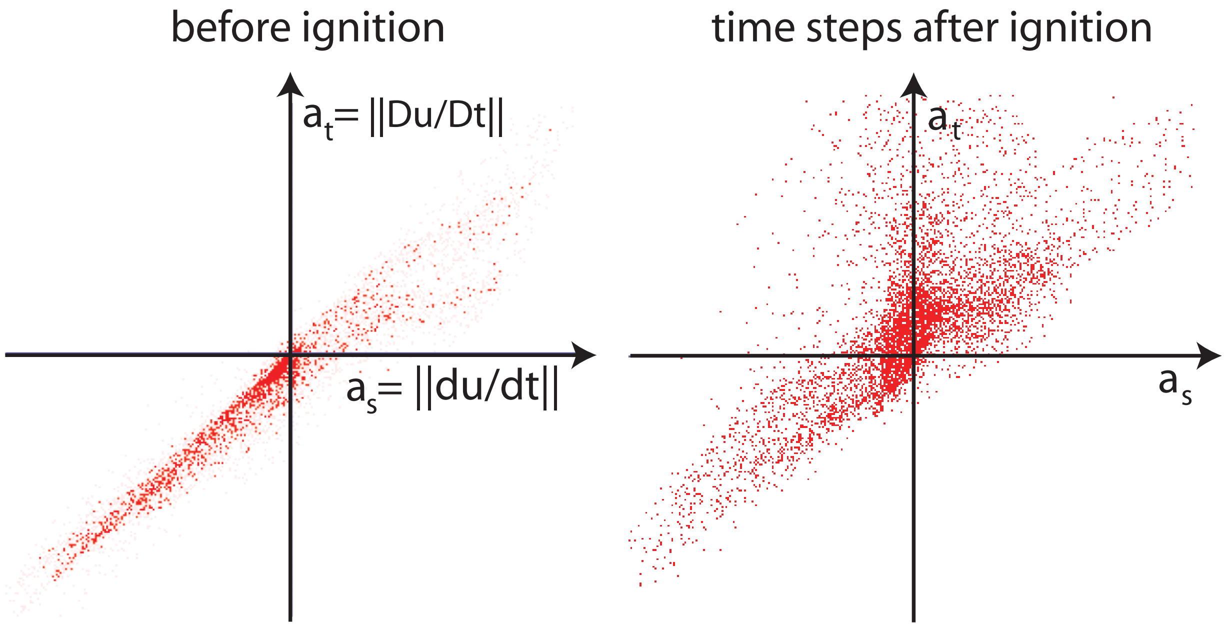


Figure 6.9: Comparison of acceleration vector magnitudes: The scatterplots show that the magnitudes of the two variants of the acceleration vectors can differ significantly in the crucial timesteps, i.e., timesteps of major changes over time, after ignition (we have normalized the magnitudes such that the center of gravity corresponds to the origin).

normalized with mean and standard deviation in scatterplots (see Figure 6.9). Very often the timesteps that include large changes over time are critical for the application. They have vital impact on mixing, material wear and engine performance and therefore the analysis benefits from improving vortex core line extraction in these time steps.

6.4.2 Equivalence Ratio

From the discussion in Chapter 4.6.1 we know that the equivalence ratio (ER) is a key attribute, related both to emission and engine performance. Equivalence Ratio is the relation between fuel and air within a volume cell. We have seen that It is crucial that ER lies in the optimal interval for large parts of the volume at the moment of ignition. The mixing process happens at earlier time steps during compression when the influence of the time derivative is less than after ignition. Even though the difference between the core lines generated by the modified version is smaller it is still not negligible. In Figure 6.10 (a) we show iso values of the λ_2 vortex detector and concentrate on the vortex core lines detected for this vortex. In the center of the combustion chamber of the two-stroke engine we can see the large vortex region that plays a central role in the mixing process. The question in this example is, why the vortex core region is not of tubular shape. The second vortex core line (3) is not detected by the original approach. Combining (1) and (3) we can gain insight into the controlling skeleton of the main tumble vortex. In Figure 6.10 (b) we can

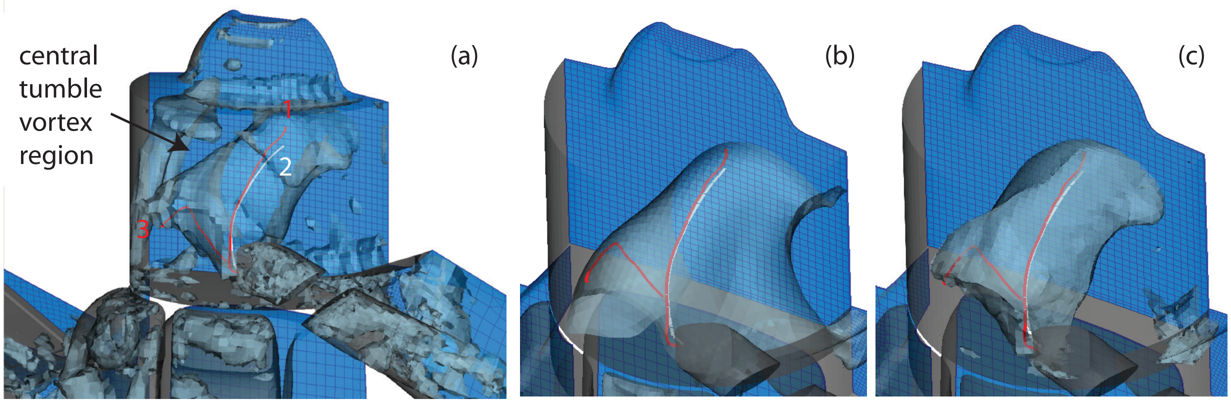


Figure 6.10: We compare the computed core lines with respect to λ_2 and equivalence ratio. (a) The modified algorithm detects two vortex core lines (red) whereas the original version only detects the main vortex core line (white). (b) An isosurface of equivalence ratio at 0.7 containing the region of optimal mixing. (c) The surface containing the region of equivalence ratio of 0.5 and $\lambda_2 \leq -1000$.

distinguish the regions of sub-optimal and optimal to very high concentrations of fuel at iso value surface of 0.7. The bend part of vortex core line (3) closely follows the boundary of this region. In Figure 6.10 (c) the surface describes the boundary of the region defined by slightly sub-optimal to high mixing and high λ_2 values. The core line generated for this vortex with the original vortex core extraction method (1) and the modified approach (2) are similar and both traverse the full region detected by the λ_2 vortex region detector. Another core line is not detected though. Obviously we miss an important aspect without the second vortex core line since we can see in Figure 6.10 (c) that it influences the region of the vortex where non optimal mixing occurs.

6.5 Assessment of Numerical Behavior

In engineering applications it is not common to store all the information computed in the course of the CFD simulation permanently. Especially time derivative information is not generally stored in the data. Furthermore, the solver does not include all the timesteps computed in the solution file. In general we can expect the simulation design regarding cell types and cell sizes to be adequately chosen by the simulation designer. The simulation designer considers the necessary resolution for postprocessing such that reliable streamlines and pathlines can be constructed. From experience we know these settings to work well for computing vortex core lines in the steady case. Since time-derivative information is not stored and not all time steps are written out into the final data set we need to evaluate the impact of larger step sizes on the feature extraction process. Our application partners from Arsenal Research [178] have computed an unsteady solution to a pulsating flow in a tube t-junction (see Figure 6.11). Time dependent boundary conditions are used to produce

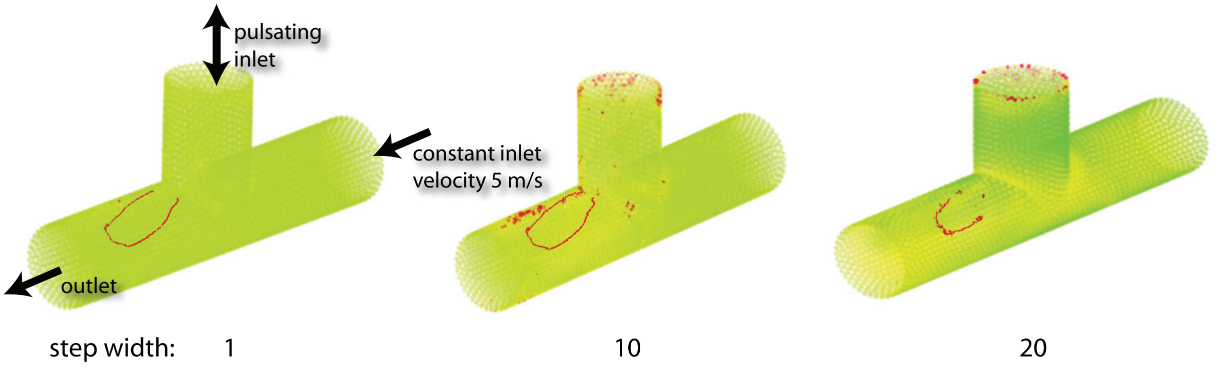


Figure 6.11: Impact of time derivative estimation. The different step sizes are measured in 1000^{-1} sec. The vortex core lines for stepwidths 1 to 10 do not differ visibly. Color is mapped to the difference between the time derivative for step size 1, and the respective step size (for step size 20 we have changed the color mapping by one order of magnitude).

flow separation inside the tube. The total mesh size is about 170000 cells.

During simulation 1570 timesteps have been generated resulting in 26 GB of compressed information. This is 10 times the temporal resolution our application partners would have stored usually for this simulation setting. To exclude possible interference from numerical problems introduced by the plane fitting technique we use to estimate the material derivatives also the Jacobian computed during the simulation have been included in the data set. This way we can analyze how strong the impact of larger timesteps is when computing vortex core lines. We can use the time derivative computed for step size 1 as reference for the other step sizes and measure the influence of larger step sizes by computing the difference between the reference derivative and the respective derivative for the given step size. In Figure 6.11 the magnitude of this difference is mapped to color. To analyze the impact on vortex extraction, we focus on a horseshoe vortex directly behind the top inlet. We see the difference between the acceleration vector from step size of 1 and step sizes 10 and 20. The vortex core lines resulting from smaller step sizes than 10 do not differ significantly from each other. This is exactly the default step size resulting from the standard simulation procedure. For larger step sizes the resulting vortex core line begins to deteriorate due to the noise introduced by the time derivative component of the acceleration vector. At step size of 20 we still get a similar but jagged result. At larger step sizes the extracted line no longer resembles the horseshoe vortex in the data set. At step size 100 the line breaks into 3 unconnected components that follow the vortex core line for some length and then trail off in random directions.

We conclude that for standard step sizes in well prepared simulations the time-aware vortex core line extraction method produces reliable results. Both for the especially designed data set and the real world examples (where the Jacobian had to be estimated) we did not find the estimation of the time derivatives to introduce significant additional noise.

6.6 Chapter Conclusions

This chapter proposes a new method to find vortex core lines in unsteady flows. Localization of vortices has been shown to be dependent on the temporal developments of the flow. We have given examples where vortex core extraction on time-frozen fields fails and have shown how to solve this problem. This result is not only relevant to vortex core extraction algorithms but to unsteady flow feature extraction methods in general. Since we could demonstrate that vortex core extraction algorithms have to include the temporal developments of the flow, it can be expected that similar results can be achieved for other flow features as well. Therefore we expect to see significant similar results in this direction in the future.

Based on the insight that it is necessary to include the time-derivative information into the feature extraction process we proposed a natural extension of the feature extraction process to unsteady flow data. By changing the underlying geometry from a streamline to a pathline based approach we can generalize existing feature extraction algorithms to unsteady flow data in a way that does not change their behavior on steady flows. We presented an algorithm that follows this approach extending parallel vectors operator criteria. Due to the consistent extension of the approach the algorithms change in a natural way and (given an implementation of the parallel vectors operator) the extension can be implemented quickly. The additional computation cost amounts to computing finite differences to estimate the time derivatives, therefore the difference to the original parallel vectors implementation is small.

We could confirm on real world data that the extracted vortices can differ significantly in position from the method of Sujudi and Haimes and in the large majority of the cases the extracted core lines are the same or better than those we got with the standard methods.

We conclude that for unsteady data the modified version of the algorithm of Sujudi and Haimes is the default choice. The higher order method generally performs very similar to the method of Sujudi and Haimes but it intensifies numerical issues. Also it requires additional computation. Therefore, only if after inspection of the data the results of the unsteady version of Sujudi and Haimes does not perform as expected, we suggest to switch to the modified higher order method.

Chapter 7

Summary

> ”The beginning of knowledge is the discovery of something we do not understand.” (Frank Herbert 1920 – 1986)

Feature-based visualization is a useful tool in understanding complex computational fluid dynamics (CFD) results. While we still do not have a generally accepted definition of what a vortex is, progress is being made.

We introduce several techniques to improve the vortex feature extraction results of classical vortex detectors. The presented framework is founded on an Integration interactive visual analysis and automatic vortex detection. In the remainder of this chapter we will summarize the contributions of this thesis. In the first section we summarize results on vortex core line extraction from time-dependent CFD data sets. In the second section we discuss how we can build non-local and non-binary detectors that give stable extraction results combining classical approaches and interactive analysis. In the first section we discuss how the formerly binary vortex detectors can be interpreted to produce smooth results that allow interactive specification of fuzzy thresholds during interactive analysis. In the second section we discuss how multiple criteria can be combined and how analysis using multiple detectors at once can improve the extraction results. In the third subsection we discuss an extension of the Eulerian detectors to the unsteady domain.

In the fourth subsection we continue work on unsteady flow field vortex feature extraction and discuss parallel vectors criteria to improve the extraction of line type vortex features on unsteady flow data.

7.1 Unsteady Criteria for Vortex Core Lines

Having unsteady region based vortex detectors available, we are interested in extracting the position of vortex core lines to improve the spatial perception of the structures in the visualization. The vortex core lines proof to be a very useful feature for understanding the volumetric nature and help orienting between multiple vortex regions. To extract meaningful vortex core lines, we need to extend existing methods for vortex core line extraction. The Sujudi-Haimes vortex core lines can be defined as the set of points where

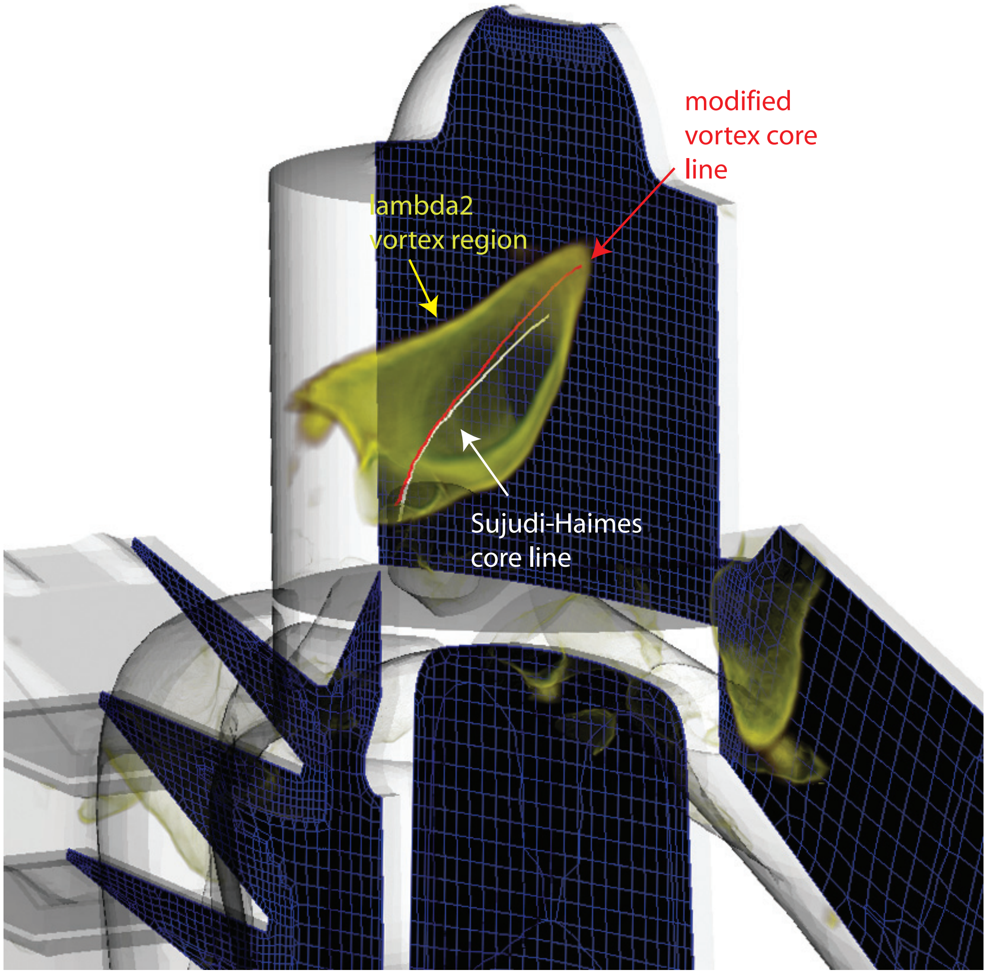


Figure 7.1: In this timestep of the two stroke data set, we can see that the vortex core line computed with the modified approach that considers the temporal developments we get a result that is in better agreement with the λ_2 vortex region. This image also highlights how the comprehensibility of the visualization improves when combining volumetric vortex region visualization with extracted vortex core line glyphs.

the velocity vector $\mathbf{u}$ and $\mathbf{a_s} := (\nabla \mathbf{u})\,\mathbf{u}$ are parallel, restricted to points where the velocity gradient has a pair of complex eigenvalues.

An modification to include the temporal developments is to use the true acceleration vector instead of instantaneous acceleration vector $\mathbf{a_s}$, i.e. to extract points where the Jerk vector and the velocity field $\mathbf{u}$ are parallel. The time-dependent acceleration (or Jerk) vector is defined as

$$\mathbf{a_t} = D\mathbf{u}/Dt = (\nabla \mathbf{u})\,\mathbf{u} + \partial \mathbf{u}/\partial t.$$

As we have seen in Chapter 6 this natural modification improves the robustness and produces vortex core line features at very reasonable positions (also see Figure 7.1).

7.2 Interactive Vortex Region Analysis

In Figure 7.2 we can see how we extend the binary classifiers throughout this thesis. In a first step we extend the classical definitions to compute non-binary vortex criteria that include a notion of strength or certainty of the result. In a second step we use this feature to combine multiple criteria and flow attributes into a combined detector interactively. The third step then extends the criteria using a physically motivated filter to the unsteady and non-local domain.

7.2.1 Step 1: Non-binary Criteria

Given two fuzzy attributes A and B on the domain D, the a fuzzy set can be defined as a mapping $A : D \rightarrow [0, 1]$, and the range between the values 0 (false) and 1 (true) allows to model partial membership to this attribute. We use linear scaling to map the detector responses to the $[0, 1]$ interval but use percentiles when scaling the detectors instead of minimum and maximum responses in the data set to improve the robustness against outliers. Let s_5 and s_{95} be the 5th and 95th percentile of a data set S, then a robust linear scaling is given by

$$scale_S(s(x)) := \begin{cases} 1 & , s_{95} \leq s(x) \\ s(x) - s_5/(s_{95} - s_5) & , s_5 < s(x) < s_{95} \\ 0 & , s(x) \leq s_5 \end{cases}$$

With this scaling function we can scale the detector outputs. Since a fixed threshold at zero is often not an appropriate requirement, we introduce an additional threshold $trsh$ for the non-binary vortex criteria λ_2, Q, δ and vorticity magnitude ($|\omega|$).

- λ_2: $E_{\lambda_2}(x, J(x), \mathbf{u}(x,t)) := scale_S(-\lambda_2(x) - trsh)$
- Q: $E_Q((x, J(x), \mathbf{u}(x,t)) := scale_S(Q - trsh)$

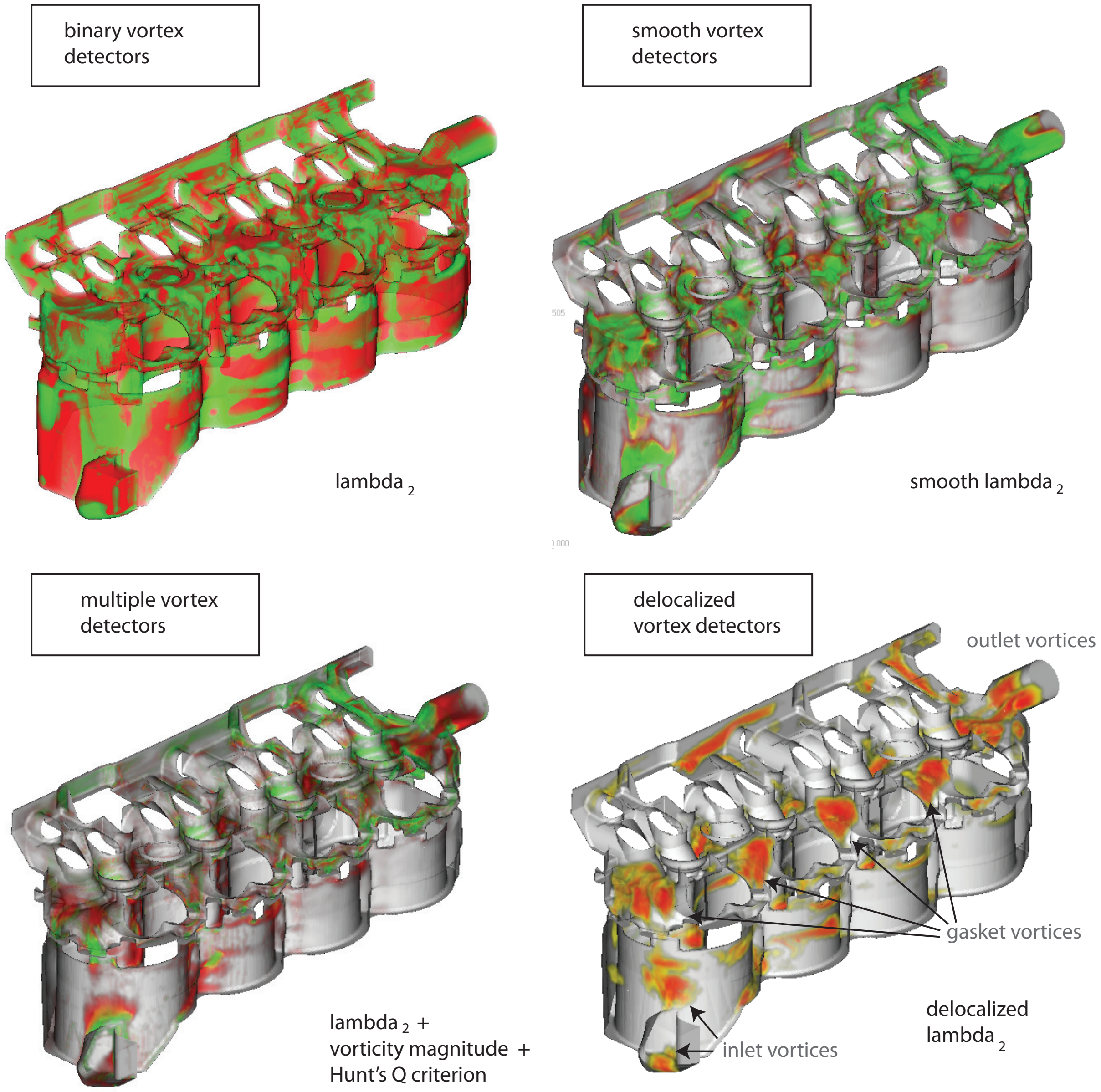


Figure 7.2: We illustrate the extensions discussed throughout this thesis. (1) We start with binary vortex detectors. It is hard to understand where the structures originate and how they are related to the geometry. (2) Using non-binary vortex detectors we can select only the parts of the volume that contain strongly vortical movements. (3) By combining multiple criteria it is possible to refine the selection further and it becomes obvious that the strongest source of vorticity are the gaskets between cylinder head and body. (4) Using delocalization we can further refine the selection and remove noise from the results.

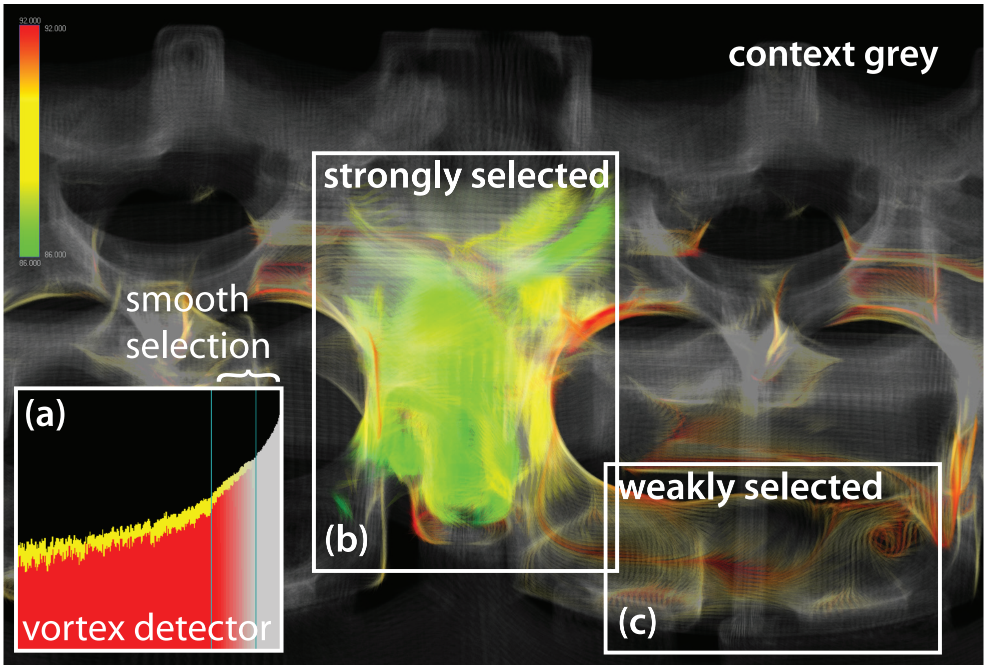


Figure 7.3: Non-binary integration of vortex detectors allows to perform fuzzy set based interaction and visualization. (a) we have selected values strongly indicating a vortex to be present fully and use a smoothly decreasing selection for borderline cases. (b) The main feature is selected with high opacity values. (c) Weaker features as the thin regions on the boundary of the geometry of the cooling jacket are rendered with high transparency. Non-selected regions are displayed in grey values and high transparency to display context.

- δ-criterion: $E_{\Delta}((x, J(x), \mathbf{u}(x,t)) := scale_S(\lambda_{ci} - trsh)$
- vorticity magnitude: $E_{\omega}((x, J(x), \mathbf{u}(x,t)) := scale_S(|\omega| - trsh)$

Using non-binary criteria allows to specify smooth selections as well and get visual feedback on the strength of the selected vortex.

7.2.2 Step 2: Multiple combined Criteria

In the second step we have a set of multiple vortex detection criteria available and want to combine these using interactive procedures. For combining two criteria we rely on standard fuzzy norm equivalents of the boolean operators.

- $'A \quad AND \quad B' := min(A, B)$
- $'A \quad OR \quad B' := max(A, B)$
- $'NOT \quad A' := 1 - A$

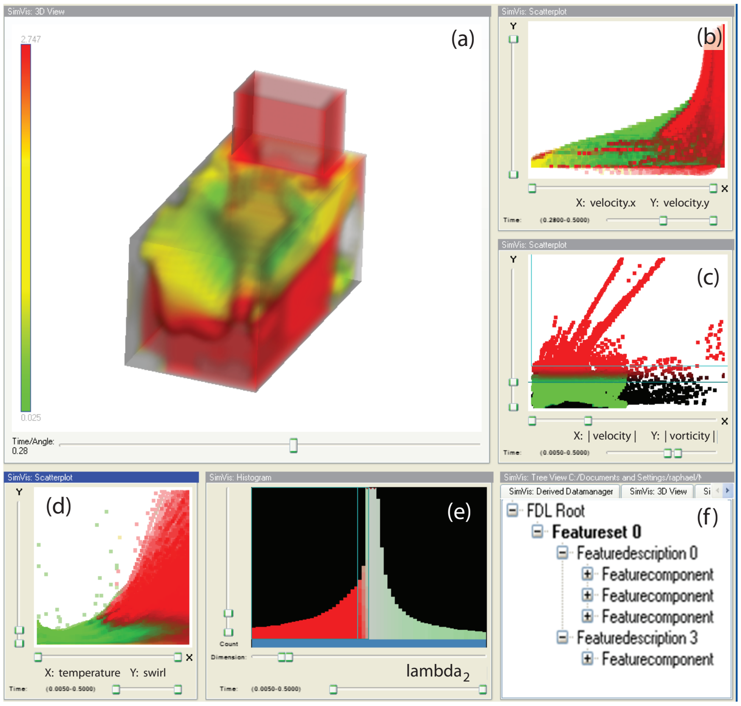


Figure 7.4: Combination and analysis of multiple detectors. We see the views linked following the expression '((b) AND (c)) OR (e)'. (a) The selected subset of the simulation is rendered with temperature mapped to color. (b) We have selected high velocities in $z - direction$ and see the distribution of $x-$ and $y - velocities$. (c) In this scatterplot data items with high vorticity are selected. (d) We can compare the relation of strongly swirling motion with the selection of $\lambda_2 < 0$ in (e). (f) The FDL-tree controls how the selection form the different views are linked together.

We can now structure features specified interactively in multiple views by placing them inside an FDL-tree [25].

In Figure 7.4 we can see an example how multiple vortex detectors and other flow attributes can be combined. The inter-view linking scheme allows to inspect the relations between multiple attributes in comparison to each other and in relation to one or multiple vortex detectors.

7.2.3 Step 3: Delocalized Criteria

Until now we have discussed how multiple criteria can be combined locally to improve feature detection. This can be considered as the Eulerian approach to feature detection. In the third step we accumulate non-binary feature criteria along trajectories. The idea is, that a particle that belongs to a vortex will travel a certain amount of time through a vortex, if the classification was good.

A pathline is defined as

$$p(t+\Delta t) = p(t) + \int_{t}^{t+\Delta t} \mathbf{u}(p(s), s)ds$$

where $p(t)$ is the position of the particle at time t, $p(t+\Delta t)$ is the new position after time Δt and $\mathbf{u}(p(t), t)$ is the velocity of the particle at position $p(t)$ at time t. The Runge-Kutta method (RK4) can be used for numeric integration of these pathlines [116]. Given a vector field $\mathbf{u}$ we call p_{t_0,x_0} the pathline starting at point x_0 and time t_0.
Following Chapter 6 we can define the Eulerian detector response for a pathline p_{t_0,x_0} as

$$E(p_{t_0,x_0}, t) = E(p_{t_0,x_0}(t), J(p_{t_0,x_0}(t), t), \mathbf{u}(p_{t_0,x_0}(t), t))$$

if $p_{t_0,x_0}(t) \in D$.

The two maximal integration length parameters $\bar{t}_f$ and $\bar{t}_b$ for a pathline represent the maximal time the particle remains inside the simulation domain D during forward (resp. backward) integration. The user can interactively specify integration length parameters t_b and t_f to select integration length in backward and forward time.

Given t_b and t_f, the delocalized version of the Eulerian detector at position x_0 and time t_0 is

$$\widetilde{E}(p_{t_0,x_0}, t_b, t_f) = \frac{\int_{\max(\bar{t}_b, t_b)}^{\min(t_f, \bar{t}_f)} w(s) \cdot E(p_{t_0,x_0}, s)ds}{\int_{t_b}^{t_f} w(s)ds} \tag{7.1}$$

with $t_b < t_0 < t_f$ and $w(x)$ a weighting function. We suggest to use linear or gaussian weighting of detector values along the pathline.

Using several real-world data sets in Chapter 5 we have shown that the delocalized feature detectors outperform their local counterparts regarding the separation of vortices. Furthermore the delocalization process improves the robustness of the detectors in the presence of noise, and using the interactive line view it becomes possible to specify these parameters non-uniformly. This improves the flexibility of the detection process and allows to concentrate on specific types of vortices, e.g., the largest or those which trap particles inside and prevent mixing.

7.3 Chapter Conclusions

In this chapter we have given an overview of the approaches presented in this thesis. The extension of vortex core line detectors to unsteady flow fields can improve the detection results, given the temporal resolution of the data is sufficiently high. We have also shown how the different extensions for interactive vortex analysis are mutually dependent on each other and contribute to the final delocalized feature detectors: combining multiple vortex detectors benefits from the non-binary form of the detectors and the delocalized detectors take much of their robustness from integration over good local selections.

Chapter 8

Thesis Conclusions

> "All proofs inevitably lead to propositions which have no proof – all things are known because we want to believe in them." (Frank Herbert 1920 – 1986)

In this thesis we have presented a set of approaches to extend classical vortex feature detection and to include the detected information into the process of visual interactive analysis. The presented framework is target at engineers and researchers that have to deal with complex simulation results. As we have seen in various case studies throughout this thesis, CFD simulation results very often include surprising features, which the engineer could not have foreseen. In other words, due to the non-linear nature of the flow, unforeseeable effects can influence the performance of a prototype simulation. When the engineer has to perform a 'debugging'-type of analysis for a prototype to figure out why the results are not as expected (e.g., a cooling jacket does not cool well enough), then the interactive analysis process including vortex features is an invaluable tool.

The field of feature based visualization develops in co-evolution with basic fluid dynamics research. As new theories on coherent structures in fluid flows are developed by fluid dynamicists, new visualizations based on these concepts become possible. One example is the transition from the Eulerian to the Lagrangian perspective regarding feature extraction. New visualization tools help engineers and researches to expand their understanding of the governing laws of turbulence, thus leading to new results in fluid dynamics. With the goal in mind to understand the underlying laws of coherent structures in turbulence, visualization can play an important role in assisting both researchers and engineers.

We have discussed that none of the vortex feature detectors is generally accepted and that there are examples where all of them fail to produce optimal results. By combining multiple detectors and taking the development over time into account we can balance many of these problems. We conclude that feature based visualization is a powerful tool for understanding simulation results. Combining interactive analysis and automatic extraction of relevant features allows the user to apply high level knowledge and experience regarding the effects of vortices into the analysis process.

For future work, many questions remain open. Here we summarize some questions which have come up during the work on this thesis:

- How can we combine and relate vortex type features with topological features? Multiple results point in this direction, but little has been done in the visualization community in this direction.
- How can we use the extracted information to produce illustrative visualization of vortex type structures in the flow? There are multiple points to attack this question, but there have been no groundbreaking successes yet.
- How can we bring simulation and visualization tools closer together, in so far as the information and understanding generated during the analysis process can be fed back to the simulation to improve the simulation results as well?

We can conclude that vortex extraction is a key tool for understanding complex fluid dynamics results and that interactive visual analysis benefits from including complex derived features.

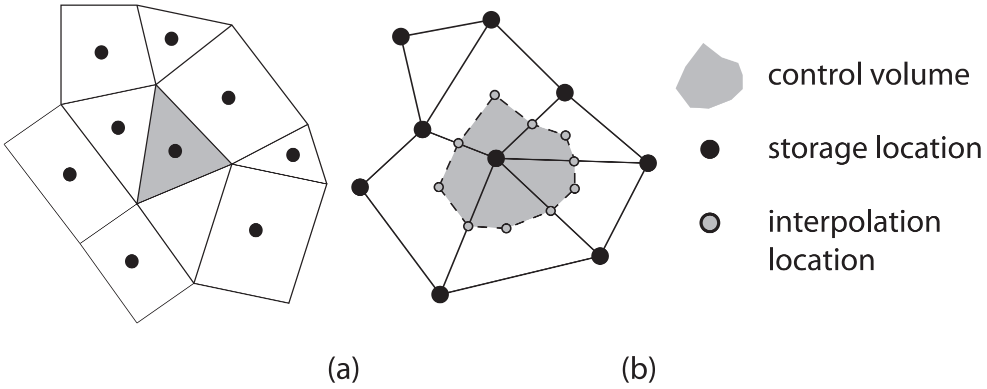


Figure A.1: Control volume variants used for numerical solution for CFD. (a) Cell-centered volume representation. (b) Vertex-centered volume representation. The segments surround the median dual control volume, i.e., the positions inside the cells are computed using the center of gravity for each cell.

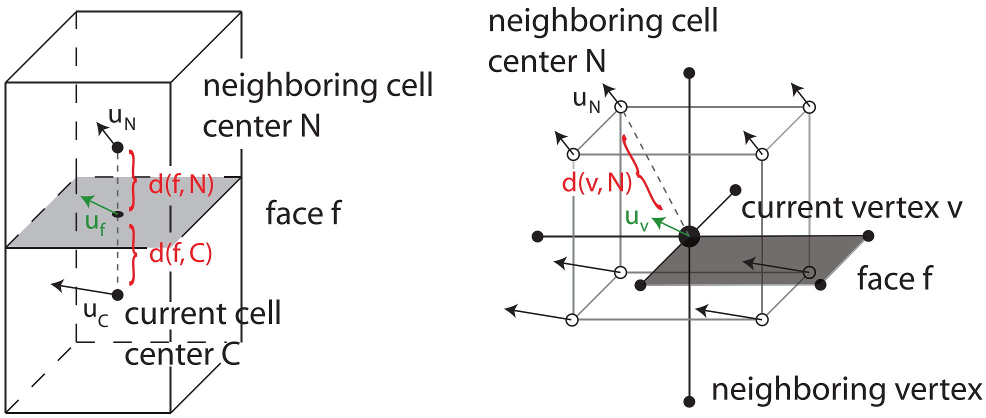
Appendix A

Gradients

The vortex core line extraction process consists of three stages:

1. estimate velocities at vertices and faces (Subsection A.1)
2. reconstruct gradients at vertices and faces using estimated velocities (Subsection A.2)
3. for each cell subdivide into tetrahedra and use reconstructed gradients to find vortex core positions (Subsection 6.3.4).

Depending on the type of simulation data storage can be either vertex or cell centered (see Figure A.1). In the third vortex core line extraction step we need gradients at the nodes of the grid, and the gradient reconstruction step varies slightly for the two storage types.



(a) (b)

Figure A.2: Velocity estimation for different mesh types. (a) Estimating face velocities from cell centered data can be done by inverse distance weighting of the adjacent cell velocities. (b) Estimating vertex velocities from cell centered data can be done by inverse distance weighting of the surrounding cell velocities

A.1 Velocity Estimation

To reconstruct velocities we use a standard inverse geometric weighted interpolation scheme.

For estimating face velocities from cell centers, we define the distance between the center of a cell and one of its faces as the distance between the cell center and the center of gravity of the face. The velocity at face u_f is computed as

$$u_f := \alpha u_C + (1 - \alpha) u_N$$

where C and N are the two cells adjacent to the face f. Here the weighting geometric factor α can be computed as $\alpha := \frac{d(f,N)}{d(f,C)+d(f,N)}$, where $d(\cdot,\cdot)$ denotes the Euclidean distance (see Figure A.2 (a)). When working with vertex-centered volume representation, the velocity at a face can be computed by taking the average of the surrounding vertices.

The velocity at a node v can be computed from the surrounding cell centers by using the cell values of the surrounding cells. Again the weight is taken as the inverse of the distance of the node from the cell center. Let N_C be the number of cells surrounding v, C_i the center of the i-th neighboring cell, and u_{C_i} its velocity vector. Then we can compute the velocity u_v at v as $u_v := \sum_{i=0}^{N_C} u_{C_i} \alpha_i$ where the weight of the i-th cell is $\alpha_i := d(v, C_i)^{-1}$ (see Figure A.2 (b)). The velocity at a cell center from the surrounding vertices for vertex centered grids can be computed by taking the inverse distance weighted average of vertices of the cell.

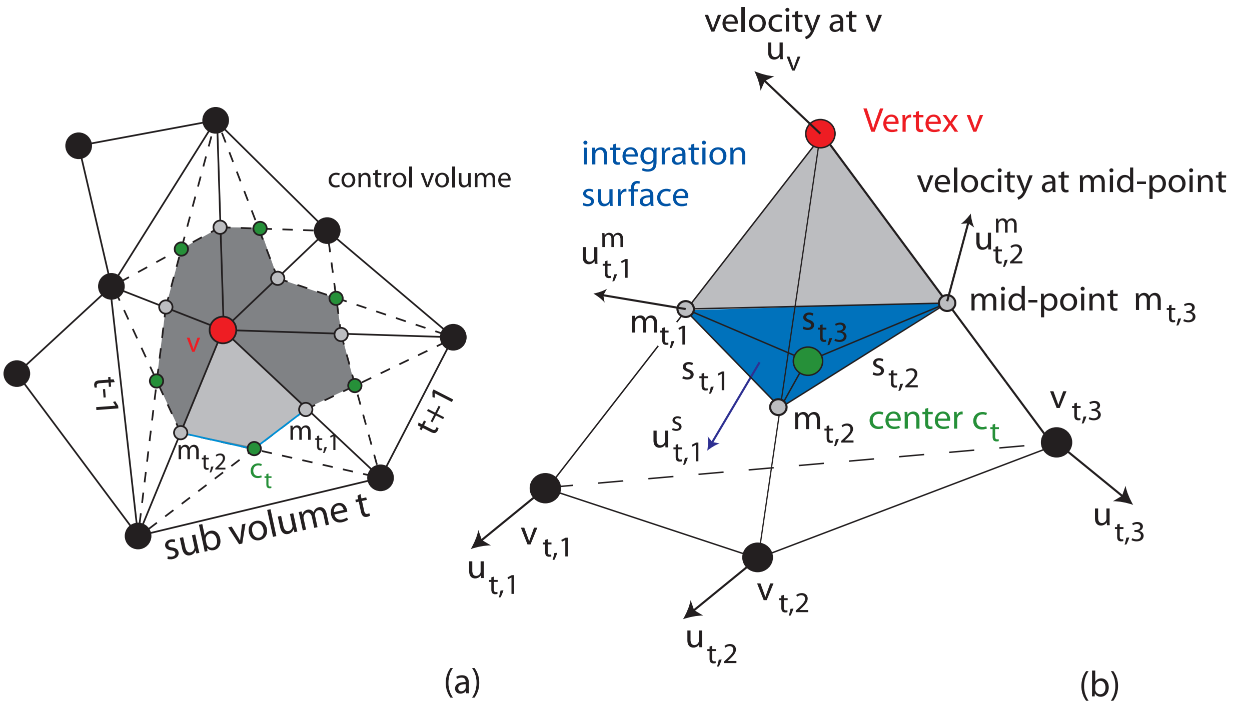


Figure A.3: Gradient estimation at a vertex (red) using the Green-Gauss theorem requires to estimate cell center velocities and mid-point velocities. (a) The surrounding surface uses cell centers (green) and mid-points (gray). (b) In this detail illustration of the lighter gray section from (a) we see the full configuration for a single surrounding tetrahedron.

A.2 Gradient Reconstruction

In addition to the velocity values at the vertices to the cell we also need the velocity gradients. To compute the gradients we suggest to use the Green-Gauss reconstruction method which works with velocity values from the faces of the cell. The least-squares linear reconstruction method can be used when no connectivity values are present (for example when working with point clouds).

A.2.1 Green-Gauss Linear Reconstruction

Let Ω be a volume (a cell of the mesh for cell centered representation or the median control volume for vertex centered representations), $S = \partial\Omega$ the bounding surface of Ω, φ some scalar function defined on Ω, and $\nabla\varphi$ the derivative of φ. Then the Green-Gauss theorem states that the surface integral of the scalar function φ times the normal vector of the surface over the surface S is equal to the volume integral of the gradient $\nabla\varphi$ over the volume Ω:

$$\int_\Omega \nabla\varphi d\Omega = \int_S \varphi \mathbf{n} dS.$$

To compute the derivative at the center of the control volume we assume that $\nabla\varphi$ is constant over the control volume and the volume integral over $\nabla\varphi$ reduces to the volume of Ω times ∇u:

$$|\Omega|\nabla\varphi \approx \int_\Omega \nabla\varphi d\Omega = \int_S \varphi \mathbf{n} dS.$$

Finally, we can approximate the integral over the bounding surface using face values. That is

$$\nabla\varphi = \frac{1}{|\Omega|}\sum_{faces} \varphi \cdot area(face_i) \cdot n_f$$

where $area(triangle)$ is the surface area of a triangle.

To compute the derivative at a vertex we can use the control volume depicted in Figure A.3 and get

$$|\Omega_C|\nabla u \approx \sum_{t=0}^{N_t(v)} \sum_{i=1}^{3} area(s_{t,i}) \cdot u^s_{t,i}.$$

Here $N_t(v)$ is the number of tetrahedra at vertex v. Here we are using an interpolated velocity vector at the mid-points $u^m_{t,i} := \frac{1}{2}(u_v + u_{t,i})$ and the velocity at the cell center to construct the surface velocity $u^s_{t,i} := \frac{1}{3}(u^m_{t,i} + u^m_{t,i+1} + u_c)$ (with $u^m_{t,4} := u^m_{t,1}$). See Figure A.3 for an illustration.

A.2.2 Least-Squares Linear Reconstruction

Here the gradient is estimated by fitting a hyperplane to the cell such that the difference between the extrapolated value for the surrounding cells and the present values of the surrounding cells are minimized.

For each edge of the resulting mesh incident to the vertex v_0, an edge projected gradient constraint equation is constructed using inverse distance weights α_i for each edge:

$$\alpha_k(\nabla u) \cdot (x_k - x_0) = \alpha_k(\varphi_k - \varphi_0).$$

The gradient construction is obtained by solving a least-squares optimization problem to minimize the sum of the distances between the estimated values and the vertex values. This approach implicitly smoothes the data and can improve the results when working with noisy data.

Which weighting scheme works best is still an open question. Mavriplis [95] stresses that the minimization problem will be much better conditioned when using inverse distance weighting. On the other hand when the mesh is irregularly sampled and on one side of a cell we have a large number of small triangles and on the other side just a few larger triangles this can lead to a gross misrepresentation of small triangles. Therefore we use unweighted direct neighbors for estimating the gradient at a cell by default and only change this procedure when necessary.

Acknowledgments

The author thanks Ronny Peikert, Meister Eduard Gröller, and Helwig Hauser for their input and discussions and for supervising this thesis. Also I would like to thank Filip Sadlo for many exiting discussions. I would like to thank Philipp Muigg and Helmut Doleisch for collaboration and help with the SimVis framework from which this thesis has greatly benefited.

This work has been funded by the FWF PVG project supported by the Austrian Science Fund (FWF) under grant no. P18547.

The cooling jacket CFD simulation data set is courtesy of AVL List GmbH, Graz, Austria. The 2-stroke engine data set is courtesy of the ”Institut für Verbrennungskraftmaschinen und Thermodynamik” at the Technical University of Graz. The t-junction data set has been prepared by Markus Trenker from Arsenal Research.

I thank Helene for her constant encouragement and support.

Curriculum Vitae

Dipl. Inf. Raphael Fuchs,
born on 8 November 1979, in Kehl, Germany.

Education

1986 – 1999 Gymnasium zum Heiligen Grab, Baden-Baden, Germany

1999 Civil Services (care for severely handicapped people)

2000 Ruprecht-Karls-Universität, Heidelberg, Germany, Computer Linguistics

2001 – 2006 Philipps-Universität, Marburg, Germany, Computer Sciences, minor: Psychology

2003 Vordiplomsprüfung (sim. Bachelor) Computer Sciences and Vordiplomsprüfung (sim. Bachelor) Mathematics

2004 6 Months student exchange to University of Virginia, USA (Department of Mathematics)

2005 Visiting student at University Erlangen, Germany in the Department of Pattern Recognition

2006 PhD student at the Vienna University of Technology

2007 Visiting student at ETH Zurich, Switzerland in the Department of Computer Graphics

Professional Activities

2002 Extension of the online-student-management system of the Computer Science Institute

2003 – 2006 Teaching assistant for lectures Logic, Theoretical Computer Sciences and Discrete Mathematics

2004 Internship at Dade Behring: Implementation of a distributed system for management of cell hybrids

since 2006 Researcher at VRVis Research Center for Virtual Reality and Visualization

Reviewing

2006 Two reviews for 2nd int. symp. on visual computing

2007 Two reviews for APVIS 08

2007 One review for Eurographics 08

2007 Three reviews for WSCG 08

2007 One review for Computers & Graphics Journal

2008 Two reviews for IEEE Visualization 08

2008 Three reviews for Computers & Graphics Journal

Student Supervision

2007 Two Seminar Theses on ’fluids in computer graphics’

2007 Internship from Damascus University

2008 One Seminar Thesis on ’volumetric shadows’

2008 Two Masters Theses in the field of scientific visualization (one recieved the Prof. Tosiyasu Laurence Kunii Award)

2008 Two student presentations at WSCG 08

Publications and Talks:

Please see the **Related Publications** and **Bibliography** sections of this thesis for the relevant publications related to the contents of this thesis.

Bibliography

[1] Web page for this thesis. See URL `http://www.cg.tuwien.ac.at/research/publications/2008/fuchs-thesis/`.

[2] J. Anderson. *Computational Fluid Dynamics: The Basics with Applications*. McGraw Hill, 1995.

[3] C. L. Bajaj, V. Pascucci, G. Rabbiolo, and D. Schikorc. Hypervolume visualization: a challenge in simplicity. In *VVS '98: Proc. of the 1998 IEEE symposium on Volume visualization*, pages 95–102, 1998.

[4] David C. Banks and Bart A. Singer. A predictor-corrector technique for visualizing unsteady flow. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):151–163, 1995.

[5] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1994.

[6] D. Bauer and R. Peikert. Vortex tracking in scale-space. In *Proceedings of the 4th Joint IEEE VGTC - EUROGRAPHICS Symposium on Visualization (VisSym 2002)*, pages 140–147, 2002.

[7] C. H. Berdahl and D. S. Thompson. Eduction of swirling structure using the velocity gradient tensor. *AIAA Journal*, 31(1):97–103, 1993.

[8] Jorik Blaas, Charl P. Botha, and Frits H. Post. Interactive visualization of multi-field medical data using linked physical and feature-space views. In *Proc. of EuroVis07*, pages 123–130, 2007.

[9] R. A. Brown, B. A. Flikinger, E. Forren, E. Schultz, D. M. Sirmans, and D. Spencer. Improved detection of severe storms using experimental fine-resolution wsr-88d measurements. *Weather and Forecasting*, 20(1):3–14, 2005.

[10] S. Bruckner, S. Grimm, A. Kanitsar, and Meister E. Gröller. Illustrative context-preserving volume rendering. In *Proceedings of the 7th Joint IEEE VGTC - EUROGRAPHICS Symposium on Visualization (VisSym 2005)*, pages 69–76, 2005.

[11] S. Bruckner and Meister E. Gröller. Volumeshop: An interactive system for direct volume illustration. In *Proc. of IEEE Visualization 2005*, pages 671–678, 2005.

[12] Raphael Bürger, Philipp Muigg, Martin Ilcìk, Helmut Doleisch, and Helwig Hauser. Integrating local feature detectors in the interactive visual analysis of flow simulation data. In *Proceedings of the 9th Joint IEEE VGTC - EUROGRAPHICS Symposium on Visualization (VisSym 2007)*, pages 171–178, 2007.

[13] W. Cai and G. Sakas. Data intermixing and multi-volume rendering. *Comput. Graph. Forum*, 18(3):359–368, 1999.

[14] P. Chakraborty, S. Balachandarand, and R. J. Adrian. On the relationships between local vortex identification schemes. *Journal of Fluid Mechanics*, 535:189–214, 2005.

[15] M. S. Chong, A. E. Perry, and B. J. Cantwell. A general classification of three-dimensional flow fields. *Physics of Fluids Archive*, 2:765–777, 1990.

[16] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms. *Computers & Graphics*, 22(1):37–54, 1998.

[17] R. A. Crawfis and M. J. Allison. A scientific visualization synthesizer. In *Proceedings IEEE Visualization 1991*, pages 262–267, 1991.

[18] R. A. Crawfis and N. Max. Texture Splats for 3D Scalar and Vector Field Visualization. In *Proceedings IEEE Visualization 1993*, pages 261–267, 1993.

[19] R. A. Crawfis, H.-W. Shen, and N. Max. Flow visualization techniques for CFD using volume rendering. 9th. International Symposium on Flow Visualization, 2000.

[20] R. Cucitore, M. Quadrio, and A. Baron. On the effectiveness and limitations of local criteria for the identification of a vortex. *European Journal of Mechanics*, 18:261–282, 1999.

[21] P. A. Davidson. *Turbulence: An Introduction for Scientists and Engineers.* Oxford University Press, 2004.

[22] T. Delmarcelle and L. Hesselink. Visualizing second-order tensor fields with hyperstreamlines. *IEEE Comput. Graph. Appl.*, 13(4):25–33, 1993.

[23] B. Dennis, S. Kocherlakota, A. Sawant, L. Tateosian, and C. G. Healey. Designing a visualization framework for multidimensional data. *IEEE Comput. Graph. Appl.*, 25(6):10–15, 2005.

[24] S. Djurcilov, K. Kim, P. F. J. Lermusiaux, and A. Pang. Volume rendering data with uncertainty information. In *Data Visualization (proceedings of the EG+IEEE VisSym)*, pages 243–52, 2001.

[25] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of the 5th Joint IEEE VGTC - EUROGRAPHICS Symposium on Visualization (VisSym 2003)*, pages 239 – 248, 2003.

[26] H. Doleisch, M. Mayer, M. Gasser, P. Priesching, and H. Hauser. Interactive feature specification for simulation data on time-varying grids. In *Proc. of the Conference Simulation and Visualization (SimVis 2005)*, pages 291–304, 2005.

[27] D. Ebert and P. Rheingans. Volume illustration: non-photorealistic rendering of volume models. In *Proceedings IEEE Visualization 2000*, pages 195–202, Los Alamitos, CA, USA, 2000. IEEE Computer Society Press.

[28] D. S. Ebert, C. J. Morris, P. Rheingans, and T. S. Yoo. Designing effective transfer functions for volume rendering from photographic volumes. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):183–197, 2002.

[29] C. P. Ellington, C. v. d. Berg, A. P. Willmott, and A. L. R. Thomas. Leading-edge vortices in insect flight. *Nature*, 384:626 – 630, 1996.

[30] Klaus Engel, Markus Hadwiger, Joe M. Kniss, Christof Rezk-Salama, and Daniel Weiskopf. *Real-Time Volume Graphics.* A K Peters Ltd., 2006.

[31] G. Erlebacher, B. Jobard, and D. Weiskopf. Flow Textures: High-Resolution Flow Visualization. In *The Visualization Handbook*, pages 279–293. Elsevier, 2005.

[32] M. Escudier. Confined vortices in flow machinery. *Annual Review of Fluid Mechanics*, 19:27–52, 1987.

[33] S. Feiner and D. D. Seligmann. Cutaways and ghosting: satisfying visibility constraints in dynamic 3d illustrations. *The Visual Computer*, 8(5&6):292–302, 1992.

[34] S. K. Feiner and C. Beshers. Worlds within worlds: metaphors for exploring n-dimensional virtual worlds. In *UIST '90: Proc. of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology*, pages 76–83, 1990.

[35] M. Ferré, A. Puig, and D. Tost. A framework for fusion methods and rendering techniques of multimodal volume data. *Computer Animation and Virtual Worlds*, 15:63–77, 2004.

[36] C. Garth, F. Gerhardt, X. Tricoche, and H. Hagen. Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1464–1471, 2007.

[37] C. Garth, G.-S. Li, X. Tricoche, C. D. Hansen, and H. Hagen. Visualization of coherent structures in transient flows. In *Proceedings of TopoInVis 2007: Topology-Based Methods in Visualization*, 2007. (to appear at Springer).

[38] Christoph Garth, Xavier Tricoche, and Gerik Scheuermann. Tracking of vector field singularities in unstructured 3d time-dependent datasets. In *Proceedings IEEE Visualization 2004*, pages 329–336, 2004.

[39] A. Ghosh, P. Prabhu, A. E. Kaufman, and K. Mueller. Hardware assisted multichannel volume rendering. In *Computer Graphics International*, pages 2–7, 2003.

[40] A. Gooch, B. Gooch, P. Shirley, and E. Cohen. A non-photorealistic lighting model for automatic technical illustration. In *SIGGRAPH '98*, pages 447–452, 1998.

[41] M. A. Green, C. W. Rowley, and G. Haller. Detection of Lagragian coherent structures in 3D turbulence. *Journal of Fluid Mechanics*, 572:111–120, 2007.

[42] C. P. Gribble and S. G. Parker. Enhancing interactive particle visualization with advanced shading models. In *APGV '06: Proc. of the 3rd symposium on Applied perception in graphics and visualization*, pages 111–118, 2006.

[43] S. Grimm, S. Bruckner, A. Kanitsar, and Meister E. Gröller. V-objects: Flexible direct multi-volume rendering in interactive scenes. Technical Report TR-186-2-04-06, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2004.

[44] M. Hadwiger, C. Berger, and H. Hauser. High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In *Proceedings IEEE Visualization 2003*, page 40, 2003.

[45] R. Haimes and D. Kenwright. On the velocity gradient tensor and fluid feature extraction. In *Proc. of AIAA 14th Computational Fluid Dynamics Conference*, 1999.

[46] G. Haller. An objective definition of a vortex. *Journal of Fluid Mechanics*, 525:1–26, 2005.

[47] Y.M.A. Hashash, J.I.C. Yao, and D. Wotring. Glyph and hypersteamline representation of stress and strain tensors and material constitutive response. *International Journal for Numerical and Analytical Methods in Geomechanics*, 27:603–626, 2003.

[48] H. Hauser. Visual analysis of differential information. In *Proc. of the International Conference of Applied Mathematics*, 2006.

[49] H. Hauser and Meister E. Gröller. Thorough insights by enhanced visualization of flow topology. In *9th International Symposium on Flow Visualization*, 2000.

[50] H. Hauser, R. Laramee, and H. Doleisch. Topology-based versus feature-based flow analysis - challenges and an application. In *Proceedings of TopoInVis 2007: Topology-Based Methods in Visualization*, 2006. (to appear at Springer).

[51] H. Hauser, L. Mroz, G.-I. Bischi, and Meister E. Gröller. Two-level volume rendering - fusing MIP and DVR. In *Proceedings IEEE Visualization 2000*, pages 211–218, 2000.

[52] J. L. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Comput. Graph. Appl.*, 11(3):36–46, 1991.

[53] L. Hesselink, F. H. Post, and J. J. van Wijk. Research issues in vector and tensor field visualization. *IEEE Computer Graphics and Applications*, 14(2):76–79, March 1994.

[54] D. H. House, A. Bair, and C. Ware. On the optimization of visualizations of complex phenomena. In *Proceedings IEEE Visualization 2005*, page 12, 2005.

[55] B. Hua, J.C. McWilliams, and P. Klein. Lagrangian accelerations in geostrophic turbulence. *Journal of Fluid Mechanics*, 366:87–108, 1998.

[56] J. C. R. Hunt, A. A. Wray, and P. Moin. Eddies, stream and convergence zones in turbulent flows. Technical report, Center for Turbulence Research Report CTR-S88, 1988.

[57] F. Hussain. Coherent structures - reality and myth. *Physics of Fluids*, 26:2816–2850, 1983.

[58] V. Interrante. Illustrating surface shape in volume data via principal direction-driven 3d line integral convolution. In *SIGGRAPH '97: Proc. of the 24th annual conference on Computer graphics and interactive techniques*, pages 109–116, 1997.

[59] V. Interrante, H. Fuchs, and S. M. Pizer. Conveying the 3D shape of smoothly curving transparent surfaces via texture. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):98–117, 1997.

[60] V. Interrante and C. Grosch. Strategies for effectively visualizing 3d flow with volume lic. In *Proceedings IEEE Visualization 1997*, pages 421–424, 1997.

[61] Monika Jankun-Kelly, Ming Jiang, David Thompson, and Raghu Machiraju. Vortex visualization for practical engineering applications. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):957–964, 2006.

[62] J. Jeong and F. Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–84, 1995.

[63] R. Jesse and T. Isenberg. Use of hybrid rendering styles for presentation. In *WSCG*, pages 57–60, 2003.

[64] Ming Jiang, Raghu Machiraju, and David Thompson. Geometric verification of swirling features in flow fields. In *Proceedings IEEE Visualization 2002*, pages 307–314, 2002.

[65] T.A. Keahey. Visualization of high-dimensional clusters using nonlinear magnification. In *Proc. SPIE Visual Data Exploration and Analysis VI*, pages 236–243, 1999.

[66] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[67] G. Kindlmann. Superquadric tensor glyphs. In *Proceeding of The Joint Eurographics - IEEE TCVG Symposium on Visualization*, pages 147–154, 2004.

[68] G. Kindlmann, D. Weinstein, and D. Hart. Strategies for direct volume rendering of diffusion tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):124–138, 2000.

[69] Gordon Kindlmann and Carl-Fredrik Westin. Diffusion tensor visualization with glyph packing. *Proceedings IEEE Visualization 2006*, 12(5), September-October 2006.

[70] M. Kirby, D. Keefe, and D. H. Laidlaw. Painting and visualization. In *Visualization Handbook*. Academic Press, June 2004.

[71] R. M. Kirby, H. Marmanis, and David H. Laidlaw. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In *Proceedings IEEE Visualization 1999*, pages 333–340, 1999.

[72] E. P. Klement, R. Mesiar, and E. Pap. *Triangular Norms, volume 8 of Trends in Logic*. Kluwer Academic Publishers, 2000.

[73] D. Knight and G. Mallinson. Visualizing unstructured flow data using dual stream functions. *IEEE Transactions on Visualization and Computer Graphics*, 2(4):355–363, 1996.

[74] J. Kniss, C. Hansen, M. Grenier, and T. Robinson. Volume rendering multivariate data to visualize meteorological simulations: a case study. In *Proceedings of the 4th Joint IEEE VGTC - EUROGRAPHICS Symposium on Visualization (VisSym 2002)*, pages 189–194, 2002.

[75] J. Kniss, G. Kindlmann, and C. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings IEEE Visualization 2001*, pages 255–262, 2001.

[76] J. Kniss, S. Premoze, M. Ikits, A. Lefohn, C. Hansen, and E. Praun. Gaussian transfer functions for multi-field volume visualization. In *Proceedings IEEE Visualization 2003*, page 65, 2003.

[77] K. Kreeger and A. Kaufman. Mixing translucent polygons with volumes. In *Proceedings IEEE Visualization 1999*, pages 191–198, 1999.

[78] R. D. Kriz, E. H. Glaessgen, and J. D. MacRae. Eigenvalue-eigenvector glyphs: Visualizing zeroth, second, fourth and higher order tensors in a continuum. In *Workshop on Modeling the Development of Residual Stresses During Thermoset Composite Curing*, 1995.

[79] J. Kruger, P. Kipfer, P. Kondratieva, and R. Westermann. A particle system for interactive visualization of 3d flows. *IEEE Transactions on Visualization and Computer Graphics*, 11(6):744–756, 2005.

[80] D. H. Laidlaw, E. T. Ahrens, D. Kremers, M. J. Avalos, R. E. Jacobs, and C. Readhead. Visualizing diffusion tensor images of the mouse spinal cord. In *Proceedings IEEE Visualization 1998*, pages 127–134, 1998.

[81] L. Lam. Classifier combinations: Implementations and theoretical issues. In *MCS '00: Proc. Workshop on Multiple Classifier Systems*, pages 77–86, 2000.

[82] G. Lapeyre. Characterization of finite-time Lyapunov exponents and vectors in two-dimensional turbulence. *Chaois*, 12:688–698, 2002.

[83] R. Laramee, B. Jobard., and H. Hauser. Image Space Based Visualization of Unsteady Flow on Surfaces. In *Proceedings IEEE Visualization 2003*, pages 131–138, 2003.

[84] R. S. Laramee, C. Garth, J. Schneider, and H. Hauser. Texture advection on stream surfaces: A novel hybrid visualization applied to CFD simulation results. In *Proc. of the Joint EUROGRAPHICS - IEEE VGTC Symposium on Visualization (EuroVis 2006)*, pages 155–162, 2006.

[85] R. S. Laramee, J. J. van Wijk, B. Jobard, and H. Hauser. Isa and ibfvs: Image space-based visualization of flow on surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):637–648, 2004.

[86] R. S. Laramee, D. Weiskopf, and J. Schneider. Investigating swirl and tumble flow with a comparison of visualization techniques. In *Proceedings IEEE Visualization 2004*, 2004.

[87] Robert S. Laramee and Helwig Hauser. Geometric flow visualization techniques for cfd simulation data. In *SCCG '05: Proc. of the 21st spring conference on Computer graphics*, pages 221–224, 2005.

[88] R.S. Laramee, H. Hauser, H. Doleisch, B. H. Vrolijk, F. Post, and D. Weiskopf. The state of the art in flow visualization: Dense and texture-based techniques. *Computer Graphics Forum*, 23(2):143–161, 2004.

[89] J. P. Lee, D. Carr, G. Grinstein, J. Kinney, and J. Saffer. The next frontier for bio- and cheminformatics visualization. *IEEE Comput. Graph. Appl.*, 22(5):6–11, 2002.

[90] Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

[91] Y. Levy, D. Degani, and A. Seginer. Graphical visualization of vortical flows by means of helicity. *AIAA Journal*, 28:1347–1352, 1990.

[92] H. J. Lugt. *Wirbelströmung in Natur und Technik*. Verlag G. Braun GmbH, 1979.

[93] I. H. Manssour, S. S. Furuie, L. P. Nedel, and C. M. Freitas. A framework to interact and visualize with multimodal medical images. In *Volume Graphics*, pages 385–398, 2001.

[94] K. Matkovic, D. Gracanin, Z..Konyha, and H. Hauser:. Color lines view: An approach to visualization of families of function graphs. In *Proceedings of 11th International Conference Information Visualisation*, 2007.

[95] D. J. Mavriplis. Revisiting the least- squares procedure for gradient reconstruction on unstructured meshes. In *Proc. of the 16th AIAA Computational Fluid Dynamics Conference*, 2003.

[96] N. Max. Progress in scientific visualization. *The Visual Computer*, 21(12):979–984, 2005.

[97] N. Max, R. Crawfis, and D. Williams. Visualization for climate modeling. *IEEE Comput. Graph. Appl.*, 13(4):34–40, 1993.

[98] D. Merhof, M. Sonntag, F. Enders, C. Nimsky, P. Hastreiter, and G. Greiner. Hybrid visualization for white matter tracts using triangle strips and point sprites. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1181–1188, 2006.

[99] A. Meyers, P. Schieße, and O. T. Bruhns. Some comments on objective rates of symmetric Eulerian tensors with application to eulerian strain. *Acta Mechanica*, Volume 139(1):91–103, 1998.

[100] H. Miura and S. Kida. Identification of tubular vortices in turbulence. *Journal of the Physical Society of Japan*, 66:1331–1334, 1997.

[101] P. Moin and K. Mahesh. Direct numerical simulation: A tool in turbulence research. *Annual Review of Fluid Mechanics*, 30:539–578, 1998.

[102] Tamara Munzner, Chris Johnson, Robert Moorhead, Hanspeter Pfister, Penny Rheingans, and Terry S. Yoo. Nih-nsf visualization research challenges report summary. *IEEE Comput. Graph. Appl.*, 26(2):20–24, 2006.

[103] S. Muraki, T. Nakai, and Y. Kita. Basic research for coloring multichannel mri data. *Proceedings IEEE Visualization 2000*, 00:33, 2000.

[104] Z. Nagy, J. Schneider, and R. Westermann. Interactive volume illustration. In *Vision, Modeling and Visualization 2002*, November 2002.

[105] NASA. Spiral galaxy ngc 4414. `http://www.nasa.gov/multimedia/imagegallery`.

[106] J. Schmaltz (NASA). Hurricane season 2005: Katrina. `http://www.nasa.gov/vision/earth/lookingatearth/h2005_katrina.html`.

[107] N. Neophytou and K. Mueller. Space-time points: 4d splatting on efficient grids. In *VVS '02: Proc. of the 2002 IEEE symposium on Volume visualization and graphics*, pages 97–106, 2002.

[108] H. J. Noordmans, H. T. M. van der Voort, and A. W. M. Smeulders. Spectral volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 6(3):196–207, 2000.

[109] National Oceanic and Atmospheric Administration. The first tornado captured by the nssl doppler radar and nssl chase personnel. `http://www.photolib.noaa.gov/`.

[110] North Carolina School of Science and Mathematics. Solar flare. `http://www.dlt.ncssm.edu/resources/images/SolarFlare.jpg`.

[111] A. Okubo. Horizontal dispersion of floatable trajectories in the vincinity of velocity singularities such as convergencies. *Deep Sea. Res.*, 17:445–454, 1970.

[112] R. Peikert and F. Sadlo. Flow topology beyond skeletons: Visualization of features in recirculating flow. In *Proceedings of TopoInVis 2007: Topology-Based Methods in Visualization*, 2007. (to appear at Springer).

[113] Ronald Peikert and Martin Roth. The 'parallel vectors' operator: a vector field visualization primitive. In *Proceedings IEEE Visualization 1999*, pages 263–270, 1999.

[114] A. Perry and M.S. Chong. A description of eddying motions and flow patterns using critical point concepts. *Annual Review of Fluid Mechanics*, 19:125–155, 1987.

[115] Frits H. Post, Benjamin Vrolijk, Helwig Hauser, Robert S. Laramee, and Helmut Doleisch. The state of the art in flow visualisation: Feature extraction and tracking. *Computer Graphics Forum*, 22(4):775–792, 2003.

[116] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, February 2002.

[117] F. Reinders, F. H. Post, and H. J. W. Spoelder. Visualization of time-dependent data with feature tracking and event detection. *The Visual Computer*, 17(1):55–71, 2001.

[118] F. Reinders, H. Spoelder, and F. Post. Experiments on the accuracy of feature extraction. In *Proc. of the 9th EG Workshop on Visualization in Scientific Computing '98*, pages 49–58, 1998.

[119] J. C. Roberts. Multiple-View and Multiform Visualization. In *Visual Data Exploration and Analysis VII, Proc. of SPIE*, volume 3960, pages 176–185, 2000.

[120] Fabio Roli and Josef Kittler. Fusion of multiple classifiers. *Information Fusion*, 3(4):243, 2002.

[121] F. Rößler, E. Tejada, T. Fangmeier, T. Ertl, and M. Knauff. Gpu-based multi-volume rendering for the visualization of functional brain images. In *SimVis*, pages 305–318, 2006.

[122] M. Roth and R. Peikert. A higher-order method for finding vortex core lines. In *Proceedings IEEE Visualization 1998*, pages 143–150, 1998.

[123] Martin Roth and Ronald Peikert. Flow visualization for turbomachinery design. In *Proceedings IEEE Visualization 1996*, pages 381–384, 1996.

[124] I.A. Sadarjoen and F.H. Post. Geometric methods for vortex extraction. In *Proceedings of the 1th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization (VisSym 1999)*, pages 53–62, 1999.

[125] F. Sadlo and R. Peikert. Efficient Visualization of Lagrangian Coherent Structures by Filtered AMR Ridge Extraction. *Transactions on Visualization and Computer Graphics*, 13(6):1456–1463, 2007.

[126] F. Sadlo and R. Peikert. Visualizing lagrangian coherent structures: A comparison to vector field topology. In *Proceedings of TopoInVis 2007: Topology-Based Methods in Visualization*, 2007. (to appear at Springer).

[127] J. Sahner, T. Weinkauf, N. Teuber, and H.-C. Hege. Vortex and strain skeletons in Eulerian and Lagrangian frames. *IEEE Transactions on Visualization and Computer Graphics*, 13(5)(5):980–990, September - October 2007.

[128] T. Salzbrunn, C. Garth, G. Scheuermann, and Joerg Meyer. Pathline predicates and unsteady flow structures. *The Visual Computer*, (accepted for publication), 2007.

[129] N. Sauber, H. Theisel, and H.-P. Seidel. Multifield-graphs: An approach to visualizing correlations in multifield scalar data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):917–924, 2006.

[130] A. Schirotzek. High-temperature superfluidity. `http://cua.mit.edu/ketterle_group/experimental_setup/BEC_I/image_gallery.html`.

[131] S. Schmidt, O. Schögl, R. Kirchberger, H. Doleisch, P. Muigg, H. Hauser, M. Grabner, A. Bornik, and D. Schmalstieg. Novel visualisation and interaction techniques for gaining insight into fluid dynamics in internal combustion engines. In *Proc. of Nafems World Congress*, 2005.

[132] W. Schroeder, C. Volpe, and W. Lorensen. The stream polygon: A technique for 3D vector field visualization. In *Proceedings IEEE Visualization 1991*, pages 126–132, 1991.

[133] H. H. Shenas and V. Interrante. Compositing color with texture for multi-variate visualization. In *GRAPHITE '05: Proc. of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 443–446, 2005.

[134] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *VL '96: Proc. of the 1996 IEEE Symposium on Visual Languages*, page 336, 1996.

[135] A. Sigfridsson, T. Ebbers, E. Heiberg, and L. Wigström. Tensor field visualisation using adaptive filtering of noise fields combined with glyph rendering. In *Proceedings IEEE Visualization 2002*, pages 371–378, 2002.

[136] J. Sobel. Scivl: A descriptive language for 2D multivariate scientific visualization synthesis. Technical Report CS-03-24, Computer Science Department, Brown University, 2003.

[137] S. Stegmaier, U. Rist, and T. Ertl. Opening the Can of Worms: An Exploration Tool for Vortical Flows. In *Proceedings IEEE Visualization 2005*, pages 463–470, 2005.

[138] M. Steinberg, K. Mueller, and R. Kelly. Engine room: A comparitive environment for volume rendering engines. In *Computer Graphics International 2005 Poster Proc.*, 2005.

[139] A. Stompel, E. B. Lum, and K.-L. Ma. Visualization of multidimensional, multivariate volume data using hardware-accelerated non-photorealistic rendering techniques. In *PG '02: Proc. of the 10th Pacific Conference on Computer Graphics and Applications*, page 394, 2002.

[140] D. Sujudi and R. Haimes. Identification of swirling flow in 3D vector fields. Technical Report AIAA-95-1715, American Institute of Aeronautics and Astronautics, 1995.

[141] N. Svakhine and D. S. Ebert. Interactive volume illustration and feature halos. In *PG '03: Proc. of the 11th Pacific Conference on Computer Graphics and Applications*, page 347, 2003.

[142] N. A. Svakhine, Y. Jang, D. S. Ebert, and K. Gaither. Illustration and Photography Inspired Visualization of Flows and Volumes. In *Proceedings IEEE Visualization 2005*, pages 687–694, 2005.

[143] R. M. Taylor. Visualizing multiple fields on the same surface. *IEEE Computer Graphics and Applications*, 22(3):6–10, 2002.

[144] H. Theisel and H.-P. Seidel. Feature flow fields. In *Proceedings of the 5th Joint IEEE VGTC - EUROGRAPHICS Symposium on Visualization (VisSym 2003)*, 2003.

[145] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. Stream line and path line oriented topology for 2D time-dependent vector fields. In *Proceedings IEEE Visualization 2004*, pages 321–328, 2004.

[146] J. J. Thomas and K. A. Cook. A visual analytics agenda. *IEEE Comput. Graph. Appl.*, 26(1):10–13, 2006.

[147] L. Treinish. Multi-resolution visualization techniques for nested weather models. In *Proceedings IEEE Visualization 2000*, pages 513–516, 2000.

[148] Lloyd A. Treinish. Task-specific visualization design. *IEEE Computer Graphics and Applications*, 19(5):72–77, 1999.

[149] X. Tricoche, C. Garth, G. Kindlmann, E. Deines, G. Scheuermann, M. Ruetten, and C. Hansen. Visualization of intricate flow structures for vortex breakdown analysis. In *Proceedings IEEE Visualization 2004*, pages 187–194, 2004.

[150] C. A. Truesdell. *The Kinematics of Vorticity.* Indiana University Science Serie 19, 1954.

[151] F.-Y. Tzeng, E. B. Lum, and K.-L. Ma. An intelligent system approach to higher-dimensional classification of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):273–284, 2005.

[152] J.R. van Dellen. Water in motion. `http://www.flickr.com/photos/van_dellen/1258995364/`.

[153] A. Vilanova, S. Zhang, G. Kindlmann, and D. H. Laidlaw. *Visualization and Processing of Tensor Fields. Proc. of the Dagstuhl Workshop*, chapter An Introduction to Visualization of Diffusion Tensor Imaging and Its Applications. Springer, 2005.

[154] I. Viola, A. Kanitsar, and Meister E. Gröller. Importance-driven feature enhancement in volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):408–418, 2005.

[155] M. O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3/4):194–210, 2002.

[156] R. Wegenkittl, Meister E. Gröller, and W. Purgathofer. Visualizing the dynamical behavior of wonderland. *IEEE Comput. Graph. Appl.*, 17(6):71–79, 1997.

[157] T. Weinkauf, J. Sahner, H. Theisel, and H.-C. Hege. Cores of swirling particle motion in unsteady flows. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1759–1766, 2007.

[158] D. Weiskopf, K. Engel, and T. Ertl. Interactive clipping techniques for texture-based volume visualization and volume shading. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):298–312, 2003.

[159] D. Weiskopf, F. Schramm, G. Erlebacher, and T. Ertl. Particle and Texture Based Spatiotemporal Visualization of Time-Dependent Vector Fields. In *Proc. of IEEE Visualization '05*, 2005.

[160] J. Weiss. The dynamics of enstrophy transfer in two-dimensional hydrodynamics. *Physica D Nonlinear Phenomena*, 48:273–294, 1991.

[161] A. Wenger, D. F. Keefe, and S. Zhang. Interactive volume rendering of thin thread structures within multivalued scientific data sets. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):664–672, 2004.

[162] C. F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis. Processing and visualization for diffusion tensor MRI. *Medical Image Analysis*, 6(2):93–108, 2002.

[163] J. van Wijk. Flow visualization with surface particles. *IEEE Comput. Graph. Appl.*, 13(4):18–24, 1993.

[164] J. van Wijk. Image Based Flow Visualization for Curved Surfaces. In *Proceedings IEEE Visualization 2003*, pages 123–130, 2003.

[165] J. van Wijk and R. van Liere. Hyperslice: visualization of scalar functions of many variables. In *Proceedings IEEE Visualization 1993*, pages 119–125, 1993.

[166] C. M. Wittenbrink, A. T. Pang, and S. K. Lodha. Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):266–279, 1996.

[167] P. C. Wong. Guest editor's introduction: Visual data mining. *IEEE Computer Graphics and Applications*, 19(5):20–21, 1999.

[168] P. C. Wong, H. Foote, D. L. Kao, R. Leung, and J. Thomas. Multivariate visualization with data fusion. *Information Visualization*, 1(3/4):182–193, 2002.

[169] J. Woodring and H.-W. Shen. Multi-variate, time varying, and comparative visualization with contextual cues. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):909–916, 2006.

[170] J. Woodring, C. Wang, and H.-W. Shen. High dimensional direct rendering of time-varying volumetric data. In *Proceedings IEEE Visualization 2003*, page 55, Washington, DC, USA, 2003. IEEE Computer Society.

[171] B. Wünsche. The visualization of 3d stress and strain tensor fields. In *Proc. of the 3rd New Zealand Computer Science Research Student Conference*, pages 109–116., 1999.

[172] R. Yagel, D. S. Ebert, J. N. Scott, and Y. Kurzion. Grouping volume renderers for enhanced visualization in computational fluid dynamics. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):117–132, 1995.

[173] E. Zhang, J. Hays, and G. Turk. Interactive tensor field design and visualization on surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):94–107, 2007.

[174] S. Zhang, G. Kindlmann, and D. H. Laidlaw. Diffusion tensor MRI visualization. In *Visualization Handbook*. Academic Press, 2004.

[175] X. Zheng and A. Pang. Interaction of light and tensor fields. In *Proceedings of the 5th Joint IEEE VGTC - EUROGRAPHICS Symposium on Visualization (VisSym 2003)*, pages 157–166, 2003.

[176] J. Zhou, S. Balachandar, and R.J. Adrian. Mechanisms for generating coherent packet of hairpin vortices in near-wall turbulence. *Journal of Fluid Mechanics*, 387:353–396, 1999.

[177] M. Zöckler, D. Stalling, and H.-C. Hege. Interactive visualization of 3D-vector fields using illuminated stream lines. In *Proceedings IEEE Visualization 1996*, pages 107–114., 1996.

[178] Homepage of Arsenal Research. See URL `http://www.arsenal.ac.at`.

[179] Homepage of the SimVis Visualization Framework. See URL `http://www.simvis.at`.